

## Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Προηγμένα Συστήματα Πληροφορικής»

### Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	<b>Προσαρμοστικό σύστημα Διδασκαλίας της Ελληνικής Μυθολογίας</b>
Όνοματεπώνυμο Φοιτητή	<b>Ιλια Παναγοπούλου</b>
Πατρώνυμο	<b>Ανδρέας</b>
Αριθμός Μητρώου	<b>ΜΠΣΠ/ 10008</b>
Επιβλέπων	<b>Μαρία Βίρβου, Καθηγήτρια</b>

Ημερομηνία Παράδοσης **Ιούλιος 2012**

---



**Τριμελής Εξεταστική Επιτροπή**

(υπογραφή)

(υπογραφή)

(υπογραφή)

Μαρία Βίρβου  
Καθηγήτρια

Γεώργιος Τσιχριτζής  
Καθηγητής

Ευάγγελος Φούντας  
Καθηγητής

## Ευχαριστίες

Η παρούσα μεταπτυχιακή διατριβή ολοκληρώθηκε υπό την επίβλεψη της καθηγήτριας κ. Μαρίας Βίρβου την οποία θα ήθελα να ευχαριστήσω ιδιαιτέρως για τα επικοινωνιακά της σχόλια, τις παρατηρήσεις της καθώς και για τις κατευθύνσεις της κατά τη διάρκεια της προσπάθειάς μου.

Επιπλέον θα ήθελα να ευχαριστήσω την συμφοιτήτριά μου Χριστίνα Παπαμιχαήλ με την οποία συνεργαστήκαμε για την εκπόνηση της διατριβής αυτής.

Τέλος θα ήθελα να ευχαριστήσω την οικογένειά μου και τους φίλους μου για την απεριόριστη υπομονή τους και την συμπαράστασή τους.

## Περίληψη

Στην παρούσα μεταπτυχιακή διατριβή παρουσιάζεται η ανάπτυξη ενός προσαρμοστικού εκπαιδευτικού λογισμικού το οποίο παρέχει μια ολοκληρωμένη εκπαιδευτική διαδικασία στους μαθητές. Το λογισμικό αυτό δίνει την δυνατότητα στους μαθητές να μελετήσουν την ελληνική μυθολογία. Οι μαθητές μπορούν να κάνουν ανάγνωση της θεωρίας, μέσα από τεστ να δοκιμάσουν τις γνώσεις τους και να δουν τα αποτελέσματα αυτών. Επιπλέον το πρόγραμμα προσαρμόζεται σύμφωνα με την απόδοση του χρήστη αφού το επίπεδο δυσκολίας των τεστ καθορίζεται από αυτή. Τέλος σε κάθε τεστ ο χρήστης δέχεται συμβουλές και καθοδηγείται από το σύστημα το οποίο του προτείνει αν πρέπει να συνεχίσει σε επόμενο κεφάλαιο ή αν πρέπει να κάνει επανάληψη. Για να γίνει η υλοποίηση των παραπάνω έγινε μελέτη για το πώς πρέπει να σχεδιάζεται ένα εκπαιδευτικό λογισμικό, ποια στοιχεία το κάνουν προσαρμοστικό και τέλος πως θα γίνει πιο φιλικό και ελκυστικό προς το χρήστη.

## Abstract

This diploma dissertation presents the development of adaptive educational software which provides an integrated educational process for students. This software gives students the opportunity to learn about Greek mythology. Students are able to read about Greek mythology, perform a test and view their results on these tests. Furthermore, the software adjusted according to user performance since the difficulty level of each test is determined by user's performance. Finally, in each test the user receives advice from the system which suggests the user if he should take another test of a different chapter or he should repeat the test. Through a study became known how educational software is designed and which elements makes it adaptive, attractive and user friendly.

## Περιεχόμενα

Εισαγωγή.....	7
<b>1. Εκπαιδευτικό λογισμικό .....</b>	<b>8</b>
1.1 Τι ορίζουμε ως εκπαιδευτικό λογισμικό.....	8
1.2 Σχεδίαση εκπαιδευτικού λογισμικού.....	8
1.3 Πλεονεκτήματα – Μειονεκτήματα εκπαιδευτικού λογισμικού.....	10
<b>2. Προσαρμοστικά Συστήματα Διδασκαλίας .....</b>	<b>11</b>
2.1 Προσαρμοστικότητα εκπαιδευτικού λογισμικού .....	11
2.2 Μοντέλο Μαθητή.....	13
2.2.1 Το πείραμα του Singley.....	13
2.3 Προσαρμοστική Πλοήγηση.....	14
2.4 Προσαρμοστική Παρουσίαση.....	15
2.5 Τα χρώματα στη μάθηση και το λογισμικό.....	17
<b>3. Υλοποίηση της εργασίας.....</b>	<b>18</b>
3.1 Περιγραφή της εφαρμογής .....	18
3.2 Ανάλυση Απαιτήσεων .....	18
3.2.1 Δυνατότητες της εφαρμογής .....	18
3.2.2 Βάση Δεδομένων .....	19
3.3 Ασφάλεια δεδομένων .....	25
3.3.1 Η κλάση Sqlqueries.cs .....	25
3.3.2 Η κλάση MD5EncryptDecrypt.cs .....	25
3.4 Περιγραφή υλοποίησης των Windows Forms της εφαρμογής.....	26
3.4.1 Εισαγωγή του χρήστη στο σύστημα (loginuser.cs).....	26
3.4.2 Εγγραφή νέου χρήστη (createuser.cs).....	28
3.4.3 Δημιουργία νέου κωδικού (createnewpassword.cs).....	30
3.4.4 Εισαγωγή στην κεντρική φόρμα της εφαρμογής (mythlab.cs).....	31

3.4.5 Ανάγνωση του υλικού για μελέτη (reader.cs) .....	37
3.4.6 Επιλογή θεού (gods.cs) .....	38
3.4.7 Επισκόπηση λογαριασμού του χρήστη (myaccount.cs) .....	40
3.4.8 Στοιχεία σε σχέση με την απόδοση του χρήστη (Stats.cs) .....	40
3.4.9 Ανάλυση της σχεδίασης των τεστ (FormTests.cs) .....	41
3.4.1 Εμφάνιση αποτελεσμάτων στον χρήστη (Analytics.cs) .....	54
<b>3.5 Προσαρμοστική πλοήγηση σε σχέση με τον χρόνο, τη βαθμολογία και το διάβασμα. 56</b>	
3.5.1 Χρόνος .....	56
3.5.2 Βαθμολογία .....	58
3.5.3 Διάβασμα .....	58
<b>3.6 Τα χρώματα στην εφαρμογή “Olympus 12” .....</b>	<b>60</b>
<b>4. Παρουσίαση του εκπαιδευτικού λογισμικού “Olympus 12” .....</b>	<b>61</b>
4.1 Εγγραφή νέου χρήστη .....	61
4.2 Παρουσίαση της κεντρικής φόρμας .....	64
4.3 Παρουσίαση της φόρμας των θεών .....	66
4.4 Παρουσίαση της φόρμας με το υλικό μελέτης .....	67
4.5 Παρουσίαση των τεστ .....	70
4.6 Παρουσίαση αποτελεσμάτων .....	73
4.7 Παρουσίαση του Μαντείου των Δελφών .....	76
4.8 Αναλυτικά στοιχεία .....	77
4.9 Παρουσίαση της φόρμας “Ο λογαριασμός μου ” .....	78
4.10 Αλλαγή κωδικού πρόσβασης .....	79
4.11 Παρουσίαση της φόρμας “Βοήθεια” .....	81
<b>Συμπεράσματα .....</b>	<b>82</b>
<b>Βιβλιογραφία .....</b>	<b>83</b>

## Εισαγωγή

Η παρούσα διπλωματική εργασία πραγματεύεται τη δημιουργία ενός εκπαιδευτικού λογισμικού. Στόχος είναι να μελετηθεί πώς γίνεται η δημιουργία και ο σχεδιασμός ενός εκπαιδευτικού λογισμικού και στη συνέχεια η ανάλυση υλοποίησης ενός τέτοιου.

Στο πρώτο κεφάλαιο γίνεται αναφορά για το τι αποτελεί εκπαιδευτικό λογισμικό και τι όχι, καθώς και ανάλυση των πλεονεκτημάτων και μειονεκτημάτων τους. Επιπλέον αναλύεται και το πώς πρέπει να σχεδιαστεί ένα εκπαιδευτικό λογισμικό έτσι ώστε να είναι ελκυστικό και να επιτυγχάνει τον σκοπό του που είναι η εκμάθηση.

Στο δεύτερο κεφάλαιο αναλύονται τα προσαρμοστικά συστήματα διδασκαλίας. Δηλαδή τα συστήματα που προσαρμόζονται ανάλογα με τις ανάγκες του χρήστη. Στο κεφάλαιο αυτό γίνεται εκτενής αναφορά για την προσαρμοστική πλοήγηση η οποία αποτελεί σημαντικό στοιχείο για την δημιουργία ενός προσαρμοστικού εκπαιδευτικού λογισμικού. Τέλος αναφέρεται το πώς πρέπει να επιλεγθούν τα χρώματα που θα έχει η εφαρμογή έτσι ώστε να είναι ελκυστική και να δημιουργεί στο μαθητή την επιθυμία για μάθηση.

Στο κεφάλαιο τρία αναλύονται οι απαιτήσεις του εκπαιδευτικού λογισμικού που σχεδιάστηκε. Ορίζονται οι απαιτήσεις, οι περιορισμοί και οι στόχοι του λογισμικού. Γίνεται ανάλυση της υλοποίησης της εφαρμογής και η περιγραφή του σχεδιασμού της κάθε φόρμας.

Το τελευταίο κεφάλαιο αποτελεί την παρουσίαση της εφαρμογής και αναλύονται όλες οι λειτουργίες του εκπαιδευτικού λογισμικού που σχεδιάστηκε.

## 1. Εκπαιδευτικό λογισμικό

### 1.1 Τι ορίζουμε ως εκπαιδευτικό λογισμικό

Τα τελευταία χρόνια στο χώρο της εκπαίδευσης η τεχνολογία έχει σημαντικό ρόλο. Οι εκπαιδευτές προσπαθούν να κάνουν το μάθημά τους περισσότερο διαδραστικό έτσι ώστε οι μαθητές να συμμετέχουν και να κατανοούν καλύτερα το περιεχόμενο της εκπαιδευτικής ύλης. Για το λόγο αυτό η ανάγκη για εκπαιδευτικά λογισμικά ως υποστηρικτικά μέσα σε όλες τις βαθμίδες μάθησης και σε όλα τα γνωστικά πεδία αυξάνεται. Η ανάπτυξη λογισμικού σε αυτόν τον τομέα έχει εξελιχθεί αρκετά αποτελώντας έτσι σημαντικό εργαλείο τόσο για τους εκπαιδευτές όσο και για τους εκπαιδευόμενους.

Με την έννοια εκπαιδευτικό λογισμικό (educational software) εννοούμε τα προγράμματα τα οποία δημιουργούνται και χρησιμοποιούνται για την εκπαίδευση του χρήστη. Κατά τον σχεδιασμό ενός τέτοιου λογισμικού πρέπει να λάβουμε υπόψη την γνώση που θα μεταδοθεί στο χρήστη (μαθητή), τον τρόπο με τον οποίο ο υπολογιστής θα μεταδώσει στον χρήστη την γνώση αυτή και το πώς το λογισμικό μπορεί να προσαρμοστεί στις ανάγκες του χρήστη για να γίνει πιο λειτουργικό και να πετύχει το στόχο του που είναι η μάθηση.

### 1.2 Σχεδίαση εκπαιδευτικού λογισμικού

Η ανάπτυξη ενός εκπαιδευτικού λογισμικού αποτελεί σύνθετη διαδικασία. Πρωταρχικές έννοιες που πρέπει να λάβουμε για τον σχεδιασμό και την ανάπτυξη του είναι οι εξής<sup>1</sup>:

- Οι μαθητές εξοικειώνονται με ένα ζήτημα ή με ένα πρόβλημα
- Οι μαθητές ασχολούνται με κάποια δραστηριότητα (π.χ. την λύση ενός προβλήματος)
- Οι μαθητές εξασκούν συγκεκριμένες ικανότητες ή αρμοδιότητες ( μαθαίνουν να μαθαίνουν).
- Οι μαθητές επανεξετάζουν την υπάρχουσα γνώση που έχουν
- Οι μαθητές αποκτούν κάποιο στόχο ή αναπτύσσουν κάποια δεξιότητα
- Στους δασκάλους μπορεί να παρέχονται υποστηρικτικά μέσα για την επίβλεψη των ενεργειών που κάνουν οι μαθητές

Ο Robert M Gagné<sup>2</sup> διατύπωσε με σαφήνεια τη στρατηγική σχεδίασης ενός εκπαιδευτικού προγράμματος. Εισήγαγε στη θεωρία της διδασκαλίας το μοντέλο «Εννέα εκδηλώσεις διδασκαλίας» (Nine events of Instruction) στο οποίο αναλύει τις εξής αρχές:

---

<sup>1</sup> Computer Science and Educational Software Design - a Resource for Multidisciplinary Work in Technology Enhanced Learning, Pierre Tchounikine, Springer 2011.

<sup>2</sup> Robert M. Gagné: Αμερικάνος Εκπαιδευτικός Ψυχολόγος γνωστός για το έργο του "Conditions of learning". Πηγή: [http://en.wikipedia.org/wiki/Robert\\_M.\\_Gagn%C3%A9](http://en.wikipedia.org/wiki/Robert_M._Gagn%C3%A9)



1. Προσέλκυση προσοχής
2. Πληροφόρηση μαθητή για τους στόχους του μαθήματος
3. Διέγερση ανάκλησης της αποκτηθείσας μάθησης
4. Παρουσίαση περιεχομένου
5. Καθοδήγηση του μαθητή
6. Συγκέντρωση απόδοσης
7. Παροχή ανατροφοδότησης
8. Αξιολόγηση των επιδόσεων
9. Βελτίωση ανάπτυξης μνήμης και μεταφοράς γνώσης

Τα εννέα αυτά βήματα έχουν ως βάση νοητικές διαδικασίες που συμβαίνουν όταν παρουσιάζονται διάφορα ερεθίσματα. Αναλυτικότερα<sup>3</sup>:

1. Προσέλκυση προσοχής  
Αρχικά το εκπαιδευτικό λογισμικό πρέπει να κερδίσει την προσοχή του μαθητή. Αυτό μπορεί να γίνει χρησιμοποιώντας κάποιο animation ή κάποια μουσική έτσι ώστε το ερέθισμα αυτό να ενεργοποιήσει τους υποδοχείς.
2. Πληροφόρηση μαθητή για τους στόχους του μαθήματος  
Στην αρχή κάθε μαθήματος ο μαθητής πρέπει να βλέπει μια λίστα με τους στόχους του μαθήματος. Έτσι δημιουργείται ένα επίπεδο προσαρμογής για την μάθηση το οποίο παρακινεί τους μαθητές να ολοκληρώσουν το μάθημα.
3. Διέγερση ανάκλησης της αποκτηθείσας μάθησης  
Συσχέτιση μιας νέας πληροφορίας με προηγούμενη γνώση διευκολύνει την μαθησιακή διαδικασία. Συγκεκριμένα όταν οι μαθητές συνδέουν τη νέα πληροφορία με προσωπική τους εμπειρία ή γνώση είναι πιο εύκολο να απομνημονεύσει σε μακροπρόθεσμη μνήμη. Ένας τρόπος είναι να τεθούν ερωτήματα προηγούμενων εμπειριών.
4. Παρουσίαση περιεχομένου  
Στο σημείο αυτό το υλικό παρουσιάζεται στον εκπαιδευόμενο. Θα πρέπει να χρησιμοποιούνται αρκετοί τρόποι παρουσίασης όπως είναι το video, η ηχητική αφήγηση και το κείμενο.
5. Καθοδήγηση του μαθητή  
Για να κωδικοποιούν οι εκπαιδευόμενοι τις πληροφορίες για τη μακροχρόνια αποθήκευση, πρόσθετες κατευθυντήριες γραμμές θα πρέπει να παρέχονται μαζί με την παρουσίαση του νέου περιεχομένου όπως είναι τα παραδείγματα και οι γραφικές παραστάσεις.

---

<sup>3</sup> Gagne's Nine Events of Instruction, Author: Kevin Kruse  
Προσαρμοστικό σύστημα Διδασκαλίας της Ελληνικής Μυθολογίας

#### 6. Συγκέντρωση απόδοσης

Σε αυτό το στάδιο ο μαθητής καλείται να ανταποκριθεί σε ερωτήσεις για να επιβεβαιώσει τη σωστή κατανόηση του περιεχομένου. Η επανάληψη θα τον βοηθήσει στην κωδικοποίηση της γνώσης.

