



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Εκπαιδευτικό Λογισμικό εκμάθησης της ελληνικής μυθολογίας
Όνοματεπώνυμο Φοιτητή	Χριστίνα Παπαμιχαήλ
Πατρώνυμο	Γεώργιος
Αριθμός Μητρώου	ΜΠΣΠ/ 10023
Επιβλέπων	Μαρία Βίρβου, Καθηγήτρια

Ηερομηγία Παράδοσης **Ιούνιος 2012**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Μαρία Βίρβου
Καθηγήτρια

Γεώργιος Τσιχριτζής
Καθηγητής

Ευάγγελος Φούντας
Καθηγητής

*Αφιερωμένη
στους γονείς μου...*

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά την καθηγήτρια και επιβλέπουσα αυτής της εργασίας Μαρία Βίρβου για όλη τη στήριξη, βοήθεια και καθοδήγηση κατά τη διάρκεια εκτέλεσης της διπλωματικής αυτής εργασίας.

Επίσης την συμφοιτήριά μου Παναγοπούλου Ίλια για την συμπαράσταση και καλή συνεργασία για τη διεκπεραίωση της εργασίας.

Περίληψη

Η παρούσα διπλωματική εργασία αποτελεί ένα εκπαιδευτικό λογισμικό για την εκμάθηση της ελληνικής μυθολογίας σε μαθητές. Βασικό στοιχείο όλης της εφαρμογής αποτελεί η προσαρμοστικότητα του λογισμικού στις ανάγκες του χρήστη. Ο χρήστης εφόσον περάσει το τεστ, μπορεί να διαβάσει όλο το διαθέσιμο υλικό του επόμενου κεφαλαίου καθώς επίσης και να διεξάγει κάποιο τεστ. Τόσο κατά τη διάρκεια της ανάγνωσης όσο και της εξέτασης το σύστημα καθοδηγεί το μαθητή στη συγκέντρωση έτσι ώστε να πετύχει καλύτερα αποτελέσματα. Υπάρχει δυνατότητα για την αποθήκευση τόσο των αποτελεσμάτων του χρήστη όσο και του χρόνου μελέτης αλλά και των κινήσεων που λαμβάνουν χώρα από τον ίδιο και τον καθιστούν αναποφάσιστο. Στόχος είναι η όσο το δυνατόν πιο σωστή ενημέρωση σχετικά με την απόδοση του χρήστη μέσω μηνυμάτων που εμφανίζονται κατά την ολοκλήρωση του τεστ. Μετά την διεξαγωγή του τεστ το σύστημα γίνεται ιδιαιτέρως προσαρμοστικό αφού παροτρύνει το χρήστη ή τον αποτρέπει στη συνέχιση του επόμενου κεφαλαίου εμφανίζοντας συγκεκριμένα μηνύματα αλλά και εικονίδια.

Abstract

This thesis is an educational software which leads students to learn the greek mythology. The adaptability of the software for the students' needs is one of the outstanding elements. If students succeed the test, they can learn the available manual and do the next one. The software guides students to be concentrated not only on studying but also on doing a test. The database stores the results of tests, the time that students spent on studying and actions of undecided students. The target of the thesis is the best information for students' performance through different messages. The software becomes more adaptive after a test because it urges or disinclines students on continuing test through messages and icons

Πίνακας Περιεχομένων

Εισαγωγή	10
1. Εκπαιδευτικά συστήματα διδασκαλίας.....	11
1.1 Τα Εκπαιδευτικά Συστήματα διδασκαλίας και η επιρροή τους στο ευρύ κοινό	11
1.2 Προσαρμοστικά Υπερμέσα (Adaptive Hypermedia).....	11
1.3 Προσαρμοστικότητα και Προσαρμοστικά Υπερμέσα.....	12
1.4 Μάθηση και διδασκαλία	12
1.5 Μεταγνωστικές δεξιότητες	13
1.6 Η γενικότητα εναντίον ειδικού τομέα των μεταγνωστικών δεξιοτήτων.	13
1.6.1 Νοητική ικανότητα και μεταγνωστικές επιδεξιότητες.....	14
2. Υλοποίηση Εργασίας	15
2.1 Περιγραφή Εφαρμογής	15
2.2 Ανάλυση Απαιτήσεων.	15
2.3 Δυνατότητες της εφαρμογής	15
3. Βάση δεδομένων.....	17
3.1 AssistImage.....	17
3.2 Questions.....	18
3.3 Reading.....	19
3.4 Results.....	19
3.5 User_Profile	21
3.6 Sqlqueries.cs	21
4. Υλοποίηση της εφαρμογής	23
4.1 Φόρμα εισαγωγής του χρήστη - loginuser.cs.....	23
4.2 Δημιουργία Χρήστη - createuser.cs.....	24
4.3 Δημιουργία κωδικού πρόσβασης - createnewpassword.cs.....	28

4.4 Κεντρική φόρμα.....	28
4.5 Μελέτη χρήστη	35
4.6 Επιλογή θεού.....	37
4.7 Εμφάνιση λογαριασμού - myaccount.cs	38
4.8 Αναλυτικά στοιχεία - Stats.cs	39
4.9 Έλεγχος γνώσεων - FormTests.cs	40
4.9.1 Υπολογισμός χρόνου του τεστ	40
4.9.2 Επίπεδα δυσκολίας	44
4.9.3 Εμφάνιση ερωτήσεων	45
4.9.4 Έλεγχος ορθότητας της απάντησης και μετάβαση στην επόμενη ερώτηση.	51
4.9.5 Υπολογισμός βαθμολογίας.....	54
4.9.6 Ολοκλήρωση του τεστ	55
4.10 Εμφάνιση αποτελεσμάτων στον χρήστη (Analytics.cs)	56
4.11 Προσαρμοστική πλοήγηση σε σχέση με τον χρόνο, τη βαθμολογία και το διάβασμα.	
4.12 Μαντείο των Δελφών.....	60
5 Εγχειρίδιο χρήστη.....	62
5.1 Είσοδος χρήστη.....	62
5.2 Δημιουργία λογαριασμού	62
5.3 Αλλαγή κωδικού	65
5.4 Είσοδος στην εφαρμογή	65
5.5 Ανάγνωση διαθέσιμου υλικού.....	66
5.6 Νέο τεστ	68
5.7 Εμφάνιση αποτελεσμάτων	70
5.8 Αλλαγή βοηθού.....	73
5.9 Αναλυτικά στοιχεία.....	75
5.10 Ο λογαριασμός μου.....	76

5.11 Βοήθεια χρήστη	77
5.12 Μαντείο Δελφών	78
Συμπεράσματα.....	80
Βιβλιογραφία.....	81

Εισαγωγή

Στην παρούσα διπλωματική εργασία γίνεται αναφορά στην εκπαίδευση μέσω του ηλεκτρονικού υπολογιστή. Στο πρώτο μέρος της εργασίας παρέχονται γενικές πληροφορίες σχετικά με τα εκπαιδευτικά συστήματα διδασκαλίας καθώς επίσης και στις σύγχρονες μεθόδους που κάνουν ένα λογισμικό πιο ελκυστικό και προσαρμοστικό στο χρήστη.

Στο δεύτερο μέρος της διπλωματικής αυτής εργασίας περιγράφεται ο στόχος ανάλυσης της εφαρμογής που έχουμε δημιουργήσει, καθώς επίσης το εγχειρίδιο του προγραμματιστή αλλά και του χρήστη. Η εφαρμογή μας αποτελεί ένα προσαρμοστικό εκπαιδευτικό λογισμικό με στόχο την εκμάθηση της ελληνικής μυθολογίας σε μαθητές.

1. Εκπαιδευτικά συστήματα διδασκαλίας

1.1 Τα Εκπαιδευτικά Συστήματα διδασκαλίας και η επιρροή τους στο ευρύ κοινό

Η τάση του μαθητή στη χρήση των Ευφυή συστημάτων διδασκαλίας (Intelligent Tutoring Systems) αποτελεί αντικείμενο μελέτης μεγάλου ενδιαφέροντος, στον τρόπο με τον οποίο ο ίδιος αλληλεπιδρά με το λογισμικό. Τα διδακτικά συστήματα με την ανάπτυξη της τεχνολογίας και την αύξηση των δυνατοτήτων ενός λογισμικού μαθητή έχουν γίνει εξαιρετικά αποτελεσματικά στην εκτίμηση των δεξιοτήτων ενός. Ωστόσο όμως δεν έχουν καταφέρει να παραγκωνίσουν τα κινητήρια προβλήματα που μαστίζουν τις παραδοσιακές μαθητικές τάξεις. Έρευνες έχουν δείξει όμως ότι μαθητές δείχνουν μεγαλύτερη προσοχή και προθυμία στα Εκπαιδευτικά Συστήματα Διδασκαλίας. Ως εκ τούτου οι ευφυείς διδάσκαλοι μπορούν να ανταποκριθούν το ίδιο με τους παραδοσιακούς δασκάλους στη γνωστική ικανότητα των μαθητών αλλά και στην αντίληψη, με αποτέλεσμα να κάνει ακόμη πιο αποτελεσματικά τα εκπαιδευτικά συστήματα διδασκαλίας.

Η ανάπτυξη ευφύων καθηγητών οι οποίοι μπορούν να προσαρμοστούν έτσι ώστε οι μαθητές να έχουν κίνητρο εξαρτάται από τη δημιουργία σωστών εργαλείων κατά την ανάπτυξη του λογισμικού.

Μια εναλλακτική προσέγγιση της εκπαιδευτικής μάθησης εστιάζει στην παροχή κινήτρων ως παράγοντα κινήτρων που επηρεάζει τη μάθηση, καθότι η βελτίωση των κινήτρων θεωρείται ως το κύριο μέσο για τη βελτίωση της μάθησης. Ένα βασικό κίνητρο το οποίο σχετίζεται με τη συμπεριφορά του μαθητή είναι η κατάχρηση της βοήθειας, όπου ένας μαθητής ζητά επανειλημμένα βοήθεια από τον δάσκαλό του, με αποτέλεσμα ο δάσκαλος να αποκαλύπτει εν τέλει στο μαθητή τη σωστή απάντηση χωρίς ο ίδιος να έχει προσπαθήσει να λύσει το πρόβλημα.

1.2 Προσαρμοστικά Υπερμέσα (Adaptive Hypermedia)

Τα Προσαρμοστικά Υπερμέσα¹ αποτελούν μια νέα σχετικά κατεύθυνση της έρευνας στο σταυροδρόμι των υπερμέσων και της μοντελοποίησης χρήστη. Ένας από τους βασικούς περιορισμούς των παραδοσιακών εφαρμογών πολυμέσων είναι ότι παρέχουν κοινό υλικό, περιεχόμενο όπως και σύνολο συνδέσεων σε όλους τους χρήστες. Ένα παραδοσιακό εκπαιδευτικό σύστημα υπερμέσων παρουσιάζει την ίδια εξήγηση και το ίδιο υλικό στους μαθητές οι οποίοι έχουν διαφορετικούς εκπαιδευτικούς στόχους και διαφορετική γνώση αντικειμένου.

Τα προσαρμοστικά υπερμέσα είναι μία εξελιγμένη εναλλακτική στην παραδοσιακά συστήματα υπερμέσων. Στοχεύουν στην οικοδόμηση ενός μοντέλου που αποτελείται από τους στόχους, τις προτιμήσεις και τις γνώσεις του κάθε χρήστη χωριστά, καθώς επίσης στη χρησιμότητα αυτού του μοντέλου για την αλληλεπίδραση του κάθε χρήστη, με τελικό σκοπό

¹ Adaptive Hypermedia: From Intelligent Tutoring Systems to Web-Based Education Peter Brusilovsky
Εκπαιδευτικό Λογισμικό εκμάθησης της ελληνικής μυθολογίας

στην προσαρμοστικότητα του μοντέλου στις ανάγκες του κάθε χρήστη ως ξεχωριστή οντότητα. Σε ένα τέτοιο σύστημα θα δοθεί μία παρουσίαση η οποία θα έχει προσαρμοστεί ειδικά στις ανάγκες του χρήστη, παραχωρώντας ειδικούς συνδέσμους οι οποίοι θα βοηθήσουν τον χρήστη να προχωρήσει περαιτέρω.

1.3 Προσαρμοστικότητα και Προσαρμοστικά Υπερμέσα.

Ένα κλασικό αρχείο συνδέσμων αποτελείται από ένα σύνολο σελίδων οι οποίες συνδέονται μεταξύ τους με δεσμούς. Στα συστήματα υπερμέσων περιλαμβάνονται ειδικά εργαλεία πλοήγησης, όπως πίνακας περιεχομένων, ευρετήριο και διάφοροι χάρτες, τα οποία χρησιμοποιούνται σε όλες τις σελίδες για πλοήγηση. Τόσο η σελίδα, όσον αφορά την περιεκτικότητα, όσο η εμφάνιση και η συμπεριφορά των συνδέσμων είναι δύο βασικές λειτουργίες οι οποίες εύκολα μπορούν να προσαρμοστούν. Πρωταρχικός στόχος αυτής της προσαρμοστικής παρουσίασης αποτελεί η προσαρμογή του περιεχομένου της σελίδας με τις γνώσεις, τους στόχους αλλά και τις πληροφορίες είναι αποθηκευμένες στο μοντέλο του κάθε χρήστη. Ο παραπάνω στόχος εκπληρώνεται με έναν από τους δύο ακόλουθους τρόπους.

- Άμεση καθοδήγηση: Στο σύστημα παρουσιάζεται ένας από τους συνδέσμους ως ο καλύτερος σύνδεσμος ή επίσης μπορεί να δημιουργηθεί μία δεύτερη σύνδεση ως η επόμενη καλύτερη επιλογή.
- Απόκρυψη συνδέσμων: Το σύστημα «καταλαβαίνει» και προσπαθεί να αποκλείσει από τον χρήστη συνδέσμους οι οποίοι δεν έχουν καμία σημασία για τον ίδιο την παρούσα χρονική στιγμή.
- Σύνδεσμος σχολίου: Το σύστημα παράγει αρκετά σχόλιο σε κάθε σύνδεσμο, τα οποία θα εμφανίζουν αρκετές πληροφορίες για τις σελίδες στις οποίες επρόκειτο να μεταβεί ο χρήστης.

1.4 Μάθηση και διδασκαλία

Η γνωστική επιστήμη και η εκπαιδευτική λογοτεχνία υποστηρίζουν ότι η διδασκαλία προς τους άλλους είναι ένας ισχυρός τρόπος για να μάθει ο ίδιος ο διδάσκων. Πολλές έρευνες στην αμοιβαία διδασκαλία, στην υποβοηθούμενη διδασκαλία, στον προγραμματισμό και σε μικρές ομάδες αλληλεπίδρασης υπαινίσσονται την πιθανή μάθηση μέσα από τη διδασκαλία. Οι Bargh και Schull² (1980) διαπίστωσαν ότι οι άνθρωποι που προετοιμάζονται να διδάξουν τους άλλους μαθαίνουν καλύτερα το αντικείμενο από ότι έκαναν αν το διάβαζαν μόνο για τον εαυτό τους. Επίσης η βιβλιογραφία έχει δείξει ότι οι διδάσκοντες επωφελούνται τόσο από την διδασκαλία όσο και από τους διδασκόμενους. Πολλοί φοιτητές έχουν αφοσιωθεί στην σαφή εννοιολογική οργάνωση των υλικών. Πέραν της προετοιμασίας για διδασκαλία, στην πραγματικότητα μπορεί να αξιοποιήσει τις κρίσιμες πτυχές των μαθησιακών αλληλεπιδράσεων.

² Gautam Biswas and Krittaya Leelawong & Department of EECs and ISIS, Vanderbilt University, Nashville, TN, USA
Εκπαιδευτικό Λογισμικό εκμάθησης της ελληνικής μυθολογίας

Ερευνητικά έργα τα οποία δίνουν έμφαση στην εκμάθηση μέσω του προγραμματισμού μοιράζονται ομοιότητες της μάθησης από τη διδασκαλία στους μαθητές που ενημερώνουν τους υπολογιστές να αναλάβουν συγκεκριμένες ενέργειες. Αλλά αυτές οι προσεγγίσεις δεν έχουν κεφαλαιοποιηθεί πλήρως στην μεταφορά πράκτορα και στο πλήθος του κοινωνικού σχήματος που παρέχεται.

1.5 Μεταγνωστικές δεξιότητες

Η μεταγνώση είναι η πτυχή της κριτικής σκέψης, που περιλαμβάνει τις ικανότητες του μαθητή για να αναπτύξουν μία στρατηγική κατά τη διάρκεια της επίλυσης των προβλημάτων και να εξετάσει και να αξιολογήσει την παραγωγικότητα του στη σκέψη. Άνθρωποι που έχουν ασχοληθεί με τη μεταγνώση θεωρούν τη διάθεση στη ερώτηση ή την αμφιβολία, καθώς και τον διπλό έλεγχο και τη δοκιμή, ως τον κύριο άξονα της κριτικής σκέψης. Οι Haller, Child and Walberg³ (1988) συνόψισαν την ουσία της μεταγνώσης, περιγράφοντας τρεις ομάδες δραστηριοτήτων:

- a) Την ευαισθητοποίηση (αναγνώριση κάποιου έμμεση ή άμεση ενημέρωση)
- b) Την παρακολούθηση
- c) Την ρύθμιση (σύγκριση των πιο πιθανών λύσεων για την επίλυση των προβλημάτων)

Οι εκπαιδευτικοί προωθούν τη μεταγνώση για την αυτονομία του μαθητή καθώς και την μαθησιακή απόκτηση του περιεχομένου. Η τόνωση της διαδικασίας της μαθησιακής σκέψης παρέχει στους μαθητές μία αίσθηση της ικανότητας στην εκμάθηση της μάθησης. Κατά συνέπεια, ο δάσκαλος θα πρέπει να επικεντρωθεί στην ανάπτυξη στις ικανότητες των μαθητών για την επίλυση των προβλημάτων και στην εμπιστοσύνη των μαθητών για τις ικανότητες της επίλυσης των προβλημάτων. Τελικά όταν οι μαθητές γίνονται πιο γνώριμοι με τις διαδικασίες που χρησιμοποιούν, και όταν μαθαίνουν τον έλεγχο αυτών των γνωστικών διαδικασιών αυξάνεται η δεξιότητα στην επίλυση των προβλημάτων.

1.6 Η γενικότητα εναντίον ειδικού τομέα των μεταγνωστικών δεξιοτήτων.

Ένα αρκετά κρίσιμο ζήτημα είναι αν οι μεταγνωστικές δεξιότητες είναι βασισμένες σε συγκεκριμένες δεξιότητες ή ως γενικές δεξιότητες ανεξαρτήτου των τομέα των δεξιοτήτων. Σύμφωνα με το μοντέλο της νοημοσύνης οι μεταγνωστικές δεξιότητες υπερβαίνουν τους τομείς ως μέρος της νοημοσύνης. Οι μεταγνωστικές δεξιότητες θα πρέπει να ανήκουν σε γενικευμένους ή ειδικούς τομείς σε ορισμένο βαθμό. Μπορούν να βρεθούν διαφορετικά πρότυπα μέσα από μία αναπτυξιακή προοπτική. Οι μεταγνωστικές δεξιότητες μπορεί να ξεκινήσουν από ένα γενικό τομέα και να καταλήξουν σε συγκεκριμένους τομείς. Η μονοτονική υπόθεση της ανάπτυξης δεν θέτει κανένα περιορισμό στη γενικότητα ή ειδικότητα του τομέα των μεταγνωστικών δεξιοτήτων.

³ Teaching & Teacher Educmion, Vol. 8, No. 2, pp. 109-121. 1992

1.6.1 Νοητική ικανότητα και μεταγνωστικές επιδεξιότητες.

Υπάρχουν τρία εναλλακτικά μοντέλα για την περιγραφή της σχέσης ανάμεσα στην πνευματική ικανότητα και στις μεταγνωστικές επιδεξιότητες ως προάγγελοι τις μάθησης. Το πρώτο μοντέλο αφορά τη μεταγνωστική δεξιότητα ως εκδήλωση της διανοητικής ικανότητας, είτε ως αναπόσπαστο μέλος της πνευματικής εργαλειοθήκης. Σύμφωνα με αυτό το μοντέλο νοημοσύνης, οι μεταγνωστικές δεξιότητες δεν μπορούν να έχουν προγνωστική αξία για μάθηση, ανεξάρτητα από την πνευματική ικανότητα. Αρκετοί ερευνητές ανέφεραν σημαντικές διαφορές στη χρήση μεταγνωστικών στρατηγικών μεταξύ φοιτητών υψηλής αυτών μέσης διανόησης. Οι διαφορές αυτές ήταν παρ' όλα αυτά μικρές και δεν είχαν ληφθεί με συνέπεια για όλες τις μεταγνωστικές στρατηγικές.

Σε ένα άλλο μοντέλο η νοητική ικανότητα και η μεταγνωστική επιδεξιότητα θεωρούνται εντελώς ανεξάρτητοι προγνωστικοί δείκτες της μάθησης. Πράγματι οι μελέτες έδειξαν ότι η μεταγνώση ήταν εν μέρει ανεξάρτητη της νοημοσύνης.

Το τελευταίο μοντέλο είναι ανάμεικτο. Σύμφωνα με αυτό το μοντέλο η μεταγνωστική επιδεξιότητα σχετίζεται με τη διανοητική ικανότητα σε κάποιο βαθμό. Αυτό το μεικτό μοντέλο έχει επιβεβαιωθεί από πολλές μελέτες, είτε με προσομοιώσεις σε ηλεκτρονικό υπολογιστή στους τομείς της ηλεκτρικής ενέργειας και στην ψυχολογία.

2. Υλοποίηση Εργασίας

2.1 Περιγραφή Εφαρμογής

Η εφαρμογή η οποία σχεδιάστηκε για το σκοπό της διπλωματικής εργασίας του μεταπτυχιακού προγράμματος «Προηγμένα Συστήματα Πληροφορικής» αποτελεί ένα λογισμικό εκμάθησης της Ελληνικής μυθολογίας και απευθύνεται κυρίως σε παιδιά. Ο χρήστης (μαθητής) μέσα από υλικό μελέτης που του δίνεται, αποκτά γνώσεις για την Ελληνική μυθολογία. Με την βοήθεια των τεστ που του παρέχονται σε κάθε ενότητα γίνεται έλεγχος των γνώσεων. Το όνομα που δόθηκε στην εφαρμογή είναι “Olympus 12”. Για την υλοποίηση του εκπαιδευτικού λογισμικού χρησιμοποιήθηκε το πρόγραμμα “Microsoft Visual Studio 2010” και το “Microsoft SQL Server 2008 R2”. Το λογισμικό σχεδιάστηκε μετά από μελέτη που έγινε και αναφέρθηκε παραπάνω έτσι ώστε να προσαρμόζεται στις ανάγκες του μαθητή, να είναι εύχρηστο ελκυστικό και να επιτυγχάνει το σκοπό του που είναι η μετάδοση της γνώσης στον μαθητή.

2.2 Ανάλυση Απαιτήσεων.

