



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής  
Πρόγραμμα Μεταπτυχιακών Σπουδών  
«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Ηλεκτρονική εκπαιδευτική πλατφόρμα-E-School
Όνοματεπώνυμο Φοιτητή	Βασιλική Λαγού
Πατρώνυμο	Παντελής
Αριθμός Μητρώου	ΜΠΣΠ/09044
Επιβλέπων	Μερία Βίρβου, Καθηγήτρια Πανεπιστημίου Πειραιώς



Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Όνομα Επώνυμο  
Βαθμίδα

Όνομα Επώνυμο  
Βαθμίδα

Όνομα Επώνυμο  
Βαθμίδα

## ΠΕΡΙΛΗΨΗ

Στις μέρες μας οι ραγδαίες εξελίξεις στο χώρο των νέων τεχνολογιών, της πληροφορικής και των τηλεπικοινωνιών υπαγορεύουν την υιοθέτηση νέων ευέλικτων λειτουργικών δομών που αναβαθμίζουν τα υπάρχοντα συστήματα εκπαίδευσης. Πολλοί εκπαιδευτικοί οργανισμοί αντιμετωπίζουν τις νέες εκπαιδευτικές προκλήσεις με τη δημιουργία προγραμμάτων εκπαίδευσης από απόσταση. Στο πιο βασικό επίπεδο, η εκπαίδευση από απόσταση υλοποιείται στις περιπτώσεις όπου ένας εκπαιδευτής και ο/οι εκπαιδευόμενοι βρίσκονται σε διαφορετικές τοποθεσίες και η τεχνολογία (δηλ. φωνή, video, δεδομένα, έντυπο υλικό, κτλ), συχνά σε συνδυασμό με επικοινωνία πρόσωπο-με-πρόσωπο, χρησιμοποιείται για να γεφυρώσει το διδακτικό κενό. Αυτού του τύπου τα προγράμματα είναι σε θέση να παράσχουν σε ενήλικες και παιδιά μια ευκαιρία για εκπαίδευση και επιμόρφωση, να απευθυνθούν σε άτομα με περιορισμούς χρόνου, με φυσικούς ή σωματικούς περιορισμούς και να αυξήσουν τις γνώσεις όλων ανεξαρτήτου ηλικίας.

Το πρόγραμμα e-School αποτελεί μια εφαρμογή ηλεκτρονικής εκπαίδευσης με αντικείμενο το μάθημα της Γεωγραφίας και Ιστορίας με δυνατότητα επέκτασης και σε άλλα μαθήματα. Ο καθηγητής έχει πλήρη έλεγχο της οργάνωσης του μαθήματος, και μπορεί να διαμορφώσει κατά βούληση την διδακτέα ύλη, χρησιμοποιώντας κείμενο, εικόνες, αλλά και πολυμέσα. Όταν έχει καταχωριστεί η διδακτέα ύλη, ο μαθητής μπορεί να χρησιμοποιήσει το πρόγραμμα για να παρακολουθήσει τα μαθήματα και να εξεταστεί σε αυτά. Το πρόγραμμα τηρεί στατιστικά στοιχεία για τον κάθε μαθητή, στα οποία έχει πρόσβαση ο καθηγητής.

## **ΚΕΦΑΛΑΙΟ 1**

### **ΕΙΣΑΓΩΓΗ**

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΑΙΑ

## 1.1 ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΤΕΧΝΟΛΟΓΙΑ ΛΟΓΙΣΜΙΚΟΥ

Είναι γεγονός ότι όλες οι αναπτυσσόμενες χώρες σήμερα βασίζονται σε πολύπλοκα υπολογιστικά συστήματα για την ανάπτυξη των προϊόντων τους. Το λογισμικό σε αυτά τα συστήματα αντιπροσωπεύει ένα ολοένα και αυξανόμενο ποσοστό του συνολικού κόστους του συστήματος. Ως εκ τούτου, η κατασκευή λογισμικού με ένα οικονομικά αποδοτικό τρόπο είναι απαραίτητη για τη λειτουργία εθνικών και διεθνών οικονομιών.

Η Τεχνολογία Λογισμικού λοιπόν έχει ως στόχο την οικονομικά αποδοτική ανάπτυξη συστημάτων λογισμικού. Το λογισμικό δεν περιορίζεται από υλικά, δεν κυβερνάται από φυσικούς νόμους ή από κατασκευαστικές διαδικασίες. Κατά κάποιο τρόπο, αυτό απλοποιεί την τεχνολογία λογισμικού, αφού δεν υπάρχουν φυσικά όρια για την ανάπτυξη λογισμικού. Αυτή όμως η έλλειψη φυσικών ορίων σημαίνει ότι το λογισμικό μπορεί εύκολα να γίνει εξαιρετικά πολύπλοκο και συνεπώς δύσκολο στην κατανόηση.

Η αρχική εμπειρία στην κατασκευή συστημάτων λογισμικού έδειξε ότι μία ανεπίσημη προσέγγιση για την ανάπτυξη λογισμικού δεν ήταν αρκετά καλή. Μεγάλες εργασίες χρειάζονταν χρόνια για να διεκπεραιωθούν, ενώ τα κόστη υπερέβαιναν τα αρχικά προβλεπόμενα. Η κρίση αυτή στην ανάπτυξη λογισμικού οδήγησε στην ανάπτυξη νέων τεχνικών, έτσι ώστε να ελεγχθεί η έμφυτη πολυπλοκότητα στα μεγάλα συστήματα λογισμικού. Με τον τρόπο αυτό, η εξέλιξη της τεχνολογίας λογισμικού έχει βελτιώσει κατά πολύ το ίδιο το λογισμικό. Τώρα πια έχουμε πολύ καλύτερη κατανόηση των δραστηριοτήτων που εμπλέκονται στην ανάπτυξη λογισμικού και έχουμε αναπτύξει αποτελεσματικές μεθόδους για τον προσδιορισμό, το σχεδιασμό και την εφαρμογή του λογισμικού.

Ας δούμε όμως στο σημείο αυτό κάποιες βασικές έννοιες που θα μας εισάγουν γενικά στο θέμα της Τεχνολογίας Λογισμικού, αλλά και ειδικότερα στην εφαρμογή που θα περιγράψουμε στη συνέχεια της εργασίας αυτής.

### Λογισμικό:

Ενώ πολλοί άνθρωποι ταυτίζουν την έννοια του λογισμικού με τα προγράμματα υπολογιστών, το λογισμικό δεν είναι μόνο τα προγράμματα αλλά και η σχετιζόμενη τεκμηρίωση και τα διαμορφωμένα δεδομένα που είναι απαραίτητα για να λειτουργούν τα προγράμματα σωστά. Ένα σύστημα λογισμικού συνήθως αποτελείται από έναν αριθμό διαφορετικών προγραμμάτων, αρχεία για την ενδοποίηση των προγραμμάτων αυτών, τεκμηρίωση του συστήματος που περιγράφει τη δομή του συστήματος και τεκμηρίωση του χρήστη που εξηγεί πως χρησιμοποιείται το σύστημα και, για προϊόντα λογισμικού, ιστοσελίδες για τους χρήστες που επιθυμούν να λαμβάνουν νέες πληροφορίες για το προϊόν.

Το λογισμικό χωρίζεται σε δύο επιμέρους κατηγορίες. Στην πρώτη ανήκουν τα αυτόνομα συστήματα που κατασκευάζονται από έναν οργανισμό ανάπτυξης λογισμικού και είναι διαθέσιμα στην αγορά για όποιον πελάτη έχει την επιθυμία και την ικανότητα να τα αγοράσει. Στην δεύτερη ανήκουν τα συστήματα, η κατασκευή των οποίων ανατίθεται σε ένα εργολήπτη από ένα συγκεκριμένο πελάτη. Το λογισμικό διαμορφώνεται ακριβώς με βάση τις απαιτήσεις του πελάτη. Η κυριότερη διαφορά μεταξύ των δύο τύπων λογισμικού είναι το γεγονός ότι στα εμπορικά προϊόντα ο οργανισμός που κατασκευάζει το λογισμικό ελέγχει τις προδιαγραφές του, ενώ στη δεύτερη κατηγορία λογισμικού οι προδιαγραφές του καθορίζονται και ελέγχονται από τον ίδιο τον οργανισμό που αγοράζει το λογισμικό.

Για την ανάπτυξη του λογισμικού υπάρχουν οι προγραμματιστές, οι οποίοι ασχολούνται με τις λεπτομέρειες της υλοποίησης, της πακετοποίησης και της αλλαγής αλγορίθμων και δομών δεδομένων που έχουν γραφτεί σε συγκεκριμένη γλώσσα προγραμματισμού.

### Τεχνολογία Λογισμικού:

Η Τεχνολογία Λογισμικού είναι μία διαδικασία που ασχολείται με όλους τους τομείς της κατασκευής λογισμικού από τα πρώτα στάδια του προσδιορισμού του συστήματος μέχρι τη συντήρηση του συστήματος αφού από έχει μπει σε λειτουργία.

Αυτός που εμπλέκεται στην διαδικασία της Τεχνολογίας Λογισμικού είναι ο μηχανικός λογισμικού, ο οποίος, επιπρόσθετα της δουλειάς του προγραμματιστή, ασχολείται με θέματα ανάλυσης συστημάτων, σχεδιασμού, επαλήθευσης, τεκμηρίωσης, συντήρησης λογισμικού και διαχείρισης έργου. Οι μηχανικοί λογισμικού είναι αυτοί που, με λίγα λόγια, κάνουν τα πράγματα να δουλεύουν. Εφαρμόζουν θεωρίες, μεθόδους και εργαλεία, όπου αυτά είναι απαραίτητα, αλλά τα χρησιμοποιούν επιλεκτικά και πάντα προσπαθούν να βρουν λύσεις σε προβλήματα, ακόμα και όταν δεν υπάρχουν εφαρμόσιμες θεωρίες και μέθοδοι για να υποστηρίξουν την προσπάθειά τους. Οι μηχανικοί, επίσης, αναγνωρίζουν το γεγονός ότι πρέπει να δουλεύουν μέσα σε οργανωτικά και οικονομικά όρια, κι έτσι αναζητούν λύσεις μέσα στα όρια αυτά.

Η Τεχνολογία Λογισμικού δεν ασχολείται μόνο με τις τεχνικές διαδικασίες της ανάπτυξης του λογισμικού, αλλά και με δραστηριότητες όπως η διαχείριση έργου και με την ανάπτυξη εργαλείων, μεθόδων και θεωριών για την υποστήριξη της κατασκευής του λογισμικού. Γενικά, οι μηχανικοί λογισμικού υιοθετούν μία συστηματική και οργανωμένη προσέγγιση για τη δουλειά τους, αφού αυτός είναι συνήθως ο πιο αποτελεσματικός τρόπος να παράγουν υψηλής ποιότητας λογισμικό.

Για την ανάπτυξη του απαιτούμενου λογισμικού πρέπει να ακολουθήσει ο κάθε μηχανικός λογισμικού κάποια βασικά στάδια για να φτάσει στο τελικό επιθυμητό προϊόν. Τα γενικά αυτά στάδια φαίνονται παρακάτω:

- Καθορισμός της λειτουργικότητας του λογισμικού και των προδιαγραφών για τη λειτουργία του.
- Κατασκευή του λογισμικού που ανταποκρίνεται στις δοθείσες προδιαγραφές. Εδώ περιλαμβάνεται η ανάπτυξη της λογικής που πρέπει να χρησιμοποιηθεί και η υλοποίηση των προγραμμάτων.
- Έλεγχος του λογισμικού. Στον έλεγχο εντάσσεται η επαλήθευση και η επικύρωση, το αν δηλαδή το προϊόν έχει αναπτυχθεί σωστά και το αν έχει αναπτυχθεί το σωστό προϊόν αντίστοιχα.
- Τεκμηρίωση του λογισμικού που έχει αναπτυχθεί. Όταν μιλάμε για τεκμηρίωση εννοούμε την εξήγηση των χαρακτηριστικών ενός εγγράφου. Υπάρχουν και δύο άλλοι τύποι τεκμηρίωσης, η εσωτερική και η εξωτερική τεκμηρίωση. Η εσωτερική τεκμηρίωση του πηγαίου κώδικα περιγράφει τα χαρακτηριστικά του κώδικα, ενώ η εξωτερική τεκμηρίωση εξηγεί τα χαρακτηριστικά των εγγράφων που συνοδεύουν τον κώδικα.
- Συντήρηση των προγραμμάτων, έτσι ώστε να ανταποκρίνονται στις μεταβαλλόμενες ανάγκες των πελατών.

Θα πρέπει να σημειωθεί ότι, αντίθετα από ότι θα περίμενε κανείς, το μεγαλύτερο χρονικό και οικονομικό κόστος κατά τον όλο κύκλο ζωής της ανάπτυξης του λογισμικού το απαιτεί ο έλεγχος και η συντήρηση του λογισμικού, ενώ πολύ μικρότερο ποσοστό του κόστους καταλαμβάνει η εν γένει ανάπτυξη του συστήματος.

### **Computer-Aided Software Engineering:**

Ο όρος Computer-Aided Software Engineering (CASE) αντιπροσωπεύει ένα πλήθος διαφορετικών τύπων προγραμμάτων που χρησιμοποιούνται για να υποστηρίξουν τις δραστηριότητες που εμπλέκονται στη διαδικασία ανάπτυξης του λογισμικού, όπως η ανάλυση απαιτήσεων, η μοντελοποίηση του συστήματος, η μεταγλώττιση και ο έλεγχος. Υπάρχουν δύο ειδών εργαλεία CASE, τα upper-CASE tools και τα lower-CASE tools. Τα πρώτα υποστηρίζουν τις αρχικές φάσεις της ανάπτυξης του λογισμικού, όπως την ανάλυση και το σχεδιασμό, ενώ τα δεύτερα είναι σχεδιασμένα για να υποστηρίξουν την υλοποίηση και τον έλεγχο.

Το εργαλείο που χρησιμοποιήθηκε στην εργασία αυτή είναι το Rational Rose, έτσι ώστε να εξηγηθεί και να τεκμηριωθεί το λογισμικό που θα κατασκευάσουμε για τη διαχείριση της εφαρμογής «e-School». Οι λεπτομέρειες της χρήσης του συγκεκριμένου CASE-tool θα παρατίθενται στην πορεία της εφαρμογής των μεθόδων που θα χρησιμοποιήσουμε.

Αφού όμως γίνει η υλοποίηση των προγραμμάτων θα πρέπει να ελέγξουμε αν το λογισμικό μας ανταποκρίνεται επαρκώς σε συγκεκριμένα χαρακτηριστικά ποιότητας. Ας δούμε ποια είναι τα χαρακτηριστικά αυτά για να κατευθύνουμε κάθε στάδιο της ανάπτυξης του λογισμικού προς αυτούς τους στόχους:

- **Χρησιμότητα:** Το λογισμικό θα πρέπει να ικανοποιεί τις ακριβείς απαιτήσεις του χρήστη.
- **Μεταφερισιμότητα:** Το λογισμικό θα πρέπει να έχει τη δυνατότητα να μεταφερθεί εύκολα μεταξύ διαφορετικών υπολογιστών ή περιβαλλόντων.
- **Αξιοπιστία:** Το λογισμικό θα πρέπει να έχει τη δυνατότητα να κάνει μια λειτουργία που απαιτείται κάτω από ορισμένες προϋποθέσεις σε καθορισμένο διάστημα χρόνου.
- **Αποδοτικότητα:** Ορίζεται ως ο βαθμός στον οποίο το λογισμικό μπορεί να εκτελέσει τις αποσκοπούμενες λειτουργίες καταναλώνοντας όσο το δυνατόν λιγότερους πόρους του συστήματος.
- **Ακρίβεια:** Ως ακρίβεια μπορεί να οριστεί είτε η ποσοτική εκτίμηση της μη ύπαρξης λαθών, είτε το ποσοτικό μέτρο του μεγέθους ενός λάθους. Λάθος αποτελεί μία ασυμβατότητα μεταξύ μίας τιμής ή μίας συνθήκης όπως έχει προκύψει από τον υπολογιστή και της πραγματικής, καθορισμένης ή θεωρητικά σωστής τιμής ή συνθήκης.
- **Ευρωστία:** Κατά πόσον το λογισμικό μπορεί να λειτουργεί σωστά παρά την εισαγωγή μη έγκυρων δεδομένων.
- **Ορθότητα:** Ο βαθμός στον οποίο το λογισμικό είναι απαλλαγμένο από σχεδιαστικές ατέλειες και ατέλειες κωδικοποίησης. Επίσης, ορθότητα ονομάζουμε το βαθμό στον οποίο το λογισμικό επιτυγχάνει τις καθορισμένες απαιτήσεις και ικανοποιεί τις προσδοκίες του χρήστη.
- **Επαναχρησιμοποιησιμότητα:** Τα διάφορα μέρη του λογισμικού πρέπει να σχεδιάζονται έτσι ώστε να μπορούν να χρησιμοποιούνται σε πολλά προγράμματα. Μέρη του λογισμικού που ικανοποιούν την αρχή της επαναχρησιμοποιησιμότητας είναι διάφοροι αλγόριθμοι (ρουτίνες και βιβλιοθήκες) και δομές δεδομένων που χρησιμοποιούμε στα προγράμματα.

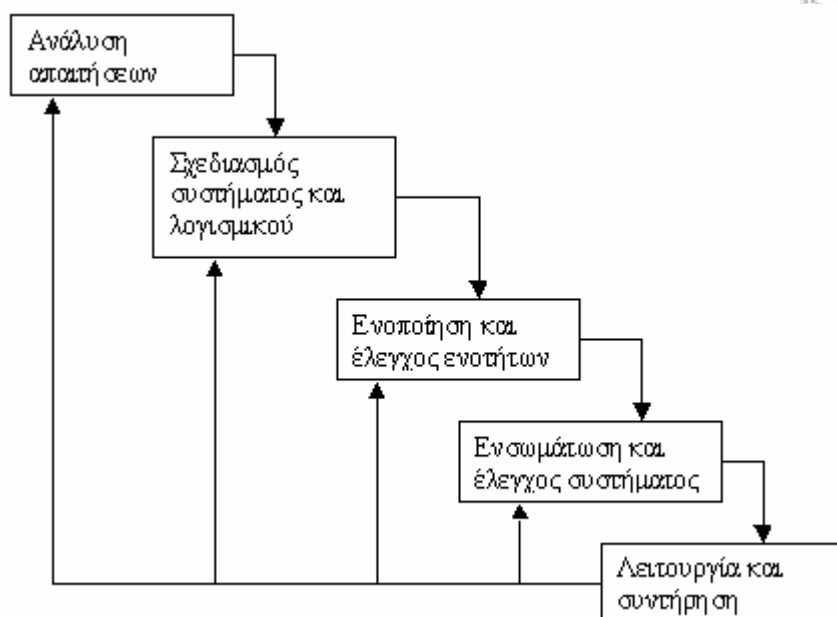
### **Κύκλος Ζωής:**

Υπάρχουν διάφορα μοντέλα για την περιγραφή του κύκλου ζωής της ανάπτυξης του λογισμικού, όπως ο Ανιχνευτικός Προγραμματισμός, η Ανάπτυξη Πρωτοτύπου και η Συναρμολόγηση Συστήματος από Επαναχρησιμοποιούμενα Κομμάτια. Το μοντέλο για τον κύκλο ζωής της ανάπτυξης του λογισμικού που θα χρησιμοποιήσουμε στην εφαρμογή που θα αναπτύξουμε είναι το μοντέλο του Καταράκτη. Η διαγραμματική απεικόνιση του μοντέλου φαίνεται στην επόμενη σελίδα.

Ας δούμε όμως αναλυτικά τις εργασίες που γίνονται σε κάθε στάδιο του κύκλου ζωής:

1. **Ανάλυση απαιτήσεων:** Μέσω συμβουλευτικής διαδικασίας με τους χρήστες του συστήματος προσδιορίζονται οι υπηρεσίες που θα παρέχει το σύστημα, οι περιορισμοί και οι στόχοι του.
2. **Σχεδιασμός συστήματος και λογισμικού:** Η διαδικασία σχεδιασμού του συστήματος χωρίζει τις απαιτήσεις σε απαιτήσεις υλικού και απαιτήσεις λογισμικού. Εγκαθιδρύει μία γενική αρχιτεκτονική συστήματος. Ο σχεδιασμός του λογισμικού περιλαμβάνει την αναγνώριση και την περιγραφή των σημαντικών μερών του συστήματος λογισμικού και των σχέσεων μεταξύ τους.
3. **Υλοποίηση και έλεγχος ενοτήτων:** Κατά τη διάρκεια αυτού του σταδίου, ο σχεδιασμός του λογισμικού γίνεται αντιληπτός ως ένα σύνολο προγραμμάτων ή ενοτήτων προγραμμάτων. Ο έλεγχος των ενοτήτων περιλαμβάνει την επαλήθευση του γεγονότος ότι κάθε ενότητα εκπληρώνει τις προδιαγραφές της.
4. **Ενσωμάτωση και έλεγχος συστήματος:** Οι ανεξάρτητες ενότητες των προγραμμάτων ή τα ίδια τα διαφορετικά προγράμματα συγχωνεύονται και ελέγχονται σαν ένα ολοκληρωμένο σύστημα για να διασφαλιστεί ότι έχουν εκπληρωθεί οι απαιτήσεις του λογισμικού. Μετά τον έλεγχο, το σύστημα λογισμικού παραδίδεται στον πελάτη.
5. **Λειτουργία και συντήρηση:** Κανονικά, αλλά όχι απαραίτητα, αυτή είναι η πιο χρονοβόρα φάση του κύκλου ζωής. Το σύστημα εγκαθίσταται και μπαίνει σε χρήση στην πράξη. Η συντήρηση περιλαμβάνει τη διόρθωση λαθών που δεν έγιναν αντιληπτά σε ενωρίτερα στάδια του κύκλου ζωής, βελτιώνοντας την εφαρμογή των

διαφόρων ενοτήτων του συστήματος και τις υπηρεσίες που προσφέρει το σύστημα, αφού ανακαλύπτονται νέες απαιτήσεις.



**Σχήμα 1:**

Κύκλος ζωής λογισμικού

Σύμφωνα με τις βασικές αρχές, το αποτέλεσμα κάθε φάσης είναι ένα ή περισσότερα έγγραφα που πρέπει να περάσουν τη διαδικασία της έγκρισης. Η επόμενη φάση δε θα μπορούσε να ξεκινήσει στην περίπτωση αυτή αν η προηγούμενη δεν είχε ολοκληρωθεί. Στην πράξη αυτά τα στάδια υπερκαλύπτονται και ανταλλάσσουν πληροφορίες. Κατά τη διάρκεια του σχεδιασμού ανακύπτουν προβλήματα σε σχέση με τις απαιτήσεις, κατά τη διάρκεια της κατασκευής του πηγαίου κώδικα βρίσκονται προβλήματα, κτλ. Η ανάπτυξη του λογισμικού δεν είναι ένα απλό γραμμικό μοντέλο αλλά περιλαμβάνει μία ακολουθία επαναλήψεων των δραστηριοτήτων εξέλιξης.

Λόγω του κόστους της κατασκευής και της έγκρισης εγγράφων, οι επαναλήψεις έχουν μεγάλο κόστος και απαιτούν σημαντική επαναληπτική εργασία. Ως εκ τούτου, μετά από ένα μικρό αριθμό επαναλήψεων, είναι φυσιολογικό να «παγώνουν» τμήματα της διαδικασίας, όπως ο καθορισμός προδιαγραφών, και να συνεχίζεται η διαδικασία στα επόμενα στάδια. Πιθανά προβλήματα αφήνονται για μεταγενέστερη επίλυση ή αγνοούνται. Το πρόωρο πάγωμα των απαιτήσεων μπορεί να σημαίνει ότι το σύστημα δεν θα κάνει αυτό που θέλει ο χρήστης. Μπορεί επίσης να οδηγήσει σε άσχημα δομημένα συστήματα, μιας και τα σχεδιαστικά προβλήματα παρακάμπτονται με κόλπα στην εφαρμογή.

Κατά τη διάρκεια της τελευταίας φάσης του κύκλου ζωής το λογισμικό μπαίνει σε λειτουργία. Λάθη και παραλείψεις στις αυθεντικές απαιτήσεις του λογισμικού ανακαλύπτονται. Προγραμματιστικά και σχεδιαστικά λάθη ενοποιούνται και αναγνωρίζεται η ανάγκη για νέα λειτουργικότητα. Το σύστημα λοιπόν πρέπει να εξελιχθεί για να παραμείνει χρήσιμο. Το να γίνουν όμως οι απαραίτητες αλλαγές (συντήρηση του λογισμικού) μπορεί να σημαίνει την επανάληψη κάποιων ή όλων των προγενέστερων φάσεων της διαδικασίας.

Επειδή το Μοντέλο του Καταρράκτη ανταποκρίνεται δύσκολα στις αλλαγές των απαιτήσεων του πελάτη (αφού χωρίζει το έργο σε σχετικά δύσκαμπτα στάδια) θα πρέπει στην αρχή της διαδικασίας να καθοριστούν σαφώς και να κατανοηθούν πλήρως οι απαιτήσεις. Έτσι τα λάθη και οι παραλείψεις θα τείνουν να ελαχιστοποιηθούν.



Το εγχειρίδιο αυτό θα οργανωθεί με βάση τις επιμέρους δραστηριότητες κάθε φάσης του κύκλου ζωής για το συγκεκριμένο σύστημα διαχείρισης των μαθημάτων Γεωγραφίας και Ιστορίας για παιδιά πρώτης τάξης γυμνασίου. Για το λόγο αυτό κάθε κεφάλαιο θα είναι αφιερωμένο στις διάφορες εργασίες και στα έγγραφα που θα συνοδεύουν κάθε εργασία. Έτσι, τα κεφάλαια θα έχουν την εξής οργάνωση:

- Κεφάλαιο 1: Εισαγωγή
- Κεφάλαιο 2: Παρούσα κατάσταση
- Κεφάλαιο 3: Ανάλυση και ορισμός απαιτήσεων
- Κεφάλαιο 4: Σχεδιασμός Λογισμικού
- Κεφάλαιο 5: Διεπαφή με Χρήστες-Παρουσίαση Εφαρμογής

Σε κάθε κεφάλαιο θα παρουσιάζονται αρχικά οι σχετικές με το αντίστοιχο θέμα μεθοδολογίες και στη συνέχεια θα γίνεται πλήρης ανάλυση της εφαρμογής των μεθοδολογιών αυτών στην πράξη κατά τη διαδικασία της κατασκευής του συστήματος διαχείρισης των εργασιών των μαθημάτων της εφαρμογής e-School.

## **1.2 ΕΙΣΑΓΩΓΗ ΚΑΙ ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ «e-School»**

Το πρώτο βήμα στη δημιουργία της εφαρμογής ήταν καταρχήν να γίνει συλλογή πληροφοριών με τη μέθοδο των συνεντεύξεων. Έγινε επικοινωνία με όσο το δυνατό περισσότερα άτομα, που εργάζονται στο χώρο της δευτεροβάθμιας εκπαίδευσης και συγκεκριμένα με δασκάλους και καθηγητές σε δημόσια σχολεία, ώστε να αποκτηθεί σφαιρική εικόνα για τον επικείμενο σχεδιασμό της εργασίας. Επίσης, έγινε μια μερική επικοινωνία με μαθητές για τη συλλογή πληροφοριών για το πώς αυτοί βλέπουν το μάθημα και πως θα μπορούσε να γίνει πιο εύκολο.

Το εύρος της διδασκαλίας των μαθημάτων τόσο για τους μαθητές όσο και για τους καθηγητές χωρίζεται σε δύο βασικά μέρη:

- Θεωρία (παράδοση μαθήματος)
- Ερωτήσεις προς εξέταση

Επίσης έγινε και μια έρευνα σχετικά με τα μαθήματα της Γεωγραφίας και Ιστορίας στα σχολικά βιβλία που μελετήθηκαν με τη βοήθεια των δασκάλων. Από τη μελέτη διαπιστώθηκε πως υπήρχε ευκολία στο αντικείμενο μιας και ήταν γνωστό στα παιδιά και πως αυτό που έλειπε ήταν μία ενσωμάτωση στοιχείων πολυμέσων για τη καλύτερη εμπέδωση της πληροφορίας. Συμπερασματικά, η παρουσία ενός εκπαιδευτικού προγράμματος το οποίο έχει την ικανότητα να εκμεταλλεύεται τη σημερινή τεχνολογία των multimedia με όμορφα και καλά γραφικά ήταν μια ιδανική λύση, ενώ σίγουρα θα ωθήσει το μαθητή περισσότερο στη γνώση. Επειδή οι υπολογιστές οι οποίοι είναι εγκατεστημένοι στα σχολεία είναι μικρής ισχύος οι πρώτοι περιορισμοί που τέθηκαν ήταν ότι το υπό κατασκευή πρόγραμμα δεν πρέπει κατά τη λειτουργία του να καταναλώνει πολλούς πόρους, αλλά όσο γίνεται πιο λίγους. Οι λόγοι ήταν για να μη υπάρχουν προβλήματα στο πρόγραμμα το οποίο θα οδηγούσε στη γρήγορη εγκατάλειψη από τα παιδιά.

Ένας από τους πρώτους και βασικότερους στόχους για τον σχεδιασμό του προγράμματος ήταν να είναι μία εφαρμογή απλή και κυρίως φιλική προς το χρήστη. Για αυτό τον λόγο έπρεπε να δημιουργηθεί ένα πρόγραμμα σε περιβάλλον που είναι διαδεδομένο και που είναι εύκολο στην εκμάθησή. Στην ανάγκη αυτή πιο θα ήταν καταλληλότερο από το περιβάλλον των Windows. Έτσι λοιπόν αποφασίστηκε ότι το σύστημα θα αλληλεπιδρά με το χρήστη με οθόνες – παράθυρα. Άλλοι στόχοι που θέσαμε ήταν η μεταφερσιμότητα, η αξιοπιστία, η αποδοτικότητα, η ακρίβεια, η συνέπεια, η ευρωστία και η ορθότητα κάτι που θα αναλυθούν πιο κάτω κατά τη διάρκεια της υλοποίησης.

## **ΚΕΦΑΛΑΙΟ 2**

### **ΠΑΡΟΥΣΑ ΚΑΤΑΣΤΑΣΗ**

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΑΙΑ

Σήμερα, στο κατώφλι του 21ου αιώνα, ένα από τα σημαντικότερα φαινόμενα της κοινωνικής και της οικονομικής δράσης αποτελεί η ραγδαία εξέλιξη της Πληροφορικής και των συγγενών με αυτήν τεχνολογιών. Η μάθηση εννοιών που αφορούν στην Πληροφορική, και γενικότερα στις Τεχνολογίες της Πληροφορίας και της Επικοινωνίας (ΤΠΕ), καθίσταται αναγκαία από τα πρώτα κιόλας χρόνια της μαθησιακής ζωής των ατόμων και επιπλέον, ισότιμη με τη μάθηση εννοιών που αφορούν σε άλλα γνωστικά αντικείμενα όπως τα Μαθηματικά, οι Φυσικές Επιστήμες ή η Γλώσσα κ.α. Ένας μεγάλος αριθμός ερευνητών και ερευνητικών οργανισμών μελετά ζητήματα που αφορούν στην «Πληροφορική και Εκπαίδευση» ενώ γίνονται προσπάθειες ένταξης της Πληροφορικής στην εκπαιδευτική διαδικασία. Τρεις προσεγγίσεις για την ένταξη και χρήση των ΤΠΕ στην εκπαιδευτική διαδικασία, διαγράφονται στο διεθνή χώρο:

- Οι ΤΠΕ ως αυτόνομο γνωστικό αντικείμενο σε επίπεδο πληροφορικού εναλφαβητισμού (ICT literacy). Η προσέγγιση αυτή έχει ως βασική επιδίωξη την απόκτηση γνώσεων όσον αφορά στη λειτουργία των υπολογιστών και στην εισαγωγή στον προγραμματισμό τους και υλοποιείται με την καθιέρωση ενός ή περισσότερων μαθημάτων.
- Οι ΤΠΕ μέσα σε όλα τα μαθήματα ως έκφραση μιας ολιστικής, διαθεματικής προσέγγισης της μάθησης. Η προσέγγιση αυτή εμφανίστηκε σχετικά πρόσφατα και χαρακτηρίζεται από το ότι η διδασκαλία της χρήσης των ΤΠΕ καθώς και η χρήση τους ενσωματώνεται στα επιμέρους γνωστικά αντικείμενα του προγράμματος σπουδών (αποδίδεται και με τον όρο *οριζόντια ή ολιστική προσέγγιση*).
- Συνδυασμός των δύο προηγούμενων προσεγγίσεων, με έμφαση στη χρήση των υπολογιστών ως εργαλείο στα επιμέρους γνωστικά αντικείμενα. Η προσέγγιση αυτή χαρακτηρίζεται από τη διδασκαλία ενός αμιγούς μαθήματος γενικών γνώσεων πληροφορικής και την προοδευτική ένταξη της χρήσης των ΤΠΕ ως μέσο στήριξης της μαθησιακής διαδικασίας σε όλα τα γνωστικά αντικείμενα του προγράμματος σπουδών.

Στη χώρα μας, οι ΤΠΕ έχουν εισαχθεί στην εκπαίδευση ως αυτόνομο γνωστικό αντικείμενο ενώ παράλληλα πραγματοποιούνται προσπάθειες χρήσης των ΤΠΕ στα επιμέρους γνωστικά αντικείμενα. Στη συνέχεια παρουσιάζεται η τρέχουσα κατάσταση στη χώρα μας και γίνεται αναφορά σε προβλήματα και προτάσεις σχετικά με τη διδασκαλία της Πληροφορικής ως γνωστικό αντικείμενο.

#### ΟΙ ΤΠΕ ΣΕ ΟΛΟ ΤΟ ΦΑΣΜΑ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ ΣΠΟΥΔΩΝ

Τα λογισμικά που έχουν αναπτυχθεί για αυτό το σκοπό συνοδεύονται από εκπαιδευτικές δραστηριότητες, των οποίων η ανάπτυξη έγινε από επιμορφωτές και καθηγητές Όσον αφορά στην επιμόρφωση των εκπαιδευτικών, λαμβάνουν χώρα κάθε χρόνο σεμινάρια. Η επιμόρφωση περιλαμβάνει:

- ο Θέματα Οργάνωσης και Διοίκησης της Εκπαίδευσης
- ο Θέματα Παιδαγωγικής (διαχείριση της σχολικής τάξης, σχέσεις εκπαιδευτικών-μαθητών, κ.λπ.)
- ο Θέματα Διδακτικής Μεθοδολογίας ανά Μάθημα (παρουσίαση του προγράμματος σπουδών ανά μάθημα, ενημέρωση για τα διδασκόμενα βιβλία ανά μάθημα, κ.λπ.)
- ο Θέματα Εκπαιδευτικής Αξιολόγησης (αξιολόγηση του εκπαιδευτικού έργου και της σχολικής μονάδας, θεσμικό πλαίσιο αξιολόγησης των μαθητών, κ.λπ.)
- ο Θέματα Διαπολιτισμικής Εκπαίδευσης.
- ο Θέματα παιδαγωγικής αξιοποίησης των ΤΠΕ,
- ο Θέματα διδακτικής μεθοδολογίας για την αξιοποίηση των ΤΠΕ,
- ο Τεχνική κατάρτιση χρήσης του εξειδικευμένου λογισμικού,
- ο Μεθόδους διδασκαλίας στην τάξη.

Από σχετικές έρευνες και συζητήσεις που έχουν πραγματοποιηθεί με εκπαιδευτικούς της Πληροφορικής τα προβλήματα συνοψίζονται στα εξής:

-Οι ώρες διδασκαλίας του μαθήματος της Πληροφορικής κρίνονται ανεπαρκείς. Προτείνεται η αύξηση των ωρών διδασκαλίας και ο εμπλουτισμός του Προγράμματος Σπουδών και του εκπαιδευτικού υλικού με δραστηριότητες διαθεματικού χαρακτήρα.

-Ορισμένα από τα βιβλία των ΤΕΕ κρίνονται δυσνόητα ή/και πυκνογραμμένα για τους μαθητές ενώ άλλα δεν καλύπτουν επαρκώς τις ανάγκες των εργαστηριακών μαθημάτων. Κρίνεται σκόπιμο τα βιβλία να γίνουν πιο προσιτά στους μαθητές, να καλύπτουν τα εργαστηριακά μαθήματα και κυρίως τις ανάγκες των μαθητών.

-Επιπλέον, παρατηρούνται ελλείψεις στην προμήθεια λογισμικού που να καλύπτει τις ανάγκες διαφόρων μαθημάτων (όπως λογισμικό για την ανάπτυξη εφαρμογών πολυμέσων) και δεν υπάρχει υποστήριξη για τη δημιουργία βιβλιοθηκών λογισμικού και εκπαιδευτικών εφαρμογών.

-Ο εργαστηριακός εξοπλισμός θεωρείται απαρχαιωμένος σε πολλά σχολεία και δε μπορεί να καλύψει τις ανάγκες διδασκαλίας του μαθήματος. Επιπλέον, παρατηρούνται ελλείψεις ως προς τις δικτυακές υποδομές και τη δυνατότητα πρόσβασης όλων των σχολείων στο Διαδίκτυο. Επίσης, πρόβλημα αποτελεί και η συντήρηση των σχολικών εργαστηρίων. Δεν υπάρχει κατάλληλα εκπαιδευμένο προσωπικό για να αναλάβει τέτοιες αρμοδιότητες ή/και οι ίδιοι εκπαιδευτικοί Πληροφορικής δεν είναι σε θέση να ανταποκριθούν σε τέτοιες απαιτήσεις.

-Υπάρχει μεγάλη ανομοιομορφία στη σύνθεση του εκπαιδευτικού προσωπικού Πληροφορικής. Σήμερα διδάσκουν στα σχολεία της Β/θμιας Εκπαίδευσης και της Τεχνικής Εκπαίδευσης πτυχιούχοι διαφόρων τμημάτων. Αρκετοί από τους εκπαιδευτικούς δεν είναι σε θέση να ανταποκριθούν επαρκώς σε ορισμένα από τα μαθήματα. Ένας μεγάλος αριθμός των εκπαιδευτικών δε γνωρίζει θέματα που σχετίζονται με τη διδακτική και τη διδασκαλία του συγκεκριμένου γνωστικού αντικείμενου με αποτέλεσμα να μην υπάρχει ενιαία αντιμετώπιση των ιεραρχήσεων και των διδακτικών στόχων των.

-Τα επιμορφωτικά προγράμματα που έχουν πραγματοποιηθεί είναι μικρής διάρκειας και δεν καλύπτουν επαρκώς τις ανάγκες διδασκαλίας του γνωστικού αντικείμενου καθώς και θέματα διδακτικής του αντικείμενου.

-Σημαντικό πρόβλημα, επίσης, αποτελεί η έλλειψη συμβούλων πληροφορικής οι οποίοι θα μπορούσαν να βοηθήσουν τους καθηγητές της Πληροφορικής όσο και άλλων ειδικοτήτων.

Η κατάσταση που επικρατεί παρουσιάζεται πιο αναλυτικά ανά βαθμίδα:

#### Πρωτοβάθμια εκπαίδευση

Μέχρι πρόσφατα δεν έχει ενταχθεί στην εκπαιδευτική διαδικασία της πρωτοβάθμιας εκπαίδευσης οποιοδήποτε μάθημα που σχετίζεται με τις ΤΠΕ, ούτε είχε υπάρξει κάποιος κεντρικός σχεδιασμός από το Υπουργείο Εθνικής Παιδείας και Θρησκευμάτων για τον εξοπλισμό των δημοτικών σχολείων με υπολογιστές. Ωστόσο, η αδήριτη ανάγκη αλφαριθμητισμού των νέων μαθητών στις νέες τεχνολογίες οδήγησε σχεδόν το σύνολο των σχολείων της ιδιωτικής πρωτοβάθμιας εκπαίδευσης, αλλά και αρκετά σχολεία του δημόσιου τομέα στον εξοπλισμό τους ήδη από τα μέσα της δεκαετίας του '90 με υπολογιστές κυρίως με πρωτοβουλίες της τοπικής αυτοδιοίκησης ή των συλλόγων γονέων και κηδεμόνων.

#### Δευτεροβάθμια εκπαίδευση

Οι Έλληνες μαθητές και μαθήτριες έρχονται 'επίσημα' για πρώτη φορά σε επαφή με τις ΤΠΕ στην πρώτη τάξη του Γυμνασίου μέσω του μαθήματος «Πληροφορική». Διδάσκεται με αυτό το όνομα σε όλες τις τάξεις του Γυμνασίου, ως μονόωρο θεωρητικό ή μικτό μάθημα, ανάλογα με την ύπαρξη εργαστηρίου Πληροφορικής στο κάθε σχολείο. Το έντυπο διδακτικό υλικό περιλαμβάνει κοινά για όλες τις τάξεις βιβλία για το μαθητή (βιβλίο και τετράδιο εργασίας), για τον καθηγητή (βιβλίο, Ενιαίο Πλαίσιο Προγράμματος Σπουδών, προγράμματα σπουδών) καθώς και για τον υπεύθυνο εργαστηρίου (εγχειρίδιο).

#### Ενιαίο Λύκειο

Ο κύκλος σπουδών στις ΤΠΕ περιλαμβάνει τα μαθήματα επιλογής «Εφαρμογές Πληροφορικής» και «Εφαρμογές Υπολογιστών», τα οποία εντάσσονται στο ωρολόγιο πρόγραμμα της Α' και Β'/Γ' τάξεων αντίστοιχα. Αποτελούν δίωρα εβδομαδιαία προαιρετικά μαθήματα, το καθένα με φόρτο διδασκαλίας 50 διδακτικών ωρών. Η διδακτέα ύλη κατανέμεται σε τρεις ενότητες. Η διδασκαλία των δύο παραπάνω μαθημάτων επιλογής του Ενιαίου Λυκείου υποστηρίζεται από ένα, κοινό για όλες τις τάξεις, σύγγραμμα με τίτλο «Εφαρμογές Πληροφορικής και Υπολογιστών», το οποίο περιλαμβάνει βιβλίο μαθητή, βιβλίο καθηγητή και τετράδιο εργασιών.

Στο πρόγραμμα σπουδών της Τεχνολογικής Κατεύθυνσης του Ενιαίου Λυκείου, στη Γ΄ τάξη, δίδεται περισσότερη έμφαση στη διδασκαλία μαθημάτων που σχετίζονται με τις ΤΠΕ. Ο κύκλος Πληροφορικής και Υπηρεσιών περιλαμβάνει μία σειρά υποχρεωτικών και επιλογής μαθημάτων.

#### Τεχνικά Επαγγελματικά Εκπαιδευτήρια (ΤΕΕ)

Αποτελούν, σε αντιδιαστολή με το Ενιαίο Λύκειο, το δεύτερο σκέλος της μετα-υποχρεωτικής δευτεροβάθμιας εκπαίδευσης και συνδυάζουν τη γενική παιδεία με την εξειδικευμένη τεχνική και επαγγελματική γνώση, με αντικειμενικό σκοπό την επαγγελματική ένταξη των αποφοίτων στην αγορά εργασίας. Παράλληλα τα ΤΕΕ αποτελούν σήμερα την πρώτη βαθμίδα της Ελληνικής εκπαίδευσης, στην οποία περιλαμβάνεται πλήρης κύκλος σπουδών στις ΤΠΕ, παρέχοντας κρατικό πτυχίο και σχετική άδεια ασκήσεως επαγγέλματος.

#### Ινστιτούτα Επαγγελματικής Κατάρτισης (ΙΕΚ)

Η εποπτεία της τυπικής επαγγελματικής εκπαίδευσης και κατάρτισης ασκείται μέσω του Εθνικού Συστήματος Επαγγελματικής Εκπαίδευσης και Κατάρτισης. Στο πλαίσιο της λειτουργίας του ΕΣΕΕΚ ιδρύθηκε ο Οργανισμός Επαγγελματικής Εκπαίδευσης και Κατάρτισης (ΟΕΕΚ). Αποτελεί οργανισμό με διοικητική και οικονομική αυτοτέλεια που εποπτεύεται από το ΥΠΕΠΘ και έχει τη συνολική ευθύνη για τα Ινστιτούτα Επαγγελματικής Κατάρτισης (ΙΕΚ). Τα ινστιτούτα αυτά δεν εντάσσονται στο εκπαιδευτικό σύστημα και σε συγκεκριμένη εκπαιδευτική βαθμίδα αποτελούν πάντως την, άτυπη μέχρι σήμερα, μετα-δευτεροβάθμια επαγγελματική εκπαίδευση και κατάρτιση. Στην πλειοψηφία των ΙΕΚ έχουν συμπεριληφθεί αρκετές ειδικότητες σχετικές με τις ΤΠΕ. Η φοίτηση είναι διετής (4 εξάμηνα) για αποφοίτους Ενιαίου Λυκείου ή μονοετής (2 εξάμηνα) για αποφοίτους ΤΕΕ της ίδιας ειδικότητας.

Ως συμπέρασμα η εκπαίδευση στις Τεχνολογίες Πληροφορίας και Επικοινωνιών όχι μόνο των ατόμων που εξειδικεύονται σε αυτόν τον τομέα, αλλά – ακόμη περισσότερο - αυτών που εκπαιδεύονται για να ακολουθήσουν μελλοντικά ένα οποιοδήποτε άλλο επάγγελμα είναι μία υπόθεση τόσο σημαντική όσο και οι ίδιες οι τεχνολογίες. Η ποιότητα και η ποιότητα των παρεχόμενων κύκλων σπουδών, ιδιαίτερα στις πρώτες βαθμίδες του ελληνικού εκπαιδευτικού συστήματος, θα καθορίσει σε μεγάλο βαθμό το μέλλον των ΤΠΕ αλλά και ολόκληρης της κοινωνίας, αφού η πληροφορία το βασικό 'προϊόν' αυτών των τεχνολογιών αποτελεί τον κύριο συνεκτικό της ιστό.

## **ΚΕΦΑΛΑΙΟ 3**

### **ΑΝΑΛΥΣΗ ΚΑΙ ΟΡΙΣΜΟΣ ΑΠΑΙΤΗΣΕΩΝ**

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΑΙΑ

Τα προβλήματα που πρέπει να επιλύσουν οι μηχανικοί λογισμικού είναι συνήθως πολύ πολύπλοκα. Η κατανόηση της φύσης του προβλήματος μπορεί να είναι πολύ δύσκολη, ειδικά αν το σύστημα είναι καινούριο, όπως το σύστημα που θα κατασκευάσουμε στην παρούσα εφαρμογή για τη διαχείριση μιας ηλεκτρονικής εφαρμογής εκπαίδευσης. Συνεπώς είναι δύσκολο να καθορίσουμε ακριβώς τι θα πρέπει να κάνει το σύστημα. Οι περιγραφές των υπηρεσιών και των περιορισμών είναι οι *απαιτήσεις* για το σύστημα και η διαδικασία της εύρεσης, της ανάλυσης, της τεκμηρίωσης και του ελέγχου αυτών των υπηρεσιών και των περιορισμών ονομάζεται *τεχνολογία απαιτήσεων*.

Ο όρος *απαίτηση* δεν χρησιμοποιείται με τον ίδιο τρόπο σε όλη τη βιομηχανία λογισμικού. Σε μερικές περιπτώσεις, μία απαίτηση αντιμετωπίζεται σαν μία υψηλού επιπέδου εισαγωγική δήλωση της υπηρεσίας που ένα να παρέχει ένα σύστημα ή σαν ένας περιορισμός του συστήματος. Από την άλλη πλευρά, είναι μία μαθηματικά επίσημη εξήγηση της λειτουργίας του συστήματος. Στην εφαρμογή που θα περιγραφεί στη συνέχεια θα δούμε ότι και οι δύο αντιλήψεις είναι αποδεκτές και εφαρμόζονται από κοινού για την επαρκή κατανόηση των στόχων του συστήματος και την επιτυχή υλοποίησή του στη συνέχεια.

Κάποια προβλήματα που προκύπτουν κατά τη διαδικασία προσδιορισμού των απαιτήσεων του συστήματος είναι αποτέλεσμα της αποτυχίας στο να γίνει ένας ξεκάθαρος διαχωρισμός ανάμεσα στα διαφορετικά επίπεδα της περιγραφής. Για το λόγο αυτό θα κάνουμε ένα διαχωρισμό μεταξύ των απαιτήσεων του χρήστη, εννοώντας τις υψηλού επιπέδου εισαγωγικές απαιτήσεις, και των απαιτήσεων του συστήματος, εννοώντας τη λεπτομερή περιγραφή του τι θα πρέπει να κάνει το σύστημα. Εκτός από αυτά τα δύο επίπεδα λεπτομέρειας μπορεί να παραχθεί και μία πιο λεπτομερής περιγραφή για να γεφυρώσει τον προσδιορισμό των απαιτήσεων και τις δραστηριότητες του σχεδιασμού. Τα τρία επίπεδα που αναφέραμε παραπάνω μπορούν να οριστούν ως εξής:

- Οι *απαιτήσεις του χρήστη* είναι δηλώσεις, εκφρασμένες σε φυσική γλώσσα και διαγράμματα, του τι υπηρεσίες αναμένεται να παρέχει το σύστημα και τους περιορισμούς κάτω από τους οποίους θα πρέπει να λειτουργεί.
- Οι *απαιτήσεις του συστήματος* εκφράζουν τις υπηρεσίες και τους περιορισμούς του συστήματος λεπτομερειακά. Το έγγραφο των απαιτήσεων του συστήματος, που κάποιες φορές αναφέρεται ως λειτουργικές προδιαγραφές, πρέπει να είναι ακριβές. Υπάρχει περίπτωση να χρησιμεύσει σαν ένα συμβόλαιο μεταξύ του αγοραστή του συστήματος και του αναλυτή λογισμικού.
- Οι *προδιαγραφές του σχεδιασμού λογισμικού* είναι μία εισαγωγική περιγραφή του σχεδιασμού του λογισμικού, η οποία είναι η βάση για πιο λεπτομερειακό σχεδιασμό και υλοποίηση. Αυτές οι προδιαγραφές προσθέτουν περαιτέρω λεπτομέρειες στον προσδιορισμό των απαιτήσεων του συστήματος.

Οι απαιτήσεις του χρήστη θα πρέπει να γράφονται για διευθυντές των πελατών και των εργοληπτών οι οποίοι δεν έχουν λεπτομερείς τεχνικές γνώσεις του συστήματος. Ο προσδιορισμός των απαιτήσεων του συστήματος θα πρέπει να στοχεύει σε τεχνικό προσωπικό και διαχειριστές έργου. Και πάλι θα χρησιμοποιείται από προσωπικό και του πελάτη και του εργολήπτη. Οι τελικοί χρήστες του συστήματος μπορούν να διαβάζουν και τα δύο αυτά έγγραφα. Τέλος, οι προδιαγραφές του σχεδιασμού λογισμικού είναι ένα έγγραφο που κατευθύνεται προς την υλοποίηση του συστήματος και γράφεται για του μηχανικούς λογισμικού που θα κατασκευάσουν το σύστημα.

Πριν προχωρήσουμε στην ανάλυση του τι πρέπει να περιέχει ένα έγγραφο προσδιορισμού απαιτήσεων θα κάνουμε μία ακόμα κατηγοριοποίηση των απαιτήσεων για ένα σύστημα η οποία, αν και καθαρά υποθετική, μπορεί να μας βοηθήσει στην κατανόηση των εκάστοτε απαιτήσεων:

1. Λειτουργικές απαιτήσεις: Είναι οι δηλώσεις των υπηρεσιών που πρέπει να παρέχει το σύστημα, πως το σύστημα θα πρέπει να αντιδράσει με την εισαγωγή συγκεκριμένων δεδομένων και πως το σύστημα θα πρέπει να συμπεριφέρεται σε συγκεκριμένες καταστάσεις. Σε μερικές περιπτώσεις οι λειτουργικές απαιτήσεις μπορεί επίσης να εκφράζουν σαφώς τι δεν θα πρέπει να κάνει το σύστημα.
2. Μη-λειτουργικές απαιτήσεις: Είναι οι περιορισμοί πάνω στις υπηρεσίες ή στις λειτουργίες που θα παρέχει το σύστημα. Συμπεριλαμβάνουν χρονικούς

περιορισμούς, περιορισμούς πάνω στη διαδικασία της ανάπτυξης, διάφορα στάνταρ, κλπ.

3. **Απαιτήσεις τομέα:** Είναι απαιτήσεις που προέρχονται από τον τομέα της εφαρμογής του συστήματος και που αντικατοπτρίζουν χαρακτηριστικά αυτού του τομέα. Μπορεί να είναι είτε λειτουργικές είτε μη-λειτουργικές απαιτήσεις.

Στο σημείο αυτό λοιπόν έχουμε την απαραίτητη υποδομή για να αναφερθούμε στο τι πρέπει να περιέχει το έγγραφο προσδιορισμού των απαιτήσεων ενός συστήματος γενικά και στη συνέχεια να εφαρμόσουμε τις μεθοδολογίες που θα περιγράψουμε στο σύστημα διαχείρισης μιας εφαρμογής ηλεκτρονικής εκπαίδευσης.

Το έγγραφο προσδιορισμού των απαιτήσεων είναι μία επίσημη δήλωση του τι απαιτείται από τους αναλυτές του συστήματος. Θα πρέπει να συμπεριλαμβάνει και τις απαιτήσεις του χρήστη σε σχέση με το σύστημα και ένα λεπτομερή προσδιορισμό των απαιτήσεων του συστήματος. Σε μερικές περιπτώσεις μπορεί να ενοποιούνται τα δύο αυτά είδη απαιτήσεων σε μία περιγραφή. Σε άλλες περιπτώσεις οι απαιτήσεις του χρήστη προσδιορίζονται σε μία εισαγωγή του προσδιορισμού των απαιτήσεων του συστήματος.

Το έγγραφο απαιτήσεων έχει ένα ποικίλο σύνολο χρηστών που κυμαίνεται από τους προϊστάμενους της διαχείρισης του οργανισμού που πληρώνουν για το σύστημα μέχρι τους μηχανικούς που είναι υπεύθυνοι για την ανάπτυξη του λογισμικού. Ας δούμε κάθε ομάδα χρηστών πώς αντιμετωπίζει το έγγραφο απαιτήσεων. Οι πελάτες που θα αγοράσουν το σύστημα προσδιορίζουν τις απαιτήσεις και τις διαβάζουν για να ελέγξουν αν ανταποκρίνονται στις ανάγκες τους. Αν χρειαστεί καθορίζουν και αλλαγές στις απαιτήσεις τους. Οι διαχειριστές του έργου (οι μάντζερ του έργου) χρησιμοποιούν το έγγραφο απαιτήσεων για να σχεδιάσουν μια εικόνα του συστήματος και να σχεδιάσουν επίσης τη διαδικασία ανάπτυξης του συστήματος. Οι μηχανικοί του συστήματος χρησιμοποιούν τις απαιτήσεις για να κατανοήσουν τι σύστημα πρέπει να αναπτυχθεί, ενώ οι μηχανικοί ελέγχου του συστήματος για αναπτύξουν ελέγχους εγκυρότητας για το σύστημα. Τέλος, οι μηχανικοί συντήρησης του συστήματος χρησιμοποιούν το έγγραφο απαιτήσεων για να τους βοηθήσει να κατανοήσουν το σύστημα και τις σχέσεις μεταξύ των μερών του, έτσι ώστε η δουλειά της συντήρησης του συστήματος να γίνει πιο εύκολη.

Η παραπάνω δομή είναι αυτή που ακολουθήθηκε για την οργάνωση του εγγράφου προσδιορισμού απαιτήσεων για το σύστημα διαχείρισης των μαθημάτων της εφαρμογής e-School. Αρχικά θα παρουσιαστεί η γενική δομή του εγγράφου και στη συνέχεια θα περιγραφούν αναλυτικά τα βήματα που απαιτούνται για να πάρει το έγγραφο ανάλυσης των δεδομένων απαιτήσεων την τελική του μορφή.

- 3.1. Σκοπός συγγραφής του εγγράφου προσδιορισμού απαιτήσεων
- 3.2. Γενική θεώρηση του προϊόντος
- 3.3 Ανάλυση Λειτουργιών-Αντικειμενοστρεφής προσέγγιση
- 3.4 Ανάλυση Λειτουργιών-Λειτουργική προσέγγιση

Διερευνώντας τα θεωρητικά θέματα που εμπλέκονται στην ανάλυση των απαιτήσεων του συστήματος θα γίνει η παρουσίαση του τελικού εγγράφου προσδιορισμού των απαιτήσεων του συγκεκριμένου συστήματος, της μηχανογράφησης των δραστηριοτήτων που περιλαμβάνονται στη λειτουργία των μαθημάτων της εφαρμογής e-School.

### **3.1. ΣΚΟΠΟΣ ΣΥΓΓΡΑΦΗΣ ΤΟΥ ΕΓΓΡΑΦΟΥ ΠΡΟΣΔΙΟΡΙΣΜΟΥ ΑΠΑΙΤΗΣΕΩΝ**

Ο προσδιορισμός των απαιτήσεων ενός υπό κατασκευή συστήματος είναι εξαιρετικής σημασίας στην όλη διαδικασία κατασκευής του συστήματος διότι αποτελεί τη βάση της κατανόησης όλων των λειτουργιών που πρέπει να φέρει σε πέρας το σύστημα κατά τη εφαρμογή του στην πράξη. Ο καθορισμός των απαιτήσεων, δηλαδή δηλώνει το τι πρέπει να γίνει και όχι το πώς θα γίνει αυτό, πράγμα που το αναλαμβάνει το τμήμα της Σχεδίασης του Λογισμικού.



Το έγγραφο αυτό απευθύνεται σε ένα μεγάλο πλήθος χρηστών και μπορεί να καλύψει πολλές διαφορετικές ανάγκες. Για παράδειγμα, οι μαθητές, οι καθηγητές και οι administrators που διαχειρίζονται το σύστημα έχουν τη δυνατότητα μέσω του εγγράφου αυτού να προσδιορίσουν καλύτερα τις ίδιες τους τις απαιτήσεις από την εφαρμογή και πιθανόν να αναθεωρήσουν τις απόψεις τους σχετικά με τις απαιτήσεις αυτές. Οι διαχειριστές έργου στον οργανισμό ανάπτυξης του λογισμικού έχουν τη δυνατότητα μέσω του εγγράφου απαιτήσεων να σχεδιάσουν τα υπόλοιπα στάδια της ανάπτυξης του συστήματος, ενώ οι μηχανικοί λογισμικού κατανοούν καλύτερα το σύστημα που κατασκευάζουν και έτσι έχουν μεγαλύτερες δυνατότητες ελέγχου και συντήρησής του.

Ας δούμε όμως στο σημείο αυτό μια γενική περιγραφή του συστήματος που καλούμαστε να κατασκευάσουμε για να μπορέσουμε να προχωρήσουμε σε περαιτέρω ανάλυση των απαιτήσεων για το συγκεκριμένο σύστημα.

## **3.2. ΓΕΝΙΚΗ ΘΕΩΡΗΣΗ ΤΟΥ ΠΡΟΪΟΝΤΟΣ**

Όπως έχει ήδη αναφερθεί η μελέτη θα επικεντρωθεί στην κατασκευή του κατάλληλου πακέτου λογισμικού που θα μπορεί να ανταποκριθεί στις ανάγκες μάθησης των μαθημάτων της Γεωγραφίας και Ιστορίας. Σκοπός της υλοποίησης μιας εφαρμογής e-School είναι η επαρκής και πιο προσιτή εκμάθηση των πληροφοριών, καθώς και η πιο ορθή εκπαιδευτική διδασκαλία. Παραδείγματα τέτοιων δραστηριοτήτων μπορεί να είναι η μάθηση (είτε μέσω διαβάσματος είτε μέσω ήχου), η εξέταση πάνω στο διδασκόμενο μάθημα, η καταχώρηση των βαθμολογιών του μαθητή σε κάθε εξέταση, η διαχείριση των αποτελεσμάτων των εξετάσεων από τον καθηγητή και πολλά άλλα.

## **3.3. ΑΝΑΛΥΣΗ ΛΕΙΤΟΥΡΓΙΩΝ-ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΠΡΟΣΕΓΓΙΣΗ**

Το πιο βασικό είναι να κατανοηθούν οι διαδικασίες που επικρατούν σε κάθε μια ομάδα δυνατοτήτων που παρέχει η εκπαιδευτική εφαρμογή e-School. Έτσι θα αναφερθούμε στους μαθητές, στους καθηγητές και στον administrator.

Για την ανάλυση και την αναπαράσταση των διαδικασιών στην εφαρμογή χρησιμοποιήθηκε το εργαλείο Rational Rose, το οποίο παρέχει τη δυνατότητα κατασκευής όλων των διαγραμμάτων, που απαιτούνται στην αντικειμενοστρεφή ανάπτυξη λογισμικού, διαγράμματα κλάσεων, αντικειμένων, συνεργασίας, σειράς, καταστάσεων, δραστηριοτήτων, εξαρτημάτων, διανομής, περιπτώσεων χρήσης. Παρακάτω παρατίθενται τα σχετικά διαγράμματα.

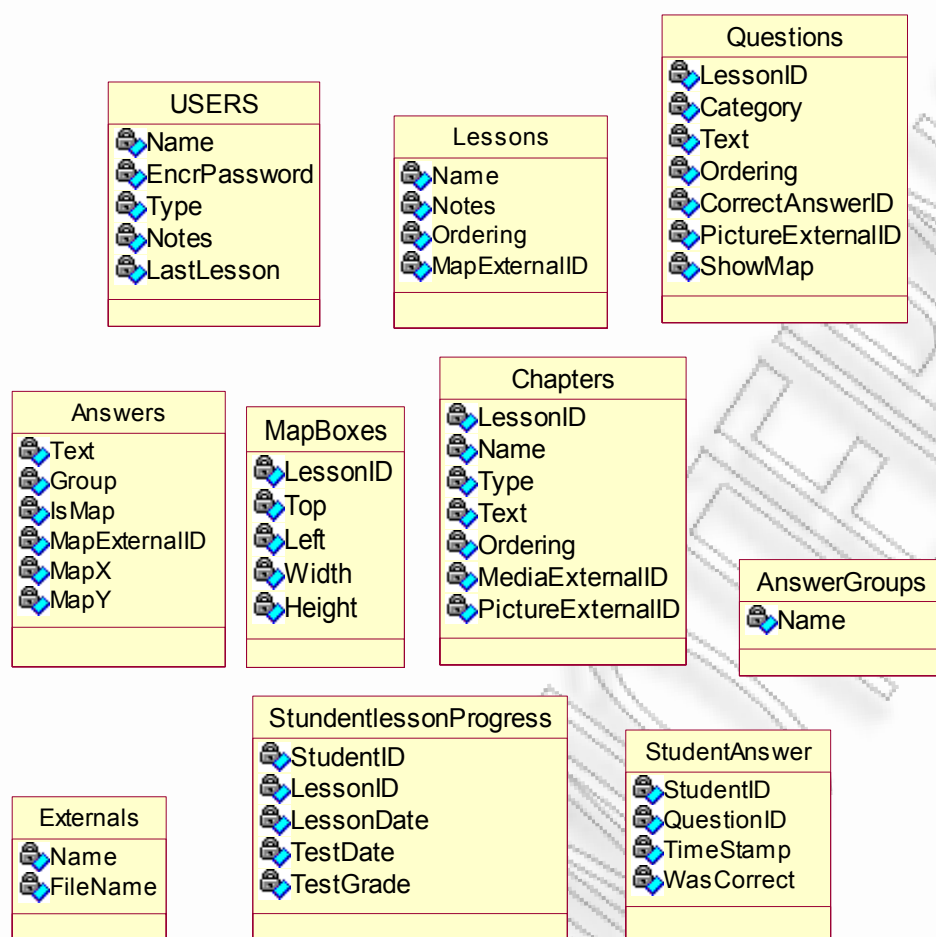
### **3.3.1 ΔΙΑΓΡΑΜΜΑΤΑ ΤΑΞΕΩΝ**

#### **A. Γενικά**

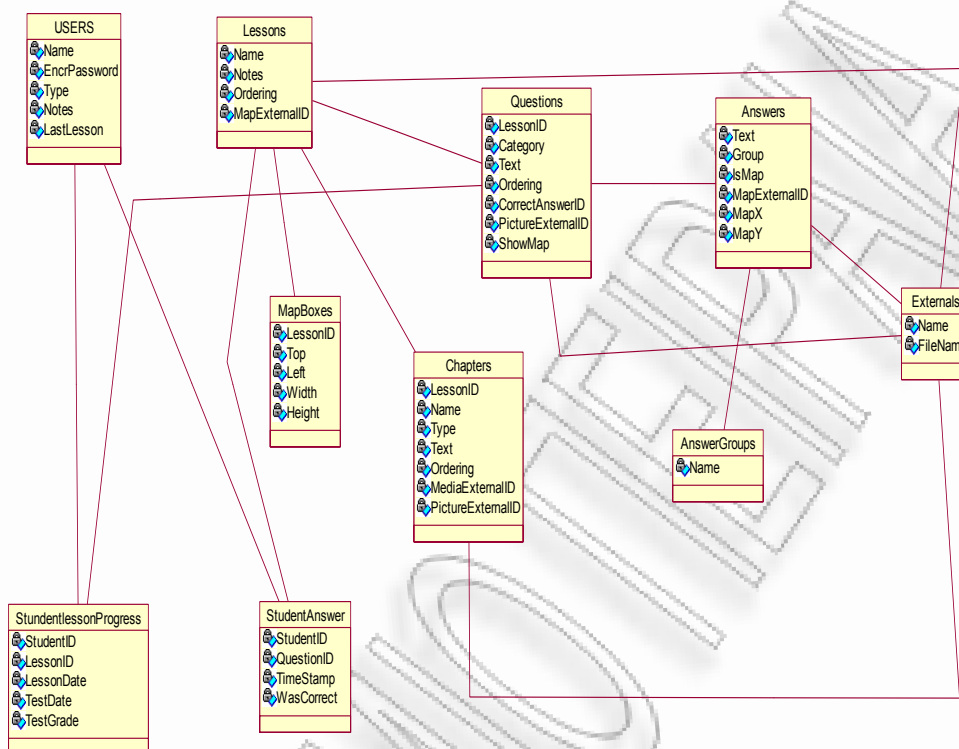
Τα διαγράμματα τάξεων αναπαριστούν τη στατική δομή όσον αφορά στις τάξεις και τις σχέσεις τους. Οι τάξεις αναπαρίστανται ως ορθογώνια τα οποία περιέχουν διαχωρισμούς. Το πρώτο τμήμα του ορθογώνιου περιέχει το όνομα της τάξης, το δεύτερο τα χαρακτηριστικά της και το τρίτο τις λειτουργίες της.

#### **B. Διαγράμματα Τάξεως Της Εφαρμογής**

Θα περιγραφούν οι βασικές τάξεις. Τα χαρακτηριστικά της κάθε τάξης παρουσιάζονται.



Οι τάξεις αυτές συνδέονται μεταξύ τους με διάφορες σχέσεις (συσχετισμούς). Οι συσχετισμοί αναπαριστούν δομικές σχέσεις μεταξύ τάξεων των αντικειμένων. Ένας συσχετισμός συμβολίζει μία πληροφορία με κύκλο ζωής. Πιο κάτω φαίνονται οι συσχετισμοί μεταξύ των τάξεων.



### 3.3.2 ΔΙΑΓΡΑΜΜΑΤΑ ΑΝΤΙΚΕΙΜΕΝΩΝ

#### A. Γενικά

Το αντικείμενο είναι μια έννοια, ή αφαίρεση, ή πράγμα με σαφή όρια και σημασία στο χώρο του προβλήματος. Το αντικείμενο είναι μοναδικό, δηλαδή ξεχωρίζει από κάθε άλλο αντικείμενο και έχει ταυτότητα.

Κάθε αντικείμενο αναπαριστάται με ένα ορθογώνιο, το οποίο περιέχει είτε το όνομα του αντικειμένου, είτε το όνομα και τη τάξη του αντικειμένου (χωρισμένα με άνω και κάτω τελεία), ή μόνο τη τάξη του αντικειμένου (σε αυτή τη περίπτωση το αντικείμενο θεωρείται ότι είναι ανώνυμο). Το όνομα από μόνο του αντιστοιχεί σε ένα ελλειπές μοντέλο, στο οποίο η τάξη του αντικειμένου δεν έχει καθοριστεί. Η τάξη από μόνη της αποφεύγει την εισαγωγή μη απαραίτητων ονομάτων στα διαγράμματα, ενώ επιτρέπει την έκφραση γενικών μηχανισμών που είναι έγκυροι για πολλά αντικείμενα.

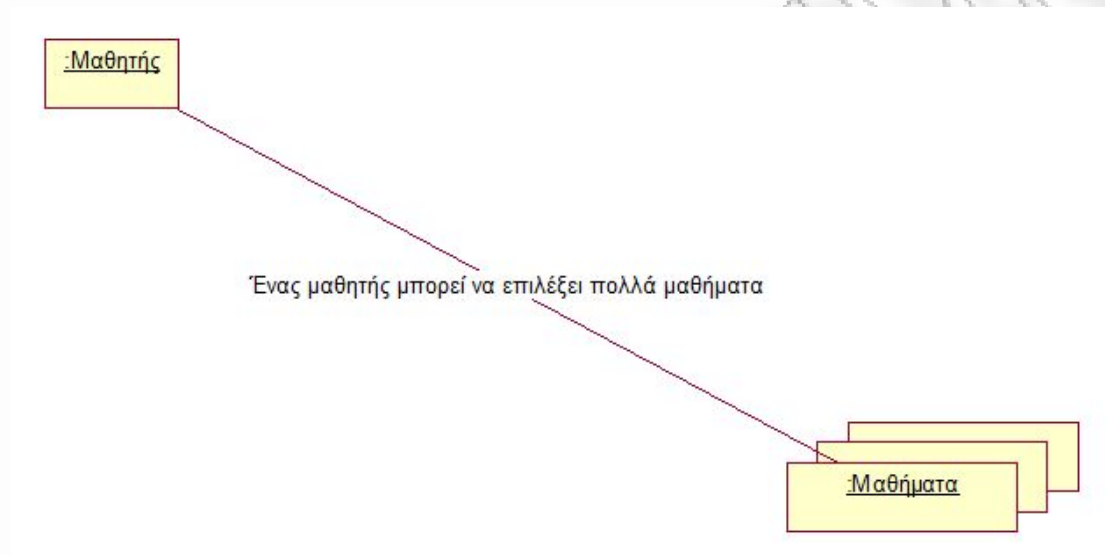
Τα αντικείμενα συνδέονται μέσω συνδέσμων οι οποίοι είναι στιγμιότυπα των συσχετισμών μεταξύ των τάξεων των αντικειμένων που μελετώνται. Η συγκεκριμένη αναπαράσταση μιας δομής με αντικείμενα είναι συχνά πιο σημαντική από την αφηρημένη αναπαράσταση μιας δομής χρησιμοποιώντας τάξεις ειδικά στην περίπτωση των επαναληπτικών δομών. Παρακάτω παραθέτουμε μερικά διαγράμματα αντικειμένων τα οποία είναι στιγμιότυπα του διαγράμματος κλάσεων.

## B. Αντικείμενα Της Εφαρμογής «e-School»

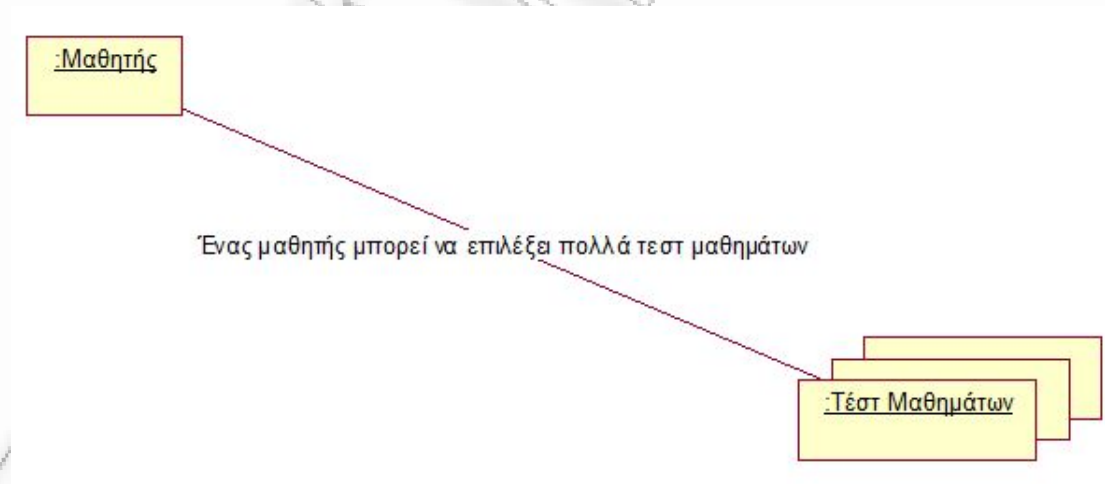
Τα αντικείμενα που χρησιμοποιούνται στην εργασία για τη διαχείριση της εφαρμογής e-School είναι τα αντίστοιχα των τάξεων Μαθητής, Καθηγητής, Administrator. Προτιμήθηκε η ονομασία των αντικειμένων μόνο με το όνομα της τάξης στην οποία ανήκει το καθένα.

Παρακάτω παρατίθενται τα διαγράμματα αντικειμένων που σχεδιάστηκαν με το Rational Rose κατά τη φάση της έναρξης:

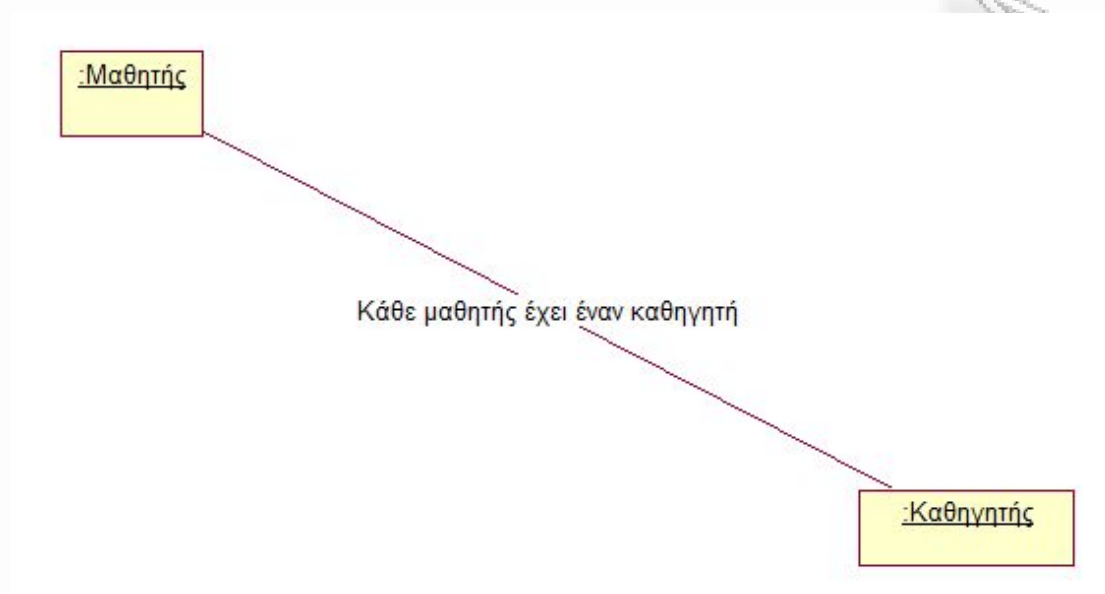
1. Ένας μαθητής μπορεί να επιλέξει πολλά μαθήματα προς παρακολούθηση και αυτό φαίνεται στο παρακάτω διάγραμμα:



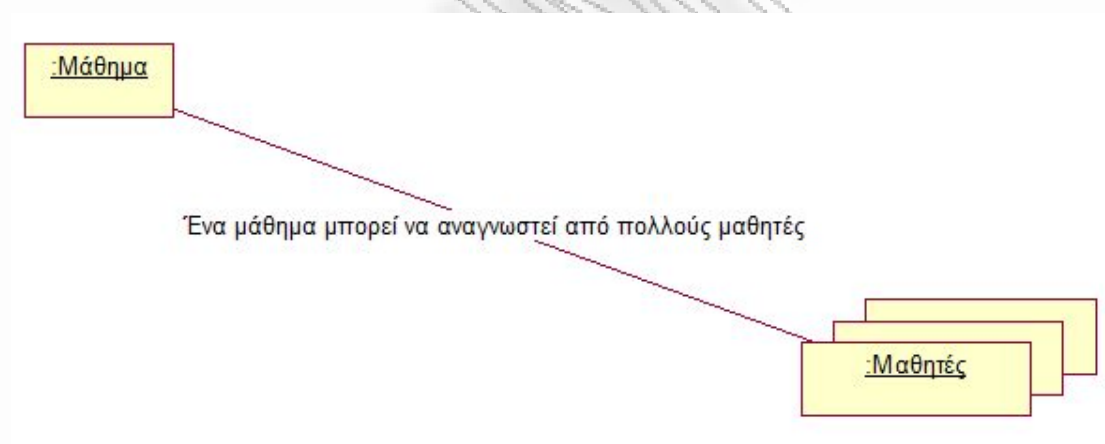
2. Ένας μαθητής μπορεί να επιλέξει πολλά τεστ προς εξέταση, ανάλογα με τα μαθήματα που έχει παρακολουθήσει και αυτό φαίνεται στο παρακάτω διάγραμμα:



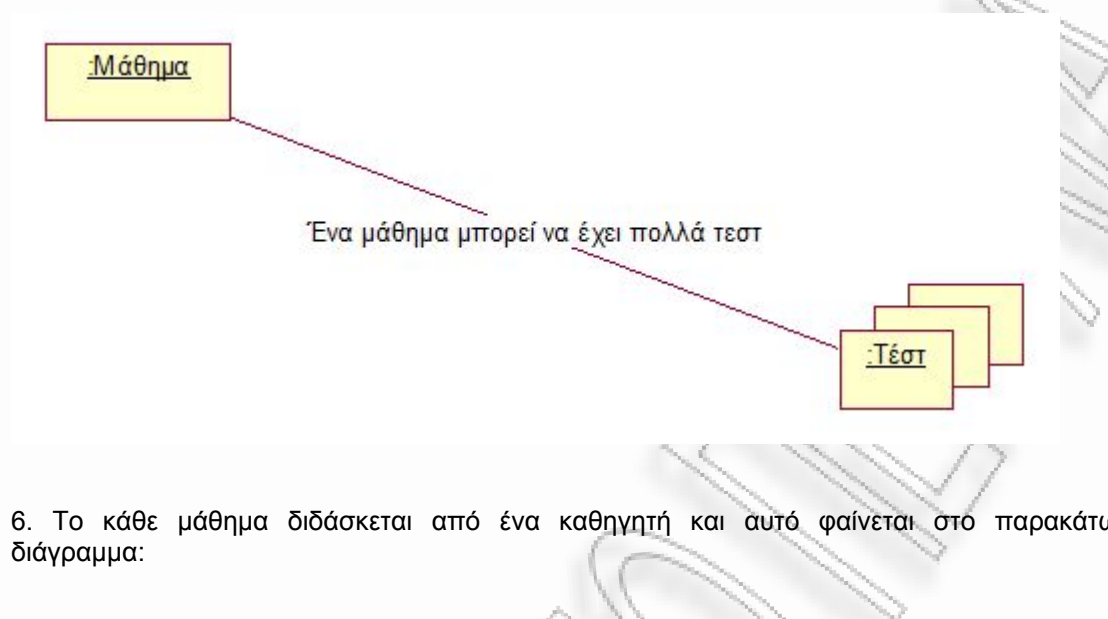
3. Κάθε μαθητής έχει έναν επιβλέποντα καθηγητή ανά μάθημα και αυτό φαίνεται στο παρακάτω διάγραμμα:



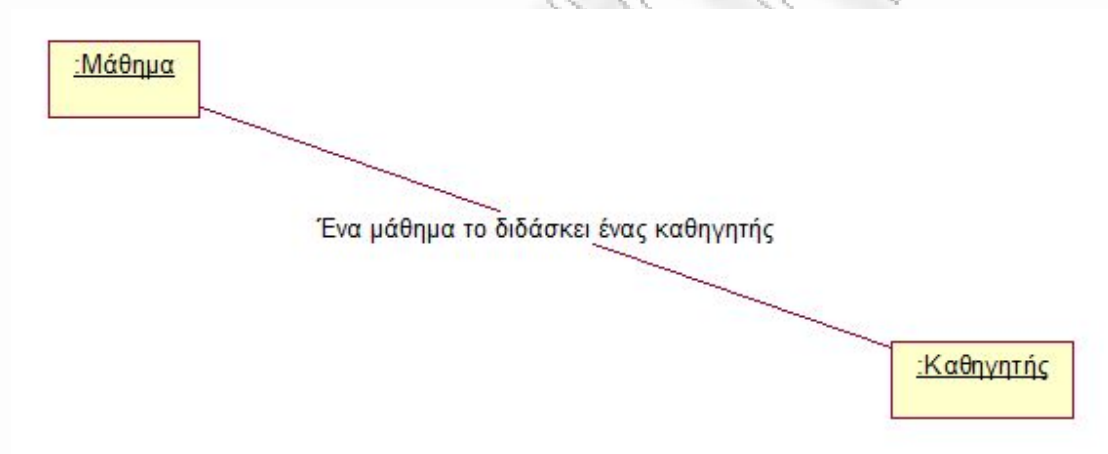
4. Ένα μάθημα μπορεί να αναγνωστεί/ παρακολουθηθεί από πολλούς μαθητές και αυτό φαίνεται στο παρακάτω διάγραμμα:



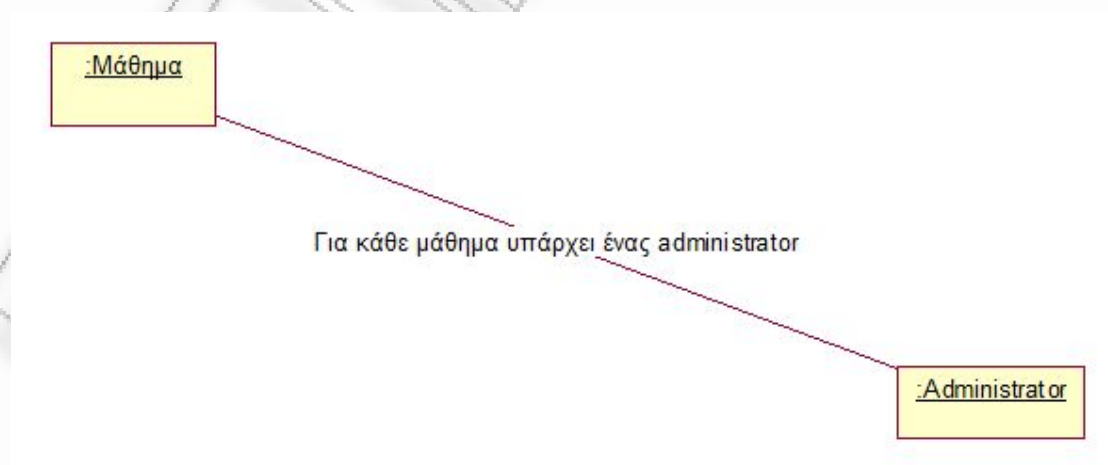
5. Ένα μάθημα μπορεί να περιλαμβάνει πολλά τεστ προς εξέταση και αυτό φαίνεται στο παρακάτω διάγραμμα:



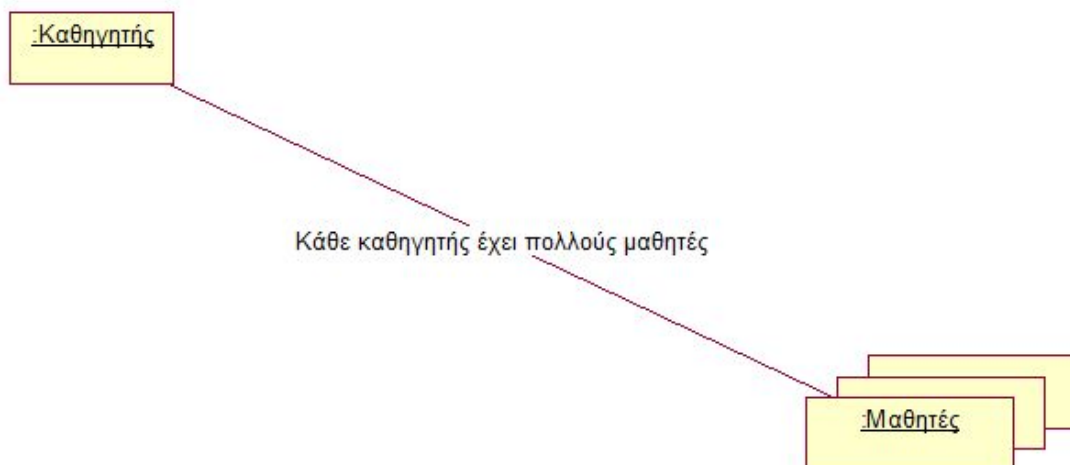
6. Το κάθε μάθημα διδάσκεται από ένα καθηγητή και αυτό φαίνεται στο παρακάτω διάγραμμα:



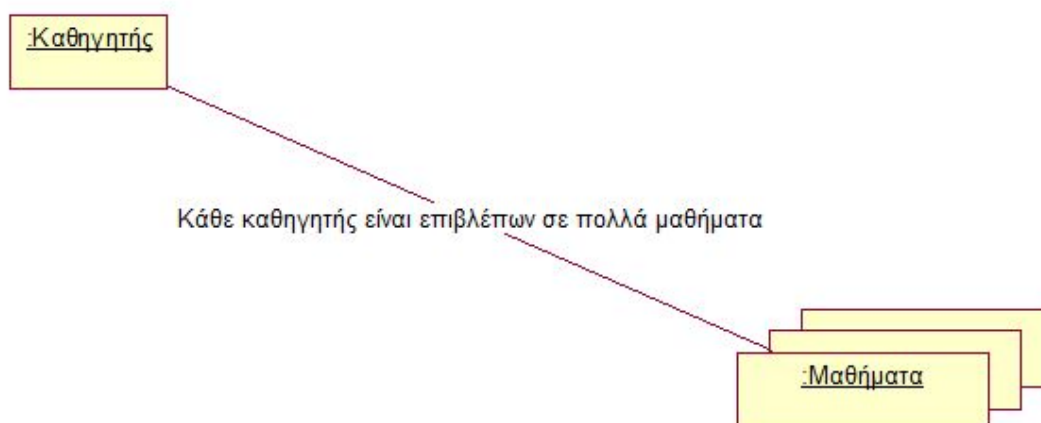
7. Για κάθε μάθημα υπάρχει ένας υπεύθυνος administrator, ο οποίος διαχειρίζεται όλες τις ρυθμίσεις του εκάστοτε μαθήματος και αυτό φαίνεται στο παρακάτω διάγραμμα:



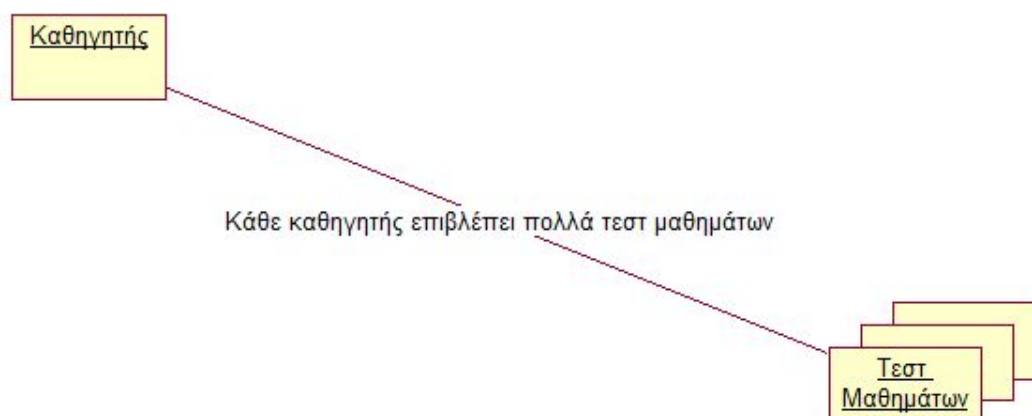
8. Ο κάθε καθηγητής έχει υπό την επίβλεψη του πολλούς μαθητές και αυτό φαίνεται στο παρακάτω διάγραμμα:



9. Ο κάθε καθηγητής είναι υπεύθυνος για την ορθή λειτουργία πολλών μαθημάτων και αυτό φαίνεται στο παρακάτω διάγραμμα:

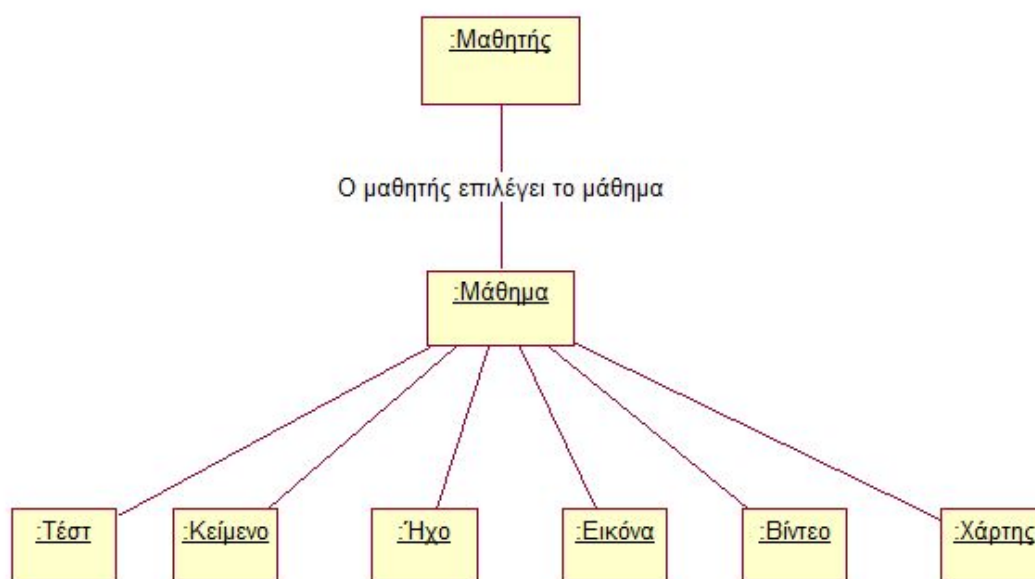


10. Ο κάθε καθηγητής επιβλέπει τα τεστ πολλών μαθημάτων και μαθητών και αυτό φαίνεται στο παρακάτω διάγραμμα:



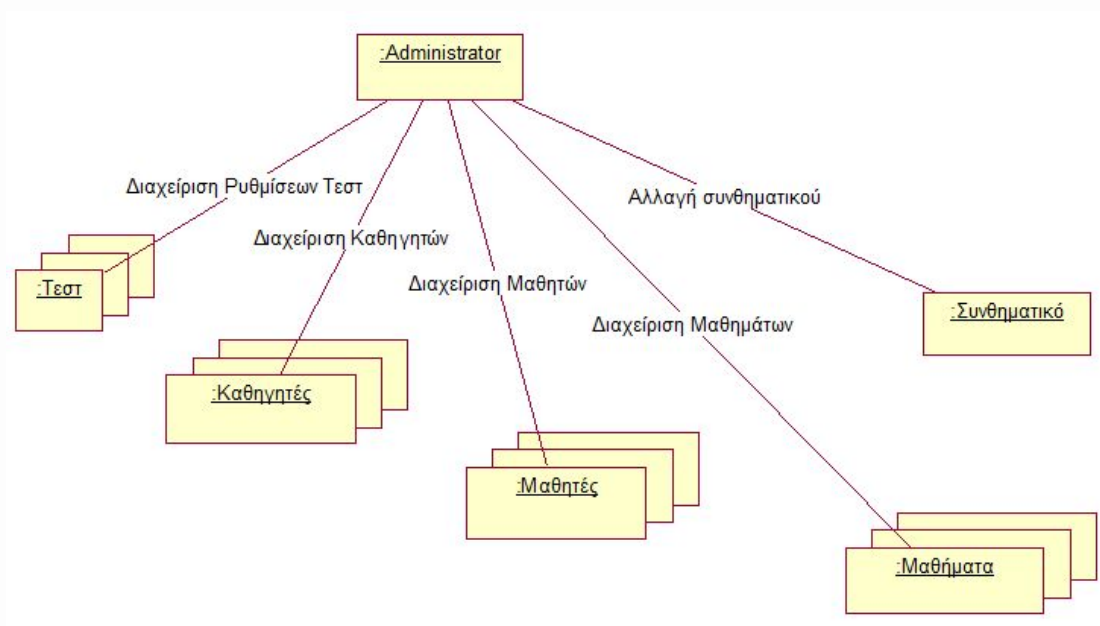
Παρακάτω παρατίθενται τα διαγράμματα αντικειμένων που σχεδιάστηκαν με το Rational Rose κατά την εκπόνηση της μελέτης, τα οποία για τη φάση εκπόνησης μελέτης έχουν μεγαλύτερη λεπτομέρεια. Στα διαγράμματα αντικειμένων σκοπός μας είναι να παρουσιάσουμε με αναλυτικό τρόπο τις διαδικασίες που λαμβάνουν χώρα στο σύστημα της εφαρμογής e-School.

1. Στο παρακάτω διάγραμμα αντικειμένων περιγράφονται οι δυνατότητες που έχει ο μαθητής με την επιλογή κάποιου μαθήματος. Η διαδικασία που ακολουθείται είναι να γίνει η επιλογή του μαθήματος από τον μαθητή και στη συνέχεια μπορεί να επιλέξει εάν επιθυμεί να διδαχτεί το μάθημα διαβάζοντας το γραπτό κείμενο, ακούγοντάς το ή κάνοντας και τα δύο. Όποιον τρόπο και να επιλέξει, θα έχει στην ευχέρειά του πλήθος εικόνων, βίντεο και χαρτών που πλαισιώνουν το μάθημα, καθώς και το τεστ που μπορεί να εξεταστεί στο αντίστοιχο κεφάλαιο. Η παραπάνω διαδικασία φαίνεται στο παρακάτω διάγραμμα:

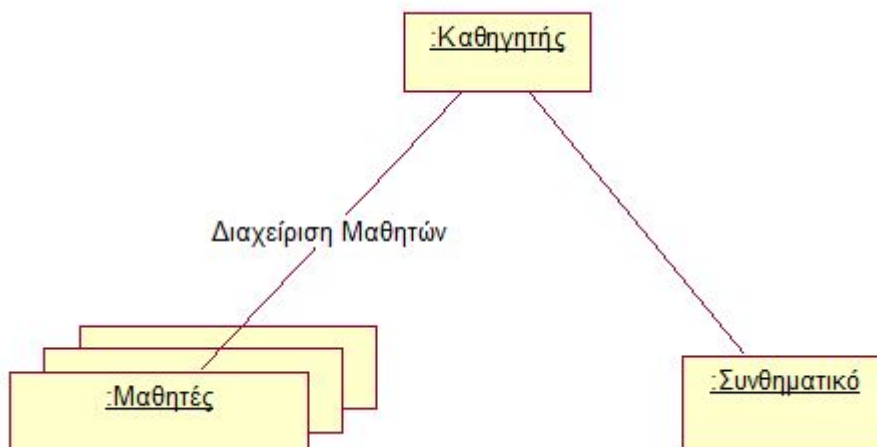


2. Στο παρακάτω διάγραμμα αντικειμένων φαίνονται οι αρμοδιότητες του administrator. Ο administrator έχει τη δυνατότητα να διαχειρίζεται όλες τις ρυθμίσεις των τεστ (διάρκεια του τεστ, αριθμός προσπαθειών σε ερωτήσεις πολλαπλής επιλογής, αριθμός προσπαθειών σε ερωτήσεις χάρτη, καθορισμός του άριστα και της βάσης, αριθμός δεκαδικών ψηφίων στην βαθμολογία), τα μαθήματα (τίτλος μαθήματος, σημειώσεις μαθήματος, ύλη μαθήματος, ερωτήσεις, χάρτης, εισαγωγή νέου μαθήματος, διαγραφή μαθήματος), τα προφίλ των καθηγητών (καθορισμών δικαιωμάτων, διαγραφή συνθηματικού καθηγητή, εισαγωγή νέου, διαγραφή καθηγητή), τους μαθητές (επίβλεψη της αξιολόγησής τους, διαγραφή συνθηματικού μαθητή, εισαγωγή νέου, διαγραφή μαθητή) και να αλλάζει το συνθηματικό του. Η παραπάνω διαδικασία φαίνεται στο παρακάτω διάγραμμα:

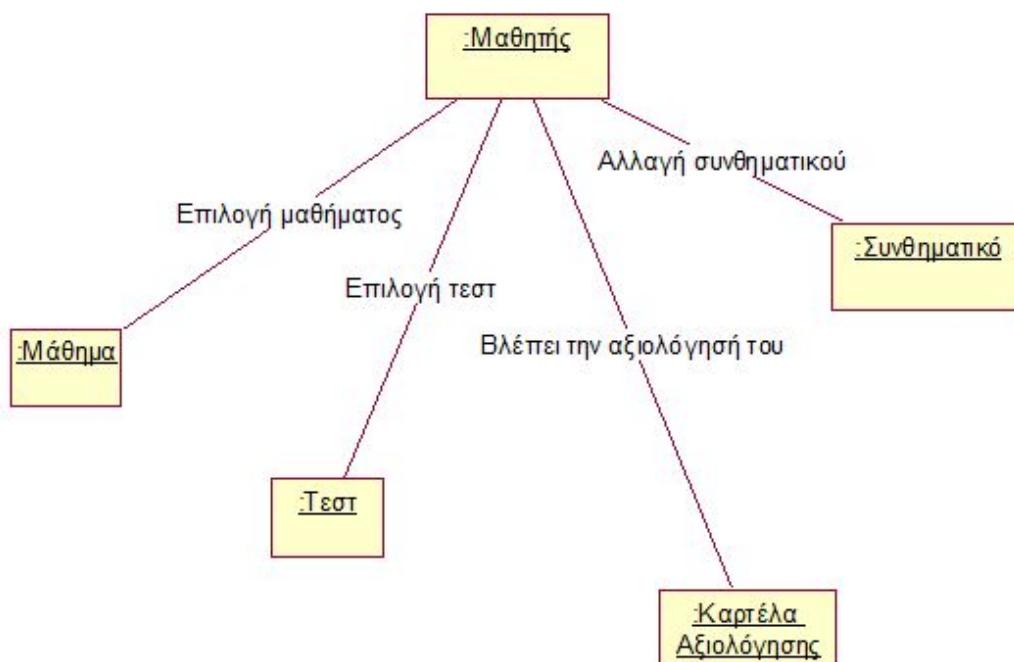




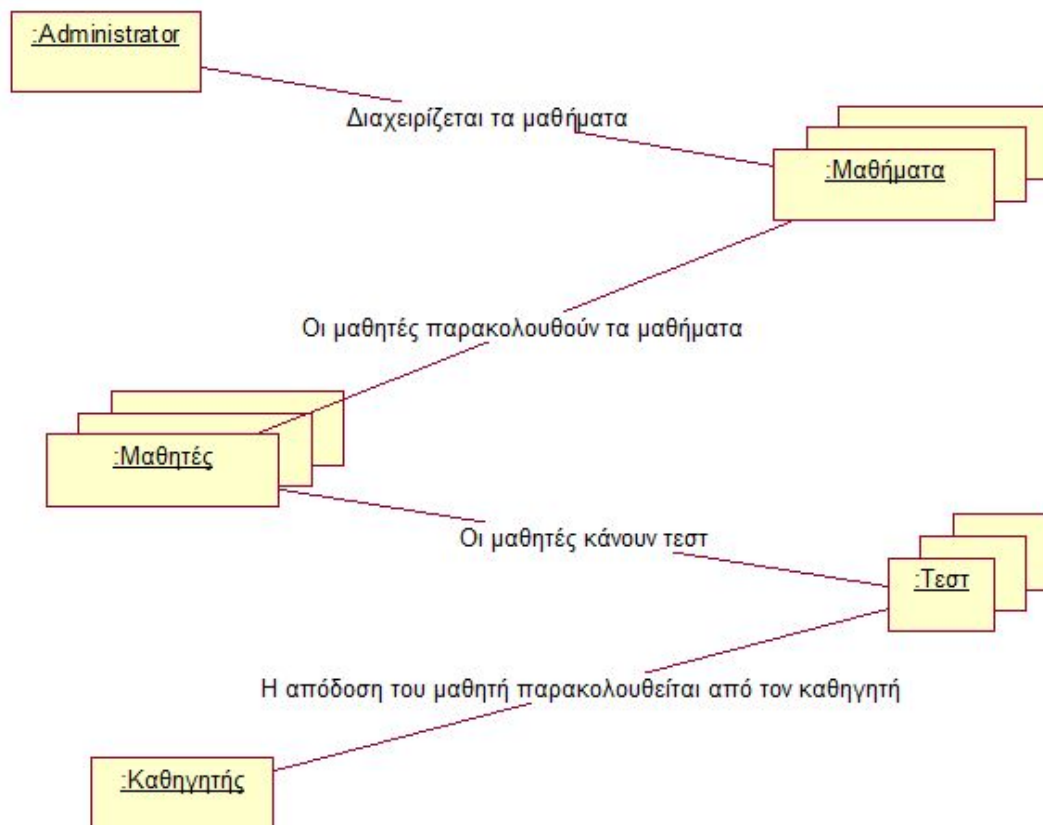
3. Παρακάτω παρατίθεται ένα διάγραμμα αντικειμένων παρόμοιο με το προηγούμενο όπου φαίνονται οι αρμοδιότητες του καθηγητή. Ο καθηγητής έχει τη δυνατότητα να διαχειρίζεται τους μαθητές (επίβλεψη της αξιολόγησής τους, διαγραφή συνθηματικού μαθητή, εισαγωγή νέου, διαγραφή μαθητή) και να αλλάζει το συνθηματικό του. Η παραπάνω διαδικασία φαίνεται στο παρακάτω διάγραμμα:



4. Στο παρακάτω διάγραμμα αντικειμένων φαίνονται οι αρμοδιότητες του μαθητή. Ο μαθητής έχει τη δυνατότητα να επιλέξει το μάθημα που επιθυμεί να παρακολουθήσει, να επιλέξει το τεστ που επιθυμεί να εξεταστεί, να παρακολουθεί την αξιολόγησή του και να αλλάζει το συνθηματικό του. Η παραπάνω διαδικασία φαίνεται στο παρακάτω διάγραμμα:



5. Στο παρακάτω διάγραμμα αντικειμένων φαίνεται μία διαδικασία στην οποία μετέχουν όλοι οι χρήστες της εφαρμογής. Ο administrator διαχειρίζεται τα μαθήματα, όπως αναλύθηκε στο δεύτερο διάγραμμα αντικειμένων και στη συνέχεια οι μαθητές μπορούν να παρακολουθήσουν το μάθημα της επιλογής τους. Έπειτα οι μαθητές εξετάζονται στο τεστ του μαθήματος που παρακολούθησαν και η απόδοσή τους παρακολουθείται και αξιολογείται από τον επιβλέπων καθηγητή. Η παραπάνω διαδικασία φαίνεται στο παρακάτω διάγραμμα:



### 3.3.3 ΔΙΑΓΡΑΜΜΑΤΑ ΣΥΝΕΡΓΑΣΙΑΣ

#### A. Γενικά

Τα διαγράμματα συνεργασίας απεικονίζουν τις αλληλεπιδράσεις ανάμεσα στα αντικείμενα, χρησιμοποιώντας μια στατική χωρική δομή, που διευκολύνει την αναπαράσταση της συνεργασίας ενός συνόλου αντικειμένων. Τα διαγράμματα συνεργασίας εκφράζουν τόσο το περιβάλλον ενός συνόλου αντικειμένων (μέσω αντικειμένων και συνδέσμων) όσο και την αλληλεπίδραση ανάμεσα σε αυτά τα αντικείμενα (αναπαριστώντας εκπομπές μηνυμάτων). Αυτά τα διαγράμματα είναι μία επέκταση των διαγραμμάτων αντικειμένων.

Επίσης, το περιεχόμενο μιας αλληλεπίδρασης περιλαμβάνει τα ορίσματα, τις τοπικές μεταβλητές που δημιουργήθηκαν κατά την διάρκεια της εκτέλεσης, και τους συνδέσμους ανάμεσα στα αντικείμενα που συμμετέχουν στην αλληλεπίδραση. Μια αλληλεπίδραση εκτελείται από ένα σύνολο αντικειμένων που συνεργάζονται ανταλλάσσοντας μηνύματα. Αυτά τα μηνύματα εμφανίζονται κατά μήκος των συνδέσμων που συνδέουν τα αντικείμενα, χρησιμοποιώντας βέλη που δείχνουν προς τον παραλήπτη του μηνύματος.

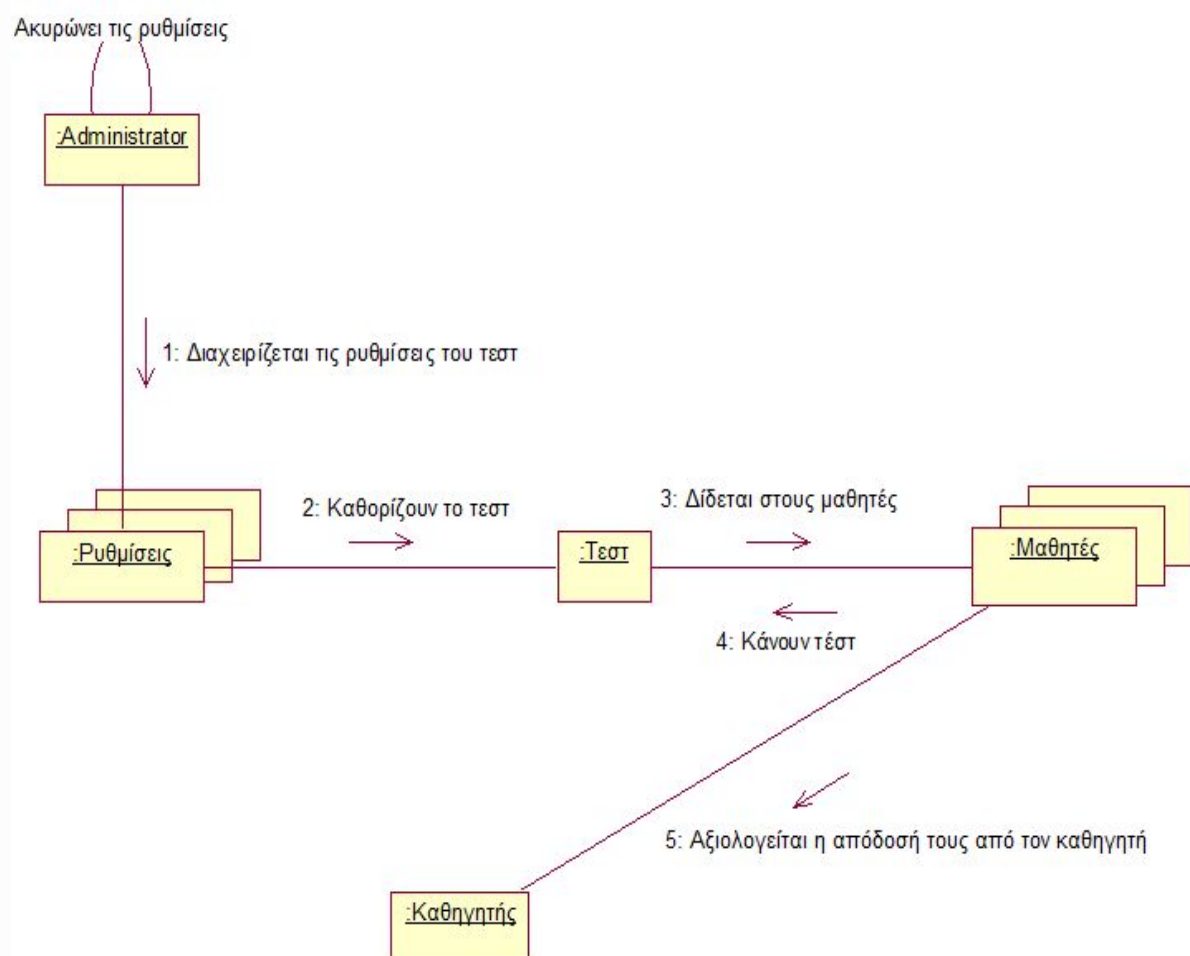
Τέλος, είναι σημαντικό να επισημανθεί πως ο χρόνος δεν αναπαριστάται σαφώς σε ένα διάγραμμα συνεργασίας και αυτό έχει ως αποτέλεσμα τα διάφορα μηνύματα να αριθμούνται για να δηλώσουν τη σειρά αποστολής.

#### B. Διαγράμματα Συνεργασίας Της Εφαρμογής

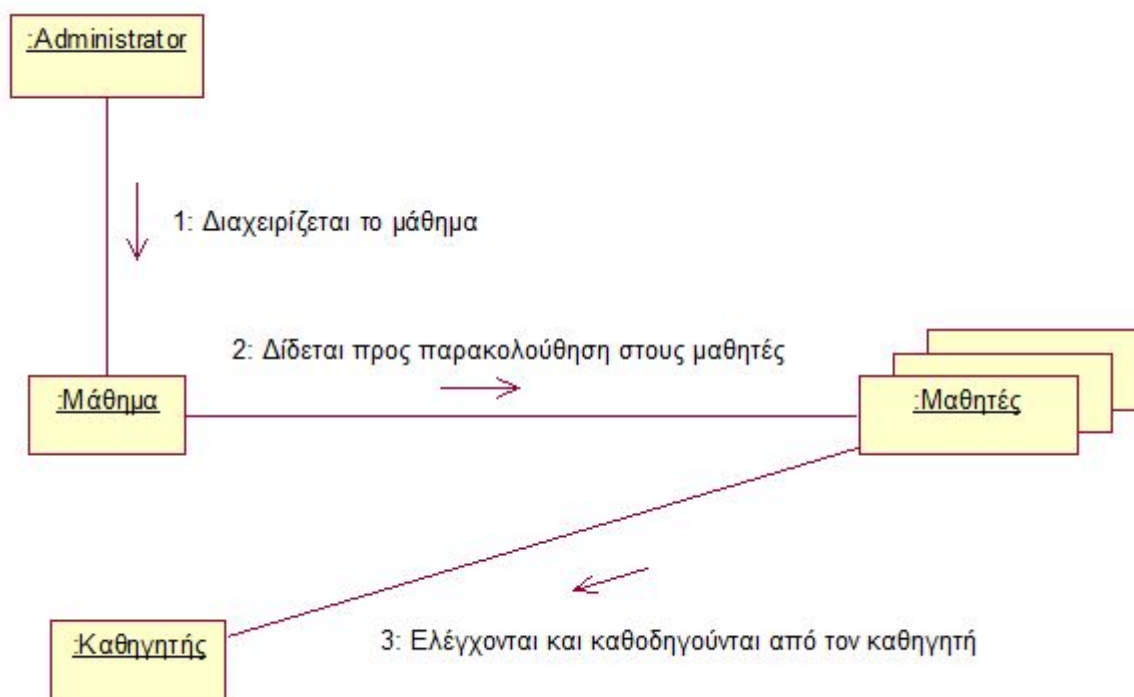
Τα αντικείμενα που χρησιμοποιούνται στην εργασία για τη διαχείριση της εφαρμογής e-School είναι τα αντίστοιχα των τάξεων Μαθητής, Καθηγητής, Administrator. Παρατηρώντας τα διαγράμματα αντικειμένων κατασκευάζουμε τα διαγράμματα συνεργασίας.