#### 7. Παροχή ανατροφοδότησης

Ο μαθητές εξασκούν τις νέες γνώσεις και ενισχύουν τη σωστή εκτέλεση. Σεμινάρια και ασκήσεις μπορούν να προστεθούν έτσι ώστε να κατανοήσουν το περιεχόμενο χωρίς βέβαια αυτά να μετρούν στην επίσημη βαθμολογία.

#### 8. Αξιολόγηση των επιδόσεων

Με την ολοκλήρωση της διδασκαλίας του περιεχομένου θα πρέπει να δοθεί στους μαθητές η ευκαιρία για την τελική αξιολόγηση. Η αξιολόγηση θα πρέπει να γίνει χωρίς επιπλέον εξάσκηση και υποδείξεις. Η πιστοποίηση της γνώσης χορηγείται μετά από την επίτευξη ορισμένων αποτελεσμάτων.

#### 9. Βελτίωση ανάπτυξης μνήμης και μεταφοράς γνώσης

Ο προσδιορισμός της κατανόησης ή όχι του περιεχομένου γίνεται όταν κληθεί να εφαρμόσει την αποκτηθείσα γνώση σε νέα κατάσταση. Η επανάληψη των εννοιών είναι μια δοκιμασμένη μέθοδος και βοηθά στην βελτίωση της απομνημόνευσης.

### **1.3 Πλεονεκτήματα – Μειονεκτήματα εκπαιδευτικού λογισμικού**

Τόσο οι μαθητές όσο και οι εκπαιδευτικοί μπορούν να κερδίσουν πολλά από την χρήση ενός εκπαιδευτικού λογισμικού όταν αυτό έχει σχεδιαστεί με τα κριτήρια που αναφέρθηκαν παραπάνω . Ο υπολογιστής δίνει την δυνατότητα στον χρήστη να αναπτύξει μεθοδικό και αναλυτικό τρόπο σκέψης. Η εκπαίδευση μπορεί να πραγματοποιηθεί από όποιο σημείο βρίσκεται ο μαθητής και σε οποιαδήποτε χρονική στιγμή. Η κατανόηση δύσκολων εννοιών μπορεί να γίνει με πολλαπλά μέσα παρουσίασης που είναι πιο ελκυστικά για τον μαθητή και τον κάνουν να θέλει να συμμετέχει στην μαθησιακή διαδικασία. Το πρόγραμμα μπορεί να προσαρμοστεί ανάλογα με τις ανάγκες του χρήστη έτσι ώστε να μπορέσει ο χρήστης να βελτιώσει την απόδοσή του.

Τα μειονεκτήματα ενός εκπαιδευτικού λογισμικού υπολείπονται των πλεονεκτημάτων του. Οι μαθητές που δεν είναι εξοικειωμένοι με την χρήση ηλεκτρονικών υπολογιστών θα δυσκολευτούν και στη χρήση ενός τέτοιου λογισμικού και ίσως να μην αποδώσουν το μέγιστο των ικανοτήτων τους. Επιπλέον ο τρόπος χρήσης τους κάποιες φορές είναι απρόσωπος με αποτέλεσμα να οδηγήσουν στην απομόνωση του χρήστη.

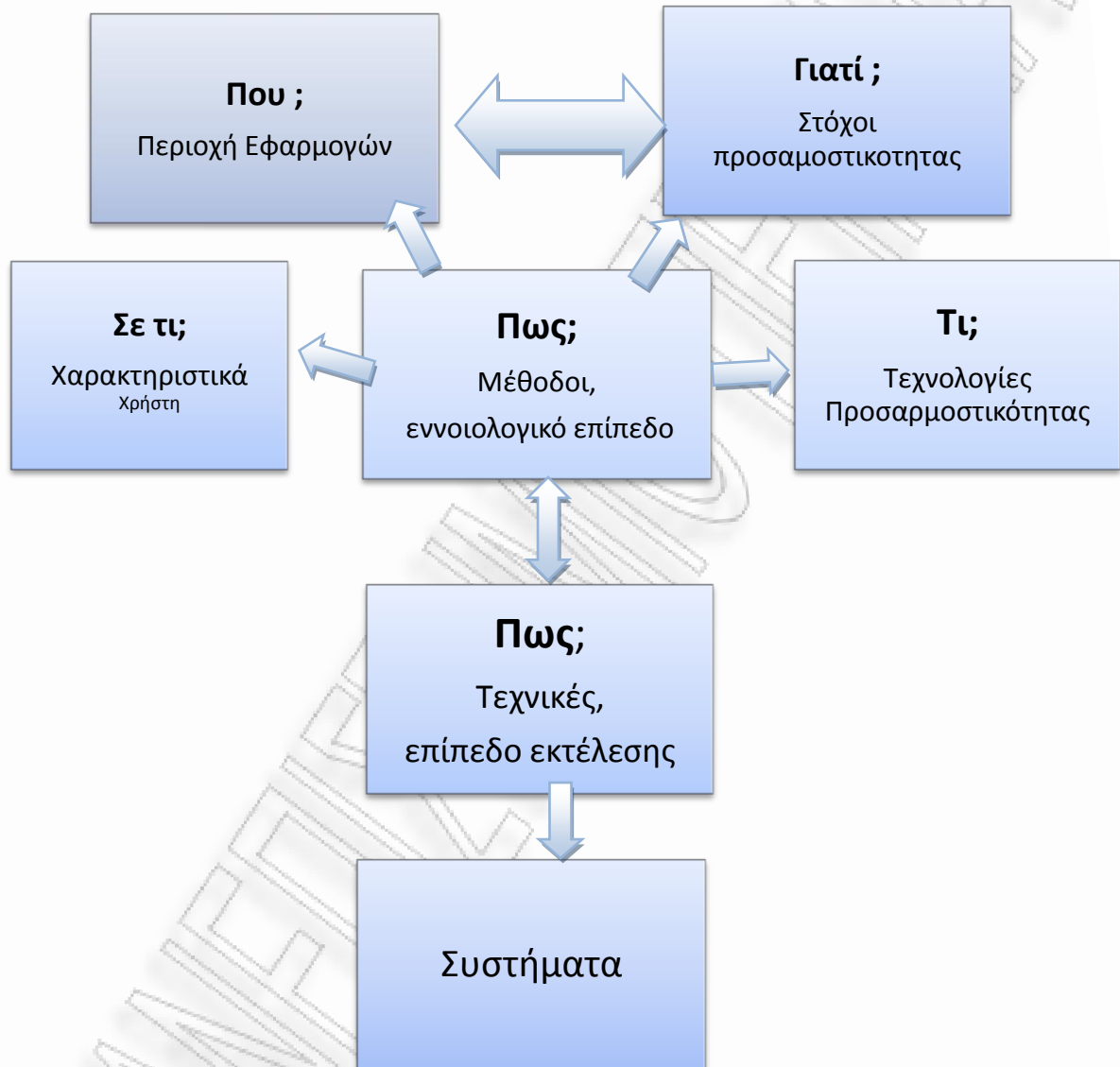
## **2. Προσαρμοστικά Συστήματα Διδασκαλίας**

### **2.1 Προσαρμοστικότητα εκπαιδευτικού λογισμικού**

Κάθε εκπαιδευτικό λογισμικό θα πρέπει να προσαρμόζεται στις ανάγκες του χρήστη, διότι κάθε μαθητής προσεγγίζει και κατανοεί διαφορετικά τις νέες έννοιες. Για παράδειγμα κάποιος μαθητής μπορεί να δυσκολεύονται με το περιεχόμενο διδασκαλίας ή να χρειάζονται περισσότερο χρόνο να ολοκληρώσουν κάποιο τεστ σε αντίθεση με άλλους.

Αν ένα εκπαιδευτικό λογισμικό είναι προσαρμοστικό (Adaptive Tutoring System) τότε γίνεται αυτομάτως και πιο ελκυστικό για τον μαθητή καθώς αυτός μπορεί να ελέγξει στοιχεία όπως το σχεδιασμό της οθόνης και διάφορα χαρακτηριστικά του συστήματος. Κατά το σχεδιασμό ενός τέτοιου συστήματος πρέπει να αναλογιστεί ο δημιουργός του πως μπορεί το πρόγραμμα να βοηθήσει τον μαθητή αποτελώντας προσαρμοστικό εκπαιδευτικό λογισμικό και ποια χαρακτηριστικά του μαθητή πρέπει να συλλεχθούν για να αποτελέσουν πηγή του προσαρμοστικού συστήματος.

Ένα προσαρμοστικό σύστημα διδασκαλίας επιτρέπει στον μαθητή να ενημερώνεται για τις αποφάσεις του συστήματος και να παρεμβαίνει σε αυτές αν επιθυμεί. Μεγαλύτερη αποτελεσματικότητα μπορεί να επιτευχθεί αν η ανάπτυξη του συστήματος βασιστεί στο θεωρητικό υπόβαθρο του χρήστη, την εμπειρία του, στο στόχο του χρήστη αλλά και τις προτιμήσεις του. Ο στόχος του χρήστη ανάλογα με το λογισμικό είναι διαφορετικός. Στόχος του μπορεί να είναι η επίλυση ενός προβλήματος ή η εκμάθηση νέων εννοιών. Οι στόχοι του μπορεί να είναι χαμηλού επιπέδου ή υψηλού επιπέδου και μπορεί να αλλάζουν μέσα σε ένα σύστημα κάθε φορά. Σε ένα εκπαιδευτικό λογισμικό ο εκπαιδευτικός στόχος μπορεί να είναι υψηλού επιπέδου ενώ το να λύσει ένα πρόβλημα να είναι χαμηλότερου επιπέδου στόχος. Το θεωρητικό υπόβαθρο του χρήστη αποτελείται από τις εμπειρίες του τις γνώσεις που μπορεί να έχει αποκτήσει στο συγκεκριμένο αντικείμενο διδασκαλίας. Οι προτιμήσεις του μπορούν να διαπιστωθούν από την καταγραφή των κινήσεων του, βλέποντας έτσι που κάνει συχνά κλικ και τι προτιμά να διαβάζει.



Σχήμα 2.1 Πιθανή ταξινόμηση για το σχεδιασμό ενός προσαρμοστικού συστήματος <sup>4</sup>

<sup>4</sup> Methods and Techniques of Adaptive Hypermedia, Peter Brusilovsky, 1996 Kluwer Academic Publishers, Netherlands

## 2.2 Μοντέλο Μαθητή

Το μοντέλου του μαθητή (student model) είναι σχεδιασμένο έτσι ώστε να αποθηκεύει πληροφορίες σχετικά με την συμπεριφορά του εκάστοτε μαθητή που χρησιμοποιεί το λογισμικό. Με τα στοιχεία που συλλέγονται διεξάγονται συμπεράσματα και μπορεί να ελεγχθεί κατά πόσο ένας μαθητής μαθαίνει αυτά που διδάσκεται. Οι πληροφορίες που συλλέγονται για τον μαθητή μπορούν να διαχωριστούν σε στατικές και δυναμικές.

Οι στατικές πληροφορίες είναι αυτές οι οποίες εξάγονται από τα προσωπικά χαρακτηριστικά του μαθητή όπως η ηλικία τα ενδιαφέροντα το φύλο ή υπάρχουσα γνώση που μπορεί να έχει σε ένα θέμα. Τα στοιχεία αυτά ζητούνται από τον μαθητή στην αρχή της εκπαιδευτικής διαδικασίας μέσα από διάφορα ερωτηματολόγια.

Οι δυναμικές πληροφορίες είναι αυτές που συλλέγονται από την αλληλεπίδραση του χρήστη με το σύστημα. Οι πληροφορίες αυτές αναφέρονται στο γνωστικό του επίπεδο, στις δεξιότητές του, στους στόχους τους οποίους θέτει αλλά και τα κίνητρα που μπορεί να έχει. Άλλες πληροφορίες μπορεί να είναι η καταγραφή των προτιμήσεων του, όπως ποιες θεματικές ενότητες του αρέσει να διαβάζει περισσότερο ή αν πριν από κάθε εξέταση κάνει επανάληψη στην διδακτέα ύλη. Επιπλέον μπορεί να καταγραφεί και πόσο γρήγορα ο μαθητής μαθαίνει κάτι αλλά και πόσο καλά θυμάται το υλικό με την πάροδο του χρόνου. Όλες αυτές οι πληροφορίες συλλέγονται μέσα από τα τεστ που κάνει, από τα προβλήματα που μπορεί να παρουσιαστούν (παρανοήσεις χρήστη), τις ενέργειες του (σε ποια κουμπιά κάνει κλικ, ποιες θεματικές ενότητες επιλέγει κτλ). Κάθε φορά που ο χρήστης εισέρχεται στο σύστημα καταγράφονται οι πληροφορίες αυτές και ανανεώνονται την επόμενη φορά που θα ξαναμπει.

### 2.2.1 Το πείραμα του Singley

Ο Singley<sup>5</sup> το 1987 σχεδίασε ένα σύστημα το οποίο θα βοηθούσε τους μαθητές στην επίλυση προβλημάτων της άλγεβρας. Δημιούργησε ένα μενού με τελεστές και ένα παράθυρο της επίλυσης του προβλήματος. Με τη χρήση του συστήματος παρατήρησε ότι οι μαθητές δεν έλυναν σωστά τα προβλήματα. Έτσι μετέτρεψε το σύστημα και δημιούργησε ένα ακόμη παράθυρο στο οποίο ο μαθητής καθόριζε το στόχο προτού επιλέξει τους τελεστές. Έτσι ο μαθητής βλέποντας τους στόχους θα μπορούσε να εφαρμόσει τους κανόνες που γνωρίζει σαν αλυσίδα και να καταφέρει τελικά να επιλύσει το πρόβλημα.

Μέσα από το πείραμα αυτό ο Singley διατύπωσε τις εξής παρατηρήσεις:

- Ο ορισμός του στόχου βελτιώνει την επίδοση στη λύση ενός προβλήματος.

---

<sup>5</sup> Singley, M.K. (1987). The effect of goal posting on operator selection, Proceedings of the Third International Conference on Artificial Intelligence and Education, Pittsburgh.

- Η επίδοση βελτιώθηκε ακόμα και όταν το παράθυρο του στόχου είχε αφαιρεθεί από την εφαρμογή καθώς οι μαθητές είχαν μάθει τα βήματα που πρέπει να ακολουθήσουν για να λύσουν το πρόβλημα.
- Οι δεξιότητες που απέκτησαν χρησιμοποιήθηκαν και σε άλλου είδους προβλήματα καθώς οι ικανότητες αυτές δεν χρησιμοποιούνται σε συγκεκριμένα προβλήματα αλλά μπορούν να φανούν χρήσιμες σε γενικά προβλήματα που προκύπτουν.

## 2.3 Προσαρμοστική Πλοήγηση

Κάθε χρήστης όταν εισέρχεται σε ένα λογισμικό κάνει κλικ στις πληροφορίες που τον ενδιαφέρουν μέσα από διαδρομές που έχουν καθοριστεί από τον σχεδιασμό του λογισμικού αλλά και από αυτές που μπορεί να ορίσει ο χρήστης.

Στην προσαρμοστική πλοήγηση (adaptive navigation) παρουσιάζονται συγκεκριμένα links ή κουμπιά σύμφωνα με τις γνώσεις του, τις προτιμήσεις του, την απόδοσή του και άλλα χαρακτηριστικά του χρήστη. Οι πιο γνωστές τεχνικές προσαρμοστική πλοήγησης είναι η άμεση καθοδήγηση (direct guidance), η προσαρμοστική ταξινόμηση (adaptive ordering), η απόκρυψη (hiding), ο προσαρμοστικός σχολιασμός (adaptive annotation) και ο προσαρμοστικός χάρτης (map adaptation).

Στην άμεση καθοδήγηση (direct guidance) το σύστημα προτείνει στον μαθητή ποιο είναι το επόμενο βήμα το οποίο θα τον οδηγήσει στην επίτευξη των στόχων. Ένα πρόβλημα της άμεσης καθοδήγησης είναι ότι δεν παρέχει καμία βοήθεια στους χρήστες που δεν θέλουν να ακολουθήσουν τις υποδείξεις του συστήματος. Αν και είναι χρήσιμη θα πρέπει να συνδυαστεί με μία πιο «υποστηρικτική» τεχνολογία.

Στην προσαρμοστική ταξινόμηση (adaptive ordering) γίνεται διάταξη όλων των συνδέσμων κάθε σελίδας σύμφωνα με το μοντέλο μαθητή. Ο πιο σχετικός σύνδεσμος μπαίνει πρώτος. Η προσαρμοστική ταξινόμηση μπορεί να εφαρμοστεί σε μη βασισμένες στα συμφραζόμενα συνδέσεις δύσκολα όμως εφαρμόζεται στο περιεχόμενο σελίδας που έχει συνήθως σταθερή μορφή συνδέσεων. Επιπλέον δεν μπορεί να χρησιμοποιηθεί σε βασισμένες στα συμφραζόμενα συνδέσεις και σε χάρτες.