Πριν ξεκινήσει ο σχεδιασμός της εφαρμογής, έγινε καταγραφή των απαιτήσεων έτσι ώστε να καθοριστεί τόσο η δομή του λογισμικού όσο και οι στόχοι. Σημαντικό ρόλο στη λήψη των αποφάσεων για την καταγραφή των απαιτήσεων έπαιξε το γεγονός ότι η εφαρμογή απευθύνεται κυρίως σε παιδιά. Αυτό σημαίνει ότι τα παιδιά ως αρχάριοι χρήστες μπορεί να δυσκολεύονται στη χρήση του υπολογιστή, με αποτέλεσμα ο προγραμματιστής να προσπαθεί να δημιουργήσει ένα λογισμικό όσο το δυνατό πιο φιλικό για το χρήστη. Ο μαθητής έχει την ικανότητα να αποτυπώνει τη γεωγραφία της οθόνης, όπως την στοίχιση των κουμπιών, τη θέση και τη μορφή των εικονιδίων. Αν ο τρόπος με τον οποίο το περιεχόμενο της εφαρμογής αλλάζει συνεχώς, δυσχεραίνει το χρήστη στην επίτευξη του στόχου του που είναι η εκμάθηση. Σε συνέχεια όσων έχουν αναφερθεί παραπάνω δεν μπορούσαμε να παραλείψουμε την προσαρμοστικότητα της εφαρμογής μας στις ανάγκες του εκάστοτε χρήστη.

Οι απαιτήσεις ενός συστήματος μπορούν να χωριστούν σε λειτουργικές και μη λειτουργικές. Λειτουργικές χαρακτηρίζονται οι απαιτήσεις μέσα από τις οποίες το λογισμικό αντιδρά και συμπεριφέρεται σε συγκεκριμένες καταστάσεις. Οι μη λειτουργικές απαιτήσεις είναι αυτές που περιλαμβάνουν χρονικούς περιορισμούς, περιορισμούς αξιοπιστίας και απόδοσης.

2.3 Δυνατότητες της εφαρμογής

Η εφαρμογή μας “Olympus 12” παρέχει στο χρήστη τις παρακάτω δυνατότητες:

- Δημιουργία λογαριασμού χρήστη (user account)
- Αλλαγή κωδικού πρόσβασης (password)
- Είσοδος στην εφαρμογή
- Ανάγνωση θεωρίας
- Διεξαγωγή τεστ

- Εμφάνιση των στοιχείων του προφίλ του χρήστη
- Καταχώρηση αποτελεσμάτων
- Εμφάνιση συμβουλής
- Προσαρμοστικότητα της εφαρμογής στις ανάγκες του εκάστοτε χρήστη
- Προσαρμογή της εφαρμογής βάσει της ηλικίας του χρήστη
- Προσαρμογή της εφαρμογής βάσει του φύλου του χρήστη
- Προσαρμογή της εφαρμογής βάσει της απόδοσης του χρήστη

3. Βάση δεδομένων.

Η βάση που χρησιμοποιήθηκε για την εφαρμογή είναι η greekmythology και το περιβάλλον υλοποίησης είναι Microsoft SQL Server Management Studio . Αποτελείται από τους έξι ακόλουθους πίνακες.

1. AssistImage
2. questions
3. Reading
4. Results
5. User_Profile

3.1 AssistImage

Στον πίνακα (table) dbo.AssistImage αποθηκεύονται όλα τα απαραίτητα στοιχεία για το βοηθό που μπορεί να επιλέξει ο χρήστης. Αποτελείται από τέσσερις στήλες (columns) όπως περιγράφεται πιο αναλυτικά παρακάτω.

- assistant: Αποτελεί το ID του βοηθού ξεκινώντας από την τιμή 1 και αυξάνοντας κατά 1 μονάδα.
- nameAssist: Αποθηκεύει το όνομα του κάθε βοηθού που έχει τη δυνατότητα να επιλέξει ο χρήστης.
- assistPhoto: Αποθηκεύει το όνομα της φωτογραφίας του κάθε βοηθού που εμφανίζεται στην εφαρμογή.
- colorAssist: Αποθηκεύει το χρώμα ανάλογα με την εικόνα του κάθε βοηθού το οποίο χρησιμοποιείται και ως background της εφαρμογής ανάλογα με την επιλογή του βοηθού.

	assistant	nameAssist	assistPhoto	colorAssist
	1	Δίας	The_Gods__Ze...	LightBlue
	2	Ήρα	0d20682ea057b...	MediumPurple
	3	Αθηνά	The_Gods__At...	Thistle
	4	Άρτεμις	The_Gods__Ar...	DarkSlateBlue
	5	Αφροδίτη	The_Gods__Ap...	LightPink
	6	Απόλλωνας	The_Gods__Ap...	Bisque
	7	Ποσειδώνας	Poseidon.jpg	Teal
	8	Ήφαιστος	Efesto.jpg	BurlyWood
	9	Δήμητρα	The_Gods__De...	PaleGreen
	10	Άρης	The_Gods__Ar...	IndianRed
	11	Άδης	Hades.jpg	GainsBoro
	12	Ερμής	The_Gods__He...	LightSteelBlue
*	NULL	NULL	NULL	NULL

Εικ. 3.1 AssistImage

3.2 Questions

Στον πίνακα (table) questions είναι καταχωρημένες όλες οι ερωτήσεις οι οποίες εμφανίζονται randomly σε κάθε τεστ που επιλέγει ο μαθητής. Αποτελείται από τις εννέα ακόλουθες στήλες

- id_q: Αποτελεί το πρωτεύον κλειδί (primary key) και αποθηκεύει το id της εκάστοτε ερώτησης.
- question: Είναι καταχωρημένες όλες οι ερωτήσεις που εμφανίζονται τυχαία σε κάθε τεστ.
- answer_right: Είναι καταχωρημένες όλες οι σωστές απαντήσεις.
- answer_wrong1: Είναι καταχωρημένη η πρώτη σωστή απάντηση.
- answer_wrong2: Είναι καταχωρημένη η δεύτερη σωστή απάντηση.
- answer_wrong3: Είναι καταχωρημένη η τρίτη σωστή απάντηση.
- level1: Καταχωρείται το επίπεδο δυσκολίας της εκάστοτε ερώτησης .
- chapter: Καταχωρείται το κεφάλαιο στο οποίο ανήκει η κάθε ερώτηση.
- typeq: Έχουμε τρεις διαφορετικούς τύπους ερωτήσεων. Στο gw ανήκουν οι ερωτήσεις Σωστού- Λάθους. Στο fb ανήκουν οι ερωτήσεις συμπλήρωσης κενού. Στο mc ανήκουν οι ερωτήσεις οι ερωτήσεις πολλαπλής επιλογής.

id_q	question	answer_right	answer_wrong1	answer_wrong2	answer_wrong3	level1	chapter
1	Τα δύοθεκα παιδι...	Σωστό	Λάθος	ΜΕΛΛ	ΜΕΛΛ	1	1
2	Ο Κρόνος παντρ...	Λάθος	Σωστό	ΜΕΛΛ	ΜΕΛΛ	1	1
3	Η Τιτανομαχία ή...	τον Δία και τα α...	τον Δία	τον Δία και την ...	τον Δία και την ...	1	1
4	Το βουνό στο οπ...	ΟΛΥΜΠΟΣ	ΜΕΛΛ	ΜΕΛΛ	ΜΕΛΛ	1	1
5	Στην Τιτανομαχί...	οι Ολύμπιοι Θεοί	οι Τιτάνες	Οι Εκατόγχερες	ο Δίας	1	1
6	Οι Θεοί του Ολύ...	12	10	8	6	1	1
7	Η Ακροδίκη ήτα...	του έρωτα	του κυνηγιού	της σοφίας	του Πολέμου	1	1
8	Ο Κρόνος είχε κ...	ΕΣΤΙΑ	ΜΕΛΛ	ΜΕΛΛ	ΜΕΛΛ	2	1
9	Ο Άρης ήταν ο ...	του πολέμου	του κυνηγιού	του έρωτα	της φωτιάς	1	1
10	Ο Άδης είναι γν...	Πλούτωνας	Άρης	Απόλλωνας	Ερμής	2	1
11	Η Πυθία:	Έδινε χρησμούς	Ήταν θεά	Ήταν μούσα	Ήταν νύμφη	2	1
12	Οι επισκέπτες πο...	Στην Κασταλία ν...	Στον Όλυμπο	Σε μια πηγή κον...	Στην Αρχαία Ολ...	2	1
13	επισκέπτες που ζ...	Σωστό	Λάθος	ΜΕΛΛ	ΜΕΛΛ	3	1
14	Οι χρησμοί δίνον...	Λάθος	Σωστό	ΜΕΛΛ	ΜΕΛΛ	2	1
15	Η ιασά ήταν η θ...	Σωστό	Λάθος	ΜΕΛΛ	ΜΕΛΛ	1	1
16	Οι Εκατόγχερες ...	Λάθος	Σωστό	ΜΕΛΛ	ΜΕΛΛ	3	1
17	Ο Κρόνος είχε ε...	Σωστό	Λάθος	ΜΕΛΛ	ΜΕΛΛ	2	1
18	Ο κατασκευαστή...	Ήρακλος	Απόλλωνας	Διώνυσος	ο Άρης	1	1
19	Αρχικά οι χρησμ...	Σωστό	Λάθος	ΜΕΛΛ	ΜΕΛΛ	3	1
20	Ο Ζευς άφησε δ...	Λάθος	Σωστό	ΜΕΛΛ	ΜΕΛΛ	3	1
21	Ο φόρος που πλ...	Σωστό	Λάθος	ΜΕΛΛ	ΜΕΛΛ	3	1

Εικ. 3.2 Questions

3.3 Reading

Στον πίνακα (table) `dbo.reading` αποθηκεύονται όλα τα απαραίτητα στοιχεία που σχετίζονται με το διάβασμα του χρήστη. Αποτελείται από τις έξι ακόλουθες στήλες (columns).

- `ID`: Αποτελεί το ID του χρήστη.
- `chapter`: Αποθηκεύει το κεφάλαιο που έχει διαβάσει ο χρήστης.
- `ReadingMin`: Αποθηκεύει τα συνολικά λεπτά που έχει αφιερώσει ο χρήστης στο διάβασμα του τεστ την τελευταία φορά.
- `ReadingSec`: Αποθηκεύει τα συνολικά δευτερόλεπτα που έχει αφιερώσει ο χρήστης στο διάβασμα του τεστ την τελευταία φορά.
- `LastRead`: Αποθηκεύει την ημερομηνία καθώς και ώρα που ο χρήστης έχει διαβάσει το διαθέσιμο υλικό για το συγκεκριμένο κεφάλαιο.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'greekmythology', including the 'dbo.reading' table. The main window shows the table's data with the following columns: ID, chapter, ReadingMin, ReadingSec, and LastRead. The data is as follows:

ID	chapter	ReadingMin	ReadingSec	LastRead
8	1	0	0	3/8/2012 1:53:4...
9	1	5	5	3/8/2012 7:07:1...
12	1	3	35	5/19/2012 12:0...
12	2	0	45	5/19/2012 12:0...
13	1	5	49	5/19/2012 5:49:...
13	1	1	8	5/19/2012 5:51:...
13	1	0	0	5/20/2012 10:3...
13	1	0	1	5/20/2012 10:5...
13	1	3	52	5/20/2012 10:5...
14	1	0	42	5/29/2012 10:3...
*	NULL	NULL	NULL	NULL

Εικ. 3.3 Reading

3.4 Results

Στον πίνακα (table) `dbo.results` αποθηκεύονται τα αποτελέσματα μετά την ολοκλήρωση του κάθε τεστ. Αποτελείται από τις δώδεκα παρακάτω στήλες (columns)

- `ID`: Αποθηκεύεται το id του χρήστη.
- `chapter`: Αποθηκεύεται το κεφάλαιο στο οποίο αναφέρεται το τεστ που έχει ολοκληρώσει ο χρήστης.

- rate: Αποθηκεύεται η βαθμολογία που έχει πάρει ο μαθητής στο συγκεκριμένο τεστ.
- datetest: Αποθηκεύεται η ώρα και η ημερομηνία που ολοκληρώθηκε το συγκεκριμένο τεστ.
- timetest: Αποθηκεύεται ο χρόνος στον οποίο ο χρήστης ολοκλήρωσε το τεστ.
- pass: Καταχωρείται 1 αν ο χρήστης έχει περάσει το τεστ και 0 αν έχει αποτύχει.
- completetest: Καταχωρείται 1 αν ο χρήστης έχει ολοκληρώσει το τεστ μέσα στο χρονικό όριο που έχει οριστεί και 0 δεν προλάβει να ολοκληρώσει το τεστ.
- radiochanges: Αποθηκεύονται οι αλλαγές που κάνει ο χρήστης στα radiobuttons.
- backspaces: Αποθηκεύονται οι φορές που πληκτρολογεί ο χρήστης το κουμπί backspace στις ερωτήσεις συμπλήρωσης κενού "fb".
- mousemoves: Αποθηκεύει τις φορές που ο χρήστης μετακινεί το ποντίκι του.
- totalmins: Αποθηκεύει τα συνολικά λεπτά που έχει αφιερώσει ο χρήστης σε κάθε τεστ.
- totalsecs: Αποθηκεύει τα συνολικά δευτερόλεπτα που έχει αφιερώσει ο χρήστης σε κάθε τεστ.

The screenshot shows the Microsoft SQL Server Management Studio interface. The main window displays a query result table with the following data:

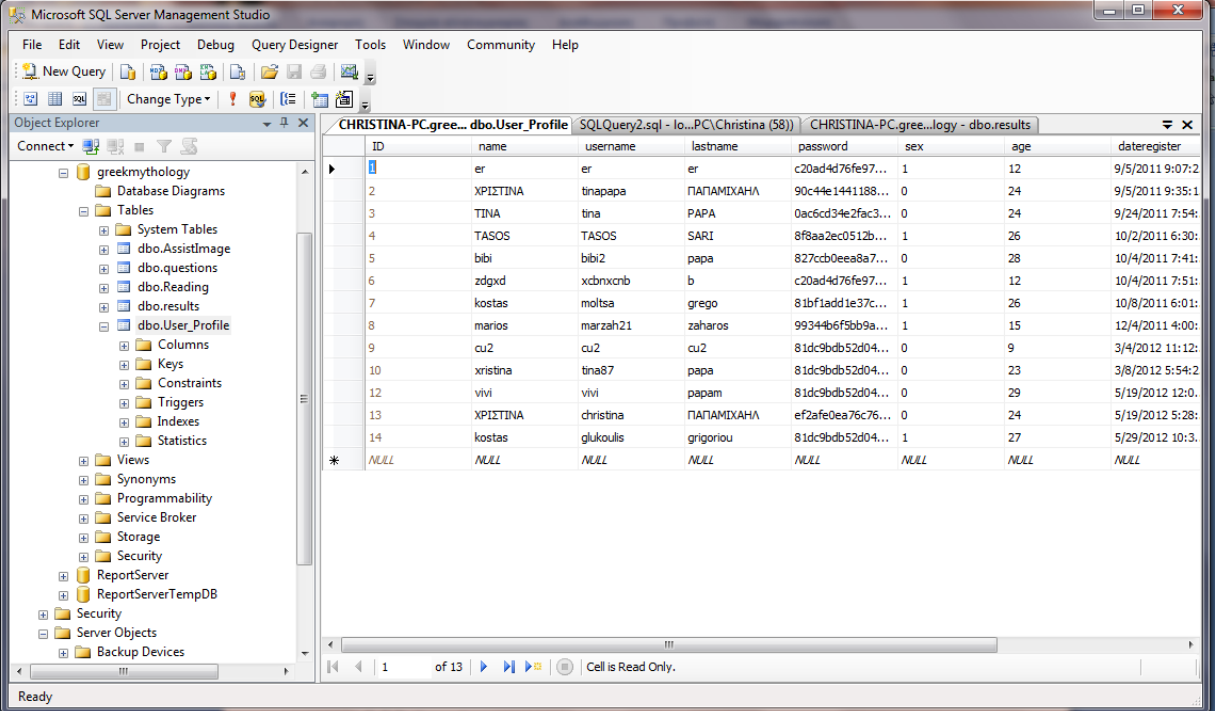
ID	chapter	rate	datetest	timetest	pass	completetest	radiochange
0	0	80	9/25/2011 6:34:...	0:45	1	NULL	NULL
0	0	80	9/25/2011 6:48:...	0:35	1	NULL	NULL
0	0	64	9/25/2011 6:51:...	0:35	1	NULL	NULL
0	0	60	9/25/2011 7:05:...	0:38	1	NULL	NULL
3	1	36	9/25/2011 8:11:...	1:20	0	NULL	NULL
3	1	64	9/25/2011 8:19:...	0:47	1	NULL	NULL
3	1	60	9/27/2011 8:31:...	0:28	1	1	92
3	1	66	9/29/2011 8:19:...	0:43	1	1	81
3	1	55	9/30/2011 7:38:...	0:37	1	1	45
3	2	13	9/30/2011 7:39:...	0:30	0	1	28
3	2	46	9/30/2011 7:40:...	0:45	0	1	35
3	2	66	9/30/2011 7:41:...	0:39	1	1	44
3	3	120	9/30/2011 7:42:...	10:00	1	0	41
3	3	43	9/30/2011 7:44:...	0:48	0	1	31
3	1	40	9/30/2011 7:50:...	0:50	0	1	53
3	1	100	9/30/2011 8:03:...	0:49	1	1	55
3	2	38	9/30/2011 8:04:...	0:43	0	1	28
3	3	26	9/30/2011 8:07:...	0:59	0	1	28
3	3	0	9/30/2011 8:08:...	0:45	0	1	16
3	4	26	9/30/2011 8:09:...	0:31	0	1	27
3	3	56	9/30/2011 8:19:...	-3:48	1	1	28

Εικ. 3.4 Results

3.5 User_Profile

Στον πίνακα (table) dbo.User_profile αποθηκεύονται όλα τα απαραίτητα στοιχεία που σχετίζονται με τον λογαριασμό του κάθε χρήστη. Αποτελείται από τις δέκα ακόλουθες στήλες.

- ID: Αποθηκεύει το id του κάθε χρήστη.
- name: Αποθηκεύει το όνομα του κάθε χρήστη.
- username: Αποθηκεύει το username του κάθε χρήστη.
- lastname: Αποθηκεύει το επώνυμο του κάθε χρήστη.
- password: Αποθηκεύει ο κωδικός πρόσβασης του κάθε χρήστη.
- sex: Αποθηκεύει 1 εάν ο χρήστης είναι αγόρι και 0 εάν είναι κορίτσι.
- age: Αποθηκεύει την ηλικία του κάθε χρήστη.
- dateregister: Καταχωρείται η ώρα και η ημερομηνία που έκανε registration ο χρήστης.
- assistant: Αποθηκεύει το ID του βοηθού που έχει επιλέξει ο χρήστης.
- coins: Αποθηκεύει τα νομίσματα που έχει στην κατοχή του ο χρήστης.



ID	name	username	lastname	password	sex	age	dateregister
1	er	er	er	c20ad4d76fe97...	1	12	9/5/2011 9:07:2
2	ΧΡΙΣΤΙΝΑ	tinapapa	ΠΑΠΑΜΙΧΑΗΛ	90c44e1441188...	0	24	9/5/2011 9:35:1
3	TINA	tina	PAPA	0ac6cd34e2fac3...	0	24	9/24/2011 7:54:
4	TASOS	TASOS	SARI	8f8aa2ec0512b...	1	26	10/2/2011 6:30:
5	bibi	bibi2	papa	827ccb0eea8a7...	0	28	10/4/2011 7:41:
6	zdgxd	xcbnxcnb	b	c20ad4d76fe97...	1	12	10/4/2011 7:51:
7	kostas	moltsa	grego	81bf1add1e37c...	1	26	10/8/2011 6:01:
8	marios	marzah21	zaharos	99344b6f5bb9a...	1	15	12/4/2011 4:00:
9	cu2	cu2		81dc9bdb52d04...	0	9	3/4/2012 11:12:
10	xristina	tina87	papa	81dc9bdb52d04...	0	23	3/8/2012 5:54:2
12	vivi	vivi	papam	81dc9bdb52d04...	0	29	5/19/2012 12:0.
13	ΧΡΙΣΤΙΝΑ	christina	ΠΑΠΑΜΙΧΑΗΛ	ef2afe0ea76c76...	0	24	5/19/2012 5:28:
14	kostas	glukoulis	grigoriou	81dc9bdb52d04...	1	27	5/29/2012 10:3.
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Εικ. 3.5 User_Profile

3.6 Sqlqueries.cs

Πρόκειται για μία από τις πιο σημαντικές κλάσεις καθώς εδώ περιέχονται όλα τα SQL statement που χρησιμοποιούνται στο πρόγραμμα . Κάθε συνάρτηση που περιέχεται εδώ

καλείται από το κυρίως πρόγραμμα και δημιουργεί μία σύνδεση με τον SQL server. Στη συνέχεια, η συνάρτηση παίρνει ορίσματα-τιμές που χρειάζονται για να εκτελέσει τα statements και επιστρέφει σε μεταβλητές τις εγγραφές από κάποιο πίνακα ή ακόμα και ολόκληρο πίνακα για να χρησιμοποιηθεί από το πρόγραμμα. Επιπλέον, συχνά κάποια συνάρτηση χρειάζεται σε περισσότερα από ένα σημεία στην ίδια φόρμα ή σε κάποια άλλη φόρμα, οπότε αντί να γράφεται πολλές φορές η ίδια συνάρτηση, γράφεται μία φορά και μπορεί να χρησιμοποιηθεί κάθε ανά πάσα στιγμή σε οποιοδήποτε σημείο του προγράμματος. Τέλος σε κάποια σημεία όπου η μεταβλητή που παίρνει το πρόγραμμα προέρχεται από είσοδο δεδομένων του χρήστη (για παράδειγμα όταν πληκτρολογεί τα στοιχεία του για να κάνει login) εξυπηρετεί την ασφάλειας καθώς καθορίζεται στο κώδικα τι είδους δεδομένα θα μπου σαν μεταβλητές στο query που θα εκτελεστεί στη βάση.

4. Υλοποίηση της εφαρμογής

4.1 Φόρμα εισαγωγής του χρήστη - loginuser.cs

Η πρώτη φόρμα που εμφανίζεται ανοίγοντας την εφαρμογή είναι η loginuser.cs, μέσω της οποίας ο ήδη υπάρχον χρήστης εισάγοντας τα στοιχεία του (Όνομα Χρήστη – Κωδικό) εισέρχεται στην εφαρμογή. Εάν δεν έχει ήδη λογαριασμό πατώντας το κουμπί της “Δημιουργία Λογαριασμού” στην createuser.cs όπου έχει τη δυνατότητα να δημιουργήσει νέο λογαριασμό. Σε περίπτωση που έχει ήδη λογαριασμό και έχει ξεχάσει τον κωδικό του πατώντας το κουμπί “Ξέχασα τον κωδικό” μεταφέρεται στη φόρμα createnewpassword.cs στην οποία δημιουργεί νέο κωδικό.

