



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Εξ' αποστάσεως εκπαίδευση μαθήματος 'Αρχών Οικονομικής Θεωρίας»
Όνοματεπώνυμο Φοιτητή	Γεώργιος Κλεονάκος
Πατρώνυμο	Κυριακούλης
Αριθμός Μητρώου	ΜΠΠΛ/06046
Επιβλέπων	Μαρία Βίρβου, Καθηγήτρια
Τριμελής Επιτροπή	Μ.Βίρβου, Γ.Τσιχριντζής, Ε.Φούντας



Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Όνομα Επώνυμο
Βαθμίδα

Όνομα Επώνυμο
Βαθμίδα

Όνομα Επώνυμο
Βαθμίδα

Περιεχόμενα

ΠΕΡΙΛΗΨΗ	4
ABSTRACT	5
1. ΕΙΣΑΓΩΓΗ	6
2. ΕΞ΄ ΑΠΟΣΤΑΣΕΩΣ ΕΚΠΑΙΔΕΥΣΗ	10
2.1 Αρχές	10
2.2 Χρήσεις	11
2.3 Σύγκριση με την παραδοσιακή εκπαίδευση	12
3. ΕΙΣΑΓΩΓΗ ΣΤΑ ΣΥΣΤΗΜΑΤΑ E-LEARNING	14
3.1 Εκπαίδευση μέσω Συστήματος Διαχείρισης Εκπαιδευτικού Περιεχομένου	14
3.2 Παιδαγωγικές αρχές στα Συστήματα Διαχείρισης Εκπαιδευτικού Περιεχομένου	16
3.3 Ανάπτυξη & παράδοση εκπαιδευτικού υλικού	18
3.4 Ασύγχρονη εκπαίδευση στα Συστήματα Διαχείρισης Εκπαιδευτικού Περιεχομένου	18
3.5 Σύγχρονη εκπαίδευση στα Συστήματα Διαχείρισης Εκπαιδευτικού Περιεχομένου	20
4. ΤΕΧΝΟΛΟΓΙΕΣ	22
5. ΣΧΕΔΙΑΣΜΟΣ ΕΦΑΡΜΟΓΗΣ	26
5.1 Περιγραφή Απαιτήσεων – Βασικές Απαιτήσεις	26
5.2 Περιγραφή Απαιτήσεων: Ειδικές Απαιτήσεις	27
5.3 Σχεδιασμός Συστήματος	27
5.3.1 Ορισμός οντοτήτων	28
5.3.2 Σχεδιασμός Διεπαφής και Διαγράμματα δραστηριότητας	28
5.3.3 Σχεδιασμός βάσης δεδομένων	40
5.3.4 UML Διαγράμματα Δομής	43
6. ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΗΣ ΣΥΣΤΗΜΑΤΟΣ	47
6.1 Εγχειρίδιο Χρήστη του Συστήματος Διδασκαλίας	48
6.2 Εγχειρίδιο Διαχειριστή	55
7. ΕΠΙΛΟΓΟΣ	59
ΒΙΒΛΙΟΓΡΑΦΙΑ	60
ΠΑΡΑΡΤΗΜΑ – ΚΩΔΙΚΑΣ	61

ΠΕΡΙΛΗΨΗ

Σκοπός αυτής της μεταπτυχιακής διατριβής είναι να δημιουργήσει ένα μοντέλο υποστήριξης της διδασκαλίας των οικονομικών μαθημάτων στο οποίο εφαρμόζεται η τεχνική της εξ' αποστάσεως εκπαίδευσης. Θα κατασκευαστεί μια πλατφόρμα η οποία θα καθοδηγεί τον μαθητή να κατανοήσει την ύλη του μαθήματος, να κάνει ασκήσεις, να αναζητήσει βοήθεια και γενικότερα θα του δοθεί ένα εργαλείο εκμάθησης με την χρήση νέων τεχνολογιών.

ABSTRACT

The purpose of this thesis is to create a model to support the teaching of finance courses with the technique of e-learning. We create a platform that will guide the student to understand the subject matter, doing exercises, seeking help and generally will be a learning tool, using new technologies.

1. ΕΙΣΑΓΩΓΗ

Το διαδίκτυο έχει αναπτυχθεί με ραγδαίους ρυθμούς τα τελευταία χρόνια. Σχεδόν καθημερινά επεκτείνεται σε μέγεθος και καινούριες εφαρμογές αναπτύσσονται οι οποίες έχουν αυξημένες απαιτήσεις από το δίκτυο. Με τη δημιουργία του Παγκόσμιου Ιστού προσφέρθηκε ελεύθερη πρόσβαση σε όλο και περισσότερους χρήστες και το διαδίκτυο είναι πια το μεγαλύτερο δίκτυο στον κόσμο. Έχουν εμφανιστεί νέες εφαρμογές όπως η τηλεδιάσκεψη, η τηλεφωνία μέσω δικτύου και το ηλεκτρονικό εμπόριο. Ιδιαίτερα το τελευταίο έχει επιφέρει σημαντικές αλλαγές στη δομή και τον τρόπο λειτουργίας των επιχειρήσεων. Η πλειοψηφία των εφαρμογών που χρησιμοποιούν το δίκτυο είναι εφαρμογές ιστού (Web applications).

Στην τεχνολογία λογισμικού, μια εφαρμογή Ιστού είναι μια εφαρμογή που προσεγγίζεται μέσω ενός φυλλομετρητή ιστοσελίδων (web browser) πάνω από το δίκτυο. Ένας φυλλομετρητής ιστοσελίδων είναι ένα λογισμικό που επιτρέπει στον χρήστη του να προβάλλει, και να αλληλεπιδρά με, κείμενα, εικόνες, βίντεο, μουσική, παιχνίδια και άλλες πληροφορίες συνήθως αναρτημένες σε μια ιστοσελίδα ενός ιστότοπου στον Παγκόσμιο Ιστό ή σε ένα τοπικό δίκτυο. Το κείμενο και οι εικόνες σε μια ιστοσελίδα μπορεί να περιέχουν συνδέσμους προς άλλες ιστοσελίδες του ίδιου ή διαφορετικού ιστότοπου. Ο φυλλομετρητής επιτρέπει στον χρήστη την γρήγορη και εύκολη πρόσβαση σε πληροφορίες που βρίσκονται σε διάφορες ιστοσελίδες και ιστότοπους εναλλάσσοντας τις ιστοσελίδες μέσω συνδέσμων (links). Η κύρια γλώσσα που χρησιμοποιείται από τις εφαρμογές ιστού και τους φυλλομετρητές είναι η γλώσσα μορφοποίησης HTML για την προβολή των ιστοσελίδων. Για την ανάπτυξη εφαρμογών που χρειάζονται περισσότερες διαδραστικά χαρακτηριστικά και δυνατότητες χρησιμοποιούνται και άλλες γλώσσες προγραμματισμού παράλληλα με την HTML όπως η JavaScript η Java και η PHP.

Οι εφαρμογές Ιστού είναι δημοφιλείς επειδή δεν υπάρχει σύγχρονο λειτουργικό σύστημα που προορίζεται για υπολογιστές γραφείου που να μην έχει προεγκατεστημένο έναν ή περισσότερους φυλλομετρητές. Οι φυλλομετρητές ουσιαστικά αποτελούν λογισμικό πελάτη του δικτυακού πρωτοκόλλου επιπέδου εφαρμογών HTTP. Για κάθε web browser διατίθενται, επίσης, και αρκετά πρόσθετα στοιχεία (add-ons), με στόχο την επαύξηση των δυνατοτήτων τους, τη βελτίωση της χρησιμότητάς τους και την προστασία του χρήστη σε θέματα ασφάλειας, και η ευκολία της χρησιμοποίησης μιας μηχανής αναζήτησης Ιστού ως πελάτη, αποκαλούμενη μερικές φορές λεπτό πελάτη. Η δυνατότητα να ενημερωθούν και να διατηρηθούν οι εφαρμογές Ιστού χωρίς τη διανομή και εγκατάσταση του λογισμικού ενδεχομένως σε χιλιάδες υπολογιστές πελατών είναι ένας βασικός λόγος για τη δημοτικότητά τους.

Η μεγάλη δυναμική που έχουν οι δικτυακές εφαρμογές και το Διαδίκτυο γενικότερα μπορούν να ωφελήσουν και την εκπαίδευση. Ένας στους τέσσερις χρήστες του διαδικτύου είναι μαθητής ή εκπαιδευτικός ο οποίος εκτός των άλλων αναζητεί ιδέες για το αναλυτικό πρόγραμμα, τα εργαλεία έρευνας, βιβλιογραφία και ευκαιρίες επαγγελματικής ανάπτυξης. Αν δεχθεί κανείς την υπόθεση ότι η πραγματική υπόσχεση της τεχνολογίας στην εκπαίδευση βασίζεται στην δυναμική της να διευκολύνει θεμελιώδεις και ποιοτικές αλλαγές στη φύση της μάθησης και της διδασκαλίας, τότε η χρήση του διαδικτύου ανοίγει μια νέα σειρά ευκαιριών μάθησης για την εκπαίδευση, όπως η κατάργηση των φυσικών ορίων της τάξης, η δυνατότητα ασύγχρονης εκπαίδευσης, η επέκταση των εμπειριών παιδιών, η διευκόλυνση της μάθησης σε παιδιά με ειδικές ανάγκες κτλ.

Το e-learning είναι ένα σύνολο εφαρμογών και διαδικασιών όπως μάθηση βασισμένη στο διαδίκτυο, στους υπολογιστές, σε εικονικές αίθουσες διδασκαλίας και ψηφιακή συνεργασία. Η on line εκμάθηση προέρχεται από τη βασισμένη σε υπολογιστή κατάρτιση, τα διαλογικά πολυμέσα (που χρονολογούνται από τους δίσκους λέιζερ μέχρι και τα πιο πρόσφατα CDs με on-line εκμάθηση) και τα ενσωματωμένα κέντρα εκμάθησης. Οι μέθοδοι περιλαμβάνουν on-line διαλέξεις, σεμινάρια, συστήματα υποστήριξης απόδοσης, προσομοιώσεις, ενισχύσεις εργασίας, παιχνίδια, και άλλα. Το αποτελεσματικό e-learning είναι συχνά ένας συνδυασμός των παραπάνω μεθόδων.

Το διαδίκτυο είναι εξαιρετικά δημοφιλές στους δημόσιους τομείς και παρέχει νέες ευκαιρίες διαχείρισης του e-learning για σκοπούς εκμάθησης και εκπαίδευσης. Βασικά, τα συστήματα e-learning πρέπει να περιλαμβάνουν πολλαπλές μορφές και ποικίλες πηγές. Επίσης χρειάζεται να υποστηρίζουν τη συνεργασία, να προσφέρουν αλληλεπίδραση και να εκτελούν τις βασισμένες στο Web δραστηριότητες σαν μέρος του εκπαιδευτικού σκελετού και να βοηθούν εξίσου μαθητές και δασκάλους. Με αυτά τα χαρακτηριστικά το e-learning είναι ικανό να προσφέρει πολλαπλές on-line και εξ' αποστάσεως ευκαιρίες εκμάθησης στους χρήστες, που δεν είναι διαθέσιμες στα εγχειρίδια ή στις παραδοσιακές διαλέξεις.

Στις πρώτες εποχές, η μάθηση εξ' αποστάσεως αποτελούνταν από εκπαίδευση δια αλληλογραφίας, τηλεοπτικά μαθήματα, συλλογές από βιντεοκασέτες και εγγραφή κασετών. Οι προβολείς διαφανειών και τα μικροφίλμ επέτρεπαν στους σπουδαστές να επαναφέρουν στη μνήμη τους την ιστορία μέσω αρνητικών από φωτογραφίες. Η καλωδιακή τηλεόραση και η δορυφορική επέτρεψαν στους σπουδαστές να συμμετέχουν στις τάξεις στο δικό τους χρόνο. Το Διαδίκτυο και η δημιουργία τοπικών δικτύων (LAN) και δικτύων ευρείας περιοχής (WAN) έδωσαν στους σπουδαστές την ευκαιρία να δοκιμάσουν τη μάθηση εξ' αποστάσεως πέρα από προ-βιντεοσκοπημένες τάξεις και φιλμ.

Η αποτελεσματικότητα μιας εμπειρίας e-learning ενισχύεται πολύ μέσω του σχεδιασμού που έχει ως επίκεντρο τον σπουδαστή. Παραδείγματος χάριν, οι σπουδαστές θυμούνται περισσότερες πληροφορίες από ένα βιβλίο κειμένων που είναι καλά οργανωμένο, με εκτενή οπτικά μέσα, σημεία αντανάκλασης / αλληλεπίδρασης, σαφείς τίτλους, κ.λπ. Οι ίδιες έννοιες υπάρχουν για τα on-line μαθήματα, στα οποία οι σπουδαστές μαθαίνουν καλύτερα μέσω της χρήσης των σαφών επικεφαλίδων, της ελαχιστοποίησης της απόστασης της προσοχής, των οπτικών μέσων, των φιλικών φόντων στην οθόνη, του κατάλληλου άσπρου διαστήματος, των ασφαλών χρωμάτων Ιστού, κ.λπ. Βασικά, η δυνατότητα χρησιμοποίησης είναι η διαδικασία ελέγχου (μέσω της παρατήρησης) της συμπεριφοράς των σπουδαστών όσο αναφορά ένα μάθημα - τι δουλεύει, τι όχι, τι μπερδεύει.

Οι επιχειρήσεις θα γίνουν πιθανόν οι μεγαλύτεροι επενδυτές και πελάτες της εκπαίδευσης εξ' αποστάσεως. Αντιμετωπίζοντας αυξανόμενα επίπεδα διεθνούς ανταγωνισμού, πολλές επιχειρήσεις έχουν σοβαρή και επείγουσα ανάγκη να αναπτύξουν γρήγορες και συνεχείς διαδικασίες μάθησης για τους εργαζομένους τους, τους συνεργάτες και τους προμηθευτές. Η εκπαίδευση εξ' αποστάσεως παρέχει ένα βιώσιμο μέσο για την αναβάθμιση των ικανοτήτων των εργαζομένων, ιδρύοντας καλύτερη επικοινωνία με τους πελάτες, και αναπτύσσοντας προϊόντα και υπηρεσίες γρήγορα.

Η παιδαγωγική που απαιτείται στα επιχειρησιακά προγράμματα της μάθησης εξ' αποστάσεως είναι διαφορετική από αυτή που απαιτείται για ακαδημαϊκούς σκοπούς. Η παραδοσιακή εκπαίδευση επικεντρώνεται στα άτομα, με δευτερεύουσα αποστολή να επιτύχει παραγωγικά αποτελέσματα. Η επιχείρηση είναι επικεντρωμένη στην παραγωγή, με δευτερεύουσα αποστολή την εκπαίδευση των ατόμων, σαν ένα μέσο τέλους του επιχειρηματικού ωφελιμισμού. Εξαιτίας όλων αυτών των περιστάσεων, η παιδαγωγική δεν περιλαμβάνει μόνο την τέχνη της διδασκαλίας, αλλά περιλαμβάνει και ένα μείγμα στρατηγικού σχεδιασμού, διαχείρισης έργου και διαχείρισης αλλαγής.

Η εκπαίδευση εξ' αποστάσεως στο χώρο της εργασίας θεωρείται γενικά λιγότερο αποτελεσματική όταν χρησιμοποιείται ως μοναδική μέθοδος εκπαίδευσης. Ένας επιτυχημένος συνδυασμός προγράμματος σπουδών εμφανίζεται να συμπεριλαμβάνει ένα μείγμα μεθόδων εκπαίδευσης με ένα υψηλό επίπεδο προσαρμοσμένης υποστήριξης. Τα προσαρμοσμένα υλικά μαθημάτων για σπουδαστές και η σχέση μεταξύ συμβούλων δασκάλων και σπουδαστών κάνουν το πρόγραμμα επιτυχημένο.

Αναφέρεται ότι οι πρόοδοι στην τεχνολογία και τα μεταβαλλόμενα περιβάλλοντα αγοράς έχουν ωθήσει τρεις νέες ευκαιρίες εκπαίδευσης εξ' αποστάσεως για τις επιχειρήσεις. Πρώτα, τα πανεπιστήμια, οι τεχνολογικές εταιρείες, και οι εταιρείες ηλεκτρονικής μάθησης είναι στρατηγικά συνδεδεμένα να επεκτείνουν την ικανότητα εκπαίδευσης διεθνώς. Κατά δεύτερον, οι εταιρείες ηλεκτρονικής εκπαίδευσης, πολλές και μικρές σε μέγεθος, προσφέρουν καινοτόμες μορφές

ηλεκτρονικής παράδοσης μαθημάτων. Τρίτον, οι μεγάλες εταιρείες έχουν ξεκινήσει να αναπτύσσουν τα δικά τους επιχειρησιακά πανεπιστήμια, με τα δικά τους συστήματα παράδοσης μαθημάτων. Κάθε μια από αυτές τις μορφές παράδοσης και τις επιλογές εκπαίδευσης μπορούν να εισάγουν αλλαγές στην οργανωτική δομή.

Έχουν προσδιοριστεί τρία σημαντικά στοιχεία της παιδαγωγικής στην επιχειρησιακή εκπαίδευση εξ' αποστάσεως : οι συνθήκες, οι αγωγοί και το πλαίσιο. Οι συνθήκες και οι αγωγοί είναι σημαντικά, αλλά το πλαίσιο εμφανίζεται να είναι ο πιο σημαντικός παράγοντας. Το πλαίσιο προσδιορίζεται ως αλλαγή διαχείρισης που οδηγεί τις απαραίτητες περιστάσεις και συνδέσμους στη θεωρία, που προβλέπει τύπους συμπεριφοράς.

Μια επιτυχημένη στρατηγική e-learning απαιτεί πολύ προσεκτικό σχεδιασμό και εκτέλεση. Γενικά, οι συνηθισμένες αρχές της διαχείρισης έργων εφαρμόζονται στο e-learning, αλλά ιδιαίτερη προσοχή πρέπει να δοθεί στις προσδοκίες διαχείρισης, διασφαλίζοντας διοικητική υποχρέωση και ανάμειξη όλων των βασικών συμμετεχόντων.

Οι ισχυρισμοί ότι το e-learning είναι καλύτερο, ταχύτερο, φθηνότερο, προσβάσιμο από οπουδήποτε, οποτεδήποτε και με εξατομικευμένο ρυθμό για τον καθένα, είναι ένα μείγμα αλήθειας και διαφημιστικής εκστρατείας. Για να προκύψει το επιθυμητό αποτέλεσμα, είναι σημαντικό να αναπτυχθεί μία στρατηγική e-learning που ξεχωρίζει την αλήθεια από τα μεγάλα λόγια.

Πολλοί οργανισμοί συνειδητοποίησαν ότι η γνώση είναι η κύρια πηγή, η οποία μπορεί να δώσει διαρκές ανταγωνιστικό πλεονέκτημα, γεγονός που τους οδήγησε στο να προσπαθούν να διαχειριστούν τη γνώση με έναν πιο συστηματικό και αποτελεσματικό τρόπο. Η διαχείριση της γνώσης χρησιμοποιείται από οργανισμούς για να ενθαρρύνει τη δημιουργία και για να μοιραστεί η γνώση, η οποία υποστηρίζεται ότι έχει αποτελέσματα στην παραγωγικότητα, στις καινοτομίες, στην ανταγωνιστικότητα και στην καλύτερη των σχέσεων μεταξύ του ανθρωπίνου δυναμικού σε αυτούς τους οργανισμούς.

Στον τομέα των επιχειρήσεων πολλοί οργανισμοί εφαρμόζουν τη διαχείριση γνώσης για να βελτιώσουν την ικανότητα και την αποτελεσματικότητα τους και να ενθαρρύνουν τη δημιουργία και το μοίρασμα της γνώσης μεταξύ των εργαζομένων. Το ερώτημα που προκύπτει είναι αν μπορούν να χρησιμοποιηθούν τα εργαλεία, η λογική και οι τεχνικές της διαχείρισης γνώσης για να λυθούν τα προβλήματα της on-line εκπαίδευσης εξ' αποστάσεως.

Ο όρος διαχείριση γνώσης περιλαμβάνει τη διαχείριση διαδικασιών που κυβερνούν τη δημιουργία, τη διάδοση και τη χρήση της γνώσης συνδυάζοντας τεχνολογίες, οργανωτικές δομές και ανθρώπους για τη δημιουργία μεθόδων αποτελεσματικής μάθησης, λύσης προβλημάτων και αποφάσεων σε έναν οργανισμό.

Για να λυθούν τα προβλήματα στην on-line εκπαίδευση εξ' αποστάσεως, η εφαρμογή σωστής ανάμειξης εργαλείων και τεχνικών της διαχείρισης θεωρείται βασικός παράγοντας επιτυχίας.

Αρκετοί ειδικοί στην αξιολόγηση της κατάρτισης έχουν αναπτύξει μοντέλα ROI, αλλά δύο εξ αυτών υπερτερούν ως πιο πρακτικά στην εφαρμογή και πιο αποτελεσματικά στην παράδοση. Αυτά τα δύο μοντέλα, ένα από τον Kirkpatrick (1994) και μία προσαυξημένη παραλλαγή του από τον συνάδελφό του Phillips (1996), δημιουργούν ένα λογικό πλαίσιο για την παρατήρηση του ROI τόσο από τη σκοπιά της ανθρώπινης απόδοσης όσο και από της επιχειρηματικής απόδοσης. Σημαντική και για τα δύο είναι η ιδέα της «αλυσίδας αποτελεσμάτων», η οποία συνδέει τα πλεονεκτήματα των επιπέδων που προέρχονται από την εκπαίδευση. Κάθε επίπεδο μέτρησης είναι αντίκτυπος των προηγούμενων αλλά και των επόμενων επιπέδων. Αν δε λάβουμε υπόψη αυτές τις συνδέσεις, είναι δύσκολο να καταλήξουμε στο συμπέρασμα ότι η κατάρτιση είναι υπεύθυνη για οποιαδήποτε βελτίωση της απόδοσης.

Πολλές εταιρίες αναγνωρίζουν την αξία των προγραμμάτων κατάρτισης των υπαλλήλων τους και θεωρούν αυτές τις προσπάθειες σαν μία επένδυση για το μέλλον της εταιρίας. Όλοι γνωρίζουν ότι ένα καλά εκπαιδευμένο εργατικό δυναμικό είναι πολύ σημαντικό για την επιτυχία μιας επιχείρησης. Αλλά και μία καλά καταρτισμένη βάση πελατών μπορεί να είναι εξίσου

σημαντική. Και οι δύο αυτές μεταβλητές μπορούν να προσφέρουν σκληρά και μαλακά οφέλη για μία εταιρία.

Στη σημερινή πτητική οικονομία, οι κατασκευαστές παγκοσμίως είναι υπό συνεχή πίεση να αυξήσουν την παραγωγικότητα, μειώνοντας παράλληλα τα κόστη. Δεν είναι μυστικό ότι οι καλά εκπαιδευμένοι υπάλληλοι, που μπορούν να χειριστούν αποτελεσματικά και σωστά πολύπλοκες αυτοματοποιημένες μηχανές, είναι περισσότερο παραγωγικοί. Επιπλέον, με λιγότερα λάθη και μικρότερους χρόνους διακοπής, οι κατασκευαστές χάνουν λιγότερα χρήματα και εξυπηρετούν τους πελάτες τους καλύτερα.

Το e-learning έχει τα εξής πλεονεκτήματα : 1) χαμηλώνει το κόστος ,2) το περιεχόμενο του είναι περισσότερο έγκαιρο και ελεγχόμενο, 3) είναι μια προσέγγιση εκμάθησης διαθέσιμη ανά πάσα στιγμή στις ανάγκες του χρήστη, 4) δημιουργεί παγκόσμιες κοινότητες και τέλος παρέχει αυξανόμενα αξιόπιστες υπηρεσίες εκμάθησης.

Τα πλεονεκτήματά του σε σύγκριση με την παραδοσιακή αίθουσα διδασκαλίας είναι :

- Εκμάθηση στον επιθυμητό ρυθμό του μαθητευόμενου.
- Περισσότερη αλληλεπίδραση με τον αρχάριο.
- Κατανόηση που ενισχύεται από τη γραφική απεικόνιση.
- Μεγαλύτερη σχετικότητα.

Αδιαφιλονίκητη είναι η δραματική μείωση κόστους και στις βιομηχανικές και στις κυβερνητικές επιχειρήσεις από την υιοθέτηση του e-learning.

Αν και απαιτείται συνήθως μια υψηλή αρχική επένδυση για να εφαρμοστεί το e-learning σε ολόκληρη την επιχείρηση, αυτή η επένδυση αντισταθμίζεται γρήγορα από την τεράστια αποταμίευση που αναπτύσσεται στην παράδοση του υλικού .

Εκτός από την αποδοτικότερη παράδοση, η μείωση του κόστους αποδίδεται κατά ένα μεγάλο μέρος στον κερδισμένο χρόνο των υπαλλήλων. Μια άλλη βασική πηγή αποταμίευσης είναι η μη αναγκαιότητα μετάβασης στο χώρο διδασκαλίας.

Λόγω των πλεονεκτημάτων του e-learning, όλο και περισσότεροι εκπαιδευτικοί φορείς και επιχειρήσεις υιοθετούν και αυξάνουν την επένδυσή τους σε αυτό. Η αυξανόμενη επένδυση έχει φέρει την ανάγκη να καταδειχθεί η οικονομική αποδοτικότητα των επενδύσεων.

Σε μια εποχή αυστηρού ανταγωνισμού και ταραχώδους αλλαγής, η επιβίωση είναι ένα βασικό πρόβλημα που αντιμετωπίζουν οι οργανισμοί. Οι οργανισμοί έχουν προσπαθήσει να βρουν τρόπους βελτίωσης της απόδοσης των υπαλλήλων τους, και η κατάρτιση έχει θεωρηθεί ως μια από τις σημαντικότερες στρατηγικές για να λυθεί το πρόβλημα.

Εκτός από την επιταχυνόμενη σημασία της κατάρτισης, η γρήγορη ανάπτυξη των τεχνολογιών και των υπολογιστών έχει αλλάξει τις μεθόδους εκπαίδευσης και κατάρτισης και έχει προκαλέσει την άνθηση των προγραμμάτων e-learning . Η χρήση της βασισμένης στην τεχνολογία κατάρτισης και της ηλεκτρονικής κατάρτισης (e-learning) είναι μια από σημαντικές τάσεις.

Οι άνθρωποι δεν είναι ποτέ ικανοποιημένοι με την κατάστασή τους, καθώς θέλουν να κάνουν τα πάντα πιο βολικά. Ότι καταφέρνουν μπορεί να τους ικανοποιήσει προσωρινά. Οι άνθρωποι νιώθουν ότι είναι πια ώρα για περισσότερους εικονικούς ανθρώπους. Μέσα στους εικονικούς γύρους όλα είναι προδημιουργημένα. Δεν υπάρχει πραγματική διαδραστική επικοινωνία σε ένα τέτοιο προκατασκευασμένο εικονικό γύρο. Έτσι, για χρόνια οι άνθρωποι έχουν κάνει μεγάλη προσπάθεια να δημιουργήσουν ένα είδωλο στην οθόνη που να προσομοιώνει ένα πραγματικό άτομο. Ένα από τα κύρια επιτεύγματα στην γραφική υπολογιστών ήταν η δημιουργία ενός εντελώς πειστικού ανθρώπινου όντος στην οθόνη, κάτι που να μοιάζει και να αντιδρά τόσο ζωντανά, που να μην μπορεί να ξεχωριστεί από ένα πραγματικό άτομο.

2. ΕΞ' ΑΠΟΣΤΑΣΕΩΣ ΕΚΠΑΙΔΕΥΣΗ

Η ραγδαία εξάπλωση του Internet έχει προκαλέσει σημαντικές αλλαγές σε πολλούς κλάδους της οικονομίας και της κοινωνίας παγκόσμια. Από αυτόν τον τυφώνα των αλλαγών δεν μπορούσε να μείνει απέξω η εκπαίδευση. Με την ραγδαία ανάπτυξη των Τεχνολογιών της Πληροφορικής και της Τεχνολογίας, μια νέα μορφή εκπαίδευσης εισέρχεται στο χώρο, που δεν είναι άλλη από το e-learning (Electronic ή Distance Learning), που έφερε την επανάσταση στο εκπαιδευτικό γίγνεσθαι.

Ένας εναλλακτικός τρόπος εκπαίδευσης (alternative training method) από αυτήν με τον καθηγητή στην τάξη, με τη δυνατότητα πρόσβασης στο εκπαιδευτικό υλικό 24 ώρες την ημέρα, 7 μέρες την εβδομάδα μέσω του Παγκόσμιου Ιστού / Internet. Το περιεχόμενο της εκπαιδευτικής ύλης (μαθήματα) είναι σε μορφή πολυμέσων (multimedia), δηλαδή, εκτός από κείμενο (text), παρουσιάζεται και σε μορφή ήχου (audio), βίντεο και άλλων μέσων αλληλεπίδρασης, ελέγχοντας και αξιολογώντας την μελέτη, την απόδοση και το βαθμό αφομοίωσης των γνώσεων των εκπαιδευομένων.

Οι εκπαιδευόμενοι έχουν την δυνατότητα επικοινωνίας με τον Καθηγητή και τους συμμαθητές τους μέσω ασύγχρονης (Asynchronous mode) ή σύγχρονης επικοινωνίας (Synchronous mode). Η επίλυση των αποριών από τον Καθηγητή, η ανταλλαγή απόψεων μεταξύ των εκπαιδευομένων, η συμμετοχή των εκπαιδευομένων σε χώρους συζητήσεων (Discussion rooms) και δωμάτια συνομιλίας, συνθέτουν την "Εικονική αίθουσα διδασκαλίας".

Η μεθοδολογία, τα προϊόντα και οι υπηρεσίες που προσφέρονται στο χώρο του e-learning, συνδυάζουν όλες τις διαθέσιμες μέχρι σήμερα τεχνολογίες, όπως υπολογιστές, Internet, ψηφιακό ήχο, εικόνα, βίντεο, κινούμενα σχέδια, προσομοιώσεις, τηλεπικοινωνιακούς φορείς και λοιπά, μέσα από ένα ολοκληρωμένο περιβάλλον σύνθεσης και διανομής του εκπαιδευτικού περιεχομένου (content).

Από την στιγμή που υιοθετήθηκε το διαδίκτυο και εξελίχθηκε ως μέσο ανταλλαγής πληροφορίας από τα εκπαιδευτικά ιδρύματα στην δεκαετία του '70 έγινε αντιληπτή από τους ακαδημαϊκούς η προοπτική του μέσου αυτού ως εργαλείο εκπαίδευσης. Τα τελευταία χρόνια οι κυβερνήσεις αρκετών τόσο των ανεπτυγμένων όσο και των υπό-ανάπτυξη χωρών παρουσιάζονται όλο και περισσότερο ενθουσιώδεις σχετικά με τις δυνατότητες της online εκπαίδευσης για τη προσφορά μιας οικονομικώς αποδοτικής, εύκολα προσβάσιμης και πάντα σύγχρονης εκπαίδευσης ανεξαρτήτως ηλικίας, κοινωνικού περιβάλλοντος, χρόνου και γεωγραφικής θέσης. Έτσι έχουν αναπτυχθεί κάποιες πλατφόρμες e-learning (τα Συστήματα Διαχείρισης Εκπαιδευτικού Περιεχομένου / Learning Management Systems), οι οποίες και θα αναλυθούν σε επόμενη ενότητα, πάνω στις οποίες στηρίζεται η εκπαίδευση.

Παράλληλα, διανύοντας την λεγόμενη εποχή της πληροφορίας, όπου η ανάγκη για 'knowledge workers' αυξάνεται και η ανάγκη για χειρονακτική εργασία μειώνεται, η δια-βίου εκπαίδευση φαίνεται να είναι το κλειδί της επιτυχίας για την μοντέρνα κοινωνία. Το e-learning θεωρείται από πολλούς ως η μοναδική βιώσιμη λύση στο πρόβλημα της προσφοράς και παράδοσης πόρων, που απαιτείται για την πραγματοποίηση της δια-βίου εκπαίδευσης.

Το περιβάλλον της ηλεκτρονικής εκπαίδευσης πρέπει να είναι προσβάσιμο και προσαρμόσιμο σε όλους τους ενδιαφερόμενους είτε αυτοί είναι φοιτητές ή εργαζόμενοι, είτε είναι άνθρωποι που απλά θέλουν να μάθουν. Για τον λόγο αυτό, το ενδιαφέρον στην ανάπτυξη του e-learning στηρίζεται, όχι τόσο στο περιεχόμενο αλλά στην τεχνολογία που χρησιμοποιείται για την παράδοση του εκπαιδευτικού υλικού, με σκοπό οι σπουδαστές να χρησιμοποιούν στο έπακρο όλες τις δυνατότητες που προσφέρονται.

2.1 Αρχές

Οι αντιλήψεις για την αποτελεσματικότητα του e-learning δεν διαφέρουν όσον αφορά τις παιδαγωγικές αρχές, οι οποίες πρέπει να υπηρετούν την εκπαίδευση και την κατάρτιση ταυτόχρονα.

Μια πλατφόρμα e-learning πρέπει να είναι :

- Προσαρμόσιμη στις ανάγκες του εκπαιδευόμενου,
- Προσβάσιμη όλο το 24ωρο μέσω του Internet ή Intranet,
- Με δυνατότητα on-line συνεργασίας μέσω των virtual classrooms τόσο μεταξύ εκπαιδευτή - εκπαιδευομένου όσο και μεταξύ των εκπαιδευομένων,
- Με δυνατότητα ελέγχου και πιστοποίησης της προσφερόμενης εκπαίδευσης, και να πληροί τους παρακάτω θεμελιώδεις κανόνες:
- Να είναι ολοκληρωμένη και αποτελεσματική,
- Να παρέχεται ταχύτητα και ολοκληρωμένα,
- Να έχει μικρό κόστος.

Οι παραπάνω κανόνες και αρχές του e-learning (e-learning specifications) αποτελούν την συνολική υποδομή για την εκπαίδευση από απόσταση και αποτελείται από τα τέσσερα (4) δομικά στοιχεία που είναι απαραίτητα για ένα ολοκληρωμένο Σύστημα Διαχείρισης Εκπαιδευτικού Περιεχομένου (Learning Management System LMS) και αυτά είναι :

- Η ευκολία χρήσης, πρέπει να είναι απλό στον χειρισμό του, χωρίς να απαιτεί ειδικές γνώσεις.
- Η εύκολη παρακολούθηση της προόδου των μαθητών από τον καθηγητή.
- Η πραγματικά σύγχρονη και ασύγχρονη μετάδοση του υλικού.
- Η απλή διαχείριση και δημιουργία της εκπαιδευτικής οντότητας.

2.2 Χρήσεις

Μέσω διαδικτύου, η ηλεκτρονική εκπαίδευση είναι εφικτή από όλους τους ενδιαφερόμενους και εστιάζοντας στο προνόμιο αυτό η χρήση της μπορεί να πάρει πολλές μορφές πέρα από αυτή της παιδαγωγικής εκπαίδευσης. Οι μορφές που αναδεικνύονται παρακάτω είναι:

1. Μη ακαδημαϊκή / Εταιρική

Τόσο οι μικρές όσο και οι μεγάλες επιχειρήσεις χρησιμοποιούν όλο και περισσότερο την ηλεκτρονική εκπαίδευση για την εκπαίδευση ή την ενημέρωση του προσωπικού τους. Στην περίπτωση αυτή χρησιμοποιούνται τόσο εξωτερικοί πόροι όσο και προγράμματα που αναπτύχθηκαν από τις ίδιες τις επιχειρήσεις και τα οποία βασίζονται στα intranets των επιχειρήσεων.

2. Ακαδημαϊκή / Εικονικά περιβάλλοντα εκπαίδευσης (VLEs)

Τα πανεπιστήμια ανοίγονται συνεχώς και περισσότερο στις δυνατότητες των εικονικών περιβαλλόντων εκπαίδευσης. Κυρίως τα νέα πανεπιστήμια χρησιμοποιούν τα περιβάλλοντα αυτά περισσότερο και πιο αποδοτικά, ενώ τα παλαιότερα και πιο παραδοσιακά πανεπιστήμια νιώθουν την ανάγκη να ακολουθήσουν, οπότε και επενδύουν σε αυτήν την τεχνολογία.