Η τεχνολογία του προσαρμοστικού χάρτη (map adaptation) περιλαμβάνει διάφορες μεθόδους για την προσαρμογή των χαρτών και των τοπικών υπερμέσων που παρουσιάζονται στο χρήστη. Στους προσαρμοστικούς χάρτες μπορεί να εφαρμοστεί η απόκρυψη η άμεση καθοδήγηση ο σχολιασμός χωρίς όμως να αλλάξουν τη δομή του χάρτη.

Στην απόκρυψη (hiding) οι μη προτεινόμενοι σύνδεσμοι είναι μεν εμφανείς αλλά απενεργοποιημένοι και κατά συνέπεια μη επισκέψιμοι. Ένας σύνδεσμος μπορεί να είναι απενεργοποιημένος ή να μην εμφανίζεται διότι μπορεί να μην είναι σχετικός με τον στόχο του χρήστη ή να μην είναι έτοιμος να κατανοήσει ακόμα νέο υλικό.

Στον προσαρμοστικό σχολιασμό (adaptive annotation) το σύστημα αυξάνει τους υπάρχοντες συνδέσμους περιεχομένων με πληροφορίες που είναι σχετικές με την κατάσταση του μαθητή,

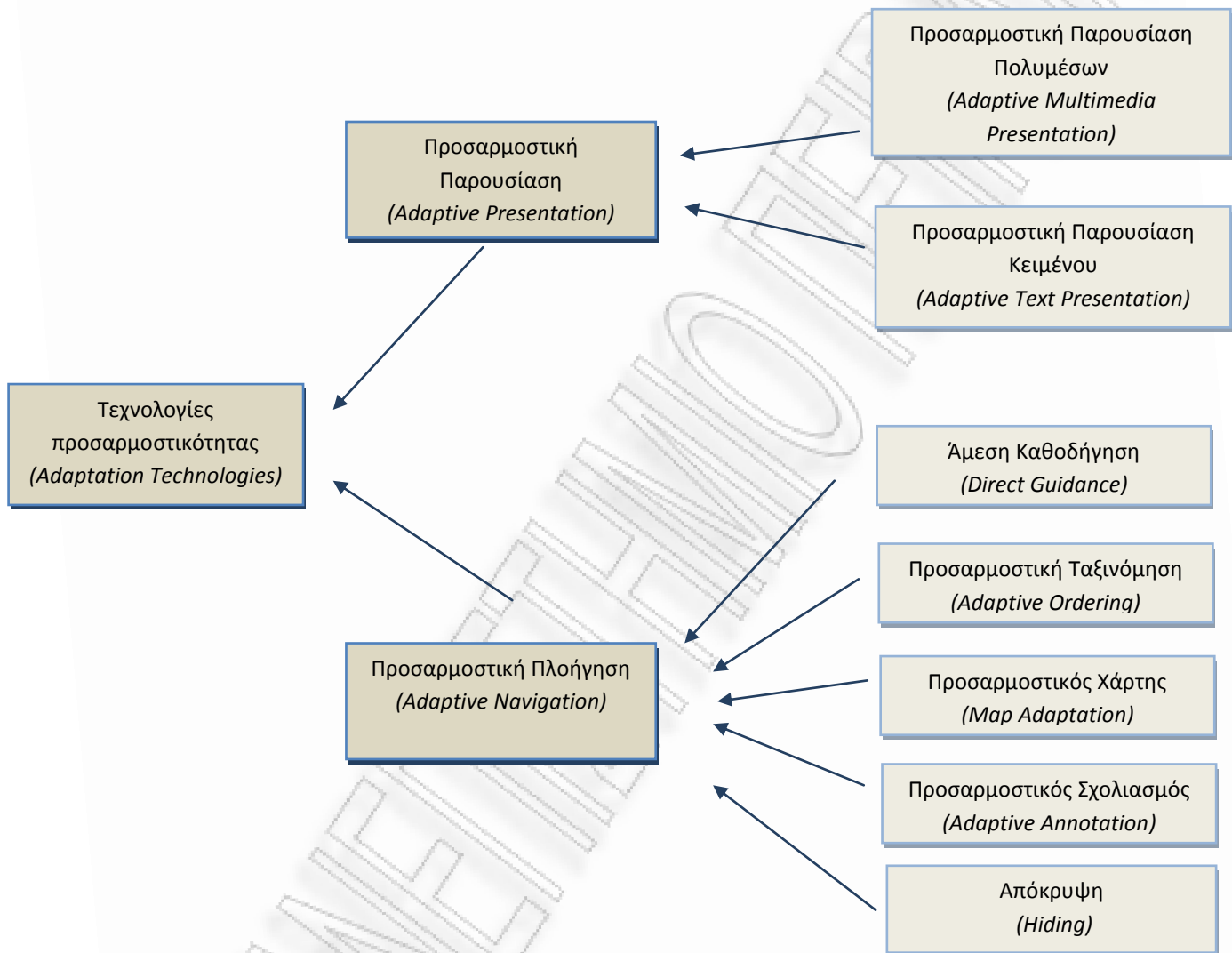
όπως τις έννοιες που έχει μάθει, αν μπορεί να επισκεφθεί νέους συνδέσμους κ.α. Τα σχόλια αυτά μπορεί να προβληθούν με την μορφή κειμένου ή με τη μορφή οπτικών μέσων όπως εικόνες και διάφορα χρώματα. Η πιο απλή εφαρμογή του προσαρμοστικού σχολιασμού είναι αυτή στους web browsers όπου ο χρήστης μπορεί να δει τα link που έχει επισκεφθεί. Τα προσαρμοστικά συστήματα διδασκαλίας μπορούν να διακριθούν και να σχολιασθούν σε έξι καταστάσεις βάση του προτύπου χρηστών.

Αυτές οι τεχνολογίες που περιγράφηκαν παραπάνω συνθέτουν την λειτουργία της προσαρμοστικής πλοήγησης. Μπορούν να χρησιμοποιηθούν σε συνδυασμό ή η καθεμία μόνη της. Συγκεκριμένα η άμεση καθοδήγηση μπορεί να συνδυαστεί με όλες τις λειτουργίες.

## 2.4 Προσαρμοστική Παρουσίαση

Η προσαρμοστική παρουσίαση (adaptive presentation) έχει ως στόχο τη διαμόρφωση του περιεχομένου μιας σελίδας βάση του γνωστικού επιπέδου του μαθητή, των στόχων του, των χαρακτηριστικών του και των πληροφοριών που έχει συλλέξει το σύστημα. Για παράδειγμα ένας χρήστης που έχει καλύτερη απόδοση από έναν άλλο δέχεται πληροφορίες με περισσότερες λεπτομέρειες και βαθύτερη ανάλυση. Δύο τεχνικές προσαρμοστικής παρουσίασης είναι οι επιπρόσθετες εξηγήσεις (comparative explanations) και η τεχνική της παραλλαγής των εξηγήσεων (explanation variants).

- Στην τεχνική των επιπρόσθετων εξηγήσεων (comparative explanations) γίνεται σύνδεση της υπάρχουσας γνώσης του μαθητή με το νέο περιεχόμενο. Στην τεχνική αυτή γίνεται απόκρυψη κάποιων πληροφοριών σε ορισμένες κατηγορίες χρηστών που δεν είναι έτοιμοι να επεξεργαστούν τις πληροφορίες αυτές.
- Στην τεχνική της παραλλαγής των εξηγήσεων (explanation variants) χρησιμοποιούνται παραλλαγές της ίδιας σελίδας με διαφορετικές παρουσιάσεις για τις ίδιες έννοιες. Οι έννοιες που παρουσιάζονται στο χρήστη είναι αυτές που αντιστοιχούν στο μοντέλο του σύμφωνα με τις πληροφορίες που έχουν συλλεχθεί γι' αυτόν.



Σχήμα 2.2 Τεχνολογίες Προσαρμοστικότητας<sup>6</sup>

<sup>6</sup> Methods and Techniques of Adaptive Hypermedia, Peter Brusilovsky, 1996 Kluwer Academic Publishers, Netherlands



## 2.5 Τα χρώματα στη μάθηση και το λογισμικό

Όταν σχεδιάζουμε ένα εκπαιδευτικό λογισμικό πρέπει, για να προσελκύει το μαθητή στο να το χρησιμοποιήσει, να είναι λειτουργικό αλλά και να καλύπτει τις αισθητικές ανάγκες του μαθητή. Συνδυάζοντας, λοιπόν τα κατάλληλα χρώματα μπορούμε να κάνουμε το λογισμικό πιο καλαίσθητο, περισσότερο λειτουργικό και εργονομικό. Μέσα από πολύχρωμα γραφικά, video, flash και ετικέτες μπορούμε να το επιτύχουμε.

Τα χρώματα μπορούν να χρησιμοποιηθούν ώστε προκαλώντας συγκεκριμένα συναισθήματα σε κάθε μαθητή να τον κάνουν να τα συνδέσει με συγκεκριμένες έννοιες. Σύμφωνα με έρευνες, οι έγχρωμες εικόνες μπορούν να αποτυπωθούν πιο εύκολα στο μυαλό μας από ότι οι ασπρόμαυρες καθώς μέσα από τα χρώματα προσελκύουν την προσοχή και τονίζουν έννοιες και μηνύματα. Τα παιδιά αρέσκονται στο να είναι τα γραφικά πολύχρωμα διότι νιώθουν ότι η διαδικασία της μάθησης γίνεται πιο διασκεδαστική.

Ωστόσο υπάρχει ο κίνδυνος στην προσπάθεια να γίνει το λογισμικό πιο ελκυστικό μέσα από την χρήση χρωμάτων να πετύχουμε το ακριβώς αντίθετο. Για παράδειγμα αν σε κάποια σημείο του λογισμικού έχουμε πολύ έντονα γράμματα με έντονο φόντο τότε το γεγονός αυτό καθιστά πολύ δύσκολη την ανάγνωση. Η χρήση απαλών χρωμάτων βοηθά στο να αποφύγουμε την κούραση των ματιών και να κάνουμε το κείμενο δυσανάγνωστο. Επιπλέον υπάρχουν χρώματα που από τη φύση τους είναι πιο κουραστικά στο μάτι, όπως το κόκκινο και άλλα που είναι λιγότερο όπως το μπλε. Ένα ακόμη στοιχείο στο οποίο πρέπει να δώσουμε προσοχή είναι ο διαχωρισμός της οθόνης σε μεγάλες περιοχές με αντίθετα χρώματα ή πολύ έντονα χρώματα. Κάτι τέτοιο θα οδηγήσει σε κούραση του μαθητή και μη ευχάριστη χρήση του.

Τα χρώματα πολλές φορές συνδέονται με διάφορες καταστάσεις. Αν θέλουμε να τονίσουμε στον μαθητή ότι πρέπει να σταματήσει ή να προσέξει μπορούμε να το συμβολίσουμε με κόκκινο χρώμα ενώ για κάτι θετικό μπορούμε να συμβολίσουμε με πράσινο χρώμα. Τέλος για μεγαλύτερη αρμονία μπορούμε να χρησιμοποιούμε όσο το δυνατόν λιγότερα συμπληρωματικά χρώματα διότι ο συνδυασμός τους κάνει δυσανάγνωστο το περιεχόμενο.

### **3. Υλοποίηση της εργασίας**

#### **3.1 Περιγραφή της 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.

#### **3.2 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.

##### **3.2.1 Δυνατότητες της εφαρμογής**

Η εφαρμογή “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

### 3.2.2 Βάση Δεδομένων

Για να σχεδιάσουμε την βάση δεδομένων αρχικά καθορίσαμε τις απαιτήσεις του χρήστη από το σύστημα. Στη συνέχεια συγκεντρώθηκαν όλες οι πληροφορίες που σχετίζονται με το πρόγραμμα. Για να σχεδιάσουμε την βάση χρησιμοποιήθηκε το πρόγραμμα της Microsoft , SQL Server 2008 R2. Οι πίνακες που δημιουργήθηκαν ήταν οι εξής:


1. User\_Profile
2. Results
3. AssistImage
4. questions
5. Reading

#### Πίνακας User\_Profile

Στον πίνακα User\_profile αποθηκεύονται όλα τα απαραίτητα στοιχεία που σχετίζονται με τον λογαριασμό του κάθε χρήστη. Αποτελείται από τις δέκα ακόλουθες στήλες.

- ID: (Primary Key) Αποθηκεύει το id του κάθε χρήστη και αποτελεί το μοναδικό χαρακτηριστικό του (int)
- name: Αποθηκεύει το όνομα του κάθε χρήστη. (nvarchar)
- username: Αποθηκεύει το username του κάθε χρήστη. (nvarchar)
- lastname: Αποθηκεύει το επώνυμο του κάθε χρήστη. (nvarchar)
- password: Αποθηκεύει ο κωδικός πρόσβασης του κάθε χρήστη. (nvarchar)
- sex: Αποθηκεύει 1 εάν ο χρήστης είναι αγόρι και 0 εάν είναι κορίτσι. (int)
- age: Αποθηκεύει την ηλικία του κάθε χρήστη. (int)
- dateregister: Καταχωρείται η ώρα και η ημερομηνία που έκανε registration ο χρήστης. (varchar)

- assistant: Αποθηκεύει το ID του βοηθού που έχει επιλέξει ο χρήστης. (int)
- coins: Αποθηκεύει τα νομίσματα που έχει στην κατοχή του ο χρήστης. (int)

User_Profile *	
	ID
	name
	lastname
	username
	password
	sex
	age
	dateregister
	assistant
	coins

Εικόνα 3.1 Ο πίνακας User\_profile

### Πίνακας AssistImage

Στον πίνακα 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: είναι αποθηκευμένα τα ID που χαρακτηρίζουν κάθε θεό (int)
- nameAssist: Αποθηκεύει το όνομα του κάθε βοηθού που έχει τη δυνατότητα να επιλέξει ο χρήστης. (nvarchar)
- assistPhoto: Αποθηκεύει το όνομα της φωτογραφίας του κάθε βοηθού που εμφανίζεται στην εφαρμογή. (nvarchar)
- 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. (nvarchar)

ILIA-PC.greekmyth...- dbo.AssistImage		ILIA-PC.greekmyth...- dbo.User_Profile		
	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

Εικόνα 3.2 : Εγγραφές του πίνακα AssistImage

### Πίνακας questions

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

- id\_q: Αποτελεί το πρωτεύον κλειδί (primary key) και εμπεριέχει το id της εκάστοτε ερώτησης. (int)
- question: Είναι καταχωρημένες όλες οι ερωτήσεις που εμφανίζονται τυχαία σε κάθε τεστ. (nvarchar)
- answer\_right: Είναι καταχωρημένες όλες οι σωστές απαντήσεις. (nvarchar)
- aswer\_wrong1: Είναι καταχωρημένη η πρώτη λάθος απάντηση. (nvarchar)
- aswer\_wrong2: Είναι καταχωρημένη η δεύτερη λάθος απάντηση. (nvarchar)
- aswer\_wrong3: Είναι καταχωρημένη η τρίτη λάθος απάντηση. (nvarchar)
- level1: Καταχωρείται το επίπεδο δυσκολίας της εκάστοτε ερώτησης. (int)
- chapter: Καταχωρείται το κεφάλαιο στο οποίο ανήκει η κάθε ερώτηση. (int)
- typeq: Έχουμε τρεις διαφορετικούς τύπους ερωτήσεων. Στο rw ανήκουν οι ερωτήσεις Σωστού- Λάθους. Στο fb ανήκουν οι ερωτήσεις συμπλήρωσης κενού. Στο mc ανήκουν οι ερωτήσεις οι ερωτήσεις πολλαπλής επιλογής. (nvarchar)

questions	
id_q	
question	
answer_right	
answer_wrong1	
answer_wrong2	
answer_wrong3	
level1	
chapter	
typeq	