Δίνοντας ο χρήστης τα στοιχεία που απαιτούνται για να εισαχθεί στο σύστημα γίνεται έλεγχος για το εάν αυτά υπάρχουν στη βάση δεδομένων. Εάν βρεθούν τα στοιχεία ο χρήστης εισέρχεται στην εφαρμογή και ταυτόχρονα ανακτώνται από βάση τα εξής στοιχεία του χρήστη, φύλο, τα κεφάλαια που μπορεί να διαβάσει και το ID του. Σε περίπτωση εισαγωγής λανθασμένων στοιχείων εμφανίζεται μήνυμα λάθους το οποίο ενημερώνει τον χρήστη για τη λανθασμένη εισαγωγή που έκανε.

```
private void button1_Click(object sender, EventArgs e)
{
    SqlQueries query = new SqlQueries();

    MD5EncryptDecrypt enc = new MD5EncryptDecrypt();
    enc.ConvertToMd5(textBoxpasswordlogin.Text.ToString());

    string newst = enc.pamd5.ToString();
    query.UserLogin(textBoxnamellogin.Text.ToString(),
newst.ToString());
    bool findit = query.FindData;
    if (findit)
    {
        SqlQueries GetUserGredentials = new SqlQueries();
        string hhhhh =
GetUserGredentials.UserHaveLogg(textBoxnamellogin.Text.ToString());

        mythlab mythform = new mythlab();

        FormTests questionform = new FormTests();
        string welname = textBoxnamellogin.Text;
        questionform.MyProperty = welname;

        mythform.MyProperty = welname;

        DataRow Idrow;
        DataTable Idtable = new DataTable();
    }
}
```

```

        SqlQueries getid = new SqlQueries();
        getid.GetIdByName(welname);
        Idtable = getid.GetId;
        Idrow = Idtable.Rows[0];
        int usersID = Convert.ToInt32(Idrow["ID"]);
        SqlQueries enable = new SqlQueries();
        enable.EnableChapters(usersID);
        int h = enable.gotchapter;
        mythform.available_chapters = h;
        DataRow GenRow;
        DataTable Gentable = new DataTable();
        SqlQueries getgen = new SqlQueries();
        getgen.GetGenByName(welname);
        Gentable = getgen.GetGender;
        GenRow = Gentable.Rows[0];
        int users_sex = Convert.ToInt32(GenRow["sex"]);
        mythform.WhatGender = users_sex;
        mythform.Show();
        this.Hide();
    }
    else { MessageBox.Show("Λάθος κωδικός ή όνομα χρήστη"); }
}

```

Κωδ. 4.1 Εισαγωγή στην εφαρμογή

4.2 Δημιουργία Χρήστη - createuser.cs

Στη φόρμα αυτή ο χρήστης εισάγει όλα τα απαραίτητα στοιχεία για να δημιουργήσει λογαριασμό. Τα στοιχεία που απαιτούνται είναι τα ακόλουθα.

- Όνομα
- Επίθετο
- Όνομα χρήστη
- Κωδικός
- Επαλήθευση κωδικού
- Ηλικία
- Φύλο
- Θεό

Όσον αφορά τα τέσσερα πρώτα στοιχεία δεν υπάρχει κάποιος περιορισμός για τον αν αυτά θα γραφτούν τόσο με κεφαλαία ή μικρά γράμματα όσο με λατινικούς ή ελληνικούς

χαρακτήρες. Το όνομα χρήστη είναι μοναδικό για κάθε χρήστη, για το λόγο αυτό γίνεται σχετικός έλεγχος αν έχει ήδη επιλεγεί από άλλο χρήστη. Το password κρυπτογραφείται έτσι ώστε να παρέχει ασφάλεια στα δεδομένα του χρήστη, κατά συνέπεια δεν υπάρχει δυνατότητα ανάκτησης του κωδικού πρόσβασης. Για το λόγο αυτό ήταν αναγκαία η δημιουργία μιας κλάσης η οποία θα περιλάμβανε τον κώδικα κρυπτογράφησης.

Η κλάση MD5EncryptDecrypt.cs περιλαμβάνει τον κώδικα που χρησιμοποιείται ώστε να κρυπτογραφηθεί το password του χρήστη. Καλείται από το κυρίως πρόγραμμα η συνάρτηση της ConvertToMd5 παίρνει σαν όρισμα ένα string που είναι ο κωδικός και αφού τον κρυπτογραφήσει τον επιστρέφει στο πρόγραμμα από όπου κλήθηκε.

```
class MD5EncryptDecrypt
{
    public string pamd5;
    public string ConvertToMd5(string UnconvertString)
    {
        //ο κώδικας που υλοποιεί τη κρυπτογράφηση MD5, παίρνει το
        string που του περνάει το κυρίως πρόγραμμα
        //και το κρυπτογραφεί μαζί με ένα hash και επιστρέφει το
        κρυπτογραφημένο string
        System.Security.Cryptography.MD5CryptoServiceProvider
md5string = new
System.Security.Cryptography.MD5CryptoServiceProvider();
        byte[] s =
System.Text.Encoding.UTF8.GetBytes(UnconvertString);
        s = md5string.ComputeHash(s);
        System.Text.StringBuilder strb = new
System.Text.StringBuilder();
        foreach (byte i in s)
        {
            strb.Append(i.ToString("x2").ToLower());
        }
        pamd5 = strb.ToString();
        return pamd5;
    }
}
```

Κωδ. 4.2 Κρυπτογράφηση κωδικού πρόσβασης

Κατά την επαλήθευση κωδικού γίνεται έλεγχος για το αν πεδίο “κωδικός” και “επαλήθευση κωδικού” έχουν το ίδιο περιεχόμενο.

Στο πεδίο ηλικία ελέγχεται αν η καταχώρηση του μαθητή είναι αριθμός ή οποιοσδήποτε άλλος χαρακτήρας. Στη δεύτερη περίπτωση εμφανίζεται μήνυμα καθοδήγησης για την εισαγωγή αριθμού.

Εάν ο χρήστης είναι αγόρι αποθηκεύεται η τιμή 1 στο πεδίο sex του πίνακα User_Profile και 0 εάν ο χρήστης είναι κορίτσι.

Πριν ο χρήστης ολοκληρώσει την εγγραφή πρέπει να επιλέξει και έναν από τους 12 θεούς του Ολύμπου. Ανάλογα με το ποιο θεό θα επιλέξει η κεντρική φόρμα θα έχει διαφορετικό χρώμα. Σε περίπτωση που ο χρήστης θελήσει να αλλάξει θεό του παρέχεται η δυνατότητα στη συνέχεια της εφαρμογής. Εάν ο χρήστης έχει συμπληρώσει χωρίς κάποιο σφάλμα γίνεται επιτυχής εγγραφή των στοιχείων του στη βάση. Σε αντίθετη περίπτωση του εμφανίζεται μήνυμα για συμπλήρωση όλων των πεδίων.

```

if (comboBox1.SelectedItem.ToString() == "Δίας")
{
    Assist = 1;
    string relPath =
System.IO.Path.Combine(Application.StartupPath);
    string path = Environment.CurrentDirectory;
    int index =
relPath.LastIndexOf(@"GreekMythology\", relPath.Length - 1);
    if (index >= 0)
        relPath = relPath.Substring(0, index);
    pictureBox2.ImageLocation = relPath +
@"\GreekMythology\images\The_Gods___Zeus.jpg";
}
else if (comboBox1.SelectedItem.ToString() ==
"Ηρα")
{
    Assist = 2;
}
else if (comboBox1.SelectedItem.ToString() ==
"Αθηνά")
{
    Assist = 3;
}
else if (comboBox1.SelectedItem.ToString() ==
"Άρτεμης")
{
    Assist = 4;
}
else if (comboBox1.SelectedItem.ToString() ==
"Αφροδίτη")
{
    Assist = 5;
}
else if (comboBox1.SelectedItem.ToString() ==
"Απόλλωνας")
{
    Assist = 6;
}
else if (comboBox1.SelectedItem.ToString() ==
"Ποσειδώνας")
{

```

```
        Assist = 7;
    }
    else if (comboBox1.SelectedItem.ToString() ==
"Ηφαιστος")
    {
        Assist = 8;
    }
    else if (comboBox1.SelectedItem.ToString() ==
"Δήμητρα")
    {
        Assist = 9;
    }
    else if (comboBox1.SelectedItem.ToString() ==
"Άρης")
    {
        Assist = 10;
    }
    else if (comboBox1.SelectedItem.ToString() ==
"Άδης")
    {
        Assist = 11;
    }
    else if (comboBox1.SelectedItem.ToString() ==
"Ερμής")
    {
        Assist = 12;
    }

    query.CreateAccount(name, lastname, username,
newst.ToString(), sex, age, dayregq, Assist, 0);

    MessageBox.Show("Η εγγραφή σας ολοκληρώθηκε
επιτυχώς!");
    Close();
```

Κωδ. 4.3 Επιλογή θεού

4.3 Δημιουργία κωδικού πρόσβασης - createnewpassword.cs

Υπάρχει η δυνατότητα δημιουργίας νέου κωδικού σε περίπτωση που ο χρήστης ξεχάσει ή θελήσει να αλλάξει κωδικό. Η υλοποίηση της αλλαγής του κωδικού παρέχεται από τη φόρμα createnewpassword.cs. Ζητείται από τον χρήστη να συμπληρώσει το όνομά του, το επίθετό του το όνομα χρήστη και τον νέο κωδικό. Στη συνέχεια ελέγχεται αν στοιχεία που έχει εισάγει ταυτοποιούνται με αυτά της βάσης και αν αυτό ισχύει ολοκληρώνεται επιτυχώς η αλλαγή του κωδικού. Ειδάλλως εμφανίζεται αντίστοιχο μήνυμα στο χρήστη για την εισαγωγή των σωστών στοιχείων.

```
private void passwordrecreate_Click(object sender, EventArgs e)
{
    string name = textBoxname.Text.ToString();
    string lastname=textBoxlastname.Text.ToString();
    string username = textBoxusername.Text.ToString();
    string newpassword = textBoxpasswordnew.Text.ToString();

    MD5EncryptDecrypt encrypt = new MD5EncryptDecrypt();
    encrypt.ConvertToMd5(newpassword);
    string encrypted = encrypt.pamd5.ToString();

    SqlQueries query = new SqlQueries();
    query.CreateNewAccount(name, lastname, username,
encrypted);
    bool AccountHasCreated = query.FindData;

    if (AccountHasCreated)
    {
        MessageBox.Show("Ο κωδικός έχει αλλάξει");
        this.Close();
    }
    else
    {
        MessageBox.Show("Η αλλαγή του κωδικού δεν
ολοκληρώθηκε ,παρακαλώ ελέξτε τα στοιχεία");
    }
}
```

Κωδ. 4.4 Αλλαγή κωδικού πρόσβασης

4.4 Κεντρική φόρμα

Για να γίνει εμφάνιση αυτής της φόρμας απαιτούνται τα εξής στοιχεία του χρήστη:

- Το ID

- Το όνομα χρήστη (username)
- Ο θεός που έχει επιλέξει
- Το φύλο
- Τα διαθέσιμα κεφάλαια
- Τα διαθέσιμα νομίσματα

Το ID του χρήστη χρησιμοποιείται για να βρούμε στη βάση διότι στους πίνακες που έχουν δημιουργηθεί στη βάση δεδομένων ο κάθε χρήστης χαρακτηρίζεται από το δικό του ID. Συνεπώς αν θέλουμε να κάνουμε ένα ερώτημα SQL θα πρέπει να χρησιμοποιήσουμε το μοναδικό χαρακτηριστικό του χρήστη που είναι το ID.

Το username μεταφέρεται από την προηγούμενη φόρμα (loginuser.cs) όπου και είχε εισαχθεί στο αντίστοιχο πεδίο από το χρήστη. Αυτό αφού μεταφερθεί χρησιμοποιείται έτσι ώστε να εξάγουμε τα στοιχεία φύλο (sex), τον θεό (assistant), τα διαθέσιμα νομίσματα (coins) και το ID (primary key) από τον πίνακα User_Profile.

Η εγγραφή που είναι αποθηκευμένη στη στήλη θεός είναι αναγκαία καθώς η φόρμα mythlab.cs αλλάζει background color ανάλογα με την επιλεγμένη θεότητα. Το background color καθορίζεται από την εγγραφή στη στήλη colorAssist στον πίνακα AssistImage. Πιο αναλυτικά δημιουργήσαμε ένα SQL Query πάνω στον πίνακα User_Profile το οποίο βρίσκει και επιστρέφει ποιόν θεό έχει επιλέξει ο εισηγμένος χρήστης. Στη συνέχεια με ένα δεύτερο SQL Query πάνω στον πίνακα AssistImage επιστρέφεται το χρώμα και η φωτογραφία για τον αντίστοιχο θεό.

```
DataRow AssistRow;  
    DataRow AssistColorRow;  
    DataRow ImageRowColor;  
    DataTable Assisttable = new DataTable();  
    SqlQueries getAssist = new SqlQueries();  
    getAssist.GetAssistByName(MyProperty);  
    Assisttable = getAssist.GetAssist;  
    AssistRow = Assisttable.Rows[0];  
    AssistColorRow = Assisttable.Rows[0];  
    int users_assist =  
Convert.ToInt32(AssistRow["assistant"]);  
  
    DataTable AsInfotable = new DataTable();  
    DataTable imageAssist = new DataTable();  
    SqlQueries getAssistInfo = new SqlQueries();  
    getAssistInfo.GetAssistInfoByName(users_assist);  
    AsInfotable = getAssistInfo.GetAssistInfo;  
    AssistColorRow = AsInfotable.Rows[0];  
    ImageRowColor = AsInfotable.Rows[0];  
    selected_color =  
AssistColorRow["colorAssist"].ToString();
```

```

        string imageofAssistan =
ImageRowColor["assistPhoto"].ToString();

public DataTable GetAssist;
    public DataTable GetAssistByName(string username)
    {
        SqlConnection conn = new SqlConnection();
        conn.ConnectionString =
@"Server=localhost;Database=greekmythology;Trusted_Connection=True;";
        try
        {
            conn.Open();
        }
        catch (Exception e)
        {

            MessageBox.Show(e.ToString());
        }
        SqlParameter p1 = new SqlParameter("@username",
SqlDbType.NVarChar);
        p1.Value = username.ToString();

        SqlCommand sqlcom = new SqlCommand("select * from
user_profile where username=@username;", conn);
        sqlcom.Parameters.Add(p1);
        // sqlcom.ExecuteNonQuery();
        SqlDataAdapter AssistAdap = new SqlDataAdapter();
        GetAssist = new DataTable();
        AssistAdap.SelectCommand = sqlcom;
        AssistAdap.Fill(GetAssist);
        conn.Close();

        return GetAssist;

    }

    public DataTable GetAssistInfo;
    public DataTable GetAssistInfoByName(int assistant)
    {
        SqlConnection conn = new SqlConnection();
        conn.ConnectionString =
@"Server=localhost;Database=greekmythology;Trusted_Connection=True;";
        try
        {
            conn.Open();
        }
        catch (Exception e)
        {

```

```

        MessageBox.Show(e.ToString());
    }
    SqlParameter p1 = new SqlParameter("@assistant",
SqlDbType.Int);
    p1.Value = assistant;

    SqlCommand sqlcom = new SqlCommand("select * from
AssistImage where assistant=@assistant;", conn);
    sqlcom.Parameters.Add(p1);

    // sqlcom.ExecuteNonQuery();
    SqlDataAdapter FeelingAdap = new SqlDataAdapter();
    GetAssistInfo = new DataTable();
    FeelingAdap.SelectCommand = sqlcom;
    FeelingAdap.Fill(GetAssistInfo);
    conn.Close();

    return GetAssistInfo;
}

```

Κωδ. 4.5 Αλλαγή χρώματος εφαρμογής ανάλογα με την επιλογή θεού

Το φύλο χρησιμοποιείται στην εναλλαγή των χρωμάτων στα labels που υπάρχουν στη φόρμα. Αν ο χρήστης είναι αγόρι τα labels έχουν μπλε χρώμα, εάν είναι κορίτσι έχουν ροζ χρώμα. Η ιδιότητα αυτή διατηρείται σε όλες τις επόμενες φόρμες.

```

if (WhatGender == 1)
{
    this.label1.ForeColor =
System.Drawing.Color.SteelBlue;
    this.label2.ForeColor =
System.Drawing.Color.SteelBlue;
    this.labelCoins.ForeColor =
System.Drawing.Color.SteelBlue;
    this.radioButton1.ForeColor =
System.Drawing.Color.SteelBlue;
    this.radioButton2.ForeColor =
System.Drawing.Color.SteelBlue;
    this.radioButton3.ForeColor =
System.Drawing.Color.SteelBlue;
}

```

```

        this.radioButton4.ForeColor =
System.Drawing.Color.SteelBlue;
        this.radioButton5.ForeColor =
System.Drawing.Color.SteelBlue;
        this.button1.ForeColor =
System.Drawing.Color.SteelBlue;
        this.button2.ForeColor =
System.Drawing.Color.SteelBlue;
        this.buttonPythia.ForeColor =
System.Drawing.Color.SteelBlue;

    }
    else
    {

        this.label1.ForeColor =
System.Drawing.Color.PaleVioletRed;
        this.label2.ForeColor =
System.Drawing.Color.PaleVioletRed;
        this.labelCoins.ForeColor =
System.Drawing.Color.PaleVioletRed;
        this.radioButton1.ForeColor =
System.Drawing.Color.PaleVioletRed;
        this.radioButton2.ForeColor =
System.Drawing.Color.PaleVioletRed;
        this.radioButton3.ForeColor =
System.Drawing.Color.PaleVioletRed;
        this.radioButton4.ForeColor =
System.Drawing.Color.PaleVioletRed;
        this.radioButton5.ForeColor =
System.Drawing.Color.PaleVioletRed;
        this.button1.ForeColor =
System.Drawing.Color.PaleVioletRed;
        this.button2.ForeColor =
System.Drawing.Color.PaleVioletRed;
        this.buttonPythia.ForeColor =
System.Drawing.Color.PaleVioletRed;

    }

```

Κωδ. 4.6 Αλλαγή χρώματος γραμμάτων ανάλογα με το φύλο

Τα διαθέσιμα κεφάλαια επιστρέφονται διότι ο χρήστης δεν μπορεί να επιλέξει οποιοδήποτε κεφάλαιο αυθαίρετα, αλλά ακολουθείται η παρακάτω λογική. Την πρώτη φορά εισαγωγής στην εφαρμογή ο μαθητής έχει τη δυνατότητα να διαβάσει αλλά και να ελέγξει τις γνώσεις του με τη βοήθεια κάποιου τεστ μόνο από το κεφάλαιο 1. Στο σημείο αυτό είναι

απαραίτητο να σημειώσουμε ότι δεν απαιτείται η ανάγνωση του διαθέσιμου υλικού για την πραγματοποίηση του τεστ. Σε περίπτωση που ο χρήστης καταφέρει να ολοκληρώσει το τεστ με βαθμολογία μεγαλύτερη ή ίση του 50 % τότε μόνο μπορεί να επιλέξει κάποιο τεστ από τα επόμενα κεφάλαια. Η ίδια λογική ακολουθείται για κάθε επόμενο κεφάλαιο που πρέπει να ξεκλειδώσει. Σε περίπτωση που εισαχθεί επόμενη φορά στο σύστημα τα κεφάλαια που είχε ξεκλειδώσει σε προηγούμενη επίσκεψή του παραμένουν διαθέσιμα. Για να επιτευχθεί η συγκεκριμένη διαδικασία δημιουργήθηκε ένα SQL Query το οποίο επιστρέφει το ID του χρήστη βάσει του username του πίνακα User_Profile. Αφού πάρουμε το ID έχουμε δημιουργήσει ένα δεύτερο SQL Query στον πίνακα results το οποίο βρίσκει τις υπάρχουσες εγγραφές που έχουν το παραπάνω ID. Στη συνέχεια ελέγχει για ποιες από αυτές τις εγγραφές η στήλη pass έχει την εγγραφή 1 η οποία χαρακτηρίζει αν κάποιο κεφάλαιο έχει περαστεί ή όχι. Οι εγγραφές αυτές ταξινομούνται με βάση το chapter κατά φθίνουσα σειρά. Εάν δεν υπάρχουν εγγραφές με την τιμή ένα στη στήλη pass αυτό σημαίνει ότι ο χρήστης είτε δεν έχει κάνει κανένα τεστ είτε δεν έχει περάσει κανένα τεστ από το κεφάλαιο 1, συνεπώς μόνο το κεφάλαιο 1 θα είναι διαθέσιμο και τα υπόλοιπα θα είναι κλειδωμένα. Στην αντίθετη περίπτωση, δηλαδή που βρεθούν εγγραφές με την τιμή 1, παίρνει το πρώτο κελί που βρίσκεται από την ταξινομημένη λίστα το οποίο αντιστοιχεί στο κεφάλαιο με τη μεγαλύτερη τιμή.

```
public int gotchapter;
    public int EnableChapters(int id)
    {
        SqlConnection conn = new SqlConnection();
        conn.ConnectionString =
@"Server=localhost;Database=greekmythology;Trusted_Connection=True;";
        try
        {
            conn.Open();
        }
        catch (Exception e)
        {
            MessageBox.Show(e.ToString());
        }
        SqlParameter p1 = new SqlParameter("@id", SqlDbType.Int);
        p1.Value = id;
        SqlCommand sqlcom = new SqlCommand("select * from results
where pass=1 and ID=@id order by chapter DESC", conn);
        sqlcom.Parameters.Add(p1);
        SqlDataAdapter Idadapter = new SqlDataAdapter();
        Idadapter.SelectCommand = sqlcom;
        DataTable idstable = new DataTable();
        Idadapter.Fill(idstable);
        DataRow idsrows;
        SqlDataReader datar = sqlcom.ExecuteReader();

        if (datar.HasRows)
```

```
{
    idsrows = idstable.Rows[0];
    gotchapter = Convert.ToInt32(idsrows["chapter"]);
    //return gotchapter;
}
else
{
    gotchapter = 0;
}
conn.Close();
return gotchapter;
}
```

```
SqlQueries enable = new SqlQueries();
enable.EnableChapters(usersID);
int h = enable.gotchapter;
```

Κωδ. 4.7 Εμφάνιση διαθέσιμων κεφαλαίων για έλεγχο γνώσεων

Όπως αναφέραμε όταν φορτώνεται η φόρμα mythlab.cs φορτώνεται και η τιμή που έχουν τα coins. Η βασική χρησιμότητα των coins είναι η επίσκεψη του χρήστη στο Μαντείο των Δελφών. Για τους νέους χρήστες αλλά και για όσους δεν έχουν πραγματοποιήσει κάποιο τεστ τα coins έχουν τιμή 0.