3. Ακαδημαϊκή / Εκπαιδευτικοί δικτυακοί τόποι

Μερικά ιδρύματα (ή και σε ορισμένες περιπτώσεις αποκλειστικά κάποιοι ακαδημαϊκοί) προτιμούν να αναπτύξουν δικούς τους online εκπαιδευτικούς πόρους, αντί να χρησιμοποιήσουν κάτι τόσο δομημένο και προκαθορισμένο όσο είναι τα εικονικά περιβάλλοντα εκπαίδευσης. Πρόκειται για ατομικά σχεδιασμένους δικτυακούς τόπους που είναι προσανατολισμένοι σε συγκεκριμένο ακροατήριο, για ένα συγκεκριμένο θέμα. Πρόκειται περισσότερο για ένα διαδραστικό βιβλίο, που περιέχει οπτικοαουστικό υλικό. Μερικοί τόποι είναι δυνατόν να περιέχουν και συγκεκριμένες δραστηριότητες, δοκιμασίες και λοιπά προκειμένου να βοηθήσουν την εκπαιδευτική διαδικασία, ενώ συνηθίζεται να περιλαμβάνουν και μέρη από τα εικονικά περιβάλλοντα εκπαίδευσης όπως χώρους για συζητήσεις.

Αυτοί οι τόποι αν και μπορούν να παρέχουν το περιεχόμενό τους τόσο σε ακαδημαϊκούς και φοιτητές όσο και στο γενικό κοινό, είναι δυσκολότερο να προσαρμοστούν σε ένα συγκεκριμένο μάθημα ή πρόγραμμα σπουδών.

2.3 Σύγκριση με την παραδοσιακή εκπαίδευση

Το e-learning μέχρι πρόσφατα χρησιμοποιούνταν για την ενίσχυση της εκπαίδευσης και όχι ως μέσο για την κατάκτησή της εξ ολοκλήρου. Έχει παραλληλιστεί με την εκπαίδευση εξ αποστάσεως αλλά μπορεί να χρησιμοποιηθεί και σε συνδυασμό με την εκπαίδευση «πρόσωπο με πρόσωπο».

Η εκπαίδευση στην κλασική αίθουσα παίζει σημαντικό ρόλο στα μικρά παιδιά και στους έφηβους και αυτό γιατί οι συγκεκριμένες ομάδες ανθρώπων πρέπει να κοινωνικοποιηθούν, να πειθαρχήσουν μέσα στο σύνολο και να επιβλέπονται για την πρόοδο τους από κάποιον υπεύθυνο.

Σε ένα πανεπιστήμιο από την άλλη μεριά, μπορεί ένας καθηγητής να είναι άριστος ερευνητής αλλά να μην έχει την απαιτούμενη γνώση για να προγραμματίσει και να παραδώσει μία διάλεξη. Με την εξ αποστάσεως εκπαίδευση η ανάπτυξη των διαλέξεων και η παράδοσή τους μπορεί να γίνουν από διαφορετικά άτομα με αποτέλεσμα η κατανόησή τους να είναι εγγυημένη.

Στις υψηλότερες βαθμίδες εκπαίδευσης που χρησιμοποιείται πιο πολύ το e-learning υπάρχει η τάση για τη δημιουργία ενός εικονικού περιβάλλοντος εκπαίδευσης στο οποίο οι σπουδαστές έχουν μία διασύνδεση με τα μαθήματα. Μία πληθώρα από ανοιχτά πανεπιστήμια γνωστά ως online κολλέγια, έχουν ξεκινήσει να προσφέρουν προγράμματα σπουδών και ακαδημαϊκά πτυχία μέσω internet σε ένα ευρύ φάσμα επιπέδων και προτιμήσεων. Ωστόσο, το e-learning δε χρησιμοποιείται μόνο για εκπαιδευτικούς ιστοχώρους, αλλά χρησιμοποιείται εκτενώς και στον επιχειρησιακό τομέα, όπου η εκπαίδευση γίνεται με απευθείας σύνδεση και είναι οικονομικώς αποδοτική.

Στη συνέχεια, αναλύεται η σύγκριση μεταξύ e-learning και του κλασσικού τρόπου εκπαίδευσης σε σχέση με τον φυσικό χώρο, τον χρόνο, το περιεχόμενο, το εκπαιδευτικό υλικό και τους εκπαιδευόμενους.

- **Φυσικός χώρος.** Η κλασσική εκπαίδευση εξαρτάται απόλυτα από τον φυσικό χώρο. Οι αίθουσες διδασκαλίας είναι περιοριστικές και η επέκτασή τους είναι δύσκολη και κοστίζει. Από την άλλη πλευρά το e-learning είναι εντελώς ανεξάρτητο από τον χώρο. Οι αίθουσες διδασκαλίας είναι εικονικές (virtual classrooms) και η "χωρητικότητα" τους απεριόριστη. Η γεωγραφική κατανομή αυτών που συμμετέχουν σε μια εικονική αίθουσα διδασκαλίας παύει πλέον να είναι περιορισμός.
- **Χρόνος.** Η κλασσική εκπαίδευση είναι άρρηκτα συνδεδεμένη με τον χρόνο. Τα μαθήματα γίνονται σε συγκεκριμένες χρονικές περιόδους και η μετακίνηση και προσέλευση των εκπαιδευομένων είναι απαραίτητως ταυτόχρονη. Από την άλλη πλευρά το e-learning είναι εντελώς ανεξάρτητο από τον χρόνο. Η πρόσβαση στο εκπαιδευτικό υλικό γίνεται οποιαδήποτε στιγμή όλο το 24ωρο είτε μέσω του Internet είτε μέσω του εσωτερικού δικτύου των επιχειρήσεων (Intranet). Ο χρόνος που ο εκπαιδευόμενος αφιερώνει στην μελέτη του εκπαιδευτικού υλικού είναι απόλυτα οριζόμενος από τον ίδιο.
- **Περιεχόμενο.** Στην κλασσική εκπαίδευση το περιεχόμενο, δηλαδή το εκπαιδευτικό υλικό είναι στις περισσότερες περιπτώσεις σε μορφή κειμένου. Στο e-learning, το εκπαιδευτικό υλικό παρουσιάζεται ως συνδυασμός κειμένου, ήχου, εικόνας, προσομοιώσεων και λοιπά.
- **Εκπαιδευτικό υλικό.** Στην κλασσική εκπαίδευση από την στιγμή της σύνθεσης του εκπαιδευτικού υλικού μέχρι την διανομή σε κάθε ενδιαφερόμενο μεσολαβεί μεγάλο χρονικό διάστημα καθότι τα συμβατικά κανάλια διανομής (τυπογραφεία, μεταφορά σε βιβλιοθήκες, βιβλιοπωλεία και λοιπά) απαιτούν πολύ χρόνο. Το e-learning, κάνει πραγματικότητα το «just-in-time» για την εκπαίδευση, καθόσον εκμηδενίζει τους χρόνους διανομής. Είναι προφανές ότι από την στιγμή που ολοκληρώνεται η σύνθεση του περιεχομένου, η πρόσβαση κάθε ενδιαφερόμενου μέσω του διαδικτύου είναι άμεση. Το γεγονός αυτό αποκτά τεράστια σημασία στις περιπτώσεις που το περιεχόμενο έχει ανάγκη συνεχούς επικαιροποίησης και αλλαγής.

- Εκπαιδευόμενοι. Στην κλασική εκπαίδευση ο ρυθμός αφομοίωσης της γνώσης είναι ανεξάρτητος από τον εκπαιδευόμενο και φυσικά σταθερός για όλους τους συμμετέχοντες. Ο εκπαιδευτής καθορίζει ένα μέσο όρο προσφοράς του εκπαιδευτικού υλικού ο οποίος είναι για άλλους πιο γρήγορος και για άλλους πιο αργός. Το e-learning, εισάγει την έννοια της προσωποποίησης / personalization στον ρυθμό της προσφοράς της εκπαίδευσης, καθόσον ο ίδιος ο εκπαιδευόμενος καθορίζει αυτόνομα τον χρόνο που θα διαθέσει προκειμένου να αφομοιώσει πλήρως το περιεχόμενο της εκπαίδευσης.

3. ΕΙΣΑΓΩΓΗ ΣΤΑ ΣΥΣΤΗΜΑΤΑ E-LEARNING

Η προσφορά ποιοτικών προγραμμάτων e-learning από τεχνολογικής σκοπιάς, βασίζεται στην αξιοποίηση των εφαρμογών που επιτρέπουν σε ένα εκπαιδευτικό ή άλλο οργανισμό να σχεδιάσει, να αναπτύξει και να προσφέρει ολοκληρωμένες εξ αποστάσεως τις εκπαιδευτικές υπηρεσίες.

Οι εφαρμογές αυτές συχνά αναφέρονται ως «Συστήματα Διαχείρισης Εκπαίδευσης», αλλά καθώς δεν υπάρχει κάποιος διεθνώς καθιερωμένος όρος συχνά κανείς συναντά και άλλους που είναι λίγο-πολύ συνώνυμοι, όπως «Συστήματα διαχείρισης μαθημάτων» (Course management systems), «Διαχειριζόμενα μαθησιακά περιβάλλοντα» (Managed Learning Environments), «Πλατφόρμα εκπαίδευσης» (Learning platform), «Σύστημα υποστήριξης εκπαίδευσης» (Learning Support System), και άλλα. Στη συγκεκριμένη αναφορά, θα αναφέρεται το Σύστημα Διαχείρισης Εκπαιδευτικού Περιεχομένου / Learning Management System LMS το οποίο για την ανάπτυξή του στηρίχθηκε στα χαρακτηριστικά των Συστημάτων Διαχείρισης Περιεχομένου, στα εργαλεία που χρησιμοποιούνται στο διαδίκτυο και στην ανάπτυξη της πλατφόρμας με εκπαιδευτικό υλικό.

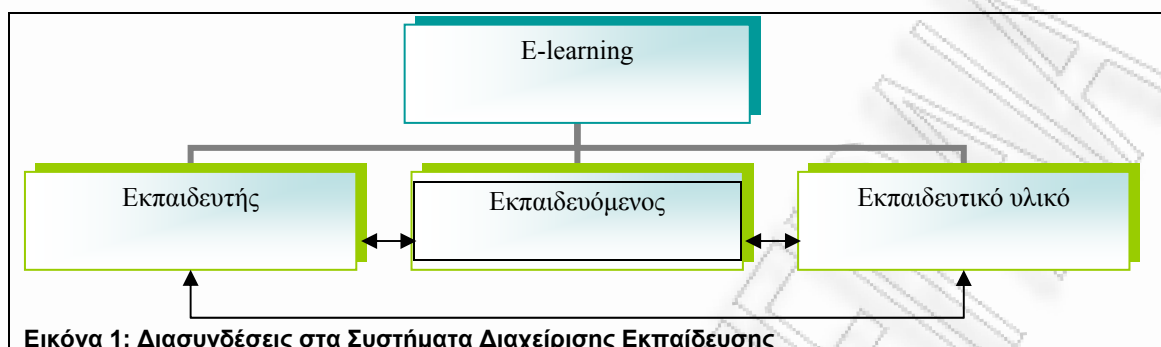
Σήμερα τέτοιου είδους συστήματα υπάρχουν πολλά τα οποία συνεχώς βελτιώνονται, ενώ διάφοροι φορείς, ομάδες και άτομα είτε για εμπορικούς λόγους, είτε αφιλοκερδώς εργάζονται για την ανάπτυξη τους. Οι δυνατότητες, τα ιδιαίτερα τεχνικά ή άλλα χαρακτηριστικά και οι απαιτήσεις, το επίπεδο ποιότητας σχεδιασμού, η λειτουργικότητα, η σταθερότητα και η ασφάλεια που παρέχουν ποικίλουν από σύστημα σε σύστημα. Το ευχάριστο είναι ότι σήμερα μπορεί ένας οργανισμός να βασιστεί ακόμα και σε συστήματα που προσφέρονται εντελώς δωρεάν καθώς αρκετά από αυτά έχουν ήδη δοκιμαστεί με επιτυχία από πολλούς εκπαιδευτικούς φορείς διεθνώς.

3.1 Εκπαίδευση μέσω Συστήματος Διαχείρισης Εκπαιδευτικού Περιεχομένου

Το εκπαιδευτικό λογισμικό διαχωρίζεται σε δύο βασικούς τομείς. Ο πρώτος αφορά τα εργαλεία λογισμικού (authoring tools), τις βάσεις δεδομένων και συζητήσεων, τα προϊόντα σύγχρονης εκπαίδευσης, που συνήθως είναι εξειδικευμένα προϊόντα στα οποία οι εταιρείες έχουν επενδύσει πολλά χρόνια εξέλιξης (για παράδειγμα οι γνωστές βάσεις δεδομένων Oracle, SQLServer & προϊόντα σύγχρονης συνεργασίας όπως Sametime, Webex, Interwise).

Ο δεύτερος τομέας αφορά το ίδιο το εκπαιδευτικό περιεχόμενο που προστατεύεται από τους νόμους περί πνευματικών δικαιωμάτων. Στην περίπτωση που ή πρόταση για άδεια χρήσης Open Source αφορά και το εκπαιδευτικό περιεχόμενο, τότε και κάθε άλλη εργασία πνευματικού δημιουργού (πχ e-Books) που είναι διαχειρίσιμη σε ηλεκτρονική μορφή θα μπορούσε να διανέμεται με παρόμοια άδεια χρήσης.

Συμβαδίζοντας με την τεχνολογία, οι «πρωταγωνιστές» στην διαδικασία του e-learning είναι ο εκπαιδευτής, ο εκπαιδευόμενος και το εκπαιδευτικό περιεχόμενο και μπορεί να έχει τρεις μορφές διασύνδεσης, εκπαιδευτής / εκπαιδευόμενος, εκπαιδευόμενος / εκπαιδευόμενος, εκπαιδευόμενος / εκπαιδευτικό υλικό.



Οι κύριες απαιτήσεις για την εφαρμογή των Συστημάτων Διαχείρισης Εκπαιδευτικού Περιεχομένου, από τη σκοπιά της εκπαίδευσης είναι η καθοδήγηση των σπουδαστών, όχι μόνο μέσω του εκπαιδευτικού υλικού, αλλά και μέσω του συστήματος. Αυτές οι απαιτήσεις αναφέρονται στη δυνατότητα της ανανέωσης και της επεξεργασίας του περιεχομένου, στη διαχείριση των μαθημάτων, στην παροχή διαμορφωτικών test και ασκήσεων και τέλος στον έλεγχο της απόδοσης.

Τα βασικά μιας πλατφόρμας Συστήματος Διαχείρισης Εκπαιδευτικού Περιεχομένου αναφέρονται στη διασύνδεση των μαθητών, στη διασύνδεση των διαχειριστών, στα εργαλεία του διαχειριστή, στη διασύνδεση του περιεχομένου που περιέχονται όλες οι σύγχρονες και ασύγχρονες υπηρεσίες, στην αποθήκευση του περιεχομένου που βασίζεται σε συστήματα διαχείρισης σχεσιακών βάσεων δεδομένων / RDBMS (Relational database management system π.χ. Oracle) και στο περιεχόμενο / content.

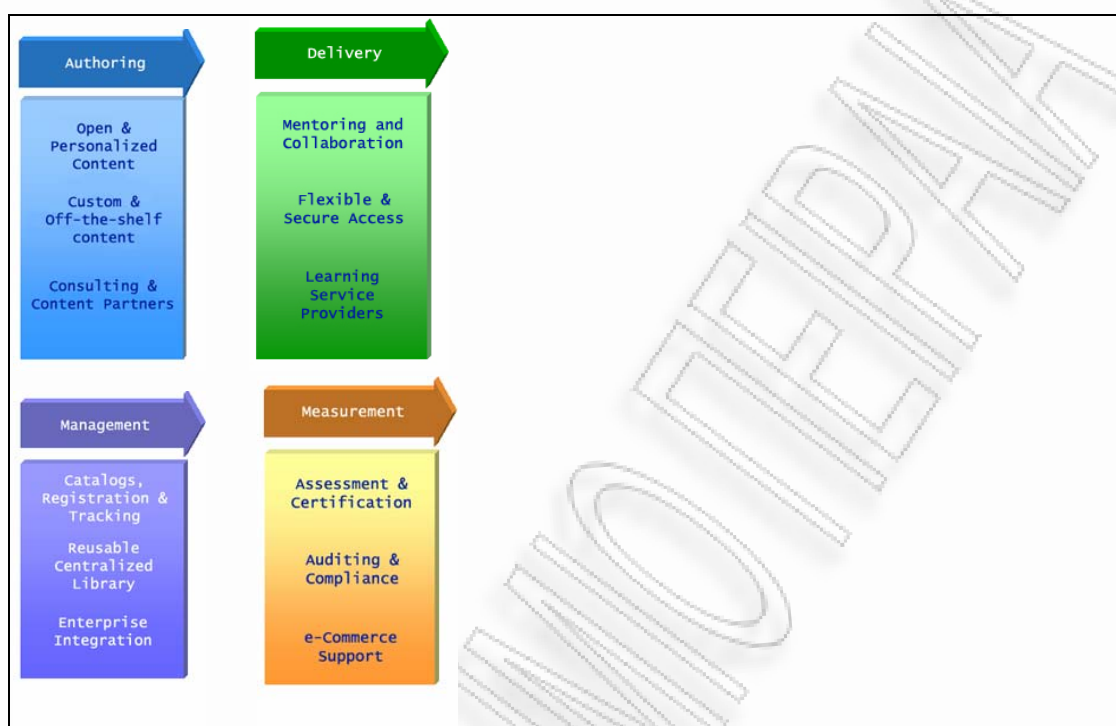
Στην παρακάτω εικόνα (Εικόνα 2: Δομικά στοιχεία Συστημάτων Διαχείρισης Εκπαίδευσης) φαίνεται η αλληλουχία των τεσσάρων βασικών δομικών στοιχείων μιας πλατφόρμας e-learning.

Συγκεκριμένα, η Δημιουργία / Authoring του περιεχομένου αφορά την ελευθερία επιλογών ως προς την ανάπτυξη του υλικού αλλά και ως προς τη μορφή ανάλογα με τις προσωπικές προτιμήσεις του δημιουργού, επίσης αφορά τη χρήση και την επεξεργασία υπάρχοντος υλικού (επαναχρησιμοποίηση) αλλά και την συνεργασία για την ανάπτυξη του βάσης κάποιων προδιαγραφών.

Η Παράδοση / Delivery αφορά την παράδοση του εκπαιδευτικού υλικού στους σπουδαστές σε συνεργασία με τους καθηγητές, την δυνατότητα πρόσβασης με ασφάλεια στην πλατφόρμα αλλά και στις υπηρεσίες που προσφέρονται για την παράδοση του υλικού όσον αφορά την τεχνολογία και τις διάφορες λειτουργικότητες.

Η Διαχείριση / Management αφορά τη διαχείριση τόσο του υλικού όσο και των χρηστών αλλά και τη δυνατότητα που δίνεται για την ολοκλήρωση της εκπαιδευτικής διαδικασίας. Συγκεκριμένα, για την πρώτη περίπτωση αναφέρονται η δυνατότητα εγγραφής, η παρακολούθηση των χρηστών και η εισαγωγή τους σε καταλόγους.

Τέλος, η Μέτρηση / Measurement αφορά την πιστοποίηση των γνώσεων από τους σπουδαστές, την αξιολόγησή τους, τον έλεγχο και τη συμμόρφωσή τους στα εκπαιδευτικά δρώμενα.



Εικόνα 2: Δομικά στοιχεία Συστημάτων Διαχείρισης Εκπαίδευσης

3.2 Παιδαγωγικές αρχές στα Συστήματα Διαχείρισης Εκπαιδευτικού Περιεχομένου

Για την υλοποίηση του σχεδιασμού, ανάπτυξης και αξιολόγησης του εκπαιδευτικού περιεχομένου στις πλατφόρμες των Συστημάτων Διαχείρισης Εκπαιδευτικού Περιεχομένου έχουν υιοθετηθεί οι παρακάτω παιδαγωγικές αρχές οι οποίες και αναλύονται.

- Ενεργητική αυτονομία
- Συνεργατική
- Στοχαστικο-κριτική
- Μεταβιβαστική
- Διερευνητική
- Αυθεντική

Ενεργητική αυτονομία

Δημιουργείται ένα “πλαίσιο στήριξης” (scaffolding) το οποίο μπορεί να οδηγήσει σταθερά τον εκπαιδευόμενο στην αυτόνομη και αυτο-ρυθμιζόμενη εκπαίδευση. Αυτό περιλαμβάνει την αυτοδύναμη δράση, την υπευθυνότητα και τη συμμετοχή στα μαθησιακά δρώμενα. Με την έννοια αυτή, το σύστημα πρέπει να παρέχει τη δυνατότητα στους εκπαιδευόμενους:

- Να συμμετέχουν στη δημιουργία, στον εμπλουτισμό και στη διαμόρφωση του εκπαιδευτικού υλικού.
- Να ασκούν σημαντικό έλεγχο στον τρόπο αλληλεπίδρασής τους με το εικονικό περιβάλλον.
- Να αναζητούν τη συναφή γνώση από άλλες περιοχές του παγκόσμιου ιστού μέσω συνδέσμων.

- Να καταθέτουν τις απόψεις, τις προθέσεις και τα επιχειρήματά τους μέσω της τεχνικής των Forum συζητήσεων.

Συνεργατική

Οι εκπαιδευόμενοι δεν αντιμετωπίζονται ατομοκεντρικά, αλλά ενθαρρύνεται η συνεργασία και η ομαδικότητα στην οικοδόμηση της νέας γνώσης αξιοποιώντας τις προσωπικές τους γνώσεις και δεξιότητες, καθώς και τη δυναμική της ομάδας. Με την έννοια αυτή, το σύστημα πρέπει να παρέχει στους εκπαιδευόμενους:

- Μηχανισμούς συνεργατικής εκπαίδευσης μέσα από την ανάπτυξη διαδραστικών ασκήσεων.
- Μηχανισμούς σύγχρονης (συνομιλία / chat), ασύγχρονης (forum συζητήσεων) και προσωπικής επικοινωνίας (e-mail) ανάμεσα στους εκπαιδευόμενους και μεταξύ εκπαιδευομένων και διδασκόντων.
- Ενθάρρυνση συμμετοχής σε εξωτερικές ομάδες ανταλλαγής πληροφοριών σε παρεμφερή θέματα.
- Δημιουργία προσωπικού μαθησιακού περιβάλλοντος και υλικού που έχουν νόημα για τον εκπαιδευόμενο.
- Έμφαση τόσο στους εσωτερικούς μηχανισμούς της συμμετοχικής επικοινωνίας όσο και στο περιεχόμενο της επικοινωνίας.
- Τη δυνατότητα να παρουσιάζουν προσχέδια των γραπτών εργασιών στους συμμετέχοντες, να ασκούν κριτική ο ένας στον άλλο στις εργασίες και να αναθεωρούν τις εργασίες τους ως αποτέλεσμα της ανατροφοδότησης που αποκτάται από αυτή τη διαδικασία.

Στοχαστικο-κριτική

Η στοχαστικο-κριτική αρχή επικεντρώνεται κυρίως στη δυνατότητα που δίνεται στον εκπαιδευόμενο να εκφράζει αυτό που έχει μάθει και να στοχάζεται πάνω στις διαδικασίες και αποφάσεις μέσα από τις οποίες έφτασε στο τελικό αποτέλεσμα.

Με την έννοια αυτή, το σύστημα πρέπει να παρέχει στους εκπαιδευόμενους:

- Αλληλεπιδραστικές ασκήσεις μέσω των οποίων οι εκπαιδευόμενοι αυτο-αξιολογούν τις δραστηριότητες που εκτέλεσαν για να φτάσουν στο αποτέλεσμα.
- Τη δυνατότητα να ανταλλάσσουν απόψεις, να κάνουν κρίσεις και συγκρίσεις για τη διαδικασία και τα αποτελέσματα των δικών τους δραστηριοτήτων τους, αλλά και των άλλων εκπαιδευομένων.

Μεταβιβαστική

Για να μπορέσει ο εκπαιδευόμενος να αφομοιώσει τη γνώση και να τη μεταφέρει σε νέες καταστάσεις θα πρέπει να συμμετέχει ενεργά στη διαδικασία οικοδόμησης της γνώσης.

Με την έννοια αυτή, το σύστημα πρέπει να παρέχει στους εκπαιδευόμενους:

- Παραδείγματα και μελέτες περίπτωσης.
- Τα εργαλεία για μια σφαιρική, ολιστική και διακλαδική προσέγγιση.
- Παραδείγματα και μελέτες περίπτωσης μέσα από τις οποίες οι εκπαιδευόμενοι θα μπορούν να ανταποκρίνονται στα βιώματα, στις παραστάσεις και τις εμπειρίες.

Διερευνητική

Οι διερευνητικές προσεγγίσεις αποσκοπούν στο να καταστήσουν ικανούς τους εκπαιδευόμενους να αντιμετωπίζουν επιτυχώς και με αυτονομία προβληματικές καταστάσεις.

Με την έννοια αυτή, το σύστημα πρέπει να παρέχει στους εκπαιδευόμενους:

- Ολοκληρωμένα παραδείγματα και αναθέσεις εργασιών πάνω σε προβληματο-κεντρικές καταστάσεις, όπως είναι η επιλογή του διδακτικού υλικού, η οργάνωση/ιεράρχηση του υλικού, η παρουσίαση και αξιολόγηση του υλικού,

- Τη δυνατότητα να θέτουν ερωτήματα, να διατυπώνουν υποθέσεις, να αναπτύσσουν τεχνικές αξιολόγησης της εκπαίδευσης, να κάνουν συνεπαγωγές βασισμένες στο γραπτό λόγο, να κατανοούν την πληροφορία μέσω διαφορετικών μέσων προφορικής συνομιλίας και άλλα.
- Ενίσχυση των ομαδικών διερευνητικών δραστηριοτήτων με βάση την εσωτερική διαφοροποίηση τους.(π.χ. ενασχόληση με διαφορετικές πτυχές του ίδιου θέματος από διαφορετικές ομάδες ή υπο-ομάδες).

Αυθεντική

Οι εκπαιδευόμενοι επιλύουν προβλήματα που σχετίζονται με πραγματικές καταστάσεις και αντιλαμβάνονται ότι οι δεξιότητες στην επίλυση αυτών των προβλημάτων είναι χρήσιμες στην επαγγελματική και κοινωνική τους ζωή.

Το σύστημα πρέπει να παρέχει στους εκπαιδευόμενους:

- Μηχανισμούς, τεχνικές και εργαλεία με τα οποία οι εκπαιδευόμενοι θα καταρτίζονται στην αξιοποίηση του ηλεκτρονικού υλικού (π.χ. εφημερίδες, περιοδικά, εικόνες, βίντεο και άλλα) που διατίθενται μέσω του Παγκόσμιου Ιστού.
- Παραδείγματα, ασκήσεις και μελέτες περίπτωσης μέσα από τα οποία οι εκπαιδευόμενοι θα μπορούν να αναπτύξουν τις βασικές τεχνικές.
- Τη δυνατότητα να αναγνωρίζουν τη συμμετοχική τους συνεισφορά στα μαθησιακά δρώμενα και αποτελέσματα.
- Χρήση διδακτικών/μαθησιακών δραστηριοτήτων σε πραγματικά και εικονικά περιβάλλοντα βασιζόμενα στις βιωματικές εμπειρίες.

3.3 Ανάπτυξη & παράδοση εκπαιδευτικού υλικού

Για την ανάπτυξη του εκπαιδευτικού υλικού χρησιμοποιούνται βασικά τρεις τεχνικές:

- Η επιλογή του υλικού-διδακτέα ύλη.
- Η οργάνωση / ιεράρχηση του υλικού.
- Η παρουσίαση του υλικού.

Από την άλλη μεριά η παράδοσή του εκπαιδευτικού υλικού γίνεται με δύο τρόπους, σύγχρονα και ασύγχρονα.

3.4 Ασύγχρονη εκπαίδευση στα Συστήματα Διαχείρισης Εκπαιδευτικού Περιεχομένου

Η Ασύγχρονη εκπαίδευση βασίζεται κυρίως στο δίκτυο και στην ασύγχρονη πρόσβαση στο υλικό του μαθήματος από τους εκπαιδευόμενους. Είναι σαφές ότι χρειάζεται να χρησιμοποιηθεί κάποιο λογισμικό για να πραγματοποιηθεί αυτό. Το λογισμικό αυτό ονομάζεται πλατφόρμα Ασύγχρονης Τηλεκπαίδευσης ή Σύστημα Διαχείρισης Μαθησιακού Υλικού (Learning Management System - LMS).

Ως πλατφόρμα Ασύγχρονης τηλεκπαίδευσης θα μπορούσε να θεωρηθεί και μία απλή ιστοσελίδα στην οποία ανεβάζει ο καθηγητής το υλικό του μαθήματος και στη συνέχεια οι μαθητές παραδίδουν τις εργασίες τους μέσω ηλεκτρονικού ταχυδρομείου. Αν και κάτι τέτοιο ίσως εξυπηρετούσε τις βασικές ανάγκες δεν θα ήταν αποτελεσματικό.

Μία πλατφόρμα για Ασύγχρονη εκπαίδευση θα πρέπει να ικανοποιεί τις παρακάτω βασικές απαιτήσεις:

- Να υποστηρίζει τον χωρισμό των χρηστών σε ομάδες έτσι ώστε η ίδια πλατφόρμα να μπορεί να χρησιμοποιηθεί για περισσότερα από ένα μαθήματα. Προφανώς θα πρέπει να υποστηρίζει κάποιου είδους πιστοποίηση των χρηστών.

- Να υποστηρίζει τη δημιουργία βημάτων συζήτησης (discussion forums) για την επικοινωνία των εκπαιδευομένων και του εκπαιδευτή ασύγχρονα.
- Να υποστηρίζει «δωμάτια συζητήσεων» (chat rooms) για συζήτηση σε πραγματικό χρόνο (σύγχρονη) και ανταλλαγή απόψεων.
- Να υλοποιεί ηλεκτρονικό ταχυδρομείο (e-mail) για την καλύτερη επικοινωνία των χρηστών.
- Εύκολο τρόπο τόσο για τον καθηγητή για να τοποθετεί το υλικό του μαθήματος όσο και για το μαθητή για την τοποθέτηση των εργασιών του.
- Να δίνει τη δυνατότητα στους μαθητές τοπικής αποθήκευσης του υλικού του μαθήματος, για επεξεργασία εκτός του δικτύου.

Αν και τα παραπάνω θεωρούνται απολύτως απαραίτητα για μία πλατφόρμα ασύγχρονης τηλεκπαίδευσης, με την εξέλιξη της τεχνολογίας, την αποκτηθείσα εμπειρία και τους ολοένα πιο απαιτητικούς χρήστες έχουν αρχίσει να προστίθενται και άλλα χαρακτηριστικά για δυνατότητες όπως:

- Να υπάρχει το υλικό του μαθήματος και σε εύκολα εκτυπώσιμη μορφή για τους χρήστες που προτιμούν το έντυπο υλικό.
- Το περιβάλλον να είναι προσβάσιμο από απλό web browser ώστε να μη χρειάζεται από τους χρήστες εγκατάσταση άλλου λογισμικού και για να είναι προσβάσιμο από παντού (π.χ. Internet cafe) και από οποιοδήποτε λειτουργικό σύστημα.
- Να έχει φιλικό περιβάλλον τόσο για το χρήστη / μαθητή όσο και για το χρήστη / καθηγητή.
- Να υποστηρίζει προσωποποίηση (customization) του περιβάλλοντος ανάλογα με το χρήστη. Επίσης να κρατάει πληροφορίες (δημιουργία profiles) για το χρήστη για να τον «βοηθάει» κατά την πλοήγηση.
- Να έχει ημερολόγιο με τις προθεσμίες και άλλα σημαντικά γεγονότα.
- Να παρακολουθεί την πρόοδο των μαθητών.
- Να υποστηρίζει την εύκολη δημιουργία διαγωνισμάτων (online tests).
- Να υποστηρίζει την παρουσίαση και άλλων πολυμεσικών υλικών όπως βίντεο, ήχου, εικόνων και άλλα.

Πρότυπα

Πολύ γρήγορα φάνηκε η ανάγκη ύπαρξης ανοικτών προτύπων για την περιγραφή του μαθησιακού υλικού. Οι βασικότεροι λόγοι που οδήγησαν στην ανάπτυξη προτύπων περιγραφής μαθησιακών αντικειμένων είναι η ανάγκη για επαναχρησιμοποίηση του μαθησιακού υλικού. Είναι πολύ σημαντικό μετά τη δημιουργία ενός μαθήματος για την ασύγχρονη τηλεκπαίδευση το υλικό αυτό να μπορεί να επαναχρησιμοποιηθεί την επόμενη φορά που θα διδαχθεί το μάθημα και να είναι απαραίτητες μόνο οι ενημερώσεις και οι βελτιώσεις. Οι εξελίξεις στον τομέα της τηλεκπαίδευσης είναι ραγδαίες και οι ανάγκες που καλείται να καλύψει μία πλατφόρμα για ασύγχρονη εκπαίδευση είναι συνεχώς αυξανόμενες με αποτέλεσμα να βγαίνουν συνεχώς καινούριες εκδόσεις και να αναπτύσσονται καινούριες πλατφόρμες. Επίσης, είναι πολύ σημαντικό μία αναβάθμιση της πλατφόρμας ή μία μετάβαση από μία πλατφόρμα σε μία άλλη, να μη συνεπάγεται και επαναδημιουργία του μαθησιακού υλικού.

Η ανάγκη για συνεργασία μεταξύ Συστημάτων Διαχείρισης Εκπαιδευτικού Περιεχομένου οφείλεται στο ότι οι εκπαιδευτές πολλές φορές θέλουν να συνεργαστούν και να ανταλλάξουν μαθησιακό υλικό. Είναι απαραίτητο λοιπόν να υπάρχει ένας ενιαίος τρόπος περιγραφής του μαθησιακού υλικού και να μπορούν διαφορετικές πλατφόρμες να συνεργαστούν για την ανταλλαγή του.

Η ανάγκη για διαθεσιμότητα πρόσβασης και εύκολης αναζήτησης. Είναι σημαντικό οι χρήστες να μπορούν να ψάξουν εύκολα στο μαθησιακό υλικό και να βρουν αυτό που τους ενδιαφέρει.

Οι παραπάνω λόγοι οδήγησαν στη δημιουργία προτύπων για την περιγραφή των μαθησιακών αντικειμένων και τα περιγραφικά μαθησιακά δεδομένα (metadata).

3.5 Σύγχρονη εκπαίδευση στα Συστήματα Διαχείρισης Εκπαιδευτικού Περιεχομένου

Όπως αναφέρθηκε και στον ορισμό της σύγχρονης εκπαίδευσης για να είναι εφικτή η πραγματοποίηση κάποιου μαθήματος θα πρέπει η εικονική αίθουσα να προσφέρει τουλάχιστον όλες τις δυνατότητες που προσφέρει και μία κανονική αίθουσα :

- Ηλεκτρονικό ασπρωπίνακα. Ο πίνακας είναι το σημαντικότερο μέσο που χρησιμοποιούν οι καθηγητές για τη διδασκαλία στην αίθουσα. Είναι απαραίτητο λοιπόν να δίνεται αυτή η δυνατότητα στον καθηγητή και σε μία εικονική αίθουσα.
- Αλληλεπιδραστική (δύο δρόμων) οπτικοακουστική επικοινωνία μεταξύ των συμμετεχόντων. Είναι πολύ σημαντικό για την επιτυχία του μαθήματος να υπάρχει πολύ καλής ποιότητα επικοινωνία μεταξύ των συμμετεχόντων έτσι ώστε να εξαλείφεται η απόσταση και να δημιουργείται η εντύπωση ότι βρίσκονται όλοι στον ίδιο χώρο. Προφανώς προτεραιότητα δίνεται στον ήχο αλλά δεν πρέπει να υποτιμηθεί η αναγκαιότητα του βίντεο αφού έχει αποδειχθεί στην πράξη ότι όταν πέφτει η ποιότητα του βίντεο χάνεται το ενδιαφέρον των συμμετεχόντων.
- Δυνατότητα για από κοινού χρήση εφαρμογής (application sharing). Είναι απαραίτητο για τον καθηγητή να μπορεί να παρουσιάσει ψηφιακό υλικό στους σπουδαστές (παρουσιάσεις με διαφάνειες, web browser, έγγραφα κειμένου και λοιπά). Όπως στην κλασική τάξη ο καθηγητής έχει τη δυνατότητα να δείξει διαφάνειες στους μαθητές, είναι απαραίτητο για τον καθηγητή να μπορεί να παρουσιάσει το υλικό του μαθήματος και στην εικονική τάξη (παρουσίαση power point).

Οι παραπάνω απαιτήσεις είναι οι ελάχιστες που πρέπει να ικανοποιεί μία εικονική αίθουσα. Από τη στιγμή όμως που προσφέρονται στην υπηρεσία του καθηγητή προηγμένες τεχνολογικές δυνατότητες μπορεί να τις εκμεταλλευτεί για να εμπλουτίσει το μάθημα του και με άλλα στοιχεία όπως για παράδειγμα προβολή βίντεο, ταυτόχρονη πλοήγηση σε δικτυακούς τόπους, χρησιμοποίηση και άλλων εφαρμογών εκτός από αυτές για παρουσιάσεις, χρησιμοποίηση προγραμμάτων προσομοίωσης. Με αυτό τον τρόπο μπορούν να πραγματοποιηθούν και εικονικά εργαστήρια (virtual laboratories), να μπορεί γενικά να μιλά και να κινείται με φυσικό τρόπο, όπως θα έκανε και σε μία παραδοσιακή διάλεξη. Να μην χρειάζεται να ασχοληθεί με την τεχνική πλευρά των συστημάτων, ώστε να μπορεί να επικεντρώσει την προσοχή του στο καθαυτό αντικείμενο της διάλεξης.