Παρακάτω παρατίθενται τα διαγράμματα αντικειμένων που σχεδιάστηκαν με το Rational Rose στην διαδικασία της φάση της έναρξης:

1. Σε αυτό το διάγραμμα αναπαριστάται η δυνατότητα του administrator να διαχειρίζεται τις ρυθμίσεις των τεστ, οι οποίες καθορίζουν το τεστ. Στη συνέχεια το τεστ δίδεται στους μαθητές, οι οποίοι εξετάζονται και αξιολογείται η απόδοσή τους από τον καθηγητή. Παρακάτω παρατίθεται το διάγραμμα συνεργασίας που σχεδιάστηκε με το Rational Rose:

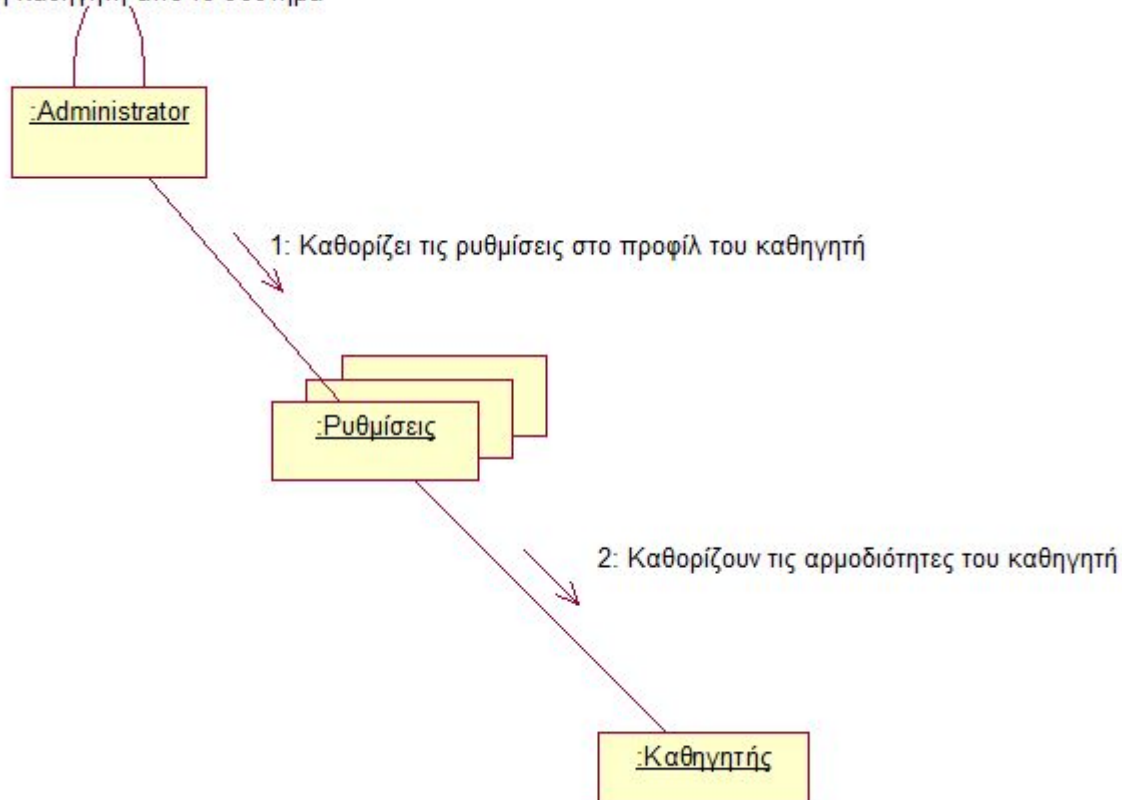


2. Σε αυτό το διάγραμμα αναπαριστάται η δυνατότητα του administrator να διαχειρίζεται τις ρυθμίσεις των μαθημάτων. Στη συνέχεια το μάθημα δίδεται στους μαθητές, οι οποίοι το παρακολουθούν και κατά τη διάρκεια της παρακολούθησης επιβλέπονται και καθοδηγούνται από τον καθηγητή. Παρακάτω παρατίθεται το διάγραμμα συνεργασίας που σχεδιάστηκε με το Rational Rose:

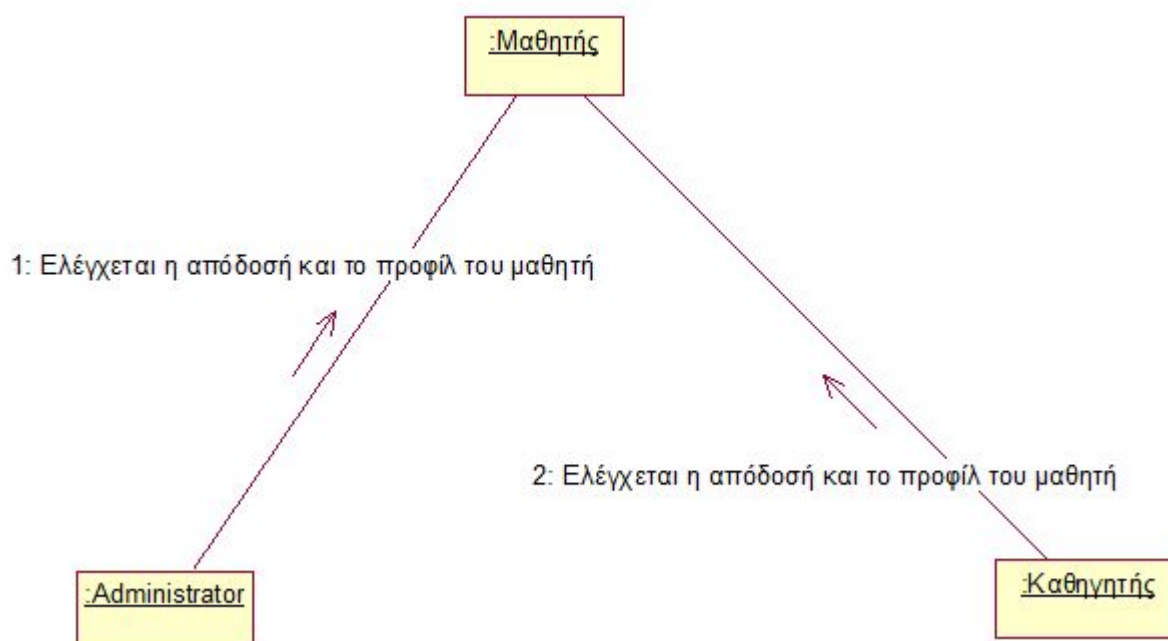


3. Σε αυτό το διάγραμμα αναπαριστάται η δυνατότητα του administrator να διαχειρίζεται τις ρυθμίσεις στα προφίλ των καθηγητών. Ο administrator διαχειρίζεται τις ρυθμίσεις που αφορούν τα προφίλ των καθηγητών, τα οποία με τη σειρά τους καθορίζουν τις αρμοδιότητες των καθηγητών. Παρακάτω παρατίθεται το διάγραμμα συνεργασίας που σχεδιάστηκε με το Rational Rose:

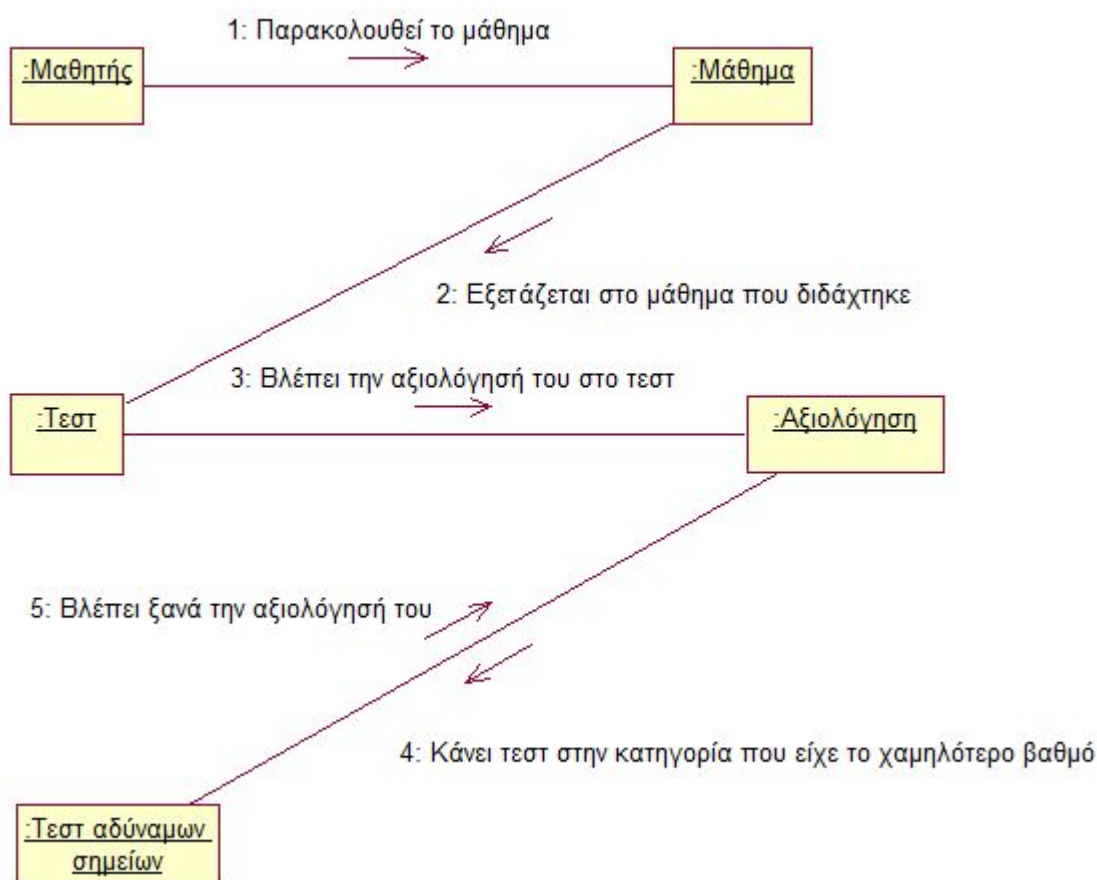
## Διαγραφή καθηγητή από το σύστημα



4. Σε αυτό το διάγραμμα συνεργασίας προσδιορίζονται οι αλληλεπιδράσεις μεταξύ των administrator, καθηγητών και μαθητών σε σχέση με τον έλεγχο της απόδοσης ενός μαθητή και το προφίλ του:



5. Σε αυτό το διάγραμμα συνεργασίας προσδιορίζονται όλες οι αλληλεπιδράσεις μεταξύ του μαθητή, των μαθημάτων και των τεστ:



### 3.3.4 ΔΙΑΓΡΑΜΜΑΤΑ ΣΕΙΡΑΣ

#### A. Γενικά

Τα διαγράμματα σειράς αναπαριστούν αλληλεπιδράσεις ανάμεσα στα αντικείμενα από μία χρονική άποψη. Σε αντίθεση με τα διαγράμματα συνεργασίας, το περιβάλλον των αντικειμένων δεν αναπαριστάται σαφώς. Για αυτόν τον λόγο η αναπαράσταση επικεντρώνεται στην έκφραση των αλληλεπιδράσεων.

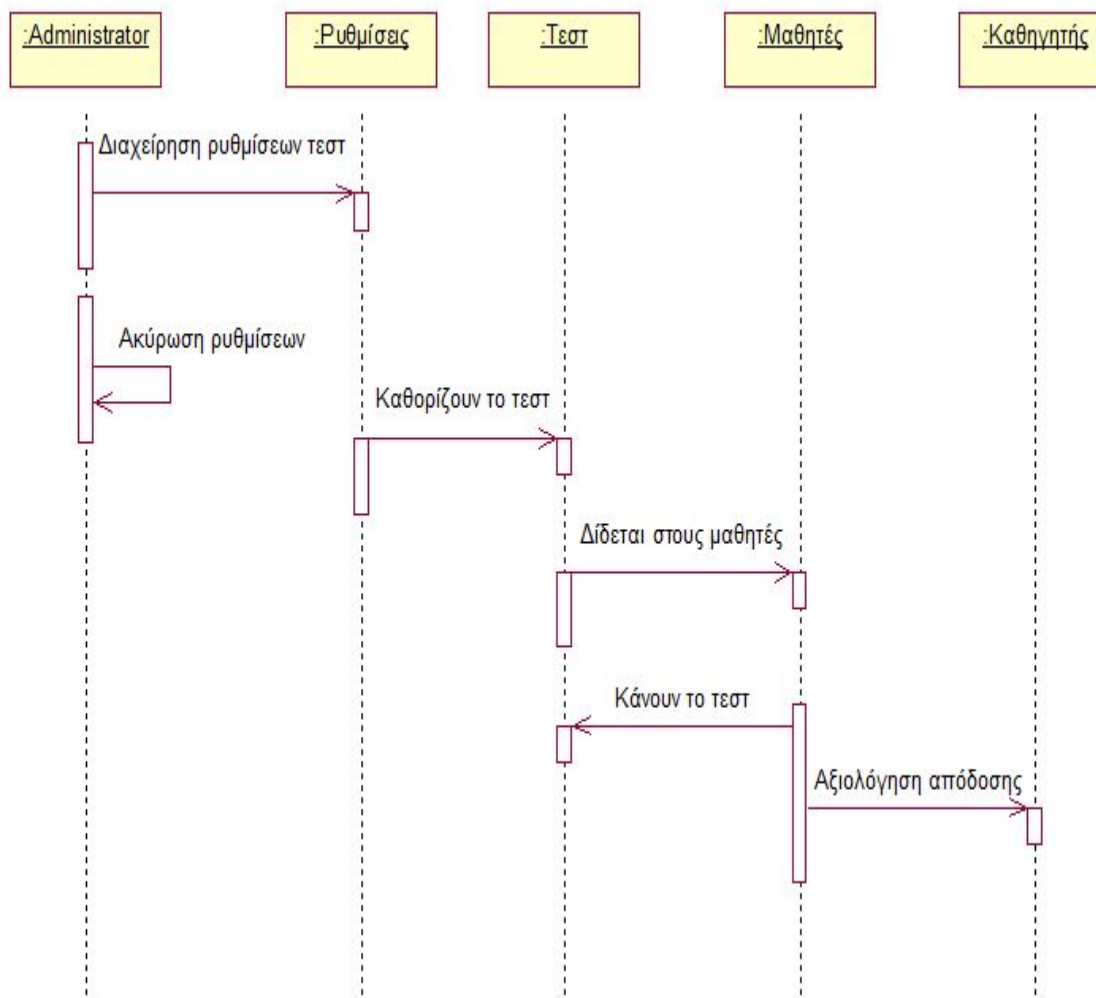
Ένα διάγραμμα σειράς αναπαριστά μια αλληλεπίδραση ανάμεσα σε αντικείμενα, που επικεντρώνεται στην χρονολογία εκπομπής μηνυμάτων. Ένα αντικείμενο αναπαρίσταται με ένα ορθογώνιο και μία κάθετη γραμμή, που καλείται γραμμή ζωής του αντικειμένου.

#### B. Διαγράμματα Σειράς Της Εφαρμογής

Τα αντικείμενα που χρησιμοποιούνται στην εργασία για τη διαχείριση της εφαρμογής e-School είναι τα αντίστοιχα των τάξεων Μαθητής, Καθηγητής, Administrator. Παρατηρώντας τα διαγράμματα αντικειμένων κατασκευάζουμε τα διαγράμματα συνεργασίας.

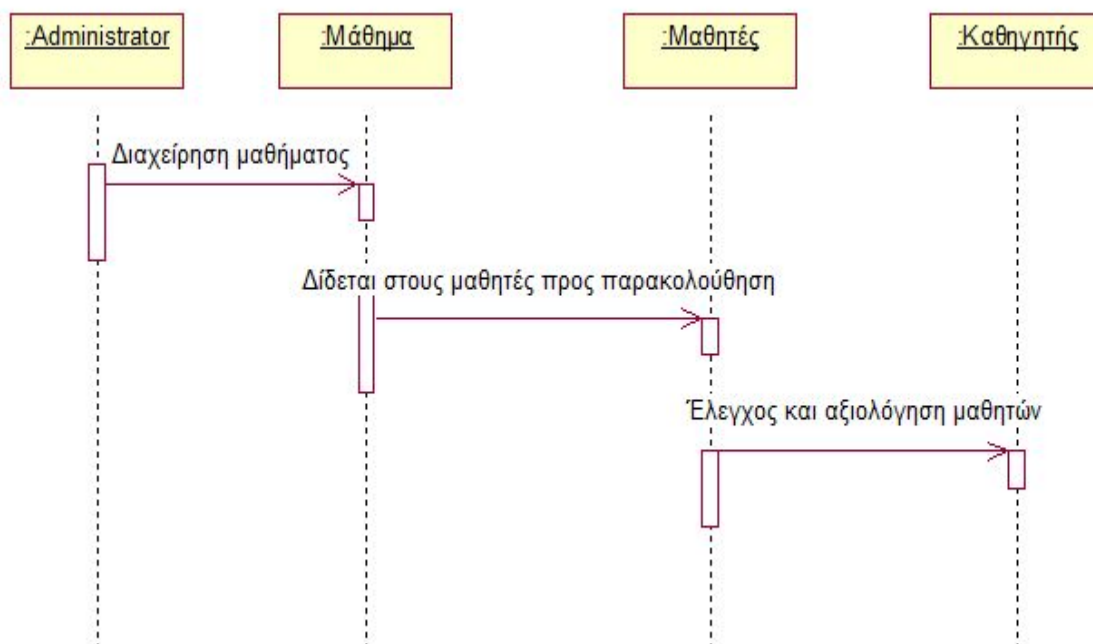
Παρακάτω παρατίθενται τα διαγράμματα αντικειμένων που σχεδιάστηκαν με το Rational Rose:

1. Σε αυτό το διάγραμμα αναπαριστάται η δυνατότητα του administrator να διαχειρίζεται τις ρυθμίσεις των τεστ, οι οποίες καθορίζουν το τεστ. Στη συνέχεια το τεστ δίδεται στους μαθητές, οι οποίοι εξετάζονται και αξιολογείται η απόδοσή τους από τον καθηγητή. Παρακάτω παρατίθεται το διάγραμμα σειράς που σχεδιάστηκε με το Rational Rose:

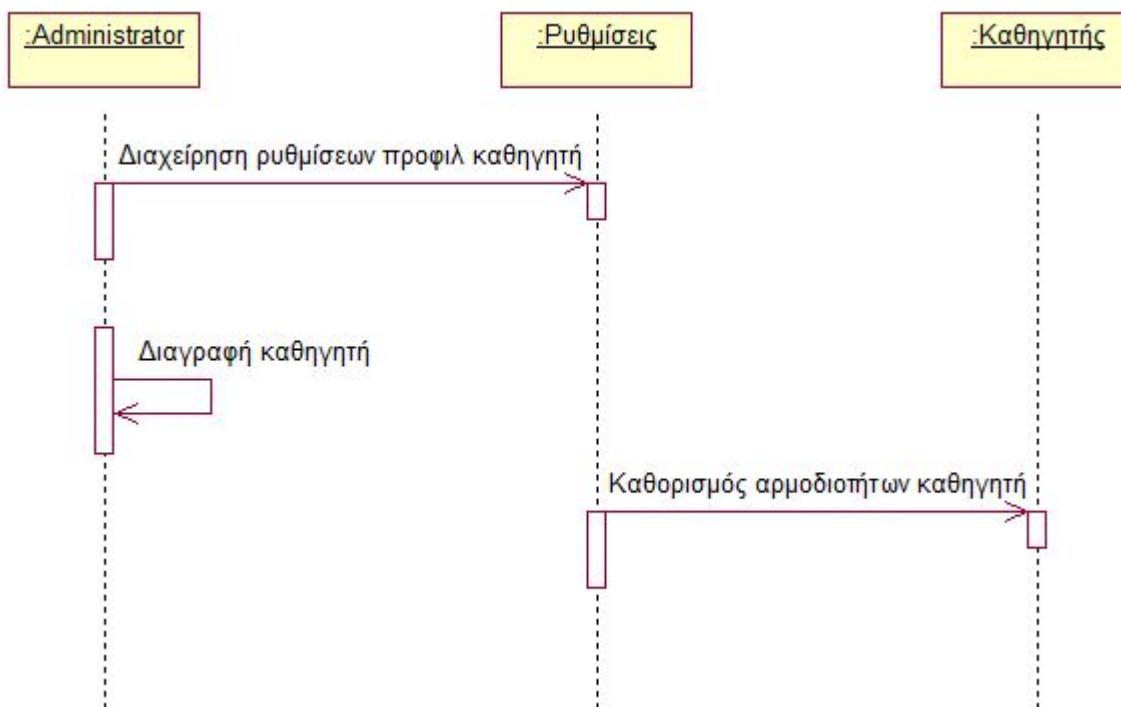


2. Σε αυτό το διάγραμμα αναπαριστάται η δυνατότητα του administrator να διαχειρίζεται τις ρυθμίσεις των μαθημάτων. Στη συνέχεια το μάθημα δίδεται στους μαθητές, οι οποίοι το παρακολουθούν και κατά τη διάρκεια της παρακολούθησης επιβλέπονται και καθοδηγούνται από τον καθηγητή. Παρακάτω παρατίθεται το διάγραμμα σειράς που σχεδιάστηκε με το Rational Rose:

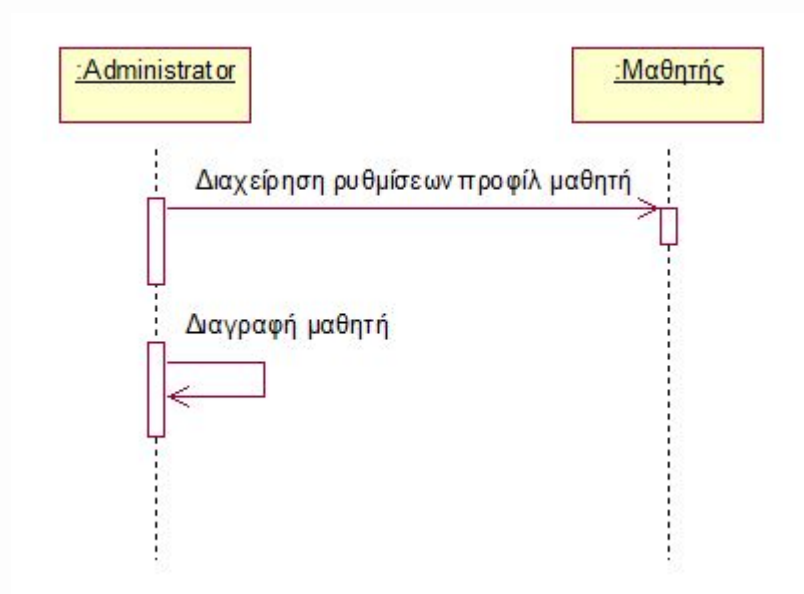




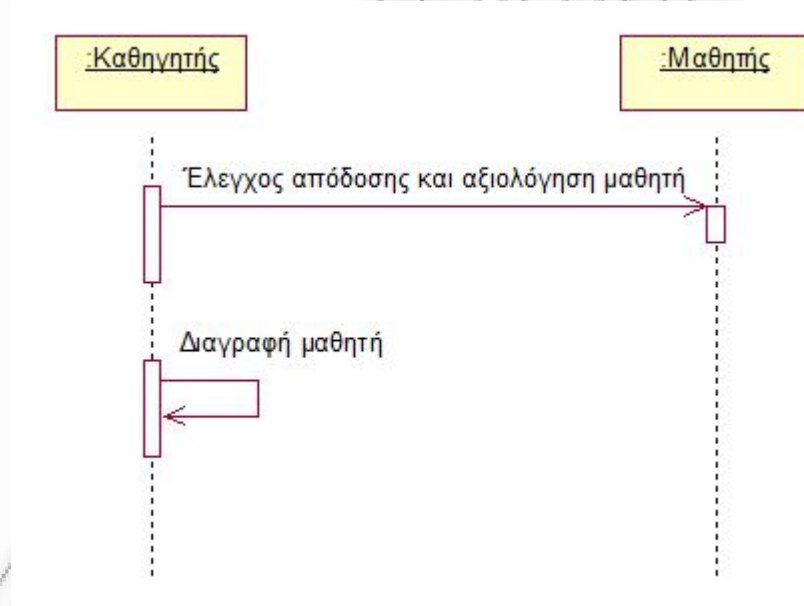
3. Σε αυτό το διάγραμμα αναπαριστάται η δυνατότητα του administrator να διαχειρίζεται τις ρυθμίσεις στα προφίλ των καθηγητών. Ο administrator διαχειρίζεται τις ρυθμίσεις που αφορούν τα προφίλ των καθηγητών, τα οποία με τη σειρά τους καθορίζουν τις αρμοδιότητες των καθηγητών. Παρακάτω παρατίθεται το διάγραμμα σειράς που σχεδιάστηκε με το Rational Rose:



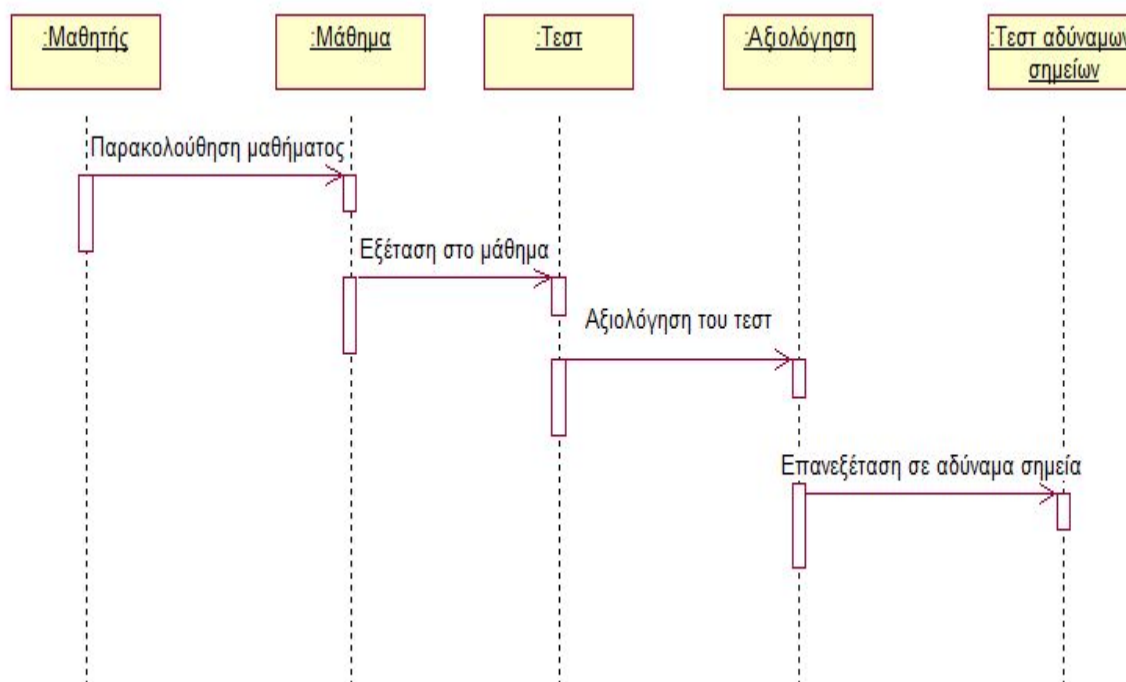
4. Σε αυτό το διάγραμμα αναπαριστάται η δυνατότητα του administrator να διαχειρίζεται τις ρυθμίσεις στα προφίλ των μαθητών. Ο administrator διαχειρίζεται τις ρυθμίσεις που αφορούν τα προφίλ των μαθητών, τα οποία με τη σειρά τους καθορίζουν τις αρμοδιότητες των μαθητών. Παρακάτω παρατίθεται το διάγραμμα σειράς που σχεδιάστηκε με το Rational Rose:



5. Σε αυτό το διάγραμμα σειράς προσδιορίζονται οι αλληλεπιδράσεις μεταξύ του καθηγητή και του μαθητή:



6. Σε αυτό το διάγραμμα σειράς προσδιορίζονται όλες οι αλληλεπιδράσεις μεταξύ του μαθητή, των μαθημάτων και των τεστ:



### 3.3.5 ΔΙΑΓΡΑΜΜΑΤΑ ΚΑΤΑΣΤΑΣΕΩΝ

#### A. Γενικά

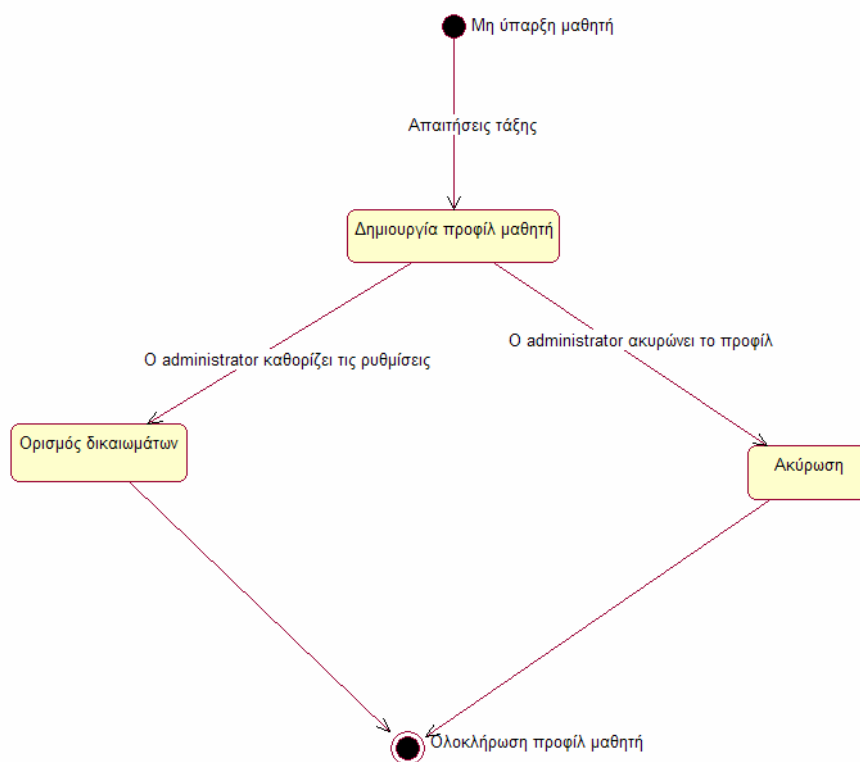
Τα διαγράμματα καταστάσεων αναπαριστούν μηχανές καταστάσεων από την άποψη των καταστάσεων και των μεταβάσεων. Κάθε αντικείμενο είναι σε μία συγκεκριμένη κατάσταση σε μία δεδομένη χρονική στιγμή. Οι καταστάσεις αναπαριστώνται σαν στρογγυλοποιημένα ορθογώνια και χαρακτηρίζονται από τις έννοιες της διάρκειας και της σταθερότητας. Ένα αντικείμενο δεν είναι πάντα στην ίδια κατάσταση σε μία δεδομένη χρονική στιγμή, και ένα αντικείμενο δεν μπορεί να είναι σε μία άγνωστη ή μη ορισμένη κατάσταση. Σε ένα διάγραμμα καταστάσεων πρέπει πάντα να περιγράψουμε την αρχική κατάσταση η οποία είναι πάντα μόνο μία και αναπαρίσταται με μία μεγάλη μαύρη τελεία. Επιπλέον, πρέπει να περιγράψουμε και την τελική κατάσταση που μπορεί να μην είναι μόνο μία αν για παράδειγμα το σύστημα δεν σταματά ποτέ και παριστάνεται με μία μεγάλη μαύρη τελεία που την περιβάλλει ένας κύκλος.

Όταν αναπτύσσονται δυναμικές συνθήκες, τα αντικείμενα αλλάζουν κατάσταση. Τα διαγράμματα καταστάσεων είναι κατευθυνόμενα διαγράμματα και οι καταστάσεις συνδέονται μέσω συνδέσεων μονής κατεύθυνσης που καλούνται μεταβάσεις. Το πέρασμα από τη μία κατάσταση στην άλλη εκτελείται όταν μία μετάβαση ενεργοποιείται από ένα γεγονός που συμβαίνει μέσα στο πεδίο του προβλήματος.

#### B. Διαγράμματα Καταστάσεων Της Εφαρμογής

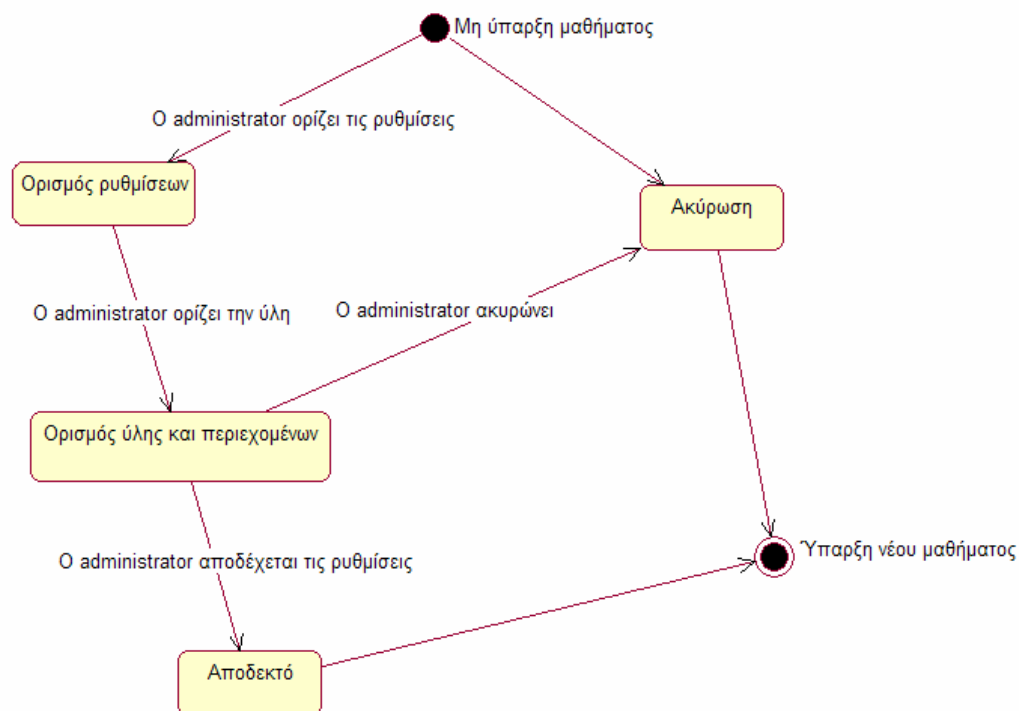
Εδώ παρατίθεται ένα διάγραμμα καταστάσεων που περιγράφει τις καταστάσεις στις οποίες μπορεί να βρεθεί η διαμόρφωση ενός προφίλ μαθητή.

Αρχικά το προφίλ του μαθητή δεν υφίσταται, όμως μετά από απαίτηση για νέο μαθητή δημιουργείται. Το προφίλ του μαθητή ελέγχεται και καθορίζεται από το διαχειριστή του συστήματος. Εάν ο διαχειριστής καθορίσει τα δικαιώματα του μαθητή, το προφίλ θεωρείται αποδεκτό, εάν όμως ο διαχειριστής κρίνει ότι τα στοιχεία του μαθητή είναι μη επαρκή, απορρίπτει τη αίτηση για εγγραφή του μαθητή και την καθιστά μη αποδεκτή. Στη συνέχεια, το προφίλ του μαθητή περνά στην κατάσταση της ακύρωσης. Εάν όμως η αίτηση είναι αποδεκτή ο διαχειριστής του συστήματος την εκτελεί και το σύστημα τερματίζει την διαδικασία κλείνοντας τον κύκλο με την ολοκλήρωση του προφίλ του νέου μαθητή.



Στη συνέχεια, παρατίθεται το διάγραμμα καταστάσεων που αναφέρεται στην δημιουργία νέου μαθήματος.

Αρχικά το μάθημα δεν υφίσταται, όμως μετά από απαίτηση για νέο μάθημα δημιουργείται. Το νέο μάθημα ελέγχεται και καθορίζεται από το διαχειριστή του συστήματος. Εάν ο διαχειριστής καθορίσει τις ρυθμίσεις, την ύλη και τα περιεχόμενα του μαθήματος, το μάθημα θεωρείται αποδεκτό, εάν όμως ο διαχειριστής κρίνει ότι τα στοιχεία του μαθήματος είναι μη επαρκή, απορρίπτει τη αίτηση για νέο μάθημα και το καθιστά μη αποδεκτό. Στη συνέχεια, το μάθημα περνά στην κατάσταση της ακύρωσης. Εάν όμως η αίτηση είναι αποδεκτή ο διαχειριστής του συστήματος την εκτελεί και το σύστημα τερματίζει την διαδικασία κλείνοντας τον κύκλο με την ολοκλήρωση του νέου μαθήματος.



### 3.3.6 ΔΙΑΓΡΑΜΜΑΤΑ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ

#### A. Γενικά

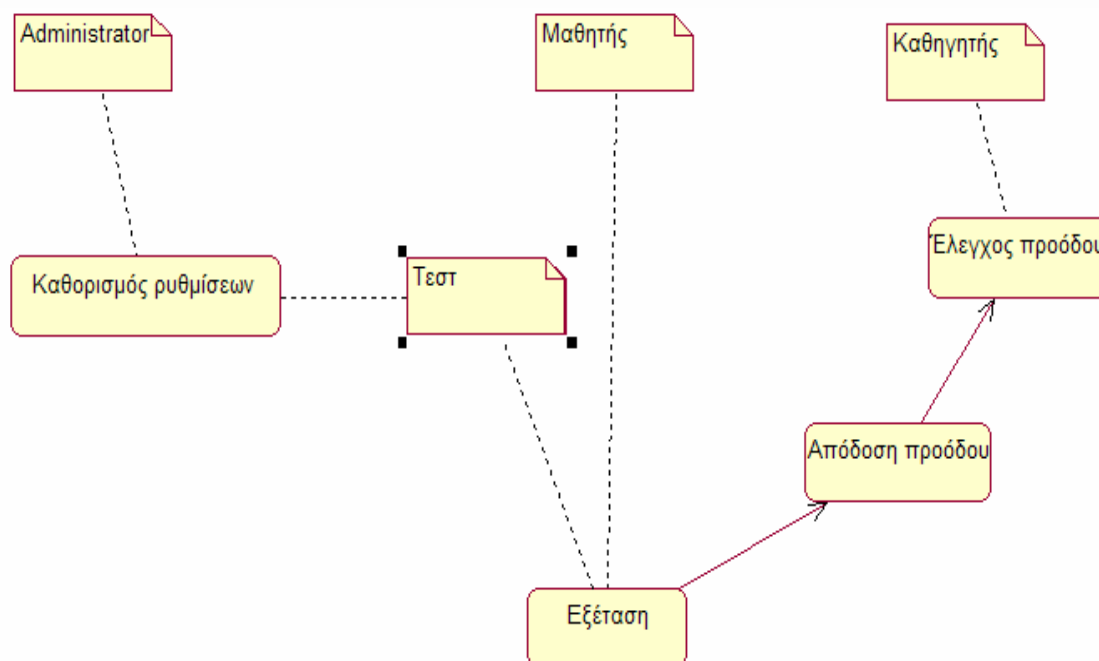
Τα διαγράμματα δραστηριοτήτων είναι μία παραλλαγή των διαγραμμάτων καταστάσεων οργανωμένα σύμφωνα με ενέργειες και στοχεύουν κυρίως στην αναπαράσταση της εσωτερικής συμπεριφοράς μίας μεθόδου ή μίας περίπτωσης χρήσης.

Ένα διάγραμμα δραστηριοτήτων αναπαριστά την κατάσταση εκτέλεσης ενός μηχανισμού σαν μία σειρά βημάτων που ομαδοποιούνται σειριακά σαν παράλληλες διακλαδώσεις ελέγχου.

Όπως και στα διαγράμματα καταστάσεων κάθε δραστηριότητα αναπαριστά μία συγκεκριμένη κατάσταση που παριστάνεται με ένα στρογγυλοποιημένο ορθογώνιο. Οι δραστηριότητες συνδέονται με αυτόματες μεταβάσεις που αναπαριστώνται με βέλη. Κάθε φορά που τερματίζει μία δραστηριότητα, ενεργοποιείται η μετάβαση και ξεκινά η επόμενη δραστηριότητα.

#### B. Διαγράμματα Δραστηριοτήτων Της Εφαρμογής

Πιο κάτω παρουσιάζεται ένα διάγραμμα δραστηριοτήτων που χωρίζεται σε λωρίδες κολύμβησης και δείχνει πως πολλές δραστηριότητες διαχειρίζονται το αντικείμενο τεστ.



Σε αυτό το διάγραμμα αναπαριστάται η δυνατότητα του administrator να διαχειρίζεται τις ρυθμίσεις των τεστ, οι οποίες καθορίζουν το τεστ. Στη συνέχεια το τεστ δίδεται στους μαθητές, οι οποίοι εξετάζονται και αξιολογείται η απόδοσή τους από τον καθηγητή.

### 3.3.7 ΔΙΑΓΡΑΜΜΑΤΑ ΕΞΑΡΤΗΜΑΤΩΝ

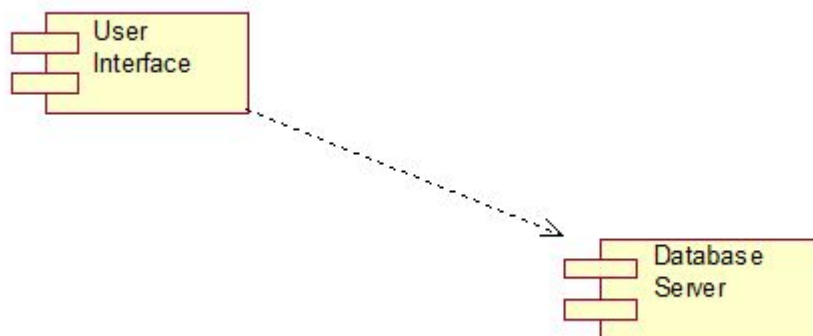
#### A. Γενικά

Τα διαγράμματα εξαρτημάτων περιγράφουν κομμάτια λογισμικού και τη σχέση τους με το περιβάλλον υλοποίησης. Υποδηλώνουν τις επιλογές που γίνονται κατά τη χρονική στιγμή υλοποίησης.

Τα εξαρτήματα αναπαριστούν όλα τα είδη των στοιχείων που αναφέρονται στην τμηματοποίηση των εφαρμογών λογισμικού. Τα εξαρτήματα μπορεί να περιλαμβάνουν εκτελέσιμα αρχεία και βιβλιοθήκες πηγαίου κώδικα που «διαβάζονται» και χρησιμοποιούνται δυναμικά.

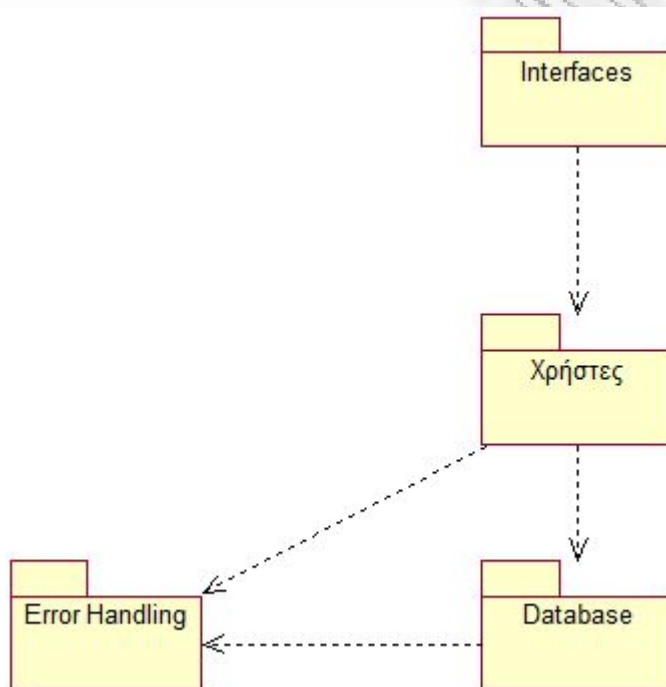
#### B. Διαγράμματα Εξαρτημάτων Της Εφαρμογής

Στην εφαρμογή που μελετάμε έχουμε δύο κομμάτια λογισμικού, την εφαρμογή του χρήστη και τη Βάση Δεδομένων. Δεδομένου ότι η εφαρμογή του χρήστη παίρνει δεδομένα από τη βάση δεδομένων, το διάγραμμα εξαρτημάτων που προκύπτει είναι το ακόλουθο.



Για να επιτύχουμε την υλοποίηση των εφαρμογών, διάφορα εξαρτήματα μπορούν να ομαδοποιηθούν μέσα σε πακέτα σύμφωνα με λογικά κριτήρια. Συχνά στερεοτυπούνται σαν υποσυστήματα για να προσθέσουμε τις έννοιες των βιβλιοθηκών μεταγλώττισης και της διαχείρισης σύνθεσης τμημάτων στην σημασιολογία της τμηματοποίησης που περιλαμβάνεται ήδη μέσα στα πακέτα.

Το μοντέλο που περιγράφει συνολικά το σύστημα και επικεντρώνεται στα πακέτα εξαρτημάτων που είναι απαραίτητο να δημιουργηθούν για την υλοποίηση του συστήματος. Στην εφαρμογή «e-School» αναπαριστάται ως εξής:



### 3.3.8 ΔΙΑΓΡΑΜΜΑΤΑ ΔΙΑΝΟΜΗΣ

#### A. Γενικά

Τα διαγράμματα διανομής δείχνουν τη φυσική διάταξη των διαφόρων εξαρτημάτων (κόμβων) υλικού, που αποτελούν ένα σύστημα, όπως και τη διανομή των εκτελέσιμων προγραμμάτων σε αυτό το υλικό.

Κάθε πόρος υλικού αναπαρίσταται με ένα μικρό κόμβο. Κάθε σύστημα μπορεί να περιγραφεί με ένα μικρό αριθμό διαγραμμάτων διανομής. Τις περισσότερες φορές ένα μόνο διάγραμμα περιγράφει ένα δεδομένο σύστημα.

Ένα διάγραμμα διανομής περιλαμβάνει επεξεργαστές, συσκευές, διαδικασίες και συνδέσεις μεταξύ επεξεργαστών και συσκευών. Δηλαδή το διάγραμμα διανομής αναπαριστά όλους τους κόμβους του δικτύου, δηλαδή τους επεξεργαστές και τις συσκευές, που αντιπροσωπεύει το σύστημα, τις συνδέσεις μεταξύ τους και τις διαδικασίες που μπορεί να τρέχουν σε κάθε μία.

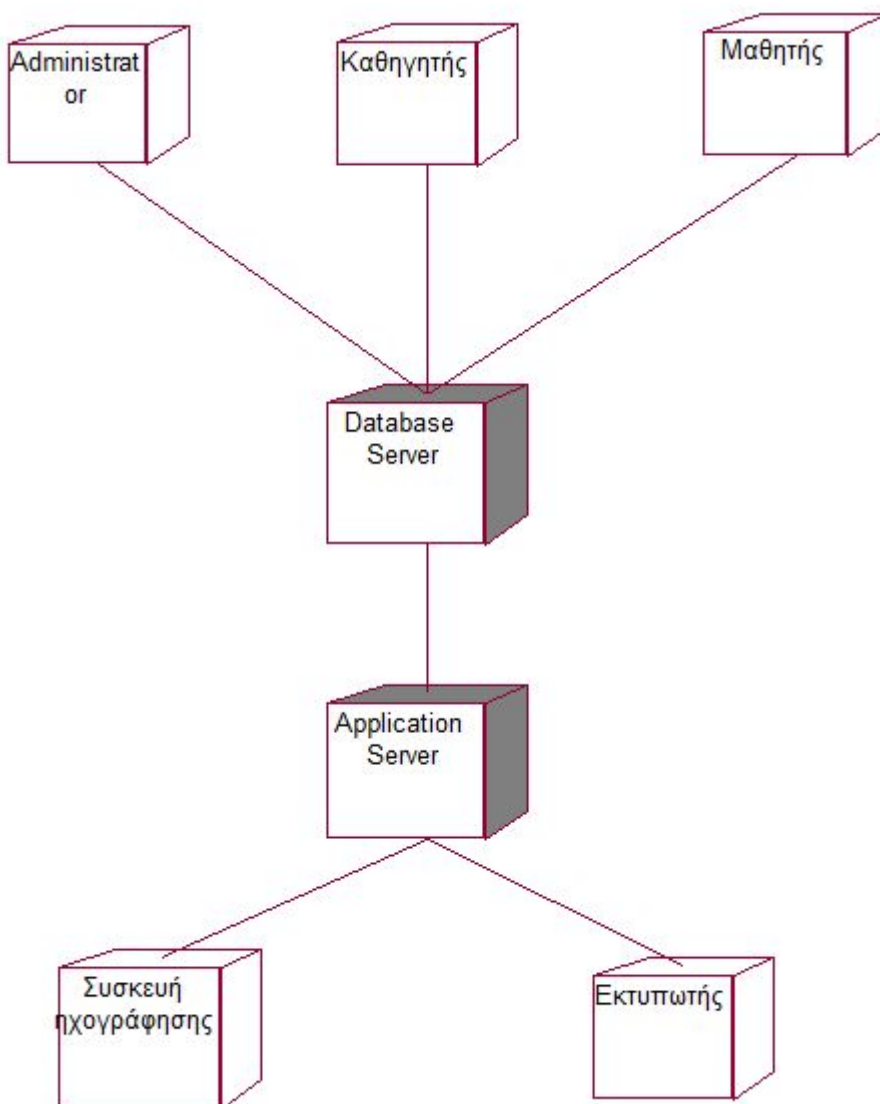
Επεξεργαστής είναι κάθε μηχάνημα που έχει δυνατότητα επεξεργασίας. Οι servers, οι σταθμοί εργασίας και άλλες μηχανές με επεξεργαστές ανήκουν σ' αυτή την κατηγορία. Μία σύνδεση αντιπροσωπεύει τη φυσική σύνδεση μεταξύ δύο επεξεργαστών, δύο συσκευών, ή ενός επεξεργαστή και μίας συσκευής. Συνήθως οι συνδέσεις αναπαριστούν τις φυσικές συνδέσεις δικτύου μεταξύ των κόμβων του δικτύου. Μία σύνδεση μπορεί επίσης να είναι μία σύνδεση Internet μεταξύ δύο κόμβων.

Μία διαδικασία είναι ένα απλό νήμα εκτέλεσης που τρέχει σε ένα επεξεργαστή. Οι διαδικασίες μπορούν είτε να παρουσιάζονται στο διάγραμμα είτε να είναι κρυμμένες. Αν είναι ορατές, τότε εμφανίζονται σε λίστα κάτω από τον επεξεργαστή ή τους επεξεργαστές πάνω στους οποίους τρέχουν. Στις διαδικασίες μπορεί να ανατίθεται μία προτεραιότητα. Αν χρησιμοποιείται προεκχωρητικός χρονοπρογραμματισμός, η προτεραιότητα της διαδικασίας θα καθορίσει πότε μπορεί να εκτελεστεί.

## B. Διαγράμματα Διανομής Της Εφαρμογής

Το παρακάτω διάγραμμα δείχνει την διάταξη του συστήματος της εφαρμογής «e-School». Το σύστημα βάσεων δεδομένων, που περιλαμβάνει τα στοιχεία των καθηγητών, των μαθητών και του administrator που λαμβάνουν χώρα στην εφαρμογή, τροφοδοτεί τον server της εφαρμογής «e-School» που προορίζεται για τους χρήστες της. Τέλος, ο server της εφαρμογής συνδέεται με ένα τερματικό, και συγκεκριμένα με ένα εκτυπωτή και με μία συσκευή ηχογράφησης, μέσω των οποίων εξυπηρετούνται οι ανάγκες της εφαρμογής.





### 3.3.9 ΔΙΑΓΡΑΜΜΑΤΑ ΠΕΡΙΠΤΩΣΕΩΝ ΧΡΗΣΗΣ

#### A. Γενικά

Οι περιπτώσεις χρήσης περιγράφουν τη συμπεριφορά ενός συστήματος από την οπτική γωνία του χρήστη. Επιτρέπουν τον ορισμό των ορίων του συστήματος και του περιβάλλοντος. Μία περίπτωση χρήσης αντιστοιχεί σε ένα συγκεκριμένο είδος χρήσης του συστήματος. Είναι μια εικόνα της λειτουργικότητας ενός συστήματος, το οποίο ενεργοποιείται για να ανταποκριθεί σε ένα εξωτερικό ενεργοποιό.

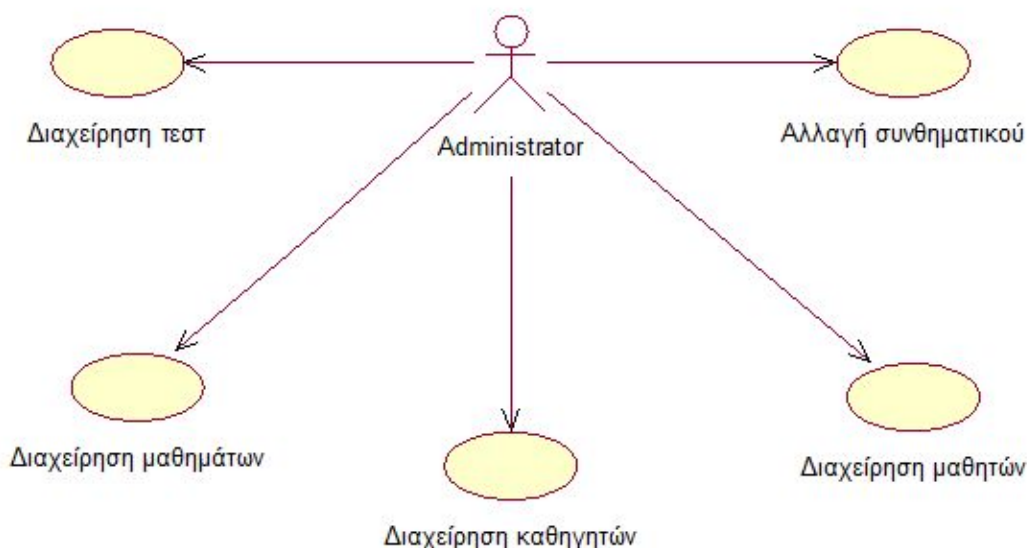
Το μοντέλο περιπτώσεων χρήσης περιλαμβάνει τους ενεργοποιούς και τις ίδιες τις περιπτώσεις χρήσης. Το σύνολο της λειτουργικότητας ενός δεδομένου συστήματος καθορίζεται μέσω της εφαρμογής των λειτουργικών απαιτήσεων κάθε ενεργοποιού. Αυτές εκφράζονται με τις περιπτώσεις χρήσης σε μορφή οικογενειακών διεπαφών.

## B. Διαγράμματα Περιπτώσεων Χρήσης Της Εφαρμογής

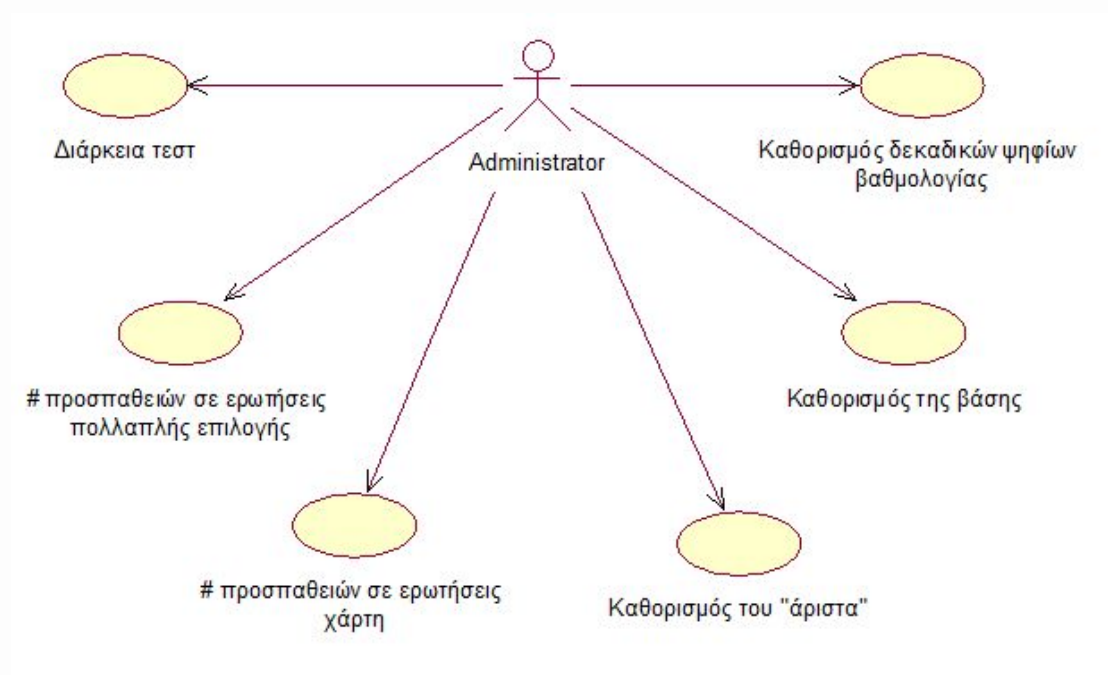
Στην εφαρμογή των μαθημάτων της εφαρμογής e-School, Γεωγραφία και Ιστορία, υπάρχουν τρεις ενεργοποιοί, ο administrator, ο καθηγητής και ο μαθητής. Παρακάτω παρατίθενται τα διαγράμματα περιπτώσεων χρήσης που σχεδιάστηκαν με το Rational Rose :

Στα έντεκα πρώτα γενικευμένα διαγράμματα περιπτώσεων χρήσης φαίνονται οι πράξεις που μπορεί να εκτελέσει ο administrator, ο καθηγητής και ο μαθητής χρησιμοποιώντας μόνο σχέσεις μεταξύ ενεργοποιών και περιπτώσεων χρήσης.

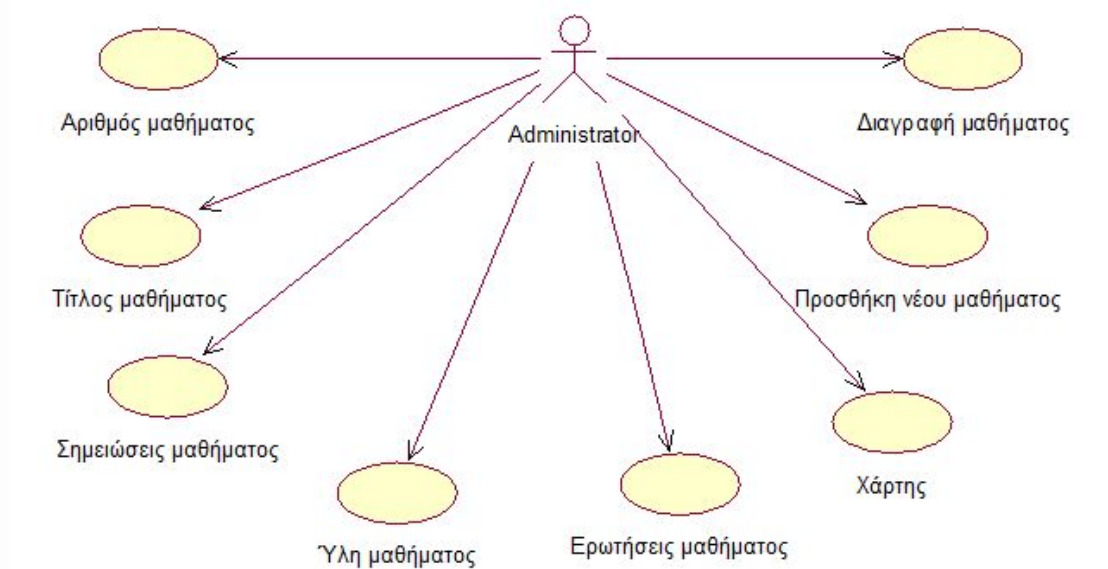
1. Στο παρακάτω διάγραμμα περιπτώσεων χρήσης φαίνονται οι αρμοδιότητες του administrator. Ο administrator έχει τη δυνατότητα να διαχειρίζεται όλες τις ρυθμίσεις των τεστ (διάρκεια του τεστ, αριθμός προσπαθειών σε ερωτήσεις πολλαπλής επιλογής, αριθμός προσπαθειών σε ερωτήσεις χάρτη, καθορισμός του άριστα και της βάσης, αριθμός δεκαδικών ψηφίων στην βαθμολογία), τα μαθήματα (τίτλος μαθήματος, σημειώσεις μαθήματος, ύλη μαθήματος, ερωτήσεις, χάρτης, εισαγωγή νέου μαθήματος, διαγραφή μαθήματος), τα προφίλ των καθηγητών (καθορισμών δικαιωμάτων, διαγραφή συνηματικού καθηγητή, εισαγωγή νέου, διαγραφή καθηγητή), τους μαθητές (επίβλεψη της αξιολόγησής τους, διαγραφή συνηματικού μαθητή, εισαγωγή νέου, διαγραφή μαθητή) και να αλλάζει το συνηματικό του. Η παραπάνω διαδικασία φαίνεται στο παρακάτω διάγραμμα:



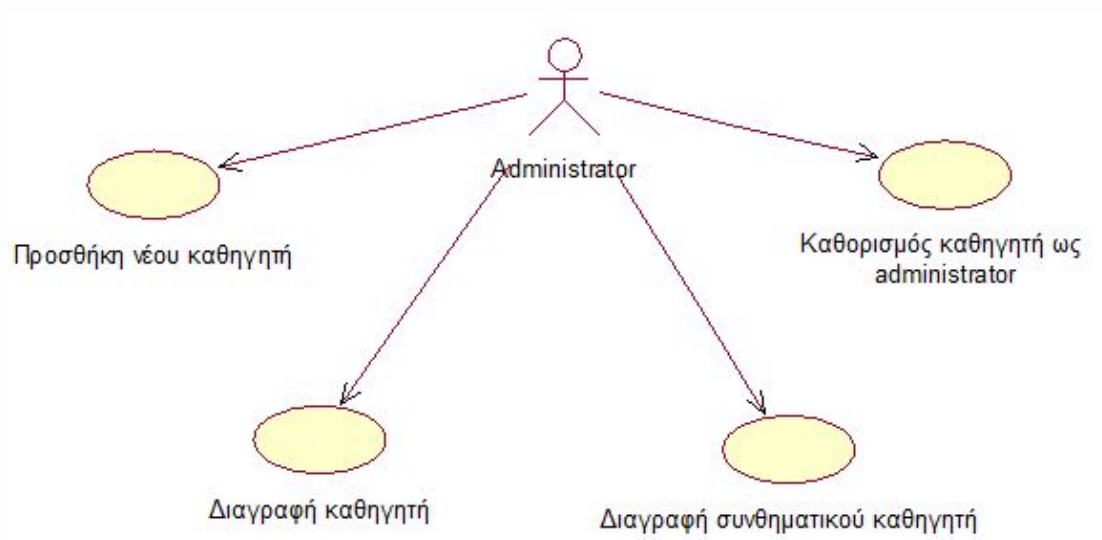
2 Στο παρακάτω διάγραμμα περιπτώσεων χρήσης φαίνονται οι αρμοδιότητες του administrator σε σχέση με τη διαχείριση των ρυθμίσεων των τεστ (διάρκεια του τεστ, αριθμός προσπαθειών σε ερωτήσεις πολλαπλής επιλογής, αριθμός προσπαθειών σε ερωτήσεις χάρτη, καθορισμός του άριστα και της βάσης, αριθμός δεκαδικών ψηφίων στην βαθμολογία). Η παραπάνω διαδικασία φαίνεται στο παρακάτω διάγραμμα:



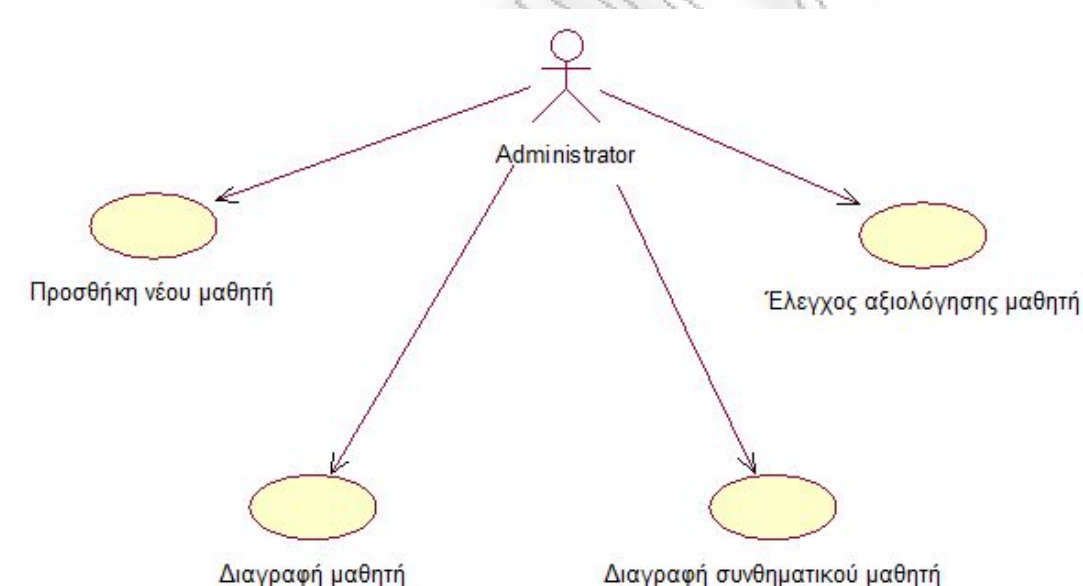
3. Στο παρακάτω διάγραμμα περιπτώσεων χρήσης φαίνονται οι αρμοδιότητες του administrator σε σχέση με τη διαχείριση των μαθημάτων (τίτλος μαθήματος, σημειώσεις μαθήματος, ύλη μαθήματος, ερωτήσεις, χάρτης, εισαγωγή νέου μαθήματος, διαγραφή μαθήματος). Η παραπάνω διαδικασία φαίνεται στο παρακάτω διάγραμμα:



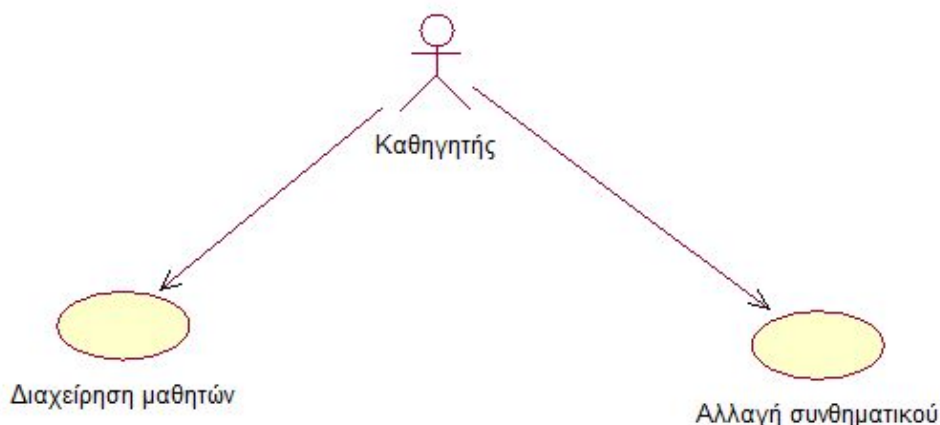
4. Στο παρακάτω διάγραμμα περιπτώσεων χρήσης φαίνονται οι αρμοδιότητες του administrator σε σχέση με τη διαχείριση των προφίλ των καθηγητών (καθορισμών δικαιωμάτων, διαγραφή συνθηματικού καθηγητή, εισαγωγή νέου, διαγραφή καθηγητή). Η παραπάνω διαδικασία φαίνεται στο παρακάτω διάγραμμα:



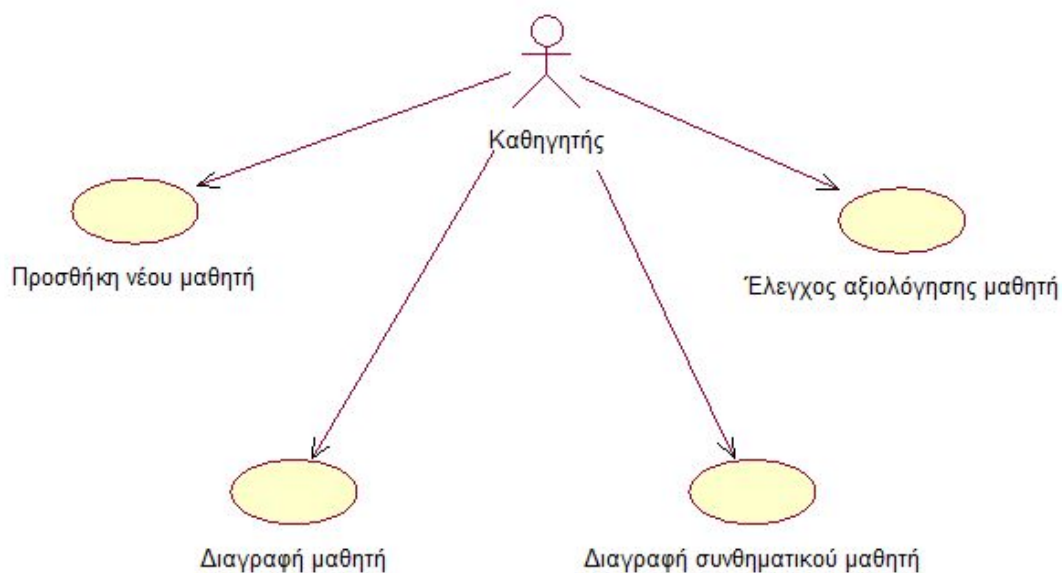
5. Στο παρακάτω διάγραμμα περιπτώσεων χρήσης φαίνονται οι αρμοδιότητες του administrator σε σχέση με τη διαχείριση των μαθητών (επίβλεψη της αξιολόγησής τους, διαγραφή συνηματικού μαθητή, εισαγωγή νέου, διαγραφή μαθητή). Η παραπάνω διαδικασία φαίνεται στο παρακάτω διάγραμμα:



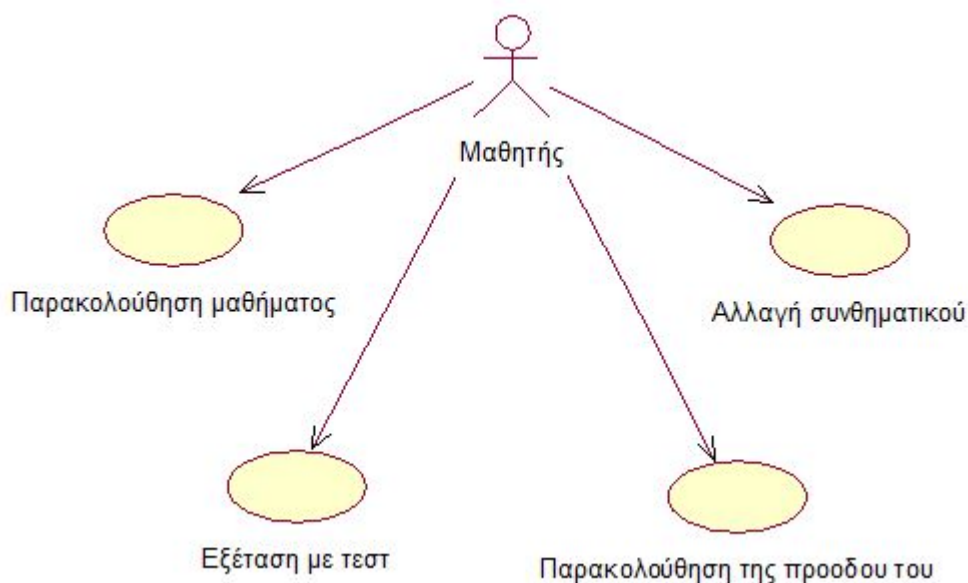
6. Παρακάτω παρατίθεται ένα διάγραμμα περιπτώσεων χρήσης, όπου φαίνονται οι αρμοδιότητες του καθηγητή. Ο καθηγητής έχει τη δυνατότητα να διαχειρίζεται τους μαθητές (επίβλεψη της αξιολόγησής τους, διαγραφή συνηματικού μαθητή, εισαγωγή νέου, διαγραφή μαθητή) και να αλλάζει το συνηματικό του. Η παραπάνω διαδικασία φαίνεται στο παρακάτω διάγραμμα:



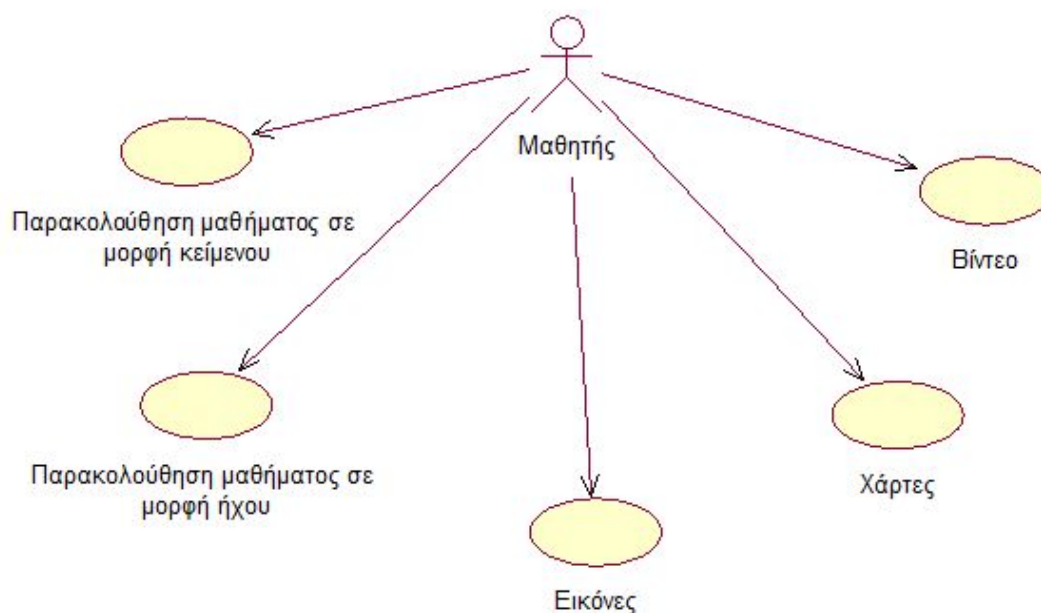
7. Παρακάτω παρατίθεται ένα διάγραμμα περιπτώσεων χρήσης, όπου φαίνονται οι αρμοδιότητες του καθηγητή σε σχέση με τη διαχείριση των μαθητών (επίβλεψη της αξιολόγησής τους, διαγραφή συνθηματικού μαθητή, εισαγωγή νέου, διαγραφή μαθητή)



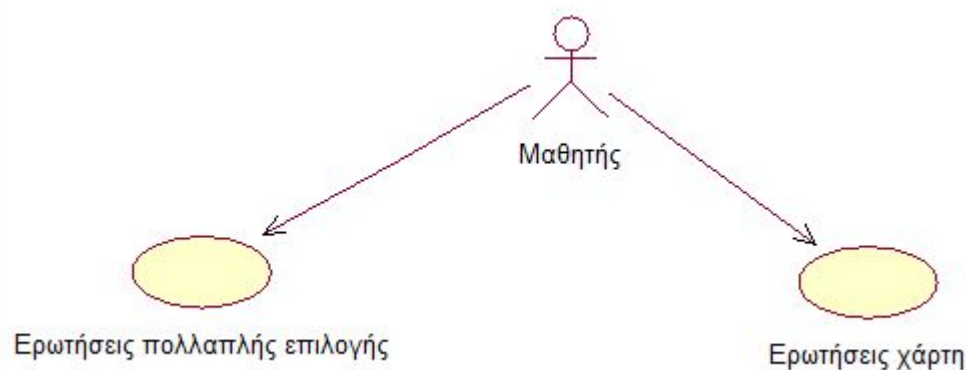
8. Στο παρακάτω διάγραμμα περιπτώσεων χρήσης φαίνονται οι αρμοδιότητες του μαθητή. Ο μαθητής έχει τη δυνατότητα να επιλέξει το μάθημα που επιθυμεί να παρακολουθήσει, να επιλέξει το τεστ που επιθυμεί να εξεταστεί, να παρακολουθεί την αξιολόγησή του και να αλλάξει το συνθηματικό του. Η παραπάνω διαδικασία φαίνεται στο παρακάτω διάγραμμα:



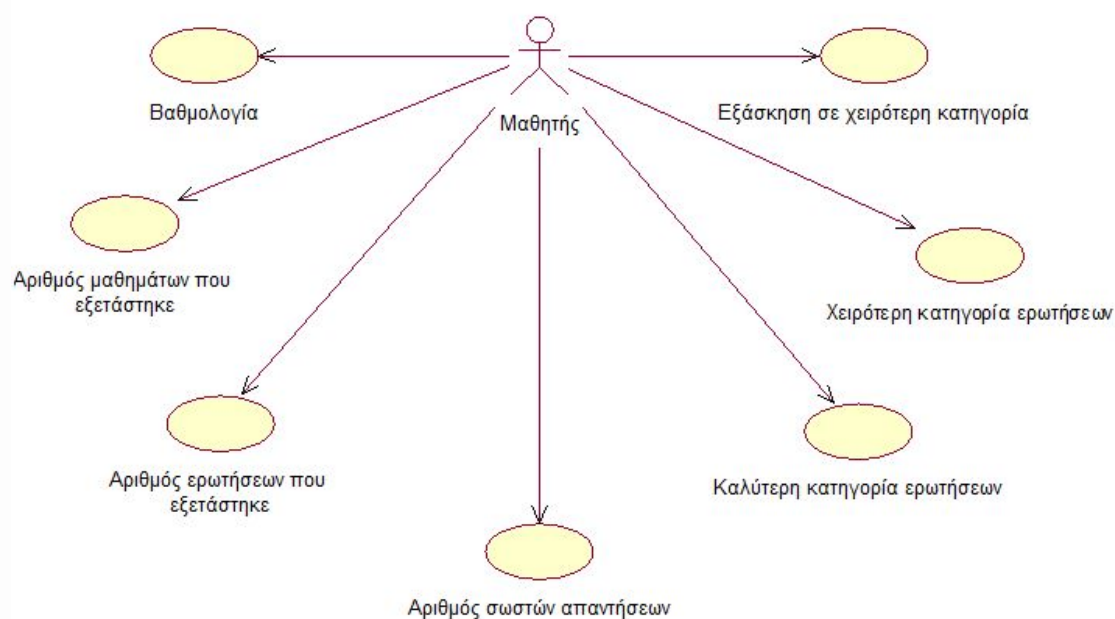
9. Στο παρακάτω διάγραμμα περιπτώσεων χρήσης περιγράφονται οι δυνατότητες που έχει ο μαθητής με την επιλογή κάποιου μαθήματος. Η διαδικασία που ακολουθείται είναι να γίνει η επιλογή του μαθήματος από τον μαθητή και στη συνέχεια μπορεί να επιλέξει εάν επιθυμεί να διδαχτεί το μάθημα διαβάζοντας το γραπτό κείμενο, ακούγοντάς το ή κάνοντας και τα δύο. Όποιον τρόπο και να επιλέξει, θα έχει στην ευχέρειά του πλήθος εικόνων, βίντεο και χαρτών που πλαισιώνουν το μάθημα, καθώς και το τεστ που μπορεί να εξεταστεί στο αντίστοιχο κεφάλαιο. Η παραπάνω διαδικασία φαίνεται στο παρακάτω διάγραμμα:



10. Στο παρακάτω διάγραμμα περιπτώσεων χρήσης περιγράφονται οι δυνατότητες που έχει ο μαθητής με την επιλογή κάποιου τεστ. Η διαδικασία που ακολουθείται είναι να γίνει η επιλογή του τεστ από τον μαθητή και στη συνέχεια έχει τη δυνατότητα να εξεταστεί με ερωτήσεις πολλαπλής επιλογής και ερωτήσεις σε χάρτη. Η παραπάνω διαδικασία φαίνεται στο παρακάτω διάγραμμα:

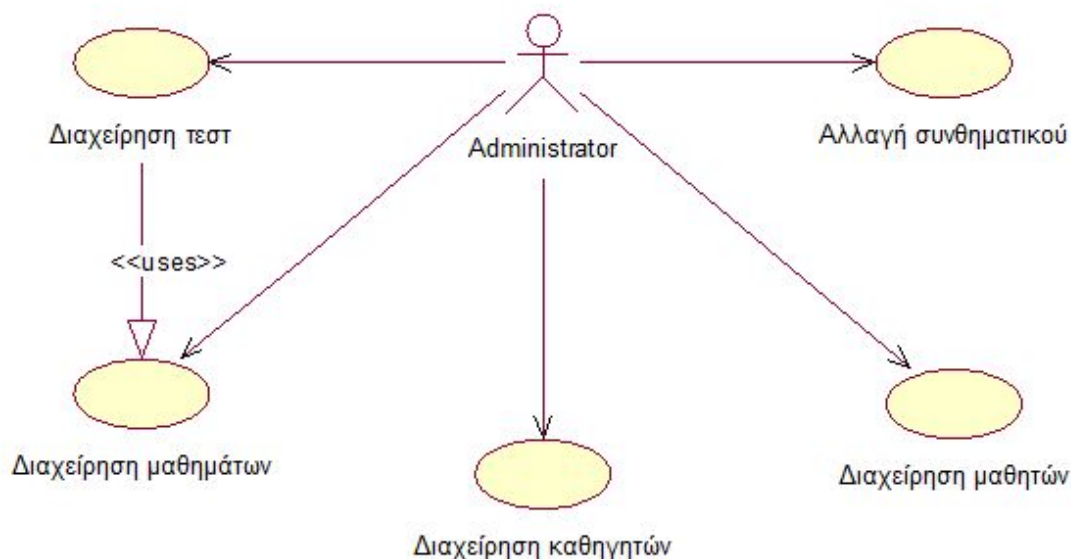


11. Στο παρακάτω διάγραμμα περιπτώσεων χρήσης περιγράφονται οι δυνατότητες που έχει ο μαθητής σε σχέση με την ολοκλήρωση του τεστ. Ο μαθητής μετά το πέρας του τεστ, όπου εξετάστηκε μπορεί να δει την βαθμολογία του, τον αριθμό των μαθημάτων που εξετάστηκε, τον αριθμό των ερωτήσεων που εξετάστηκε, τον αριθμό των σωστών απαντήσεων, την καλύτερη κατηγορία ερωτήσεων, την χειρότερη κατηγορία ερωτήσεων, καθώς επίσης και να κάνει περισσότερη εξάσκηση στην κατηγορία που έχει τη χαμηλότερη βαθμολογία. Η παραπάνω διαδικασία φαίνεται στο παρακάτω διάγραμμα:

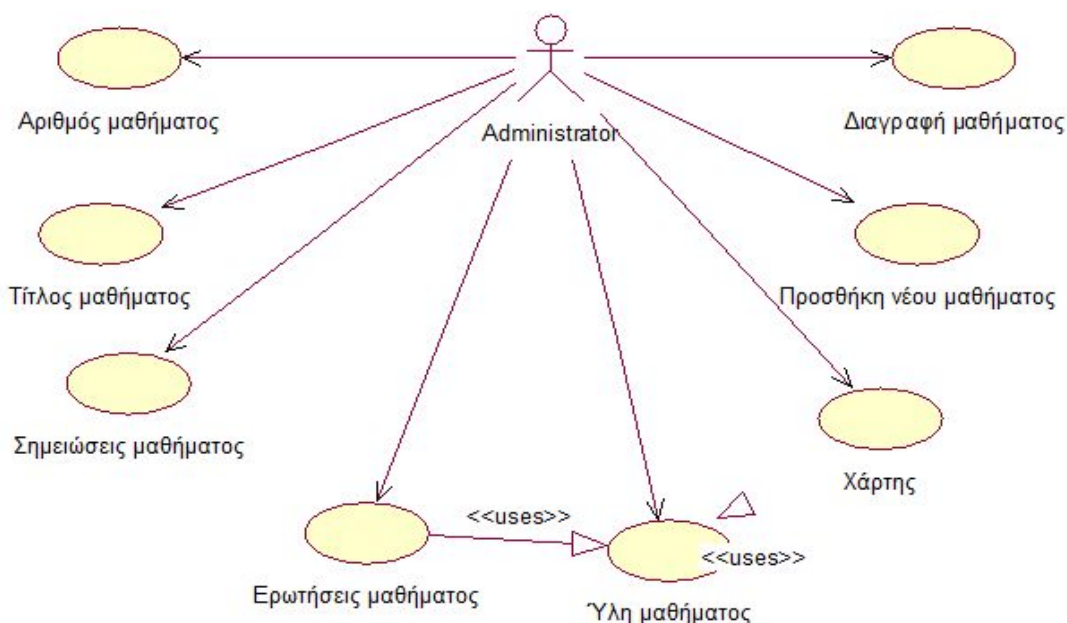


Παρακάτω παρατίθενται τα διαγράμματα χρήσης που παρουσιάζουν μεγαλύτερο βαθμό λεπτομέρειας μέσω της εισαγωγής των σχέσεων <<uses>> και <<extends>>.

1. Η σχέση <<uses>> χρησιμοποιείται για να δηλώσει ότι ο administrator για να διαχειριστεί κάποιο τεστ πρέπει πρώτα να έχει ολοκληρωθεί η διαχείριση του αντίστοιχου μαθήματος.

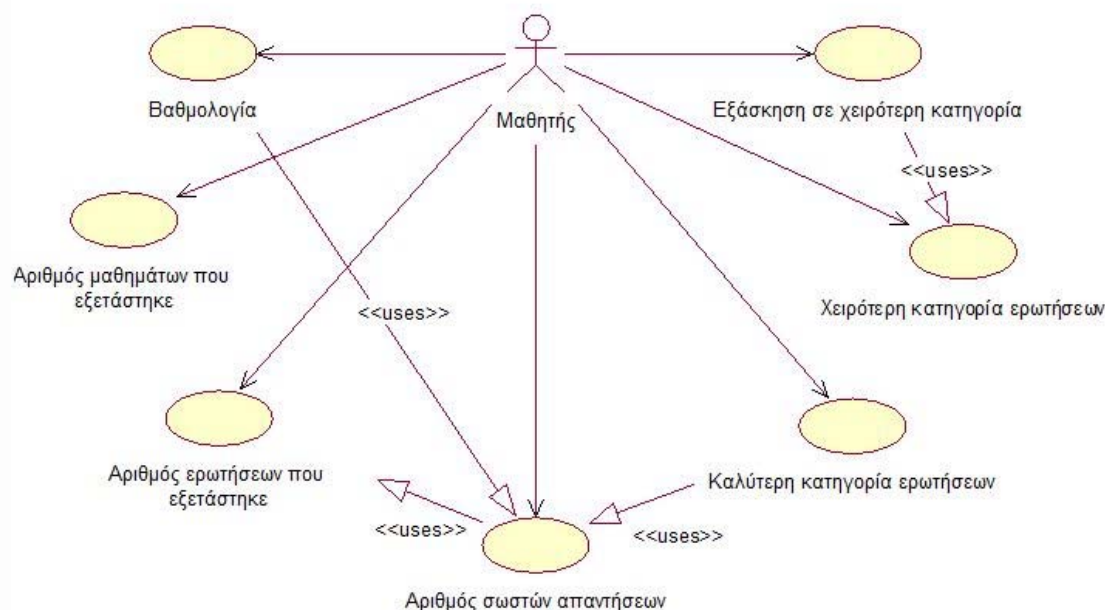


2. Στο παρακάτω διάγραμμα με τη σχέση <<uses>> δηλώνεται πως οι ερωτήσεις κάθε μαθήματος, καθώς και ο χάρτης εξαρτώνται από την ύλη του μαθήματος.



3. Στο παρακάτω διάγραμμα με τη σχέση <<uses>> δηλώνεται πως η βαθμολογία κάθε τεστ εξαρτάται από τον αριθμό των σωστών απαντήσεων κάθε μαθήματος. Επίσης, η καλύτερη κατηγορία ερωτήσεων, δηλαδή η κατηγορία που συγκέντρωσε ο μαθητής την καλύτερη βαθμολογία κατά την εξέτασή του στο τεστ, εξαρτάται από τον αριθμό των σωστών απαντήσεων κάθε μαθήματος. Οι σωστές απαντήσεις εξαρτώνται από τον αριθμό των ερωτήσεων στον οποίο εξετάστηκε ο μαθητής. Τέλος, η εξάσκηση στην χειρότερη κατηγορία εξαρτάται από την κατηγορία των ερωτήσεων με τη χαμηλότερη βαθμολογία. Με βάση τα παραπάνω προκύπτει το ακόλουθο διάγραμμα χρήσης.





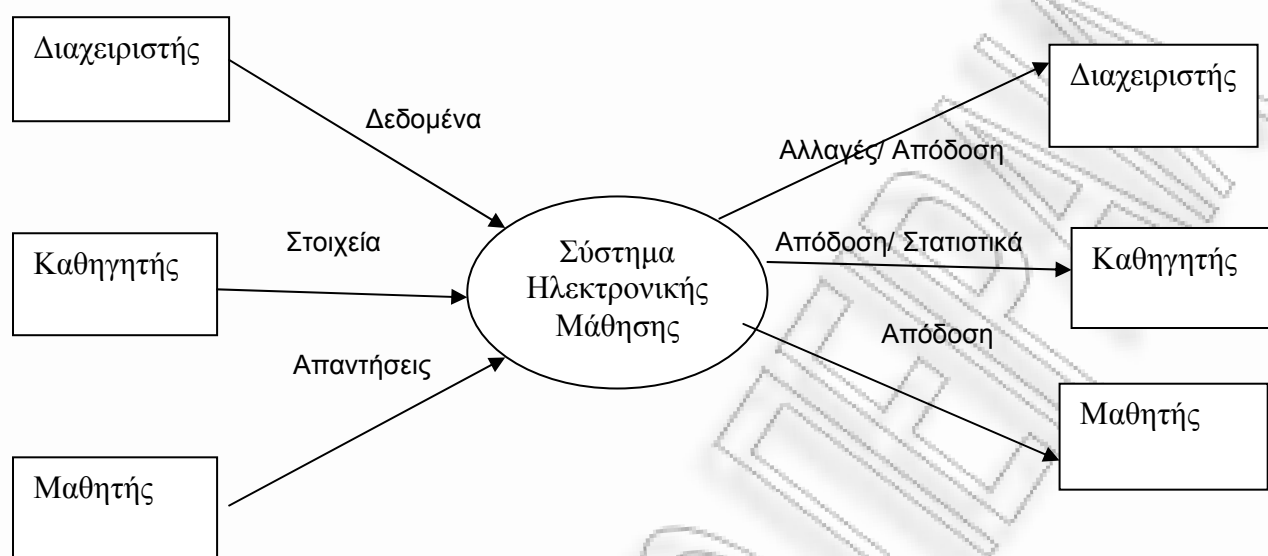
### 3.4 ΑΝΑΛΥΣΗ ΛΕΙΤΟΥΡΓΙΩΝ-ΛΕΙΤΟΥΡΓΙΚΗ ΠΡΟΣΕΓΓΙΣΗ

Η λειτουργική ανάλυση απαιτήσεων θα γίνει με τη χρήση διαγραμμάτων ροής δεδομένων και διαγραμμάτων δομών. Σε αυτό το κεφάλαιο θα γίνει ανάλυση των διαγραμμάτων ροής δεδομένων.

Τα διαγράμματα ροής δεδομένων – ΔΡΔ (Data Flow Diagrams- DFDs) αναπαριστούν ένα σύστημα λογισμικού με βάση τα δεδομένα που παράγονται ή διακινούνται σε αυτό. Αποτελούν μια λογική αναπαράσταση του συστήματος, χωρίς να περιέχουν πληροφορίες για το υλικό, το λογισμικό ή τα αρχεία που το αποτελούν.

Ένα σύστημα μπορεί να αναπαρασταθεί με ΔΡΔ σε οποιοδήποτε επίπεδο λεπτομέρειας. Συνήθως, στο πρώτο βήμα γίνεται αναπαράσταση του συστήματος με ένα ΔΡΔ που έχει μόνο μία διεργασία, στην οποία εισέρχονται οι ροές δεδομένων που προέρχονται από το περιβάλλον του συστήματος και από την οποία εξέρχονται οι ροές δεδομένων προς το περιβάλλον. Το διάγραμμα σε αυτό το επίπεδο ονομάζεται Διάγραμμα Μηδενικού Επιπέδου.

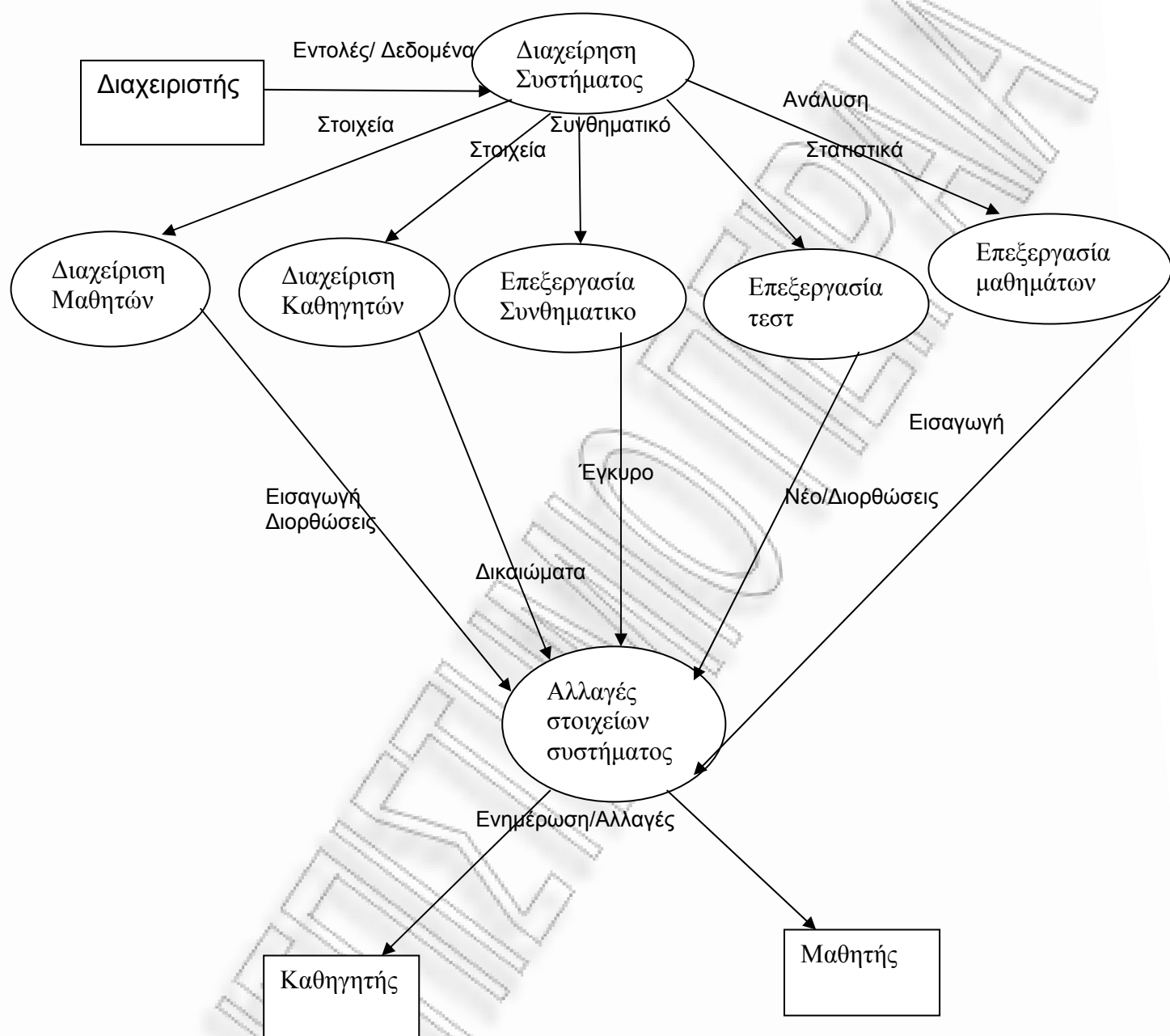
Στην εφαρμογή e-School το σύστημα ηλεκτρονικής μάθησης λαμβάνει πληροφορίες από τον διαχειριστή και τον καθηγητή ώστε να διαμορφωθούν η ύλη των μαθημάτων και τα τεστ, καθώς και η διαχείριση των προφίλ των ατόμων. Επίσης, οι μαθητές εισάγουν δεδομένα πραγματοποιώντας παρακολουθήσεις μαθημάτων και κάνοντας τεστ. Εξάγονται δεδομένα και στατιστικά στοιχεία για την απόδοση των μαθητών προς όλους τους χρήστες, Παρακάτω παρατίθεται το διάγραμμα μηδενικού επιπέδου:



Σχήμα 1.1: Διάγραμμα Μηδενικού Επιπέδου

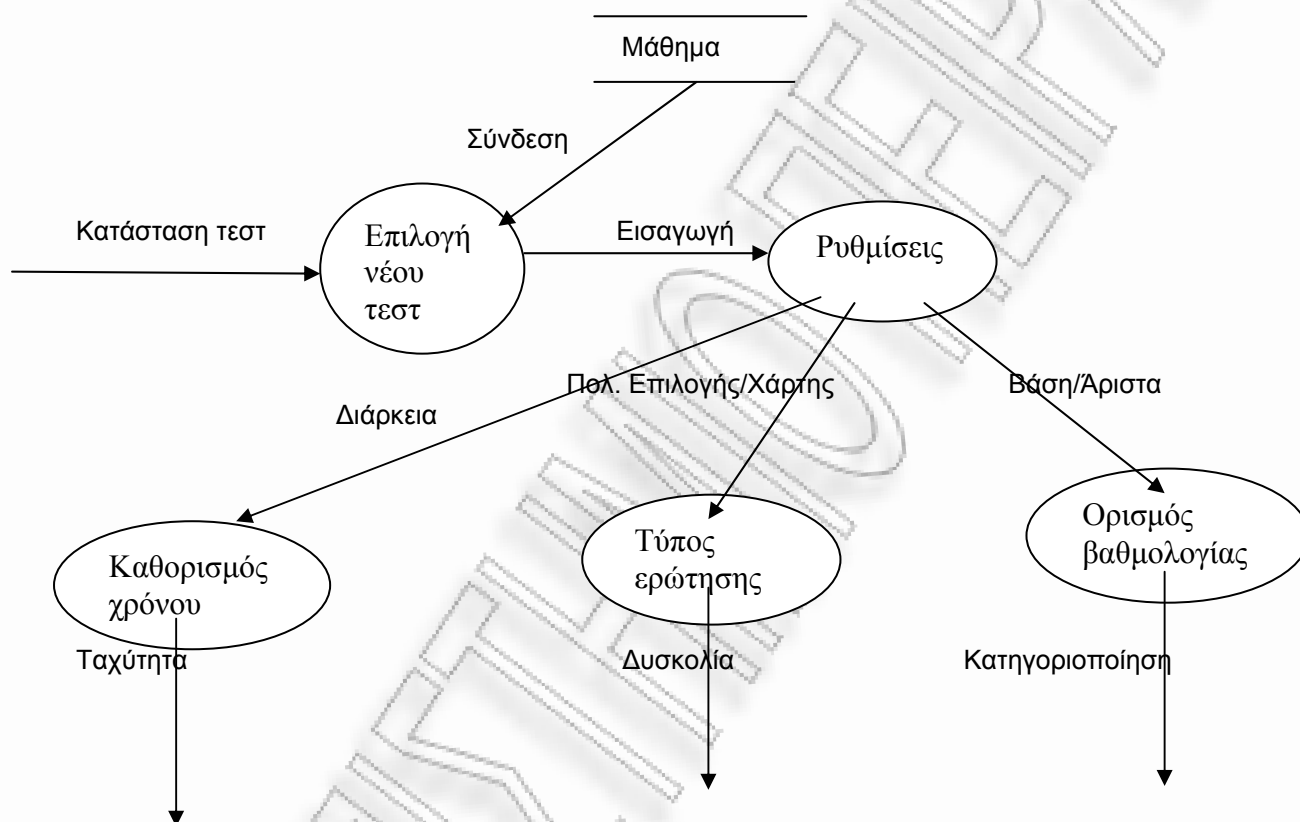
Στη συνέχεια η παραπάνω διεργασία αναλύεται σε άλλες διεργασίες, και κάθε διεργασία αναλύεται με τη σειρά της, έως ότου να οριστεί ένα ικανοποιητικό επίπεδο λεπτομέρειας.

Στην εφαρμογή e-School επειδή είναι πολύπλοκη και κάθε χρήστης έχει πολλές αρμοδιότητες, θα αναλυθεί μόνο ο διαχειριστής στα επόμενα διαγράμματα ροής δεδομένων. Ο διαχειριστής έχει τη δυνατότητα να διαχειρίζεται όλες τις ρυθμίσεις των τεστ (διάρκεια του τεστ, αριθμός προσπαθειών σε ερωτήσεις πολλαπλής επιλογής, αριθμός προσπαθειών σε ερωτήσεις χάρτη, καθορισμός του άριστα και της βάσης, αριθμός δεκαδικών ψηφίων στην βαθμολογία), τα μαθήματα (τίτλος μαθήματος, σημειώσεις μαθήματος, ύλη μαθήματος, ερωτήσεις, χάρτης, εισαγωγή νέου μαθήματος, διαγραφή μαθήματος), τα προφίλ των καθηγητών (καθορισμών δικαιωμάτων, διαγραφή συνθηματικού καθηγητή, εισαγωγή νέου, διαγραφή καθηγητή), τους μαθητές (επίβλεψη της αξιολόγησής τους, διαγραφή συνθηματικού μαθητή, εισαγωγή νέου, διαγραφή μαθητή) και να αλλάζει το συνθηματικό του. Παρακάτω παρατίθεται διάγραμμα πρώτου επιπέδου:



Σχήμα 1.2: Διάγραμμα Πρώτου Επιπέδου

Στο επόμενο (δεύτερο) επίπεδο επιλέγεται τυχαία η δραστηριότητα «Επεξεργασία Τεστ», ώστε να γίνει μεγαλύτερη και λεπτομερέστερη ανάλυση. Οι αρμοδιότητες του διαχειριστή σε σχέση με τη διαχείριση των ρυθμίσεων των τεστ (διάρκεια του τεστ, αριθμός προσπαθειών σε ερωτήσεις πολλαπλής επιλογής, αριθμός προσπαθειών σε ερωτήσεις χάρτη, καθορισμός του άριστα και της βάσης, αριθμός δεκαδικών ψηφίων στην βαθμολογία). Η παραπάνω διαδικασία φαίνεται στο παρακάτω διάγραμμα:



Σχήμα 1.3: Διάγραμμα Δευτέρου Επιπέδου

Ένα ΔΡΔ από μόνο του δεν επαρκεί για την περιγραφή των απαιτήσεων, γιατί το λόγο υπάρχει το λεξικό δεδομένων. Στο λεξικό δεδομένων καταγράφονται όλες οι ροές δεδομένων του ΔΡΔ με επεξήγηση της δομής τους.

Το λεξικό δεδομένων για την εφαρμογή e-School παρατίθεται παρακάτω:

#### Λεξικό Δεδομένων

Όνομα	Περιγραφή
Διαχειριστής	Η οντότητα διαχειριστής αποτελεί έναν από τους χρήστες του συστήματος. Περιέχει στοιχεία και αρμοδιότητες.
Καθηγητής	Η οντότητα καθηγητής αποτελεί έναν από τους χρήστες του συστήματος. Περιέχει στοιχεία και αρμοδιότητες.
Μαθητής	Η οντότητα μαθητής αποτελεί έναν από τους χρήστες του συστήματος. Περιέχει στοιχεία και

	αρμοδιότητες.
Δεδομένα	Τα στοιχεία που υπάρχουν ή προστίθενται στην εφαρμογή.
Στοιχεία	Τα στοιχεία που αντλούνται από την εφαρμογή.
Απαντήσεις	Τα στοιχεία που εισάγονται από τους μαθητές κατά την διάρκεια των τεστ.
Αλλαγές	Οι διαφορές μεταξύ των στοιχείων.
Απόδοση	Τα στοιχεία που δείχνουν την πρόοδο του μαθητή.
Στατιστικά	Τα στοιχεία σε ποσοστά που αφορούν την απόδοση των μαθητών και άλλα ποσοστιαία μεγέθη.
Εντολές	Οι οδηγίες αλλαγών στο σύστημα.
Συνθηματικό	Τα μοναδικοποιημένα στοιχεία κάθε χρήστη. Το log in.
Ενημέρωση	Η αλλαγή κάποιων στοιχείων.
Μάθημα	Τα στοιχεία με τα οποία γίνεται η εκπαίδευση των μαθητών.
Τεστ	Τα στοιχεία με τα οποία ελέγχεται η απόδοση του μαθητή. Συνδέεται άμεσα με το μάθημα.
Ρυθμίσεις	Ο καθορισμός κάποιων παραμέτρων.
Διάρκεια	Ο χρόνος που απαιτείται για την ολοκλήρωση του τεστ. Τα στοιχεία που αποτελούν ένα τεστ. Μπορεί να είναι πολλαπλής επιλογής, σωστό/λάθος ή σε χάρτη.
Ερώτηση	
Βαθμολογία	Καθορισμός στοιχείων για την μέτρηση της απόδοσης των μαθητών.

## **ΚΕΦΑΛΑΙΟ 4**

### **ΣΧΕΔΙΑΣΜΟΣ ΛΟΓΙΣΜΙΚΟΥ**

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΑΙΑ

Για τον σχεδιασμό και την υλοποίηση ενός προγράμματος λογισμικού είναι απαραίτητο να γίνει διαχωρισμός του σχεδιασμού του προγράμματος σε:

1. Εξωτερικό Σχεδιασμό
2. Εσωτερικό Σχεδιασμό
3. Δομημένος Σχεδιασμός
4. Αναλυτικός Σχεδιασμός
5. Σχεδιασμός Βάσης

Με τον όρο εξωτερικό εννοούμε τα εξωτερικά χαρακτηριστικά του προγράμματος, ενώ με τον όρο εσωτερικό τον σχεδιασμό του προγράμματος με τα κατάλληλα εργαλεία. Δηλαδή στην συγκεκριμένη περίπτωση οι delphi φόρμες - σελίδες και η βάση δεδομένων.

#### **4.1 ΕΞΩΤΕΡΙΚΟΣ ΣΧΕΔΙΑΣΜΟΣ**

Ο εξωτερικός σχεδιασμός περιλαμβάνει τη σύλληψη και τον προσδιορισμό εξωτερικών χαρακτηριστικών ενός προϊόντος λογισμικού. Κατ' αρχάς πρέπει να επισημανθεί πως ο εξωτερικός σχεδιασμός δεν ξεκινάει ξαφνικά, αλλά αρχίζει κατά τη φάση της ανάλυσης και συνεχίζει στη φάση του σχεδιασμού. Σκοπό έχει την εκλέπτυνση του προσδιορισμού των απαιτήσεων και τον καθορισμό της δομής του συστήματος σε υψηλό επίπεδο.

Στα χαρακτηριστικά του προς υλοποίηση προγράμματος υπάρχει αρχικά μία φόρμα εισαγωγής, από την οποία μπορεί να κάνουν log in όλοι οι χρήστες της εφαρμογής. Ανάλογα το προφίλ τους, έχουν την δυνατότητα πλοήγησης στις οθόνες του προγράμματος που απευθύνονται σε αυτούς. Για να υλοποιηθεί κάτι τέτοιο δημιουργήθηκε το λογισμικό με τη χρήση της γλώσσας BDS Delphi 2006 και τα δεδομένα αποθηκεύονται και επεξεργάζονται από μια βάση δεδομένων η οποία έχει υλοποιηθεί στην ίδια γλώσσα.

Η επιλογή της δημιουργίας των φορμών/σελίδων με τη χρήση της γλώσσας BDS Delphi 2006 έχει να κάνει λόγο με την διασύνδεση με τον χρήστη, ώστε αυτή να είναι όσο το δυνατότερο πιο φιλική. Τα δεδομένα εισάγονται στην βάση από το χρήστη μέσω των επιλογών από τα μενού των σελίδων. Όσο αναφορά τα αποτελέσματα αυτά επιστρέφονται στον χρήστη από τη βάση ανάλογα με το ερώτημα που κάνει αυτός κάθε φορά, κάνοντας χρήση της γλώσσας Delphi στην οποία και έχουν υλοποιηθεί τα ερωτήματα.

Καταρχήν, θα πρέπει να πούμε ότι πρόκειται να φτιάξουμε ένα πρόγραμμα μέσω του οποίου θα γίνεται η διαχείριση της εφαρμογής των μαθημάτων της e-School. Για την υλοποίηση του επιλέχθηκε η χρήση της γλώσσας προγραμματισμού Delphi τόσο για βάση δεδομένων όσο και για το interface, καθώς και για την διασύνδεση αυτών ώστε να παραχθεί το επιθυμητό αποτέλεσμα.

Προκειμένου να επιτευχθεί το επιθυμητό αποτέλεσμα στο επίπεδο του χρήστη τότε πρέπει να δομηθεί το σύστημα στο επίπεδο αυτού. Αυτό γίνεται με το να υλοποιήσουμε το user interface σε τέτοιο επίπεδο ώστε να είναι φιλικό προς το χρήστη τόσο από θέμα χειρισμού όσο και από θέμα κατανόησης.

Ο τελικός χρήστης του προγράμματος πρέπει να δει ένα προσχέδιο τόσο του interface όσο και των λειτουργιών που θα κάνει αυτό προκειμένου να προχωρήσουμε στην φάση της υλοποίησης του προγράμματος. Το ίδιο πρέπει να γίνει και με την δημιουργία της βάσης δεδομένων. Αυτό το βήμα είναι από τα πλέον απαιτητικά και σίγουρα θα πρέπει να γίνει κάτω από την επίβλεψη του χρήστη ώστε αυτό να καλύπτει πλήρως τις ανάγκες του. Εάν σε αυτό το στάδιο προκύψει κάποιο λάθος τότε υπάρχει περίπτωση το πρόγραμμα είτε να μην καλύπτει τις ανάγκες του χρήστη είτε να χαθεί πολύτιμος χρόνος και χρήμα κατά την φάση της υλοποίησης του.

Στην σχεδίαση ενός λογισμικού πρέπει να καταφεύγουμε σε 2 κρίσιμες ερωτήσεις: 1. τι πρόκειται να φτιάξουμε και 2. πώς θα το φτιάξουμε. Αυτά τα ερωτήματα πρέπει να γίνουν τόσο σε αυτούς που θα φτιάξουν το πρόγραμμα όσο και σε αυτούς που θα το χρησιμοποιήσουν. Η βαρύτητα αυτού του σταδίου είναι τέτοια ώστε θα κρίνει το κόστος, το χρόνο και την επιτυχία μιας θετικής κατάληξης.

Ας δούμε όμως τι έχει υλοποιηθεί στην συγκεκριμένη εφαρμογή. Καταρχήν έχει υλοποιηθεί ένα πρόγραμμα που επιτρέπει στον χρήστη, όποιες αρμοδιότητες και να έχει, να διαχειρίζεται μία εφαρμογή ενός μαθήματος με άνεση. Το πρόγραμμα πρέπει να είναι απλό και εύκολο στη χρήση ώστε να το διαχειρίζεται κάποιος χωρίς ιδιαίτερες γνώσεις. Για τον λόγο αυτό αποφασίστηκε να γίνει η υλοποίηση με τη χρήση φορμών της γλώσσας Delphi. Τα δεδομένα αποθηκεύονται σε μια βάση δεδομένων, γεγονός που επιτρέπει την εύκολη αναβάθμιση του προγράμματος με νέες λειτουργίες σύμφωνα με τις μελλοντικές ανάγκες. Με την επιλογή αυτών των εργαλείων δίνεται η απάντηση στο κρίσιμο ερώτημα «πώς θα το φτιάξουμε». Η επιλογή τους επιτρέπει την προσθήκη στοιχείων και την εύκολη εισαγωγή νέων δεδομένων χωρίς να επηρεάζονται οι ήδη υπάρχουσες εγγραφές κάτι πολύ σημαντικό. Επίσης, επιτρέπεται και η ενημέρωση των υπάρχόντων δεδομένων από τους κατάλληλα εξουσιοδοτημένους χρήστες (administrator).

## 4.2. ΕΣΩΤΕΡΙΚΟΣ ΣΧΕΔΙΑΣΜΟΣ

Με τον όρο εσωτερικό σχεδιασμό στην τεχνολογία λογισμικού εννοούμε το πώς θα υλοποιήσουμε το πρόγραμμα που θέλουμε να φτιάξουμε. Δηλαδή θα αναφερθούμε στην υλοποίηση της βάσης μας και στο interface του προγράμματος και για ποιο λόγο θα εισάγουμε το κάθε πεδίο.

Ο εσωτερικός σχεδιασμός χωρίζεται στις εξής κατηγορίες :

### A) Αρχιτεκτονικός σχεδιασμός

Στον αρχιτεκτονικό σχεδιασμό γίνεται αναγνώριση όλων των υποσυστημάτων που συνθέτουν το πρόγραμμα καθώς και των σχέσεων τους. Η εφαρμογή «e-School» αποτελείται από δύο κύρια μέρη -υποσυστήματα. Αυτά είναι η διδασκαλία- εκπαίδευση των μαθητών και η αξιολόγησή τους με χρήση τεστ. Η σχέση μεταξύ αυτών των δύο υποσυστημάτων είναι αυτονόητη. Όταν ένας μαθητής νιώσει ότι έχει κατανοήσει την ύλη των μαθημάτων που αναφέρεται το μάθημα, τότε θα πρέπει να συνεχίσει στα τεστ (διαγωνίσματα) για να σιγουρευτεί για τις γνώσεις του. Τα δύο υποσυστήματα είναι στενά συνδεδεμένα μεταξύ τους. Το ίδιο ισχύει και για τις αρμοδιότητες του administrator, ο οποίος διαχειρίζεται τα διδασκόμενα μαθήματα και τα τεστ, από τα οποία προκύπτει η αξιολόγηση των μαθητών.

### B) Περιληπτικός προσδιορισμός

Στον περιληπτικό προσδιορισμό για κάθε υποσύστημα προσδιορίζονται περιληπτικά οι υπηρεσίες που παρέχει και οι περιορισμοί κάτω από τους οποίους λειτουργεί. Το πρώτο υποσύστημα έχει ως κύριο σκοπό να διδάξει στον μαθητή τους νομούς της Ελλάδος με χρήση εικόνων, βίντεο, ήχου και κειμένου. Το δεύτερο υποσύστημα έχει ως σκοπό να αξιολογήσει το μαθητή σύμφωνα με αυτά που διδάχτηκε.

### Γ) Σχεδιασμός διασύνδεσης

Στον σχεδιασμό της διασύνδεσης σχεδιάζεται και τεκμηριώνεται η διασύνδεση (interface) κάθε υποσυστήματος με άλλα υποσυστήματα. Ο προσδιορισμός διασύνδεσης πρέπει να είναι σαφής και να επιτρέπει την χρήση του υποσυστήματος χωρίς γνώση της λειτουργίας του.

### Δ) Σχεδιασμός στοιχείων

Στον σχεδιασμό στοιχείων οι υπηρεσίες που παρέχει ένα υποσύστημα χωρίζονται στα στοιχεία που συνθέτουν το υποσύστημα.

### Ε) Σχεδιασμός δομών δεδομένων

Στον σχεδιασμό δομών δεδομένων σχεδιάζονται και προσδιορίζονται οι δομές δεδομένων με λεπτομέρεια. Στην εφαρμογή «e-School» για κάθε νομό υπάρχουν κείμενα, βίντεο, εικόνες, σχήματα, χάρτες και ηχητικό υλικό.



### ΣΤ) Σχεδιασμός αλγορίθμων

Στην εφαρμογή «e-School» δεν σχεδιάσθηκαν αλγόριθμοι για τη παρουσίαση του διδακτικού υλικού.

Καταρχήν κατασκευάστηκε η βάση όπου θα αποθηκεύονται τα δεδομένα. Για να γίνει κάτι τέτοιο, έγινε συνεννόηση με τους χρήστες (καθηγητές και μαθητές) του προγράμματος ώστε να καταγραφούν οι απαιτήσεις τους. Αυτό το βήμα είναι απαραίτητο, ώστε να στηθούν σωστά οι πίνακες της βάσης. Δημιουργήθηκαν οι πίνακες μέσα στους οποίους αποθηκεύονται όλες οι πληροφορίες για την υλοποίηση της εφαρμογής. Τα πεδία κάθε πίνακα είναι αυτά που ζητήσαν οι χρήστες, ωστόσο μια μελλοντική εισαγωγή ενός νέου πεδίου είναι δυνατή χωρίς να μας δημιουργήσει πρόβλημα στη βάση. Επόμενο στάδιο είναι η ένωση των πινάκων μεταξύ τους, ώστε να υπάρχει η δυνατότητα εξαγωγής των επιθυμητών αποτελεσμάτων αλλά και για την αποφυγή των διπλών εισαγωγών σε δεδομένα εφόσον τα ίδια ακριβώς στοιχεία θα υπάρχουν σε κάποιον άλλο πίνακα της βάσης. Συνεχίζοντας την υλοποίηση της βάσης, ακολουθεί στο στάδιο δημιουργίας των φορμών. Ο σκοπός τους είναι η εισαγωγή δεδομένων στους πίνακες, αλλά και η έξοδος αποτελεσμάτων χρήσιμων για τον χρήστη του προγράμματος π.χ. εμφάνιση αξιολόγησης μαθητών με βάση τα τεστ. Ως τελευταίο βήμα υλοποίησης της βάσης είναι τα ερωτήματα με τα οποία εξάγονται τα αποτελέσματα από τη βάση. Με αυτό τον τρόπο ο μαθητής μπορεί να δει τα διαθέσιμα μαθήματα βάσει του τίτλου του μαθήματος, καθώς και τότε παρακολούθησε και τότε εξετάστηκε σε αυτό το μάθημα, όπως επίσης και τον βαθμό του στο τεστ.

Μετά το πέρας του ενός σταδίου του προγράμματος σειρά έχει το επόμενο που είναι η υλοποίηση του interface. Οι φόρμες θα συνδεθούν με τη βάση προκειμένου να περαστούν τα δεδομένα μέσα σε αυτήν. Μέσω των φορμών αυτών γίνονται οι εγγραφές στη βάση, καθώς επίσης και ανακαλούνται οι πληροφορίες από αυτήν. Ο σκοπός των φορμών αυτών είναι η καλύτερη διαχείριση του προγράμματος από τους χρήστες. Οι φόρμες έχουν μια μηχανή αναζήτησης η οποία ανακαλεί στοιχεία από τη βάση μέσω των ερωτήσεων που έχουν αναπτυχθεί μέσα στη βάση. Το κεντρικό interface του προγράμματος αποτελείται από κουμπιά τα οποία μας εισάγουν στις άλλες φόρμες με τις οποίες γίνεται η διαχείριση της βάσης και της εφαρμογής.