Τέλος στην φόρμα mythlab.cs υπάρχει και ένα ημερολόγιο από το οποίο ο χρήστης μπορεί να επιλέξει μία ημερομηνία και να ελέγξει αν εκείνη την ημέρα είχε κάνει κάποιο τεστ και τι βαθμολογία είχε πάρει. Εάν ο μαθητής είχε κάνει τεστ από διάφορα κεφάλαια ή πολλά τεστ του ίδιου κεφαλαίου επιστρέφεται το τεστ με την υψηλότερη βαθμολογία που είχε επιτευχθεί την επιλεγμένη μέρα. Η διαδικασία αυτή λειτουργεί ως εξής, όταν ο μαθητής επιλέξει ημέρα γίνεται ένα SQL ερώτημα για το αν υπάρχει εγγραφή στη στήλη datetest του πίνακα results με την ημερομηνία αυτή. Τα δεδομένα που βρέθηκαν με αυτό το χαρακτηριστικό ταξινομούνται με φθίνουσα σειρά, από την μεγαλύτερη προς τη μικρότερη βαθμολογία και στην οθόνη επιστρέφεται η εγγραφή με την μεγαλύτερη βαθμολογία.

```
private void dateTimePickerRate_ValueChanged(object sender, EventArgs e)
```

```

{
    DataRow Idrow;
    DataTable Idtable = new DataTable();
    SqlQueries getid = new SqlQueries();
    getid.GetIdByName(label1.Text.ToString());
    Idtable = getid.GetId;
    Idrow = Idtable.Rows[0];
    usersID = Convert.ToInt32(Idrow["ID"]);
    string mydate =
dateTimePickerRate.Value.ToString("MM/dd/yyyy");
    DataRow TestRateRow;
    DataTable GetTestRatetable = new DataTable();
    SqlQueries showmyresults = new SqlQueries();
    showmyresults.GetTestRateByname(usersID, mydate);
    GetTestRatetable = showmyresults.GetTestRate;
    TestRateRow = GetTestRatetable.Rows[0];
    // MessageBox.Show(TestRateRow["chapter"].ToString());
    label3.Text = "Είχες επιλέξει το κεφάλαιο:" +
TestRateRow["chapter"].ToString() + "\r\n και η βαθμολογία σου ήταν:"
+ TestRateRow["rate"].ToString();
}

```

Κωδ. 4.8 Εμφάνιση ημερολογίου

4.5 Μελέτη χρήστη

Ο χρήστης μπορεί να επιλέξει από τα διαθέσιμα κεφάλαια που έχει ξεκλειδώσει ποιο θέλει να διαβάσει. Κάνοντας κλικ στο κουμπί “Ανάγνωση” μεταφέρεται στη φόρμα Reader.cs. Στη φόρμα αυτή φορτώνεται ένα βιβλίο σε μορφή flash μέσα από το οποίο μπορεί να κάνει ανάγνωση του περιεχομένου του μαθήματος. Στη φόρμα αυτή μεταφέρεται το username του χρήστη και το ID του. Το username χρησιμοποιείται διότι στο επάνω μέρος της οθόνης εμφανίζεται το username του χρήστη με ένα μήνυμα που του εύχεται καλή ανάγνωση. Η ανάγνωση του υλικού δεν είναι υποχρεωτική για να ξεκινήσει ο χρήστης κάποιο τεστ. Μόλις η φόρμα φορτώσει ξεκινάει ένας timer, ο οποίος μετράει πόσο χρόνο είναι ανοικτή η φόρμα. Αν ο χρήστης κατά τη διάρκεια που έχει ανοικτή τη φόρμα με το υλικό δεν είναι αφοσιωμένος στη μελέτη και έχει ανοικτές άλλες εφαρμογές με τις οποίες ασχολείται εμφανίζεται στην οθόνη του σχετικό μήνυμα το οποίο του λέει ότι δεν είναι αφοσιωμένος στη μελέτη. Το μήνυμα αυτό εμφανίζεται μόλις συμπληρωθεί ένα λεπτό κατά το οποίο ο χρήστης είχε την προσοχή του στραμμένη αλλού. Αυτός ο έλεγχος γίνεται με τη χρήση ενός δεύτερου timer.

Όταν ο χρήστης θελήσει να βγει από την φόρμα και να σταματήσει την ανάγνωση γίνεται έλεγχος για το αν κατά τη διάρκεια που διάβαζε υπήρξαν κάποια λεπτά που καταγράφηκαν από τον δεύτερο timer κατά τα οποία δεν ήταν συγκεντρωμένος και είτε είχε κάποιο άλλο πρόγραμμα ανοικτό είτε δεν διάβαζε το περιεχόμενο. Αν υπήρχαν τέτοια λεπτά τότε εμφανίζεται μήνυμα για το πόσο χρόνο διάβασε και πόσο χρόνο δεν ήταν συγκεντρωμένος.

Κατά την έξοδο γίνεται αποθήκευση στη βάση δεδομένων στον πίνακα "Reading" το id του μαθητή (ID), το κεφάλαιο που διάβασε (chapter), ο χρόνος που αφιέρωσε στο διάβασμα (διαχωρίζουμε τα λεπτά και τα δευτερόλεπτα, ReadingMin, ReadingSec), από τον οποίο έχει αφαιρεθεί ο χρόνος που είχε την προσοχή του αλλού, και η ημερομηνία μελέτης (LastRead). Τα στοιχεία αυτά αποθηκεύονται καθότι χρησιμεύουν στην τελική αξιολόγηση του χρήστη η οποία θα αναλυθεί παρακάτω.

```
void timemouse_Tick(object sender, EventArgs e)
{
    if (h == "j")
    { timemouse.Stop(); }
    else
    {
        if (timemousesecs < 59)
        {
            timemousesecs = timemousesecs + 1;
            timecountmouse = timemousemins + ":" +
timemousesecs;
        }

        if (timemousesecs == 59)
        {
            timemousesecs = 0;
            timemousemins = timemousemins + 1;
            timecountmouse = timemousemins + ":" +
timemousesecs;
        }
    }
    if (((timemousemins * 60) + timemousesecs > 60) && xaz ==
"no")
    {
        xaz = "yes";
        MessageBox.Show("Καλό θα ήταν να είσαι συγκεντρωμένος
και να μην ασχολείσαι με κάτι άλλο!");
    }
}
```

Κωδ. 4.9 Καταγραφή χρόνου διαβάσματος

4.6 Επιλογή Θεού.

Μία από τις επιλογές που υπάρχουν στο κεντρικό παράθυρο της εφαρμογής είναι αυτή της αλλαγής του θεού που φαίνεται στο παράθυρο του λογισμικού. Κάνοντας κλικ στο κουμπί “12” εμφανίζεται ένα παράθυρο το οποίο επιτρέπει στο χρήστη να αλλάξει τον θεό τον οποίο είχε επιλέξει για πρώτη φορά όταν εγγράφηκε στο σύστημα. Φορτώνοντας, λοιπόν, η φόρμα gods.cs εμφανίζεται ένα drop down list το οποίο περιέχει τους δώδεκα θεούς του Ολύμπου. Επιλέγοντας κάποιον από τους δώδεκα εμφανίζεται στα δεξιά της οθόνης η φωτογραφία του και στο κάτω μέρος η ιδιότητα που έχει κάθε θεός. Αν δε θέλει να κάνει κάποια αλλαγή απλά κλείνει το παράθυρο αυτό. Αν θέλει να καταχωρήσει την επιλογή του κάνει κλικ στο κουμπί “Αλλαγή” και αμέσως γίνεται update στο προφίλ του χρήστη που είναι αποθηκευμένο στη βάση δεδομένων. Κλείνοντας το παράθυρο αυτό και έχοντας αποθηκευθεί η αλλαγή που έκανε, η φόρμα mythlab.cs δηλαδή το κεντρικό παράθυρο της εφαρμογής φορτώνεται ξανά έχοντας νέο background color και νέα εικόνα στα αριστερά της οθόνης.

```
private void gods_FormClosed(object sender, FormClosedEventArgs e)
{
    mythlab mythform = new mythlab();
    mythform.MyProperty = idname;
    DataRow Idrow;
    DataTable Idtable = new DataTable();
    SqlQueries getid = new SqlQueries();
    getid.GetIdByName(idname);
    Idtable = getid.GetId;
    Idrow = Idtable.Rows[0];
    int usersID = Convert.ToInt32(Idrow["ID"]);
    SqlQueries enable = new SqlQueries();
    enable.EnableChapters(usersID);
    int h = enable.gotchapter;
    mythform.available_chapters = h;

    DataRow GenRow;
    DataTable Gentable = new DataTable();
    SqlQueries getgen = new SqlQueries();
    getgen.GetGenByName(idname);
    Gentable = getgen.GetGender;
    GenRow = Gentable.Rows[0];
    int users_sex = Convert.ToInt32(GenRow["sex"]);

    mythform.WhatGender = users_sex;
    mythform.Show();
}
```

```
public string UpdateAssistByname(string username, int assistant)
{
    SqlConnection conn = new SqlConnection();
    conn.ConnectionString =
@"Server=localhost;Database=greekmythology;Trusted_Connection=True;";
    try
    {
        conn.Open();
    }
    catch (Exception e)
    {
        MessageBox.Show(e.ToString());
    }

    SqlParameter p1 = new SqlParameter("@username",
SqlDbType.NVarChar);
    p1.Value = username.ToString();
    SqlParameter p2 = new SqlParameter("@assistant",
SqlDbType.Int);
    p2.Value = assistant;

    SqlCommand sqlcom = new SqlCommand("update User_Profile
set assistant=@assistant where username=@username", conn);
    sqlcom.Parameters.Add(p1);
    sqlcom.Parameters.Add(p2);
    sqlcom.ExecuteNonQuery();

    conn.Close();

    return null;
}
```

Κωδ. 4.10 Αλλαγή θεού

4.7 Εμφάνιση λογαριασμού - myaccount.cs

Μία ακόμα δυνατότητα που παρέχεται στον χρήστη είναι η απεικόνιση κάποιων συγκεντρωτικών στοιχείων. Συγκεκριμένα πατώντας το κουμπί "Ο λογαριασμός μου" μπορεί να δει την ημερομηνία εγγραφής του στο σύστημα, την ηλικία την οποία είχε δηλώσει, το ονοματεπώνυμό του, τον αριθμό των τεστ που έχει ολοκληρώσει, καθώς και την μεγαλύτερη και τη μικρότερη βαθμολογία σε κάθε κεφάλαιο που έχει επιτύχει. Για να υλοποιηθεί αυτή η λειτουργία δημιουργήθηκε ένα SQL statement το οποίο τραβάει τα στοιχεία εγγραφής του

χρήστη από τον πίνακα User_Profile αυτά αποθηκεύονται σε έναν πίνακα που δημιουργούμε και κάθε στοιχείο του πίνακα εμφανίζεται στο αντίστοιχο label.

4.8 Αναλυτικά στοιχεία - Stats.cs

Το κουμπί “Αναλυτικά στοιχεία” που βρίσκεται στο κεντρικό παράθυρο της εφαρμογής (mythlab.cs) οδηγεί σε μία νέα φόρμα την stats.cs στην οποία περιέχονται οι βαθμολογίες όλων των τεστ που έχουν κάνει οι μαθητές. Ο χρήστης μπορεί να επιλέξει κεφάλαιο και να δει πόσα τεστ έχει κάνει, πότε τα έκανε και τι βαθμολογία είχε πάρει. Δημιουργήσαμε ένα SQL Query το οποίο επιστρέφει όσες εγγραφές υπάρχουν στον πίνακα results και το πεδίο chapter έχει την τιμή του επιλεγμένου κεφαλαίου. Η εμφάνιση των αποτελεσμάτων γίνεται με βάση την ημερομηνία που είχε γίνει το τεστ. Για την απεικόνισή του χρησιμοποιήθηκε ένα datagridview το οποίο εμφανίζει τα περιεχόμενα μιας βάσης δεδομένων. Το datagridview είναι ένα control των Windows forms που χρησιμοποιεί μια μορφή πίνακα για την παρουσίαση των εγγραφών. Το datagridview αποτελείται από της εξής στήλες:

- Κεφάλαιο (select chapter as Κεφάλαιο)
- Βαθμολογία (rate as Βαθμός)
- Ημερομηνία (datetest as Ημερομηνία)
- Διάρκεια του τεστ (timetest as Διάρκεια)

```
dataGridView1.Visible = true;
        dataGridView1.DataSource = bindingSource1;

        GetData("select chapter as Κεφάλαιο,rate as Βαθμός
,datetest as Ημερομηνία ,timetest as Διάρκεια from results where
chapter=1 and id=" + userID + "order by Ημερομηνία");
```

Κωδ. 4.11 Επιλογή αποτελεσμάτων

Για βαθμολογίες κάτω του 50% οι στήλες χρωματίζονται με κόκκινο χρώμα. Για βαθμολογίες από 50% μέχρι 65% με μπλε και για βαθμολογίες άνω του 65% με πράσινο.

```
if (Convert.ToInt32(r.Cells["Βαθμός"].Value) < 50)
    {
        r.DefaultCellStyle = red;
```

```

    }
    if ((Convert.ToInt32(r.Cells["Βαθμός"].Value) >= 50)
    && (Convert.ToInt32(r.Cells["Βαθμός"].Value) <= 65))
    {
        r.DefaultCellStyle = blue;
    }
    if (Convert.ToInt32(r.Cells["Βαθμός"].Value) > 65)
    {
        r.DefaultCellStyle = green;
    }

```

Κωδ. 4.12 Εμφάνιση Χρώματος ανάλογα με τη βαθμολογία

Το κουμπί “όλα τα τεστ” εμφανίζει τα αποτελέσματα των τεστ από όλα τα κεφάλαια που έχει κάνει ο μαθητής. Όταν ο μαθητής επιλέγει συγκεκριμένο κεφάλαιο τότε στο SQL Query που γράψαμε ορίζουμε και το chapter με την τιμή που έχει επιλέξει για παράδειγμα αν έχει επιλέξει το κεφάλαιο 1 τότε θα πρέπει να γράψουμε “where chapter=1” ενώ αν επιλέξει “όλα τα τεστ” θα γράψουμε τα παρακάτω:

```

dataGridView1.Visible = true;
dataGridView1.DataSource = bindingSource1;

GetData("select chapter as Κεφάλαιο,rate as Βαθμός
,datetest as Ημερομηνία ,timetest as Διάρκεια from results where id="
+ usersID + "order by Ημερομηνία");

```

Κωδ. 4.13 Εμφάνιση όλων των αποτελεσμάτων

4.9 Έλεγχος γνώσεων - FormTests.cs

Η φόρμα στην οποία ο μαθητής μπορεί να κάνει έλεγχο των γνώσεων για κάθε κεφάλαιο ξεχωριστά είναι η FormTests.cs. Η φόρμα αυτή χωρίζεται σε δύο μέρη με ένα panel, το επάνω μέρος είναι στατικό όπου και εμφανίζεται το logo και το username του χρήστη. Το κάτω μέρος είναι δυναμικό και εκεί εμφανίζονται οι ερωτήσεις του κεφαλαίου που έχει επιλέξει και ο χρόνος που έχει στη διάθεσή του.

4.9.1 Υπολογισμός χρόνου του τεστ

Ο χρόνος που έχει στη διάθεσή του εξαρτάται και από την ηλικία του. Σημαντικό ρόλο στη προσαρμοστικότητα του συστήματος στο χρόνο, έπαιξε το γεγονός ότι τα παιδιά μικρότερης ηλικίας και συγκεκριμένα κάτω των οκτώ ετών έχουν χαμηλότερη αντίληψη καθώς επίσης και δυσκολία στην ανάγνωση, άρα χρειάζονται περισσότερο χρόνο για την ανάγνωση μίας ερώτησης και την κατανόηση αυτής. Για το λόγο αυτό για τους χρήστες που είναι μέχρι και οκτώ ετών ο χρόνος ορίστηκε στα 15 λεπτά και για τους χρήστες άνω των οκτώ ετών ορίστηκε

στα 10 λεπτά. Για την υλοποίησή του έχοντας φορτωθεί το username του χρήστη από την προηγούμενη φόρμα (mythlab.cs) γίνεται έλεγχος στον πίνακα User_profile για την ηλικία που είχε δηλώσει ο χρήστης κατά την εγγραφή του. Ο παραπάνω έλεγχος πραγματοποιείται με το SQL Query που αναφέρεται παρακάτω. Στη συνέχεια γίνεται ο έλεγχος της ηλικίας για την προσαρμογή του χρόνου.

```

DataRow AgeRow;
    DataTable Agetable = new DataTable();
    SqlQueries getAge = new SqlQueries();
    getAge.GetAgeByName(usernameLabel);
    Agetable = getAge.GetAge();
    AgeRow = Agetable.Rows[0];

    users_age = Convert.ToInt32(AgeRow["age"]);
    string ageuser = users_age.ToString();
    if (users_age > 8)
    {
       agemins = 9;
        mins = 9;
    }
    else
    {
       agemins = 14;
        mins = 14;
    }

public DataTable GetAge;
public DataTable GetAgeByName(string username)
{
    SqlConnection conn = new SqlConnection();
    conn.ConnectionString =
@"Server=localhost;Database=greekmythology;Trusted_Connection=True;";
    try
    {
        conn.Open();
    }
    catch (Exception e)
    {
        MessageBox.Show(e.ToString());
    }
    SqlParameter p1 = new SqlParameter("@username",
SqlDbType.NVarChar);
    p1.Value = username.ToString();

    SqlCommand sqlcom = new SqlCommand("select * from
user_profile where username=@username;", conn);
    sqlcom.Parameters.Add(p1);

```

```

// sqlcom.ExecuteNonQuery();
SqlDataAdapter AgeAdap = new SqlDataAdapter();
GetAge = new DataTable();
AgeAdap.SelectCommand = sqlcom;
AgeAdap.Fill(GetAge);
conn.Close();

return GetAge;
}

```

Κωδ. 4.14 Υπολογισμός διαθέσιμου χρόνου διεξαγωγής τεστ

Δημιουργήθηκε η ανάγκη να χρησιμοποιήσουμε δύο διαφορετικές μεταβλητές για τον χρόνο, την *agemins* και την *mins*. Η δεύτερη αρχικοποιείται ανάλογα με την ηλικία του χρήστη, οπότε για τους χρήστες που είναι μεγαλύτεροι των οκτώ ετών έχει οριστεί η τιμή 9 και για τους υπόλοιπους έχει οριστεί η τιμή 14. Ο λόγος που δεν ορίσαμε τις τιμές 10 και 15 αντίστοιχα είναι διότι στο τελευταίο λεπτό της αντίστροφης μέτρησης ο χρόνος που δείχνει το χρονόμετρο είναι 0 λεπτά και 59 δευτερόλεπτα, οπότε μόλις ο χρόνος σταματήσει, στα 0 λεπτά και στα 0 δευτερόλεπτα έχουν ολοκληρωθεί 10 ή 15 λεπτά αντίστοιχα. Εκτός από τα *mins* έχουμε ορίσει και άλλη μία μεταβλητή που αντιστοιχεί στα δευτερόλεπτα του timer, τα *secs* με αρχική τιμή 59. Όταν τρέχει ο timer (on timer tick) κάθε η τιμή της μεταβλητής *secs* μειώνεται κατά 1. Για τιμή ίση με 0 τότε τα *mins* μειώνονται κατά 1 και τα *secs* παίρνουν ξανά την τιμή 59 για να αρχίσει η μέτρηση του επόμενου λεπτού. Η διαδικασία αυτή επαναλαμβάνεται μέχρις ότου το *mins* πάρει την τιμή -1. Για τιμή ίση με -1 έχει τελειώσει ο χρόνος, όπου και εμφανίζεται αντίστοιχο μήνυμα στο χρήστη για τη μη ολοκλήρωση του τεστ. Όταν απομένει στο χρήστη ένα λεπτό για την ολοκλήρωση του τεστ εμφανίζεται προειδοποιητικό μήνυμα για την άμεση ολοκλήρωση.

```

if (secs == 0)
{
    secs = 59;
    if (mins < 10)
    {
        label1.Text = "0" + mins.ToString();
    }
    else
    {
        label1.Text = mins.ToString();
    }
    label3.Text = secs.ToString();
    mins--;
    if (mins == agemins)

```

```
        {
            label5.Visible = true;
        }
    }

    else
    {
        secs--;

        if (secs < 10)
        {
            label3.Text = "0" + secs.ToString();
        }
        else
        {
            label3.Text = secs.ToString();
        }
    }
}
```

Κωδ. 4.15 Διαθέσιμος χρόνος ανάλογα με την ηλικία

Για την αποθήκευση του αποτελέσματος του τεστ έχει οριστεί η μεταβλητή `passtest` η οποία αποθηκεύει την τιμή 1 στον πίνακα `results` στη στήλη `pass` στην περίπτωση που ο χρήστης έχει προλάβει να απαντήσει όλες τις ερωτήσεις του τεστ, διαφορετικά παίρνει την τιμή 0. Ακόμη και στην περίπτωση που ο χρήστης δεν έχει προλάβει να ολοκληρώσει το τεστ αποθηκεύονται τα αποτελέσματα, παρακάτω αναλυτική περιγραφή της αποθήκευσης των αποτελεσμάτων. Αν ο χρόνος προλάβει να ολοκληρώσει το τεστ στον απαιτούμενο χρόνο, ο υπολογισμός του χρόνου ολοκλήρωσης γίνεται ως εξής. Όπως αναφέρθηκε παραπάνω έχουμε ορίσει μια μεταβλητή που ονομάζεται `agemins`, από την `agemins` αφαιρείται η τιμή που έχει η μεταβλητή `mins` για να υπολογιστούν τα λεπτά που διήρκεσε το τεστ. Για τον υπολογισμό των δευτερολέπτων από την τιμή 60 αφαιρούμε την τιμή που έχει η μεταβλητή `secs`. Οι δύο αυτές μεταβλητές αποθηκεύονται στη βάση δεδομένων στον πίνακα `results` στη στήλη `timetest`.

```
Timeoutoftest.Stop();
timer1.Stop();
int completemins = agemins - (mins + 1);
int completesecs = 60 - secs;
```

Κωδ. 4.16 Τέλος χρόνου

4.9.2 Επίπεδα δυσκολίας

Οι ερωτήσεις είναι αποθηκευμένες στον πίνακα questions. Ο λόγος που τις αποθηκεύσαμε στη βάση είναι γιατί σε κάθε νέο τεστ που κάνει ο μαθητής έπρεπε να επιστρέφονται διαφορετικές ερωτήσεις με τυχαία σειρά. Οι ερωτήσεις κάθε κεφαλαίου είναι χωρισμένες σε τρία επίπεδα δυσκολίας ως εξής: εύκολες, μέτριες και δύσκολες. Στη βάση δεδομένων η στήλη level1 χαρακτηρίζει το επίπεδο δυσκολίας και παίρνει την τιμή 1, 2 ή 3 για εύκολο, μέτριο και δύσκολο αντίστοιχα. Το τεστ αποτελείται από δέκα ερωτήσεις, οι ερωτήσεις αυτές μπορεί να είναι πολλαπλής επιλογής, συμπλήρωσης κενού ή σωστού λάθους. Όταν ο χρήστης εξεταστεί για πρώτη φορά σε κάποιο κεφάλαιο τότε εμφανίζονται έξι ερωτήσεις επιπέδου 1, τρεις ερωτήσεις επιπέδου 2 και μία ερώτηση επιπέδου 3. Για κάθε επόμενο τεστ οι ερωτήσεις εμφανίζονται ως εξής:

- Για βαθμολογία μέχρι και 60% εμφανίζονται έξι ερωτήσεις επιπέδου 1, τρεις ερωτήσεις επιπέδου 2 και μία ερώτηση επιπέδου 3.
- Για βαθμολογία από 61% μέχρι και 75% εμφανίζονται τέσσερις ερωτήσεις επιπέδου 1, τέσσερις ερωτήσεις επιπέδου 2 και δύο ερωτήσεις επιπέδου 3.
- Για βαθμολογία από 75% μέχρι και 86% εμφανίζονται τρεις ερωτήσεις επιπέδου 1, τέσσερις ερωτήσεις επιπέδου 2 και τρεις ερωτήσεις επιπέδου 3.
- Για βαθμολογία από 87% και άνω εμφανίζεται μία ερώτηση επιπέδου 1, πέντε ερωτήσεις επιπέδου 2 και τέσσερις ερωτήσεις επιπέδου 3.

```
if (this.PickDifficultyLevel < 61)
{
    level1questions = 6;
    level2questions = 3;
    level3questions = 1;
}
if (this.PickDifficultyLevel >= 61 &&
this.PickDifficultyLevel <= 75)
{
    level1questions = 4;
    level2questions = 4;
    level3questions = 2;
}
if (this.PickDifficultyLevel > 75 &&
this.PickDifficultyLevel <= 86)
{
    level1questions = 3;
    level2questions = 4;
    level3questions = 3;
}
if (this.PickDifficultyLevel >= 87)
{
    level1questions = 1;
    level2questions = 5;
```

```

        level3questions = 4;
    }

```

Κωδ. 4.17 Επίπεδα δυσκολίας

4.9.3 Εμφάνιση ερωτήσεων

Όπως αναφέραμε το ποιες ερωτήσεις θα εμφανιστούν εξαρτάται από τη βαθμολογία που έχει επιτύχει ο χρήστης. Γι' αυτό λοιπόν γίνεται αναζήτηση στη βάση δεδομένων για τη μεγαλύτερη βαθμολογία που έχει επιτύχει στο επιλεγμένο κεφάλαιο. Έχοντας βρει λοιπόν τη μεγαλύτερη βαθμολογία δημιουργούμε ένα νέο SQL Query το οποίο επιστρέφει τις εγγραφές του πίνακα questions ανάλογα με το επιλεγμένο κεφάλαιο και ανάλογα με το επίπεδο που αντιστοιχεί στη βαθμολογία, όπως έχει αναλυθεί παραπάνω.

```

public DataTable GetTable(int chapter, int level1q, int level2q, int
level3q, int q1, int q2, int q3)
{
    if (chapter == 5)
    {
        int qnum1 = q1;
        int qnum2 = q2;
        int qnum3 = q3;

        SqlConnection conn = new SqlConnection();
        conn.ConnectionString =
@"Server=localhost;Database=greekmythology;Trusted_Connection=True;";
        try
        {
            conn.Open();
        }
        catch (Exception e)
        {

            MessageBox.Show(e.ToString());

        }
        dtGet = new DataTable();

        SqlParameter p2 = new SqlParameter("@level1q",
SqlDbType.Int);
        p2.Value = Convert.ToInt32(level1q);
        SqlCommand sqlcom = new SqlCommand(" SELECT TOP " +
qnum1 + " * FROM questions where level1=@level1q ORDER BY NEWID()",

```

```
conn);

        sqlcom.Parameters.Add(p2);

        SqlDataAdapter da = new SqlDataAdapter();

        da.SelectCommand = sqlcom;

        //γεμίζει το πίνακα με τις εγγραφές
        da.Fill(dtGet);

        SqlParameter p3 = new SqlParameter("@level2q",
SqlDbType.Int);
        p3.Value = level2q;

        SqlCommand sqlcom1 = new SqlCommand("SELECT TOP " +
qnum2 + " * FROM questions where level1=@level2q ORDER BY NEWID()",
conn);

        sqlcom1.Parameters.Add(p3);

        da.SelectCommand = sqlcom1;
        da.Fill(dtGet);

        SqlParameter p5 = new SqlParameter("@level3q",
SqlDbType.Int);
        p5.Value = level3q;

        SqlCommand sqlcom2 = new SqlCommand("SELECT TOP " +
qnum3 + " * FROM questions where level1=@level3q ORDER BY NEWID()",
conn);

        sqlcom2.Parameters.Add(p5);

        da.SelectCommand = sqlcom2;
        da.Fill(dtGet);

        conn.Close();
        return dtGet;
    }
    else
    {

        int qnum1 = q1;
        int qnum2 = q2;
        int qnum3 = q3;
```

```

        SqlConnection conn = new SqlConnection();
        conn.ConnectionString =
@"Server=localhost;Database=greekmythology;Trusted_Connection=True;";
        try
        {
            conn.Open();
        }
        catch (Exception e)
        {

            MessageBox.Show(e.ToString());

        }
        dtGet = new DataTable();
        SqlParameter p1 = new SqlParameter("@chapter",
SqlDbType.Int);
        p1.Value = Convert.ToInt32(chapter);

        SqlParameter p2 = new SqlParameter("@level1q",
SqlDbType.Int);
        p2.Value = Convert.ToInt32(level1q);
        SqlCommand sqlcom = new SqlCommand(" SELECT TOP " +
qnum1 + " * FROM questions where level1=@level1q and chapter=@chapter
ORDER BY NEWID()", conn);
        sqlcom.Parameters.Add(p1);
        sqlcom.Parameters.Add(p2);

        SqlDataAdapter da = new SqlDataAdapter();

        da.SelectCommand = sqlcom;

        da.Fill(dtGet);

        SqlParameter p3 = new SqlParameter("@level2q",
SqlDbType.Int);
        p3.Value = level2q;
        SqlParameter p4 = new SqlParameter("@chapter",
SqlDbType.Int);
        p4.Value = Convert.ToInt32(chapter);
        SqlCommand sqlcom1 = new SqlCommand("SELECT TOP " +
qnum2 + " * FROM questions where level1=@level2q and chapter=@chapter
ORDER BY NEWID()", conn);
        sqlcom1.Parameters.Add(p4);
        sqlcom1.Parameters.Add(p3);

        da.SelectCommand = sqlcom1;
        da.Fill(dtGet);

```

```

        SqlParameter p5 = new SqlParameter("@level3q",
        SqlDbType.Int);
        p5.Value = level3q;
        SqlParameter p6 = new SqlParameter("@chapter",
        SqlDbType.Int);
        p6.Value = chapter;
        SqlCommand sqlcom2 = new SqlCommand("SELECT TOP " +
        qnum3 + " * FROM questions where level1=@level3q and chapter=@chapter
        ORDER BY NEWID()", conn);
        sqlcom2.Parameters.Add(p6);
        sqlcom2.Parameters.Add(p5);

        da.SelectCommand = sqlcom2;
        da.Fill(dtGet);

        conn.Close();
        return dtGet;
    }

```

Κωδ. 4.18 Εμφάνιση ερωτήσεων

Αφού βρέθηκαν οι ερωτήσεις από τις οποίες θα αποτελείται το τεστ έπρεπε να εξεταστεί με ποιο τρόπο θα εμφανίζονται στη φόρμα. Μόλις φορτώνεται το τεστ σχεδιάζονται εκείνη τη στιγμή τα labels των ερωτήσεων και τα radiobuttons για τις απαντήσεις πολλαπλής επιλογής και σωστού λάθους καθώς επίσης τα textboxes για τις απαντήσεις συμπλήρωσης κενού. Ορίσαμε λοιπόν σε ποια τοποθεσία θα βρίσκονται οι ερωτήσεις και σε ποια οι απαντήσεις. Σε κάθε τεστ η τοποθεσία των ερωτήσεων, απαντήσεων και κουμπιών είναι ίδια έτσι ώστε ο μαθητής να αποτυπώσει τη γεωγραφία της οθόνης.

Επειδή στις ερωτήσεις πολλαπλής επιλογής και σωστού λάθους οι απαντήσεις είναι αποθηκευμένες με την ίδια σειρά (αρχικά αποθηκεύεται η σωστή και συνέχεια οι λανθασμένες) δημιουργήθηκε η ανάγκη να επιστρέφονται οι απαντήσεις με διαφορετική σειρά έτσι ώστε η πρώτη προτεινόμενη απάντηση να μην είναι πάντα και η σωστή. Για να υλοποιηθεί προγραμματιστικά δημιουργήσαμε μια νέα κλάση την ShuffleAnswers.cs. Αυτό που κάνει η συνάρτηση είναι να χρησιμοποιεί τα millisecond, από την ώρα του συστήματος, και βάσει αυτών να παράγει τυχαίες τιμές ορίζοντας έτσι τις θέσεις των απαντήσεων. Συνεπώς η σειρά εμφάνισης κάθε φορά θα είναι διαφορετική.


```
public object[] AnswersQuestions(int QuestStringCount, string
QuestionType)

{
    //όταν καλείται παίρνει σαν ορίσματα το μήκος του string
    ερώτησης και τι τύπου ερώτηση είναι

    //χρησιμοποιεί την ώρα του συστήματος και κρατάει τα
    milliseconds

    DateTime dt = new DateTime();

    dt = DateTime.Now;

    int d = dt.Millisecond;

    //δημιουργεί μια τυχαία τιμη απο το μηδέν μέχρι τη τιμή
    των milliseconds την οποία περνάει στις μεταβλητές

    //που θα ορίσουν ποια απάντηση θα πάρει τυχαία μία απο τις
    καθορισμένες τιμές πάνω στη φόρμα

    Random RandomNum = new Random();

    int RandPos1 = RandomNum.Next(0, d);
    int RandPos2 = RandomNum.Next(0, d);
    int RandPos3 = RandomNum.Next(0, d);
    int RandPos4 = RandomNum.Next(0, d);

    //ελέγχει το τύπο της ερώτησης

    if (QuestionType.ToString() == "rw")
    {
        if (RandPos1 >= RandPos2)
        {
            pos1 = 180;

            pos2 = 210;

        }
    }
}
```

```
else
{
    pos1 = 210;
    pos2 = 180;
}
}

if (QuestionType.ToString() == "mc")
{
    int[] positions={180,210,250,290};
    int j = RandomNum.Next(0,3);
    for (int ii = 3; ii > 0; ii--)
    {
        int temp = positions[ii];
        positions[ii] = positions[j];
        positions[j] = temp;
    }

    pos1 = positions[0];
    pos2 = positions[1];
    pos3 = positions[2];
    pos4 = positions[3];
}

//επιστρέφει τις μεταβλητές που περιέχουν την θέση της
κάθε απάντησης

return new object[]{pos1,pos2,pos3,pos4};
```

}

Κωδ. 4.19 Εμφάνιση ερωτήσεων με τυχαία σειρά

4.9.4 Έλεγχος ορθότητας της απάντησης και μετάβαση στην επόμενη ερώτηση.

Για να επιλέξει ο χρήστης την επόμενη ερώτηση θα πρέπει απαραίτητα να δώσει κάποια απάντηση, μέχρι τότε το κουμπί “Επόμενο” είναι απενεργοποιημένο. Κατά τη διάρκεια εκτέλεσης ενός test καταγράφεται αν ο χρήστης είναι σίγουρος για τις απαντήσεις του και αν είναι προσεκτικός, έτσι ώστε να επιτευχθεί η μοντελοποίηση και η προσαρμοστικότητα του λογισμικού. Πιο συγκεκριμένα γίνεται καταγραφή τριών παραμέτρων:

- I. Συχνή αλλαγή των radiobuttons
Οι ερωτήσεις πολλαπλής επιλογής και οι ερωτήσεις τύπου «Σωστό- Λάθος» είναι ορισμένες με radiobuttons. Αν ο χρήστης αλλάζει συνεχώς την απάντηση του, δηλαδή επιλέγει διαφορετικά radiobuttons και δεν επιλέγει αμέσως την απάντηση του τότε αυτό σημαίνει ότι δεν είναι σίγουρος για την απάντηση του.
- II. Μεγάλος αριθμός Backspaces
Αν ο χρήστης στις ερωτήσεις συμπλήρωσης κενού πατήσει στο πληκτρολόγιο του πάνω από 10 backspaces συνολικά σε όλο το τεστ, δείχνει ότι δεν είναι πολύ προσεκτικός κατά την πληκτρολόγηση της απάντησης του.
- III. Μεγάλος αριθμός κινήσεων του ποντικιού
Γίνεται έλεγχος για το κατά πόσο ο χρήστης κινεί το ποντίκι του σε υπερβολικό βαθμό. Μετά από μέτρηση που έγινε σε ένα τεστ ένα χρήστης δεν ξεπερνά τις 250 κινήσεις. Ορίσαμε ως κατώφλι τις 250 κινήσεις αν ο χρήστης τις ξεπεράσει εμφανίζεται αντίστοιχο μήνυμα που τον ενημερώνει ότι δεν ήταν ιδιαίτερα αφοσιωμένος.

Τα παραπάνω στοιχεία αποθηκεύονται στον πίνακα results που χρησιμοποιούνται στην αξιολόγηση του χρήστη και θα αναλυθούν αργότερα. Στη περίπτωση που υπάρχει textbox για την εισαγωγή κειμένου στις ερωτήσεις συμπλήρωσης κενού και η επιλεγμένη γλώσσα του πληκτρολογίου είναι οποιαδήποτε άλλη εκτός της ελληνικής εμφανίζεται αντίστοιχο μήνυμα που ενημερώνει τον χρήστη για την εισαγωγή ελληνικών χαρακτήρων.

```
public int checklang = 0;
private void tb_KeyPress(object sender, KeyPressEventArgs e)
{
    int lenghtstring = tb.Text.ToString().Length;
    InputLanguage myCurrentLanguage =
InputLanguage.CurrentInputLanguage;
    CultureInfo myCultureInfo = myCurrentLanguage.Culture;
```

```
string hh = myCultureInfo.EnglishName;

if ((hh != "Greek (Greece)") && checklang == 0) {
    MessageBox.Show("Άλλαξε τη γλώσσα στα ελληνικά 'cntrl+Shift'"); }

if (e.KeyChar == (Char)Keys.Delete)
{
    BackspacesCounter = BackspacesCounter + lenghtstring;
}
if (e.KeyChar == '\b')
{
    checklang = 1;
    BackspacesCounter = BackspacesCounter + 1;
}
}
```

Κωδ. 4.20 Έλεγχος γλώσσας εισαγωγής

Όπως κατά την ανάγνωση του περιεχομένου έτσι και στην διεξαγωγή του τεστ γίνεται έλεγχος στην αφοσίωση του μαθητή. Σε περίπτωση που ο μαθητής ελέγχει οτιδήποτε άλλο εκτός του τεστ ενεργοποιείται ένας δεύτερος timer ο Timeoutofstest ο οποίος χρησιμοποιείται να μετρήσει για πόση ώρα ο μαθητή δεν είναι αφοσιωμένος στο τεστ. Αν συμπληρωθούν 30 δευτερόλεπτα εμφανίζεται σχετικό μήνυμα στο χρήστη που τον παροτρύνει να επιστρέψει στο τεστ.

```

if ((Form.ActiveForm == null) && (!panelTest1.ContainsFocus) &&
(!buttonNextQ1.ContainsFocus) && timehasstr == 0 &&
(!this.ContainsFocus) && (!groupBoxCh1.ContainsFocus))
{
    timehasstr = 1;

    Timeoutoftest.Start();
    Timeoutoftest.Tick += new
EventHandler(Timeoutoftest_Tick);
}
if ((this.ContainsFocus || buttonNextQ1.ContainsFocus ||
Form.ActiveForm != null) && timehasstr == 1)
{
    Timeoutoftest.Stop();
    timehasstr = 0;
}

void Timeoutoftest_Tick(object sender, EventArgs e)
{
    if (timeoutsecs < 59)
    {
        timeoutsecs = timeoutsecs + 1;
        totaltimeout = timeoutmins + ":" + timeoutsecs;
    }

    if (timeoutsecs == 59)
    {
        timeoutsecs = 0;
        timeoutmins = timeoutmins + 1;
        totaltimeout = timeoutmins + ":" + timeoutsecs;
    }

    if (countermessag > 30) { countermessag = 0;
MessageBox.Show("Μάλλον έχεις στρέψει αλλού τη προσοχή σου, επέστρεψε
στο τεστ"); }
    countermessag = countermessag + 1;
}

```

Κωδ. 4.21 Έλεγχος αφοσίωσης μαθητή

Μόλις ο χρήστης επιλέξει το κουμπί “Επόμενη” γίνεται έλεγχος ορθότητας της απάντησης που έχει δώσει. Η απάντηση που δίνει αποθηκεύεται σε μία μεταβλητή και ελέγχεται αν αυτή είναι ίδια με την εγγραφή που υπάρχει στη στήλη `answer_right` για την αντίστοιχη ερώτηση στον πίνακα `questions`. Δημιουργήθηκε ένας μετρητής ο `rightAns` ο οποίος μετράει τις σωστές απαντήσεις. Στη συνέχεια φορτώνεται η επόμενη ερώτηση.

4.9.5 Υπολογισμός βαθμολογίας

Η βαθμολογία υπολογίζεται αφού ο χρήστης ολοκληρώσει το τεστ ή αφού λήξει ο χρόνος πριν την ολοκλήρωση. Ο υπολογισμός της βαθμολογίας ολοκληρώνεται ακολουθώντας τα επόμενα βήματα.

- Υπολογισμός των ερωτήσεων που είναι επιπέδου 1, 2 και 3
- Πολλαπλασιάζουμε τον αριθμό των ερωτήσεων κάθε επιπέδου επί 10, 20, 30 αντίστοιχα
- Αθροίζουμε τα τρία γινόμενα (`totalcount`)
- Για κάθε σωστή απάντηση που δίνεται ο χρήστης παίρνει 10 μονάδες για τις ερωτήσεις επιπέδου 1, 20 για τις επιπέδου 2 και 30 για τις επιπέδου 3 (μεταβλητή `resultpc`)
- Η `resultpc` διαιρείται με το `totalcount` και πολλαπλασιάζεται με 100

```
totalcount = levele1questions * 10 + levele2questions * 20 +
levele3questions * 30;
if (rw["level1"].ToString() == "1")
{
    resultpc = resultpc + 10;
    rightAns = rightAns + 1;
}
else if (rw["level1"].ToString() == "2")
{
    resultpc = resultpc + 20;
    rightAns = rightAns + 1;
}
else if (rw["level1"].ToString() == "3")
{
    resultpc = resultpc + 30;
    rightAns = rightAns + 1;
}
totalcount = (resultpc * 100) / totalcount;
```

Κωδ. 4.23 Υπολογισμός βαθμολογίας

4.9.6 Ολοκλήρωση του τεστ

Ένα τεστ ολοκληρώνεται όταν ο χρήστης απαντήσει και τις δέκα ερωτήσεις που υπάρχουν στο τεστ ή όταν τελειώσει ο χρόνος. Όταν συμβεί κάποιο από τα δύο τότε αποθηκεύονται στον πίνακα results της βάσης δεδομένων τα εξής στοιχεία:

- Το ID του χρήστη (iduser)
- Το κεφάλαιο στο οποίο ανήκει το τεστ που έκανε (chaptertest)
- Η βαθμολογία (ratetest)
- Η ημερομηνία διεξαγωγής του τεστ (dateT)
- Ο χρόνος που του πήρε για να ολοκληρώσει το τεστ (timeT)
- Αν πέρασε το τεστ αποθηκεύεται η τιμή 1 διαφορετικά η τιμή 0 (passtest)
- Αν ολοκλήρωσε το τεστ πριν λήξει ο χρόνος αποθηκεύεται η τιμή 1 διαφορετικά η τιμή 0 (completetest)
- Ο αριθμός των radio changes (radioscnt)
- Ο αριθμός των backspaces που έχουν πατηθεί (backscnt)
- Οι κινήσεις του ποντικιού (mousemvs)
- Τα λεπτά που έκανε για να ολοκληρώσει το τεστ (totalmins)
- Και τα δευτερόλεπτα (totalsecs)

```
public string SaveResults(int iduser, int chaptertest, int
ratetest, string dateT, string timeT, int passtest, int completetest,
int radioscnt, int backscnt, int mousemvs, int totalmins, int
totalsecs)
{

    SqlConnection conn = new SqlConnection();
    conn.ConnectionString =
@"Server=localhost;Database=greekmythology;Trusted_Connection=True;";
    try
    {
        conn.Open();
    }
    catch (Exception e)
    {

        MessageBox.Show(e.ToString());

    }
}
```

Κωδ. 4.24 Ολοκλήρωση τεστ και αποθήκευση αποτελεσμάτων

Με την αποθήκευση των στοιχείων εμφανίζεται η analytics.cs

4.10 Εμφάνιση αποτελεσμάτων στον χρήστη (Analytics.cs)

Όταν ολοκληρωθεί ένα τεστ εμφανίζονται στον χρήστη κάποια στοιχεία που αφορούν την απόδοσή του αλλά και συμβουλές που θα τον βοηθήσουν στο να κατανοήσει καλύτερα το υλικό και να επιτύχει το στόχο του που είναι η μάθηση.

Από την φόρμα formtests.cs φορτώνεται το όνομα του χρήστη, η βαθμολογία που πέτυχε στο τεστ, τα νομίσματα τα οποία έχει συγκεντρώσει, ο αριθμός των σωστών απαντήσεων, ο χρόνος που του πήρε να ολοκληρώσει το τεστ καθώς και τα Mousemoves τα radiochanges και τα backspaces.

Στο κέντρο της φόρμας αυτής υπάρχει ένα tab με την ονομασία αποτελέσματα στο οποίο εμπεριέχονται τα στοιχεία βαθμολογία, χρόνος, σωστές απαντήσεις, η διάρκεια του τεστ, ο αριθμός των νομισμάτων που κέρδισε και ο συνολικός αριθμός των νομισμάτων. Επιπλέον στο συγκεκριμένο tab εμφανίζεται και ένα flash με κάποιον θεό ανάλογα με την απόδοσή του.

Ο αριθμός των νομισμάτων που κερδίζει εξαρτάται από την βαθμολογία του. Έτσι έχουμε ορίσει:

- Για βαθμολογία κάτω από 50% συγκεντρώνει 10 νομίσματα
- Για βαθμολογία από 50% μέχρι 60% συγκεντρώνει 20 νομίσματα
- Για βαθμολογία από 60% μέχρι 85% συγκεντρώνει 30 νομίσματα
- Για βαθμολογία από 85% μέχρι 100% συγκεντρώνει 50 νομίσματα
- Για βαθμολογία από 100% συγκεντρώνει 100 νομίσματα

Δίπλα από το tab “Αποτελέσματα” υπάρχει και το tab “Συμβουλές”. Κάνοντας κλικ στο tab αυτό δίνονται κάποιες συμβουλές στον χρήστη. Συγκεκριμένα γίνονται οι εξής έλεγχοι:

- Αν τα radiochanges είναι 150 σημαίνει ότι ο χρήστης αλλάζει πολύ συχνά την απάντησή του μέχρι να δώσει την τελική και να προχωρήσει στην επόμενη. Αυτό δείχνει ότι δεν είναι σίγουρος για την απάντησή του και εμφανίζεται η συμβουλή: “Την επόμενη φορά να είσαι πιο σίγουρος για τις απαντήσεις σου!”
- Αν τα backspaces είναι πάνω από 20 σημαίνει ότι ο χρήστης ίσως να μην ξέρει τη σωστή απάντηση και να μην αισθάνεται σίγουρος για την απάντηση που υποβάλει. Στην περίπτωση αυτή εμφανίζεται το μήνυμα “Να είσαι πιο προσεκτικός όταν δίνεις τις απαντήσεις σου.”
- Στην περίπτωση που οι κινήσεις του ποντικιού είναι πάρα πολλές, συγκεκριμένα πάνω από 500, φανερώνει ότι ο χρήστης δεν είναι συγκεντρωμένος και ίσως είναι και αγχωμένος. Η συμβουλή για αυτήν την περίπτωση είναι η εξής: “Μην ξεχνάς όταν ξεκινάς το τεστ να είσαι αφοσιωμένος! Έτσι θα πετύχεις καλύτερο αποτέλεσμα σε λιγότερο χρόνο!!!!”
- Ανάλογα με το ποσοστό της βαθμολογίας του εμφανίζονται μηνύματα που τον ενθαρρύνουν να συνεχίσει την προσπάθειά του. Για παράδειγμα για βαθμολογίες από 50% μέχρι 60% εμφανίζεται το μήνυμα: “Διάβασε περισσότερο!! Η σωστή μελέτη φέρνει πάντα επιτυχία!!”

- Αν έχει περάσει το τεστ χωρίς όμως να έχει προλάβει να απαντήσει σε όλες τις ερωτήσεις το πρόγραμμα τον συμβουλεύει να διαχειρίζεται καλύτερα το χρόνο του.
- Αν ο χρόνος του είναι μεγαλύτερος ή ίσος με εννέα λεπτά τότε δίνεται η συμβουλή να προσπαθεί να διαχειρίζεται σωστά το χρόνο.
- Σε περίπτωση που ο χρήστης δεν ήταν αφοσιωμένος στο τεστ και είχε την προσοχή του στραμμένη αλλού τότε εμφανίζεται ανάλογο μήνυμα.
- Εάν ο χρήστης έχει βαθμολογία κάτω από 50% και δεν έχει τελειώσει το τεστ στον επιθυμητό χρόνο εμφανίζεται το παρακάτω μήνυμα «Δεν έχεις απαντήσει σε όλες τις ερωτήσεις, διαχειρίσου καλύτερα το χρόνο σου. Διάβασε περισσότερο».
- Εάν ο χρήστης έχει βαθμολογία από 50% μέχρι 60% και παρ' όλα αυτά δεν έχει τελειώσει το τεστ στον επιθυμητό χρόνο εμφανίζεται το παρακάτω μήνυμα «Πέρασες το τεστ αλλά δεν απάντησες σε όλες τις ερωτήσεις! Διαχειρίσου καλύτερα το χρόνο σου».
- Εάν ο χρήστης έχει βαθμολογία από 60% μέχρι 85% και παρ' όλα αυτά δεν έχει τελειώσει το τεστ στον επιθυμητό χρόνο εμφανίζεται το παρακάτω μήνυμα «Τα πήγες περίφημα αλλά κάποιες ερωτήσεις δεν πρόλαβες να τις κάνεις!».
- Εάν ο χρήστης έχει βαθμολογία από 85% μέχρι 100% και παρ' όλα αυτά δεν έχει τελειώσει το τεστ στον επιθυμητό χρόνο εμφανίζεται το παρακάτω μήνυμα «Έχεις σπουδαία βαθμολογία αν πρόσεχες λίγο θα είχες ακόμα καλύτερα αποτελέσματα!».

4.11 Προσαρμοστική πλοήγηση σε σχέση με τον χρόνο, τη βαθμολογία και το διάβασμα.

Για να πετύχουμε την προσαρμοστικότητα στο σύστημά μας και δεδομένου ότι απευθύνεται σε παιδιά έχουμε χρησιμοποιήσει εικόνες οι οποίες υποδεικνύουν στο χρήστη αν πρέπει να προχωρήσει σε επόμενη ενότητα ή όχι. Για τα σκοπό αυτό δημιουργήσαμε ένα πίνακα που περιέχει τα στοιχεία χρόνος, βαθμολογία και διάβασμα. Σε κάθε μια από τις κατηγορίες αυτές γίνεται αξιολόγηση των επιδόσεών του. Σε κάθε κατηγορία μπορεί να πάρει άδαιο, μισό ή ολόκληρο αστέρι καθώς επίσης και απαγορευτικό, προειδοποιητική σήμανση και σήμανση για επανάληψη. Η σημασία του καθενός ξεχωριστά θα αναλυθεί παρακάτω. Στο τέλος της αξιολόγησης αυτής εμφανίζεται το αποτέλεσμα που ενημερώνει τον μαθητή αν πρέπει να συνεχίσει ή όχι.



Εικ.4.1 Εικονίδια που εμφανίζονται στην οθόνη του χρήστη ανάλογα με την απόδοση

4.11.1 Χρόνος

Όταν ολοκληρώνεται ένα τεστ ο χρήστης αξιολογείται και για την ταχύτητα με την οποία ολοκλήρωσε το τεστ. Όμως μαζί με τον χρόνο ολοκλήρωσης πρέπει να συσχετίσουμε και την βαθμολογία του χρήστη διότι αν για παράδειγμα ολοκληρώσει το τεστ μέσα σε λίγα

δευτερόλεπτα και η απόδοση του δεν είναι καλή σημαίνει ότι ο μαθητής μάλλον τυχαία απαντούσε τις ερωτήσεις και δεν έδινε την απαραίτητη προσοχή. Για το λόγο αυτό γίνονται οι έλεγχοι που φαίνονται αναλυτικά στον πίνακα 3.1 που ακολουθεί:

Έλεγχοι για την διάρκεια ολοκλήρωσης του τεστ				
Χρόνος διάρκειας τεστ	Βαθμολογία	Μήνυμα	Εικονίδιο	Μεταβλητή
χρόνος \geq 7 Λεπτά	Δεν υπολογίζεται	Θα μπορούσες να επιτύχεις καλύτερο χρόνο. Προσπάθησε ξανά για καλύτερο αποτέλεσμα		stop=stop+1
5= \leq Χρόνος \leq 7 λεπτά	Δεν υπολογίζεται	Ο χρόνος ολοκλήρωσης του τεστ είναι αρκετά ικανοποιητικός		star=star+2
1= \leq Χρόνος \leq 5λεπτά	Βαθμολογία \leq 50%	Προσπάθησε να ξανακάνεις το τεστ και κοίτα τις ερωτήσεις περισσότερη ώρα για να τις κατανοήσεις		star=star+1
	Βαθμολογία \geq 50%	Ο χρόνος ολοκλήρωσης του τεστ είναι πολύ καλός.		star=star+3
	Βαθμολογία \leq 50%	Ολοκλήρωσες το test πολύ γρήγορα και η απόδοσή σου δεν είναι καλή. Ίσως να μην πρόσεχες τόσο τι έγραφαν οι ερωτήσεις		star=star+0
Χρόνος \leq 1 λεπτό	50% \leq Βαθμολογία \leq 60%	Ολοκλήρωσες το test πολύ γρήγορα. Αν κάνεις το τεστ πιο προσεκτικά θα πετύχεις καλύτερη βαθμολογία		star=star + 1
	60% \leq Βαθμολογία \leq 80%	Ολοκλήρωσες το test πολύ γρήγορα. Να θυμάσαι ότι όταν απαντάς τυχαία τις ερωτήσεις η τύχη μπορεί να μην είναι πάντα με το μέρος σου		star=star+ 2
	80% \leq Βαθμολογία \leq 100 %	Ολοκλήρωσες το test πολύ γρήγορα. Να θυμάσαι όμως ότι αν ολοκληρώνεις τόσο γρήγορα τα τεστ μπορεί να κάνεις λάθη απροσεξίας!		star=star+ 3

Πίνακας 4.1: Επεξήγηση για το πώς και πότε εμφανίζονται τα εικονίδια σε σχέση με την διάρκεια ολοκλήρωσης του τεστ

4.11.2 Βαθμολογία

- Για βαθμολογία κάτω από 50% παίρνει ένα απαγορευτικό εικονίδιο και η μεταβλητή stor αυξάνεται κατά μία μονάδα. Ο χρήστης ενημερώνεται ότι πρέπει να προσπαθήσει ξανά.
- Για βαθμολογία από 50% μέχρι 60% ο χρήστης παίρνει ένα άδειο αστεράκι γιατί μπορεί να έχει περάσει το τεστ αλλά η βαθμολογία του δεν είναι αρκετά υψηλή. Η μεταβλητή star αυξάνεται κατά μία μονάδα.
- Για βαθμολογία από 60% μέχρι 85% παίρνει μισό αστέρι και η μεταβλητή star αυξάνεται κατά δύο μονάδες. Ο χρήστης ενημερώνεται ότι η βαθμολογία του είναι αρκετά υψηλή.
- Για βαθμολογία από 85% μέχρι και 100% παίρνει ένα ολόκληρο αστέρι και η μεταβλητή star αυξάνεται κατά τρεις μονάδες. Ο χρήστης έχει σημειώσει μεγάλη βαθμολογία.

4.11.3 Διάβασμα

Στην περίπτωση αυτή γίνεται έλεγχος για το πόσο και αν ο χρήστης έχει διαβάσει. Όπως αναφέραμε σε προηγούμενη ενότητα, όταν ο χρήστης διαβάζει μία θεματική ενότητα αποθηκεύεται ο χρόνος που διαβάζει από τον οποίο έχει αφαιρεθεί ο χρόνος που είχε την προσοχή του αλλού. Για να αξιολογήσουμε αν ο χρήστης έχει διαβάσει αρκετά την ενότητα ορίσαμε ένα όριο για κάθε κεφάλαιο το οποίο ο χρήστης πρέπει να ξεπερνά για να θεωρείται επαρκής ο χρόνος που αφιέρωσε στο διάβασμα. Τα όρια έχουν οριστεί με βάση την δυσκολία των κεφαλαίων αλλά και τον όγκο των πληροφοριών. Συγκεκριμένα έχουν οριστεί τα εξής:

- Για το κεφάλαιο 1 χρειάζονται 30 λεπτά
- Για το κεφάλαιο 2 χρειάζονται 40 λεπτά
- Για το κεφάλαιο 3 χρειάζονται 50 λεπτά
- Για το κεφάλαιο 1 χρειάζονται 60 λεπτά

Ωστόσο δεν αρκεί μόνο το μέτρο αυτό διότι ο χρήστης μπορεί να έχει προηγούμενη γνώση και να μην χρειάζεται να αφιερώσει όσο χρόνο απαιτείται σε κάποιο κεφάλαιο και γι αυτό το λόγο η απόδοση του να είναι πολύ καλή. Ή σε άλλη περίπτωση, ο χρήστης μπορεί να έχει διαβάσει παραπάνω από τον χρόνο που έχουμε ορίσει όμως να μην καταφέρνει να περάσει το τεστ του επιλεγμένου κεφαλαίου. Αυτό σημαίνει ότι εκτός από τον έλεγχο που κάνουμε για το πόση ώρα έχει καταγραφεί στη βάση δεδομένων πρέπει να συνδυάσουμε και την απόδοση που είχε στο συγκεκριμένο κεφάλαιο. Για παράδειγμα αν ένας μαθητής έχει επιλέξει το κεφάλαιο 1 για να κάνει ένα τεστ και έχει διαθέσει πάνω από 30 λεπτά στο διάβασμα του και η βαθμολογία που παίρνει είναι κάτω από 50% δείχνει ότι ο μαθητής ίσως να μην ήταν τόσο αφοσιωμένος κατά το διάβασμά του ή δεν έχει κατανοήσει το περιεχόμενο του υλικού. Σε αυτήν την περίπτωση το πρόγραμμα τον συμβουλεύει να ξαναδιαβάσει.

4.12 Μαντείο των Δελφών

Για να εισέλθει ο χρήστης στο μαντείο των Δελφών πρέπει να έχει στην κατοχή από 10 νομίσματα και πάνω. Μόλις λοιπόν κάνει κλικ στο κουμπί για να επισκεφθεί το μαντείο γίνεται ο έλεγχος για το αν επαρκούν τα νομίσματα που έχει. Αν επαρκούν εισέρχεται στο μαντείο διαφορετικά ενημερώνεται ότι πρέπει να συγκεντρώσει 10 νομίσματα. Ο αριθμός των νομισμάτων που έχει στην κατοχή του ο χρήστης φαίνεται στο πάνω αριστερό μέρος της οθόνης στο κεντρικό παράθυρο της εφαρμογής (mythlab.cs).

Το μαντείο των Δελφών παρέχει στον χρήστη τρεις συμβουλές. Μία σε σχέση με το χρόνο, μία σε σχέση με τη βαθμολογία του και μία που έχει να κάνει με τη συμπεριφορά του η οποία έχει καταγραφεί μέσω των τεστ αλλά και μέσω της μελέτης που έχει κάνει (κινήσεις ποντικιού, συχνές αλλαγές στην απάντηση, μη αφοσίωση την ώρα της μελέτης κτλ). Όταν εισέλθει στο μαντείο γίνονται κάποιες αναζητήσεις στην βάση δεδομένων έτσι ώστε να έρθουν οι κατάλληλες συμβουλές στον μαθητή οι οποίες έχουν ως στόχο να τον ενθαρρύνουν και να τον κάνουν να συνεχίσει με μεγαλύτερη διάθεση την μελέτη. Συγκεκριμένα υπολογίζεται ο μέσος όρος των βαθμολογιών όλων των τεστ που έχει πραγματοποιήσει έτσι ώστε να διαπιστωθεί σε τι επίπεδο κυμαίνεται.

- Για μέσο όρο βαθμολογίας κάτω από 50% εμφανίζεται το εξής μήνυμα: “Η ολοκλήρωση των επιπέδων δεν είναι πάντα εύκολη υπόθεση, \r\n όμως μπορείς να βελτιώσεις την βαθμολογία σου!”
- Για μέσο όρο βαθμολογίας πάνω από 50% μέχρι 60 % εμφανίζεται το εξής μήνυμα: Η συνεχής εξάσκηση και το διάβασμα \r\n βελτιώνει τις επιδόσεις.
- Για μέσο όρο βαθμολογίας πάνω από 60% μέχρι 70 % εμφανίζεται το εξής μήνυμα Μόνο με θυσίες και κόπους μπορούμε να αποκτήσουμε τη μάθηση
- Για μέσο όρο βαθμολογίας πάνω από 70 % εμφανίζεται το εξής μήνυμα Η μεθοδική μελέτη φέρνει αποτελέσματα!

Όσον αφορά την μελέτη υπολογίζεται ο μέσος όρος που αφιερώνει ο μαθητής σε κάθε κεφάλαιο και συνδυάζονται με την απόδοση του αναλόγως εμφανίζονται τα παρακάτω μηνύματα:

- Για χρόνο κάτω από μισή ώρα:
 - Αν η μέση βαθμολογία του είναι κάτω από 50% εμφανίζεται το μήνυμα : “Όποιος βιάζεται σκοντάφτει”
 - Αν η μέση βαθμολογία του είναι πάνω από 50% εμφανίζεται το μήνυμα: “Ο πιο σύντομος δρόμος είναι αυτός που γνωρίζεις”
- Για χρόνο πάνω από μισή ώρα και κάτω από 2 ώρες εμφανίζεται το μήνυμα: “Η σωστή διαχείριση του χρόνου δείχνει συγκέντρωση”
- Πάνω από 2 ώρες: “Με περισσότερο διάβασμα μπορείς να πετύχεις καλύτερους χρόνους.”

Υπολογίζεται επίσης και ο μέσος όρος των κινήσεων του ποντικιού, των backspaces και των radiochanges.

- Στην περίπτωση που τα radiochanges είναι πάνω από 150 ή ο αριθμός των backspaces είναι πάνω από 15 ή οι κινήσεις του ποντικιού πάνω από 500 τότε:
 - Αν ο μέσος όρος της βαθμολογίας είναι πάνω από 60% δείχνει ότι ο μαθητής δεν αισθάνεται σίγουρος για τις απαντήσεις του και λαμβάνει το ακόλουθο μήνυμα : "Πρέπει να πιστεύεις στον εαυτό σου."
 - Αν ο μέσος όρος της βαθμολογίας είναι κάτω από 60% εμφανίζεται το μήνυμα: "Το διάβασμα μπορεί να φέρει αυτοπεποίθηση και περισσότερη συγκέντρωση."
- Διαφορετικά αν ο μέσος όρος radiochanges, backspaces και κινήσεων του ποντικιού δεν είναι πάνω από το όριο που έχουμε ορίσει ακολουθούνται οι παρακάτω ενέργειες:
 - Αν ο μέσος όρος της βαθμολογίας είναι πάνω από 60% εμφανίζεται το ρητό : "Εάν ης φιλομαθής, έσει και πολυμαθής."
 - Αν ο μέσος όρος της βαθμολογίας είναι κάτω από 60% εμφανίζεται το μήνυμα: "Ζήλος χωρίς γνώση είναι φωτιά χωρίς φως."

```

if ((usersRC > 150) || (usersBS > 15) || (usersMM > 500))
{
    if (usersAvg > 60)
    {
        labelMoves.Text = "Πρέπει να πιστεύεις στον εαυτό σου";
    }
    else
    {
        labelMoves.Text = "Το διάβασμα μπορεί να φέρει
\r\n αυτοπεποίθηση και περισσότερη συγκέντρωση";
    }
}
else
{
    if (usersAvg > 60)
    {
        labelMoves.Text = "Εάν ης φιλομαθής, έσει και πολυμαθής.";
    }
    else
    {
        labelMoves.Text = "Ζήλος χωρίς γνώση είναι φωτιά χωρίς φως.";
    }
}

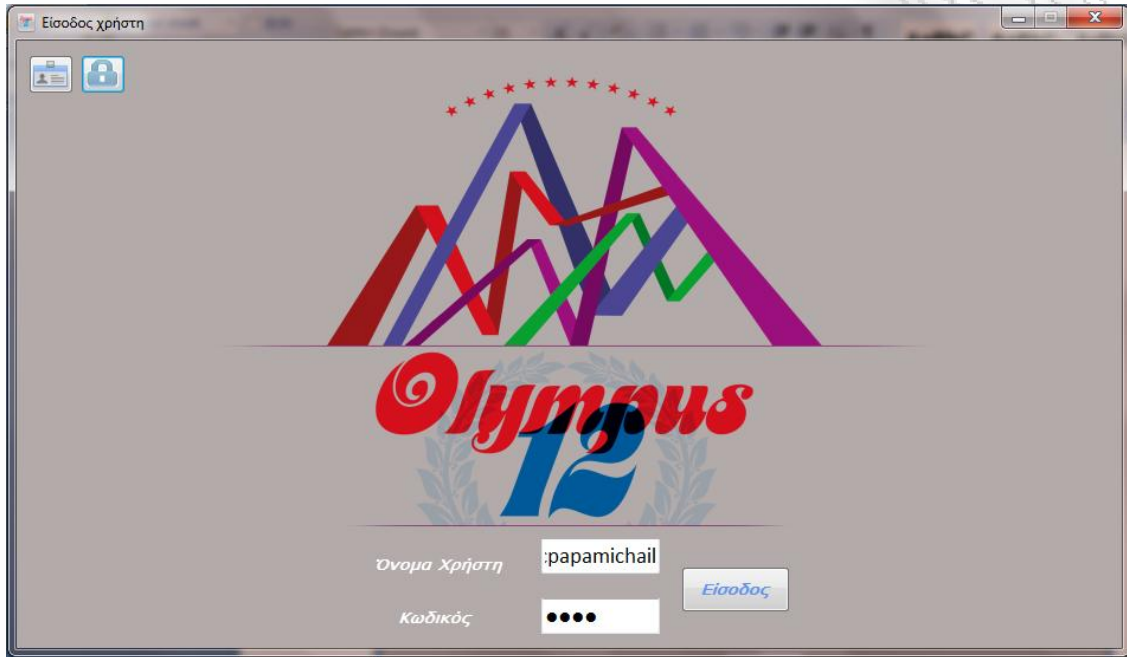
```

Κωδ. 4.25 Εμφάνιση συμβουλών

5 Εγχειρίδιο χρήστη

5.1 Είσοδος χρήστη

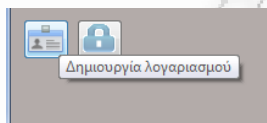
Ανοίγοντας την εφαρμογή εμφανίζεται η παρακάτω φόρμα:



Εικ. 5.1 Εισαγωγή στην εφαρμογή

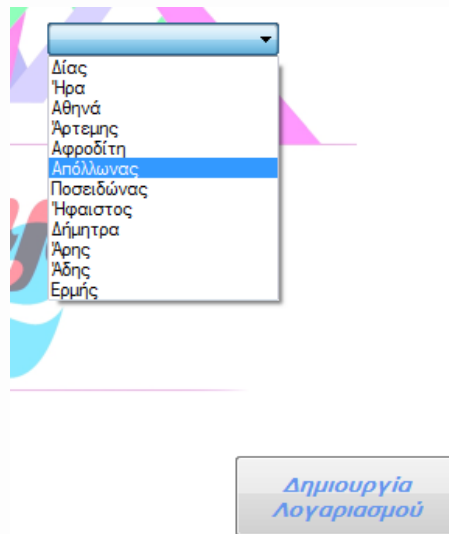
5.2 Δημιουργία λογαριασμού

Για τη δημιουργία νέου λογαριασμού επιλέγουμε το παρακάτω πεδίο



Όπου και εμφανίζεται μία νέα φόρμα για συμπλήρωση των στοιχείων του νέου χρήστη:

Ο νέος χρήστης πρέπει αυστηρά να συμπληρώσει όλα τα στοιχεία για να ολοκληρωθεί επιτυχώς η δημιουργία του νέου χρήστη. Επίσης θα πρέπει να επιλέξει και έναν από τους 12 θεούς για βοηθό, όπως απεικονίζεται παρακάτω.



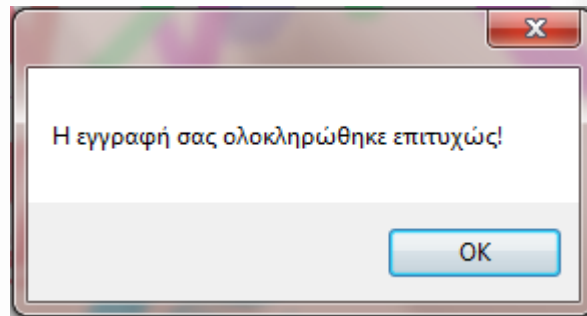
Εικ. 5.2 Επιλογή θεού

Κάνοντας κλικ στο κουμπί “Δημιουργία Λογαριασμού” εμφανίζεται και το αντίστοιχο μήνυμα για την ολοκλήρωση της επιτυχούς εγγραφής.