Προϋποθέσεις

Όπως φαίνεται και από την προηγούμενη παράγραφο δεν μπορεί οποιοδήποτε μάθημα να γίνει με τη μορφή της σύγχρονης εκπαίδευσης. Θα πρέπει όλοι οι συμμετέχοντες να είναι συνδεδεμένοι σε δίκτυο υψηλών ταχυτήτων έτσι ώστε να εξασφαλίζεται η καλή ποιότητα βίντεο και ήχου και να είναι εφικτή η από κοινού χρήση εφαρμογών.

Επίσης, χρειάζεται τουλάχιστον ένα άτομο για τεχνική υποστήριξη στο μάθημα, προκειμένου να ασχολείται με δικτυακά και άλλα προβλήματα που μπορεί να προκύψουν από τη χρήση νέων τεχνολογιών και να υποστηρίζει τον καθηγητή ο οποίος μπορεί να μην είναι εξοικειωμένος με τα τεχνολογικά μέσα.

Όλοι οι συμμετέχοντες θα πρέπει να έχουν στη διάθεσή τους αρκετά προηγμένο εξοπλισμό για τις ανάγκες της σύγχρονης τηλεεκπαίδευσης και τουλάχιστον ο καθηγητής θα πρέπει να βρίσκεται σε αίθουσα ειδικά διαμορφωμένη για να καλύπτει αυτές τις ανάγκες.

Τεχνολογίες στην σύγχρονη εκπαίδευση

Για να υλοποιηθεί μία εικονική αίθουσα αλλά και για να είναι διαθέσιμα όσα προαναφέρθηκαν για την παράδοση της σύγχρονης εκπαίδευσης που θα ικανοποιούνται οι απαιτήσεις που προαναφέρθηκαν έχουν αναπτυχθεί κατάλληλα εργαλεία που χρησιμοποιούν συγκεκριμένα πρωτόκολλα υλοποίησης αλλά και νέες τεχνολογίες δικτύων

4. ΤΕΧΝΟΛΟΓΙΕΣ

a. Η HTML και το Internet

Κάθε σελίδα που εμφανίζεται στο Internet είναι ένα αρχείο γραμμένο με τη γλώσσα HTML (HyperText Markup Language, Γλώσσα Χαρακτηρισμού Υπερ-Κειμένου), που περιλαμβάνει το κείμενο της σελίδας, τη δομή της και τους συνδέσμους προς άλλα έγγραφα, εικόνες ή άλλα μέσα.

Ο φυλλομετρητής (Web browser) παίρνει τις πληροφορίες από τον Web server, τις μορφοποιεί και τις εμφανίζει κατάλληλα για το σύστημά μας. Διαφορετικά προγράμματα φυλλομετρητή μπορεί να μορφοποιούν και να εμφανίζουν το ίδιο αρχείο με διαφορετικό τρόπο, ανάλογα με τις δυνατότητες του συστήματος στο οποίο τρέχουν και τις επιλογές διαμόρφωσης του προγράμματος του φυλλομετρητή.

Μια Web σελίδα ή ιστοσελίδα (Web page) είναι ένα μεμονωμένο στοιχείο μιας παρουσίασης για το Web και περιέχεται σ' ένα αρχείο στον δίσκο, το οποίο ανακτάται από έναν Web server και μορφοποιείται μέσω ενός φυλλομετρητή.

Η αρχική σελίδα (home page) είναι η πρώτη ή κορυφαία σελίδα μιας παρουσίασης για το Web, είναι δηλαδή το σημείο εισόδου ή εκκίνησης για τις υπόλοιπες σελίδες της παρουσίασης και η πρώτη σελίδα που θα συναντήσουν οι αναγνώστες της παρουσίασής μας. Η αρχική σελίδα περιέχει συνήθως μια σύνοψη του περιεχομένου της παρουσίασης με τη μορφή ενός πίνακα περιεχομένων ή μιας ομάδας εικονιδίων.

i. Τι Είναι η HTML

Η HTML είναι το ακρωνύμιο των λέξεων HyperText Markup Language, δηλ. Γλώσσα Χαρακτηρισμού Υπερ-Κειμένου και βασίζεται στη γλώσσα SGML, Standard Generalized Markup Language, που είναι ένα πολύ μεγαλύτερο σύστημα επεξεργασίας εγγράφων.

Η HTML ορίζει ένα σύνολο κοινών στυλ για τις Web σελίδες, όπως τίτλοι (titles), επικεφαλίδες (headings), παράγραφοι (paragraphs), λίστες (lists) και πίνακες (tables). Ορίζει επίσης στυλ χαρακτήρων, όπως η έντονη γραφή (boldface) και οι ενότητες κώδικα.

Κάθε στοιχείο έχει ένα όνομα και περιέχεται μέσα στα σύμβολα <>, που αποκαλούνται tags (ετικέτες). Όταν γράφουμε μια Web σελίδα με την HTML, στην ουσία δίνουμε τίτλους στα διάφορα στοιχεία της σελίδας μ' αυτά τα tags.

Οι φυλλομετρητές, μαζί με τη δυνατότητά τους να ανακτούν σελίδες από το Web, λειτουργούν επίσης και σαν μορφοποιητές για την HTML. Όταν διαβάζουμε μια σελίδα γραμμένη με την HTML σ' έναν φυλλομετρητή, ο φυλλομετρητής διαβάζει (διερμηνεύει) τα tags της HTML και μορφοποιεί το κείμενο και τις εικόνες στην οθόνη.

Διαφορετικοί φυλλομετρητές, οι οποίοι τρέχουν σε διαφορετικούς υπολογιστές, μπορεί να αντιστοιχίζουν διαφορετικά στυλ σε κάθε στοιχείο μιας σελίδας. Αυτό σημαίνει ότι οι σελίδες που δημιουργούμε με την HTML μπορεί να δείχνουν εντελώς διαφορετικές από σύστημα σε σύστημα και από φυλλομετρητή σε φυλλομετρητή. Δηλαδή, οι πραγματικές πληροφορίες και οι σύνδεσμοι που περιέχουν οι σελίδες μας θα είναι πάντα εκεί, αλλά η εμφάνιση των σελίδων στην οθόνη θα είναι διαφορετική.

ii. Η Δομή μιας HTML Σελίδας

Οι σελίδες που γράφουμε με την HTML είναι απλά αρχεία κειμένου σε μορφή ASCII, που σημαίνει ότι δεν περιέχουν πληροφορίες για κάποιο λειτουργικό σύστημα ή πρόγραμμα, αλλά μπορούν να διαβαστούν από οποιοδήποτε συντάκτη υποστηρίζει απλό κείμενο.

Τα αρχεία της HTML περιέχουν τα ακόλουθα :

- Το κείμενο της σελίδας.

- Τα tags της HTML, τα οποία υποδεικνύουν τα στοιχεία, τη δομή και τη μορφοποίηση των σελίδων, καθώς επίσης και τους συνδέσμους υπερ-κειμένου προς άλλες σελίδες ή προς αρχεία άλλων μορφών (πολυμέσα).

Τα περισσότερα tags της HTML έχουν την εξής μορφή :

<ΌνομαTag> επηρεαζόμενο κείμενο </ΌνομαTag>

Τα tags της HTML έχουν γενικά ένα tag αρχής και ένα tag τέλους ή ένα tag ανοίγματος και ένα tag κλεισίματος, τα οποία περικλείουν το κείμενο που επηρεάζουν. Το tag αρχής ενεργοποιεί μια λειτουργία ή ένα χαρακτηριστικό, όπως είναι για παράδειγμα η έντονη γραφή, ενώ το tag τέλους την απενεργοποιεί. Τα tags τέλους έχουν το ίδιο όνομα με τα tags αρχής, αλλά με πρόθεμα τον χαρακτήρα /.

Δεν αποτελούν ζευγάρι όλα τα tags της HTML, καθώς ορισμένα είναι “μονομελή”, ενώ άλλα περιέχουν επιπλέον πληροφορίες και κείμενο μέσα στα σύμβολα <>. Ακόμη, όλα τα tags της HTML δεν κάνουν διάκριση μεταξύ κεφαλαίων και πεζών γραμμάτων, δηλ. μπορούμε να τα γράφουμε είτε με κεφαλαίους είτε με πεζούς χαρακτήρες είτε με οποιονδήποτε συνδυασμό τους. Συνήθως τα γράφουμε κεφαλαία για να μπορούμε να τα ξεχωρίζουμε ευκολότερα από το κείμενο της σελίδας.

Τα αρχεία της HTML πρέπει να τα γράφουμε σε συντάκτες κειμένων, δηλ. προγράμματα που μπορούν να αποθηκεύσουν αρχεία κειμένου σε μορφή ASCII. Τέτοια προγράμματα είναι τα Notepad, Wordpad και Write των Windows, το Edit του DOS και τα vi, emacs και pico του UNIX.

Το όνομα του αρχείου θα πρέπει να έχει επέκταση .html ή .htm σε συστήματα DOS ή Windows που επιτρέπουν επεκτάσεις μόνο τριών χαρακτήρων.

Η HTML χρησιμοποιεί τρία tags για την περιγραφή της συνολικής δομής μιας σελίδας, τα οποία παρέχουν ορισμένες απλές πληροφορίες κεφαλίδας. Αυτά τα tags προσδιορίζουν τη σελίδα μας στους φυλλομετρητές και παρέχουν επίσης απλές πληροφορίες για τη σελίδα, όπως τον τίτλο ή τον συγγραφέα της, πριν από τη φόρτωση ολόκληρης της σελίδας.

b. Mysql

Η MySQL ανήκει στην γενική κατηγορία των DBMS (Database Management Systems) που έχουν σαν πρωταρχικό τους ρόλο την αποθήκευση των δεδομένων, την ελεγχόμενη και ασφαλή πρόσβαση στις πληροφορίες, και την διαχείριση των δικαιωμάτων με τα οποία οι χρήστες μπορούν να επέμβουν και να αλλάξουν στοιχεία πληροφοριών. Κάθε φορά που ένας χρήστης προσπαθεί να διαχειριστεί ένα όγκο δεδομένων προσθέτοντας νέα στοιχεία ή αφαιρώντας κάποια άλλα που δεν ισχύουν πια, το DBMS είναι υπεύθυνο να ακολουθήσει αυτή την διαδικασία από την αρχή της ενεργοποίησης της μέχρι το τέλος της. Παίζει το ρόλο του “μεσάζων” γιατί απλά το DBMS δεν αφήνει το χρήστη να έχει άμεση πρόσβαση στα δεδομένα. Οι εντολές απευθύνονται αποκλειστικά στο DBMS και αφού ελεγχθούν για την εγκυρότητα τους τότε το σύστημα αναλαμβάνει την μεταφορά της πληροφορίας στο χρήστη που την αναζητήσε. Η MySQL ανήκοντας στην κατηγορία των DBMS μπορεί να αναλάβει την διαχείριση πολλών βάσεων μαζί. Κατά την εγκατάσταση έχει ήδη δημιουργηθεί μια βάση με το όνομα mysql η οποία χρησιμοποιείται αποκλειστικά για την καταγραφή πληροφοριών του συστήματος.

Αυτές οι πληροφορίες καταγράφουν την δημιουργία νέων βάσεων, νέων πινάκων, ή ακόμα ενέργειες που έχουν ζητηθεί από τον χρήστη. Με άλλα λόγια, για οποιοδήποτε νέο στοιχείο που δημιουργείται μετά την εγκατάσταση της MySQL, πρέπει το σύστημα να ενημερώνεται και να καταγράφει την ύπαρξη του. Όταν θα δημιουργήσουμε το δικό μας μικρό γκρουπ από δεδομένα, θα πρέπει να τα αποθηκεύσουμε σε μια καινούργια βάση. Έτσι η MySQL πολύ εύκολα θα μπορεί να διαχειριστεί τα καινούργια δεδομένα χωρίς να υπάρξει καμία σύγχυση με αυτά του συστήματος. Η δημιουργία μιας βάσης στην MySQL είναι απλή υπόθεση, αφού το μόνο που απαιτείται είναι η εντολή CREATE DATABASE και το όνομα που θα δώσουμε στην βάση. Εάν προσπαθήσουμε να ονομάσουμε την βάση μας με κάποιο όνομα που ήδη υπάρχει κάτω από την διαχείριση της MySQL, τότε το σύστημα θα μας ειδοποιήσει ότι δεν είναι δυνατή να ολοκληρωθεί αυτή η ενέργεια.

Μια βάση στην MySQL πρακτικά είναι ένα directory το οποίο περιέχει αρχεία τα οποία αντιπροσωπεύουν τους πίνακες (tables). Αυτό σημαίνει ότι απλά εάν φτιάξουμε ένα φάκελο (folder) στο directory της MySQL (/var/lib/mysql) τότε στο αποτέλεσμα της εκτέλεσης της εντολής SHOW DATABASES θα εμφανιζόταν σαν βάση.

c. Java

Στα πλαίσια της διπλωματικής αυτής εργασίας, χρησιμοποιήθηκαν για την ανάπτυξη των εφαρμογών μαθηματικών για μηχανικούς τα applets της Java. Η ραγδαία εξάπλωση του Internet και του παγκόσμιου ιστού (World-Wide Web) δημιούργησαν την ανάγκη νέων τρόπων ανάπτυξης και διανομής του λογισμικού. Οι απαιτήσεις αυτές οδήγησαν στην δημιουργία της γλώσσας προγραμματισμού Java, από την εταιρία Sun microsystems TM. Η Java σχεδιάστηκε με σκοπό την ανάπτυξη εφαρμογών που θα τρέχουν σε ετερογενή δικτυακά περιβάλλοντα.

Τα applets εισήχθησαν στην πρώτη έκδοση της Java το 1995 και χρησιμοποιούνται για να παρέχουν δια-δραστικά χαρακτηριστικά και δυνατότητες στις εφαρμογές Ιστού που δεν μπορούν να παρασχεθούν από τη γλώσσα HTML. Δεδομένου ότι τα bytecode της Java δεν εξαρτώνται από κάποια πλατφόρμα, τα applets μπορούν να εκτελεστούν από οποιονδήποτε browser σε οποιαδήποτε πλατφόρμα, συμπεριλαμβανομένων των Windows, Unix, MAC OS και Linux, αρκεί να έχει εγκατασταθεί πρωτίτερα η Java.

i. Τα εργαλεία της Java

Ακολούθως παρουσιάζονται εν συντομία όλα τα εργαλεία που έρχονται με Java Developers Kit (JDK), της Sun microsystem.

- javac. Είναι ο compiler της Java. Η χρήση του στο command-line είναι : javac <όνομα αρχείου>. Εδώ να σημειώσουμε ότι το javac δεν παράγει ένα αρχείο με όλον τον κώδικα, αλλά χωριστό αρχείο για κάθε κλάση. Τα αρχεία των κλάσεων ονομάζονται : <όνομα κλάσης>.class.
- java. Είναι ο interpreter της Java. Η χρήση του είναι η εξής : java <κλάση>, πχ java myClass και όχι java myClass.class.
- javaw. Είναι παρόμοιο με το java με μόνη την διαφορά ότι δεν χρειάζεται την ύπαρξη φλοιού (shell) για να τρέξει.
- javadoc. Είναι ένα πρόγραμμα για αυτόματη κατασκευή documentation. Είναι αρκετά χρήσιμο στην κατασκευή βοηθημάτων και τεχνικών αναφορών για εφαρμογές οποιουδήποτε μεγέθους.
- appletviewer. Είναι ένα πρόγραμμα το οποίο επιτρέπει να τρέχει κανείς και να χρησιμοποιεί τα διάφορα Java applets χωρίς να χρειάζεται να τα ενσωματώσει σε κάποια html σελίδα. Οι stand-alone εφαρμογές, ωστόσο, δεν τρέχουν στον appletviewer αλλά κατευθείαν στον java ή javaw.

Η Java έχει τα ακόλουθα χαρακτηριστικά

- Αντικειμενοστραφής (ομοιότητες εντολών με τη C++).
- Είναι Interpreted γλώσσα. Αυτό σημαίνει ότι ο java compiler δεν παράγει εκτελέσιμο κώδικα αλλά μια μορφή ψευδοκώδικα (bytecode) το οποίο από μόνο του δεν τρέχει σε καμία μηχανή. Προκειμένου λοιπόν να εκτελεστεί απαιτείται η χρήση ενός interpreter (=διερμηνέα) για να μετατρέψει το bytecode σε πραγματικό εκτελέσιμο κώδικα. Αυτό το χαρακτηριστικό δίνει τη δυνατότητα στα java bytecodes να μπορούν να τρέξουν σε οποιοδήποτε μηχάνημα, κάτω από οποιοδήποτε λειτουργικό, αρκεί να έχει εγκατασταθεί ένας java interpreter. Επίσης ένα άλλο χαρακτηριστικό του java bytecode είναι το μικρό του μέγεθος, (μόλις λίγα Kilobytes). Αυτό το κάνει ιδανικό για μετάδοση μέσω του δικτύου.
- Κατανεμημένα (distributed). Δηλαδή ένα πρόγραμμα σε Java είναι δυνατό να το φέρουμε από το δίκτυο και να το τρέξουμε. Επίσης είναι δυνατό διαφορετικά κομμάτια του προγράμματος να έρθουν από διαφορετικά sites.

- Ασφαλής (secure). Στο δίκτυο όμως ελλοχεύουν πολλοί κίνδυνοι για τον χρήστη - παραλήπτη μιας δικτυακής εφαρμογής, γι' αυτό η Java έχει σχεδιαστεί έτσι ώστε να ελαχιστοποιείται η πιθανότητα προσβολής του συστήματος του χρήστη από κάποιο applet γραμμένο για τέτοιο σκοπό.
- Είναι multithreaded. Η Java υποστηρίζει εγγενώς την χρήση πολλών threads. Προκειμένου να το επιτύχει αυτό σε συστήματα με έναν επεξεργαστή, το Java runtime system (interpreter) υλοποιεί ένα δικό χρονοδρομολογητή (scheduler), ενώ σε συστήματα που υποστηρίζουν πολυεπεξεργασία η δημιουργία των threads ανατίθεται στο λειτουργικό σύστημα. Φυσικά όλα αυτά είναι αόρατα τόσο στον προγραμματιστή όσο και στον χρήστη.
- Υποστηρίζει multimedia εφαρμογές. Με αυτό εννοούμε ότι η Java παρέχει ευκολίες στη δημιουργία multimedia εφαρμογών. Αυτό επιτυγχάνεται τόσο με την ευελιξία της σαν γλώσσα όσο και με τις πλούσιες και συνεχώς εμπλουτιζόμενες βιβλιοθήκες της.

5. ΣΧΕΔΙΑΣΜΟΣ ΕΦΑΡΜΟΓΗΣ

5.1 Περιγραφή Απαιτήσεων – Βασικές Απαιτήσεις

Φιλικό περιβάλλον. Στόχος η ύπαρξη μιας φιλικής προς το χρήστη διεπαφής η οποία να μην αποτρέπει τη χρήση του συστήματος και να βοηθά τους μη έμπειρους χρήστες. Επίσης πρέπει να υπάρχει στην ίδια την εφαρμογή Online βοήθεια που να εξηγεί στους χρήστες τις δυνατότητες της εφαρμογής και τα αποτελέσματα κάθε ενέργειας.

Έλλειψη ανάγκης για ειδικό λογισμικό σύνδεσης. Απαιτείται η ελάχιστη συμμετοχή χρηστών στην εγκατάσταση συστημάτων και τη λειτουργία του συστήματος. Η έλλειψη ειδικού λογισμικού για τη χρήση της εφαρμογής ελαχιστοποιεί το κόστος για τη χρήση του συστήματος. Η εφαρμογή για το λόγο αυτό πρέπει να σχεδιαστεί ως Εφαρμογή Ιστού.

Στην τεχνολογία λογισμικού, μια εφαρμογή Ιστού είναι μια εφαρμογή που προσεγγίζεται μέσω ενός φυλλομετρητή ιστοσελίδων (web browser) πάνω από το δίκτυο. Ένας φυλλομετρητής ιστοσελίδων είναι ένα λογισμικό που επιτρέπει στον χρήστη του να προβάλλει, και να αλληλεπιδρά με, κείμενα, εικόνες, βίντεο, μουσική, παιχνίδια και άλλες πληροφορίες συνήθως αναρτημένες σε μια ιστοσελίδα ενός ιστότοπου στον Παγκόσμιο Ιστό ή σε ένα τοπικό δίκτυο. Το κείμενο και οι εικόνες σε μια ιστοσελίδα μπορεί να περιέχουν συνδέσμους προς άλλες ιστοσελίδες του ίδιου ή διαφορετικού ιστότοπου. Ο φυλλομετρητής επιτρέπει στον χρήστη την γρήγορη και εύκολη πρόσβαση σε πληροφορίες που βρίσκονται σε διάφορες ιστοσελίδες και ιστότοπους εναλλάσσοντας τις ιστοσελίδες μέσω συνδέσμων (links). Η κύρια γλώσσα που χρησιμοποιείται από τις εφαρμογές ιστού και τους φυλλομετρητές είναι η γλώσσα μορφοποίησης HTML για την προβολή των ιστοσελίδων. Για την ανάπτυξη εφαρμογών που χρειάζονται περισσότερες διαδραστικά χαρακτηριστικά και δυνατότητες χρησιμοποιούνται και άλλες γλώσσες προγραμματισμού παράλληλα με την HTML όπως η JavaScript η Java και η PHP.

Οι εφαρμογές Ιστού είναι δημοφιλείς επειδή δεν υπάρχει σύγχρονο λειτουργικό σύστημα που προορίζεται για υπολογιστές γραφείου που να μην έχει προ-εγκατεστημένο έναν ή περισσότερους φυλλομετρητές. Οι φυλλομετρητές ουσιαστικά αποτελούν λογισμικό πελάτη του δικτυακού πρωτοκόλλου επιπέδου εφαρμογών HTTP. Για κάθε web browser διατίθενται, επίσης, και αρκετά πρόσθετα στοιχεία (add-ons), με στόχο την επαύξηση των δυνατοτήτων τους, τη βελτίωση της χρηστικότητας τους και την προστασία του χρήστη σε θέματα ασφάλειας. και η ευκολία της χρησιμοποίησης μιας μηχανής αναζήτησης Ιστού ως πελάτη, αποκαλούμενη μερικές φορές λεπτό πελάτη. Η δυνατότητα να ενημερωθούν και να διατηρηθούν οι εφαρμογές Ιστού χωρίς τη διανομή και εγκατάσταση του λογισμικού ενδεχομένως σε χιλιάδες υπολογιστές πελατών είναι ένας βασικός λόγος για τη δημοτικότητά τους.

Γρήγορη πρόσβαση στην εφαρμογή. Επειδή η εφαρμογή είναι δικτυακή, επηρεάζεται από την απόδοση του δικτύου. Στόχος είναι η ελαχιστοποίηση των δεδομένων που ανταλλάσσονται ώστε η επικοινωνία να είναι όσο το δυνατόν γρηγορότερη. Όταν μια διεπαφή είναι αργή επηρεάζει αρνητικά τους χρήστες και μειώνεται σημαντικά η χρηστικότητά της.

Ασφάλεια. Το σύστημα πρέπει να είναι ασφαλές στην ανταλλαγή στοιχείων, και να μην επιτρέπει τη διαρροή στοιχείων σε κακόβουλους χρήστες. Επίσης πρέπει να εξασφαλίζει ότι καμία πληροφορία για προσφορά δεν θα χαθεί. Είναι επίσης ουσιαστικό ότι το σύστημα υποστηρίζει τη μυστικότητα προσωπικών και κρίσιμων δεδομένων.

Ταυτοποίηση. Κάθε χρήστης που έχει πρόσβαση στο περιβάλλον εξ'αποστάσεως μάθησης πρέπει να έχει ταυτοποιηθεί πρωτίτερα.

5.2 Περιγραφή Απαιτήσεων: Ειδικές Απαιτήσεις

Η εφαρμογή που αναπτύσσεται στα πλαίσια της παρούσας εργασίας χρησιμοποιείται για την εξ' αποστάσεως εκπαίδευση μαθητών/φοιτητών σε κάποιο μάθημα. Πιο συγκεκριμένα, οι ειδικές απαιτήσεις της εφαρμογής είναι οι παρακάτω:

Θεωρία Μαθήματος. Η εφαρμογή πρέπει να έχει την θεωρία του μαθήματος. Η ύλη πρέπει να χωρίζεται σε κεφάλαια. Η θεωρία πρέπει να είναι ευανάγνωστη μέσα από την εφαρμογή. Αυτό δεν έχει σχέση με το περιεχόμενο, η ποιότητα του οποίου δεν αποτελεί απαίτηση της εφαρμογής, αλλά έχει να κάνει με την εύκολη περιήγηση. Για το λόγο αυτό προτείνεται η περιήγηση να είναι σαν ενός βιβλίου, ώστε να είναι πιο οικεία στο μαθητή. Η ύλη λοιπόν χωρίζεται σε κεφάλαια και στη συνέχεια σε σελίδες, προσομοιάζοντας ένα βιβλίο.

Ηλεκτρονικές Ασκήσεις. Προκειμένου να γίνει εμπέδωση της διδαχθείσας ύλης, πρέπει να υπάρχουν online ασκήσεις. Οι ασκήσεις μπορεί να είναι διαφορετικού τύπου και δυσκολίας. Για παράδειγμα μπορεί να έχουμε ερωτήσεις πολλαπλής επιλογής με 2 πιθανές απαντήσεις, αλλά και με περισσότερες. Τα διαφορετικά επίπεδα και είδη ασκήσεων βοηθούν τον καθηγητή να αντιληφθεί και αν εντοπίσει προβλήματα στην κατανόηση του μαθήματος από τους μαθητές. Καθώς οι ασκήσεις αφορούν διαφορετικές ενότητες της θεωρίας και έχουν διαφορετικό τύπο ή δυσκολία, πρέπει να χωρίζονται σε διαφορετικά Online διαγωνίσματα (tests).

Επικοινωνία με το διδάσκοντα. Σε online συστήματα είναι χρήσιμο να υπάρχει δυνατότητα επικοινωνίας με τους υπευθύνους. Για το λόγο αυτό, η δικτυακή εφαρμογή πρέπει να παρέχει στοιχεία επικοινωνίας με το διδάσκοντα. Ιδανικό είναι να υποστηρίζεται τόσο ασύγχρονη όσο και σύγχρονη επικοινωνία ανάμεσα στα δυο μέρη. Συνεπώς η εφαρμογή πρέπει να υποστηρίζει άμεση ανταλλαγή μηνυμάτων (Instant Messaging). Όταν κάποιος από τα δυο μέρη δεν είναι Online, το μήνυμα μπορεί να αποθηκεύεται και ο χρήστης να μπορεί να δει το μήνυμα και να απαντήσει αργότερα.

Στατιστικά στοιχεία. Ο καθηγητής πρέπει να έχει τη δυνατότητα ελέγχου της προόδου των μαθητών. Πιο συγκεκριμένα, πρέπει να έχει πρόσβαση στα αποτελέσματα των διαγωνισμάτων των μαθητών, με δυνατότητα ελέγχου του τρόπου εμφάνισης των αποτελεσμάτων. Δηλαδή, τα αποτελέσματα πρέπει να εμφανίζονται είτε με βάση το μαθητή, ώστε να ελέγχονται οι επιδόσεις ενός συγκεκριμένου ατόμου, είτε με βάση τις κατηγορίες των διαγωνισμάτων, ώστε να ελέγχονται οι επιδόσεις όλων σε κάποιο συγκεκριμένο κομμάτι της θεωρίας. Με τον τρόπο αυτό ο καθηγητής μπορεί να εντοπίσει αδυναμίες ή κενά στην διαδικασία εκμάθησης.

5.3 Σχεδιασμός Συστήματος

Στην ενότητα αυτή γίνεται αναλυτικός σχεδιασμός της εφαρμογής, με βάση τις απαιτήσεις που παρουσιάστηκαν προηγουμένως. Ο σχεδιασμός του συστήματος αφορά αρχικά τον ορισμό των οντοτήτων που υπάρχουν στο σύστημα. Οι οντότητες αυτές αφορούν βασικές έννοιες και στοιχεία του συστήματος και αφού οριστούν θα συσχετιστούν με διάφορες διαδικασίες.

Στη συνέχεια λοιπόν θα γίνει αναλυτική περιγραφή των διαδικασιών του συστήματος σε μορφή διαγραμμάτων UML. Τα διαγράμματα αυτά θα παρουσιάσουν συσχετίσεις ανάμεσα στις οντότητες του συστήματος.

Οι οντότητες και οι συσχετίσεις τους θα είναι παρούσες και στο σχεδιασμό της βάσης δεδομένων του συστήματος. Για το λόγο αυτό θα παρουσιαστεί ένα αναλυτικό διάγραμμα οντοτήτων-συσχετίσεων (E-R diagram) και στη συνέχεια θα γίνει αντιστοίχιση του σε πίνακες οι οποίοι θα δημιουργηθούν στο σύστημα βάσεων δεδομένων.

5.3.1 Ορισμός οντοτήτων

Χρήστης. Η οντότητα αυτή περιγράφει τους εξωτερικούς χρήστες του συστήματος. Κάθε χρήστης χαρακτηρίζεται από τα παρακάτω γνωρίσματα: όνομα, επίθετο, username (μοναδικό), κωδικό πρόσβασης, e-mail. Επίσης, χρειάζεται ένα ακόμα πεδίο τύπου Boolean, που να ορίζει αν ο χρήστης είναι διαχειριστής (καθηγητής) ή όχι στο σύστημα.

Κεφάλαιο. Η οντότητα αυτή χωρίζει τη διδαχθείσα ύλη σε ξεχωριστές ενότητες. Κάθε κεφάλαιο πρέπει να έχει σα πεδία το αντίστοιχο όνομα και τη σειρά προτεραιότητας.

Σελίδα. Η οντότητα αυτή χωρίζει τη διδαχθείσα ύλη ενός κεφαλαίου σε ξεχωριστές σελίδες. Κάθε σελίδα πρέπει να έχει σα πεδία το αντίστοιχο αριθμό προτεραιότητας, την αντιστοίχιση με το κεφάλαιο στο οποίο ανήκει και το περιεχόμενο.

Κατηγορία. Η οντότητα αυτή χωρίζει τις ασκήσεις/ερωτήσεις ενός κεφαλαίου σε ξεχωριστές κατηγορίες. Σε κάθε κεφάλαιο υπάρχουν βασικές κατηγορίες και υποκατηγορίες (δύο επίπεδα). Κάθε κατηγορία πρέπει να έχει σα πεδία το αντίστοιχο αριθμό προτεραιότητας, την αντιστοίχιση με το κεφάλαιο στο οποίο ανήκει, την μητρική κατηγορία (αν πρόκειται για υποκατηγορία).

Ερώτηση. Η οντότητα αυτή αφορά ερωτήσεις που ανήκουν σε κάποιο κεφάλαιο και κατηγορία. Κάθε ερώτηση πρέπει να έχει σα πεδία το αντίστοιχο αριθμό προτεραιότητας, την αντιστοίχιση με το κεφάλαιο και την κατηγορία στα οποία ανήκει και το περιεχόμενο της ερώτησης. Επίσης υπάρχει ένα ακόμα πεδίο που ορίζει το επίπεδο/τύπο ερώτησης.

Απάντηση. Η οντότητα αυτή αφορά πιθανές απαντήσεις που αφορούν κάποια ερώτηση. Κάθε ερώτηση πρέπει να έχει δύο ή περισσότερες πιθανές ερωτήσεις. Συνεπώς για την οντότητα αυτή χρειάζεται ένα πεδίο για το περιεχόμενο της απάντησης και ένα ακόμα πεδίο που ορίζει αν η συγκεκριμένη ερώτηση είναι η σωστή.

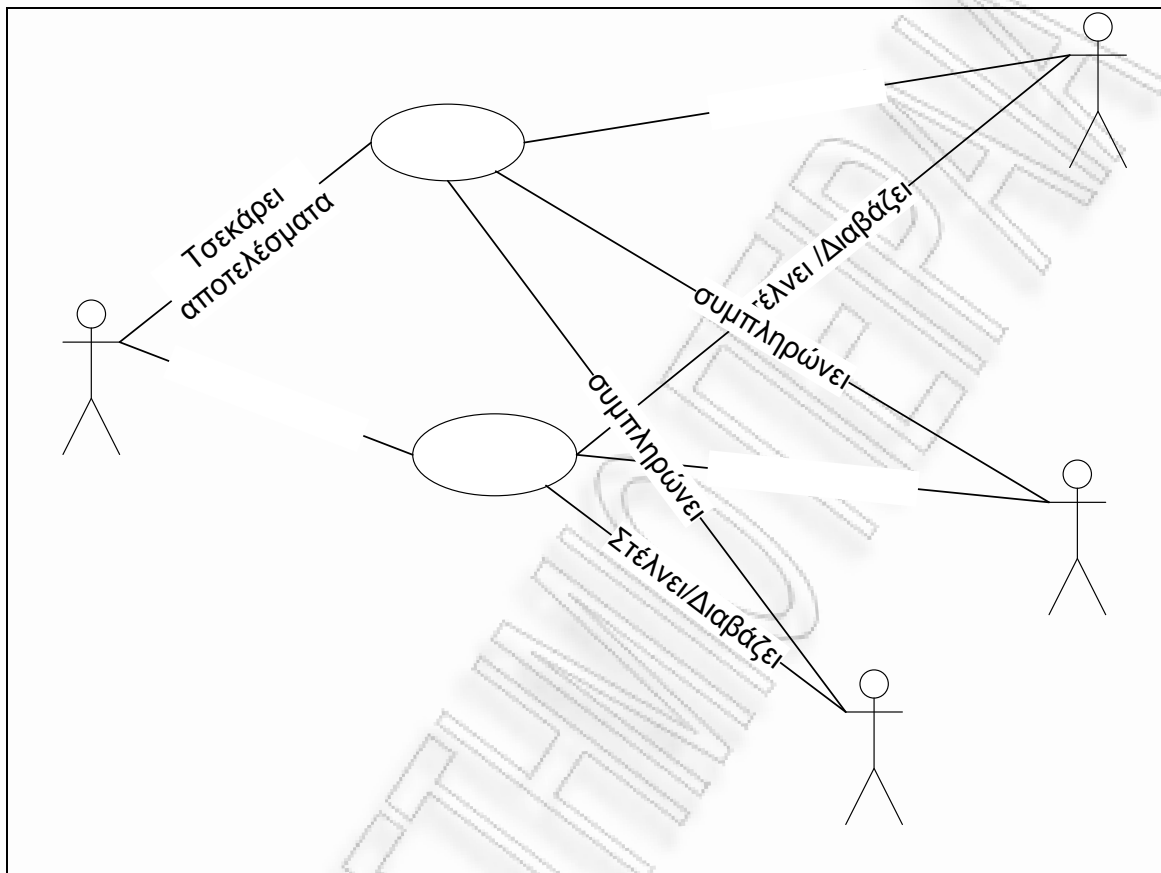
Διαγώνισμα. Η οντότητα αυτή έχει ως στόχο την αποθήκευση των αποτελεσμάτων ενός τεστ (ενός συνόλου από ερωτήσεις). Η συσχέτιση στο σύνολο από ερωτήσεις γίνεται με τον ορισμό του επιπέδου και της κατηγορίας στην οποία ανήκουν οι ερωτήσεις. Κάθε «διαγώνισμα» έχει ακόμα ως πεδία το username του φοιτητή στο οποίο αναφέρεται, τη βαθμολογία του μαθητή καθώς και την ημερομηνία διεξαγωγής.

Μήνυμα. Η οντότητα αυτή αφορά τα μηνύματα που ανταλλάσσονται ανάμεσα σε καθηγητή και μαθητές. Συνεπώς χρειάζεται ως πεδία την ημερομηνία αποστολής, τον παραλήπτη και τον αποδέκτη του μηνύματος (usernames) και το περιεχόμενο του μηνύματος.

5.3.2 Σχεδιασμός Διεπαφής και Διαγράμματα δραστηριότητας

Στη συνέχεια ακολουθεί ανάλυση των προβλεπόμενων διαδικασιών που θα εκτελούνται στο σύστημα. Οι διαδικασίες αυτές θα παρουσιαστούν με τη μορφή διαγραμμάτων περίπτωσης (use case diagram) και διαγραμμάτων δραστηριότητας (activity diagram). Ένα use case diagram περιγράφει σε αφηρημένο επίπεδο την αλληλεπίδραση ανάμεσα στους συμμετέχοντες σε μια περίπτωση λειτουργίας του συστήματος. Τα διαγράμματα δραστηριότητας είναι γραφικές αναπαραστάσεις ροών εργασίας (workflows) διαδικασιών οι οποίες μπορούν να χωριστούν σε ξεχωριστά λογικά βήματα. Τα διαγράμματα αυτά αποτελούνται από κόμβους διεργασιών, αποφάσεων και υποστηρίζουν βρόγχους επανάληψης και παράλληλες διεργασίες.