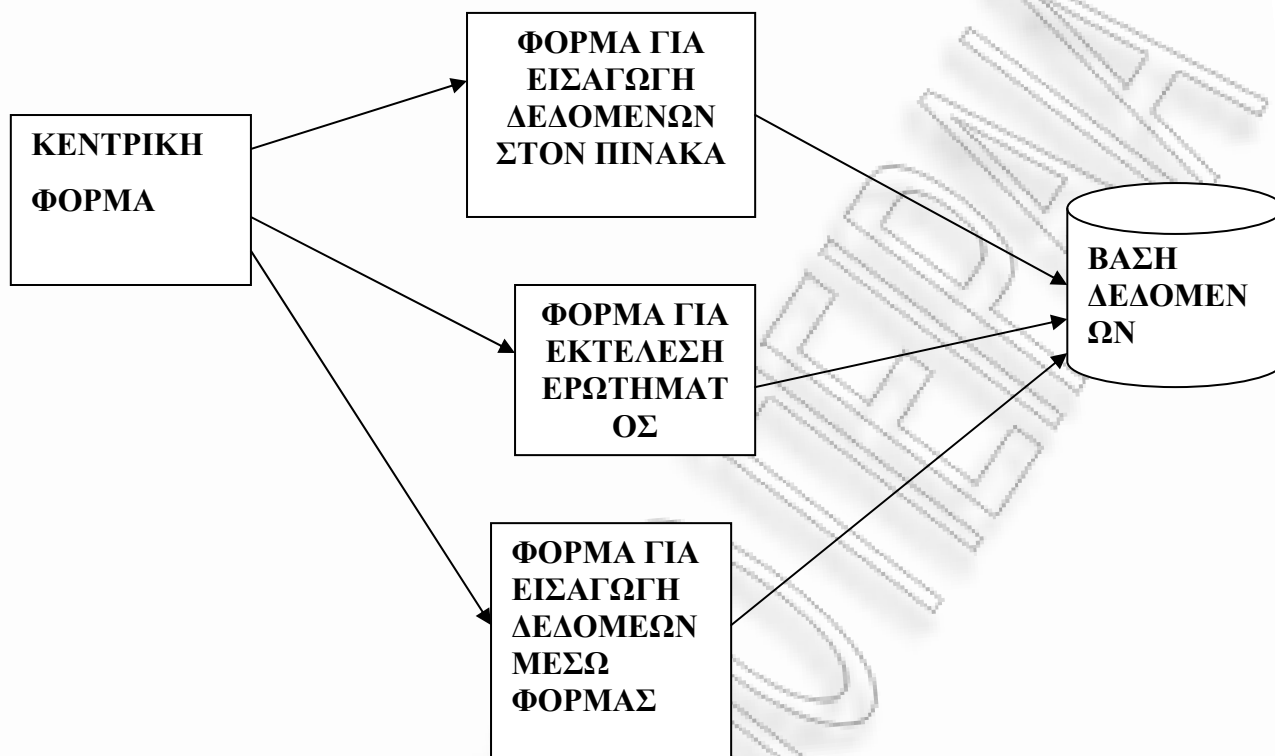
Αυτά που αναφέρθηκαν παραπάνω αποτελούν μέρος του αρχιτεκτονικού σχεδιασμού του προγράμματος, ο οποίος είναι ένα από τα βήματα του εσωτερικού σχεδιασμού της τεχνολογίας λογισμικού.

Οι φόρμες που δημιουργήθηκαν με σκοπό την εισαγωγή στοιχείων στη βάση περιλαμβάνουν button και layers ώστε να είναι φιλικές προς τον χρήστη. Ως περιορισμούς στις φόρμες έχει τεθεί η εισαγωγή ορισμένων μόνο στοιχείων στη βάση και αυτά εισάγονται μόνο από τον administrator. Επίσης, από τις φόρμες μπορούμε να δούμε μόνο μερικές εγγραφές ανάλογα με το query που τρέχει πίσω από την επιλογή της κάθε φόρμας. Επομένως, οι μαθητές και καθηγητές που χρησιμοποιούν την εφαρμογή έχουν μόνο μερική πρόσβαση στη βάση και όχι πλήρη έλεγχο αυτής κάτι το οποίο αποζητείται προκειμένου πρόσβαση να έχουν μόνο τα αρμόδια άτομα, όπως ο administrator. Όσο αφορά τη βάση, οι περιορισμοί που έχουμε τεθεί είναι στα πεδία των πινάκων όπου και καταχωρούνται όλα τα δεδομένα και αυτό για να αποτραπεί κάποια λανθασμένη εισαγωγή (π.χ. την εισαγωγή αρνητικής τιμής στο πεδίο όπου αποθηκεύεται η βαθμολογία κάποιου μαθητή στην εξέτασή του σε ένα τεστ). Σε περίπτωση ενός τέτοιου λάθους η καταχώρηση δεν γίνεται και το πρόγραμμα προειδοποιεί για το λάθος.

Για να λειτουργήσει σωστά η βάση, αλλά και γενικότερα το όλο πρόγραμμα να μην έχει λάθη τα οποία ίσως αργότερα αποβούν μοιραία για τους χρήστες, γίνεται εισαγωγή περιορισμών στα πεδία της βάσης, ώστε να αποφευχθεί η λανθασμένη καταχώρηση δεδομένων.

Προκειμένου οι χρήστες να έχουν τη δυνατότητα να λαμβάνουν τα αποτελέσματα, πρέπει να ενωθούν οι φόρμες με την βάση δεδομένων. Αυτό ονομάζεται σχεδιασμός διασύνδεσης και σκοπός είναι να συνδέσουμε τις φόρμες με τα κατάλληλα στοιχεία της βάσης. Ενώνονται οι φόρμες με τα layers με τα διάφορα ερωτήματα που έχουν δημιουργηθεί, ώστε οι χρήστες να λαμβάνουν τα αποτελέσματα που επιθυμούν. Είναι, δηλαδή, σαν να δημιουργούνται διαδικασίες οι οποίες καλούν τα συγκεκριμένα ερωτήματα της βάσης. Αυτό υλοποιείται με τη χρήση buttons και layers, με την επιλογή των οποίων καλούνται τα

αντίστοιχα ερωτήματα-φόρμες της βάσης. Στο ακόλουθο διάγραμμα παρουσιάζεται η διαδικασία σύνδεσης των φορμών με τα ερωτήματα και με τους πίνακες της βάσης.



Μετά τον σχεδιασμό των πινάκων και την καταχώρηση εγγραφών σε αυτούς συνεχίζεται το «κτίσιμο» της βάσης με την υλοποίηση των φορμών.

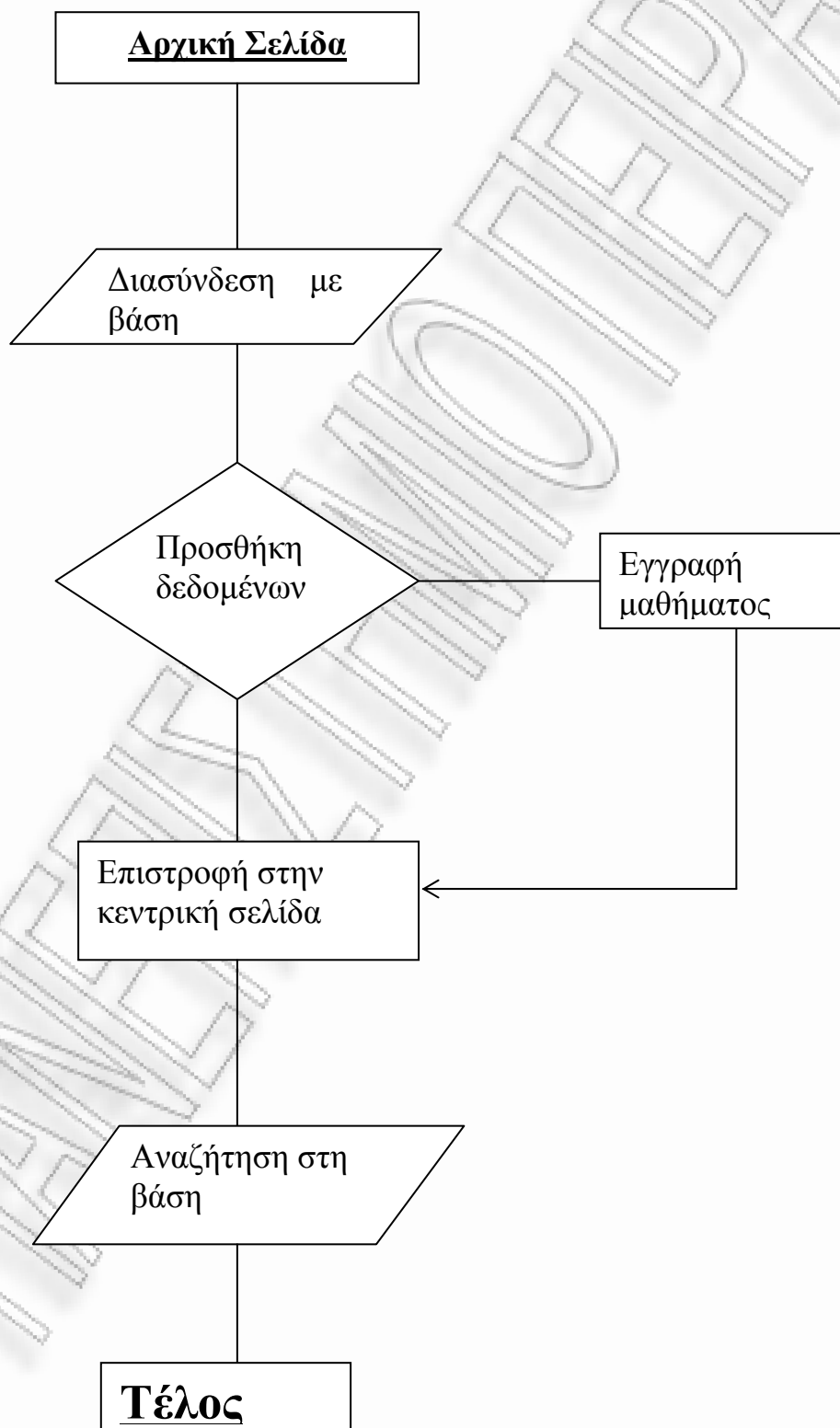
Ο ρόλος των φορμών σε μία βάση είναι η εισαγωγή στοιχείων στους πίνακες με φιλικό προς τον χρήστη περιβάλλον καθώς και η εξαγωγή στοιχείων. Στην προκειμένη περίπτωση και ύστερα από άμεση συνεννόηση με τους χρήστες του προγράμματος μας δημιουργήθηκαν οι φόρμες της εφαρμογής. Αρχικά κατασκευάστηκε μια κεντρική φόρμα, από την οποία κάνουν log in οι χρήστες και στην συνέχεια ανάλογα με το προφίλ του κάθε χρήστη κατασκευάστηκαν τριών κατηγοριών φόρμες. Μέσω αυτών των φορμών γίνεται η καταχώρηση των δεδομένων στους πίνακες. Αυτή η κίνηση είναι πολύ σημαντική, διότι ο μαθητής δε θα πρέπει να έχει πρόσβαση στις εγγραφές του πίνακα, αλλά μόνο να εισάγει εγγραφές σε αυτούς. Με αυτό τον τρόπο αυξάνεται το επίπεδο ασφαλείας στη βάση.

Δημιουργήθηκαν φόρμες, ώστε να οι χρήστες να λαμβάνουν τα κυριότερα στοιχεία που τους ενδιαφέρουν. Δεν χρησιμοποιήθηκαν reports για την εξαγωγή των στοιχείων, διότι κατόπιν συνεννοήσεως με τους χρήστες του προγράμματος διαπιστώθηκε ότι η χρήση των φορμών κάλυπτε απολύτως τις ανάγκες τους. Στη συνέχεια δημιουργήθηκαν οι ερωτήσεις. Τα ερωτήματα έχουν σαν σκοπό να εξαχθούν δεδομένα από τη βάση σύμφωνα με τις άμεσες ανάγκες του τελικού χρήστη. Εξάλλου είναι εύκολο να προστεθούν κάποια ερωτήματα, ώστε να καλυφθούν και ενδεχόμενες μελλοντικές ανάγκες. Αυτός είναι και ο κυρίως λόγος για τον οποίο επιλέχθηκαν αυτά τα εργαλεία σχεδίασης για την εφαρμογή «e-School».

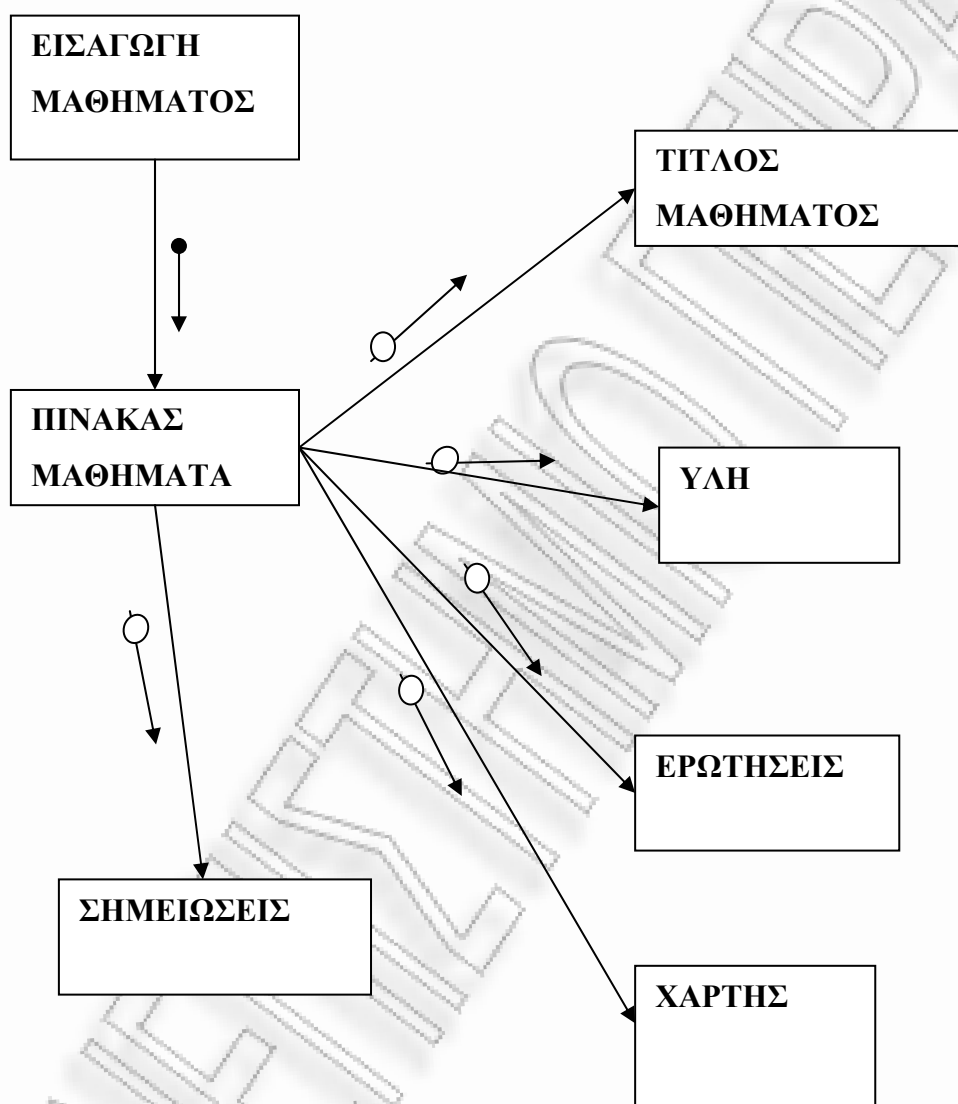
Όλα όσα αναφέρθηκαν παραπάνω ανήκουν στον εσωτερικό σχεδιασμό της εφαρμογής, δηλαδή στον τρόπο και τα εργαλεία με τα οποία υλοποιήθηκε το πρόγραμμα.

### 4.3. ΔΟΜΗΜΕΝΟΣ ΣΧΕΔΙΑΣΜΟΣ

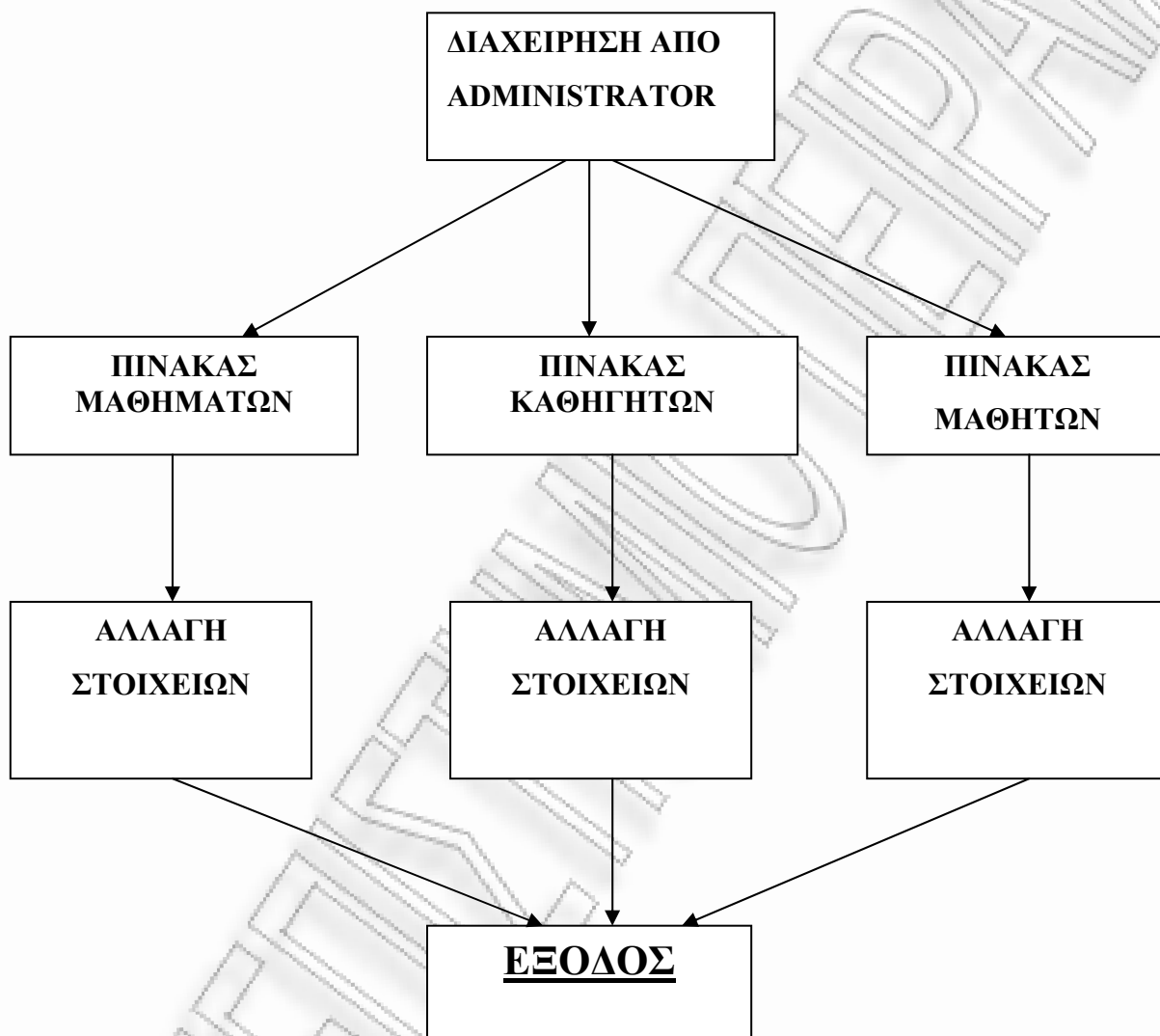
Στη συνέχεια παρουσιάζεται ένα block διάγραμμα που δείχνει μία από τις λειτουργίες της εφαρμογής «e-School».



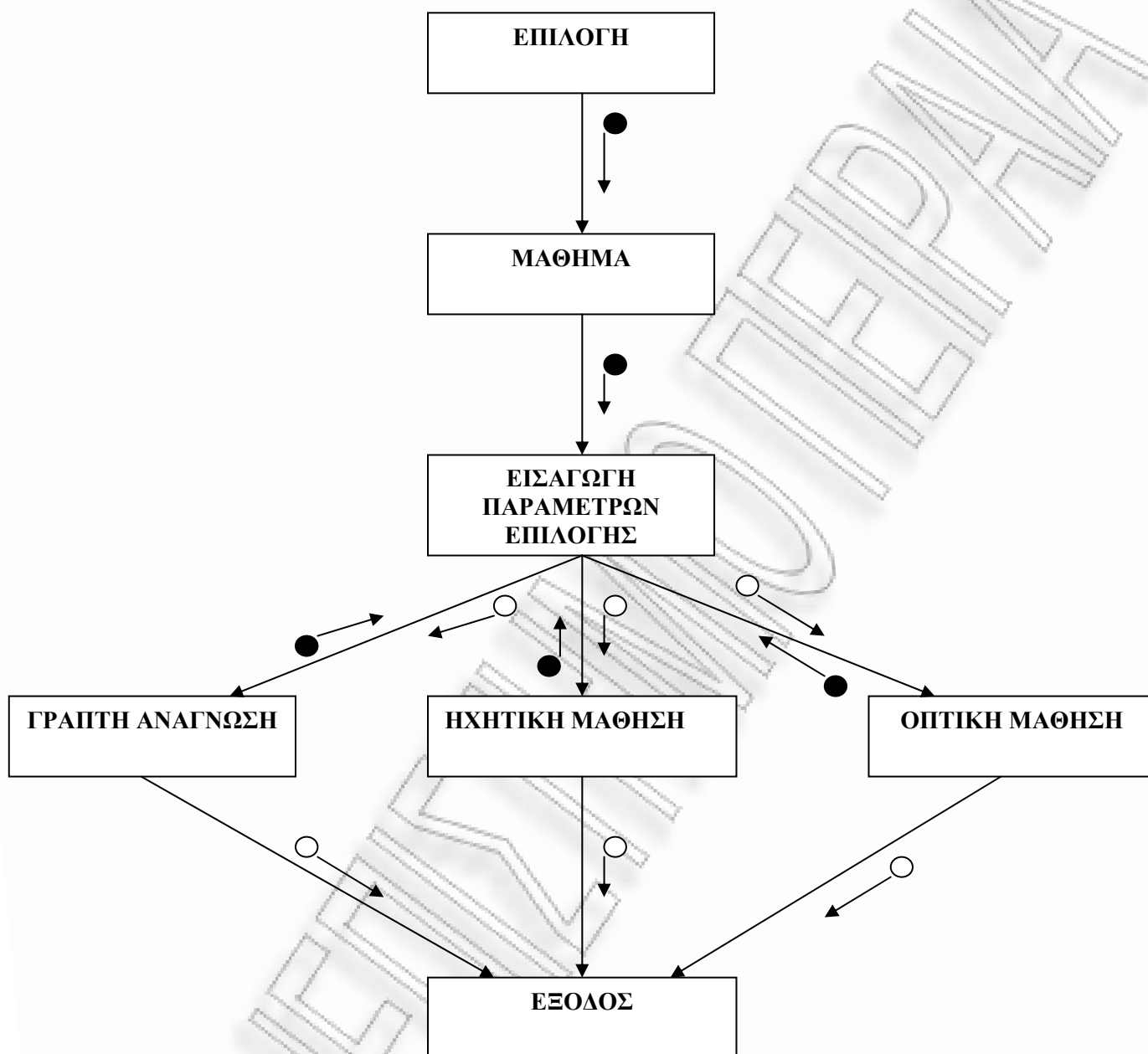
Στο παρακάτω διάγραμμα δομής για εισαγωγή δεδομένων στην βάση και συγκεκριμένα στον πίνακα μαθημάτων, παρατηρείται ο αρχικός έλεγχος της πληροφορίας ελέγχου, ώστε να γίνει η εισαγωγή των στοιχείων στα σωστά πεδία του πίνακα της βάσης. Μετά τον επιτυχή έλεγχο, γίνεται η εισαγωγή των δεδομένων στα αντίστοιχα πεδία και κάτω από τους ανάλογους περιορισμούς (οι περιορισμοί επισημαίνεται πως βρίσκονται στα πεδία της βάσης).



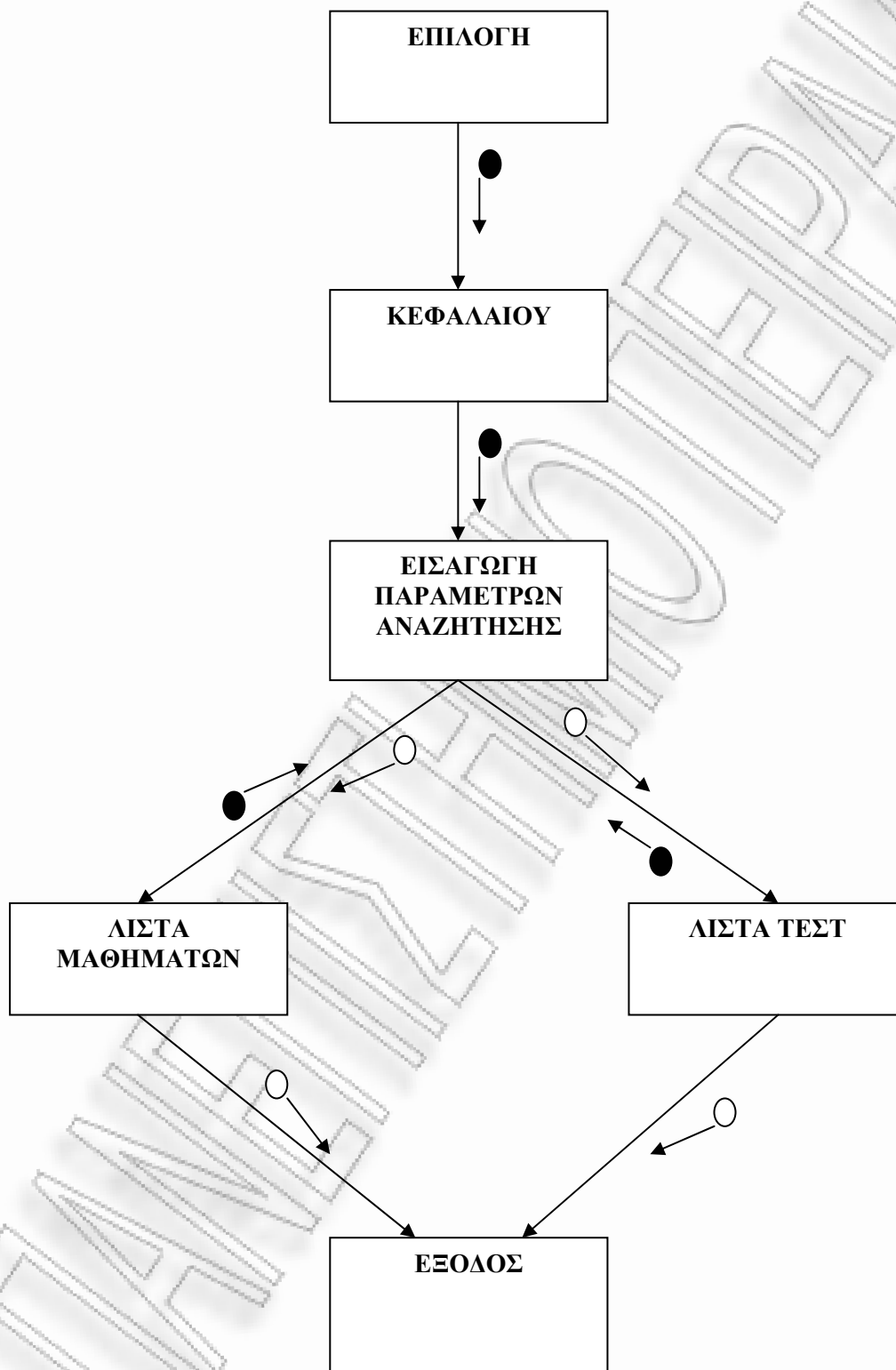
Στο παρακάτω διάγραμμα δομής δεδομένων, παρουσιάζεται η δυνατότητα του προγράμματός μέσω του administrator να αλλάζουν τα στοιχεία των μαθητών, καθηγητών και μαθημάτων.



Το παρακάτω διάγραμμα απεικονίζει την διαδικασία επιλογής από τον μαθητή του τρόπου παρακολούθησης του μαθήματος.



Το παρακάτω διάγραμμα απεικονίζει την διαδικασία επιλογής από τον μαθητή του μαθήματος ή του τεστ στο αντίστοιχο κεφάλαιο της ύλης.



#### 4.4. ΑΝΑΛΥΤΙΚΟΣ ΣΧΕΔΙΑΣΜΟΣ

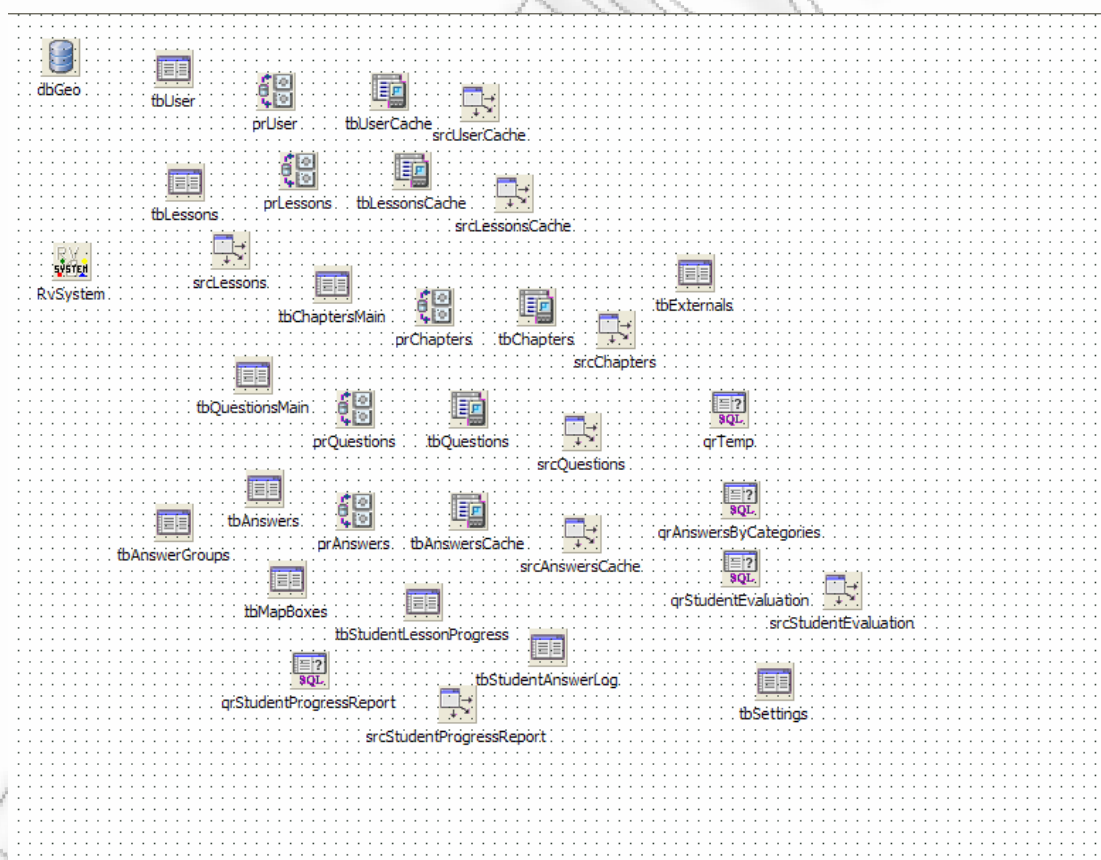
Τελευταίο κομμάτι ενός προς ανάπτυξη προγράμματος είναι ο αναλυτικός του σχεδιασμός. Δηλαδή, στον τρόπο με τον οποίο γίνονται οι διάφορες εντολές μέσω αυτού. Ο ψευδοκώδικας είναι σαν κώδικας προγράμματος, αλλά δεν ανήκει σε καμία γλώσσα προγραμματισμού και χρησιμοποιείται για την καλύτερη κατανόηση πριν αρχίσει η εγγραφή του πραγματικού κώδικα.

Επειδή το συγκεκριμένο πρόγραμμα είναι αρκετά πολύπλοκο και σχεδιάστηκε βήμα, δεν χρησιμοποιήθηκε ψευδοκώδικας. Οι βασικές λειτουργίες του προγράμματος περιγράφονται και παρατίθενται τα αντίστοιχα παραρτήματα κώδικα, στο Παράρτημα 2.

#### 4.5. ΣΧΕΔΙΑΣΜΟΣ ΒΑΣΗΣ

Ένας από τους σημαντικότερους τομείς για την καλή ανάπτυξη και λειτουργία ενός σύγχρονου έργου λογισμικού είναι η δημιουργία μίας σωστής και αξιόπιστης βάσης που εξυπηρετεί τα συμφέροντα μας και αναπαριστά πλήρως την εικόνα του οργανισμού για τον οποίο αναπτύχθηκε το λογισμικό του. Για τη δημιουργία της βάσης επιλέχθηκε η DELPHI.

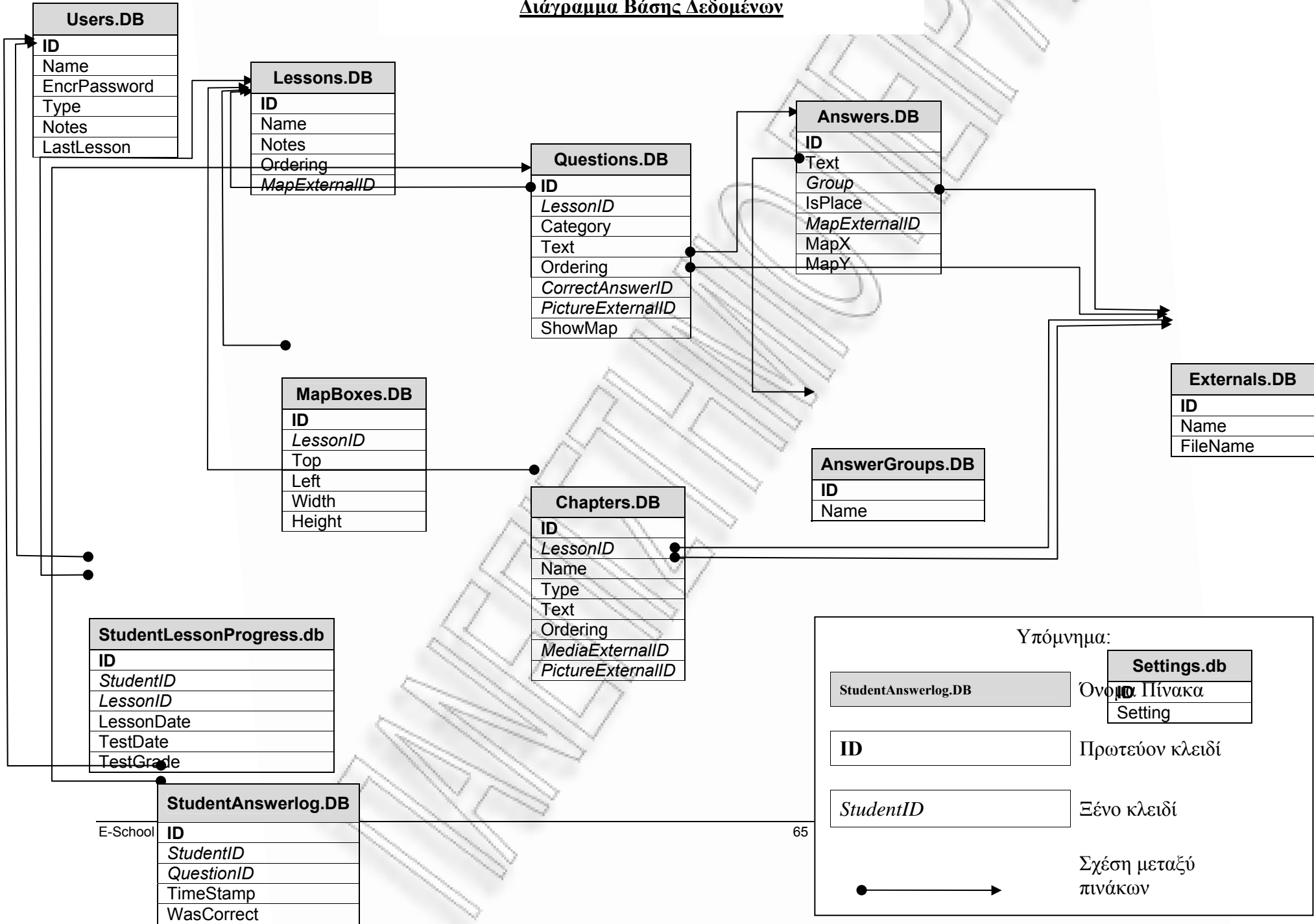
Παρακάτω φαίνεται η βάση δεδομένων της εφαρμογής e-School :



Παρακάτω αναπαριστάται σχηματικά η βάση που υποστηρίζει την εφαρμογή :



**Διάγραμμα Βάσης Δεδομένων**



## ΑΝΑΛΥΣΗ ΣΧΗΜΑΤΟΣ:

- Ο πίνακας Users συνδέεται με τον πίνακα StudentLessonProgress με τα κλειδιά ID, QuestionID και το StudentID
- Ο πίνακας Lessons συνδέεται με τους πίνακες Questions, MapBoxes, Chapters και StudentLessonProgress
- Ο πίνακας Questions συνδέεται με τους πίνακες Lessons, StudentAnswer, Answers και Externals
- Ο πίνακας Answers συνδέεται με τους πίνακες Questions, AnswerGroups και External
- Ο πίνακας Externals συνδέεται με τους πίνακες Answers, Questions και Chapters
- Ο πίνακας MapBoxes συνδέεται με τον πίνακα Lessons
- Ο πίνακας Chapters συνδέεται με τους πίνακες Lessons και Externals
- Ο πίνακας AnswerGroups συνδέεται με τον πίνακα Answers
- Ο πίνακας StudentLessonProgress συνδέεται με τους πίνακες Users και Lessons
- Ο πίνακας StudentAnswer συνδέεται με τους πίνακες Users και Questions

Το παραπάνω διάγραμμα περιλαμβάνει όλα τα γνωρίσματα που έχει κάθε οντότητα-πίνακας ενώ τα πρωτεύοντα κλειδιά κάθε πίνακα είναι αντίστοιχα τα ID, το LessonID, QuestionID και το StudentID.

Παρακάτω παρατίθεται μια αναλυτική περιγραφή των οντοτήτων-πινάκων της βάσης δεδομένων.

## USERS

Η οντότητα αυτή περιλαμβάνει όλους τους χρήστες που χειρίζονται την εφαρμογή. Η οντότητα αυτή περιλαμβάνει τα εξής γνωρίσματα:

- ID, ανάλογα με τη σειρά που έγινε η καταχώρηση κάθε χρήστη
- το user name κάθε χρήστη
- τον κωδικό που χρησιμοποιεί κάθε χρήστης κωδικοποιημένος
- τον τύπο κάθε χρήστη, 0 για τους μαθητές, 1 για τους καθηγητές και 2 για τον administrator
- οι σημειώσεις που καταχωρούνται
- το τελευταίο μάθημα που παρακολούθησε ο μαθητής. Για τους άλλους χρήστες οι τιμές είναι 0 ή null.

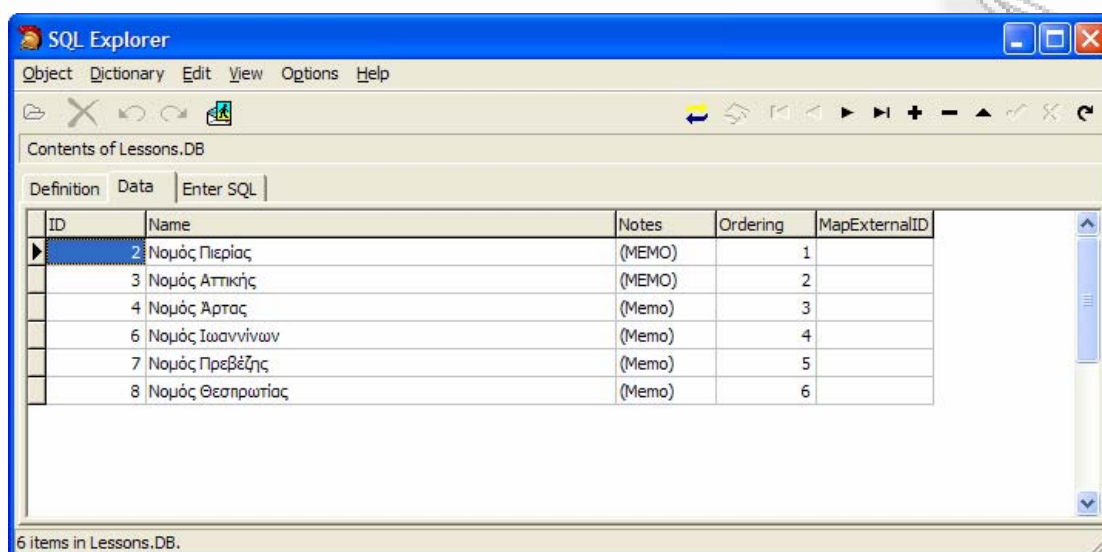
ID	Name	EncrPassword	Type	Notes	LastLesson
1	Administrator	89E37EAB2163E273655E9091ADA678E13E54E4EC1C5426881		2 (Memo)	0
2	TestStudent	9A687EF3C2BAB6ED8D9D2D34DE4D5C2BF7E3905EDD4A1BA1		0 (Memo)	6
3	TestTeacher	9A687EF3C2BAB6ED8D9D2D34DE4D5C2BF7E3905EDD4A1BA1		1 (Memo)	
25	Student2	9A687EF3C2BAB6ED8D9D2D34DE4D5C2BF7E3905EDD4A1BA1		0 (Memo)	

## LESSONS

Η οντότητα αυτή περιλαμβάνει όλα τα μαθήματα της e-School και αποτελείται από τα εξής γνωρίσματα:

- το ID
- τον τίτλο του κάθε μαθήματος
- τυχόν σημειώσεις

- τη σειρά με την οποία εμφανίζονται τα μαθήματα στην εφαρμογή
- το ID του χάρτη που περιλαμβάνεται σε κάθε μάθημα



The screenshot shows the SQL Explorer interface with a table titled 'Contents of Lessons.DB'. The table has columns for ID, Name, Notes, Ordering, and MapExternalID. The data is as follows:

ID	Name	Notes	Ordering	MapExternalID
2	Νομός Πιερίας	(MEMO)	1	
3	Νομός Αττικής	(MEMO)	2	
4	Νομός Άρτας	(Memo)	3	
6	Νομός Ιωαννίνων	(Memo)	4	
7	Νομός Πρεβέζης	(Memo)	5	
8	Νομός Θεσπρωτίας	(Memo)	6	

6 items in Lessons.DB.

## CHAPTERS

Η οντότητα αυτή περιλαμβάνει όλα τα περιεχόμενα των διαφόρων κεφαλαίων κάθε μαθήματος και αποτελείται από τα εξής γνωρίσματα:

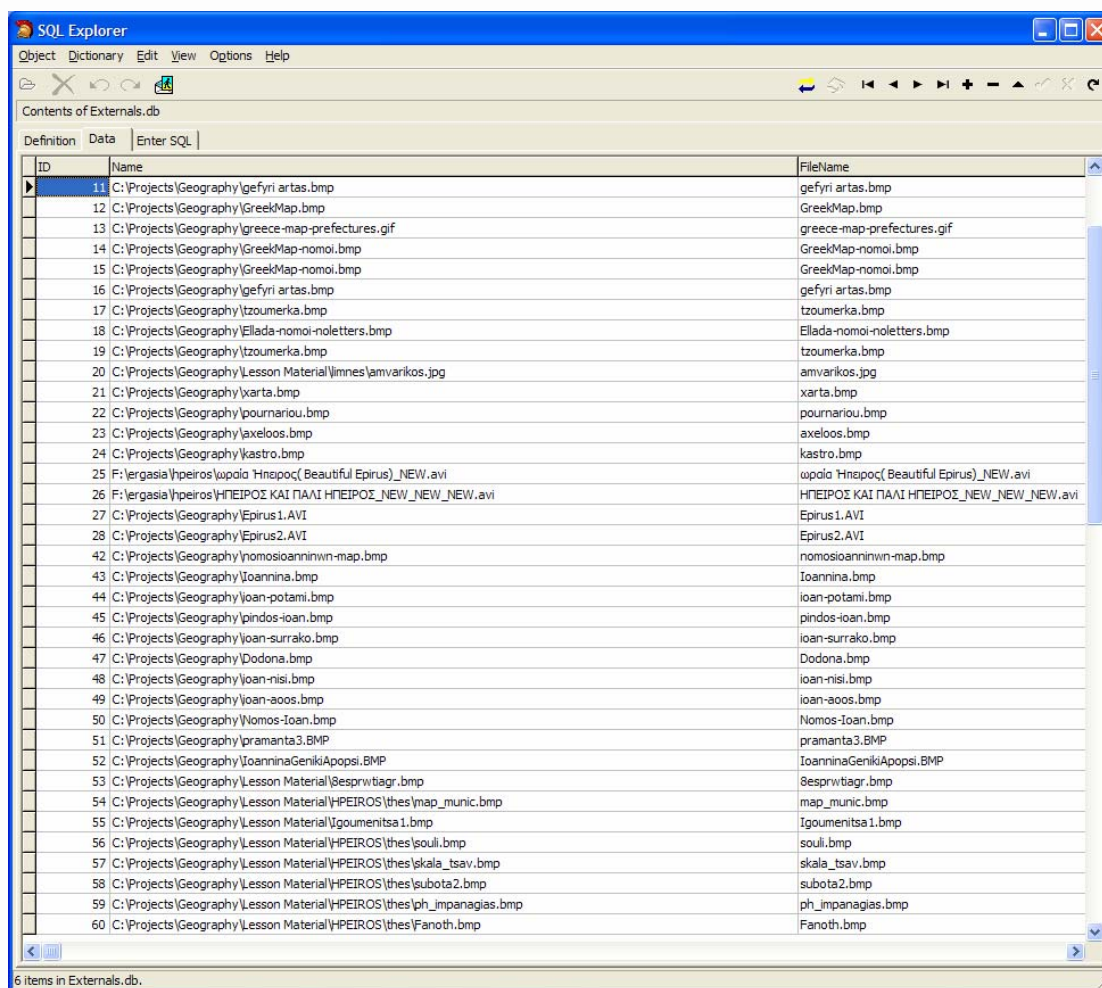
- το ID
- το ID του κάθε μαθήματος
- το όνομα του κάθε κεφαλαίου
- το ID των πολυμέσων που χρησιμοποιούνται σε κάθε κεφάλαιο
- το ID των εικόνων που χρησιμοποιούνται σε κάθε κεφάλαιο
- τυχόν σημειώσεις
- η σειρά των κεφαλαίων κάθε μαθήματος

ID	LessonID	Name	Type	MediaExternalID	PictureExternalID	Text	Ordering
4	2	Εισαγωγή				-1 (Memo)	1
5	2	Βασικά Στοιχεία				(Memo)	2
6	2	Το Όρος Όλυμπος			-1	(MEMO)	3
7	2	Η πόλη της Κατερίνης				(Memo)	4
8	2	Άλλα ενδιαφέροντα στοιχεία				(Memo)	5
9	3	Εισαγωγικά				(Memo)	1
10	3	Η Αθήνα				(Memo)	2
11	3	Ο Πειραιάς				(Memo)	3
12	3	Άλλα στοιχεία				(Memo)	4
30	4	Εισαγωγή			27	8 (MEMO)	1
31	4	Δημογραφικά Στοιχεία				21 (MEMO)	2
32	4	Βουνά και πεδιάδες				19 (MEMO)	4
33	4	Ποτάμια				23 (MEMO)	5
34	4	Λίμνες, λιμνοθάλασσες και ακτές				9 (MEMO)	6
35	4	Κλίμα				22 (MEMO)	7
36	4	Η πόλη της Άρτας				11 (MEMO)	3
37	4	Αξιοθέατα - Μνημεία			28	24 (MEMO)	8
38	6	Εισαγωγή				42 (MEMO)	1
39	6	Δημογραφικά Στοιχεία				50 (MEMO)	2
40	6	Η πόλη των Ιωαννίνων				43 (MEMO)	3
41	6	Βουνά και πεδιάδες				45 (MEMO)	4
42	6	Ποτάμια				44 (MEMO)	5
43	6	Λίμνες, λιμνοθάλασσες και ακτές				52 (MEMO)	6
44	6	Κλίμα				51 (MEMO)	7
45	6	Αξιοθέατα - Μνημεία				46 (MEMO)	8
46	7	Εισαγωγή				66 (MEMO)	1
47	7	Δημογραφικά Στοιχεία				64 (MEMO)	2
48	7	Η πόλη της Πρέβεζας				65 (MEMO)	3
49	7	Βουνά και πεδιάδες				67 (MEMO)	4
50	7	Ποτάμια				68 (MEMO)	5
51	7	Λίμνες, λιμνοθάλασσες και ακτές				69 (MEMO)	6
52	7	Κλίμα			-1	70 (MEMO)	7
53	7	Αξιοθέατα - Μνημεία				71 (MEMO)	8
54	8	Εισαγωγή				53 (MEMO)	1
55	8	Δημογραφικά Στοιχεία				54 (MEMO)	2
56	8	Η πόλη της Ηγουμενίτσας				55 (MEMO)	3
57	8	Βουνά και πεδιάδες				56 (MEMO)	4
58	8	Ποτάμια				57 (MEMO)	5

## EXTERNALS

Η οντότητα αυτή περιλαμβάνει όλα τα περιεχόμενα των διαφόρων εικόνων, πολυμέσων και χαρτών που χρησιμοποιήθηκαν στα κεφάλαια κάθε μαθήματος και αποτελείται από τα εξής γνωρίσματα:

- ID
- το path κάθε στοιχείου
- το όνομα κάθε στοιχείου



ID	Name	FileName
11	C:\Projects\Geography\gefyrri artas.bmp	gefyrri artas.bmp
12	C:\Projects\Geography\GreekMap.bmp	GreekMap.bmp
13	C:\Projects\Geography\greece-map-prefectures.gif	greece-map-prefectures.gif
14	C:\Projects\Geography\GreekMap-nomoi.bmp	GreekMap-nomoi.bmp
15	C:\Projects\Geography\GreekMap-nomoi.bmp	GreekMap-nomoi.bmp
16	C:\Projects\Geography\gefyrri artas.bmp	gefyrri artas.bmp
17	C:\Projects\Geography\tzoumerka.bmp	tzoumerka.bmp
18	C:\Projects\Geography\Ellada-nomoi-noletters.bmp	Ellada-nomoi-noletters.bmp
19	C:\Projects\Geography\tzoumerka.bmp	tzoumerka.bmp
20	C:\Projects\Geography\Lesson Material\imnes\amvarikos.jpg	amvarikos.jpg
21	C:\Projects\Geography\xarta.bmp	xarta.bmp
22	C:\Projects\Geography\pournariou.bmp	pournariou.bmp
23	C:\Projects\Geography\axeloos.bmp	axeloos.bmp
24	C:\Projects\Geography\kastro.bmp	kastro.bmp
25	F:\ergasia\hpeiros\ωποια Ηνερος( Beautiful Epirus)_NEW.avi	ωποια Ηνερος( Beautiful Epirus)_NEW.avi
26	F:\ergasia\hpeiros\ΗΠΕΙΡΟΣ ΚΑΙ ΠΑΛΙ ΗΠΕΙΡΟΣ_NEW_NEW_NEW.avi	ΗΠΕΙΡΟΣ ΚΑΙ ΠΑΛΙ ΗΠΕΙΡΟΣ_NEW_NEW_NEW.avi
27	C:\Projects\Geography\Epirus1.AVI	Epirus1.AVI
28	C:\Projects\Geography\Epirus2.AVI	Epirus2.AVI
42	C:\Projects\Geography\nomosioanninwn-map.bmp	nomosioanninwn-map.bmp
43	C:\Projects\Geography\Ioannina.bmp	Ioannina.bmp
44	C:\Projects\Geography\ioan-potami.bmp	ioan-potami.bmp
45	C:\Projects\Geography\pindos-ioan.bmp	pindos-ioan.bmp
46	C:\Projects\Geography\ioan-surrako.bmp	ioan-surrako.bmp
47	C:\Projects\Geography\Dodona.bmp	Dodona.bmp
48	C:\Projects\Geography\ioan-nisi.bmp	ioan-nisi.bmp
49	C:\Projects\Geography\ioan-aoos.bmp	ioan-aoos.bmp
50	C:\Projects\Geography\Nomos-Ioan.bmp	Nomos-Ioan.bmp
51	C:\Projects\Geography\pramanta3.BMP	pramanta3.BMP
52	C:\Projects\Geography\IoanninaGenikiApopsi.BMP	IoanninaGenikiApopsi.BMP
53	C:\Projects\Geography\Lesson Material\8esprwtiagr.bmp	8esprwtiagr.bmp
54	C:\Projects\Geography\Lesson Material\HPEIROS\thes\map_munic.bmp	map_munic.bmp
55	C:\Projects\Geography\Lesson Material\Igoumenitsa 1.bmp	Igoumenitsa 1.bmp
56	C:\Projects\Geography\Lesson Material\HPEIROS\thes\souli.bmp	souli.bmp
57	C:\Projects\Geography\Lesson Material\HPEIROS\thes\skala_tsav.bmp	skala_tsav.bmp
58	C:\Projects\Geography\Lesson Material\HPEIROS\thes\subota2.bmp	subota2.bmp
59	C:\Projects\Geography\Lesson Material\HPEIROS\thes\ph_impanagias.bmp	ph_impanagias.bmp
60	C:\Projects\Geography\Lesson Material\HPEIROS\thes\Fanoth.bmp	Fanoth.bmp

## QUESTIONS

Η οντότητα αυτή περιλαμβάνει όλες τις ερωτήσεις που περιλαμβάνονται στα τεστ των μαθημάτων και αποτελείται από τα εξής γνωρίσματα:

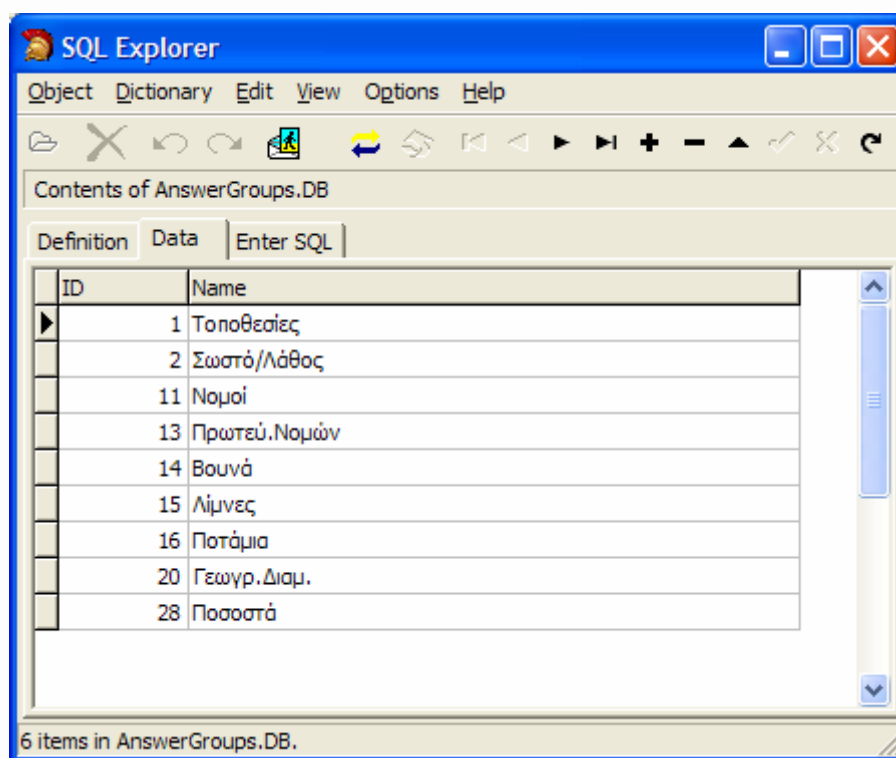
- το ID
- το ID του κάθε μαθήματος
- η σειρά με την οποία εμφανίζονται οι ερωτήσεις κάθε εξεταζόμενου μαθήματος
- το κείμενο κάθε ερώτησης
- την κατηγορία που ανήκει κάθε ερώτηση. Μπορεί να ανήκει στην κατηγορία ΠΟΤΑΜΙΑ, εάν πρόκειται για ερώτηση για ποτάμια
- το ID της σωστής απάντησης
- το ID της εικόνας που χρησιμοποιείται στην ερώτηση
- αν είναι ερώτηση χάρτη (True) ή όχι (null)

ID	LessonID	Ordering	Text	Category	CorrectAnswerID	PictureExternalID	ShowMap
31	4	1	Δείξε τον Νομό Άρτας στον χάρτη της Ελλάδας	ΧΑΡΤΗΣ	25		True
32	4	2	Σε ποιο γεωγραφικό διαμέρισμα ανήκει ο νομός Άρτας;	ΓΕΩΓΡ.ΔΙΑΜΕΡ.	55		
33	4	3	Ποιο είναι η πρωτεύουσα του νομού Άρτας;	ΠΡΩΤ.ΝΟΜΩΝ	38		
34	4	4	Ο νομός Άρτας συνορεύει με τον νομό Αιτωλοακαρνανίας;	ΣΥΝΟΡΑ ΝΟΜΩΝ	16		
35	4	5	Ο νομός Άρτας συνορεύει με τον νομό Ξάνθης;	ΣΥΝΟΡΑ ΝΟΜΩΝ	17		
36	4	6	Ο δήμος Πέτα ανήκει στον νομό Άρτας;	ΔΗΜΟΓΡΑΦΙΚΑ	16		
37	4	7	Σε τι ποσοστό καλύπτεται από βουνά ο νομός Άρτας;	ΔΗΜΟΓΡΑΦΙΚΑ	98		
38	4	8	Ποιο είναι το ψηλότερο βουνό του νομού Άρτας;	ΒΟΥΝΑ	44	17	
39	4	9	Το Σερβοβούνι είναι βουνό του νομού Άρτας;	ΒΟΥΝΑ	16		
40	4	10	Ο Αραχθός και ο Λούρος είναι ποτάμια του νομού Άρτας;	ΠΟΤΑΜΙΑ	16		
41	4	11	Ο Αλιάκμονας είναι ποτάμι του νομού Άρτας;	ΠΟΤΑΜΙΑ	17		
42	4	12	Ποια από τις παρακάτω λίμνες βρίσκεται στο νομό Άρτας;	ΛΙΜΝΕΣ	71		
43	4	13	Πάνω σε ποιον ποταμό είναι κτισμένο το γεφύρι της Άρτας;	ΠΟΤΑΜΙΑ	77	16	
44	6	1	Δείξε το νομό Ιωαννίνων στον χάρτη της Ελλάδας	ΧΑΡΤΗΣ	26		True
45	6	2	Σε ποιο γεωγραφικό διαμέρισμα ανήκει ο νομός Ιωαννίνων;	ΓΕΩΓΡ.ΔΙΑΜΕΡ.	55		
46	6	3	Ποιο είναι η πρωτεύουσα του νομού Ιωαννίνων;	ΠΡΩΤ.ΝΟΜΩΝ	39	43	
47	6	6	Ο δήμος Δαυδίνης ανήκει στον νομό Ιωαννίνων;	ΔΗΜΟΓΡΑΦΙΚΑ	16	47	
48	6	7	Το νησί στη λίμνη των Ιωαννίνων αποτελεί καινήττα;	ΔΗΜΟΓΡΑΦΙΚΑ	16	48	
49	6	8	Ποιο είναι το ψηλότερο βουνό του νομού Ιωαννίνων;	ΒΟΥΝΑ	104		
50	6	9	Το Αθαμανικό Όρη (Τουμέρικο) είναι βουνό του νομού Ιωαννίνων	ΒΟΥΝΑ	16		
51	6	4	Ο νομός Ιωαννίνων συνορεύει με το νομό Άρτας;	ΣΥΝΟΡΑ ΝΟΜΩΝ	16		
52	6	5	Ο νομός Ιωαννίνων συνορεύει με το νομό Πέλλας;	ΣΥΝΟΡΑ ΝΟΜΩΝ	17		
53	6	10	Ποιος από τους παρακάτω ποταμούς διαρρέει τον νομό Ιωαννίνων και συνεχίζει προς Αλβανικό έδαφος;	ΠΟΤΑΜΙΑ	105	49	
54	6	11	Ο ποταμός Αχέρωντας διαρρέει τον νομό Ιωαννίνων;	ΠΟΤΑΜΙΑ	16		
55	6	12	Η πόλη των Ιωαννίνων είναι χτισμένη στις όχθες της λίμνης Παυβάπδας;	ΛΙΜΝΕΣ	16		
56	6	13	Η λίμνη Μικρή Πρέσα βρίσκεται στο νομό Ιωαννίνων;	ΛΙΜΝΕΣ	17		
57	7	1	Δείξε το νομό Πρεβέζης στον χάρτη της Ελλάδας	ΧΑΡΤΗΣ	114		True
58	7	2	Σε ποιο γεωγραφικό διαμέρισμα ανήκει ο νομός Πρεβέζης;	ΓΕΩΓΡ.ΔΙΑΜΕΡ.	55		
59	7	3	Ποιο είναι η πρωτεύουσα του νομού Πρεβέζης;	ΠΡΩΤ.ΝΟΜΩΝ	115	65	
60	7	4	Ο νομός Πρεβέζης συνορεύει με το νομό Άρτας;	ΣΥΝΟΡΑ ΝΟΜΩΝ	16		
61	7	5	Ο νομός Πρεβέζης συνορεύει με το νομό Λαρίσης;	ΣΥΝΟΡΑ ΝΟΜΩΝ	17		
62	7	6	Ο δήμος Πάργας ανήκει στον νομό Πρεβέζης;	ΔΗΜΟΓΡΑΦΙΚΑ	16	72	
63	7	7	Ο δήμος Ηγουμενίτσας ανήκει στο νομό Πρεβέζης;	ΔΗΜΟΓΡΑΦΙΚΑ	17		
64	7	8	Ποιο είναι το ψηλότερο βουνό του νομού Πρεβέζης;	ΒΟΥΝΑ	116	71	
65	7	9	Τα Θεσπρωτικά Όρη είναι βουνό και του νομού Πρεβέζης;	ΒΟΥΝΑ	16		
66	7	10	Ποιος από τους παρακάτω ποταμούς του νομού Πρεβέζης εκβάλει στο Ιόνιο;	ΠΟΤΑΜΙΑ	107	68	
67	7	12	Η λίμνη Ζηρού είναι η μεγαλύτερη λίμνη της Ηπείρου;	ΛΙΜΝΕΣ	17	69	
68	7	13	Η λίμνη Βιστωνίδα βρίσκεται στο νομό Πρεβέζης;	ΛΙΜΝΕΣ	17		

## ANSWER GROUPS

Η οντότητα αυτή περιλαμβάνει όλες τις ομάδες που έχουν χωριστεί οι απαντήσεις των ερωτήσεων που περιλαμβάνονται στα τεστ των μαθημάτων και αποτελείται από τα εξής γνωρίσματα:

- το ID
- το όνομα της κάθε ομάδας απαντήσεων. Αν κάποια απάντηση αφορά κάποια λίμνη, τότε είναι στην κατηγορία Λίμνες



ID	Name
1	Τοποθεσίες
2	Σωστό/Λάθος
11	Νομοί
13	Πρωτεύ.Νομών
14	Βουνά
15	Λίμνες
16	Ποτάμια
20	Γεωγρ.Διαμ.
28	Ποσοστά

### **ANSWERS**

Η οντότητα αυτή περιλαμβάνει όλες τις απαντήσεις των ερωτήσεων που περιλαμβάνονται στα τεστ των μαθημάτων και αποτελείται από τα εξής γνωρίσματα:

- το ID
- το κείμενο της απάντησης
- το ID της ομάδας που ανήκει η απάντηση
- αν ανήκει σε τοποθεσία (true) ή όχι (false)
- το ID της τοποθεσίας στον χάρτη
- η διάσταση x στον χάρτη
- η διάσταση y στον χάρτη

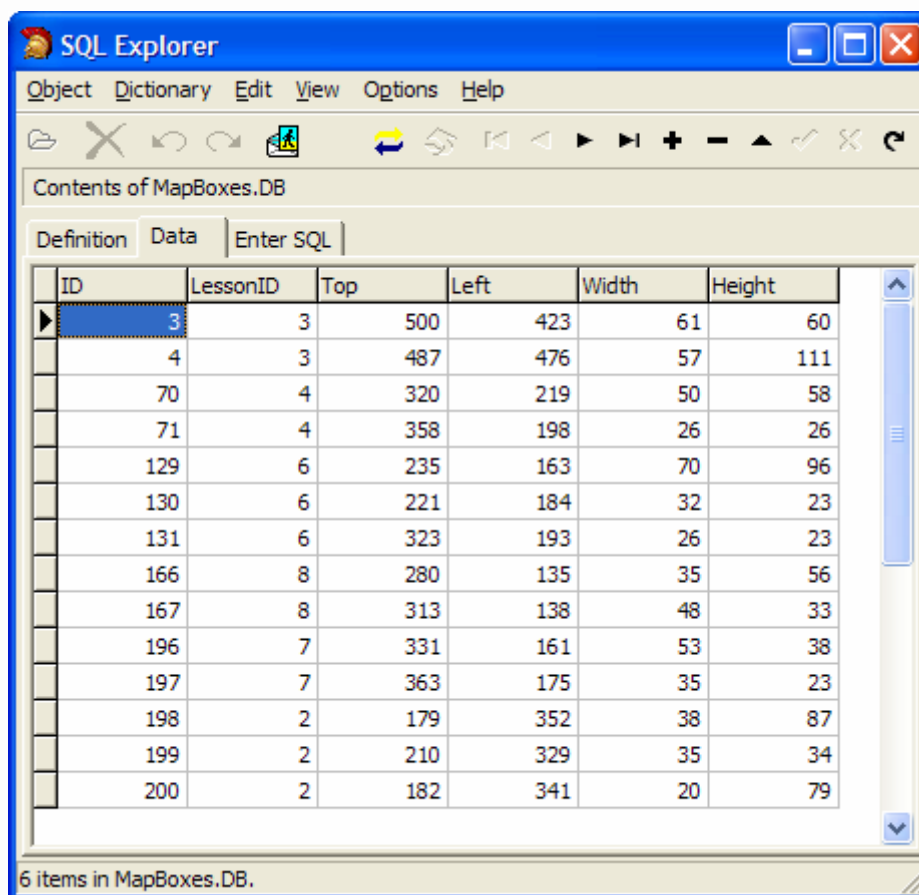
ID	Text	Group	IsPlace	MapExternalID	MapX	MapY
18	Νομός Αττικής	11	True	18	393	445
19	Νομός Εύβοιας	11	True	18	383	377
20	Νομός Ευρυτανίας	11	True	18	188	311
21	Νομός Ημαθίας	11	True	18	265	124
22	Νομός Πιερίας	11	True	18	275	166
23	Νομός Κιλκίς	11	True	18	319	62
24	Νομός Πέλλας	11	True	18	267	78
25	Νομός Άρτας	11	True	18	140	275
26	Νομός Ιωαννίνων	11	True	18	120	200
27	Νομός Απωλοκαρνανίας	11	True	18	150	353
28	Νομός Τρικάλων	11	True	18	192	236
29	Νομός Αχαΐας	11	True	18	218	433
30	Νομός Καστοριάς	11	True	18	156	124
31	Χαλκίδα	13	True	18	383	385
32	Καρπενήσι	13	True	18	186	321
33	Αθήνα	13	True	18	383	451
34	Βέροια	13	True	18	265	128
35	Κατερίνη	13	True	18	277	166
36	Κιλκίς	13	True	18	323	68
37	Έδεσσα	13	True	18	257	84
38	Άρτα	13	True	18	142	283
39	Ιωάννινα	13	True	18	126	204
40	Μεσολόγγι	13	True	18	150	373
41	Τρίκαλα	13	True	18	186	236
42	Πάτρα	13	True	18	194	419
43	Καστοριά	13	True	18	158	130
44	Αθωμανικά Όρη (Τζουμέρκα)	14	True	18	138	279
45	Γράμμος	14	True	18	186	144
46	Βίτσι	14	True	18	196	196
47	Όλυμπος	14	True	18	279	172
48	Παρνασσός	14	True	18	373	435
49	Οίτη	14	True	18	234	313
50	Πήλιο	14	True	18	299	283
51	Τόη (Ψηλορείτης)	14	True	18	473	796
52	Ταΰγετος	14	True	18	234	561
53	Πάρνωνας	14	True	18	234	505
54	Τυμφριστός (Βελούχι)	14	True	18	259	359
55	Ήπειρος	20	True	18	96	222

### MAP BOXES

Η οντότητα αυτή περιλαμβάνει όλες τις διαστάσεις των τοποθεσιών στον χάρτη και αποτελείται από τα εξής γνωρίσματα:

- το ID
- το ID του μαθήματος
- τη διάσταση στο πάνω μέρος του χάρτη
- τη διάσταση στο δεξί μέρος του χάρτη
- τη διάσταση του πλάτους στον χάρτη
- τη διάσταση του ύψους στον χάρτη



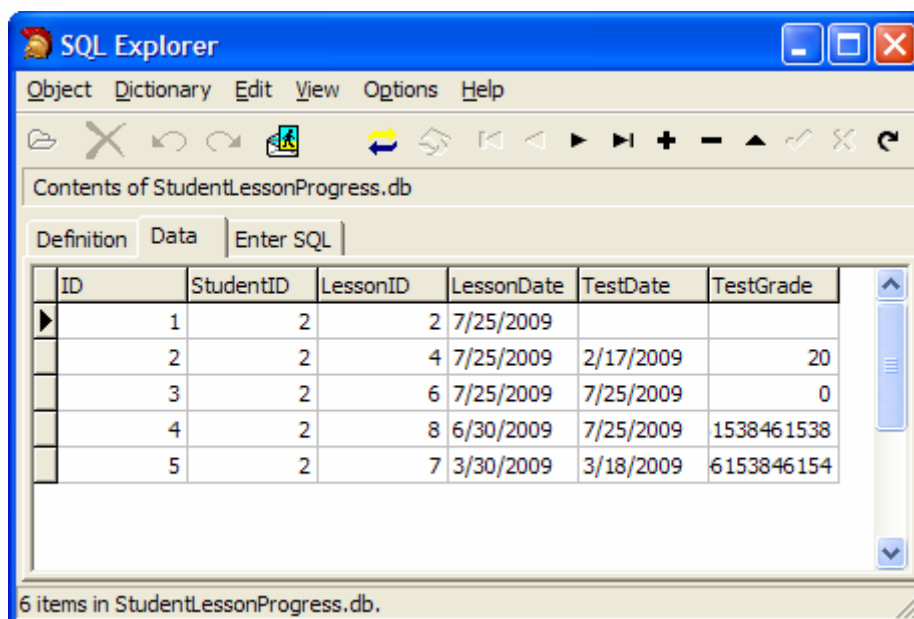


ID	LessonID	Top	Left	Width	Height
3	3	500	423	61	60
4	3	487	476	57	111
70	4	320	219	50	58
71	4	358	198	26	26
129	6	235	163	70	96
130	6	221	184	32	23
131	6	323	193	26	23
166	8	280	135	35	56
167	8	313	138	48	33
196	7	331	161	53	38
197	7	363	175	35	23
198	2	179	352	38	87
199	2	210	329	35	34
200	2	182	341	20	79

### **STUDENT LESSON PROGRESS**

Η οντότητα αυτή περιλαμβάνει την απόδοση του μαθητή σε κάθε τεστ και αποτελείται από τα εξής γνωρίσματα:

- το ID
- το ID του μαθητή
- το ID του μαθήματος
- η ημερομηνία παρακολούθησης του μαθήματος
- η ημερομηνία εξέτασης του τεστ
- ο βαθμός που έλαβε ο μαθητής στο τεστ, εάν το έχει ολοκληρώσει. Εάν δεν το έχει ολοκληρώσει, τότε εμφανίζεται η τιμή μηδέν.



Contents of StudentLessonProgress.db

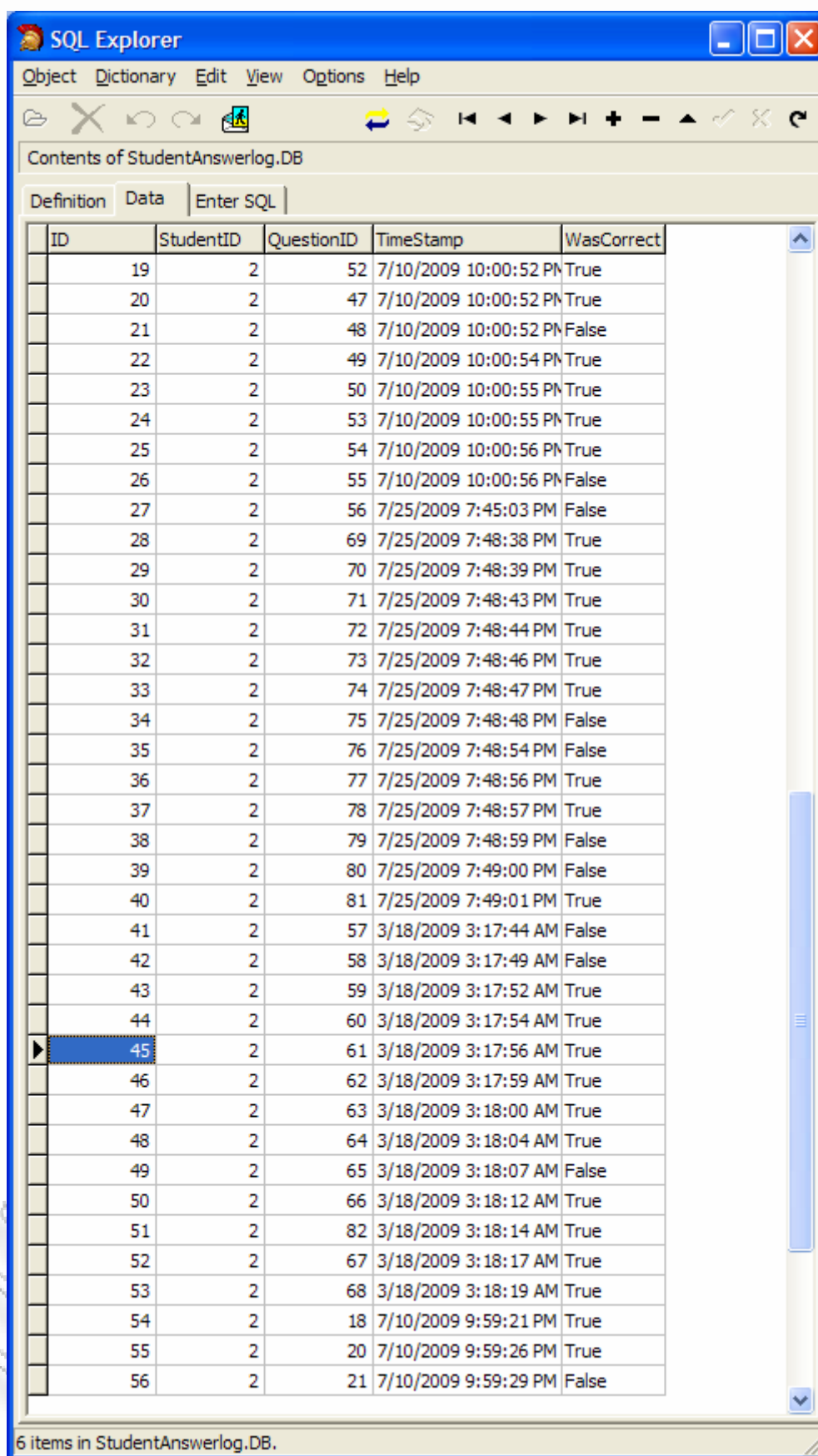
ID	StudentID	LessonID	LessonDate	TestDate	TestGrade
1	2	2	7/25/2009		
2	2	4	7/25/2009	2/17/2009	20
3	2	6	7/25/2009	7/25/2009	0
4	2	8	6/30/2009	7/25/2009	1538461538
5	2	7	3/30/2009	3/18/2009	6153846154

6 items in StudentLessonProgress.db.

### **STUDENT ANSWER**

Η οντότητα αυτή περιλαμβάνει όλες τις απαντήσεις των ερωτήσεων που έδωσε ο μαθητής κατά τη διάρκεια των τεστ και αποτελείται από τα εξής γνωρίσματα:

- το ID
- το ID του μαθητή που έδωσε την απάντηση
- το ID της ερώτησης που εξετάστηκε ο μαθητής
- την ημερομηνία και ώρα που δόθηκε η κάθε απάντηση
- αν η απάντηση ήταν σωστή (true) ή λανθασμένη (false)



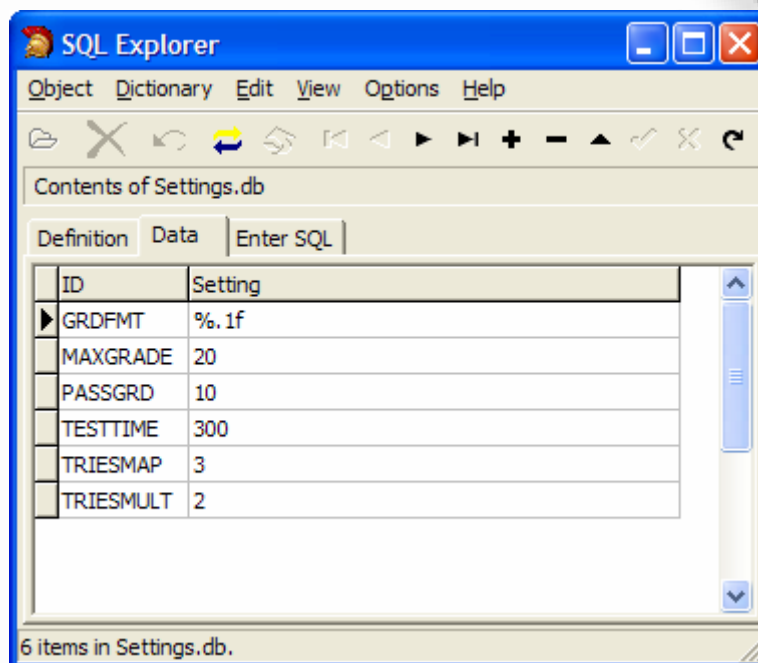
The screenshot shows the SQL Explorer application window. The title bar reads "SQL Explorer". The menu bar includes "Object", "Dictionary", "Edit", "View", "Options", and "Help". Below the menu bar is a toolbar with various icons for navigation and editing. The main area displays the "Contents of StudentAnswerlog.DB" with tabs for "Definition", "Data", and "Enter SQL". The "Data" tab is active, showing a table with the following columns: ID, StudentID, QuestionID, TimeStamp, and WasCorrect. The table contains 38 rows of data, with row 45 highlighted. The status bar at the bottom indicates "6 items in StudentAnswerlog.DB".

ID	StudentID	QuestionID	TimeStamp	WasCorrect
19	2	52	7/10/2009 10:00:52 PM	True
20	2	47	7/10/2009 10:00:52 PM	True
21	2	48	7/10/2009 10:00:52 PM	False
22	2	49	7/10/2009 10:00:54 PM	True
23	2	50	7/10/2009 10:00:55 PM	True
24	2	53	7/10/2009 10:00:55 PM	True
25	2	54	7/10/2009 10:00:56 PM	True
26	2	55	7/10/2009 10:00:56 PM	False
27	2	56	7/25/2009 7:45:03 PM	False
28	2	69	7/25/2009 7:48:38 PM	True
29	2	70	7/25/2009 7:48:39 PM	True
30	2	71	7/25/2009 7:48:43 PM	True
31	2	72	7/25/2009 7:48:44 PM	True
32	2	73	7/25/2009 7:48:46 PM	True
33	2	74	7/25/2009 7:48:47 PM	True
34	2	75	7/25/2009 7:48:48 PM	False
35	2	76	7/25/2009 7:48:54 PM	False
36	2	77	7/25/2009 7:48:56 PM	True
37	2	78	7/25/2009 7:48:57 PM	True
38	2	79	7/25/2009 7:48:59 PM	False
39	2	80	7/25/2009 7:49:00 PM	False
40	2	81	7/25/2009 7:49:01 PM	True
41	2	57	3/18/2009 3:17:44 AM	False
42	2	58	3/18/2009 3:17:49 AM	False
43	2	59	3/18/2009 3:17:52 AM	True
44	2	60	3/18/2009 3:17:54 AM	True
45	2	61	3/18/2009 3:17:56 AM	True
46	2	62	3/18/2009 3:17:59 AM	True
47	2	63	3/18/2009 3:18:00 AM	True
48	2	64	3/18/2009 3:18:04 AM	True
49	2	65	3/18/2009 3:18:07 AM	False
50	2	66	3/18/2009 3:18:12 AM	True
51	2	82	3/18/2009 3:18:14 AM	True
52	2	67	3/18/2009 3:18:17 AM	True
53	2	68	3/18/2009 3:18:19 AM	True
54	2	18	7/10/2009 9:59:21 PM	True
55	2	20	7/10/2009 9:59:26 PM	True
56	2	21	7/10/2009 9:59:29 PM	False

## SETTINGS

Η οντότητα αυτή περιλαμβάνει όλες τις ρυθμίσεις των τεστ που καθορίζονται από τον administrator και αποτελείται από τα εξής γνωρίσματα:

- το όνομα της ρύθμισης ή το ID
- την τιμή της παραμέτρου



Η γλώσσα προγραμματισμού που χρησιμοποιήθηκε είναι η BDS Delphi 2006, η γλώσσα αυτή παρέχει την δυνατότητα χειριστηρίων μέσω των οποίων εκτελούμε τα sql queries αλλά προτιμήθηκε η διασύνδεση της βάσης με κάθε φόρμα κατευθείαν μέσω κώδικα για την πιο σωστή δόμηση του κώδικα και την καλύτερη εποπτεία του κώδικα έτσι ώστε να αποφευχθεί οποιοδήποτε λάθος.

Πάνω απ' όλα οι στόχοι που τέθηκαν ήταν να σχεδιαστεί μια εφαρμογή με τα εξής χαρακτηριστικά:

- Φιλική προς το χρήστη : Η εφαρμογή δεν μπερδεύει το χρήστη, αλλά τον καθοδηγεί με τρόπο φιλικό και απλό. Είναι ιδιαίτερα εύχρηστη ακόμα και για τους όχι και τόσο εξοικειωμένους με τους ηλεκτρονικούς υπολογιστές, και στην συγκεκριμένη περίπτωση για τους μαθητές, εφόσον παρέχει ευκολία εκμάθησης και μέσω του on-line help. Επίσης δόθηκε ιδιαίτερη προσοχή, ώστε η εφαρμογή να είναι εύκαμπτη υποστηρίζοντας πολλούς τρόπους επικοινωνίας με τον χρήστη και ανθεκτική ώστε να υποστηρίζει τον χρήστη στα όποια λάθη ενδέχεται να κάνει.
- Συνεπής : Η εφαρμογή ακολουθεί κάποιους κανόνες που έχουν να κάνουν κυρίως με το σχεδιαστικό κομμάτι της. Για παράδειγμα τα χρώματα και οι λέξεις που χρησιμοποιήθηκαν σημαίνουν πάντα τα ίδια πράγματα κάθε φορά. Η συνέπεια βοηθά και διευκολύνει το χρήστη.
- Γρήγορα δοκιμασμένη : Έγιναν πολλά παραδείγματα και έλεγχοι, ώστε η εφαρμογή να εκτελεί τις διαδικασίες επιτυχώς.
- Προβλέψιμη: Δηλαδή να υποστηρίζει το χρήστη, ώστε να μπορεί εύκολα να καθορίσει το αποτέλεσμα μίας μελλοντικής ενέργειας βασισμένος σε εμπειρία παλιότερης επικοινωνίας με το σύστημα.

## **ΚΕΦΑΛΑΙΟ 5**

### **ΣΧΕΔΙΑΣΜΟΣ ΔΙΕΠΑΦΗΣ ΜΕ ΧΡΗΣΤΕΣ-ΠΑΡΟΥΣΙΑΣΗ ΕΦΑΡΜΟΓΗΣ**

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΡΑΙΑ

## 5.1. ΕΙΣΑΓΩΓΗ

Ο σχεδιασμός ενός συστήματος διεπαφής του χρήστη με το σύστημα, ότι δικαιοδοσίες κι αν έχει ο χρήστης αυτός σε σχέση με το σύστημα, είναι αναπόσπαστο κομμάτι του σχεδιασμού του λογισμικού. Στο έκτο μέρος του manual της κατασκευής του συστήματος του μαθήματος της «e-School» παρουσιάζονται οι βασικές αρχές για την κατασκευή ενός αποδοτικού περιβάλλοντος διεπαφής με τους χρήστες και η ακριβής περιγραφή της μεθόδου που ακολουθήθηκε για την κατασκευή του κατάλληλου περιβάλλοντος για τη συγκεκριμένη εφαρμογή.

Σε κάθε σύστημα λογισμικού πρέπει να υπάρχει κάποιος τρόπος επικοινωνίας και αλληλεπίδρασης του χρήστη με το σύστημα. Ο πιο προφανής τρόπος επίτευξης αυτού του στόχου είναι ο σχεδιασμός του κατάλληλου κάθε φορά περιβάλλοντος, μέσα από το οποίο ο εκάστοτε χρήστης θα μπορεί με εύκολο τρόπο να προσπελαύνει τις πληροφορίες που τον ενδιαφέρουν.

Ο καλός σχεδιασμός του συστήματος διεπαφής με το χρήστη είναι αποφασιστικής σημασίας για την επιτυχία του συστήματος. Ένα δύσκολο στη χρήση σύστημα διεπαφής στην καλύτερη περίπτωση θα έχει ως αποτέλεσμα πολλά λάθη από πλευράς του χρήστη. Στη χειρότερη περίπτωση απλά οι χρήστες θα αρνηθούν να χρησιμοποιήσουν το σύστημα λογισμικού ανεξάρτητα από τη λειτουργικότητά του. Αν οι πληροφορίες παρουσιάζονται με μπερδεμένο ή παραπλανητικό τρόπο, οι χρήστες μπορεί να παρανοήσουν τη σημασία των πληροφοριών. Μπορεί να ακολουθήσουν μία ακολουθία ενεργειών που διαστρεβλώνουν τα δεδομένα ή ακόμα μπορεί να προκαλέσουν ανεπανόρθωτη βλάβη στο σύστημα.

Για την κατασκευή ενός περιβάλλοντος που εκπληρώνει τις απαιτήσεις του συστήματος θα πρέπει κατ' αρχήν να ληφθεί υπ' όψη σε ποιόν απευθύνεται. Η κάθε κατηγορία χρήστη έχει και ένα διαφορετικό τρόπο να μαθαίνει να αλληλεπιδρά με τη μηχανή, έχει διαφορετικές προσδοκίες από το σύστημα και ερμηνεύει με διαφορετικό τρόπο τις πληροφορίες που παράγει το σύστημα. Οι χρήστες του περιβάλλοντος διεπαφής για το σύστημα μηχανογράφησης του εφαρμογής είναι κυρίως ο administrator, αλλά και οι καθηγητές που τους έχουν δοθεί δικαιώματα administrator.

Ο κάθε χρήστης αντιλαμβάνεται διαφορετικά κάθε είδους πληροφορία και ειδικά αυτές που προέρχονται από ένα μηχανογραφημένο σύστημα. Ένα σύστημα διεπαφής επιτρέπει στο χρήστη να αντιληφθεί κάποιες πληροφορίες, να τις αποθηκεύσει στη μνήμη του και να τις επεξεργαστεί χρησιμοποιώντας τη λογική του. Η αλληλεπίδραση ενός περιβάλλοντος με το χρήστη επιτυγχάνεται κυρίως διαμέσου οπτικών εικόνων, όπως κειμένου, γραφικών και άλλων τεχνικών. Το μάτι και το μυαλό συνεργάζονται στην αντίληψη και ερμηνεία των οπτικών πληροφοριών. Ο άνθρωπος πρέπει να αποκωδικοποιήσει τα οπτικά σχήματα και να κάνει ανάκτηση των εννοιών των λέξεων. Όταν η πληροφορία εξαχθεί από το σύστημα διεπαφής πρέπει να αποθηκευτεί στη μνήμη. Επίσης ο χρήστης πρέπει να θυμάται εντολές, καταστάσεις λαθών, κλπ. Για το λόγο αυτό το σύστημα πρέπει να κατασκευαστεί με τέτοιο τρόπο ώστε να είναι απλό στη χρήση, εύκολο στην εκμάθηση, να μην προκαλεί σύγχυση και να συγχωρεί πιθανά λάθη του χρήστη.

Οι σχεδιαστές των συστημάτων διεπαφής πρέπει να λαμβάνουν υπ' όψη τους τις φυσικές και πνευματικές ικανότητες των ανθρώπων που χρησιμοποιούν το λογισμικό. Οι άνθρωποι έχουν περιορισμένη Μνήμη Μικρής Διάρκειας και κάνουν λάθη, ειδικά όταν έχουν να κάνουν με πολλές πληροφορίες ή έχουν κάποιου είδους άγχος. Οι ανθρώπινες ικανότητες είναι η βάση των αρχών που διέπουν το σχεδιασμό των συστημάτων διεπαφής, οι οποίες περιγράφονται στη συνέχεια. Αυτές είναι γενικές αρχές που είναι εφαρμόσιμες σε όλα τα περιβάλλοντα διεπαφής με το χρήστη και θα πρέπει να ενταχθούν στο σχεδιασμό για συγκεκριμένους οργανισμούς ή τύπους του συστήματος.

Η οικειότητα του χρήστη με το σύστημα υποστηρίζει ότι οι χρήστες δε θα πρέπει να αναγκάζονται να προσαρμόζονται σε κάθε περιβάλλον διεπαφής επειδή είναι βολικό στην εφαρμογή. Το περιβάλλον που θα κατασκευαστεί πρέπει να χρησιμοποιεί όρους οικείους στο χρήστη και τα αντικείμενα που χρησιμοποιούνται από το σύστημα πρέπει να συνδέονται άμεσα με το περιβάλλον του χρήστη. Για παράδειγμα, στην εφαρμογή της μηχανογράφησης

της εφαρμογής της «e-School» θα ήταν χρήσιμο να χρησιμοποιούνται αντικείμενα, όπως τα χάρτες και εικόνες των νομών, που να θυμίζουν στο χρήστη το σκοπό της εφαρμογής και τις αντίστοιχες δυνατότητές του. Τα αρχεία και οι δομές δεδομένων που χρησιμοποιούνται για την υλοποίηση του περιβάλλοντος δεν θα πρέπει να είναι ορατά από τον τελικό χρήστη.

Ο κανόνας της συνέπειας του περιβάλλοντος διεπαφής με το χρήστη σημαίνει ότι οι εντολές και τα μενού του συστήματος θα πρέπει να έχουν την ίδια μορφή και οι παράμετροι θα πρέπει να εισάγονται σε όλες τις εντολές με τον ίδιο τρόπο. Τα συνεπή περιβάλλοντα μειώνουν το χρόνο εκμάθησης του χρήστη. Η γνώση που λαμβάνει ο χρήστης με μία εντολή ή εφαρμογή θα πρέπει να είναι εφαρμόσιμη και σε άλλα μέρη του συστήματος διεπαφής.

Η αρχή της ελάχιστης έκπληξης είναι απαραίτητη επειδή οι χρήστες ενοχλούνται όταν το σύστημα συμπεριφέρεται με απροσδόκητο τρόπο. Καθώς χρησιμοποιείται το σύστημα, οι χρήστες κατασκευάζουν ένα νοητικό μοντέλο για το πώς λειτουργεί το σύστημα. Εάν μία ενέργεια σε κάποιο σημείο προκαλεί μία συγκεκριμένη αλλαγή, είναι λογικό να υποθεθεί ότι η ίδια ενέργεια σε ένα διαφορετικό σημείο θα προκαλέσει μία παρόμοια αλλαγή. Αν συμβεί κάτι εντελώς διαφορετικό ο χρήστης θα εκπλαγεί και θα μπερδευτεί. Οι σχεδιαστές λοιπόν των περιβαλλόντων διεπαφής θα πρέπει να διασφαλίζουν ότι παρόμοιες ενέργειες θα έχουν και παρόμοιες επιπτώσεις.

Ένας πολύ χρήσιμος κανόνας για το σχεδιασμό ενός σταθερού συστήματος διεπαφής με το χρήστη είναι αυτός της δυνατότητας επαναφοράς της κατάστασης πριν από μια ενέργεια. Αυτό είναι απαραίτητο γιατί ο χρήστης μπορεί να μην καταφέρει να αξιολογήσει επαρκώς την κατάσταση στην οποία βρίσκεται μετά από κάποια ενέργεια. Μπορεί επίσης να μην έχει συνειδητοποιήσει καθόλου ποια ενέργεια τον οδήγησε στην παρούσα κατάσταση. Το αποτέλεσμα αυτών μπορεί να είναι η αδυναμία του να βρει τρόπο ανάνηψης από κάποιο λάθος που μπορεί να έκανε. Ο σχεδιασμός του περιβάλλοντος πρέπει από μόνος του να είναι σε θέση να ελαχιστοποιεί την πιθανότητα διάπραξης κάποιου λάθους, αν και η πιθανότητα αυτή, όσο καλός και να είναι ο σχεδιασμός, δεν θα μηδενιστεί ποτέ. Υπάρχουν δύο τύπου δραστηριότητες που μπορεί να εμπεριέχει ένα σύστημα διεπαφής και να υποστηρίζουν την αρχή της επαναφοράς από πιθανά λάθη του χρήστη. Σύμφωνα με τον πρώτο τύπο δραστηριοτήτων, αν οι χρήστες για παράδειγμα θελήσουν να κάνουν μία ενέργεια που είναι πιθανώς καταστροφική, θα πρέπει να τους ζητείται επιβεβαίωση ότι πραγματικά θέλουν να κάνουν αυτή την ενέργεια πριν καταστραφεί οποιαδήποτε πληροφορία. Μια άλλη περίπτωση που ανήκει στην ίδια κατηγορία είναι αυτή στην οποία το σύστημα προστατεύει το χρήστη από ενδεχόμενα λάθη απενεργοποιώντας επιλογές που μπορεί να μην έχουν νόημα για μία συγκεκριμένη εργασία. Μια δεύτερη τακτική είναι η δυνατότητα του συστήματος να επανέρχεται στην κατάσταση στην οποία βρισκόταν πριν λάβει χώρα η ανεπιθύμητη πράξη. Πολλαπλά επίπεδα επαναφοράς είναι χρήσιμα, καθώς οι χρήστες δεν αναγνωρίζουν πάντα αμέσως ότι έχει γίνει κάποιο λάθος.

Ένας σχετικός κανόνας είναι αυτός της υποστήριξης των χρηστών. Τα περιβάλλοντα διεπαφής θα πρέπει να περιλαμβάνουν κάποια μορφή παροχής βοήθειας προς το χρήστη, αφού όλοι οι χρήστες αλληλεπιδραστικών συστημάτων χρειάζονται εκπαίδευση για να καταλάβουν πώς να χρησιμοποιούν σωστά το σύστημα. Η μεγαλύτερη δυσκολία στην εκμάθηση απορρέει από τον κακό σχεδιασμό των οθονών, των μενού επιλογών, των εντολών που μπορεί να οδηγήσουν σε λάθη ή από την αδυναμία των χρηστών να καταλάβουν τι πρέπει να κάνουν. Όσο καλός λοιπόν κι αν είναι ο σχεδιασμός του περιβάλλοντος διεπαφής, θα πρέπει να συνοδεύεται και από κατάλληλο συνοδευτικό υλικό το οποίο θα ήταν καλό να βρίσκεται και σε μορφή εκτύπωσης αλλά και σε μορφή on-line.

Το σύστημα διεπαφής θα πρέπει εκτός των άλλων να αναγνωρίζει ότι για τα περισσότερα αλληλεπιδραστικά συστήματα υπάρχουν πολλοί τύποι χρηστών. Κάποιοι χρήστες θα χρησιμοποιούν το σύστημα κατά περιόδους, ενώ άλλοι θα χρησιμοποιούν το σύστημα αρκετές ώρες την ημέρα. Στην εφαρμογή που αναπτύσσεται δεν υπάρχουν περιστασιακοί χρήστες. Όλοι οι χρήστες θα προσπελαίνουν το σύστημα σχεδόν καθημερινά. Οι περιστασιακοί χρήστες χρειάζονται περιβάλλοντα που παρέχουν καθοδήγηση, ενώ οι εν δυνάμει χρήστες απαιτούν συντομεύσεις που θα τους επιτρέπουν να αλληλεπιδρούν με το σύστημα όσο το δυνατόν ταχύτερα. Επιπλέον, οι χρήστες μπορεί να υποφέρουν από διαφορετικού τύπου ανικανότητες και, αν είναι εφικτό, το περιβάλλον διεπαφής θα πρέπει να είναι προσαρμόσιμο για να μπορεί να αντιμετωπίζει τέτοιες ανικανότητες.

Η αρχή της αναγνώρισης της ποικιλομορφίας των χρηστών μπορεί να έρχεται σε σύγκρουση με τους άλλους κανόνες σχεδίασης των συστημάτων διεπαφής, κυρίως επειδή κάποιες κατηγορίες χρηστών μπορεί να προτιμούν πιο γρήγορη αλληλεπίδραση έναντι, για παράδειγμα, της συνέπειας του περιβάλλοντος. Όμοια, ο βαθμός της απαιτούμενης καθοδήγησης του χρήστη μπορεί να είναι ριζικά διαφορετική για διαφορετικές κατηγορίες χρηστών και μπορεί να είναι αδύνατο να αναπτυχθεί υποστήριξη κατάλληλη για όλες τις κατηγορίες χρηστών. Ο σχεδιαστής του περιβάλλοντος πρέπει αναπόφευκτα να κάνει συμβιβασμούς, βασιζόμενος πάντα στους πραγματικούς χρήστες του συστήματος. Όπως έχει ήδη αναφερθεί, οι χρήστες του συστήματος που θα καθορίσουν το σχεδιασμό του περιβάλλοντος διεπαφής για την εφαρμογή «e-School» είναι οι καθηγητές, οι μαθητές και ο administrator.

## 5.2. ΠΑΡΟΥΣΙΑΣΗ ΤΩΝ ΠΛΗΡΟΦΟΡΙΩΝ ΣΤΟΥΣ ΧΡΗΣΤΕΣ

Αφού αναλύθηκαν οι γενικές αρχές που πρέπει να διέπουν ένα σύστημα διεπαφής για να αποδοτικό και να εκπληρώνει τους στόχους του, θα γίνει μια παρουσίαση κάποιων θεμάτων που αφορούν την παρουσίαση των πληροφοριών που θα παρέχει το σύστημα στους χρήστες. Η πολιτική που θα ακολουθείται είναι, όπως και στα προηγούμενα κεφάλαια, να παρουσιαστούν αρχικά κάποιες γενικές θεωρητικές αρχές και μετά να γίνει η αναλυτική περιγραφή του τρόπου κατασκευής του περιβάλλοντος διεπαφής για την εφαρμογή «e-School».

Όλα τα αλληλεπιδραστικά συστήματα παρέχουν κάποιο τρόπο παρουσίασης των διαθέσιμων πληροφοριών στους χρήστες. Οι διαθέσιμες πληροφορίες είναι προφανώς διαφορετικές για κάθε κατηγορία χρηστών για λόγους που έχουν ήδη αναφερθεί προηγούμενες σε αυτή την ενότητα. Η παρουσίαση των πληροφοριών μπορεί απλά να είναι μία άμεση αναπαράσταση των πληροφοριών που εισάγονται στο σύστημα, όπως κείμενο σε ένα επεξεργαστή κειμένου, ή μπορεί να παρουσιάζει τις πληροφορίες με γραφικό τρόπο. Είναι μια καλή πρακτική σχεδίασης να διαχωριστεί το λογισμικό που είναι απαραίτητο για την παρουσίαση των πληροφοριών από τις ίδιες τις πληροφορίες. Μέχρι κάποιου σημείου, αυτό έρχεται σε αντίθεση με την αντικειμενοστραφή φιλοσοφία που υποστηρίζει ότι οι λειτουργίες στις οποίες εμπλέκονται τα δεδομένα πρέπει να ορίζονται από κοινού με τα δεδομένα. Παρόλα αυτά, αυτό προϋποθέτει ότι ο σχεδιαστής των αντικειμένων γνωρίζει πάντα ποιος είναι ο καλύτερος τρόπος παρουσίασης των πληροφοριών. Αυτό φυσικά δεν είναι πάντα αληθές. Συνήθως είναι δύσκολο να ξέρουμε τον καλύτερο τρόπο παρουσίασης των δεδομένων όταν γίνεται ο ορισμός τους και οι δομές αντικειμένων δεν θα πρέπει να περιορίζει τις λειτουργίες της παρουσίασης. Διαχωρίζοντας το σύστημα παρουσίασης από τα δεδομένα μπορεί να αλλάξει η μορφή της οθόνης που λαμβάνει ο χρήστης χωρίς να είναι απαραίτητη η αλλαγή του αντίστοιχου υπολογιστικού συστήματος.

Η εύρεση της καλύτερης παρουσίασης των πληροφοριών χρειάζεται γνώση του υπόβαθρου των χρηστών που θα προσπελάσουν τις πληροφορίες και του τρόπου με τον οποίο αυτοί χρησιμοποιούν το σύστημα. Στην απόφαση του πως θα παρουσιαστούν οι πληροφορίες ο σχεδιαστής πρέπει να λάβει ένα πλήθος παραμέτρων υπ' όψιν του:

1. Ο χρήστης ενδιαφέρεται για ακριβείς πληροφορίες ή για τις σχέσεις μεταξύ διαφορετικών τιμών δεδομένων;
2. Πόσο αλλάζουν οι τιμές των πληροφοριών; Θα πρέπει η αλλαγή κάποιας τιμής να γίνεται άμεσα γνωστή στο χρήστη;
3. Ο χρήστης θα πρέπει να κάνει κάποια ενέργεια ως αποτέλεσμα της αλλαγής της πληροφορίας;
4. Χρειάζεται να αλληλεπιδρά ο χρήστης με τις πληροφορίες που παρουσιάζονται μέσω ενός άμεσου περιβάλλοντος χρήσης;
5. Οι πληροφορίες που παρουσιάζονται είναι αριθμητικές ή σε μορφή κειμένου; Είναι σημαντικές οι τιμές σχετικών αντικειμένων πληροφορίας;

Πληροφορίες που δεν αλλάζουν μπορούν να παρασταθούν είτε γραφικά είτε σε μορφή κειμένου, ανάλογα με την εφαρμογή. Η παρουσίαση σε μορφή κειμένου χρειάζεται λιγότερο χώρο στην οθόνη αλλά δεν μπορεί να διαβαστεί με μία ματιά. Οι πληροφορίες που δεν



αλλάζουν θα πρέπει να ξεχωρίζουν από τις δυναμικές πληροφορίες χρησιμοποιώντας διαφορετικό στυλ παρουσίασης. Για παράδειγμα, όλες οι στατικές πληροφορίες θα μπορούσαν να παρουσιάζονται σε ένα συγκεκριμένο φόντο ή να υπογραμμίζονται με ένα συγκεκριμένο χρώμα ή να συνοδεύονται πάντα από ένα σχετικό αντικείμενο.

Οι πληροφορίες θα πρέπει να παρουσιάζονται σαν κείμενο όταν απαιτούνται ακριβείς αριθμητικές πληροφορίες και οι πληροφορίες αλλάζουν σχετικά αργά. Εάν τα δεδομένα αλλάζουν γρήγορα ή αν οι σχέσεις μεταξύ των δεδομένων είναι σημαντικές, τότε κατά προτίμηση θα πρέπει να χρησιμοποιείται γραφική παρουσίαση.

Η χρήση γραφικής υπογράμμισης είναι συχνή κυρίως για την προσέλκυση του ενδιαφέροντος του χρήστη σε αλλαγές τμημάτων της παρουσίασης. Αν όμως οι αλλαγές αυτές συμβαίνουν ξαφνικά, τότε αυτή η τακτική δεν θα πρέπει να χρησιμοποιείται, αφού οι αιφνίδιες αλλαγές μπορεί να προκαλέσουν την αστραπιαία εμφάνιση της παρουσίασης. Αυτό αποσπά και ενοχλεί τους χρήστες.

Όταν πρέπει να παρουσιαστεί μεγάλος όγκος πληροφοριών, τότε μπορεί να χρησιμοποιηθούν εισαγωγικές οπτικοποιήσεις που συνδέουν σχετικά αντικείμενα δεδομένων. Αυτό μπορεί να εκθέσει σχέσεις που δεν είναι φανερές από τα προφανή δεδομένα. Οι σχεδιαστές των συστημάτων διεπαφής με τους χρήστες θα πρέπει να έχουν γνώση των πιθανοτήτων της οπτικοποίησης, ειδικά όταν το σύστημα του περιβάλλοντος διεπαφής πρέπει να αναπαριστά φυσικές οντότητες.

Στην παρουσίαση των διαθέσιμων πληροφοριών μεγάλο ρόλο παίζει και η χρήση του χρώματος με πολλούς διαφορετικούς τρόπους. Το χρώμα μπορεί να βελτιώσει τα περιβάλλοντα διεπαφής βοηθώντας τους χρήστες να αντιληφθούν και αντιμετωπίσουν την έμφυτη πολυπλοκότητα των συστημάτων. Ας δούμε κάποιες βασικές αρχές που θα πρέπει να διέπουν τα περιβάλλοντα διεπαφής σε σχέση με το χρώμα για να κατανοήσουμε καλύτερα τα πιθανά λάθη που μπορεί να γίνουν κατά το σχεδιασμό του περιβάλλοντος και το πώς μπορούν να αποφευχθούν.

1. Πρέπει να είμαστε φειδωλοί στον αριθμό των χρωμάτων που χρησιμοποιούμε και στον τρόπο με τον οποίο τα χρησιμοποιούμε. Τα χρώματα πρέπει να χρησιμοποιούμε επιλεκτικά και με συνέπεια, όχι για να «φωτίσουμε» απλώς το περιβάλλον.
2. Η αλλαγή στα χρώματα θα πρέπει να συνοδεύεται και από κάποια αλλαγή στην κατάσταση του συστήματος. Αν μία παρουσίαση αλλάξει χρώμα, αυτό θα σημαίνει ότι συνέβη ένα σημαντικό γεγονός. Η χρωματιστή υπογράμμιση είναι ιδιαίτερα σημαντική σε πολύπλοκες παρουσιάσεις όπου μπορεί να παρατίθενται εκατοντάδες διαφορετικές οντότητες.
3. Οι χρήστες θα πρέπει να υποστηρίζονται στην εργασία που προσπαθούν να ολοκληρώσουν από μια κωδικοποίηση μέσω των χρωμάτων. Αν, για παράδειγμα, πρέπει να αναγνωρίσουν κάποιες ανώμαλες επιφάνειες, καλό θα ήταν να γίνει υπογράμμιση με κάποιο χρώμα των επιφανειών αυτών. Αν πάλι πρέπει να βρεθούν ομοιότητες, επιβάλλεται η χρήση διαφορετικού χρώματος στην υπογράμμιση.
4. Η κωδικοποίηση με βάση τα χρώματα θα πρέπει να γίνεται με συντηρητικό και συνεπή τρόπο. Αν ένα μέρος του συστήματος παρουσιάζει τα μηνύματα λάθους με κάποιο συγκεκριμένο χρώμα, όλα τα υπόλοιπα μέρη θα πρέπει να κάνουν το ίδιο. Το χρώμα αυτό δεν θα πρέπει να χρησιμοποιείται για κανένα άλλο σκοπό. Αν χρησιμοποιηθεί για άλλο σκοπό ο χρήστης μπορεί να ερμηνεύσει το χρώμα αυτό ως μήνυμα λάθους. Πρέπει πάντα να έχουμε στο μυαλό μας ότι κάποιες κατηγορίες χρηστών μπορεί να κάνουν υποθέσεις για τη σημασία ορισμένων χρωμάτων.
5. Τέλος, θα πρέπει να είμαστε προσεκτικοί με τους συνδυασμούς των χρωμάτων. Κάποιοι συνδυασμοί, όπως για παράδειγμα το κόκκινο και το μπλε, είναι ενοχλητικοί για τα μάτια ή δύσκολοι στην ανάγνωση.

### 5.3. ΠΑΡΟΧΗ ΥΠΟΣΤΗΡΙΞΗΣ ΣΤΟΥΣ ΧΡΗΣΤΕΣ

Εκτός από την παρουσίαση των διαθέσιμων πληροφοριών στους χρήστες ανάλογα με την ιδιότητα και τις δικαιοδοσίες τους, όπως έχουμε ήδη αναφέρει στην αρχή του κεφαλαίου, πολύ σημαντικό ρόλο στον σχεδιασμό ενός λειτουργικού περιβάλλοντος διεπαφής είναι η παροχή υποστήριξης στους χρήστες του συστήματος. Η υποστήριξη αυτή μπορεί να είναι είτε υπό τη μορφή μηνυμάτων παραγομένων από το σύστημα μετά από δεδομένες ενέργειες του χρήστη, είτε υπό τη μορφή on-line βοήθειας μέσα από το ίδιο το σύστημα, είτε μέσω του εγχειριδίου χρήσης που παρέχεται μαζί με το σύστημα.

Ο σχεδιασμός χρήσιμων και ενημερωτικών πληροφοριών για τους χρήστες θα πρέπει να αντιμετωπίζεται σοβαρά από τους σχεδιαστές του συστήματος και να υποβάλλεται σε κάποιο έλεγχο ποιότητας, όπως και τα σχέδια και τα προγράμματα. Οι διαχειριστές έργου θα πρέπει να αφιερώνουν επαρκή χρόνο και προσπάθεια για το σχεδιασμό μηνυμάτων, κάτι που μπορεί να απαιτεί τη συμμετοχή επαγγελματιών συγγραφέων και γραφιστών στην όλη διαδικασία. Παρακάτω αναλύεται κάθε κατηγορία μορφής υποστήριξης του συστήματος προς το χρήστη ξεχωριστά, έτσι ώστε να γίνει αναλυτική προσέγγιση στο θέμα και να περιγραφούν λεπτομερειακά οι μέθοδοι που έχουν χρησιμοποιηθεί στην εφαρμογή της μηχανογράφησης της εφαρμογής.

#### 5.3.1. Μηνύματα λάθους

Η πρώτη εντύπωση που θα έχουν οι χρήστες από ένα σύστημα λογισμικού είναι τα μηνύματα λάθους του συστήματος. Οι άπειροι χρήστες μπορεί να ξεκινήσουν να δουλεύουν με το σύστημα, να κάνουν ένα αρχικό λάθος αμέσως θα πρέπει να καταλάβουν το συνεπαγόμενο μήνυμα λάθους. Αυτό μπορεί να είναι αρκετά δύσκολο για έμπειρους μηχανικούς λογισμικού και συχνά αδύνατο για άπειρους χρήστες.

Το σύστημα καθοδήγησης του χρήστη θα πρέπει να γνωρίζει τι κάνει ο χρήστης και να προσαρμόζει το μήνυμα στη συγκεκριμένη περίπτωση. Επίσης, κατά το σχεδιασμό των μηνυμάτων λάθους θα πρέπει να υπολογίζεται το υπόβαθρο και η εμπειρία των χρηστών. Για παράδειγμα, άλλη εξοικείωση με το σύστημα έχει ο administrator της εκπαιδευτικής εφαρμογής, ο οποίος έχει ουσιαστικά στήσει όλη την εφαρμογή τροφοδοτώντας την με πληροφορίες και διαφορετικά αντιμετωπίζει την εφαρμογή ένας μαθητής που έρχεται σε επαφή με το περιβάλλον του συστήματος για πρώτη φορά. Για το λόγο αυτό το σύστημα θα πρέπει να έχει τη δυνατότητα να παρέχει δύο τύπους μηνυμάτων και να επιτρέπει στο χρήστη να καθορίζει το βαθμό ακρίβειας των μηνυμάτων. Τα μηνύματα θα πρέπει επίσης να προσαρμόζονται στις ικανότητες του χρήστη και την ορολογία στην οποία είναι συνηθισμένος, εκτός από την εμπειρία του. Τέλος, τα μηνύματα πρέπει να έχουν θετικό στυλ γραφής, χωρίς να προσβάλλουν το χρήστη ή να είναι αστεία, και ο σχεδιαστής τους θα πρέπει να έχει κάποιο βαθμό εξοικείωσης με τον πολιτισμό της χώρας όπου πρόκειται να πωληθεί το λογισμικό.

#### 5.3.2. Σχεδιασμός του συστήματος βοήθειας

Όταν οι χρήστες έρχονται αντιμέτωποι με ένα μήνυμα λάθους που δεν καταλαβαίνουν, στρέφονται στο σύστημα βοήθειας για περισσότερες πληροφορίες. Τα εκτυπωμένα εγχειρίδια χρήσης είναι απαραίτητα συνοδευτικά για κάθε λογισμικό και είναι καλό να υπάρχουν και μέσα στο σύστημα της βοήθειας του λογισμικού. Εκτός όμως των εγχειριδίων που παρέχονται για τη διευκόλυνση του χρήστη με αυτή τη μορφή, απαραίτητες είναι και κάποιες άλλες μορφές παροχής βοήθειας, όπως ένα αλληλεπιδραστικό σύστημα βοήθειας (on-line help). Για το λόγο αυτό, μέσα από το περιβάλλον διεπαφής του συστήματος ο χρήστης έχει τη δυνατότητα να ανατρέξει για βοήθεια στο παρόν εγχειρίδιο χρήσης του συστήματος, στο εγχειρίδιο που απευθύνεται αποκλειστικά στους χρήστες και περιγράφει αναλυτικά τις δυνατότητες που τους παρέχει το σύστημα και τέλος σε ένα σύστημα on-line βοήθειας.

Τα αλληλεπιδραστικά συστήματα βοήθειας θα πρέπει να παρέχουν ένα πλήθος διαφορετικών σημείων εισαγωγής του χρήστη στη βοήθεια. Θα πρέπει να επιτρέπουν στο χρήστη να μπει στο σύστημα βοήθειας στην κορυφή της ιεραρχίας των μηνυμάτων και να αναζητάει πληροφορίες. Εναλλακτικά, μπορεί να μπει στο σύστημα βοήθειας για να του

εξηγηθεί ένα μήνυμα λάθους ή μπορεί να ζητήσει να μάθει τη σημασία μιας συγκεκριμένης εντολής. Για παράδειγμα, ο χρήστης μπορεί να έχει στη διάθεσή του ηλεκτρονική αναζήτηση λέξεων ή προτάσεων, πρόσβαση υπερκειμένου, αυτόματο ιστορικό αναζητήσεων, πολλαπλούς πίνακες περιεχομένων, κλπ.