Εικόνα 3.3: Ο πίνακας questions

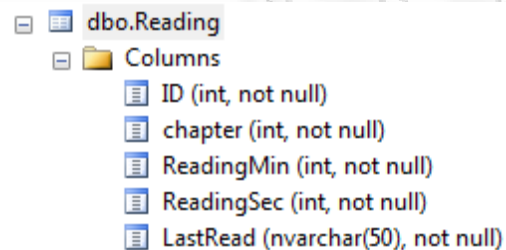
id_q	question	answer_right	answer_wrong1	answer_wrong2	answer_wrong3	level1
1	Τα δώδεκα παιδι...	Σωστό	Λάθος	NULL	NULL	1
2	Ο Κρόνος παντρ...	Λάθος	Σωστό	NULL	NULL	1
3	Η Τιτανομαχία ή...	τον Δία και τα α...	τον Δία	τον Δία και την ...	τον Δία και την ...	1
4	Το βουνό στο οη...	ΟΛΥΜΠΟΣ	NULL	NULL	NULL	1
5	Στην Τιτανομαχί...	οι Ολύμπιοι Θεοί	οι Τιτάνες	Οι Εκατόγχοιρες	ο Δίας	1
6	Οι Θεοί του Ολύ...	12	10	8	6	1
7	Η Αφροδίτη ήτα...	του έρωτα	του κυνηγιού	της σοφίας	του Πολέμου	1
8	Ο Κρόνος είχε κ...	ΕΣΤΙΑ	NULL	NULL	NULL	2
9	Ο Άρης ήταν ο ...	του πολέμου	του κυνηγιού	του έρωτα	της φωτιάς	1
10	Ο Άδης είναι γν...	Πλούτωνας	Άρης	Απόλλωνας	Ερμής	2
11	Η Πυθία:	Έδινε χρησμούς	Ήταν Θεά	Ήταν μούσα	Ήταν νύμφη	2
12	Οι επισκέπτες πο...	Στην Κασταλία π...	Στον Όλυμπο	Σε μια πηγή κον...	Στην Αρχαία Ολ...	2
13	Οι επισκέπτες πο...	Σωστό	Λάθος	NULL	NULL	3
14	Οι χρησμοί δίνον...	Λάθος	Σωστό	NULL	NULL	2
15	Η Ιασώ ήταν η θ...	Σωστό	Λάθος	NULL	NULL	1
16	Οι Εκατόγχοιρες ...	Λάθος	Σωστό	NULL	NULL	3
17	Ο Κρόνος είχε ε...	Σωστό	Λάθος	NULL	NULL	2
18	Ο κατασκευαστή...	Ήφαιστος	Απόλλωνας	Διώνυσος	Άρης	1
19	Αρχικά οι χρησμ...	Σωστό	Λάθος	NULL	NULL	3
20	Ο Ζεύς άφησε δ...	Λάθος	Σωστό	NULL	NULL	3
21	Ο φόρος που πλ...	Λάθος	Σωστό	NULL	NULL	3
22	Πόσες ήταν οι μ...	9	15	6	4	1
23	Η πρώτη γενιά μ...	Λάθος	Σωστό	NULL	NULL	1

Εικόνα 3.4: Εγγραφές του πίνακα questions

## Πίνακας 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: Γίνεται εγγραφή του ID χρήστη κάθε φορά που διαβάζει ένα κεφάλαιο. (int)
- chapter: Αποθηκεύει το κεφάλαιο που έχει διαβάσει ο χρήστης. (int)
- ReadingMin: Αποθηκεύει τα λεπτά που έχει αφιερώσει ο χρήστης στο διάβασμα του τεστ την τελευταία φορά. (int)
- ReadingSec: Αποθηκεύει τα δευτερόλεπτα που έχει αφιερώσει ο χρήστης στο διάβασμα του τεστ την τελευταία φορά. (int)
- LastRead: Αποθηκεύει την ημερομηνία καθώς και ώρα που ο χρήστης έχει διαβάσει το διαθέσιμο υλικό για το συγκεκριμένο κεφάλαιο. (nvarchar)



Εικόνα 3.5: Ο πίνακας Reading

## Πίνακας Results

Στον πίνακα 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.
- completest: 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".

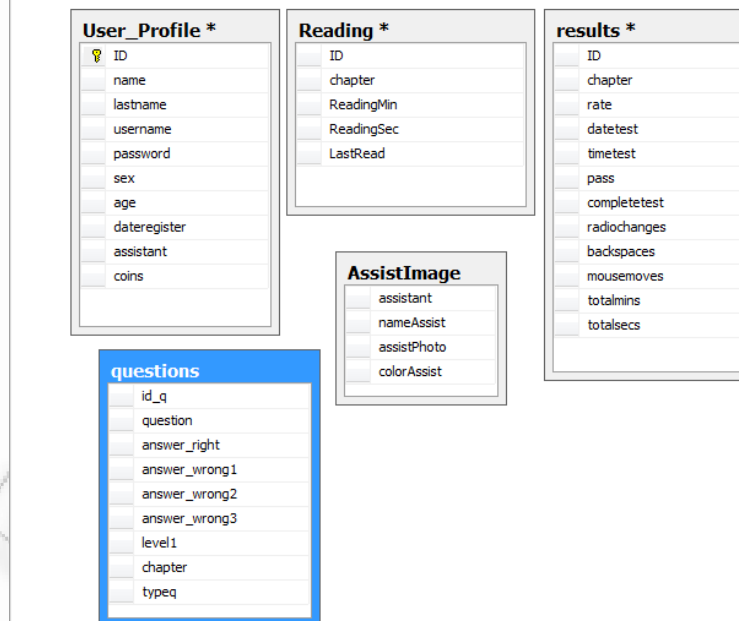


- mousemoves: Αποθηκεύει τις φορές που ο χρήστης μετακινεί το ποντίκι του.
- totalmins: Αποθηκεύει τα συνολικά λεπτά που έχει αφιερώσει ο χρήστης σε κάθε τεστ.
- totalsecs: Αποθηκεύει τα συνολικά δευτερόλεπτα που έχει αφιερώσει ο χρήστης σε κάθε τεστ.

Εικόνα 3.6: Εγγραφές του πίνακα results

ID	chapter	rate	datetest	timetest	pass	completetest	radiochanges	backspaces	mousemove	totalmi...	tr	
196	3...	1	73	02-06-2012	0:46	1	1	37	3	424	0	4
197	3...	1	30	02-12-2012	0:50	0	1	45	0	1019	0	5
198	3...	1	34	02-14-2012	0:35	0	1	52	0	852	0	6
199	3...	1	39	02-14-2012	0:38	0	1	44	1	738	0	7
200	4...	1	53	02-16-2012	0:27	1	1	40	0	473	0	8
201	4...	1	53	02-16-2012	0:42	1	1	35	1	666	0	9
202	4...	1	66	02-16-2012	0:28	1	1	36	0	587	0	10
203	4...	1	44	02-16-2012	0:25	0	1	54	0	502	0	11
204	4...	1	35	03-05-2012	0:29	0	1	43	6	700	0	12
205	4...	2	53	03-05-2012	0:35	1	1	33	0	819	0	13
206	4...	1	17	03-06-2012	0:32	0	1	53	0	763	0	14
207	4...	1	60	03-06-2012	0:37	1	1	38	1	838	0	15
208	4...	1	43	03-06-2012	0:24	0	1	61	0	740	0	16
209	4...	1	30	03-06-2012	0:28	0	1	46	0	749	0	17
210	4...	1	69	03-21-2012	0:36	1	1	55	0	857	0	18

Query executed successfully. | localhost (10.50 RTM) | ilia-PC\ilia (54) | master | 00:00:00 | 210 rows



Εικόνα 3.7: Απεικόνιση όλων των πινάκων της βάσης



### 3.3 Ασφάλεια δεδομένων

Για να αποφύγουμε ανεπιθύμητες καταστάσεις όπως είναι η «επίθεση» στο λογισμικό που φτιάξαμε θα πρέπει να είμαστε προσεκτικοί με τα θέματα ασφάλειας. Υπάρχει περίπτωση κάποιος να αντλήσει σημαντικές πληροφορίες από την βάση δεδομένων κάνοντας μια επίθεση (SQL Injection). Για να αποτρέψουμε να γίνει κάτι τέτοιο δημιουργήσαμε δύο κλάσεις που το επιτυγχάνουν, την `Sqlqueries.cs` και την `MD5EncryptDecrypt.cs`.

#### 3.3.1 Η κλάση `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 statement 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.

#### 3.3.2 Η κλάση `MD5EncryptDecrypt.cs`

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)
    {
        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;
    }
}
```

Κώδικας 3.1: Κώδικας από την κλάση MD5EncryptDecrypt.cs

### 3.4 Περιγραφή υλοποίησης των Windows Forms της εφαρμογής

#### 3.4.1 Εισαγωγή του χρήστη στο σύστημα (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(textBoxnamelogin.Text.ToString(), newst.ToString());
    bool findit = query.FindData;
    if (findit)
    {
        SqlQueries GetUserGredentials = new SqlQueries();
        string hhhhh =
        GetUserGredentials.UserHaveLogg(textBoxnamelogin.Text.ToString());

        mythlab mythform = new mythlab();

        FormTests questionform = new FormTests();
        string welname = textBoxnamelogin.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"); }
}

```

Κώδικας 3.2: Απόσπασμα από την loginuser.cs

### 3.4.2 Εγγραφή νέου 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.

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;
}
else if (comboBox1.SelectedItem.ToString() == "Ήρα")
{
    Assist = 2;
}
else if (comboBox1.SelectedItem.ToString() == "Αθηνά")
{
    Assist = 3;
}
else if (comboBox1.SelectedItem.ToString() == "Άρτεμις")
{
    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() == "Ήφαιστος")
{
    Assist = 8;
}
else if (comboBox1.SelectedItem.ToString() == "Δήμητρα")
{
    Assist = 9;
}
else if (comboBox1.SelectedItem.ToString() == "Άρης")
{
    Assist = 10;
}
else if (comboBox1.SelectedItem.ToString() == "Αδης")
{
    Assist = 11;
}
else if (comboBox1.SelectedItem.ToString() == "Ερμής")
{
    Assist = 12;
}
```

**Κώδικας 3.3:** Καταχώρηση τιμής στην μεταβλητή Assist ανάλογα με τον επιλεγμένο θεό

```

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

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

```

**Κώδικας 3.4:** Καταχώρηση όλων των δεδομένων στη βάση και εμφάνιση μηνύματος στην οθόνη

### 3.4.3 Δημιουργία νέου κωδικού (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");
    }
}

```

**Κώδικας 3.5:** Δημιουργία νέου κωδικού

### 3.4.4 Εισαγωγή στην κεντρική φόρμα της εφαρμογής (mythlab.cs)

Εφόσον ο χρήστης δημιουργήσει και εισάγει τα σωστά στοιχεία που του ζητούνται για να εισαχθεί στο σύστημα, οδηγείται στην φόρμα όπου προβάλλονται όλες οι δυνατότητες που παρέχονται από το εκπαιδευτικό λογισμικό.

Όταν φορτώνεται η φόρμα αυτή φορτώνονται και τα εξής στοιχεία:

- 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();

```

Κώδικας 3.6: Φόρτωση όλων των δεδομένων που χρειάζονται από την βάση δεδομένων

```

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;

```

Κώδικας 3.7: Δημιουργία SQL query για εμφάνιση δεδομένων



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;
}
}
```

### Κώδικας 3.8: Επιλογή στο χρώμα των ετικετών 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 η 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;
```

**Κώδικας 3.9:** Εύρεση των κεφαλαίων που έχει ξεκλειδώσει ο χρήστης

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(label11.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();
}
```

### Κώδικας 3.10: Λειτουργία του ημερολογίου

### 3.4.5 Ανάγνωση του υλικού για μελέτη (reader.cs)

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 ανοιχτές 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 ανοιχτό 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 καθώς χρησιμοποιούν 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!");
        }
    }
}

```

Κώδικας 3.11: Εμφάνιση μηνυμάτων την ώρα που έχει στρέψει την προσοχή του αλλού

### 3.4.6 Επιλογή θεού (gods.cs)

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();
}
```

Κώδικας 3.12: Εύρεση του ID του χρήστη

```
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;
}
```

Κώδικας 3.13: Ανανέωση στη βάση δεδομένων με την νέα επιλογή του θεού



### 3.4.7 Επισκόπηση λογαριασμού του χρήστη (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.

### 3.4.8 Στοιχεία σε σχέση με την απόδοση του χρήστη (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=" + usersID
+ "order by Ημερομηνία");
```

#### Κώδικας 3.14: Εμφάνιση του dataGridView

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;
}

```

**Κώδικας 3.15:** Χρωματισμός γραμμών του DataGrid ανάλογα με την απόδοση

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");

```

**Κώδικας 3.16:** Εμφάνιση στοιχείων στο DataGrid ανάλογα με το επιλεγμένο κεφάλαιο

### 3.4.9 Ανάλυση της σχεδίασης των τεστ (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.

### 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;
}
```

Κώδικας 3.17: Ορισμός της χρονικής διάρκειας του τεστ ανάλογα με την ηλικία

```

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;
}

```

**Κώδικας 3.18: Εύρεση ηλικίας μαθητή από την βάση δεδομένων**

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)
    {
        label11.Text = "0" + mins.ToString();
    }
    else
    {
        label11.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();
    }
}
```

**Κώδικας 3.19: Λειτουργία του timer**

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`.

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

#### Κώδικας 3.20: Τερματισμός του timer

#### 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 εύκολος, μέτριο και δύσκολος 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;
}
```

Κώδικας 3.21: Επιλογή επιπέδου δυσκολίας ανάλογα με την μεγαλύτερη απόδοση

### 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;
    }

```

### Κώδικας 3.22: Επιλογή επιπέδου δυσκολίας ανάλογα με την μεγαλύτερη απόδοση

Αφού βρέθηκαν οι ερωτήσεις από τις οποίες θα αποτελείται το τεστ έπρεπε να εξεταστεί με ποιό 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 milliseconds, 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.

### **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;
    }
}
}

```

**Κώδικας 3.23:** Έλεγχος γλώσσας πληκτρολογίου

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

Κώδικας 3.24: Έλεγχος για το αν ο χρήστης έχει στρέψει την προσοχή του αλλού

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.

### 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;
```

### Κώδικας 3.25: Υπολογισμός βαθμολογίας

#### 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)

Με την αποθήκευση των στοιχείων εμφανίζεται η analytics.cs