Στη συνέχεια ακολουθεί το use case diagram του συστήματος που περιγράφει και τη βασική λειτουργικότητά του όσον αφορά την εισαγωγή πληροφορίας στο σύστημα και την ανταλλαγή πληροφορίας ανάμεσα στους χρήστες του συστήματος. Οι δυο actors του διαγράμματος είναι ο διαχειριστής και οι φοιτητές/μαθητές. Το σύστημα ενσωματώνει ένα σύστημα για την ανταλλαγή μηνυμάτων ανάμεσα στο διαχειριστή/καθηγητή και τους απλούς χρήστες/φοιτητές. Επίσης, κάθε φοιτητής μπορεί να συμπληρώσει κάποιο διαγώνισμα, το οποίο στη συνέχεια ελέγχεται από τον καθηγητή.



Σχήμα 1: Use case Diagram

Το λογισμικό χωρίζεται σε διαφορετικά Modules κάθε ένα από τα οποία αφορούν διαφορετική λειτουργικότητα. Τα modules επιτρέπουν τη ξεχωριστή ανάπτυξη της λειτουργικότητας σε λογικά ξεχωριστά τμήματα του λογισμικού. Η σχεδίαση με βάση αυτή τη λογική συνίσταται στην ανάπτυξη λογισμικού, καθώς έτσι το λογισμικό μπορεί να αναβαθμιστεί ευκολότερα. Η ανάπτυξη σε κάποιο Module και η προσθήκη νέων δυνατοτήτων ή διόρθωση σφαλμάτων γίνεται ευκολότερα όταν κάνοντας αλλαγές σε κάποιο τμήμα του λογισμικού δεν επηρεάζονται τα υπόλοιπα.

Τα Modules από τα οποία αποτελείται το σύστημα είναι:

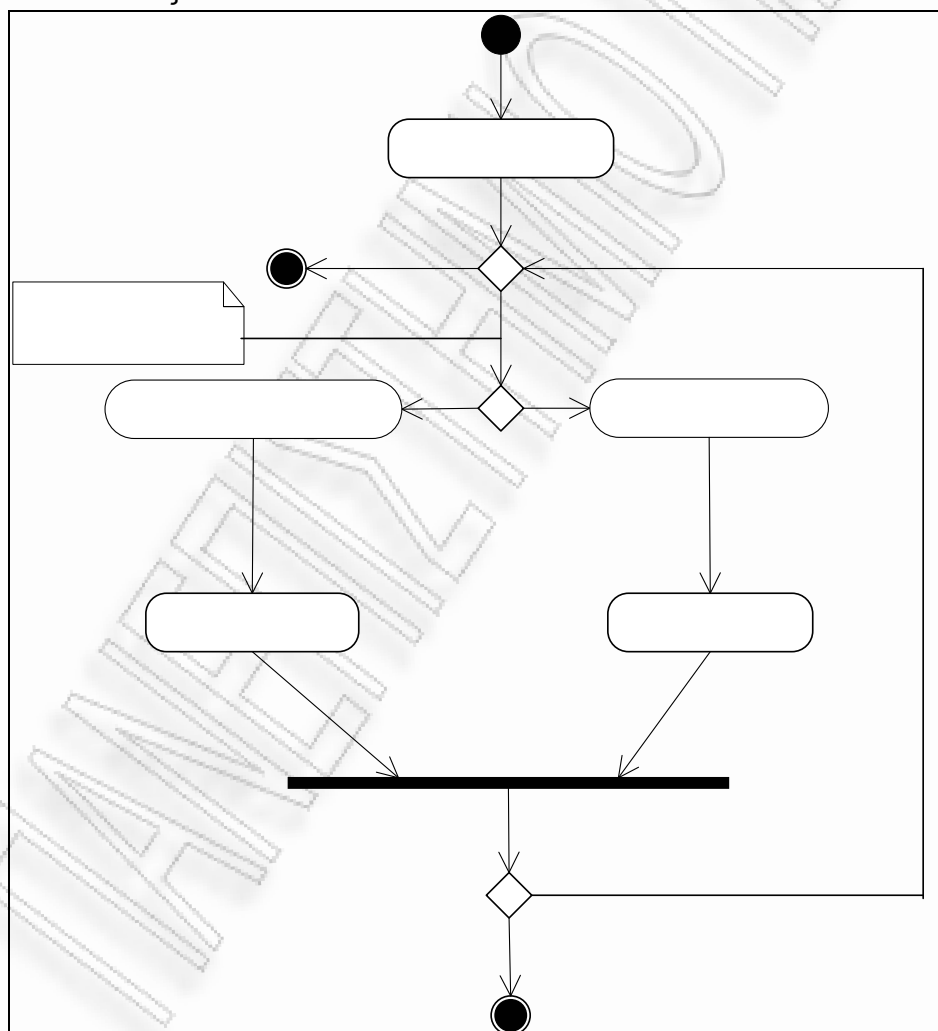
- Εισαγωγή – Εισαγωγή στο σύστημα και ενημέρωση για τη βασική λειτουργία του
- Θεωρία – Στην ενότητα αυτή μπορεί ο χρήστης να περιηγηθεί στη θεωρία η οποία χωρίζεται σε Κεφάλαια, και τα Κεφάλαια σε σελίδες, προσομοιάζοντας έτσι την ανάγνωση ενός βιβλίου.
- Ασκήσεις – Στην ενότητα αυτή ο χρήστης μπορεί να περιηγηθεί στις ασκήσεις του κάθε κεφαλαίου, με αντιστοίχιση στα κεφάλαια της θεωρίας. Υπάρχει η δυνατότητα συμπλήρωσης των ερωτήσεων που αποτελούν ένα «Διαγώνισμα» όπως αυτό ορίστηκε σε προηγούμενη ενότητα, καθώς και η δυνατότητα αναιρέσης (για την επανάληψη του διαγωνίσματος).

- Διαχείριση – Η ενότητα αυτή είναι εμφανής μόνο στο διαχειριστή/καθηγητή του συστήματος. Μέσα από την ενότητα αυτή μπορεί να ανταλλάσσει μηνύματα με τους μαθητές και να ελέγχει την επίδοσή τους στα «Διαγωνίσματα».
- Μηνύματα – Η ενότητα αυτή είναι διαθέσιμη μόνο στους μαθητές και χρησιμεύει για την ανταλλαγή μηνυμάτων με τον καθηγητή.
- Βοήθεια – Η ενότητα αυτή έχει Online οδηγίες χρήσεως της εφαρμογής.

Στη συνέχεια παρουσιάζεται ο σχεδιασμός κάθε ενότητας του προγράμματος.

Θεωρία

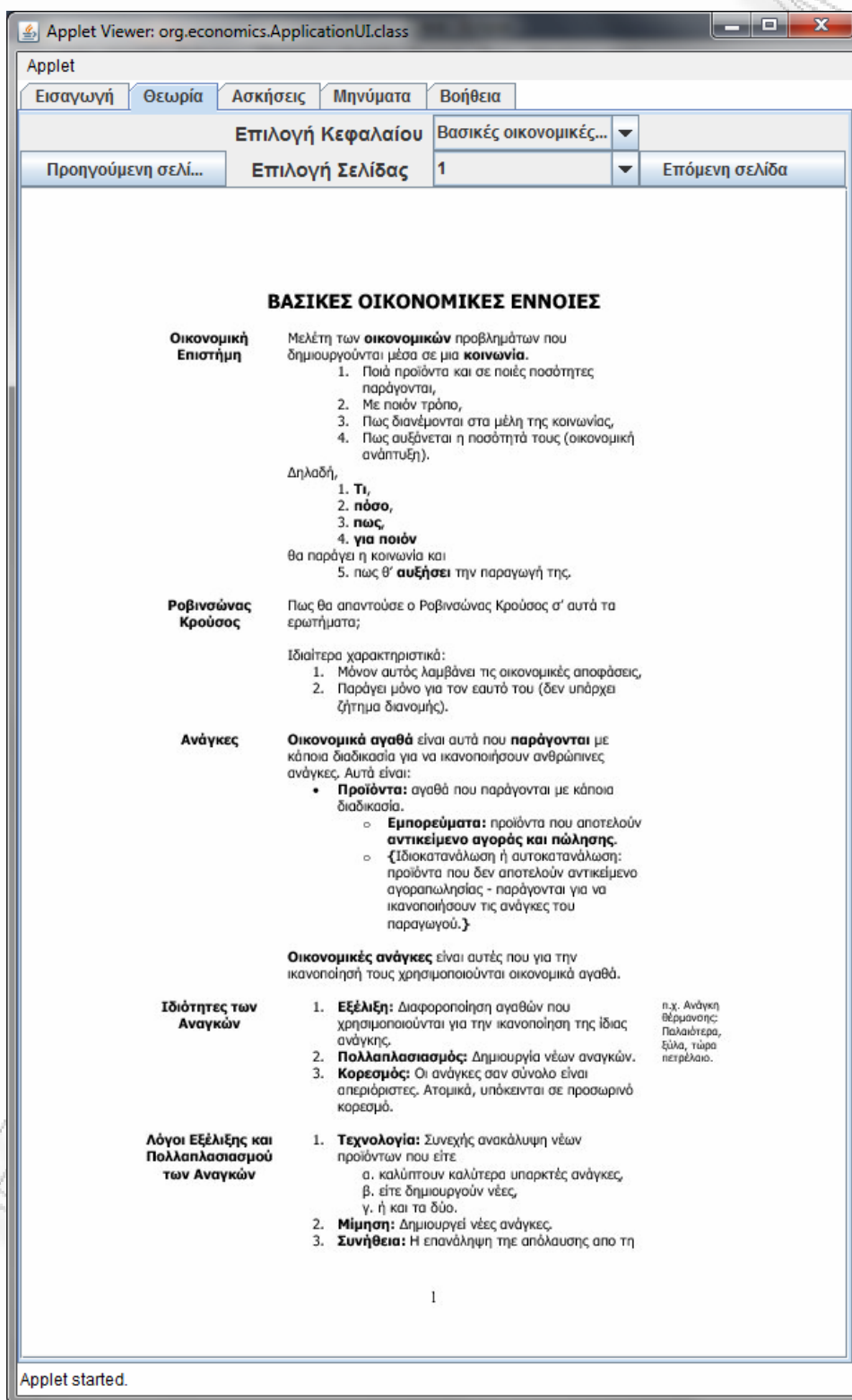
Μια ενότητα του λογισμικού αφορά την ανάγνωση της θεωρίας. Η θεωρία όπως αναλύσαμε προηγουμένως χωρίζεται σε Κεφάλαια και κάθε Κεφάλαιο αποτελείται από 1 ή περισσότερες σελίδες. Στο παρακάτω διάγραμμα δραστηριότητας προδιαγράφεται ο μηχανισμός επιλογής Κεφαλαίου και σελίδας.



Σχήμα 2: Διάγραμμα δραστηριότητας - Επιλογή Κεφαλαίου/Σελίδας

Για την λειτουργία του παραπάνω σχήματος, χρειάζεται η διεπαφή να έχει κατάλληλα πλήκτρα για την επιλογή. Για την εύκολη επιλογή Κεφαλαίου, προτιμάται η ύπαρξη ενός drop down

menu, τύπου JComboBox στη Java. Με τη βοήθεια του JComboBox ο χρήστης μπορεί να δει ποια είναι τα διαθέσιμα Κεφάλαια και να επιλέξει. Κατά αναλογία, χρειάζεται ένα JComboBox για την εμφάνιση των διαθέσιμων σελίδων. Ειδικά στις σελίδες, όπου προτείνεται η σειριακή επιλογή τους, ώστε το περιεχόμενο να διαβάζεται με τη ροή με την οποία έχει προδιαγραφεί, συνίσταται και η χρήση πλήκτρων για την επιλογή της αμέσως επόμενης και προηγούμενης σελίδας. Τα παραπάνω εμφανίζονται στο παρακάτω σχήμα με το στιγμιότυπο της εφαρμογής.



Σχήμα 3: Μενού θεωρίας

Ασκήσεις

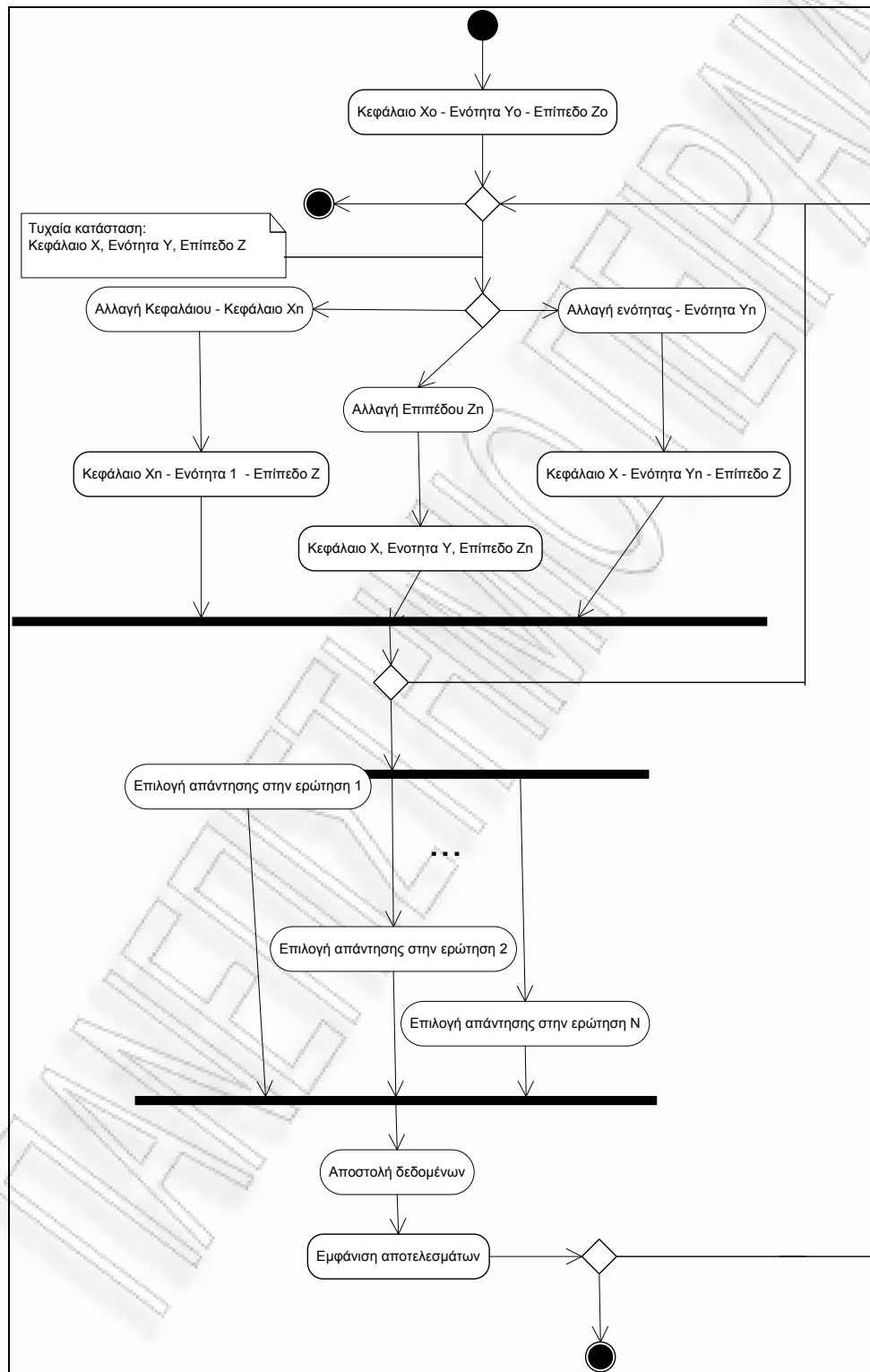
Η έχει αυτή έχει ως στόχο την αποθήκευση των αποτελεσμάτων ενός τεστ (ενός συνόλου από ερωτήσεις). Η συσχέτιση στο σύνολο από ερωτήσεις γίνεται με τον ορισμό του επιπέδου και της κατηγορίας στην οποία ανήκουν οι ερωτήσεις. Κάθε «διαγώνισμα» έχει ακόμα ως πεδία το username του φοιτητή στο οποίο αναφέρεται, τη βαθμολογία του μαθητή καθώς και την ημερομηνία διεξαγωγής.

Συνεπώς, απαιτείται η εφαρμογή να δίνει τη δυνατότητα περιήγησης στα κεφάλαια και τις ενότητες καθώς και η επιλογή του επιπέδου από τον χρήστη. Για την λειτουργία αυτή, προτιμάται η ύπαρξη ενός drop down menu , τύπου JComboBox στη Java. Με τη βοήθεια του JComboBox ο χρήστης μπορεί να δει ποια είναι τα διαθέσιμα Κεφάλαια και να επιλέξει. Κατά αναλογία, χρειάζεται ένα JComboBox για την εμφάνιση των διαθέσιμων ενοτήτων και επιπέδων. Ειδικά στις ενότητες υπάρχει και η δυνατότητα χρήσης πλήκτρων για την επιλογή της αμέσως επόμενης και προηγούμενης ενότητας.

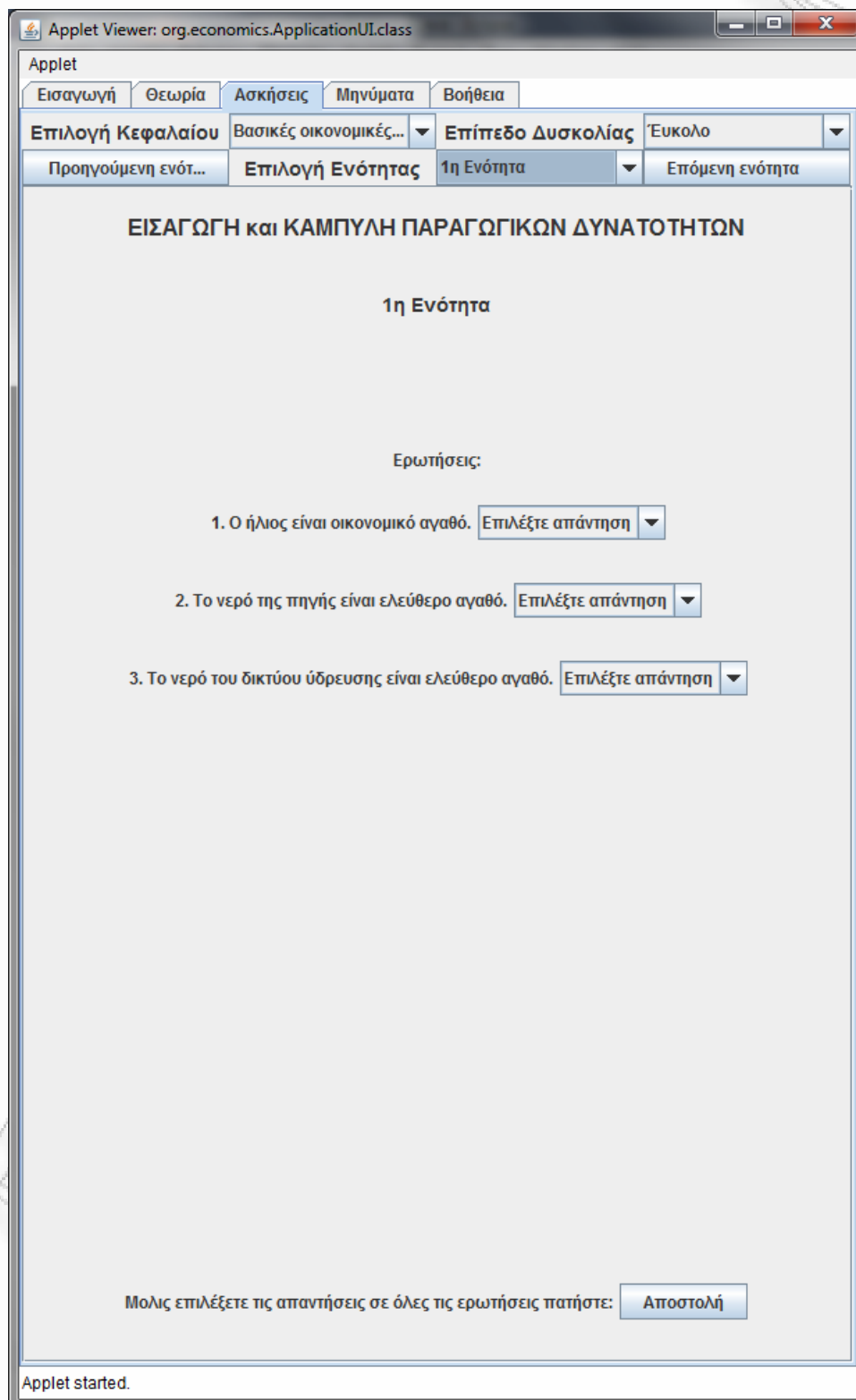
Η ερωτήσεις είναι πολλαπλών επιλογών, με το χρήστη να καλείται να επιλέξει τη σωστή με τη χρήση ενός JComboBox για κάθε ερώτηση.

Τελικά πρέπει να υπάρχει και η επιλογή «Αποστολή» για την καταχώρηση των αποτελεσμάτων.

Ακολουθούν το διάγραμμα ροής που περιγράφει την παραπάνω διαδικασία και η αντίστοιχη διεπαφή.



Σχήμα 4: Διάγραμμα δραστηριότητας - Επιλογή διαγωνίσματος και εκτέλεση

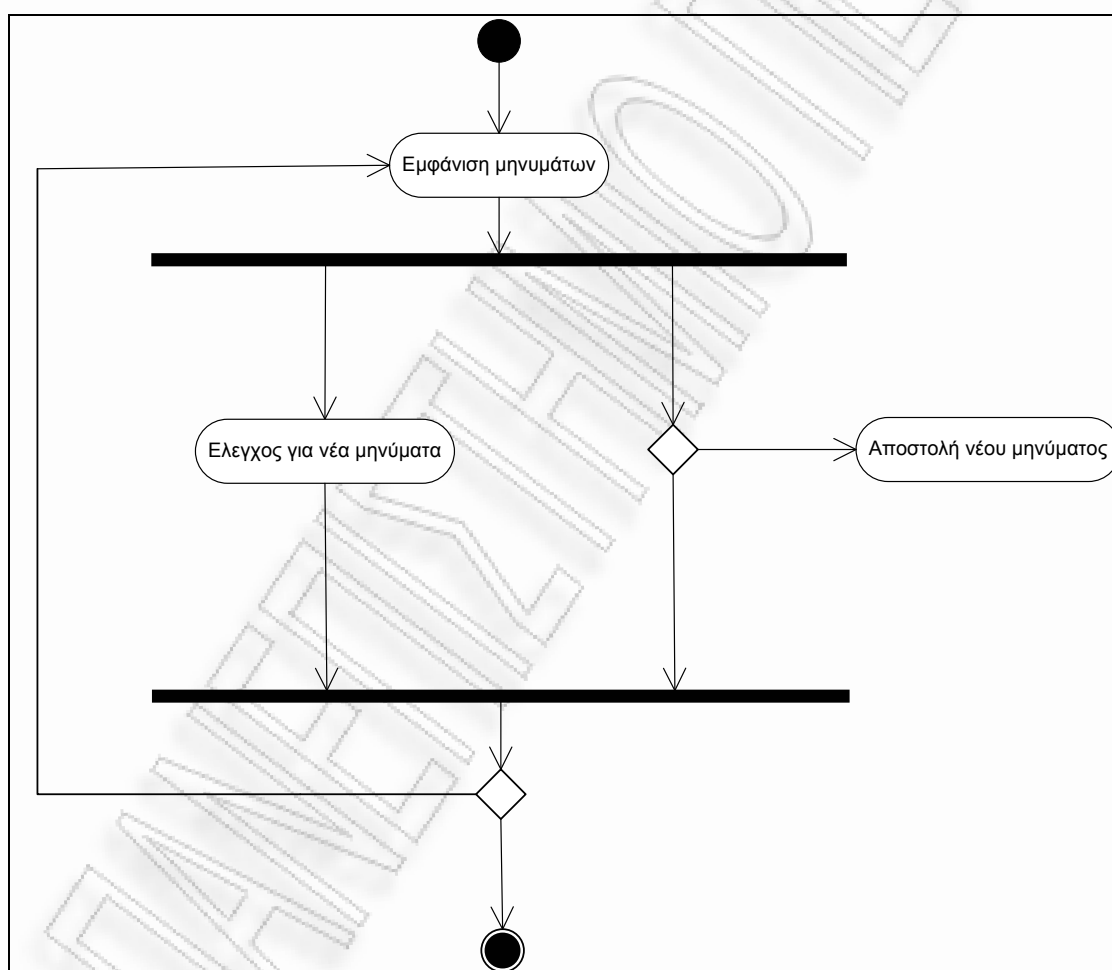


Σχήμα 5: Διεπαφή "Διαγωνίσματος"

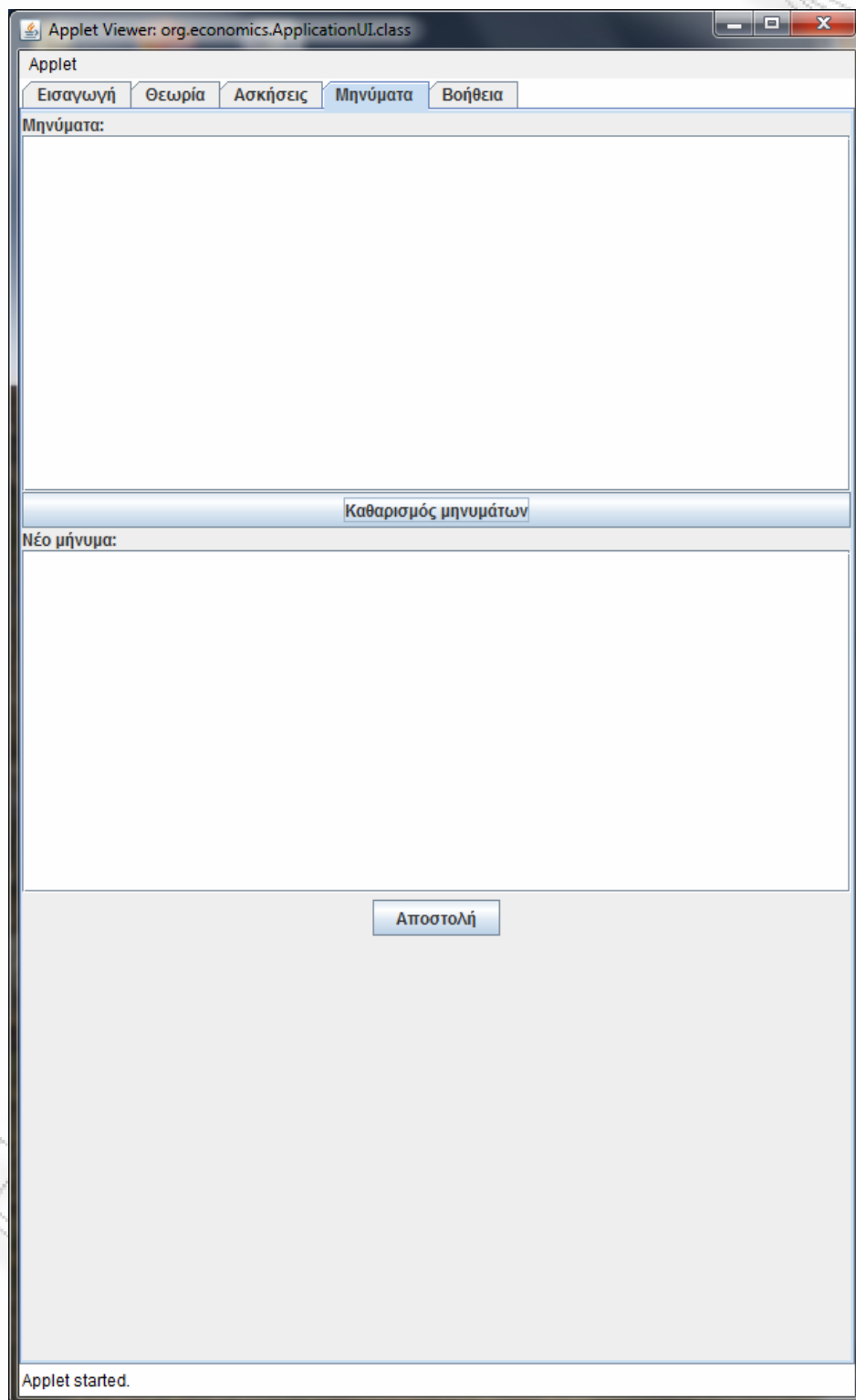
Μηνύματα

Είναι ενότητα του λογισμικού που αφορά την επικοινωνία ανάμεσα στους χρήστες. Με την ανταλλαγή μηνυμάτων επιτυγχάνεται τόσο σύγχρονη όσο και ασύγχρονη επικοινωνία ανάμεσα σε καθηγητή και φοιτητές. Όταν το μήνυμα στέλνεται από το καθηγητή, απαιτείται να οριστεί ο αποδέκτης του μηνύματος (username) και το περιεχόμενο του μηνύματος, ενώ όταν στέλνεται από το μαθητή απαιτείται μονάχα το περιεχόμενο. Παράλληλα, πρέπει να υπάρχει η δυνατότητα διαγραφής των παλαιότερων μηνυμάτων.

Προκειμένου να γίνεται ανανέωση των μηνυμάτων στην αντίστοιχη οθόνη, χρειάζεται σε κάθε σύστημα να τρέχει στο background μια διαδικασία η οποία να ελέγχει περιοδικά για την λήψη νέων μηνυμάτων. Η διαδικασία αυτή υλοποιείται με την κλάση Timer η οποία στην ουσία «ξυπνά» ανά τακτά χρονικά διαστήματα και υλοποιεί μια δεδομένη λειτουργία.



Σχήμα 6: Διάγραμμα δραστηριότητας - Αποστολή και λήψη μηνυμάτων



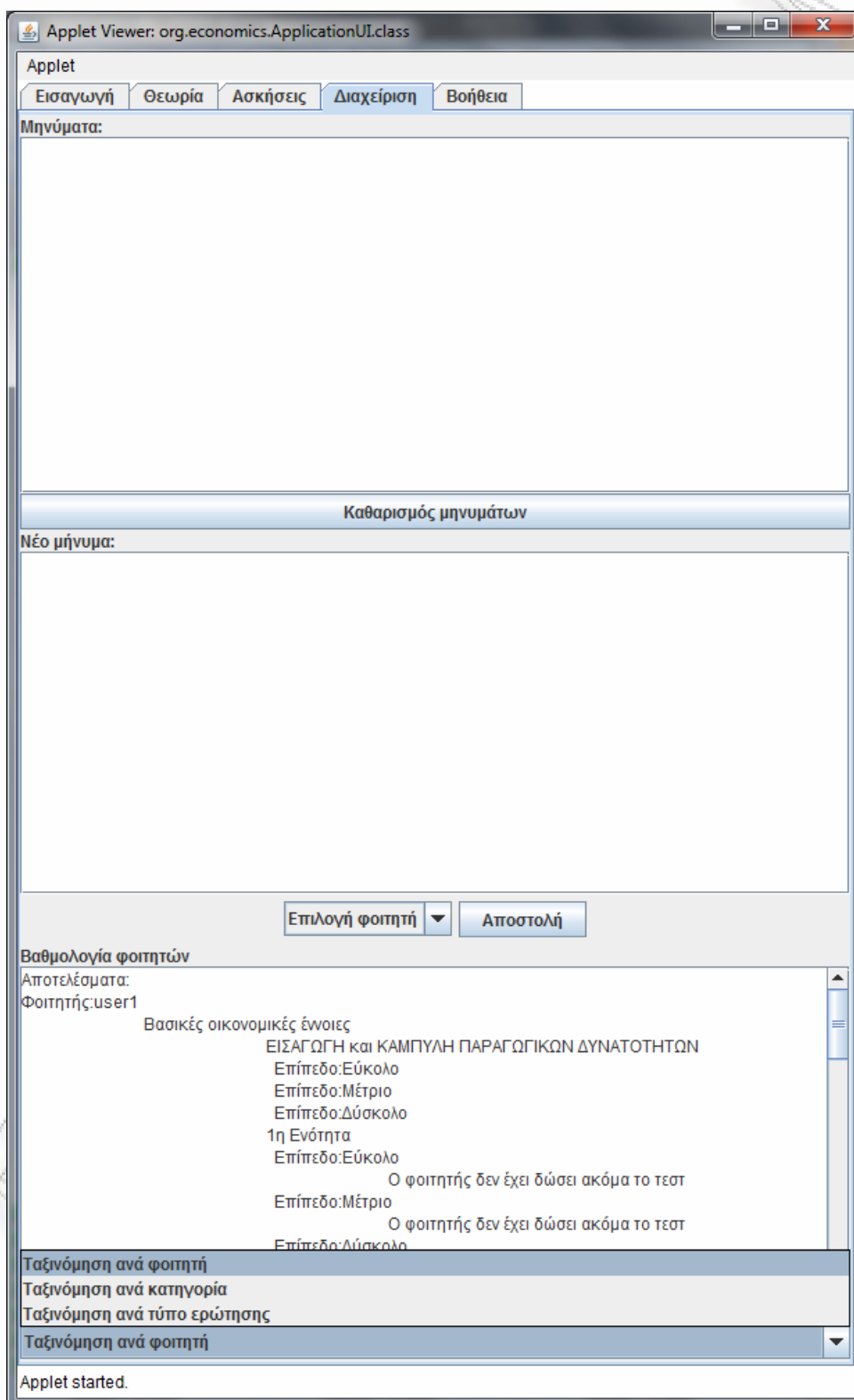
Σχήμα 7: Διεπαφή μηνυμάτων

Διαχείριση

Η διαχείριση είναι τμήμα του λογισμικού που είναι διαθέσιμο μόνο στον διαχειριστή. Μέσα από το Module αυτό, ο διαχειριστής μπορεί να λάβει και να στείλει μηνύματα με τους χρήστες της εφαρμογής.