Εικ. 5.3 Δημιουργία λογαριασμού

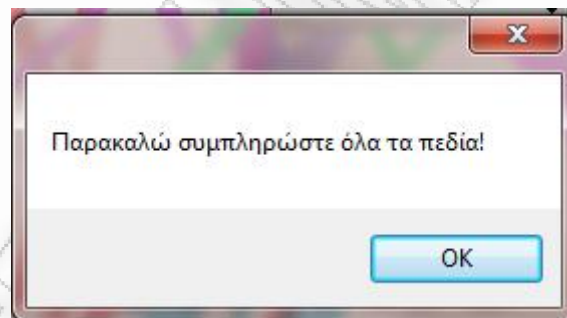
Δημιουργία
Λογαριασμού

Πατώντας το κουμπί σε περίπτωση που ο χρήστης έχει συμπληρώσει σωστά όλα τα πεδία εμφανίζεται το παρακάτω ενημερωτικό μήνυμα για την ολοκλήρωση της επιτυχούς δημιουργίας λογαριασμού.



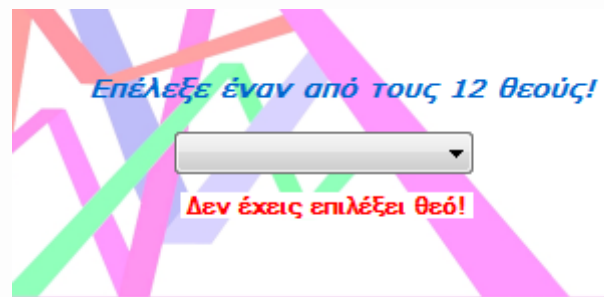
Εικ. 5.4 Μήνυμα ολοκλήρωσης επιτυχούς εγγραφής

Σε περίπτωση που ο χρήστης δεν έχει συμπληρώσει όλα τα πεδία εμφανίζεται το παρακάτω ενημερωτικό μήνυμα καθότι δεν επιτρέπεται η δημιουργία του λογαριασμού.



Εικ. 5.5 Μήνυμα ενημέρωσης συμπλήρωσης όλων των πεδίων

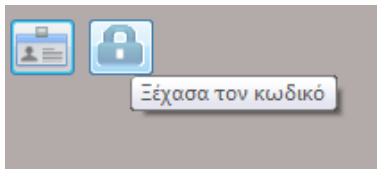
Εάν ο χρήστης δεν έχει επιλέξει κάποιο θεό επίσης δεν είναι εφικτή η δημιουργία του λογαριασμού .



Εικ. 5.6 Ενημέρωση για την επιλογή θεού

5.3 Αλλαγή κωδικού

Για την αλλαγή του κωδικού πρόσβασης επιλέγουμε το παρακάτω πεδίο



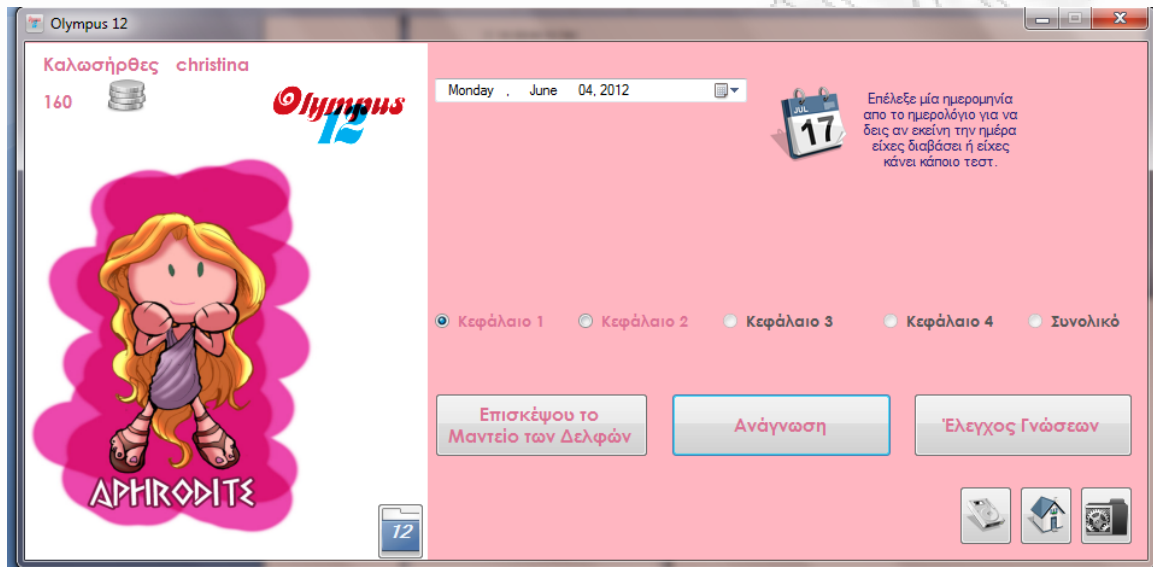
Όπου και εμφανίζεται μία νέα φόρμα για τη συμπλήρωση του νέου κωδικού. Κάνοντας κλικ στο κουμπί “Αλλαγή” εμφανίζεται το αντίστοιχο μήνυμα.

Εικ. 5.7 Αλλαγή κωδικού πρόσβασης

5.4 Είσοδος στην εφαρμογή

Εφόσον ο χρήστης εισαχθεί στην εφαρμογή εμφανίζεται μία νέα φόρμα που του παρέχει τη δυνατότητα να μελετήσει το διαθέσιμο υλικό, να επισκεφτεί το μαντείο των Δελφών για να Εκπαιδευτικό Λογισμικό εκμάθησης της ελληνικής μυθολογίας

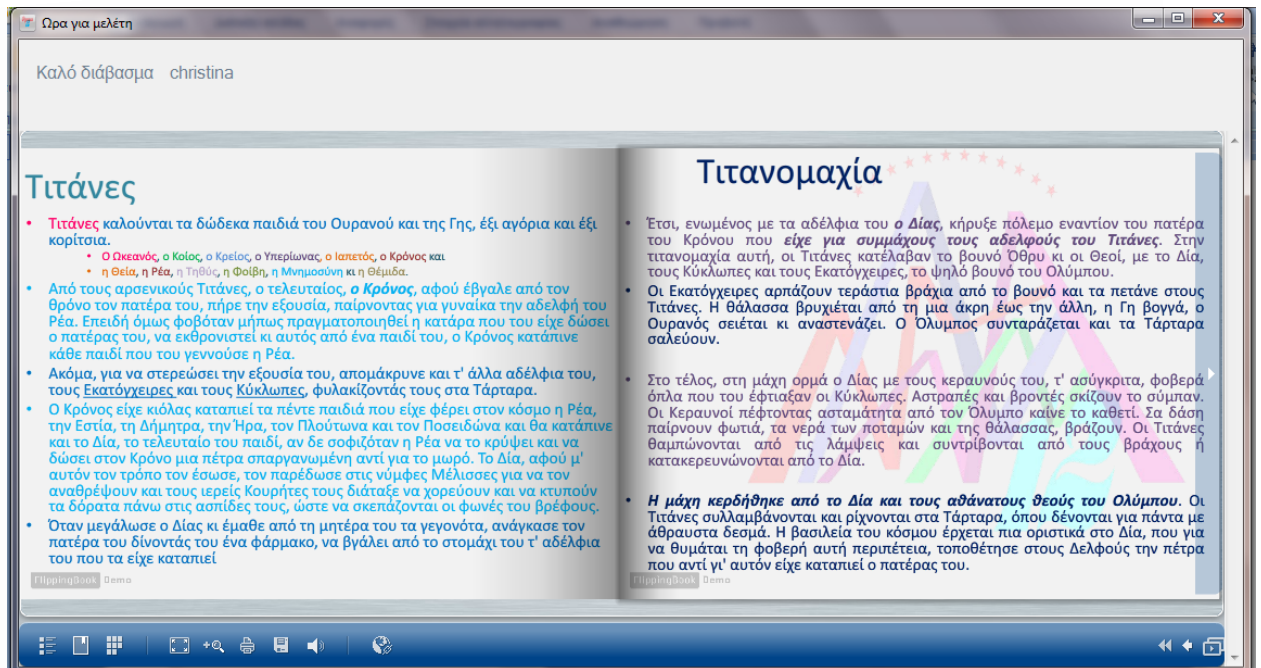
πάρει το χρησμό από την Πυθία, να κάνει νέο τεστ, να ελέγξει εάν είχε διαβάσει ή είχε κάνει τεστ κάποια συγκεκριμένη ημερομηνία, να αλλάξει βοηθό και να μάθει περισσότερες λεπτομέρειες για τους εχθρούς, να δει αναλυτικά τις βαθμολογίες του, να ελέγξει στοιχεία του λογαριασμού του, και να πάρει την κατάλληλη βοήθεια για την εφαρμογή. Το χρώμα της γραμματοσειράς της εφαρμογής είναι ροζ εάν ο χρήστης είναι κορίτσι και μπλε εάν είναι αγόρι. Επίσης το φόντο της παρακάτω φόρμας έχει χρώμα ανάλογο με το βοηθό που έχει επιλέξει ο χρήστης.



Εικ. 5.8 Δυνατότητες εφαρμογής

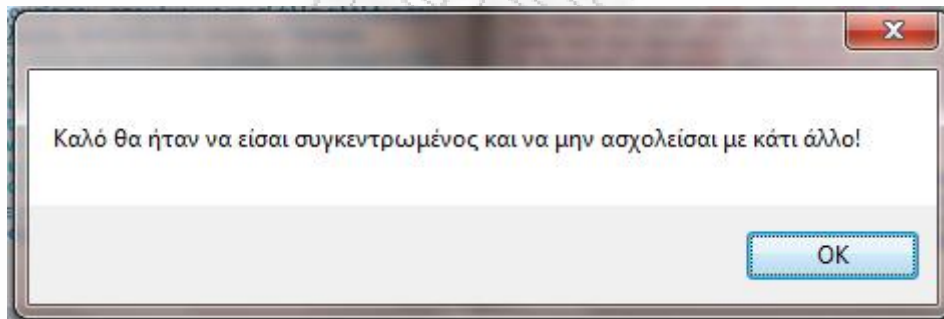
5.5 Ανάγνωση διαθέσιμου υλικού.

Επιλέγοντας το κουμπί “Ανάγνωση” εμφανίζεται ένα νέο παράθυρο με όλο το διαθέσιμο υλικό προς ανάγνωση ανάλογα με το κεφάλαιο που έχει επιλέξει ο χρήστης.



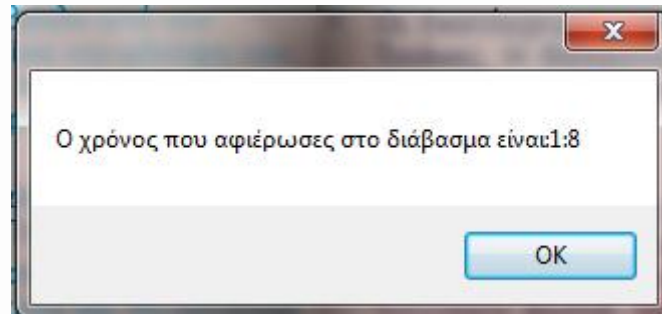
Εικ. 5.9 Ανάγνωση υλικού

Η εφαρμογή αποθηκεύει τον χρόνο που ο χρήστης αφιερώνει στο διάβασμα. Εάν δεν είναι αφοσιωμένος στο υλικό εμφανίζονται αντίστοιχα μηνύματα.



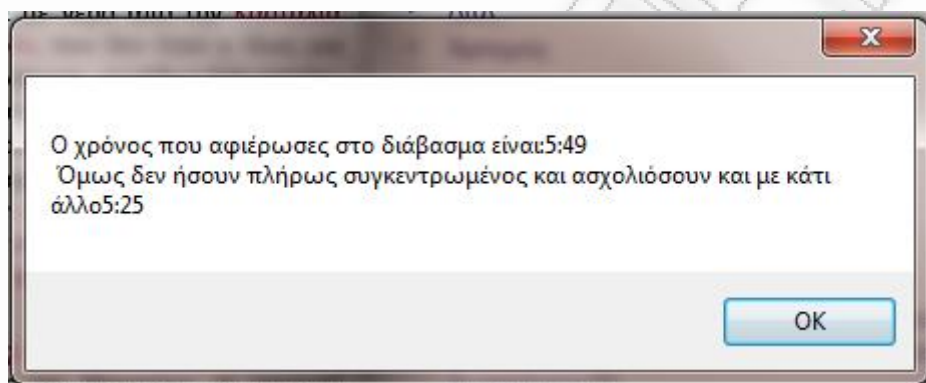
Εικ. 5.10 Μήνυμα ενημέρωσης του χρήστη για συγκέντρωση

Όταν ο χρήστης ολοκληρώσει το διάβασμα, ενημερώνεται για το χρονικό διάστημα που έχει αφιερώσει στο διάβασμα.



Εικ. 5.11 Ενημέρωση χρόνου ανάγνωσης υλικού

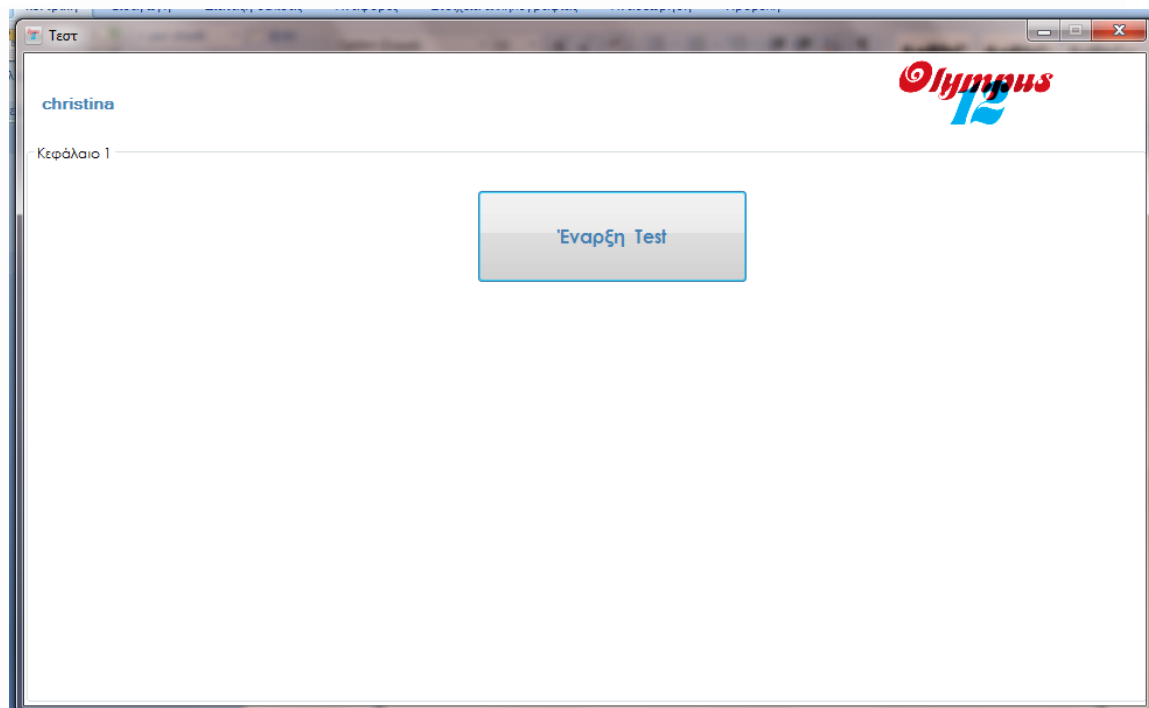
Στην περίπτωση που δεν είναι απόλυτα αφοσιωμένος εμφανίζεται και ο χρόνος που χαζεύει.



Εικ. 5.12 Ενημέρωση χρόνου διαβάσματος μη συγκεντρωμένου μαθητή.

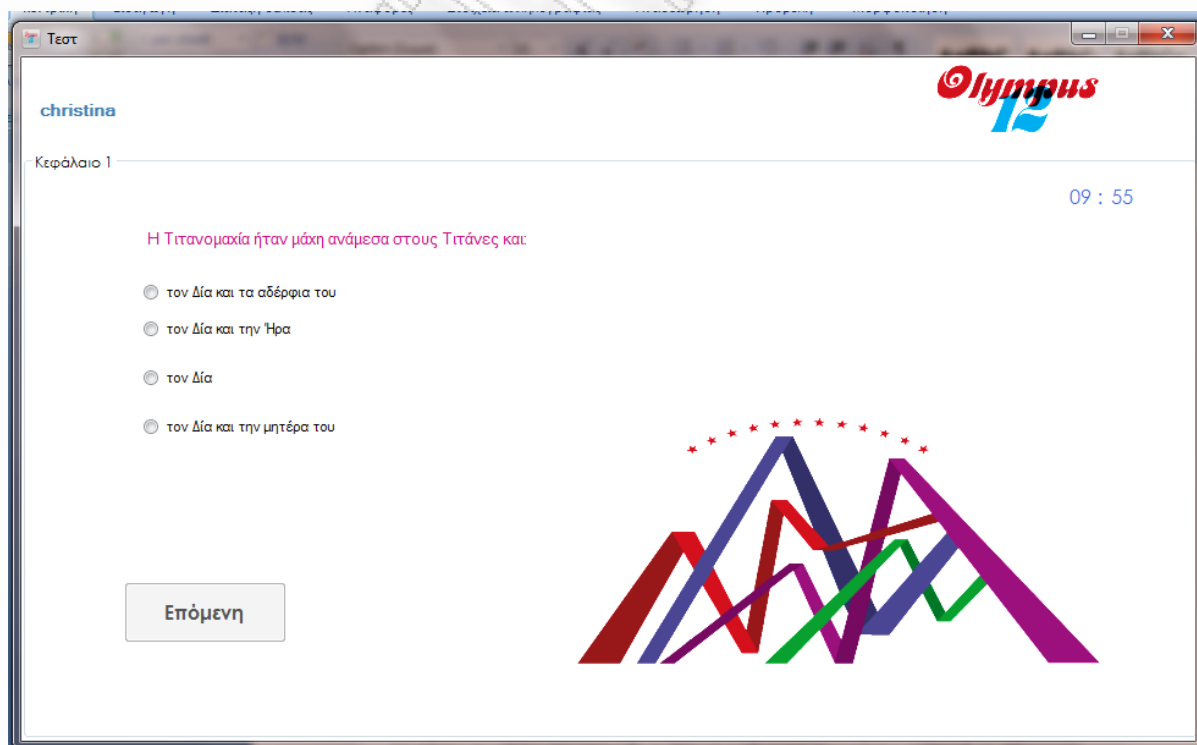
5.6 Νέο τεστ

Με την επιλογή "Έλεγχος Γνώσεων" ο χρήστης μπορεί να πραγματοποιήσει ένα νέο τεστ.



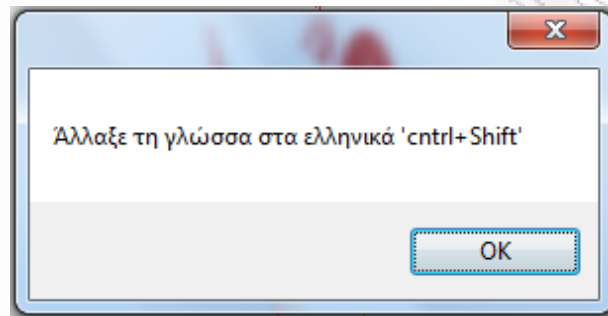
Εικ. 5.13 Έναρξη τεστ

Πατώντας το κουμπί “Έναρξη Test” αρχίζει ο χρόνος να μετράει αντίστροφα και εμφανίζονται οι ερωτήσεις του κεφαλαίου που έχει επιλεγεί.



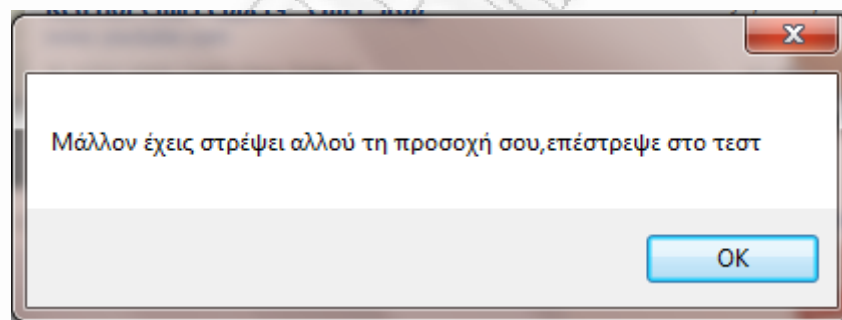
Εικ. 5.14 Εμφάνιση ερωτήσεων

Εφόσον ο χρήστης επιλέξει κάποια απάντηση τότε και μόνο μπορεί να πατήσει το κουμπί “Επόμενη” για να εμφανιστεί η επόμενη ερώτηση. Ο χρόνος για τους χρήστε ηλικίας άνω των 8 ετών είναι 10’ ενώ για τους υπόλοιπους χρήστες είναι 15’. Οι ερωτήσεις είναι διαφορετικές σε κάθε τεστ και τόσο οι ερωτήσεις όσο και οι απαντήσεις εμφανίζονται με τυχαία σειρά. Στις ερωτήσεις συμπλήρωσης κενού θα πρέπει να εισάγονται απαντήσεις αυστηρά με ελληνικούς χαρακτήρες ειδάλλως εμφανίζεται αντίστοιχο μήνυμα.



Εικ. 5.15 Μήνυμα ενημέρωσης επιλογής σωστής γλώσσας

Εάν ο χρήστης κατά τη διάρκεια του τεστ δεν είναι αφοσιωμένος η εφαρμογή τον ενημερώνει να είναι πιο συγκεντρωμένος για να ολοκληρώσει επιτυχώς το τεστ.



Εικ. 5.16 Μήνυμα ενημέρωσης προσήλωσης στο τεστ

5.7 Εμφάνιση αποτελεσμάτων

Όταν ο χρήστης ολοκληρώσει το τεστ εμφανίζεται μία νέα φόρμα με τα αποτελέσματα, τη βαθμολογία, τη διάρκεια του τεστ, το πλήθος των σωστών απαντήσεων, τα νομίσματα που έχει κερδίσει, τον συνολικό αριθμό νομισμάτων καθώς επίσης με συμβουλές ανάλογα με τον χρόνο, τη βαθμολογία και το διάβασμα. Επίσης η εφαρμογή παροτρύνει τον χρήστη ή όχι να συνεχίσει σε επόμενο κεφάλαιο ανάλογα με τη βαθμολογία. Παραπάνω έχουμε εξηγήσει τις παραμέτρους που λαμβάνει υπ’ όψιν η εφαρμογή για να δοθούν οι σωστές συμβουλές.