```
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());
    }
}
```

**Κώδικας 3.26:** Αποθήκευση όλων των στοιχείων που κατεγράφησαν κατά τη διάρκεια του τεστ στη βάση δεδομένων

### 3.4.1 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 απάντηση 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!».

### 3.5 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 άδειο, 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.











Εικόνα 3.8: Εικονίδια που εμφανίζονται στην οθόνη του χρήστη ανάλογα με την απόδοση

#### 3.5.1 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
<b>χρόνος&gt;=7 Λεπτά</b>	Δεν υπολογίζεται	Θα μπορούσες να επιτύχεις καλύτερο χρόνο. Προσπάθησε ξανά για καλύτερο αποτέλεσμα		stop=stop+1
<b>5=&lt;Χρόνος&lt;7 λεπτά</b>	Δεν υπολογίζεται	Ο χρόνος ολοκλήρωσης του τεστ είναι αρκετά ικανοποιητικός		star=star+2
<b>1=&lt;Χρόνος&lt;5λεπτά</b>	Βαθμολογία<50%	Προσπάθησε να ξανακάνεις το τεστ και κοίτα τις ερωτήσεις περισσότερη ώρα για να τις κατανοήσεις		star=star+1
	Βαθμολογία>=50%	Ο χρόνος ολοκλήρωσης του τεστ είναι πολύ καλός.		star=star+3
<b>Χρόνος&lt;1 λεπτό</b>	Βαθμολογία<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
	50%=<Βαθμολογία<60%	Ολοκλήρωσες το test πολύ γρήγορα. Αν κάνεις το τεστ πιο προσεκτικά θα πετύχεις καλύτερη βαθμολογία		star=star + 1
	60%=<Βαθμολογία<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%=<Βαθμολογία<=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 3.1: Επεξήγηση για το πώς και πότε εμφανίζονται τα εικονίδια σε σχέση με την διάρκεια ολοκλήρωσης του τεστ**

### 3.5.2 Βαθμολογία

- Για βαθμολογία κάτω από 50% παίρνει ένα απαγορευτικό εικονίδιο και η μεταβλητή stop 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.









### 3.5.3 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 4 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.

Πιο αναλυτικά γίνονται οι εξής έλεγχοι όπως φαίνεται στον ακόλουθο πίνακα:

Έλεγχοι για την διάρκεια μελέτης				
Βαθμολογία	Χρόνος Διαβάσματος	Μήνυμα	Εικονίδιο	Μεταβλητή
<50%	Πάνω από τον ενδεδειγμένο	Αν και έχεις διαβάσει αρκετή ώρα η βαθμολογία σου δεν είναι καλή. Προσπάθησε κάθε φορά που διαβάζεις μια παράγραφο να προσπαθείς να την λες φωναχτά μόνος σου!		star=star+2
	Κάτω από τον ενδεδειγμένο	Μπορείς να κάνεις κλικ στο κουμπί '12' στην αρχική σελίδα της εφαρμογής και να μάθεις την ιδιότητα κάθε θεού. Είναι μία καλή αρχή!";		stop=stop+1
50% ≤ βαθμός < 60%	Πάνω από τον ενδεδειγμένο	Έχεις διαβάσει αρκετά! Μπράβο!		star=star+3
	Κάτω από τον ενδεδειγμένο	Αν διαβάσεις ξανά την ενότητα η βαθμολογία σου θα βελτιωθεί		stop=stop+1
60% ≤ βαθμός < 85%	Πάνω από τον ενδεδειγμένο	"Το διάβασμα απέδωσε! Μπράβο"		star=star+3
	Κάτω από τον ενδεδειγμένο	Αν και δεν έχεις διαβάσει αρκετά η βαθμολογία σου είναι καλή! \r\n Μπορεί όμως να μην είσαι πάντα τόσο τυχερός!";		star=star+1
85% ≤ βαθμός < 100%	Πάνω από τον ενδεδειγμένο	Το διάβασμα απέδωσε! Μπράβο"		star=star+3
	Κάτω από τον ενδεδειγμένο	Αν και δεν έχεις διαβάσει αρκετά η βαθμολογία σου είναι καλή! \r\n Μπορεί όμως να μην είσαι πάντα τόσο τυχερός!";		star=star+1

100%	Πάνω από τον ενδεδειγμένο	Η βαθμολογία σου είναι άριστη		star=star+3
	Κάτω από τον ενδεδειγμένο	Αν και δεν έχεις διαβάσει αρκετά η βαθμολογία σου είναι άριστη!		star=star+1

**Πίνακας 3.2: Επεξήγηση για το πώς και πότε εμφανίζονται τα εικονίδια σε σχέση με την διάρκεια μελέτης**

Μετά από αυτούς τους ελέγχους εξετάζεται η τιμή της μεταβλητής *star* και η τιμή της μεταβλητής *stop*. Στην περίπτωση που η μεταβλητή *star* έχει τιμή πάνω από τρία αλλά μικρότερη ή ίση από τέσσερα ( $3 < star \leq 4$ ) προτείνεται στον μαθητή να επαναλάβει το τεστ για ακόμα καλύτερα αποτελέσματα!" και εμφανίζεται το εξής εικονίδιο: . Αν η μεταβλητή *stop* έχει τιμή πάνω από δύο ( $stop \geq 2$ ) τότε εμφανίζεται το ακόλουθο μήνυμα: "Θα ήταν καλύτερα να κάνεις επανάληψη και να προσπαθήσεις να κάνεις τεστ του ίδιου κεφαλαίου!" και εμφανίζεται το εξής εικονίδιο: . Σε κάθε άλλη περίπτωση ο χρήστης μπορεί να συνεχίσει σε επόμενο τεστ όπως του καταδεικνύει το αντίστοιχο εικονίδιο .

### 3.6 Τα χρώματα στην εφαρμογή "Olympus 12"

Όπως αναλύθηκε στην ενότητα 2.5 ένα εκπαιδευτικό λογισμικό πρέπει να είναι ελκυστικό ώστε να τραβάει την προσοχή του χρήστη. Στην εφαρμογή "Olympus 12" έχουν χρησιμοποιηθεί γραφικά τα οποία επιτελούν αυτόν το σκοπό.

Συγκεκριμένα στα βιβλία εκμάθησης του υλικού έχουν χρησιμοποιηθεί γραμματοσειρές οι οποίες δεν έχουν έντονο χρώμα. Τα κείμενα είναι γραμμένα με μπλε χρώμα το οποίο δημιουργεί μικρότερη κούραση στο μάτι. Οι θεοί απεικονίζονται με ευχάριστα χρώματα και ο κάθε θεός έχει το δικό του χρώμα καθώς θυμόμαστε περισσότερο αντικείμενα στα οποία δώσαμε προσοχή γιατί το χρώμα αυξάνει την ταχύτητα και την ακρίβεια με την οποία αφομοιώνουμε τα αντικείμενα.

Στην παρουσίαση των αποτελεσμάτων εμφανίζεται και ένα flash το οποίο έχει και ήχο και κάνει πιο ευχάριστη την φόρμα. Τέλος ο μαθητής μπορεί να επιλέξει μόνος του μέσα από 12 χρώματα και να ορίσει αυτός το χρώμα φόντου που θέλει να έχει η εφαρμογή.

## 4. Παρουσίαση του εκπαιδευτικού λογισμικού “Olympus 12”

### 4.1 Εγγραφή νέου χρήστη




Όταν ο χρήστης ανοίξει την εφαρμογή πατώντας το εικονίδιο βλέπει την παρακάτω εικόνα:



Εικόνα 4.1: Παράθυρο εισόδου

Στο σημείο αυτό υπάρχει η περίπτωση να είναι εγγεγραμμένος χρήστης και να έχει χρησιμοποιήσει την εφαρμογή στο παρελθόν, ή να την χρησιμοποιεί για πρώτη φορά.

Αν χρησιμοποιεί για πρώτη φορά την εφαρμογή θα πρέπει να δημιουργήσει έναν λογαριασμό για να αποθηκευθούν τα στοιχεία του στη βάση δεδομένων έτσι ώστε κάθε φορά που μπαίνει να καταγράφονται οι κινήσεις του και να βλέπει την εξέλιξή του. Για να κάνει νέο λογαριασμό πρέπει

να κάνει κλικ στο εικονίδιο  (περνώντας τον κέρσορα πάνω από το εικονίδιο εμφανίζεται η επεξήγηση: Δημιουργία Λογαριασμού). Κάνοντας κλικ στο εικονίδιο αυτό εμφανίζεται η φόρμα στην οποία ο χρήστης πρέπει να συμπληρώσει τα στοιχεία που του ζητούνται. Όλα τα πεδία της φόρμας είναι υποχρεωτικά.