Επίσης μέσα από τη διαχείριση γίνεται εμφάνιση των αποτελεσμάτων των διαγωνισμάτων που έχουν ολοκληρώσει οι χρήστες. Για την εξαγωγή χρήσιμων στοιχείων και συμπερασμάτων, ο διαχειριστής/καθηγητής μπορεί να κάνει ταξινόμηση των αποτελεσμάτων με τρεις διαφορετικούς τρόπους:

- ταξινόμηση ανά φοιτητή
- ταξινόμηση ανά ενότητα/κεφάλαιο
- ταξινόμηση ανά επίπεδο δυσκολίας

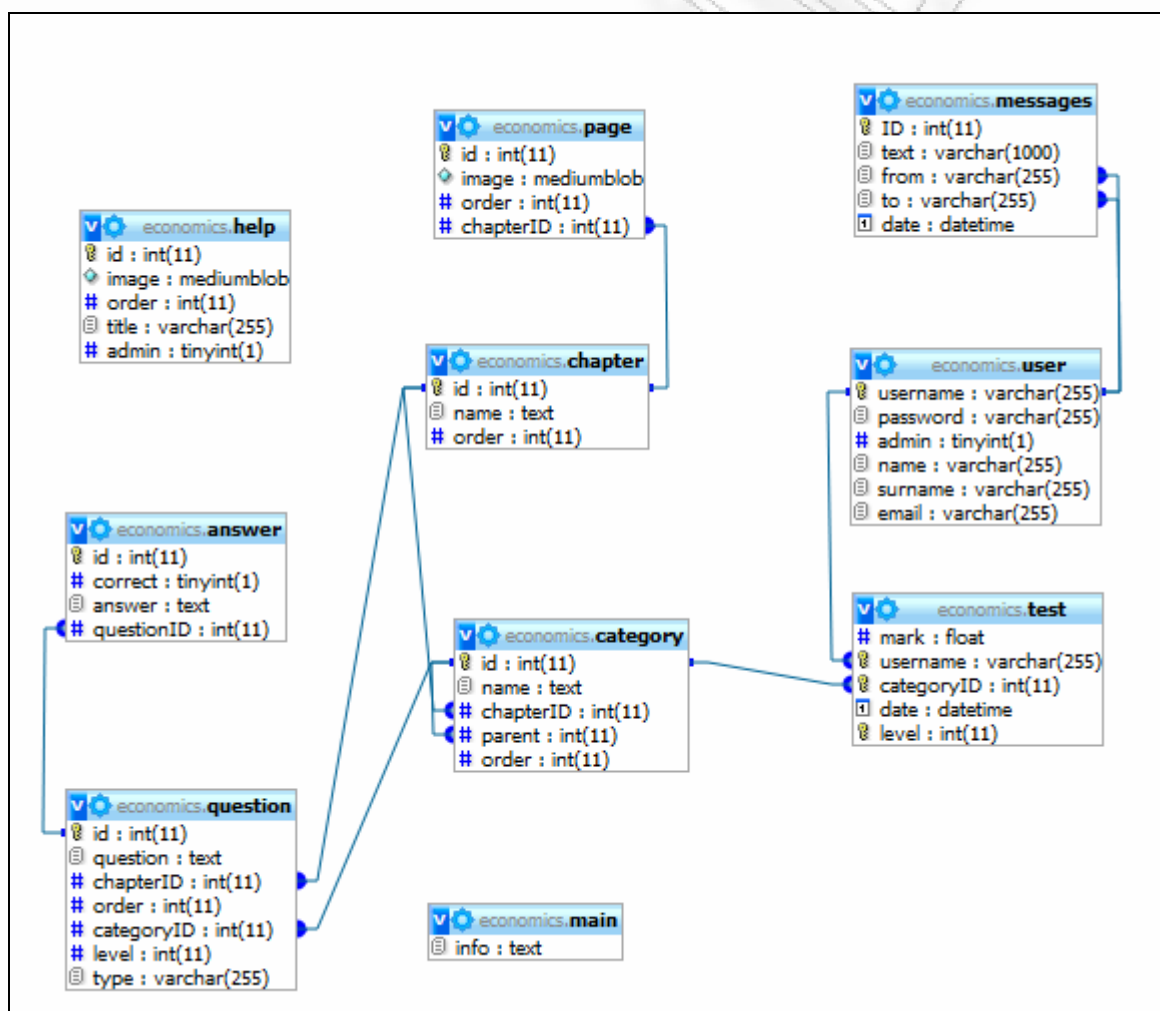


Σχήμα 8: Διεπαφή διαχείρισης

5.3.3 Σχεδιασμός βάσης δεδομένων

Στο παρακάτω σχεδιάγραμμα παρουσιάζεται το σχεσιακό μοντέλο της βάσης δεδομένων του συστήματος. Βασισμένο στις 8 οντότητες που παρουσιάστηκαν στην ενότητα 5.2.1, το εικονιζόμενο μοντέλο περιγράφει τις συσχετίσεις ανάμεσα στις οντότητες. Κάθε συσχέτιση περιγράφεται με μια γραμμή που ενώνει πρωτεύοντα και δευτερεύοντα κλειδιά. Για παράδειγμα, το questioned στο πίνακα answer είναι δευτερεύον κλειδί στο πρωτεύον κλειδί id του πίνακα question, περιγράφοντας το ότι μπορεί να υπάρχουν πολλές απαντήσεις για μια συγκεκριμένη ερώτηση.

Οι πίνακες main και help δεν συσχετίζονται με τους υπόλοιπους και χρησιμοποιούνται για την αποθήκευση κειμένου για το εισαγωγικό κείμενο του συστήματος και για την Online βοήθεια αντίστοιχα.



Σχήμα 9: Διάγραμμα E-R

Ακολουθεί ο κώδικας σε SQL του σχήματος της βάσης. Περιγράφεται αναλυτικά η δομή κάθε πίνακα σε γλώσσα SQL. Ο παρακάτω κώδικας μπορεί να χρησιμοποιηθεί για τη δημιουργία της βάσης (των πινάκων, χωρίς δεδομένα).


```
--  
-- Table structure for table `answer`  
--  
CREATE TABLE IF NOT EXISTS `answer` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `correct` tinyint(1) NOT NULL,  
  `answer` text COLLATE utf8_bin NOT NULL,  
  `questionID` int(11) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_bin AUTO_INCREMENT=13 ;  
  
--  
-- Table structure for table `category`  
--  
CREATE TABLE IF NOT EXISTS `category` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `name` text COLLATE utf8_bin NOT NULL,  
  `chapterID` int(11) NOT NULL,  
  `parent` int(11) NOT NULL,  
  `order` int(11) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_bin AUTO_INCREMENT=11 ;  
  
--  
-- Table structure for table `chapter`  
--  
CREATE TABLE IF NOT EXISTS `chapter` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `name` text COLLATE utf8_bin NOT NULL,  
  `order` int(11) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_bin AUTO_INCREMENT=3 ;  
  
--  
-- Table structure for table `help`  
--  
CREATE TABLE IF NOT EXISTS `help` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `image` mediumblob NOT NULL,  
  `order` int(11) NOT NULL,  
  `title` varchar(255) COLLATE utf8_bin NOT NULL,
```

```
`admin` tinyint(1) NOT NULL DEFAULT '0',
PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_bin AUTO_INCREMENT=3 ;

--
-- Table structure for table `main`
--

CREATE TABLE IF NOT EXISTS `main` (
  `info` text COLLATE utf8_bin NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

--
-- Table structure for table `messages`
--

CREATE TABLE IF NOT EXISTS `messages` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `text` varchar(1000) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL,
  `from` varchar(255) COLLATE utf8_bin NOT NULL,
  `to` varchar(255) COLLATE utf8_bin NOT NULL,
  `date` datetime NOT NULL,
  PRIMARY KEY (`ID`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_bin AUTO_INCREMENT=16 ;

--
-- Table structure for table `page`
--

CREATE TABLE IF NOT EXISTS `page` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `image` mediumblob NOT NULL,
  `order` int(11) NOT NULL,
  `chapterID` int(11) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_bin AUTO_INCREMENT=7 ;

--
-- Table structure for table `question`
--

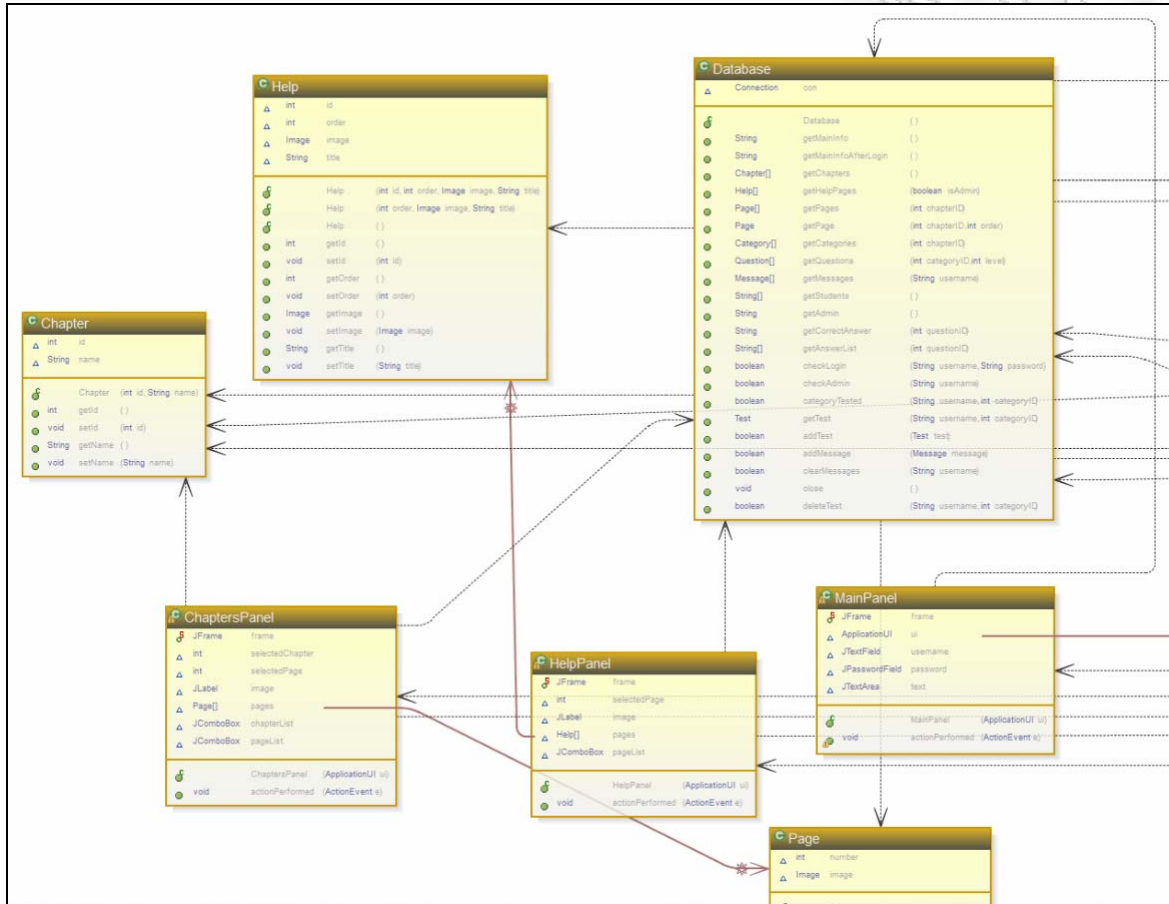
CREATE TABLE IF NOT EXISTS `question` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
```

```
`question` text COLLATE utf8_bin NOT NULL,  
`chapterID` int(11) NOT NULL,  
`order` int(11) NOT NULL,  
`categoryID` int(11) NOT NULL,  
`level` int(11) NOT NULL,  
`type` varchar(255) COLLATE utf8_bin NOT NULL,  
PRIMARY KEY (`id`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_bin AUTO_INCREMENT=7 ;  
  
--  
-- Table structure for table `test`  
--  
  
CREATE TABLE IF NOT EXISTS `test` (  
  `mark` float NOT NULL,  
  `username` varchar(255) COLLATE utf8_bin NOT NULL,  
  `categoryID` int(11) NOT NULL,  
  `date` datetime NOT NULL,  
  `level` int(11) NOT NULL,  
  PRIMARY KEY (`username`,`categoryID`,`level`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_bin;  
  
--  
-- Table structure for table `user`  
--  
  
CREATE TABLE IF NOT EXISTS `user` (  
  `username` varchar(255) COLLATE utf8_bin NOT NULL,  
  `password` varchar(255) COLLATE utf8_bin NOT NULL,  
  `admin` tinyint(1) NOT NULL,  
  `name` varchar(255) COLLATE utf8_bin NOT NULL,  
  `surname` varchar(255) COLLATE utf8_bin NOT NULL,  
  `email` varchar(255) COLLATE utf8_bin NOT NULL,  
  PRIMARY KEY (`username`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
```

5.3.4 UML Διαγράμματα Δομής

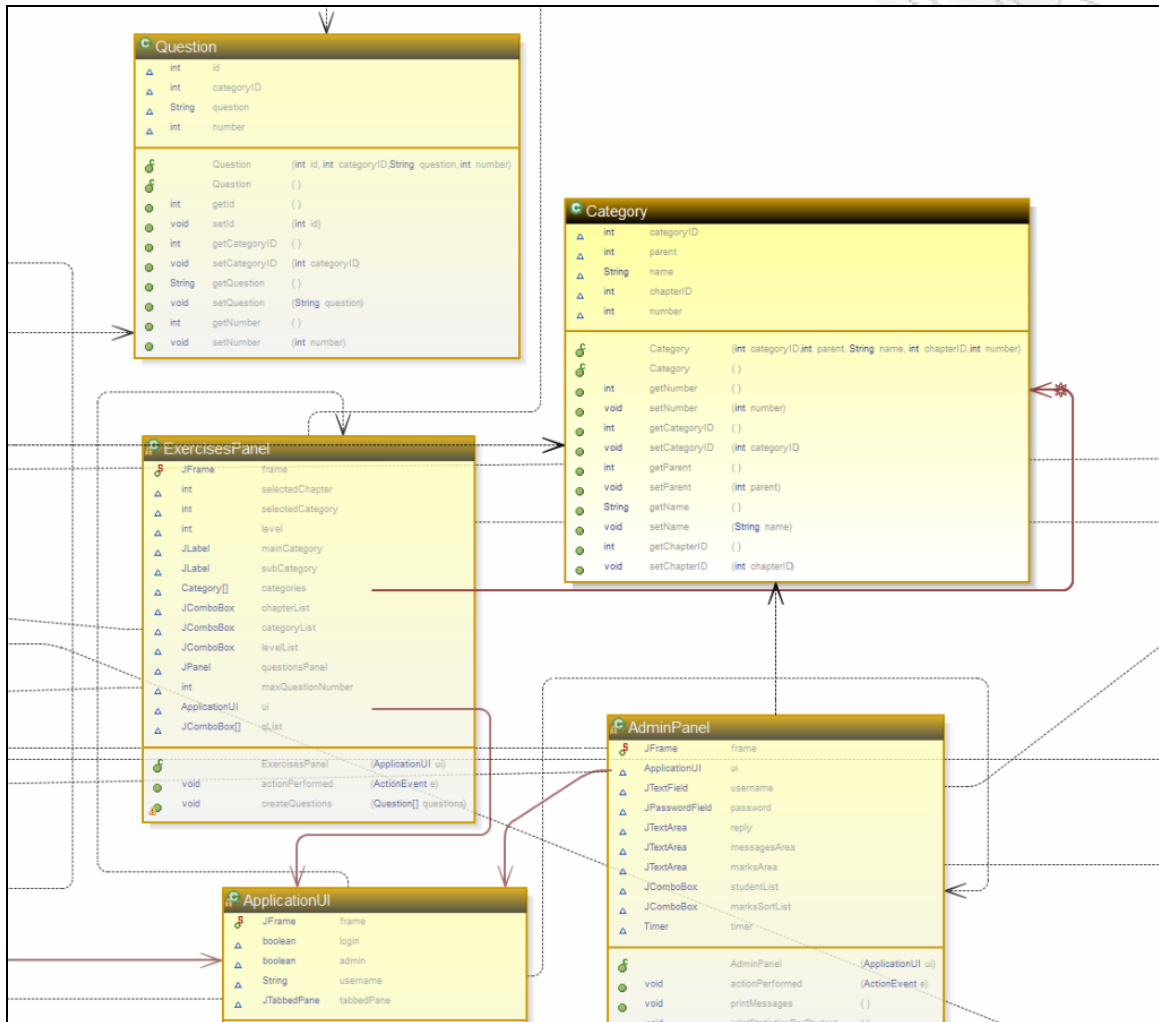
Στην ενότητα αυτή εμφανίζονται τα UML διαγράμματα δομής. Στα διαγράμματα αυτά περιγράφονται όλες οι κλάσεις του λογισμικού, με τις μεθόδους τους. Επίσης, με βέλη φαίνεται η

συσχέτιση ανάμεσα στις κλάσεις αυτές. Ένα βέλος ανάμεσα σε δυο κλάσεις δηλώνουν ότι καλείται η μια μέσα από κάποια μέθοδο της άλλης.

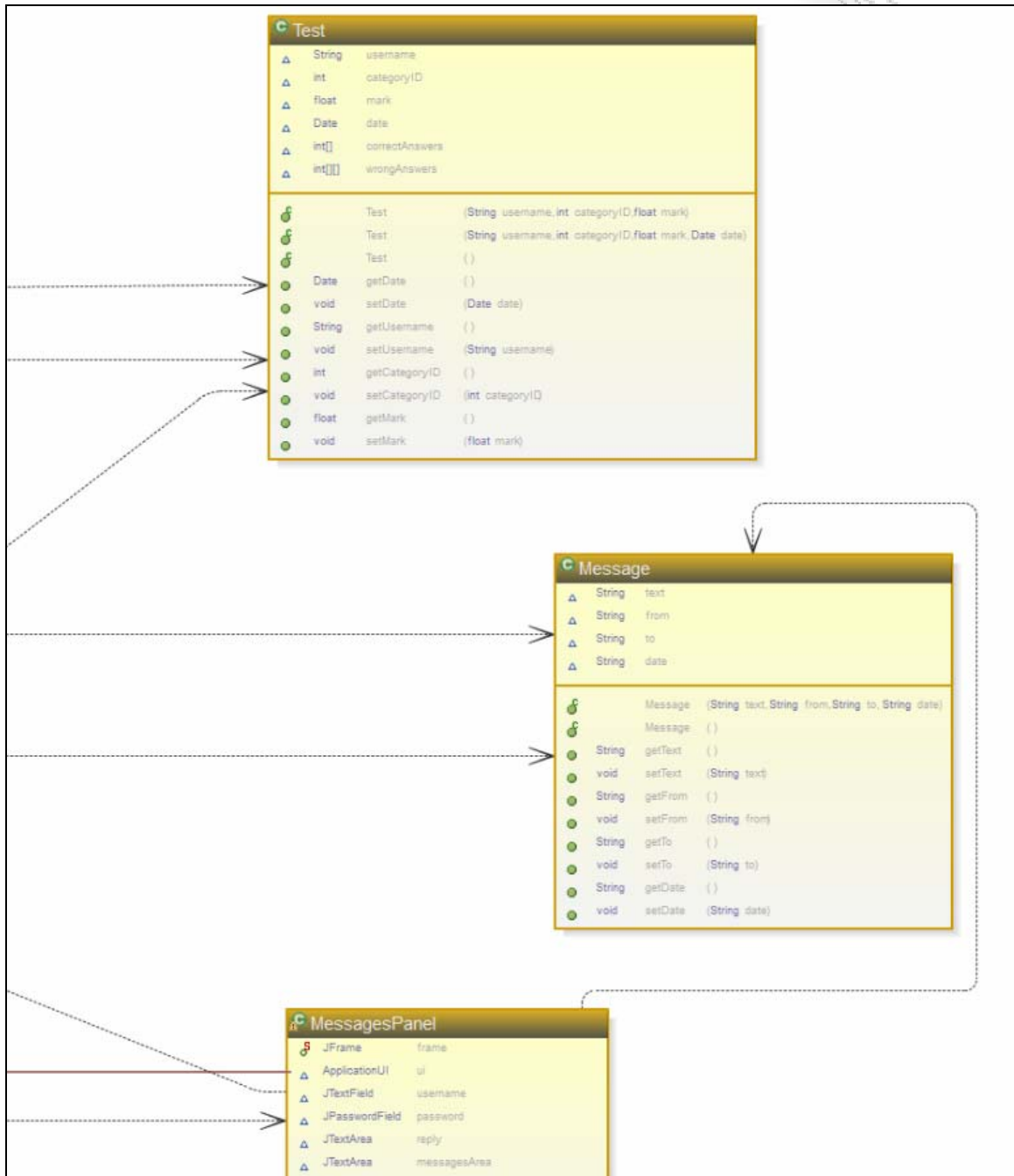


Σχήμα 10: Κλάσεις του λογισμικού και συσχετίσεις





Σχήμα 11: Κλάσεις του λογισμικού και συσχετίσεις



Σχήμα 12: Κλάσεις του λογισμικού και συσχετίσεις

6. ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΗΣ ΣΥΣΤΗΜΑΤΟΣ

Το Σύστημα Διδασκαλίας χρησιμοποιείται για την εξ' αποστάσεως εκπαίδευση μαθητών/φοιτητών σε θέματα οικονομικής θεωρίας. Είναι ένα διαδραστικό σύστημα που θα σας βοηθήσει στη μελέτη του μαθήματος των οικονομικών.

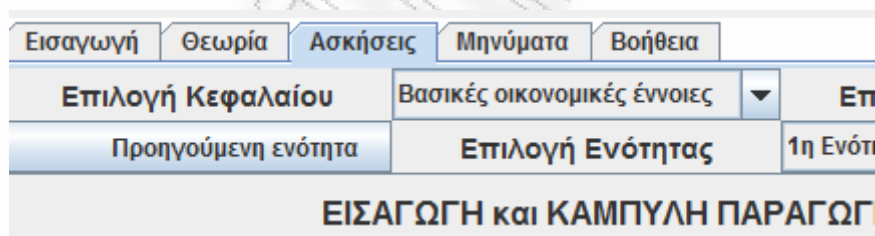
Το Σύστημα αποτελείται από τις παρακάτω βασικές ενότητες:

Θεωρία. Η εφαρμογή εμπεριέχει τη θεωρία του μαθήματος των οικονομικών. Η ύλη που καλύπτεται χωρίζεται σε κεφάλαια. Προκειμένου η θεωρία να είναι ευανάγνωστη η περιήγηση γίνεται σαν σε βιβλίο, ώστε να είναι πιο οικεία στο μαθητή. Η ύλη λοιπόν χωρίζεται σε κεφάλαια και στη συνέχεια σε σελίδες, προσομοιάζοντας ένα βιβλίο.

Ασκήσεις. Προκειμένου να γίνει εμπέδωση της διδαχθείσας ύλης, πρέπει να υπάρχουν online ασκήσεις. Οι ασκήσεις είναι διαφορετικού τύπου και δυσκολίας. Για παράδειγμα μπορεί να έχουμε ερωτήσεις πολλαπλής επιλογής με 2 πιθανές απαντήσεις, αλλά και με περισσότερες. Καθώς οι ασκήσεις αφορούν διαφορετικές ενότητες της θεωρίας και έχουν διαφορετικό τύπο ή δυσκολία, χωρίζονται σε διαφορετικά Online διαγωνίσματα (tests).

Επικοινωνία με το διδάσκοντα. Σε online συστήματα είναι χρήσιμο να υπάρχει δυνατότητα επικοινωνίας με τους υπευθύνους. Για το λόγο αυτό, το Σύστημα Διδασκαλίας παρέχει στοιχεία επικοινωνίας με το διδάσκοντα. Υποστηρίζεται τόσο ασύγχρονη όσο και σύγχρονη επικοινωνία ανάμεσα στα δυο μέρη με την ανταλλαγή μηνυμάτων (Instant Messaging). Όταν κάποιος από τα δυο μέρη δεν είναι Online, το μήνυμα μπορεί να αποθηκεύεται και ο χρήστης να μπορεί να δει το μήνυμα και να απαντήσει αργότερα.

Κάθε ενότητα βρίσκεται σε διαφορετικό παράθυρο της εφαρμογής, ενώ η μετάβαση ανάμεσα στις διαφορετικές ενότητες γίνεται με την επιλογή της αντίστοιχης καρτέλας (tab) όπως φαίνεται στο παρακάτω σχήμα. Στη περίπτωση αυτή έχει επιλεγεί η ενότητα «Ασκήσεις».

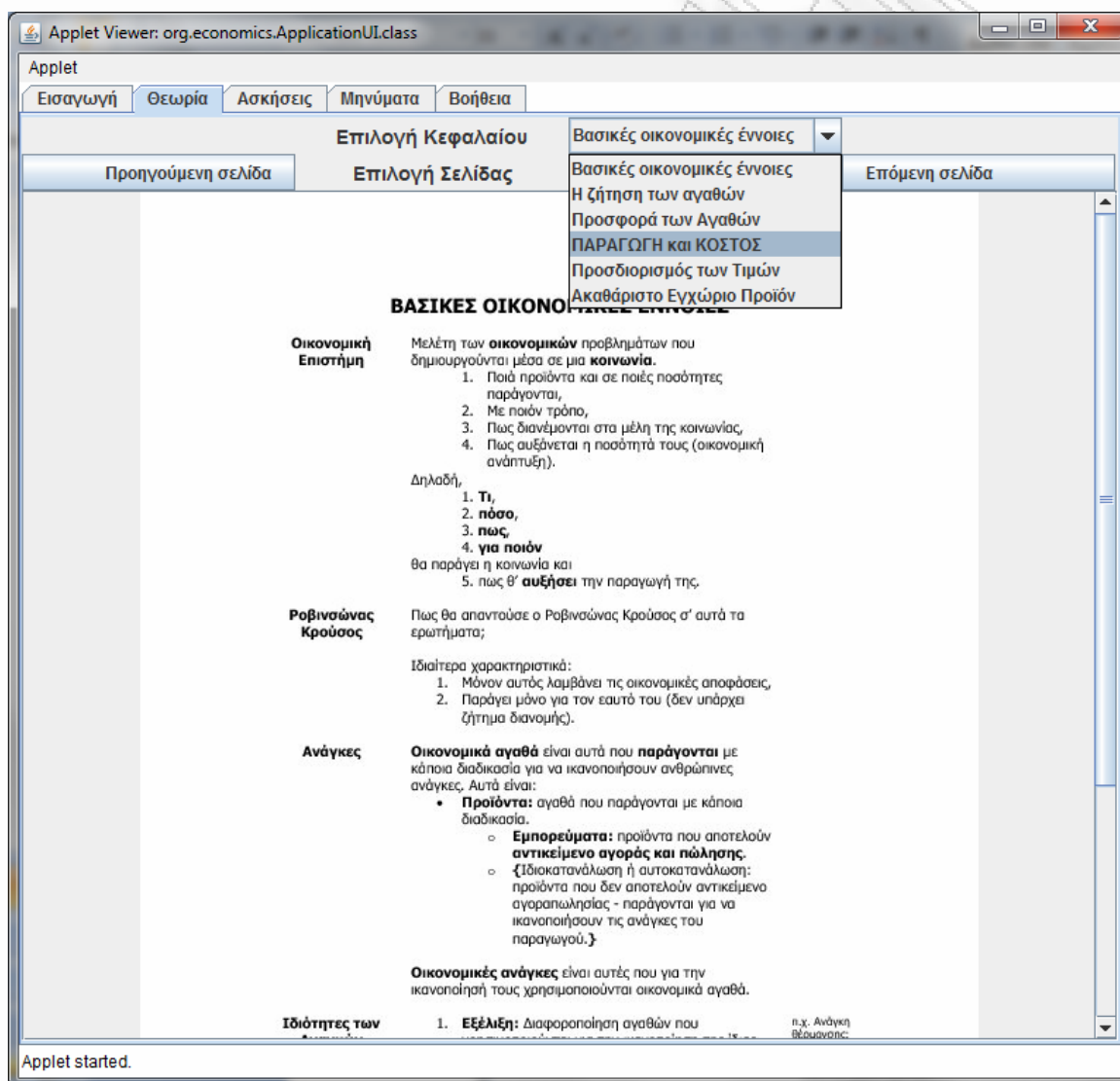


Σχήμα 13: Ενότητες συστήματος

6.1 Εγχειρίδιο Χρήστη του Συστήματος Διδασκαλίας

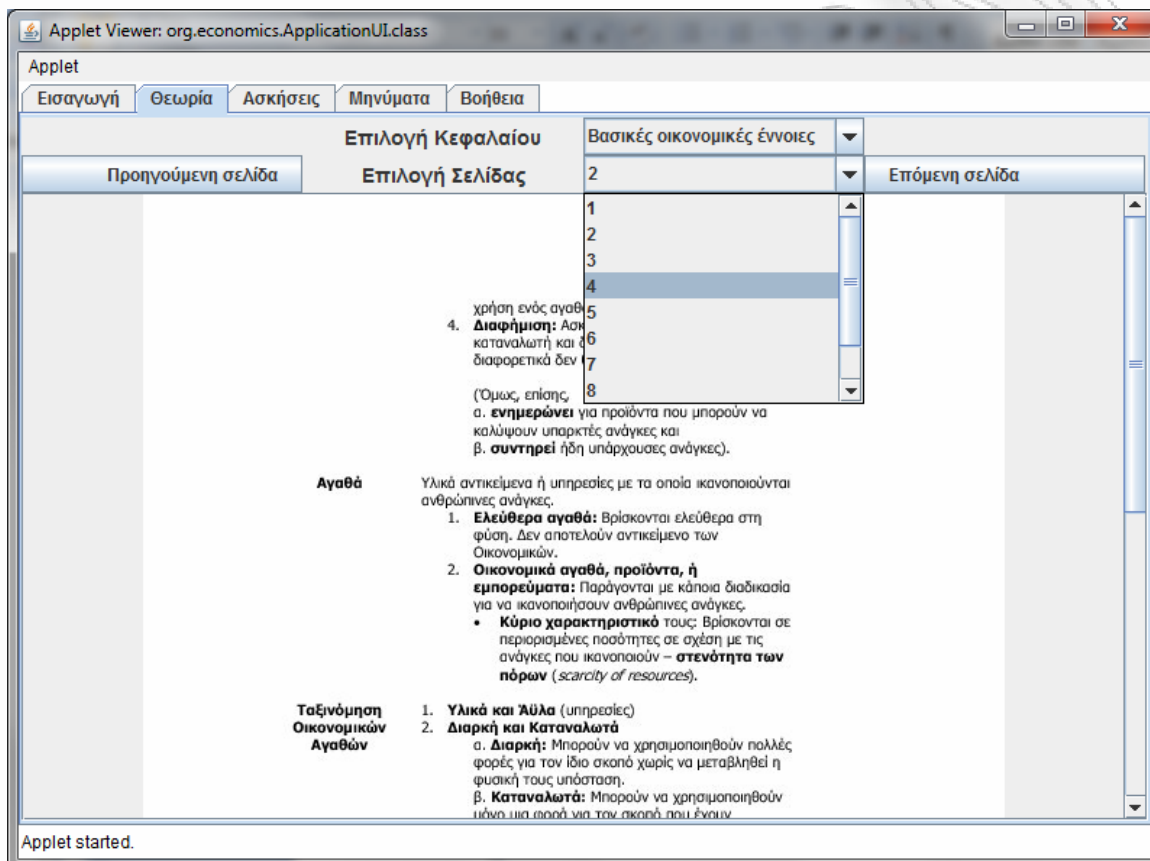
i. Θεωρία

Η περιήγηση στη θεωρία που καλύπτεται στο Σύστημα Διδασκαλίας προσομοιάζει το διάβασμα ενός βιβλίου. Ο χρήστης μπορεί να επιλέξει το κεφάλαιο το οποίο θέλει να διαβάσει επιλέγοντας το τίτλο στο κεντρικό «Drop Down Menu». Στο παρακάτω στιγμιότυπο φαίνεται η επιλογή του κεφαλαίου «ΠΑΡΑΓΩΓΗ και ΚΟΣΤΟΣ».



Σχήμα 14: Οθόνες θεωρίας

Η αλλαγή σελίδας μπορεί να πραγματοποιηθεί με 2 τρόπους, είτε με απευθείας επιλογή του αριθμού της σελίδας στο «Drop Down Menu», είτε σειριακά, επιλέγοντας την αμέσως προηγούμενη ή αμέσως επόμενη σελίδα.



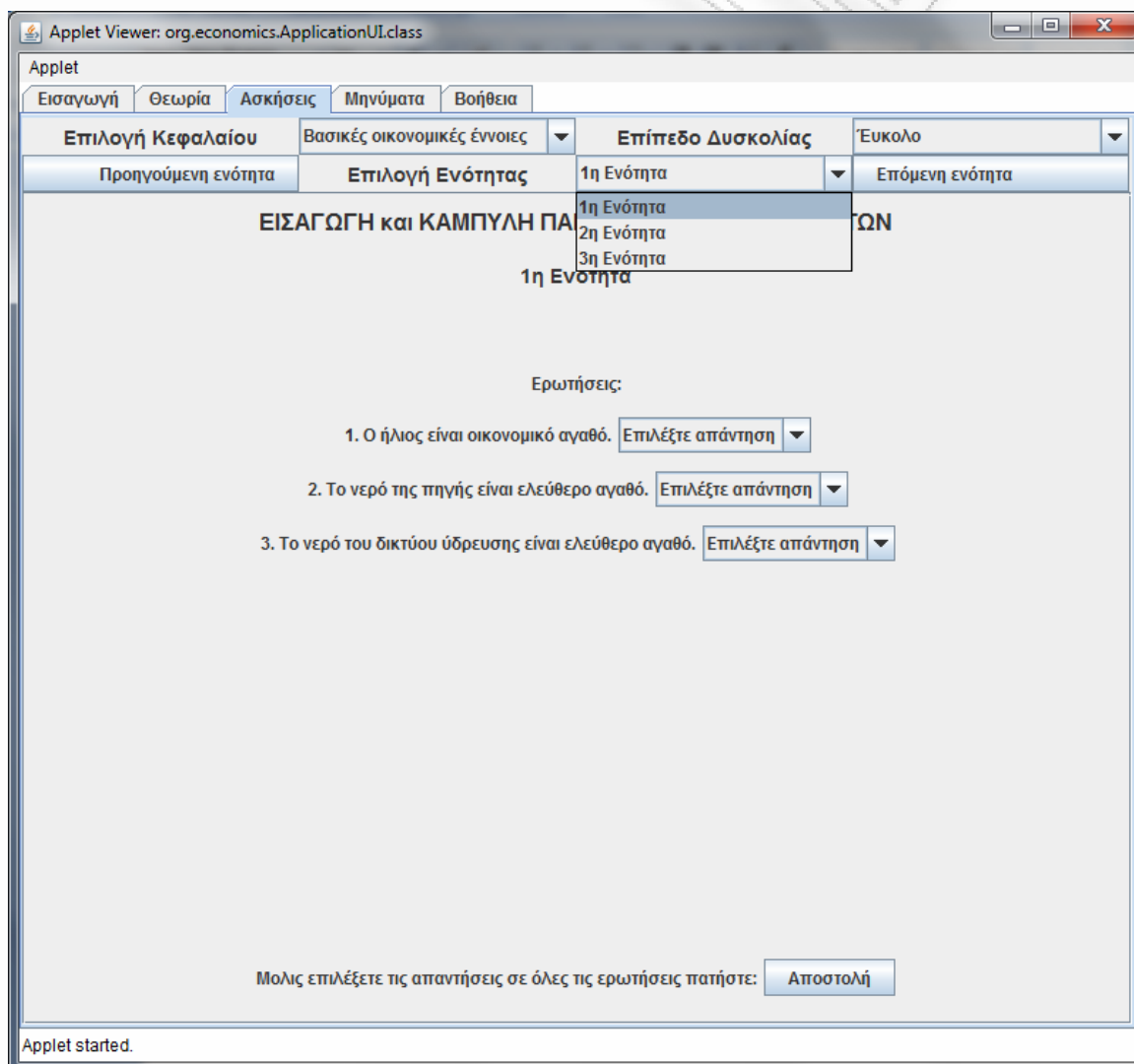
Σχήμα 15: Οθόνες θεωρίας

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΑΣ

ii. Ασκήσεις

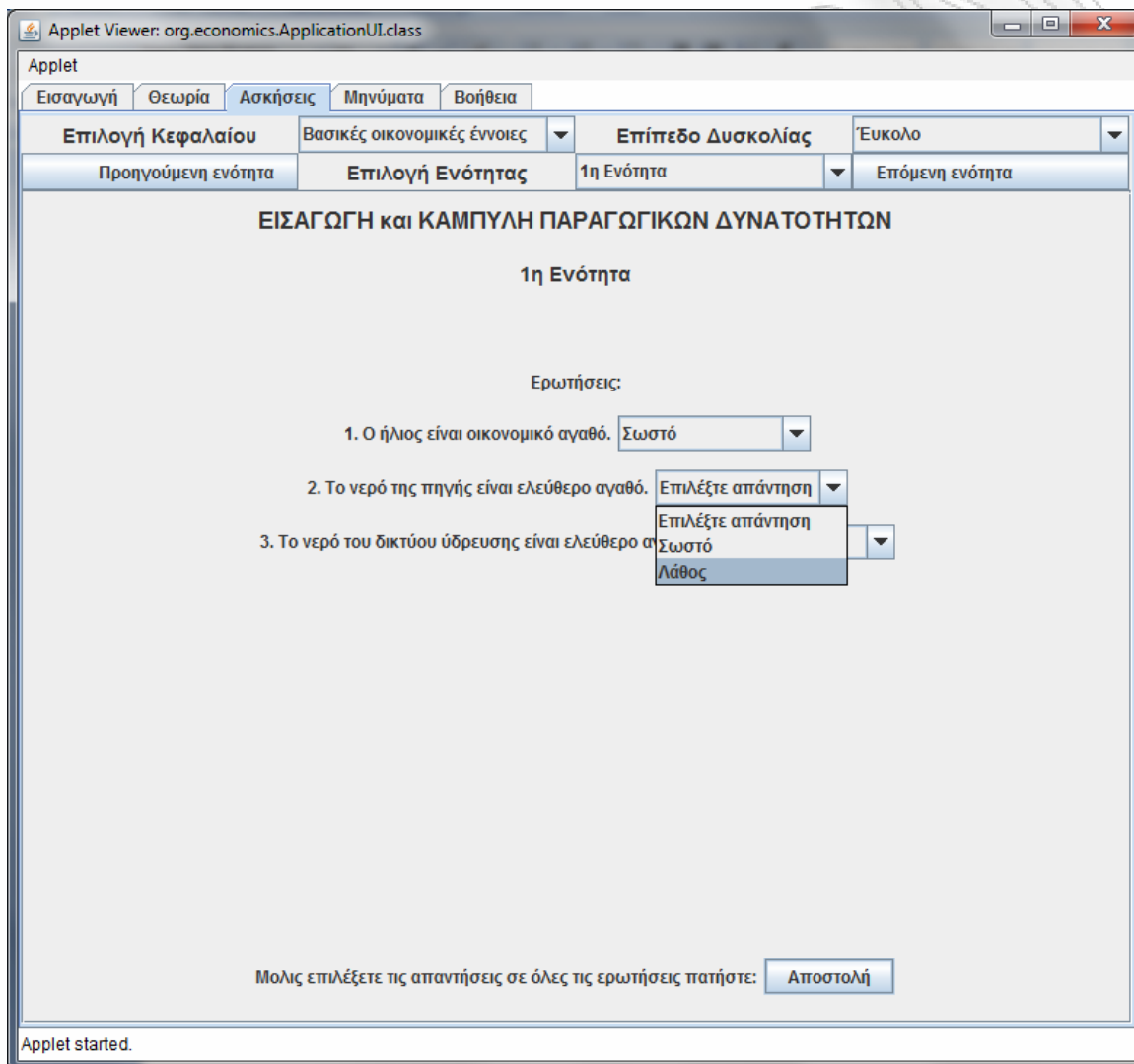
Οι ασκήσεις είναι χωρισμένες σε διαγωνίσματα. Η οντότητα αυτή έχει ως στόχο την αποθήκευση των αποτελεσμάτων ενός τεστ (ενός συνόλου από ερωτήσεις). Η συσχέτιση στο σύνολο από ερωτήσεις γίνεται με τον ορισμό του επιπέδου και της κατηγορίας στην οποία ανήκουν οι ερωτήσεις. Συνεπώς, το διαγώνισμα είναι ένα σύνολο από ερωτήσεις που ανήκουν στην ίδια ενότητα ενός κεφαλαίου και έχουν το ίδιο επίπεδο δυσκολίας.

Στην παρακάτω εικόνα ο χρήστης έχει ήδη επιλέξει το «Ευκολο» Επίπεδο δυσκολίας, στο Κεφάλαιο «Βασικές οικονομικές έννοιες» και είναι έτοιμος να επιλέξει την ενότητα Ασκήσεων με όνομα: «1η Ενότητα». Ο παραπάνω συνδυασμός αποτελεί ένα ξεχωριστό διαγώνισμα.



Σχήμα 16: Οθόνες ασκήσεων

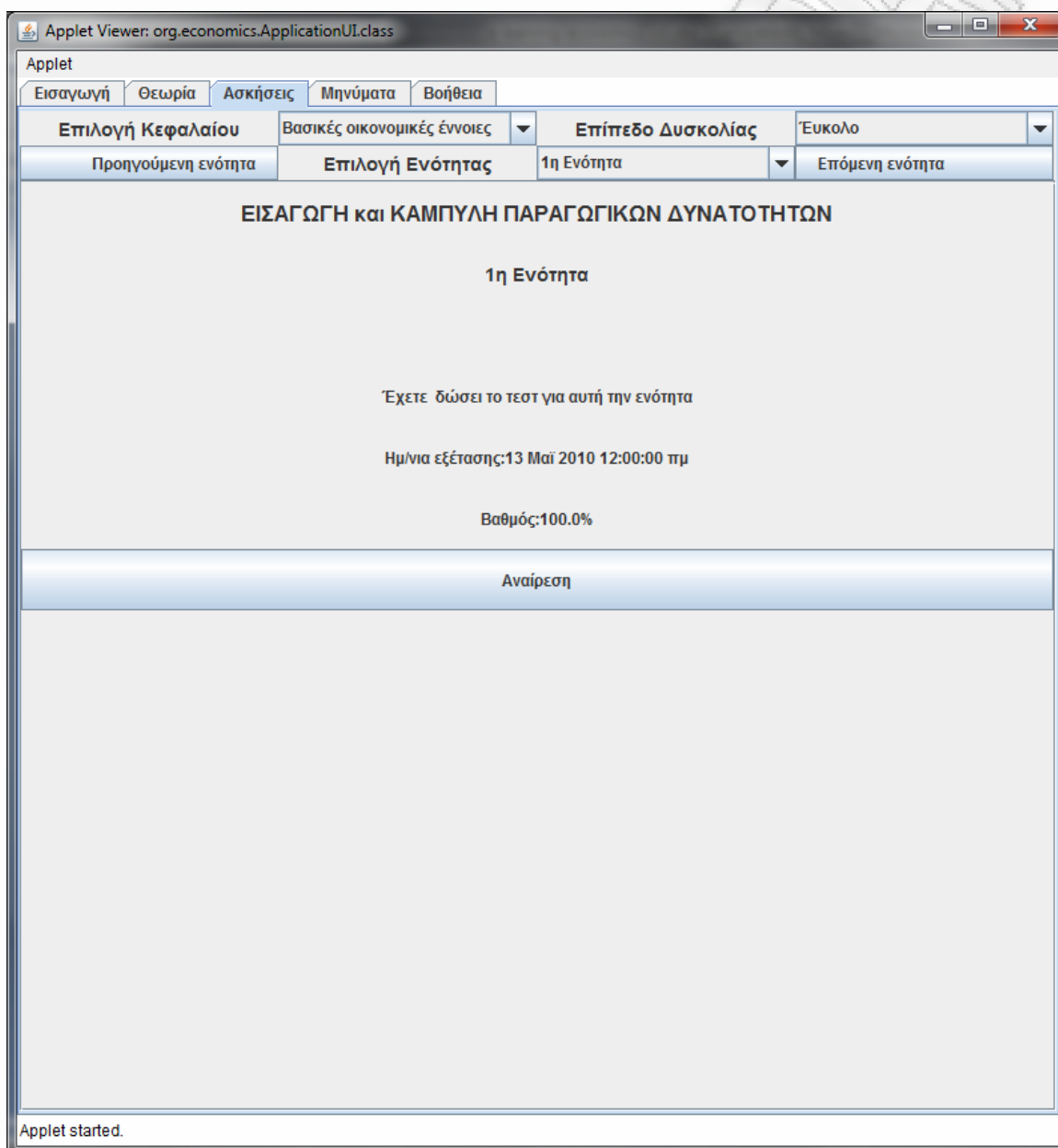
Αφού επιλέξετε το διαγώνισμα το οποίο θέλει να συμπληρώσετε, καλείστε να επιλέξετε τη σωστή απάντηση από τις ερωτήσεις πολλαπλών επιλογών. Αφού συμπληρώσετε όλες τις ερωτήσεις, πατήστε «Αποστολή».



Σχήμα 17: Οθόνες ασκήσεων

Αν ένα διαγώνισμα έχει ήδη συμπληρωθεί, τότε αντί για τις ερωτήσεις πολλαπλών εφαρμογών, εμφανίζεται στην εφαρμογή ο βαθμός που έχετε πάρει, διαιρώντας το πλήθος των σωστών απαντήσεων με το συνολικό πλήθος.

Εμφανίζεται επίσης η ημερομηνία εξέτασης.

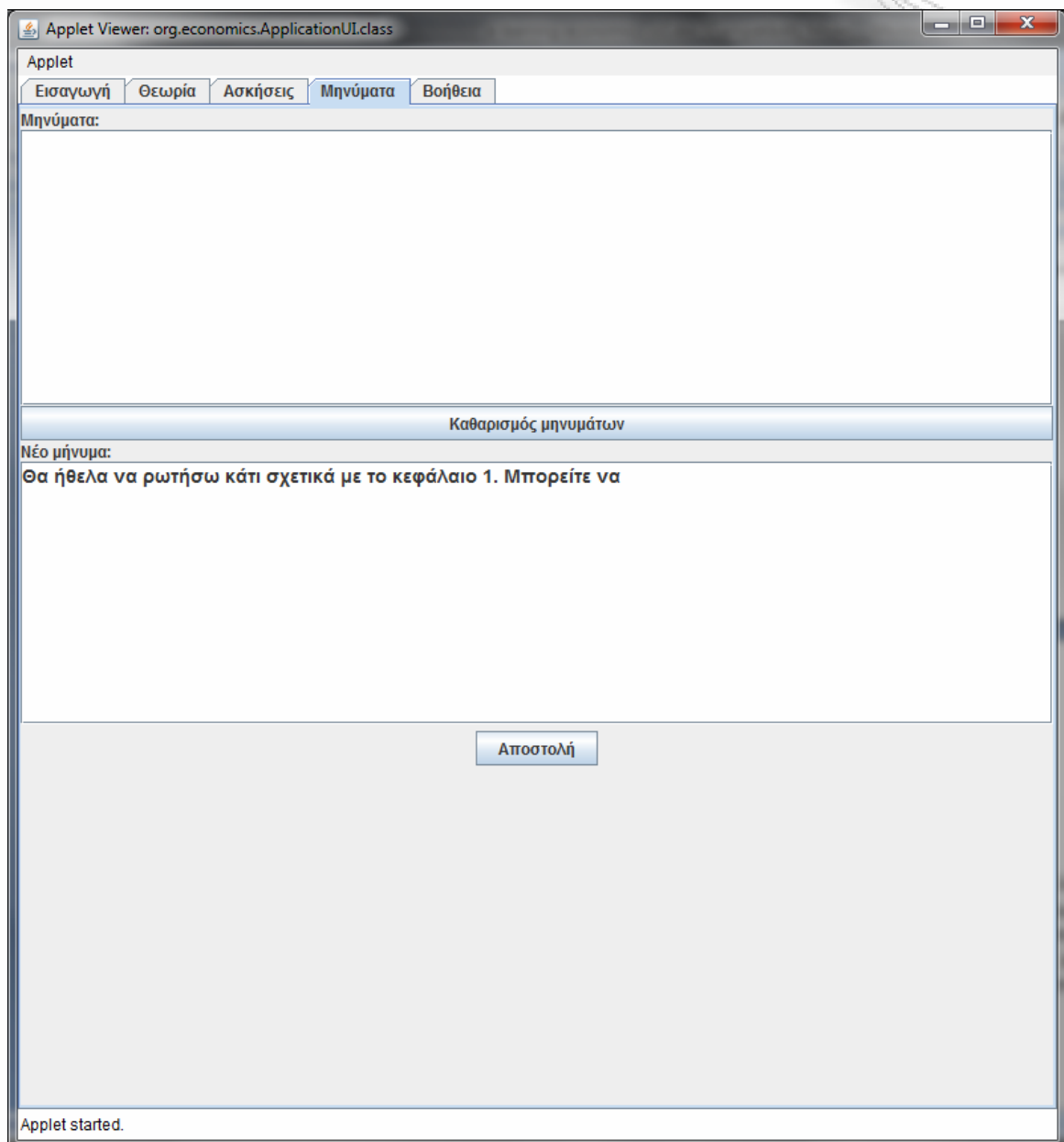


Σχήμα 18: Οθόνες ακήσεων

Τέλος, έχετε τη δυνατότητα αναίρεσης του τεστ και διαγραφής των αποτελεσμάτων, έτσι ώστε να μπορείτε να το ξαναδοκιμάσετε. Αυτό γίνεται επιλέγοντας το πλήκτρο «Αναίρεση».

iii. Μηνύματα

Έχετε τη δυνατότητα ανταλλαγής μηνυμάτων με τον διδάσκοντα κατά τη διάρκεια της χρήσης του Συστήματος Διδασκαλίας. Αυτό είναι εφικτό από την ενότητα «Μηνύματα». Στην κονσόλα στο πάνω μέρος του παραθύρου, εμφανίζονται τα προηγούμενα μηνύματα που έχετε λάβει από τον καθηγητή. Παράλληλα, στο πεδίο στο κάτω μέρος του παραθύρου, μπορείτε να γράψετε ένα νέο μήνυμα. Η εφαρμογή φροντίζει για την αποστολή και λήψη μηνυμάτων σε πραγματικό χρόνο, λειτουργώντας σαν ένα σύστημα Instant Messaging.



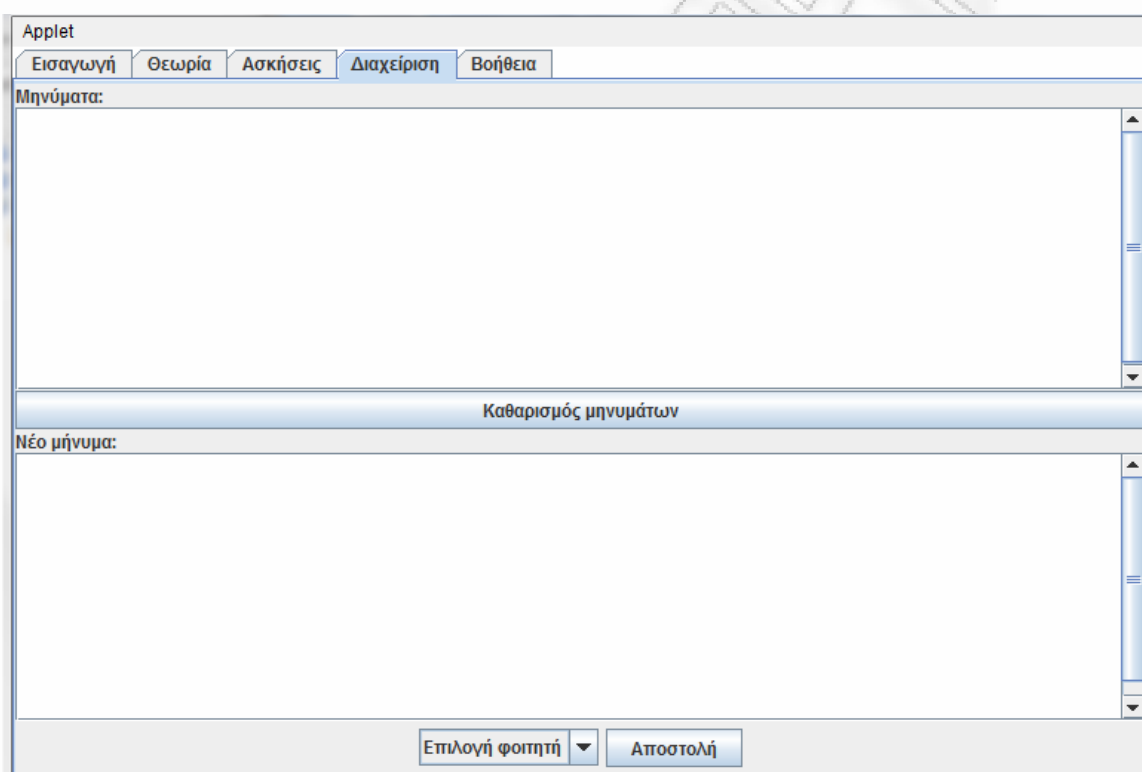
Σχήμα 19: Οθόνες Μηνυμάτων

Τέλος, επιλέγοντας το πλήκτρο «Καθαρισμός Μηνυμάτων» τα προηγούμενα μηνύματα που έχετε λάβει διαγράφονται από το αρχείο και δεν είναι πλέον διαθέσιμα.

6.2 Εγχειρίδιο Διαχειριστή

i. Διαχείριση

Στο διαχειριστή του συστήματος διατίθεται μια ακόμα ενότητα με όνομα «Διαχείριση». Από την ενότητα αυτή μπορείτε να επικοινωνήσετε με τους μαθητές μέσω της ανταλλαγής μηνυμάτων σε πραγματικό χρόνο. Επίσης, στο παράθυρο αυτό παρέχεται η δυνατότητα προβολής των αποτελεσμάτων των διαγωνισμάτων τα οποία έχουν δώσει οι χρήστες του Συστήματος Διδασκαλίας.

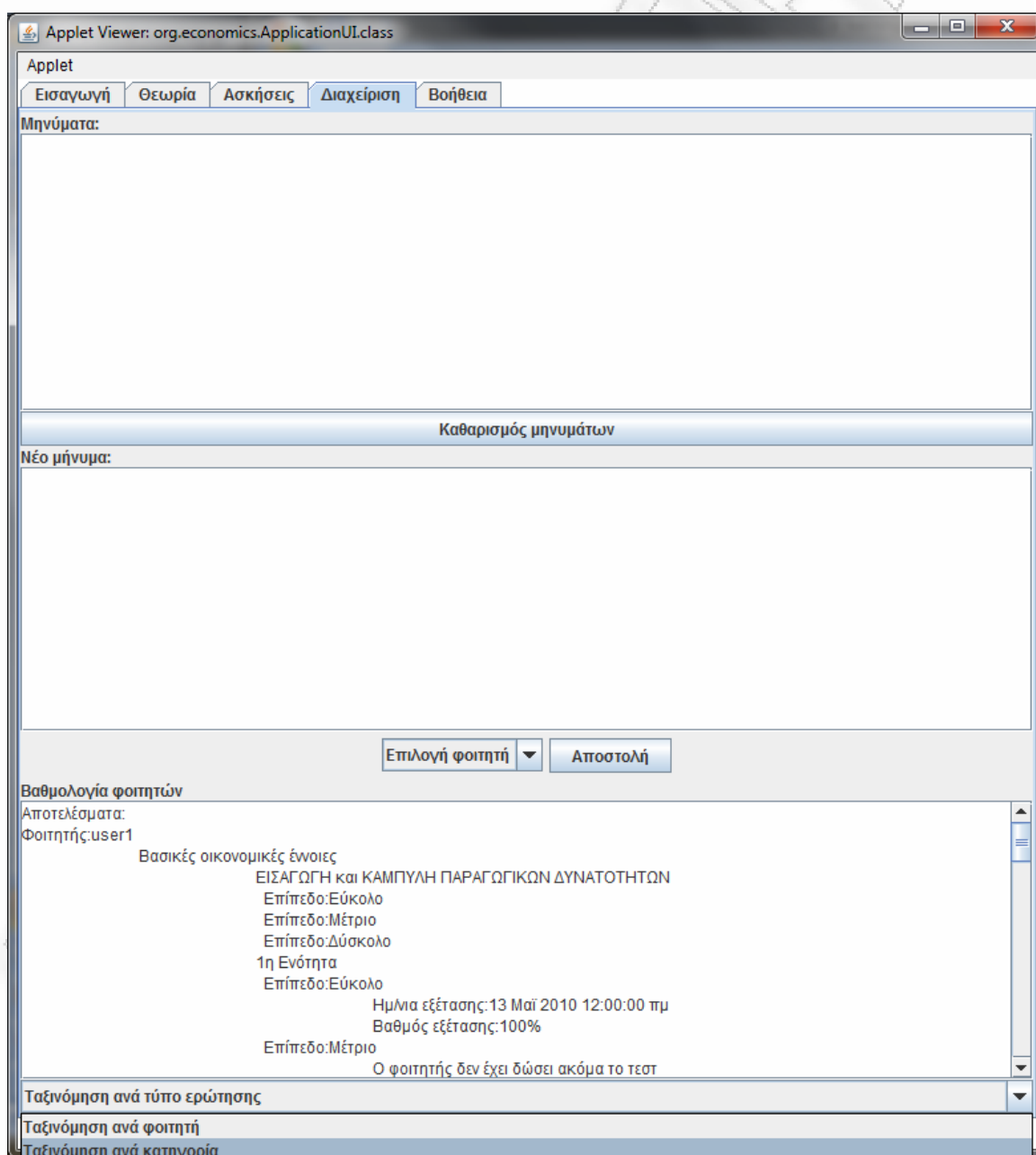


Σχήμα 20: Οθόνες Διαχείρισης

Για την αποστολή σε κάποιον φοιτητή, επιλέξτε από το «Drop Down Menu» το όνομα του φοιτητή και στη συνέχεια πατήστε «Αποστολή».

Για την προβολή των αποτελεσμάτων από τα διαγωνίσματα, υπάρχουν τρεις διαφορετικές επιλογές απεικόνισης:

- Ταξινόμηση ανά φοιτητή: Τυπώνει για κάθε φοιτητή τα επιμέρους αποτελέσματα από τα διαγωνίσματα που έχει συμπληρώσει
- Ταξινόμηση ανά κατηγορία: Τυπώνει ανά κατηγορία τα αποτελέσματα όλων των φοιτητών από τα διαγωνίσματα που έχουν συμπληρώσει
- Ταξινόμηση ανά τύπο ερώτησης: Τυπώνει τα αποτελέσματα ανά επίπεδο δυσκολίας.



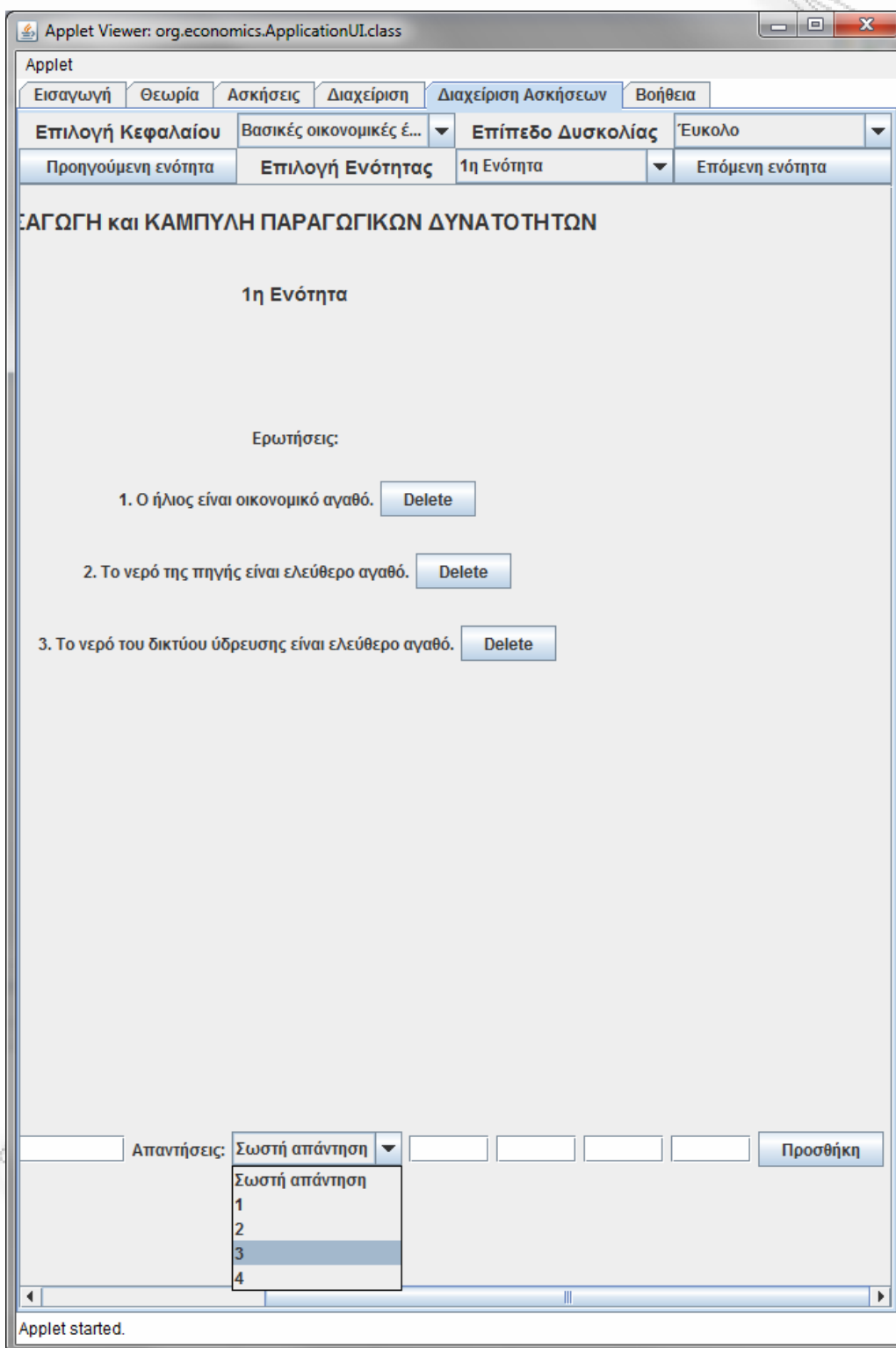
Σχήμα 21: Οθόνες Διαχείρισης

ii. Διαχείριση ερωτήσεων/απαντήσεων

Στο διαχειριστή του συστήματος διατίθεται μια ακόμα ενότητα με όνομα «Διαχείριση Ασκήσεων». Από την ενότητα αυτή μπορείτε να προσθαφαιρέσετε ερωτήσεις και τις αντίστοιχες απαντήσεις στο Σύστημα Διδασκαλίας. Πιο συγκεκριμένα, η ενότητα αυτή ακολουθεί τη δομή της ενότητας «Ασκήσεις», εμφανίζονται τα κεφάλαια και οι ενότητές του.

Η διαγραφή των ερωτήσεων και των απαντήσεων που αντιστοιχούν σε αυτές γίνεται επιλέγοντας το πλήκτρο Delete. Πατώντας το πλήκτρο, διαγράφεται η αντίστοιχη ερώτηση, και όλες απαντήσεις την αφορούν.

Για την προσθήκη νέας ερώτησης και απαντήσεων συμπληρώνετε την ερώτηση στο πεδίο «Νέα ερώτηση», στη συνέχεια συμπληρώνεται όσες από τις απαντήσεις χρειάζεστε στα επόμενα τέσσερα πεδία. Πρέπει να συμπληρώσετε τουλάχιστον 2 πεδία, δίνοντας έτσι 2 πιθανές απαντήσεις. Δεν είναι όμως υποχρεωτική η συμπλήρωση και των τεσσάρων πεδίων που αφορούν τις απαντήσεις. Η επιλογή της σωστής ερώτησης γίνεται από τη «Drop Down» λίστα, επιλέγοντας ένα αριθμό από το 1 ως το 4, ο οποίος αντιστοιχεί στη σωστή απάντηση.



Σχήμα 22: Οθόνες Διαχείρισης

7. ΕΠΙΛΟΓΟΣ

Μετά την ανάλυση του σχεδιασμού και της ανάπτυξης του υλικού για το Σύστημα Διδασκαλίας, κρίνεται απαραίτητο να παρατεθούν κάποια γενικά συμπεράσματα σχετικά με την ικανοποίηση των αρχικών προδιαγραφών και στόχων.

Το σύστημα απευθύνεται σε φοιτητές και στοχεύει στην παρουσίαση εισαγωγικών εννοιών σχετικών με τη βασική θεωρία οικονομικών. Σκοπός του είναι να εισαχθεί ο φοιτητής σε βασικές έννοιες της, όπως για παράδειγμα η προσφορά και ζήτηση αγαθών, το Εγχώριο Ακαθάριστο Προϊόν και άλλα.

Η σε βάθος κατανόηση των εννοιών επιτυγχάνεται με τη διάθεση ασκήσεων, οι οποίες όχι μόνο είναι χωρισμένες σε διαφορετικές ενότητες, σε αναλογία με τα κεφάλαια της θεωρίας, αλλά έχουν και διαφορετικά επίπεδα δυσκολίας. Το βασικό πλεονέκτημα του Συστήματος Διδασκαλίας είναι ότι προσαρμόζει την παρουσίαση του εκπαιδευτικού υλικού στις διαφορετικές ανάγκες των χρηστών. Τα διαφορετικά επίπεδα ασκήσεων προσφέρουν στο σύστημα προσαρμοστικότητα στις ανάγκες κάθε φοιτητή. Η προσαρμοστικότητά του αυτή βοηθά στην αύξηση της απόδοσης των χρηστών, καθώς γίνεται καλύτερη αφομοίωση των εννοιών.

Παράλληλα, ένα σημαντικό στοιχείο στην εξ'αποστάσεως εκπαίδευση είναι η επικοινωνία με τον διδάσκοντα. Ο περιορισμός αυτός επιτυγχάνεται με τη χρήση υποσυστήματος για την ανταλλαγή μηνυμάτων σε πραγματικό χρόνο. Το ίδιο σύστημα μπορεί να λειτουργήσει και ασύγχρονα, όταν δεν είναι δυνατή η άμεση επικοινωνία, καθώς τα μηνύματα αποθηκεύονται στο κεντρικό εξυπηρετητή και είναι διαθέσιμα μόλις συνδεθεί ο χρήστης στον οποίο απευθύνονται.

Τέλος πρέπει να παρατηρήσουμε ότι το σύστημα είναι βασισμένο στον Παγκόσμιο Ιστό, η οικουμενικότητά του οποίου και η ευκολία πρόσβασης θα δώσει κίνητρο σε αρκετούς χρήστες να το χρησιμοποιήσουν. Η σχεδίαση έγινε με βάση την ευκολία στη χρήση, και με γνώμονα τη μη αναγκαιότητα για εξειδικευμένο λογισμικό.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] «Αλληλεπίδραση Ανθρώπου Υπολογιστή», Σπύρος Συρμακέσης, ΕΛΛΗΝΙΚΑ ΓΡΑΜΜΑΤΑ-Ε.Α.Ι.Τ.Υ. cti press, 2003
- [2] Deitel Harvey M., Deitel Paul J., «Java προγραμματισμός», Εκδόσεις Μ. Γιούρδας, 2005
- [3] Bachman, K. Corporate E-learning : Exploring a new frontier.
- [4] Dumme Audrey, B. T. Beyond Knowledge Management: Introducing Learning Management Systems.
- [5] Managing open Systems (Open & Distance Learning).
- [6] Patricia McGee, C. C. Course Management Systems for Learning: Beyond Accidental Pedagogy.
- [7] Researcher, G. D. (May 2004). An Introduction to e-learning.
- [8] W3schools online web tutorials: www.w3schools.com
- [9] Php – MySQL Manual: <http://gr2.php.net/mysql>
- [10] Intelligent tutoring system: http://en.wikipedia.org/wiki/Intelligent_tutoring_system
- [11] John Psarras, Education and training in the knowledge-based economy, pp 85-95.
- [12] Kimberly C. Harper, Kuanchin Chen, David C. Yen , Distance learning, virtual classrooms, and teaching pedagogy in the Internet environment, Technology in Society, Vol. 26 (2004), pp. 585–598.
- [13] Shu-Sheng Liaw Hsiu-Mei Huang , Gwo-Dong Chen, An activity-theoretical approach to investigate learners' factors toward e-learning systems, Computers in Human Behavior (2006) .
- [14] Martin Gotthardt, Maria J. Siegert, Anja Schlieck, Stefan Schneider, Alfred Kohnert, Markus W. Gro_Christine Schäfer, Richard Wagner, Stefan Hörmann, Thomas M. Behr, Rita Engenhardt-Cabillic, Klaus J. Klose Hartmut Jungclas, Ulrich Glowalla, How to Successfully Implement E-learning for both Students and Teachers, Acad Radiol; Vol. 13, 2006, pp. 379–390.
- [15] Chenn-Jung Huang , San-Shine Chu, Chih-Tai Guan, Implementation and performance evaluation of parameter improvement mechanisms for intelligent e-learning systems, Computers & Education (2005).
- [16] Gabriela Kennedy, Lovells, Hong Kong, INTELLECTUAL PROPERTY ISSUES IN E-LEARNING, Computer Law & Security Report Vol. 18 no. 2 2002.

ΠΑΡΑΡΤΗΜΑ – ΚΩΔΙΚΑΣ

```
ApplicationUI Class
package org.economics;

import java.applet.Applet;
import java.awt.GridLayout;
import java.awt.event.KeyEvent;
import java.util.HashMap;

import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JSpinner;
import javax.swing.JTabbedPane;
import javax.swing.SwingUtilities;
import javax.swing.UIManager;

/**
 * Η βασική κλάση από την οποία αρχικοποιείται το πρόγραμμα.
 * Περιέχει και την συνάρτηση main.
 */
public class ApplicationUI extends Applet {

    public static JFrame frame = null;
    boolean login=false;
    boolean admin=false;
    String username;
    JTabbedPane tabbedPane;

    public ApplicationUI(){
        // Το παράθυρο θα έχει διάφορα tabs.

        this.setLayout(new GridLayout(1, 1));
        // Φτιάχνουμε το tabbed περιβάλλον και προσθέτουμε το πρώτο
tab
        // που θα είναι panel τύπου MainPanel
        tabbedPane = new JTabbedPane();
        tabbedPane.addTab("Εισαγωγή", new MainPanel(this));
        // κουμπί συντόμευσης - με Alt-1 θα πηγαίνει σε αυτό το tab
        tabbedPane.setMnemonicAt(0, KeyEvent.VK_1);

        //Add the tabbed pane to this panel.
        add(tabbedPane);

        //The following line enables to use scrolling tabs.
        tabbedPane.setTabLayoutPolicy(JTabbedPane.SCROLL_TAB_LAYOUT);
    }
}
```

```

        tabbedPane.setFont(tabbedPane.getFont().deriveFont((float)
12.0));

    }

    public void deleteTabs(){
        tabbedPane.remove(1);
        tabbedPane.remove(1);
        tabbedPane.remove(1);
        tabbedPane.remove(1);
    }

    public void addTabs(){

        // προσθέτουμε το δεύτερο tab που θα είναι panel τύπου
CategoryCoursePanel
        tabbedPane.addTab("Θεωρία", new ChaptersPanel(this) );
        // κουμπί συντόμευσης - με Alt-2 θα πηγαίνει σε αυτό το tab
        tabbedPane.setMnemonicAt(1, KeyEvent.VK_2);

        tabbedPane.addTab("Ασκήσεις", new ExercisesPanel(this));
        // κουμπί συντόμευσης - με Alt-2 θα πηγαίνει σε αυτό το tab
        tabbedPane.setMnemonicAt(1, KeyEvent.VK_3);

        if(isAdmin()){
            tabbedPane.addTab("Διαχείριση", new AdminPanel(this) );
            // κουμπί συντόμευσης - με Alt-2 θα πηγαίνει σε αυτό το
tab
            tabbedPane.setMnemonicAt(1, KeyEvent.VK_4);

            tabbedPane.addTab("Διαχείριση Ασκήσεων", new
AdminExercisesPanel(this) );

        }
        else{
            tabbedPane.addTab("Μηνύματα", new MessagesPanel(this)
);
            // κουμπί συντόμευσης - με Alt-2 θα πηγαίνει σε αυτό το
tab
            tabbedPane.setMnemonicAt(1, KeyEvent.VK_4);
        }

        tabbedPane.addTab("Βοήθεια", new HelpPanel(this) );
        // κουμπί συντόμευσης - με Alt-2 θα πηγαίνει σε αυτό το tab
        tabbedPane.setMnemonicAt(1, KeyEvent.VK_5);

    }

    /**
     * Create the GUI and show it. For thread safety,
     * this method should be invoked from the
     * event dispatch thread.