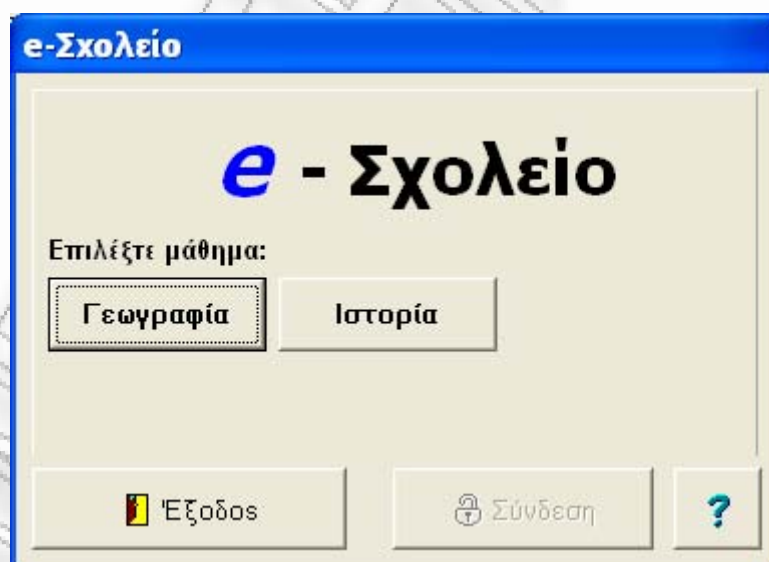
Όλα τα νοήμονα συστήματα βοήθειας έχουν μία πολύπλοκη δομή δικτύου όπου κάθε πλαίσιο πληροφορίας βοήθειας μπορεί να αναφέρεται σε πολλά άλλα πλαίσια πληροφορίας. Η δομή αυτού του δικτύου είναι συνήθως ιεραρχική με διασταυρούμενους συνδέσμους. Γενικές πληροφορίες βρίσκονται στην κορυφή της ιεραρχίας, ενώ πιο εξειδικευμένες πληροφορίες στο τέλος.

Προβλήματα μπορεί να προκύψουν όταν οι χρήστες μπαίνουν στο σύστημα πληροφοριών αφού έχουν κάνει κάποιο λάθος και μετά περιπλανιούνται μέσα στο δίκτυο. Μετά από ελάχιστο χρόνο μπορεί να χαθούν χωρίς να μπορούν να προσανατολιστούν. Στην περίπτωση αυτή θα πρέπει να εγκαταλείψουν την προσπάθεια και να ξεκινήσουν ξανά από κάποιο γνωστό σημείο του δικτύου.

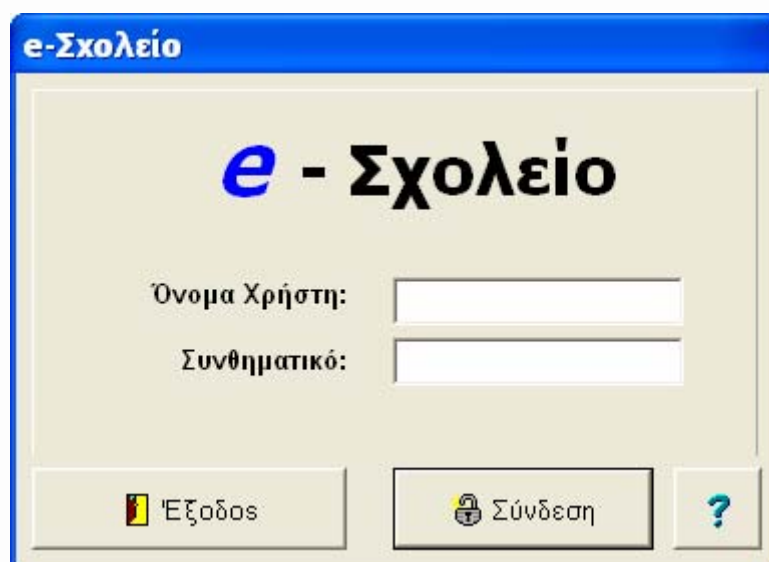
#### **5.4. ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΔΙΕΠΑΦΗΣ ΜΕ ΤΟ ΧΡΗΣΤΗ**

Για την υλοποίηση του συστήματος διεπαφής με το χρήστη στη συγκεκριμένη εφαρμογή χρησιμοποιήθηκε η γλώσσα προγραμματισμού BDS Delphi 2006 και με τη βοήθεια της οποίας να κατασκευάστηκαν όλες οι φόρμες.

Λαμβάνοντας υπόψη τους χρήστες του συστήματος και τις ανάγκες τους, θεωρήθηκε καλό η εφαρμογή να διαθέτει μία βασική φόρμα. Η βασική οθόνη σύνδεσης στο πρόγραμμα είναι η πρώτη που θα συναντάται κάθε φορά:



Ο χρήστης μπορεί να επιλέξει μεταξύ των δύο μαθημάτων που υπάρχουν αυτή τη στιγμή στην εφαρμογή. Στη συνέχεια υπάρχει η δυνατότητα προσθήκης νέων μαθημάτων και έτσι ο χρήστης θα μπορεί να επιλέξει ποιο από τα μαθήματα θα παρακολουθήσει και θα εξεταστεί. Όταν επιλεγεί το μάθημα, εμφανίζεται η παρακάτω οθόνη:





**e-Σχολείο**


**e - Σχολείο**

Όνομα Χρήστη:

Συνθηματικό:

 Έξοδος

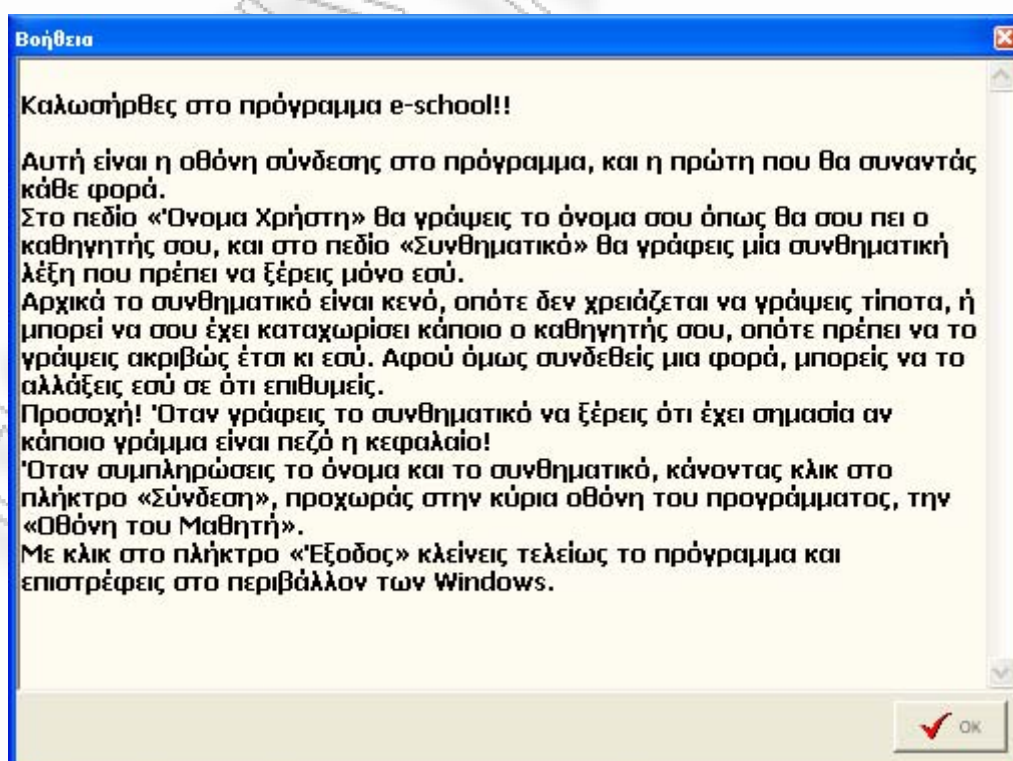
 Σύνδεση



Στο πεδίο «Όνομα Χρήστη» θα αναγράφεται το όνομα του εκάστοτε χρήστη και στο πεδίο «Συνθηματικό» θα αναγράφεται μία συνθηματική λέξη που πρέπει να γνωρίζει ο κάθε χρήστης μόνο. Στην περίπτωση των μαθητών, το «Όνομα Χρήστη» και το «Συνθηματικό» θα τα γνωρίζει και ο καθηγητής και ο administrator. Αρχικά το συνθηματικό είναι κενό, οπότε δεν χρειάζεται να αναγράφεται κάτι, ή εάν έχει καταχωρηθεί κάποιος, πρέπει να το γράφεται ακριβώς με τον ίδιο τρόπο. Έχει σημασία πως γράφεται το συνθηματικό, αν κάποιο γράμμα είναι πεζό η κεφαλαίο.

Όταν συμπληρωθεί το όνομα και το συνθηματικό, κάνοντας κλικ στο πλήκτρο «Σύνδεση», γίνεται μετάβαση στην κύρια οθόνη του προγράμματος ανάλογα με τον χρήστη, για τον μαθητή στην βασική «Οθόνη του Μαθητή», για τον καθηγητή στην βασική «Οθόνη του Καθηγητή» και για τον administrator στην βασική «Οθόνη του Administrator».

Με κλικ στο κουμπί «Βοήθεια», ο χρήστης μπορεί να δει οδηγίες για τη σύνδεσή του στην εφαρμογή :



**Βοήθεια**

**Καλωσήρθες στο πρόγραμμα e-school!!**

**Αυτή είναι η οθόνη σύνδεσης στο πρόγραμμα, και η πρώτη που θα συναντάς κάθε φορά.**


**Στο πεδίο «Όνομα Χρήστη» θα γράφεις το όνομα σου όπως θα σου πει ο καθηγητής σου, και στο πεδίο «Συνθηματικό» θα γράφεις μία συνθηματική λέξη που πρέπει να ξέρεις μόνο εσύ.**

**Αρχικά το συνθηματικό είναι κενό, οπότε δεν χρειάζεται να γράφεις τίποτα, ή μπορεί να σου έχει καταχωρίσει κάποιος ο καθηγητής σου, οπότε πρέπει να το γράφεις ακριβώς έτσι κι εσύ. Αφού όμως συνδεθείς μια φορά, μπορείς να το αλλάξεις εσύ σε ότι επιθυμείς.**

**Προσοχή! Όταν γράφεις το συνθηματικό να ξέρεις ότι έχει σημασία αν κάποιο γράμμα είναι πεζό η κεφαλαίο!**

**Όταν συμπληρώσεις το όνομα και το συνθηματικό, κάνοντας κλικ στο πλήκτρο «Σύνδεση», προχωράς στην κύρια οθόνη του προγράμματος, την «Οθόνη του Μαθητή».**

**Με κλικ στο πλήκτρο «Έξοδος» κλείνεις τελείως το πρόγραμμα και επιστρέφεις στο περιβάλλον των Windows.**



Με κλικ στο κουμπί «Έξοδος» κλείνει τελείως το πρόγραμμα και γίνεται επιστροφή στο περιβάλλον των Windows.

Θα παρουσιαστεί το μάθημα της Γεωγραφίας και στο Παράρτημα 1, όπου υπάρχει το user manual, θα γίνει παρουσίαση και των δύο μαθημάτων.

## ΒΑΣΙΚΕΣ ΟΘΟΝΕΣ ΜΑΘΗΤΗ

Ο μαθητής αφού συνδεθεί στην εφαρμογή, μεταβαίνει στην βασική «Οθόνη του Μαθητή», η οποία παρατίθεται παρακάτω:

Καλωσήρθες!

Το τελευταίο μάθημα που παρακολούθησες ήταν το: **Νομός Άρτας**

Επέλεξε μάθημα από τη λίστα ή κάνοντας κλικ στον χάρτη και ύστερα κάνε κλικ στο κουμπί "Έναρξη μαθήματος" για να ξεκινήσει το μάθημα

**Νομός Άρτας**

Έναρξη μαθήματος!

Τεστ προόδου

Καρτέλα Αξιολόγησης

Αλλαγή συνηματικού

Εξοδος

Τελευταία παρακολούθηση: 20/11/2012  
Τελευταίο τεστ προόδου: 20/11/2012 Βαθμολογία: 10,8

Στα αριστερά θα βρίσκονται μία σειρά από πλήκτρα για την πραγματοποίηση των σημαντικότερων λειτουργιών, όπως η παρακολούθηση ή η εξέταση κάποιου μαθήματος.

Στα δεξιά και στο μεγαλύτερο κομμάτι της οθόνης εμφανίζεται ο βασικός χάρτης του προγράμματος. Μετακινώντας τον δείκτη του ποντικιού πάνω στον χάρτη υπάρχει η δυνατότητα να δει ο μαθητής για ποιες περιοχές υπάρχουν διαθέσιμα μαθήματα. Το πλαίσιο που εμφανίζεται κοντά στο δείκτη του ποντικιού δείχνει τότε ο μαθητής παρακολούθησε τελευταία φορά το μάθημα, εάν βέβαια το παρακολούθησε, και αν έχει εξεταστεί στο μάθημα αυτό γράφει και τότε εξετάστηκε, καθώς και τι βαθμό πήρε στην εξέταση.

Κάνοντας κλικ με το ποντίκι σε κάποια περιοχή που διαθέτει μάθημα, υπάρχει η δυνατότητα επιλογής του μαθήματος αυτού, ώστε ο μαθητής να το παρακολουθήσει ή να εξεταστεί σε αυτό. Επίσης, υπάρχει η δυνατότητα επιλογής μαθήματος από το μενού που βρίσκεται ακριβώς πάνω από το πλήκτρο «Έναρξη μαθήματος».

Στο πλαίσιο ακριβώς κάτω από τον χάρτη εμφανίζεται η ονομασία του επιλεγμένου μαθήματος, η ημερομηνία παρακολούθησης της τελευταίας φοράς, η ημερομηνία εξέτασης, καθώς και ο βαθμός κατά την εξέταση.

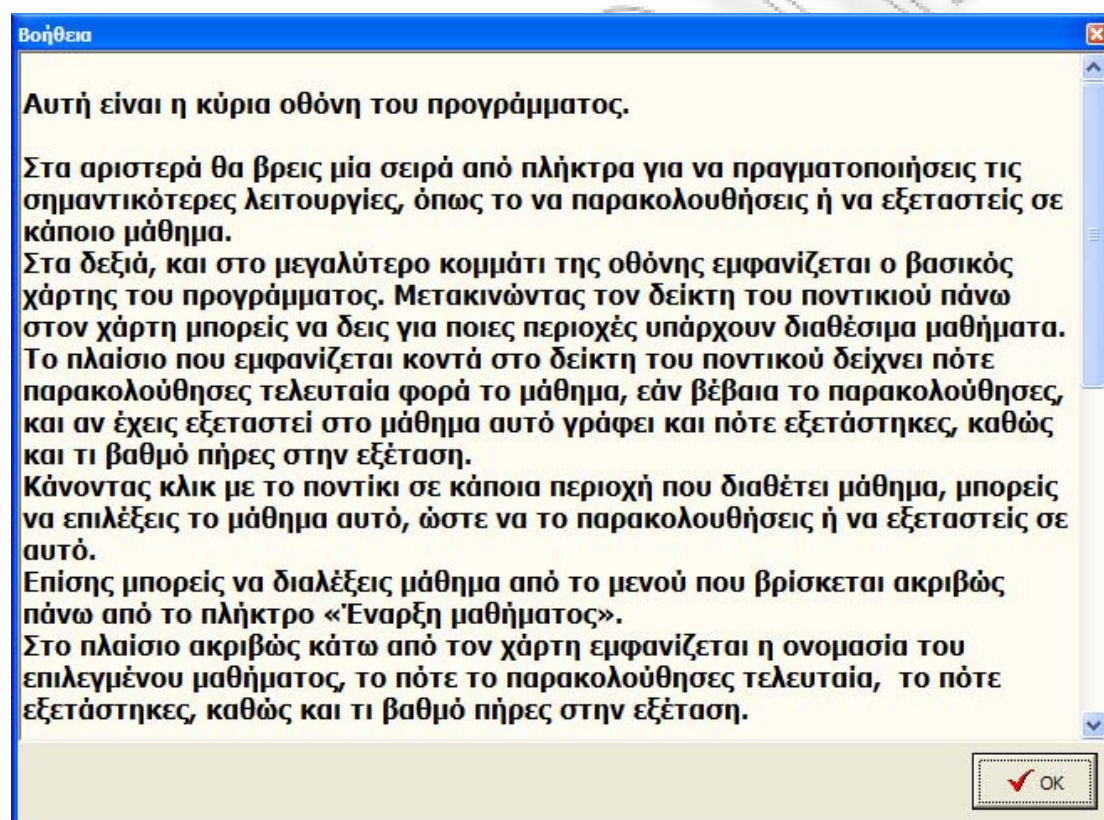
Αφού επιλεγεί το μάθημα από τον χάρτη ή από το μενού, ο μαθητής μπορεί να κάνει κλικ στο πλήκτρο «Έναρξη μαθήματος» για να παρακολουθήσει το συγκεκριμένο μάθημα. Τότε θα μεταφερθεί στην Οθόνη Παρουσίασης Μαθήματος.

Όταν ο μαθητής παρακολουθήσει το μάθημα, μπορείς να κάνει κλικ στο πλήκτρο «Τεστ προόδου» για να εξεταστεί σε αυτό το μάθημα. Σε αυτήν την περίπτωση θα μεταφερθεί στην Οθόνη Εξέτασης.

Όταν ο μαθητής έχει παρακολουθήσει αρκετά μαθήματα, και έχει κάνει αρκετά τεστ, μπορεί να δει την αξιολόγησή του στα τεστ κάνοντας κλικ στο πλήκτρο «Καρτέλα Αξιολόγησης», οπότε θα μεταφερθεί στην αντίστοιχη οθόνη.

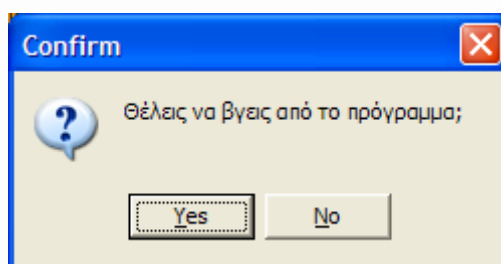
Από αυτή την οθόνη, επίσης, δίδεται στον μαθητή η δυνατότητα να αλλάξει το συνθηματικό του σε οποιαδήποτε λέξη ή συνδυασμό επιθυμεί. Με κλικ στο πλήκτρο «Αλλαγή Συνθηματικού», ανοίγει μία οθόνη στην οποία ο μαθητής μπορεί να πραγματοποιήσει την αλλαγή.

Με κλικ στο κουμπί «Βοήθεια», ο χρήστης μπορεί να δει οδηγίες για την λειτουργικότητα της κύριας οθόνης του μαθητή :



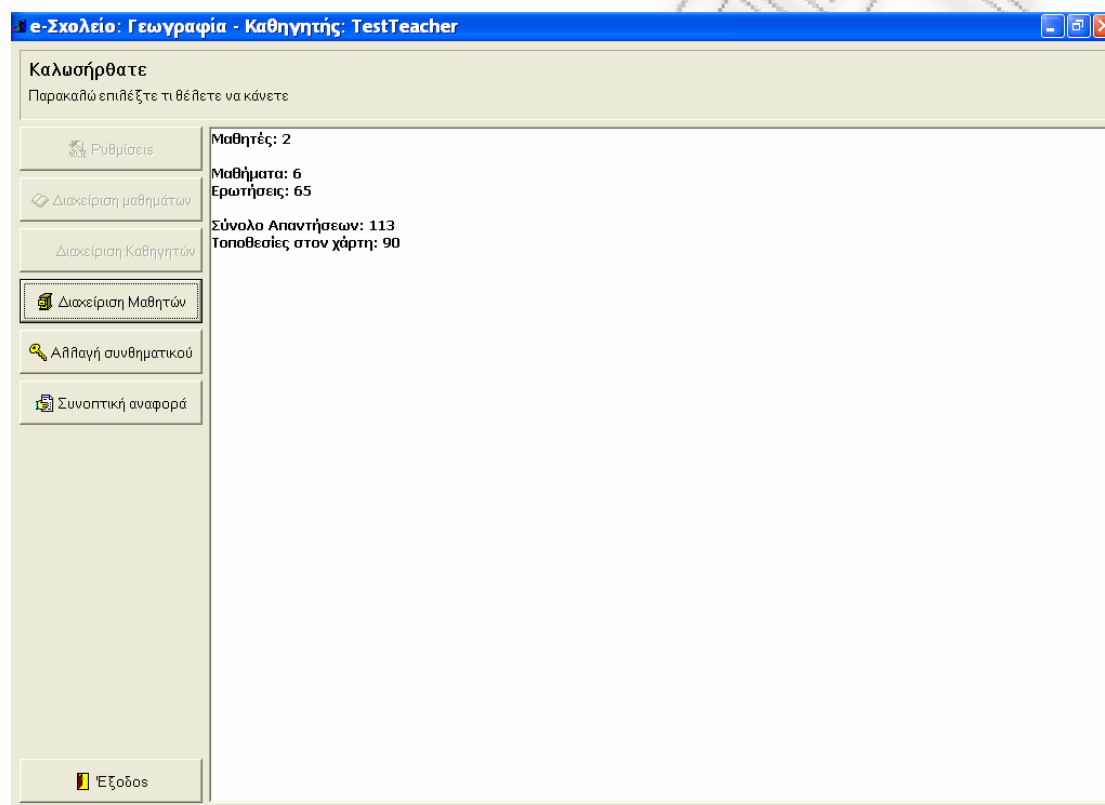
Βοήθεια υπάρχει επίσης και από τον agent που υπάρχει στη εφαρμογή. Βρίσκεται στα αριστερά της οθόνης και δίνει συμβουλές στον μαθητή. Αν κάνει κλικ πάνω του υπάρχει η βασική οθόνη της βοήθειας, όπως παρουσιάστηκε και παραπάνω.

Τέλος, κάνοντας κλικ στο πλήκτρο «Έξοδος» ο μαθητής επιστρέφει στην Οθόνη Σύνδεσης, αφού πρώτα ερωτηθεί από το πρόγραμμα αν επιθυμεί να βγει :



## ΒΑΣΙΚΕΣ ΟΘΟΝΕΣ ΚΑΘΗΓΗΤΗ

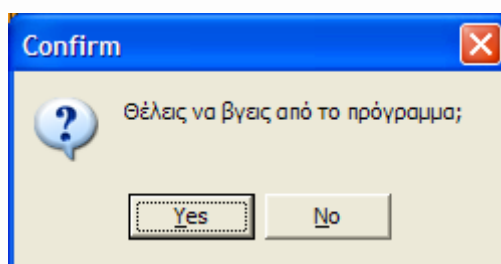
Ο καθηγητής αφού συνδεθεί στην εφαρμογή, μεταβαίνει στην βασική «Οθόνη του Καθηγητή», η οποία παρατίθεται παρακάτω :



Στα αριστερά θα βρίσκονται μία σειρά από πλήκτρα για την πραγματοποίηση των σημαντικότερων λειτουργιών, όπως η διαχείριση των μαθητών, η αλλαγή του συνθηματικού και η συνοπτική αναφορά για την πρόοδο των μαθητών.

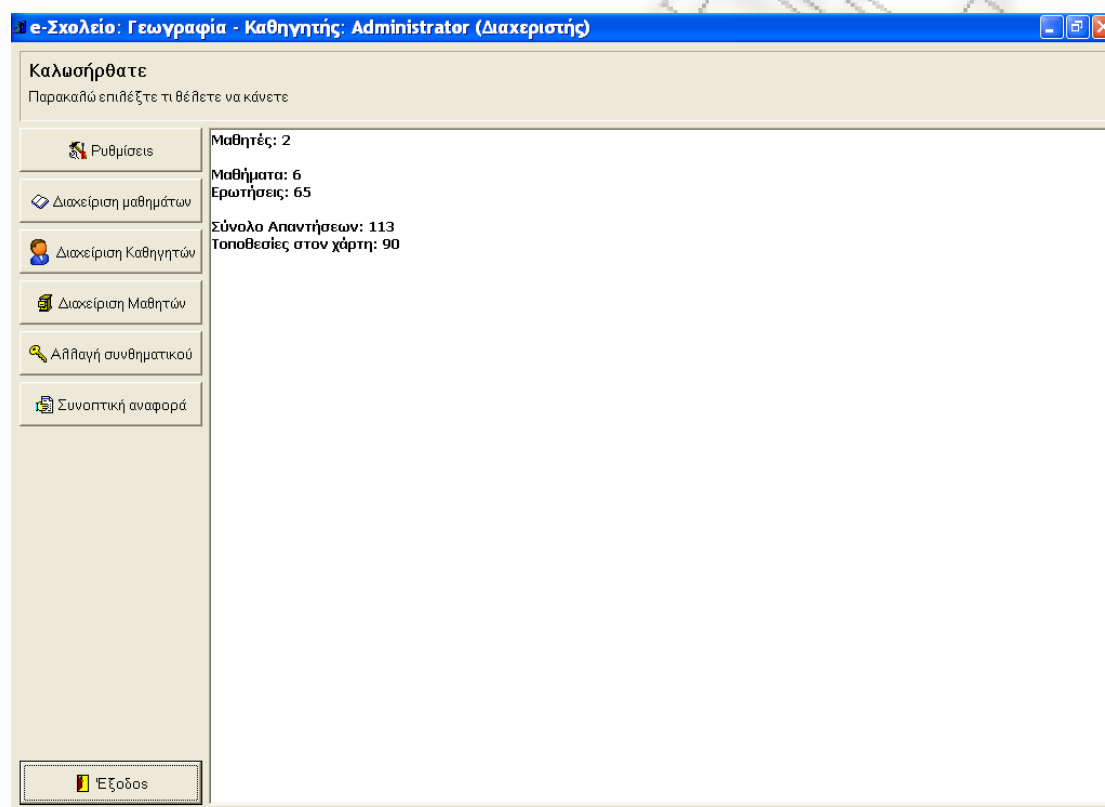
Στα δεξιά και στο μεγαλύτερο κομμάτι της οθόνης εμφανίζεται ένα grid με στοιχεία για τους μαθητές και τα μαθήματα. Ο καθηγητής μπορεί να δει τον αριθμό των μαθητών που παρακολουθούν το μάθημα, τον αριθμό των διαθέσιμων μαθημάτων, τον αριθμό των ερωτήσεων που συμμετέχουν στα εξεταζόμενα τεστ, το συνολικό πλήθος των απαντήσεων και τον αριθμό των τοποθεσιών που βρίσκονται στον χάρτη.

Τέλος, κάνοντας κλικ στο πλήκτρο «Έξοδος» ο καθηγητής επιστρέφει στην Οθόνη Σύνδεσης, αφού πρώτα ερωτηθεί από το πρόγραμμα αν επιθυμεί να βγει :



## ΒΑΣΙΚΕΣ ΟΘΟΝΕΣ ΔΙΑΧΕΙΡΙΣΤΗ

Ο administrator αφού συνδεθεί στην εφαρμογή, μεταβαίνει στην βασική «Οθόνη του Administrator», η οποία παρατίθεται παρακάτω :

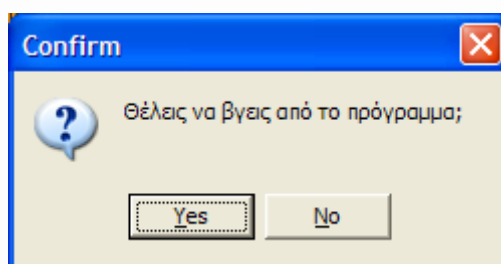


Στα αριστερά θα βρίσκονται μία σειρά από πλήκτρα για την πραγματοποίηση των σημαντικότερων λειτουργιών, όπως οι ρυθμίσεις των τεστ, η διαχείριση των μαθητών, των μαθημάτων και των καθηγητών, η αλλαγή του συνθηματικού και η συνοπτική αναφορά για την πρόοδο των μαθητών.

Στα δεξιά και στο μεγαλύτερο κομμάτι της οθόνης εμφανίζεται ένα grid με στοιχεία για τους μαθητές και τα μαθήματα. Ο administrator μπορεί να δει τον αριθμό των μαθητών που παρακολουθούν το μάθημα, τον αριθμό των διαθέσιμων μαθημάτων, τον αριθμό των ερωτήσεων που συμμετέχουν στα εξεταζόμενα τεστ, το συνολικό πλήθος των απαντήσεων και τον αριθμό των τοποθεσιών που βρίσκονται στον χάρτη.

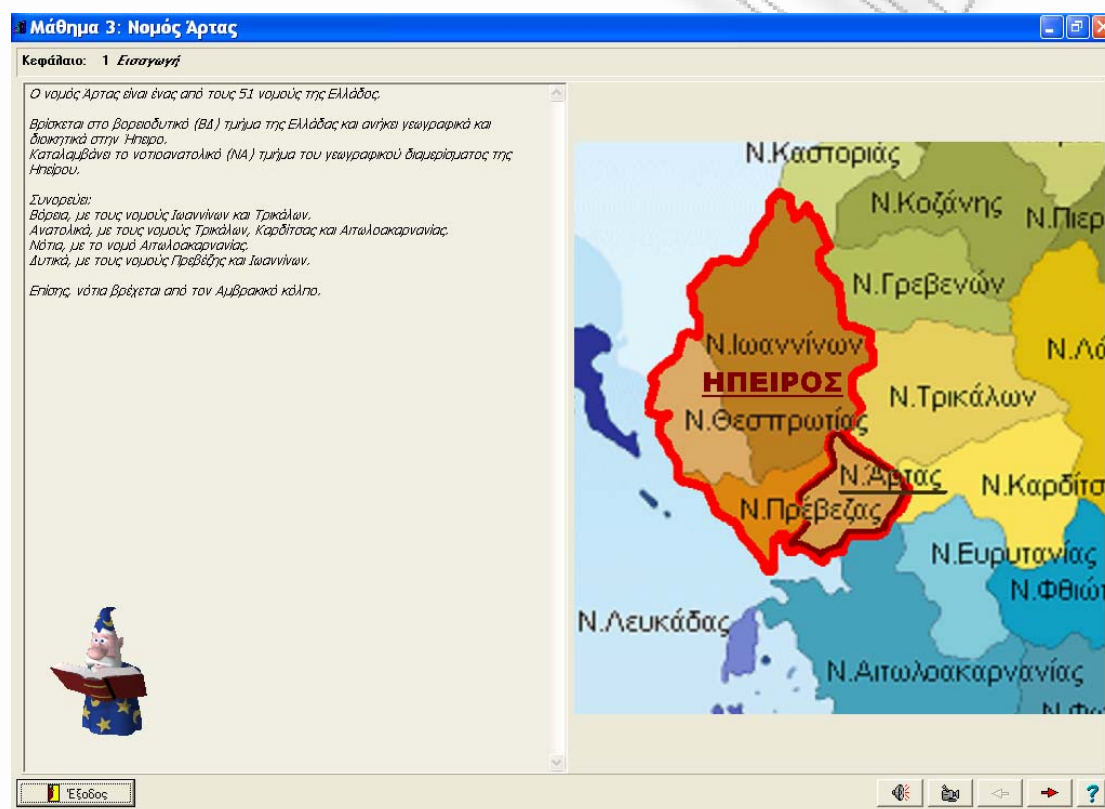
Τέλος, κάνοντας κλικ στο πλήκτρο «Έξοδος» ο administrator επιστρέφει στην Οθόνη Σύνδεσης, αφού πρώτα ερωτηθεί από το πρόγραμμα αν επιθυμεί να βγει :





## ΟΘΟΝΕΣ ΜΑΘΗΤΗ

Η παρακάτω οθόνη αποτελεί μία από τις πολλές οθόνες που έχουν σχεδιαστεί για την παρακολούθηση των μαθημάτων από τους μαθητές :



Η οθόνη αποτελείται βασικά από δύο μέρη. Στα αριστερά γράφει το κείμενο του κεφαλαίου, την ύλη του μαθήματος που έχει καθοριστεί από τον administrator και πρέπει να μάθει ο μαθητής. Στα δεξιά υπάρχει μια εικόνα σχετική με το κεφάλαιο.

Όταν ο μαθητής διαβάσει το κάθε κεφάλαιο μπορεί να προχωρήσει στο επόμενο με κλικ στο πλήκτρο κάτω-δεξιά, που απεικονίζει ένα βέλος προς τα δεξιά. Δίπλα από αυτό υπάρχει ένα πλήκτρο με το αντίθετο βέλος, σε περίπτωση που επιθυμεί να επιστρέψει και να ξαναδιαβάσει ένα κεφάλαιο που πέρασε πιο πριν. Αυτά τα δύο πλήκτρα μπορούν να θεωρηθούν σαν το γύρισμα μιας σελίδας μπρος και πίσω σε ένα παραδοσιακό βιβλίο.

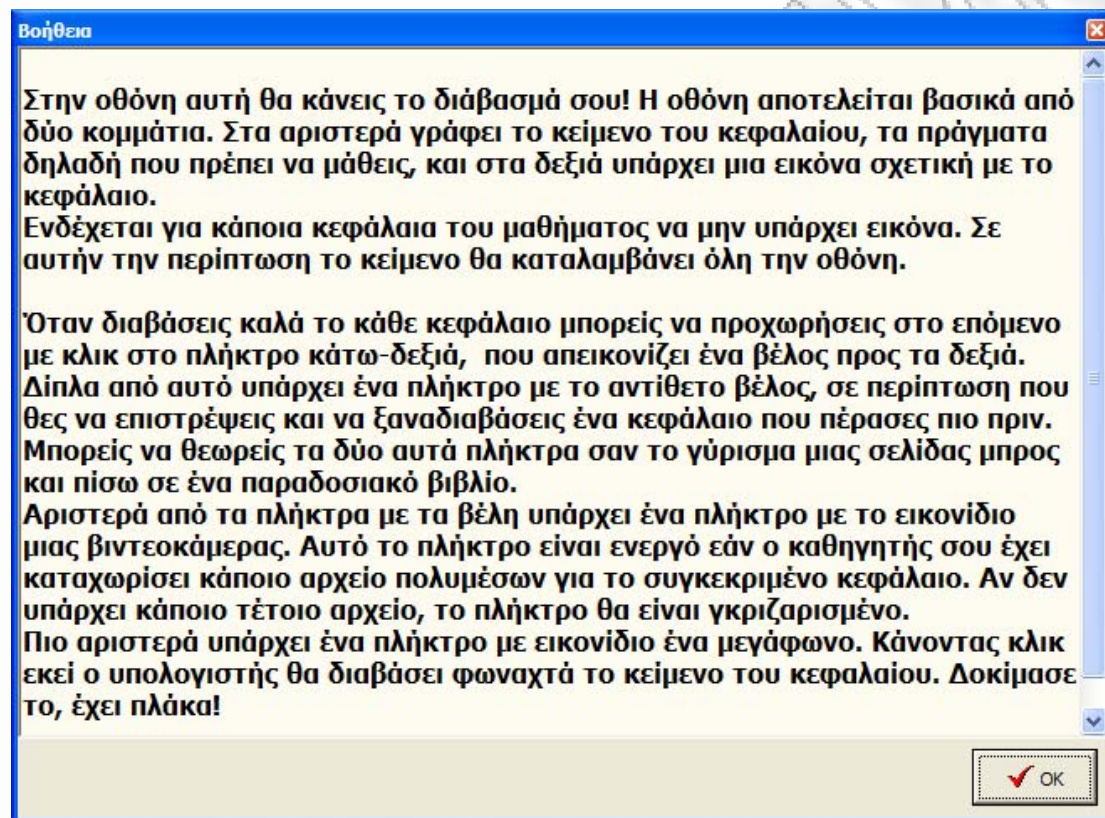
Αριστερά από τα πλήκτρα με τα βέλη υπάρχει ένα πλήκτρο με το εικονίδιο μιας βιντεοκάμερας. Αυτό το πλήκτρο είναι ενεργό, εάν ο καθηγητής έχει καταχωρήσει κάποιο

αρχείο πολυμέσων για το συγκεκριμένο κεφάλαιο. Αν δεν υπάρχει κάποιο τέτοιο αρχείο, το πλήκτρο θα είναι γκριζαρισμένο.

Πιο αριστερά υπάρχει ένα πλήκτρο με εικονίδιο ένα megáφωνο. Κάνοντας κλικ εκεί ο υπολογιστής θα διαβάσει φωναχτά το κείμενο του κεφαλαίου.

Επίσης, όταν το κείμενο ενός κεφαλαίου είναι αρκετά μεγάλο, ο μαθητής μπορεί να χρησιμοποιήσει το schroll down.

Με κλικ στο κουμπί «Βοήθεια», ο χρήστης μπορεί να δει οδηγίες για την παρακολούθηση του μαθήματος :



Βοήθεια υπάρχει επίσης και από τον agent που υπάρχει στη εφαρμογή. Βρίσκεται στα αριστερά της οθόνης και δίνει συμβουλές στον μαθητή. Αν κάνει κλικ πάνω του υπάρχει η βασική οθόνη της βοήθειας, όπως παρουσιάστηκε και παραπάνω.


Τέλος, κάνοντας κλικ στο πλήκτρο «Έξοδος» ο μαθητής επιστρέφει στην βασική «Οθόνη του Μαθητή».


Οι παρακάτω οθόνες αποτελούν δύο από τις πολλές οθόνες που έχουν σχεδιαστεί για την εξέταση των μαθημάτων από τους μαθητές. Το κάθε τεστ αποτελείται από κάποιες ερωτήσεις που μπορεί να είναι δύο ειδών:

Ερωτήσεις στον χάρτη :



**Ερωτήσεις πάνω στο μάθημα 3: Νομός Άρτας**

Ερώτηση: 1

Ολοκλήρωση τεστ:  Μένουν 12 ερωτήσεις

Χρόνος που απομένει:  Μένουν 4 λεπτά

Δείξε τον Νομό Άρτας στον χάρτη της Ελλάδας

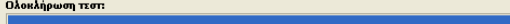
Έξοδος

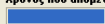
Σάββατο, 24 Νοεμβρίου 2012

και ερωτήσεις πολλαπλής επιλογής :

**Ερωτήσεις πάνω στο μάθημα 3: Νομός Άρτας**

Ερώτηση: 8

Ολοκλήρωση τεστ:  Μένουν 5 ερωτήσεις

Χρόνος που απομένει:  Μένουν 4 λεπτά

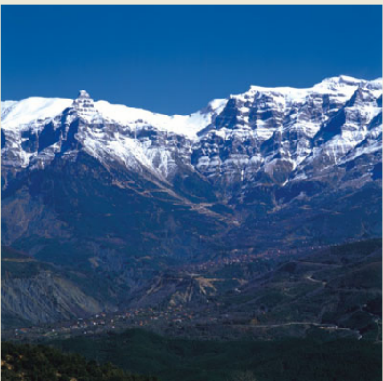

Ποιο είναι το ψηλότερο βουνό του νομού Άρτας;

**A** Ίδη (Ψηλορείτης)

**B** Τιμφριστός (Βελούχι)

**Γ** Αθαμανικά Όρη (Τζουμέρκα)

**Δ** Οίτη

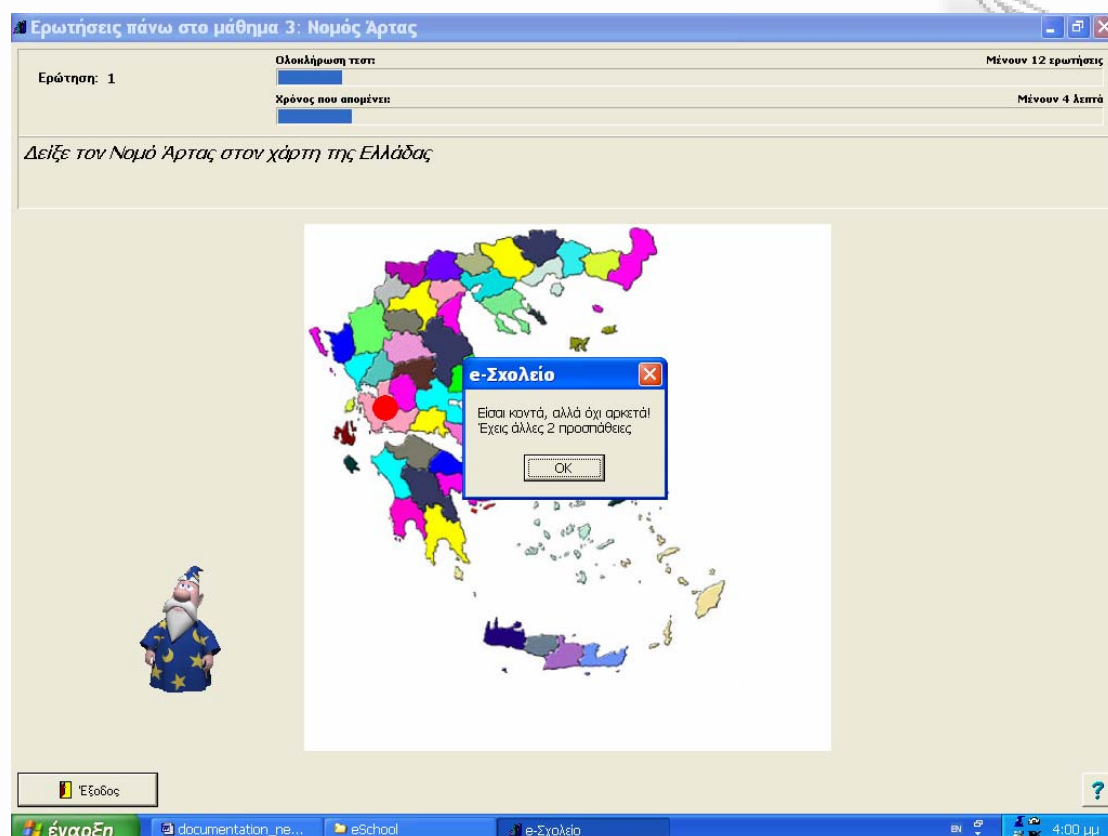



Έξοδος

?

Στις ερωτήσεις στον χάρτη, ζητείται από τον μαθητή να κάνει κλικ σε μια συγκεκριμένη περιοχή του χάρτη. Αν επιλέξει το σωστό σημείο, το τεστ προχωράει στην

επόμενη ερώτηση. Εάν κάνει κλικ σε λάθος σημείο, εμφανίζεται ένα μήνυμα που λέει αν είναι κοντά ή μακριά, και αν το έχει επιτρέψει ο καθηγητής, ο μαθητής μπορεί να ξαναδοκιμάσει.



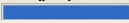
The screenshot shows a web browser window titled "Ερωτήσεις πάνω στο μάθημα 3: Νομός Άρτας". The interface includes a progress bar for "Ολοκλήρωση τεστ" (Test completion) and "Μένουν 12 ερωτήσεις" (12 questions remaining). A timer shows "Χρόνος που απομένει" (Time remaining) and "Μένουν 4 λεπτά" (4 minutes remaining). The question text is "Δείξε τον Νομό Άρτας στον χάρτη της Ελλάδας" (Show the Arta region on the map of Greece). A map of Greece is displayed with a red dot indicating the user's selection. A dialog box titled "e-Σχολείο" (e-School) is overlaid on the map, containing the message: "Είσαι κοντά, αλλά όχι αρκετά! Έχεις άλλες 2 προσπάθειες" (You are close, but not quite there! You have 2 more attempts) and an "OK" button. The Windows taskbar at the bottom shows the "e-School" application and the system clock at 4:00 μμ.

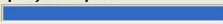
Αν κάνει ξανά λάθος η απάντηση θα θεωρηθεί τελικά λανθασμένη και το τεστ θα προχωρήσει στην επόμενη ερώτησή ή θα δοθεί νέα ευκαιρία από τον καθηγητή.

Στις ερωτήσεις πολλαπλής επιλογής, θα εμφανιστούν από δύο έως τέσσερα πλήκτρα απάντησης και ο μαθητής πρέπει να κάνει κλικ σε αυτό που πιστεύει ότι είναι η σωστή απάντηση. Αν επιλέξει την σωστή απάντηση, το τεστ θα προχωρήσει. Αν κάνει λάθος, το πλήκτρο που επέλεξε θα πάρει γκριζό χρώμα και θα μπορεί να ξαναδοκιμάσει να δώσει κάποια από τις άλλες διαθέσιμες απαντήσεις. Αν η ερώτηση αφορά τοποθεσία, το πρόγραμμα θα κάνει ένα σχόλιο στην περίπτωση λάθους, ανάλογα με το πόσο κοντά ή μακριά είναι η τοποθεσία που επέλεξε ο μαθητής από αυτήν που ζητάει η ερώτηση. Στην παρακάτω περίπτωση που ο μαθητής επέλεξε λάθος απάντηση, εμφανίστηκε το μήνυμα «Λάθος! Πολύ μακριά» :

Ερωτήσεις πάνω στο μάθημα 3: Νομός Άρτας

Ερώτηση: 2

Ολοκλήρωση τεστ:  Μένουν 11 ερωτήσεις

Χρόνος που απομένει:  Μένουν 3 λεπτά

Σε ποιο γεωγραφικό διαμέρισμα ανήκει ο νομός Άρτας;

A Πελοπόννησος

B Ήπειρος

Γ Νησιά Ιονίου Πελάγους

Δ Κρήτη

e-Σχολείο

Λάθος! Πολύ μακριά!

OK

Έξοδος

έναρξη

documentation\_ne...

eSchool

e-Σχολείο

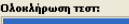
EN

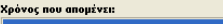
4:01 μμ

και γίνεται γκρι η λανθασμένη απάντηση που επιλέχθηκε :

Ερωτήσεις πάνω στο μάθημα 3: Νομός Άρτας

Ερώτηση: 2

Ολοκλήρωση τεστ:  Μένουν 11 ερωτήσεις

Χρόνος που απομένει:  Μένουν 3 λεπτά

Σε ποιο γεωγραφικό διαμέρισμα ανήκει ο νομός Άρτας;

A Πελοπόννησος

B Ήπειρος

Γ Νησιά Ιονίου Πελάγους

Δ Κρήτη

Έξοδος

έναρξη

documentation\_ne...

eSchool

e-Σχολείο

EN

4:02 μμ

Επίσης, στο πάνω μέρος της οθόνης αναγράφεται ο αύξων αριθμός της κάθε ερώτησης, καθώς και πόσες ερωτήσεις απομένουν για την ολοκλήρωση του τεστ και πόσος

χρόνος απομένει. Εάν ο μαθητής δεν μπορέσει να ολοκληρώσει το τεστ στον προβλεπόμενο χρόνο, τότε εμφανίζεται το παρακάτω μήνυμα :

Η νέος μηχανισμός που υλοποιήθηκε αφορά τις ερωτήσεις πολλαπλής επιλογής. Εφαρμόστηκε ένας αλγόριθμος, ο οποίος συγκρίνει τις δύο πιθανές απαντήσεις ως προς την ομοιότητα των γραμμάτων που περιέχουν. Ο αλγόριθμος λειτουργεί ως εξής:

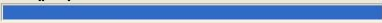
Κάθε γράμμα της μικρότερης εκ των δύο λέξεων συγκρίνεται με το αντίστοιχο γράμμα της άλλης λέξης. Αν το γράμμα είναι το ίδιο, προστίθενται δέκα μονάδες σε έναν αθροιστή. Αν δεν είναι το ίδιο, αλλά είναι το ίδιο με ένα από τα γειτονικά του, προστίθενται πέντε μονάδες στον αθροιστή. Το τελικό αποτέλεσμα της σύγκρισης είναι το εκατοστιαίο ποσοστό της τελικής τιμής του αθροιστή προς τον αριθμό των γραμμάτων της μεγαλύτερης λέξης επί δέκα. Οι ερωτήσεις δεν έχουν καμία διαφοροποίηση σε σχέση με τις ερωτήσεις πολλαπλής επιλογής. Υπάρχουν τέσσερα πλήκτρα απάντησης και ο μαθητής πρέπει να κάνει κλικ σε αυτό που πιστεύει ότι είναι η σωστή απάντηση. Αν επιλέξει την σωστή απάντηση, το τεστ θα προχωρήσει. Αν κάνει λάθος, το πλήκτρο που επέλεξε θα πάρει γκριζό χρώμα. Αν ο καθηγητής έχει κάνει την σωστή παραμετροποίηση ο μαθητής θα μπορεί να ξαναδοκιμάσει να δώσει κάποιες από τις άλλες διαθέσιμες απαντήσεις μία ή και παραπάνω φορές.

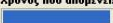
Ο νέος μηχανισμός εφαρμόζεται ως πρώτη συνθήκη και στη συνέχεια η συνθήκη της απόστασης που αναφέρεται παραπάνω.

Αν η ομοιότητα την λανθασμένης επιλογής είναι άνω του ποσοστού επί τοις εκατό που θα ορίσει ο καθηγητής, τότε το πρόγραμμα θα κάνει ένα σχόλιο, ανάλογα με το πόσο όμοιες ή όχι είναι η απάντηση που επέλεξε ο μαθητής από αυτήν που ζητάει η ερώτηση. Στην παρακάτω περίπτωση που ο μαθητής επέλεξε λάθος απάντηση, εμφανίστηκε το μήνυμα «Περίπου! Αλλά όχι ακριβώς!»:

Ερωτήσεις πάνω στο μάθημα 5: Νομός Πρεβέζης

Ερώτηση: 6

Ολοκλήρωση τεστ:  Μένουν 7 ερωτήσεις

Χρόνος που απομένει:  Μένουν 4 λεπτά


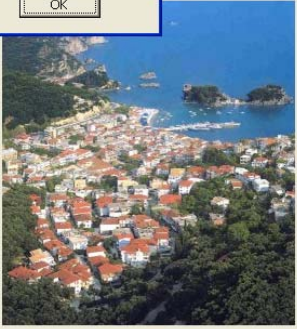
Πώς λέγεται η εικονιζόμενη παραθαλάσσια πόλη του νομού Πρεβέζης;

A Πάργα

B Πράγα

Γ Ελασσόνα

**e-Σχολείο**  
Περίπου! Αλλά όχι ακριβώς!  
OK

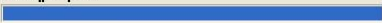
Έξοδος


έναρξη documentation\_ne... eSchool e-Σχολείο EN 4:04 μμ

Αν όμως επέλεγε μία απάντηση με κανένα βαθμό ομοιότητας ή κάτω από το ποσοστό που έχει οριστεί, τότε θα εφαρμοζόταν ο επόμενος μηχανισμός με την απόσταση. Στο παρακάτω παράδειγμα το μήνυμα λάθους είναι «Λάθος! Πολύ μακριά»:

Ερωτήσεις πάνω στο μάθημα 5: Νομός Πρεβέζης

Ερώτηση: 6

Ολοκλήρωση τεστ:  Μένουν 7 ερωτήσεις

Χρόνος που απομένει:  Μένουν 3 λεπτά


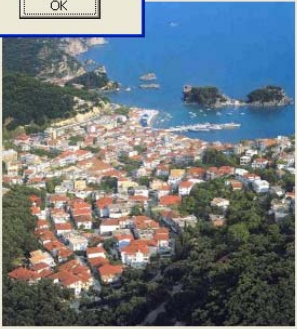
Πώς λέγεται η εικονιζόμενη παραθαλάσσια πόλη του νομού Πρεβέζης;

A Πάργα

B Πράγα

Γ Ελασσόνα

**e-Σχολείο**  
Λάθος! Πολύ μακριά!  
OK

Έξοδος

έναρξη documentation\_ne... eSchool e-Σχολείο EN 4:04 μμ

Με βάση τον παραπάνω αλγόριθμο ομοιότητας λέξεων προστέθηκαν δύο νέες κατηγορίες λαθών. Οι υπάρχουσες είναι «Ποτάμια», «Βουνά», «Λίμνες», «Γεωγραφικά Διαμερίσματα», «Δημογραφικά», «Πρωτεύουσες Νομών» και «Σύνορα Νομών». Οι νέες κατηγορίες είναι «Απροσεξία» και «Εκτός Ελλάδας».

Οι δύο κατηγορίες βασίζονται στον αλγόριθμο ομοιότητας των λέξεων. Στις νέες περιπτώσεις, στις δύο νέες κατηγορίες, υπάρχουν πιθανές απαντήσεις όμοιες λεξικά με την σωστή απάντηση.

Στην πρώτη περίπτωση, στην κατηγορία «Απροσεξία» οι πιθανές απαντήσεις στις ερωτήσεις πολλαπλής επιλογής είναι ορθογραφικά κοντά στην σωστή απάντηση. Παράδειγμα παρατίθεται παρακάτω:

Ερώτηση: 3 Ολοκλήρωση τεστ:   Μένουν 10 ερωτήσεις

  Χρόνος που απομένει: Μένουν 4 λεπτά

**Ποια είναι η πρωτεύουσα του νομού Άρτας;**

Α	Άτρα
Β	Άρτη
Γ	Άρτος
Δ	Άρτα

Εξοδος ?

Η ερώτηση είναι «Ποια είναι η πρωτεύουσα του νομού Άρτας?». Οι πιθανές απαντήσεις είναι «Άτρα», «Άρτη», «Άρτος» και η σωστή απάντηση «Άρτα». Όλες οι απαντήσεις είναι όμοιες με την σωστή απάντηση. Αν ο μαθητής επιλέξει μία εκ των λανθασμένων απαντήσεων, εμφανίζεται το μήνυμα λάθους «Περίπου! Αλλά όχι ακριβώς!»

Αν τα περισσότερα λάθη γίνουν στην κατηγορία «Απροσεξία», τότε στην καρτέλα αξιολόγησης εμφανίζεται ως η χειρότερη κατηγορία και ζητείται από τον μαθητή να κάνει περισσότερη εξάσκηση στην συγκεκριμένη κατηγορία.



**Καρτέλα Αξιολόγησης**

<b>Γενική Βαθμολογία:</b>	<b>17,0</b>
Μαθήματα που παρακολούθησες:	<b>4</b>
Ερωτήσεις που εξετάστηκες:	<b>56</b>
Σωστές απαντήσεις:	<b>47</b>

Επιδόσεις ανά κατηγορία ερωτήσεων:

Καλύτερη Κατηγορία:

Χειρότερη Κατηγορία:

Στην δεύτερη περίπτωση, στην κατηγορία «Εκτός Ελλάδας» οι πιθανές απαντήσεις στις ερωτήσεις πολλαπλής επιλογής είναι ορθογραφικά κοντά στην σωστή απάντηση, αλλά η λανθασμένες απαντήσεις είναι εκτός Ελλάδας. Παράδειγμα παρατίθεται παρακάτω:

Ερώτηση: 6 Ολοκλήρωση 100% Μένουν 7 ερωτήσεις

Χρόνος που απομένει: Μένουν 3 λεπτά


**Πώς λέγεται η εικονιζόμενη παραθαλάσσια πόλη του νομού Πρεβέζης;**

**A** Πράγα

**B** Πάργα

Geography

Είναι σε άλλη χώρα!



Η ερώτηση είναι «Πως λέγεται η εικονιζόμενη παραθαλάσσια πόλη του νομού Πρεβέζης?». Οι πιθανές απαντήσεις είναι «Πράγα» και η σωστή απάντηση «Πάργα». Όλες οι απαντήσεις είναι όμοιες με την σωστή απάντηση, αλλά είναι εκτός ελλαδικού χώρου. Αν ο μαθητής επιλέξει μία εκ των λανθασμένων απαντήσεων, εμφανίζεται το μήνυμα λάθους «Είναι εκτός Ελλάδας».

Αν τα περισσότερα λάθη γίνουν στην κατηγορία «Εκτός Ελλάδας», τότε στην καρτέλα αξιολόγησης εμφανίζεται ως η χειρότερη κατηγορία και ζητείται από τον μαθητή να κάνει περισσότερη εξάσκηση στην συγκεκριμένη κατηγορία.

Κατηγορία	Αποτέλεσμα
Γενική Βαθμολογία:	14,7
Μαθήματα που παρακολούθησες:	4
Ερωτήσεις που εξετάστηκες:	56
Σωστές απαντήσεις:	45
Επιδόσεις ανά κατηγορία ερωτήσεων:	
Καλύτερη Κατηγορία:	ΣΥΝΟΡΑ ΝΟΜΩΝ (100% Επιτυχία)
Χειρότερη Κατηγορία:	ΕΚΤΟΣ ΕΛΛΑΔΑΣ (0% Επιτυχία) <span style="float: right;">Εξάσκηση!</span>

Οι παράμετροι του χρόνου και του πλήθους των ερωτήσεων μπορούν να αλλάξουν και καθορίζονται από τον administrator.

Ακόμη, ολοκληρώνοντας το τεστ ο μαθητής λαμβάνει πληροφορίες σχετικά με την απόδοσή του κατά την εξέταση :

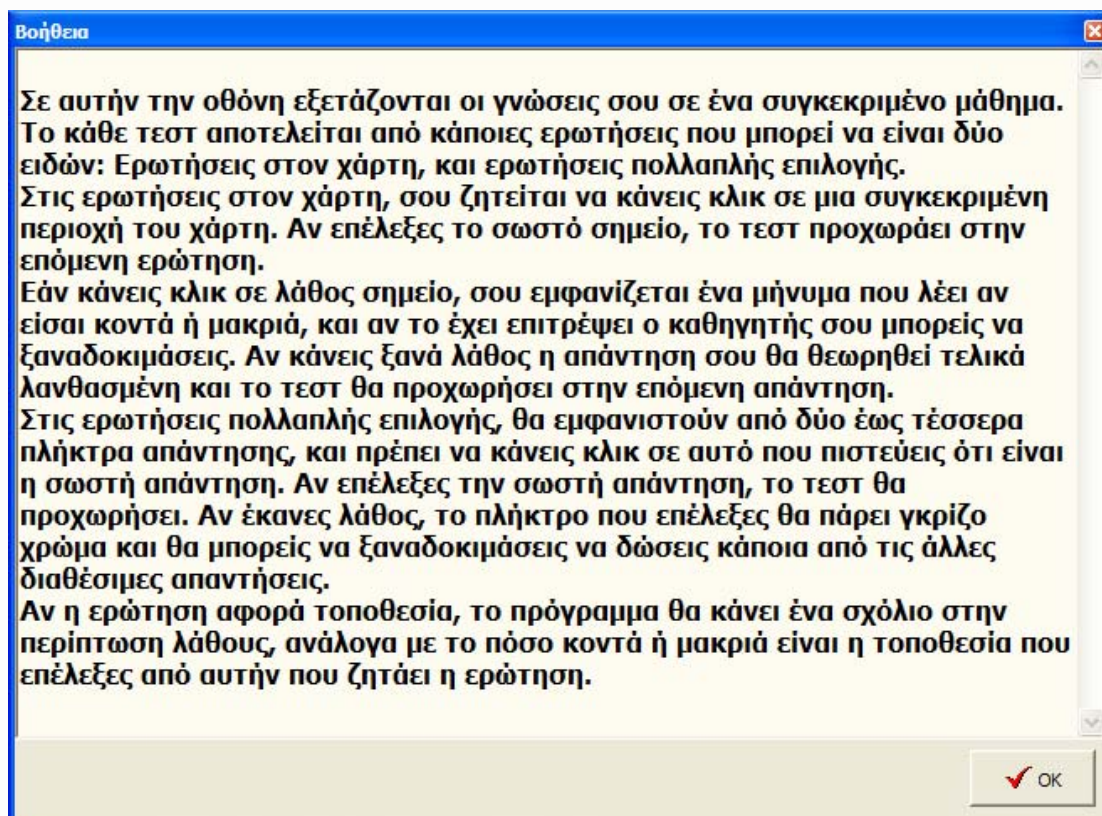
**Information**

Συγχαρητήρια! Πέρασες το τεστ με επιτυχία!  
Απάντησες σωστά τις 9 από τις 13 ερωτήσεις.  
Η βαθμολογία σου είναι: 13.8

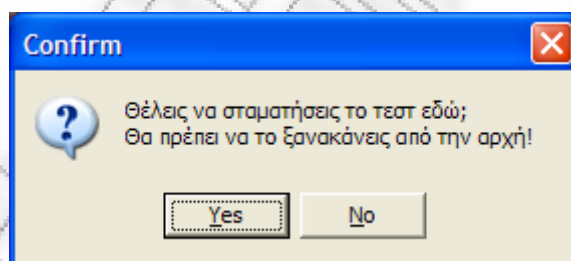
Μπορείς να προχωρήσεις στο επόμενο μάθημα!

OK

Με κλικ στο κουμπί «Βοήθεια», ο χρήστης μπορεί να δει οδηγίες για την εξέταση του τεστ :



Τέλος, κάνοντας κλικ στο πλήκτρο «Έξοδος» ο μαθητής επιστρέφει στην βασική «Οθόνη του Μαθητή». Πριν όμως επιστρέψει βγαίνει το παρακάτω μήνυμα :



Και ο μαθητής επιλέγει αν θέλει να συνεχίσει το τεστ ή να το επαναλάβει κάποια άλλη φορά.

Στην παρακάτω οθόνη, ο μαθητής μπορεί να αλλάξει το συνθηματικό που χρησιμοποιεί για να συνδεθεί στο πρόγραμμα. Η οθόνη αυτή εμφανίζεται, όταν ο μαθητής κάνει κλικ στο πλήκτρο «Αλλαγή Συνθηματικού» στην βασική «Οθόνη του Μαθητή» :

**Αλλαγή συνθηματικού**

Γράψτε το παλιό σας συνθηματικό στο επάνω πλαίσιο, και το νέο συνθηματικό και στα δύο κάτω πλαίσια.

Παλιό συνθηματικό:

Νέο συνθηματικό:

Επανάληψη:

**ΠΡΟΣΟΧΗ: Για το συνθηματικό έχει σημασία αν γράφετε με κεφαλαία ή μικρά!**

? OK Άκυρο

Στο πάνω πλαίσιο ο μαθητής γράφει το παλιό του συνθηματικό, και στα δύο κάτω πλαίσια (και στα δύο) γράφει το καινούριο που επιθυμεί να έχει. Κάνοντας κλικ στο «OK» αποθηκεύεται το νέο συνθηματικό. Αν ο μαθητής αλλάξει γνώμη και θέλει να κρατήσει το παλιό συνθηματικό, κάνει κλικ στο «Άκυρο».

Με κλικ στο κουμπί «Βοήθεια», ο χρήστης μπορεί να δει οδηγίες για την αλλαγή του συνθηματικού του :

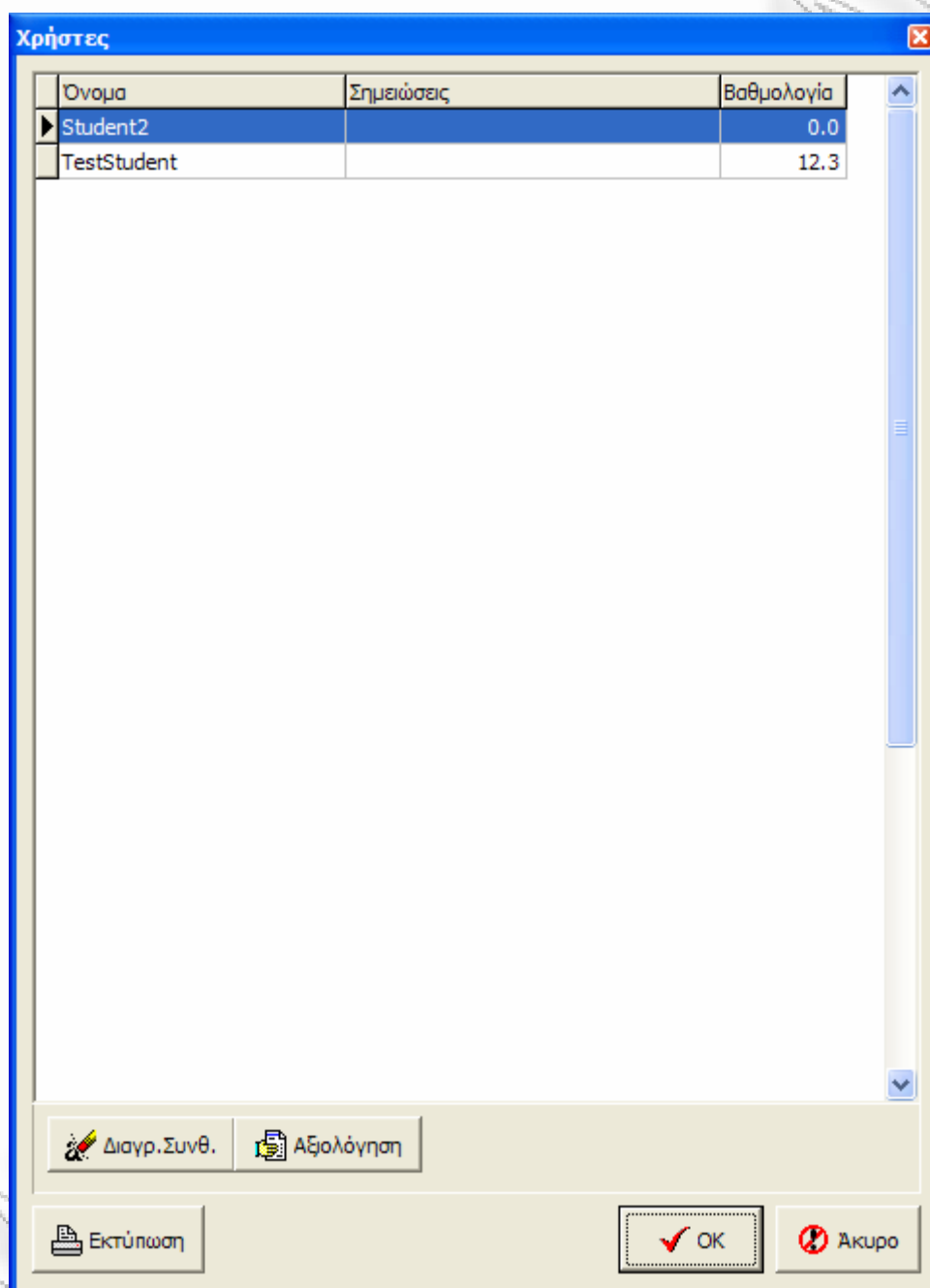
**Βοήθεια**

Σε αυτήν την οθόνη μπορείς να αλλάξεις το συνθηματικό που χρησιμοποιείς για να συνδεθείς στο πρόγραμμα. Στο επάνω πλαίσιο γράφεις το παλιό συνθηματικό, και στα δύο κάτω πλαίσια (και στα δύο!) γράφεις το καινούριο που θέλεις να έχεις. Όταν είσαι σίγουρος ότι έγραψες το συνθηματικό που θέλεις, κάνε κλικ στο «OK». Αν άλλαξες γνώμη και θες να κρατήσεις το παλιό συνθηματικό, κάνε κλικ στο «Άκυρο».

OK

## ΟΘΟΝΕΣ ΚΑΘΗΓΗΤΗ

Ο καθηγητής έχει τη δυνατότητα να «διαχειριστεί» τους μαθητές που παρακολουθούν τα μαθήματα και εξετάζονται στα τεστ. Η οθόνη αυτή εμφανίζεται, όταν ο καθηγητής κάνει κλικ στο πλήκτρο «Διαχείριση Μαθητών» στην βασική «Οθόνη του Καθηγητή» :



Στο πάνω μέρος της οθόνης εμφανίζονται οι ενεργοί μαθητές. Ο καθηγητής έχει τη δυνατότητα να δει τα ονόματα των μαθητών, τυχόν σημειώσεις που έχει κρατήσει και τη συνολική τους βαθμολογία.

Στο κάτω μέρος της οθόνης υπάρχει το κουμπί «Διαγραφή Συνθηματικού», με το οποίο ο καθηγητής μπορεί να διαγράψει το συνθηματικό κάποιου μαθητή, εφόσον τον έχει επιλέξει.

Στο κάτω μέρος της οθόνης υπάρχει, επίσης το κουμπί «Αξιολόγηση». Πατώντας το ο καθηγητής, αφού έχει επιλέξει κάποιον μαθητή, βλέπει την καρτέλα αξιολόγησης του :

The screenshot shows a window titled "Αξιολόγηση Μαθητή: TestStudent" with a close button in the top right corner. The window contains the following information:

<b>Γενική Βαθμολογία:</b>	<b>12.3</b>
Μαθήματα που παρακολούθησε:	<b>3</b>
Ερωτήσεις που εξετάστηκε:	<b>55</b>
Σωστές απαντήσεις:	<b>42</b>

Below this table, there is a section titled "Επιδόσεις ανά κατηγορία ερωτήσεων:" containing two rows:

Καλύτερη Κατηγορία:	ΣΥΝΟΡΑ ΝΟΜΩΝ (100% Επιτυχία)
Χειρότερη Κατηγορία:	ΠΡΩΤ.ΝΟΜΩΝ (50% Επιτυχία)

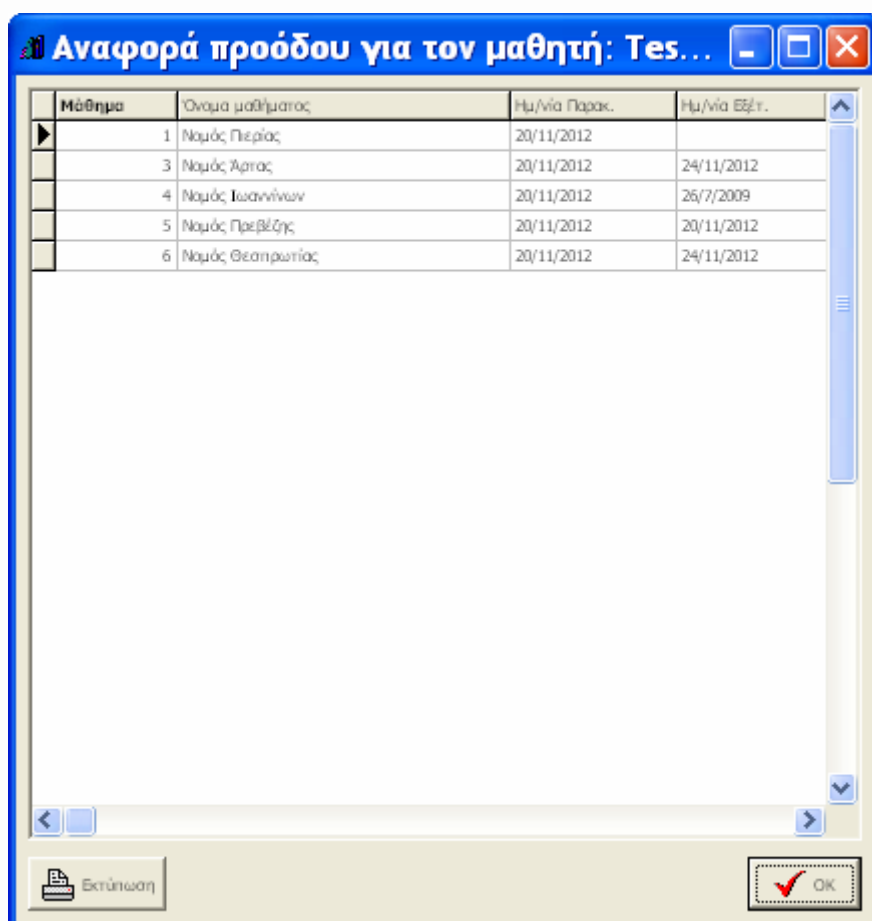
At the bottom of the window, there are two buttons: a button with a question mark icon and the text "Αναλυτικά..." and a button with a checkmark icon and the text "OK".

Στην οθόνη αυτή εμφανίζεται η Καρτέλα Αξιολόγησης, όπου ο καθηγητής μπορεί να δει τα εξής στοιχεία για την πρόοδο του μαθητή:

- **Γενική Βαθμολογία:** Ο μέσος όρος της βαθμολογίας μέχρι τώρα, ανάλογα με το ποσοστό των σωστών απαντήσεων που έδωσε στα τεστ
- **Ερωτήσεις που εξετάστηκε:** Ο συνολικός αριθμός των ερωτήσεων στα διάφορα τεστ που έχει απαντήσει
- **Σωστές Απαντήσεις:** Πόσες από τις απαντήσεις που έδωσε στις παραπάνω ερωτήσεις ήταν σωστές.

Μέσα στο πλαίσιο της οθόνης, ο καθηγητής έχει τη δυνατότητα να δει σε ποια κατηγορία ερωτήσεων ο μαθητής έχει την καλύτερη απόδοση και το ποσοστό επιτυχίας που έχει σε αυτές τις ερωτήσεις. Ακριβώς από κάτω μπορεί να δει σε ποια κατηγορία είχε τη χαμηλότερη βαθμολογία, δηλαδή το χαμηλότερο ποσοστό σωστών απαντήσεων.

Εάν το επιθυμεί, ο καθηγητής μπορεί να κάνει κλικ στο κουμπί «Αναλυτικά» και θα δει αναλυτικά τους τίτλους των μαθημάτων που παρακολούθησε ο μαθητής, την ημερομηνία παρακολούθησης κάθε μαθήματος, την ημερομηνία εξέτασης κάθε τεστ και τη βαθμολογία που έλαβε σε κάθε εξέταση :



The screenshot shows a software window with a blue title bar containing the text 'Αναφορά προόδου για τον μαθητή: Tes...'. The window contains a table with four columns: 'Μάθημα', 'Όνομα μαθήματος', 'Ημ/νία Παρακ.', and 'Ημ/νία Εξέτ.'. Below the table, there are two buttons: 'Εκτύπωση' (Print) and 'OK'.

Μάθημα	Όνομα μαθήματος	Ημ/νία Παρακ.	Ημ/νία Εξέτ.
1	Νομός Περίας	20/11/2012	
3	Νομός Άρτας	20/11/2012	24/11/2012
4	Νομός Ιωαννίνων	20/11/2012	26/7/2009
5	Νομός Πρεβέζης	20/11/2012	20/11/2012
6	Νομός Θεσπρωτίας	20/11/2012	24/11/2012

Τόσο στην οθόνη που εμφανίζεται, όταν ο καθηγητής κάνει κλικ στο πλήκτρο «Διαχείριση Μαθητών» στην βασική «Οθόνη του Καθηγητή», όσο και στην οθόνη «Αναφορά προόδου για τον μαθητή» υπάρχει η επιλογή εκτύπωσης.

Επιλέγοντας τη συνοπτική αναφορά, ο καθηγητής μπορεί να δει σε κάθε μάθημα το μέσο όρο βαθμολογίας σε όλα τα τεστ που έκανε ο μαθητής.

Μαθητής	Γεωγραφία	Ιστορία
TestStudent	12,4 (1 μαθήματα)	8,6 (2 μαθήματα)

Στην παρακάτω οθόνη, ο καθηγητής μπορεί να αλλάξει το συνθηματικό που χρησιμοποιεί για να συνδεθεί στο πρόγραμμα. Η οθόνη αυτή εμφανίζεται, όταν ο καθηγητής κάνει κλικ στο πλήκτρο «Αλλαγή Συνθηματικού» στην βασική «Οθόνη του Καθηγητή» :

**Αλλαγή συνθηματικού**

Γράψτε το παλιό σας συνθηματικό στο επάνω πλαίσιο, και το νέο συνθηματικό και στα δύο κάτω πλαίσια.

Παλιό συνθηματικό:

Νέο συνθηματικό:

Επανάληψη:

**ΠΡΟΣΟΧΗ: Για το συνθηματικό έχει σημασία αν γράφετε με κεφαλαία ή μικρά!**

? OK Άκυρο

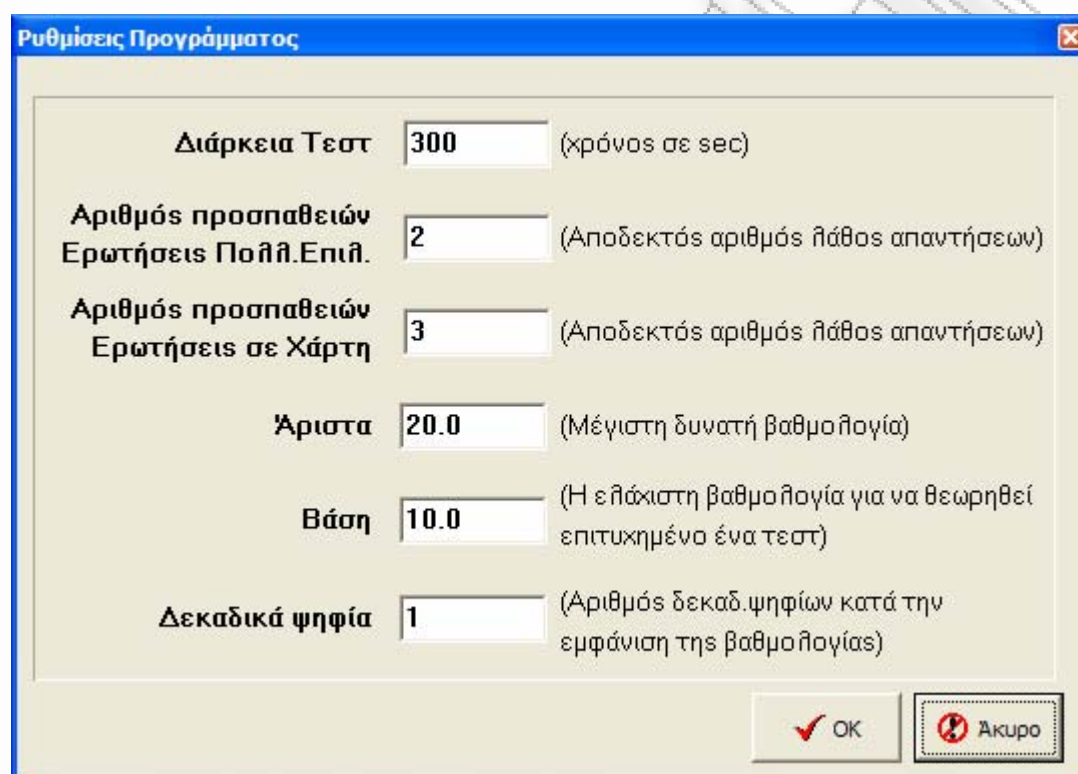
Στο πάνω πλαίσιο ο καθηγητή γράφει το παλιό του συνθηματικό, και στα δύο κάτω πλαίσια (και στα δύο) γράφει το καινούριο που επιθυμεί να έχει. Κάνοντας κλικ στο «OK»



αποθηκεύεται το νέο συνθηματικό. Αν ο καθηγητής αλλάξει γνώμη και θέλει να κρατήσει το παλιό συνθηματικό, κάνει κλικ στο «Άκυρο».

## ΟΘΟΝΕΣ ΔΙΑΧΕΙΡΙΣΤΗ

Ο administrator έχει τη δυνατότητα να αλλάζει και να καθορίζει τις ρυθμίσεις των τεστ. Πιο αναλυτικά μπορεί να καθορίζει την διάρκεια του τεστ σε δευτερόλεπτα, τον αριθμό των προσπαθειών σε ερωτήσεις πολλαπλής επιλογής, τον αριθμό των προσπαθειών σε ερωτήσεις με χάρτη, τη μέγιστη δυνατή βαθμολογία (άριστα), την ελάχιστη βαθμολογία για να θεωρηθεί επιτυχημένο ένα τεστ (βάση) και τα δεκαδικά ψηφία κατά την εμφάνιση της βαθμολογίας. Η παρακάτω οθόνη είναι ενδεικτική :



Παράμετρος	Τιμή	Περιγραφή
Διάρκεια Τεστ	300	(χρόνος σε sec)
Αριθμός προσπαθειών Ερωτήσεις Πολλ.Επιλ.	2	(Αποδεκτός αριθμός λάθος απαντήσεων)
Αριθμός προσπαθειών Ερωτήσεις σε Χάρτη	3	(Αποδεκτός αριθμός λάθος απαντήσεων)
Άριστα	20.0	(Μέγιστη δυνατή βαθμολογία)
Βάση	10.0	(Η ελάχιστη βαθμολογία για να θεωρηθεί επιτυχημένο ένα τεστ)
Δεκαδικά ψηφία	1	(Αριθμός δεκαδ.ψηφίων κατά την εμφάνιση της βαθμολογίας)

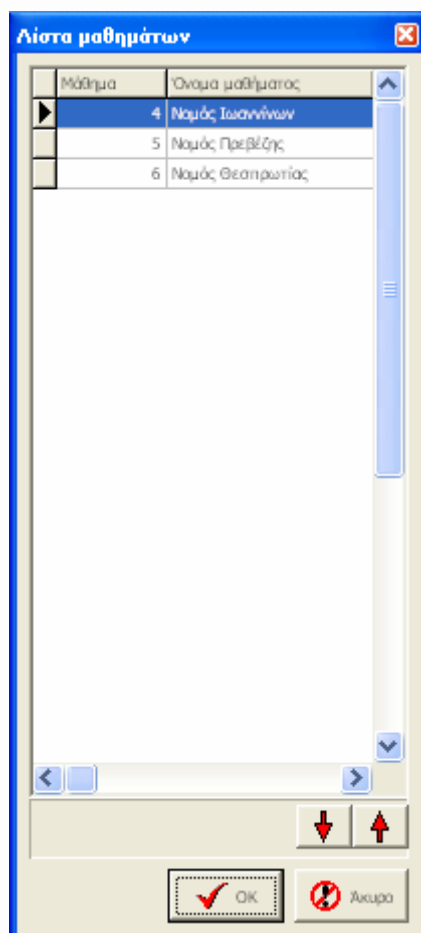
Πατώντας το κουμπί «OK», γίνονται αποδεκτές όλες οι αλλαγές, ενώ με το κουμπί «Άκυρο» δεν πραγματοποιείται καμία αλλαγή.

Ο administrator έχει τη δυνατότητα να «διαχειριστεί» τα μαθήματα που παρακολουθούν οι μαθητές. Η οθόνη αυτή εμφανίζεται, όταν ο administrator κάνει κλικ στο πλήκτρο «Διαχείριση Μαθημάτων» στην βασική «Οθόνη του Administrator» :

Κατάσταση	Τίτλος Καρτέλας	Κείμενο	Αρχείο Πακέτων	Αρχείο Σελίδας
1	Εισαγωγή	Ο κατάλογος των μαθημάτων είναι ένας κατάλογος που περιλαμβάνει όλα τα μαθήματα που υπάρχουν στο σύστημα.		catalogo.html
2	Διαγραφή Μαθήματος	Ο κατάλογος των μαθημάτων είναι ένας κατάλογος που περιλαμβάνει όλα τα μαθήματα που υπάρχουν στο σύστημα.		delete.html
3	Η πύλη των μαθημάτων	Το μεγαλύτερο μέρος του κατάλογου των μαθημάτων είναι η πύλη των μαθημάτων.		portal.html
4	Βασικά και προγράμματα	Οι πληροφορίες που διαβάζονται τον κατάλογο των μαθημάτων.		basic.html
5	Πρόγραμμα	Η πύλη των μαθημάτων είναι χωρισμένη σε δύο μέρη.		program.html
6	Μηνιαία λειτουργία και αρχείο	Το κείμενο του κατάλογου είναι χωρισμένο σε δύο μέρη.		monthly.html
7	Κείμενο	Το κείμενο του κατάλογου είναι χωρισμένο σε δύο μέρη.		text.html
8	Αρχείο - Μηνιαία	Ο κατάλογος των μαθημάτων είναι χωρισμένος σε δύο μέρη.		file.html

Στα αριστερά θα βρίσκονται μία σειρά από πλήκτρα για την πραγματοποίηση των σημαντικότερων λειτουργιών, όπως η δημιουργία νέου μαθήματος και η διαγραφή υπάρχοντος μαθήματος. Όπως, επίσης, και βέλη τα οποία μπορούν να μεταφέρουν τον administrator στο πρώτο ή το τελευταίο μάθημα και στο προηγούμενο ή το επόμενο.

Στα δεξιά και στο μεγαλύτερο κομμάτι της οθόνης εμφανίζεται ένα grid με στοιχεία για των μαθημάτων. Στο πάνω μέρος φαίνεται η σειρά στην οποία βρίσκεται ο τίτλος του μαθήματος και ο αριθμός που καθορίζει τη σειρά παρακολούθησης του μαθήματος. Πατώντας τις τελίτσες εμφανίζεται η παρακάτω οθόνη :



Η οποία δίνει τη δυνατότητα στον administrator να αλλάξει τη σειρά παρακολούθησης των μαθημάτων. Αυτό γίνεται επιλέγοντας το μάθημα και πατώντας τα βέλη για να ανέβει ή να κατέβει προτεραιότητα.

Πιο κάτω βρίσκεται ένα grid όπου ο administrator μπορεί να καταχωρήσει κάποιες σημειώσεις που αφορούν το μάθημα, ώστε να τις χρησιμοποιήσει ο ίδιος ή κάποιος καθηγητής.

Ακόμη πιο κάτω βρίσκεται ένα grid όπου ο administrator μπορεί να διαχειριστεί την ύλη του μαθήματος, τις ερωτήσεις των τεστ και τον χάρτη που εμφανίζεται σε κάθε μάθημα. Στην πρώτη καρτέλα μπορεί να καθορίσει ή να αλλάξει την ύλη του μαθήματος :

Διαχείριση μαθημάτων

Μάθημα: 5 Τίτλος Μαθήματος: Γενικά Περιβάλλον

Σημαντικές μαθήματα

Νέο μάθημα  
Διαγραφή Μαθήματα  
Αποθήκευση Αλλαγών  
Ακύρωση Αλλαγών

Κεφάλαιο	Τίτλος Κεφαλαίου	Κείμενο	Αρχείο Πολυμέσων	Αρχείο Εικόνας
1	Εισαγωγή	Ο ναυός Περιβάλλον είναι ένας από τους 31 ναυό...		εισεως1.jpg
2	Διαμορφωσιών Στοιχεία	Επίπεδο έκταση 1.026 π. γλ.μ. και μήκος 99,25...		εισεως2.jpg
3	Η πόλη της Περιβάλλον	Η Περιβάλλον είναι η πρωτεύουσα του ναυού Πε...		εισεως3.jpg
4	Ποιότητα και περιβάλλον	Το έδαφος του ναυού Περιβάλλον είναι γενικά κ...		εισεως4.jpg
5	Ποιότητα	Οι ποταμοί που διαρρέουν τον ναυό Περιβάλλον...		εισεως5.jpg
6	Αίματα, λιμνοθάλασσες και αετός	Η σημαντικότερη λίμνη του ναυού είναι η σελ...		εισεως6.jpg
7	Κλίμα	Το κλίμα του ναυού στις παράλιες περιοχές...		εισεως7.jpg
8	Αρφάλματα - Μηνιαία	Ο ναυός Περιβάλλον είναι η κορυφή σε φυσική κ...		εισεως8.jpg

Έξοδος

Όπως φαίνεται και στην παραπάνω οθόνη, ο administrator μπορεί να καθορίσει την σειρά κάθε κεφαλαίου του μαθήματος, να αλλάξει τον τίτλο και το κείμενο που αναφέρεται στο κάθε κεφάλαιο και να προσθέσει ή να αφαιρέσει αρχείο πολυμέσων και εικόνων. Όλα τα παραπάνω πραγματοποιούνται με τη χρήση των κουμπιών που βρίσκονται στο κάτω μέρος της οθόνης, δεξιά. Με το πρώτο κουμπί γίνεται προσθήκη νέου κεφαλαίου και προσθήκη αρχείου πολυμέσων ή εικόνων. Με το δεύτερο κουμπί γίνεται διαγραφή κεφαλαίου και διαγραφή αρχείου πολυμέσων ή εικόνων. Με τα δύο επόμενα κουμπιά, που αποτελούν βέλη, πραγματοποιείται η αλλαγή σειράς των κεφαλαίων. Πραγματοποιώντας οποιαδήποτε αλλαγή, ενεργοποιούνται τα κουμπιά «Αποθήκευση Αλλαγών» και «Ακύρωση Αλλαγών», που βρίσκονται στο αριστερό μέρος της οθόνης :

**Διαχείριση μαθημάτων**

Μάθημα: 6 Τίτλος Μαθήματος: Νεώσις Θεωρητικής

Σημειώσεις μαθηματος:

Υψηλ. Μαθήματος: Ερωτήσεις | Χάρτες

Κεφάλαιο	Τίτλος Κεφαλαίου	Κείμενο	Αρχείο Πακέτων	Αρχείο Ενός
1	Εισαγωγή	Ο νεώσις Θεωρητικής είναι ένας ένας ένας		θεωρητικα.zip
2	Διμορφικό Στοιχείο	Ο νεώσις Θεωρητικής έχει ύψος 1.515 τ. γλ		neoi_1mu1c.zip
3	Η πόλη της Ηνωμένης	Η Ηνωμένη είναι παραρτή πόλη, πρώτη		krumenitsa1.zip
4	Βουνά και πεδιάδες	Η Θεωρητική είναι περιοχή κατάλληλη ορεινή		soai.zip
5	Ποτάμια	Οι κερφότα ποταμοί που διαρρέουν τον νεώσι		skala_bsan.zip
6	Λίμνες, λιμνοθάλασσες και αλιεία	Στον νεώσι Θεωρητικής υπάρχουν αρκετές λίμ		subota.zip
7	Κλίμα	Στα παράλια κα στη χαμηλή ζώνη του εσωτερ		ph_1ptmavajia.zip
8	Αρδάσεις - Μηχανή	Ο νεώσις Θεωρητικής είναι πλούσιος σε φυσ		Fanoti.zip
9				

Έξοδος

Επιλέγοντας το κουμπί «Αποθήκευση Αλλαγών», αποθηκεύονται όλες οι αλλαγές που πραγματοποιήθηκαν, ενώ με το κουμπί «Ακύρωση Αλλαγών», ακυρώνονται όλες οι τροποποιήσεις.

Στην δεύτερη καρτέλα ο administrator μπορεί να καθορίσει ή να αλλάξει τις ερωτήσεις που αφορούν κάθε μάθημα :

**Διαχείριση μαθημάτων**

Μάθημα: 6 Τίτλος Μαθήματος: Νεώσις Θεωρητικής

Σημειώσεις μαθηματος:

Υψηλ. Μαθήματος: Ερωτήσεις | Χάρτες

Ερώτηση	Κατηγορία	Κείμενο ερώτησης	Απάντηση	Σε Χάρτη	Αρχείο ενός
1	ΧΑΡΤΗΣ	Δείξε το νεώσι Θεωρητικής στον χάρτη της Ελλάδας	Νεώσις Θεωρητικής	ΝΑΙ	
2	ΓΕΩΓΡ.ΔΙΑΜΟΡΦ.	Σε ποια γεωγραφικό θεματικό ανήκει ο νεώσις Θεωρητικής;	Υψίμετρος	ΟΧΙ	
3	ΠΡΟΤ.ΜΟΜΩΝ	Ποια είναι η πρωτεύουσα του νεώσι Θεωρητικής;	Ηνωμένη	ΟΧΙ	krumenitsa1.zip
4	ΣΥΝΟΡΑ ΝΟΜΩΝ	Ο νεώσις Θεωρητικής συνορεύει με το νεώσι Πρεβέζης;	Σωστό	ΟΧΙ	
5	ΣΥΝΟΡΑ ΝΟΜΩΝ	Ο νεώσις Θεωρητικής συνορεύει με το νεώσι Έβρου;	Λάθος	ΟΧΙ	
6	ΔΙΡΡΟΤΡΑΦΙΚΑ	Η κανότητα Σουλίου ανήκει στον νεώσι Θεωρητικής;	Σωστό	ΟΧΙ	soai_ghfuti.zip
7	ΔΙΡΡΟΤΡΑΦΙΚΑ	Ο Δήμος Σούλων ανήκει στον νεώσι Θεωρητικής;	Σωστό	ΟΧΙ	subota.zip
8	ΒΟΥΝΑ	Ποια είναι το ψηλότερο βουνό του νεώσι Θεωρητικής;	Μαυροβούνια	ΟΧΙ	
9	ΒΟΥΝΑ	Το Όρη Φιλιπών είναι βουνό του νεώσι Θεωρητικής;	Σωστό	ΟΧΙ	
10	ΠΟΤΑΜΙΑ	Ποιας από τους παρακάτω ποταμούς διαρρέει τον νεώσι Θεωρητικής;	Αχελώϊας	ΟΧΙ	skala_bsan.zip
11	ΠΟΤΑΜΙΑ	Ο ποταμός Καλάμιας εκβάλλει στα παράλια του νεώσι Θεωρητικής;	Σωστό	ΟΧΙ	
12	ΛΙΜΝΕΣ	Η λίμνη Παμβώτιδα βρίσκεται στον νεώσι Θεωρητικής;	Λάθος	ΟΧΙ	
13	ΛΙΜΝΕΣ	Ποια από τις παρακάτω λίμνες βρίσκεται στον νεώσι Θεωρητικής;	Λίμνη των Νικαράδων	ΟΧΙ	

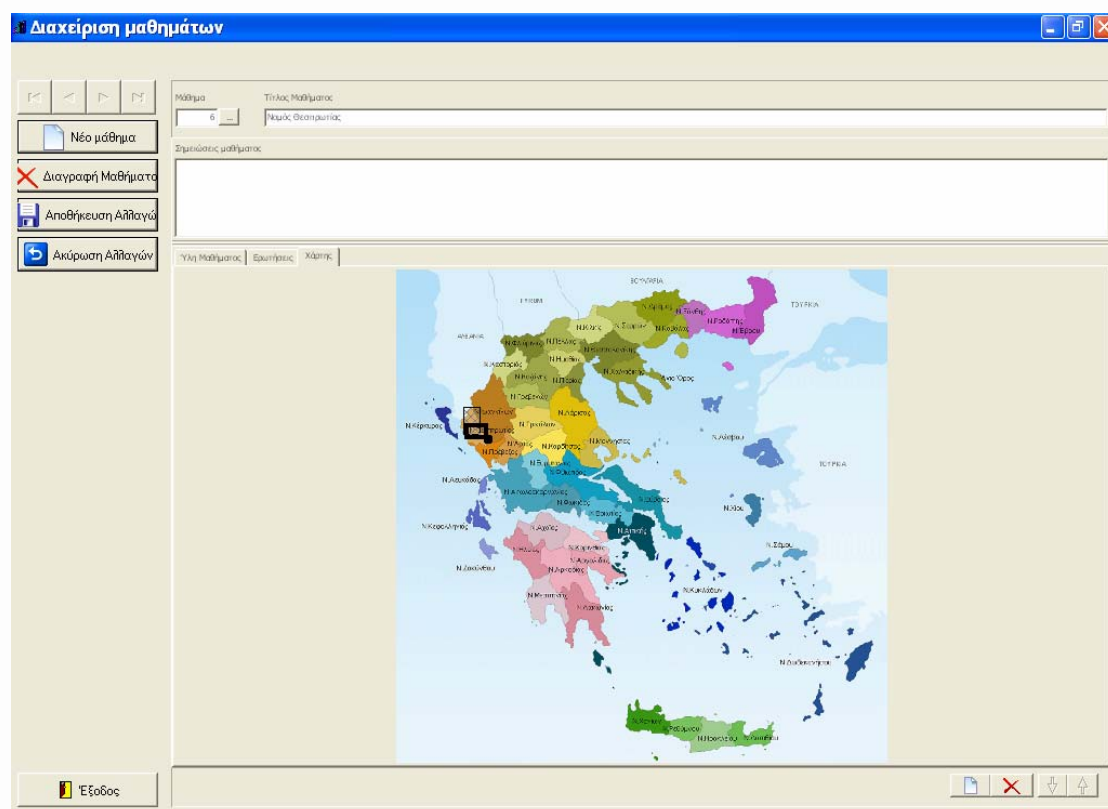
Έξοδος

Όπως φαίνεται και στην παραπάνω οθόνη, ο administrator μπορεί να καθορίσει την σειρά κάθε ερώτησης του τεστ, να αλλάξει την κατηγορία που ανήκει κάθε ερώτηση, να αλλάξει το κείμενο που αναφέρεται στο κάθε ερώτηση και την απάντηση, να προσθέσει ή να αφαιρέσει ερώτηση από χάρτη και αρχείο εικόνων. Όλα τα παραπάνω πραγματοποιούνται με τη χρήση των κουμπιών που βρίσκονται στο κάτω μέρος της οθόνης, δεξιά. Με το πρώτο κουμπί γίνεται προσθήκη νέας ερώτησης και προσθήκη αρχείου εικόνων ή ερώτησης στον χάρτη. Με το δεύτερο κουμπί γίνεται διαγραφή ερώτησης, απάντησης και διαγραφή αρχείου εικόνων. Με τα δύο επόμενα κουμπιά, που αποτελούν βέλη, πραγματοποιείται η αλλαγή σειράς των ερωτήσεων. Πραγματοποιώντας οποιαδήποτε αλλαγή, ενεργοποιούνται τα κουμπιά «Αποθήκευση Αλλαγών» και «Ακύρωση Αλλαγών», που βρίσκονται στο αριστερό μέρος της οθόνης:

Ερώτηση	Κατηγορία	Κείμενο ερώτησης	Απάντηση	Στ. Χάρτη	Αρχείο εικόνας
1	ΣΥΝΘΡΑ ΝΟΜΩΝ	Ο νομικός Θεσπρωτίας αναγκάζο με το ναυό Έβρου	Αόθος	ΟΧΙ	
2	ΧΑΡΤΗΣ	Δείξε το ναυό Θεσπρωτίας στον χάρτη της Ελλάδας	Νομικός Θεσπρωτίας	ΝΑΙ	
3	ΓΕΩΓΡ.ΔΙΑΜΕΡ.	Σε ποιο γεωγραφικό διαμέρισμα ανήκει ο νομικός Θεσπρωτίας	Ήπειρος	ΟΧΙ	
4	ΠΡΩΤ.ΝΟΜΩΝ	Παρά είναι η πραγματικότητα του νομικού Θεσπρωτίας	Μοταμηνίσις	ΟΧΙ	krumenitsa1.bmp
5	ΣΥΝΘΡΑ ΝΟΜΩΝ	Ο νομικός Θεσπρωτίας αναγκάζο με το ναυό Πετρίτζης	Σωστό	ΟΧΙ	
6	ΔΗΡΟΤΡΑΦΙΚΑ	Η κανόνισσα Σωλίου ανήκει στον ναυό Θεσπρωτίας	Σωστό	ΟΧΙ	souli_gfari1.bmp
7	ΔΗΡΟΤΡΑΦΙΚΑ	Ο Δέμος Σωλίου ανήκει στον ναυό Θεσπρωτίας	Σωστό	ΟΧΙ	subota1.bmp
8	ΒΟΥΛΑ	Παρά είναι το ψηφίσις βουλό του νομικού Θεσπρωτίας	Μακρυκόνα	ΟΧΙ	
9	ΒΟΥΛΑ	Το Όρι Φιλισίων είναι βουλό του νομικού Θεσπρωτίας	Σωστό	ΟΧΙ	
10	ΠΟΤΑΜΙΑ	Παρά από τους παραπάνω ποταμούς διασχίζει τον ναυό Θεσπρωτίας	Αρδωνίσις	ΟΧΙ	skala_tsiav1.bmp
11	ΠΟΤΑΜΙΑ	Ο ποταμός Κολυμής εκβάλλει στον ναυό Θεσπρωτίας	Σωστό	ΟΧΙ	
12	ΛΙΜΝΕΣ	Η λίμνη Παμβώτιδα βρίσκεται στον ναυό Θεσπρωτίας	Αόθος	ΟΧΙ	
13	ΛΙΜΝΕΣ	Παρά από τις παραπάνω λίμνες βρίσκεται στον ναυό Θεσπρωτίας	Λίμνη των Νικαράδων	ΟΧΙ	

Επιλέγοντας το κουμπί «Αποθήκευση Αλλαγών», αποθηκεύονται όλες οι αλλαγές που πραγματοποιήθηκαν, ενώ με το κουμπί «Ακύρωση Αλλαγών», ακυρώνονται όλες οι τροποποιήσεις.

Στην τρίτη καρτέλα ο administrator μπορεί να καθορίσει ή να αλλάξει την τοποθεσία του νομού που αφορά το μάθημα στον χάρτη :

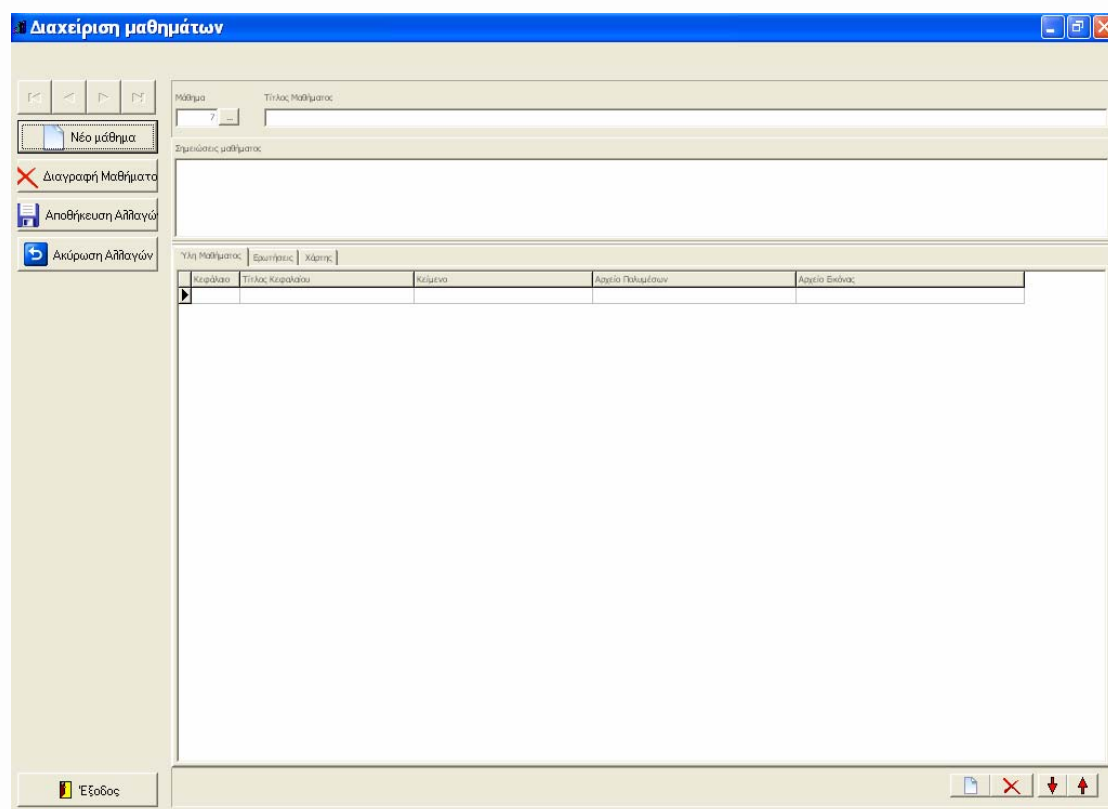


Ο καθορισμός του νομού πάνω στον χάρτη ή η αλλαγή και η διαγραφή πραγματοποιούνται με τη χρήση των κουμπιών που βρίσκονται στο κάτω μέρος της οθόνης, δεξιά. Με το πρώτο κουμπί γίνεται προσθήκη νέας τοποθεσίας πάνω στον χάρτη, ενώ με το δεύτερο κουμπί γίνεται διαγραφή της τοποθεσίας. Επιλέγοντας το κουμπί «Αποθήκευση Αλλαγών», αποθηκεύονται όλες οι αλλαγές που πραγματοποιήθηκαν, ενώ με το κουμπί «Ακύρωση Αλλαγών», ακυρώνονται όλες οι τροποποιήσεις.

Τέλος, κάνοντας κλικ στο πλήκτρο «Έξοδος» ο administrator επιστρέφει στην βασική «Οθόνη του Administrator».

Στην βασική «Οθόνη του Administrator» υπάρχουν και οι επιλογές προσθήκης νέου μαθήματος, πατώντας το κουμπί «Νέο Μάθημα» και διαγραφής μαθήματος πατώντας το κουμπί «Διαγραφή Μαθήματος».

Για να προστεθεί νέο μάθημα, εμφανίζεται η παρακάτω οθόνη :

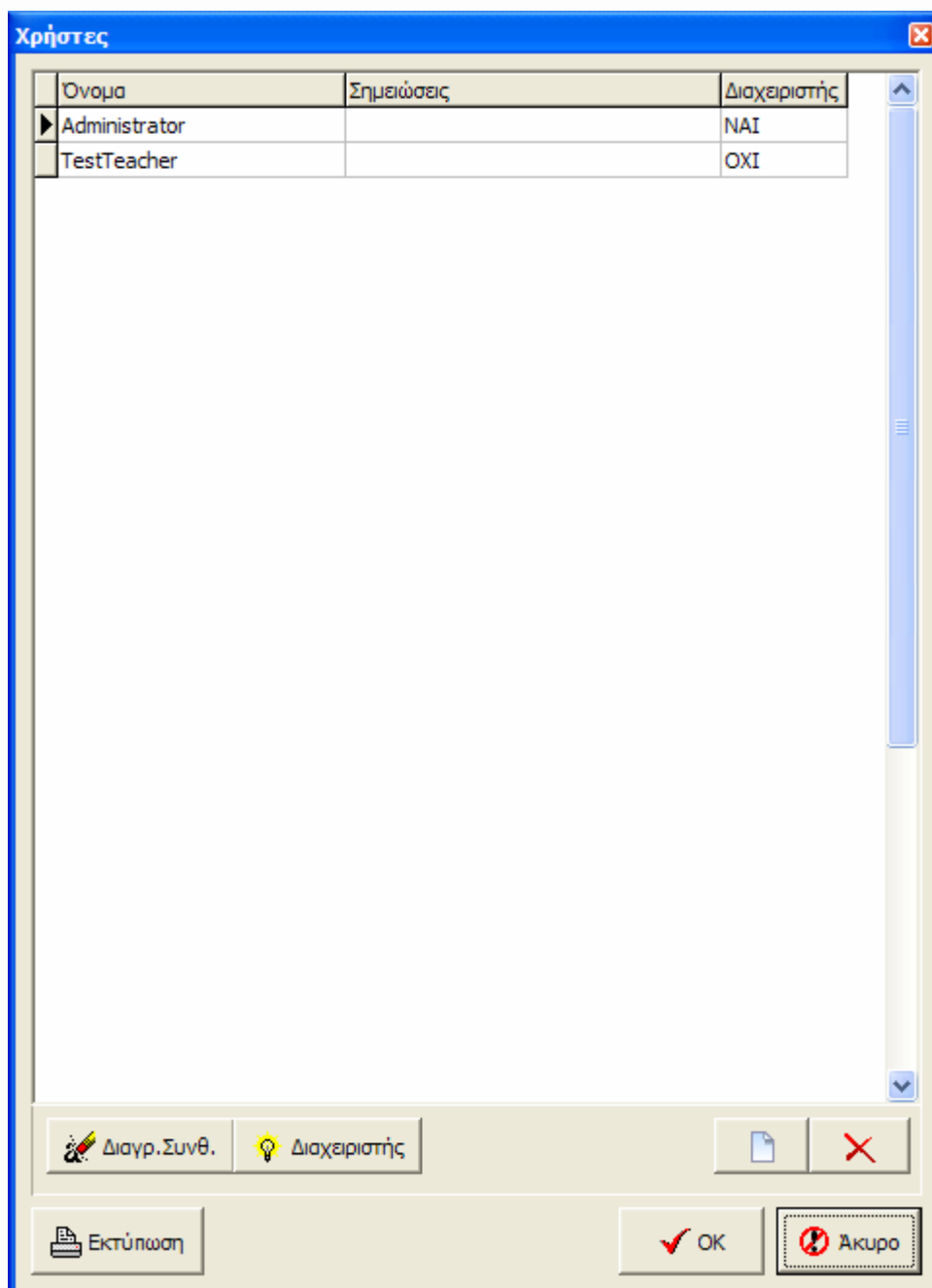


Το μάθημα παίρνει τον τελευταίο αριθμό σειράς, ο οποίος μπορεί να αλλάξει μετά την ενεργοποίηση των βελών, όπως έχει αναφερθεί και προηγουμένως. Επίσης, συμπληρώνονται από τον administrator ο τίτλος του μαθήματος, οι σημειώσεις, η ύλη του μαθήματος (αριθμός κεφαλαίου, τίτλος κεφαλαίου, κείμενο, αρχεία πολυμέσων, αρχεία εικόνων), οι ερωτήσεις που αφορούν το μάθημα (αριθμός ερώτησης, κατηγορία, κείμενο ερώτησης, απάντηση, αν ανήκει σε ερώτηση χάρτη, αρχείο εικόνων) και η τοποθεσία του νομού στον χάρτη. Όλες οι παραπάνω λειτουργίες, καθώς και η διαγραφή πραγματοποιούνται με τη χρήση των κουμπιών που βρίσκονται στο κάτω μέρος της οθόνης, δεξιά. Με το πρώτο κουμπί γίνεται προσθήκη, ενώ με το δεύτερο κουμπί γίνεται διαγραφή. Επιλέγοντας το κουμπί «Αποθήκευση Αλλαγών», αποθηκεύονται όλες οι αλλαγές που πραγματοποιήθηκαν, ενώ με το κουμπί «Ακύρωση Αλλαγών», ακυρώνονται όλες οι τροποποιήσεις.

Για να διαγραφεί κάποιο μάθημα, ο administrator πρέπει να κάνει κλικ στο κουμπί «Διαγραφή Μαθήματος».

Ο administrator έχει τη δυνατότητα να «διαχειριστεί» τους καθηγητές που συμμετέχουν στο μάθημα. Η οθόνη αυτή εμφανίζεται, όταν ο administrator κάνει κλικ στο πλήκτρο «Διαχείριση Καθηγητών» στην βασική «Οθόνη του Administrator»:





Στο πάνω μέρος της οθόνης εμφανίζονται οι ενεργοί καθηγητές. Ο administrator έχει τη δυνατότητα να δει τα ονόματα των καθηγητών και το δικό του, τυχόν σημειώσεις που έχει κρατήσει και εάν κάποιος καθηγητής έχει τα δικαιώματα του administrator.

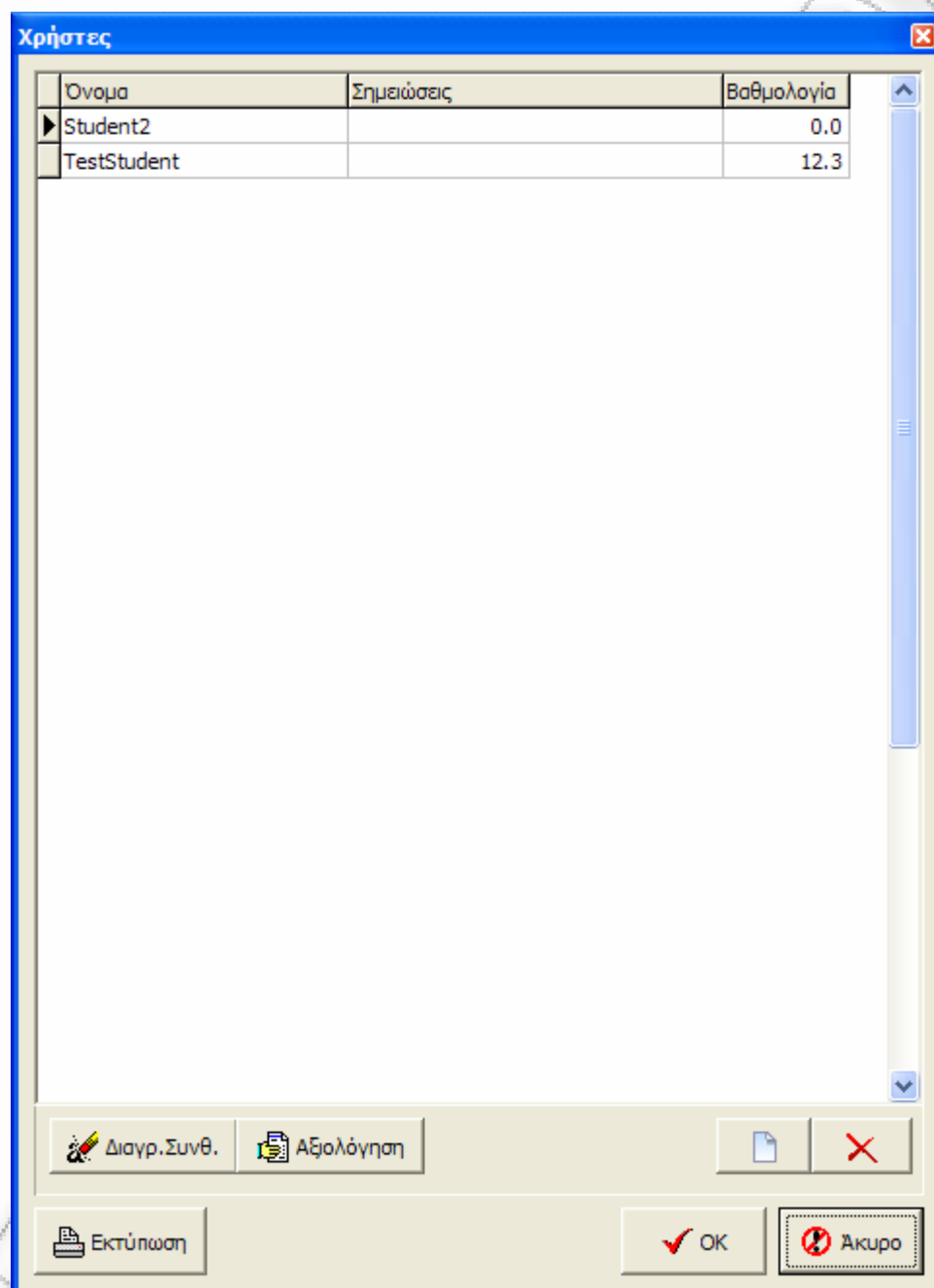
Στο κάτω μέρος της οθόνης, αριστερά, υπάρχει το κουμπί «Διαγραφή Συνθηματικού», με το οποίο ο administrator μπορεί να διαγράψει το συνθηματικό κάποιου καθηγητή, εφόσον τον έχει επιλέξει.

Στο κάτω μέρος της οθόνης, αριστερά, υπάρχει, επίσης, το κουμπί «Διαχειριστής». Πατώντας το ο administrator, αφού έχει επιλέξει κάποιον καθηγητή, μπορεί να του δώσει τα δικαιώματα του administrator.

Στο κάτω μέρος της οθόνης «Χρήστες», δεξιά, υπάρχει το κουμπί «Προσθήκη Καθηγητή», με το οποίο ο administrator μπορεί να προσθέσει έναν νέο καθηγητή και το κουμπί «Διαγραφή Καθηγητή», με το οποίο ο administrator μπορεί να διαγράψει έναν καθηγητή.

Τέλος στο κάτω μέρος, αριστερά υπάρχει η επιλογή εκτύπωσης, κάνοντας κλικ στο κουμπί «Εκτύπωση».

Ο administrator έχει τη δυνατότητα να «διαχειριστεί» τους μαθητές που παρακολουθούν τα μαθήματα και εξετάζονται στα τεστ. Η οθόνη αυτή εμφανίζεται, όταν ο administrator κάνει κλικ στο πλήκτρο «Διαχείριση Μαθητών» στην βασική «Οθόνη του Administrator»:



Στο πάνω μέρος της οθόνης εμφανίζονται οι ενεργοί μαθητές. Ο administrator έχει τη δυνατότητα να δει τα ονόματα των μαθητών, τυχόν σημειώσεις που έχει κρατήσει και τη συνολική τους βαθμολογία.

Στο κάτω μέρος της οθόνης, αριστερά, υπάρχει το κουμπί «Διαγραφή Συνθηματικού», με το οποίο ο administrator μπορεί να διαγράψει το συνθηματικό κάποιου μαθητή, εφόσον τον έχει επιλέξει.

Στο κάτω μέρος της οθόνης, αριστερά, υπάρχει, επίσης το κουμπί «Αξιολόγηση». Πατώντας το ο administrator, αφού έχει επιλέξει κάποιον μαθητή, βλέπει την καρτέλα αξιολόγησης του :

Αξιολόγηση Μαθητή: TestStudent	
<b>Γενική Βαθμολογία:</b>	12.3
Μαθήματα που παρακολούθησε:	3
Ερωτήσεις που εξετάστηκε:	55
Σωστές απαντήσεις:	42
Επιδόσεις ανά κατηγορία ερωτήσεων:	
Καλύτερη Κατηγορία:	ΣΥΝΟΡΑ ΝΟΜΩΝ (100% Επιτυχία)
Χειρότερη Κατηγορία:	ΠΡΩΤ.ΝΟΜΩΝ (50% Επιτυχία)
<input type="button" value="Αναλυτικά..."/> <input type="button" value="OK"/>	

Στην οθόνη αυτή εμφανίζεται η Καρτέλα Αξιολόγησης, όπου ο administrator μπορεί να δει τα εξής στοιχεία για την πρόοδο του μαθητή:

- **Γενική Βαθμολογία:** Ο μέσος όρος της βαθμολογίας μέχρι τώρα, ανάλογα με το ποσοστό των σωστών απαντήσεων που έδωσε στα τεστ
- **Ερωτήσεις που εξετάστηκε:** Ο συνολικός αριθμός των ερωτήσεων στα διάφορα τεστ που έχει απαντήσει
- **Σωστές Απαντήσεις:** Πόσες από τις απαντήσεις που έδωσε στις παραπάνω ερωτήσεις ήταν σωστές.

Μέσα στο πλαίσιο της οθόνης, ο administrator έχει τη δυνατότητα να δει σε ποια κατηγορία ερωτήσεων ο μαθητής έχει την καλύτερη απόδοση και το ποσοστό επιτυχίας που έχει σε αυτές τις ερωτήσεις. Ακριβώς από κάτω μπορεί να δει σε ποια κατηγορία είχε τη χαμηλότερη βαθμολογία, δηλαδή το χαμηλότερο ποσοστό σωστών απαντήσεων.

Η unit UserManagementUnit που περιγράφηκε παραπάνω χρησιμοποιείται και σε αυτό το σημείο για την περιγραφή της φόρμας «Χρήστες».

Εάν το επιθυμεί, ο administrator μπορεί να κάνει κλικ στο κουμπί «Αναλυτικά» και θα δει αναλυτικά τους τίτλους των μαθημάτων που παρακολούθησε ο μαθητής, την ημερομηνία παρακολούθησης κάθε μαθήματος, την ημερομηνία εξέτασης κάθε τεστ και τη βαθμολογία που έλαβε σε κάθε εξέταση :

Μάθημα	Όνομα μαθήματος	Ημ/νία Παρακ.	Ημ/νία Εξέτ.
1	Ναμός Περίας	20/11/2012	
3	Ναμός Άρτας	20/11/2012	24/11/2012
4	Ναμός Ιωαννίνων	20/11/2012	26/7/2009
5	Ναμός Πρεβέζης	20/11/2012	20/11/2012
6	Ναμός Θεσπρωτίας	20/11/2012	24/11/2012

Στο κάτω μέρος της οθόνης «Χρήστες», δεξιά, υπάρχει το κουμπί «Προσθήκη Μαθητή», με το οποίο ο administrator μπορεί να προσθέσει έναν νέο μαθητή και το κουμπί «Διαγραφή Μαθητή», με το οποίο ο administrator μπορεί να διαγράψει έναν μαθητή. Τόσο στην οθόνη που εμφανίζεται, όταν ο administrator κάνει κλικ στο πλήκτρο «Διαχείριση Μαθητών» στην βασική «Οθόνη του Administrator», όσο και στην οθόνη «Αναφορά προόδου για τον μαθητή» υπάρχει η επιλογή εκτύπωσης.

Στην παρακάτω οθόνη, ο administrator μπορεί να αλλάξει το συνθηματικό που χρησιμοποιεί για να συνδεθεί στο πρόγραμμα. Η οθόνη αυτή εμφανίζεται, όταν ο καθηγητής κάνει κλικ στο πλήκτρο «Αλλαγή Συνθηματικού» στην βασική «Οθόνη του Administrator» :

**Αλλαγή συνθηματικού**



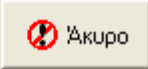
Γράψτε το παλιό σας συνθηματικό στο επάνω πλαίσιο, και το νέο συνθηματικό και στα δύο κάτω πλαίσια.

Παλιό συνθηματικό:

Νέο συνθηματικό:

Επανάληψη:

**ΠΡΟΣΟΧΗ: Για το συνθηματικό έχει σημασία αν γράφετε με κεφαλαία ή μικρά!**


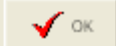
  

Στο πάνω πλαίσιο ο administrator γράφει το παλιό του συνθηματικό, και στα δύο κάτω πλαίσια (και στα δύο) γράφει το καινούριο που επιθυμεί να έχει. Κάνοντας κλικ στο «OK» αποθηκεύεται το νέο συνθηματικό. Αν ο administrator αλλάξει γνώμη και θέλει να κρατήσει το παλιό συνθηματικό, κάνει κλικ στο «Ακυρο».

Στην συνοπτική αναφορά, υπάρχει η πρόοδος του μαθητή στα εξεταζόμενα τεστ. Συγκεκριμένα, φαίνεται ο μέσος όρος βαθμολογίας του μαθητή στα τεστ που έλαβε μέρος ανά μάθημα. Επίσης, φαίνεται και σε πόσα τεστ εξετάστηκε ο μαθητής ανά μάθημα.

**StudentSummaryForm**

Μαθητής	Γεωγραφία	Ιστορία
TestStudent	12,4 (1 μαθήματα)	8,6 (2 μαθήματα)

 Εκτύπωση 

## ΣΥΜΠΕΡΑΣΜΑΤΑ

Η πληθώρα προϊόντων ηλεκτρονικής εκπαίδευσης παρέχει τη δυνατότητα επιλογής ανάλογα με τις συγκεκριμένες ανάγκες στις οποίες θα πρέπει να ανταποκριθεί για την παροχή υπηρεσιών ηλεκτρονικής εκπαίδευσης. Πέρα από την ικανοποίηση των λειτουργικών απαιτήσεων της υπηρεσίας ηλεκτρονικής εκπαίδευσης, υπάρχουν στην πράξη και διάφοροι άλλοι παράγοντες και περιορισμοί οι οποίοι θα πρέπει να ικανοποιούνται προκειμένου να ληφθεί η απόφαση υιοθέτησης κάποιας πλατφόρμας ηλεκτρονικής εκπαίδευσης. Πιο συγκεκριμένα, θέματα όπως το κόστος λειτουργίας και συντήρησης της υπηρεσίας ή των πόρων που αυτή απαιτεί για την ικανοποιητική και αποδοτική λειτουργία της θα πρέπει να συνεκτιμηθούν.

Όσον αφορά πιο συγκεκριμένα το χώρο των εκπαιδευτικών ιδρυμάτων, τα θέματα κόστους αποτελούν σημαντικό περιορισμό κατά την επιλογή μιας λύσης ηλεκτρονικής εκπαίδευσης. Παράλληλα θα πρέπει να ληφθεί υπόψη το γεγονός ότι είναι πολύ πιθανόν οι υπάρχουσες πλατφόρμες ηλεκτρονικής εκπαίδευσης να μην ανταποκρίνονται επακριβώς στις ανάγκες των ελληνικών εκπαιδευτικών ιδρυμάτων.

Στόχος της παρούσας εφαρμογής είναι η διερεύνηση των παραγόντων που επηρεάζουν την αποτελεσματικότητα της διδακτικής-μαθησιακής διαδικασίας μέσω ηλεκτρονικής εκπαίδευσης. Τα εργαλεία που χρησιμοποιήθηκαν σε αυτή την εφαρμογή διατηρούν τον πιλοτικό τους χαρακτήρα και τα συμπεράσματα που εξήχθηκαν εμφανίζονται ενθαρρυντικά όσον αφορά την αξιοπιστία και την εγκυρότητά τους, υπό την έννοια ότι φαίνεται να συμφωνούν με τη διεθνή βιβλιογραφία/ έρευνα, ενώ αποκαλύπτουν ενδιαφέρουσες ιδιαιτερότητες που σχετίζονται με τα δεδομένα της ελληνικής πραγματικότητας.

Σε ότι αφορά στους παράγοντες που διαμορφώνουν την αποτελεσματικότητα της διδασκαλίας μάθησης σε ηλεκτρονικό περιβάλλον, τα συμπεράσματα που προέκυψαν αφορούν τόσο την διαμόρφωση του παιδαγωγικού πλαισίου όσο και την οργάνωση και αξιοποίηση των τεχνολογικών μέσων.

Όπως διαπιστώνεται από της αξιολόγηση της παρούσας εφαρμογής, η αξιοποίηση της ηλεκτρονικής εκπαίδευσης μπορεί να συμβάλλει στη δημιουργία καινοτόμων διδακτικών/ μαθησιακών περιβαλλόντων για την υποστήριξη προγραμμάτων σπουδών δευτεροβάθμιας εκπαίδευσης. Τα συστήματα ηλεκτρονικής εκπαίδευσης με εικόνα και ήχο είναι σε θέση να εξασφαλίσουν την ισότιμη συμμετοχή των μαθητών κατά τη διδακτική διαδικασία, καθώς και την αποστολή και παρουσίαση ποικιλίας εκπαιδευτικού υλικού.

Τα αναμενόμενα αποτελέσματα είναι οι συμμετέχοντες μαθητές να εμφανίζονται σε μεγάλο βαθμό ικανοποιημένοι από την παρακολούθηση των ηλεκτρονικών μαθημάτων. Βασικοί παράγοντες που επέδρασαν ενισχυτικά ως προς την ικανοποίηση των προσδοκιών τους ήταν η ίδια η τεχνολογική καινοτομία και η καλή οργάνωση και η προγραμματισμένη διεξαγωγή των διδακτικών δραστηριοτήτων. Αντίθετα, οι παράγοντες που φαίνεται πως θα δημιουργήσουν δυσκολίες κατά την παρακολούθηση της διδασκαλίας σχετίζονται με τη μικρή εξοικείωση των μαθητών με τα τεχνολογικά μέσα.

Προκειμένου να εκμεταλλευτούμε τις δυνατότητες που προσφέρει η ραγδαία τεχνολογική εξέλιξη απαιτούνται:

- Νέες ιδέες διδακτικών προσεγγίσεων οι οποίες θα προσαρμόζουν τις διδακτικές τεχνικές όχι μόνο στις δυνατότητες των νέων μέσων αλλά περισσότερο στις μαθησιακές απαιτήσεις των συμμετεχόντων.
- Σχεδιασμός «ευρηματικού» - πολυμεσικού εκπαιδευτικού υλικού, το οποίο θα ανταποκρίνεται στις προδιαγραφές των εικονικών περιβαλλόντων αφενός και αφετέρου θα μπορεί να καλύψει τις γνωστικές ανάγκες των διδασκομένων.
- Στενή συνεργασία τεχνολόγων και παιδαγωγών για τον σχεδιασμό και την υποστήριξη της διδακτικής διαδικασίας.

- Καλύτερη ενημέρωση και εμπλοκή της Ακαδημαϊκής κοινότητας για τις δυνατότητες και τις προδιαγραφές των νέων περιβαλλόντων.
- Διαρκής αξιολόγηση και βελτίωση τόσο τεχνολογική όσο και παιδαγωγική.

Τα κυριότερα πλεονεκτήματα από την εφαρμογή της ηλεκτρονικής πλατφόρμας «e-School» μπορούν να συνοψιστούν στα πιο κάτω:

- Προοπτικές ομοιόμορφης αντιμετώπισης της εκπαιδευτικής λειτουργίας στο σύνολο της επικράτειας δίνοντας ίσες ευκαιρίες σε όλους, σε ό,τι αφορά στην απόσταση και στην επιλογή του χρόνου εκπαίδευσης και στο γνωστικό αντικείμενο.
- Συνεχιζόμενη κατάρτιση και εκπαίδευση για τη βελτίωση των ικανοτήτων των εκπαιδευομένων.
- Επιτάχυνση της διαδικασίας μεταφοράς γνώσης από τους εκπαιδευτές στους εκπαιδευόμενους.
- Άμεση διασύνδεση ατόμων και ομάδων με εξωτερικές πηγές γνώσεων και αυξημένες δυνατότητες συνεργασίας μεταξύ εκπαιδευομένων και εκπαιδευτών.
- Δημιουργία μιας «ηλεκτρονικής τάξης» με εξομίωση όλων των λειτουργιών μιας παραδοσιακής τάξης (παράδοση διαλέξεων, επίλυση ασκήσεων, διόρθωση ασκήσεων, υποβολή ερωτήσεων κλπ).
- Εξοικονόμηση χρόνου και εκπαιδευτικού προσωπικού.
- Αύξηση του αριθμού των εκπαιδευομένων με ταυτόχρονη μείωση των λειτουργικών αναγκών.
- Ευελιξία στο χρόνο, στο χώρο και στο ρυθμό μάθησης.
- Έλεγχος από την πλευρά του εκπαιδευόμενου για τον ρυθμό προόδου που σημειώνει κατά τη μαθησιακή διαδικασία.
- Οικονομικά οφέλη για κάθε εμπλεκόμενο, με τη δυνατότητα ελαχιστοποίησης κόστους προς απόδοση.

**ΒΙΒΛΙΟΓΡΑΦΙΑ**

- Βίρβου Μαρία, «Τεχνολογία Λογισμικού, Σημειώσεις Διδασκαλίας»
- Βίρβου Μαρία, «Αντικειμενοστρεφής Τεχνολογία Λογισμικού»
- Shirley Williams, Sue Walmsley (1999), «Μάθετε την Delphi», εκδόσεις Κλειδάριθμος
- Ian Sommerville (2001), «Software Engineering», Addison-Wesley
- Terry Quatrani (2000), «Visual Modeling with Rational Rose 2000 and UML», Addison-Wesley
- Μοίρα Πολυξένη (2005), «Τουριστική Γεωγραφία Της Ελλάδας», εκδόσεις Interbooks
- Συλλογικό έργο (2006), «Γεωγραφία της Ελλάδος - Έρευνα στην Ελλάδα CD», εκδόσεις Πατάκη
- Παυλίδου Μαριάνθη, Ασλανίδης Άρης (2003), «Γνωρίζω την Ελλάδα, class Ε' δημοτικού», εκδόσεις Πατάκη
- (2007), «Ελλάδα Γεωφυσικός Χάρτης Προστατευόμενης», εκδόσεις Ανάβαση
- Ασλανίδης, Άρης (2007), «Class ΣΤ' δημοτικού», εκδόσεις Πατάκη
- (2004), «Πολιτικός και γεωφυσικός χάρτης της Ελλάδας», εκδόσεις Κέδρος
- «Ανακαλύπτω Την Ελλάδα Και Την Κύπρο!» (2004), Πλατφόρμα PC, Εκπαιδευτικό Παιχνίδι
- «Class της Ελλάδας», CD-ROM με την Γεωγραφία της Ελλάδας, <http://schoolnet.protovoulia.org/>
- «Ο Ξεφτέρης Ταξιδεύει Στην Ελλάδα», Εκπαιδευτικό Λογισμικό Εκμάθησης CD-ROM (siem)
- <http://www.google.gr>
- <http://www.geocities.com/barousi/pages/geografia.htm>
- <http://el.wikipedia.org/>
- <http://www.athensguide.org/>
- <http://www.ccifhel.org.gr>
- <http://schoolnet.protovoulia.org/>
- [http://www.geocities.com/world\\_greek\\_geografia/](http://www.geocities.com/world_greek_geografia/)
- <http://www.worldatlas.com/>
- <http://www.culture.gr>
- <http://hellas.teipir.gr/general/ellas.htm>
- <http://www.mapgreece.gr/maps/index.php>
- <http://www.in2greece.com/mappages/greeceall.htm>
- <http://www.e-city.gr/>
- <http://www.anodos.edu.gr/>
- <http://www.meteo.gr/>
- <http://www.oramaeditions.gr/>
- <http://www.neolaia.gr/wordpress/index.php/about/xarths-ellados/>
- <http://www.united-hellas.com/map-of-greece-gr.html>
- <http://www.mapgr.gr/>
- <http://www.sunsailing.gr/p.homeGR.html>
- <http://greece-ellas.com/>
- <http://www.hellogreece.gr/>
- <http://www.youtube.com/>
- <http://www.nomosartas.gr/>
- <http://www.artino.gr>
- <http://www.artainfo.gr>
- <http://www.arta.gr>
- <http://www.about-ioannina.gr>
- [http://www.ioannina.gr/DI/index\\_gr.htm](http://www.ioannina.gr/DI/index_gr.htm)
- <http://www.ioannina24.gr/>
- <http://ioannina.uoi.gr/>
- <http://www.preveza.gr/>
- <http://www.dimosprevezas.gr/>
- <http://www.preveza-info.gr/>
- <http://www.thesprotia.gr/>
- <http://www.nthesprotias.com/>
- <http://e-thesprotia.eu/>
- <http://demosthenes.di.uoa.gr/>



- <http://www.teleteaching.gr/>
- <http://eclass.upatras.gr/>

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΠΑ

## **ΠΑΡΑΡΤΗΜΑ 1**

### **ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΤΩΝ**

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΡΑΙΑ

## Εισαγωγή

Το πρόγραμμα e-School αποτελεί μια εφαρμογή ηλεκτρονικής εκπαίδευσης με αντικείμενο το μαθήματα της Γεωγραφίας και Ιστορίας.

Ο καθηγητής έχει πλήρη έλεγχο της οργάνωσης του μαθήματος, και μπορεί να διαμορφώσει κατά βούληση την διδακτέα ύλη, χρησιμοποιώντας κείμενο, εικόνες, αλλά και πολυμέσα.

Όταν έχει καταχωριστεί η διδακτέα ύλη, ο μαθητής μπορεί να χρησιμοποιήσει το πρόγραμμα για να παρακολουθήσει τα μαθήματα και να εξεταστεί σε αυτά. Το πρόγραμμα τηρεί στατιστικά στοιχεία για τον κάθε μαθητή, στα οποία έχει πρόσβαση ο καθηγητής.

Το παρόν εγχειρίδιο δίνει οδηγίες χρήσης του προγράμματος για τον μαθητή (Α΄ μέρος), τον καθηγητή (Β΄ μέρος) και τον Administrator (Γ΄ μέρος).

## **Α΄ Μέρος**

### **Εγχειρίδιο για τον μαθητή**

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΡΑΙΑ

## 1. Αρχική οθόνη

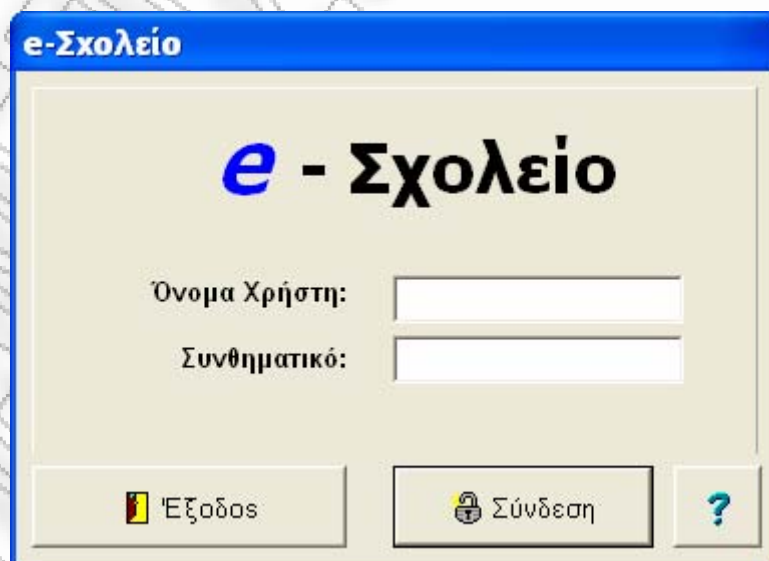
Αυτή είναι η οθόνη σύνδεσης στο πρόγραμμα, και η πρώτη που θα συναντά ο μαθητής κάθε φορά.



Ο μαθητής μπορεί να επιλέξει μεταξύ των δύο μαθημάτων που υπάρχουν αυτή τη στιγμή στην εφαρμογή. Στη συνέχεια υπάρχει η δυνατότητα προσθήκης νέων μαθημάτων και έτσι ο μαθητής θα μπορεί να επιλέξει ποιο από τα μαθήματα θα παρακολουθήσει και θα εξεταστεί.

## 2. Οθόνη σύνδεσης

Όταν επιλεγεί το μάθημα, εμφανίζεται η παρακάτω οθόνη:

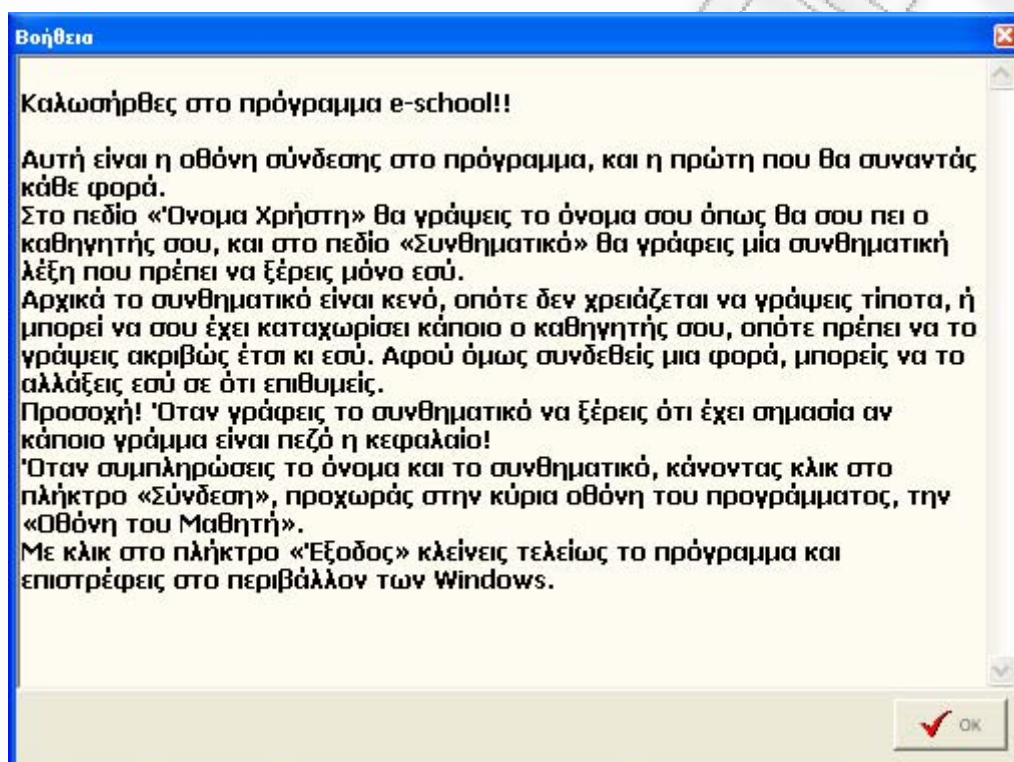


Στο πεδίο «Όνομα Χρήστη» θα αναγράφεται το όνομα του εκάστοτε χρήστη και στο πεδίο «Συνθηματικό» θα αναγράφεται μία συνθηματική λέξη που πρέπει να γνωρίζει ο κάθε

χρήστης μόνο. Στην περίπτωση των μαθητών, το «Όνομα Χρήστη» και το «Συνθηματικό» θα τα γνωρίζει και ο καθηγητής και ο administrator. Αρχικά το συνθηματικό είναι κενό, οπότε δεν χρειάζεται να αναγράφεται κάτι, ή εάν έχει καταχωρηθεί κάποιος, πρέπει να το γράφεται ακριβώς με τον ίδιο τρόπο. Έχει σημασία πως γράφεται το συνθηματικό, αν κάποιο γράμμα είναι πεζό ή κεφαλαίο.

Όταν συμπληρωθεί το όνομα και το συνθηματικό, κάνοντας κλικ στο πλήκτρο «Σύνδεση», γίνεται μετάβαση στην κύρια οθόνη του προγράμματος για τον μαθητή στην βασική «Οθόνη του Μαθητή».

Με κλικ στο κουμπί «Βοήθεια», ο χρήστης μπορεί να δει οδηγίες για τη σύνδεσή του στην εφαρμογή :



Με κλικ στο κουμπί «Έξοδος» κλείνει τελείως το πρόγραμμα και γίνεται επιστροφή στο περιβάλλον των Windows.

### 3.Βασική Οθόνη Μαθητή

#### 3.1 Μάθημα Γεωγραφίας

Ο μαθητής αφού συνδεθεί στην εφαρμογή, μεταβαίνει στην βασική «Οθόνη του Μαθητή», η οποία παρατίθεται παρακάτω:

**e-Σχολείο: Γεωγραφία - Μαθητής: TestStudent**

Καλωσήρθες!  
Το τελευταίο μάθημα που παρακολούθησες ήταν το: **Νομός Άρτας**  
Επέλεξε μάθημα από τη λίστα ή κάνοντας κλικ στον χάρτη και ύστερα κάνε κλικ στο κουμπί "Έναρξη μαθήματος" για να ξεκινήσει το μάθημα

**Νομός Άρτας**

Έναρξη μαθήματος!

Τεστ προόδου

Καρτέλα Αξιολόγησης

Αλλαγή συνηματικού

Ξεσόδος

**Νομός Άρτας**

Τελευταία παρακολούθηση: 20/11/2012  
Τελευταίο τεστ προόδου: 20/11/2012 Βαθμολογία: 10.8

### 3.2 Μάθημα Ιστορίας

**Καλωσήρθες!**  
Το τελευταίο μάθημα που παρακολούθησες ήταν το: **Προεπαναστατική περίοδος**  
Επέλεξε μάθημα από τη λίστα ή κάνοντας κλικ στον χάρτη και ύστερα κάνε κλικ στο κουμπί "Εναρξη μαθήματος" για να ξεκινήσει το μάθημα

Προεπαναστατική πε...

Εναρξη μαθήματος!

Τεστ προόδου

Καρτέλα Αξιολόγησης

Αλλαγή συνηματικού

Εξοδος

Τελευταία παρακολούθηση: 11/7/2012  
Τελευταίο τεστ προόδου: 11/7/2012 Βαθμολογία: 14.3

Προεπαναστατική περίοδος

Η έναρξη της Επανάστασης

Πρώτη φάση της Επανάστασης

Το Χρονικό του Αγώνα

Στα αριστερά θα βρίσκονται μία σειρά από πλήκτρα για την πραγματοποίηση των σημαντικότερων λειτουργιών, όπως η παρακολούθηση ή η εξέταση κάποιου μαθήματος.

Στα δεξιά και στο μεγαλύτερο κομμάτι της οθόνης εμφανίζεται η βασική εικόνα του προγράμματος. Μετακινώντας τον δείκτη του ποντικιού πάνω στον χάρτη ή στον πάπυρο αντίστοιχα υπάρχει η δυνατότητα να δει ο μαθητής για ποιες περιοχές υπάρχουν διαθέσιμα μαθήματα. Το πλαίσιο που εμφανίζεται κοντά στο δείκτη του ποντικου δείχνει πότε ο μαθητής παρακολούθησε τελευταία φορά το μάθημα, εάν βέβαια το παρακολούθησε, και αν έχει εξεταστεί στο μάθημα αυτό γράφει και πότε εξετάστηκε, καθώς και τι βαθμό πήρε στην εξέταση.

Κάνοντας κλικ με το ποντίκι σε κάποια περιοχή που διαθέτει μάθημα, υπάρχει η δυνατότητα επιλογής του μαθήματος αυτού, ώστε ο μαθητής να το παρακολουθήσει ή να εξεταστεί σε αυτό. Επίσης, υπάρχει η δυνατότητα επιλογής μαθήματος από το μενού που βρίσκεται ακριβώς πάνω από το πλήκτρο «Εναρξη μαθήματος».

Στο πλαίσιο ακριβώς κάτω από την εικόνα εμφανίζεται η ονομασία του επιλεγμένου μαθήματος, η ημερομηνία παρακολούθησης της τελευταίας φοράς, η ημερομηνία εξέτασης, καθώς και ο βαθμός κατά την εξέταση.

Αφού επιλεγεί το μάθημα από τον χάρτη ή από το μενού, ο μαθητής μπορεί να κάνει κλικ στο πλήκτρο «Εναρξη μαθήματος» για να παρακολουθήσει το συγκεκριμένο μάθημα. Τότε θα μεταφερθεί στην Οθόνη Παρουσίασης Μαθήματος.

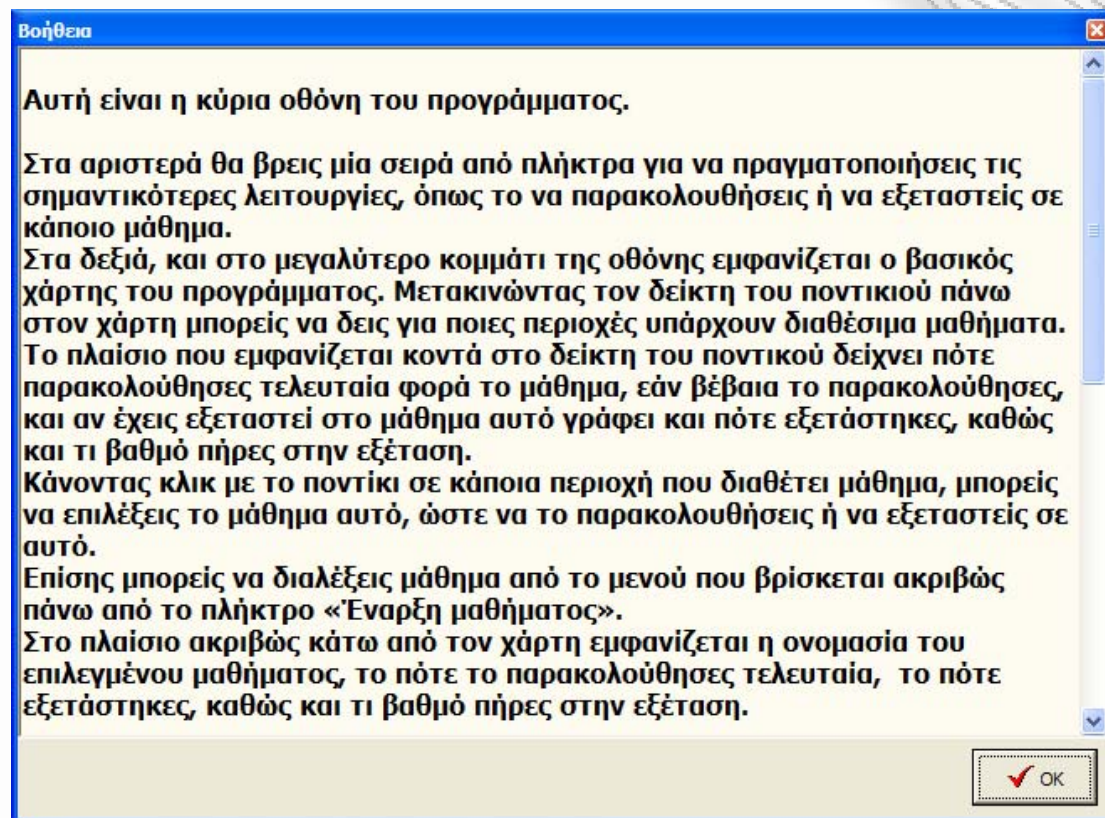
Όταν ο μαθητής παρακολουθήσει το μάθημα, μπορείς να κάνει κλικ στο πλήκτρο «Τεστ προόδου» για να εξεταστεί σε αυτό το μάθημα. Σε αυτήν την περίπτωση θα μεταφερθεί στην Οθόνη Εξέτασης.

Όταν ο μαθητής έχει παρακολουθήσει αρκετά μαθήματα, και έχει κάνει αρκετά τεστ, μπορεί να δει την αξιολόγησή του στα τεστ κάνοντας κλικ στο πλήκτρο «Καρτέλα Αξιολόγησης», οπότε θα μεταφερθεί στην αντίστοιχη οθόνη.



Από αυτή την οθόνη, επίσης, δίδεται στον μαθητή η δυνατότητα να αλλάξει το συνθηματικό του σε οποιαδήποτε λέξη ή συνδυασμό επιθυμεί. Με κλικ στο πλήκτρο «Αλλαγή Συνθηματικού», ανοίγει μία οθόνη στην οποία ο μαθητής μπορεί να πραγματοποιήσει την αλλαγή.

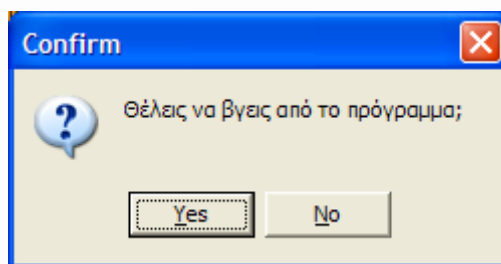
Με κλικ στο κουμπί «Βοήθεια», ο χρήστης μπορεί να δει οδηγίες για την λειτουργικότητα της κύριας οθόνης του μαθητή :



Βοήθεια υπάρχει επίσης και από τον agent που υπάρχει στη εφαρμογή. Βρίσκεται στα αριστερά της οθόνης και δίνει συμβουλές στον μαθητή. Αν κάνει κλικ πάνω του υπάρχει η βασική οθόνη της βοήθειας, όπως παρουσιάστηκε και παραπάνω. Ο agent:

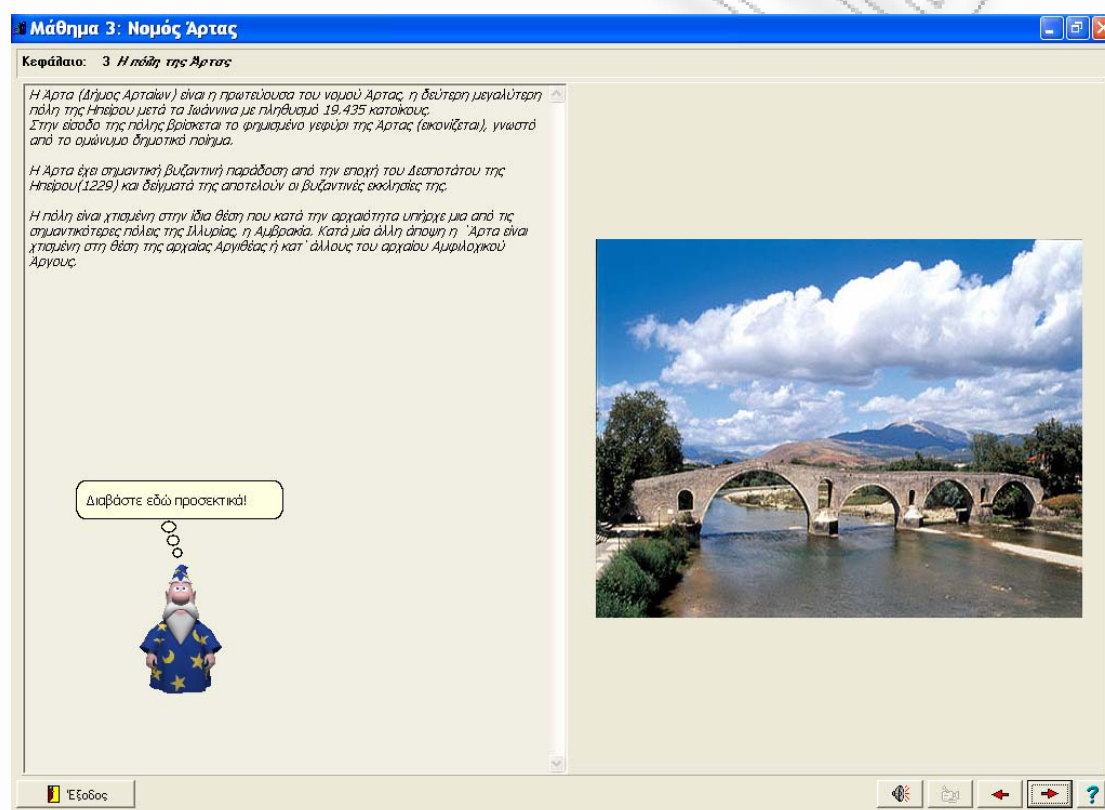


Τέλος, κάνοντας κλικ στο πλήκτρο «Έξοδος» ο μαθητής επιστρέφει στην Οθόνη Σύνδεσης, αφού πρώτα ερωτηθεί από το πρόγραμμα αν επιθυμεί να βγει :



## 4. Οθόνη Παρουσίασης Μαθήματος

### 4.1 Μάθημα Γεωγραφίας

A screenshot of a lesson presentation window titled "Μάθημα 3: Νομός Άρτας". The window contains text about the city of Arta, a cartoon character with a speech bubble saying "Διαβάστε εδώ προσεκτικά!" (Read carefully here!), and a photograph of a stone bridge over a river. The text in the window reads:

**Κεφάλαιο: 3 Η πόλη της Άρτας**

Η Άρτα (Δήμος Αρταίων) είναι η πρωτεύουσα του νομού Άρτας, η δεύτερη μεγαλύτερη πόλη της Ηπείρου μετά τα Ιωάννινα με πληθυσμό 19.435 κατοίκους. Στην είσοδο της πόλης βρίσκεται το φημισμένο γεφύρι της Άρτας (εικονίζεται), γνωστό από το ομώνυμο δημοτικό ποίημα.

Η Άρτα έχει σημαντική βυζαντινή παράδοση από την εποχή του Δεσποτάτου της Ηπείρου (1229) και δείγματά της αποτελούν οι βυζαντινές εκκλησίες της.

Η πόλη είναι χτισμένη στην ίδια θέση που κατά την αρχαιότητα υπήρχε μια από τις σημαντικότερες πόλεις της Ιλλυρίας, η Αμβρακία. Κατά μια άλλη άποψη η Άρτα είναι χτισμένη στη θέση της αρχαίας Αργινέας ή και άλλους του αρχαίου Αμφιλοχικού Άργους.

Διαβάστε εδώ προσεκτικά!

Έξοδος

## 4.2 Μάθημα Ιστορίας

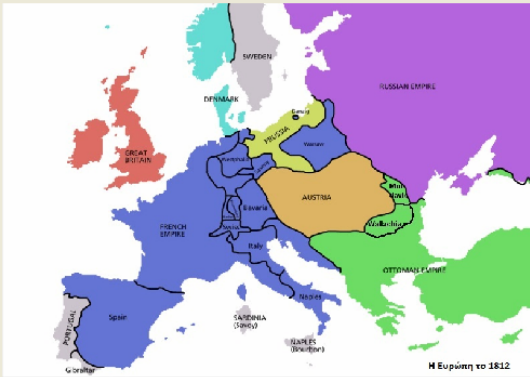
**Μάθημα 1: Προεπαναστατική περίοδος**

Κεφάλαιο: 1 *Πρώτες προσπάθειες*


Πολλές επαναστάσεις ενάντια στην οθωμανική κυριαρχία είχαν λάβει χώρα στον ελληνικό γεωγραφικό χώρο πριν από την μεγάλη επανάσταση του 1821. Άλλες από αυτές ήταν μικρότερης και άλλες μεγαλύτερης σημασίας, όλες όμως είχαν γενικά τοπικό χαρακτήρα. Η επανάσταση του 1821 ήταν η μόνη που οργανώθηκε προσηκτικά, πολλά χρόνια πριν την άρση της στην Ελλάδα (σαν Ελλάδα εννοείται την περίοδο αυτή ο ιστορικός ελληνικός γεωγραφικός χώρος).

Στον απόηχο των μεγάλων γεγονότων που συγκλόνισαν την Ευρώπη (Γαλλική Επανάσταση του 1789, Ναπολεόντιο πόλεμοι), αλλά και τον κόσμο (Αμερικανική Επανάσταση), που δεν είχαν αντίκτυπο στις οπισθοδρομικές πλέον θωμές της αχανούς Οθωμανικής Αυτοκρατορίας του Μαχμούτ Β' (1808-1839), οι Έλληνες της διασποράς που αποτελούσαν το πιο ενθουσιωμένο και προοδευτικό κομμάτι του ελληνισμού, ίδρυσαν αρχικά, το 1809, την οργάνωση Ελληνόγλωσσα Ξενοδοχείο με πρωταρχικό του Θεσσαλονίκης λογιό Γρηγορίου Ζαλίκη και της Κύπρου λογίας Ελισάβετ Σαντή Λουμάκη Σενιέ.

Η οργάνωση πέτυχε τη στήριξη του Ναπολέοντα Βοναπάρτη και συγκέντρωσε οπλισμό που διαμοιράστηκε στη Μακεδονία και την Ήπειρο προκειμένου να προετοιμαστεί η επανάσταση. Η κίνηση αυτή προδόθηκε στις αρχές της Οθωμανικής Αυτοκρατορίας και τελικά το σχέδιό της απέτυχε.



Η Ευρώπη το 1812



Εξέδος

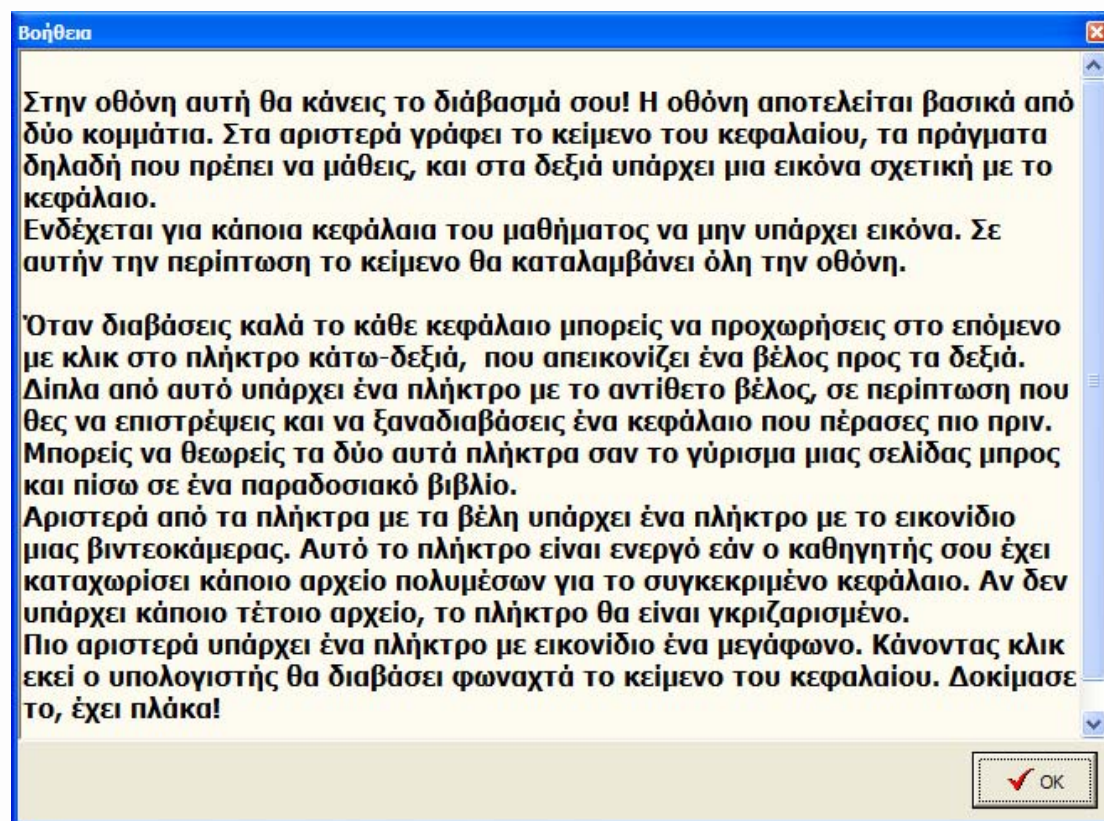
Η οθόνη αποτελείται βασικά από δύο μέρη. Στα αριστερά γράφει το κείμενο του κεφαλαίου, την ύλη του μαθήματος που έχει καθοριστεί από τον administrator και πρέπει να μάθει ο μαθητής. Στα δεξιά υπάρχει μια εικόνα σχετική με το κεφάλαιο.

Όταν ο μαθητής διαβάσει το κάθε κεφάλαιο μπορεί να προχωρήσει στο επόμενο με κλικ στο πλήκτρο κάτω-δεξιά, που απεικονίζει ένα βέλος προς τα δεξιά. Δίπλα από αυτό υπάρχει ένα πλήκτρο με το αντίθετο βέλος, σε περίπτωση που επιθυμεί να επιστρέψει και να ξαναδιαβάσει ένα κεφάλαιο που πέρασε πιο πριν. Αυτά τα δύο πλήκτρα μπορούν να θεωρηθούν σαν το γύρισμα μιας σελίδας μπρος και πίσω σε ένα παραδοσιακό βιβλίο.

Αριστερά από τα πλήκτρα με τα βέλη υπάρχει ένα πλήκτρο με το εικονίδιο μιας βιντεοκάμερας. Αυτό το πλήκτρο είναι ενεργό, εάν ο καθηγητής έχει καταχωρήσει κάποιο αρχείο πολυμέσων για το συγκεκριμένο κεφάλαιο. Αν δεν υπάρχει κάποιο τέτοιο αρχείο, το πλήκτρο θα είναι γκριζαρισμένο.

Πιο αριστερά υπάρχει ένα πλήκτρο με εικονίδιο ένα megáφωνο. Κάνοντας κλικ εκεί ο υπολογιστής θα διαβάσει φωναχτά το κείμενο του κεφαλαίου. Επίσης, όταν το κείμενο ενός κεφαλαίου είναι αρκετά μεγάλο, ο μαθητής μπορεί να χρησιμοποιήσει το scroll down.

Με κλικ στο κουμπί «Βοήθεια», ο χρήστης μπορεί να δει οδηγίες για την παρακολούθηση του μαθήματος :



Βοήθεια υπάρχει επίσης και από τον agent που υπάρχει στη εφαρμογή. Βρίσκεται στα αριστερά της οθόνης και δίνει συμβουλές στον μαθητή. Αν κάνει κλικ πάνω του υπάρχει η βασική οθόνη της βοήθειας, όπως παρουσιάστηκε και παραπάνω.


Τέλος, κάνοντας κλικ στο πλήκτρο «Έξοδος» ο μαθητής επιστρέφει στην βασική «Οθόνη του Μαθητή».


Οι παρακάτω οθόνες αποτελούν δύο από τις πολλές οθόνες που έχουν σχεδιαστεί για την εξέταση των μαθημάτων από τους μαθητές. Το κάθε τεστ αποτελείται από κάποιες ερωτήσεις που μπορεί να είναι δύο ειδών:

Ερωτήσεις στον χάρτη:

**Ερωτήσεις πάνω στο μάθημα 3: Νομός Άρτας**

Ερώτηση: 1



Ολοκλήρωση τεστ: 

Χρόνος που απομένει: 

Μένουν 12 ερωτήσεις

Μένουν 4 λεπτά

*Δείξε τον Νομό Άρτας στον χάρτη της Ελλάδας*

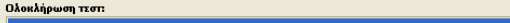
Έξοδος

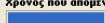
Σάββατο, 24 Νοεμβρίου 2012

και ερωτήσεις πολλαπλής επιλογής:

**Ερωτήσεις πάνω στο μάθημα 3: Νομός Άρτας**

Ερώτηση: 8

Ολοκλήρωση τεστ: 

Χρόνος που απομένει: 

Μένουν 5 ερωτήσεις

Μένουν 4 λεπτά

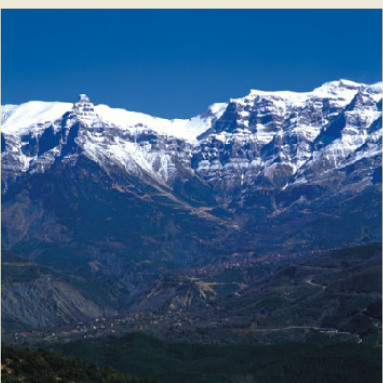

*Ποιο είναι το ψηλότερο βουνό του νομού Άρτας;*

**A** Ίδη (Ψηλορείτης)

**B** Τιμφριστός (Βελούχι)

**Γ** Αθαμανικά Όρη (Τζουμέρκα)

**Δ** Οίτη

Έξοδος

Στις ερωτήσεις στον χάρτη, ζητείται από τον μαθητή να κάνει κλικ σε μια συγκεκριμένη περιοχή του χάρτη. Αν επιλέξει το σωστό σημείο, το τεστ προχωράει στην επόμενη ερώτηση. Εάν κάνει κλικ σε λάθος σημείο, εμφανίζεται ένα μήνυμα που λέει αν είναι κοντά ή μακριά, και αν το έχει επιτρέψει ο καθηγητής, ο μαθητής μπορεί να ξαναδοκιμάσει.

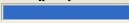
The screenshot shows a web browser window titled "Ερωτήσεις πάνω στο μάθημα 3: Νομός Άρτας". The interface includes a progress bar for "Ολοκλήρωση τεστ" and "Χρόνος που απομένει", and a timer for "Μένουν 12 ερωτήσεις" and "Μένουν 4 λεπτά". The question text is "Δείξε τον Νομό Άρτας στον χάρτη της Ελλάδας". Below the text is a map of Greece with various regions highlighted in different colors. A red dot is placed on the map, and a dialog box titled "e-Σχολείο" appears with the message: "Είσαι κοντά, αλλά όχι αρκετά! Έχεις άλλες 2 προσπάθειες" and an "OK" button. At the bottom left of the map area is a cartoon character of a wizard. The Windows taskbar at the bottom shows the "έναρξη" (Start) button and several open applications.

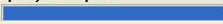
Αν κάνει ξανά λάθος η απάντηση θα θεωρηθεί τελικά λανθασμένη και το τεστ θα προχωρήσει στην επόμενη ερώτηση ή θα δοθεί νέα ευκαιρία από τον καθηγητή.

Στις ερωτήσεις πολλαπλής επιλογής, θα εμφανιστούν από δύο έως τέσσερα πλήκτρα απάντησης και ο μαθητής πρέπει να κάνει κλικ σε αυτό που πιστεύει ότι είναι η σωστή απάντηση. Αν επιλέξει την σωστή απάντηση, το τεστ θα προχωρήσει. Αν κάνει λάθος, το πλήκτρο που επέλεξε θα πάρει γκριζό χρώμα και θα μπορεί να ξαναδοκιμάσει να δώσει κάποια από τις άλλες διαθέσιμες απαντήσεις. Αν η ερώτηση αφορά τοποθεσία, το πρόγραμμα θα κάνει ένα σχόλιο στην περίπτωση λάθους, ανάλογα με το πόσο κοντά ή μακριά είναι η τοποθεσία που επέλεξε ο μαθητής από αυτήν που ζητάει η ερώτηση. Στην παρακάτω περίπτωση που ο μαθητής επέλεξε λάθος απάντηση, εμφανίστηκε το μήνυμα «Λάθος! Πολύ μακριά»:

**Ερωτήσεις πάνω στο μάθημα 3: Νομός Άρτας**

Ερώτηση: 2

Ολοκλήρωση τεστ: 

Χρόνος που απομένει: 

Μένουν 11 ερωτήσεις

Μένουν 3 λεπτά


Σε ποιο γεωγραφικό διαμέρισμα ανήκει ο νομός Άρτας;

**A** Πελοπόννησος

**B** Ήπειρος

**Γ** Νησιά Ιονίου Πελάγους

**e-Σχολείο**  
Λάθος! Πολύ μακριά!  
OK



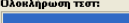
Έξοδος

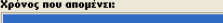
έναρξη documentation\_ne... eSchool e-Σχολείο 4:01 μμ

και γίνεται γκρι η λανθασμένη απάντηση που επιλέχθηκε :

**Ερωτήσεις πάνω στο μάθημα 3: Νομός Άρτας**

Ερώτηση: 2

Ολοκλήρωση τεστ: 

Χρόνος που απομένει: 

Μένουν 11 ερωτήσεις


Μένουν 3 λεπτά


Σε ποιο γεωγραφικό διαμέρισμα ανήκει ο νομός Άρτας;

**A** Πελοπόννησος

**B** Ήπειρος

**Γ** Νησιά Ιονίου Πελάγους

 Κρήτη



Έξοδος

έναρξη documentation\_ne... eSchool e-Σχολείο 4:02 μμ

Επίσης, στο πάνω μέρος της οθόνης αναγράφεται ο αύξων αριθμός της κάθε ερώτησης, καθώς και πόσες ερωτήσεις απομένουν για την ολοκλήρωση του τεστ και πόσος χρόνος απομένει. Εάν ο μαθητής δεν μπορέσει να ολοκληρώσει το τεστ στον προβλεπόμενο χρόνο, τότε εμφανίζεται το παρακάτω μήνυμα :

The screenshot shows a web-based test interface. At the top, it says "Ερωτήσεις πάνω στο μάθημα 3: Νομός Άρτας". Below this, there are progress bars for "Ολοκλήρωση τεστ" (10 questions remaining) and "Χρόνος που απομένει" (0 seconds remaining). The question is: "Πάνω σε ποιον ποταμό είναι κτισμένο το γεφύρι της Άρτας;". Two options are visible: "A Χαλκίδα" and "B Ηγουμενίτσα". An "Error" dialog box is overlaid on the screen, stating: "Δυστυχώς δεν κατάφερες να περάσεις το τεστ. Απάντησες σωστά τις 1 από τις 13 ερωτήσεις. Η βαθμολογία σου είναι: 1,5 ενώ για να περάσεις χρειάζεσαι: 10,0. Διάβασε πάλι το μάθημα και ξαναδοκίμασε!". The interface also includes a "Έξοδος" button and a Windows taskbar at the bottom with the "έναρξη" logo.

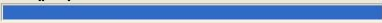
Επίσης, υπάρχουν ερωτήσεις πολλαπλής επιλογής όπου ο μαθητής πρέπει να κάνει κλικ σε αυτό που πιστεύει ότι είναι η σωστή απάντηση. Αν επιλέξει την σωστή απάντηση, το τεστ θα προχωρήσει. Αν κάνει λάθος, το πλήκτρο που επέλεξε θα πάρει γκριζό χρώμα και θα μπορεί να ξαναδοκιμάσει να δώσει κάποιες από τις άλλες διαθέσιμες απαντήσεις μία ή και παραπάνω φορές.

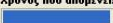
Αν η ομοιότητα την λανθασμένης επιλογής είναι άνω του ποσοστού επί τοις εκατό που θα ορίσει ο καθηγητής, τότε το πρόγραμμα θα κάνει ένα σχόλιο, ανάλογα με το πόσο όμοιες ή όχι είναι οι απάντηση που επέλεξε ο μαθητής από αυτήν που ζητάει η ερώτηση. Στην παρακάτω περίπτωση που ο μαθητής επέλεξε λάθος απάντηση, εμφανίστηκε το μήνυμα «Περίπου! Αλλά όχι ακριβώς!»:



Ερωτήσεις πάνω στο μάθημα 5: Νομός Πρεβέζης

Ερώτηση: 6

Ολοκλήρωση τεστ:  Μένουν 7 ερωτήσεις

Χρόνος που απομένει:  Μένουν 4 λεπτά


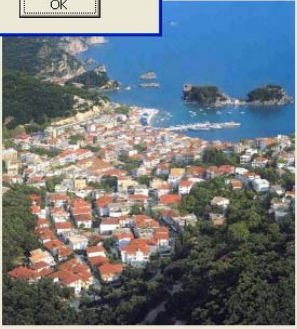
Πώς λέγεται η εικονιζόμενη παραθαλάσσια πόλη του νομού Πρεβέζης;

A Πάργα

B Πράγα

Γ Ελασσόνα

**e-Σχολείο**  
Περίπου! Αλλά όχι ακριβώς!  
OK


Έξοδος

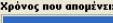
έναρξη documentation\_ne... eSchool e-Σχολείο 4:04 μμ

Αν όμως επέλεγε μία απάντηση με κανένα βαθμό ομοιότητας ή κάτω από το ποσοστό που έχει οριστεί, τότε θα εφαρμοζόταν ο επόμενος μηχανισμός με την απόσταση. Στο παρακάτω παράδειγμα το μήνυμα λάθους είναι «Λάθος! Πολύ μακριά»:

Ερωτήσεις πάνω στο μάθημα 5: Νομός Πρεβέζης

Ερώτηση: 6

Ολοκλήρωση τεστ:  Μένουν 7 ερωτήσεις

Χρόνος που απομένει:  Μένουν 3 λεπτά

Πώς λέγεται η εικονιζόμενη παραθαλάσσια πόλη του νομού Πρεβέζης;


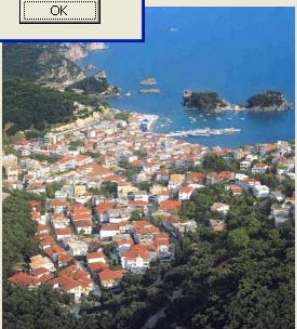
A Πάργα

B Πράγα

Γ Ελασσόνα

Δ Βα

**e-Σχολείο**  
Λάθος! Πολύ μακριά!  
OK

Έξοδος

έναρξη documentation\_ne... eSchool e-Σχολείο 4:04 μμ

Αν ο μαθητής από απροσεξία απαντήσει λανθασμένα, αλλά η απάντηση είναι ορθογραφικά κοντά στην σωστή απάντηση, εμφανίζεται το παρακάτω μήνυμα λάθους:

Ερώτηση: 3

Ολοκλήρωση τεστ:                      Μένουν 10 ερωτήσεις

Χρόνος που απομένει:                      Μένουν 4 λεπτά

**Ποια είναι η πρωτεύουσα του νομού Άρτας;**

Α	Άτρα
Β	Άρτη
Γ	Άρτος
Δ	Άρτα

Geography

Περίπου! Αλλά όχι ακριβώς!


OK

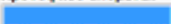
Εξοδος ?

Η ερώτηση είναι «Ποια είναι η πρωτεύουσα του νομού Άρτας?». Οι πιθανές απαντήσεις είναι «Άτρα», «Άρτη», «Άρτος» και η σωστή απάντηση «Άρτα». Όλες οι απαντήσεις είναι όμοιες με την σωστή απάντηση. Αν ο μαθητής επιλέξει μία εκ των λανθασμένων απαντήσεων, εμφανίζεται το μήνυμα λάθους «Περίπου! Αλλά όχι ακριβώς!»

Αν ο μαθητής από απροσεξία απαντήσει λανθασμένα, αλλά η απάντηση είναι ορθογραφικά κοντά στην σωστή απάντηση, αλλά εκτός Ελλάδας, τότε εμφανίζεται το παρακάτω μήνυμα λάθους:

Ερώτηση: 6


Ολοκλήρωση τεστ:  Μένουν 7 ερωτήσεις


Χρόνος που απομένει:  Μένουν 3 λεπτά


**Πώς λέγεται η εικονιζόμενη παραθαλάσσια πόλη του νομού Πρεβέζης;**

**A** Πράγα

**B** Πάργα







Εξοδος 

Οι παράμετροι του χρόνου και του πλήθους των ερωτήσεων μπορούν να αλλάξουν και καθορίζονται από τον administrator.

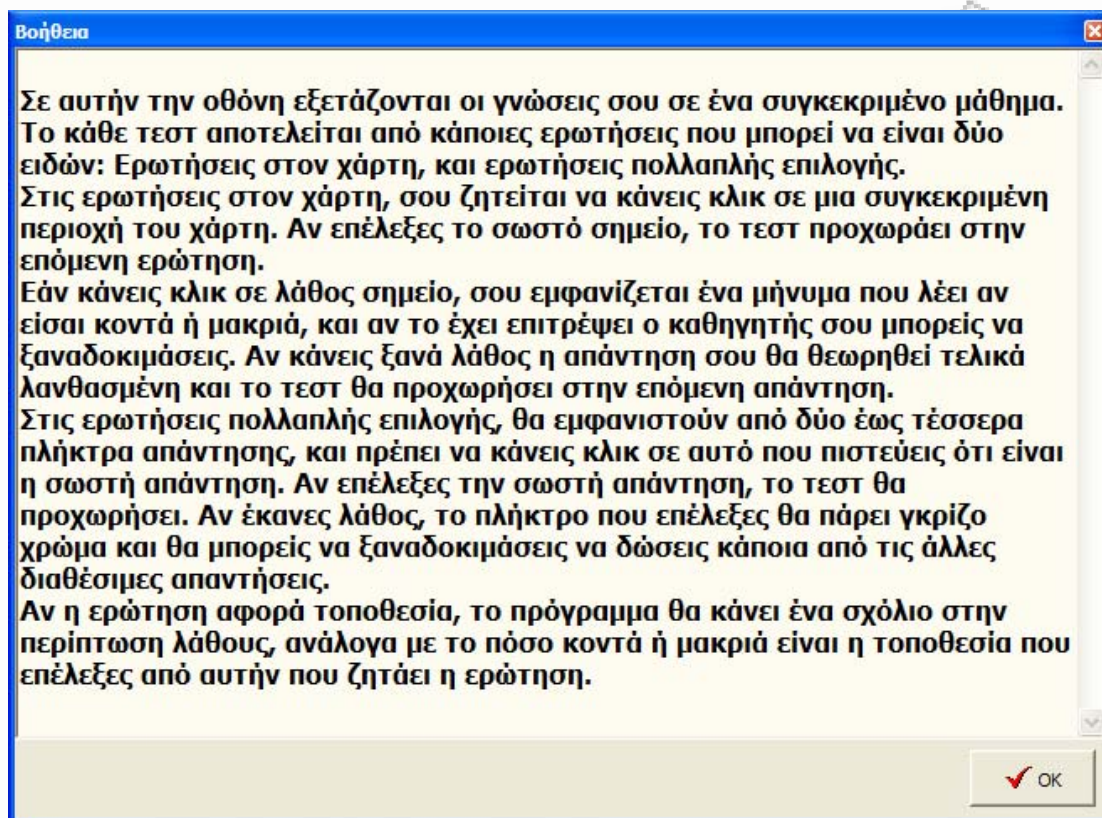
Ακόμη, ολοκληρώνοντας το τεστ ο μαθητής λαμβάνει πληροφορίες σχετικά με την απόδοσή του κατά την εξέταση :

**Information** 

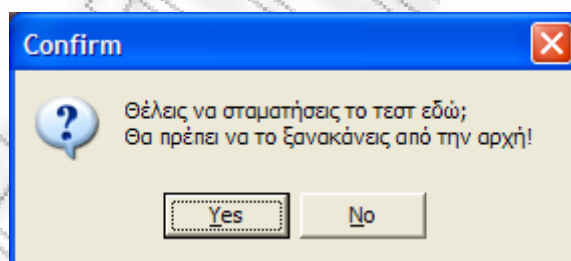
 Συγχαρητήρια! Πέρασες το τεστ με επιτυχία!  
Απάντησες σωστά τις 9 από τις 13 ερωτήσεις.  
Η βαθμολογία σου είναι: 13.8

Μπορείς να προχωρήσεις στο επόμενο μάθημα!

Με κλικ στο κουμπί «Βοήθεια», ο χρήστης μπορεί να δει οδηγίες για την εξέταση του τεστ :



Τέλος, κάνοντας κλικ στο πλήκτρο «Έξοδος» ο μαθητής επιστρέφει στην βασική «Οθόνη του Μαθητή». Πριν όμως επιστρέψει βγαίνει το παρακάτω μήνυμα:



Και ο μαθητής επιλέγει αν θέλει να συνεχίσει το τεστ ή να το επαναλάβει κάποια άλλη φορά.

Στην παρακάτω οθόνη, ο μαθητής μπορεί να αλλάξει το συνθηματικό που χρησιμοποιεί για να συνδεθεί στο πρόγραμμα. Η οθόνη αυτή εμφανίζεται, όταν ο μαθητής κάνει κλικ στο πλήκτρο «Αλλαγή Συνθηματικού» στην βασική «Οθόνη του Μαθητή» :

**Αλλαγή συνθηματικού**

Γράψτε το παλιό σας συνθηματικό στο επάνω πλαίσιο, και το νέο συνθηματικό και στα δύο κάτω πλαίσια.

Παλιό συνθηματικό:

Νέο συνθηματικό:

Επανάληψη:

**ΠΡΟΣΟΧΗ: Για το συνθηματικό έχει σημασία αν γράφετε με κεφαλαία ή μικρά!**

? OK Άκυρο

Στο πάνω πλαίσιο ο μαθητής γράφει το παλιό του συνθηματικό, και στα δύο κάτω πλαίσια (και στα δύο) γράφει το καινούριο που επιθυμεί να έχει. Κάνοντας κλικ στο «OK» αποθηκεύεται το νέο συνθηματικό. Αν ο μαθητής αλλάξει γνώμη και θέλει να κρατήσει το παλιό συνθηματικό, κάνει κλικ στο «Άκυρο».

Με κλικ στο κουμπί «Βοήθεια», ο χρήστης μπορεί να δει οδηγίες για την αλλαγή του συνθηματικού του :

**Βοήθεια**

**Σε αυτήν την οθόνη μπορείς να αλλάξεις το συνθηματικό που χρησιμοποιείς για να συνδεθείς στο πρόγραμμα. Στο επάνω πλαίσιο γράφεις το παλιό συνθηματικό, και στα δύο κάτω πλαίσια (και στα δύο!) γράφεις το καινούριο που θέλεις να έχεις.**

**Όταν είσαι σίγουρος ότι έγραψες το συνθηματικό που θέλεις, κάνε κλικ στο «OK». Αν άλλαξες γνώμη και θες να κρατήσεις το παλιό συνθηματικό, κάνε κλικ στο «Άκυρο».**

OK

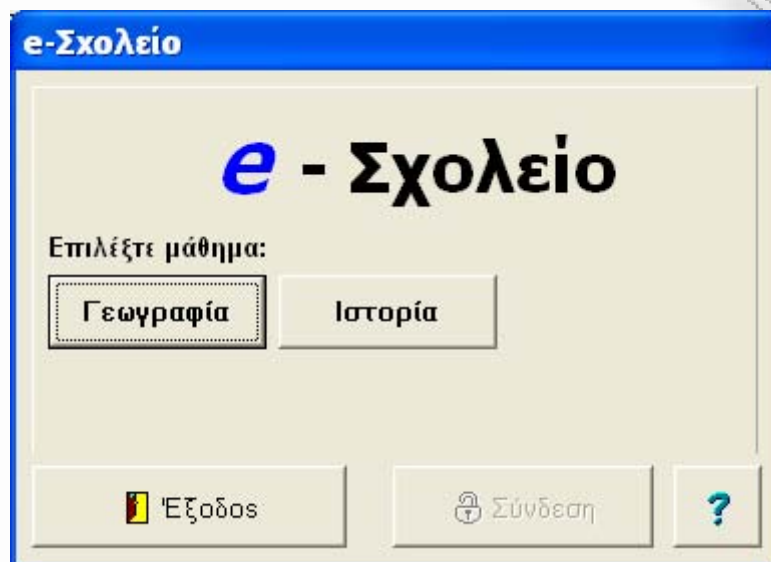
## **Β' Μέρος**

### **Εγχειρίδιο για τον καθηγητή**

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΡΑΙΑ

## 1. Αρχική οθόνη

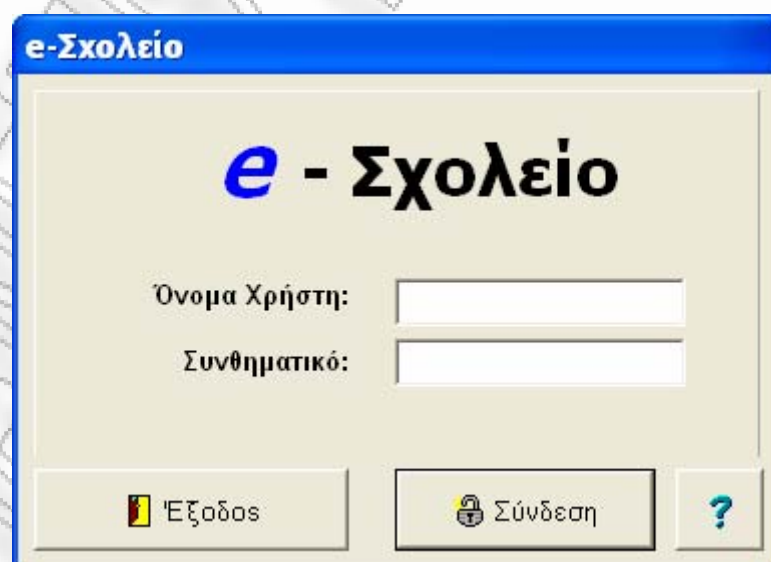
Αυτή είναι η οθόνη σύνδεσης στο πρόγραμμα, και η πρώτη που θα συναντά ο καθηγητής κάθε φορά.



Ο καθηγητής μπορεί να επιλέξει μεταξύ των δύο μαθημάτων που υπάρχουν αυτή τη στιγμή στην εφαρμογή. Στη συνέχεια υπάρχει η δυνατότητα προσθήκης νέων μαθημάτων.

## 2. Οθόνη σύνδεσης

Όταν επιλεγεί το μάθημα, εμφανίζεται η παρακάτω οθόνη:

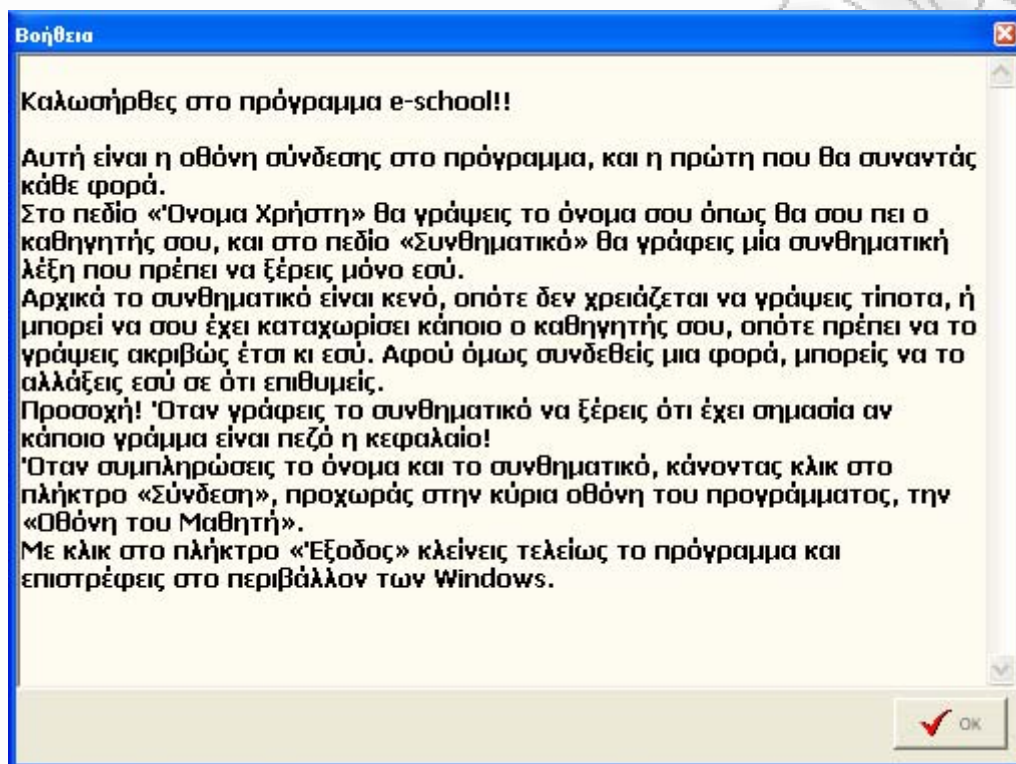


Στο πεδίο «Όνομα Χρήστη» θα αναγράφεται το όνομα του εκάστοτε χρήστη και στο πεδίο «Συνθηματικό» θα αναγράφεται μία συνθηματική λέξη που πρέπει να γνωρίζει ο κάθε χρήστης μόνο. Αρχικά το συνθηματικό είναι κενό, οπότε δεν χρειάζεται να αναγράφεται κάτι, ή

εάν έχει καταχωρηθεί κάποιος, πρέπει να το γράφεται ακριβώς με τον ίδιο τρόπο. Έχει σημασία πως γράφεται το συνθηματικό, αν κάποιος γράμμα είναι πεζό η κεφαλαίο.

Όταν συμπληρωθεί το όνομα και το συνθηματικό, κάνοντας κλικ στο πλήκτρο «Σύνδεση», γίνεται μετάβαση στην κύρια οθόνη «Οθόνη του Καθηγητή».

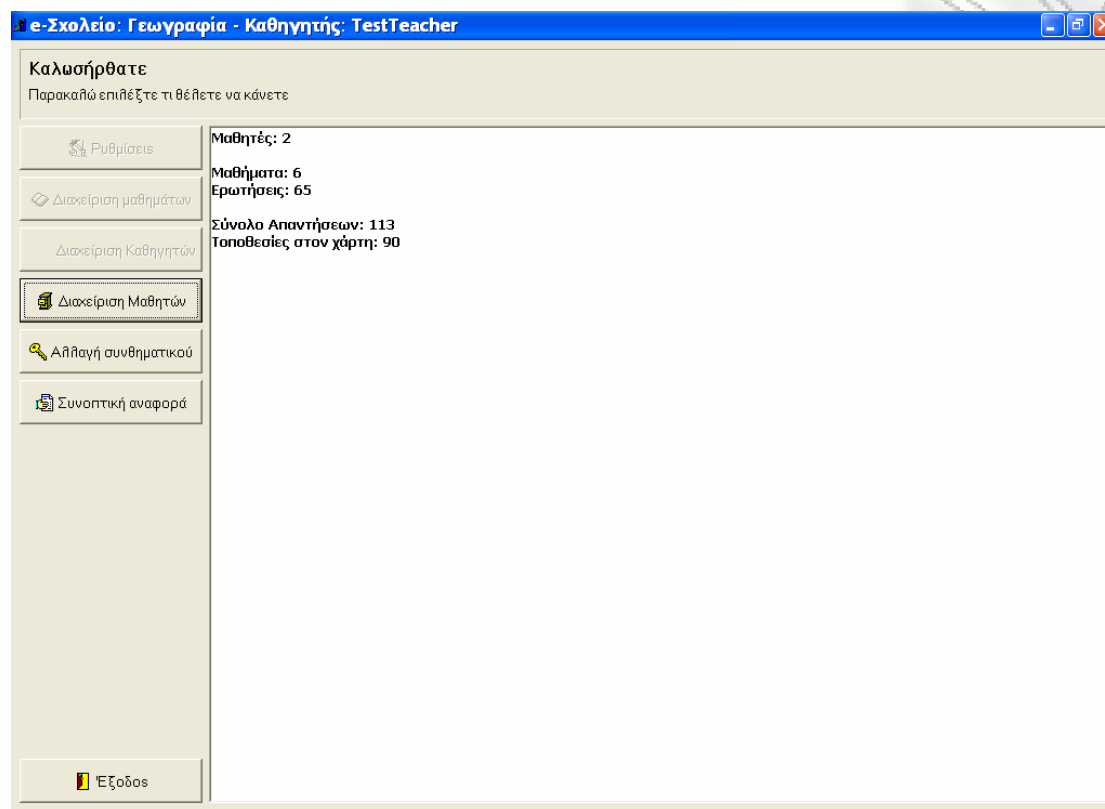
Με κλικ στο κουμπί «Βοήθεια», ο χρήστης μπορεί να δει οδηγίες για τη σύνδεσή του στην εφαρμογή :



Με κλικ στο κουμπί «Έξοδος» κλείνει τελείως το πρόγραμμα και γίνεται επιστροφή στο περιβάλλον των Windows.



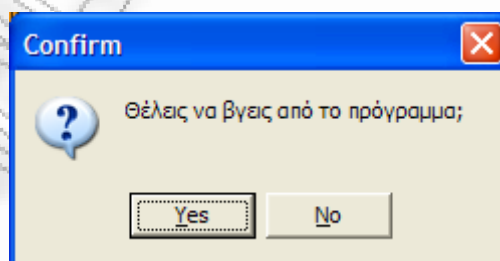
### 3.Βασική Οθόνη Καθηγητή



Στα αριστερά θα βρίσκονται μία σειρά από πλήκτρα για την πραγματοποίηση των σημαντικότερων λειτουργιών, όπως η διαχείριση των μαθητών, η αλλαγή του συνθηματικού και η συνοπτική αναφορά για την πρόοδο των μαθητών.

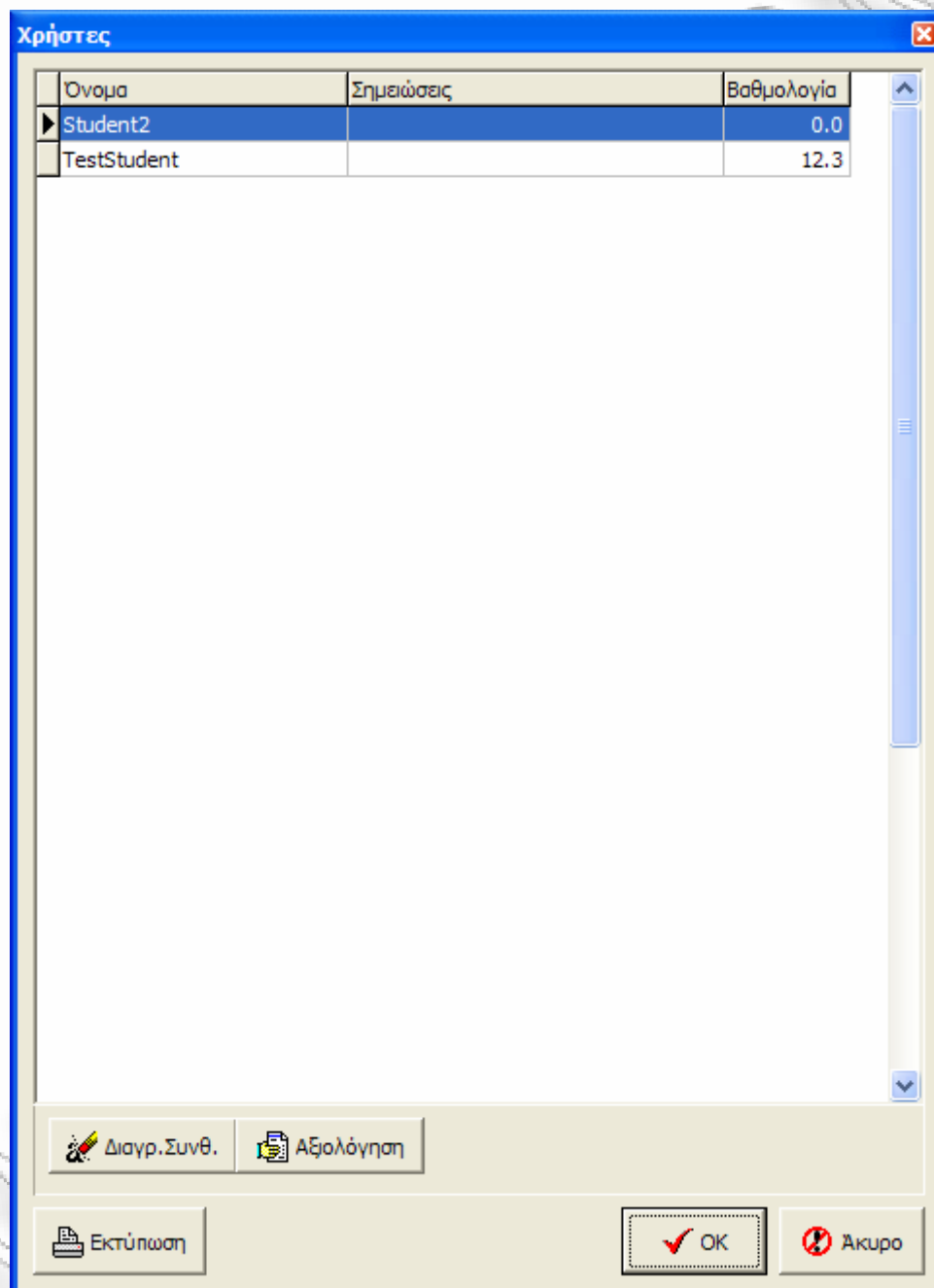
Στα δεξιά και στο μεγαλύτερο κομμάτι της οθόνης εμφανίζεται ένα grid με στοιχεία για τους μαθητές και τα μαθήματα. Ο καθηγητής μπορεί να δει τον αριθμό των μαθητών που παρακολουθούν το μάθημα, τον αριθμό των διαθέσιμων μαθημάτων, τον αριθμό των ερωτήσεων που συμμετέχουν στα εξεταζόμενα τεστ, το συνολικό πλήθος των απαντήσεων και τον αριθμό των τοποθεσιών που βρίσκονται στον χάρτη.

Τέλος, κάνοντας κλικ στο πλήκτρο «Έξοδος» ο καθηγητής επιστρέφει στην Οθόνη Σύνδεσης, αφού πρώτα ερωτηθεί από το πρόγραμμα αν επιθυμεί να βγει :



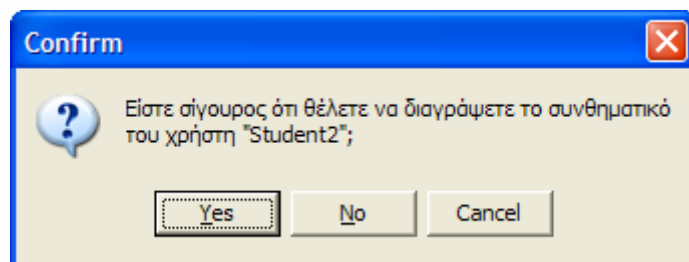
#### 4. Οθόνη Επεξεργασίας Μαθήματος

Ο καθηγητής έχει τη δυνατότητα να «διαχειριστεί» τους μαθητές που παρακολουθούν τα μαθήματα και εξετάζονται στα τεστ. Η οθόνη αυτή εμφανίζεται, όταν ο καθηγητής κάνει κλικ στο πλήκτρο «Διαχείριση Μαθητών» στην βασική «Οθόνη του Καθηγητή» :



Στο πάνω μέρος της οθόνης εμφανίζονται οι ενεργοί μαθητές. Ο καθηγητής έχει τη δυνατότητα να δει τα ονόματα των μαθητών, τυχόν σημειώσεις που έχει κρατήσει και τη συνολική τους βαθμολογία.

Στο κάτω μέρος της οθόνης υπάρχει το κουμπί «Διαγραφή Συνθηματικού», με το οποίο ο καθηγητής μπορεί να διαγράψει το συνθηματικό κάποιου μαθητή, εφόσον τον έχει επιλέξει. Το πρόγραμμα ζητά επιβεβαίωση για την διαγραφή :



Στο κάτω μέρος της οθόνης υπάρχει, επίσης το κουμπί «Αξιολόγηση». Πατώντας το ο καθηγητής, αφού έχει επιλέξει κάποιον μαθητή, βλέπει την καρτέλα αξιολόγησης του :

**Αξιολόγηση Μαθητή: TestStudent**

<b>Γενική Βαθμολογία:</b>	<b>12.3</b>
Μαθήματα που παρακολούθησε:	<b>3</b>
Ερωτήσεις που εξετάστηκε:	<b>55</b>
Σωστές απαντήσεις:	<b>42</b>

Επιδόσεις ανά κατηγορία ερωτήσεων:

Καλύτερη Κατηγορία:  
ΣΥΝΟΡΑ ΝΟΜΩΝ (100% Επιτυχία)

Χειρότερη Κατηγορία:  
ΠΡΩΤ.ΝΟΜΩΝ (50% Επιτυχία)

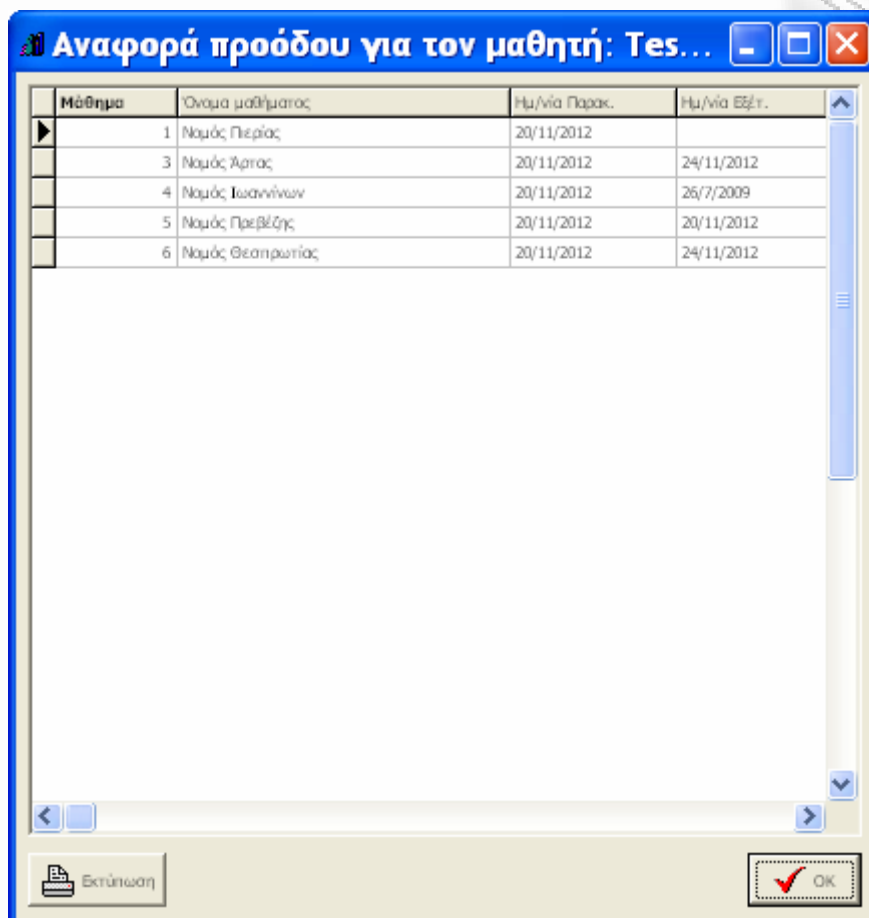
? Αναλυτικά... OK

Στην οθόνη αυτή εμφανίζεται η Καρτέλα Αξιολόγησης, όπου ο καθηγητής μπορεί να δει τα εξής στοιχεία για την πρόοδο του μαθητή:

- **Γενική Βαθμολογία:** Ο μέσος όρος της βαθμολογίας μέχρι τώρα, ανάλογα με το ποσοστό των σωστών απαντήσεων που έδωσε στα τεστ
- **Ερωτήσεις που εξετάστηκε:** Ο συνολικός αριθμός των ερωτήσεων στα διάφορα τεστ που έχει απαντήσει
- **Σωστές Απαντήσεις:** Πόσες από τις απαντήσεις που έδωσε στις παραπάνω ερωτήσεις ήταν σωστές.

Μέσα στο πλαίσιο της οθόνης, ο καθηγητής έχει τη δυνατότητα να δει σε ποια κατηγορία ερωτήσεων ο μαθητής έχει την καλύτερη απόδοση και το ποσοστό επιτυχίας που έχει σε αυτές τις ερωτήσεις. Ακριβώς από κάτω μπορεί να δει σε ποια κατηγορία είχε τη χαμηλότερη βαθμολογία, δηλαδή το χαμηλότερο ποσοστό σωστών απαντήσεων.

Εάν το επιθυμεί, ο καθηγητής μπορεί να κάνει κλικ στο κουμπί «Αναλυτικά» και θα δει αναλυτικά τους τίτλους των μαθημάτων που παρακολούθησε ο μαθητής, την ημερομηνία παρακολούθησης κάθε μαθήματος, την ημερομηνία εξέτασης κάθε τεστ και τη βαθμολογία που έλαβε σε κάθε εξέταση :



Μάθημα	Όνομα μαθήματος	Ημ/νία Παρακ.	Ημ/νία Εξέτ.
1	Ναυός Περίας	20/11/2012	
3	Ναυός Άστιας	20/11/2012	24/11/2012
4	Ναυός Ιωαννίνων	20/11/2012	26/7/2009
5	Ναυός Πρεβέζης	20/11/2012	20/11/2012
6	Ναυός Θεσσαλονίκης	20/11/2012	24/11/2012

Τόσο στην οθόνη που εμφανίζεται, όταν ο καθηγητής κάνει κλικ στο πλήκτρο «Διαχείριση Μαθητών» στην βασική «Οθόνη του Καθηγητή», όσο και στην οθόνη «Αναφορά προόδου για τον μαθητή» υπάρχει η επιλογή εκτύπωσης.

Επιλέγοντας τη συνοπτική αναφορά, ο καθηγητής μπορεί να δει σε κάθε μάθημα το μέσο όρο βαθμολογίας σε όλα τα τεστ που έκανε ο μαθητής.

The screenshot shows a window titled "StudentSummaryForm" with a table containing the following data:

Μαθητής	Γεωγραφία	Ιστορία
TestStudent	12,4 (1 μαθήματα)	8,6 (2 μαθήματα)

At the bottom of the window, there are two buttons: "Εκτύπωση" (Print) and "OK".

Στην παρακάτω οθόνη, ο καθηγητής μπορεί να αλλάξει το συνθηματικό που χρησιμοποιεί για να συνδεθεί στο πρόγραμμα. Η οθόνη αυτή εμφανίζεται, όταν ο καθηγητής κάνει κλικ στο πλήκτρο «Αλλαγή Συνθηματικού» στην βασική «Οθόνη του Καθηγητή» :

The screenshot shows a dialog box titled "Αλλαγή συνθηματικού" (Change Password) with the following text and input fields:

Γράψτε το παλιό σας συνθηματικό στο επάνω πλαίσιο, και το νέο συνθηματικό και στα δύο κάτω πλαίσια.

Παλιό συνθηματικό:

Νέο συνθηματικό:

Επανάληψη:

**ΠΡΟΣΟΧΗ: Για το συνθηματικό έχει σημασία αν γράφετε με κεφαλαία ή μικρά!**

At the bottom, there are three buttons: a help button (question mark), "OK", and "Άκυρο" (Cancel).

Στο πάνω πλαίσιο ο καθηγητή γράφει το παλιό του συνθηματικό, και στα δύο κάτω πλαίσια (και στα δύο) γράφει το καινούριο που επιθυμεί να έχει. Κάνοντας κλικ στο «OK»

αποθηκεύεται το νέο συνθηματικό. Αν ο καθηγητής αλλάξει γνώμη και θέλει να κρατήσει το παλιό συνθηματικό, κάνει κλικ στο «Άκυρο».

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΡΑΙΑ

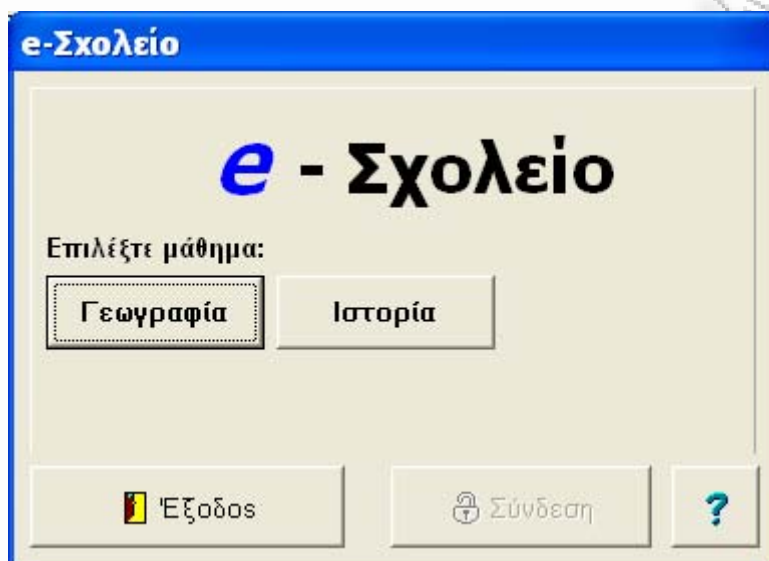
## **Γ Μέρος**

### **Εγχειρίδιο για τον Administrator**

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΡΑΙΑ

## 1. Αρχική οθόνη

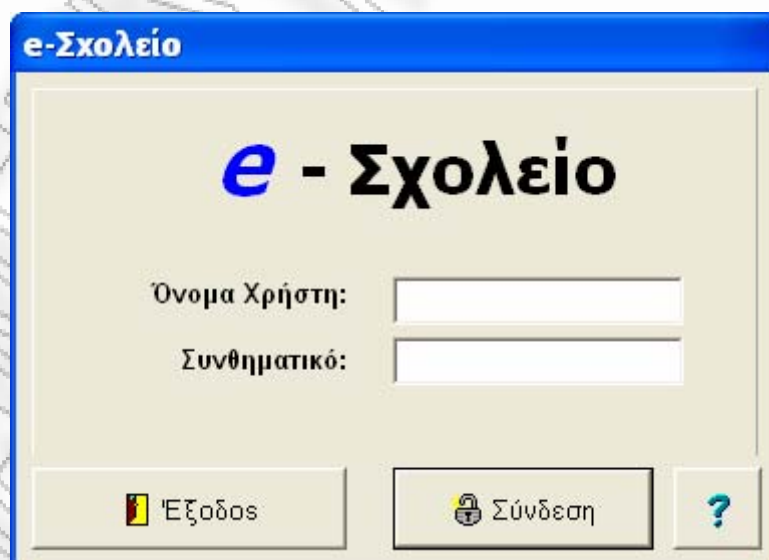
Αυτή είναι η οθόνη σύνδεσης στο πρόγραμμα, και η πρώτη που θα συναντά ο Administrator κάθε φορά.



Ο Administrator μπορεί να επιλέξει μεταξύ των δύο μαθημάτων που υπάρχουν αυτή τη στιγμή στην εφαρμογή. Στη συνέχεια υπάρχει η δυνατότητα προσθήκης νέων μαθημάτων.

## 2. Οθόνη σύνδεσης

Όταν επιλεγεί το μάθημα, εμφανίζεται η παρακάτω οθόνη:

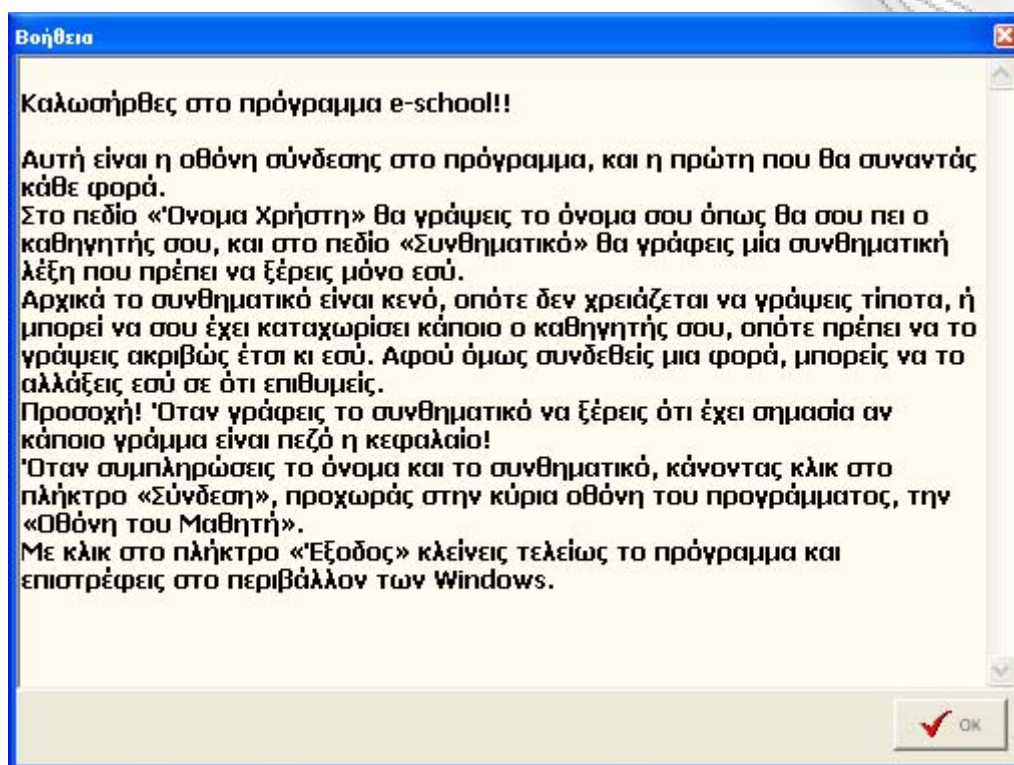


Στο πεδίο «Όνομα Χρήστη» θα αναγράφεται το όνομα του εκάστοτε χρήστη και στο πεδίο «Συνθηματικό» θα αναγράφεται μία συνθηματική λέξη που πρέπει να γνωρίζει ο κάθε χρήστης μόνο. Έχει σημασία πως γράφεται το συνθηματικό, αν κάποιο γράμμα είναι πεζό η κεφαλαίο.



Όταν συμπληρωθεί το όνομα και το συνθηματικό, κάνοντας κλικ στο πλήκτρο «Σύνδεση», γίνεται μετάβαση στην κύρια οθόνη του προγράμματος για τον μαθητή στην βασική «Οθόνη του Administrator».

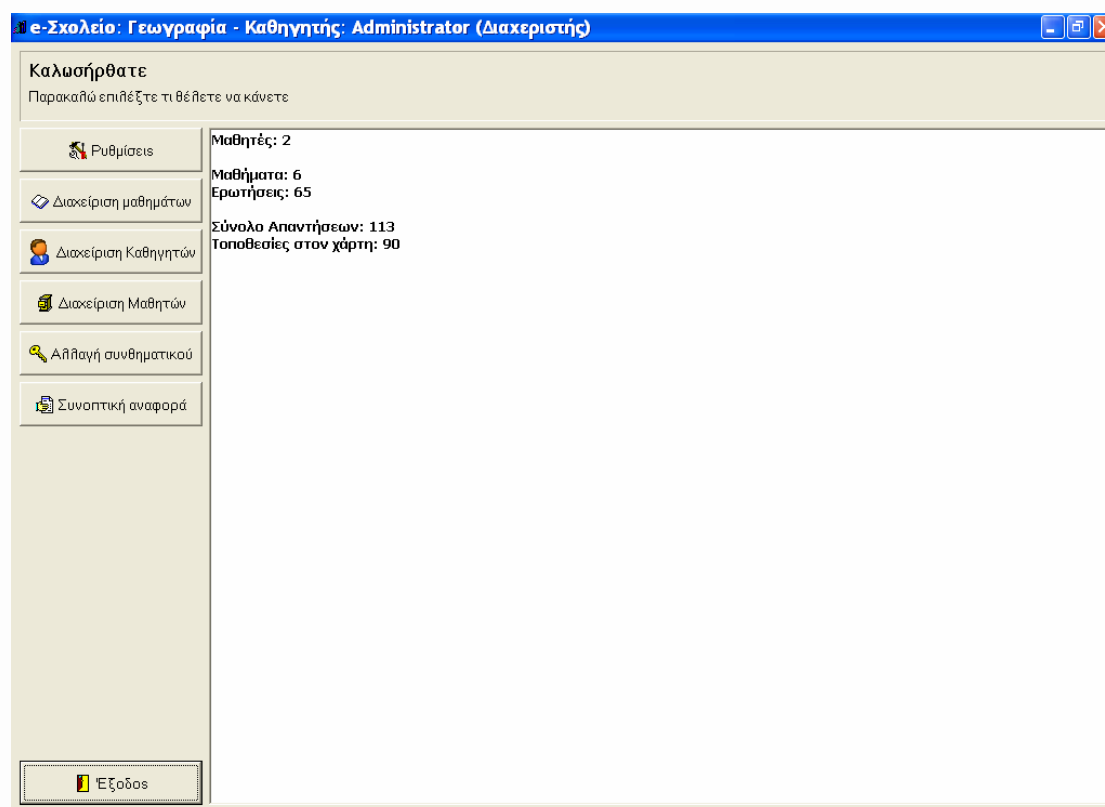
Με κλικ στο κουμπί «Βοήθεια», ο χρήστης μπορεί να δει οδηγίες για τη σύνδεσή του στην εφαρμογή :



Με κλικ στο κουμπί «Έξοδος» κλείνει τελείως το πρόγραμμα και γίνεται επιστροφή στο περιβάλλον των Windows.

### 3.Βασική Οθόνη Administrator

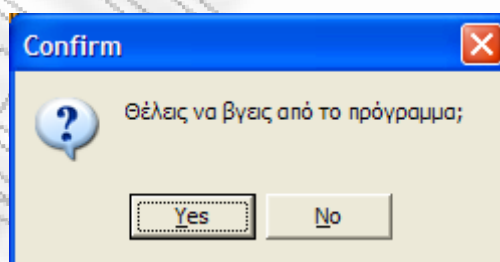
Ο administrator αφού συνδεθεί στην εφαρμογή, μεταβαίνει στην βασική «Οθόνη του Administrator», η οποία παρατίθεται παρακάτω :



Στα αριστερά θα βρίσκονται μία σειρά από πλήκτρα για την πραγματοποίηση των σημαντικότερων λειτουργιών, όπως οι ρυθμίσεις των τεστ, η διαχείριση των μαθητών, των μαθημάτων και των καθηγητών, η αλλαγή του συνθηματικού και η συνοπτική αναφορά για την πρόοδο των μαθητών.

Στα δεξιά και στο μεγαλύτερο κομμάτι της οθόνης εμφανίζεται ένα grid με στοιχεία για τους μαθητές και τα μαθήματα. Ο administrator μπορεί να δει τον αριθμό των μαθητών που παρακολουθούν το μάθημα, τον αριθμό των διαθέσιμων μαθημάτων, τον αριθμό των ερωτήσεων που συμμετέχουν στα εξεταζόμενα τεστ, το συνολικό πλήθος των απαντήσεων και τον αριθμό των τοποθεσιών που βρίσκονται στον χάρτη.

Τέλος, κάνοντας κλικ στο πλήκτρο «Εξοδος» ο administrator επιστρέφει στην Οθόνη Σύνδεσης, αφού πρώτα ερωτηθεί από το πρόγραμμα αν επιθυμεί να βγει :



#### 4. Οθόνη Επεξεργασίας Μαθήματος

Ο administrator έχει τη δυνατότητα να αλλάζει και να καθορίζει τις ρυθμίσεις των τεστ. Πιο αναλυτικά μπορεί να καθορίζει την διάρκεια του τεστ σε δευτερόλεπτα, τον αριθμό των προσπαθειών σε ερωτήσεις πολλαπλής επιλογής, τον αριθμό των προσπαθειών σε ερωτήσεις με χάρτη, τη μέγιστη δυνατή βαθμολογία (άριστα), την ελάχιστη βαθμολογία για να θεωρηθεί επιτυχημένο ένα τεστ (βάση) και τα δεκαδικά ψηφία κατά την εμφάνιση της βαθμολογίας. Η παρακάτω οθόνη είναι ενδεικτική :

Παράμετρος	Τιμή	Περιγραφή
Διάρκεια Τεστ	300	(χρόνος σε sec)
Αριθμός προσπαθειών Ερωτήσεις Πολλ.Επιλ.	2	(Αποδεκτός αριθμός λάθος απαντήσεων)
Αριθμός προσπαθειών Ερωτήσεις σε Χάρτη	3	(Αποδεκτός αριθμός λάθος απαντήσεων)
Άριστα	20.0	(Μέγιστη δυνατή βαθμολογία)
Βάση	10.0	(Η ελάχιστη βαθμολογία για να θεωρηθεί επιτυχημένο ένα τεστ)
Δεκαδικά ψηφία	1	(Αριθμός δεκαδ.ψηφίων κατά την εμφάνιση της βαθμολογίας)

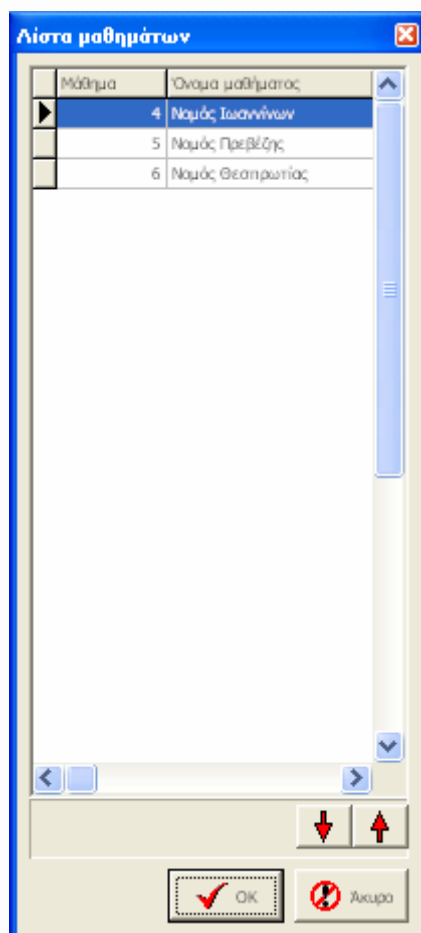
Πατώντας το κουμπί «OK», γίνονται αποδεκτές όλες οι αλλαγές, ενώ με το κουμπί «Άκυρο» δεν πραγματοποιείται καμία αλλαγή.

Ο administrator έχει τη δυνατότητα να «διαχειριστεί» τα μαθήματα που παρακολουθούν οι μαθητές. Η οθόνη αυτή εμφανίζεται, όταν ο administrator κάνει κλικ στο πλήκτρο «Διαχείριση Μαθημάτων» στην βασική «Οθόνη του Administrator» :

Κατάσταση	Τίτλος Μαθήματος	Κείμενο	Αριθμός Παικιδιών	Αρχείο Εικόνας
1	Εισαγωγή	Ο κατάλογος των μαθημάτων είναι ένας ένας ένας		matheio\matheio\matheio.jpg
2	Διευρηματικά Στοιχεία	Ο κατάλογος των μαθημάτων έχει έκταση 4999 π. γράμ		matheio\matheio.jpg
3	Η πόλη των Ιωαννίνων	Το Ιωάννινα ή Γιάννινα είναι η Γάλλικα είναι η παρ		matheio\matheio.jpg
4	Βουνά και πεδιάδες	Το μεγαλύτερο μέρος του νομού Ιωαννίνων ε		matheio\matheio.jpg
5	Ποτάμια	Οι ποτάμια που διαρρέουν τον νομό Ιωαννίνων ε		matheio\matheio.jpg
6	Λίμνες, λιμνοθάλασσες και αετές	Η πόλη των Ιωαννίνων είναι χτισμένη στις όρ		matheio\matheio.jpg
7	Κλίμα	Το κλίμα του νομού είναι ηπειρωτικό. □□□□		matheio\matheio.jpg
8	Αρβάδια - Μημεία	Ο κατάλογος των μαθημάτων είναι πλάτος σε φυσική		matheio\matheio.jpg

Στα αριστερά θα βρίσκονται μία σειρά από πλήκτρα για την πραγματοποίηση των σημαντικότερων λειτουργιών, όπως η δημιουργία νέου μαθήματος και η διαγραφή υπάρχοντος μαθήματος. Όπως, επίσης, και βέλη τα οποία μπορούν να μεταφέρουν τον administrator στο πρώτο ή το τελευταίο μάθημα και στο προηγούμενο ή το επόμενο.

Στα δεξιά και στο μεγαλύτερο κομμάτι της οθόνης εμφανίζεται ένα grid με στοιχεία για των μαθημάτων. Στο πάνω μέρος φαίνεται η σειρά στην οποία βρίσκεται ο τίτλος του μαθήματος και ο αριθμός που καθορίζει τη σειρά παρακολούθησης του μαθήματος. Πατώντας τις τελίτσες εμφανίζεται η παρακάτω οθόνη :



Η οποία δίνει τη δυνατότητα στον administrator να αλλάξει τη σειρά παρακολούθησης των μαθημάτων. Αυτό γίνεται επιλέγοντας το μάθημα και πατώντας τα βέλη για να ανέβει ή να κατέβει προτεραιότητα.

Πιο κάτω βρίσκεται ένα grid όπου ο administrator μπορεί να καταχωρήσει κάποιες σημειώσεις που αφορούν το μάθημα, ώστε να τις χρησιμοποιήσει ο ίδιος ή κάποιος καθηγητής.

Ακόμη πιο κάτω βρίσκεται ένα grid όπου ο administrator μπορεί να διαχειριστεί την ύλη του μαθήματος, τις ερωτήσεις των τεστ και τον χάρτη που εμφανίζεται σε κάθε μάθημα. Στην πρώτη καρτέλα μπορεί να καθορίσει ή να αλλάξει την ύλη του μαθήματος :

Διαχείριση μαθημάτων

Μάθημα: 5 Τίτλος Μαθήματος: Γενικά Περιβάλλον

Σημαντικές μαθήματα

Νέο μάθημα  
Διαγραφή Μαθήματα  
Αποθήκευση Αλλαγών  
Ακύρωση Αλλαγών

Κεφάλαιο	Τίτλος Κεφαλαίου	Κείμενο	Αρχείο Πολυμέσων	Αρχείο Εικόνας
1	Εισαγωγή	Ο ναυός Περιβάλλον είναι ένας από τους 31 ναυό...		εισεως1.jpg
2	Διαμορφωσιών Στοιχεία	Επίσης υπάρχει 1.026 η. γλφ. και πηγήματα 99...		εισεως2.jpg
3	Η πόλη της Περιβάλλον	Η Περιβάλλον είναι η παρουσίαση του ναυού Πε...		εισεως3.jpg
4	Σουά και πεδιάδες	Το έδαφος του ναυού Περιβάλλον είναι γενικά κ...		εισεως4.jpg
5	Ποτάμια	Οι ποταμοί που διαρρέουν τον ναυό Περιβάλλον...		εισεως5.jpg
6	Αίμνες, λιμνοθάλασσες και αετός	Η σημαντικότερη λίμνη του ναυού είναι η σελ...		εισεως6.jpg
7	Κλίμα	Το κλίμα του ναυού στις παράκτιες περιοχές...		εισεως7.jpg
8	Αβράκωτα - Μηνιαία	Ο ναυός Περιβάλλον είναι η κορυφή σε φυσική κ...		εισεως8.jpg

Έξοδος

Όπως φαίνεται και στην παραπάνω οθόνη, ο administrator μπορεί να καθορίσει την σειρά κάθε κεφαλαίου του μαθήματος, να αλλάξει τον τίτλο και το κείμενο που αναφέρεται στο κάθε κεφάλαιο και να προσθέσει ή να αφαιρέσει αρχείο πολυμέσων και εικόνων. Όλα τα παραπάνω πραγματοποιούνται με τη χρήση των κουμπιών που βρίσκονται στο κάτω μέρος της οθόνης, δεξιά. Με το πρώτο κουμπί γίνεται προσθήκη νέου κεφαλαίου και προσθήκη αρχείου πολυμέσων ή εικόνων. Με το δεύτερο κουμπί γίνεται διαγραφή κεφαλαίου και διαγραφή αρχείου πολυμέσων ή εικόνων. Με τα δύο επόμενα κουμπιά, που αποτελούν βέλη, πραγματοποιείται η αλλαγή σειράς των κεφαλαίων. Πραγματοποιώντας οποιαδήποτε αλλαγή, ενεργοποιούνται τα κουμπιά «Αποθήκευση Αλλαγών» και «Ακύρωση Αλλαγών», που βρίσκονται στο αριστερό μέρος της οθόνης :

Κατάλογος	Τίτλος Κατάρτασης	Κείμενο	Αρχείο Πολυμέσων	Αρχείο Εικόνας
1	Εισαγωγή	Ο νωμός Θεσπερίας είναι ένας ένας ένας		θεσπ100.jpg
2	Διευρηματικό Στοιχείο	Ο νωμός Θεσπερίας έχει έκταση 1.515 τ. γλ		piar_piuric.jpg
3	Η πόλη της Ηνωμένητας	Η Ηνωμένητα είναι παραλαιή πόλη, πρωτεύ		hounmentisai.jpg
4	Βουνά και πεδιάδες	Η Θεσπερία είναι περιοχή κατάχλην ορεινή		sou1.jpg
5	Ποτάμια	Οι κερφότερα ποταμά που διαρρέουν τον νωμό		skala_isan.jpg
6	Λίμνες, λιμνοθάλασσες και ακτές	Στον νωμό Θεσπερίας υπάρχουν αρκετές λιμ		subota2.jpg
7	Κλίμα	Στα παράλια κα' στη χαμηλή ζώνη του εσωτερ		ph_ιππασαγια10.jpg
8	Αρφάλιατα - Μηνμεία	Ο νωμός Θεσπερίας είναι ηλιόσκιος σε φυσική		Fanoti.jpg
9				

Επιλέγοντας το κουμπί «Αποθήκευση Αλλαγών», αποθηκεύονται όλες οι αλλαγές που πραγματοποιήθηκαν, ενώ με το κουμπί «Ακύρωση Αλλαγών», ακυρώνονται όλες οι τροποποιήσεις.

Στην δεύτερη καρτέλα ο administrator μπορεί να καθορίσει ή να αλλάξει τις ερωτήσεις που αφορούν κάθε μάθημα :

Ερώτηση	Κατηγορία	Κείμενο ερώτησης	Απάντηση	Σε Χάρτη	Αρχείο εικόνας
1	ΧΑΡΤΗΣ	Δείξε το νωμό Θεσπερίας στον χάρτη της Ελλάδας	Νωμός Θεσπερίας	ΝΑΙ	
2	ΓΕΩΓΡ.ΔΙΑΜΕΡ.	Σε ποια γεωγραφικό διαμέρισμα ανήκει ο νωμός Θεσπερίας;	Ήπειρος	ΟΧΙ	
3	ΠΡΟΤ.ΜΟΜΩΝ	Ποια είναι η πρωτεύουσα του νωμού Θεσπερίας;	Ηνωμένητα	ΟΧΙ	hounmentisai.jpg
4	ΣΥΝΟΡΑ ΝΟΜΩΝ	Ο νωμός Θεσπερίας συνορεύει με το νωμό Πρεβέζης;	Σωστό	ΟΧΙ	
5	ΣΥΝΟΡΑ ΝΟΜΩΝ	Ο νωμός Θεσπερίας συνορεύει με το νωμό Έβρου;	Λάθος	ΟΧΙ	
6	ΔΙΡΗΓΩΡΑΦΙΚΑ	Η κοινότητα Σαυλού ανήκει στον νωμό Θεσπερίας;	Σωστό	ΟΧΙ	sou1_ghfuti.jpg
7	ΔΙΡΗΓΩΡΑΦΙΚΑ	Ο δήμος Σαβίνων ανήκει στο νωμό Θεσπερίας;	Σωστό	ΟΧΙ	subota.jpg
8	ΒΟΥΝΑ	Ποια είναι το ψηλότερο βουνό του νωμού Θεσπερίας;	Μαυρηλάδα	ΟΧΙ	
9	ΒΟΥΝΑ	Το Όρη Φιλιζάν είναι βουνό του νωμού Θεσπερίας;	Σωστό	ΟΧΙ	
10	ΠΟΤΑΜΙΑ	Ποιος από τους παρακάτω ποταμούς διαρρέει τον νωμό Θεσπ	Αχελώϊας	ΟΧΙ	skala_isan.jpg
11	ΠΟΤΑΜΙΑ	Ο ποταμός Καλάμιας εκβάλλει στο παράλια του νωμού Θεσπερ	Σωστό	ΟΧΙ	
12	ΛΙΜΝΕΣ	Η λίμνη Παμβότιδα βρίσκεται στο νωμό Θεσπερίας;	Λάθος	ΟΧΙ	
13	ΛΙΜΝΕΣ	Ποια από τις παρακάτω λίμνες βρίσκεται στο νωμό Θεσπερίας	Λίμνη των Νικαράδων	ΟΧΙ	

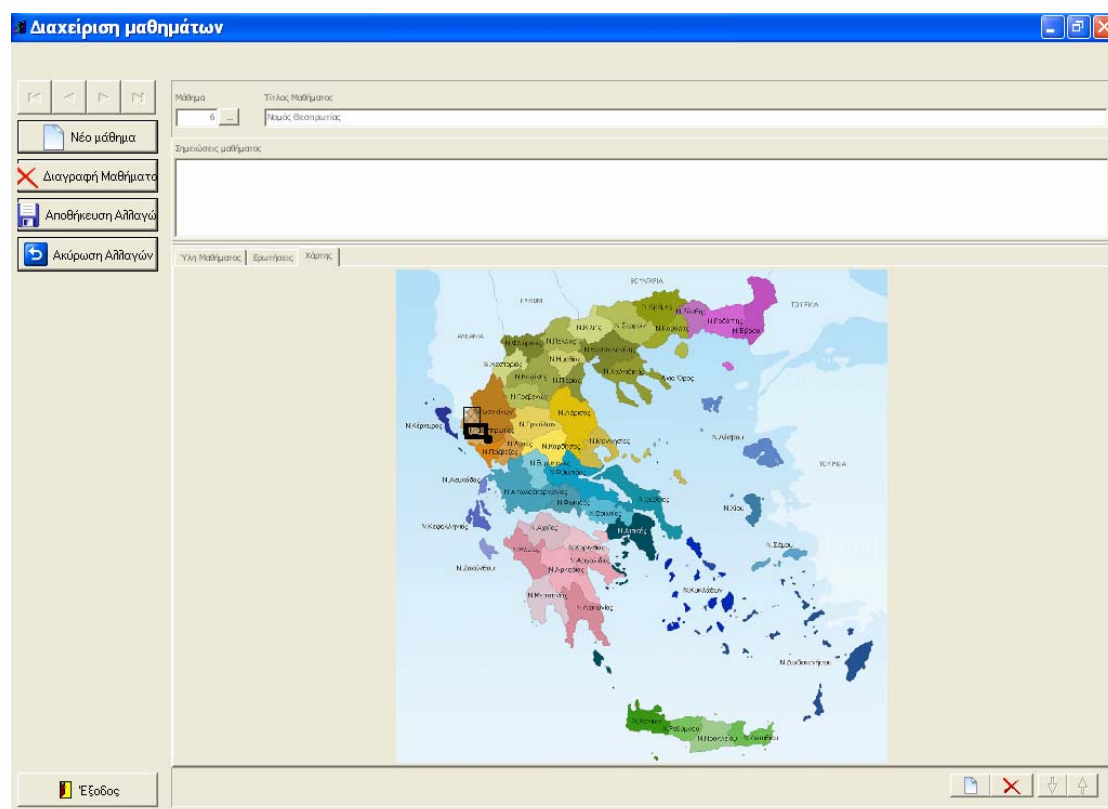
Όπως φαίνεται και στην παραπάνω οθόνη, ο administrator μπορεί να καθορίσει την σειρά κάθε ερώτησης του τεστ, να αλλάξει την κατηγορία που ανήκει κάθε ερώτηση, να αλλάξει το κείμενο που αναφέρεται στο κάθε ερώτηση και την απάντηση, να προσθέσει ή να αφαιρέσει ερώτηση από χάρτη και αρχείο εικόνων. Όλα τα παραπάνω πραγματοποιούνται με τη χρήση των κουμπιών που βρίσκονται στο κάτω μέρος της οθόνης, δεξιά. Με το πρώτο κουμπί γίνεται προσθήκη νέας ερώτησης και προσθήκη αρχείου εικόνων ή ερώτησης στον χάρτη. Με το δεύτερο κουμπί γίνεται διαγραφή ερώτησης, απάντησης και διαγραφή αρχείου εικόνων. Με τα δύο επόμενα κουμπιά, που αποτελούν βέλη, πραγματοποιείται η αλλαγή σειράς των ερωτήσεων. Πραγματοποιώντας οποιαδήποτε αλλαγή, ενεργοποιούνται τα κουμπιά «Αποθήκευση Αλλαγών» και «Ακύρωση Αλλαγών», που βρίσκονται στο αριστερό μέρος της οθόνης:

Ερώτηση	Κατηγορία	Κείμενο ερώτησης	Απάντηση	Στ. Χάρτη	Αρχείο εικόνας
1	ΣΥΝΘΡΑ ΝΟΜΩΝ	Ο νομικός Θεσπρωτίας αναγκάσει με το ναυό Έβρου	Αόδης	ΟΧΙ	
2	ΧΑΡΤΗΣ	Δείξε το ναυό Θεσπρωτίας στον χάρτη της Ελλάδας	Ναυός Θεσπρωτίας	ΝΑΙ	
3	ΓΕΩΓΡ.ΔΙΑΜΕΡ.	Σε ποιο γεωγραφικό διαμέρισμα ανήκει ο ναυός Θεσπρωτίας	Ήπειρος	ΟΧΙ	
4	ΠΡΩΤ.ΝΟΜΩΝ	Παρά είναι η πραγματικότητα του ναυού Θεσπρωτίας	Μοταμηνίσις	ΟΧΙ	krumenitsa1.bmp
5	ΣΥΝΘΡΑ ΝΟΜΩΝ	Ο ναυός Θεσπρωτίας αναγκάσει με το ναυό Πελοπόννησος	Συστό	ΟΧΙ	
6	ΔΗΡΟΤΡΑΦΙΚΑ	Η κανόνισσα Σαβίνας ανήκει στον ναυό Θεσπρωτίας	Συστό	ΟΧΙ	souli_gfari1.bmp
7	ΔΗΡΟΤΡΑΦΙΚΑ	Ο Δέμος Σαβίνας ανήκει στον ναυό Θεσπρωτίας	Συστό	ΟΧΙ	subota1.bmp
8	ΒΟΥΛΑ	Παρά είναι το ψηφίσις βουλό του ναυού Θεσπρωτίας	Μακρονάσις	ΟΧΙ	
9	ΒΟΥΛΑ	Το Όρι Φιλισίων είναι βουλό του ναυού Θεσπρωτίας	Συστό	ΟΧΙ	
10	ΠΟΤΑΜΙΑ	Παρά από τους παραπάνω ποταμούς διασχίζει τον ναυό Θεσπρωτίας	Αρδωνίσις	ΟΧΙ	skala_tsiav1.bmp
11	ΠΟΤΑΜΙΑ	Ο ποταμός Καλαμάς εκβάσει στον ναυό Θεσπρωτίας	Συστό	ΟΧΙ	
12	ΛΙΜΝΕΣ	Η λίμνη Παμβώνας βρίσκεται στον ναυό Θεσπρωτίας	Αόδης	ΟΧΙ	
13	ΛΙΜΝΕΣ	Παρά από τις παραπάνω λίμνες βρίσκεται στον ναυό Θεσπρωτίας	Λίμνη των Πικυράδων	ΟΧΙ	

Επιλέγοντας το κουμπί «Αποθήκευση Αλλαγών», αποθηκεύονται όλες οι αλλαγές που πραγματοποιήθηκαν, ενώ με το κουμπί «Ακύρωση Αλλαγών», ακυρώνονται όλες οι τροποποιήσεις.

Στην τρίτη καρτέλα ο administrator μπορεί να καθορίσει ή να αλλάξει την τοποθεσία του νομού που αφορά το μάθημα στον χάρτη :



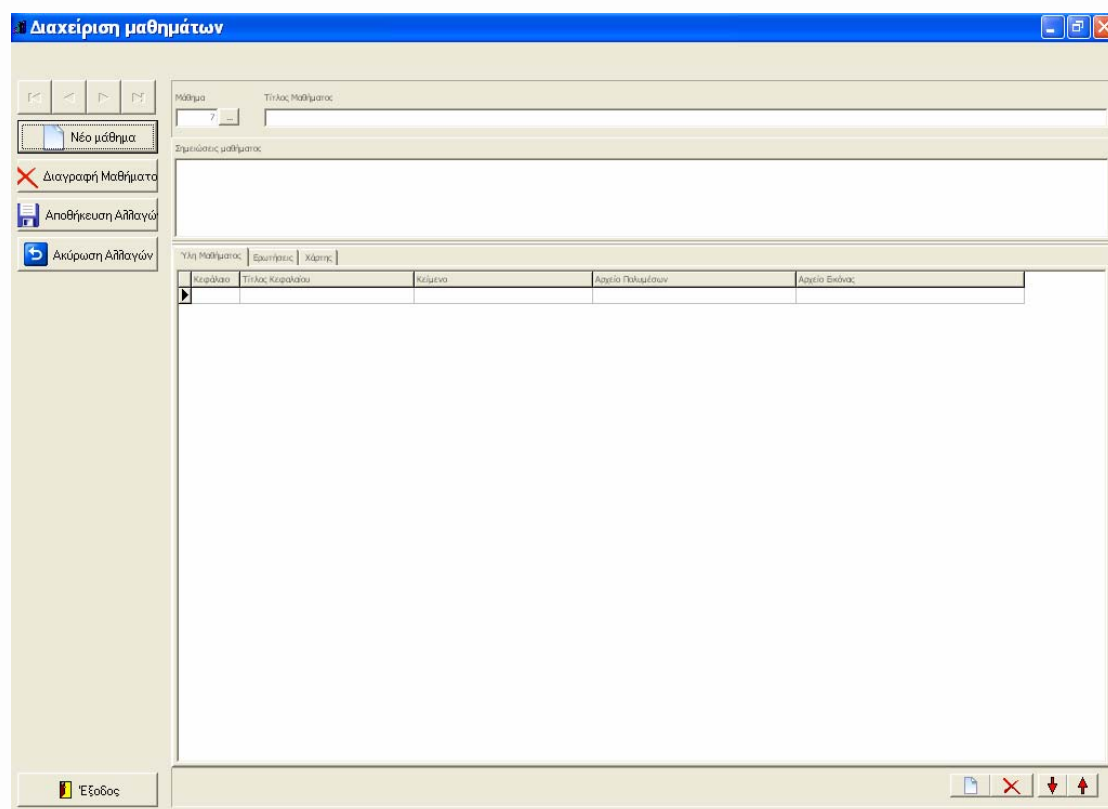


Ο καθορισμός του νομού πάνω στον χάρτη ή η αλλαγή και η διαγραφή πραγματοποιούνται με τη χρήση των κουμπιών που βρίσκονται στο κάτω μέρος της οθόνης, δεξιά. Με το πρώτο κουμπί γίνεται προσθήκη νέας τοποθεσίας πάνω στον χάρτη, ενώ με το δεύτερο κουμπί γίνεται διαγραφή της τοποθεσίας. Επιλέγοντας το κουμπί «Αποθήκευση Αλλαγών», αποθηκεύονται όλες οι αλλαγές που πραγματοποιήθηκαν, ενώ με το κουμπί «Ακύρωση Αλλαγών», ακυρώνονται όλες οι τροποποιήσεις.

Τέλος, κάνοντας κλικ στο πλήκτρο «Εξοδος» ο administrator επιστρέφει στην βασική «Οθόνη του Administrator».

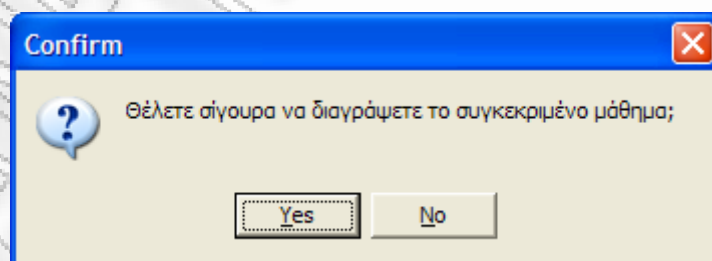
Στην βασική «Οθόνη του Administrator» υπάρχουν και οι επιλογές προσθήκης νέου μαθήματος, πατώντας το κουμπί «Νέο Μάθημα» και διαγραφής μαθήματος πατώντας το κουμπί «Διαγραφή Μαθήματος».

Για να προστεθεί νέο μάθημα, εμφανίζεται η παρακάτω οθόνη :



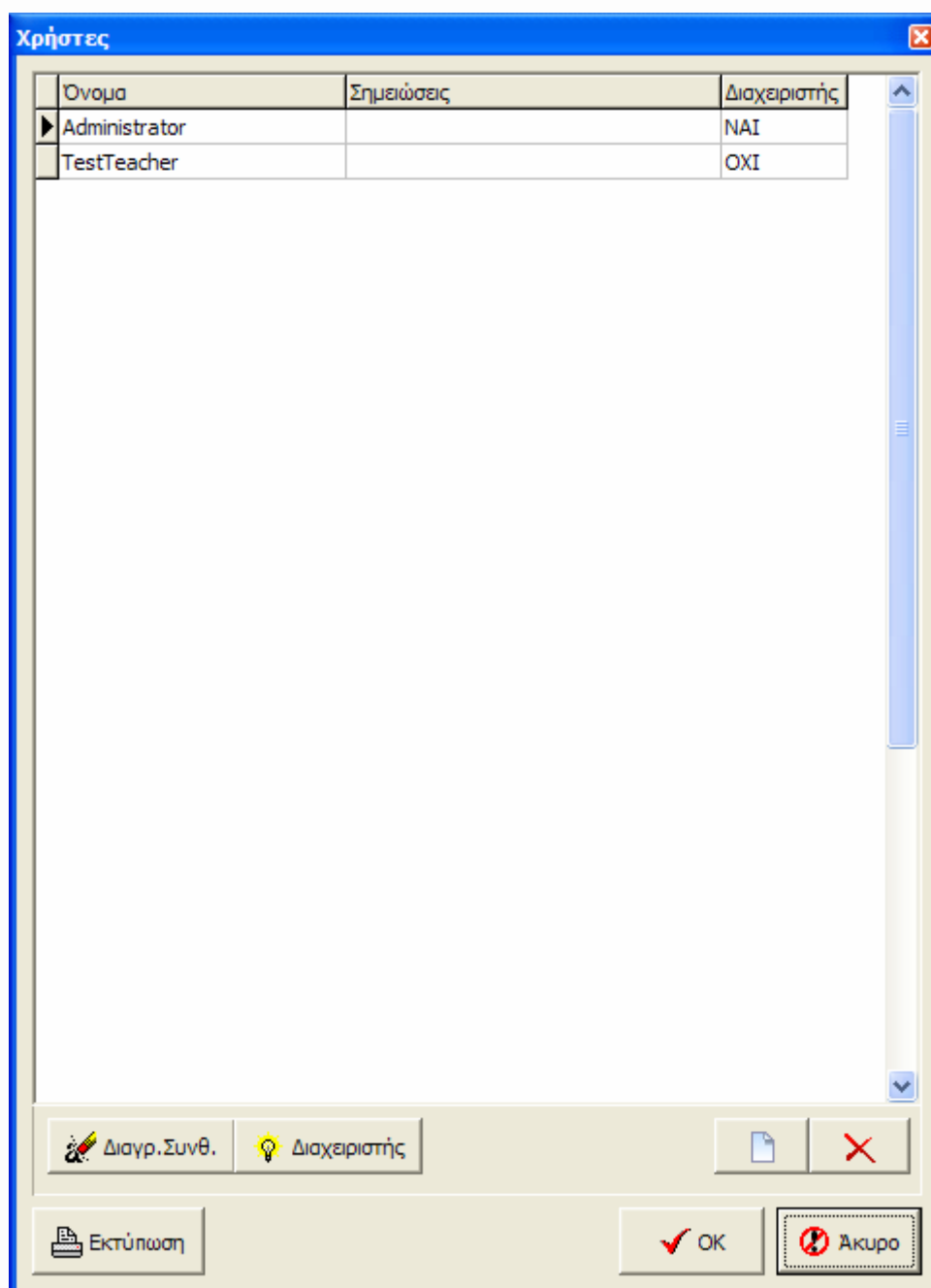
Το μάθημα παίρνει τον τελευταίο αριθμό σειράς, ο οποίος μπορεί να αλλάξει μετά την ενεργοποίηση των βελών, όπως έχει αναφερθεί και προηγουμένως. Επίσης, συμπληρώνονται από τον administrator ο τίτλος του μαθήματος, οι σημειώσεις, η ύλη του μαθήματος (αριθμός κεφαλαίου, τίτλος κεφαλαίου, κείμενο, αρχεία πολυμέσων, αρχεία εικόνων), οι ερωτήσεις που αφορούν το μάθημα (αριθμός ερώτησης, κατηγορία, κείμενο ερώτησης, απάντηση, αν ανήκει σε ερώτηση χάρτη, αρχείο εικόνων) και η τοποθεσία του νομού στον χάρτη. Όλες οι παραπάνω λειτουργίες, καθώς και η διαγραφή πραγματοποιούνται με τη χρήση των κουμπιών που βρίσκονται στο κάτω μέρος της οθόνης, δεξιά. Με το πρώτο κουμπί γίνεται προσθήκη, ενώ με το δεύτερο κουμπί γίνεται διαγραφή. Επιλέγοντας το κουμπί «Αποθήκευση Αλλαγών», αποθηκεύονται όλες οι αλλαγές που πραγματοποιήθηκαν, ενώ με το κουμπί «Ακύρωση Αλλαγών», ακυρώνονται όλες οι τροποποιήσεις.

Για να διαγραφεί κάποιο μάθημα, ο administrator πρέπει να κάνει κλικ στο κουμπί «Διαγραφή Μαθήματος» και εμφανίζεται η παρακάτω οθόνη :



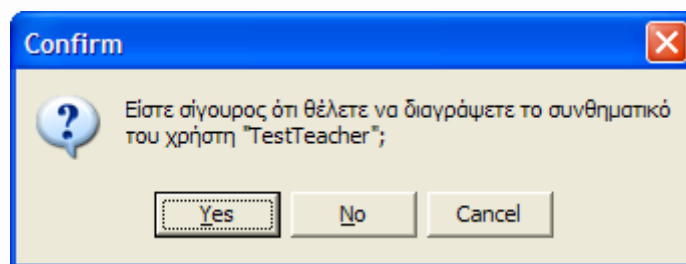
Έτσι ο administrator επιβεβαιώνει αν επιθυμεί ή όχι να διαγράψει το μάθημα.

Ο administrator έχει τη δυνατότητα να «διαχειριστεί» τους καθηγητές που συμμετέχουν στο μάθημα. Η οθόνη αυτή εμφανίζεται, όταν ο administrator κάνει κλικ στο πλήκτρο «Διαχείριση Καθηγητών» στην βασική «Οθόνη του Administrator»:

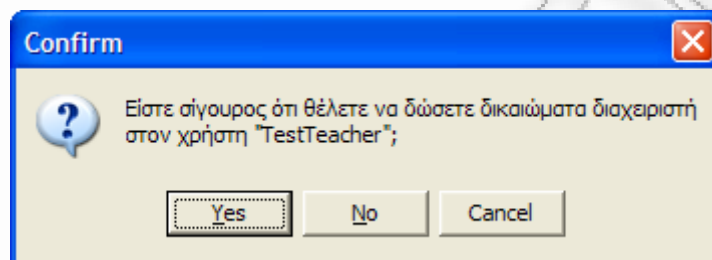


Στο πάνω μέρος της οθόνης εμφανίζονται οι ενεργοί καθηγητές. Ο administrator έχει τη δυνατότητα να δει τα ονόματα των καθηγητών και το δικό του, τυχόν σημειώσεις που έχει κρατήσει και εάν κάποιος καθηγητής έχει τα δικαιώματα του administrator.

Στο κάτω μέρος της οθόνης, αριστερά, υπάρχει το κουμπί «Διαγραφή Συνθηματικού», με το οποίο ο administrator μπορεί να διαγράψει το συνθηματικό κάποιου καθηγητή, εφόσον τον έχει επιλέξει. Το πρόγραμμα ζητά επιβεβαίωση για την διαγραφή :



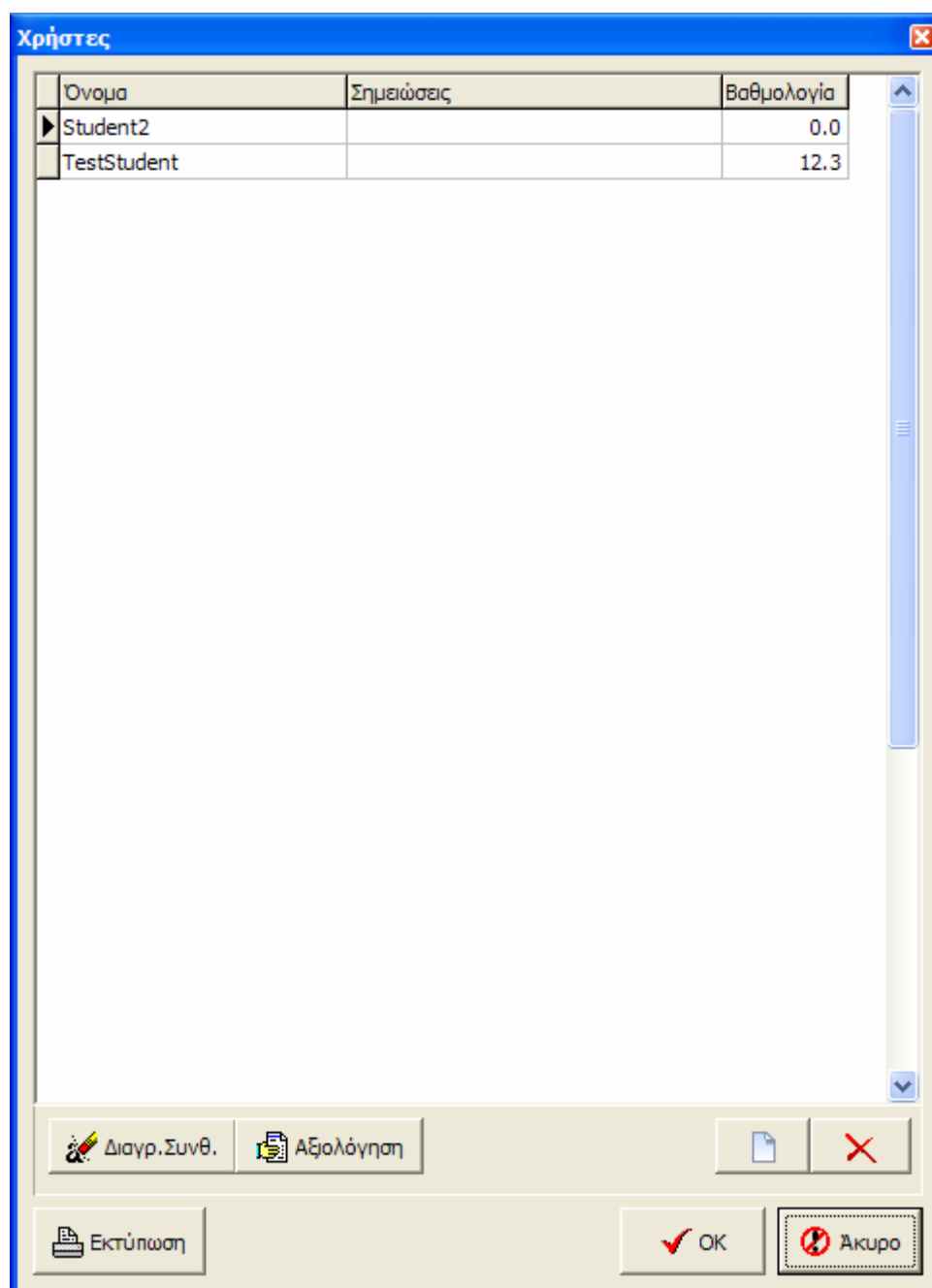
Στο κάτω μέρος της οθόνης, αριστερά, υπάρχει, επίσης, το κουμπί «Διαχειριστής». Πατώντας το ο administrator, αφού έχει επιλέξει κάποιον καθηγητή, μπορεί να του δώσει τα δικαιώματα του administrator. Το πρόγραμμα ζητά επιβεβαίωση για την αλλαγή αυτή :



Στο κάτω μέρος της οθόνης «Χρήστες», δεξιά, υπάρχει το κουμπί «Προσθήκη Καθηγητή», με το οποίο ο administrator μπορεί να προσθέσει έναν νέο καθηγητή και το κουμπί «Διαγραφή Καθηγητή», με το οποίο ο administrator μπορεί να διαγράψει έναν καθηγητή.

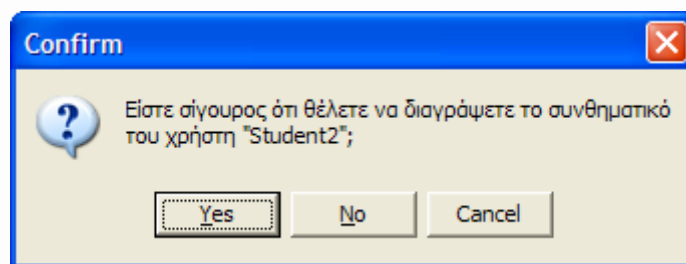
Τέλος στο κάτω μέρος, αριστερά υπάρχει η επιλογή εκτύπωσης, κάνοντας κλικ στο κουμπί «Εκτύπωση».

Ο administrator έχει τη δυνατότητα να «διαχειριστεί» τους μαθητές που παρακολουθούν τα μαθήματα και εξετάζονται στα τεστ. Η οθόνη αυτή εμφανίζεται, όταν ο administrator κάνει κλικ στο πλήκτρο «Διαχείριση Μαθητών» στην βασική «Οθόνη του Administrator»:



Στο πάνω μέρος της οθόνης εμφανίζονται οι ενεργοί μαθητές. Ο administrator έχει τη δυνατότητα να δει τα ονόματα των μαθητών, τυχόν σημειώσεις που έχει κρατήσει και τη συνολική τους βαθμολογία.

Στο κάτω μέρος της οθόνης, αριστερά, υπάρχει το κουμπί «Διαγραφή Συνθηματικού», με το οποίο ο administrator μπορεί να διαγράψει το συνθηματικό κάποιου μαθητή, εφόσον τον έχει επιλέξει. Το πρόγραμμα ζητά επιβεβαίωση για την διαγραφή :



Στο κάτω μέρος της οθόνης, αριστερά, υπάρχει, επίσης το κουμπί «Αξιολόγηση». Πατώντας το ο administrator, αφού έχει επιλέξει κάποιον μαθητή, βλέπει την καρτέλα αξιολόγησης του :

**Αξιολόγηση Μαθητή: TestStudent**

<b>Γενική Βαθμολογία:</b>	12.3
Μαθήματα που παρακολούθησε:	3
Ερωτήσεις που εξετάστηκε:	55
Σωστές απαντήσεις:	42

Επιδόσεις ανά κατηγορία ερωτήσεων:

Καλύτερη Κατηγορία:  
ΣΥΝΟΡΑ ΝΟΜΩΝ (100% Επιτυχία)

Χειρότερη Κατηγορία:  
ΠΡΩΤ.ΝΟΜΩΝ (50% Επιτυχία)

? Αναλυτικά... OK

Στην οθόνη αυτή εμφανίζεται η Καρτέλα Αξιολόγησης, όπου ο administrator μπορεί να δει τα εξής στοιχεία για την πρόοδο του μαθητή:

- **Γενική Βαθμολογία:** Ο μέσος όρος της βαθμολογίας μέχρι τώρα, ανάλογα με το ποσοστό των σωστών απαντήσεων που έδωσε στα τεστ
- **Ερωτήσεις που εξετάστηκε:** Ο συνολικός αριθμός των ερωτήσεων στα διάφορα τεστ που έχει απαντήσει
- **Σωστές Απαντήσεις:** Πόσες από τις απαντήσεις που έδωσε στις παραπάνω ερωτήσεις ήταν σωστές.

Μέσα στο πλαίσιο της οθόνης, ο administrator έχει τη δυνατότητα να δει σε ποια κατηγορία ερωτήσεων ο μαθητής έχει την καλύτερη απόδοση και το ποσοστό επιτυχίας που έχει σε αυτές τις ερωτήσεις. Ακριβώς από κάτω μπορεί να δει σε ποια κατηγορία είχε τη χαμηλότερη βαθμολογία, δηλαδή το χαμηλότερο ποσοστό σωστών απαντήσεων.

Η unit UserManagementUnit που περιγράφηκε παραπάνω χρησιμοποιείται και σε αυτό το σημείο για την περιγραφή της φόρμας «Χρήστες».

Εάν το επιθυμεί, ο administrator μπορεί να κάνει κλικ στο κουμπί «Αναλυτικά» και θα δει αναλυτικά τους τίτλους των μαθημάτων που παρακολούθησε ο μαθητής, την ημερομηνία παρακολούθησης κάθε μαθήματος, την ημερομηνία εξέτασης κάθε τεστ και τη βαθμολογία που έλαβε σε κάθε εξέταση :

Μάθημα	Όνομα μαθήματος	Ημ/νία Παρακ.	Ημ/νία Εξ.τ.
1	Μαμάς Περίας	20/11/2012	
3	Μαμάς Άστρας	20/11/2012	24/11/2012
4	Μαμάς Ιωαννίνων	20/11/2012	26/7/2009
5	Μαμάς Πρεβέζης	20/11/2012	20/11/2012
6	Μαμάς Θεσσαλονίκης	20/11/2012	24/11/2012

Στο κάτω μέρος της οθόνης «Χρήστες», δεξιά, υπάρχει το κουμπί «Προσθήκη Μαθητή», με το οποίο ο administrator μπορεί να προσθέσει έναν νέο μαθητή και το κουμπί «Διαγραφή Μαθητή», με το οποίο ο administrator μπορεί να διαγράψει έναν μαθητή. Τόσο στην οθόνη που εμφανίζεται, όταν ο administrator κάνει κλικ στο πλήκτρο «Διαχείριση Μαθητών» στην βασική «Οθόνη του Administrator», όσο και στην οθόνη «Αναφορά προόδου για τον μαθητή» υπάρχει η επιλογή εκτύπωσης.

Στην παρακάτω οθόνη, ο administrator μπορεί να αλλάξει το συνθηματικό που χρησιμοποιεί για να συνδεθεί στο πρόγραμμα. Η οθόνη αυτή εμφανίζεται, όταν ο καθηγητής κάνει κλικ στο πλήκτρο «Αλλαγή Συνθηματικού» στην βασική «Οθόνη του Administrator» :

**Αλλαγή συνθηματικού**

Γράψτε το παλιό σας συνθηματικό στο επάνω πλαίσιο, και το νέο συνθηματικό και στα δύο κάτω πλαίσια.

Παλιό συνθηματικό:

Νέο συνθηματικό:

Επανάληψη:

**ΠΡΟΣΟΧΗ:** Για το συνθηματικό έχει σημασία αν γράφετε με κεφαλαία ή μικρά!

Στο πάνω πλαίσιο ο administrator γράφει το παλιό του συνθηματικό, και στα δύο κάτω πλαίσια (και στα δύο) γράφει το καινούριο που επιθυμεί να έχει. Κάνοντας κλικ στο «OK» αποθηκεύεται το νέο συνθηματικό. Αν ο administrator αλλάξει γνώμη και θέλει να κρατήσει το παλιό συνθηματικό, κάνει κλικ στο «Άκυρο».

Στην συνοπτική αναφορά, υπάρχει η πρόοδος του μαθητή στα εξεταζόμενα τεστ. Συγκεκριμένα, φαίνεται ο μέσος όρος βαθμολογίας του μαθητή στα τεστ που έλαβε μέρος ανά μάθημα. Επίσης, φαίνεται και σε πόσα τεστ εξετάστηκε ο μαθητής ανά μάθημα.

**StudentSummaryForm**

Μαθητής	Γεωγραφία	Ιστορία
TestStudent	12,4 (1 μαθήματα)	8,6 (2 μαθήματα)



## **ΠΑΡΑΡΤΗΜΑ 2**

### **ΚΩΔΙΚΑΣ ΕΦΑΡΜΟΓΗΣ**

ΠΑΡΑΡΤΗΜΑ 2  
ΚΩΔΙΚΑΣ ΕΦΑΡΜΟΓΗΣ

Για την υλοποίηση του συστήματος διεπαφής με το χρήστη στη συγκεκριμένη εφαρμογή χρησιμοποιήθηκε η γλώσσα προγραμματισμού BDS Delphi 2006 και με τη βοήθεια της οποίας να κατασκευάστηκαν όλες οι φόρμες.

Παρακάτω παρατίθεται ο κώδικας με κυριότερες procedures που χρησιμοποιεί η εφαρμογή, καθώς και εξωτερικά προγράμματα.

```

unit GlobalsUnit;

interface

uses Windows,Controls,ExtCtrls,Classes,db;

var TotalTestTime:Integer = 300; // Test time allowed in seconds
var MaxTriesMultChoice:Integer = 1; // Max. tries for multiple choice questions
var MaxAttemptsMap:Integer = 3; // Max. tries for map questions
var MaxGrade:Real = 20.0; // The max grade
var PassingGrade:Real = 10.0; // The max grade
var GradesFormat:String = '%.1f'; // The format of grade numbers

var RNGSeed:Integer;

function Speak(txt:String):boolean;

Function Encrypted(aPlainText:String):String;
Function FindWordMatch(strWord1, strWord2: String): Integer;

Function ExecuteFile (FileName:string):THandle;
Function ExecuteFileEx(const Operation, FileName, Params, DefaultDir: string;
    ShowCmd: Integer): THandle;

procedure LoadSettings;
procedure SaveSettings;

type TUser = Class
private
    mID: Integer;
    mUserName:string;
    mEncrPassword:string;
    mUserType:Integer;
    mLastLesson:Integer;
public
    property ID: Integer read mID;
    property UserName:string read mUserName;
    property LastLesson:Integer read mLastLesson write mLastLesson;
    property EncrPassword:string read mEncrPassword;
    function IsStudent:boolean;
    function IsTeacher:boolean;
    function IsAdmin:boolean;
    function Login(aUsername,aEncrPassword:string):boolean;
    function ChangePassword(aEncrOldPassword,aEncrNewPassword:string):boolean;
    procedure Store;
end;

type TMapImageConv = Class
private
    mRefImage:TImage;
public
    property RefImage:TImage read mRefImage;
    constructor Create(aRefImage:TImage);
    function ImageAddedLeft: Integer;

```

```

function ImageAddedTop: Integer;
function MapToImage(MapCoords:TPoint): TPoint;
function ImageToMap(ImageCoords:TPoint): TPoint;
function MapToImageX(MapX:Integer): Integer;
function ImageToMapX(ImageX:Integer): Integer;
function MapToImageY(MapY:Integer): Integer;
function ImageToMapY(ImageY:Integer): Integer;
end;

```

```

type TMapBoxItem = Class (TCollectionItem)
private
  mTag:Integer;
  mBoxShape:TShape;
  procedure AssignShape(aBoxShape:TShape);
  function GetTop:Integer;
  function GetLeft:Integer;
  function GetTopLeft:TPoint;
  function GetWidth:Integer;
  function GetHeight:Integer;
public
  property Left:Integer read GetLeft;
  property Top:Integer read GetTop;
  property TopLeft:TPoint read GetTopLeft;
  property Width:Integer read GetWidth;
  property Height:Integer read GetHeight;
  property BoxShape:TShape read mBoxShape write AssignShape;
  property Tag:Integer read mTag write mTag;
  destructor Destroy; override;
end;

```

```

type TMapBoxEvent = procedure(aBoxIndex,aBoxTag:Integer) of Object;

```

```

type TMapBoxCollection = Class (TCollection)
private
  mImage: TImage;
  mMapImageConv: TMapImageConv;
  mOnBoxClick:TMapBoxEvent;
  mOnBoxMouseOver:TMapBoxEvent;
  procedure BoxMouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer); //TMouseEvent
  procedure BoxMouseMove(Sender: TObject; Shift: TShiftState;
    X, Y: Integer); //TMouseMoveEvent
public
  constructor Create(almage:TImage); overload;
  property MapImageConv:TMapImageConv read mMapImageConv;
  property OnBoxClick:TMapBoxEvent read mOnBoxClick write mOnBoxClick;
  property OnBoxMouseOver:TMapBoxEvent read mOnBoxMouseOver write
mOnBoxMouseOver;
  function NewBox(aTop,aLeft,aWidth,aHeight:Integer):Integer; overload;
  function NewBox(aTop,aLeft,aWidth,aHeight,Tag:Integer):Integer; overload;
  function NewBox(aTop,aLeft,aWidth,aHeight,Tag:Integer;IsVisible:Boolean):Integer;
overload;
end;

```

```

type TSettingsReader = Class
private
  mDS: TDataSet;
public
  constructor Create(aDS:TDataSet); overload;
  destructor Destroy; override;
  function ReadSettingStr(SettingKey:String;DefValue:String):String; overload;

```

```

function ReadSettingInt(SettingKey:String;DefValue:Integer):Integer; overload;
function ReadSettingReal(SettingKey:String;DefValue:Real):Real; overload;
function ReadSettingStr(SettingKey:String):String; overload;
function ReadSettingInt(SettingKey:String):Integer; overload;
function ReadSettingReal(SettingKey:String):Real; overload;
procedure WriteSettingStr(SettingKey:String;Value:String);
procedure WriteSettingInt(SettingKey:String;Value:Integer);
procedure WriteSettingReal(SettingKey:String;Value:Real);
end;

var
  User:TUser;

implementation

uses Forms,SysUtils,GeoDataUnit,Dialogs,ShellAPI,Graphics;

type
  TDemoSend = function (data: PChar): boolean; stdcall;

function Speak(txt:String):boolean;
var Handle: Integer;
var DemoSend: TDemoSend;
////////////////////////////////////
/// Όνομα: Speak
/// Τύπος: Public Function
/// Επιστρέφει:
///   Boolean - True = Επιτυχής εκτέλεση
///             - False = Απέτυχε
/// Παράμετροι:
///   txt:String - το κείμενο προς εκφώνηση
/// Περιγραφή:
///   Η Speak καλείται από το υπόλοιπο πρόγραμμα για να εκφωνηθεί κάποιο
///   κείμενο με ανθρώπινη ομιλία (text-to-speech) χρησιμοποιώντας τον
///   συνθέτη ομιλίας "Δημοσθένης"
////////////////////////////////////
begin
  Result := False;

  Handle := LoadLibrary('demosend.dll');
  if Handle <> 0 then
  begin
    @DemoSend := GetProcAddress(Handle, 'demosend');
    if @DemoSend <> nil then
    try
      if DemoSend(PChar('<play>'+txt)) then
        Result := True;
      finally
        FreeLibrary(Handle);
      end;
    end;
  end;
end;

constructor TMapBoxCollection.Create(alimage:TImage);
////////////////////////////////////
/// Όνομα: Create
/// Τύπος: Constructor
/// Παράμετροι:
///   alimage:TImage - το αντικείμενο τύπου εικόνας πάνω στην οποία θα
///                   γίνει η σχεδίαση
/// Περιγραφή:

```

```

/// Τα αντικείμενα της κλάσεως TMapBoxCollection αποθηκεύουν μία συλλογή
/// (collection) από αντικείμενα TMapBoxItem που αντιπροσωπεύουν τα
/// τετράγωνα πάνω σε έναν χάρτη. Ο constructor δημιουργεί μία νέα συλλογή
/// χρησιμοποιώντας ως βάση την εικόνα που δίνεται στην παράμετρο aImage
////////////////////////////////////
begin
  inherited Create(TMapBoxItem);
  mImage := aImage;
  mMapImageConv := TMapImageConv.Create(mImage);
end;

procedure TMapBoxCollection.BoxMouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
////////////////////////////////////
/// Όνομα: BoxMouseMove
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Μετακίνηση του δείκτη του ποντικιού πάνω σε ένα από τα τετράγωνα
/// της συλλογής
/// Παράμετροι:
/// Sender:TObject - Το τετράγωνο πάνω από το οποίο κινείται το ποντίκι
/// Shift: TShiftState - Η κατάσταση των πλήκτρων Ctrl,Alt,Shift, κτλ (αγνοείται)
/// X,Y: Integer - Η θέση του ποντικιού σε σχέση με την πάνω αριστερή γωνία
/// του Sender
/// Περιγραφή:
/// Όλα τα events MouseMove των boxes της συλλογής καταλήγουν σε αυτόν τον
/// event handler ο οποίος πυροδοτεί το event OnBoxMouseOver που συλλαμβάνεται
/// από το υπόλοιπο πρόγραμμα όταν απαιτείται κάποια ενέργεια την στιγμή
/// που ο χρήστης μετακινεί το ποντίκι πάνω από κάποιο τετράγωνο στον χάρτη
////////////////////////////////////
var BoxShape: TShape;
    BoxItem: TMapBoxItem;
begin
  if (Sender is TShape) then
  begin
    BoxShape := (Sender as TShape);
    BoxItem := (Self.FindItemID(BoxShape.Tag - 9999) as TMapBoxItem);
    if Assigned(BoxItem) then
    begin
      if Assigned(OnBoxMouseOver) then
      begin
        OnBoxMouseOver(BoxItem.Index, BoxItem.Tag);
      end;
    end;
  end;
end;

procedure TMapBoxCollection.BoxMouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
////////////////////////////////////
/// Όνομα: BoxMouseDown
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του ποντικιού πάνω σε ένα από τα τετράγωνα της συλλογής
/// Παράμετροι:
/// Sender:TObject - Το τετράγωνο πάνω από το οποίο κινείται το ποντίκι
/// Button: TMouseButton - Ποιο από τα πλήκτρα του ποντικιού πατήθηκε
/// Shift: TShiftState - Η κατάσταση των πλήκτρων Ctrl,Alt,Shift, κτλ (αγνοείται)
/// X,Y: Integer - Η θέση του ποντικιού σε σχέση με την πάνω αριστερή γωνία
/// του Sender
/// Περιγραφή:

```

```

/// Όλα τα events MouseDown των boxes της συλλογής καταλήγουν σε αυτόν τον
/// event handler ο οποίος πυροδοτεί το event OnBoxClick που συλλαμβάνεται
/// από το υπόλοιπο πρόγραμμα όταν απαιτείται κάποια ενέργεια την στιγμή
/// που ο χρήστης κάνει κλικ σε κάποιο τετράγωνο στον χάρτη.
/// Αν δεν ήταν μεμονωμένο κλικ αλλά παρατεταμένο, που δηλώνει dragging
/// του τετραγώνου, τότε καλείται η BeginDrag του τετραγώνου, για να
/// γίνει ο χειρισμός του dragging.
////////////////////////////////////
var BoxShape: TShape;
    BoxItem: TMapBoxItem;
begin
    if (Sender is TShape) then
        begin
            BoxShape := (Sender as TShape);
            BoxItem := (Self.FindItemID(BoxShape.Tag - 9999) as TMapBoxItem);
            if Assigned(BoxItem)
                and (Button = mbLeft) then //αριστερό κλικ σε τετράγωνο της συλλογής
                begin
                    if Assigned(OnBoxClick) and (not BoxShape.Dragging) then
                        begin
                            OnBoxClick(BoxItem.Index, BoxItem.Tag);
                        end;
                    If Assigned(BoxShape.OnStartDrag) then
                        begin
                            BoxShape.BeginDrag(False,3);
                        end;
                    end;
                end;
            end;
        end;
end;

function TMapBoxCollection.NewBox(aTop, aLeft, aWidth, aHeight, Tag: Integer; IsVisible:
Boolean): Integer;
////////////////////////////////////
/// Όνομα: NewBox
/// Τύπος: Public Function (overloaded)
/// Επιστρέφει:
///     Integer - το Index του νέου τετραγώνου
/// Παράμετροι:
///     aTop,aLeft: Integer - Οι συντεταγμένες του νέου τετραγώνου,
///     κανονικοποιημένες από (0,0) έως (1000,1000)
///     aWidth,aHeight: Integer - το πλάτος του τετραγώνου, κανονικοποιημένο
///     με τον ίδιο τρόπο
///     Tag: Integer - Ένας ακέραιος αριθμός που χρησιμοποιείται για την
///     ταυτοποίηση του τετραγώνου
///     IsVisible: Boolean - True = Το τετράγωνο είναι ορατό
///     - False = Το τετράγωνο δεν είναι ορατό
/// Περιγραφή:
///     Με την NewBox προστίθεται ένα νέο τετράγωνο στην συλλογή, στις
///     συντεταγμένες που δίνονται, και με το μέγεθος που δίνεται.
///     Οι κανονικοποιημένες συντεταγμένες μετατρέπονται σε πραγματικά pixels
///     μέσω της κλάσης TMapImageConv και το τετράγωνο εμφανίζεται στην οθόνη
///     ως ένα αντικείμενο TShape.
////////////////////////////////////
var NewMapBoxItem: TMapBoxItem;
    NewShape: TShape;
begin
    NewMapBoxItem := (Self.Add as TMapBoxItem);
    NewShape := TShape.Create(mImage.Owner);
    with NewShape do
        begin
            Shape := stRectangle;

```

```

Top := mMapImageConv.ImageAddedTop + mMapImageConv.MapToImageX(aTop);
Left := mMapImageConv.ImageAddedLeft + mMapImageConv.MapToImageY(aLeft);
Width := mMapImageConv.MapToImageX(aWidth);
Height := mMapImageConv.MapToImageY(aHeight);
Brush.Color := clGray;
Brush.Style := bsDiagCross;
Pen.Color := clBlack;
if isVisible then Pen.Mode := pmCopy
  else Pen.Mode := pmNop;
Pen.Style := psSolid;
Pen.Width := 1;
Parent := mImage.Parent;
Visible := True;
Enabled := True;
OnMouseDown := BoxMouseDown;
OnMouseMove := BoxMouseMove;
end;
NewMapBoxItem.BoxShape := NewShape;
NewMapBoxItem.Tag := Tag;
Result := NewMapBoxItem.Index;
end;

```

```

function TMapBoxCollection.NewBox(aTop,aLeft,aWidth,aHeight,Tag:Integer):Integer;

```

```

////////////////////////////////////
/// Όνομα: NewBox
/// Τύπος: Public Function (overloaded)
/// Επιστρέφει:
///   Integer - το Index του νέου τετραγώνου
/// Παράμετροι:
///   aTop,aLeft: Integer - Οι συντεταγμένες του νέου τετραγώνου,
///                   κανονικοποιημένες από (0,0) έως (1000,1000)
///   aWidth,aHeight: Integer - το πλάτος του τετραγώνου, κανονικοποιημένο
///                   με τον ίδιο τρόπο
///   Tag: Integer - Ένας ακέραιος αριθμός που χρησιμοποιείται για την
///                   ταυτοποίηση του τετραγώνου
/// Περιγραφή:
///   Σε αυτή την εκδοχή της NewBox το τετράγωνο που δημιουργείται
///   δεν είναι ορατό (isVisible = False)
////////////////////////////////////
begin
  Result := NewBox(aTop,aLeft,aWidth,aHeight,Tag,False);
end;

```

```

function TMapBoxCollection.NewBox(aTop,aLeft,aWidth,aHeight:Integer):Integer;

```

```

////////////////////////////////////
/// Όνομα: NewBox
/// Τύπος: Public Function (overloaded)
/// Επιστρέφει:
///   Integer - το Index του νέου τετραγώνου
/// Παράμετροι:
///   aTop,aLeft: Integer - Οι συντεταγμένες του νέου τετραγώνου,
///                   κανονικοποιημένες από (0,0) έως (1000,1000)
///   aWidth,aHeight: Integer - το πλάτος του τετραγώνου, κανονικοποιημένο
///                   με τον ίδιο τρόπο
/// Περιγραφή:
///   Σε αυτή την εκδοχή της NewBox το τετράγωνο που δημιουργείται
///   δεν είναι ορατό (isVisible = False) και η ιδιότητα Tag τίθεται 0
////////////////////////////////////
begin
  Result := NewBox(aTop,aLeft,aWidth,aHeight,0,False);
end;

```

```

procedure TMapBoxItem.AssignShape(aBoxShape:TShape);
////////////////////////////////////////////////////
/// Όνομα: AssignShape
/// Τύπος: Public Procedure
/// Παράμετροι:
///   aBoxShape: TShape
/// Περιγραφή:
///   Μέσω αυτής της procedure γίνεται η σύνδεση ενός τετραγώνου TMapBoxItem
///   με το αντίστοιχο TShape που το απεικονίζει στην οθόνη. Χρησιμοποιείται
///   η ιδιότητα Tag του αντικειμένου TShape για να αποθηκευτεί ο κωδικός
///   του αντίστοιχου TMapBoxItem
////////////////////////////////////////////////////
begin
  if not Assigned(mBoxShape) then
    begin
      mBoxShape := aBoxShape;
      mBoxShape.Tag := 9999 + Self.ID;
    end;
  end;
end;

function TMapBoxItem.GetTop:Integer;
////////////////////////////////////////////////////
/// Όνομα: GetTop
/// Τύπος: Public Function
/// Επιστρέφει:
///   Integer - Η κατακόρυφη θέση του τετραγώνου, κανονικοποιημένη από 0 έως 1000
/// Παράμετροι:
///   (καμία)
/// Περιγραφή:
///   Η θέση του αντικειμένου TShape σε pixels μετατρέπεται σε κανονικοποιημένες
///   τιμές μέσω της MapImageConv και επιστρέφεται η τιμή που προκύπτει
////////////////////////////////////////////////////
begin
  If Assigned(mBoxShape) then
    begin
      Result :=
        (Collection as TMapBoxCollection).MapImageConv.ImageToMapY
        (mBoxShape.Top - Collection as
TMapBoxCollection).MapImageConv.ImageAddedTop);
    end else Result := -1;
  end;
end;

function TMapBoxItem.GetLeft:Integer;
////////////////////////////////////////////////////
/// Όνομα: GetLeft
/// Τύπος: Public Function
/// Επιστρέφει:
///   Integer - Η οριζόντια θέση του τετραγώνου, κανονικοποιημένη από 0 έως 1000
/// Παράμετροι:
///   (καμία)
/// Περιγραφή:
///   Η θέση του αντικειμένου TShape σε pixels μετατρέπεται σε κανονικοποιημένες
///   τιμές μέσω της MapImageConv και επιστρέφεται η τιμή που προκύπτει
////////////////////////////////////////////////////
begin
  If Assigned(mBoxShape) then
    begin
      Result :=
        (Collection as TMapBoxCollection).MapImageConv.ImageToMapX

```



```

        (mBoxShape.Left
TMapBoxCollection).MapImageConv.ImageAddedLeft);
    end else Result := -1;
end;

function TMapBoxItem.GetTopLeft:TPoint;
////////////////////////////////////
/// Όνομα: GetTopLeft
/// Τύπος: Public Function
/// Επιστρέφει:
/// Integer - Οι συντεταγμένες της πάνω-αριστερής γωνίας του τετραγώνου,
/// κανονικοποιημένες από 0 έως 1000
/// Παράμετροι:
/// (καμία)
/// Περιγραφή:
/// Καλούνται οι GetLeft και GetTop και το αποτέλεσμά τους συνδυάζεται
/// σε μία μεταβλητή τύπου TPoint
////////////////////////////////////
begin
    Result.X := GetLeft;
    Result.Y := GetTop;
end;

function TMapBoxItem.GetWidth:Integer;
////////////////////////////////////
/// Όνομα: GetWidth
/// Τύπος: Public Function
/// Επιστρέφει:
/// Integer - Το πλάτος του τετραγώνου, κανονικοποιημένο από 0 έως 1000
/// Παράμετροι:
/// (καμία)
/// Περιγραφή:
/// Το πλάτος του αντικειμένου TShape σε pixels μετατρέπεται σε κανονικοποιημένες
/// τιμές μέσω της MapImageConv και επιστρέφεται η τιμή που προκύπτει
////////////////////////////////////
begin
    If Assigned(mBoxShape) then
        begin
            Result := (Collection as TMapBoxCollection).MapImageConv.ImageToMapX
                (mBoxShape.Width);
        end else Result := -1;
    end;
end;

function TMapBoxItem.GetHeight:Integer;
////////////////////////////////////
/// Όνομα: GetHeight
/// Τύπος: Public Function
/// Επιστρέφει:
/// Integer - Το ύψος του τετραγώνου, κανονικοποιημένο από 0 έως 1000
/// Παράμετροι:
/// (καμία)
/// Περιγραφή:
/// Το ύψος του αντικειμένου TShape σε pixels μετατρέπεται σε κανονικοποιημένες
/// τιμές μέσω της MapImageConv και επιστρέφεται η τιμή που προκύπτει
////////////////////////////////////
begin
    If Assigned(mBoxShape) then
        begin
            Result := (Collection as TMapBoxCollection).MapImageConv.ImageToMapY
                (mBoxShape.Height);
        end else Result := -1;
    end;
end;

```

```

end;

destructor TMapBoxItem.Destroy;
////////////////////////////////////
/// Όνομα: Destroy
/// Τύπος: Destructor
/// Παράμετροι:
/// (καμία)
/// Περιγραφή:
/// Πριν καταστραφεί το αντικείμενο TMapBoxItem φροντίζει να καταστραφεί
/// και το αντίστοιχο αντικείμενο σχεδίασης TShape, εάν υπάρχει
////////////////////////////////////
begin
  if Assigned(mBoxShape) then mBoxShape.Destroy;
  inherited Destroy;
end;

Constructor TMapImageConv.Create(aRefImage:TImage);
////////////////////////////////////
/// Όνομα: Create
/// Τύπος: Constructor
/// Παράμετροι:
/// aRefImage:TImage - το αντικείμενο τύπου εικόνας σε σχέση με το οποίο θα
/// γίνονται οι μετατροπές
/// Περιγραφή:
/// Η κλάση TMapImageConv αναλαμβάνει τις μετατροπές από πραγματικές
/// συντεταγμένες της οθόνης σε pixels στις κανονικοποιημένες συντεταγμένες
/// από (0,0) έως (1000,1000) που χρησιμοποιούνται εσωτερικά στο πρόγραμμα
/// και αποθηκεύονται στη βάση. Με τη μετατροπή αυτή εξασφαλίζουμε ότι
/// τα τετράγωνα θα σχεδιάζονται πάντα στην ίδια θέση στο χάρτη, ανεξάρτητα
/// από τις ρυθμίσεις της οθόνης του χρήστη. Η εικόνα aRefImage αποτελεί το
/// σύστημα αναφοράς των μετατροπών
////////////////////////////////////
begin
  inherited Create;
  mRefImage := aRefImage;
end;

function TMapImageConv.ImageAddedLeft: Integer;
////////////////////////////////////
/// Όνομα: ImageAddedLeft
/// Τύπος: Public Function
/// Επιστρέφει:
/// Integer - Το αριστερό άκρο της εμφανιζόμενης εικόνας σε pixels,
/// λαμβάνοντας υπόψη την ύπαρξη λευκού πλαισίου αριστερά
/// Παράμετροι:
/// (καμία)
/// Περιγραφή:
/// Εάν οι αναλογίες του πλαισίου της mPRefImage διαφέρουν από τις αναλογίες
/// της εικόνας mPRefImage.Picture, μπορεί να εμφανιστεί λευκό πλαίσιο αριστερά
/// και δεξιά, ή πάνω και κάτω, καθώς η Picture κεντράρεται μέσα στο πλαίσιο.
/// Αυτή η function επιστρέφει την θέση σε pixels που ξεκινάει η εικόνα
/// απο αριστερά αφού πρώτα υπολογίσει αν υπάρχει λευκό πλαίσιο ή όχι
////////////////////////////////////
var ZoomScaleX,ZoomScaleY,ZoomScale,a:Real;
begin
  ZoomScaleX := mRefImage.ClientWidth / mRefImage.Picture.Width;
  ZoomScaleY := mRefImage.ClientHeight / mRefImage.Picture.Height;
  if ZoomScaleX<ZoomScaleY then
    Result := mRefImage.Left
  else

```

```

begin
  ZoomScale := ZoomScaleY;
  if ZoomScale > 1 then ZoomScale := 1;
  a := (mReflImage.ClientWidth - (mReflImage.Picture.Width * ZoomScale)) / 2;
  Result := mReflImage.Left + Round(a);
end;
end;

function TMapImageConv.ImageAddedTop: Integer;
////////////////////////////////////////////////////////////////////
// Όνομα: ImageAddedLeft
// Τύπος: Public Function
// Επιστρέφει:
//   Integer - Το άνω άκρο της εμφανιζόμενης εικόνας σε pixels,
//             λαμβάνοντας υπόψη την ύπαρξη λευκού πλαισίου επάνω
// Παράμετροι:
//   (καμία)
// Περιγραφή:
//   Εάν οι αναλογίες του πλαισίου της mPREFImage διαφέρουν από τις αναλογίες
//   της εικόνας mPREFImage.Picture, μπορεί να εμφανιστεί λευκό πλαίσιο αριστερά
//   και δεξιά, ή πάνω και κάτω, καθώς η Picture κεντράρεται μέσα στο πλαίσιο.
//   Αυτή η function επιστρέφει την θέση σε pixels που ξεκινάει η εικόνα
//   από επάνω αφού πρώτα υπολογίσει αν υπάρχει λευκό πλαίσιο ή όχι.
////////////////////////////////////////////////////////////////////
var ZoomScaleX,ZoomScaleY,ZoomScale,a:Real;
begin
  ZoomScaleX := mReflImage.ClientWidth / mReflImage.Picture.Width;
  ZoomScaleY := mReflImage.ClientHeight / mReflImage.Picture.Height;
  if ZoomScaleX<ZoomScaleY then
    begin
      ZoomScale := ZoomScaleX;
      if ZoomScale > 1 then ZoomScale := 1;
      a := (mReflImage.ClientHeight - (mReflImage.Picture.Height * ZoomScale)) / 2;
      Result := mReflImage.Top + Round(a);
    end else
      Result := mReflImage.Top ;
  end;
end;

function TMapImageConv.MapToImage(MapCoords: TPoint):TPoint;
////////////////////////////////////////////////////////////////////
// Όνομα: MapToImage
// Τύπος: Public Function
// Επιστρέφει:
//   TPoint - Θέση σε pixels
// Παράμετροι:
//   MapCoords: TPoint - Θέση σε κανονικοποιημένες συντεταγμένες
// Περιγραφή:
//   Η function αυτή καλεί τις MapToImageX και MapToImageY και επιστρέφει
//   την θέση σε pixels της κανονικοποιημένης θέσης που δόθηκε ως παράμετρος
////////////////////////////////////////////////////////////////////
begin
  Result.X := MapToImageX(MapCoords.X);
  Result.Y := MapToImageY(MapCoords.Y);
end;

function TMapImageConv.ImageToMap(ImageCoords:TPoint):TPoint;
////////////////////////////////////////////////////////////////////
// Όνομα: MapToImage
// Τύπος: Public Function
// Επιστρέφει:
//   TPoint - Θέση σε κανονικοποιημένες συντεταγμένες

```

```

/// Παράμετροι:
/// MapCoords: TPoint - Θέση σε pixels
/// Περιγραφή:
/// Η function αυτή καλεί τις ImageToMapX και ImageToMapY και επιστρέφει
/// την κανονικοποιημένη μορφή της θέσης σε pixels που δόθηκε ως παράμετρος
////////////////////////////////////
begin
  Result.X := ImageToMapX(ImageCoords.X);
  Result.Y := ImageToMapY(ImageCoords.Y);
end;

function TMapImageConv.MapToImageX(MapX:Integer): Integer;
////////////////////////////////////
/// Όνομα: MapToImageX
/// Τύπος: Public Function
/// Επιστρέφει:
/// Integer - Οριζόντια Θέση σε pixels
/// Παράμετροι:
/// MapX: Integer - Οριζόντια Θέση σε κανονικοποιημένες συντεταγμένες
/// Περιγραφή:
/// Η function αυτή υπολογίζει την θέση σε πραγματικά pixels που αντιστοιχεί
/// στην κανονικοποιημένη θέση που δόθηκε
////////////////////////////////////
var ZoomScaleX,ZoomScaleY,ZoomScale:Real;
begin
  ZoomScaleX := mReflImage.ClientWidth / mReflImage.Picture.Width;
  ZoomScaleY := mReflImage.ClientHeight / mReflImage.Picture.Height;
  if ZoomScaleX<ZoomScaleY then
    ZoomScale := ZoomScaleX
  else
    ZoomScale := ZoomScaleY;
  if ZoomScale > 1 then ZoomScale := 1;
  Result := Round((mReflImage.Picture.Width*ZoomScale * MapX) / 1000);
  if Result > Round(mReflImage.Picture.Width*ZoomScale) then
    Result := Round(mReflImage.Picture.Width*ZoomScale);
  if Result < 0 then Result := 0;
end;

function TMapImageConv.MapToImageY(MapY:Integer): Integer;
////////////////////////////////////
/// Όνομα: MapToImageY
/// Τύπος: Public Function
/// Επιστρέφει:
/// Integer - Κατακόρυφη Θέση σε pixels
/// Παράμετροι:
/// MapY: Integer - Κατακόρυφη Θέση σε κανονικοποιημένες συντεταγμένες
/// Περιγραφή:
/// Η function αυτή υπολογίζει την θέση σε πραγματικά pixels που αντιστοιχεί
/// στην κανονικοποιημένη θέση που δόθηκε
////////////////////////////////////
var ZoomScaleX,ZoomScaleY,ZoomScale:Real;
begin
  ZoomScaleX := mReflImage.ClientWidth / mReflImage.Picture.Width;
  ZoomScaleY := mReflImage.ClientHeight / mReflImage.Picture.Height;
  if ZoomScaleX<ZoomScaleY then
    ZoomScale := ZoomScaleX
  else
    ZoomScale := ZoomScaleY;
  if ZoomScale > 1 then ZoomScale := 1;
  Result := Round((mReflImage.Picture.Height*ZoomScale * MapY) / 1000);
  if Result > Round(mReflImage.Picture.Height*ZoomScale) then

```

```

    Result := Round(mReflImage.Picture.Height*ZoomScale);
    if Result < 0 then Result := 0;
end;

```

```

function TMapImageConv.ImageToMapX(ImageX:Integer): Integer;
////////////////////////////////////////////////////////////////////
/// Όνομα: ImageToMapX
/// Τύπος: Public Function
/// Επιστρέφει:
/// Integer - Οριζόντια Θέση σε κανονικοποιημένες συντεταγμένες
/// Παράμετροι:
/// ImageX: Integer - Οριζόντια Θέση σε pixels
/// Περιγραφή:
/// Η function αυτή υπολογίζει την κανονικοποιημένη θέση που αντιστοιχεί
/// στην θέση σε πραγματικά pixels που δόθηκε
////////////////////////////////////////////////////////////////////
var ZoomScaleX,ZoomScaleY,ZoomScale:Real;
begin
    ZoomScaleX := mReflImage.ClientWidth / mReflImage.Picture.Width;
    ZoomScaleY := mReflImage.ClientHeight / mReflImage.Picture.Height;
    if ZoomScaleX<ZoomScaleY then
        ZoomScale := ZoomScaleX
    else
        ZoomScale := ZoomScaleY;
    if ZoomScale > 1 then ZoomScale := 1;
    Result := Round((1000 * ImageX) / (mReflImage.Picture.Width*ZoomScale));
    if Result > 1000 then Result := 1000;
    if Result < 0 then Result := 0;
end;

```

```

function TMapImageConv.ImageToMapY(ImageY:Integer): Integer;
////////////////////////////////////////////////////////////////////
/// Όνομα: ImageToMapY
/// Τύπος: Public Function
/// Επιστρέφει:
/// Integer - Κατακόρυφη Θέση σε κανονικοποιημένες συντεταγμένες
/// Παράμετροι:
/// ImageY: Integer - Κατακόρυφη Θέση σε pixels
/// Περιγραφή:
/// Η function αυτή υπολογίζει την κανονικοποιημένη θέση που αντιστοιχεί
/// στην θέση σε πραγματικά pixels που δόθηκε
////////////////////////////////////////////////////////////////////
var ZoomScaleX,ZoomScaleY,ZoomScale:Real;
begin
    ZoomScaleX := mReflImage.ClientWidth / mReflImage.Picture.Width;
    ZoomScaleY := mReflImage.ClientHeight / mReflImage.Picture.Height;
    if ZoomScaleX<ZoomScaleY then
        ZoomScale := ZoomScaleX
    else
        ZoomScale := ZoomScaleY;
    if ZoomScale > 1 then ZoomScale := 1;
    Result := Round((1000 * ImageY) / (mReflImage.Picture.Height*ZoomScale));
    if Result > 1000 then Result := 1000;
    if Result < 0 then Result := 0;
end;

```

```

Function ExecuteFile(FileName:string):THandle;
////////////////////////////////////////////////////////////////////
/// Όνομα: ExecuteFile
/// Τύπος: Public Function
/// Επιστρέφει:

```

```

/// THandle - To Windows Handle για την εφαρμογή που θα εκτελεστεί
/// Παράμετροι:
/// FileName: String - Το αρχείο που θέλουμε να ανοίξουν τα Windows
/// Περιγραφή:
/// Η function αυτή χρησιμοποιείται από το υπόλοιπο πρόγραμμα όταν θέλουμε
/// να εκτελεστεί ένα εξωτερικό αρχείο. Καλεί την ExecuteFileEx με τις
/// κατάλληλες παραμέτρους ώστε απλά να ανοιχτεί το αρχείο, όπως αν
/// έκανε ο χρήστης διπλό κλικ σε αυτό από τα Windows (τύπος εκτέλεσης 'Open',
/// σε κατάσταση SW_NORMAL)
////////////////////////////////////
begin
  Result := ExecuteFileEx('Open',Filename,",ExtractFileDir(Filename),SW_NORMAL);
end;

Function ExecuteFileEx(const Operation, FileName, Params, DefaultDir: string;
  ShowCmd: Integer): THandle;
////////////////////////////////////
/// Όνομα: ExecuteFileEx
/// Τύπος: Public Function
/// Επιστρέφει:
/// THandle - To Windows Handle για την εφαρμογή που θα εκτελεστεί
/// Παράμετροι:
/// const Operation: string - Τύπος εκτέλεσης του αρχείου
/// const FileName: string - Όνομα του αρχείου
/// const Params: string - Παράμετροι
/// const DefaultDir: string - Ο κατάλογος στον οποίο θέλουμε να εκτελεστεί
/// ShowCmd: Integer - κατάσταση εκτέλεσης
/// Περιγραφή:
/// Η function αυτή αποτελεί wrapper για την Win32 function ShellExecute
/// που ορίζεται στο ShellAPI
////////////////////////////////////
var
  zOperation : array[0..79] of Char;
  zFileName   : array[0..79] of Char;
  zParams     : array[0..79] of Char;
  zDir        : array[0..79] of Char;
begin
  Result := ShellExecute(Application.Handle,

    StrPCopy(zOperation, Operation),

    StrPCopy(zFileName, FileName),

    StrPCopy(zParams, Params),

    StrPCopy(zDir, DefaultDir), ShowCmd);

  // αποτέλεσμα <= 32 σημαίνει λάθος κατά την ShellExecute
  if Result <= 32 then
    MessageDlg('ERROR - Can't ' + Operation + ' file ' +
      FileName, mtError, [mbOK],
0);
end;

function TUser.IsStudent: boolean;
////////////////////////////////////
/// Όνομα: IsStudent
/// Τύπος: Public Function
/// Επιστρέφει:
/// Boolean - True = ο τρέχον χρήστης είναι μαθητής
///          - False = ο τρέχον χρήστης δεν είναι μαθητής

```

```

/// Παράμετροι:
/// (καμία)
/// Περιγραφή:
/// Η function αυτή χρησιμοποιείται για να ελεγχθεί αν ο χρήστης είναι μαθητής
////////////////////////////////////
begin
  IsStudent := (mUserType = 0);
end;

function TUser.IsTeacher: boolean;
////////////////////////////////////
/// Όνομα: IsTeacher
/// Τύπος: Public Function
/// Επιστρέφει:
/// Boolean - True = ο τρέχον χρήστης είναι καθηγητής
/// - False = ο τρέχον χρήστης δεν είναι καθηγητής
/// Παράμετροι:
/// (καμία)
/// Περιγραφή:
/// Η function αυτή χρησιμοποιείται για να ελεγχθεί αν ο χρήστης είναι καθηγητής
////////////////////////////////////
begin
  IsTeacher := (mUserType = 1) or (mUserType = 2);
end;

function TUser.IsAdmin: boolean;
////////////////////////////////////
/// Όνομα: IsAdmin
/// Τύπος: Public Function
/// Επιστρέφει:
/// Boolean - True = ο τρέχον χρήστης έχει δικαιώματα διαχειριστή
/// - False = ο τρέχον χρήστης δεν έχει δικαιώματα διαχειριστή
/// Παράμετροι:
/// (καμία)
/// Περιγραφή:
/// Η function αυτή χρησιμοποιείται για να ελεγχθεί αν ο χρήστης
/// έχει δικαιώματα διαχειριστή
////////////////////////////////////
begin
  IsAdmin := (mUserType = 2);
end;

procedure TUser.Store;
////////////////////////////////////
/// Όνομα: Store
/// Τύπος: Public Procedure
/// Παράμετροι:
/// (καμία)
/// Περιγραφή:
/// Η Store αποθηκεύει στην βάση τυχόν αλλαγές που έγιναν στο αντικείμενο TUser,
/// δηλαδή στα στοιχεία του τρέχοντος χρήστη
////////////////////////////////////
begin
  with GeoData do
  try
    tbUser.Open;
    if tbUser.Locate('ID',mID,[]) then
      begin //ο χρήστης βρέθηκε
        tbUser.Edit;
      end else
      begin //ο χρήστης δεν βρέθηκε -- σφάλμα

```

```

    raise EDatabaseError.Create('ΣΦΑΛΜΑ ΔΕΔΟΜΕΝΩΝ: Ο τρέχων χρήστης δεν βρέθηκε
στην βάση!');
end;
tbUserName.AsString := mUserName;
tbUserEncrPassword.AsString := mEncrPassword;
tbUserType.AsInteger := mUserType;
tbUserLastLesson.AsInteger := mLastLesson;
tbUser.Post;
tbUser.Close;
except
on E:Exception do
begin
    ShowMessage('Αποτυχία πρόσβασης στο αρχείο χρηστών!' + chr(13) +
        'Παρακαλούμε ξαναπροσπαθήστε. Εάν το πρόβλημα επιμένει' + chr(13) +
        'κλείστε το πρόγραμμα και εκκινήστε ξανά' + chr(13) + chr(13) +
        'Μήνυμα λάθους:' + chr(13) +
        ''' + e.Message + ''');
    if tbUser.Active then tbUser.Close;
end;
end;

end;

function TUser.Login(aUsername,aEncrPassword:string):boolean;
////////////////////////////////////
/// Όνομα: Login
/// Τύπος: Public Function
/// Επιστρέφει:
/// Boolean - True = ο χρήστης συνδέθηκε επιτυχώς
///           - False = αποτυχία σύνδεσης χρήστη
/// Παράμετροι:
/// aUsername: string - Το όνομα χρήστη
/// aEncrPassword: string - Το συνθηματικό του χρήστη, κρυπτογραφημένο
/// Περιγραφή:
/// Η Login επιχειρεί την σύνδεση ενός χρήστη στο πρόγραμμα, χρησιμοποιώντας
/// τα στοιχεία που δίνονται (aUsername, aEncrPassword). Αν η σύνδεση είναι
/// επιτυχής, διαβάζονται και τα υπόλοιπα δεδομένα του χρήστη στο αντικείμενο
/// της κλάσης TUser απ' όπου στην συνέχεια τα χρησιμοποιεί το υπόλοιπο
/// πρόγραμμα.
////////////////////////////////////
begin
    Login := False;
    with GeoData do
    try
        tbUser.Open;
        if tbUser.Locate('Name',aUsername,[loCaseInsensitive]) then
            begin
                mUserName := tbUserName.AsString;
                mEncrPassword := tbUserEncrPassword.AsString;
                mUserType := tbUserType.AsInteger;
                mLastLesson := tbUserLastLesson.AsInteger;
                mID := tbUserID.AsInteger;
                Login := (mEncrPassword = aEncrPassword);
            end;
            tbUser.Close;
        except
        on E:Exception do
            begin
                ShowMessage('Αποτυχία πρόσβασης στο αρχείο χρηστών!' + chr(13) +
                    'Παρακαλούμε ξαναπροσπαθήστε. Εάν το πρόβλημα επιμένει' + chr(13) +
                    'κλείστε το πρόγραμμα και εκκινήστε ξανά' + chr(13) + chr(13) +

```



```

        'Μήνυμα λάθους:'+chr(13)+
        ""+e.Message+"";
    if tbUser.Active then tbUser.Close;
end;
end;
end;

function TUser.ChangePassword(aEncrOldPassword,aEncrNewPassword:string):boolean;
////////////////////////////////////////////////////////////////////
// Όνομα: ChangePassword
// Τύπος: Public Function
// Επιστρέφει:
//   Boolean - True = το συνθηματικό άλλαξε επιτυχώς
//             - False = το συνθηματικό δεν άλλαξε
// Παράμετροι:
//   aEncrOldPassword: string - Το παλιό συνθηματικό του χρήστη, κρυπτογραφημένο
//   aEncrNewPassword: string - Το συνθηματικό του χρήστη, κρυπτογραφημένο
// Περιγραφή:
//   Η ChangePassword αλλάζει το password του τρέχοντος χρήστη, ελέγχοντας
//   πρώτα αν το παλιό είναι σωστό
////////////////////////////////////////////////////////////////////
begin
    ChangePassword := False;
    if (mEncrPassword = aEncrOldPassword) then
        begin
            mEncrPassword := aEncrNewPassword;
            ChangePassword := True;
        end;
end;

constructor TSettingsReader.Create(aDS:TDataSet);
////////////////////////////////////////////////////////////////////
// Όνομα: Create
// Τύπος: Constructor
// Παράμετροι:
//   aDS:TDataSet - το DataSet του πίνακα των ρυθμίσεων
// Περιγραφή:
//   Η κλάση TSettingsReader αναλαμβάνει να διαβάσει και να γράψει τις ρυθμίσεις
//   του προγράμματος από και προς την βάση.
////////////////////////////////////////////////////////////////////
begin
    inherited Create;
    mDS := aDS;
    mDS.Open;
end;

destructor TSettingsReader.Destroy;
////////////////////////////////////////////////////////////////////
// Όνομα: Destroy
// Τύπος: Destructor
// Παράμετροι:
//   (καμία)
// Περιγραφή:
//   Πριν καταστραφεί, η κλάση TSettingsReader κλείνει το Dataset που
//   χρησιμοποίησε
////////////////////////////////////////////////////////////////////
begin
    mDS.Close;
    inherited Destroy;
end;

```

```

function TSettingsReader.ReadSettingStr(SettingKey:String;DefValue:String):String;
////////////////////////////////////////////////////////////////////
/// Όνομα: ReadSettingStr
/// Τύπος: Public Function (overloaded)
/// Επιστρέφει:
///   String - Η ρύθμιση
/// Παράμετροι:
///   SettingKey: String - Το κλειδί της ρύθμισης που επιθυμείται να διαβαστεί
///   DefValue: String - Η προεπιλεγμένη τιμή της ρύθμισης
/// Περιγραφή:
///   Η function ReadSettingStr διαβάζει μία ρύθμιση τύπου string από τον
///   πίνακα των ρυθμίσεων με κλειδί την παράμετρο SettingKey και την επιστρέφει.
///   Αν δεν βρέθηκε καταχώρηση στον πίνακα ρυθμίσεων με αυτό το κλειδί,
///   επιστρέφει την προεπιλεγμένη τιμή DefValue
////////////////////////////////////////////////////////////////////
begin
  if mDS.Locate('ID',SettingKey,[loCaseInsensitive]) then
    begin
      Result := mDS.FieldByName('Setting').AsString;
    end else
    begin
      Result := DefValue;
    end;
end;

function TSettingsReader.ReadSettingStr(SettingKey:String):String;
////////////////////////////////////////////////////////////////////
/// Όνομα: ReadSettingStr
/// Τύπος: Public Function (overloaded)
/// Επιστρέφει:
///   String - Η ρύθμιση
/// Παράμετροι:
///   SettingKey: String - Το κλειδί της ρύθμισης που επιθυμείται να διαβαστεί
/// Περιγραφή:
///   Η function ReadSettingStr διαβάζει μία ρύθμιση τύπου string από τον
///   πίνακα των ρυθμίσεων με κλειδί την παράμετρο SettingKey και την επιστρέφει.
///   Αν δεν βρέθηκε καταχώρηση στον πίνακα ρυθμίσεων με αυτό το κλειδί,
///   επιστρέφει την τιμή ""
////////////////////////////////////////////////////////////////////
begin
  result := ReadSettingStr(SettingKey,"");
end;

function TSettingsReader.ReadSettingInt(SettingKey:String;DefValue:Integer):Integer;
////////////////////////////////////////////////////////////////////
/// Όνομα: ReadSettingInt
/// Τύπος: Public Function (overloaded)
/// Επιστρέφει:
///   Integer - Η ρύθμιση
/// Παράμετροι:
///   SettingKey: String - Το κλειδί της ρύθμισης που επιθυμείται να διαβαστεί
///   DefValue: Integer - Η προεπιλεγμένη τιμή της ρύθμισης
/// Περιγραφή:
///   Η function ReadSettingInt διαβάζει μία ρύθμιση τύπου Integer από τον
///   πίνακα των ρυθμίσεων με κλειδί την παράμετρο SettingKey και την επιστρέφει.
///   Αν δεν βρέθηκε καταχώρηση στον πίνακα ρυθμίσεων με αυτό το κλειδί,
///   επιστρέφει την προεπιλεγμένη τιμή DefValue
////////////////////////////////////////////////////////////////////
begin
  result := StrToInt(ReadSettingStr(SettingKey,IntToStr(DefValue)));
end;

```

```

end;

function TSettingsReader.ReadSettingInt(SettingKey:String):Integer;
////////////////////////////////////////////////////////////////////
/// Όνομα: ReadSettingInt
/// Τύπος: Public Function (overloaded)
/// Επιστρέφει:
/// Integer - Η ρύθμιση
/// Παράμετροι:
/// SettingKey: String - Το κλειδί της ρύθμισης που επιθυμείται να διαβαστεί
/// Περιγραφή:
/// Η function ReadSettingInt διαβάζει μία ρύθμιση τύπου Integer από τον
/// πίνακα των ρυθμίσεων με κλειδί την παράμετρο SettingKey και την επιστρέφει.
/// Αν δεν βρέθηκε καταχώρηση στον πίνακα ρυθμίσεων με αυτό το κλειδί,
/// επιστρέφει την τιμή 0
////////////////////////////////////////////////////////////////////
begin
  result := ReadSettingInt(SettingKey,0);
end;

function TSettingsReader.ReadSettingReal(SettingKey:String;DefValue:Real):Real;
////////////////////////////////////////////////////////////////////
/// Όνομα: ReadSettingReal
/// Τύπος: Public Function (overloaded)
/// Επιστρέφει:
/// Real - Η ρύθμιση
/// Παράμετροι:
/// SettingKey: String - Το κλειδί της ρύθμισης που επιθυμείται να διαβαστεί
/// DefValue: Real - Η προεπιλεγμένη τιμή της ρύθμισης
/// Περιγραφή:
/// Η function ReadSettingReal διαβάζει μία ρύθμιση τύπου Real από τον
/// πίνακα των ρυθμίσεων με κλειδί την παράμετρο SettingKey και την επιστρέφει.
/// Αν δεν βρέθηκε καταχώρηση στον πίνακα ρυθμίσεων με αυτό το κλειδί,
/// επιστρέφει την προεπιλεγμένη τιμή DefValue
////////////////////////////////////////////////////////////////////
begin
  result := StrToFloat(ReadSettingStr(SettingKey,FloatToStr(DefValue)));
end;

function TSettingsReader.ReadSettingReal(SettingKey:String):Real;
////////////////////////////////////////////////////////////////////
/// Όνομα: ReadSettingReal
/// Τύπος: Public Function (overloaded)
/// Επιστρέφει:
/// Real - Η ρύθμιση
/// Παράμετροι:
/// SettingKey: String - Το κλειδί της ρύθμισης που επιθυμείται να διαβαστεί
/// Περιγραφή:
/// Η function ReadSettingReal διαβάζει μία ρύθμιση τύπου Real από τον
/// πίνακα των ρυθμίσεων με κλειδί την παράμετρο SettingKey και την επιστρέφει.
/// Αν δεν βρέθηκε καταχώρηση στον πίνακα ρυθμίσεων με αυτό το κλειδί,
/// επιστρέφει την τιμή 0.0
////////////////////////////////////////////////////////////////////
begin
  result := ReadSettingReal(SettingKey,0.0);
end;

procedure TSettingsReader.WriteSettingStr(SettingKey: String; Value: String);
////////////////////////////////////////////////////////////////////
/// Όνομα: WriteSettingStr
/// Τύπος: Public Procedure

```

```

/// Παράμετροι:
///   SettingKey: String - Το κλειδί της ρύθμισης που επιθυμείται να γραφτεί
///   Value: String - Η ρύθμιση
/// Περιγραφή:
///   Η Procedure WriteSettingStr γράφει μία ρύθμιση τύπου string στον
///   πίνακα των ρυθμίσεων με τιμή την παράμετρο Value και κλειδί
///   την παράμετρο SettingKey.
////////////////////////////////////
begin
  if mDS.Locate('ID',SettingKey,[!oCaseInsensitive]) then
    begin
      mDS.Edit;
    end else
    begin
      mDS.Append;
      mDS.FieldByName('ID').AsString := AnsiUpperCase(SettingKey);
    end;
    mDS.FieldByName('Setting').AsString := Value;
    mDS.Post;
  end;

procedure TSettingsReader.WriteSettingInt(SettingKey: String; Value: Integer);
////////////////////////////////////
/// Όνομα: WriteSettingInt
/// Τύπος: Public Procedure
/// Παράμετροι:
///   SettingKey: String - Το κλειδί της ρύθμισης που επιθυμείται να γραφτεί
///   Value: String - Η ρύθμιση
/// Περιγραφή:
///   Η Procedure WriteSettingStr γράφει μία ρύθμιση τύπου Integer στον
///   πίνακα των ρυθμίσεων με τιμή την παράμετρο Value και κλειδί
///   την παράμετρο SettingKey.
////////////////////////////////////
begin
  WriteSettingStr(SettingKey, IntToStr(Value));
end;

procedure TSettingsReader.WriteSettingReal(SettingKey: String; Value: Real);
////////////////////////////////////
/// Όνομα: WriteSettingReal
/// Τύπος: Public Procedure
/// Παράμετροι:
///   SettingKey: String - Το κλειδί της ρύθμισης που επιθυμείται να γραφτεί
///   Value: String - Η ρύθμιση
/// Περιγραφή:
///   Η Procedure WriteSettingStr γράφει μία ρύθμιση τύπου Real στον
///   πίνακα των ρυθμίσεων με τιμή την παράμετρο Value και κλειδί
///   την παράμετρο SettingKey.
////////////////////////////////////
begin
  WriteSettingStr(SettingKey, FloatToStr(Value));
end;

function HashOf(const Key: string): Integer;
////////////////////////////////////
/// Όνομα: HashOf
/// Τύπος: Public Function
/// Επιστρέφει:
///   Integer - Hash-code του κειμένου που δόθηκε
/// Παράμετροι:
///   const Key: string - Το κείμενο από το οποίο θέλουμε να παράγουμε hash-code

```

```

/// Περιγραφή:
/// Η HashOf είναι μια απλή hashing function όπως υλοποιείται από την Delphi
/// στην unit IniFiles, κλάση TStringHash. Το hash του συνθηματικού
/// χρησιμοποιείται για να δώσει αρχική τιμή στη γεννήτρια τυχαίων
/// αριθμών η οποία χρησιμοποιείται στη συνέχεια για την κρυπτογράφηση
/// του συνθηματικού.
////////////////////////////////////
var
  I: Integer;
begin
  Result := 0;
  for I := 1 to Length(Key) do
    Result := ((Result shl 2) or (Result shr (SizeOf(Result) * 8 - 2))) xor
      Ord(Key[I]);
end;

function RNG:Byte;
////////////////////////////////////
/// Όνομα: RNG
/// Τύπος: Public Function
/// Επιστρέφει:
/// Byte - Τυχαίος αριθμός από 0 έως 255
/// Παράμετροι:
/// (καμία)
/// Περιγραφή:
/// Η RNG είναι μία υλοποίηση του Linear congruential generator, μίας
/// δημοφιλούς γεννήτριας τυχαίων αριθμών, με τις παραμέτρους που χρησιμοποιούνται
/// και στην υλοποίηση της rand() στην Ansi C.
/// Από το αποτέλεσμα κρατιούνται μόνο τα bits 29-22 γιατί μας ενδιαφέρουν
/// μόνο αριθμοί των 8 bits, και σύμφωνα με την τεκμηρίωση του LCG τα bits
/// υψηλής τάξης παρουσιάζουν την μεγαλύτερη τυχαιότητα
////////////////////////////////////
var i:integer;
begin
  RNGSeed := ($41C64E6D * RNGSeed + 12345);
  i := (RNGSeed shr 22) and $FF; //μας ενδιαφέρουν μόνο τα bits 29-22
  RNG := lo(i);
end;

Function Encrypted(aPlainText:String):String;
////////////////////////////////////
/// Όνομα: Encrypted
/// Τύπος: Public Function
/// Επιστρέφει:
/// String - το κρυπτογραφημένο κείμενο, σαν σειρά δεκαεξαδικών αριθμών
/// Παράμετροι:
/// aPlainText: String - το κείμενο προς κρυπτογράφηση
/// Περιγραφή:
/// Η λειτουργία της Encrypted είναι η εξής:
/// Υπολογίζεται ένα hash-code του αρχικού κειμένου, και χρησιμοποιείται
/// ως αρχική τιμή της γεννήτριας RNG.
/// Ο κάθε χαρακτήρας του κειμένου γίνεται XOR με διαδοχικές τιμές
/// της RNG και προκύπτει το κρυπτογραφημένο κείμενο
/// Αυτό τελικά επιστρέφεται ως σειρά δεκαεξαδικών αριθμών, για να
/// αποφευχθούν προβλήματα κατά την εγγραφή του string στην βάση, λόγω
/// των ειδικών χαρακτήρων που μπορεί να εμφανιστούν.
////////////////////////////////////
var i:integer;
    CryptText:String;
    NextChar:byte;
begin

```

```

for i:= length(CryptText) + 1 to 25 do
begin
  aPlainText := aPlainText + chr(255);
end;
RNGSeed := HashOf(aPlainText);
CryptText := "";
for i:= 1 to 25 do
begin
  NextChar := (ord(aPlainText[i]) XOR RNG);
  CryptText := CryptText + IntToHex(NextChar,2);
end;
Encrypted := CryptText;
end;

procedure SaveSettings;
////////////////////////////////////////////////////////////////
/// Όνομα: SaveSettings
/// Τύπος: Public procedure
/// Παράμετροι:
///   (καμία)
/// Περιγραφή:
///   Η procedure SaveSettings χρησιμοποιεί τις WriteSettingInt, WriteSettingReal
///   και WriteSettingStr για να αποθηκεύσει όλες τις ρυθμίσεις του προγράμματος
///   στην βάση
////////////////////////////////////////////////////////////////
var mSettingsReader: TSettingsReader;
begin
  mSettingsReader:= TSettingsReader.Create(GeoData.tbSettings);

  mSettingsReader.WriteSettingInt('TestTime',TotalTestTime);
  mSettingsReader.WriteSettingInt('TriesMult',MaxTriesMultChoice);
  mSettingsReader.WriteSettingInt('TriesMap',MaxAttemptsMap);
  mSettingsReader.WriteSettingReal('MaxGrade',MaxGrade);
  mSettingsReader.WriteSettingReal('PassGrd',PassingGrade);
  mSettingsReader.WriteSettingStr('GrdFmt',GradesFormat);

  mSettingsReader.Destroy;
end;

procedure LoadSettings;
////////////////////////////////////////////////////////////////
/// Όνομα: LoadSettings
/// Τύπος: Public procedure
/// Παράμετροι:
///   (καμία)
/// Περιγραφή:
///   Η procedure LoadSettings χρησιμοποιεί τις ReadSettingInt, ReadSettingReal
///   και ReadSettingStr για να διαβάσει όλες τις ρυθμίσεις του προγράμματος από
///   την βάση
////////////////////////////////////////////////////////////////
var mSettingsReader: TSettingsReader;
begin
  mSettingsReader := TSettingsReader.Create(GeoData.tbSettings);

  TotalTestTime := mSettingsReader.ReadSettingInt('TestTime',TotalTestTime);
  MaxTriesMultChoice := mSettingsReader.ReadSettingInt('TriesMult',MaxTriesMultChoice);
  MaxAttemptsMap := mSettingsReader.ReadSettingInt('TriesMap',MaxAttemptsMap);
  MaxGrade := mSettingsReader.ReadSettingReal('MaxGrade',MaxGrade);
  PassingGrade := mSettingsReader.ReadSettingReal('PassGrd',PassingGrade);
  GradesFormat := mSettingsReader.ReadSettingStr('GrdFmt',GradesFormat);

```

```
mSettingsReader.Destroy;
end;
```

```
function UpperCaseNoStress(const s:string): String;
////////////////////////////////////////////////////////////////
/// Όνομα: UpperCaseNoStress
/// Τύπος: Public Function
/// Επιστρέφει:
/// String - αντίγραφο του s, με τα γράμματα γυρισμένα σε κεφαλαία
/// Παράμετροι:
/// const s: string - το κείμενο που επιθυμούμε μετατροπή σε κεφαλαία
/// Περιγραφή:
/// Η UpperCaseNoStress κάνει ότι και η AnsiUpperCase αλλά χωρίς να διατηρεί
/// τους τόνους στα κεφαλαία. Δηλαδή για είσοδο 'Ελλάδα', η AnsiUpperCase
/// θα έδινε αποτέλεσμα 'ΕΛΛΑΔΑ', ενώ η UpperCaseNoStress θα δώσει
/// 'ΕΛΛΑΔΑ'
////////////////////////////////////////////////////////////////
```

```
var i:integer;
begin
  Result := AnsiUpperCase(s);
  for i := 1 to length(Result) do
    begin
      case result[i] of
        'A': result[i] := 'A';
        'E': result[i] := 'E';
        'H': result[i] := 'H';
        'I': result[i] := 'I';
        'İ': result[i] := 'I';
        'O': result[i] := 'O';
        'Y': result[i] := 'Y';
        'ÿ': result[i] := 'Y';
        'Ω': result[i] := 'Ω';
      end;
    end;
  end;
end;
```

```
function FindWordMatch(strWord1, strWord2: String): Integer;
////////////////////////////////////////////////////////////////
/// Όνομα: FindWordMatch
/// Τύπος: Public Function
/// Επιστρέφει:
/// Integer - βαθμός ομοιότητας των λέξεων (0 έως 100)
/// Παράμετροι:
/// strWord1, strWord2: String - Οι δύο λέξεις προς σύγκριση
/// Περιγραφή:
/// Η FindWordMatch υπολογίζει τον βαθμό ομοιότητας δύο λέξεων
/// Η λειτουργία της FindWordMatch είναι η εξής:
/// Κάθε γράμμα της μικρότερης από τις δύο λέξεις συγκρίνεται με το
/// αντίστοιχο γράμμα της άλλης λέξης. Αν το γράμμα είναι το ίδιο
/// προστίθεται 10 σε έναν αθροιστή. Αν δεν είναι το ίδιο αλλά είναι ίδιο με
/// ένα γειτονικό, προστίθεται 5 στον αθροιστή. Το τελικό αποτέλεσμα
/// της σύγκρισης είναι το εκατοστιαίο ποσοστό της τελικής τιμής του αθροιστή προς
/// τον αριθμό των γραμμάτων της μεγαλύτερης λέξης επί 10.
////////////////////////////////////////////////////////////////
var i:integer; s:string;
    iSum:integer; iMax:integer;
begin
  strWord1 := Trim(UpperCaseNoStress(strWord1));
  strWord2 := Trim(UpperCaseNoStress(strWord2));
  // κράτησε την πιο μικρή από τις δύο λέξεις στην strWord1
  if length(strWord1) > length(strWord2) then
```


```
begin
  s := strWord1;
  strWord1 := strWord2;
  strWord2 := s;
end;
iSum := 0;
for i := 1 to length(strWord1) do
begin
  if strWord2[i] = strWord1[i] then
    iSum := iSum + 10
  else
begin
  if (i > 1) and (strWord2[i-1] = strWord1[i]) then
    iSum := iSum + 5
  else
    if (i < length(strWord2)) and (strWord2[i+1] = strWord1[i]) then
      iSum := iSum + 5;
    end;
  end;
end;
iMax := length(strWord2) * 10;
FindWordMatch := ((iSum * 100) DIV iMax);
end;

end.
```

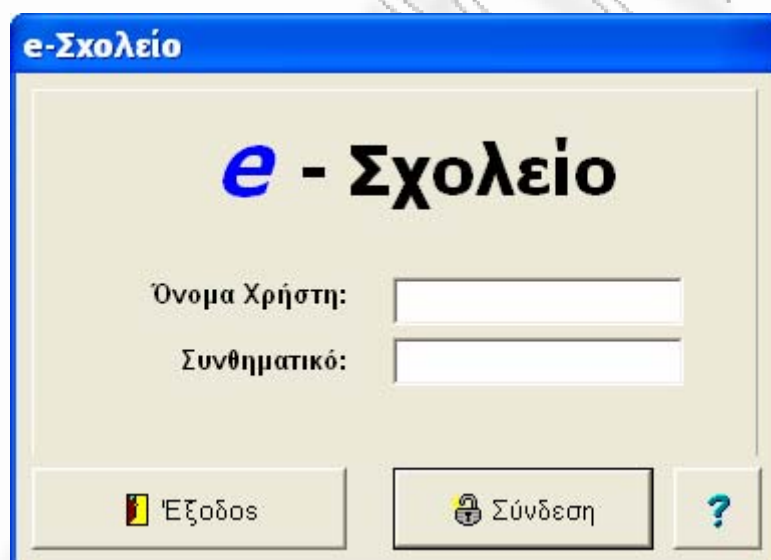
### **-Αρχική Οθόνη**

Λαμβάνοντας υπόψη τους χρήστες του συστήματος και τις ανάγκες τους, θεωρήθηκε καλό η εφαρμογή να διαθέτει κάποιες βασικές φόρμες. Η βασικές οθόνες σύνδεσης στο πρόγραμμα είναι οι εξής:





Μετά την επιλογή του μαθήματος εμφανίζεται η παρακάτω φόρμα:



Παρακάτω παρατίθεται ο κώδικας υλοποίησης των παραπάνω φορμών :

```

object LoginForm: TLoginForm
  Left = 413
  Top = 200
  BorderIcons = []
  BorderStyle = bsSingle
  Caption = 'e-#931#967#959#955#949#943#959'
  ClientHeight = 246
  ClientWidth = 380
  Color = clBtnFace
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -8
  Font.Name = 'MS Sans Serif'
  Font.Style = []
  OldCreateOrder = False
  Position = poScreenCenter
  OnCreate = FormCreate

```

```
OnShow = FormShow
DesignSize = (
  380
  246)
PixelsPerInch = 72
TextHeight = 13
object Bevel1: TBevel
  Left = 8
  Top = 8
  Width = 362
  Height = 183
  Style = bsRaised
end
object Label3: TLabel
  Left = 99
  Top = 18
  Width = 52
  Height = 52
  Caption = 'e '
  Font.Charset = GREEK_CHARSET
  Font.Color = clBlue
  Font.Height = -43
  Font.Name = 'Tahoma'
  Font.Style = [fsBold, fsItalic]
  ParentFont = False
end
object Label4: TLabel
  Left = 141
  Top = 26
  Width = 159
  Height = 42
  Caption = '- '#931#967#959#955#949#943#959
  Color = clBtnFace
  Font.Charset = GREEK_CHARSET
  Font.Color = clWindowText
  Font.Height = -35
  Font.Name = 'Tahoma'
  Font.Style = [fsBold]
  ParentColor = False
  ParentFont = False
end
object CoursePanel: TPanel
  Left = 9
  Top = 74
  Width = 360
  Height = 117
  BevelOuter = bvNone
  TabOrder = 0
  object Label5: TLabel
    Left = 7
    Top = 6
    Width = 109
    Height = 16
    Alignment = taRightJustify
    Caption = '#917#960#953#955#941#958#964#949' '#956#940#952#951#956#945':'
    Font.Charset = GREEK_CHARSET
    Font.Color = clWindowText
    Font.Height = -13
    Font.Name = 'Arial'
    Font.Style = [fsBold]
    ParentFont = False
```

```
end
object BtnLesson1: TButton
  Left = 7
  Top = 28
  Width = 109
  Height = 37
  Caption = #915#949#969#947#961#945#966#943#945
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -13
  Font.Name = 'MS Sans Serif'
  Font.Style = [fsBold]
  ParentFont = False
  TabOrder = 0
  OnClick = BtnLesson1Click
end
object BtnLesson2: TButton
  Tag = 1
  Left = 122
  Top = 28
  Width = 109
  Height = 37
  Caption = #921#963#964#959#961#943#945
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -13
  Font.Name = 'MS Sans Serif'
  Font.Style = [fsBold]
  ParentFont = False
  TabOrder = 1
  OnClick = BtnLesson1Click
end
object BtnLesson3: TButton
  Tag = 2
  Left = 237
  Top = 28
  Width = 109
  Height = 37
  Caption = #917#955#955#951#957#953#954#940
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -13
  Font.Name = 'MS Sans Serif'
  Font.Style = [fsBold]
  ParentFont = False
  TabOrder = 2
  OnClick = BtnLesson1Click
end
object BtnLesson4: TButton
  Tag = 3
  Left = 7
  Top = 71
  Width = 109
  Height = 37
  Caption = #902#955#955#959
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -13
  Font.Name = 'MS Sans Serif'
  Font.Style = [fsBold]
  ParentFont = False
```

```
    TabOrder = 3
    OnClick = BtnLesson1Click
end
object BtnLesson5: TButton
    Tag = 4
    Left = 122
    Top = 71
    Width = 109
    Height = 37
    Caption = #902#955#955#959
    Font.Charset = DEFAULT_CHARSET
    Font.Color = clWindowText
    Font.Height = -13
    Font.Name = 'MS Sans Serif'
    Font.Style = [fsBold]
    ParentFont = False
    TabOrder = 4
    OnClick = BtnLesson1Click
end
object BtnLesson6: TButton
    Tag = 5
    Left = 237
    Top = 71
    Width = 109
    Height = 37
    Caption = #902#955#955#959
    Font.Charset = DEFAULT_CHARSET
    Font.Color = clWindowText
    Font.Height = -13
    Font.Name = 'MS Sans Serif'
    Font.Style = [fsBold]
    ParentFont = False
    TabOrder = 5
    OnClick = BtnLesson1Click
end
end
object LoginPanel: TPanel
    Left = 9
    Top = 74
    Width = 360
    Height = 117
    BevelOuter = bvNone
    TabOrder = 1
    object Label2: TLabel
        Left = 73
        Top = 61
        Width = 83
        Height = 16
        Alignment = taRightJustify
        Caption = #931#965#957#952#951#956#945#964#953#954#972!:'
        Font.Charset = GREEK_CHARSET
        Font.Color = clWindowText
        Font.Height = -13
        Font.Name = 'Arial'
        Font.Style = [fsBold]
        ParentFont = False
    end
    object Label1: TLabel
        Left = 58
        Top = 28
        Width = 98
```

```
Height = 16
Alignment = taRightJustify
Caption = #908#957#959#956#945' '#935#961#942#963#964#951':
Font.Charset = GREEK_CHARSET
Font.Color = clWindowText
Font.Height = -13
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
end
object editPassword: TEdit
Left = 179
Top = 58
Width = 145
Height = 24
Font.Charset = GREEK_CHARSET
Font.Color = clWindowText
Font.Height = -13
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
PasswordChar = '*'
TabOrder = 1
end
object editUsername: TEdit
Left = 179
Top = 28
Width = 145
Height = 24
Font.Charset = GREEK_CHARSET
Font.Color = clWindowText
Font.Height = -13
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
TabOrder = 0
end
object btnCancel: TBitBtn
Left = 8
Top = 197
Width = 157
Height = 41
Anchors = [akLeft, akBottom]
Caption = #904#958#959#948#959#962
Font.Charset = DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 2
OnClick = btnCancelClick
Glyph.Data = {
76010000424D760100000000000076000000280000002000000010000000100
04000000000000010000120B0000120B000010000000000000000000000000
800000800000008080008000000080008000808000007F7F7F00BFBFBF000000
FF0000FF000000FFFF00FF000000FF00FF00FFFF0000FFFFFF00330000000000
03333377777777777777F333301BBBBBBBB033333773F3333337F3333011BBBBBBB
0333337F73F333337F3333011BBBBBB0333337F373F33337F333301110BBBBB
0333337F337F33337F33330110BBBBB0333337F337F33337F333301110BBBBB
```

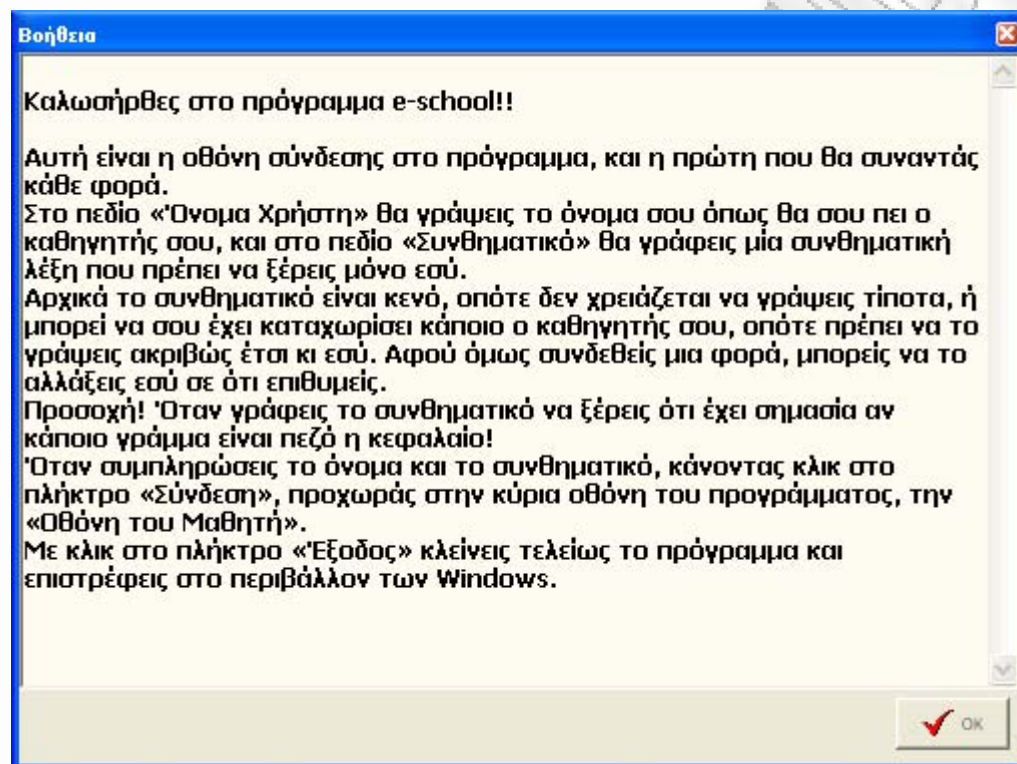
```
0333337F337F33337F333301110BBBBB033337F337F33337F333301110BBBBB
0333337F337F33337F333301110BBBBB033337F337FF3337F33330111B0BBBB
0333337F337733337F333301110BBBBB033337F337F33337F333301110BBBBB
0333337F3F7F33337F333301E10BBBBB033337F7F7F33337F333301EE0BBBBB
0333337F777FFFFF7F333300000000003333377777777777333}
NumGlyphs = 2
end
object btnLogin: TBitBtn
  Left = 188
  Top = 197
  Width = 130
  Height = 41
  Anchors = [akLeft, akBottom]
  Caption = #931#973#957#948#949#963#951
  Default = True
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -13
  Font.Name = 'MS Sans Serif'
  Font.Style = []
  ParentFont = False
  TabOrder = 3
  OnClick = btnLoginClick
  Glyph.Data = {
    76010000424D7601000000000000760000002800000020000000100000000100
    04000000000000010000120B0000120B000010000000000000000000000000
    800000800000008080008000000080008000808000007F7F7F00BFBFBF000000
    FF0000FF000000FFFF00FF000000FF00FF00FFFF0000FFFFFF00333333000003
    33333333F777773FF333333008888800333333377333F3773F3333077870787
    7033333733337F33373F3308888707888803337F33337F33337F330777880887
    7703337F33337FF3337F3308888000888803337F333777F3337F330777700077
    7703337F33377733337FB3088888888888033373FFFFFFFFFFF733B3000000000
    0033333777777777773333BBBB3333080333333333F3337F7F33BBBB707BB308
    03333333373F337F7F3333BB08033308033333337F7F337F7F333B3B08033308
    033333337F73FF737F33B33B778000877333333373F777337333333B308888880
    33333333373FFFFF73333333B330000033333333337777733333}
NumGlyphs = 2
end
object btnHelp: TBitBtn
  Left = 328
  Top = 197
  Width = 44
  Height = 41
  Hint = #914#959#942#952#949#953#945
  Anchors = [akRight, akBottom]
  ParentShowHint = False
  ShowHint = True
  TabOrder = 4
  OnClick = btnHelpClick
  Glyph.Data = {
    DE010000424DDE01000000000000760000002800000024000000120000000100
    04000000000068010000000000000000001000000000000000000000000000
    80000080000000808000800000008000800080800000C0C0C000808080000000
    FF0000FF000000FFFF00FF000000FF00FF00FFFF0000FFFFFF003333333336633
    333333333333FF3333333330000333333364463333333333333388F33333333
    0003333333E66433333333333338F38F3333333000033333333E66333333333
    33338FF8F3333333000033333333333333333333333338833333333300003333
    3333446333333333333333FF333333330000333333366643333333333333888
    F333333300003333333E6643333333333338F38F333333300003333333E6664
    333333333338F38F333333300003333333E6664333333333338F338F333333
    000033333333E666433333333338F338F3333300003333344333E666433333
```

```

333F338F338F3333000033336664333E664333333388F338F338F33300003333
E6664446664333338F38FFF8338F333000033333E666666666333338F33888
3338F3330000333333EE6666663333338FF33333383333000033333333EEEE
E33333333388FFFFF83333300003333333333333333333333333388888333333
0000}
NumGlyphs = 2
end
end

```

### -Φόρμα Βοήθειας



Παρακάτω παρατίθεται ο κώδικας φόρμας της Βοήθειας, ο οποίος είναι ο ίδιος για όλες τις φόρμες βοήθειας του προγράμματος :

*unit* HelpUnit;

*interface*

*uses*

*Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, Buttons, AppEvnts;*

*type*

*THelpForm = class(TForm)*

*TheMemo: TMemo;*

*btnOK: TBitBtn;*

*AppEvents: TApplicationEvents;*

*procedure AppEventsShortCut(var Msg: TWMKey; var Handled: Boolean);*

*function AppEventsHelp(Command: Word; Data: Integer;*

*var CallHelp: Boolean): Boolean;*

*procedure btnOKClick(Sender: TObject);*

*private*

*{ Private declarations }*

*procedure WriteHelpFileTemplate(aTextFilename: String; aForm: TForm);*

```

    procedure ReadHelpFile(aTextFilename: String);
    public
    { Public declarations }
    procedure ShowFormHelp(aForm:TForm);
    end;

var
    HelpForm: THelpForm;

implementation
{$R *.dfm}

////////////////////////////////////////////////////////////////
/// Όνομα: AppEventsHelp
/// Τύπος: Event Handler
/// Πυροδότηση:
///   Ενεργοποίηση της λήψης βοήθειας από οποιοδήποτε σημείο του προγράμματος
/// Παράμετροι:
///   Command: Word - Παράμετρος για το σύστημα βοήθειας των Windows (αγνοείται)
///   Data: Integer - Παράμετρος για το σύστημα βοήθειας των Windows (αγνοείται)
///   CallHelp:Boolean - Τίθεται True για να μην ενεργοποιηθεί το σύστημα
///                     βοήθειας των Windows
/// Περιγραφή:
///   Όταν ο χρήστης καλέσει βοήθεια από οποιοδήποτε σημείο του προγράμματος
///   το πρόγραμμα καλεί την procedure ShowFormHelp για να εμφανιστεί η βοήθεια
///   του προγράμματος, αντί να ενεργοποιηθεί το σύστημα βοήθειας των Windows
////////////////////////////////////////////////////////////////
function THelpForm.AppEventsHelp(Command: Word; Data: Integer;
    var CallHelp: Boolean): Boolean;
begin
    CallHelp := False;
    ShowFormHelp(Screen.ActiveForm);
    AppEventsHelp := True;
end;

////////////////////////////////////////////////////////////////
/// Όνομα: AppEventsShortCut
/// Τύπος: Event Handler
/// Πυροδότηση:
///   Κατά το πάτημα οποιουδήποτε πλήκτρου στο πληκτρολόγιο,
///   σε οποιοδήποτε σημείο του προγράμματος
/// Παράμετροι:
///   Msg: TWMKey - Το μήνυμα συστήματος τύπου TWMKey που περιγράφει
///                 ποιο πλήκτρο πατήθηκε
///   Handled:Boolean - Τίθεται True για να μην συνεχιστεί η επεξεργασία
///                     του μηνύματος
/// Περιγραφή:
///   Όταν ο χρήστης πατήσει το πλήκτρο F1 από οποιοδήποτε σημείο του προγράμματος,
///   το πρόγραμμα καλεί την procedure ShowFormHelp για να εμφανιστεί η βοήθεια
///   του προγράμματος. Για οποιοδήποτε άλλο πλήκτρο, η παράμετρος Handled παραμένει
///   False και συνεχίζεται κανονικά η επεξεργασία του μηνύματος πληκτρολόγησης
////////////////////////////////////////////////////////////////
procedure THelpForm.AppEventsShortCut(var Msg: TWMKey; var Handled: Boolean);
begin
    //Respond to F1 Click
    if msg.CharCode = VK_F1 then
        begin
            ShowFormHelp(Screen.ActiveForm);
            Handled := True;
        end;
end;
end;

```



```
////////////////////////////////////
/// Όνομα: btnOKClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnOK
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο OK, κλείνει η οθόνη βοήθειας
////////////////////////////////////
procedure THelpForm.btnOKClick(Sender: TObject);
begin
  ModalResult := mrOk;
end;

////////////////////////////////////
/// Όνομα: ReadHelpFile
/// Τύπος: Private Procedure
/// Παράμετροι:
/// aTextFilename: String - Το όνομα του αρχείου βοήθειας
/// Περιγραφή:
/// Η ReadHelpFile διαβάζει ένα αρχείο κειμένου που περιέχει το κείμενο βοήθειας
/// για την συγκεκριμένη οθόνη, και το εμφανίζει στο πλαίσιο TheMemo
////////////////////////////////////
procedure THelpForm.ReadHelpFile(aTextFilename: String);
var HelpTxt:TextFile;
    s:string;
begin
  TheMemo.Lines.Clear;
  AssignFile(HelpTxt,aTextFilename);
  Reset(HelpTxt);
  While not eof(HelpTxt) do
  begin
    Readln(HelpTxt,s);
    //Αγνοούνται οι γραμμές που ξεκινάν με '/'
    if (length(s) < 2) or (copy(s,1,2) <> '/') then
    begin
      TheMemo.Lines.Add(s);
    end;
  end;
  CloseFile(HelpTxt);
end;

////////////////////////////////////
/// Όνομα: WriteHelpFile
/// Τύπος: Private Procedure
/// Παράμετροι:
/// aTextFilename: String - Το όνομα του αρχείου βοήθειας
/// aForm: TForm - Η φόρμα για την οποία πρέπει να δημιουργηθεί αρχείο βοήθειας
/// Περιγραφή:
/// Η WriteHelpFile δημιουργεί ένα νέο αρχείο βοήθειας για την φόρμα που
/// είναι στην οθόνη. Γράφονται στην αρχή κάποιες οδηγίες προς τον
/// διαχειριστή του συστήματος που πρόκειται να γράψει το αρχείο βοήθειας
////////////////////////////////////
procedure THelpForm.WriteHelpFileTemplate(aTextFilename: String; aForm:TForm);
var HelpTxt:TextFile;
begin
  AssignFile(HelpTxt,aTextFilename);
  Rewrite(HelpTxt);
  Writeln(HelpTxt,'// Χρησιμοποιήστε αυτό το αρχείο για να εμφανιστεί βοήθεια');
```

```
WriteLn(HelpTxt, // στην οθόνη ' + aForm.Name + ' ('" + aForm.Caption + '"') του
προγράμματος. ');
WriteLn(HelpTxt, // οι γραμμές που ξεκινάν με "/" αγνοούνται');
CloseFile(HelpTxt);
end;
```

```
////////////////////////////////////
// Ονομα: ShowFormHelp
// Τύπος: Public Procedure
// Παράμετροι:
// aForm: TForm - Η φόρμα για την οποία πρέπει να εμφανιστεί βοήθεια
// Περιγραφή:
// Η ShowFormHelp κοιτάει αν υπάρχει αρχείο βοήθειας για την
// συγκεκριμένη φόρμα, και αν υπάρχει το εμφανίζει, αλλιώς δημιουργεί
// ένα κενό.
////////////////////////////////////
procedure TForm.ShowFormHelp(aForm:TForm);
var HelpTextFileName: String;
begin
  HelpTextFileName := ExtractFilePath(Application.ExeName)
    + 'HelpText\'+ aForm.Name + '.txt';
  If FileExists(HelpTextFileName) then
    begin
      ReadHelpFile(HelpTextFileName);
      if TheMemo.Lines.Count > 0 then Self.ShowModal;
    end else
      begin
        WriteHelpFileTemplate(HelpTextFileName,aForm);
      end;
  end;
end.
end.
```

### -Αρχική Οθόνη Μαθητή

**e-Σχολείο: Γεωγραφία - Μαθητής: TestStudent**

**Καλωσήρθες!**  
 Το τελευταίο μάθημα που παρακολούθησες ήταν το: **Νομός Άρτας**  
 Επέλεξε μάθημα από τη λίστα ή κάνοντας κλικ στον χάρτη και ύστερα κάνε κλικ στο κουμπί "Έναρξη μαθήματος" για να ξεκινήσει το μάθημα


**Νομός Άρτας**

Έναρξη μαθήματος!

Τεστ προόδου

Καρτέλα Αξιολόγησης

Αβλαγή συνηματικού



Εξοδος

Τελευταία παρακολούθηση: 20/11/2012  
 Τελευταίο τεστ προόδου: 20/11/2012 Βαθμολογία: 10.8

Παρακάτω παρατίθεται ο κώδικας υλοποίησης των φορμών για τις οθόνες του μαθητή :

*unit StudentUnit;*

*interface*

*uses*

*Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, ExtCtrls, StdCtrls, Buttons, GlobalsUnit;*

*type*

```
TStudentForm = class(TForm)
  btnStartLesson: TBitBtn;
  Label1: TLabel;
  lbLessonFixed: TLabel;
  lbLesson: TLabel;
  Label3: TLabel;
  Bevel1: TBevel;
  btnStartTest: TBitBtn;
  btnShowEvaluation: TBitBtn;
  btnClose: TBitBtn;
  cmbLessons: TComboBox;
  MapPanel: TPanel;
  MapImage: TImage;
  Panel1: TPanel;
  LessonLabel: TLabel;
  Label5: TLabel;
  Label6: TLabel;
  StudyDateLabel: TLabel;
  TestDateLabel: TLabel;
  TestGradeFixedLabel: TLabel;
  TestGradeLabel: TLabel;
```

```

btnChangePassword: TBitBtn;
btnHelp: TBitBtn;
procedure btnHelpClick(Sender: TObject);
procedure btnShowEvaluationClick(Sender: TObject);
procedure btnChangePasswordClick(Sender: TObject);
procedure MapImageMouseLeave(Sender: TObject);
procedure cmbLessonsChange(Sender: TObject);
procedure MapImageMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure MapPanelResize(Sender: TObject);
procedure FormDestroy(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure btnCloseClick(Sender: TObject);
procedure btnStartTestClick(Sender: TObject);
procedure btnStartLessonClick(Sender: TObject);
procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
private
  { Private declarations }
  LessonIDs: TStringList;
  Boxes: TMapBoxCollection;
  procedure BoxClick(aBoxIndex, aBoxTag: Integer);
  procedure BoxMouseOver(aBoxIndex, aBoxTag: Integer);
  procedure ReadDBBoxes;
public
  { Public declarations }
end;

var
  StudentForm: TStudentForm;

implementation

uses LoginUnit, ShowLessonUnit, ShowTestUnit, MapUnit, GeoDataUnit,
  ChangePasswordUnit, LessonToolTipUnit, StudentEvaluationUnit;

{$R *.dfm}

procedure TStudentForm.ReadDBBoxes;
////////////////////////////////////
  /// Όνομα: ReadDBBoxes
  /// Τύπος: Private Procedure
  /// Παράμετροι:
  ///   (καμία)
  /// Περιγραφή:
  ///   Η ReadDBBoxes διαβάζει τα τετράγωνα όλων των μαθημάτων από τη βάση και
  ///   τα τοποθετεί στο χάρτη, αλλά αόρατα. Για το κάθε τετράγωνο, τίθεται
  ///   η ιδιότητα Tag ίση με τον κωδικό του μαθήματος που ανήκει το τετράγωνο
  //////////////////////////////////////
begin
  if Assigned(Boxes) then
  begin
    Boxes.Clear;
    GeoData.tbMapBoxes.Filter := '';
    GeoData.tbMapBoxes.Filtered := False;
    GeoData.tbMapBoxes.Open;
    While not GeoData.tbMapBoxes.EOF do
    begin
      Boxes.NewBox(
        GeoData.tbMapBoxesTop.AsInteger,

```

```

GeoData.tbMapBoxesLeft.AsInteger,
GeoData.tbMapBoxesWidth.AsInteger,
GeoData.tbMapBoxesHeight.AsInteger,
GeoData.tbMapBoxesLessonID.AsInteger,False);
GeoData.tbMapBoxes.Next;
end;
GeoData.tbMapBoxes.Close;
end;
end;

procedure TStudentForm.BoxClick(aBoxIndex,aBoxTag:Integer);
////////////////////////////////////////////////////
/// Όνομα: BoxClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ κάποιου από τα τετράγωνα στον χάρτη
/// Παράμετροι:
/// aBoxIndex: Integer - ο αύξων αριθμός του τετραγώνου στη συλλογή
/// aBoxTag: Integer - η ιδιότητα Tag του τετραγώνου
/// Περιγραφή:
/// Όταν ο μαθητής κάνει κλικ πάνω στον χάρτη σε σημείο που υπάρχει τετράγωνο,
/// η παράμετρος aBoxTag δίνει το ID του αντίστοιχου μαθήματος. Ενημερώνεται
/// η property ItemIndex του cmbLessons ώστε να είναι επιλεγμένο και στο
/// combobox το ίδιο μάθημα, και καλείται η cmbLessonsChange ώστε να ενημερωθούν
/// τα διάφορα σχετικά με το μάθημα controls
////////////////////////////////////////////////////
begin
  cmbLessons.ItemIndex := LessonIDs.IndexOf(IntToStr(aBoxTag));
  cmbLessonsChange(self);
end;

procedure TStudentForm.BoxMouseOver(aBoxIndex,aBoxTag:Integer);
////////////////////////////////////////////////////
/// Όνομα: BoxMouseOver
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Μετακίνηση του δείκτη του ποντικιού πάνω από κάποιο τετράγωνο στον χάρτη
/// Παράμετροι:
/// aBoxIndex: Integer - ο αύξων αριθμός του τετραγώνου στη συλλογή
/// aBoxTag: Integer - η ιδιότητα Tag του τετραγώνου
/// Περιγραφή:
/// Όταν ο μαθητής κάνει μετακινεί το ποντίκι σε σημείο που υπάρχει τετράγωνο,
/// η παράμετρος aBoxTag δίνει το ID του αντίστοιχου μαθήματος, και εμφανίζεται
/// ένα ToolTip (φόρμα LessonToolTip) που γράφει διάφορα στατιστικά για το
/// συγκεκριμένο μάθημα
////////////////////////////////////////////////////
begin
  LessonToolTip.LessonLabel.Caption :=
  cmbLessons.Items[LessonIDs.IndexOf(IntToStr(aBoxTag))];
  if GeoData.tbStudentLessonProgress.Locate('LessonID',aBoxTag,[]) then
  begin
    if GeoData.tbStudentLessonProgressLessonDate.IsNull then
    begin
      LessonToolTip.StudyDateLabel.Caption := 'Δεν το παρακολούθησες!';
      LessonToolTip.StudyDateLabel.Font.Color := clRed;
    end else
    begin
      LessonToolTip.StudyDateLabel.Caption :=
      GeoData.tbStudentLessonProgressLessonDate.AsString;
      LessonToolTip.StudyDateLabel.Font.Color := clGreen;
    end;
  end;
end;

```

```

if GeoData.tbStudentLessonProgressTestDate.IsNull then
begin
LessonToolTip.TestDateLabel.Caption := 'Δεν έχεις εξεταστεί!';
LessonToolTip.TestDateLabel.Font.Color := clRed;
LessonToolTip.TestGradeFixedLabel.Visible := False;
LessonToolTip.TestGradeLabel.Visible := False;
end else
begin
LessonToolTip.TestDateLabel.Caption :=
GeoData.tbStudentLessonProgressTestDate.AsString;
LessonToolTip.TestDateLabel.Font.Color := clGreen;
LessonToolTip.TestGradeFixedLabel.Visible := True;
LessonToolTip.TestGradeLabel.Caption :=
Format(GradesFormat,[GeoData.tbStudentLessonProgressTestGrade.AsFloat]);
if GeoData.tbStudentLessonProgressTestGrade.AsFloat > PassingGrade then
LessonToolTip.TestGradeLabel.Font.Color := ClGreen
else
LessonToolTip.TestGradeLabel.Font.Color := ClRed;
LessonToolTip.TestGradeLabel.Visible := True;
end;
end else
begin
LessonToolTip.StudyDateLabel.Caption := 'Δεν το παρακολούθησες!';
LessonToolTip.TestDateLabel.Caption := 'Δεν έχεις εξεταστεί!';
LessonToolTip.StudyDateLabel.Font.Color := clRed;
LessonToolTip.TestDateLabel.Font.Color := clRed;
LessonToolTip.TestGradeFixedLabel.Visible := False;
LessonToolTip.TestGradeLabel.Visible := False;
end;
LessonToolTip.Top := Mouse.CursorPos.Y + 15;
LessonToolTip.Left := Mouse.CursorPos.X + 15;
if Not LessonToolTip.Visible then LessonToolTip.Show;
end;

procedure TStudentForm.FormCloseQuery(Sender: TObject;
var CanClose: Boolean);
////////////////////////////////////
/// Όνομα: FormCloseQuery
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Έλεγχος πριν από το κλείσιμο της φόρμας
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// var CanClose:Boolean - Τίθεται False αν ο χρήστης επιλέξει να μην κλείσει
/// Περιγραφή:
/// Πριν κλείσει η φόρμα εμφανίζεται ένα παράθυρο επιβεβαίωσης στον χρήστη
////////////////////////////////////
begin
CanClose := MessageDlg( 'Θέλεις να βγεις από το πρόγραμμα;',
mtConfirmation, [mbYes,mbNo], 0) = mrYes;
end;

procedure TStudentForm.FormCreate(Sender: TObject);
////////////////////////////////////
/// Όνομα: FormCreate
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Μετά από τη δημιουργία της φόρμας
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:

```

```

/// Όταν δημιουργείται η φόρμα δημιουργείται και μία λίστα τύπου TStringList
/// όπου κρατούνται τα μαθήματα και οι κωδικοί τους.
////////////////////////////////////
begin
  LessonIDs := TStringList.Create;
end;

procedure TStudentForm.FormDestroy(Sender: TObject);
////////////////////////////////////
/// Όνομα: FormDestroy
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Πριν την καταστροφή της φόρμας
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Πριν καταστραφεί η φόρμα καταστρέφεται το αντικείμενο TStringList που
/// είχε δημιουργηθεί
////////////////////////////////////
begin
  LessonIDs.Destroy;
end;

procedure TStudentForm.FormShow(Sender: TObject);
////////////////////////////////////
/// Όνομα: FormShow
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κατά το άνοιγμα της φόρμας
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ανοίγει η φόρμα, ανοίγονται τα σχετικά tables στην βάση, ανατίθενται
/// οι event handlers των Boxes, και συμπληρώνονται τα διάφορα labels
////////////////////////////////////
begin
  Caption := 'ε-Γεωγραφία - Μαθητής: ' + User.UserName;
  cmbLessons.Clear;
  LessonIDs.Clear;
  GeoData.tbLessons.Open;
  while not GeoData.tbLessons.eof do
  begin
    cmbLessons.AddItem(GeoData.tbLessonsName.AsString,NIL);
    LessonIDs.Add(GeoData.tbLessonsID.AsString);
    GeoData.tbLessons.Next;
  end;
  Boxes := TMapBoxCollection.Create(MapImage);
  ReadDBBoxes;
  Boxes.OnBoxClick := BoxClick;
  Boxes.OnBoxMouseOver := BoxMouseOver;
  GeoData.tbLessons.First;
  if (User.LastLesson > 0) AND (GeoData.tbLessons.Locate('ID',User.LastLesson,[])) then
  begin
    lbLessonFixed.Caption := 'Το τελευταίο μάθημα που παρακολούθησες ήταν το: ';
    lbLesson.Caption := GeoData.tbLessonsName.AsString;
    cmbLessons.ItemIndex := LessonIDs.IndexOf(GeoData.tbLessonsID.AsString);
  end else
  begin
    lbLessonFixed.Caption := 'Δεν έχεις παρακολουθήσει ακόμα κανένα μάθημα!';
    lbLesson.Caption := '';
    cmbLessons.ItemIndex := -1;
  end;
end;

```

```

end;
GeoData.tbLessons.Close;
GeoData.tbStudentLessonProgress.Filter := 'StudentID = ' + IntToStr(User.ID);
GeoData.tbStudentLessonProgress.Filtered := True;
GeoData.tbStudentLessonProgress.Open;
cmbLessonsChange(self);
end;

procedure TStudentForm.MapImageMouseLeave(Sender: TObject);
////////////////////////////////////////////////////////////////
/// Όνομα: MapImageMouseLeave
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Όταν ο δείκτης του ποντικιού βγει εκτός της περιοχής του χάρτη
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο δείκτης του ποντικιού βγει εκτός της περιοχής του χάρτη κρύβεται το
/// ToolTip αν τυχόν ήταν ανοιχτό
////////////////////////////////////////////////////////////////
begin
LessonToolTip.Hide;
end;

procedure TStudentForm.MapImageMouseMove(Sender: TObject; Shift: TShiftState; X,
Y: Integer);
////////////////////////////////////////////////////////////////
/// Όνομα: MapImageMouseLeave
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Όταν ο δείκτης του ποντικιού βρίσκεται πάνω στον χάρτη αλλά όχι
/// σε τετράγωνο
/// Παράμετροι:
/// Sender: TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Shift: TShiftState - η κατάσταση των Ctrl,Alt,Shift (αγνοείται)
/// X,Y: Integer - οι συντεταγμένες του δείκτη του ποντικιού (αγνοούνται)
/// Περιγραφή:
/// Όταν ο δείκτης του ποντικιού βρίσκεται πάνω στον χάρτη αλλά όχι σε κάποιο
/// τετράγωνο, το tooltip κρύβεται.
////////////////////////////////////////////////////////////////
begin
LessonToolTip.Hide;
end;

procedure TStudentForm.MapPanelResize(Sender: TObject);
////////////////////////////////////////////////////////////////
/// Όνομα: MapPanelResize
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Αλλαγή του μεγέθους του MapPanel
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Αν αλλάξει το μέγεθος του MapPanel, ξανακαλείται η ReadDBBoxes για να
/// τοποθετηθεί τα τετράγωνα στα σημεία που πρέπει.
////////////////////////////////////////////////////////////////
begin
ReadDBBoxes;
end;

procedure TStudentForm.btnChangePasswordClick(Sender: TObject);

```



```

////////////////////////////////////
/// Όνομα: btnChangePasswordClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnChangePassword
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnChangePassword, ανοίγει η φόρμα
/// ChangePasswordForm, όπου μπορεί ο χρήστης να αλλάξει το συνθηματικό του
////////////////////////////////////
begin
  ChangePasswordForm.ShowModal;
end;

procedure TStudentForm.btnCloseClick(Sender: TObject);
////////////////////////////////////
/// Όνομα: btnCloseClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnClose
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnClose, κλείνει η φόρμα
////////////////////////////////////
begin
  Close;
end;

procedure TStudentForm.btnHelpClick(Sender: TObject);
////////////////////////////////////
/// Όνομα: btnCloseClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnClose
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnHelp, εμφανίζεται η βοήθεια
////////////////////////////////////
begin
  Application.HelpJump('Help');
end;

procedure TStudentForm.btnShowEvaluationClick(Sender: TObject);
////////////////////////////////////
/// Όνομα: btnShowEvaluationClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnShowEvaluation
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnShowEvaluation, ανοίγει η φόρμα
/// StudentEvaluationForm, όπου ο μαθητής μπορεί να δει μία επισκόπηση
/// της προόδου του. Αν η StudentEvaluationForm επιστρέψει με αποτέλεσμα
/// mYYes τότε ξεκινάει ένα τεστ προσαρμοσμένο στην κατηγορία που τον δυσκολεύει
/// περισσότερο
////////////////////////////////////
var TestCategory:String;

```

```

    OldMasterFields:String;
begin
    if StudentEvaluationForm.ShowForStudent(TestCategory) = mrYes then
    //mrYes: ο Μαθητής επέλεξε να εξασκηθεί στην κατηγορία TestCategory
    begin
        GeoData.tbQuestions.Filter := 'Category = ' + TestCategory + '';
        GeoData.tbQuestions.Filtered := True;
        OldMasterFields := GeoData.tbQuestions.MasterFields;
        GeoData.tbQuestions.MasterSource := NIL;
        GeoData.tbQuestions.MasterFields := '';
        Self.Hide;
        ShowTestForm.ShowModal;
        GeoData.tbQuestions.Filter := '';
        GeoData.tbQuestions.Filtered := False;
        GeoData.tbQuestions.MasterSource := GeoData.srcLessons;
        GeoData.tbQuestions.MasterFields := OldMasterFields;
        Self.Show;
    end;
end;

procedure TStudentForm.btnStartLessonClick(Sender: TObject);
/////////////////////////////////////////////////////////////////
/// Όνομα: btnStartLessonClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnStartLesson
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnStartLesson, ανοίγει η φόρμα
/// ShowLessonForm, όπου ο μαθητής θα παρακολουθήσει το επιλεγμένο μάθημα
/////////////////////////////////////////////////////////////////
begin
    GeoData.tbLessons.Open;
    GeoData.tbLessons.Locate('ID',StrToInt(LessonIDs[cmbLessons.ItemIndex]),[]);
    Self.Hide;
    ShowLessonForm.ShowModal;
    GeoData.tbLessons.Close;
    Self.Show;
    cmbLessonsChange(self);
end;

procedure TStudentForm.btnStartTestClick(Sender: TObject);
/////////////////////////////////////////////////////////////////
/// Όνομα: btnStartTestClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnStartTest
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnStartTest, ανοίγει η φόρμα
/// ShowTestForm και ξεκινάει το τεστ προόδου στο επιλεγμένο μάθημα
/////////////////////////////////////////////////////////////////
begin
    GeoData.tbLessons.Open;
    GeoData.tbLessons.Locate('ID',StrToInt(LessonIDs[cmbLessons.ItemIndex]),[]);
    Self.Hide;
    ShowTestForm.ShowModal;
    GeoData.tbLessons.Close;
    Self.Show;
end;

```

```

    cmbLessonsChange(self);
end;

procedure TStudentForm.cmbLessonsChange(Sender: TObject);
////////////////////////////////////////////////////////////////////
// Όνομα: btnStartTestClick
// Τύπος: Event Handler
// Πυροδότηση:
//   Επιλογή κάποιου μαθήματος από τη λίστα του cmbLessons
// Παράμετροι:
//   Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
// Περιγραφή:
//   Όταν ο χρήστης επιλέξει κάποιο μάθημα από το combobox cmbLessons,
//   ενημερώνονται διάφορα controls της φόρμας που εμφανίζουν στατιστικά
//   στοιχεία (πότε το παρακολούθησε τελευταία φορά, τι βαθμό πήρε, κτλ)
//   και επίσης ενεργοποιείται το πλήκτρο btnStartLesson και αν έχει
//   παρακολουθήσει το μάθημα, ενεργοποιείται και το btnStartTest
////////////////////////////////////////////////////////////////////
begin
    if cmbLessons.ItemIndex > -1 then
        begin
            LessonLabel.Caption := cmbLessons.Text;
            if GeoData.tbStudentLessonProgress.Locate
                ('LessonID',StrToInt(LessonIDs[cmbLessons.ItemIndex]),[]) then
                begin
                    if GeoData.tbStudentLessonProgressLessonDate.IsNull then
                        begin
                            StudyDateLabel.Caption := 'Δεν έχεις παρακολουθήσει αυτό το μάθημα ακόμα!';
                            StudyDateLabel.Font.Color := clRed;
                        end else
                        begin
                            StudyDateLabel.Caption := GeoData.tbStudentLessonProgressLessonDate.AsString;
                            StudyDateLabel.Font.Color := clGreen;
                        end;
                    if GeoData.tbStudentLessonProgressTestDate.IsNull then
                        begin
                            TestDateLabel.Caption := 'Δεν έχεις εξεταστεί σε αυτό το μάθημα ακόμα!';
                            TestDateLabel.Font.Color := clRed;
                            TestGradeFixedLabel.Visible := False;
                            TestGradeLabel.Visible := False;
                        end else
                        begin
                            TestDateLabel.Caption := GeoData.tbStudentLessonProgressTestDate.AsString;
                            TestDateLabel.Font.Color := clGreen;
                            TestGradeFixedLabel.Visible := True;
                            TestGradeLabel.Caption
                                :=
                                Format(GradesFormat,[GeoData.tbStudentLessonProgressTestGrade.AsFloat]);
                            if GeoData.tbStudentLessonProgressTestGrade.AsFloat > PassingGrade then
                                TestGradeLabel.Font.Color := clGreen
                            else
                                TestGradeLabel.Font.Color := clRed;
                                TestGradeLabel.Visible := True;
                            end;
                        end else
                        begin
                            StudyDateLabel.Caption := 'Δεν έχεις παρακολουθήσει αυτό το μάθημα ακόμα!';
                            TestDateLabel.Caption := 'Δεν έχεις εξεταστεί σε αυτό το μάθημα ακόμα!';
                            StudyDateLabel.Font.Color := clRed;
                            TestDateLabel.Font.Color := clRed;
                            TestGradeFixedLabel.Visible := False;
                            TestGradeLabel.Visible := False;
                        end;
                    end;
                end;
            end;
        end;
    end;
end;

```

```

    end;
  end else
  begin
    LessonLabel.Caption := 'Διάλεξε μάθημα...';
    StudyDateLabel.Caption := "";
    TestDateLabel.Caption := "";
    StudyDateLabel.Font.Color := clBlack;
    TestDateLabel.Font.Color := clBlack;
    TestGradeFixedLabel.Visible := False;
    TestGradeLabel.Visible := False;
  end;

  btnStartLesson.Enabled := (cmbLessons.ItemIndex > -1);
  if btnStartLesson.Enabled then
  begin
    btnStartTest.Enabled :=
      (StrToInt(LessonIDs[cmbLessons.ItemIndex]) =
        GeoData.tbStudentLessonProgressLessonID.AsInteger) AND
      (NOT GeoData.tbStudentLessonProgressLessonDate.IsNull);
  end;

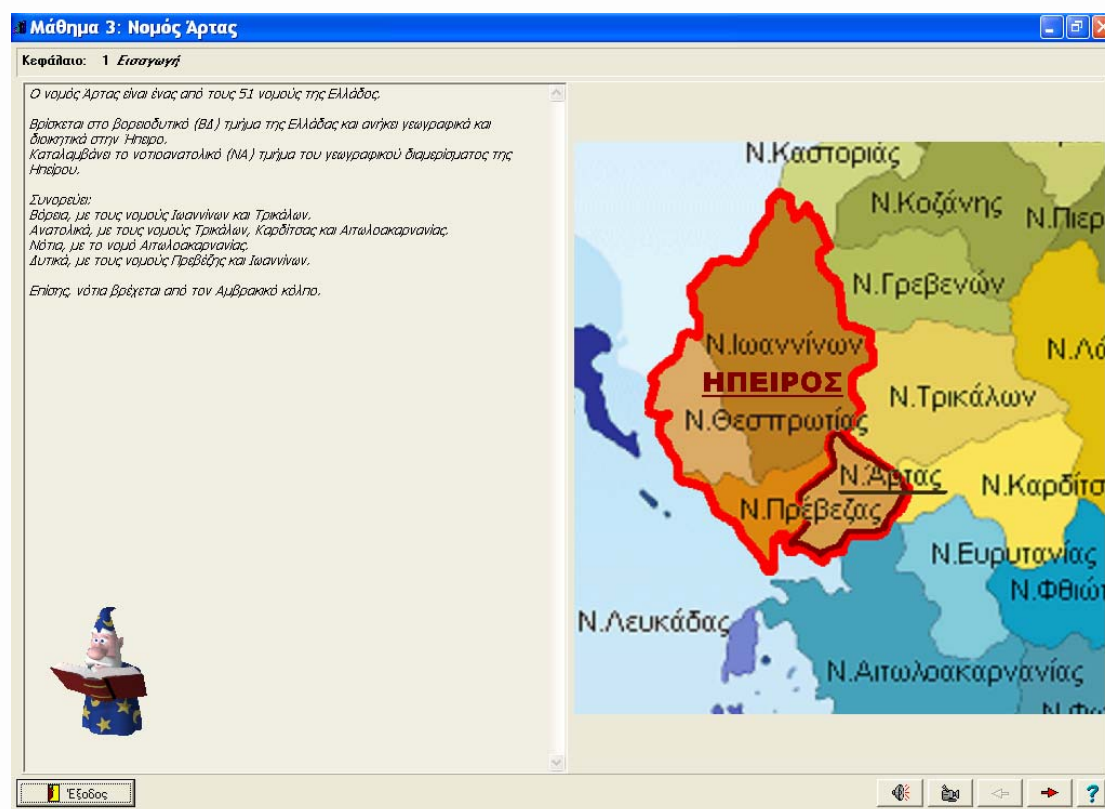
end;

procedure TStudentForm.FormClose(Sender: TObject,
  var Action: TCloseAction);
////////////////////////////////////////////////////
// Όνομα: FormClose
// Τύπος: Event Handler
// Πυροδότηση:
//   Κατά το κλείσιμο της φόρμας
// Παράμετροι:
//   Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
//   var Action:TCloseAction - δεν χρησιμοποιείται
// Περιγραφή:
//   Όταν κλείνει η φόρμα, πραγματοποιείται και το κλείσιμο του πίνακα
//   tbStudentLessonProgress, και εμφανίζεται η αρχική οθόνη του προγράμματος,
//   η LoginForm
////////////////////////////////////////////////////
begin
  GeoData.tbStudentLessonProgress.Close;
  GeoData.tbStudentLessonProgress.Filtered := False;
  GeoData.tbStudentLessonProgress.Filter := "";
  LoginForm.Show;
end;

end.

```

Η παρακάτω οθόνη αποτελεί μία από τις πολλές οθόνες που έχουν σχεδιαστεί για την παρακολούθηση των μαθημάτων από τους μαθητές :



Παρακάτω παρατίθεται ο κώδικας υλοποίησης των παραπάνω φορμών:  
*unit ShowLessonUnit;*

*interface*

*uses*

*Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
 Dialogs, ExtCtrls, OleCtrls, StdCtrls, DBCtrls, Buttons;*

*type*

```
TShowLessonForm = class(TForm)
  Bevel1: TBevel;
  Label1: TLabel;
  btnClose: TBitBtn;
  btnNext: TBitBtn;
  btnPrevious: TBitBtn;
  DBText1: TDBText;
  DBText2: TDBText;
  btnVideo: TBitBtn;
  btnSpeak: TBitBtn;
  LessonPanel: TPanel;
  DBMemo1: TDBMemo;
  Splitter: TSplitter;
  LessonImage: TImage;
  btnHelp: TBitBtn;
  procedure btnHelpClick(Sender: TObject);
  procedure FormResize(Sender: TObject);
  procedure btnSpeakClick(Sender: TObject);
  procedure btnVideoClick(Sender: TObject);
  procedure btnPreviousClick(Sender: TObject);
  procedure FormClose(Sender: TObject; var Action: TCloseAction);
  procedure btnCloseClick(Sender: TObject);
  procedure btnNextClick(Sender: TObject);
```

```

    procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
    procedure FormShow(Sender: TObject);
private
    { Private declarations }
    procedure UpdatePictureAndButtons;
    procedure LogLesson;
public
    { Public declarations }
end;

var
    ShowLessonForm: TShowLessonForm;
    LastChapter: Integer;

implementation

uses GeoDataUnit, GlobalsUnit, db;

{$R *.dfm}

procedure TShowLessonForm.btnCloseClick(Sender: TObject);
////////////////////////////////////////////////////
/// Όνομα: btnCloseClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnClose
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnClose, κλείνει η φόρμα
////////////////////////////////////////////////////
begin
    Close;
end;

procedure TShowLessonForm.UpdatePictureAndButtons;
////////////////////////////////////////////////////
/// Όνομα: UpdatePictureAndButtons
/// Τύπος: Private Procedure
/// Παράμετροι:
/// (καμία)
/// Περιγραφή:
/// Η UpdatePictureAndButtons καλείται όταν αλλάζει το κεφάλαιο, για να
/// δείξει την εικόνα του νέου κεφαλαίου, εάν υπάρχει. Επίσης ενεργοποιεί
/// η απενεργοποιεί τα σχετικά controls της φόρμας.
/// Για να διαβάσει την αποθηκευμένη εικόνα χρησιμοποιεί
/// την GetFile της κλάσης TExternal και εμφανίζει την εικόνα
/// στη φόρμα με την Picture.LoadFromFile
////////////////////////////////////////////////////
var extPicture: TExternal;
    ExtID: Integer;
begin
    ExtID := GeoData.tbChaptersPictureExternalID.AsInteger;
    if ExtID > 0 then
        begin
            extPicture := TExternal.Create(ExtID);
            LessonImage.Width := LessonPanel.Width div 2 - 10;
            Splitter.Width := 6;
            LessonImage.Visible := True;
            Splitter.Visible := True;
            Splitter.Enabled := True;
        end;
    end;
end;

```

```

LessonImage.Picture.LoadFromFile(extPicture.GetFile);
extPicture.Destroy;
end else
begin
  Splitter.Enabled := False;
  LessonImage.Visible := False;
  Splitter.Visible := False;
  LessonImage.Width := 0;
  Splitter.Width := 0;
end;
ExtID := GeoData.tbChaptersMediaExternalID.AsInteger;
btnVideo.Enabled := (ExtID > 0);

end;

procedure TShowLessonForm.LogLesson;
/////////////////////////////////////////////////////////////////
/// Όνομα: LogLesson
/// Τύπος: Private Procedure
/// Παράμετροι:
/// (καμία)
/// Περιγραφή:
/// Η LogLesson καταγράφει στην βάση, στον πίνακα tbStudentLessonProgress,
/// την παρακολούθηση του συγκεκριμένου μαθήματος από τον τρέχον χρήστη.
/////////////////////////////////////////////////////////////////
var LessonID: Integer;
begin
  LessonID := GeoData.tbLessonsID.AsInteger;
  if GeoData.tbStudentLessonProgress.Locate('LessonID', LessonID, []) then
    begin
      GeoData.tbStudentLessonProgress.Edit;
    end else
    begin //πρώτη φορά που παρακολουθεί το συγκεκριμένο μάθημα
      GeoData.tbStudentLessonProgress.Append;
      GeoData.tbStudentLessonProgressLessonID.AsInteger := LessonID;
      GeoData.tbStudentLessonProgressStudentID.AsInteger := User.ID;
    end;
    GeoData.tbStudentLessonProgressLessonDate.AsDateTime := Now;
    GeoData.tbStudentLessonProgress.Post;
    User.LastLesson := LessonID;
    User.Store;
  end;

  procedure TShowLessonForm.btnHelpClick(Sender: TObject);
  ///////////////////////////////////////////////////////////////////////
  /// Όνομα: btnCloseClick
  /// Τύπος: Event Handler
  /// Πυροδότηση:
  /// Κλικ του πλήκτρου btnClose
  /// Παράμετροι:
  /// Sender: TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
  /// Περιγραφή:
  /// Όταν ο χρήστης πατήσει το πλήκτρο btnHelp, εμφανίζεται η βοήθεια
  ///////////////////////////////////////////////////////////////////////
  begin
    Application.HelpJump('Help');
  end;

  procedure TShowLessonForm.btnNextClick(Sender: TObject);
  ///////////////////////////////////////////////////////////////////////
  /// Όνομα: btnNextClick

```

```

/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnNext
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnNext, μετακινείται στο επόμενο
/// κεφάλαιο του μαθήματος. Το πλήκτρο γίνεται disabled αν έφτασε στο
/// τελευταίο κεφάλαιο
////////////////////////////////////
begin
  GeoData.tbChapters.Next;
  UpdatePictureAndButtons;
  if GeoData.tbChaptersOrdering.AsInteger > LastChapter then
    LastChapter := GeoData.tbChaptersOrdering.AsInteger;
    btnNext.Enabled := (GeoData.tbChapters.RecNo < GeoData.tbChapters.RecordCount )
      AND (Not GeoData.tbChapters.Eof);
    btnPrevious.Enabled := Not GeoData.tbChapters.Bof;
  end;

procedure TShowLessonForm.btnPreviousClick(Sender: TObject);
////////////////////////////////////
/// Όνομα: btnPreviousClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnPrevious
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnPrevious, μετακινείται στο προηγούμενο
/// κεφάλαιο του μαθήματος. Το πλήκτρο γίνεται disabled αν έφτασε στο
/// πρώτο κεφάλαιο
////////////////////////////////////
begin
  GeoData.tbChapters.Prior;
  UpdatePictureAndButtons;
  btnNext.Enabled := Not GeoData.tbChapters.Eof;
  btnPrevious.Enabled := Not GeoData.tbChapters.Bof;
end;

procedure TShowLessonForm.btnSpeakClick(Sender: TObject);
////////////////////////////////////
/// Όνομα: btnSpeakClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnSpeak
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnSpeak, το κείμενο του κεφαλαίου
/// στέλνεται στον συνθέτη φωνής "Δημοσθένης" για να αναγνωστεί φωναχτά
////////////////////////////////////
begin
  Speak(GeoData.tbChaptersText.AsString);
end;

procedure TShowLessonForm.btnVideoClick(Sender: TObject);
////////////////////////////////////
/// Όνομα: btnVideoClick
/// Τύπος: Event Handler
/// Πυροδότηση:

```



```

/// Κλικ του πλήκτρου btnVideo
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnVideo, εάν υπάρχει κάποιο αρχείο
/// πολυμέσων συνδεδεμένο με το συγκεκριμένο κεφάλαιο, αυτό εκτελείται.
////////////////////////////////////
var extPicture: TExternal;
  ExtID: Integer;
  Filename: String;
begin
  ExtID := GeoData.tbChaptersMediaExternalID.AsInteger;
  if ExtID > 0 then
    begin
      extPicture := TExternal.Create(ExtID);
      Filename := extPicture.GetFile;
      if Filename <> " then ExecuteFile(Filename);
      extPicture.Free;
    end;
  end;
end;

procedure TShowLessonForm.FormClose(Sender: TObject; var Action: TCloseAction);
////////////////////////////////////
/// Όνομα: FormClose
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κατά το κλείσιμο της φόρμας
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// var Action:TCloseAction - δεν χρησιμοποιείται
/// Περιγραφή:
/// Όταν κλείνει η φόρμα, πραγματοποιείται και το κλείσιμο όλων των
/// πινάκων που χρησιμοποιούνται από αυτήν
////////////////////////////////////
begin
  GeoData.tbChaptersMain.Close;
  GeoData.tbChapters.Close;
end;

procedure TShowLessonForm.FormCloseQuery(Sender: TObject;
var CanClose: Boolean);
////////////////////////////////////
/// Όνομα: FormCloseQuery
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Έλεγχος πριν από το κλείσιμο της φόρμας
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// var CanClose:Boolean - Τίθεται False αν ο χρήστης επιλέξει να μην κλείσει
/// Περιγραφή:
/// Αν το μάθημα δεν έχει ολοκληρωθεί ακόμα, εμφανίζεται ένα παράθυρο
/// επιβεβαίωσης στον χρήστη για το αν θέλει να σταματήσει πριν ολοκληρωθεί
////////////////////////////////////
var BM: TBookmark;
begin
  BM := GeoData.tbChapters.GetBookmark;
  GeoData.tbChapters.Last;
  if GeoData.tbChaptersOrdering.AsInteger = LastChapter then
    begin
      CanClose := True;
      LogLesson;
    end;
end;

```

```

end else
begin
  CanClose :=
    MessageDlg('Το μάθημα δεν ολοκληρώθηκε ακόμα!' + #13 + 'Θέλεις να σταματήσεις;',
      mtConfirmation, [mbYes,mbNo], 0) = mrYes;
  if not CanClose then
    GeoData.tbChapters.GotoBookmark(BM);
    GeoData.tbChapters.FreeBookmark(BM);
  end;
end;

procedure TShowLessonForm.FormResize(Sender: TObject);
////////////////////////////////////////////////////
// Όνομα: FormResize
// Τύπος: Event Handler
// Πυροδότηση:
//   Αλλαγή του μεγέθους της φόρμας
// Παράμετροι:
//   Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
// Περιγραφή:
//   Αν αλλάξει το μέγεθος της φόρμας, αλλάζει αναλογικά το μέγεθος της εικόνας
////////////////////////////////////////////////////
begin
  LessonImage.Width := LessonPanel.Width div 2 - 10;
end;

procedure TShowLessonForm.FormShow(Sender: TObject);
////////////////////////////////////////////////////
// Όνομα: FormShow
// Τύπος: Event Handler
// Πυροδότηση:
//   Κατά το άνοιγμα της φόρμας
// Παράμετροι:
//   Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
// Περιγραφή:
//   Όταν ανοίγει η φόρμα, ανοίγονται τα σχετικά tables στην βάση, τίθεται
//   ο τίτλος του μαθήματος στην γραμμή τίτλου της φόρμας, και εμφανίζεται
//   το πρώτο κεφάλαιο του μαθήματος
////////////////////////////////////////////////////
begin
  GeoData.tbChaptersMain.Open;
  GeoData.tbChapters.Open;
  GeoData.tbChapters.First;
  ShowLessonForm.Caption := 'Μάθημα ' + GeoData.tbLessonsOrdering.AsString +
    ':' + GeoData.tbLessonsName.AsString;
  btnNext.Enabled := Not GeoData.tbChapters.Eof;
  btnPrevious.Enabled := Not GeoData.tbChapters.Bof;
  LastChapter := 0;
  UpdatePictureAndButtons;
end;

end.

```


### **-Ερωτήσεις σε τεστ μαθημάτων**

Οι παρακάτω οθόνες αποτελούν δύο από τις πολλές οθόνες που έχουν σχεδιαστεί για την εξέταση των μαθημάτων από τους μαθητές. Το κάθε τεστ αποτελείται από κάποιες ερωτήσεις που μπορεί να είναι δύο ειδών: Ερωτήσεις στον χάρτη :

**Ερωτήσεις πάνω στο μάθημα 3: Νομός Άρτας**

Ερώτηση: 1


Ολοκλήρωση τεστ: 

Χρόνος που απομένει: 

Μένουν 12 ερωτήσεις

Μένουν 4 λεπτά

Δείξε τον Νομό Άρτας στον χάρτη της Ελλάδας

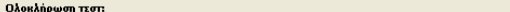
Έξοδος


Σάββατο, 24 Νοεμβρίου 2012

και ερωτήσεις πολλαπλής επιλογής :

**Ερωτήσεις πάνω στο μάθημα 3: Νομός Άρτας**

Ερώτηση: 8

Ολοκλήρωση τεστ: 

Χρόνος που απομένει: 

Μένουν 5 ερωτήσεις

Μένουν 4 λεπτά



Ποιο είναι το ψηλότερο βουνό του νομού Άρτας;

**A** Ίδη (Ψηλορείτης)

**B** Τιμφριστός (Βελούχι)

**Γ** Αθαμανικά Όρη (Τζουμέρκα)

**Δ** Οίτη

Έξοδος

Παρακάτω παρατίθεται ο κώδικας υλοποίησης των παραπάνω φορμών :

```
unit ShowTestUnit;
```

*interface*

*uses*

*Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, ComCtrls, ExtCtrls, DBCtrls, StdCtrls, Buttons, GlobalsUnit;*

*type*

```
TShowTestForm = class(TForm)
  pbTimeLeft: TProgressBar;
  btnClose: TBitBtn;
  Label1: TLabel;
  pbQuestionsLeft: TProgressBar;
  Label2: TLabel;
  lbQuestionsLeft: TLabel;
  lbTimeLeft: TLabel;
  Label5: TLabel;
  DBText2: TDBText;
  Bevel1: TBevel;
  Bevel2: TBevel;
  btnAnswerA: TBitBtn;
  btnAnswerB: TBitBtn;
  btnAnswerC: TBitBtn;
  btnAnswerD: TBitBtn;
  TestTimer: TTimer;
  imgQuestion: TImage;
  imgMap: TImage;
  DotShape: TShape;
  Shape1: TShape;
  lbQuestionNo: TLabel;
  btnHelp: TBitBtn;
  procedure btnHelpClick(Sender: TObject);
  procedure imgMapMouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
  procedure btnAnswerAClick(Sender: TObject);
  procedure btnCloseClick(Sender: TObject);
  procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
  procedure TestTimerTimer(Sender: TObject);
  procedure FormClose(Sender: TObject; var Action: TCloseAction);
  procedure FormShow(Sender: TObject);
private
  { Private declarations }
  MapImageConv: TMapImageConv;
  procedure LogTestGrade(aTestGrade:real);
  procedure LogAnswer(WasCorrect: Boolean);
  procedure LogAnswerAndMoveToNext(WasCurrentCorrect: Boolean);
  procedure SetRandomButtonText(aBtn: TBitBtn);
  procedure ShowCurrentQuestion;
  procedure UpdatePicture;
  procedure UpdateMap;
  procedure ShowDistanceComment(WrongAnswerID:Integer);
public
  { Public declarations }
end;
```

*var*

*ShowTestForm: TShowTestForm;*

*implementation*

uses GeoDataUnit, MapUnit;

{\$R \*.dfm}

var

LastCaption: String;  
 LastTag: Integer;  
 CurrentTime: Integer;  
 TotalQuestions: Integer;  
 CurrentQuestion: Integer;  
 CorrectAnswers: Integer;  
 TestFinished: Boolean;  
 Retries: Integer;  
 CorrectAnswerID: Integer;  
 CorrectAnswerText: String;

procedure TShowTestForm.LogTestGrade(aTestGrade:real);

///  
 /// Όνομα: LogTestGrade  
 /// Τύπος: Private Procedure  
 /// Παράμετροι:  
 /// aTestGrade:real - Ο τελικός βαθμός του τεστ  
 /// Περιγραφή:  
 /// Η LogTestGrade καταγράφει στην βάση, στον πίνακα tbStudentLessonProgress,  
 /// την τελική βαθμολογία του τρέχοντος χρήστη στο συγκεκριμένου τεστ.

///  
 var LessonID: Integer;  
 begin  
 LessonID := GeoData.tbLessonsID.AsInteger;  
 if GeoData.tbStudentLessonProgress.Locate('LessonID',LessonID,[]) then  
 begin  
 GeoData.tbStudentLessonProgress.Edit;  
 GeoData.tbStudentLessonProgress.TestDate.AsDateTime := Now;  
 GeoData.tbStudentLessonProgress.TestGrade.AsFloat := aTestGrade;  
 GeoData.tbStudentLessonProgress.Post;  
 end;  
 end;

procedure TShowTestForm.LogAnswer(WasCorrect:Boolean);

///  
 /// Όνομα: LogAnswer  
 /// Τύπος: Private Procedure  
 /// Παράμετροι:  
 /// WasCorrect:Boolean - True = Έδωσε σωστή απάντηση  
 /// False = Έδωσε λάθος απάντηση  
 /// Περιγραφή:  
 /// Η LogAnswer καταγράφει στην βάση, στον πίνακα tbStudentAnswerLog,  
 /// την επιτυχία ή όχι του τρέχοντος χρήστη στην συγκεκριμένη ερώτηση.

///  
 var QuestionID: Integer;  
 begin  
 QuestionID := GeoData.tbQuestionsID.AsInteger;  
 if not GeoData.tbStudentAnswerLog.Locate('QuestionID',QuestionID,[]) then  
 begin  
 GeoData.tbStudentAnswerLog.Insert;  
 GeoData.tbStudentAnswerLog.StudentID.AsInteger := User.ID;  
 GeoData.tbStudentAnswerLog.QuestionID.AsInteger := QuestionID;  
 end else  
 GeoData.tbStudentAnswerLog.Edit;  
 GeoData.tbStudentAnswerLog.TimeStamp.AsDateTime := Now;  
 GeoData.tbStudentAnswerLog.WasCorrect.AsBoolean := WasCorrect;

```

GeoData.tbStudentAnswerLog.Post;
end;

procedure TShowTestForm.LogAnswerAndMoveToNext(WasCurrentCorrect: Boolean);
////////////////////////////////////////////////////////////////////
// Όνομα: LogAnswerAndMoveToNext
// Τύπος: Private Procedure
// Παράμετροι:
//   WasCurrentCorrect:Boolean - True = Έδωσε σωστή απάντηση
//   False = Έδωσε λάθος απάντηση
// Περιγραφή:
//   Η LogAnswerAndMoveToNext καλεί την LogAnswer για να γίνει η καταγραφή στην
//   βάση
//   και υπολογίζει την συνολική βαθμολογία του τεστ αν τελείωσαν όλες οι ερωτήσεις.
//   Αν υπάρχουν και άλλες ερωτήσεις καλείται η ShowCurrentQuestion για να εμφανίσει
//   την επόμενη ερώτηση
////////////////////////////////////////////////////////////////////
var TestGrade:Real;
    TestPassed:Boolean;
begin
    TestTimer.Enabled := False;
    if WasCurrentCorrect then
        begin
            LogAnswer(True);
            CorrectAnswers := CorrectAnswers + 1;
        end else
        begin
            LogAnswer(False);
        end;
    GeoData.tbQuestions.Next;

    if GeoData.tbQuestions.Eof then
        begin //test finished
            TestGrade := (CorrectAnswers * MaxGrade) / TotalQuestions;
            TestPassed := (TestGrade >= PassingGrade);
            if TestPassed then
                begin
                    MessageDlg(
                        'Συγχαρητήρια! Πέρασες το τεστ με επιτυχία!' + chr(13) +
                        'Απάντησες σωστά τις ' + IntToStr(CorrectAnswers) + ' από τις ' +
                        IntToStr(TotalQuestions) + ' ερωτήσεις.' + chr(13) +
                        'Η βαθμολογία σου είναι: ' + Format(GradesFormat,[TestGrade]) + chr(13) + chr(13) +
                        'Μπορείς να προχωρήσεις στο επόμενο μάθημα!',mtInformation,[mbOk],0);
                end else
                begin
                    MessageDlg(
                        'Δυστυχώς δεν κατάφερες να περάσεις το τεστ.' + chr(13) +
                        'Απάντησες σωστά τις ' + IntToStr(CorrectAnswers) + ' από τις ' +
                        IntToStr(TotalQuestions) + ' ερωτήσεις.' + chr(13) +
                        'Η βαθμολογία σου είναι: ' + Format(GradesFormat,[TestGrade]) + chr(13) +
                        'ένώ για να περάσεις χρειάζεσαι: ' + Format(GradesFormat,[PassingGrade]) + chr(13) +
                        chr(13) +
                        'Διάβασε πάλι το μάθημα και ξαναδοκίμασε!',mtError,[mbOk],0);
                end;
            TestFinished := True;
            LogTestGrade(TestGrade);
            Close;
        end else
        begin
            TestTimer.Enabled := True;
            ShowCurrentQuestion;
        end;
    end;
end;

```

```

end;
end;

procedure TShowTestForm.ShowDistanceComment(WrongAnswerID:Integer);
////////////////////////////////////////////////////////////////
/// Όνομα: ShowDistanceComment
/// Τύπος: Private Procedure
/// Παράμετροι:
/// WrongAnswerID:Integer - Ο κωδικός της λάθος απάντησης που δόθηκε
/// Περιγραφή:
/// Εάν η απάντηση στην τρέχουσα ερώτηση αφορούσε τοποθεσία, όταν
/// δοθεί λάθος απάντηση, καλείται η ShowDistanceComment που εμφανίζει ένα
/// σχόλιο σχετικά με το πόσο κοντά ή μακριά είναι οι δύο τοποθεσίες
////////////////////////////////////////////////////////////////
var mX,mY,aX,aY: Integer;
    dX, dY, dist: Real;
    s: string;
begin
    //Move to the correct answer record
    GeoData.tbAnswersCache.Filtered := False;
    GeoData.tbAnswersCache.First;
    if GeoData.tbAnswersCache.Locate('ID',CorrectAnswerID,[]) and
        GeoData.tbAnswersCache.IsPlace.AsBoolean then
    begin
        aX := GeoData.tbAnswersCacheMapX.AsInteger;
        aY := GeoData.tbAnswersCacheMapY.AsInteger;
        //Move to the wrong answer record
        if GeoData.tbAnswersCache.Locate('ID',WrongAnswerID,[]) and
            GeoData.tbAnswersCache.IsPlace.AsBoolean then
        begin
            //get coordinates of the place the student clicked on
            mX := GeoData.tbAnswersCacheMapX.AsInteger;
            mY := GeoData.tbAnswersCacheMapY.AsInteger;
            dX := mX - aX;
            dY := mY - aY;
            dist := sqrt(sqrt(dX) + sqrt(dY));
            if dist < 50 then
                s := 'Λάθος! Αλλά πολύ κοντά!'
            else if dist < 100 then
                s := 'Λάθος! Όχι και πολύ μακριά όμως!'
            else
                s := 'Λάθος! Πολύ μακριά!';
            ShowMessage(s);
        end;
    end;
end;

procedure TShowTestForm.btnAnswerAClick(Sender: TObject);
////////////////////////////////////////////////////////////////
/// Όνομα: btnAnswerAClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ κάποιου από τα πλήκτρα απαντήσεων (btnAnswerA, btnAnswerB,
/// btnAnswerC, btnAnswerD)
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Αν επελέγη η σωστή απάντηση, καταχωρείται και προχωράει στην επόμενη.
/// Αν είναι λάθος, εμφανίζεται ένα σχόλιο αν η απάντηση που επιλέχτηκε
/// μοιάζει λεκτικά στην σωστή, ή αν οι απαντήσεις αφορούν τοποθεσίες, ένα
/// σχόλιο σχετικά με την απόσταση των δύο τοποθεσιών. Στη συνέχεια

```

```

/// αυξάνεται ο μετρητής των διαθέσιμων προσπαθειών και αν φτάσει
/// το όριο που δίνεται από την global μεταβλητή MaxTriesMultChoice,
/// η ερώτηση καταχωρείται ως λάθος και προχωράει στην επόμενη.
////////////////////////////////////
var WasCorrect:Boolean;
    WordMatch: Integer;
begin
    if Sender is TBitBtn then
        begin
            WasCorrect := ((Sender as TBitBtn).Tag = CorrectAnswerID);
            if not WasCorrect then
                begin
                    WordMatch := FindWordMatch((Sender as TBitBtn).Caption, CorrectAnswerText);
                    if WordMatch > 70 then
                        begin
                            ShowMessage ('Περίπου! Αλλά όχι ακριβώς!')
                        end else
                            if GeoData.tbAnswersCacheIsPlace.AsBoolean then
                                ShowDistanceComment((Sender as TBitBtn).Tag);
                            (Sender as TBitBtn).Enabled := False;
                            Retries := Retries + 1;
                            if Retries >= MaxTriesMultChoice then LogAnswerAndMoveToNext(False);
                        end else
                            begin
                                LogAnswerAndMoveToNext(True);
                            end;
                        end;
                    end;
                end;

procedure TShowTestForm.btnCloseClick(Sender: TObject);
////////////////////////////////////
/// Όνομα: btnCloseClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnClose
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnClose, κλείνει η φόρμα
////////////////////////////////////
begin
    Close;
end;

procedure TShowTestForm.btnHelpClick(Sender: TObject);
////////////////////////////////////
/// Όνομα: btnCloseClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnClose
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnHelp, εμφανίζεται η βοήθεια
////////////////////////////////////
begin
    Application.HelpJump("Help");
end;

procedure TShowTestForm.FormClose(Sender: TObject; var Action: TCloseAction);
////////////////////////////////////

```



```

/// Όνομα: FormClose
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κατά το κλείσιμο της φόρμας
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// var Action:TCloseAction - δεν χρησιμοποιείται
/// Περιγραφή:
/// Όταν κλείνει η φόρμα, πραγματοποιείται και το κλείσιμο όλων των
/// πινάκων που χρησιμοποιούνται από αυτήν
////////////////////////////////////
begin
  TestTimer.Enabled := False;
  GeoData.tbAnswersCache.Close;
  GeoData.tbAnswers.Close;
  GeoData.tbQuestions.Close;
  GeoData.tbQuestionsMain.Close;
  GeoData.tbStudentAnswerLog.Close;
  GeoData.tbStudentAnswerLog.Filter := "";
  GeoData.tbStudentAnswerLog.Filtered := False;
end;

procedure TShowTestForm.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
////////////////////////////////////
/// Όνομα: FormCloseQuery
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Έλεγχος πριν από το κλείσιμο της φόρμας
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// var CanClose:Boolean - Τίθεται False αν ο χρήστης επιλέξει να μην κλείσει
/// Περιγραφή:
/// Αν το τεστ δεν έχει ολοκληρωθεί ακόμα, εμφανίζεται ένα παράθυρο
/// επιβεβαίωσης στον χρήστη για το αν θέλει να σταματήσει πριν ολοκληρωθεί
////////////////////////////////////
begin
  if not GeoData.tbQuestions.EOF then
    begin
      CanClose :=
        TestFinished or
        (MessageDlg('Θέλεις να σταματήσεις το τεστ εδώ; ' + chr(13) +
          'Θα πρέπει να το ξανακάνεις από την αρχή!',
          mtConfirmation, [mbYes,mbNo], 0) = mrYes);
    end;
end;

procedure TShowTestForm.UpdatePicture;
////////////////////////////////////
/// Όνομα: UpdatePicture
/// Τύπος: Private Procedure
/// Παράμετροι:
/// (καμία)
/// Περιγραφή:
/// Η UpdatePicture καλείται όταν αλλάζει η ερώτηση, για να
/// δείξει την εικόνα της νέας ερώτησης, εάν υπάρχει.
/// Για να διαβάσει την αποθηκευμένη εικόνα χρησιμοποιεί
/// την GetFile της κλάσης TExternal και εμφανίζει την εικόνα
/// στη φόρμα με την Picture.LoadFromFile
////////////////////////////////////
var extPicture: TExternal;
    ExtID: Integer;

```

```

begin
  DotShape.Visible := False;
  ExtID := GeoData.tbQuestionsPictureExternalID.AsInteger;
  if ExtID > 0 then
    begin
      extPicture := TExternal.Create(ExtID);
      imgQuestion.Visible := True;
      imgQuestion.Picture.LoadFromFile(extPicture.GetFile);
      extPicture.Destroy;
    end else
      imgQuestion.Visible := False;
  end;

procedure TShowTestForm.UpdateMap;
////////////////////////////////////////////////////////////////////
// Όνομα: UpdateMap
// Τύπος: Private Procedure
// Παράμετροι:
//   (καμία)
// Περιγραφή:
//   Η UpdateMap καλείται όταν αλλάζει η ερώτηση, για να δείξει
//   τον χάρτη της νέας ερώτησης, εάν η ερώτηση είναι από χάρτη.
//   Για να διαβάσει την εικόνα του χάρτη χρησιμοποιεί
//   την GetFile της κλάσης TExternal και την εμφανίζει
//   στη φόρμα με την Picture.LoadFromFile
////////////////////////////////////////////////////////////////////
var extPicture: TExternal;
    ExtID: Integer;
begin
  ExtID := GeoData.tbAnswersCacheMapExternalID.AsInteger;
  if ExtID > 0 then
    begin
      extPicture := TExternal.Create(ExtID);
      imgMap.Visible := True;
      imgMap.Picture.LoadFromFile(extPicture.GetFile);
      extPicture.Destroy;
      if Assigned (MapImageConv) then MapImageConv.Free;
      MapImageConv := TMapImageConv.Create(imgMap);
    end else
      imgQuestion.Visible := False;
  end;

procedure TShowTestForm.FormShow(Sender: TObject);
////////////////////////////////////////////////////////////////////
// Όνομα: FormShow
// Τύπος: Event Handler
// Πυροδότηση:
//   Κατά το άνοιγμα της φόρμας
// Παράμετροι:
//   Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
// Περιγραφή:
//   Όταν ανοίγει η φόρμα, ανοίγονται τα σχετικά tables στην βάση, τίθεται
//   ο τίτλος του τεστ στην γραμμή τίτλου της φόρμας, αρχικοποιούνται οι
//   μετρητές χρόνου και ερωτήσεων και εμφανίζεται η πρώτη ερώτηση
////////////////////////////////////////////////////////////////////
begin
  GeoData.tbStudentAnswerLog.Filter := 'StudentID = ' + IntToStr(User.ID);
  GeoData.tbStudentAnswerLog.Filtered := True;
  GeoData.tbStudentAnswerLog.Open;
  GeoData.tbQuestionsMain.Open;
  GeoData.tbQuestions.Open;

```

```

GeoData.tbQuestions.Last;
TotalQuestions := GeoData.tbQuestions.RecordCount;
GeoData.tbQuestions.First;
GeoData.tbAnswers.Open;
GeoData.tbAnswersCache.Open;
if GeoData.tbQuestions.Filtered then
begin
  ShowTestForm.Caption := 'Τεστ Εξάσκησης στην κατηγορία ' +
    GeoData.tbQuestionsCategory.AsString;
end else
begin
  ShowTestForm.Caption := 'Ερωτήσεις πάνω στο μάθημα ' +
    GeoData.tbLessonsOrdering.AsString +
    ':' + GeoData.tbLessonsName.AsString;
end;
CurrentTime := 0;
CurrentQuestion := GeoData.tbQuestionsOrdering.AsInteger;
DotShape.Visible := False;
CorrectAnswers := 0;
pbQuestionsLeft.Max := TotalQuestions;

if TotalTestTime > 0 then
begin
  pbTimeLeft.Visible := True;
  lbTimeLeft.Visible := True;
  Label1.Visible := True;
  pbTimeLeft.Max := TotalTestTime;
  TestTimer.Enabled := True;
end else
begin
  pbTimeLeft.Visible := False;
  lbTimeLeft.Visible := False;
  Label1.Visible := False;
  TestTimer.Enabled := False;
end;

Randomize;
TestFinished := False;
ShowCurrentQuestion;
end;

procedure TShowTestForm.imgMapMouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
////////////////////////////////////
/// Όνομα: imgMapMouseDown
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του ποντικιού στην περιοχή του χάρτη
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Button: TMouseButton - Ποιο από τα πλήκτρα του ποντικιού πατήθηκε
/// Shift: TShiftState - Η κατάσταση των πλήκτρων Ctrl,Alt,Shift, κτλ (αγνοείται)
/// X,Y: Integer - Η θέση του ποντικιού σε σχέση με την πάνω αριστερή γωνία
/// του χάρτη
/// Περιγραφή:
/// Αν γίνει κλικ επάνω στον χάρτη, υπολογίζεται η θέση και συγκρίνεται
/// με την θέση της σωστής απάντησης. Εμφανίζεται ένα σχετικό σχόλιο
/// και κατόπιν η απάντηση καταγράφεται ως σωστή ή λανθασμένη
////////////////////////////////////
var mX,mY,aX,aY: Integer;
    sz, i: Integer;

```

```

    dX, dY, dist: Real;
    s: String;
begin
  If Button = mbLeft then
  begin
    mX      :=      MapImageConv.ImageToMapX(imgMap.Left      +      X      -
MapImageConv.ImageAddedLeft);
    mY      :=      MapImageConv.ImageToMapY(imgMap.Top      +      Y      -
MapImageConv.ImageAddedTop);
    aX := GeoData.tbAnswersCacheMapX.AsInteger;
    aY := GeoData.tbAnswersCacheMapY.AsInteger;
    Shape1.Left := MapImageConv.ImageAddedLeft + MapImageConv.MapToImageX(aX) -
10;
    Shape1.Top := MapImageConv.ImageAddedTop + MapImageConv.MapToImageY(aY) -
10;

    sz := MapImageConv.MapToImageX(50);
    DotShape.Width := sz;
    DotShape.Height := sz;
    DotShape.Left := imgMap.Left + X - sz div 2;
    DotShape.Top := imgMap.Top + Y - sz div 2;
    DotShape.Visible := True;
    dX := mX - aX;
    dY := mY - aY;
    dist := sqrt(sqr(dX) + sqr(dY));
    if dist < 40 then
    begin
      ShowMessage('Πολύ Σωστά!');
      LogAnswerAndMoveToNext(True);
    end else
    begin
      if dist > 300 then
        s := 'Είσαι πολύ μακριά!!' else
        s := 'Είσαι κοντά, αλλά όχι αρκετά!';
      Retries := Retries + 1;
      if Retries < MaxAttemptsMap then
      begin
        i:= MaxAttemptsMap-Retries;
        if i>1 then
          s := s + #13 + 'Έχεις άλλες ' + IntToStr(i) + ' προσπάθειες' else
          s := s + #13 + 'Έχεις άλλη 1 προσπάθεια';
        end;
        ShowMessage(s);
        if Retries >= MaxAttemptsMap then
          LogAnswerAndMoveToNext(False);
        end;
      end;
    end;
  end;
end;

procedure TShowTestForm.SetRandomButtonText(aBtn:TBitBtn);
////////////////////////////////////
/// Όνομα: SetRandomButtonText
/// Τύπος: Private Procedure
/// Παράμετροι:
/// aBtn:TBitBtn - το πλήκτρο στο οποίο θέλουμε να γραφεί μια τυχαία απάντηση
/// Περιγραφή:
/// Η SetRandomButtonText ψάχνει ανάμεσα στις πιθανές απαντήσεις της τρέχουσας
/// ερώτησης, και βάζει τυχαία κάποια από τις απαντήσεις ως κείμενο του πλήκτρου.
/// Κάθε φορά που χρησιμοποιείται μία από τις απαντήσεις, σβήνεται από τον
/// προσωρινό πίνακα για να μην ξαναχρησιμοποιηθεί σε άλλο πλήκτρο

```

```

////////////////////////////////////
var RecPos:Integer;
begin
  aBtn.Visible := True;
  aBtn.Enabled := True;
  if GeoData.tbAnswersCache.RecordCount > 0 then
    begin
      GeoData.tbAnswersCache.First;
      RecPos := Random(GeoData.tbAnswersCache.RecordCount);
      GeoData.tbAnswersCache.MoveBy(RecPos);
      aBtn.Tag := GeoData.tbAnswersCacheID.AsInteger;
      aBtn.Caption := GeoData.tbAnswersCacheText.AsString;
      GeoData.tbAnswersCache.Delete; //Σβήνεται από τον προσωρινό πίνακα, όχι από τη βάση
      LastCaption := aBtn.Caption;
      LastTag := aBtn.Tag;
    end else
    begin // Δεν έμειναν άλλες απαντήσεις
      if LastCaption <> " then begin
        aBtn.Tag := LastTag;
        aBtn.Caption := LastCaption;
        LastCaption := "";
      end else
      begin
        aBtn.Visible := False;
      end;
    end;
  end;
end;

procedure TShowTestForm.ShowCurrentQuestion;
////////////////////////////////////
/// Όνομα: ShowCurrentQuestion
/// Τύπος: Private Procedure
/// Παράμετροι:
/// (καμία)
/// Περιγραφή:
/// Η ShowCurrentQuestion εμφανίζει την ερώτηση που πρέπει να απαντήσει ο μαθητής.
/// Αν είναι ερώτηση πολλαπλής επιλογής τότε καλεί την SetRandomButtonText για να
/// ορίσει τα κείμενα των πλήκτρων απάντησης, αλλιώς καλεί την UpdateMap για να
/// εμφανιστεί ο χάρτης
////////////////////////////////////
var CorrectButton: Integer;
    AnswerGroup: Integer;
    AnswerText: String;
    TotalAnswers: Integer;
    ShowMap: Boolean;
begin
  btnAnswerA.Tag := -1;
  btnAnswerB.Tag := -1;
  btnAnswerC.Tag := -1;
  btnAnswerD.Tag := -1;
  Retries := 0;

  CurrentQuestion := GeoData.tbQuestions.RecNo;
  lbQuestionNo.Caption := IntToStr(CurrentQuestion);
  pbQuestionsLeft.Position := CurrentQuestion;
  lbQuestionsLeft.Caption := 'Μένουν '+IntToStr(TotalQuestions-CurrentQuestion)+'
  ερωτήσεις';

  CorrectAnswerID := GeoData.tbQuestionsCorrectAnswerID.AsInteger;

  //Βρες την σωστή απάντηση και αποθήκευσε την

```

```

GeoData.tbAnswersCache.Filtered := False;
GeoData.tbAnswersCache.First;
GeoData.tbAnswersCache.Locate('ID', CorrectAnswerID, []);
AnswerText := GeoData.tbAnswersCache.Text.AsString;
CorrectAnswerText := AnswerText;
AnswerGroup := GeoData.tbAnswersCache.Group.AsInteger;
ShowMap := GeoData.tbQuestionsShowMap.AsBoolean;

if ShowMap then
begin
  //ερώτηση από χάρτη
  btnAnswerA.Visible := False;
  btnAnswerB.Visible := False;
  btnAnswerC.Visible := False;
  btnAnswerD.Visible := False;
  imgQuestion.Visible := False;
  UpdateMap;
  imgMap.Visible := True;
end else
begin
  imgMap.Visible := False;
  //Γράψε 4 πιθανές απαντήσεις στα πλήκτρα απάντησης
  GeoData.tbAnswersCache.Filter := '(Group = ' + IntToStr(AnswerGroup) +
    ') AND (ID <> ' + IntToStr(CorrectAnswerID) + ')';
  GeoData.tbAnswersCache.Filtered := True;

  TotalAnswers := 1;
  LastCaption := AnswerText;

  SetRandomButtonText(btnAnswerA);
  SetRandomButtonText(btnAnswerB);
  if btnAnswerB.Visible then TotalAnswers := 2;
  SetRandomButtonText(btnAnswerC);
  if btnAnswerC.Visible then TotalAnswers := 3;
  SetRandomButtonText(btnAnswerD);
  if btnAnswerD.Visible then TotalAnswers := 4;

  // Αν οι διαθέσιμες απαντήσεις είναι λιγότερες απο 4, μείωσε
  // ανάλογα τον διαθέσιμο αριθμό προσπαθειών
  Retries := 4 - TotalAnswers;

  //Διάλεξε ένα από τα ενεργά πλήκτρα απάντησης και βάλε σε αυτό την
  //σωστή απάντηση
  CorrectButton := 1 + random(TotalAnswers);
  case CorrectButton of
    1:
      begin
        btnAnswerA.Tag := CorrectAnswerID;
        btnAnswerA.Caption := AnswerText;
      end;
    2:
      begin
        btnAnswerB.Tag := CorrectAnswerID;
        btnAnswerB.Caption := AnswerText;
      end;
    3:
      begin
        btnAnswerC.Tag := CorrectAnswerID;
        btnAnswerC.Caption := AnswerText;
      end;
    4:

```

```

begin
  btnAnswerD.Tag := CorrectAnswerID;
  btnAnswerD.Caption := AnswerText;
end;
end;

//ακύρωσε τις αλλαγές στον προσωρινό πίνακα tbAnswersCache
GeoData.tbAnswersCache.CancelUpdates;
UpdatePicture;
end;
end;

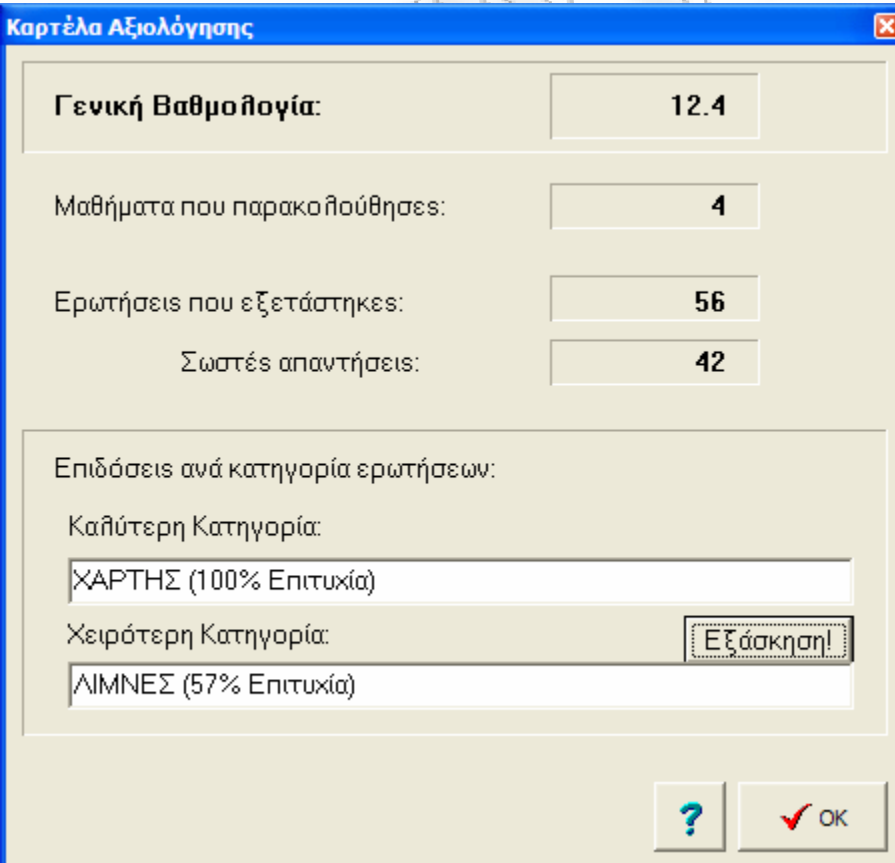
procedure TShowTestForm.TestTimerTimer(Sender: TObject);
////////////////////////////////////////////////////////////////////
// Όνομα: TestTimerTimer
// Τύπος: Event Handler
// Πυροδότηση:
// Κάθε δευτερόλεπτο (TestTimer.Interval = 1000 ms)
// Παράμετροι:
// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
// Περιγραφή:
// Ενημερώνεται η μπάρα του χρόνου pbTimeLeft και το κείμενο της label
// lbTimeLeft, για τον διαθέσιμο χρόνο. Αν ο χρόνος έληξε, τελειώνει το τεστ
//////////////////////////////////////////////////////////////////
var TimeLeft:Integer;
begin
  if CurrentTime < TotalTestTime then
    begin
      CurrentTime := CurrentTime + 1;
      pbTimeLeft.Position := CurrentTime;
      TimeLeft := TotalTestTime - CurrentTime;
      if TimeLeft > 60 then
        begin
          lbTimeLeft.Caption := 'Μένουν '+IntToStr(TimeLeft div 60)+' λεπτά'
        end else
          if TimeLeft >= 0 then
            begin
              lbTimeLeft.Caption := 'Μένουν '+IntToStr(TimeLeft)+' δευτερόλεπτα'
            end;

          if TimeLeft <= 0 then
            begin
              GeoData.tbQuestions.Last;
              LogAnswerAndMoveToNext(false);
            end;
          end;
        end;
      end;
    end.

```

### -Καρτέλα αξιολόγησης μαθητή

Ο μαθητής έχει τη δυνατότητα να δει την αξιολόγησή του στα τεστ που έχει εξεταστεί κάνοντας κλικ στο πλήκτρο «Καρτέλα Αξιολόγησης» στην βασική «Οθόνη του Μαθητή» και εμφανίζεται η παρακάτω οθόνη :



Περιγραφή	Αποτέλεσμα
Γενική Βαθμολογία:	12.4
Μαθήματα που παρακολούθησες:	4
Ερωτήσεις που εξετάστηκες:	56
Σωστές απαντήσεις:	42

Επιδόσεις ανά κατηγορία ερωτήσεων:

Καλύτερη Κατηγορία:	ΧΑΡΤΗΣ (100% Επιτυχία)
Χειρότερη Κατηγορία:	Εξάσκηση!
	ΛΙΜΝΕΣ (57% Επιτυχία)

Buttons: ? (Help), OK (Confirm)

Παρακάτω παρατίθεται ο κώδικας υλοποίησης της παραπάνω φόρμας :

```
unit StudentEvaluationUnit;
```

```
interface
```

```
uses
```



*Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, ExtCtrls, Mask, DBCtrls, Buttons;*

type

```
TStudentEvaluationForm = class(TForm)
  Label1: TLabel;
  Label2: TLabel;
  Label3: TLabel;
  Label4: TLabel;
  Label6: TLabel;
  Label7: TLabel;
  Label8: TLabel;
  btnPractice: TButton;
  edWorstCat: TEdit;
  edBestCat: TEdit;
  Bevel1: TBevel;
  Bevel2: TBevel;
  btnOK: TBitBtn;
  DBText2: TDBText;
  DBText1: TDBText;
  lbCorrectAnswers: TLabel;
  lbAllAnswers: TLabel;
  Bevel3: TBevel;
  Bevel4: TBevel;
  Bevel5: TBevel;
  Bevel6: TBevel;
  btnDetails: TBitBtn;
  btnHelp: TBitBtn;
  procedure btnHelpClick(Sender: TObject);
  procedure btnDetailsClick(Sender: TObject);
  procedure FormClose(Sender: TObject; var Action: TCloseAction);
  procedure FormShow(Sender: TObject);
private
  { Private declarations }
  mStudentID: Integer;
  mWorstCategory: String;
public
  { Public declarations }
  function ShowForTeacher(StudentID: Integer; StudentName: String): Integer;
  function ShowForStudent(var aCategory: String): Integer;
end;
```

var

```
StudentEvaluationForm: TStudentEvaluationForm;
```

implementation

```
uses GeoDataUnit, GlobalsUnit, StudentProgressReportUnit;
```

```
{ $R *.dfm }
```

```
function TStudentEvaluationForm.ShowForTeacher(StudentID: Integer;
StudentName: String): Integer;
////////////////////////////////////
/// Όνομα: ShowForTeacher
/// Τύπος: Public Function
/// Επιστρέφει:
/// Integer - mrOk ή mrNone
/// Παράμετροι:
/// StudentID: Integer - ο κωδικός του μαθητή
/// StudentName: String - το όνομα του μαθητή
```

```

/// Περιγραφή:
/// Η ShowForTeacher καλείται από το υπόλοιπο πρόγραμμα για να εμφανιστεί
/// η καρτέλα αξιολόγησης ενός μαθητή, εάν ο τρέχον χρήστης του προγράμματος
/// είναι καθηγητής
////////////////////////////////////
begin
  mStudentID := StudentID;
  Caption := 'Αξιολόγηση Μαθητή: ' + StudentName;
  Label2.Caption := 'Μαθήματα που παρακολούθησε:.';
  Label3.Caption := 'Ερωτήσεις που εξετάστηκαν:.';
  btnPractice.Visible := False;
  btnDetails.Visible := True;
  Result := ShowModal;
end;

function TStudentEvaluationForm.ShowForStudent(var aCategory:String):Integer;
////////////////////////////////////
/// Όνομα: ShowForStudent
/// Τύπος: Public Function
/// Επιστρέφει:
/// Integer - mrOk, mrNone (αν κλείσει με το btnOk, ή το Close Button πάνω δεξιά)
/// mrYes (αν πατηθεί το btnPractice)
/// Παράμετροι:
/// var aCategory: String - σε αυτή την παράμετρο επιστρέφεται η κατηγορία
/// ερωτήσεων που έχει το χαμηλότερο ποσοστό επιτυχίας
/// ο μαθητής
/// Περιγραφή:
/// Η ShowForStudent καλείται από το υπόλοιπο πρόγραμμα για να εμφανιστεί
/// η καρτέλα αξιολόγησης του τρέχοντος χρήστη, αν είναι μαθητής.
////////////////////////////////////
begin
  mStudentID := User.ID;
  Caption := 'Καρτέλα Αξιολόγησης';
  Label2.Caption := 'Μαθήματα που παρακολούθησες:.';
  Label3.Caption := 'Ερωτήσεις που εξετάστηκες:.';
  btnPractice.Visible := True;
  btnDetails.Visible := False;
  Result := ShowModal;
  aCategory := mWorstCategory;
end;

procedure TStudentEvaluationForm.btnDetailsClick(Sender: TObject);
////////////////////////////////////
/// Όνομα: btnDetailsClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnDetails
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnDetails, εμφανίζεται η φόρμα
/// StudentProgressReportForm με λεπτομέρειες για την πρόοδο του μαθητή
////////////////////////////////////
begin
  StudentProgressReportForm.ShowModal;
end;

procedure TStudentEvaluationForm.btnHelpClick(Sender: TObject);
////////////////////////////////////
/// Όνομα: btnCloseClick
/// Τύπος: Event Handler

```

```

/// Πυροδότηση:
/// Κλικ του πλήκτρου btnClose
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnHelp, εμφανίζεται η βοήθεια
////////////////////////////////////
begin
  Application.HelpJump('Help');
end;

procedure TStudentEvaluationForm.FormClose(Sender: TObject;
  var Action: TCloseAction);
////////////////////////////////////
/// Όνομα: FormClose
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κατά το κλείσιμο της φόρμας
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// var Action:TCloseAction - δεν χρησιμοποιείται
/// Περιγραφή:
/// Όταν κλείνει η φόρμα, πραγματοποιείται και το κλείσιμο όλων των
/// πινάκων που χρησιμοποιούνται από αυτήν
////////////////////////////////////
begin
  GeoData.qrStudentEvaluation.Close;
  GeoData.qrAnswersByCategories.Close;
end;

procedure TStudentEvaluationForm.FormShow(Sender: TObject);
////////////////////////////////////
/// Όνομα: FormShow
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κατά το άνοιγμα της φόρμας
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ανοίγει η φόρμα, ανοίγονται τα σχετικά tables στην βάση, και
/// με τη χρήση του query qrAnswersByCategories υπολογίζεται σε ποια
/// κατηγορία ερωτήσεων έχει την χειρότερη και σε ποια την καλύτερη επίδοση
/// και τα αποτελέσματα γράφονται στα σχετικά πεδία.
/// Αν η φόρμα έχει ανοιχτεί από μαθητή, τότε μπορεί να κάνει κλικ
/// στο btnPractice, και να εξεταστεί σε ένα προσαρμοσμένο τεστ με ερωτήσεις
/// μόνο από την κατηγορία που αντιμετωπίζει πρόβλημα
////////////////////////////////////
var cntAllAnswers: Integer;
    cntCorrectAnswers: Integer;
begin
  GeoData.qrStudentEvaluation.ParamByName('ID').AsInteger := mStudentID;
  GeoData.qrStudentEvaluation.Open;
  mWorstCategory := '';
  edWorstCat.Text := '';
  edBestCat.Text := '';
  lbAllAnswers.Caption := '';
  lbCorrectAnswers.Caption := '';
  GeoData.qrAnswersByCategories.ParamByName('StudentID').AsInteger := mStudentID;
  GeoData.qrAnswersByCategories.Open;
  if GeoData.qrAnswersByCategories.RecordCount > 0 then
    begin

```

```

GeoData.qrAnswersByCategories.First; // Η χειρότερη κατηγορία
mWorstCategory := GeoData.qrAnswersByCategoriesCategory.AsString;
if mWorstCategory <> " then
  edWorstCat.Text := mWorstCategory +
    ' (' + GeoData.qrAnswersByCategoriesPercentCorrect.AsString + '%
Επιτυχία)';
cntAllAnswers := 0;
cntCorrectAnswers := 0;
While not GeoData.qrAnswersByCategories.EOF do
begin
  cntAllAnswers := cntAllAnswers +
GeoData.qrAnswersByCategoriesAllAnswers.AsInteger;
  cntCorrectAnswers := cntCorrectAnswers +
GeoData.qrAnswersByCategoriesCorrectAnswers.AsInteger;
  GeoData.qrAnswersByCategories.Next;
end;
lbAllAnswers.Caption := IntToStr(cntAllAnswers);
lbCorrectAnswers.Caption := IntToStr(cntCorrectAnswers);
// Η καλύτερη κατηγορία
if GeoData.qrAnswersByCategoriesCategory.AsString <> " then
  edBestCat.Text := GeoData.qrAnswersByCategoriesCategory.AsString +
    ' (' + GeoData.qrAnswersByCategoriesPercentCorrect.AsString + '%
Επιτυχία)';
end;
if edWorstCat.Text = " then btnPractice.Visible := False;
end;

end.

```

### -Συνθηματικό μαθητή

Στην παρακάτω οθόνη, ο μαθητής μπορεί να αλλάξει το συνθηματικό που χρησιμοποιεί για να συνδεθεί στο πρόγραμμα. Η οθόνη αυτή εμφανίζεται, όταν ο μαθητής κάνει κλικ στο πλήκτρο «Αλλαγή Συνθηματικού» στην βασική «Οθόνη του Μαθητή» :

**Αλλαγή συνθηματικού**

Γράψτε το παλιό σας συνθηματικό στο επάνω πλαίσιο, και το νέο συνθηματικό και στα δύο κάτω πλαίσια.

Παλιό συνθηματικό:

Νέο συνθηματικό:

Επανάληψη:

**ΠΡΟΣΟΧΗ: Για το συνθηματικό έχει σημασία αν γράφετε με κεφαλαία ή μικρά!**

? OK Άκυρο

Παρακάτω παρατίθεται ο κώδικας υλοποίησης της παραπάνω φόρμας :

```

unit ChangePasswordUnit;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls, Buttons;

type
  TChangePasswordForm = class(TForm)
    Label2: TLabel;
    editOldPassword: TEdit;
    Label1: TLabel;
    editNewPass1: TEdit;
    Label3: TLabel;
    editNewPass2: TEdit;
    Label4: TLabel;
    Label5: TLabel;
    Bevel1: TBevel;
    btnOK: TBitBtn;
    btnCancel: TBitBtn;
    btnHelp: TBitBtn;
    procedure btnHelpClick(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure btnOKClick(Sender: TObject);
    procedure btnCancelClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  ChangePasswordForm: TChangePasswordForm;

implementation

uses GlobalsUnit;

{$R *.dfm}

procedure TChangePasswordForm.btnCancelClick(Sender: TObject);
  ////////////////////////////////////////////////////////////////////
  /// Όνομα: btnCancelClick
  /// Τύπος: Event Handler
  /// Πυροδότηση:
  /// Κλικ του πλήκτρου btnCancel
  /// Παράμετροι:
  /// Sender: TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
  /// Περιγραφή:
  /// Όταν ο χρήστης πατήσει το πλήκτρο btnCancel, η φόρμα κλείνει χωρίς να γίνει
  /// καμία αλλαγή στο συνθηματικό του χρήστη
  ////////////////////////////////////////////////////////////////////
begin
  ModalResult := mrCancel;
end;

procedure TChangePasswordForm.btnHelpClick(Sender: TObject);
  ////////////////////////////////////////////////////////////////////

```

```
/// Όνομα: btnCloseClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnClose
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnHelp, εμφανίζεται η βοήθεια
////////////////////////////////////
begin
  Application.HelpJump('Help');
end;

procedure TChangePasswordForm.btnOKClick(Sender: TObject);
////////////////////////////////////
/// Όνομα: btnOKClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnOk
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnOk, γίνεται έλεγχος αν δόθηκε σωστά το
/// παλιό συνθηματικό, και αν το νέο συνθηματικό έχει γραφτεί με τον ίδιο τρόπο
/// και στα δύο πεδία εισόδου. Αν ναι, τότε αποθηκεύεται το νέο συνθηματικό
/// και η φόρμα κλείνει
////////////////////////////////////
var OldEncrPassword,NewEncrPassword: String;
begin

  if (length(editOldPassword.Text) > 25) or
    (length(editNewPass1.Text) > 25) or
    (length(editNewPass2.Text) > 25) then
    begin
      ShowMessage('Τα συνθηματικά δεν μπορούν να υπερβαίνουν τους 25 χαρακτήρες!');
    end else
    begin
      OldEncrPassword := Encrypted(editOldPassword.Text);
      if User.EncrPassword = OldEncrPassword then
        begin
          if editNewPass1.Text = editNewPass2.Text then
            begin
              NewEncrPassword := Encrypted(editNewPass1.Text);
              if User.ChangePassword(OldEncrPassword,NewEncrPassword) then
                begin
                  User.Store;
                  MessageDlg('Το συνθηματικό άλλαξε με επιτυχία!',mtInformation,[mbOk],0);
                  ModalResult := mrOk;
                end else
                  begin
                    ShowMessage('Δεν ήταν εφικτή η αλλαγή του συνθηματικού! Άγνωστο λάθος!');
                  end;
                end else
                  begin
                    ShowMessage('Το νέο συνθηματικό δεν είναι το ίδιο με την επανάληψη!');
                  end;
                end else
                  begin
                    ShowMessage('Το παλιό συνθηματικό δεν είναι σωστό!');
                  end;
                end;
            end;
          end;
        end;
      end;
    end;
  end;
```

end;

```

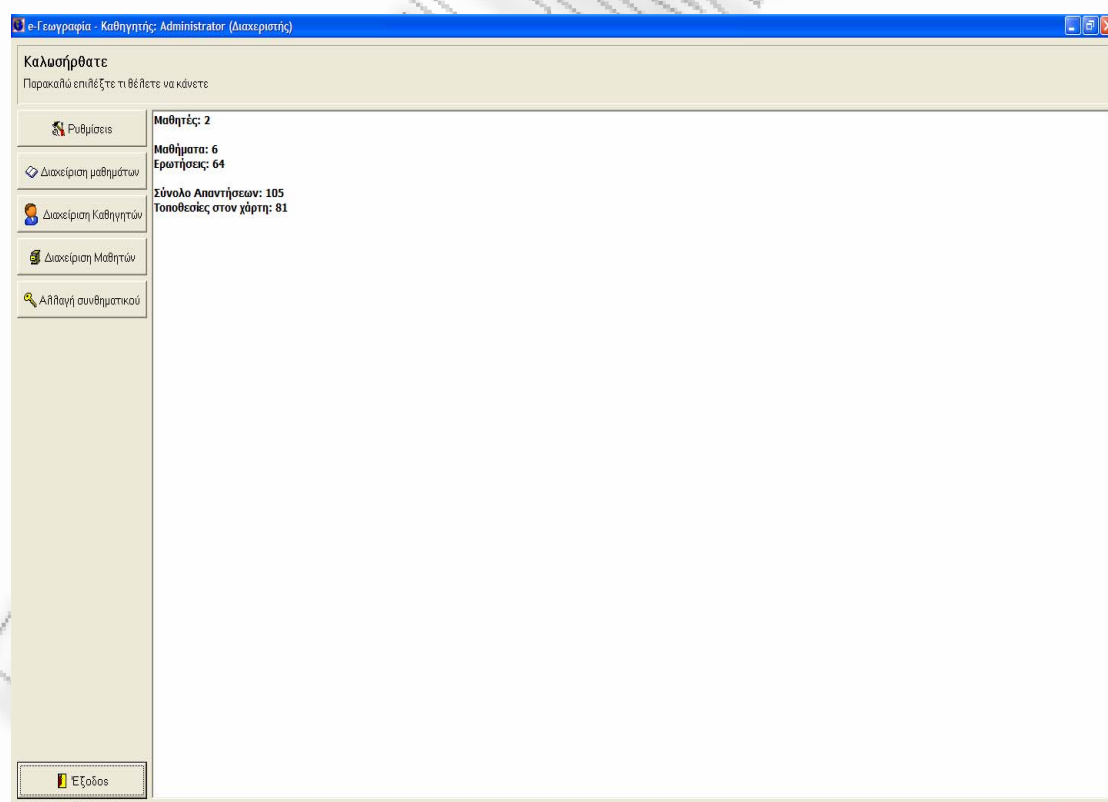
procedure TChangePasswordForm.FormShow(Sender: TObject);
////////////////////////////////////////////////////
/// Όνομα: FormShow
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κατά το άνοιγμα της φόρμας
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ανοίγει η φόρμα, διαγράφονται τα περιεχόμενα των πεδίων εισόδου
////////////////////////////////////////////////////
begin
  editOldPassword.Text := "";
  editNewPass1.Text := "";
  editNewPass2.Text := "";
end;

end.

```

### **-Αρχική οθόνη Καθηγητή-Administrator**

Ο administrator αφού συνδεθεί στην εφαρμογή, μεταβαίνει στην βασική «Οθόνη του Administrator», η οποία παρατίθεται παρακάτω :



Παρακάτω παρατίθεται ο κώδικας υλοποίησης της παραπάνω φόρμας :

*unit TeacherUnit;*

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, Buttons, ExtCtrls;

type

TTeacherForm = class(TForm)

Bevel1: TBevel;

Label1: TLabel;

Label3: TLabel;

btnEditLesson: TBitBtn;

Memo: TMemo;

btnEditStudents: TBitBtn;

btnChangePassword: TBitBtn;

btnClose: TBitBtn;

btnEditTeachers: TBitBtn;

btnSettings: TBitBtn;

procedure btnSettingsClick(Sender: TObject);

procedure btnChangePasswordClick(Sender: TObject);

procedure btnEditStudentsClick(Sender: TObject);

procedure btnEditTeachersClick(Sender: TObject);

procedure FormShow(Sender: TObject);

procedure btnEditLessonClick(Sender: TObject);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);

procedure btnCloseClick(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

TeacherForm: TTeacherForm;

implementation

uses LoginUnit, GlobalsUnit, EditLessonUnit, UserManagementUnit,  
ChangePasswordUnit, SettingsUnit, GeoDataUnit;

{ \$R \*.dfm }

procedure TTeacherForm.btnChangePasswordClick(Sender: TObject);

////////////////////////////////////

/// Όνομα: btnChangePasswordClick

/// Τύπος: Event Handler

/// Πυροδότηση:

/// Κλικ του πλήκτρου btnChangePassword

/// Παράμετροι:

/// Sender: TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)

/// Περιγραφή:

/// Όταν ο χρήστης πατήσει το πλήκτρο btnChangePassword, ανοίγει η φόρμα

/// ChangePasswordForm, όπου μπορεί ο χρήστης να αλλάξει το συνθηματικό του

////////////////////////////////////

begin

ChangePasswordForm.ShowModal;

end;

procedure TTeacherForm.btnCloseClick(Sender: TObject);



```
////////////////////////////////////
/// Όνομα: btnCloseClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnClose
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnClose, κλείνει η φόρμα
////////////////////////////////////
begin
  Close;
end;

procedure TTeacherForm.btnEditLessonClick(Sender: TObject);
////////////////////////////////////
/// Όνομα: btnEditLessonClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnEditLesson
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnEditLesson, εμφανίζεται η EditLessonForm
/// όπου μπορεί να αλλαχτεί η διδακτέα ύλη του προγράμματος
////////////////////////////////////
begin
  Self.Hide;
  EditLessonForm.ShowModal;
  Self.Show;
end;

procedure TTeacherForm.btnEditStudentsClick(Sender: TObject);
////////////////////////////////////
/// Όνομα: btnEditStudentsClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnEditStudents
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnEditStudents, καλείται η ShowList της
/// UserManagementForm με παράμετρο False, οπότε εμφανίζεται η λίστα των μαθητών
/// για επεξεργασία
////////////////////////////////////
begin
  UserManagementForm.ShowList(False);
end;

procedure TTeacherForm.btnEditTeachersClick(Sender: TObject);
////////////////////////////////////
/// Όνομα: btnEditTeachersClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnEditTeachers
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnEditTeachers, καλείται η ShowList της
/// UserManagementForm με παράμετρο True, οπότε εμφανίζεται η λίστα των καθηγητών
/// για επεξεργασία
```

```

////////////////////////////////////
begin
  UserManagementForm.ShowList(True);
end;

procedure TTeacherForm.btnSettingsClick(Sender: TObject);
////////////////////////////////////
/// Όνομα: btnSettingsClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnSettings
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnSettings, εμφανίζεται η SettingsForm
/// όπου μπορούν να αλλαχτούν διάφορες ρυθμίσεις του προγράμματος
////////////////////////////////////
begin
  SettingsForm.ShowModal;
end;

procedure TTeacherForm.FormClose(Sender: TObject; var Action: TCloseAction);
////////////////////////////////////
/// Όνομα: FormClose
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κατά το κλείσιμο της φόρμας
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// var Action:TCloseAction - δεν χρησιμοποιείται
/// Περιγραφή:
/// Όταν κλείνει η φόρμα, εμφανίζεται η αρχική οθόνη του προγράμματος,
/// η LoginForm
////////////////////////////////////
begin
  LoginForm.Show;
end;

procedure TTeacherForm.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
////////////////////////////////////
/// Όνομα: FormCloseQuery
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Έλεγχος πριν από το κλείσιμο της φόρμας
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// var CanClose:Boolean - Τίθεται False αν ο χρήστης επιλέξει να μην κλείσει
/// Περιγραφή:
/// Πριν κλείσει η φόρμα εμφανίζεται ένα παράθυρο επιβεβαίωσης στον χρήστη
////////////////////////////////////
begin
  CanClose := MessageDlg('Θέλετε να βγείτε από το πρόγραμμα;',
    mtConfirmation, [mbYes,mbNo], 0) = mrYes;
end;

procedure TTeacherForm.FormShow(Sender: TObject);
////////////////////////////////////
/// Όνομα: FormShow
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κατά το άνοιγμα της φόρμας

```

```

/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ανοίγει η φόρμα, ανοίγονται τα σχετικά tables στην βάση, γράφονται
/// κάποια στατιστικά στοιχεία στο Memo, γράφεται ο τίτλος της φόρμας, και
/// αν ο χρήστης έχει δικαιώματα διαχειριστή, ενεργοποιούνται τα πλήκτρα
/// btnEditLesson, btnEditTeachers και btnSettings
////////////////////////////////////
begin
  Caption := 'ε-Γεωγραφία - Καθηγητής: ';
  Caption := Caption + User.UserName;
  if User.IsAdmin then
    begin
      Caption := Caption + ' (Διαχειριστής)';
    end;

  Memo.Lines.Clear;
  GeoData.qrTemp.SQL.Text := 'SELECT Count(*) AS CNT From Users where Type = 0';
  GeoData.qrTemp.Open;
  Memo.Lines.Add('Μαθητές: ' + GeoData.qrTempCNT.AsString);
  GeoData.qrTemp.Close;
  GeoData.qrTemp.SQL.Text := 'SELECT Count(*) AS CNT From Lessons';
  GeoData.qrTemp.Open;
  Memo.Lines.Add("");
  Memo.Lines.Add('Μαθήματα: ' + GeoData.qrTempCNT.AsString);
  GeoData.qrTemp.Close;
  GeoData.qrTemp.SQL.Text := 'SELECT Count(*) AS CNT From Questions';
  GeoData.qrTemp.Open;
  Memo.Lines.Add('Ερωτήσεις: ' + GeoData.qrTempCNT.AsString);
  GeoData.qrTemp.Close;
  GeoData.qrTemp.SQL.Text := 'SELECT Count(*) AS CNT From Answers';
  GeoData.qrTemp.Open;
  Memo.Lines.Add("");
  Memo.Lines.Add('Σύνολο Απαντήσεων: ' + GeoData.qrTempCNT.AsString);
  GeoData.qrTemp.Close;
  GeoData.qrTemp.SQL.Text := 'SELECT Count(*) AS CNT From Answers WHERE IsPlace =
True';
  GeoData.qrTemp.Open;
  Memo.Lines.Add('Τοποθεσίες στον χάρτη: ' + GeoData.qrTempCNT.AsString);
  GeoData.qrTemp.Close;

  btnEditLesson.Enabled := User.IsAdmin;
  btnEditTeachers.Enabled := User.IsAdmin;
  btnSettings.Enabled := User.IsAdmin;
end;

end.

```

#### **-Φόρμα αλλαγής ρυθμίσεων σε τεστ**

Η παρακάτω οθόνη είναι ενδεικτική:

**Ρυθμίσεις Προγράμματος**

<b>Διάρκεια Τεστ</b>	<input type="text" value="300"/>	(χρόνος σε sec)
<b>Αριθμός προσπαθειών Ερωτήσεις Πολλ.Επιλ.</b>	<input type="text" value="2"/>	(Αποδεκτός αριθμός λάθος απαντήσεων)
<b>Αριθμός προσπαθειών Ερωτήσεις σε Χάρτη</b>	<input type="text" value="3"/>	(Αποδεκτός αριθμός λάθος απαντήσεων)
<b>Άριστα</b>	<input type="text" value="20.0"/>	(Μέγιστη δυνατή βαθμολογία)
<b>Βάση</b>	<input type="text" value="10.0"/>	(Η ελάχιστη βαθμολογία για να θεωρηθεί επιτυχημένο ένα τεστ)
<b>Δεκαδικά ψηφία</b>	<input type="text" value="1"/>	(Αριθμός δεκαδ.ψηφίων κατά την εμφάνιση της βαθμολογίας)

Παρακάτω παρατίθεται ο κώδικας υλοποίησης της παραπάνω φόρμας :

```
unit SettingsUnit;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, Mask, ExtCtrls, Buttons;
```

```
type
```

```
TSettingsForm = class(TForm)
```

```
Label1: TLabel;
```

```
Label2: TLabel;
```

```
Label3: TLabel;
```

```
Label4: TLabel;
```

```
Label7: TLabel;
```

```
Label8: TLabel;
```

```
Label9: TLabel;
```

```
Label10: TLabel;
```

```
btnOK: TBitBtn;
```

```
btnCancel: TBitBtn;
```

```
Label11: TLabel;
```

```
Label5: TLabel;
```

```
Label6: TLabel;
```

```
Label12: TLabel;
```

```
Label13: TLabel;
```

```
Label14: TLabel;
```

```
Label15: TLabel;
```

```
Label16: TLabel;
```

```
Bevel1: TBevel;
```

```
edGrdDigits: TEdit;
```

```
edPassGrd: TEdit;
```

```
edMaxGrd: TEdit;
```

```

edMaxTriesMap: TEdit;
edMaxTriesMult: TEdit;
edTotalTime: TEdit;
procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
procedure btnOKClick(Sender: TObject);
procedure btnCancelClick(Sender: TObject);
procedure FormShow(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  SettingsForm: TSettingsForm;

implementation

uses GlobalsUnit;

{$R *.dfm}

var SettingsChanged:Boolean;

procedure TSettingsForm.btnCancelClick(Sender: TObject);
/////////////////////////////////////////////////////////////////
/// Όνομα: btnCancelClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnCancel
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnCancel, η SettingsForm κλείνει
/// χωρίς να αποθηκευτεί η όποια αλλαγή στις ρυθμίσεις
/////////////////////////////////////////////////////////////////
begin
  SettingsChanged := False;
  ModalResult := mrCancel;
end;

procedure TSettingsForm.btnOKClick(Sender: TObject);
/////////////////////////////////////////////////////////////////
/// Όνομα: btnOKClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnOk
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnOk, η SettingsForm κλείνει
/// και αποθηκεύονται οι νέες ρυθμίσεις
/////////////////////////////////////////////////////////////////
begin
  TotalTestTime := StrToInt(edTotalTime.Text);
  MaxTriesMultChoice := StrToInt(edMaxTriesMult.Text);
  MaxAttemptsMap := StrToInt(edMaxTriesMap.Text);
  MaxGrade := StrToFloat(edMaxGrd.Text);
  PassingGrade := StrToFloat(edPassGrd.Text);
  GradesFormat := '%.'+IntToStr(StrToInt(edGrdDigits.Text))+'f';
  SaveSettings;

```

```

SettingsChanged := False;
ModalResult := mrOk;
end;

procedure TSettingsForm.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
////////////////////////////////////////////////////////////////////
// Όνομα: FormCloseQuery
// Τύπος: Event Handler
// Πυροδότηση:
// Έλεγχος πριν από το κλείσιμο της φόρμας
// Παράμετροι:
// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
// var CanClose:Boolean - Τίθεται False αν ο χρήστης επιλέξει να μην κλείσει
// Περιγραφή:
// Εμφανίζεται ένα παράθυρο προειδοποίησης στον χρήστη αν έχουν πραγματοποιηθεί
// αλλαγές, ώστε να επιλέξει αν θέλει να αποθηκευτούν ή όχι
////////////////////////////////////////////////////////////////////
var DlgResult: Integer;
begin
if SettingsChanged then begin
DlgResult := MessageDlg ('Πραγματοποιήσατε αλλαγές!' + chr(13) +
'Αποθήκευση;', mtWarning, mbYesNoCancel, 0);
if DlgResult = mrYes then btnOKClick(Self);
if DlgResult = mrNo then btnCancelClick(Self);
if DlgResult = mrCancel then CanClose := False;
end;
end;

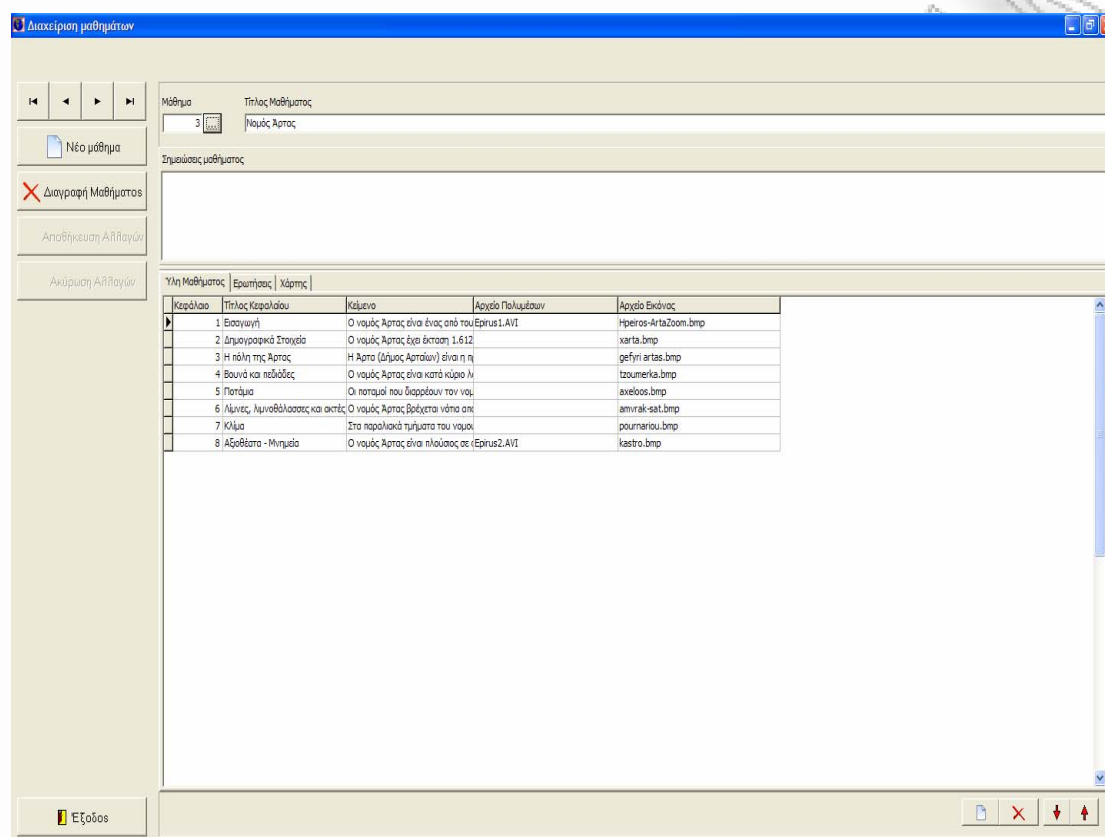
procedure TSettingsForm.FormShow(Sender: TObject);
////////////////////////////////////////////////////////////////////
// Όνομα: FormShow
// Τύπος: Event Handler
// Πυροδότηση:
// Κατά το άνοιγμα της φόρμας
// Παράμετροι:
// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
// Περιγραφή:
// Όταν ανοίγει η φόρμα, θέτονται οι τιμές των διαφόρων πεδίων
////////////////////////////////////////////////////////////////////
var p1,p2:integer;
begin
edTotalTime.Text := IntToStr(TotalTestTime);
edMaxTriesMult.Text := IntToStr(MaxTriesMultChoice);
edMaxTriesMap.Text := IntToStr(MaxAttemptsMap);
edMaxGrd.Text := FloatToStrF(MaxGrade, ffFixed, 3, 1);
edPassGrd.Text := FloatToStrF(PassingGrade, ffFixed, 3, 1);
p1 := pos('.', GradesFormat);
p2 := pos('f', GradesFormat);
if (p1>0) and (p2>p1) and (length(GradesFormat)>0) then
edGrdDigits.Text := Copy(GradesFormat, p1+1, p2-p1-1)
else
edGrdDigits.Text := '0';
SettingsChanged := False;
end;

end.

```

### **-Φόρμα διαχείρισης μαθημάτων**

Ο administrator έχει τη δυνατότητα να «διαχειριστεί» τα μαθήματα που παρακολουθούν οι μαθητές. Η οθόνη αυτή εμφανίζεται, όταν ο administrator κάνει κλικ στο πλήκτρο «Διαχείριση Μαθημάτων» στην βασική «Οθόνη του Administrator» :



Παρακάτω παρατίθεται ο κώδικας της παραπάνω φόρμας :

```
unit LessonPickerUnit;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, ExtCtrls, Grids, DBGrids, StdCtrls, Buttons;
```

```
type
```

```
TLessonPickerForm = class(TForm)
```

```
  btnUp: TBitBtn;
```

```
  btnDown: TBitBtn;
```

```
  DBGrid1: TDBGrid;
```

```
  btnOK: TBitBtn;
```

```
  btnCancel: TBitBtn;
```

```
  Bevel1: TBevel;
```

```
  procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
```

```
  procedure btnCancelClick(Sender: TObject);
```

```
  procedure btnOKClick(Sender: TObject);
```

```
  procedure btnUpClick(Sender: TObject);
```

```
  procedure btnDownClick(Sender: TObject);
```

```
  procedure FormClose(Sender: TObject; var Action: TCloseAction);
```

```
  procedure FormShow(Sender: TObject);
```

```
private
```

```
  { Private declarations }
```

```
  SelectedID: Integer;
```

```

public
{ Public declarations }
function ShowList(Var aSelectedID:Integer):Integer;
end;

var
LessonPickerForm: TLessonPickerForm;

implementation

uses db,GeoDataUnit;

{$R *.dfm}

/////////////////////////////////////////////////////////////////
/// Όνομα: ShowList
/// Τύπος: Public Function
/// Επιστρέφει:
/// Integer - mrOk ή mrCancel, ανάλογα με την επιλογή του χρήστη
/// Παράμετροι:
/// var SelectedID:Integer - Αρχικά επιλεγμένο μάθημα
/// Περιγραφή:
/// Η ShowList καλείται από το υπόλοιπο πρόγραμμα για να εμφανιστεί στην οθόνη
/// η λίστα μαθημάτων. Ο χρήστης μπορεί να αλλάξει την σειρά ταξινόμησης.
/// Αν ο χρήστης πατήσει το πλήκτρο OK, οι αλλαγές αποθηκεύονται και στην
/// μεταβλητή παραμέτρου SelectedID αποθηκεύεται το τελευταίο μάθημα που επελέγη
/////////////////////////////////////////////////////////////////
function TLessonPickerForm.ShowList(Var aSelectedID:Integer):Integer;
begin
SelectedID := aSelectedID;
Result := Self.ShowModal;
if Result = mrOk then aSelectedID := SelectedID;
end;

/////////////////////////////////////////////////////////////////
/// Όνομα: btnCancelClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnCancel
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο Cancel, ακυρώνονται οι τυχόν αλλαγές
/////////////////////////////////////////////////////////////////
procedure TLessonPickerForm.btnCancelClick(Sender: TObject);
begin
if (GeoData.tbLessonsCache.State in dsEditModes) then
GeoData.tbLessonsCache.Cancel;
if (GeoData.tbLessonsCache.ChangeCount > 0) then
GeoData.tbLessonsCache.CancelUpdates;
ModalResult := mrCancel;
end;

/////////////////////////////////////////////////////////////////
/// Όνομα: btnDownClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnDown
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:

```



```

/// Όταν ο χρήστης πατήσει το πλήκτρο btnDown (κάτω βέλος), το τρέχον επιλεγμένο
/// μάθημα μετακινείται μια θέση κάτω
////////////////////////////////////
procedure TLessonPickerForm.btnDownClick(Sender: TObject);
var CurID: Integer;
begin
  GeoData.tbLessonsCache.DisableControls;
  CurID := GeoData.tbLessonsCacheID.AsInteger;
  GeoData.tbLessonsCache.Next;
  if not GeoData.tbLessonsCache.Eof then
  begin
    GeoData.tbLessonsCache.Edit;
    GeoData.tbLessonsCacheOrdering.AsInteger :=
      GeoData.tbLessonsCacheOrdering.AsInteger - 1;
    GeoData.tbLessonsCache.Post;
    GeoData.tbLessonsCache.Locate('ID', CurID, []);
    GeoData.tbLessonsCache.Edit;
    GeoData.tbLessonsCacheOrdering.AsInteger :=
      GeoData.tbLessonsCacheOrdering.AsInteger + 1;
    GeoData.tbLessonsCache.Post;
  end;
  GeoData.tbLessonsCache.EnableControls;
end;

////////////////////////////////////
/// Όνομα: btnOKClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnOK
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο OK, επικυρώνονται οι αλλαγές που
/// πραγματοποιήθηκαν.
////////////////////////////////////
procedure TLessonPickerForm.btnOKClick(Sender: TObject);
begin
  if (GeoData.tbLessonsCache.State in dsEditModes) then
    GeoData.tbLessonsCache.Post;
  if (GeoData.tbLessonsCache.ChangeCount > 0) then
    GeoData.tbLessonsCache.ApplyUpdates(0);
  ModalResult := mrOk;
end;

////////////////////////////////////
/// Όνομα: btnUpClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnUp
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnUp (άνω βέλος), το τρέχον επιλεγμένο
/// μάθημα μετακινείται μια θέση πάνω
////////////////////////////////////
procedure TLessonPickerForm.btnUpClick(Sender: TObject);
var CurID: Integer;
begin
  GeoData.tbLessonsCache.DisableControls;
  CurID := GeoData.tbLessonsCacheID.AsInteger;
  GeoData.tbLessonsCache.Prior;

```

```

if not GeoData.tbLessonsCache.Bof then
begin
  GeoData.tbLessonsCache.Edit;
  GeoData.tbLessonsCacheOrdering.AsInteger :=
    GeoData.tbLessonsCacheOrdering.AsInteger + 1;
  GeoData.tbLessonsCache.Post;
  GeoData.tbLessonsCache.Locate('ID', CurID, []);
  GeoData.tbLessonsCache.Edit;
  GeoData.tbLessonsCacheOrdering.AsInteger :=
    GeoData.tbLessonsCacheOrdering.AsInteger - 1;
  GeoData.tbLessonsCache.Post;
end;
GeoData.tbLessonsCache.EnableControls;
end;

/////////////////////////////////////////////////////////////////
/// Όνομα: FormClose
/// Τύπος: Event Handler
/// Πυροδότηση:
///   Κατά το κλείσιμο της φόρμας
/// Παράμετροι:
///   Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
///   var Action:TCloseAction - δεν χρησιμοποιείται
/// Περιγραφή:
///   Όταν κλείνει η φόρμα, πραγματοποιείται και το κλείσιμο του προσωρινού
///   πίνακα tbLessonsCache
/////////////////////////////////////////////////////////////////
procedure TLessonPickerForm.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin
  GeoData.tbLessonsCache.Close;
end;

/////////////////////////////////////////////////////////////////
/// Όνομα: FormCloseQuery
/// Τύπος: Event Handler
/// Πυροδότηση:
///   Έλεγχος πριν από το κλείσιμο της φόρμας
/// Παράμετροι:
///   Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
///   var CanClose:Boolean - Τίθεται False αν ο χρήστης επιλέξει να μην κλείσει
/// Περιγραφή:
///   Εμφανίζεται ένα παράθυρο προειδοποίησης στον χρήστη αν έχουν πραγματοποιηθεί
///   αλλαγές, ώστε να επιλέξει αν θέλει να αποθηκευτούν ή όχι
/////////////////////////////////////////////////////////////////
procedure TLessonPickerForm.FormCloseQuery(Sender: TObject;
  var CanClose: Boolean);
var DlgResult: Integer;
begin
  CanClose := True;
  if (GeoData.tbLessonsCache.State in dsEditModes) or
    (GeoData.tbLessonsCache.ChangeCount > 0)
  then begin
    DlgResult := MessageDlg ('Πραγματοποιήσατε αλλαγές!' + chr(13) +
      'Αποθήκευση;', mtWarning, mbYesNoCancel, 0);
    if DlgResult = mrYes then btnOKClick(Self);
    if DlgResult = mrNo then btnCancelClick(Self);
    if DlgResult = mrCancel then CanClose := False;
  end;
end;
end;

```

```

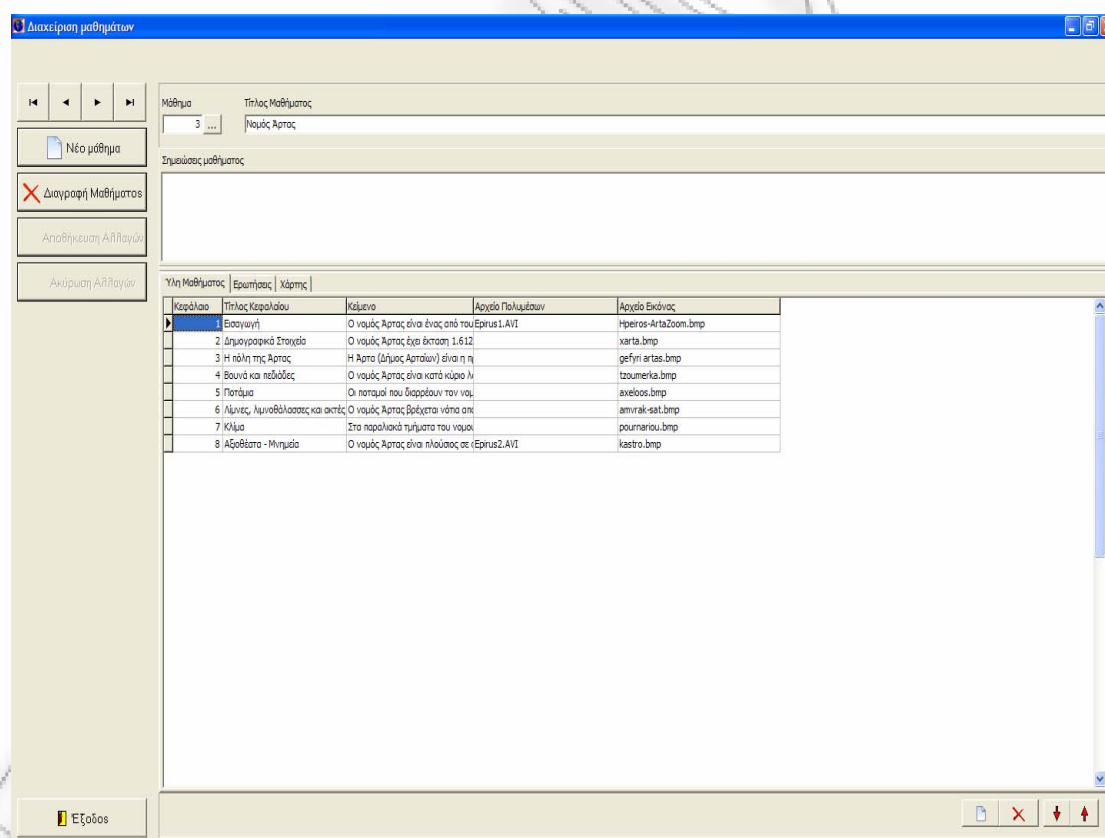
////////////////////////////////////
/// Όνομα: FormShow
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κατά το άνοιγμα της φόρμας
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ανοίγει η φόρμα, πραγματοποιείται και το άνοιγμα του προσωρινού
/// πίνακα tbLessonsCache
////////////////////////////////////
procedure TLessonPickerForm.FormShow(Sender: TObject);
begin
  GeoData.tbLessonsCache.Open;
  GeoData.tbLessonsCache.Locate('ID',SelectedID,[]);
end;

end.

```

### -Υπόλοιπες φόρμες του Administrator

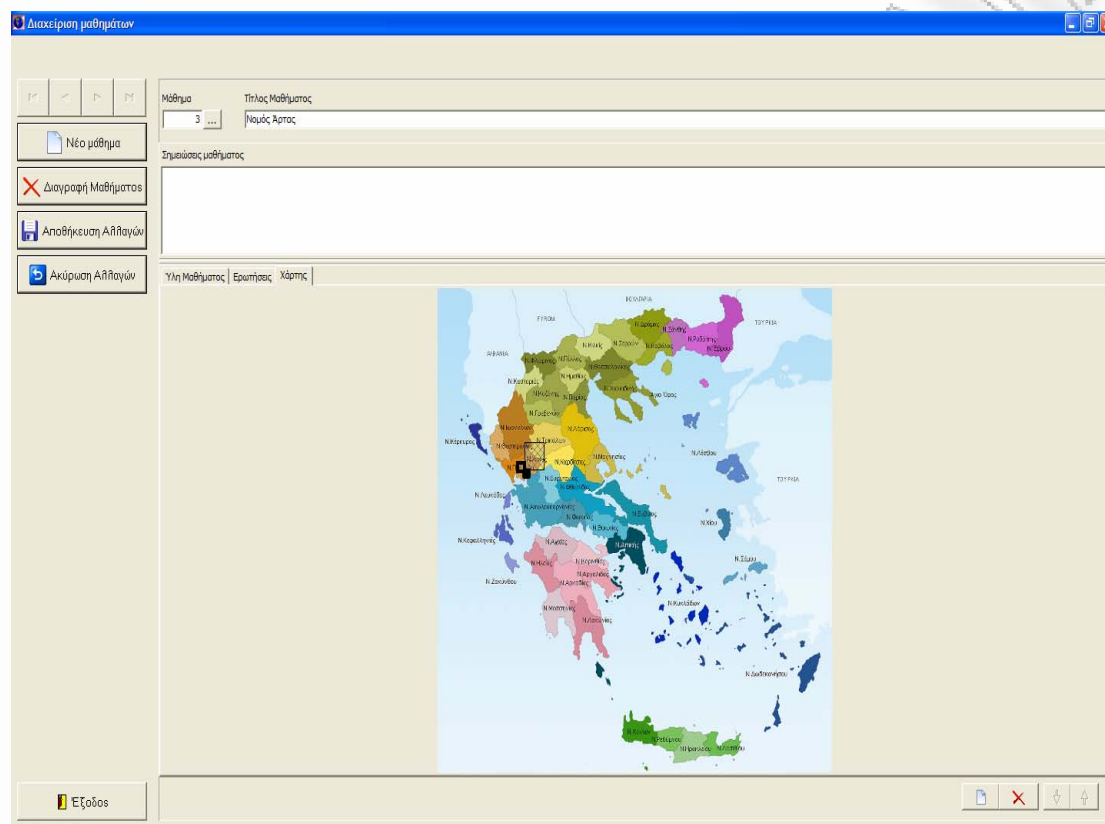
Πιο κάτω βρίσκεται ένα grid όπου ο administrator μπορεί να καταχωρήσει κάποιες σημειώσεις που αφορούν το μάθημα, ώστε να τις χρησιμοποιήσει ο ίδιος ή κάποιος καθηγητής.



Όπως φαίνεται και στην παραπάνω οθόνη, ο administrator μπορεί να καθορίσει την σειρά κάθε κεφαλαίου του μαθήματος ή των ερωτήσεων, να αλλάξει τον τίτλο και το κείμενο που αναφέρεται στο κάθε κεφάλαιο και να προσθέσει ή να αφαιρέσει αρχείο πολυμέσων και εικόνας. Όλα τα παραπάνω πραγματοποιούνται με τη χρήση των κουμπιών που βρίσκονται στο κάτω μέρος της οθόνης, δεξιά. Με το πρώτο κουμπί γίνεται προσθήκη νέου κεφαλαίου και προσθήκη αρχείου πολυμέσων ή εικόνας. Με το δεύτερο κουμπί γίνεται διαγραφή κεφαλαίου και διαγραφή αρχείου πολυμέσων ή εικόνας. Με τα δύο επόμενα κουμπιά, που αποτελούν

βέλη, πραγματοποιείται η αλλαγή σειράς των κεφαλαίων. Πραγματοποιώντας οποιαδήποτε αλλαγή, ενεργοποιούνται τα κουμπιά «Αποθήκευση Αλλαγών» και «Ακύρωση Αλλαγών».

Επίσης, ο administrator μπορεί να καθορίσει ή να αλλάξει την τοποθεσία του νομού που αφορά το μάθημα στον χάρτη :



Παρακάτω παρατίθεται ο κώδικας υλοποίησης των παραπάνω φορμών :

```
unit EditLessonUnit;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, db,  
Dialogs, ExtCtrls, StdCtrls, DBCtrls, Mask, Buttons, DBCGrids, Grids, DBGrids,  
ComCtrls, GlobalsUnit, RpDefine, RpBase, RpSystem;
```

```
type
```

```
TEditLessonForm = class(TForm)
```

```
  btnUp: TBitBtn;
```

```
  btnDown: TBitBtn;
```

```
  Label1: TLabel;
```

```
  DBEdit1: TDBEdit;
```

```
  Label2: TLabel;
```

```
  Bevel1: TBevel;
```

```
  DBEdit2: TDBEdit;
```

```
  Label4: TLabel;
```

```
  btnDeleteChapter: TBitBtn;
```

```
  btnInsertChapter: TBitBtn;
```

```
  LessonNavigator: TDBNavigator;
```

```
  btnChangeLessonOrder: TBitBtn;
```

```

btnNewLesson: TBitBtn;
btnDelLesson: TBitBtn;
btnSave: TBitBtn;
btnCancel: TBitBtn;
btnClose: TBitBtn;
OpenDialog: TOpenDialog;
Panel1: TPanel;
ChaptersQuestionsTab: TPageControl;
ChaptersSheet: TTabSheet;
ChaptersGrid: TDBGrid;
QuestionsSheet: TTabSheet;
Splitter1: TSplitter;
Panel2: TPanel;
DBMemo1: TDBMemo;
Label3: TLabel;
QuestionsGrid: TDBGrid;
MapSheet: TTabSheet;
MapImage: TImage;
BoxShape: TShape;
CornerShape: TShape;
procedure ChaptersQuestionsTabResize(Sender: TObject);
procedure MapImageClick(Sender: TObject);
procedure BoxShapeDragOver(Sender, Source: TObject; X, Y: Integer;
  State: TDragState; var Accept: Boolean);
procedure MapImageDragDrop(Sender, Source: TObject; X, Y: Integer);
procedure MapImageDragOver(Sender, Source: TObject; X, Y: Integer;
  State: TDragState; var Accept: Boolean);
procedure BoxShapeStartDrag(Sender: TObject; var DragObject: TDragObject);
procedure ChaptersQuestionsTabChange(Sender: TObject);
procedure ChaptersGridKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure btnChangeLessonOrderClick(Sender: TObject);
procedure ChaptersGridDbClick(Sender: TObject);
procedure btnUpClick(Sender: TObject);
procedure btnDownClick(Sender: TObject);
procedure btnDeleteChapterClick(Sender: TObject);
procedure btnInsertChapterClick(Sender: TObject);
procedure ChaptersGridEditButtonClick(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
procedure btnCloseClick(Sender: TObject);
procedure btnCancelClick(Sender: TObject);
procedure btnSaveClick(Sender: TObject);
procedure btnNewLessonClick(Sender: TObject);
procedure btnDelLessonClick(Sender: TObject);
procedure FormShow(Sender: TObject);
private
{ Private declarations }
Boxes: TMapBoxCollection;
SelectedBox: Integer;
procedure BoxClick(aBoxIndex, aBoxTag: Integer);
procedure BoxMouseOver(aBoxIndex, aBoxTag: Integer);
procedure SelectBox(aBox: Integer);
procedure DeSelectBox;
procedure NewMapBox(aTop, aLeft, aWidth, aHeight: Integer);
procedure DeleteSelectedMapBox;
procedure ReadDBBoxes;
procedure SaveDBBoxes;
procedure srcLessonsDataChange(Sender: TObject; Field: TField);
procedure srcChaptersStateChange(Sender: TObject);
procedure tbChaptersBeforeInsert(DataSet: TDataSet);

```

```

public
  { Public declarations }
end;

var
  EditLessonForm: TEditLessonForm;

implementation

uses GeoDataUnit, MemoEditUnit, LessonPickerUnit, AnswerPickerUnit;

{$R *.dfm}

var BoxesChanged: Boolean;

procedure TEditLessonForm.srcLessonsDataChange(Sender: TObject; Field: TField);
////////////////////////////////////////////////////
// όνομα: srcLessonsDataChange
// τύπος: event handler
// πυροδότηση:
//   όταν αλλάξει η τρέχουσα εγγραφή του tbLessons, δηλαδή όταν ο χρήστης
//   μεταβεί σε άλλο μάθημα, ή όταν πραγματοποιηθούν αλλαγές στα δεδομένα
// παράμετροι:
//   Sender:TObject - Το αντικείμενο που πυροδότησε το συμβάν (αγνοείται)
//   Field:TField - Το πεδίο που αλλάχθηκε (αγνοείται)
// περιγραφή:
//   Καλείται η ReadDBBoxes ώστε να ενημερωθεί ο χάρτης του τρέχοντος μαθήματος.
////////////////////////////////////////////////////
begin
  if (GeoData.tbLessons.State = dsBrowse) then
    begin
      ReadDBBoxes;
    end;
end;

procedure TEditLessonForm.srcChaptersStateChange(Sender: TObject);
////////////////////////////////////////////////////
// όνομα: srcChaptersStateChange
// τύπος: event handler
// πυροδότηση:
//   Όταν αλλάζει η κατάσταση λειτουργίας του πίνακα tbChapters, δηλαδή:
//   1) Όταν αρχίζει ο χρήστης να πραγματοποιεί αλλαγές (από κατάσταση dsBrowse σε
//   dsEdit)
//   1) Όταν ο χρήστης δημιουργεί καινούριο κεφάλαιο (dsBrowse σε dsInsert)
//   3) Όταν επιβεβαιώνει ή ματαιώνει κάποιες αλλαγές (dsEdit ή dsInsert σε dsBrowse)
// παράμετροι:
//   Sender:TObject - Το αντικείμενο που πυροδότησε το συμβάν (αγνοείται)
// περιγραφή:
//   Ενημερώνεται η ιδιότητα enabled των buttons Save,Cancel, καθώς και του
//   LessonNavigator, ανάλογα με την τρέχουσα κατάσταση λειτουργίας
////////////////////////////////////////////////////
begin
  if (GeoData.tbLessons.State in dsEditModes) or
    (GeoData.tbChapters.State in dsEditModes) or
    (GeoData.tbQuestions.State in dsEditModes) or
    (GeoData.tbChapters.ChangeCount > 0) or
    (GeoData.tbQuestions.ChangeCount > 0) or
    BoxesChanged
  then begin
    btnSave.Enabled := True;
    btnCancel.Enabled := True;
  end;
end;

```

```

LessonNavigator.Enabled := False;
end else
begin
  btnSave.Enabled := False;
  btnCancel.Enabled := False;
  LessonNavigator.Enabled := True;
end;
end;

procedure TEditLessonForm.BoxMouseOver(aBoxIndex,aBoxTag:Integer);
begin

end;

procedure TEditLessonForm.BoxClick(aBoxIndex,aBoxTag:Integer);
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
/// όνομα: BoxClick
/// τύπος: event handler
/// πυροδότηση:
/// Όταν ο χρήστης κάνει κλικ σε ένα από τα τετράγωνα (Boxes) στον χάρτη
/// του μαθήματος
/// παράμετροι:
/// aBoxIndex:Integer - Ο κωδικός του Box
/// aBoxTag:Integer - Η τιμή της ιδιότητας tag του επιλεγμένου Box (αγνοείται)
/// περιγραφή:
/// Καλείται η SelectBox για το τετράγωνο που επελέγη
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
begin
  SelectBox(aBoxIndex);
end;

procedure TEditLessonForm.SelectBox(aBox: Integer);
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
/// όνομα: SelectBox
/// τύπος: procedure
/// παράμετροι:
/// aBox:Integer - Ο κωδικός του Box που θέλουμε να γίνει ενεργό
/// περιγραφή:
/// Το τετράγωνο με κωδικό την παράμετρο aBox γίνεται το ενεργό
/// τετράγωνο στον χάρτη του μαθήματος. Το πλαίσιο του
/// σχεδιάζεται εντονότερα, και προστίθεται στην κάτω δεξιά
/// άκρη του τό τετραγωνίδιο μεταβολής μεγέθους (CornerShape)
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
var S:TShape;
begin
  DeselectBox;
  SelectedBox := aBox;
  S := (Boxes.Items[SelectedBox] as TMapBoxItem).BoxShape;
  S.BringToFront;
  S.Pen.Width := 4;
  CornerShape.Left := S.Left + S.Width - (CornerShape.Width DIV 2);
  CornerShape.Top := S.Top + S.Height - (CornerShape.Height DIV 2);
  CornerShape.Visible := True;
  CornerShape.BringToFront;
end;

procedure TEditLessonForm.DeSelectBox;
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
/// όνομα: DeSelectBox
/// τύπος: procedure
/// παράμετροι:

```

```

/// (καμία)
/// περιγραφή:
/// Ακύρωση της επιλογής του ενεργού τετραγώνου στον χάρτη.
/// Αφαιρείται το έντονο πλαίσιο αν υπάρχει, και το CornerShape
/// καθίσταται αόρατο (Visible = False)
/////////////////////////////////////////////////////////////////
begin
  if SelectedBox > -1 then
    begin
      with (Boxes.Items[SelectedBox] as TMapBoxItem).BoxShape do
        begin
          Pen.Width := 1;
          CornerShape.Visible := False;
        end;
      end;
      SelectedBox := -1;
    end;

procedure TEditLessonForm.SaveDBBoxes;
/////////////////////////////////////////////////////////////////
/// όνομα: SaveDBBoxes
/// τύπος: procedure
/// παράμετροι:
/// (καμία)
/// περιγραφή:
/// Αποθηκεύονται στην βάση όλα τα τετράγωνα(boxes) που έχουν σχεδιαστεί
/// στον χάρτη
/////////////////////////////////////////////////////////////////
var i:Integer;
begin
  GeoData.tbMapBoxes.Filter := 'LessonID='+GeoData.tbLessonsID.AsString;
  GeoData.tbMapBoxes.Filtered := True;
  GeoData.tbMapBoxes.Open;
  //Διαγραφή όλων των boxes που είναι ήδη αποθηκευμένα
  While not GeoData.tbMapBoxes.EOF do
    begin
      GeoData.tbMapBoxes.Delete;
    end;
  //Αποθήκευση των καινούριων
  For i:= 0 to Boxes.Count - 1 do
    begin
      GeoData.tbMapBoxes.Insert;
      GeoData.tbMapBoxesLessonID.AsInteger := GeoData.tbLessonsID.AsInteger;
      GeoData.tbMapBoxesTop.AsInteger := (Boxes.Items[i] as TMapBoxItem).Top;
      GeoData.tbMapBoxesLeft.AsInteger := (Boxes.Items[i] as TMapBoxItem).Left;
      GeoData.tbMapBoxesWidth.AsInteger := (Boxes.Items[i] as TMapBoxItem).Width;
      GeoData.tbMapBoxesHeight.AsInteger := (Boxes.Items[i] as TMapBoxItem).Height;
      GeoData.tbMapBoxes.Post;
    end;
  GeoData.tbMapBoxes.Close;
end;

procedure TEditLessonForm.ReadDBBoxes;
/////////////////////////////////////////////////////////////////
/// όνομα: ReadDBBoxes
/// τύπος: procedure
/// παράμετροι:
/// (καμία)
/// περιγραφή:
/// Διαβάζονται από την βάση τα τετράγωνα του χάρτη για το συγκεκριμένο μάθημα
/////////////////////////////////////////////////////////////////

```



```

begin
  CornerShape.Visible := False;
  if Assigned(Boxes) then
    begin
      Boxes.Clear;
      GeoData.tbMapBoxes.Filter := 'LessonID='+GeoData.tbLessonsID.AsString;
      GeoData.tbMapBoxes.Filtered := True;
      GeoData.tbMapBoxes.Open;
      While not GeoData.tbMapBoxes.EOF do
        begin
          NewMapBox(
            GeoData.tbMapBoxesTop.AsInteger,
            GeoData.tbMapBoxesLeft.AsInteger,
            GeoData.tbMapBoxesWidth.AsInteger,
            GeoData.tbMapBoxesHeight.AsInteger);
          GeoData.tbMapBoxes.Next;
        end;
      GeoData.tbMapBoxes.Close;
      SelectedBox := -1;
      BoxesChanged := False;
    end;
end;

procedure TEditLessonForm.NewMapBox(aTop,aLeft,aWidth,aHeight:Integer);
////////////////////////////////////////////////////
// όνομα: NewMapBox
// τύπος: procedure
// παράμετροι:
//   aTop,aLeft: Integer - Συντεταγμένες σε pixels της πάνω-αριστερής γωνίας
//   aWidth,aHeight: Integer - Πλάτος και ύψος σε pixels
// περιγραφή:
//   Ορισμός ενός νέου τετραγώνου στις δεδομένες συντεταγμένες, και με
//   το δεδομένο μέγεθος.
////////////////////////////////////////////////////
var NewBox:Integer;
begin
  NewBox := Boxes.NewBox(aTop,aLeft,aWidth,aHeight,-1,True);
  SelectBox(NewBox);
  with (Boxes.Items[SelectedBox] as TMapBoxItem).BoxShape do
    begin
      OnStartDrag := BoxShapeStartDrag;
      OnDragOver := BoxShapeDragOver;
    end;
  BoxesChanged := True;
end;

procedure TEditLessonForm.DeleteSelectedMapBox;
////////////////////////////////////////////////////
// όνομα: DeleteSelectedMapBox
// τύπος: procedure
// παράμετροι:
//   (καμία)
// περιγραφή:
//   Διαγραφή του επιλεγμένου τετραγώνου από τον χάρτη.
////////////////////////////////////////////////////
var OldSelection: Integer;
begin
  if SelectedBox > -1 then
    begin
      OldSelection := SelectedBox;
      DeselectBox;
    end;
end;

```

```

    Boxes.Delete(OldSelection);
end;
BoxesChanged := True;
end;

procedure TEditLessonForm.btnInsertChapterClick(Sender: TObject);
////////////////////////////////////////////////////////////////////
// Όνομα: btnInsertChapterClick
// Τύπος: Event Handler
// Πυροδότηση:
// Κλικ του πλήκτρου btnInsertChapter
// Παράμετροι:
// Sender: TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
// Περιγραφή:
// Όταν ο χρήστης πατήσει το πλήκτρο btnInsertChapter, προστίθεται ένα νέο
// κεφάλαιο στο τρέχον μάθημα.
////////////////////////////////////////////////////////////////////
var LastChapter: Integer;
    RndID: Integer;
    DS: TDataSet;
begin

    if ChaptersQuestionsTab.ActivePage = MapSheet then
        begin
            NewMapBox(450,450,100,100);
        end else
        begin
            if ChaptersQuestionsTab.ActivePage = QuestionsSheet then
                begin
                    DS := GeoData.tbQuestions;
                end else
                begin
                    DS := GeoData.tbChapters;
                end;
            end;
            try
                if DS.State in dsEditModes then DS.Post;
                DS.Last;
                LastChapter := DS.FieldByName('Ordering').AsInteger;
                DS.Append;
                RndID := -1 - random(maxint - 1);
                DS.FieldByName('ID').AsInteger := RndID;
                DS.FieldByName('Ordering').AsInteger := LastChapter + 1;
            except
                on E:Exception do
                    begin
                        ShowMessage('Δεν ήταν εφικτή η εισαγωγή!' + chr(13) +
                            'Μήνυμα λάθους:' + chr(13) +
                            "" + e.Message + "");
                    end;
                end;
            end;

            srcChaptersStateChange(self);

        end;

procedure TEditLessonForm.BoxShapeDragOver(Sender, Source: TObject; X,
Y: Integer; State: TDragState; var Accept: Boolean);
////////////////////////////////////////////////////////////////////
// Όνομα: BoxShapeDragOver

```

```

/// Τύπος: Event Handler
/// Πυροδότηση:
///   Dragging (σύρσιμο) κάποιου αντικείμενου πάνω από ένα από τα τετράγωνα του χάρτη
/// Παράμετροι:
///   Sender: TObject - Το αντικείμενο που σύρεται
///   X, Y: Integer - Η συντεταγμένες του δείκτη του ποντικιού σε σχέση με την
///                   πάνω-αριστερή γωνία του τετραγώνου πάνω από το οποίο σύρεται
///   State: TDragState - Ο τρόπος με τον οποίο γίνεται το dragging (αγνοείται)
///   var Accept: Boolean - True = το συρόμενο αντικείμενο μπορεί να γίνει δεκτό
///                       - False = το συρόμενο αντικείμενο είναι ακατάλληλο
/// Περιγραφή:
///   Όταν ο χρήστης σύρει το ποντίκι πάνω από κάποιο τετράγωνο, καλείται
///   απλά ο αντίστοιχος event handler του χάρτη που διαχειρίζεται τις
///   μετακινήσεις των τετραγώνων από τον χρήστη.
////////////////////////////////////
begin
  MapImageDragOver(Source, Source, (Sender as TShape).Left + X,
    (Sender as TShape).Top + Y, State, Accept);
end;

procedure TEditLessonForm.BoxShapeStartDrag(Sender: TObject;
  var DragObject: TDragObject);
////////////////////////////////////
/// Όνομα: BoxShapeDragOver
/// Τύπος: Event Handler
/// Πυροδότηση:
///   Dragging (σύρσιμο) του ποντικιού πάνω από ένα από τα τετράγωνα του χάρτη
/// Παράμετροι:
///   Sender: TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
///   DragObject: TDragObject - Στην παράμετρο αυτή μπορεί να τεθεί κάποιο
///   αντικείμενο εικόνας που θα εμφανίζεται καθώς σύρεται το ποντίκι
/// Περιγραφή:
///   Απλά τίθεται η παράμετρος DragObject ως NIL που σημαίνει ότι πρόκειται
///   να συρθεί το ίδιο το αντικείμενο
////////////////////////////////////
begin
  DragObject := NIL;
end;

procedure TEditLessonForm.btnCancelClick(Sender: TObject);
////////////////////////////////////
/// Όνομα: btnCancelClick
/// Τύπος: Event Handler
/// Πυροδότηση:
///   Κλικ του πλήκτρου btnCancel
/// Παράμετροι:
///   Sender: TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
///   Όταν ο χρήστης πατήσει το πλήκτρο Cancel, ακυρώνονται οι τυχόν εκκρεμείς
///   αλλαγές και επαναφέρονται τα αποθηκευμένα δεδομένα
////////////////////////////////////
begin
  try
    if (GeoData.tbLessons.State in dsEditModes) then
      GeoData.tbLessons.Cancel;
    if (GeoData.tbChapters.State in dsEditModes) then
      GeoData.tbChapters.Cancel;
    if (GeoData.tbQuestions.State in dsEditModes) then
      GeoData.tbQuestions.Cancel;
    if (GeoData.tbChapters.ChangeCount > 0) then
      GeoData.tbChapters.CancelUpdates;
  end;
end;

```

```

if (GeoData.tbQuestions.ChangeCount > 0 ) then
  GeoData.tbQuestions.CancelUpdates;
  GeoData.tbChapters.Refresh;
  GeoData.tbQuestions.Refresh;
  ReadDBBoxes;
  srcChaptersStateChange(Self);
except
  on E:Exception do
  begin
    ShowMessage('Δεν ήταν εφικτή η ακύρωση των αλλαγών!'+chr(13)+
      'Μήνυμα λάθους:'+chr(13)+
      '"+e.Message+"'");
  end;
end;
end;

procedure TEditLessonForm.btnChangeLessonOrderClick(Sender: TObject);
/////////////////////////////////////////////////////////////////
/// Όνομα: btnChangeLessonOrderClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnChangeLessonOrder
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnChangeLessonOrder εμφανίζεται η λίστα
/// μαθημάτων (LessonPickerForm), όπου και μπορεί να αλλάξει την σειρά τους
/////////////////////////////////////////////////////////////////
var CurID: Integer;
begin
  GeoData.tbLessons.DisableControls;
  CurID := GeoData.tbLessonsID.asInteger;
  if LessonPickerForm.ShowList(CurID)= mrOk then
  begin
    GeoData.tbLessons.Refresh;
    GeoData.tbLessons.Locate('ID',CurID,[]);
  end;
  GeoData.tbLessons.EnableControls;
end;

procedure TEditLessonForm.btnCloseClick(Sender: TObject);
/////////////////////////////////////////////////////////////////
/// Όνομα: btnCloseClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnClose
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnClose, κλείνει η φόρμα
/////////////////////////////////////////////////////////////////
begin
  Close;
end;

procedure TEditLessonForm.btnDeleteChapterClick(Sender: TObject);
/////////////////////////////////////////////////////////////////
/// Όνομα: btnDeleteChapterClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnDeleteChapter

```

```

/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnDeleteChapter, διαγράφεται το επιλεγμένο
/// κεφάλαιο στο τρέχον μάθημα.
////////////////////////////////////
var DS:TDataSet;
    CurlID:Integer;
begin

    if ChaptersQuestionsTab.ActivePage = MapSheet then
    begin
        DeleteSelectedMapBox;
    end else
    begin
        if ChaptersQuestionsTab.ActivePage = QuestionsSheet then
        begin
            DS := GeoData.tbQuestions;
        end else
        begin
            DS := GeoData.tbChapters;
        end;
    try
        if (not DS.IsEmpty) then
        begin
            if DS.State = dsInsert then
                DS.Cancel
            else
            begin
                DS.Delete;
                CurlID := DS.FieldByName('ID').asInteger;
                FixOrdering(DS);
                DS.Locate('ID', CurlID, []);
            end;
        end;
    except
        on E:Exception do
        begin
            ShowMessage('Δεν ήταν εφικτή η διαγραφή!'+chr(13)+
                'Μήνυμα λάθους:'+chr(13)+
                '''+e.Message+''');
        end;
    end;
end;

    srcChaptersStateChange(self);

end;

procedure TEditLessonForm.btnDelLessonClick(Sender: TObject);
////////////////////////////////////
/// Όνομα: btnDelLessonClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnDelLesson
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnDelLesson διαγράφεται το τρέχον μάθημα,
/// εφόσον δοθεί η επιβεβαίωση από τον χρήστη.

```

```

////////////////////////////////////
var CurlID: Integer;
begin
  try
    if (not GeoData.tbLessons.IsEmpty) and
      (MessageDlg('Θέλετε σίγουρα να διαγράψετε το συγκεκριμένο μάθημα;',
        mtConfirmation, mbYesNo, 0) = mrYes) then
      begin
        GeoData.tbChapters.DisableControls;
        GeoData.tbQuestions.DisableControls;
        GeoData.tbLessons.DisableControls;
        if GeoData.tbChapters.State in dsEditModes then
          GeoData.tbChapters.Cancel;
        GeoData.tbChapters.First;
        while not GeoData.tbChapters.Eof do
          begin
            GeoData.tbChapters.Delete;
          end;
        GeoData.tbChapters.ApplyUpdates(0);
        if GeoData.tbQuestions.State in dsEditModes then
          GeoData.tbQuestions.Cancel;
        GeoData.tbQuestions.First;
        while not GeoData.tbQuestions.Eof do
          begin
            GeoData.tbQuestions.Delete;
          end;
        GeoData.tbQuestions.ApplyUpdates(0);
        GeoData.tbLessons.Delete;
        CurlID := GeoData.tbLessonsID.Value;
        FixOrdering(GeoData.tbLessons);
        GeoData.tbLessons.Locate('ID', CurlID, []);
        GeoData.tbLessons.EnableControls;
        GeoData.tbChapters.EnableControls;
        GeoData.tbQuestions.EnableControls;
      end;
    except
      on E:Exception do
        begin
          ShowMessage('Δεν ήταν εφικτή η διαγραφή!' + chr(13) +
            'Μήνυμα λάθους:' + chr(13) +
            "" + e.Message + "");
        end;
      end;
    end;
  end;

  procedure TEditLessonForm.btnDownClick(Sender: TObject);
  //////////////////////////////////////
  /// Όνομα: btnDownClick
  /// Τύπος: Event Handler
  /// Πυροδότηση:
  /// Κλικ του πλήκτρου btnDown
  /// Παράμετροι:
  /// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
  /// Περιγραφή:
  /// Όταν ο χρήστης πατήσει το πλήκτρο btnDown, η επιλεγμένη ερώτηση ή το
  /// επιλεγμένο κεφάλαιο (ανάλογα με το ποιο από τα δύο tabs είναι ενεργό)
  /// μετακινείται μία θέση κάτω στη σειρά
  //////////////////////////////////////
  var CurlID: Integer;
      DS:TDataSet;
  begin

```

```

DS := NIL;

if ChaptersQuestionsTab.ActivePage = QuestionsSheet then
begin
  DS := GeoData.tbQuestions;
end else
if ChaptersQuestionsTab.ActivePage = ChaptersSheet then
begin
  DS := GeoData.tbChapters;
end;

if Assigned(DS) then
begin
  DS.DisableControls;
  CurID := DS.FieldByName('ID').AsInteger;
  DS.Next;
  if not DS.Eof then
  begin
    DS.Edit;
    DS.FieldByName('Ordering').AsInteger :=
      DS.FieldByName('Ordering').AsInteger - 1;
    DS.Post;
    DS.Locate('ID', CurID, []);
    DS.Edit;
    DS.FieldByName('Ordering').AsInteger :=
      DS.FieldByName('Ordering').AsInteger + 1;
    DS.Post;
  end;
  DS.EnableControls;
  srcChaptersStateChange(self);
end;
end;

procedure TEditLessonForm.btnNewLessonClick(Sender: TObject);
////////////////////////////////////////////////////////////////
/// Όνομα: btnNewLessonClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnNewLesson
/// Παράμετροι:
/// Sender: TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnNewLesson, δημιουργείται ένα νέο
/// κενό μάθημα.
////////////////////////////////////////////////////////////////
var LastLesson: Integer;
begin
  try
    if GeoData.tbLessons.State = dsBrowse then
    begin
      GeoData.tbLessons.Last;
      LastLesson := GeoData.tbLessonsOrdering.AsInteger;
      GeoData.tbLessons.Append;
      GeoData.tbLessonsOrdering.AsInteger := LastLesson + 1;
    end;
  except
    on E: Exception do
    begin
      ShowMessage('Δεν ήταν εφικτή η εισαγωγή!' + chr(13) +
        'Μήνυμα λάθους:' + chr(13) +
        "" + e.Message + "");
    end;
  end;
end;

```

```

end;
end;
end;

procedure TEditLessonForm.FormClose(Sender: TObject; var Action: TCloseAction);
////////////////////////////////////////////////////
// Όνομα: FormClose
// Τύπος: Event Handler
// Πυροδότηση:
// Κατά το κλείσιμο της φόρμας
// Παράμετροι:
// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
// var Action:TCloseAction - δεν χρησιμοποιείται
// Περιγραφή:
// Όταν κλείνει η φόρμα, πραγματοποιείται και το κλείσιμο όλων των
// πινάκων που χρησιμοποιούνται από αυτήν
////////////////////////////////////////////////////
begin
  GeoData.tbLessons.Close;
  GeoData.tbChaptersMain.Close;
  GeoData.tbChapters.Close;
  GeoData.tbExternals.Close;
  GeoData.srcLessons.OnDataChange := NIL;
  GeoData.srcLessons.OnStateChange := NIL;
  GeoData.srcChapters.OnStateChange := NIL;
  GeoData.tbChapters.BeforeInsert := NIL;
  GeoData.tbQuestions.BeforeInsert := NIL;
  Boxes.Free;
  Boxes := NIL;
end;

procedure TEditLessonForm.FormCloseQuery(Sender: TObject;
////////////////////////////////////////////////////
// Όνομα: FormCloseQuery
// Τύπος: Event Handler
// Πυροδότηση:
// Έλεγχος πριν από το κλείσιμο της φόρμας
// Παράμετροι:
// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
// var CanClose:Boolean - Τίθεται False αν ο χρήστης επιλέξει να μην κλείσει
// Περιγραφή:
// Εμφανίζεται ένα παράθυρο προειδοποίησης στον χρήστη αν έχουν πραγματοποιηθεί
// αλλαγές, ώστε να επιλέξει αν θέλει να αποθηκευτούν ή όχι
////////////////////////////////////////////////////
var CanClose: Boolean);
var DlgResult: Integer;
begin
  CanClose := True;
  if (GeoData.tbLessons.State in dsEditModes) or
    (GeoData.tbChapters.State in dsEditModes) or
    (GeoData.tbQuestions.State in dsEditModes) or
    (GeoData.tbChapters.ChangeCount > 0) or
    (GeoData.tbQuestions.ChangeCount > 0)
  then begin
    DlgResult := MessageDlg ('Πραγματοποιήσατε αλλαγές!' + chr(13) +
      'Αποθήκευση;', mtWarning, mbYesNoCancel, 0);
    if DlgResult = mrYes then btnSaveClick(Self);
    if DlgResult = mrNo then btnCancelClick(Self);
    if DlgResult = mrCancel then CanClose := False;
  end;
end;
end;

```



```

procedure TEditLessonForm.FormShow(Sender: TObject);
////////////////////////////////////////////////////////////////
/// Όνομα: FormShow
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κατά το άνοιγμα της φόρμας
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ανοίγει η φόρμα, ανοίγονται όλα τα σχετικά tables στην βάση
/// και γίνονται οι αναθέσεις των διαφόρων event handlers
////////////////////////////////////////////////////////////////
begin
  GeoData.srcLessons.OnDataChange := srcLessonsDataChange;
  GeoData.srcLessons.OnStateChange := srcChaptersStateChange;
  GeoData.srcChapters.OnStateChange := srcChaptersStateChange;
  GeoData.srcQuestions.OnStateChange := srcChaptersStateChange;
  GeoData.tbChapters.BeforeInsert := tbChaptersBeforeInsert;
  GeoData.tbQuestions.BeforeInsert := tbChaptersBeforeInsert;
  GeoData.tbLessons.Open;
  GeoData.tbExternals.Open;
  GeoData.tbChaptersMain.Open;
  GeoData.tbChapters.Open;
  GeoData.tbQuestionsMain.Open;
  GeoData.tbQuestions.Open;
  GeoData.RefreshCategories;
  QuestionsGrid.Columns[1].PickList.Clear;
  QuestionsGrid.Columns[1].PickList.AddStrings(GeoData.QuestionCategories);
  srcChaptersStateChange(self);
  ChaptersQuestionsTab.ActivePage := ChaptersSheet;
  Boxes := TMapBoxCollection.Create(MapImage);
  ReadDBBoxes;
  Boxes.OnBoxClick := BoxClick;
  Boxes.OnBoxMouseOver := BoxMouseOver;
  Randomize;
end;

procedure TEditLessonForm.MapImageClick(Sender: TObject);
////////////////////////////////////////////////////////////////
/// Όνομα: MapImageClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ πάνω στον χάρτη (όχι σε κάποιο τετράγωνο)
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης κάνει κλικ πάνω στον χάρτη, σε σημείο που δεν υπάρχει
/// κάποιο τετράγωνο, αν υπήρχε πριν κάποιο επιλεγμένο τετράγωνο ακυρώνεται
/// η επιλογή
////////////////////////////////////////////////////////////////
begin
  DeselectBox;
end;

procedure TEditLessonForm.MapImageDragDrop(Sender, Source: TObject; X,
  Y: Integer);
////////////////////////////////////////////////////////////////
/// Όνομα: MapImageDragDrop
/// Τύπος: Event Handler
/// Πυροδότηση:

```

```

/// Τέλος του Dragging (σύρσιμο) ενός αντικείμενου πάνω από τον χάρτη
/// Παράμετροι:
/// Sender: TObject - Το αντικείμενο που συρόταν και αφέθηκε στον χάρτη
/// X,Y: Integer - Οι συντεταγμένες όπου αφέθηκε το αντικείμενο που συρόταν,
/// αναφορικά με την πάνω-αριστερή γωνία του χάρτη
/// Περιγραφή:
/// Αν το συρόμενο αντικείμενο ήταν ένα τετράγωνο, τότε ενημερώνεται η Θέση του.
/// Αν ήταν το CornerShape, τότε ενημερώνεται το μέγεθος του επιλεγμένου τετραγώνου.
////////////////////////////////////
begin
  if (Source is TShape) then
    begin
      if (Source as TShape) = CornerShape then
        With (Boxes.Items[SelectedBox] as TMapBoxItem).BoxShape do
          begin
            If X > Left then
              begin
                Width := X - Left;
              end;
            If Y > Top then
              begin
                Height := Y - Top;
              end;
            CornerShape.Left := Left + Width - (CornerShape.Width DIV 2);
            CornerShape.Top := Top + Height - (CornerShape.Height DIV 2);
          end else
            With (Source as TShape) do
              begin
                Left := X - (Width div 2);
                Top := Y - (Height div 2);
                CornerShape.Left := Left + Width - (CornerShape.Width DIV 2);
                CornerShape.Top := Top + Height - (CornerShape.Height DIV 2);
              end;
            end;
          end;
        end;
      end;

procedure TEditLessonForm.MapImageDragOver(Sender, Source: TObject; X,
Y: Integer; State: TDragState; var Accept: Boolean);
////////////////////////////////////
/// Όνομα: MapImageDragOver
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Dragging (σύρσιμο) κάποιου αντικείμενου πάνω από τον χάρτη
/// Παράμετροι:
/// Sender:TObject - Το συρόμενο αντικείμενο
/// X,Y:Integer - Η συντεταγμένες του δείκτη του ποντικιού σε σχέση με
/// το τετράγωνο πάνω από το οποίο σύρεται
/// State: TDragState - Ο τρόπος με τον οποίο γίνεται το dragging (αγνοείται)
/// var Accept: Boolean - True = το συρόμενο αντικείμενο μπορεί να γίνει δεκτο
/// - False = το συρόμενο αντικείμενο είναι ακατάλληλο
/// Περιγραφή:
/// Αν το συρόμενο αντικείμενο ήταν ένα τετράγωνο, τότε ενημερώνεται η Θέση του.
/// Αν ήταν το CornerShape, τότε ενημερώνεται το μέγεθος του επιλεγμένου τετραγώνου.
////////////////////////////////////
begin
  if not (Source is TShape) then
    begin
      Accept := False;
    end else
      begin
        if (Source as TShape) = CornerShape then

```

```

With (Boxes.Items[SelectedBox] as TMapBoxItem).BoxShape do
begin
  Accept := (X > Left);
  If Accept then
  begin
    Accept := (Y > Top);
  end;
  If Accept then
  begin
    Width := X - Left;
    Height := Y - Top;
  end;
  CornerShape.Left := Left + Width - (CornerShape.Width DIV 2);
  CornerShape.Top := Top + Height - (CornerShape.Height DIV 2);
end else
With (Source as TShape) do
begin
  Left := X - (Width div 2);
  Top := Y - (Height div 2);
  CornerShape.Left := Left + Width - (CornerShape.Width DIV 2);
  CornerShape.Top := Top + Height - (CornerShape.Height DIV 2);
end;
Accept := True;
BoxesChanged := True;
srcChaptersStateChange(self);
end;
end;

procedure TEditLessonForm.ChaptersGridKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
////////////////////
/// Όνομα: ChaptersGridKeyDown
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Πάτημα κάποιου πλήκτρο όταν το ChaptersGrid είναι ενεργό
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// var Key: Word - Ο κωδικός του πλήκτρου που πατήθηκε
/// Shift: TShiftState - η κατάσταση των πλήκτρων Ctrl,Alt,Shift κτλ
/// Περιγραφή:
/// Ελέγχονται τα πλήκτρα που πατιούνται όταν ο χρήστης βρίσκεται πάνω
/// στο ChaptersGrid για να αποφευχθεί η καταλάθος εισαγωγή ή διαγραφή
/// κεφαλαίου, και να γίνεται μόνο μέσω των πλήκτρων btnInsertChapter και
/// btnDeleteChapter
////////////////////
var atBottom: Boolean;
    DS:TDataSet;
begin
  DS := (Sender as TDBGrid).DataSource.DataSet;
  atBottom := (DS.RecNo = DS.RecordCount) or
    (DS.State = dsInsert);
  if key = VK_INSERT then key:=0;
  if (key = VK_DOWN) and atBottom then key:= 0;
  if (key = VK_TAB) and atBottom and
    ((Sender as TDBGrid).SelectedField.Index = DS.Fields.Count) then key:= 0;
  if (key = VK_DELETE) and (ssCtrl in Shift) then key:= 0;
end;

procedure TEditLessonForm.ChaptersQuestionsTabChange(Sender: TObject);
////////////////////

```

```

/// Όνομα: ChaptersQuestionsTabChange
/// Τύπος: Event Handler
/// Πυροδότηση:
///   Αλλαγή του ενεργού tab στο ChaptersQuestionsTab (Κεφάλαια, Ερωτήσεις, Χάρτης)
/// Παράμετροι:
///   Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
///   Ενεργοποιούνται τα πλήκτρα btnUp και btnDown όταν είναι επιλεγμένο
///   το tab των Κεφαλαίων ή των ερωτήσεων, και στην περίπτωση που επελέγη
///   το tab του χάρτη, διαβάζονται από την βάση τα τετράγωνα του μαθήματος
////////////////////////////////////
begin
  btnUp.Enabled := (ChaptersQuestionsTab.ActivePage = ChaptersSheet) or
    (ChaptersQuestionsTab.ActivePage = QuestionsSheet);
  btnDown.Enabled := btnUp.Enabled;
  if ChaptersQuestionsTab.ActivePage = MapSheet then
    ReadDBBoxes;
end;

procedure TEditLessonForm.ChaptersQuestionsTabResize(Sender: TObject);
////////////////////////////////////
/// Όνομα: ChaptersQuestionsTabResize
/// Τύπος: Event Handler
/// Πυροδότηση:
///   Αλλαγή του μεγέθους του ChaptersQuestionsTab
/// Παράμετροι:
///   Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
///   Διαβάζονται από την βάση τα τετράγωνα του μαθήματος ώστε να εμφανιστούν
///   στις νέες θέσεις εφόσον το μέγεθος του χάρτη άλλαξε
////////////////////////////////////
begin
  ReadDBBoxes;
  srcChaptersStateChange(self);
end;

procedure TEditLessonForm.tbChaptersBeforeInsert(DataSet: TDataSet);
////////////////////////////////////
/// Όνομα: tbChaptersBeforeInsert
/// Τύπος: Event Handler
/// Πυροδότηση:
///   Πριν από την εισαγωγή νέας εγγραφής στον πίνακα tbChapters
/// Παράμετροι:
///   DataSet: TDataSet - Ο πίνακας στον οποίο θα γίνει η εισαγωγή (αγνοείται)
/// Περιγραφή:
///   Εμφανίζεται ειδοποίηση στον χρήστη στην περίπτωση που έκανε εισαγωγή
///   κεφαλαίου ή ερώτησης σε μάθημα που δεν έχει αποθηκευτεί ακόμα.
///   Ο χρήστης έχει την επιλογή να αποθηκεύσει το μάθημα, ή να ακυρώσει
///   όλες τις αλλαγές.
////////////////////////////////////
var DlgResult: Integer;
begin
  if GeoData.tbLessonsID.IsNull then
    begin
      DlgResult := MessageDlg ('Για να εισάγετε κεφάλαια και ερωτήσεις, ' + chr(13) +
        'πρέπει να αποθηκευτεί το νέο μάθημα!' + chr(13) + chr(13) +
        'Αποθήκευση του μαθήματος "' + GeoData.tbLessonsName.Value +
        "";', mtWarning, mbYesNo, 0);
      if DlgResult = mrYes then GeoData.tbLessons.Post;
      if DlgResult = mrNo then btnCancelClick(Self);
    end;
end;

```

```

end;

procedure TEditLessonForm.btnSaveClick(Sender: TObject);
////////////////////////////////////////////////////
// Όνομα: btnSaveClick
// Τύπος: Event Handler
// Πυροδότηση:
// Κλικ του πλήκτρου btnSave
// Παράμετροι:
// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
// Περιγραφή:
// Όταν ο χρήστης κάνει κλικ στο πλήκτρο btnSave, αποθηκεύονται όλες οι
// αλλαγές που πραγματοποίησε στο μάθημα και στα κεφάλαια, τις ερωτήσεις και
// τα τετράγωνα του μαθήματος.
////////////////////////////////////////////////////
begin
try
if (GeoData.tbLessons.State in dsEditModes) then
GeoData.tbLessons.Post;
if (GeoData.tbChapters.State in dsEditModes) then
GeoData.tbChapters.Post;
if (GeoData.tbQuestions.State in dsEditModes) then
GeoData.tbQuestions.Post;
if (GeoData.tbChapters.ChangeCount > 0 ) then
GeoData.tbChapters.ApplyUpdates(0);
if (GeoData.tbQuestions.ChangeCount > 0 ) then
GeoData.tbQuestions.ApplyUpdates(0);
GeoData.tbChapters.Refresh;
GeoData.tbQuestions.Refresh;
SaveDBBoxes;
BoxesChanged := False;
srcChaptersStateChange(Self);
GeoData.RefreshCategories;
QuestionsGrid.Columns[1].PickList.Clear;
QuestionsGrid.Columns[1].PickList.AddStrings(GeoData.QuestionCategories);
except
on E:Exception do
begin
ShowMessage('Δεν ήταν εφικτή η αποθήκευση!'+chr(13)+
'Mήνυμα λάθους:'+chr(13)+
'''+e.Message+''');
end;
end;
end;

procedure TEditLessonForm.btnUpClick(Sender: TObject);
////////////////////////////////////////////////////
// Όνομα: btnUpClick
// Τύπος: Event Handler
// Πυροδότηση:
// Κλικ του πλήκτρου btnUp
// Παράμετροι:
// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
// Περιγραφή:
// Όταν ο χρήστης πατήσει το πλήκτρο btnUp, η επιλεγμένη ερώτηση ή το
// επιλεγμένο κεφάλαιο (ανάλογα με το ποιο από τα δύο tabs είναι ενεργό)
// μετακινείται μία θέση πάνω στη σειρά
////////////////////////////////////////////////////
var CurlD: Integer;
DS:TDataSet;
begin

```

```

if ChaptersQuestionsTab.ActivePage = QuestionsSheet then
begin
  DS := GeoData.tbQuestions;
end else
begin
  DS := GeoData.tbChapters;
end;
DS.DisableControls;
CurID := DS.FieldByName('ID').AsInteger;
DS.Prior;
if not DS.Bof then
begin
  DS.Edit;
  DS.FieldByName('Ordering').AsInteger :=
    DS.FieldByName('Ordering').AsInteger + 1;
  DS.Post;
  DS.Locate('ID', CurID, []);
  DS.Edit;
  DS.FieldByName('Ordering').AsInteger :=
    DS.FieldByName('Ordering').AsInteger - 1;
  DS.Post;
end;
DS.EnableControls;
srcChaptersStateChange(self);
end;

procedure TEditLessonForm.ChaptersGridDbClick(Sender: TObject);
////////////////////////////////////////////////////////////////
// Όνομα: ChaptersGridDbClick
// Τύπος: Event Handler
// Πυροδότηση:
// Διπλό κλικ στο εσωτερικό του ChaptersGrid ή του QuestionsGrid
// Παράμετροι:
// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (ένα από
// τα δύο grids)
// Περιγραφή:
// Όταν ο χρήστης κάνει διπλό κλικ σε κάποιο από τα δύο grids, αν βρίσκεται
// σε κάποιο πεδίο που αφορά εξωτερικό αρχείο, δηλαδή ένα από τα
tbChaptersMediaFilename,
// tbChaptersPictureFilename ή tbQuestionsPictureFilename, τότε θα ανοίξει το
// εξωτερικό αρχείο. Αν βρίσκεται στο πεδίο tbChaptersText (τύπου Memo), θα ανοίξει
// η οθόνη MemoEditForm για να επεξεργαστεί το κείμενο του κεφαλαίου. Αν βρίσκεται
// στο πεδίο tbQuestionsCorrectAnswerTextLU τότε θα ανοίξει η οθόνη
AnswerPickerForm
// ώστε να επιλέξει την σωστή απάντηση της ερώτησης
////////////////////////////////////////////////////////////////
var ExtFile:TExternal;
    DS:TDataSet;
    Filename:String;
    IDFieldName:String;
    ExtID:Integer;
    AnsID:Integer;
begin
  If Assigned((Sender as TDBGrid).SelectedField) then
  begin
    if ((Sender as TDBGrid).SelectedField.name = 'tbChaptersMediaFilename') or
      ((Sender as TDBGrid).SelectedField.name = 'tbChaptersPictureFilename') or
      ((Sender as TDBGrid).SelectedField.name = 'tbQuestionsPictureFilename') then
    begin
      DS := (Sender as TDBGrid).SelectedField.DataSet;
      IDFieldName := (Sender as TDBGrid).SelectedField.KeyFields;
    end;
  end;
end;

```

```

ExtID := (DS.FieldByName(IDFieldName).AsInteger);
if ExtID > 0 then
begin
  ExtFile := TExternal.Create(ExtID);
  Filename := ExtFile.GetFile;
  if Filename <> '' then ExecuteFile(Filename);
  ExtFile.Free;
end;
end else
If (Sender as TDBGrid).SelectedField.name = 'tbChaptersText' then
begin
  MemoEditForm.ShowMemo(GeoData.tbChaptersText,'Κείμενο κεφαλαίου')
end else
If (Sender as TDBGrid).SelectedField.name = 'tbQuestionsCorrectAnswerTextLU' then
begin
  AnsID := GeoData.tbQuestionsCorrectAnswerID.AsInteger;
  if AnswerPickerForm.ShowList(AnsID) = mrOk then
  begin
    if not (GeoData.tbQuestions.State in dsEditModes) then
      GeoData.tbQuestions.Edit;
    GeoData.tbQuestionsCorrectAnswerID.AsInteger := AnsID;
  end;
end;
end;
end;

procedure TEditLessonForm.ChaptersGridEditButtonClick(Sender: TObject);
////////////////////////////////////////////////////////////////////
// Όνομα: ChaptersGridEditButtonClick
// Τύπος: Event Handler
// Πυροδότηση:
// Κλικ στο EditButton κάποιου πεδίου στο ChaptersGrid ή το QuestionsGrid
// Παράμετροι:
// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (ένα από
// τα δύο grids)
// Περιγραφή:
// Όταν ο χρήστης κάνει κλικ στο πλήκρο EditButton ενός πεδίου σε κάποιο από τα δύο
// grids,
// αν το πεδίο αφορά εξωτερικό αρχείο, δηλαδή είναι ένα από τα
// tbChaptersMediaFilename,
// tbChaptersPictureFilename ή tbQuestionsPictureFilename, τότε θα ανοίξει μία
// οθόνη επιλογής του εξωτερικού αρχείου. Αν πατήσει Cancel εκεί, τότε
// θα διαγραφεί το εξωτερικό αρχείο που προϋπήρχε.
// Αν είναι το πεδίο tbChaptersText (τύπου Memo), θα ανοίξει η οθόνη MemoEditForm
// για να επεξεργαστεί το κείμενο του κεφαλαίου. Αν είναι το πεδίο
// tbQuestionsCorrectAnswerTextLU τότε θα ανοίξει η οθόνη AnswerPickerForm
// ώστε να επιλέξει την σωστή απάντηση της ερώτησης
////////////////////////////////////////////////////////////////////
var ExtFile:TExternal;
    DS:TDataSet;
    IDFieldName:String;
    AnsID:Integer;
begin
  if Assigned((Sender as TDBGrid).SelectedField) then
  begin
    If ((Sender as TDBGrid).SelectedField.name = 'tbChaptersMediaFilename') or
      ((Sender as TDBGrid).SelectedField.name = 'tbChaptersPictureFilename') or
      ((Sender as TDBGrid).SelectedField.name = 'tbQuestionsPictureFilename') then
    begin
      DS := (Sender as TDBGrid).SelectedField.DataSet;
      IDFieldName := (Sender as TDBGrid).SelectedField.KeyFields;
    end;
  end;
end;

```

```

if OpenFileDialog.Execute then
begin
  if not (DS.State in dsEditModes) then DS.Edit;
  ExtFile := TExternal.Create(DS.FieldByName(IDFieldName).AsInteger);
  ExtFile.AddFile(OpenDialog.FileName);
  DS.FieldByName(IDFieldName).AsInteger := ExtFile.ID;
  DS.Post;
  ExtFile.Free;
end else
begin
  if not (DS.State in dsEditModes) then DS.Edit;
  DS.FieldByName(IDFieldName).AsInteger := -1;
  DS.Post;
end;
end else
If (Sender as TDBGrid).SelectedField.name = 'tbChaptersText' then
begin
  MemoEditForm.ShowMemo(GeoData.tbChaptersText, 'Κείμενο κεφαλαίου')
end else
If (Sender as TDBGrid).SelectedField.name = 'tbQuestionsCorrectAnswerTextLU' then
begin
  AnsID := GeoData.tbQuestionsCorrectAnswerID.AsInteger;
  if AnswerPickerForm.ShowList(AnsID) = mrOk then
  begin
    if not (GeoData.tbQuestions.State in dsEditModes) then
      GeoData.tbQuestions.Edit;
    GeoData.tbQuestionsCorrectAnswerID.AsInteger := AnsID;
  end;
end;
end;
end;

end.

unit AnswerPickerUnit;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, Grids, DBGrids, StdCtrls, Buttons, ComCtrls;

type
  TAnswerPickerForm = class(TForm)
    btnOK: TBitBtn;
    btnCancel: TBitBtn;
    Bevel1: TBevel;
    btnInsertAnswer: TBitBtn;
    btnDeleteAnswer: TBitBtn;
    Label1: TLabel;
    btnInsertGroup: TBitBtn;
    tcGroups: TTabControl;
    grdAnswers: TDBGrid;
    Label2: TLabel;
    btnDeleteGroup: TBitBtn;
    procedure FormResize(Sender: TObject);
    procedure grdAnswersDbClick(Sender: TObject);
    procedure grdAnswersEditButtonClick(Sender: TObject);
    procedure tcGroupsChange(Sender: TObject);
    procedure btnDeleteGroupClick(Sender: TObject);
    procedure btnInsertGroupClick(Sender: TObject);
  end;

```



```

procedure btnDeleteAnswerClick(Sender: TObject);
procedure btnInsertAnswerClick(Sender: TObject);
procedure btnCancelClick(Sender: TObject);
procedure btnOKClick(Sender: TObject);
procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure FormShow(Sender: TObject);
private
  { Private declarations }
  SelectedID: Integer;
  GroupIDs: TStringList;
public
  { Public declarations }
  function ShowList(Var aSelectedID:Integer):Integer;
end;

var
  AnswerPickerForm: TAnswerPickerForm;

implementation

uses GeoDataUnit, db, MapUnit;

{$R *.dfm}

var MaxGroupID: Integer;

function TAnswerPickerForm.ShowList(Var aSelectedID:Integer):Integer;
////////////////////////////////////////////////////////////////////
// Όνομα: ShowList
// Τύπος: Public Function
// Επιστρέφει:
// Integer - mrOk ή mrCancel, ανάλογα με την επιλογή του χρήστη
// Παράμετροι:
// var SelectedID:Integer - Αρχικά επιλεγμένο μάθημα
// Περιγραφή:
// Η ShowList καλείται από το υπόλοιπο πρόγραμμα για να εμφανιστεί στην οθόνη
// η λίστα απαντήσεων. Ο χρήστης μπορεί να προσθέσει και να αφαιρέσει
// απαντήσεις και είδη απαντήσεων.
// Αν πατηθεί το πλήκτρο OK, οι αλλαγές αποθηκεύονται και στην μεταβλητή
// παραμέτρου aSelectedID αποθηκεύεται η τελευταία απάντηση που επελέγη
////////////////////////////////////////////////////////////////////
begin
  SelectedID := aSelectedID;
  Result := Self.ShowModal;
  if Result = mrOk then aSelectedID := SelectedID;
end;

procedure TAnswerPickerForm.tcGroupsChange(Sender: TObject);
////////////////////////////////////////////////////////////////////
// Όνομα: tcGroupsChange
// Τύπος: Event Handler
// Πυροδότηση:
// Αλλαγή του τρέχοντος είδους απαντήσεων
// Παράμετροι:
// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
// Περιγραφή:
// Όταν ο χρήστης αλλάξει το τρέχον είδος απαντήσεων, αλλάζει η τιμή
// της μεταβλητής GroupID του Data Module του προγράμματος, ώστε να
// εμφανίζονται μόνο οι απαντήσεις που ανήκουν στο συγκεκριμένο είδος
////////////////////////////////////////////////////////////////////

```

```

var i:Integer;
begin
  if (GeoData.tbAnswersCache.State in dsEditModes) then
    GeoData.tbAnswersCache.Post;
  i := StrToInt(GroupIDs[tcGroups.TabIndex]);
  GeoData.GroupID := i;
  GeoData.tbAnswersCache.Filter := 'Group = ' + IntToStr(i);
end;

procedure TAnswerPickerForm.btnCancelClick(Sender: TObject);
////////////////////////////////////////////////////////////////
/// Όνομα: btnCancelClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnCancel
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnCancel, ακυρώνονται οι αλλαγές που
/// πραγματοποιήθηκαν
////////////////////////////////////////////////////////////////
begin
  if (GeoData.tbAnswersCache.State in dsEditModes) then
    GeoData.tbAnswersCache.Cancel;
  if (GeoData.tbAnswersCache.ChangeCount > 0) then
    GeoData.tbAnswersCache.CancelUpdates;
  ModalResult := mrCancel;
end;

procedure TAnswerPickerForm.btnDeleteAnswerClick(Sender: TObject);
////////////////////////////////////////////////////////////////
/// Όνομα: btnDeleteAnswerClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnDeleteAnswer
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnDeleteAnswer, διαγράφεται η τρέχουσα
/// επιλεγμένη απάντηση.
////////////////////////////////////////////////////////////////
begin
  GeoData.tbAnswersCache.Delete;
end;

procedure TAnswerPickerForm.btnDeleteGroupClick(Sender: TObject);
////////////////////////////////////////////////////////////////
/// Όνομα: btnDeleteGroupClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnDeleteGroup
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnDeleteGroup, διαγράφεται το τρέχον
/// επιλεγμένο είδος ερωτήσεων.
////////////////////////////////////////////////////////////////
var i:Integer;
begin

  if StrToInt(GroupIDs[tcGroups.TabIndex]) <= 10 then

```

```

begin
  ShowMessage('Τα είδη ερωτήσεων Τοποθεσίες, Ναι/Όχι και Σωστό/Λάθος' + #13 +
    'δεν μπορούν να διαγραφούν!');
end else
begin
  While GeoData.tbAnswersCache.RecordCount > 0 do
  begin
    GeoData.tbAnswersCache.Delete;
  end;
  i := tcGroups.TabIndex;
  tcGroups.Tabs.Delete(i);
  GroupIDs.Delete(i);
  if i < tcGroups.Tabs.Count then
    tcGroups.TabIndex := i else
    tcGroups.TabIndex := tcGroups.Tabs.Count - 1;
  tcGroupsChange(self);
end;
end;

procedure TAnswerPickerForm.btnInsertAnswerClick(Sender: TObject);
////////////////////////////////////////////////////////////////
/// Όνομα: btnInsertAnswerClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnInsertAnswer
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnInsertAnswer, εισάγεται μία κενή γραμμή
/// στο grid των απαντήσεων, για να πραγματοποιηθεί η εισαγωγή μιας νέας απάντησης.
////////////////////////////////////////////////////////////////
begin
  GeoData.tbAnswersCache.Insert;
end;

procedure TAnswerPickerForm.btnInsertGroupClick(Sender: TObject);
////////////////////////////////////////////////////////////////
/// Όνομα: btnInsertGroupClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnInsertGroup
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnInsertGroup, εμφανίζεται ένα πλαίσιο
/// εισόδου για να δώσει ο χρήστης την ονόμασία και πραγματοποιείται εισαγωγή
/// ενός νέου είδους απαντήσεων.
////////////////////////////////////////////////////////////////
var i:Integer;
    s:String;
begin
  i := MaxGroupID;
  s := Inputbox('Νέο Είδος Απαντήσεων','Ονομασία:');
  if s <> '' then
  begin
    tcGroups.Tabs.Add(s);
    tcGroups.TabIndex := tcGroups.Tabs.IndexOf(s);
    GroupIDs.Add(IntToStr(i));
    Inc(MaxGroupID);
    tcGroupsChange(self);
  end;
end;

```

```

end;

procedure TAnswerPickerForm.btnOKClick(Sender: TObject);
////////////////////////////////////////////////////
// Όνομα: btnOKClick
// Τύπος: Event Handler
// Πυροδότηση:
// Κλικ του πλήκτρου btnOK
// Παράμετροι:
// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
// Περιγραφή:
// Όταν ο χρήστης πατήσει το πλήκτρο OK, επικυρώνονται οι αλλαγές που
// πραγματοποιήθηκαν στο αρχείο των απαντήσεων και στα είδη απαντήσεων.
////////////////////////////////////////////////////
var i:Integer;
    AnsText:String;
    AnsGroup:Integer;
begin
    GeoData.tbAnswersCache.DisableControls;
    AnsText := GeoData.tbAnswersCache.Text.AsString;
    AnsGroup := GeoData.tbAnswersCache.Group.AsInteger;
    if (GeoData.tbAnswersCache.State in dsEditModes) then
        GeoData.tbAnswersCache.Post;
    if (GeoData.tbAnswersCache.ChangeCount > 0) then
        GeoData.tbAnswersCache.ApplyUpdates(0);
    While GeoData.tbAnswerGroups.RecordCount > 0 do
    begin
        GeoData.tbAnswerGroups.Delete;
    end;
    for i:= 0 to tcGroups.Tabs.Count - 1 do
    begin
        GeoData.tbAnswerGroups.Insert;
        GeoData.tbAnswerGroups.ID.AsString := GroupIDs[i];
        GeoData.tbAnswerGroups.Name.AsString := tcGroups.Tabs[i];
        GeoData.tbAnswerGroups.Post;
    end;
    GeoData.tbAnswersCache.Refresh;
    GeoData.tbAnswersCache.Locate('Group;Text',
        VarArrayOf([AnsGroup,AnsText]),[]);
    GeoData.tbAnswersCache.EnableControls;
    ModalResult := mrOk;
end;

procedure TAnswerPickerForm.FormClose(Sender: TObject;
var Action: TCloseAction);
////////////////////////////////////////////////////
// Όνομα: FormClose
// Τύπος: Event Handler
// Πυροδότηση:
// Κατά το κλείσιμο της φόρμας
// Παράμετροι:
// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
// var Action:TCloseAction - δεν χρησιμοποιείται
// Περιγραφή:
// Όταν κλείνει η φόρμα, πραγματοποιείται και το κλείσιμο του προσωρινού
// πίνακα tbAnswersCache καθώς και του πίνακα tbAnswerGroups
////////////////////////////////////////////////////
begin
    SelectedID := GeoData.tbAnswersCache.ID.AsInteger;
    GeoData.tbAnswersCache.Close;
    GeoData.tbAnswerGroups.Close;

```

```

end;

procedure TAnswerPickerForm.FormCloseQuery(Sender: TObject;
  var CanClose: Boolean);
  ////////////////////////////////////////////////////
  // Όνομα: FormCloseQuery
  // Τύπος: Event Handler
  // Πυροδότηση:
  // Έλεγχος πριν από το κλείσιμο της φόρμας
  // Παράμετροι:
  // Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
  // var CanClose:Boolean - Τίθεται False αν ο χρήστης επιλέξει να μην κλείσει
  // Περιγραφή:
  // Εμφανίζεται ένα παράθυρο προειδοποίησης στον χρήστη αν έχουν πραγματοποιηθεί
  // αλλαγές, ώστε να επιλέξει αν θέλει να αποθηκευτούν ή όχι
  ////////////////////////////////////////////////////
  var DlgResult: Integer;
  begin
    CanClose := True;
    if (GeoData.tbAnswersCache.State in dsEditModes) or
      (GeoData.tbAnswersCache.ChangeCount > 0)
    then begin
      DlgResult := MessageDlg ('Πραγματοποιήσατε αλλαγές!' + chr(13) +
        'Αποθήκευση;', mtWarning, mbYesNoCancel, 0);
      if DlgResult = mrYes then btnOKClick(Self);
      if DlgResult = mrNo then btnCancelClick(Self);
      if DlgResult = mrCancel then CanClose := False;
    end;
  end;

procedure TAnswerPickerForm.FormResize(Sender: TObject);
  ////////////////////////////////////////////////////
  // Όνομα: FormResize
  // Τύπος: Event Handler
  // Πυροδότηση:
  // Μετά από την αλλαγή μεγέθους της φόρμας
  // Παράμετροι:
  // Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
  // Περιγραφή:
  // Όταν αλλάζει το μέγεθος της φόρμας, αλλάζουν αναλογικά
  // τα μεγέθη των στηλών του Grid
  ////////////////////////////////////////////////////
  begin
    grdAnswers.Columns[0].Width := Width * 3 div 5 - 65;
    grdAnswers.Columns[1].Width := 60;
    grdAnswers.Columns[2].Width := Width * 2 div 5 - 65;
  end;

procedure TAnswerPickerForm.FormShow(Sender: TObject);
  ////////////////////////////////////////////////////
  // Όνομα: FormShow
  // Τύπος: Event Handler
  // Πυροδότηση:
  // Κατά το άνοιγμα της φόρμας
  // Παράμετροι:
  // Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
  // Περιγραφή:
  // Όταν ανοίγει η φόρμα, πραγματοποιείται και το άνοιγμα του προσωρινού
  // πίνακα tbLessonsCache
  ////////////////////////////////////////////////////
  var SelectedGroup: Integer;

```

```

begin
  GeoData.tbAnswersCache.Filtered := False;
  GeoData.tbAnswersCache.Open;
  if GeoData.tbAnswersCache.Locate('ID', SelectedID, []) then
    SelectedGroup := GeoData.tbAnswersCache.Group.AsInteger
  else
    SelectedGroup := -1;

  GroupIDs := TStringList.Create;
  GroupIDs.Sorted := False;
  GroupIDs.Duplicates := dupError;
  GeoData.tbAnswerGroups.Open;
  GeoData.tbAnswerGroups.First;
  tcGroups.Tabs.Clear;
  While not GeoData.tbAnswerGroups.Eof do
    begin
      tcGroups.Tabs.Add(GeoData.tbAnswerGroupsName.AsString);
      GroupIDs.Add(GeoData.tbAnswerGroupsID.AsString);
      if GeoData.tbAnswerGroupsID.AsInteger > MaxGroupID then
        MaxGroupID := GeoData.tbAnswerGroupsID.AsInteger;
      GeoData.tbAnswerGroups.Next;
    end;
  Inc(MaxGroupID);

  if SelectedGroup > -1 then
    tcGroups.TabIndex := GroupIDs.IndexOf(IntToStr(SelectedGroup))
  else
    tcGroups.TabIndex := 1;

  tcGroupsChange(self);
  GeoData.tbAnswersCache.Filtered := True;
  GeoData.tbAnswersCache.Locate('ID', SelectedID, [])
end;

procedure TAnswerPickerForm.grdAnswersDbClick(Sender: TObject);
////////////////////////////////////////////////////////////////////
// Όνομα: grdAnswersDbClick
// Τύπος: Event Handler
// Πυροδότηση:
// Διπλό κλικ επάνω σε μια γραμμή του grid των απαντήσεων (grdAnswers)
// Παράμετροι:
// Sender: TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
// Περιγραφή:
// Αν γίνει διπλό κλικ πάνω στο πεδίο του χάρτη, ο χάρτης εμφανίζεται στην
// οθόνη ώστε ο χρήστης να επιλέξει την τοποθεσία της απάντησης
//////////////////////////////////////////////////////////////////
var DS: TDataSet;
    IDFieldName: String;
    ExtID, X, Y: Integer;
begin
  if ((Sender as TDBGrid).SelectedField.name = 'tbAnswersCacheMapFilename') then
    begin
      DS := (Sender as TDBGrid).SelectedField.DataSet;
      IDFieldName := (Sender as TDBGrid).SelectedField.KeyFields;
      ExtID := (DS.FieldByName(IDFieldName).AsInteger);
      X := DS.FieldByName('MapX').AsInteger;
      Y := DS.FieldByName('MapY').AsInteger;
      if MapForm.ShowMap(DS.FieldByName('Text').AsString, ExtID, X, Y) = mrOk then
        begin
          if not (DS.State in dsEditModes) then DS.Edit;
          DS.FieldByName('IsPlace').AsBoolean := True;
        end;
    end;
end;

```

```

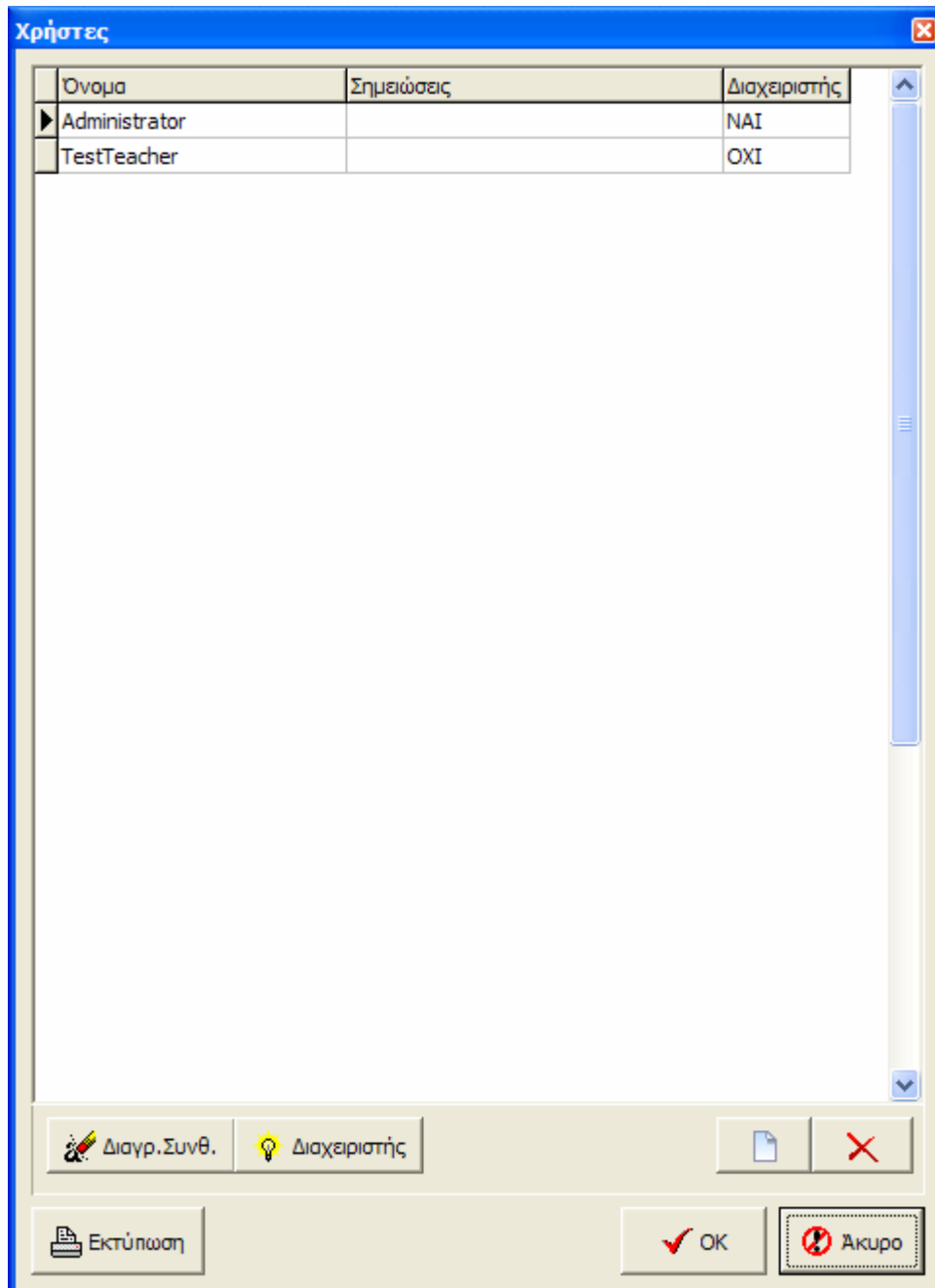
    DS.FieldByName(IDFieldName).AsInteger := ExtID;
    DS.FieldByName('MapX').AsInteger := X;
    DS.FieldByName('MapY').AsInteger := Y;
end;
end;
end;

procedure TAnswerPickerForm.grdAnswersEditButtonClick(Sender: TObject);
////////////////////////////////////////////////////////////////////
// Όνομα: grdAnswersEditButtonClick
// Τύπος: Event Handler
// Πυροδότηση:
// Κλικ στο πλήκτρο edit (...) στο πεδίο χάρτη του grid των απαντήσεων (grdAnswers)
// Παράμετροι:
// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
// Περιγραφή:
// Αν γίνει κλικ στο πλήκτρο edit στο πεδίο χάρτη, εμφανίζεται ο χάρτης στην
// οθόνη ώστε ο χρήστης να επιλέξει την τοποθεσία της απάντησης
////////////////////////////////////////////////////////////////////
var DS:TDataSet;
    IDFieldName: String;
    ExtID,X,Y:Integer;
begin
    if ((Sender as TDBGrid).SelectedField.name = 'tbAnswersCacheMapFilename') then
        begin
            DS := (Sender as TDBGrid).SelectedField.DataSet;
            IDFieldName := (Sender as TDBGrid).SelectedField.KeyFields;
            ExtID := (DS.FieldByName(IDFieldName).AsInteger);
            X := DS.FieldByName('MapX').AsInteger;
            Y := DS.FieldByName('MapY').AsInteger;
            if MapForm.ShowMap(DS.FieldByName('Text').AsString,ExtID,X,Y) = mrOk then
                begin
                    if not (DS.State in dsEditModes) then DS.Edit;
                    DS.FieldByName('IsPlace').AsBoolean := True;
                    DS.FieldByName(IDFieldName).AsInteger := ExtID;
                    DS.FieldByName('MapX').AsInteger := X;
                    DS.FieldByName('MapY').AsInteger := Y;
                end;
            end;
        end;
    end;
end.

```

#### **-Διαχείριση καθηγητών**

Ο administrator έχει τη δυνατότητα να «διαχειριστεί» τους καθηγητές που συμμετέχουν στο μάθημα. Η οθόνη αυτή εμφανίζεται, όταν ο administrator κάνει κλικ στο πλήκτρο «Διαχείριση Καθηγητών» στην βασική «Οθόνη του Administrator»:



Παρακάτω παρατίθεται ο κώδικας υλοποίησης της παραπάνω φόρμας :

```
unit UserManagementUnit;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, Grids, DBGrids, StdCtrls, Buttons, ExtCtrls, db;
```

```
type
```

```
TUserManagementForm = class(TForm)
```

```
  Bevel1: TBevel;
```

```
  btnOK: TBitBtn;
```

```
  btnCancel: TBitBtn;
```

```
  Grid: TDBGrid;
```

```
  btnInsertUser: TBitBtn;
```

```
  btnDeleteUser: TBitBtn;
```



```

btnResetPassword: TBitBtn;
btnSetAdmin: TBitBtn;
btnEvaluation: TBitBtn;
btnPrint: TBitBtn;
procedure btnPrintClick(Sender: TObject);
procedure btnEvaluationClick(Sender: TObject);
procedure btnSetAdminClick(Sender: TObject);
procedure btnDeleteUserClick(Sender: TObject);
procedure btnInsertUserClick(Sender: TObject);
procedure GridEditButtonClick(Sender: TObject);
procedure btnCancelClick(Sender: TObject);
procedure btnOKClick(Sender: TObject);
procedure GridDbClick(Sender: TObject);
procedure btnResetPasswordClick(Sender: TObject);
procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure FormShow(Sender: TObject);
private
  procedure UserBeforeDelete(DataSet: TDataSet);
  { Private declarations }
public
  function ShowList(ShowTeachers:Boolean):Integer;
  { Public declarations }
end;

var
  UserManagementForm: TUserManagementForm;

implementation

uses GeoDataUnit, GlobalsUnit, MemoEditUnit, StudentEvaluationUnit;

{$R *.dfm}

function TUserManagementForm.ShowList(ShowTeachers:Boolean):Integer;
////////////////////////////////////////////////////////////////////
/// Όνομα: ShowList
/// Τύπος: Public Function
/// Επιστρέφει:
/// Integer - mrOk ή mrCancel, ανάλογα με την επιλογή του χρήστη
/// Παράμετροι:
/// ShowTeachers:Boolean - True = Εμφάνιση λίστας καθηγητών
/// - False = Εμφάνιση λίστας μαθητών
/// Περιγραφή:
/// Η ShowList καλείται από το υπόλοιπο πρόγραμμα για να εμφανιστεί στην οθόνη
/// η λίστα των χρηστών του προγράμματος. Η λίστα φιλτράρεται ως προς τον
/// τύπο χρήστη (μαθητής/καθηγητής) ανάλογα με την τιμή της παραμέτρου ShowTeachers.
/// Μόνο χρήστης με δικαιώματα διαχειριστή μπορεί να ανοίξει την φόρμα με παράμετρο
/// οπότε και μπορεί να προσθέσει ή να αφαιρέσει καθηγητές και να δώσει
/// ή να αποσπάσει δικαιώματα διαχειριστή σε κάποιον καθηγητή.
/// Χρήστες τύπου καθηγητή αλλά όχι διαχειριστές, μπορούν να έχουν πρόσβαση
/// μόνο στη λίστα των μαθητών αλλά μόνο για ανάγνωση, και για να λάβουν
/// στατιστικά στοιχεία για την πρόοδο των μαθητών.
////////////////////////////////////////////////////////////////////
begin
  if ShowTeachers then
    begin
      GeoData.tbUserCache.Filter := 'Type = 1 OR Type = 2';
      GeoData.tbUserCache.Filtered := True;
      Grid.Columns[2].Title.Caption := 'Διαχειριστής';
      Grid.Columns[2].FieldName := 'IsAdmin';
    end
  end;
end;

```

```

Grid.Columns[2].PickList.Clear;
Grid.Columns[2].PickList.Add ('NAI');
Grid.Columns[2].PickList.Add ('OXI');
btnSetAdmin.Visible := True;
btnEvaluation.Visible := False;
btnInsertUser.Visible := True;
btnDeleteUser.Visible := True;
Grid.Options := Grid.Options + [dgEditing];
Grid.Options := Grid.Options - [dgAlwaysShowSelection, dgRowSelect];
end else
begin
GeoData.tbUserCache.Filter := 'Type = 0';
GeoData.tbUserCache.Filtered := True;
Grid.Columns[2].Title.Caption := 'Βαθμολογία';
Grid.Columns[2].FieldName := 'StudentGrade';
Grid.Columns[2].PickList.Clear;
btnSetAdmin.Visible := False;
if User.IsAdmin then
begin
Grid.Options := Grid.Options + [dgEditing];
Grid.Options := Grid.Options - [dgAlwaysShowSelection, dgRowSelect];
btnInsertUser.Visible := True;
btnDeleteUser.Visible := True;
end else begin
Grid.Options := Grid.Options - [dgEditing];
Grid.Options := Grid.Options + [dgAlwaysShowSelection, dgRowSelect];
btnInsertUser.Visible := False;
btnDeleteUser.Visible := False;
end;
btnEvaluation.Visible := True;
end;
Result := Self.ShowModal;
end;

procedure TUserManagementForm.UserBeforeDelete(DataSet: TDataSet);
////////////////////////////////////
/// Όνομα: UserBeforeDelete
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Πριν γίνει διαγραφή ενός χρήστη από τον πίνακα tbUserCache
/// Παράμετροι:
/// DataSet: TDataSet - Το αντικείμενο TDataSet (table, query, κτλ) στο οποίο
/// θα γίνει η διαγραφή
/// Περιγραφή:
/// Πριν εκτελεστεί η διαγραφή, γίνεται έλεγχος μήπως ο προς διαγραφή χρήστης
/// είναι ο τρέχον συνδεδεμένος χρήστης. Σε αυτή την περίπτωση η διαγραφή
/// ακυρώνεται και εμφανίζεται ένα μήνυμα λάθους στο χρήστη
////////////////////////////////////
begin
if GeoData.tbUserCache.ID.Value = User.ID then
begin
MessageDlg ('Δεν μπορείτε να διαγράψετε τον τρέχοντα χρήστη!', mtError, [mbOK], 0);
Abort;
end;
end;

procedure TUserManagementForm.btnCancelClick(Sender: TObject);
////////////////////////////////////
/// Όνομα: btnCancelClick
/// Τύπος: Event Handler
/// Πυροδότηση:

```

```

/// Κλικ του πλήκτρου btnCancel
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnCancel, ακυρώνονται οι αλλαγές που
/// πραγματοποιήθηκαν
////////////////////////////////////
begin
  if (GeoData.tbUserCache.State in dsEditModes) then
    GeoData.tbUserCache.Cancel;
  if (GeoData.tbUserCache.ChangeCount > 0) then
    GeoData.tbUserCache.CancelUpdates;
  ModalResult := mrCancel;
end;

procedure TUserManagementForm.btnDeleteUserClick(Sender: TObject);
////////////////////////////////////
/// Όνομα: btnDeleteUserClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnDeleteUser
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnDeleteUser, διαγράφεται ο
/// επιλεγμένος στο grid χρήστης.
////////////////////////////////////
begin
  GeoData.tbUserCache.Delete
end;

procedure TUserManagementForm.btnEvaluationClick(Sender: TObject);
////////////////////////////////////
/// Όνομα: btnEvaluationClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnEvaluation
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnEvaluation, ανοίγει η φόρμα
/// StudentEvaluationForm, όπου ο καθηγητής μπορεί να δει μία επισκόπηση
/// της προόδου κάποιου μαθητή.
////////////////////////////////////
begin
  StudentEvaluationForm.ShowForTeacher
  (GeoData.tbUserCacheID.AsInteger,GeoData.tbUserCacheName.AsString);
end;

procedure TUserManagementForm.btnInsertUserClick(Sender: TObject);
////////////////////////////////////
/// Όνομα: btnInsertUserClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnInsertUser
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnInsertUser, εισάγεται μία κενή γραμμή
/// στο grid των χρηστών, για να πραγματοποιηθεί η εισαγωγή ενός νέου χρήστη.
////////////////////////////////////

```

```

begin
  GeoData.tbUserCache.Insert;
end;

procedure TUserManagementForm.btnOKClick(Sender: TObject);
////////////////////////////////////////////////////
/// Όνομα: btnOKClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnOK
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο OK, επικυρώνονται οι αλλαγές που
/// πραγματοποιήθηκαν στο αρχείο των απαντήσεων και στα είδη απαντήσεων.
////////////////////////////////////////////////////
begin
  if (GeoData.tbUserCache.State in dsEditModes) then
    GeoData.tbUserCache.Post;
  if (GeoData.tbUserCache.ChangeCount > 0) then
    GeoData.tbUserCache.ApplyUpdates(0);
  ModalResult := mrOk;
end;

procedure TUserManagementForm.btnPrintClick(Sender: TObject);
////////////////////////////////////////////////////
/// Όνομα: btnPrintClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnPrint
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnPrint, καλείται η PrintGrid
/// για να εκτυπώσει την λίστα χρηστών που εμφανίζεται στο grid
////////////////////////////////////////////////////
begin
  GeoData.PrintGrid(Grid,'Χρήστες');
end;

procedure TUserManagementForm.btnResetPasswordClick(Sender: TObject);
////////////////////////////////////////////////////
/// Όνομα: btnResetPasswordClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnResetPassword
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnResetPassword, αφού βγει πρώτα μια
/// οθόνη επιβεβαίωσης, το password του επιλεγμένου στο grid χρήστη θα γίνει
/// κενό ".
////////////////////////////////////////////////////
begin
  if GeoData.tbUserCache.ID.Value = User.ID then
    begin
      MessageDlg ('Δεν μπορείτε να διαγράψετε το συνθηματικό του τρέχοντος
χρήστη!',mtError,[mbOK],0);
    end else
      if MessageDlg ('Είστε σίγουρος ότι θέλετε να διαγράψετε το συνθηματικό' + chr(13) +
'του χρήστη "' + GeoData.tbUserCacheName.AsString + "';',

```

```

        mtConfirmation,mbYesNoCancel,0) = mrYes then
    begin
        if not (GeoData.tbUserCache.State in dsEditModes) then
            GeoData.tbUserCache.Edit;
            GeoData.tbUserCacheEncrPassword.AsString := Encrypted("");
        end;
    end;

procedure TUserManagementForm.btnSetAdminClick(Sender: TObject);
////////////////////////////////////////////////////
// Όνομα: btnSetAdminClick
// Τύπος: Event Handler
// Πυροδότηση:
// Κλικ του πλήκτρου btnSetAdmin
// Παράμετροι:
// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
// Περιγραφή:
// Όταν ο χρήστης πατήσει το πλήκτρο btnSetAdmin, αφού βγει πρώτα μια
// οθόνη επιβεβαίωσης, ο επιλεγμένος στο grid χρήστη θα αποκτήσει δικαιώματα
// διαχειριστή
////////////////////////////////////////////////////
begin
    if MessageDlg ('Είστε σίγουρος ότι θέλετε να δώσετε δικαιώματα διαχειριστή' + chr(13) +
        'στον χρήστη "' + GeoData.tbUserCacheName.AsString + '";',
        mtConfirmation,mbYesNoCancel,0) = mrYes then
        begin
            if not (GeoData.tbUserCache.State in dsEditModes) then
                GeoData.tbUserCache.Edit;
                GeoData.tbUserCacheType.AsInteger := 2;
            end;
        end;

procedure TUserManagementForm.GridDbClick(Sender: TObject);
////////////////////////////////////////////////////
// Όνομα: GridDbClick
// Τύπος: Event Handler
// Πυροδότηση:
// Διπλό κλικ επάνω σε μια γραμμή του grid των χρηστών
// Παράμετροι:
// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
// Περιγραφή:
// Αν γίνει διπλό κλικ πάνω στο πεδίο Notes, ανοίγει η MemoEditForm για να
// γίνει η επεξεργασία του πεδίου
////////////////////////////////////////////////////
begin
    If (Sender as TDBGrid).SelectedField.name = 'tbUserCacheNotes' then
        begin
            MemoEditForm.ShowMemo(GeoData.tbUserCacheNotes,'Σημειώσεις Χρήστη');
        end;
    end;

procedure TUserManagementForm.GridEditButtonClick(Sender: TObject);
////////////////////////////////////////////////////
// Όνομα: grdAnswersDbClick
// Τύπος: Event Handler
// Πυροδότηση:
// Κλικ στο edit button σε κάποιο πεδίο του grid των χρηστών
// Παράμετροι:
// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
// Περιγραφή:
// Αν γίνει κλικ του edit button στο πεδίο Notes, ανοίγει η MemoEditForm για να

```

```

///   γίνει η επεξεργασία του πεδίου
////////////////////////////////////
begin
  If (Sender as TDBGrid).SelectedField.name = 'tbUserCacheNotes' then
    begin
      MemoEditForm.ShowMemo(GeoData.tbUserCacheNotes,'Σημειώσεις Χρήστη');
    end;
  end;

procedure TUserManagementForm.FormClose(Sender: TObject;
  var Action: TCloseAction);
////////////////////////////////////
/// Όνομα: FormClose
/// Τύπος: Event Handler
/// Πυροδότηση:
///   Κατά το κλείσιμο της φόρμας
/// Παράμετροι:
///   Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
///   var Action:TCloseAction - δεν χρησιμοποιείται
/// Περιγραφή:
///   Όταν κλείνει η φόρμα, πραγματοποιείται και το κλείσιμο του προσωρινού
///   πίνακα tbUserCache καθώς και του πίνακα tbUser
////////////////////////////////////
begin
  GeoData.tbUserCache.Filtered := False;
  GeoData.tbUserCache.Filter := "";
  GeoData.tbUserCache.BeforeDelete := NIL;
  GeoData.tbUserCache.Close;
  GeoData.tbUser.Close;
end;

procedure TUserManagementForm.FormCloseQuery(Sender: TObject;
  var CanClose: Boolean);
////////////////////////////////////
/// Όνομα: FormCloseQuery
/// Τύπος: Event Handler
/// Πυροδότηση:
///   Έλεγχος πριν από το κλείσιμο της φόρμας
/// Παράμετροι:
///   Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
///   var CanClose:Boolean - Τίθεται False αν ο χρήστης επιλέξει να μην κλείσει
/// Περιγραφή:
///   Εμφανίζεται ένα παράθυρο προειδοποίησης στον χρήστη αν έχουν πραγματοποιηθεί
///   αλλαγές, ώστε να επιλέξει αν θέλει να αποθηκευτούν ή όχι
////////////////////////////////////
  var DlgResult: Integer;
begin
  CanClose := True;
  if (GeoData.tbUserCache.State in dsEditModes) or
    (GeoData.tbUserCache.ChangeCount > 0 )
  then begin
    DlgResult := MessageDlg ('Πραγματοποιήσατε αλλαγές!' + chr(13) +
      'Αποθήκευση;', mtWarning, mbYesNoCancel, 0);
    if DlgResult = mrYes then btnOKClick(Self);
    if DlgResult = mrNo then btnCancelClick(Self);
    if DlgResult = mrCancel then CanClose := False;
  end;
end;

procedure TUserManagementForm.FormShow(Sender: TObject);
////////////////////////////////////

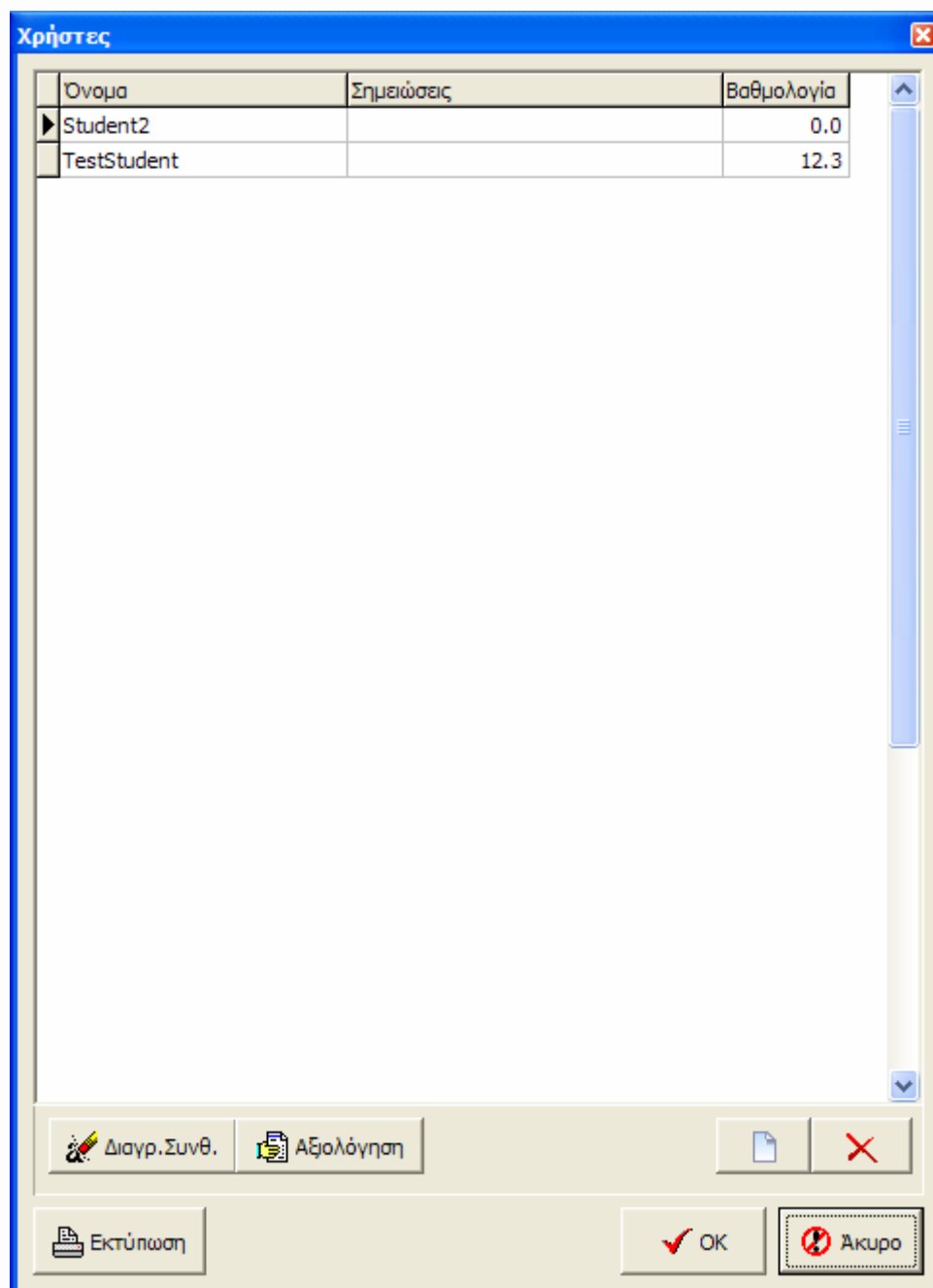
```

```
/// Όνομα: FormShow
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κατά το άνοιγμα της φόρμας
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ανοίγει η φόρμα, πραγματοποιείται και το άνοιγμα του προσωρινού
/// πίνακα tbUserCache
////////////////////////////////////
begin
  GeoData.tbUser.Open;
  GeoData.tbUserCache.Open;
  GeoData.tbUserCache.BeforeDelete := UserBeforeDelete;
end;

end.
```

### **-Διαχείριση μαθητών**

Ο administrator έχει τη δυνατότητα να «διαχειριστεί» τους μαθητές που παρακολουθούν τα μαθήματα και εξετάζονται στα τεστ. Η οθόνη αυτή εμφανίζεται, όταν ο administrator κάνει κλικ στο πλήκτρο «Διαχείριση Μαθητών» στην βασική «Οθόνη του Administrator»:



Στο κάτω μέρος της οθόνης, αριστερά, υπάρχει, επίσης το κουμπί «Αξιολόγηση». Πατώντας το ο administrator, αφού έχει επιλέξει κάποιον μαθητή, βλέπει την καρτέλα αξιολόγησης του :



**Αξιολόγηση Μαθητή: TestStudent**

<b>Γενική Βαθμολογία:</b>	<b>12.3</b>
Μαθήματα που παρακολούθησε:	<b>3</b>
Ερωτήσεις που εξετάστηκε:	<b>55</b>
Σωστές απαντήσεις:	<b>42</b>

Επιδόσεις ανά κατηγορία ερωτήσεων:

Καλύτερη Κατηγορία:  
ΣΥΝΟΡΑ ΝΟΜΩΝ (100% Επιτυχία)

Χειρότερη Κατηγορία:  
ΠΡΩΤ.ΝΟΜΩΝ (50% Επιτυχία)

? Αναλυτικά...      ✓ OK

Παρακάτω παρατίθεται ο κώδικας της παραπάνω φόρμας :

```
unit StudentEvaluationUnit;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, ExtCtrls, Mask, DBCtrls, Buttons;
```

```
type
```

```
TStudentEvaluationForm = class(TForm)  
Label1: TLabel;  
Label2: TLabel;  
Label3: TLabel;  
Label4: TLabel;  
Label6: TLabel;  
Label7: TLabel;  
Label8: TLabel;  
btnPractice: TButton;  
edWorstCat: TEdit;  
edBestCat: TEdit;  
Bevel1: TBevel;  
Bevel2: TBevel;  
btnOK: TBitBtn;  
DBText2: TDBText;  
DBText1: TDBText;  
lbCorrectAnswers: TLabel;  
lbAllAnswers: TLabel;  
Bevel3: TBevel;  
Bevel4: TBevel;
```

```

Bevel5: TBevel;
Bevel6: TBevel;
btnDetails: TBitBtn;
btnHelp: TBitBtn;
procedure btnHelpClick(Sender: TObject);
procedure btnDetailsClick(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure FormShow(Sender: TObject);
private
{ Private declarations }
mStudentID: Integer;
mWorstCategory: String;
public
{ Public declarations }
function ShowForTeacher(StudentID: Integer; StudentName:String):Integer;
function ShowForStudent(var aCategory:String):Integer;
end;

var
StudentEvaluationForm: TStudentEvaluationForm;

implementation

uses GeoDataUnit, GlobalsUnit, StudentProgressReportUnit;

{$R *.dfm}

function TStudentEvaluationForm.ShowForTeacher(StudentID: Integer;
StudentName:String):Integer;
////////////////////////////////////
/// Όνομα: ShowForTeacher
/// Τύπος: Public Function
/// Επιστρέφει:
/// Integer - mrOk ή mrNone
/// Παράμετροι:
/// StudentID: Integer - ο κωδικός του μαθητή
/// StudentName: String - το όνομα του μαθητή
/// Περιγραφή:
/// Η ShowForTeacher καλείται από το υπόλοιπο πρόγραμμα για να εμφανιστεί
/// η καρτέλα αξιολόγησης ενός μαθητή, εάν ο τρέχον χρήστης του προγράμματος
/// είναι καθηγητής
////////////////////////////////////
begin
mStudentID := StudentID;
Caption := 'Αξιολόγηση Μαθητή: ' + StudentName;
Label2.Caption := 'Μαθήματα που παρακολούθησε:';
Label3.Caption := 'Ερωτήσεις που εξετάστηκε:';
btnPractice.Visible := False;
btnDetails.Visible := True;
Result := ShowModal;
end;

function TStudentEvaluationForm.ShowForStudent(var aCategory:String):Integer;
////////////////////////////////////
/// Όνομα: ShowForStudent
/// Τύπος: Public Function
/// Επιστρέφει:
/// Integer - mrOk, mrNone (αν κλείσει με το btnOk, ή το Close Button πάνω δεξιά)
/// mrYes (αν πατηθεί το btnPractice)
/// Παράμετροι:
/// var aCategory: String - σε αυτή την παράμετρο επιστρέφεται η κατηγορία

```

```

///          ερωτήσεων που έχει το χαμηλότερο ποσοστό επιτυχίας
///          ο μαθητής
/// Περιγραφή:
/// Η ShowForStudent καλείται από το υπόλοιπο πρόγραμμα για να εμφανιστεί
/// η καρτέλα αξιολόγησης του τρέχοντος χρήστη, αν είναι μαθητής.
////////////////////////////////////
begin
  mStudentID := User.ID;
  Caption := 'Καρτέλα Αξιολόγησης';
  Label2.Caption := 'Μαθήματα που παρακολούθησες:';
  Label3.Caption := 'Ερωτήσεις που εξετάστηκες:';
  btnPractice.Visible := True;
  btnDetails.Visible := False;
  Result := ShowModal;
  aCategory := mWorstCategory;
end;

procedure TStudentEvaluationForm.btnDetailsClick(Sender: TObject);
////////////////////////////////////
/// Όνομα: btnDetailsClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnDetails
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnDetails, εμφανίζεται η φόρμα
/// StudentProgressReportForm με λεπτομέρειες για την πρόοδο του μαθητή
////////////////////////////////////
begin
  StudentProgressReportForm.ShowModal;
end;

procedure TStudentEvaluationForm.btnHelpClick(Sender: TObject);
////////////////////////////////////
/// Όνομα: btnCloseClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnClose
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnHelp, εμφανίζεται η βοήθεια
////////////////////////////////////
begin
  Application.HelpJump("Help");
end;

procedure TStudentEvaluationForm.FormClose(Sender: TObject;
  var Action: TCloseAction);
////////////////////////////////////
/// Όνομα: FormClose
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κατά το κλείσιμο της φόρμας
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// var Action:TCloseAction - δεν χρησιμοποιείται
/// Περιγραφή:
/// Όταν κλείνει η φόρμα, πραγματοποιείται και το κλείσιμο όλων των
/// πινάκων που χρησιμοποιούνται από αυτήν

```

```

////////////////////////////////////
begin
  GeoData.qrStudentEvaluation.Close;
  GeoData.qrAnswersByCategories.Close;
end;

procedure TStudentEvaluationForm.FormShow(Sender: TObject);
////////////////////////////////////
// Όνομα: FormShow
// Τύπος: Event Handler
// Πυροδότηση:
// Κατά το άνοιγμα της φόρμας
// Παράμετροι:
// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
// Περιγραφή:
// Όταν ανοίγει η φόρμα, ανοίγονται τα σχετικά tables στην βάση, και
// με τη χρήση του query qrAnswersByCategories υπολογίζεται σε ποια
// κατηγορία ερωτήσεων έχει την χειρότερη και σε ποια την καλύτερη επίδοση
// και τα αποτελέσματα γράφονται στα σχετικά πεδία.
// Αν η φόρμα έχει ανοιχτεί από μαθητή, τότε μπορεί να κάνει κλικ
// στο btnPractice, και να εξεταστεί σε ένα προσαρμοσμένο τεστ με ερωτήσεις
// μόνο από την κατηγορία που αντιμετωπίζει πρόβλημα
////////////////////////////////////
var cntAllAnswers: Integer;
    cntCorrectAnswers: Integer;
begin
  GeoData.qrStudentEvaluation.ParamByName('ID').AsInteger := mStudentID;
  GeoData.qrStudentEvaluation.Open;
  mWorstCategory := "";
  edWorstCat.Text := "";
  edBestCat.Text := "";
  lbAllAnswers.Caption := "";
  lbCorrectAnswers.Caption := "";
  GeoData.qrAnswersByCategories.ParamByName('StudentID').AsInteger := mStudentID;
  GeoData.qrAnswersByCategories.Open;
  if GeoData.qrAnswersByCategories.RecordCount > 0 then
    begin
      GeoData.qrAnswersByCategories.First; // Η χειρότερη κατηγορία
      mWorstCategory := GeoData.qrAnswersByCategories.Category.AsString;
      if mWorstCategory <> " then
        edWorstCat.Text := mWorstCategory +
          (' + GeoData.qrAnswersByCategoriesPercentCorrect.AsString + '%
Επιτυχία)';
      cntAllAnswers := 0;
      cntCorrectAnswers := 0;
      While not GeoData.qrAnswersByCategories.EOF do
        begin
          cntAllAnswers := cntAllAnswers +
GeoData.qrAnswersByCategoriesAllAnswers.AsInteger;
          cntCorrectAnswers := cntCorrectAnswers +
GeoData.qrAnswersByCategoriesCorrectAnswers.AsInteger;
          GeoData.qrAnswersByCategories.Next;
        end;
      lbAllAnswers.Caption := IntToStr(cntAllAnswers);
      lbCorrectAnswers.Caption := IntToStr(cntCorrectAnswers);
      // Η καλύτερη κατηγορία
      if GeoData.qrAnswersByCategories.Category.AsString <> " then
        edBestCat.Text := GeoData.qrAnswersByCategories.Category.AsString +
          (' + GeoData.qrAnswersByCategoriesPercentCorrect.AsString + '%
Επιτυχία)';
    end;
end;

```

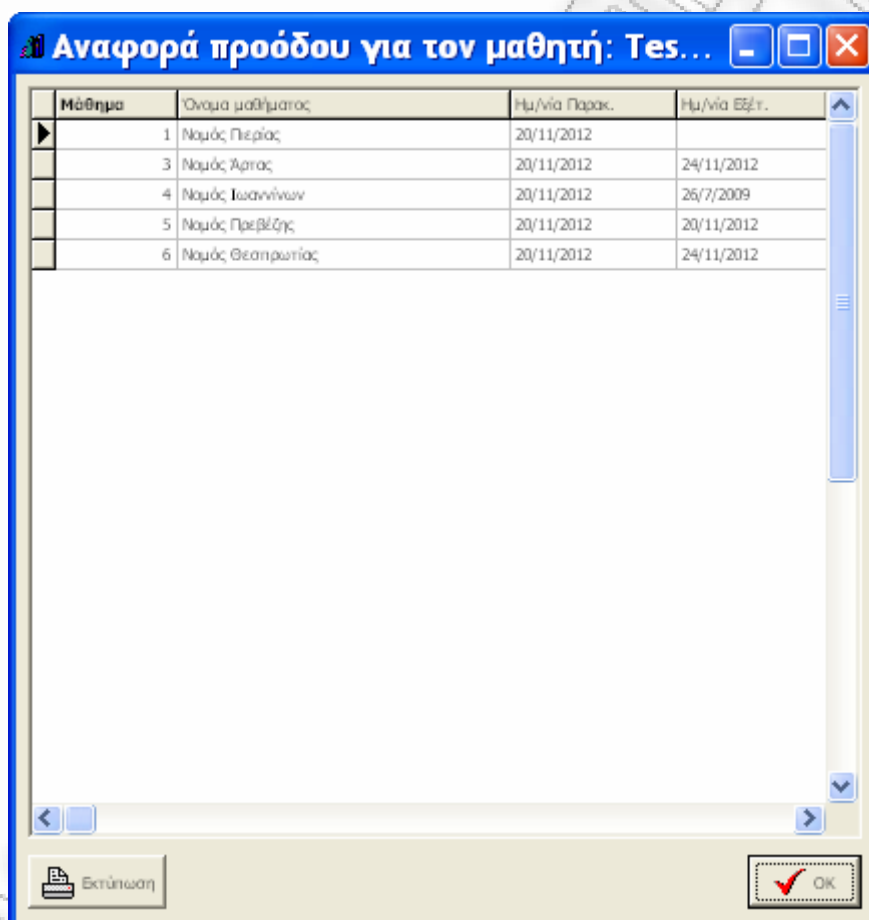
```
if edWorstCat.Text = " then btnPractice.Visible := False;
end;
```

```
end.
```

### -Φόρμα προόδου του μαθητή

Η unit `UserManagementUnit` που περιγράφηκε παραπάνω χρησιμοποιείται και σε αυτό το σημείο για την περιγραφή της φόρμας «Χρήστες».

Εάν το επιθυμεί, ο administrator μπορεί να κάνει κλικ στο κουμπί «Αναλυτικά» και θα δει αναλυτικά τους τίτλους των μαθημάτων που παρακολούθησε ο μαθητής, την ημερομηνία παρακολούθησης κάθε μαθήματος, την ημερομηνία εξέτασης κάθε τεστ και τη βαθμολογία που έλαβε σε κάθε εξέταση :



Παρακάτω παρατίθεται ο κώδικας υλοποίησης της παραπάνω φόρμας :

```
unit StudentProgressReportUnit;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, DBCtrls, StdCtrls, Mask, Grids, DBGrids, Buttons;
```

```
type
```

```
TStudentProgressReportForm = class(TForm)
```

```
  grdProgress: TDBGrid;
```

```
  btnOK: TBitBtn;
```

```

    btnPrint: TBitBtn;
    procedure btnPrintClick(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure FormShow(Sender: TObject);
    procedure grdProgressTitleClick(Column: TColumn);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    StudentProgressReportForm: TStudentProgressReportForm;

implementation

uses GeoDataUnit;

{$R *.dfm}

procedure TStudentProgressReportForm.btnPrintClick(Sender: TObject);
////////////////////////////////////////////////////////////////////
/// Όνομα: btnPrintClick
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κλικ του πλήκτρου btnPrint
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
/// Όταν ο χρήστης πατήσει το πλήκτρο btnPrint, καλείται η PrintGrid
/// για να εκτυπώσει την αναφορά προόδου του μαθητή
////////////////////////////////////////////////////////////////////
begin
    GeoData.PrintGrid(grdProgress,'Αναφορά προόδου για τον μαθητή: ' +
        GeoData.tbUserCacheName.Text);
end;

procedure TStudentProgressReportForm.FormClose(Sender: TObject;
    var Action: TCloseAction);
////////////////////////////////////////////////////////////////////
/// Όνομα: FormClose
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κατά το κλείσιμο της φόρμας
/// Παράμετροι:
/// Sender:TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// var Action:TCloseAction - δεν χρησιμοποιείται
/// Περιγραφή:
/// Όταν κλείνει η φόρμα, πραγματοποιείται και το κλείσιμο του
/// query qrStudentProgressReport
////////////////////////////////////////////////////////////////////
begin
    GeoData.qrStudentProgressReport.Close;
end;

procedure TStudentProgressReportForm.FormShow(Sender: TObject);
////////////////////////////////////////////////////////////////////
/// Όνομα: FormShow
/// Τύπος: Event Handler
/// Πυροδότηση:
/// Κατά το άνοιγμα της φόρμας

```

```

/// Παράμετροι:
///   Sender: TObject - Το αντικείμενο που προκάλεσε την πυροδότηση (αγνοείται)
/// Περιγραφή:
///   Όταν ανοίγει η φόρμα, ανοίγεται το query qrStudentProgressReport
///   γράφεται ο τίτλος της φόρμας, αρχικοποιείται το grid
////////////////////////////////////
begin
  GeoData.qrStudentProgressReport.ParambyName('StudentId').Value :=
  GeoData.tbUserCacheID.AsInteger;
  GeoData.qrStudentProgressReport.Open;
  Caption := 'Αναφορά προόδου για τον μαθητή: ' + GeoData.tbUserCacheName.Text;
  //make sure the grid is sorted by first field
  grdProgressTitleClick(grdProgress.Columns[0]);
end;

procedure TStudentProgressReportForm.grdProgressTitleClick(Column: TColumn);
////////////////////////////////////
/// Όνομα: grdProgressTitleClick
/// Τύπος: Event Handler
/// Πυροδότηση:
///   Κατά το κλικ στη γραμμή τίτλου μιας στήλης του grdProgress
/// Παράμετροι:
///   Column: TColumn - Η στήλη στην οποία έγινε κλικ
/// Περιγραφή:
///   Όταν ο χρήστης κάνει κλικ στην γραμμή τίτλου μιας από τις στήλες
///   του grid, τα περιεχόμενα ταξινομούνται με βάση το πεδίο αυτής της
///   στήλης
////////////////////////////////////
var s:String;
    i:Integer;
begin
  //κάνε τον τίτλο της στήλης έντονο
  for i := 0 to grdProgress.Columns.Count - 1 do
  begin
    grdProgress.Columns[i].Title.Font.Style := [];
  end;
  Column.Title.Font.Style := [fsBold];
  //αντικατάστησε το πεδίο ταξινόμησης στην εντολή ORDER BY
  s := GeoData.qrStudentProgressReport.SQL.Text;
  i := pos('ORDER BY',s);
  s := copy(s,1,i+8);
  s := s + Column.FieldName;
  //Ανανέωσε το query, εάν χρειάζεται
  if s <> GeoData.qrStudentProgressReport.SQL.Text then
  begin
    GeoData.qrStudentProgressReport.DisableControls;
    GeoData.qrStudentProgressReport.Close;
    GeoData.qrStudentProgressReport.SQL.Text := s;
    GeoData.qrStudentProgressReport.Open;
    GeoData.qrStudentProgressReport.EnableControls;
  end;
end;

end.

```

**ΠΕΡΙΕΧΟΜΕΝΑ**

ΠΕΡΙΛΗΨΗ.....	3
ΚΕΦΑΛΑΙΟ 1	
ΕΙΣΑΓΩΓΗ.....	4
1.1 ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΤΕΧΝΟΛΟΓΙΑ ΛΟΓΙΣΜΙΚΟΥ.....	5
1.2 ΕΙΣΑΓΩΓΗ ΚΑΙ ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ «e-School».....	9
ΚΕΦΑΛΑΙΟ 2	
ΠΑΡΟΥΣΑ ΚΑΤΑΣΤΑΣΗ.....	10
ΚΕΦΑΛΑΙΟ 3	
ΑΝΑΛΥΣΗ ΚΑΙ ΟΡΙΣΜΟΣ ΑΠΑΙΤΗΣΕΩΝ.....	14
3.1. ΣΚΟΠΟΣ ΣΥΓΓΡΑΦΗΣ ΤΟΥ ΕΓΓΡΑΦΟΥ ΠΡΟΣΔΙΟΡΙΣΜΟΥ ΑΠΑΙΤΗΣΕΩΝ.....	16
3.2. ΓΕΝΙΚΗ ΘΕΩΡΗΣΗ ΤΟΥ ΠΡΟΪΟΝΤΟΣ.....	17
3.3. ΑΝΑΛΥΣΗ ΛΕΙΤΟΥΡΓΙΩΝ-ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΠΡΟΣΕΓΓΙΣΗ.....	17
3.3.1 ΔΙΑΓΡΑΜΜΑΤΑ ΤΑΞΕΩΝ.....	17
3.3.2 ΔΙΑΓΡΑΜΜΑΤΑ ΑΝΤΙΚΕΙΜΕΝΩΝ.....	19
3.3.3 ΔΙΑΓΡΑΜΜΑΤΑ ΣΥΝΕΡΓΑΣΙΑΣ.....	27
3.3.4 ΔΙΑΓΡΑΜΜΑΤΑ ΣΕΙΡΑΣ.....	31
3.3.5 ΔΙΑΓΡΑΜΜΑΤΑ ΚΑΤΑΣΤΑΣΕΩΝ.....	35
3.3.6 ΔΙΑΓΡΑΜΜΑΤΑ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ.....	37
3.3.7 ΔΙΑΓΡΑΜΜΑΤΑ ΕΞΑΡΤΗΜΑΤΩΝ.....	38
3.3.8 ΔΙΑΓΡΑΜΜΑΤΑ ΔΙΑΝΟΜΗΣ.....	39
3.3.9 ΔΙΑΓΡΑΜΜΑΤΑ ΠΕΡΙΠΤΩΣΕΩΝ ΧΡΗΣΗΣ.....	41
3.4 ΑΝΑΛΥΣΗ ΛΕΙΤΟΥΡΓΙΩΝ-ΛΕΙΤΟΥΡΓΙΚΗ ΠΡΟΣΕΓΓΙΣΗ.....	49
ΚΕΦΑΛΑΙΟ 4	
ΣΧΕΔΙΑΣΜΟΣ ΛΟΓΙΣΜΙΚΟΥ.....	54
4.1 ΕΞΩΤΕΡΙΚΟΣ ΣΧΕΔΙΑΣΜΟΣ.....	55
4.2. ΕΣΩΤΕΡΙΚΟΣ ΣΧΕΔΙΑΣΜΟΣ.....	56
4.3. ΔΟΜΗΜΕΝΟΣ ΣΧΕΔΙΑΣΜΟΣ.....	59
4.4. ΑΝΑΛΥΤΙΚΟΣ ΣΧΕΔΙΑΣΜΟΣ.....	64
4.5. ΣΧΕΔΙΑΣΜΟΣ ΒΑΣΗΣ.....	64
ΚΕΦΑΛΑΙΟ 5	
ΣΧΕΔΙΑΣΜΟΣ ΔΙΕΠΑΦΗΣ ΜΕ ΧΡΗΣΤΕΣ-ΠΑΡΟΥΣΙΑΣΗ ΕΦΑΡΜΟΓΗΣ.....	77
5.1. ΕΙΣΑΓΩΓΗ.....	78
5.2. ΠΑΡΟΥΣΙΑΣΗ ΤΩΝ ΠΛΗΡΟΦΟΡΙΩΝ ΣΤΟΥΣ ΧΡΗΣΤΕΣ.....	80
5.3. ΠΑΡΟΧΗ ΥΠΟΣΤΗΡΙΞΗΣ ΣΤΟΥΣ ΧΡΗΣΤΕΣ.....	82
5.3.1. Μηνύματα λάθους.....	82
5.3.2. Σχεδιασμός του συστήματος βοήθειας.....	82
5.4. ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΔΙΕΠΑΦΗΣ ΜΕ ΤΟ ΧΡΗΣΤΗ.....	83
ΣΥΜΠΕΡΑΣΜΑΤΑ.....	118
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	120
ΠΑΡΑΡΤΗΜΑ 1	
ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΤΩΝ.....	122
Εισαγωγή.....	123
Α΄ Μέρος	
Εγχειρίδιο για τον μαθητή.....	124
1. Αρχική οθόνη.....	125
2. Οθόνη σύνδεσης.....	125
3. Βασική Οθόνη Μαθητή.....	127
3.1 Μάθημα Γεωγραφίας.....	127
3.2 Μάθημα Ιστορίας.....	128
4. Οθόνη Παρουσίασης Μαθήματος.....	130
4.1 Μάθημα Γεωγραφίας.....	130
4.2 Μάθημα Ιστορίας.....	131
Β΄ Μέρος	
Εγχειρίδιο για τον καθηγητή.....	142
1. Αρχική οθόνη.....	143
2. Οθόνη σύνδεσης.....	143
3. Βασική Οθόνη Καθηγητή.....	145



4. Οθόνη Επεξεργασίας Μαθήματος.....	146
Γ΄ Μέρος	
Εγχειρίδιο για τον Administrator.....	151
1. Αρχική οθόνη.....	152
2. Οθόνη σύνδεσης.....	152
3. Βασική Οθόνη Administrator.....	153
4. Οθόνη Επεξεργασίας Μαθήματος.....	155
ΠΑΡΑΡΤΗΜΑ 2	
ΚΩΔΙΚΑΣ ΕΦΑΡΜΟΓΗΣ.....	169