Εγγραφή Νέου χρήστη

Όνομα

Επίθετο

Όνομα Χρήστη

Κωδικός

Επαλήθευση Κωδικού

Ηλικία

Φύλο  Αγόρι  Κορίτσι

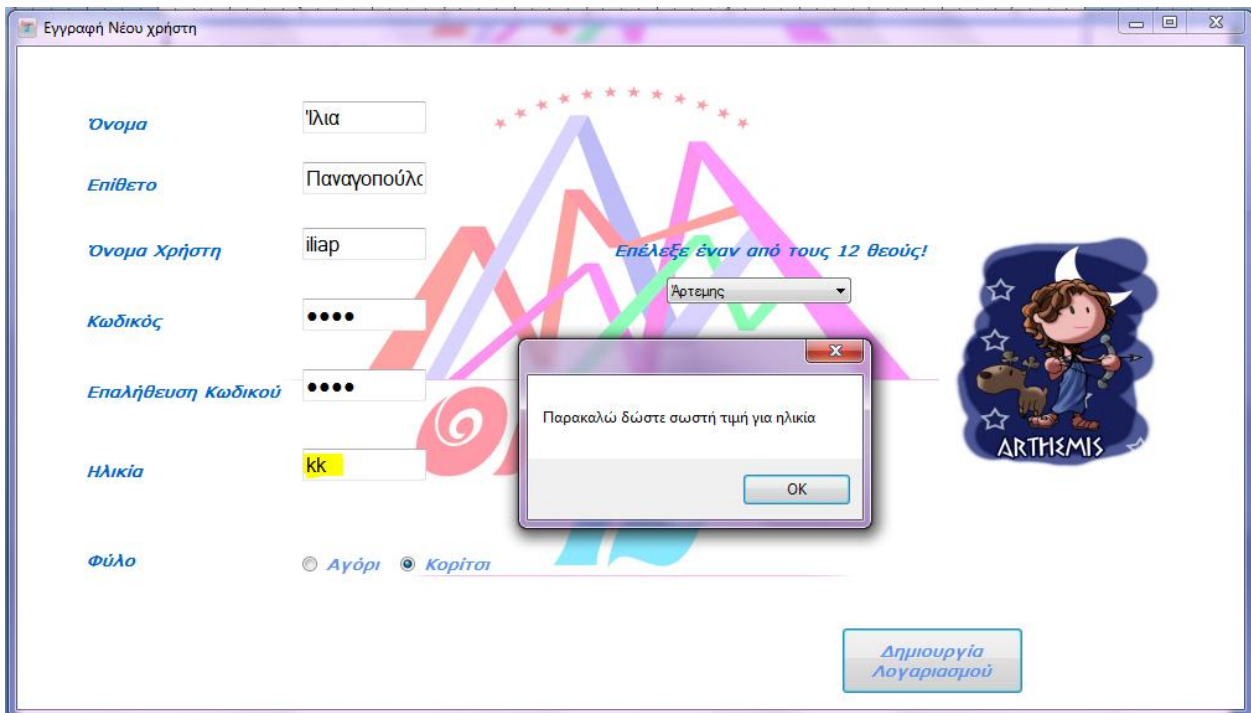
Επέλεξε έναν από τους 12 θεούς!

Δημιουργία Λογαριασμού

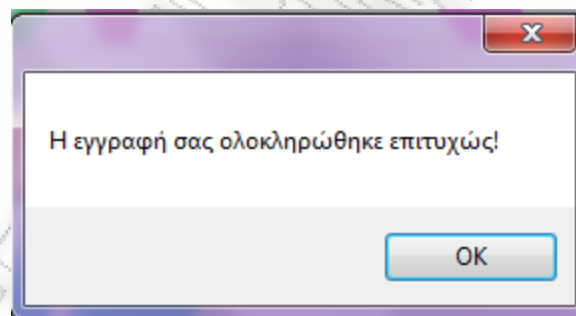
Εικόνα 4.2: Παράθυρο δημιουργίας νέου χρήστη

Αν ο χρήστης ξεχάσει κάποιο πεδίο ή πληκτρολογήσει κάτι το οποίο είναι λάθος ενημερώνεται έτσι ώστε να κάνει την σχετική διόρθωση. Μόλις ολοκληρώσει την διαδικασία κάνει κλικ στο κουμπί «Δημιουργία λογαριασμού» και αν είναι όλα τα πεδία σωστά τα δεδομένα καταχωρούνται στη βάση. Ακολουθεί ενδεικτικό παράδειγμα συμπλήρωσης της φόρμας εγγραφής.



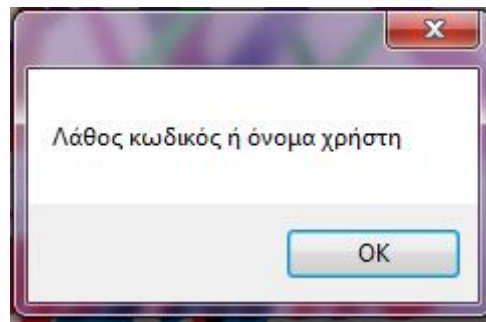


Εικόνα 4.3: Ο χρήστης έχει πληκτρολογήσει μη έγκυρους χαρακτήρες στο πεδίο ηλικία



Εικόνα 4.4: Μήνυμα ολοκλήρωσης εγγραφής

Αφού ολοκληρωθεί η εγγραφή του χρήστη μπορεί να εισάγει το username του και το password στα αντίστοιχα πεδία της φόρμας (Εικόνα 4.1). Αν εισάγει σωστά τα στοιχεία εισέρχεται στην εφαρμογή διαφορετικά εμφανίζεται στην οθόνη μήνυμα λάθους για να εισάγει ξανά τα στοιχεία του.



Εικόνα 4.5: Μήνυμα λανθασμένων στοιχείων εισαγωγής

## 4.2 Παρουσίαση της κεντρικής φόρμας

Όταν εισέλθει στην κεντρική φόρμα του προγράμματος βλέπει στο πάνω μέρος το όνομα χρήστη που έχει δηλώσει, τα διαθέσιμα νομίσματα, τη θεότητα που έχει επιλέξει, τα κεφάλαια που είναι ξεκλειδωτά, διαθέσιμα δηλαδή για ανάγνωση ή για εξάσκηση και ένα ημερολόγιο για να επιλέξει ημερομηνία και να δει αν την συγκεκριμένη ημέρα είχε κάποια δραστηριότητα στην εφαρμογή. Στην φόρμα αυτή υπάρχουν κουμπιά που του δίνουν τις εξής δυνατότητες:

- Επίσκεψη στο μαντείο των Δελφών
- Ανάγνωση
- Έλεγχος γνώσεων
- Αλλαγή επιλεγμένης θεότητας
- Αναλυτικά στοιχεία
- Ο Λογαριασμός μου
- Βοήθεια

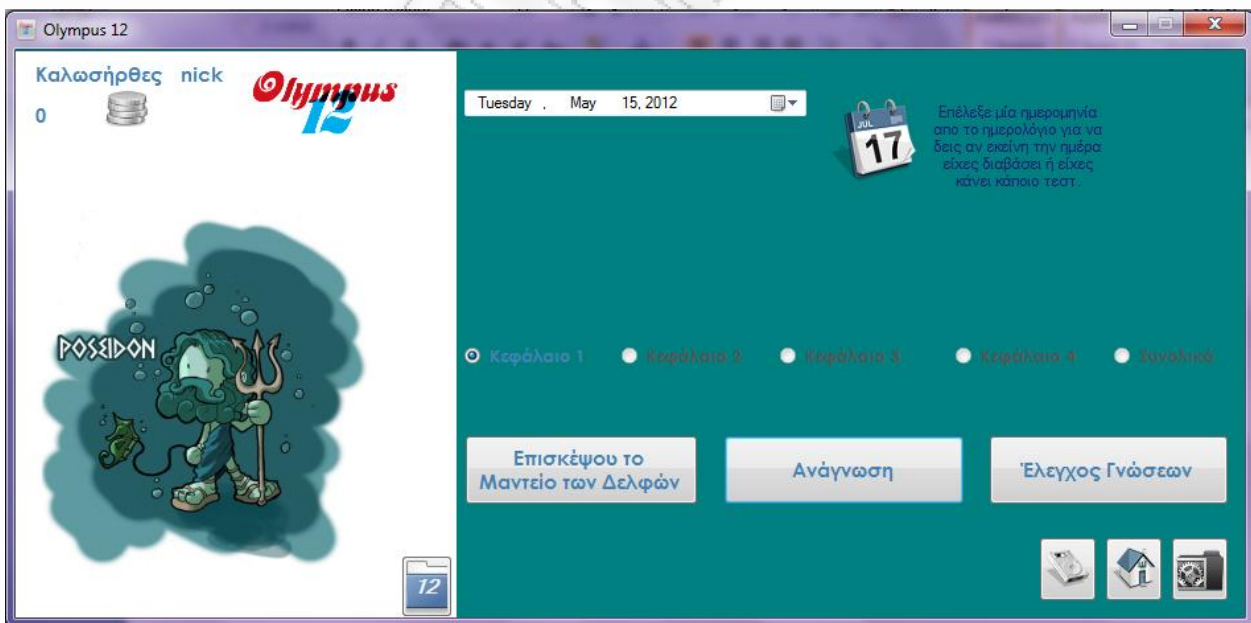
Αν ο χρήστης είναι κορίτσι όλα τα γράμματα των ετικετών της εφαρμογής έχουν το χρώμα ροζ αν είναι αγόρι το χρώμα μπλε. Η εφαρμογή έχει για χρώμα φόντου το χρώμα που αντιστοιχεί σε κάθε θεό. Για παράδειγμα στην εικόνα 4.6 ο χρήστης "iliar" είναι κορίτσι ως αποτέλεσμα όλες οι ετικέτες να είναι ροζ και έχει επιλέξει τη θεά Άρτεμη ορίζοντας δηλαδή η εφαρμογή έχει ως χρώμα φόντου το μωβ.





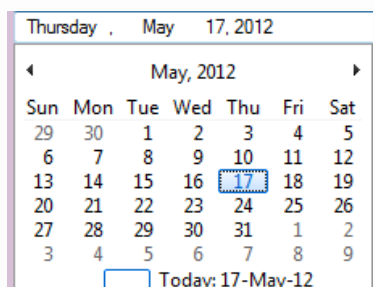
Εικόνα 4.6: Κεντρική φόρμα της εφαρμογής

Στην παρακάτω εικόνα ο χρήστης είναι αγόρι και έχει επιλέξει για θεό τον Ποσειδώνα. Οι ετικέτες έχουν τώρα χρώμα μπλε και φόντο μπλε.



Εικόνα 4.7: Αλλαγή χρώματος φόντου

Στο κεντρικό παράθυρο της εφαρμογής υπάρχει ένα ημερολόγιο από το οποίο μπορεί να επιλέξει ημερομηνία και να δει αν την επιλεγμένη ημέρα είχε κάνει κάποιο τεστ.



Εικόνα 4.8: Ημερολόγιο δραστηριοτήτων

### 4.3 Παρουσίαση της φόρμας των θεών

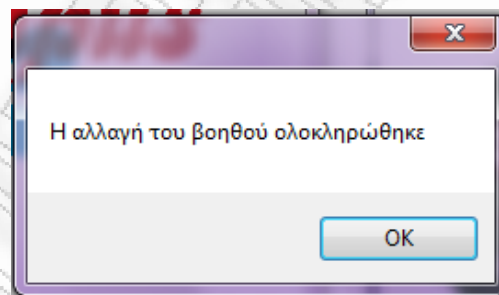
Κάνοντας κλικ στο κουμπί “12” ο χρήστης μπορεί να αλλάξει τον θεό που επέλεξε κατά την εγγραφή του και να ενημερωθεί για την ιδιότητα που έχει. Όταν εμφανιστεί η φόρμα μπορεί να επιλέξει από μία λίστα που έχει και τους 12 θεούς. Κάνοντας κλικ σε κάποιον θεό βλέπει την ιδιότητα του και το χρώμα φόντου που θα έχει η εφαρμογή. Για να επιβεβαιώσει την αλλαγή που θέλει κάνει κλικ στο κουμπί “Αλλαγή θεού.” Ένα μήνυμα τον ενημερώνει ότι έχει Όταν επιστρέψει στην αρχική φόρμα είναι αλλαγμένο το χρώμα φόντου της εφαρμογής.



Εικόνα 4.8: Ο χρήστης επιλέγει τον θεό Αδη



Εικόνα 4.9 Ο χρήστης επιλέγει την θεά Δήμητρα



Εικόνα 4.10: Μήνυμα ότι έχει αλλάξει θεότητα

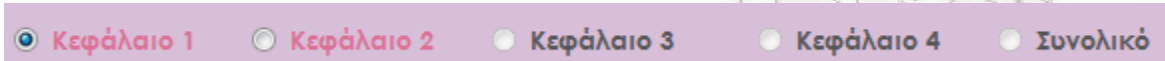
#### 4.4 Παρουσίαση της φόρμας με το υλικό μελέτης

Κάνοντας ο χρήστης κλικ στο κουμπί “Ανάγνωση” φορτώνεται ένα νέο παράθυρο το οποίο περιέχει το υλικό για το κεφάλαιο που έχει επιλέξει. Το “βιβλίο” είναι σε μορφή flash και ο χρήστης μπορεί να γυρίζει τις σελίδες όπως θα έκανε σε ένα κανονικό βιβλίο. Αυτό βοηθά στο να γίνει πιο εύκολη η χρήση του λογισμικού την ώρα της ανάγνωσης και να μην αποσπάται ο χρήστης από περιττά κουμπιά. Τα βιβλία μελέτης του υλικού έχουν σχεδιαστεί με τέτοιο τρόπο ώστε να είναι ευανάγνωστα. Έχουν χρησιμοποιηθεί χρώματα τα οποία είναι ξεκούραστα στο μάτι και δεν υπάρχουν μεγάλες αντιθέσεις. Για να αυξήσουμε την αναγνωσιμότητα των κειμένων δεν έχουν χρησιμοποιηθεί πολλές πολύχρωμες γραμματοσειρές για να είναι το υλικό περισσότερο

ευανάγνωστο. Από το μενού που υπάρχει στο κάτω μέρος του παραθύρου της ανάγνωσης ο χρήστης μπορεί να κάνει zoom στο υλικό να το εκτυπώσει και να πολογηθεί στον αριθμό της σελίδας που θέλει.

Διαδικασία ανάγνωσης

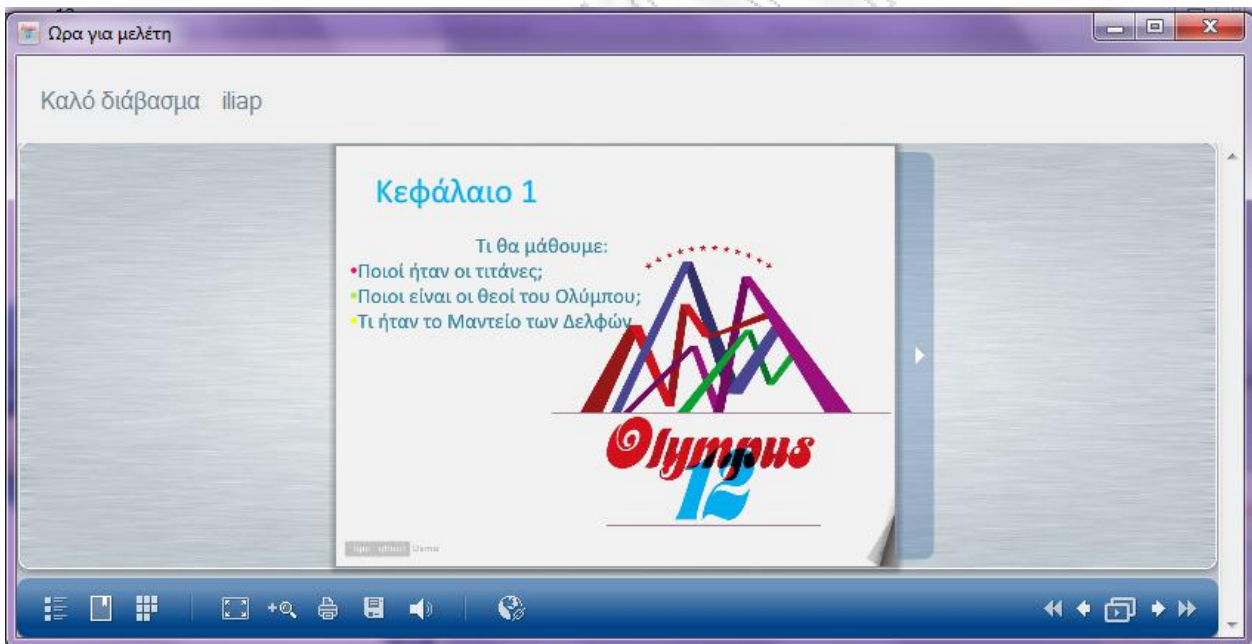
- Επιλογή κεφαλαίου



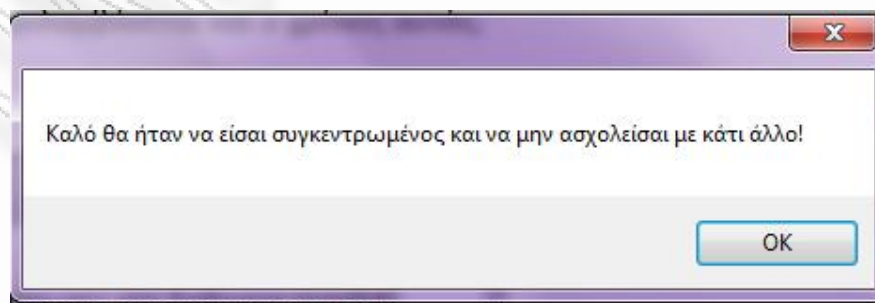
- Κλικ στο κουμπί “Ανάγνωση”



- Φόρτωση του παραθύρου με το υλικό



Εικόνα 4.11: Το παράθυρο με το υλικό για μελέτη



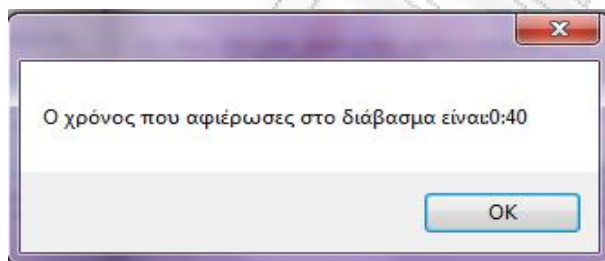


Εικόνα 4.12: Μήνυμα στον χρήστη το οποίο εμφανίζεται όταν δεν είναι αφωσιωμένος στη μελέτη

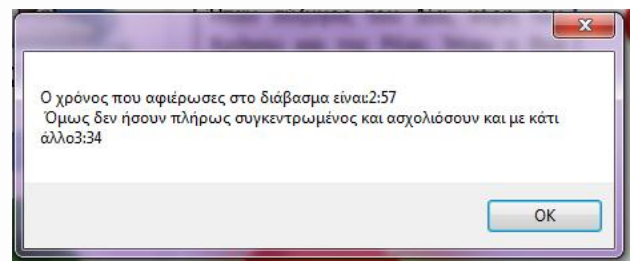


Εικόνα 4.13: Άποψη του υλικού μελέτης όταν ο χρήστης έχει κάνει zoom

Μόλις ολοκληρώσει την ανάγνωση και κλείσει το παράθυρο εμφανίζεται ο χρόνος που αφιέρωσε στο διάβασμα. Σε περίπτωση που είχε την προσοχή του στραμμένη αλλού κατά τη διάρκεια της ανάγνωσης στο μήνυμα συμπεριλαμβάνεται και ο χρόνος αυτός.



Εικόνα.4.14: ο χρόνος που αφιέρωσε στο διάβασμα ο χρήστης είναι 40''

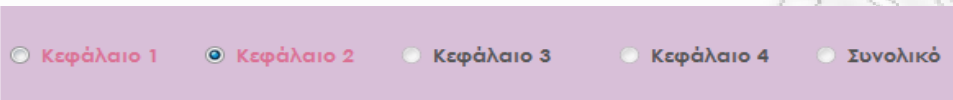


εικόνα 4.15: ο χρόνος που αφιέρωσε στο διάβασμα ο χρήστης είναι 2' και 57'' ενώ είχε την προσοχή του αλλού για 3' και 34''

#### 4.5 Παρουσίαση των τεστ

Για να γίνει η διεξαγωγή ενός τεστ πρέπει να γίνει η ακόλουθη διαδικασία:

- Ο χρήστης επιλέγει κεφάλαιο από αυτά που έχει ξεκλειδώσει. Τα κεφάλαια που είναι με γκρι χρώμα γραμματοσειράς είναι τα απενεργοποιημένα κεφάλαια.



- Στη συνέχεια κάνει κλικ στο κουμπί έλεγχος γνώσεων



- Το τεστ ανοίγεται μπροστά του