```

```
    * Required only if run as application
    */
private static void createAndShowGUI() {
    //Create and set up the window.
    frame = new JFrame("Σύστημα Διδασκαλίας Οικονομικών");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    //Add content to the window.
    frame.add(new ApplicationUI());

    //Display the window.
    frame.setVisible(true);
    frame.setSize(650,870 );
}

public boolean isLogin() {
    return login;
}
public void setLogin(boolean login) {
    this.login = login;
}

public boolean isAdmin() {
    return admin;
}
public void setAdmin(boolean admin) {
    this.admin = admin;
}

public String getUsername() {
    return username;
}
public void setUsername(String username) {
    this.username = username;
}

public void init(){
    new ApplicationUI();
}
}
```

```
package org.economics;

import java.awt.BorderLayout;
import java.awt.Color;
```

```
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.StringTokenizer;

import javax.swing.JButton;
import javax.swing.JComboBox;
AdminExercisesPanel

import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.SwingConstants;

/**
 * Η κλάση που αρχικοποιεί το πρώτο tab με την εισαγωγή
 *
 */
public class AdminExercisesPanel extends JPanel implements
ActionListener {
    public static JFrame frame = null;

    int selectedChapter =1;
    int selectedCategory = 1;
    int level = 0;
    JLabel mainCategory;
    JLabel subCategory;
    Category[] categories;
    JComboBox chapterList;
    JComboBox categoryList ;
    JComboBox levelList;
    JPanel questionsPanel;
    int maxQuestionNumber=10;
    ApplicationUI ui;
    JComboBox[] qList;
    JComboBox aList;
    JTextField questionField;
    JTextField answerField1;
    JTextField answerField2;
    JTextField answerField3;
    JTextField answerField4;

    public AdminExercisesPanel(ApplicationUI ui){
        super(new BorderLayout());

        if(ui.isLogin()){

            this.ui=ui;
```



```

//Παρε τα κεφάλαια από τη βάση
Database db = new Database();
Chapter[] chapters = db.getChapters();
categories = db.getCategories(selectedChapter);
Question[] questions =
db.getQuestions(categories[selectedCategory].getCategoryID(),level);
db.close();

String[] stringChapterList = new
String[chapters.length];
for(int i=0;i<chapters.length;i++){
stringChapterList[i] =
chapters[i].getName();
}
String[] stringCategoryList = new
String[categories.length-1];
for(int i=0;i<categories.length-1;i++){
stringCategoryList[i] =
categories[i+1].getName();
}

chapterList = new JComboBox(stringChapterList);
chapterList.setName("chapters");
chapterList.setSelectedIndex(0);
chapterList.addActionListener(this);

categoryList = new
JComboBox(stringCategoryList);
categoryList.setName("categories");
categoryList.setSelectedIndex(0);
categoryList.addActionListener(this);

JButton forward = new JButton("Επόμενη
ενότητα");

forward.setHorizontalAlignment(SwingConstants.LEFT );
forward.setActionCommand("forward");
forward.addActionListener((ActionListener)
this);

JButton back = new JButton("Προηγούμενη
ενότητα");

back.setHorizontalAlignment(SwingConstants.RIGHT );
back.setActionCommand("back");
back.addActionListener((ActionListener) this);

JPanel north = new JPanel();
north.setLayout(new GridLayout(2 , 6));

//Προσθήκη ετικέτας

```

```
Κεφαλαίου");
        JLabel label = new JLabel("Επιλογή
        label.setHorizontalAlignment(JLabel.CENTER);

        label.setFont(label.getFont().deriveFont(Font.BOLD).deriveFont(
(float) 14.0));
        north.add(label);
        //Προσθήκη λίστας επιλογής κεφαλαίων
        north.add(chapterList);

        //Προσθήκη ετικέτας
        label = new JLabel("Επίπεδο Δυσκολίας");
        label.setHorizontalAlignment(JLabel.CENTER);

        label.setFont(label.getFont().deriveFont(Font.BOLD).deriveFont(
(float) 14.0));
        north.add(label);

        String[] levels = {"Ευκολο", "Μέτριο",
        "Δύσκολο"};

        levelList = new JComboBox(levels);
        levelList.setName("levels");
        levelList.setSelectedIndex(0);
        levelList.addActionListener(this);
        north.add(levelList);

        north.add(back);
        //Προσθήκη ετικέτας
        label = new JLabel("Επιλογή Ενότητας");
        label.setHorizontalAlignment(JLabel.CENTER);

        label.setFont(label.getFont().deriveFont(Font.BOLD).deriveFont(
(float) 14.0));
        north.add(label);
        //Προσθήκη λίστας επιλογής σελίδων
        north.add(categoryList);
        //Προσθήκη κενού
        north.add(forward);

        add(north, BorderLayout.NORTH);

        //προσθήκη του Panel με τις ερωταποκρίσεις
        questionsPanel = new JPanel();
        questionsPanel.setLayout(new GridLayout(15,1));

        mainCategory = new JLabel();

        mainCategory.setHorizontalAlignment(JLabel.CENTER);

        mainCategory.setFont(label.getFont().deriveFont(Font.BOLD).deri
veFont((float) 16.0));
        mainCategory.setText(categories[0].getName());
```

```

        subCategory = new JLabel();

        subCategory.setHorizontalAlignment(JLabel.CENTER);

        subCategory.setFont(label.getFont().deriveFont(Font.BOLD).deriveFont((float) 14.0));

        subCategory.setText(categories[selectedCategory].getName());

        createQuestions(questions);

        //Δημιουργία του Panel για τις ερωτήσεις
        JScrollPane jp = new
JScrollPane(questionsPanel);
        add(jp, BorderLayout.CENTER);
    }

    public void actionPerformed(ActionEvent e) {

        if ("submit".equals(e.getActionCommand())) {
            int counter=0;
            boolean ready=true;
            Database db = new Database();
            Question[] questions =
db.getQuestions(categories[selectedCategory].getCategoryID(), level);
            if(questions.length>0){
                for(int i=0;i<questions.length;i++){
                    String correctAnswer =
db.getCorrectAnswer(questions[i].getId());
                    String givenAnswer = (String)
qList[i].getSelectedItem();
                    if(givenAnswer.equals("Επιλέξτε
απάντηση"))
                        ready=false;

                    if(correctAnswer.equals(givenAnswer))
                        counter++;
                }
                if(ready){ //έχουν επιλεγεί όλες οι
ερωτήσεις
                    float mark =
(float) counter/questions.length;
                    Test test = new
Test(ui.getUsername(), questions[0].getCategoryID(), mark, level);
                    boolean result =
db.addTest(test);
                    if(result)

                        createQuestions(questions);
                }
            }
        }
    }
}

```



```

        Question[] questions =
db.getQuestions(categories[selectedCategory].getCategoryID(), level);
        db.close();
        createQuestions(questions);

//Χρησιμοποιείται σα hack για να αναγκάσει τον
ανασχεδιασμό του GUI
        subCategory.setText("");

subCategory.setText(categories[selectedCategory].getName());
    }
    else if(e.getActionCommand().startsWith("add")){
        System.out.println(e.getActionCommand());
        StringTokenizer st = new
StringTokenizer(e.getActionCommand(), ":");
        int level=-1;
        int categoryID=-1;
        st.nextToken();
        categoryID = new
Integer(st.nextToken()).intValue();
        level = new
Integer(st.nextToken()).intValue();
        if(level>-1 && categoryID>-1){
            int correctAnswer =
aList.getSelectedIndex();
                int answerCount=0;
                if(correctAnswer>0){

if(!answerField1.getText().equals(""))
                    answerCount++;

if(!answerField2.getText().equals(""))
                    answerCount++;

if(!answerField3.getText().equals(""))
                    answerCount++;

if(!answerField4.getText().equals(""))
                    answerCount++;

                    if(answerCount>1){
                        Database db = new
Database();
                            int questionID =
db.insertQuestion(categoryID, level, questionField.getText());
                            int correct=0;
                            if(questionID>0){

if(!answerField1.getText().equals("")){
if(correctAnswer==1)
correct=1;

```



```

        Database db = new Database();
        categories =
db.getCategories(selectedChapter);
        Question[] questions =
db.getQuestions(categories[selectedCategory].getCategoryID(), level);
        db.close();

        //Στο νέο κεφάλαιο προεπιλέγεται η 1η
ενότητα
        categoryList.removeAllItems();
        for(int i=0;i<categories.length-1;i++){
            categories[i+1].getName();
            categoryList.addItem(
                categories[i+1].getName());
            categoryList.setSelectedIndex(0);

            createQuestions(questions);
        }
        //ο χρήστης επέλεξε άλλη ενότητα
        else if(cb.getName().equals("categories")){
            selectedCategory=cb.getSelectedIndex()+1;

            subCategory.setText(categories[selectedCategory].getName());
            //φόρτωσε τις ερωτήσεις
            Database db = new Database();
            Question[] questions =
db.getQuestions(categories[selectedCategory].getCategoryID(), level);
            db.close();
            createQuestions(questions);
        }
        else if(cb.getName().equals("levels")){
            level = cb.getSelectedIndex();

            //φόρτωσε τις ερωτήσεις
            Database db = new Database();
            Question[] questions =
db.getQuestions(categories[selectedCategory].getCategoryID(), level);
            db.close();
            createQuestions(questions);

            //Χρησιμοποιείται σα hack για να
αναγκάσει τον ανασχεδιασμό του GUI
            subCategory.setText("");

            subCategory.setText(categories[selectedCategory].getName());
        }
    }
}

public void createQuestions(Question[] questions){

```

```

questionsPanel.removeAll();
questionsPanel.add(mainCategory);
questionsPanel.add(subCategory);
questionsPanel.doLayout();

if(questions.length>0){
    Database db = new Database();
    questionsPanel.add(new JLabel());
    JLabel label = new JLabel("Ερωτήσεις:");
    label.setHorizontalAlignment(JLabel.CENTER);

    label.setFont(label.getFont().deriveFont(Font.BOLD).deriveFont(
(float) 12.0));

    questionsPanel.add(label);
    qList = new JComboBox[questions.length];

    for(int i=0;i<maxQuestionNumber-1;i++){
        if(i<questions.length){
            //JLabel questionText = new
JLabel();

            //questionText.setHorizontalAlignment(JLabel.LEFT);
            JTextArea questionText = new
JTextArea(2,30);

            questionText.setBackground(Color.LIGHT_GRAY);

            questionText.setFont(label.getFont().deriveFont((float) 12.0));
            questionText.setEditable(false);
            questionText.setLineWrap(true);
            JPanel qPanel = new JPanel();
            qPanel.add(questionText);
            JButton delete = new
JButton("Delete");

            delete.addActionListener(this);

            delete.setActionCommand("Delete"+questions[i].getId());
            qPanel.add( delete );
            int j=i+1;
            questionText.setText(" " + j +
". " +questions[i].getQuestion());
            questionsPanel.add(qPanel);
        }
        else{
            JPanel qPanel = new JPanel();
            qPanel.add(new JLabel());
            qPanel.add(new JLabel());
            questionsPanel.add(qPanel);
        }
    }
    JPanel qPanel = new JPanel();
    qPanel.add(new JLabel("Νέα ερώτηση:"));
    questionField = new JTextField(20);

```



```

        qPanel.add(questionField);
        qPanel.add(new JLabel("Απάντησεις:"));
        String[] answerList = { "Σωστή απάντηση", "1",
"2", "3", "4" };

        aList = new JComboBox(answerList);
        aList.setSelectedIndex(0);
        qPanel.add(aList);
        answerField1 = new JTextField(5);
        qPanel.add(answerField1);
        answerField2 = new JTextField(5);
        qPanel.add(answerField2);
        answerField3 = new JTextField(5);
        qPanel.add(answerField3);
        answerField4 = new JTextField(5);
        qPanel.add(answerField4);
        JButton add = new JButton("Προσθήκη");
        add.addActionListener(this);

        add.setActionCommand("add:"+categories[selectedCategory].getCategoryID() + ":" + level);
        qPanel.add(add);
        questionsPanel.add(qPanel);
        db.close();
    }
    else{
        questionsPanel.add(new JLabel());
        JLabel label = new JLabel("Ερωτήσεις:");
        label.setHorizontalAlignment(JLabel.CENTER);

        label.setFont(label.getFont().deriveFont(Font.BOLD).deriveFont(
(float) 12.0));

        questionsPanel.add(label);
        for(int i=0;i<maxQuestionNumber;i++){
            JPanel qPanel = new JPanel();
            qPanel.add(new JLabel());
            qPanel.add(new JLabel());
            questionsPanel.add(qPanel);
        }

        JPanel qPanel = new JPanel();
        qPanel.add(new JLabel("Νέα ερώτηση:"));
        questionField = new JTextField(20);
        qPanel.add(questionField);
        qPanel.add(new JLabel("Απάντησεις:"));
        String[] answerList = { "Επιλογή απάντησης",
"1", "2", "3", "4" };

        aList = new JComboBox(answerList);
        aList.setSelectedIndex(0);
        qPanel.add(aList);
        answerField1 = new JTextField(5);
        qPanel.add(answerField1);
        answerField2 = new JTextField(5);
        qPanel.add(answerField2);

```

```

        answerField3 = new JTextField(5);
        qPanel.add(answerField3);
        answerField4 = new JTextField(5);
        qPanel.add(answerField4);
        JButton add = new JButton("Προσθήκη");
        add.addActionListener(this);
        add.setActionCommand("add:"+
categories[selectedCategory].getCategoryID()+ ":" +level);
        qPanel.add(add);
        questionsPanel.add(qPanel);
    }
}
}

```

```

package org.economics;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Date;

import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JPasswordField;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.JTextField;

import javax.swing.Timer;

/**
 * Η κλάση που αρχικοποιεί το πρώτο tab με την εισαγωγή
 *
 */
public class AdminPanel extends JPanel implements ActionListener {
    public static JFrame frame = null;

    ApplicationUI ui;
    JTextField username ;
    JPasswordField password;
    JTextArea reply;
    JTextArea messagesArea;
    JTextArea marksArea;

```

```
JComboBox studentList;
JComboBox marksSortList;
Timer timer;

public AdminPanel(ApplicationUI ui) {
    super(new GridLayout(3,1));

    this.ui=ui;

    JPanel receivedMessages = new JPanel();
    receivedMessages.setLayout(new BorderLayout());
    //main.setBackground(Color.WHITE);
    add(receivedMessages);

    messagesArea = new JTextArea(10,20);
    messagesArea.setEditable(false);

    messagesArea.setFont(this.getFont().deriveFont(Font.BOLD).deriveFont((float) 14.0));

    printMessages();
    receivedMessages.add(new
JLabel("Μηνύματα:"),BorderLayout.NORTH);
    receivedMessages.add(new
JScrollPane(messagesArea),BorderLayout.CENTER);

    JButton clear =new JButton("Καθαρισμός μηνυμάτων");
    clear.setActionCommand("clear");
    clear.addActionListener(this);
    receivedMessages.add(clear, BorderLayout.SOUTH);

    JPanel replyPanel = new JPanel();
    replyPanel.setLayout(new BorderLayout());
    replyPanel.add(new JLabel("Νέο
μήνυμα:"),BorderLayout.NORTH);
    reply = new JTextArea(10,10);

    reply.setFont(this.getFont().deriveFont(Font.BOLD).deriveFont((float) 14.0));
    replyPanel.add(new JScrollPane(reply));
    JButton go =new JButton("Αποστολή");
    go.setActionCommand("go");
    go.addActionListener(this);

    //Παρε τα δεδομένα από τη βάση
    Database db = new Database();
    String[] usernames = db.getStudents();
    db.close();

    String[] stringStudentList = new
String[usernames.length+1];
```

```

        stringStudentList[0] = "Επιλογή φοιτητή";
        for(int i=0;i<usernames.length;i++)
            stringStudentList[i+1]=usernames[i];
        studentList = new JComboBox(stringStudentList);
        studentList.setName("students");
        studentList.setSelectedIndex(0);
        JPanel lowPanel = new JPanel();
        lowPanel.add(studentList);
        lowPanel.add(go);
        replyPanel.add(lowPanel, BorderLayout.SOUTH);
        add(replyPanel);

        JPanel marksPanel = new JPanel();
        marksPanel.setLayout(new BorderLayout());
        marksPanel.add(new JLabel("Βαθμολογία
φοιτητών"), BorderLayout.NORTH);
        marksArea = new JTextArea();
        marksArea.setEditable(false);
        marksArea.setFont(this.getFont().deriveFont((float)
12.0));
        marksPanel.add(new
JScrollPane(marksArea), BorderLayout.CENTER);

        marksSortList = new JComboBox();
        marksSortList.addItem("Ταξινομηση ανά φοιτητή");
        marksSortList.addItem("Ταξινομηση ανά κατηγορία");
        //marksSortList.addItem("Ταξινομηση ανά τύπο
ερώτησης");
        marksSortList.setName("marksSort");
        marksSortList.setSelectedIndex(0);
        marksSortList.addActionListener(this);
        marksPanel.add(marksSortList, BorderLayout.SOUTH);

        add(marksPanel);

        printStatisticsPerStudent();

        //Set up timer to drive animation events.
        timer = new Timer(5000, this);
        timer.setActionCommand("timer");
        timer.setInitialDelay(5000);
        timer.start();
    }

    public void actionPerformed(ActionEvent e) {
        if ("go".equals(e.getActionCommand())) {
            if(studentList.getSelectedIndex()>0){
                Message message = new Message();
                message.setFrom(ui.getUsername());
                message.setTo((String)
studentList.getSelectedItem());
                message.setText(reply.getText());
            }
        }
    }

```

```

        Database db = new Database();
        boolean result = db.addMessage(message);
        db.close();
        if(result)
            reply.setText("To μήνυμα
εστάλη!");
    }
}
else if ("clear".equals(e.getActionCommand())) {
    Database db = new Database();
    db.clearMessages(ui.getUsername());
    db.close();
    messagesArea.setText("Τα μηνύματα σβήστηκαν");
}
else if ("timer".equals(e.getActionCommand())) {
    printMessages();
}
else {
    JComboBox cb = (JComboBox)e.getSource();
    if(cb.getSelectedIndex()==0)
        printStatisticsPerStudent();
    else if(cb.getSelectedIndex()==1)
        printStatisticsPerCategory();
    else if(cb.getSelectedIndex()==2)
        printStatisticsPerQuestionType();
}
}

private void printStatisticsPerQuestionType() {
/*
    Database db = new Database();
    String[] students = db.getStudents();
    Chapter[] chapters = db.getChapters();
    String info="Αποτελέσματα:\n";
    String[] levels = { "Εύκολο", "Μέτριο", "Δύσκολο"};

    for(int i=0;i<students.length;i++){
        info = info + "Φοιτητής:" + students[i] +
\n";
        for(int j=0;j<chapters.length;j++){
            Category[] categories =
db.getCategories(chapters[j].getId());
            //info = info + "\t" +
chapters[j].getName() + "\n";
            for(int k=0;k<categories.length;k++){
                //info = info + "\t\t" +
categories[k].getName() + "\n";
                for(int
level=0;level<3;level++){
                    //info = info + "\t\t
Επίπεδο:" + levels[level] + "\n";

```

```
        if(categories[k].getParent()!=0){
            Test test =
            db.getTest(students[i], categories[k].getCategoryID(),level);
            if(test!=null){

                Question[] questions =
                db.getQuestions(categories[k].getCategoryID(),level);

                if(questions.length>0){

                    for(int q=0;q<questions.length;q++){

                        String
                        correctAnswer = db.getCorrectAnswer(questions[q].getId());

                        String givenAnswer = (String) qList[q].getSelectedItem();

                        if(correctAnswer.equals(givenAnswer)){

                                                    }
                                                }
                                            }
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
        marksArea.setText(info);
    */
}

public void printMessages(){
    Database db = new Database();
    Message[] messages = db.getMessages(ui.getUsername());
    db.close();
    String info="";
    for(int i=0;i<messages.length;i++){
        info = info + messages[i].getDate() + "\n";
        info = info + "From:" + messages[i].getFrom()
+ "\n";
        info = info + messages[i].getText() +
"\n\n\n";
    }
    messagesArea.setText(info);
}
```

```

public void printStatisticsPerStudent() {

    Database db = new Database();
    String[] students = db.getStudents();
    Chapter[] chapters = db.getChapters();
    String info="Αποτελέσματα:\n";
    String[] levels = { "Εύκολο", "Μέτριο", "Δύσκολο"};

    for(int i=0;i<students.length;i++){
        info = info + "Φοιτητής:" + students[i] +
"\n";
        for(int j=0;j<chapters.length;j++){
            Category[] categories =
db.getCategories(chapters[j].getId());
            info = info + "\t" +
chapters[j].getName() + "\n";
            for(int k=0;k<categories.length;k++){
                info = info + "\t\t" +
categories[k].getName() + "\n";
                for(int
level=0;level<3;level++){
                    info = info + "\t\t
Επίπεδο:" + levels[level] + "\n";

                    if(categories[k].getParent() !=0) {
                        Test test =
db.getTest(students[i], categories[k].getCategoryID(), level);
                        if(test!=null){
                            info =
info + "\t\t\t" + "Ημ/νία εξέτασης:" + test.getDate().toLocaleString()
+ "\n";
                            float
mark = test.getMark()*100;
                            info =
info + "\t\t\t" + "Βαθμός εξέτασης:" + mark + "%\n";
                        }
                        else{
                            info =
info + "\t\t\t" + "Ο φοιτητής δεν έχει δώσει ακόμα το τεστ\n";
                        }
                    }
                }
            }
        }
        marksArea.setText(info);
    }

public void printStatisticsPerCategory() {

    Database db = new Database();

```

```

String[] students = db.getStudents();
Chapter[] chapters = db.getChapters();
String info="Αποτελέσματα:\n";
String[] levels = { "Εύκολο", "Μέτριο", "Δύσκολο"};

for(int j=0;j<chapters.length;j++){
    Category[] categories =
db.getCategories(chapters[j].getId());
    info = info + chapters[j].getName() +"\n";
    for(int k=0;k<categories.length;k++){
        info = info + "\t" +
categories[k].getName() +"\n";
        if(categories[k].getParent()!=0){
            for(int
level=0;level<3;level++){
                info = info + "\t\t
Επίπεδο:" + levels[level] +"\n";
                for(int
i=0;i<students.length;i++){
                    info = info +
"\t\tΦοιτητής:" + students[i] + "\n";
                    Test test =
db.getTest(students[i], categories[k].getCategoryID(),level);
                    if(test!=null){
                        info =
info + "\t\t\t" + "Ημ/νία εξέτασης:"+test.getDate().toLocaleString()
+"\n";
                        float
mark = test.getMark()*100;
                        info =
info + "\t\t\t" + "Βαθμός εξέτασης:"+mark + "%\n";
                    }
                    else{
                        info =
info + "\t\t\t" + "Ο φοιτητής δεν έχει δώσει ακόμα το τεστ\n";
                    }
                }
            }
        }
    }
    marksArea.setText(info);
}
}

```

```
package org.economics;
```



```
public class Category {
    int categoryID;
    int parent;
    String name;
    int chapterID;
    int number;

    public Category(int categoryID, int parent, String name, int
chapterID, int number) {
        super();
        this.categoryID = categoryID;
        this.parent = parent;
        this.name = name;
        this.chapterID = chapterID;
        this.number = number;
    }

    public Category(){
        super();
    }

    public int getNumber() {
        return number;
    }

    public void setNumber(int number) {
        this.number = number;
    }

    public int getCategoryID() {
        return categoryID;
    }

    public void setCategoryID(int categoryID) {
        this.categoryID = categoryID;
    }

    public int getParent() {
        return parent;
    }

    public void setParent(int parent) {
        this.parent = parent;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
```

```
        public int getChapterID() {
            return chapterID;
        }

        public void setChapterID(int chapterID) {
            this.chapterID = chapterID;
        }
    }
}
```

```
    package org.economics;

    public class Chapter {
        int id;
        String name;

        public Chapter(int id, String name) {
            super();
            this.id = id;
            this.name = name;
        }

        public int getId() {
            return id;
        }

        public void setId(int id) {
            this.id = id;
        }

        public String getName() {
            return name;
        }

        public void setName(String name) {
            this.name = name;
        }
    };
}
package org.economics;

import java.awt.BorderLayout;
import java.awt.Dimension;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.Image;
import java.awt.event.ActionEvent;
```

```
import java.awt.event.ActionListener;
import java.io.BufferedInputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;

import javax.imageio.ImageIO;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.SwingConstants;

/**
 * Η κλάση που αρχικοποιεί το πρώτο tab με την εισαγωγή
 *
 */
public class ChaptersPanel extends JPanel implements ActionListener {
    public static JFrame frame = null;

    int selectedChapter = 1;
    int selectedPage = 1;
    JLabel image;
    Page[] pages;
    JComboBox chapterList;
    JComboBox pageList ;

    public ChaptersPanel(ApplicationUI ui){
        super(new BorderLayout());

        if(ui.isLogin()){
            //Παρε τα κεφάλαια από τη βάση
            Database db = new Database();
            Chapter[] chapters = db.getChapters();
            pages = db.getPages(selectedChapter);
            db.close();

            String[] stringChapterList = new
String[chapters.length];
            for(int i=0;i<chapters.length;i++){
                stringChapterList[i] =
chapters[i].getName();
            }
            String[] stringPageList = new
String[pages.length];
            for(int i=0;i<pages.length;i++){
                stringPageList[i] = new
Integer(pages[i].getNumber()).toString();
            }
        }
    }
}
```

```

    }

    chapterList = new JComboBox(stringChapterList);
    chapterList.setName("chapters");
    chapterList.setSelectedIndex(0);
    chapterList.addActionListener(this);

    pageList = new JComboBox(stringPageList);
    pageList.setName("pages");
    pageList.setSelectedIndex(0);
    pageList.addActionListener(this);

    JButton forward = new JButton("Επόμενη
σελίδα");

    forward.setHorizontalAlignment(SwingConstants.LEFT );
    forward.setActionCommand("forward");
    forward.addActionListener((ActionListener)
this);

    JButton back = new JButton("Προηγούμενη
σελίδα");

    back.setHorizontalAlignment(SwingConstants.RIGHT );
    back.setActionCommand("back");
    back.addActionListener((ActionListener) this);

    JPanel north = new JPanel();
    north.setLayout(new GridLayout(2 , 6));
    //Προσθήκη κενού
    north.add(new JLabel(""));
    //Προσθήκη ετικέτας
    JLabel label = new JLabel("Επιλογή
Κεφαλαίου");

    label.setHorizontalAlignment(JLabel.CENTER);

    label.setFont(label.getFont().deriveFont(Font.BOLD).deriveFont(
(float) 14.0));

    north.add(label);
    //Προσθήκη λίστας επιλογής κεφαλαίων
    north.add(chapterList);
    //Προσθήκη κενού
    north.add(new JLabel(""));
    //Προσθήκη κενού
    north.add(back);
    //Προσθήκη ετικέτας
    label = new JLabel("Επιλογή Σελίδας");
    label.setHorizontalAlignment(JLabel.CENTER);

    label.setFont(label.getFont().deriveFont(Font.BOLD).deriveFont(
(float) 14.0));

    north.add(label);

```

```

        //Προσθήκη λίστας επιλογής σελίδων
        north.add(pageList);
        //Προσθήκη κενού
        north.add(forward);

        add(north, BorderLayout.NORTH);

        //φορτωσε την πρώτη εικόνα του πρώτου κεφαλαίου
        if(pages.length>0){
            image = new JLabel(new
ImageIcon(pages[selectedPage-1].getImage()));
            add(new
JScrollPane(image), BorderLayout.CENTER);
        }
    }

    public void actionPerformed(ActionEvent e) {
        if ("forward".equals(e.getActionCommand())) {
            if(selectedPage<pages.length){
                selectedPage++;
                pageList.setSelectedIndex(selectedPage-
1);
                image.setIcon(new
ImageIcon(pages[selectedPage-1].getImage()));
            }
            else if ("back".equals(e.getActionCommand())) {
                if(selectedPage>1){
                    selectedPage--;

                    pageList.setSelectedIndex(selectedPage-1);
                    image.setIcon(new
ImageIcon(pages[selectedPage-1].getImage()));
                }
            }
            else{
                JComboBox cb = (JComboBox)e.getSource();
                //Ο χρήστης επέλεξε άλλο κεφάλαιο
                if(cb.getName().equals("chapters")){
                    selectedChapter=cb.getSelectedIndex()+1;
                    selectedPage=1;

                    Database db = new Database();
                    pages = db.getPages(selectedChapter);
                    db.close();

                    //Στο νέο κεφάλαιο εμφανίζεται η 1η
σελίδα

                    pageList.removeAllItems();
                    for(int i=0;i<pages.length;i++){

```

```
                pageList.addItem( new
Integer(pages[i].getNumber()).toString());
            }
            pageList.setSelectedIndex(0);

            if(pages.length>0)
                image.setIcon(new
ImageIcon(pages[0].getImage()));

        }
        //ο χρήστης επέλεξε άλλη σελίδα
        else if(cb.getName().equals("pages")){
            selectedPage=cb.getSelectedIndex()+1;
            if(selectedPage<1)
                selectedPage=1;
            image.setIcon(new
ImageIcon(pages[selectedPage-1].getImage()));
        }
    }
}
```

```
package org.economics;

import java.io.BufferedInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import javax.imageio.ImageIO;

public class Database {

    Connection con;

    public Database(){

        // Ορισμός του URL του database server στο localhost
        // με port number 3306.
```

```

        // String url ="jdbc:mysql://localhost:3306/economics";

        String
url="jdbc:mysql://localhost/economics?useUnicode=true&characterEncodin
g=UTF8&user=root";

        try {
            try {
                Class.forName("com.mysql.jdbc.Driver");
            } catch (ClassNotFoundException e) {
                e.printStackTrace();
            }
            //con = DriverManager.getConnection(url,"root",
main");

            con = DriverManager.getConnection(url);
        } catch (SQLException e) {
            e.printStackTrace();
        }

    }

    public String getMainInfo(){

        Statement stmt;
        String info= "";

        try {
            stmt = con.createStatement();
            //Δημιουργία του statement object
            stmt = con.createStatement(
                ResultSet.TYPE_SCROLL_INSENSITIVE,
                ResultSet.CONCUR_READ_ONLY);

            //Query the database, storing the result
            // in an object of type ResultSet
            ResultSet rs = stmt.executeQuery("SELECT * from
main");

            while(rs.next()){
                info = rs.getString("info");
                break;
            }

        } catch (SQLException e) {
            e.printStackTrace();
        }
        return info;
    }

    public String getMainInfoAfterLogin(){

        Statement stmt;
        String info= "";

```

```
        try {
            stmt = con.createStatement();
            //Δημιουργία του statement object
            stmt = con.createStatement(
                ResultSet.TYPE_SCROLL_INSENSITIVE,
                ResultSet.CONCUR_READ_ONLY);

            //Query the database, storing the result
            // in an object of type ResultSet
            ResultSet rs = stmt.executeQuery("SELECT * from
main");

            while(rs.next()){
                info = rs.getString("info");
            }

        } catch (SQLException e) {
            e.printStackTrace();
        }
        return info;
    }

    public Chapter[] getChapters(){

        Statement stmt;
        Chapter[] chapters = null;

        try {
            stmt = con.createStatement();
            //Δημιουργία του statement object
            stmt = con.createStatement(
                ResultSet.TYPE_SCROLL_INSENSITIVE,
                ResultSet.CONCUR_READ_ONLY);

            //Query the database, storing the result
            // in an object of type ResultSet
            ResultSet rs = stmt.executeQuery("SELECT * from
Chapter");

            //Υπολογισμός του πλήθους των κεφαλαίων
            int size=0;
            while(rs.next())
                size++;

            chapters= new Chapter[size];

            //reset row index
            rs.first();rs.previous();
            int i=0;
            while(rs.next()){
                chapters[i] = new
Chapter(rs.getInt("id"),rs.getString("name"));
            }
        }
    }
}
```



```
        i++;
    }

    } catch (SQLException e) {
        e.printStackTrace();
    }
    return chapters;
}

public Help[] getHelpPages(boolean isAdmin){

    Statement stmt;
    Help[] pages = null;
    int admin=0;

    if(isAdmin)
        admin=2;
    else
        admin=1;

    try {
        stmt = con.createStatement();
        //Δημιουργία του statement object
        stmt = con.createStatement(
            ResultSet.TYPE_SCROLL_INSENSITIVE,
            ResultSet.CONCUR_READ_ONLY);

        //Query the database, storing the result
        // in an object of type ResultSet
        ResultSet rs = stmt.executeQuery("SELECT
image,title FROM `help` WHERE admin<"+admin+" ORDER BY `order`");

        //Υπολογισμός του πλήθους των σελίδων
        int size=0;
        while(rs.next())
            size++;

        pages= new Help[size];

        //reset row index
        rs.first();rs.previous();
        int i=0;
        while(rs.next()){
            pages[i] = new Help();
            InputStream is = new
BufferedInputStream(rs.getBinaryStream("image"));
            try {
                pages[i] = new
Help(i+1,ImageIO.read(is),rs.getString("title"));
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```

```
        i++;
    }

    } catch (SQLException e) {
        e.printStackTrace();
    }
    return pages;
}

public Page[] getPages(int chapterID){
    Statement stmt;
    Page[] pages = null;

    try {
        stmt = con.createStatement();
        //Δημιουργία του statement object
        stmt = con.createStatement(
            ResultSet.TYPE_SCROLL_INSENSITIVE,
            ResultSet.CONCUR_READ_ONLY);

        //Query the database, storing the result
        // in an object of type ResultSet
        ResultSet rs = stmt.executeQuery("SELECT image
FROM `page` WHERE chapterID="+chapterID +" ORDER BY `order`");

        //Υπολογισμός του πλήθους των σελίδων
        int size=0;
        while(rs.next())
            size++;

        pages= new Page[size];

        //reset row index
        rs.first();rs.previous();
        int i=0;
        while(rs.next()){
            pages[i] = new Page();
            InputStream is = new
BufferedInputStream(rs.getBinaryStream("image"));
            try {
                pages[i] = new
Page(i+1, ImageIO.read(is));
            } catch (IOException e) {
                e.printStackTrace();
            }
            i++;
        }

    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

```
        return pages;
    }

    public Page getPage(int chapterID,int order){

        Statement stmt;
        Page page = new Page();
        try {
            stmt = con.createStatement();
            //Δημιουργία του statement object
            stmt = con.createStatement(
                ResultSet.TYPE_SCROLL_INSENSITIVE,
                ResultSet.CONCUR_READ_ONLY);

            //Query the database, storing the result
            // in an object of type ResultSet
            ResultSet rs = stmt.executeQuery("SELECT
id,image FROM `page` " +
                                           "WHERE `order`="+order +" AND
chapterID="+chapterID );

            while(rs.next()){

                InputStream is = new
BufferedInputStream(rs.getBinaryStream("image"));
                try {
                    page = new
Page(rs.getInt("id"),ImageIO.read(is));
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }

        } catch (SQLException e) {
            e.printStackTrace();
        }
        return page;
    }

    public Category[] getCategories(int chapterID){

        Statement stmt;
        Category[] categories = null;

        try {
            stmt = con.createStatement();
            //Δημιουργία του statement object
            stmt = con.createStatement(
                ResultSet.TYPE_SCROLL_INSENSITIVE,
                ResultSet.CONCUR_READ_ONLY);
```

```

        //Query the database, storing the result
        // in an object of type ResultSet
        ResultSet rs = stmt.executeQuery("SELECT * FROM
`category` WHERE chapterID="+chapterID +" ORDER BY `order`");

        //Υπολογισμός του πλήθους των σελίδων
        int size=0;
        while(rs.next())
            size++;

        categories= new Category[size];

        //reset row index
        rs.first();rs.previous();
        int i=0;
        while(rs.next()){
            categories[i] = new Category();
            categories[i].setCategoryID(rs.getInt("id"));

            categories[i].setChapterID(rs.getInt("chapterID"));
            categories[i].setParent(rs.getInt("parent"));
            categories[i].setName(rs.getString("name"));
            i++;
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return categories;
}

public Question[] getQuestions(int categoryID, int level){

    Statement stmt;
    Question[] questions = null;

    try {
        stmt = con.createStatement();
        //Δημιουργία του statement object
        stmt = con.createStatement(
            ResultSet.TYPE_SCROLL_INSENSITIVE,
            ResultSet.CONCUR_READ_ONLY);

        //Query the database, storing the result
        // in an object of type ResultSet
        ResultSet rs = stmt.executeQuery("SELECT * FROM
`question` WHERE categoryID="+categoryID + " AND level="+level +
            " ORDER BY `order`");

        //Υπολογισμός του πλήθους των σελίδων
        int size=0;
        while(rs.next())
            size++;
    }
}

```

```

        questions= new Question[size];

        //reset row index
        rs.first();rs.previous();
        int i=0;
        while(rs.next()){
            questions[i] = new Question();
            questions[i].setId(rs.getInt("id"));

            questions[i].setCategoryID(rs.getInt("categoryID"));

            questions[i].setQuestion(rs.getString("question"));
            questions[i].setNumber(rs.getInt("order"));
            i++;
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return questions;
}

public Message[] getMessages(String username){

    Statement stmt;
    Message[] messages = null;

    try {
        stmt = con.createStatement();
        //Δημιουργία του statement object
        stmt = con.createStatement(
            ResultSet.TYPE_SCROLL_INSENSITIVE,
            ResultSet.CONCUR_READ_ONLY);

        //Query the database, storing the result
        // in an object of type ResultSet
        ResultSet rs = stmt.executeQuery("SELECT * FROM
`messages` WHERE `to`='"+username+"' ORDER BY `date` ASC ");

        //Υπολογισμός του πλήθους των σελίδων
        int size=0;
        while(rs.next())
            size++;

        messages= new Message[size];

        //reset row index
        rs.first();rs.previous();
        int i=0;
        while(rs.next()){
            messages[i] = new Message();
            messages[i].setText(rs.getString("text"));
            messages[i].setFrom(rs.getString("from"));

```

```
                messages[i].setTo(rs.getString("to"));
                messages[i].setDate(rs.getString("date"));
                i++;
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
    return messages;
}

public String[] getStudents(){

    Statement stmt;
    String[] students = null;

    try {
        stmt = con.createStatement();
        //Δημιουργία του statement object
        stmt = con.createStatement(
            ResultSet.TYPE_SCROLL_INSENSITIVE,
            ResultSet.CONCUR_READ_ONLY);

        //Query the database, storing the result
        // in an object of type ResultSet
        ResultSet rs = stmt.executeQuery("SELECT username FROM
`user` WHERE `admin`<>1 ");

        //Υπολογισμός του πλήθους των σελίδων
        int size=0;
        while(rs.next())
            size++;

        students= new String[size];

        //reset row index
        rs.first();rs.previous();
        int i=0;
        while(rs.next()){
            students[i] = rs.getString("username");
            i++;
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return students;
}

public String getAdmin(){

    Statement stmt;
    String username = null;

    try {
```

```
        stmt = con.createStatement();
        //Δημιουργία του statement object
        stmt = con.createStatement(
            ResultSet.TYPE_SCROLL_INSENSITIVE,
            ResultSet.CONCUR_READ_ONLY);

        //Query the database, storing the result
        // in an object of type ResultSet
        ResultSet rs = stmt.executeQuery("SELECT username FROM
`user` WHERE `admin`=1 ");

        while(rs.next()){
            username = rs.getString("username");
            break;
        }

    } catch (SQLException e) {
        e.printStackTrace();
    }
    return username;
}

public String getCorrectAnswer(int questionID){

    Statement stmt;
    String answer = null;

    try {
        stmt = con.createStatement();
        //Δημιουργία του statement object
        stmt = con.createStatement(
            ResultSet.TYPE_SCROLL_INSENSITIVE,
            ResultSet.CONCUR_READ_ONLY);

        //Query the database, storing the result
        // in an object of type ResultSet
        ResultSet rs = stmt.executeQuery("SELECT `answer` FROM
`answer` WHERE `correct`=1 and `questionID`="+questionID);
        while(rs.next()){
            answer = rs.getString("answer");
        }

    } catch (SQLException e) {
        e.printStackTrace();
    }
    return answer;
}

public String[] getAnswerList(int questionID){

    Statement stmt;
    String[] answerList = null;
```

```
try {
    stmt = con.createStatement();
    //Δημιουργία του statement object
    stmt = con.createStatement(
        ResultSet.TYPE_SCROLL_INSENSITIVE,
        ResultSet.CONCUR_READ_ONLY);

    //Query the database, storing the result
    // in an object of type ResultSet
    ResultSet rs = stmt.executeQuery("SELECT `answer` FROM
`answer` WHERE `questionID`="+questionID);

    //Υπολογισμός του πλήθους των σελίδων
    int size=0;
    while(rs.next())
        size++;

    size++;
    answerList= new String[size];
    answerList[0]="Επιλέξτε απάντηση";
    //reset row index
    rs.first();rs.previous();

    //προσθεσε τις πιθανές απαντήσεις
    int i=1;
    while(rs.next()){
        answerList[i] = rs.getString("answer");
        i++;
    }
} catch (SQLException e) {
    e.printStackTrace();
}
return answerList;
}

public boolean checkLogin(String username,String password){

    Statement stmt;
    boolean login=false;

    try {
        stmt = con.createStatement();
        //Get another statement object initialized
        // as shown.
        stmt = con.createStatement(
            ResultSet.TYPE_SCROLL_INSENSITIVE,
            ResultSet.CONCUR_READ_ONLY);

        //Query the database, storing the result
        // in an object of type ResultSet
```



```
        ResultSet rs = stmt.executeQuery("SELECT * from user
WHERE username= '" +username + "' and password='" + password + "'");
        if(rs.next()){
            login=true;
        }

    } catch (SQLException e) {
        e.printStackTrace();
    }
    return login;
}

public boolean checkAdmin(String username){

    Statement stmt;
    boolean admin=false;

    try {
        stmt = con.createStatement();
        //Get another statement object initialized
        // as shown.
        stmt = con.createStatement(
            ResultSet.TYPE_SCROLL_INSENSITIVE,
            ResultSet.CONCUR_READ_ONLY);

        //Query the database, storing the result
        // in an object of type ResultSet
        ResultSet rs = stmt.executeQuery("SELECT admin from
user WHERE username= '" +username + "'");
        while(rs.next()){
            if(rs.getInt("admin")==1 )
                admin=true;
        }

    } catch (SQLException e) {
        e.printStackTrace();
    }
    return admin;
}

public boolean categoryTested(String username,int categoryID, int
level){

    Statement stmt;
    boolean tested=false;

    try {
        stmt = con.createStatement();
        //Get another statement object initialized
        // as shown.
        stmt = con.createStatement(
            ResultSet.TYPE_SCROLL_INSENSITIVE,
            ResultSet.CONCUR_READ_ONLY);
```

```

        //Query the database, storing the result
        // in an object of type ResultSet
        ResultSet rs = stmt.executeQuery("SELECT * from `test`
WHERE `username`= '" +username + "' AND `categoryID`="+categoryID +
" AND `level`=" + level );
        if(rs.next()){
            tested=true;
        }

    } catch (SQLException e) {
        e.printStackTrace();
    }
    return tested;
}

public Test getTest(String username,int categoryID, int level){

    Statement stmt;
    Test test=null;

    try {
        stmt = con.createStatement();
        //Get another statement object initialized
        // as shown.
        stmt = con.createStatement(
            ResultSet.TYPE_SCROLL_INSENSITIVE,
            ResultSet.CONCUR_READ_ONLY);

        //Query the database, storing the result
        // in an object of type ResultSet
        ResultSet rs = stmt.executeQuery("SELECT * from `test`
WHERE `username`= '" +username + "' AND `categoryID`="+categoryID + "
AND `level`=" + level );
        if(rs.next()){
            test = new Test(username, categoryID,
rs.getFloat("mark"), rs.getDate("date"),rs.getInt("level") );
        }

    } catch (SQLException e) {
        e.printStackTrace();
    }
    return test;
}

public boolean addTest(Test test){
    Statement stmt;
    boolean status;
    try {
        stmt = con.createStatement();
        String insertString = "INSERT INTO test
(`mark`,`username`,`categoryID`,`date`,`level`) VALUES('"

```

```

+test.getMark() + "','" + test.getUsername() + "','" +
test.getCategoryID() + "','" + NOW() + "','" + test.getLevel() + " )";
        stmt.executeUpdate(insertString);
        //if statement executes without exception, set status
to true
        status = true;
    } catch (SQLException e) {
        e.printStackTrace();
        status=false;
    }

    return status;
}

public boolean addMessage(Message message){
    Statement stmt;
    boolean status;
    try {
        stmt = con.createStatement();
        String insertString = "INSERT INTO messages
(`from`,`to`,`text`,`date`) VALUES ('" +message.getFrom() + "','" +
message.getTo() + "','" + message.getText() + "','" + NOW() )";
        stmt.executeUpdate(insertString);
        //if statement executes without exception, set status
to true
        status = true;
    } catch (SQLException e) {
        e.printStackTrace();
        status=false;
    }

    return status;
}

public boolean clearMessages(String username){
    Statement stmt;
    boolean status;
    try {
        stmt = con.createStatement();
        String insertString = "DELETE FROM `messages` WHERE
`to`='" + username + "'";
        stmt.executeUpdate(insertString);
        //if statement executes without exception, set status
to true
        status = true;
    } catch (SQLException e) {
        e.printStackTrace();
        status=false;
    }

    return status;
}
}

```

```
public void close(){
    try {
        con.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

public boolean deleteTest(String username, int categoryID) {
    Statement stmt;
    boolean status;
    try {
        stmt = con.createStatement();
        String deleteString = "DELETE FROM `test` WHERE
`username`='" + username + "' AND categoryID="+categoryID;
        stmt.executeUpdate(deleteString);
        //if statement executes without exception, set
status to true
        status = true;
    } catch (SQLException e) {
        e.printStackTrace();
        status=false;
    }

    return status;
}

public boolean deleteQuestion(int questionID) {
    Statement stmt;
    boolean status;
    try {
        stmt = con.createStatement();
        String deleteString = "DELETE FROM `answer`
WHERE `questionID`="+questionID;
        stmt.executeUpdate(deleteString);

        deleteString = "DELETE FROM `question` WHERE
`id`="+questionID;
        stmt.executeUpdate(deleteString);

        //if statement executes without exception, set
status to true
        status = true;
    } catch (SQLException e) {
        e.printStackTrace();
        status=false;
    }

    return status;
}
```

```

    }

    public int insertQuestion(int categoryID,int level, String
text) {
        Statement stmt;
        boolean status;
        int chapterID=0;
        int questionID=0;
        int order=0;
        try {
            stmt = con.createStatement();
            //Get another statement object initialized
            // as shown.
            stmt = con.createStatement(
                ResultSet.TYPE_SCROLL_INSENSITIVE,
                ResultSet.CONCUR_READ_ONLY);

            //Query the database, storing the result
            // in an object of type ResultSet
            ResultSet rs = stmt.executeQuery("SELECT
chapterID from `category` WHERE `id`=" +categoryID );
            if(rs.next()){
                chapterID=rs.getInt("chapterID") ;
            }

        } catch (SQLException e) {
            e.printStackTrace();
        }

        try {
            stmt = con.createStatement();
            //Get another statement object initialized
            // as shown.
            stmt = con.createStatement(
                ResultSet.TYPE_SCROLL_INSENSITIVE,
                ResultSet.CONCUR_READ_ONLY);

            //Query the database, storing the result
            // in an object of type ResultSet
            ResultSet rs = stmt.executeQuery("SELECT
max(`order`) FROM question WHERE categoryID =" +categoryID );
            if(rs.next()){
                order=rs.getInt("max(`order`")+1;
            }

        } catch (SQLException e) {
            e.printStackTrace();
        }

        try {
            stmt = con.createStatement();
            String insertString = "INSERT INTO question
(`question`,`chapterID`,`categoryID`,`level`,`order`) VALUES('"+text

```

```

+ "','" + chapterID + "','" + categoryID + "','" + level + "','" + order
+ "');"
        stmt.executeUpdate(insertString);
        //if statement executes without exception, set
status to true
        status = true;
    } catch (SQLException e) {
        e.printStackTrace();
        status=false;
    }
    //System.out.println("questionID:"+questionID);
    if(status){
        try {
            stmt = con.createStatement();
            //Get another statement object
initialized
            // as shown.
            stmt = con.createStatement(
ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);

            //Query the database, storing the result
            // in an object of type ResultSet
            ResultSet rs = stmt.executeQuery("SELECT
LAST_INSERT_ID()" );
            if(rs.next()){
                questionID=rs.getInt("LAST_INSERT_ID()" );
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
    //System.out.println("questionID:"+questionID);
    return questionID;
}

public boolean insertAnswer(String text, int questionID, int
correct) {
    Statement stmt;
    boolean status;
    try {
        stmt = con.createStatement();
        String insertString = "INSERT INTO answer
(`answer`,`questionID`,`correct`) VALUES('" +text + "','" +
questionID + "','" + correct + "')";
        stmt.executeUpdate(insertString);
        //if statement executes without exception, set
status to true
        status = true;
    } catch (SQLException e) {

```

```
        e.printStackTrace();
        status=false;
    }
    return status;
}
}
```

```
package org.economics;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.Image;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;

import javax.imageio.ImageIO;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.SwingConstants;

/**
 * Η κλάση που αρχικοποιεί το πρώτο tab με την εισαγωγή
 *
 */
public class ExercisesPanel extends JPanel implements ActionListener
{
    public static JFrame frame = null;

    int selectedChapter =1;
    int selectedCategory = 1;
    int level = 0;
    JLabel mainCategory;
```

```
JLabel subCategory;
Category[] categories;
JComboBox chapterList;
JComboBox categoryList ;
JComboBox levelList;
JPanel questionsPanel;
int maxQuestionNumber=10;
ApplicationUI ui;
JComboBox[] qList;

public ExercisesPanel(ApplicationUI ui){
    super(new BorderLayout());

    if(ui.isLogin()){

        this.ui=ui;

        //Παρε τα κεφάλαια από τη βάση
        Database db = new Database();
        Chapter[] chapters = db.getChapters();
        categories = db.getCategories(selectedChapter);
        Question[] questions =
db.getQuestions(categories[selectedCategory].getCategoryID(),level);
        db.close();

        String[] stringChapterList = new
String[chapters.length];
        for(int i=0;i<chapters.length;i++){
            stringChapterList[i] =
chapters[i].getName();
        }
        String[] stringCategoryList = new
String[categories.length-1];
        for(int i=0;i<categories.length-1;i++){
            stringCategoryList[i] =
categories[i+1].getName();
        }

        chapterList = new JComboBox(stringChapterList);
        chapterList.setName("chapters");
        chapterList.setSelectedIndex(0);
        chapterList.addActionListener(this);

        categoryList = new
JComboBox(stringCategoryList);
        categoryList.setName("categories");
        categoryList.setSelectedIndex(0);
        categoryList.addActionListener(this);

        JButton forward = new JButton("Επόμενη
ενότητα");
```



```
        forward.setHorizontalAlignment(SwingConstants.LEFT );
        forward.setActionCommand("forward");
        forward.addActionListener((ActionListener)
this);

        JButton back = new JButton("Προηγούμενη
ενότητα");

        back.setHorizontalAlignment(SwingConstants.RIGHT );
        back.setActionCommand("back");
        back.addActionListener((ActionListener) this);

        JPanel north = new JPanel();
        north.setLayout(new GridLayout(2 , 6));

        //Προσθήκη ετικέτας
        JLabel label = new JLabel("Επιλογή
Κεφαλαίου");

        label.setHorizontalAlignment(JLabel.CENTER);

        label.setFont(label.getFont().deriveFont(Font.BOLD).deriveFont(
(float) 14.0));

        north.add(label);
        //Προσθήκη λίστας επιλογής κεφαλαίων
        north.add(chapterList);

        //Προσθήκη ετικέτας
        label = new JLabel("Επίπεδο Δυσκολίας");
        label.setHorizontalAlignment(JLabel.CENTER);

        label.setFont(label.getFont().deriveFont(Font.BOLD).deriveFont(
(float) 14.0));

        north.add(label);

        String[] levels = {"Ευκολο", "Μέτριο",
"Δύσκολο"};

        levelList = new JComboBox(levels);
        levelList.setName("levels");
        levelList.setSelectedIndex(0);
        levelList.addActionListener(this);
        north.add(levelList);

        north.add(back);
        //Προσθήκη ετικέτας
        label = new JLabel("Επιλογή Ενότητας");
        label.setHorizontalAlignment(JLabel.CENTER);

        label.setFont(label.getFont().deriveFont(Font.BOLD).deriveFont(
(float) 14.0));

        north.add(label);
        //Προσθήκη λίστας επιλογής σελίδων
        north.add(categoryList);
```

```

        //Προσθήκη κενού
        north.add(forward);

        add(north, BorderLayout.NORTH);

        //προσθήκη του Panel με τις ερωταποκρίσεις
        questionsPanel = new JPanel();
        questionsPanel.setLayout(new GridLayout(15,1));

        mainCategory = new JLabel();

        mainCategory.setHorizontalAlignment(JLabel.CENTER);

        mainCategory.setFont(label.getFont().deriveFont(Font.BOLD).deriveFont((float) 16.0));
        mainCategory.setText(categories[0].getName());

        subCategory = new JLabel();

        subCategory.setHorizontalAlignment(JLabel.CENTER);

        subCategory.setFont(label.getFont().deriveFont(Font.BOLD).deriveFont((float) 14.0));

        subCategory.setText(categories[selectedCategory].getName());

        createQuestions(questions);

        //Δημιουργία του Panel για τις ερωτήσεις
        JScrollPane jp = new
JScrollPane(questionsPanel);
        add(jp, BorderLayout.CENTER);
    }

    public void actionPerformed(ActionEvent e) {

        if ("submit".equals(e.getActionCommand())) {
            int counter=0;
            boolean ready=true;
            Database db = new Database();
            Question[] questions =
db.getQuestions(categories[selectedCategory].getCategoryID(), level);
            if(questions.length>0){
                for(int i=0;i<questions.length;i++){
                    String correctAnswer =
db.getCorrectAnswer(questions[i].getId());
                    String givenAnswer = (String)
qList[i].getSelectedItem();
                    if(givenAnswer.equals("Επιλέξτε
απάντηση"))

```

```

ready=false;

if(correctAnswer.equals(givenAnswer))
    counter++;
}
if(ready){ //έχουν επιλεγεί όλες οι
ερωτήσεις
    float mark =
(float)counter/questions.length;
    Test test = new
Test(ui.getUsername(), questions[0].getCategoryID(), mark, level);
    boolean result =
db.addTest(test);
    if(result)
        createQuestions(questions);
    }
    db.close();
}
else if ("forward".equals(e.getActionCommand())) {
    if(selectedCategory<categories.length-1){
        selectedCategory++;

        categoryList.setSelectedIndex(selectedCategory-1);
        //load questions
        Database db = new Database();
        Question[] questions =
db.getQuestions(categories[selectedCategory].getCategoryID(),level);
        db.close();
        createQuestions(questions);

        //Χρησιμοποιείται σα hack για να
αναγκάσει τον ανασχεδιασμό του GUI
        subCategory.setText("");
subCategory.setText(categories[selectedCategory].getName());
    }
}
else if ("back".equals(e.getActionCommand())) {
    if(selectedCategory>1){
        selectedCategory--;

        categoryList.setSelectedIndex(selectedCategory-1);
        //load questions
        Database db = new Database();
        Question[] questions =
db.getQuestions(categories[selectedCategory].getCategoryID(),level);
        db.close();
        createQuestions(questions);

        //Χρησιμοποιείται σα hack για να
αναγκάσει τον ανασχεδιασμό του GUI

```

```

subCategory.setText("");

subCategory.setText(categories[selectedCategory].getName());
    }
    else if ("reset".equals(e.getActionCommand())) {
        Database db = new Database();

db.deleteTest(ui.getUsername(), categories[selectedCategory].getCategoryID());

        Question[] questions =
db.getQuestions(categories[selectedCategory].getCategoryID(), level);
        db.close();
        createQuestions(questions);

//Χρησιμοποιείται σα hack για να αναγκάσει τον
ανασχεδιασμό του GUI
        subCategory.setText("");

subCategory.setText(categories[selectedCategory].getName());
    }
    else{
        JComboBox cb = (JComboBox)e.getSource();
        //Ο χρήστης επέλεξε άλλο κεφάλαιο
        if(cb.getName().equals("chapters")){
            selectedChapter=cb.getSelectedIndex()+1;
            selectedCategory=1;

            Database db = new Database();
            categories =
db.getCategories(selectedChapter);
            Question[] questions =
db.getQuestions(categories[selectedCategory].getCategoryID(), level);
            db.close();

            //Στο νέο κεφάλαιο προεπιλέγεται η 1η
            ενότητα
            categoryList.removeAllItems();
            for(int i=0;i<categories.length-1;i++){
                categoryList.addItem(
categories[i+1].getName());
            }
            categoryList.setSelectedIndex(0);

            createQuestions(questions);

mainCategory.setText(categories[0].getName());
subCategory.setText("");

subCategory.setText(categories[selectedCategory].getName());

```

```

    }
    //ο χρήστης επέλεξε άλλη ενότητα
    else if (cb.getName().equals("categories")) {

        selectedCategory=cb.getSelectedIndex()+1;

        subCategory.setText(categories[selectedCategory].getName());
        //φόρτωσε τις ερωτήσεις
        Database db = new Database();
        Question[] questions =
db.getQuestions(categories[selectedCategory].getCategoryID(), level);
        categories =
db.getCategories(selectedChapter);
        db.close();
        createQuestions(questions);

    }
    else if (cb.getName().equals("levels")) {
        level = cb.getSelectedIndex();

        //φόρτωσε τις ερωτήσεις
        Database db = new Database();
        Question[] questions =
db.getQuestions(categories[selectedCategory].getCategoryID(), level);
        db.close();
        createQuestions(questions);

        //Χρησιμοποιείται σα hack για να
αναγκάσει τον ανασχεδιασμό του GUI
        subCategory.setText("");

        subCategory.setText(categories[selectedCategory].getName());

    }
}

public void createQuestions(Question[] questions){
    questionsPanel.removeAll();
    questionsPanel.add(mainCategory);
    questionsPanel.add(subCategory);
    questionsPanel.doLayout();

    if(questions.length>0){
        Database db = new Database();

        if(!db.categoryTested(ui.getUsername(), questions[0].getCategory
ID(), level) ){
            questionsPanel.add(new JLabel());
            JLabel label = new JLabel("Ερωτήσεις:");

            label.setHorizontalAlignment(JLabel.CENTER);

```

```

        label.setFont(label.getFont().deriveFont(Font.BOLD).deriveFont(
(float) 12.0));
        questionsPanel.add(label);
        qList = new JComboBox[questions.length];
        for(int i=0;i<maxQuestionNumber;i++){
            if(i<questions.length){
                //JLabel questionText =
new JLabel();
                JTextArea questionText =
new JTextArea(2,30);
                questionText.setBackground(Color.LIGHT_GRAY);
                questionText.setFont(label.getFont().deriveFont((float) 12.0));
                questionText.setEditable(false);
                questionText.setLineWrap(true);
                //questionText.setHorizontalAlignment(JLabel.LEFT);
                JPanel qPanel = new
JPanel();
                qPanel.add(questionText);
                String[] answerList =
db.getAnswerList(questions[i].getId());
                JComboBox(answerList);
                qList[i].setSelectedIndex(0);
                qPanel.add(qList[i]);
                int j=i+1;
                questionText.setText(" "
+ j + ". " +questions[i].getQuestion());
                questionsPanel.add(qPanel);
            }
            else{
                JPanel qPanel = new
JPanel();
                qPanel.add(new JLabel());
                qPanel.add(new JLabel());
                questionsPanel.add(qPanel);
            }
        }
        if(!ui.isAdmin()){
            JPanel qPanel = new JPanel();
            JButton submit = new
JButton("Αποστολή");
            submit.setActionCommand("submit");

```

```

submit.addActionListener(this);
qPanel.add(new JLabel("Μολις
επιλέξετε τις απαντήσεις σε όλες τις ερωτήσεις πατήστε:"));
qPanel.add(submit);
questionsPanel.add(qPanel);
    }
}
else{
    Test test =
db.getTest(ui.getUsername(),questions[0].getCategoryID(),level);
    questionsPanel.add(new JLabel());
    JLabel label = new JLabel("Έχετε δώσει
το τεστ για αυτή την ενότητα");

    label.setHorizontalAlignment(JLabel.CENTER);

    label.setFont(label.getFont().deriveFont(Font.BOLD).deriveFont(
(float) 12.0));

    questionsPanel.add(label);
    label = new JLabel();

    label.setHorizontalAlignment(JLabel.CENTER);

    label.setFont(label.getFont().deriveFont(Font.BOLD).deriveFont(
(float) 12.0));

    label.setText("Ημ/νία
εξέτασης:"+test.getDate().toLocaleString());
    questionsPanel.add(label);
    float mark = test.getMark()*100;
    label = new JLabel();

    label.setHorizontalAlignment(JLabel.CENTER);

    label.setFont(label.getFont().deriveFont(Font.BOLD).deriveFont(
(float) 12.0));

    questionsPanel.add(label);
    label.setText("Βαθμός:"+mark + "%");

    JButton reset = new JButton("Ανάφραση");
    reset.setActionCommand("reset");
    reset.addActionListener(this);
    questionsPanel.add(reset);

    for(int i=0;i<maxQuestionNumber-2;i++){
        JPanel qPanel = new JPanel();
        qPanel.add(new JLabel());
        qPanel.add(new JLabel());
        questionsPanel.add(qPanel);
    }
}
db.close();
}
else{

```

```
        questionsPanel.add(new JLabel());
        JLabel label = new JLabel("Ερωτήσεις:");
        label.setHorizontalAlignment(JLabel.CENTER);

        label.setFont(label.getFont().deriveFont(Font.BOLD).deriveFont(
(float) 12.0));

        questionsPanel.add(label);
        for(int i=0;i<maxQuestionNumber;i++){
            JPanel qPanel = new JPanel();
            qPanel.add(new JLabel());
            qPanel.add(new JLabel());
            questionsPanel.add(qPanel);
        }
    }
}
```

```
package org.economics;

import java.awt.Image;

public class Help {
    int id;
    int order;
    Image image;
    String title;
    //boolean admin;

    public Help(int id, int order, Image image, String title) {
        super();
        this.id = id;
        this.order = order;
        this.image = image;
        this.title = title;
    }

    public Help(int order, Image image, String title) {
        super();
        this.order = order;
        this.image = image;
        this.title = title;
    }

    public Help() {

    }

    public int getId() {
        return id;
    }
}
```



```
public void setId(int id) {
    this.id = id;
}

public int getOrder() {
    return order;
}

public void setOrder(int order) {
    this.order = order;
}

public Image getImage() {
    return image;
}

public void setImage(Image image) {
    this.image = image;
}

public String getTitle() {
    return title;
}

public void setTitle(String title) {
    this.title = title;
}
}
```

```
package org.economics;

import java.awt.BorderLayout;
import java.awt.Dimension;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.Image;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.BufferedInputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;

import javax.imageio.ImageIO;
import javax.swing.ImageIcon;
import javax.swing.JButton;
```

```
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.SwingConstants;

/**
 * Η κλάση που αρχικοποιεί το πρώτο tab με την εισαγωγή
 *
 */
public class HelpPanel extends JPanel implements ActionListener {
    public static JFrame frame = null;

    int selectedPage = 1;
    JLabel image;
    Help[] pages;
    JComboBox pageList ;

    public HelpPanel(ApplicationUI ui){
        super(new BorderLayout());

        if(ui.isLogin()){
            //Παρε τα κεφάλαια από τη βάση
            Database db = new Database();

            pages = db.getHelpPages(ui.isAdmin());
            db.close();

            String[] stringPageList = new
String[pages.length];
            for(int i=0;i<pages.length;i++){
                stringPageList[i] = new
Integer(pages[i].getOrder()).toString();
            }

            pageList = new JComboBox(stringPageList);
            pageList.setName("pages");
            pageList.setSelectedIndex(0);
            pageList.addActionListener(this);

            JButton forward = new JButton("Επόμενη
σελίδα");

            forward.setHorizontalAlignment(SwingConstants.LEFT );
            forward.setActionCommand("forward");
            forward.addActionListener((ActionListener)
this);
        }
    }
}
```

```

        JButton back = new JButton("Προηγούμενη
σελίδα");

        back.setHorizontalAlignment(SwingConstants.RIGHT );
        back.setActionCommand("back");
        back.addActionListener((ActionListener) this);

        JPanel north = new JPanel();
        north.setLayout(new GridLayout(2 , 6));
        //Προσθήκη κενού
        north.add(new JLabel(""));
        //Προσθήκη ετικέτας
        JLabel label = new JLabel("");
        label.setHorizontalAlignment(JLabel.CENTER);

        label.setFont(label.getFont().deriveFont(Font.BOLD).deriveFont(
(float) 14.0));

        north.add(label);
        //Προσθήκη λίστας επιλογής κεφαλαίων
        north.add(new JLabel(""));
        //Προσθήκη κενού
        north.add(new JLabel(""));
        //Προσθήκη κενού
        north.add(back);
        //Προσθήκη ετικέτας
        label = new JLabel("Επιλογή Σελίδας:");
        label.setHorizontalAlignment(JLabel.CENTER);

        label.setFont(label.getFont().deriveFont(Font.BOLD).deriveFont(
(float) 14.0));

        north.add(label);

        //Προσθήκη λίστας επιλογής σελίδων
        north.add(pageList);
        //Προσθήκη κενού
        north.add(forward);

        add(north, BorderLayout.NORTH);

        //φορτωσε την πρώτη εικόνα του πρώτου κεφαλαίου
        if(pages.length>0){
            image = new JLabel(new
ImageIcon(pages[selectedPage-1].getImage()));
            add(new
JScrollPane(image), BorderLayout.CENTER);
        }
    }

    public void actionPerformed(ActionEvent e) {
        if ("forward".equals(e.getActionCommand())) {

```

```
        if(selectedPage<pages.length) {
            selectedPage++;
            pageList.setSelectedIndex(selectedPage-
1);
            image.setIcon(new
ImageIcon(pages[selectedPage-1].getImage()));
        }
        else if ("back".equals(e.getActionCommand())) {
            if(selectedPage>1){
                selectedPage--;

                pageList.setSelectedIndex(selectedPage-1);
                image.setIcon(new
ImageIcon(pages[selectedPage-1].getImage()));
            }
            else{
                JComboBox cb = (JComboBox)e.getSource();
                //Ο χρήστης επέλεξε άλλο κεφάλαιο
                if(cb.getName().equals("pages")){
                    selectedPage=cb.getSelectedIndex()+1;
                    if(selectedPage<1)
                        selectedPage=1;
                    image.setIcon(new
ImageIcon(pages[selectedPage-1].getImage()));
                }
            }
        }
    }
```

```
package org.economics;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JPasswordField;
import javax.swing.JTextArea;
import javax.swing.JTextField;

/**
 * Η κλάση που αρχικοποιεί το πρώτο tab με την εισαγωγή
```

```
*
*/
public class MainPanel extends JPanel implements ActionListener {
    public static JFrame frame = null;

    ApplicationUI ui;
    JTextField username ;
    JPasswordField password;
    JTextArea text;

    public MainPanel(ApplicationUI ui) {
        super(new BorderLayout());

        this.ui=ui;

        //Παρε τα δεδομένα από τη βάση
        Database db = new Database();
        String info = db.getMainInfo();
        db.close();
        // Βάλε το τίτλο του tab
        text = new JTextArea();
        JPanel main = new JPanel();
        //main.setLayout(new BorderLayout());

        text.setEditable(false);
        text.setText(info);

        text.setFont(main.getFont().deriveFont(Font.BOLD).deriveFont((float) 14.0));
        add(text, BorderLayout.NORTH);

        JButton Go = new JButton("Εισαγωγή");
        Go.setActionCommand("go");
        Go.addActionListener(this);
        JButton logout = new JButton("Αποσύνδεση");
        logout.setActionCommand("logout");
        logout.addActionListener(this);
        username = new JTextField(10);
        password = new JPasswordField(10);
        main.add(new JLabel("Όνομα χρήστη"));
        main.add(username);
        main.add(new JLabel("Κωδικός χρήστη"));
        main.add(password);
        add(main, BorderLayout.CENTER);
        main.setBackground(Color.WHITE);
        main.add(Go);
        main.add(logout);

        java.net.URL imageURL = MainPanel.class.getResource("e-learning.jpg");
        add(new JLabel(new ImageIcon(imageURL)), BorderLayout.SOUTH);
    }
}
```

```

    }

    public void actionPerformed(ActionEvent e) {
        if ("go".equals(e.getActionCommand())) {
            Database db = new Database();

            ui.setLogin(db.checkLogin(username.getText(), password.getText()
) );

            ui.setAdmin(db.checkAdmin(username.getText()));
            if(ui.isLogin()){
                ui.setUsername(username.getText());
                String info = db.getMainInfoAfterLogin(
+ username.getText());

                text.setText(info);
                ui.addTab();
            }
            db.close();
        }
        else if ("logout".equals(e.getActionCommand())) {
            if(ui.isLogin()){
                username.setText("");
                password.setText("");
                ui.deleteTabs();
                ui.setLogin(false);
                ui.setAdmin(false);
                Database db = new Database();
                String info = db.getMainInfoAfterLogin()

;

                db.close();
                text.setText(info);
                //ui.setUsername("");
            }
        }
    }
}
}

```

```
package org.economics;
```

```
public class Message {
    String text;
    String from;
    String to;
    String date;
}

```

```
        public Message(String text, String from, String to, String
date) {
            super();
            this.text = text;
            this.from = from;
            this.to = to;
            this.date = date;
        }

        public Message() {
        }

        public String getText() {
            return text;
        }

        public void setText(String text) {
            this.text = text;
        }

        public String getFrom() {
            return from;
        }

        public void setFrom(String from) {
            this.from = from;
        }

        public String getTo() {
            return to;
        }

        public void setTo(String to) {
            this.to = to;
        }

        public String getDate() {
            return date;
        }

        public void setDate(String date) {
            this.date = date;
        }
    }
}
```

```
package org.economics;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Date;

import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JPasswordField;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.JTextField;

import javax.swing.Timer;

/**
 * Η κλάση που αρχικοποιεί το πρώτο tab με την εισαγωγή
 *
 */
public class MessagesPanel extends JPanel implements ActionListener
{
    public static JFrame frame = null;

    ApplicationUI ui;
    JTextField username ;
    JPasswordField password;
    JTextArea reply;
    JTextArea messagesArea;
    Timer timer;

    public MessagesPanel(ApplicationUI ui) {
        super(new GridLayout(3,1));

        this.ui=ui;

        JPanel receivedMessages = new JPanel();
        receivedMessages.setLayout(new BorderLayout());
        //main.setBackground(Color.WHITE);
        add(receivedMessages);

        messagesArea = new JTextArea(10,20);
        messagesArea.setEditable(false);
```



```
        messagesArea.setFont(this.getFont().deriveFont(Font.BOLD).deriveFont((float) 14.0));

        printMessages();
        receivedMessages.add(new
JLabel("Μηνύματα:"), BorderLayout.NORTH);
        receivedMessages.add(new
JScrollPane(messagesArea), BorderLayout.CENTER);

        JButton clear =new JButton("Καθαρισμός μηνυμάτων");
        clear.setActionCommand("clear");
        clear.addActionListener(this);
        receivedMessages.add(clear, BorderLayout.SOUTH);

        JPanel replyPanel = new JPanel();
        replyPanel.setLayout(new BorderLayout());
        replyPanel.add(new JLabel("Νέο
μήνυμα:"), BorderLayout.NORTH);
        reply = new JTextArea(10,10);

        reply.setFont(this.getFont().deriveFont(Font.BOLD).deriveFont((float) 14.0));
        replyPanel.add(new JScrollPane(reply));
        JButton go =new JButton("Αποστολή");
        go.setActionCommand("go");
        go.addActionListener(this);

        JPanel lowPanel = new JPanel();
        lowPanel.add(go);
        replyPanel.add(lowPanel, BorderLayout.SOUTH);
        add(replyPanel);

        //Set up timer to drive animation events.
        timer = new Timer(5000, this);
        timer.setActionCommand("timer");
        timer.setInitialDelay(5000);
        timer.start();
    }

    public void actionPerformed(ActionEvent e) {
        if ("go".equals(e.getActionCommand())) {
            Database db = new Database();
            Message message = new Message();
            message.setFrom(ui.getUsername());
            message.setTo(db.getAdmin());
            message.setText(reply.getText());
            boolean result = db.addMessage(message);
            db.close();
            if(result)
                reply.setText("Το μήνυμα εστάλη!");
        }
    }
}
```

```
    }
    else if ("clear".equals(e.getActionCommand())) {
        Database db = new Database();
        db.clearMessages(ui.getUsername());
        db.close();
        messagesArea.setText("Τα μηνύματα σβήστηκαν");
    }
    else if ("timer".equals(e.getActionCommand())) {
        printMessages();
    }
}

public void printMessages(){
    Database db = new Database();
    Message[] messages = db.getMessages(ui.getUsername());
    db.close();
    String info="";
    for(int i=0;i<messages.length;i++){
        info = info + messages[i].getDate() + "\n";
        info = info + "From:"  + messages[i].getFrom()
+ "\n";
        info = info + messages[i].getText() +
"\n\n\n";
    }
    messagesArea.setText(info);
}

}

package org.economics;

import java.awt.Image;

public class Page {
    int number;
    Image image;

    public Page() {
        super();
    }

    public Page(int number, Image image) {
        super();
        this.number = number;
        this.image = image;
    }

    public int getNumber() {
        return number;
    }
}
```

```
public void setNumber(int number) {
    this.number = number;
}

public Image getImage() {
    return image;
}

public void setImage(Image image) {
    this.image = image;
}
}
```

```
package org.economics;

public class Question {
    int id;
    int categoryID;
    String question;
    int number;
    String type;

    public Question(int id, int categoryID, String question, int
number, String type) {
        super();
        this.id = id;
        this.categoryID = categoryID;
        this.question = question;
        this.number = number;
        this.type = type;
    }

    public Question(){

    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public int getCategoryID() {
        return categoryID;
    }
}
```

```
public void setCategoryID(int categoryID) {
    this.categoryID = categoryID;
}

public String getQuestion() {
    return question;
}

public void setQuestion(String question) {
    this.question = question;
}

public int getNumber() {
    return number;
}

public void setNumber(int number) {
    this.number = number;
}

public String getType() {
    return type;
}

public void setType(String type) {
    this.type = type;
}
}
```

```
package org.economics;

public class QuestionType {

    int count;
    int correctAnswers;

    public QuestionType(int count, int correctAnswers) {
        super();
        this.count = count;
        this.correctAnswers = correctAnswers;
    }

    public QuestionType() {
    }

    public int getCount() {
        return count;
    }
}
```

```
    }  
    public void setCount(int count) {  
        this.count = count;  
    }  
    public int getCorrectAnswers() {  
        return correctAnswers;  
    }  
    public void setCorrectAnswers(int correctAnswers) {  
        this.correctAnswers = correctAnswers;  
    }  
}
```

```
    package org.economics;  
  
    import java.sql.Date;  
  
    public class Test {  
  
        String username;  
        int categoryID;  
        float mark;  
        Date date;  
        int level;  
  
        int[] correctAnswers;  
        int[] wrongAnswers[];  
  
        public Test(String username, int categoryID, float mark, int  
level) {  
            super();  
            this.username = username;  
            this.categoryID = categoryID;  
            this.mark = mark;  
            this.level = level;  
        }  
  
        public Test(String username, int categoryID, float mark, Date  
date , int level) {  
            super();  
            this.username = username;  
            this.categoryID = categoryID;  
            this.mark = mark;  
            this.date = date;  
            this.level = level;  
        }  
    }
```

```
public Test(){  
  
}  
  
public Date getDate() {  
    return date;  
}  
  
public void setDate(Date date) {  
    this.date = date;  
}  
  
public String getUsername() {  
    return username;  
}  
  
public void setUsername(String username) {  
    this.username = username;  
}  
  
public int getCategoryID() {  
    return categoryID;  
}  
  
public void setCategoryID(int categoryID) {  
    this.categoryID = categoryID;  
}  
  
public float getMark() {  
    return mark;  
}  
  
public void setMark(float mark) {  
    this.mark = mark;  
}  
  
public int getLevel() {  
    return level;  
}  
  
public void setLevel(int level) {  
    this.level = level;  
}  
  
}
```