Εάν ο χρήστης έχει μία αρκετά καλή, αλλά όχι άριστη βαθμολογία και έχει τελειώσει το τεστ σε μικρό χρονικό διάστημα εμφανίζεται η παρακάτω φόρμα.

christina

Αποτελέσματα Συμβουλές

Συγχαρητήρια!!! Τα πήγες πολύ καλά!!
Μπορείς να κάνεις ξάνα το τεστ
ή να κάνεις το τεστ άλλου κεφαλαίου!!!

Η βαθμολογία σου είναι: 70 %

Διάρκεια τεστ: 0:34

Σωστές απαντήσεις 8/10

Έχεις κερδίσει 30 νομίσματα

Συνολικός αριθμός νομισμάτων: 50

ATHENAS
Πολύ καλή Βαθμολογία!!!
Συνέχισε έτσι!!


Χρόνος			Ολοκλήρωσες το test πολύ γρήγορα!
Βαθμολογία			Η βαθμολογία σου είναι αρκετά υψηλή!
Διάβασμα			Αν και δεν έχεις διαβάσει αρκετά η βαθμολογία σου είναι καλή! Μπορεί όμως να μην είσαι πάντα τόσο τυχερός!

Συνέχισε για νέο τεστ σε άλλο κεφάλαιο!

Εικ. 5.17 Εμφάνιση αποτελεσμάτων επιτυχούς διεξαγωγής τεστ

Επιλέγοντας το κουμπί συμβουλές εμφανίζεται ένα παράθυρο με τις συμβουλές προς το χρήστη.







Τα αποτελέσματά μου



christina 

Αποτελέσματα Συμβουλές

Η σωστή δουλειά φέρνει αποτελέσματα!!!!

Μην ξεχνάς όταν ξεκινάς το τεστ να είσαι αφοσιωμένος! Έτσι θα πετύχεις καλύτερο αποτέλεσμα σε λιγότερο χρόνο!!!!

Χρόνος			Ολοκλήρωσες το test πολύ γρήγορα!
Βαθμολογία			Η βαθμολογία σου είναι αρκετά υψηλή!
Διάβασμα			Αν και δεν έχεις διαβάσει αρκετά η βαθμολογία σου είναι καλή! Μπορεί όμως να μην είσαι πάντα τόσο τυχερός!

  Συνέχισε για νέο τεστ σε άλλο κεφάλαιο! 

Εικ. 5.18 Εμφάνιση συμβουλών

Εάν ο χρήστης δεν καταφέρει να περάσει το τεστ εμφανίζεται η παρακάτω φόρμα που ενημερώνει ότι δεν μπορεί να συνεχίσει λόγω της χαμηλής επίδοσης.

christina

Αποτελέσματα Συμβουλές

Δυστυχώς δεν κατάφερες να περάσεις στο τεστ.
Διάβασε ξανά το κεφάλαιο και προσπάθησε να πετύχεις ένα αποτέλεσμα καλύτερο από αυτό!!

Η βαθμολογία σου είναι: 15 %

Διάρκεια τεστ: 0:21

Σωστές απαντήσεις 1/10

Έχεις κερδίσει 10 νομίσματα

Συνολικός αριθμός νομισμάτων: 60

Χρόνος			Ολοκλήρωσες το test πολύ γρήγορα!
Βαθμολογία			Πρέπει να προσπαθήσεις ξανά.
Διάβασμα			Μπορείς να κάνεις κλικ στο κουμπί '12' στην αρχική σελίδα της εφαρμογής και να μάθεις την ιδιότητα κάθε θεού.

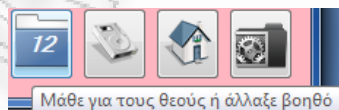
Θα ήταν καλύτερα να κάνεις επανάληψη και να προσπαθήσεις να κάνεις τεστ του ίδιου κεφαλαίου!

Εικ. 5.19 Εμφάνιση αποτελεσμάτων σε περίπτωση αποτυχίας

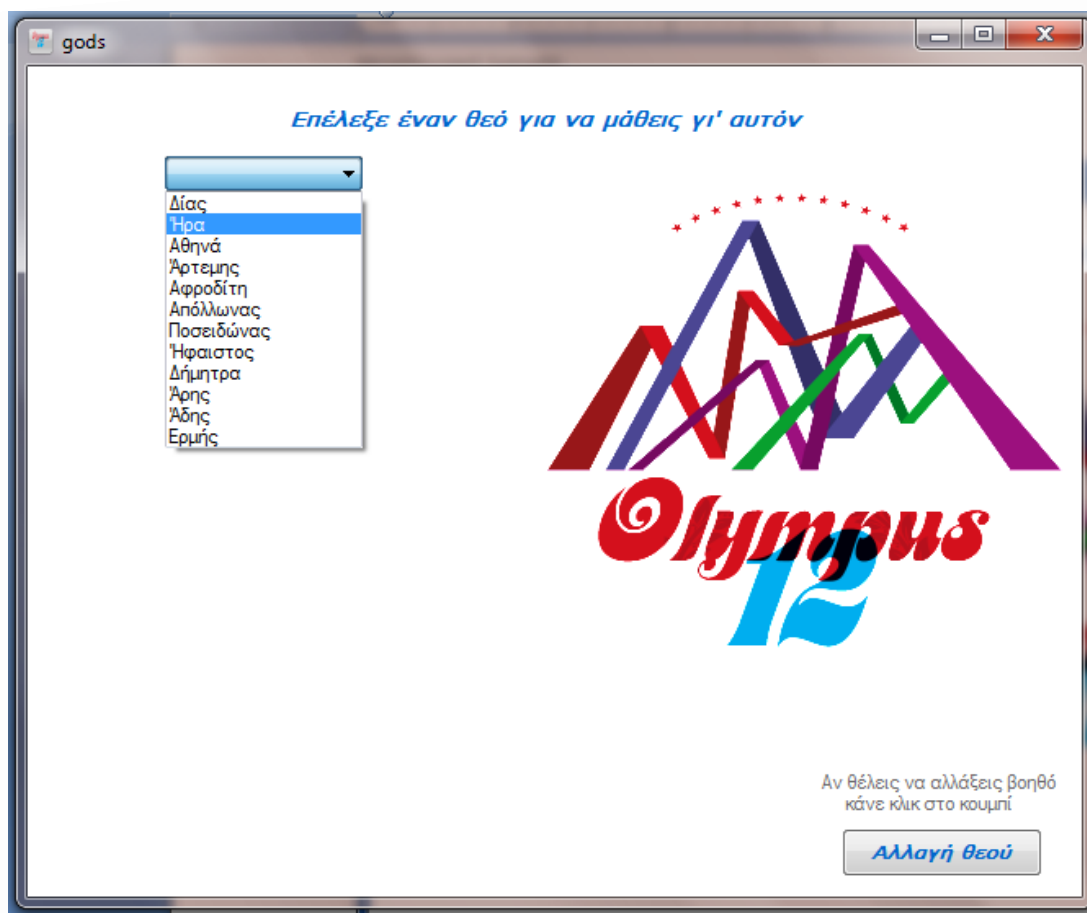
Για να επιστρέψει ο χρήστης στην κεντρική φόρμα επιλέγει το παρακάτω κουμπί.



5.8 Αλλαγή βοηθού.

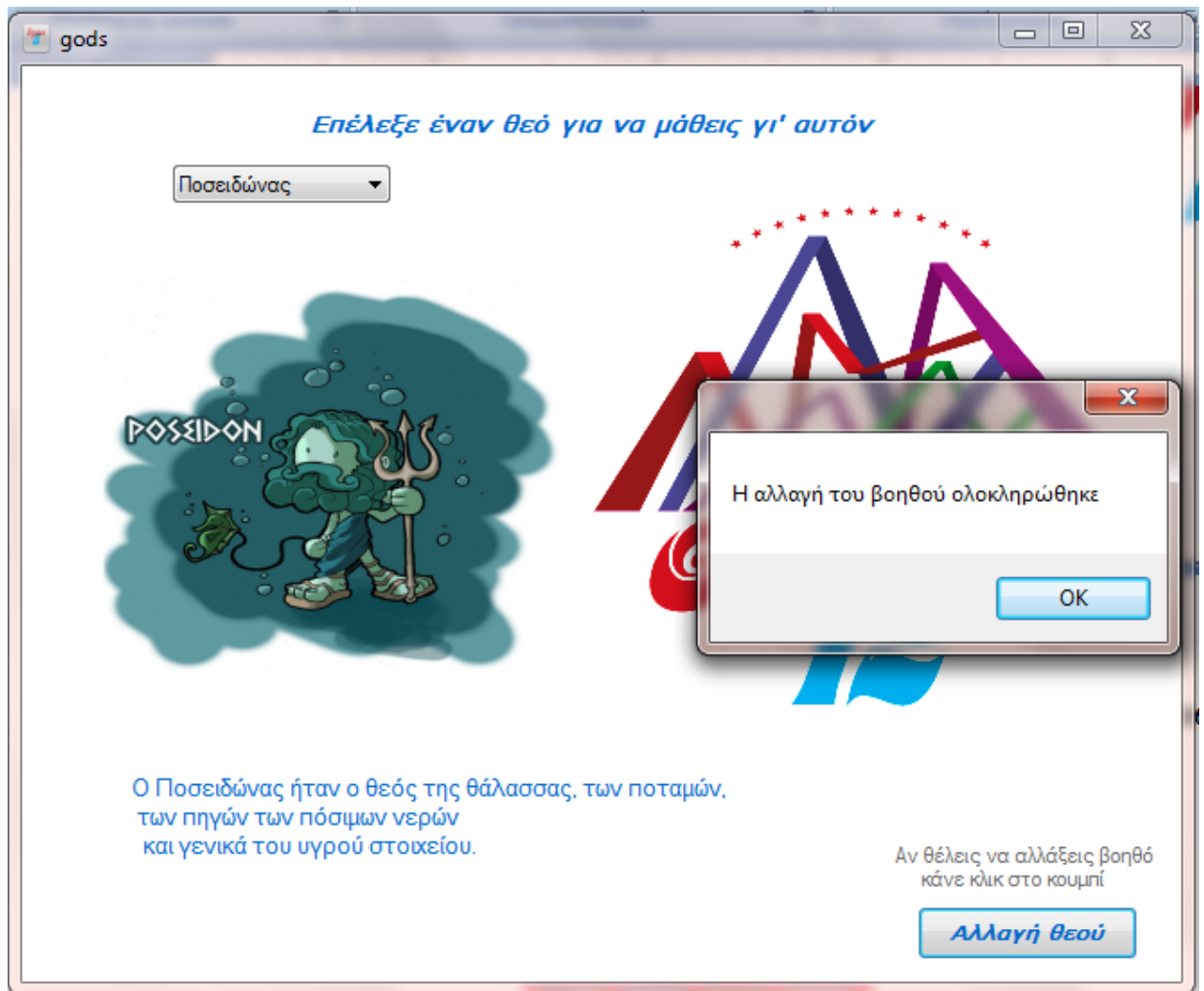


Επιλέγοντας το πεδίο εμφανίζεται ένα νέο παράθυρο όπου ο χρήστης μπορεί να πάρει χρήσιμες πληροφορίες για τους θεούς καθώς επίσης να αλλάξει και βοηθό πατώντας το κουμπί “Αλλαγή βοηθού”.



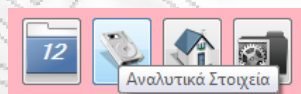
Εικ. 5.20 Αλλαγή βοηθού


Εφόσον ο χρήστης επιθυμεί να αλλάξει θεό, όταν επιστρέψει στην εφαρμογή θα εμφανιστεί και το αντίστοιχο χρώμα ανάλογα με το θεό που έχει επιλέξει.



Εικ. 5.21 Ενημέρωση επιτυχούς αλλαγής βοηθού

5.9 Αναλυτικά στοιχεία



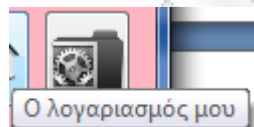
Επιλέγοντας το πεδίο  ο χρήστης έχει τη δυνατότητα να ελέγξει τα αποτελέσματα που έχει συγκεντρώσει σε κάθε τεστ καθώς επίσης την ακριβή ώρα και ημερομηνία που πραγματοποίησε το τεστ.

cu2
Κάνε κλικ στο κεφάλαιο που θές, για να δεις τα αποτελέσματα που έχεις συγκεντρώσει!

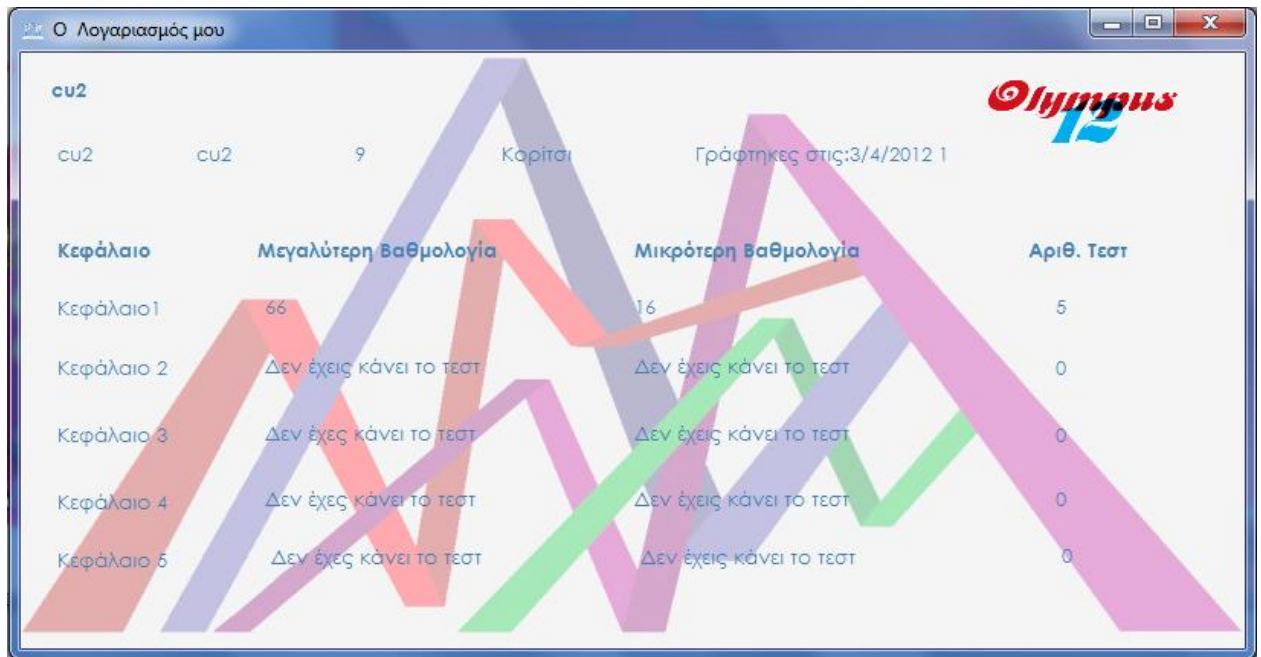
Κεφάλαιο	Βαθμός	Ημερομηνία	Διάρκεια
1	66	03/07/2012	0.15
1	33	03/07/2012	0.26
1	61	03/07/2012	0.20
1	33	03/08/2012	0.19
1	16	03/08/2012	8.5

Εικ. 5.22 Εμφάνιση αποτελεσμάτων ανά κεφάλαιο

5.10 Ο λογαριασμός μου

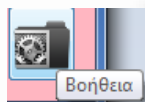


Επιλέγοντας το πεδίο **Ο λογαριασμός μου** ο χρήστης έχει τη δυνατότητα να ελέγξει πότε πραγματοποίησε την εγγραφή στην εφαρμογή, ποια είναι τα στοιχεία που έχει εισάγει (όνομα, επώνυμο, username, ηλικία, φύλο) καθώς επίσης το μεγαλύτερο και το μικρότερο βαθμό του τεστ όπως και τον αριθμό που έχει επιλέξει το τεστ.

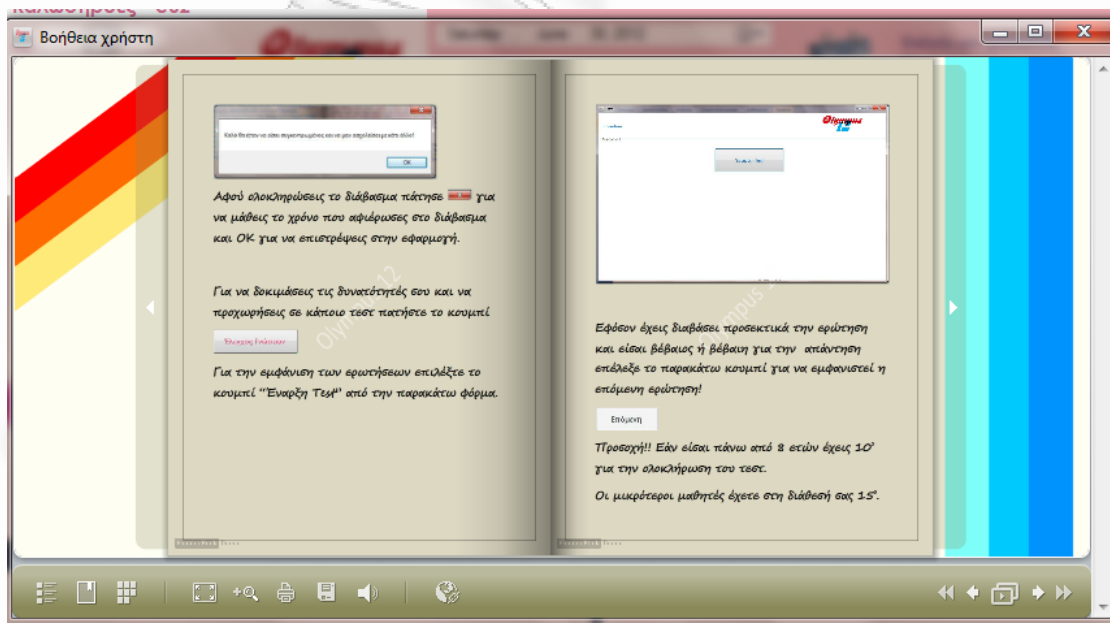


Εικ. 5.23 Εμφάνιση στοιχείων του χρήστη

5.11 Βοήθεια χρήστη



Επιλέγοντας το πεδίο **Βοήθεια** ο χρήστης έχει τη δυνατότητα να πάρει την κατάλληλη βοήθεια σχετικά με τη χρήση και τη λειτουργία της εφαρμογής.



Εικ. 5.24 Εμφάνιση βοήθειας χρήστη

5.12 Μαντείο Δελφών

Για να πάρει ο Χρήστης χρησμό από την Πυθία πρέπει να επιλέξει το παρακάτω:

Επισκέψου το
Μαντείο των Δελφών

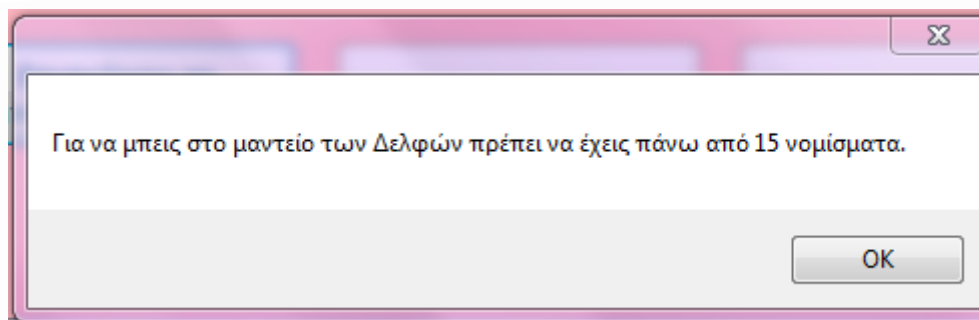
Απαραίτητη προϋπόθεση είναι να έχει ο χρήστης πάνω από 50 νομίσματα. Τότε εμφανίζεται το παρακάτω παράθυρο.



Εικ. 5.25 Εμφάνιση ενημερώσεων πυθίας

Τα μηνύματα που εμφανίζει η πυθία σχετίζονται με την απόδοση, τον χρόνο και με τη γενικότερη συμπεριφορά του χρήστη.

Σε περίπτωση που ο χρήστης δεν έχει συγκεντρώσει 50 νομίσματα εμφανίζεται το παρακάτω ενημερωτικό μήνυμα.



Εικ. 5.26 Ενημέρωση απαραίτητου αριθμού νομισμάτων για επίσκεψη στο μαντείο.

Συμπεράσματα

Το εκπαιδευτικό λογισμικό αποτελεί ένα ολοκληρωμένο σύστημα διδασκαλίας το οποίο διευρύνει την αντίληψη και τις γνώσεις του μαθητή. Ο ίδιος ξεκινά το διάβασμα με ευχάριστο και ξεκούραστο τρόπο μέσα από ένα γραφικό περιβάλλον προσαρμοστικό στις δικές του ανάγκες. Δοκιμάζει τις γνώσεις και τις δυνατότητές του μέσα από τη διεξαγωγή των διαφόρων τεστ.

Στην παρούσα διπλωματική εργασία δόθηκε ιδιαίτερο ενδιαφέρον στις ανάγκες του χρήστη έτσι ώστε να πετύχουμε ένα όσο το δυνατό πιο προσαρμοστικό λογισμικό. Ο χρήστης μέσα από διάφορα μηνύματα μαθαίνει να οργανώνει το χρόνο του τόσο στη μελέτη όσο και στη διεξαγωγή του τεστ. Επίσης δέχεται συμβουλές για την επίτευξη καλύτερων αποτελεσμάτων.

Ένα τέτοιου είδους λογισμικό μπορεί να εφοδιάσει το μαθητή με γνώση και να αντικαταστήσει με επιτυχημένο τρόπο τις παραδοσιακές αίθουσες διδασκαλίας. Μέσα από έξυπνες μεθόδους μπορεί να αναγνωρίζει την ψυχολογία, την αντίληψη και τις γνώσεις του ίδιου του μαθητή.

Βιβλιογραφία

1. Learning by teaching: A new agent paradigm for educational software **Gautam Biswas a , Krittaya Leelawong a , Daniel Schwartz b , NancyVye c & The Teachable Agents Group at Vanderbilt**
2. Intellectual and metacognitive skills of novices while studying texts under conditions of text difficulty and time constraint **Marcel V.J. Veenmana,b,* , Jos J. Beishuizena**
3. Stereotypes, Student Models and Scrutability **Judy Kay Basser Dept of Computer Science**
4. Adaptive Hypermedia: From Intelligent Tutoring Systems to Web-Based Education **Peter Brusilovsky**
5. Gautam Biswas and Krittaya Leelawong & Department of EECS and ISIS, Vanderbilt University, Nashville, TN, USA
6. Teaching & Teacher Educumion, Vol. 8, No. 2, pp. 109-121. 1992
7. An object-oriented software life cycle of an intelligent tutoring system. Journal of Computer Assisted Learning. 2001, 17. **M. Virvou, V. Tsiriga.**
8. Student and Instructor Models: Two Kinds of User Model and Their Interaction in an ITS Authoring Tool. **Maria Virvou, Maria Moundridou.**