Στο πάνω μέρος της οθόνης ο χρήστης μπορεί να βλέπει το χρόνο που μετρά αντίστροφα. Υπάρχουν τρεις τύποι ερωτήσεων. Οι ερωτήσεις πολλαπλής επιλογής, που δίνονται τέσσερις πιθανές απαντήσεις από τις οποίες ο χρήστης θα διαλέξει μία. Οι ερωτήσεις “Σωστό- Λάθος” στις οποίες ο χρήστης κρίνει αν η πρόταση που του δίνεται είναι σωστή ή λάθος. Τέλος, υπάρχουν οι ερωτήσεις συμπλήρωσης κενού στις οποίες ο χρήστης πληκτρολογεί την απάντηση στα ελληνικά. Αν το πληκτρολόγιο του είναι ρυθμισμένο σε άλλη γλώσσα εμφανίζεται μήνυμα που του λέει πώς να αλλάξει την γλώσσα. Ο χρήστης πρέπει να απαντήσει στην ερώτηση που του εμφανίζεται διαφορετικά το κουμπί “Επόμενη” που βρίσκεται κάτω αριστερά δεν ενεργοποιείται. Μόλις δώσει μια απάντηση γίνεται ενεργοποίηση του κουμπιού για να πάει στην επόμενη ερώτηση.

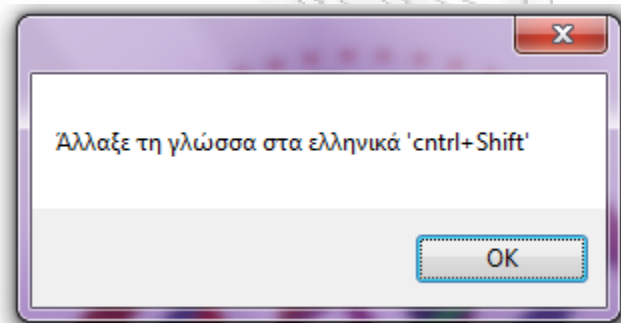
Οι Θεοί του Ολύμπου ήταν:

- 12
- 6
- 10
- 8

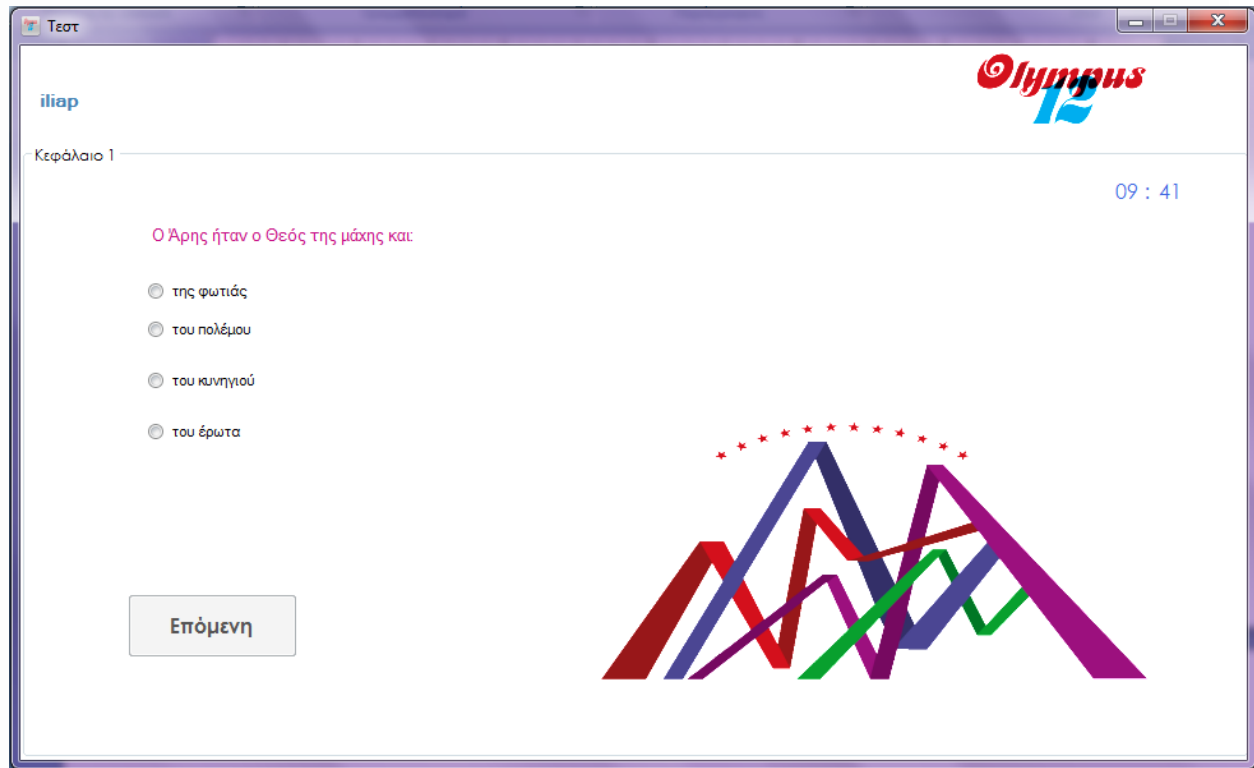
Ερώτηση  
Πολλαπλής  
Επιλογής

<p>Η Ιασώ ήταν η θεά της ίασης.</p> <p><input type="radio"/> Σωστό</p> <p><input type="radio"/> Λάθος</p>	Ερώτηση “Σωστό- Λάθος”
<p>Ο Κρόνος είχε καταπιεί τα πέντε παιδιά του, τη Δήμητρα, την Ήρα, τον Πλούτωνα, τον Ποσειδώνα και την...</p> <input type="text"/>	Ερώτηση Συμπλήρωσης Κενού

Πίνακας 4.1: Παρουσίαση Ερωτήσεων

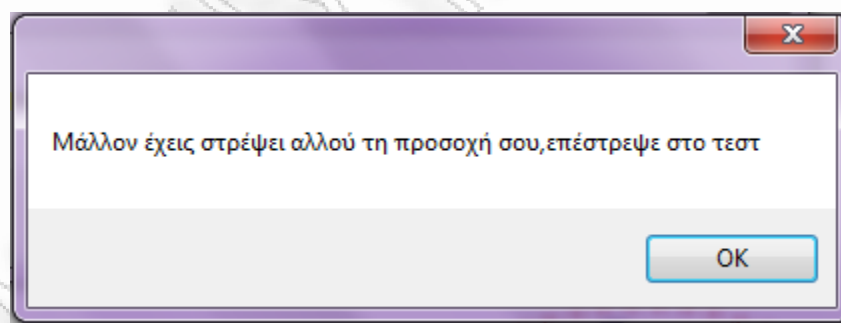


Εικόνα 4.16: Ο χρήστης δεν έχει ως επιλεγμένη γλώσσα τα ελληνικά στο πληκτρολόγιό του



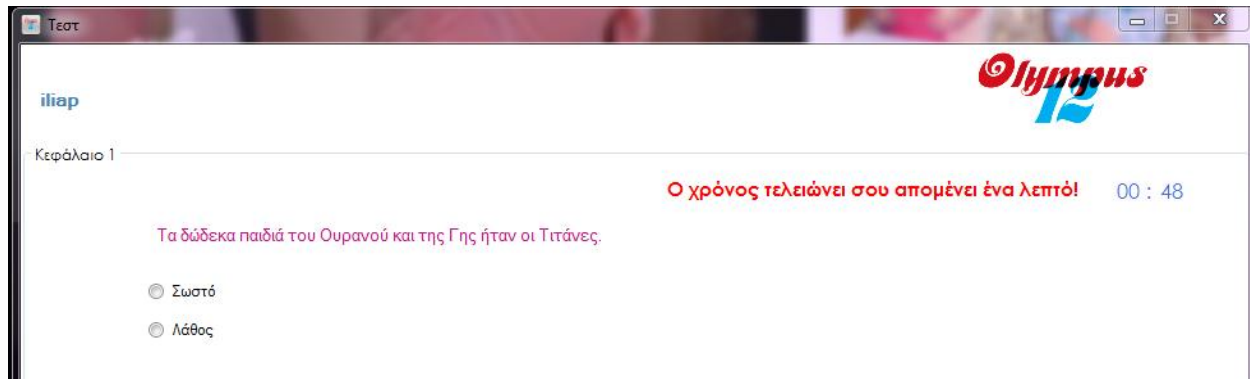
Εικόνα 4.17: Απομένουν στον χρήστη 9 λεπτά και 41 δευτερόλεπτα. Δεν έχει δώσει καμία απάντηση, το κουμπί “Επόμενη είναι απενεργοποιημένο”.

Αν ο χρήστης κατά τη διάρκεια που κάνει κάποιο τεστ έχει την προσοχή του στραμμένη αλλού εμφανίζεται ένα παράθυρο διαλόγου το οποίο τον προτρέπει να επιστρέψει στο τεστ που έχει ξεκινήσει να κάνει.



Εικόνα 4.18: Παράθυρο διαλόγου. Το σύστημα προτρέπει τον χρήστη να επιστρέψει στο τεστ





Εικόνα 4.19:Εμφάνιση λεζάντας στο πάνω μέρος της οθόνης για να τονίσει στο χρήστη ότι του απομένει ένα λεπτό.

#### 4.6 Παρουσίαση αποτελεσμάτων

Όταν ολοκληρωθεί το τεστ εμφανίζεται μία φόρμα η οποία έχει τα συγκεντρωτικά αποτελέσματα. Από τη φόρμα αυτή ο χρήστης ενημερώνεται για :

- Την βαθμολογία του
- Για το αν πέρασε το τεστ (βαθμολογία πάνω από 50%)
- Για την χρονική διάρκεια του τεστ
- Για τα νομίσματα που συγκέντρωσε
- Για τον συνολικό αριθμό νομισμάτων

Επιπλέον εμφανίζονται συμβουλές και εικονίδια που φανερώνουν την απόδοσή του. Στο τέλος της φόρμας ένα μήνυμα τον ενημερώνει για το αν πρέπει να συνεχίσει σε τεστ άλλου κεφαλαίου ή πρέπει να επιλέξει ξανά τεστ από το ίδιο κεφάλαιο.

Τα αποτελέσματά μου

iliap

Θημύρις

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: 30 %

Διάρκεια τεστ: 0:20

Σωστές απαντήσεις 3/10

Έχεις κερδίσει 10 νομίσματα

Συνολικός αριθμός νομισμάτων: 115

Χρόνος		Ολοκλήρωσες το test πολύ γρήγορα και η απόδοσή σου δεν είναι καλή. Ίσως να μην πρόσεχες τόσο τι έγραφαν οι ερωτήσεις
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!

**Εικόνα 4.20: Απεικόνιση αποτελεσμάτων**

Στην παραπάνω εικόνα παρατηρούμε ότι ο χρήστης ολοκλήρωσε το τεστ μέσα σε 20 δευτερόλεπτα που είναι πολύ γρήγορος χρόνος και είναι αδύνατο να ολοκληρωθεί ένα τεστ μέσα σε τόσα λίγα δευτερόλεπτα. Για το λόγο αυτό στο πεδίο “Χρόνος” έχει προειδοποιητική σήμανση. Η βαθμολογία που πήρε είναι 30%, δηλαδή κάτω από τη βάση, επειδή το ποσοστό δεν είναι υψηλό παίρνει απαγορευτική σήμανση όπως φαίνεται και από το πεδίο “Βαθμολογία”. Τέλος επειδή από τον έλεγχο που έκανε το πρόγραμμα ο χρήστης δεν είχε διαβάσει αρκετά παίρνει πάλι απαγορευτική σήμανση. Επειδή δεν έχει συγκεντρώσει αστέρια σε καμία από τις κατηγορίες στο κάτω μέρος της οθόνης το σύστημα τον προτρέπει να συνεχίσει σε νέο τεστ επόμενου κεφαλαίου. Επιπλέον εμφανίζεται ένα flash με τον θεό Δία ο οποίος εμφανίζεται μέσα από το άκουσμα ενός κεραυνού διότι δεν κατάφερε να περάσει το τεστ.

Όπως αναφέραμε στην προηγούμενη ενότητα οι συμβουλές εμφανίζονται με βάση την απόδοση του χρήστη αλλά και την συμπεριφορά του κατά τη διάρκεια του τεστ (αριθμός backspaces, κινήσεις ποντικιού κτλ.).

Αποτελέσματα

Συμβουλές

Η σωστή δουλεία φέρνει αποτελέσματα!!!!

Θα πρέπει όταν κάνεις το τεστ να μην ασχολήσε με κάτι άλλο

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

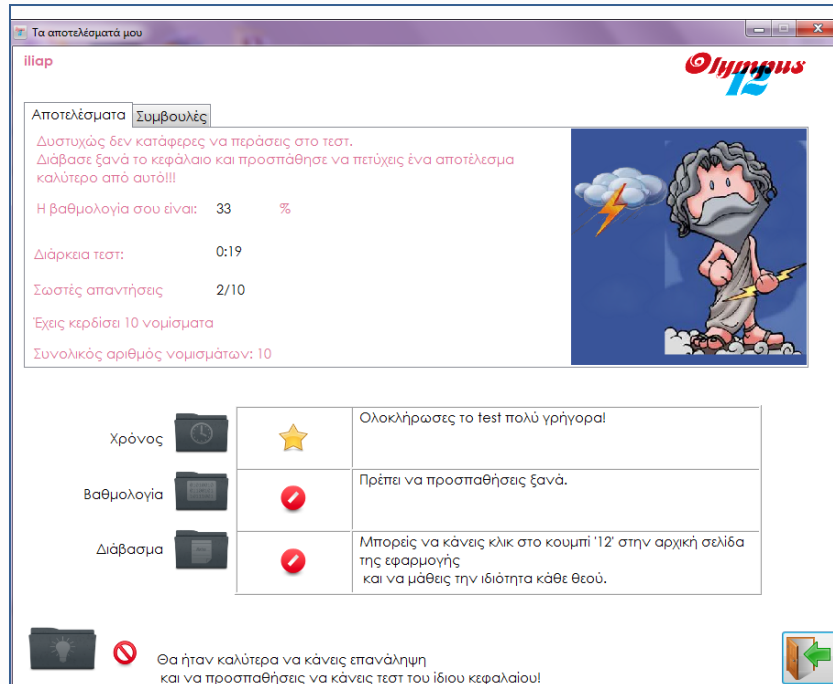
Εικόνα 4.21: Απεικόνιση διάφορων συμβουλών που αφορούν τον χρήστη.

Όταν ο χρήστης επιστέψει στην αρχική οθόνη βλέπει ότι τα νομίσματα που έχει στην κατοχή του έχουν ανανεωθεί βάση του τελευταίου τεστ που έκανε.



Εικόνα 4.22: Ανανέωση νομισμάτων

Στην εικόνα 4.22 μπορούμε να δούμε τα αποτελέσματα από την διεξαγωγή κάποιου άλλου τεστ που ο χρήστης έχει ολοκληρώσει πολύ γρήγορα το τεστ όμως δεν έχει καταφέρει να περάσει με επιτυχία. Επιπλέον δεν έχει διαβάσει αρκετή ώρα και γι αυτό εμφανίζεται το εικονίδιο που δηλώνει απαγόρευση. Ο χρήστης δεν καταφέρνει να συγκεντρώσει αστέρια και στις τρεις κατηγορίες και γι αυτό το σύστημα τον ενημερώνει να ξανακάνει το τεστ.



Εικόνα 4.23: Προσθήκη νομισμάτων στα ήδη υπάρχοντα νομίσματα

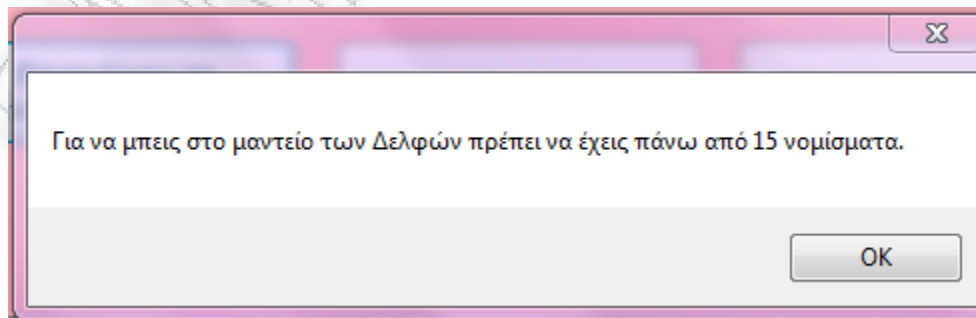
#### 4.7 Παρουσίαση του Μαντείου των Δελφών

Αν ο χρήστης έχει στην κατοχή του πάνω από 15 νομίσματα μπορεί να επισκεφθεί το μαντείο των Δελφών και να πάρει χρησμό από την Πυθία. Αν δεν έχει συγκεντρώσει 15 νομίσματα πρέπει να κάνει τεστ για να μπορέσει να τα συγκεντρώσει.

Για να γίνει η επίσκεψη ο χρήστης κάνει κλικ στο κουμπί που υπάρχει στην κεντρική οθόνη:

Επισκέψου το  
Μαντείο των Δελφών

. Αν δεν έχει τα νομίσματα που χρειάζονται ενημερώνεται από ένα παράθυρο διαλόγου ότι πρέπει να συγκεντρώσει το απαιτούμενο ποσό για να πάρει χρησμό. Όπως αναλύθηκε στην προηγούμενη ενότητα υπάρχει συγκεκριμένη διαδικασία με την οποία ο χρήστης συγκεντρώνει νομίσματα και συγκεκριμένη διαδικασία για το ποιοι χρησμοί θα εμφανιστούν.



Εικόνα 4.24: Ο χρήστης έχει λιγότερο από 15 νομίσματα



Εικόνα 4.25: Οι τρεις χρησμοί που δίνονται στον χρήστη από την Πυθία

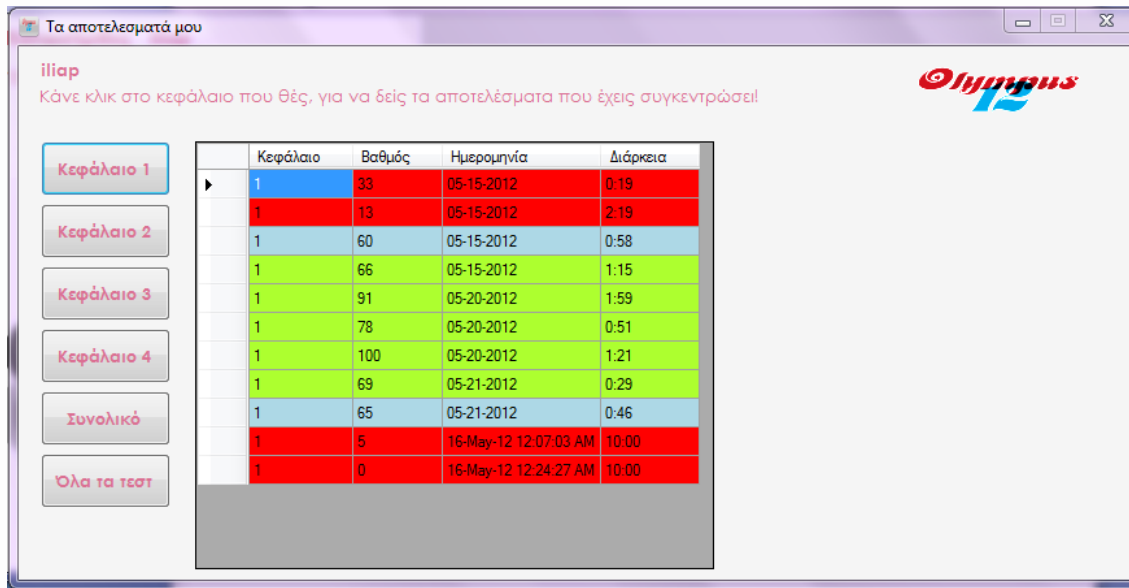
#### 4.8 Αναλυτικά στοιχεία

Η επιλογή “Αναλυτικά στοιχεία” που υπάρχει στην κεντρική φόρμα του προγράμματος εμφανίζει ένα νέο παράθυρο όπου περιέχονται όλα τα αποτελέσματα του χρήστη από τα τεστ που έχει κάνει. Στην φόρμα αυτή μπορεί να επιλέξει κεφάλαιο και να δει τα τεστ που είχε κάνει, πότε τα είχε κάνει και τι απόδοση είχε σε αυτά. Οι βαθμολογίες που είναι κάτω από 50% εμφανίζονται με κόκκινο, οι βαθμολογίες μέχρι 65% με μπλε και από 65% και πάνω με πράσινο.



Εικόνα 4.26: Κουμπί για εμφάνιση των αναλυτικών στοιχείων




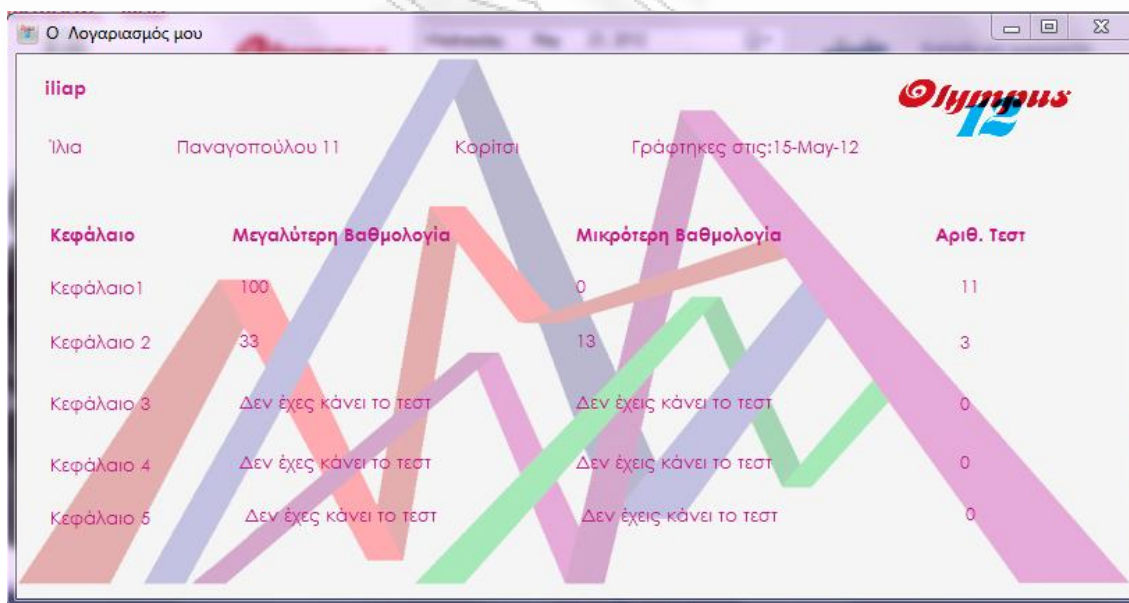


Εικόνα 4.27: Αναλυτικά στοιχεία

### 4.9 Παρουσίαση της φόρμας “Ο λογαριασμός μου ”




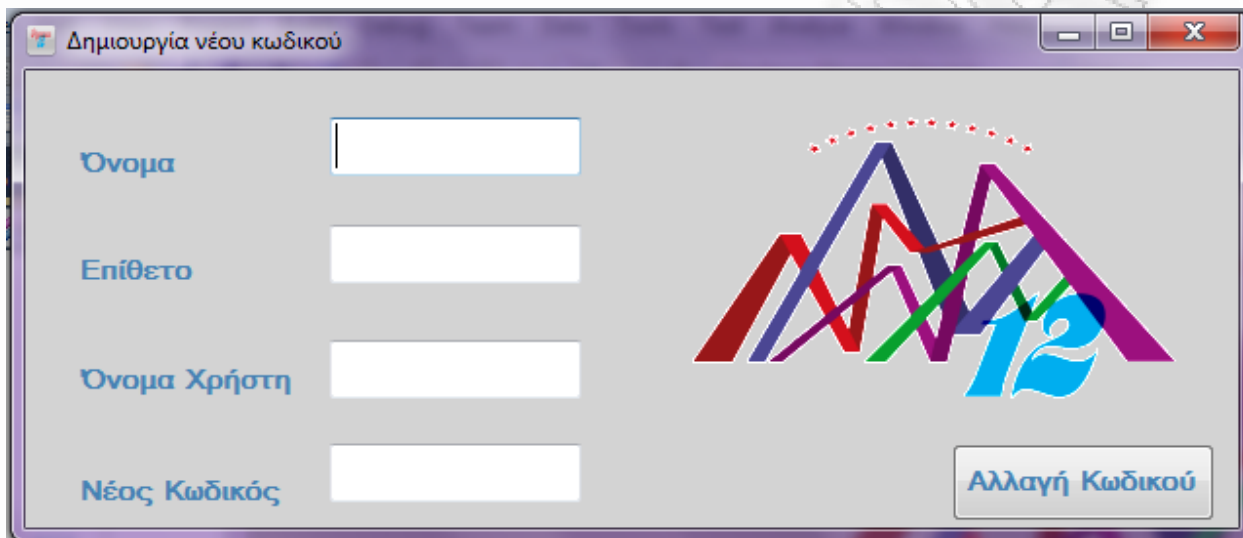
Κάνοντας κλικ στο κουμπί  εμφανίζεται ένα νέο παράθυρο όπου έχει κάποια στοιχεία του χρήστη. Στο παράθυρο αυτό εμφανίζεται το ονοματεπώνυμο του χρήστη η ηλικία που είχε δηλώσει κατά την εγγραφή του καθώς και η ημερομηνία εγγραφής του. Πέρα αυτών των στοιχείων εμφανίζεται η μεγαλύτερη και η μικρότερη βαθμολογία που έχει επιτύχει σε κάθε κεφάλαιο.



Εικόνα 4.28: Ο λογαριασμός μου

#### 4.10 Αλλαγή κωδικού πρόσβασης

Για να αλλάξει ο χρήστης κωδικό πρόσβασης πρέπει να κάνει στο κουμπί . Μπροστά του ανοίγει το παράθυρο της εικόνας 4.29 όπου πρέπει να συμπληρώσει το όνομα του το επίθετό του και το όνομα χρήστη που είχε δηλώσει κατά την εγγραφή του. Τέλος συμπληρώνει και τον νέο κωδικό που θέλει να έχει.



Δημιουργία νέου κωδικού

Όνομα

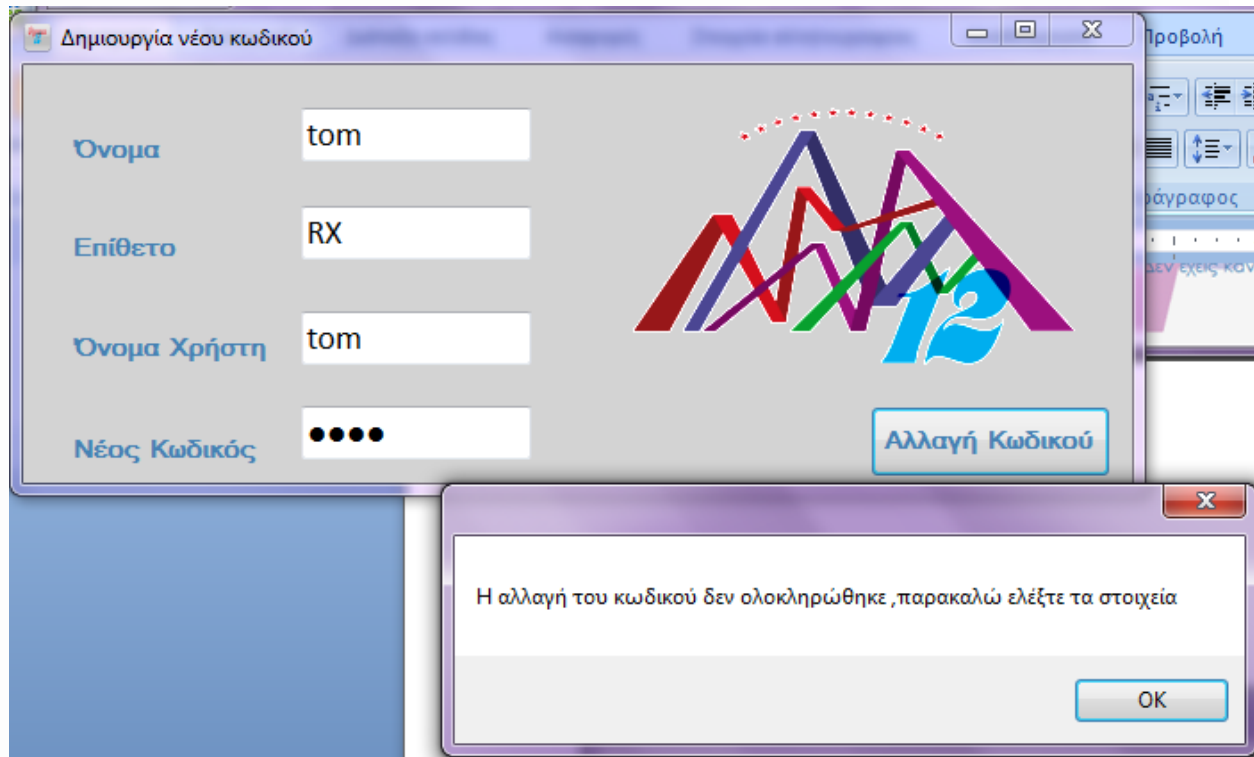
Επίθετο

Όνομα Χρήστη

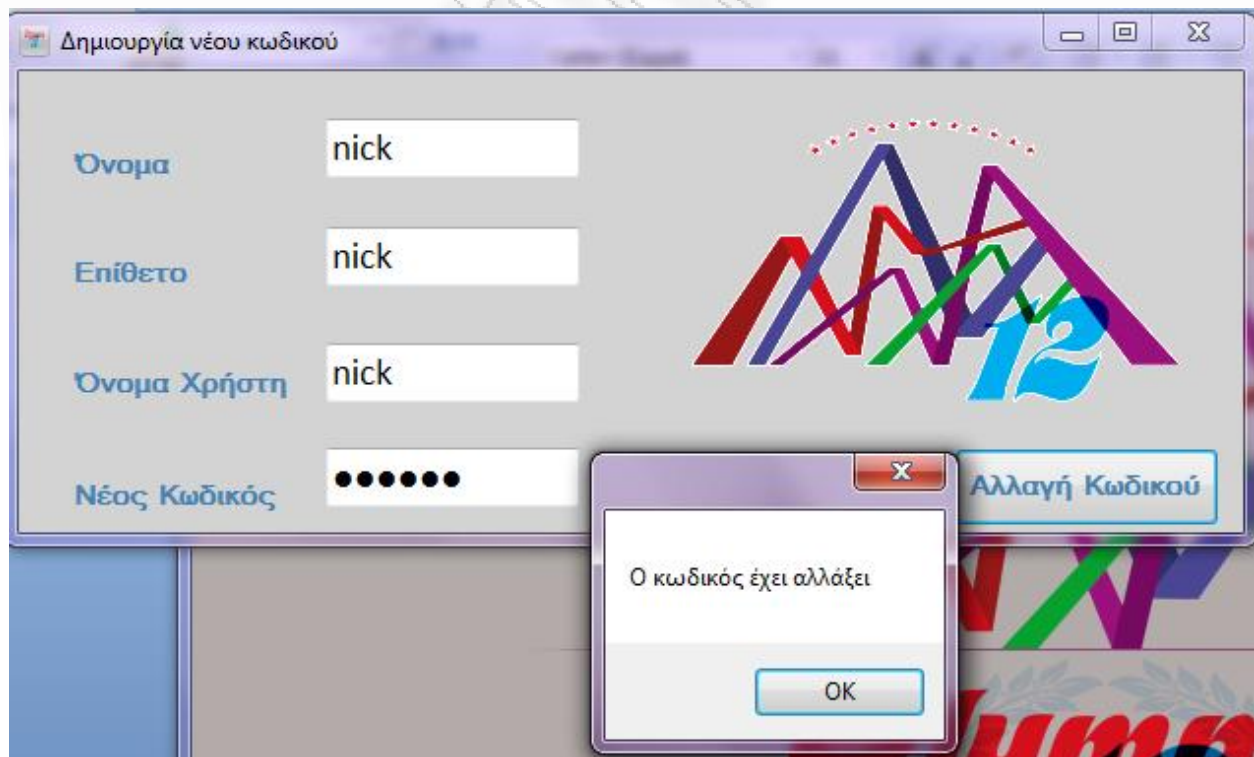
Νέος Κωδικός

Αλλαγή Κωδικού

Εικόνα 4.29: Δημιουργία νέου κωδικού



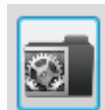
Εικόνα 4.30: Τα στοιχεία που πληκτρολόγησε ο χρήστης δεν υπάρχουν στην βάση



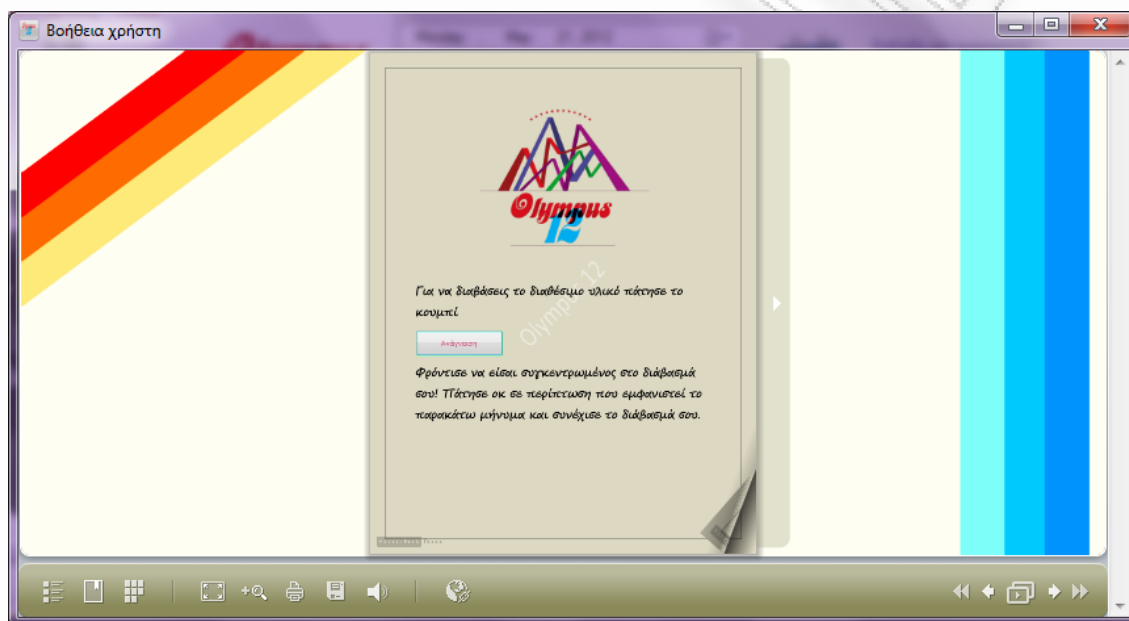


Εικόνα 4.31: Επιτυχής αλλαγή κωδικού

#### 4.11 Παρουσίαση της φόρμας “Βοήθεια”



Ο χρήστης ανά πάσα στιγμή μπορεί να πατήσει το κουμπί το οποίο βρίσκεται στην κεντρική φόρμα της εφαρμογής. Αν έχει κάποιο πρόβλημα με το χειρισμό του λογισμικού εκεί θα βρει όλες τις οδηγίες χρήσης της εφαρμογής.



Εικόνα 4.32: Παράθυρο που περιέχει τις οδηγίες για την χρήση της εφαρμογής

## Συμπεράσματα

Τα προσαρμοστικά συστήματα διδασκαλίας επηρεάζουν την αντίληψη του μαθητή και του δίνουν το έναυσμα για μάθηση. Όταν σχεδιάζουμε ένα τέτοιο μοντέλο πρέπει να έχουμε λάβει υπόψη ότι είναι ένα σύνθετο μοντέλο το οποίο αποτελείται από το μοντέλο μαθητή το παιδαγωγικό μοντέλο και το γνωστικό πεδίο.

Συστήματα τα οποία σχεδιάζονται με τις μεθόδους που αναλύθηκαν σε αυτήν την διπλωματική εργασία μπορούν να συμβάλουν θετικά στην εξέλιξη του μαθητή. Ωστόσο όταν σχεδιάζουμε ένα προσαρμοστικό σύστημα θα πρέπει να δίνεται προσοχή στο πως τα προσαρμοστικά μέσα συμβάλουν στην ευκολία χρήσης του λογισμικού και δεν επιτυγχάνουν το αντίθετο.

Όταν ένα εκπαιδευτικό λογισμικό αποτελεί μέρος ενός εκπαιδευτικού οργανισμού και όχι ένα απλό εργαλείο εκμάθησης για το σπίτι θα πρέπει να αξιολογείται συχνά από τους χρήστες (καθηγητές και μαθητές), να αναφέρονται τυχόν λάθη που μπορεί να εντοπιστούν κατά τη χρήση του και να υποβάλλονται αναβαθμίσεις και ενημερώσεις όταν κρίνεται απαραίτητο.

Το προσαρμοστικό εκπαιδευτικό λογισμικό που έχει σχεδιαστεί για την παρούσα διπλωματική εργασία αποτελεί ένα ολοκληρωμένο εργαλείο μάθησης το οποίο όμως δεν έχει σχεδιαστεί σε συνεργασία με εκπαιδευτικούς και δεν επιδιώκει στην εκμάθηση όλης της ύλης της ελληνικής μυθολογίας αλλά ένα μέρος αυτής.

## Βιβλιογραφία

1. **Frasson, Claude, Gauthier, Gilles και Lesgold, Alan.** *Intelligent Tutoring Systems*. Heidelberg : Springer Verlag, 1996.
2. **Tchounikine, Pierre.** *Computer Science and Educational Software Design*. Grenoble : Springer Verlag, 2011.
3. **Blease, Derek.** *Evaluating Educational Software*. Great Britain : Routledge.
4. **Julie A. Jacko, Constantine Stephanidis, Don Harris.** *Human-computer Interaction: Theory and Practice*. New Jersey : Routledge, 2003. Τόμ. ΙΙ.
5. **Kruse, Kevin.** *Gagne's Nine Events of Instruction*. s.l. : [http://www.e-learningguru.com/articles/art3\\_3.htm](http://www.e-learningguru.com/articles/art3_3.htm), 2008.
6. **Aleksandar Davidovic, Jim Warren, Elena Trichina.** *Learning Benefits of Structural Example-Based*. s.l. : IEEE TRANSACTIONS ON EDUCATION, 2003.
7. **BRUSILOVSKY, PETER.** *Methods and Techniques of Adaptive Hypermedia*. Netherlands : Kluwer Academic Publishers, 1996.
8. **Brusilovsky, Peter.** *Adaptive Hypermedia: From Intelligent Tutoring Systems to Web-Based Education*. Pittsburgh : Carnegie Technology Education and HCI Institute.
9. **Kavcic, Alenka.** *The Role of User Models in Adaptive Hypermedia Systems*. s.l. : Mediterranean Electrotechnical Conference, 2000.
10. **Joseph Beck, Mia Stern, Erik Haugsjaa.** Applications of AI in Education. *Crossroads, The ACM Student Magazine*.
11. **M. Virvou, V. Tsiriga.** An object-oriented software life cycle of an intelligent tutoring system. *Journal of Computer Assisted Learning*. 2001, 17.
12. **Αναστασία Β. Παπαβασιλείου, Γεωργίου Δημήτριος, Καραάκος Αλέξανδρος.** Η χρήση των χρωμάτων στην σχεδίαση εκπαιδευτικού λογισμικού.
13. **Aleksandar Davidovic, Jim Warren, and Elena Trichina.** Learning Benefits of Structural Example-Based Adaptive Tutoring Systems. *IEEE TRANSACTIONS ON EDUCATION*. 2003, 46.
14. **Maria Virvou, Maria Moundridou.** Student and Instructor Models: Two Kinds of User Model and Their Interaction in an ITS Authoring Tool.
15. **Singley, M.K.** *The effect of goal posting on operator selection*. Pittsburg : Proceedings of the Third International Conference on Artificial Intelligence and Education, 1987.

16. Wikipedia. <http://en.wikipedia.org/>. [Ηλεκτρονικό] 2012. [Παραπομπή: 5 Απρίλιος 2012.]  
[http://en.wikipedia.org/wiki/Robert\\_M.\\_Gagn%C3%A9](http://en.wikipedia.org/wiki/Robert_M._Gagn%C3%A9).

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΡΑΙΑ

