



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Εκπαιδευτικό λογισμικό στον τομέα των Μαθηματικών για παιδιά Α' και Β' τάξεως δημοτικού.
Όνοματεπώνυμο Φοιτητή	Ηλίας Αραπάκης
Πατρώνυμο	Χρήστος
Αριθμός Μητρώου	ΜΠΣΠ / 09030
Επιβλέπων	Μαρία Βίρβου, Καθηγήτρια

Ημερομηνία Παράδοσης **Οκτώβριος 2012**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

Μαρία Βίρβου
Καθηγήτρια

(υπογραφή)

Γεώργιος Τσιχριντζής
Καθηγητής

(υπογραφή)

Ευάγγελος Φούντας
Καθηγητής

Περιεχόμενα

Περίληψη.....	σελ.4
Abstract.....	σελ.4
1. Εισαγωγή.....	σελ.4
1.1 Είδη Εκπαιδευτικού Λογισμικού.....	σελ.7
1.2 Προδιαγραφές και Πρότυπα Εκπαιδευτικού Λογισμικού.....	σελ.9
1.3 Αξιολόγηση Εκπαιδευτικού Λογισμικού.....	σελ.10
1.3.1 Τύποι Αξιολόγησης.....	σελ.10
1.3.2 Μέθοδοι Αξιολόγησης.....	σελ.11
1.4 Ποιοτικός Έλεγχος Εκπαιδευτικού Λογισμικού.....	σελ.11
2. Παρούσα Κατάσταση.....	σελ.13
3. Ανάλυση και Σχεδιασμός.....	σελ.16
3.1 Κύκλος Ζωής Ανάπτυξης Λογισμικού.....	σελ.17
3.1.1 Μοντέλο Καταρράκτη.....	σελ.17
3.1.2 Μοντέλο Πρωτοτυποποίησης.....	σελ.18
3.1.3 Μοντέλο Λειτουργικής Επαύξησης.....	σελ.20
3.1.4 Εναλλακτικά Μοντέλα Ανάπτυξης Εκπαιδευτικού Λογισμικού.....	σελ.21
4. Παρουσίαση εφαρμογής.....	σελ.25
4.1 Επίπεδα χρηστών και είδη ασκήσεων.....	σελ.27
4.2 Ο αλγόριθμος.....	σελ.29
4.2.1 Είδη Λαθών.....	σελ.29
4.2.2 Πολλαπλά Είδη Λαθών.....	σελ.32
4.2.3 Διφορούμενα Λάθη.....	σελ.37
4.3 Βάση Δεδομένων.....	σελ.37
4.4 Σενάρια Εκτέλεσης.....	σελ.40
4.4.1 Σενάριο Εκτέλεσης χρήστη με ρόλο μαθητή.....	σελ.40
4.4.2 Σενάριο Εκτέλεσης χρήστη με ρόλο καθηγητή.....	σελ.62
5. Σύγκριση με άλλες εφαρμογές.....	σελ.68
6. Συμπεράσματα.....	σελ.69
7. Βιβλιογραφία.....	σελ.70
Παράρτημα Α - User Manual.....	σελ.70
Παράρτημα Β - Τμήματα Κώδικα.....	σελ.85

Περίληψη

Στην εργασία αυτή θα γίνει παρουσίαση μιας εκπαιδευτικής ιστοσελίδας στον κλάδο των μαθηματικών για παιδιά Α' και Β' τάξης δημοτικού και συγκεκριμένα για τις 4 βασικές πράξεις (πρόσθεση, αφαίρεση, πολλαπλασιασμός, διαίρεση). Αρχικά θα αναφερθούμε στην έννοια του εκπαιδευτικού λογισμικού, τα πλεονεκτήματά του σε σχέση με την παραδοσιακή διδασκαλία, τα είδη του εκπαιδευτικού λογισμικού που υπάρχουν, τα είδη και μεθόδους αξιολόγησής του και τον ποιοτικό του έλεγχο ως εισαγωγή. Στη συνέχεια θα κάνουμε μία ανασκόπηση πεδίου και το πού βρισκόμαστε στην παρούσα χρονική περίοδο. Θα γίνει αναφορά στην ανάλυση και το σχεδιασμό γενικότερα για το εκπαιδευτικό λογισμικό και ιδιαίτερα για την εφαρμογή μας. Παρακάτω περιγράφεται σε εκτενές κεφάλαιο η λειτουργία της ιστοσελίδας, η σύγκρισή της με παρόμοιες εφαρμογές που υπάρχουν και τέλος θα δούμε κάποια συμπεράσματα σχετικά με την ανάπτυξη της ιστοσελίδας.

Abstract

In this study there will be a presentation of an educational website in the field of mathematics for kids of the first and second class of primary school and specifically the 4 basic operations (addition, subtraction, multiplication, division). Initially, we will mention the definition of educational software, its advantages against the traditional teaching, the kinds of educational software that there are, the kinds and methods of its evaluation and the quality checking as introduction. Furthermore, we will make a review and we will mention to which level we have reached nowadays concerning the educational software. There will be a reference to Analysis and Design generally for the educational software and also for the website that we developed. Later, the operation of the website will be described in a later chapter, its comparison with similar applications and finally we will reach in some conclusions for the development of the website.

Εισαγωγή

Το εκπαιδευτικό λογισμικό έχει ήδη εισβάλλει στην εκπαιδευτική διαδικασία όλων των βαθμίδων αφού έκανε πριν μερικά χρόνια πρώτα την είσοδο του στην επαγγελματική κατάρτιση. Επειδή μεταξύ άλλων δεν υπάρχει ακόμη ομοφωνία στην επιστημονική κοινότητα για το τι ακριβώς είναι και πώς ορίζεται το εκπαιδευτικό λογισμικό θα ξεκινήσουμε παραθέτοντας τους διάφορους τύπους εκπαιδευτικού λογισμικού όπως αυτοί εμφανίζονται μέχρι στιγμής στην διεθνή βιβλιογραφία. Ατυχώς, η κατάσταση στον χώρο αυτό, παρά τις προσπάθειες που έχουν ήδη ξεκινήσει, δεν είναι ακόμα ώριμη. Υπάρχουν βέβαια κάποιες κατευθυντήριες γραμμές που μπορούν να οριοθετήσουν κάποιες γενικές απαιτήσεις και προδιαγραφές και οι οποίες μπορούν να οδηγήσουν σε ποιοτικό εκπαιδευτικό λογισμικό, αρκεί να ενταχθούν σε ένα συστηματικό πλαίσιο ανάπτυξης και ελέγχου της ποιότητας τον λογισμικού παράλληλα με την φάση της ανάπτυξης του. Στη συνέχεια εξετάζουμε σύντομα τους διάφορους τύπους και μεθόδους αξιολόγησης εκπαιδευτικού λογισμικού για να προχωρήσουμε στην τεκμηρίωση της ανάγκης του ποιοτικού ελέγχου του. Ορίζουμε την ποιότητα, τις μετρικές και άλλα χαρακτηριστικά όπως

αυτά έχουν μελετηθεί στα πλαίσια της Τεχνολογίας Λογισμικού και για τα οποία έχουμε στη διάθεση μας κάποια γενικά στάνταρντ.

Αποτελεί κοινή διαπίστωση ότι αρκετοί μαθητές αντιμετωπίζουν δυσκολίες σε πολλά από τα μαθήματα που διδάσκονται. Για αυτό το λόγο η έρευνα στον τομέα της Διδακτικής και Μεθοδολογίας ασχολείται σε μεγάλο βαθμό τόσο με την καταγραφή αυτών των προβλημάτων όσο και με την ερμηνεία και την επίλυσή τους. Η δημιουργία κατάλληλων συνθηκών για την προώθηση της αποτελεσματικής μάθησης αποτελεί άλλωστε τον απώτερο σκοπό της Διδακτικής και Μεθοδολογίας των διαφόρων επιστημών. Οι ερευνητές σε αυτό τον τομέα προσπαθούν να καθορίσουν το αντικείμενο διδασκαλίας και τον τρόπο μετάδοσης της γνώσης, αναπτύσσοντας πρότυπα και μοντέλα αποτελεσματικών μεθόδων διδασκαλίας. Τα τελευταία χρόνια αυτές οι προσπάθειες έχουν επεκταθεί και στο χώρο της εκπαιδευτικής τεχνολογίας με κύριο αποτέλεσμα το σχεδιασμό και την υλοποίηση εκπαιδευτικού λογισμικού. Γενικά, ως εκπαιδευτικό λογισμικό ορίζεται το προϊόν της τεχνολογίας με το οποίο προσπαθούμε να διδάξουμε ένα γνωστικό αντικείμενο υλοποιώντας συγκεκριμένη παιδαγωγική φιλοσοφία και συγκεκριμένη εκπαιδευτική στρατηγική. Τα μαθησιακά περιβάλλοντα και εργαλεία που έχουν αναπτυχθεί στο πλαίσιο αυτής της προσπάθειας και φέρουν τον τίτλο εκπαιδευτικό λογισμικό είναι πολλά αλλά και διαφορετικά μεταξύ τους. Κυρίως διαφέρουν ως προς τα επί μέρους χαρακτηριστικά τους, αλλά και ως προς τη φιλοσοφία σχεδιασμού τους και τη διδακτική προσέγγιση που χρησιμοποιούν. Δεδομένου ότι η έρευνα για την εκπαιδευτική τεχνολογία και τα αποτελέσματά της ενσωμάτωσής της στην εκπαιδευτική διαδικασία είναι πολύ πρόσφατη, δεν είναι δυνατόν να καθοριστούν απόλυτα τα χαρακτηριστικά ενός προγράμματος ώστε αυτό να θεωρείται εκπαιδευτικό λογισμικό. Πολλά προγράμματα θεωρούνται λογισμικά γιατί χρησιμοποιούν κάποιο κώδικα για την εκτέλεσή τους. Επίσης, μπορούν να χαρακτηριστούν και ως εκπαιδευτικά επειδή θέτουν συγκεκριμένους μαθησιακούς στόχους και χρησιμοποιούν μία ή περισσότερες διδακτικές μεθόδους για να τους επιτύχουν. Αρκεί όμως αυτό για να χαρακτηριστούν εκπαιδευτικά λογισμικά; Για πολλούς εκπαιδευτικό είναι εκείνο το λογισμικό που μπορεί να υποκαταστήσει το δάσκαλο, ή ακόμα καλύτερα να τον αντικαταστήσει ή τέλος να διευρύνει την έννοια δάσκαλος παρέχοντας συγχρόνως υπηρεσίες, πληροφορίες και διεξόδους που ένα μεμονωμένο άτομο δεν μπορεί να παρέχει. Για άλλους εκπαιδευτικό είναι εκείνο το λογισμικό που ανταποκρίνεται στις γνωστικές δομές και ιδιαιτερότητες του μαθητή, υλοποιεί παιδαγωγικές και κοινωνικές αρχές και τον βοηθά να οικοδομήσει τη γνώση σύμφωνα με τις ατομικές του ανάγκες σε ένα μαθησιακό περιβάλλον όπου ο εκπαιδευτικός τον καθοδηγεί και τον διευκολύνει στην πορεία του προς τη γνώση. Για να ενημερωθούν οι εκπαιδευτικοί σχετικά με το εκπαιδευτικό λογισμικό γίνεται παρακάτω μια προσπάθεια προσέγγισης των εξής θεμάτων:

- Διατύπωση ορισμού για το εκπαιδευτικό λογισμικό.
- Ταξινόμηση / κατηγοριοποίηση εκπαιδευτικού λογισμικού
- Περιγραφή των χαρακτηριστικών ανά είδος εκπαιδευτικού λογισμικού.
- Παιδαγωγική αξιοποίηση εκπαιδευτικού λογισμικού και ο ρόλος του εκπαιδευτικού.
- Η εισαγωγή των Τεχνολογιών της Πληροφορίας και της Επικοινωνίας στην εκπαίδευση και οι στάσεις των εκπαιδευτικών.

Η χρήση του εκπαιδευτικού λογισμικού στη γενικότερή του μορφή παρουσιάζει τα εξής πλεονεκτήματα σε σχέση με τις παραδοσιακές μορφές διδασκαλίας:

1. Δίνει τη δυνατότητα εξατομικευμένης διδασκαλίας. Ο κάθε μαθητής προχωράει ακολουθώντας το δικό του ρυθμό. Ο ηλεκτρονικός υπολογιστής του εξηγεί τη θεωρία α) με την παράθεση πληροφοριών, κανόνων και εικόνων, β) με προσομοίωση πειραμάτων, χημικών αντιδράσεων και μορίων, μαχών, τρόπου λειτουργίας, χαρτών κ.ο.κ. Στη συνέχεια ελέγχει αν αυτή η θεωρία έχει εμπεδωθεί με την παράθεση ερωτήσεων και ασκήσεων. Αν ο μαθητής έχει κατανοήσει το «μάθημά του» μπορεί να συνεχίσει στο επόμενο κεφάλαιο. Αν ο μαθητής έχει κενά τότε το εκπαιδευτικό λογισμικό εξηγεί πάλι το θέμα και αξιολογεί την προσπάθεια του μαθητή. (Σε ορισμένα προγράμματα υπάρχει και επιβράβευση του μαθητή. Αν απαντήσει σωστά, τότε κερδίζει κάποια εικόνα ή βλέπει Video ή του επιτρέπεται να παίξει για συγκεκριμένο χρόνο κάποιο παιχνίδι.)
2. Δίνει τη δυνατότητα διαφοροποίησης του χρόνου διδασκαλίας και του χρόνου μελέτης – μάθησης. Το αντικείμενο μπορεί να διδαχθεί σε μια δεδομένη χρονική στιγμή και η εμπέδωσή του να γίνει μερικές ώρες αργότερα.
3. Έχει τη δυνατότητα άμεσης και αμφίδρομης επικοινωνίας μεταξύ διδάσκοντος και διδασκόμενων. Μόλις ο Η/Υ (δάσκαλος) εντοπίσει λάθος αμέσως δίνει σήμα και ο μαθητής πρέπει να προσπαθήσει πάλι. Αν ο μαθητής δε μπορεί να εντοπίσει το λάθος του, τότε ο Η/Υ κάνει τη διόρθωση. Η διόρθωση γίνεται την ώρα που συμβαίνει το λάθος και όχι όπως γίνεται στις παραδοσιακές μεθόδους διδασκαλίας, μετά την πάροδο κάποιας χρονικής περιόδου, όταν δηλαδή ο δάσκαλος θα πάρει το γραπτό του μαθητή για να το διορθώσει και να του το επιστρέψει. Πολύ συχνά τα γραπτά διορθώνονται σπίτι και την άλλη μέρα που επιστρέφονται απλά τοποθετούνται στην τσάντα χωρίς κανένα σχολιασμό των λαθών. Με το δάσκαλο ηλεκτρονικό – υπολογιστή ο μαθητής μαθαίνει από το λάθος του, αφού παραπέμπεται στους σχετικούς κανόνες, ενώ στις παραδοσιακές μεθόδους γυρίζει στο διορθωμένο γραπτό μόνο και μόνο για να δει το βαθμό που πήρε.
4. Επιτρέπει στο μαθητή να αυτενεργήσει, ώστε να ανακαλύψει το αντικείμενο της μελέτης του. Ο μαθητής μπορεί να προσπαθήσει όσες φορές θέλει να βρει τη σωστή απάντηση ή λύση ή και διαδικασία, αφού στη διδασκαλία με ηλεκτρονικό υπολογιστή δεν τίθενται χρονικά όρια μέσα στα οποία πρέπει να ολοκληρωθεί η παράδοση και η εμπέδωση του αντικειμένου. Αντίθετα, στο παραδοσιακό σχολείο έχουμε συγκεκριμένο αριθμό ωρών για κάθε μάθημα, που είναι ποσοτικά ο ίδιος για όλους τους μαθητές και ανεξάρτητος από την προσωπική τους ικανότητα.
5. Με τη χρήση της τηλεματικής παρέχει τη δυνατότητα διαφοροποίησης του τόπου όπου βρίσκεται ο καθένας από τους μαθητές του. Με αυτή την τεχνική ο δάσκαλος ή ο καθηγητής αντί να βρίσκεται στην τάξη του βρίσκεται στο σπίτι του, που μπορεί να είναι στη γειτονική πόλη ή και χώρα. Οι μαθητές είναι συνδεδεμένοι σε ένα WAN (Wide Area Network – Δίκτυο Ευρείας Περιοχής) και είναι σε θέση να δέχονται και να στέλνουν πληροφορίες στους υπόλοιπους μαθητές και στο δάσκαλο. Μπορούμε να δούμε το τεράστιο κέρδος από αυτή την εφαρμογή στα ελληνικά σχολεία. Αντί να έχουμε μονοθέσια και γενικότερα ολιγοθέσια σχολεία, όπου ένας δάσκαλος διδάσκει σε όλες τις τάξεις όλα τα μαθήματα και άρα δε μπορεί να παρακολουθεί τη δουλειά όλων των μαθητών του αποτελεσματικά, θα μπορούσαμε να έχουμε ηλεκτρονικούς υπολογιστές και το δάσκαλο να παρακολουθεί την πρόοδο των μαθητών του με τη βοήθεια του μηχανήματος. Με τη μέθοδο αυτή αφιερώνοντας τον ίδιο χρόνο προσφέρει στους μαθητές του περισσότερα.

6. Η τηλεματική μας δίνει τη δυνατότητα σχηματισμού εξειδικευμένων διδακτικών ομάδων (τάξεων) η ύπαρξη των οποίων θα ήταν για οικονομικούς ή φυσικούς λόγους αδύνατη (π.χ. διδασκαλία ξένων γλωσσών ή γλωσσών προγραμματισμού σε ολιγοθέσια σχολεία).
7. Εξασφαλίζει ίσες ευκαιρίες σε όλους τους μαθητές, αφού όλοι μπορούν να έχουν πρόσβαση στον καλύτερο δάσκαλο, στην οργανωμένη βιβλιοθήκη, στο πειραματικό εργαστήριο (προσομοιώσεις), ενώ το υπάρχον εκπαιδευτικό σύστημα ευνοούσε τους μαθητές των μεγάλων αστικών κέντρων που είχαν περισσότερες δυνατότητες.
8. Επιτρέπει την προσομοίωση πειραμάτων που είναι πρακτικά δύσκολο ή και φυσικώς αδύνατο να γίνουν πράξη. Στον Η/Υ μπορούμε να προσομοιώσουμε την περιστροφή των πλανητών, πειράματα μηχανικής (π.χ. τριβών όπου στην πράξη είναι πάρα πολύ δύσκολο να δούμε με ποια ακριβώς δύναμη το σώμα αρχίζει να ολισθαίνει ενώ στην προσομοίωση έχουμε τα διάφορα όργανα που μας δίνουν τις μετρήσεις όταν αλλάξουμε τις δυνάμεις). Μπορούμε να μεταβάλουμε παραμέτρους όπως η βαρύτητα και να παρακολουθήσουμε την εξέλιξη του πειράματός μας. Με τον Η/Υ μπορούμε να ενεργοποιήσουμε αλυσιδωτές αντιδράσεις. (π.χ. αν η πίεση φτάσει σε μία την τιμή και η θερμοκρασία σε μία άλλη τότε θα ανοίξει κάποια βαλβίδα και.....).

1.1 Είδη Εκπαιδευτικού Λογισμικού

Το λογισμικό το οποίο διατίθεται για χρήση στην εκπαίδευση μπορεί να διακριθεί σε δύο βασικές κατηγορίες [1]: *Λογισμικό Γενικού Σκοπού* το οποίο χρησιμοποιείται σαν εργαλείο διδασκαλίας με την έννοια της επέκτασης των εποπτικών μέσων (είτε εντάσσεται στα πλαίσια της εισαγωγής της πληροφορικής στα σχολεία), και σε *Εκπαιδευτικό Λογισμικό* το οποίο έχει σχεδιασθεί ειδικά για να ικανοποιήσει παιδαγωγικούς, διδακτικούς και μαθησιακούς στόχους. Η πρώτη κατηγορία περιλαμβάνει λογισμικό όπως Word, Excel, PowerPoint και Access το οποίο δεν το θεωρούμε εκπαιδευτικό λογισμικό και για αυτό το λόγο δεν θα αναφερόμαστε σε τέτοιο λογισμικό στην συνέχεια αυτής της εργασίας. Η δεύτερη κατηγορία είναι αυτή η οποία μας ενδιαφέρει και με την οποία θα ασχοληθούμε στην συνέχεια. Τυπικό παράδειγμα εκπαιδευτικού λογισμικού μπορεί να θεωρηθεί το λογισμικό που αναπτύσσεται για λογαριασμό του ΥΠΕΠΘ στα πλαίσια των έργων Σειρήνες, Ναυσικά κλπ. Στη βιβλιογραφία το εκπαιδευτικό λογισμικό γενικά διαχωρίζεται στις εξής κατηγορίες ή τύπους[1]: tutorial, drill and practice, simulation, problem solving, και educational games. Κάποιο λογισμικό μπορεί φυσικά να εντάσσεται σε περισσότερες από μία κατηγορίες ενώ κάποιο άλλο, οι εγκυκλοπαίδειες για παράδειγμα, δεν εντάσσεται σε καμία κατηγορία.

Tutorial (Παιδαγωγικά)

Αυτού του τύπου το λογισμικό κατά κανόνα ομοιάζει περισσότερο με ηλεκτρονικό βιβλίο το οποίο παρουσιάζει νέες ιδέες και επιδεξιότητες μέσω υπερμεσικής πληροφορίας, ερωτήσεων και προβλημάτων. Κατά κανόνα επαναλαμβάνει τον κύκλο πληροφορία, ερώτηση, ανάδραση.

Στην ιδανική περίπτωση ένα tutorial λογισμικό θα πρέπει να προσαρμόζει το διδακτικό υλικό στις ιδιαίτερες ανάγκες και ικανότητες του μαθητή. Κάτι τέτοιο υποδηλώνει την παρουσία μέσα στο λογισμικό μοντέλων μαθητή, μοντέλων διδακτικών στρατηγικών, και παραπέμπει σε ιδιαίτερη κατηγορία εκπαιδευτικού λογισμικού το οποίο αναφέρεται σαν Έμπειρο Εκπαιδευτικό Λογισμικό (Intelligent Tutoring System - ITS). Δεν θεωρούμε σωστό ότι σε ένα tutorial λογισμικό η πληροφορία πρέπει, να παρουσιάζεται σε σειριακή μορφή. Αυτό ήταν μάλλον αποτέλεσμα των τεχνολογικών περιορισμών των πρώτων πολυμεσικών εφαρμογών και της μορφής που είχαν σαν ηλεκτρονικό βιβλίο.

Drill and Practice (Πρακτική και Εξάσκηση)

Ένα καλό λογισμικό αυτού του τύπου πρέπει να προσφέρει στον μαθητή απεριόριστη πρακτική άσκηση, να παρέχει συνεχή ανάδραση, να εξηγεί πώς να βρεθεί η σωστή απάντηση σε ένα πρόβλημα, και να περιλαμβάνει και ένα υποσύστημα παρακολούθησης της προόδου του μαθητή. Πρέπει να προσδιορίζει τις προαπαιτούμενες γνώσεις, να προσαρμόζει τις ασκήσεις και τις επεξηγήσεις του στις ανάγκες του κάθε μαθητή βασιζόμενο στην ανταπόκριση του μαθητή, και να παρέχει εύχρηστο σύστημα λεξιλογίου και βιβλιογραφικής αναφοράς. Η ιδανική περίπτωση και πάλι παραπέμπει σε Έμπειρο Εκπαιδευτικό Λογισμικό.

Simulation (Προσομοίωση)

Αυτού του τύπου το λογισμικό χρησιμοποιείται για την προσομοίωση εκπαιδευτικών δραστηριοτήτων οι οποίες χαρακτηρίζονται από κάποιο βαθμό επικινδυνότητας (π.χ. πειράματα Χημείας), έχουν υψηλό κόστος, είναι δύσκολα ή μη πρακτικά για κάποιο λόγο. Έτσι, το λογισμικό δημιουργεί μια κατά το δυνατόν ρεαλιστική προσομοίωση ενός πραγματικού συστήματος ή φαινομένου με ασφαλή, φθηνό και αποτελεσματικό τρόπο μέσω του οποίου ο μαθητής αποκτά εμπειρία και γνώση.

Problem Solving (Επίλυση Προβλημάτων)

Το λογισμικό αυτού του τύπου παρέχει ένα περιβάλλον μέσω του οποίου βοηθά τον μαθητή να βελτιώσει τις ικανότητες του στην επίλυση προβλημάτων. Το περιβάλλον μπορεί να περιέχει ή και να μην περιέχει κάποια προσομοίωση ενός φαινομένου του πραγματικού κόσμου. Το λογισμικό πρέπει να δίνει στον μαθητή τη δυνατότητα να δημιουργήσει ή να αναλύσει παραλλαγές του προβλήματος μέσω αλλαγών των δεδομένων του προβλήματος, να περιλαμβάνει επεξηγηματικές γραφικές απεικονίσεις των δραστηριοτήτων του μαθητή στην προσπάθειά του για επίλυση του προβλήματος, να τον υποστηρίζει στην κατανόηση αλγοριθμικών μεθόδων και να τον αποθαρρύνει από προσεγγίσεις τύπου trial-and-error.

Educational Games (Εκπαιδευτικά παιχνίδια)

Το λογισμικό του τύπου εκπαιδευτικό παιχνίδι εκμεταλλεύεται τον ενθουσιασμό, την υποκίνηση, και την προσοχή του μαθητή στο παιχνίδι για να του μεταφέρει γνώση, εμπειρία και ικανότητες. Προσφέρεται ιδιαίτερα για συνεργατική μάθηση. Στοιχεία εκπαιδευτικού παιχνιδιού είναι σημαντικό να υπάρχουν και στους υπόλοιπους τύπους εκπαιδευτικού λογισμικού. Είναι σημαντικό το εκπαιδευτικό παιχνίδι να ενθαρρύνει την επίτευξη των εκπαιδευτικών στόχων και να αποθαρρύνει κάθε άλλη χρήση που συνήθως σχετίζεται με τα ειδικά πολυμεσικά εφέ του παιχνιδιού.

1.2 Προδιαγραφές και Πρότυπα Εκπαιδευτικού Λογισμικού

Σαν εκπαιδευτικό λογισμικό χαρακτηρίζουμε το λογισμικό για εκπαίδευση μέσω υπολογιστή το οποίο ικανοποιεί πλήρως τις διδακτικές, παιδαγωγικές, γνωστικές και τεχνολογικές απαιτήσεις για τις οποίες σχεδιάστηκε και υλοποιήθηκε. Αν και υπάρχουν και άλλοι ορισμοί [4], ο παραπάνω θεωρούμε ότι είναι ο πληρέστερος. Η εκπαίδευση μέσω υπολογιστή, και κατά επέκταση και το εκπαιδευτικό λογισμικό, περιλαμβάνει και όλες τις σύγχρονες τεχνολογίες πληροφορικής, όπως είναι το διαδίκτυο, η εικονική πραγματικότητα, και οι νεώτερες εξελίξεις της τεχνητής νοημοσύνης. Υπάρχουν διεθνείς οργανισμοί όπως το IEEE LTSC (Learning Technology Standards Committee), το οποίο έχει δημιουργήσει ομάδες εργασίας μερικές από τις οποίες έχουν προχωρήσει σημαντικά στον προσδιορισμό στάνταρτ. Τέτοιες ομάδες, για παράδειγμα είναι η ομάδα για μοντέλο μαθητή, η ομάδα αρχιτεκτονικού μοντέλου, η ομάδα εργαλείων, πρακτόρων και επικοινωνίας κλπ. Στο μεσοδιάστημα, και αναφορικά με τον Ελλαδικό χώρο, υπάρχουν κάποιες εξελίξεις γύρω από το εκπαιδευτικό λογισμικό, μέσω των οποίων έχει καταγραφεί ένας σημαντικός αριθμός απαιτήσεων εκπαιδευτικού λογισμικού που είναι άμεσα χρησιμοποιούμενες. Το ΥΠΕΠΘ μέσω του προγράμματος ΟΔΥΣΣΕΙΑ, και σε συνεργασία με το Ινστιτούτο Τεχνολογίας Υπολογιστών και το Παιδαγωγικό Ινστιτούτο, έχει καθορίσει απαιτήσεις για ανάπτυξη εκπαιδευτικού λογισμικού. Οι απαιτήσεις αυτές διαχωρίζονται σε τεχνολογικές, παιδαγωγικές, εκπαιδευτικές (instructional), γνωστικού περιεχομένου, διαπροσωπείας χρήστη-υπολογιστή κλπ. Ακόμη, το Παιδαγωγικό Ινστιτούτο έχει ξεκινήσει την υπηρεσία πιστοποίησης εκπαιδευτικού λογισμικού στην οποία γίνεται ο ποιοτικός έλεγχος του λογισμικού που πρόκειται να διατεθεί για σχολική χρήση. Ο έλεγχος αυτός γίνεται με βάση αναλυτικά και εκτεταμένα ερωτηματολόγια τα οποία επεξεργάζεται ομάδα ειδικών και τα οποία στην ουσία είναι ένα μεγάλο σύνολο απαιτήσεων που εφόσον τα ικανοποιεί το ελεγχόμενο λογισμικό θα του δίνει, κατά κάποιο τρόπο, πιστοποιητικό ποιοτικού ελέγχου. Μια άλλη κατηγορία λογισμικού είναι το λογισμικό για σύγχρονη και ασύγχρονη ανοικτή και εξ αποστάσεως εκπαίδευση (ODL-Open and Distance Learning). Αν και αυτού του τύπου η εκπαίδευση (κατά κανόνα τριτοβάθμια ή κατάρτισης) είναι στην διεθνή πρακτική για πολλά χρόνια, ενώ πρόσφατα χρησιμοποιείται και στις άλλες βαθμίδες. Εκεί οι απαιτήσεις και τα στάνταρτ είναι διαφορετικά και κυρίως αναφέρονται στον τρόπο συγγραφής του διδακτικού υλικού. Στον Ελλαδικό χώρο το Ελληνικό Ανοικτό Πανεπιστήμιο (ΕΑΠ) χρησιμοποιεί προδιαγραφές οι οποίες αναφέρονται στον τρόπο συγγραφής του διδακτικού υλικού σε μορφή κατάλληλη για ανοικτή και εξ αποστάσεως εκπαίδευση.

1.3 Αξιολόγηση Εκπαιδευτικού Λογισμικού

Ο όρος "αξιολόγηση" Εκπαιδευτικού Λογισμικού υποδηλώνει την εξαγωγή ενός συμπεράσματος για το Εκπαιδευτικό Λογισμικό. Ένας καλύτερος ορισμός θα μπορούσε να αναφέρεται στην "ανακάλυψη του πώς λειτουργεί ένα εκπαιδευτικό λογισμικό" μέσω παρατήρησης και μετρήσεων της διδακτικής και μαθησιακής διαδικασίας. Όπως θα δούμε στη συνέχεια και ο ορισμός αυτός, όπως και άλλοι που συναντώνται στην βιβλιογραφία, δεν μας βοηθάει στο να σχεδιάσουμε και να υλοποιήσουμε ένα σωστό εκπαιδευτικό λογισμικό. Ένα εκπαιδευτικό λογισμικό δεν μπορεί να αξιολογηθεί σαν ένα αυτόνομο πακέτο λογισμικού. Η αξιολόγηση του πρέπει να γίνεται σε σχέση με κάποιο υπάρχον ή στοχευόμενο εκπαιδευτικό περιβάλλον μέσα στο οποίο το λογισμικό θα λειτουργεί.

1.3.1 Τύποι Αξιολόγησης

Έχουν αναφερθεί διάφοροι τύποι αξιολόγησης οι οποίοι συνήθως κατατάσσονται στις εξής τέσσερις κατηγορίες [1], 1) predictive, 2) interpretative, 3) formative και 4) summative.

Η Predictive προσέγγιση ενδιαφέρεται να αξιολογήσει την ποιότητα του λογισμικού και τις δυνατότητες χρήσης του μέσα στην τάξη πριν δοθεί για χρήση στους μαθητές. Αυτός ο τύπος αξιολόγησης γίνεται κατά κανόνα από τους εκπαιδευτικούς ή από οργανισμούς που είναι υπεύθυνοι για αγορές εκπαιδευτικού λογισμικού για τα σχολεία. Η Predictive αξιολόγηση μπορεί να χρησιμοποιηθεί και σε συνδυασμό με την διαδικασία ανάπτυξης εκπαιδευτικού λογισμικού όταν η ανάπτυξη αυτή γίνεται μέσω κάποιου αυξητικού μοντέλου ή κάποιου μοντέλου πρωτοτυποποίησης. Αν και στην βιβλιογραφία θεωρείται ότι η Predictive αξιολόγηση δεν συνδέεται με το διδακτικό περιεχόμενο, αλλά αναφέρεται στα τεχνικά χαρακτηριστικά του λογισμικού, δεν συμφωνούμε με αυτό διότι απλά έχει να κάνει με το κατά πόσον προβλέπονται ερωτήσεις επί του περιεχομένου στις οποίες απαντά ο αξιολογητής εκπαιδευτικός ή όχι (εφόσον για παράδειγμα η αξιολόγηση γίνεται μέσω λίστας ερωτήσεων).

Η Interpretative αξιολόγηση ενδιαφέρεται για την αξιολόγηση της παρατηρούμενης χρήσης του λογισμικού από τους μαθητές. Η αξιολόγηση αυτού του τύπου συνδέεται άμεσα με το διδακτικό περιεχόμενο (γνωστικό αντικείμενο) του υπό αξιολόγηση εκπαιδευτικού λογισμικού. Και η Interpretative αξιολόγηση μπορεί να χρησιμοποιηθεί κατά την διαδικασία ανάπτυξης εκπαιδευτικού λογισμικού που αναφέρθηκε παραπάνω.

Η Formative αξιολόγηση ενδιαφέρεται να παρακολουθεί την υλοποίηση των τεχνολογικών, διδακτικών, παιδαγωγικών και άλλων στόχων (αλλά όχι των αντίστοιχων απαιτήσεων και προδιαγραφών) ενός εκπαιδευτικού λογισμικού κατά την διάρκεια των φάσεων υλοποίησης του. Δεν στοχεύει στο να μετρήσει την αποτελεσματικότητα του αναπτυσσόμενου εκπαιδευτικού λογισμικού αλλά να εντοπίσει τις απαραίτητες διορθωτικές παρεμβάσεις ώστε το λογισμικό να μην αποκλίνει από τους αρχικούς του στόχους.

Η Summative αξιολόγηση ενδιαφέρεται για την αποτελεσματικότητα υλοποίησης των αρχικών στόχων μετά την υλοποίηση του εκπαιδευτικού λογισμικού. Τόσο η Formative όσο και η Summative αξιολόγηση μπορούν να χρησιμοποιηθούν όχι μόνο για την παρακολούθηση της υλοποίησης και την επιβεβαίωση της υλοποίησης αντίστοιχα των αρχικών στόχων, αλλά και των συνολικών απαιτήσεων του εκπαιδευτικού λογισμικού. Οι δύο αυτοί τύποι αξιολόγησης κάνουν κατά κανόνα χρήση των ίδιων τεχνικών αξιολόγησης. Η Formative αξιολόγηση επομένως χρησιμοποιείται να πιστοποιήσει ότι το εκπαιδευτικό λογισμικό και το αντίστοιχο υλικό του ικανοποιούν τις ανάγκες του χρήστη (μαθητή, καθηγητή, προγράμματος σπουδών ή όλων αυτών). Αντίστοιχα, η Summative αξιολόγηση μπορεί να χρησιμοποιηθεί για να ελεγχθεί η

καταλληλότητα του εκπαιδευτικού λογισμικού για συγκεκριμένες λειτουργίες (εξομοιώσεις, παιχνίδια, κατασκευές κλπ) ή κατηγορίες χρηστών (μαθητές, καθηγητές, διευθυντές), ή για να συγκριθεί το εκπαιδευτικό λογισμικό με κάποιο άλλο.

1.3.2 Μέθοδοι Αξιολόγησης

Στην παρούσα εργασία δεν θα αναφερθούμε σε τεχνικές αξιολόγησης που σχετίζονται με την χρησιμότητα του συστήματος (λογισμικού και υλικού), την αντίληψη του χρήστη για το σύστημα, και το αποτέλεσμα της χρήσης του συστήματος. Τέτοιες τεχνικές όπως παρατήρηση, μέτρα αυτό-αναφοράς, συνεντεύξεις, ψυχομετρικά τεστ κλπ είναι έξω από τα ενδιαφέροντα αυτής της εργασίας. Οι αρχικές μέθοδοι υλοποίησης της αξιολόγησης όπως Reviews and Software Evaluation Forms, απέτυχαν να συνυπολογίσουν και την σημασία του περιεχομένου (γνωστικού αντικειμένου) και έτσι δίνουν την θέση τους σε άλλες νεώτερες μεθόδους όπως Frameworks, Heuristics και Field Studies. Τα frameworks στοχεύουν στην υποβοήθηση του έργου των αξιολογητών με την κατηγοριοποίηση του λογισμικού (drill and practice, simulation, tutorial, game, programming language, spreadsheet, CAD/CAM, κλπ.), την περιγραφή των ρόλων τους οποίους το λογισμικό στοχεύει να ικανοποιήσει (π.χ. εκπαιδευτής, εκπαιδευόμενος, εργαλείο, αναπλήρωση του εκπαιδευτικού κλπ), και περιγραφή του λογισμικού με βάση το περιβάλλον λειτουργίας του (και κατηγοριοποίηση του είδους: πολυμεσικό, υπερμεσικό, διαδικτυακό, CD-ROM κλπ.). Τα Heuristics περιλαμβάνουν κριτήρια αξιολόγησης τα οποία στοχεύουν στην αξιολόγηση χαρακτηριστικών του λογισμικού, όπως η ορατότητα της κατάστασης λειτουργίας του λογισμικού από τον χρήστη και η συνέπεια στην ορολογία ή τα πρότυπα. Τα Field Studies αναφέρονται στον πιλοτικό έλεγχο του λογισμικού από τους σπουδαστές κάτω από τον έλεγχο εξειδικευμένων εκπαιδευτικών και με βάση ένα συγκεκριμένο πλάνο ελέγχου.

1.4 Ποιοτικός Έλεγχος Εκπαιδευτικού Λογισμικού

Τίθεται το ερώτημα του πώς εξασφαλίζεται ότι ένα λογισμικό για εκπαίδευση μέσω υπολογιστή ικανοποιεί τον ορισμό του εκπαιδευτικού λογισμικού που δόθηκε προηγούμενα. Εάν το λογισμικό έχει ήδη αναπτυχθεί την απάντηση θα την δώσει η αξιολόγηση του λογισμικού. Βεβαίως, σε μια τέτοια περίπτωση η αξιολόγηση δεν μπορεί βελτιώσει το λογισμικό. Το μόνο που κάνει είναι να καταγράψει τα θετικά και τα αρνητικά στοιχεία του λογισμικού, σε σχέση με τους αρχικούς του στόχους και να προτείνει την εισαγωγή ή απόρριψη της χρήσης του στο σχολικό περιβάλλον. Εάν το λογισμικό πρόκειται να αναπτυχθεί, τότε την απάντηση στο ερώτημα την δίνει ένα καλά σχεδιασμένο Πλάνο Ποιοτικού Ελέγχου (ΠΠΕ) στο οποίο κεντρικό ρόλο παίζει η αξιολόγηση. Το ΠΠΕ εντάσσεται στο Πρόγραμμα Εξασφάλισης Ποιότητας (ΠΕΠ) μιας επιχείρησης. Η αξιολόγηση δεν πρέπει να θεωρείται σαν μια αυτόνομη διαδικασία αλλά σαν μέρος ενός συνόλου διαδικασιών για την εξασφάλιση της ποιότητας του λογισμικού ώστε αυτό να χαρακτηριστεί τελικά σαν εκπαιδευτικό λογισμικό. Τι σημαίνει όμως ποιότητα στο εκπαιδευτικό λογισμικό; Για να το διερευνήσουμε αυτό πρέπει να ψάξουμε πρώτα το τί σημαίνει και πώς ορίζεται η ποιότητα γενικότερα. Μερικοί από τους πιο διαδεδομένους ορισμούς της ποιότητας είναι: "η συλλογή των χαρακτηριστικών και των ιδιοτήτων του προϊόντος που σχετίζονται με τη δυνατότητα του να εκπληρώνει τις ζητούμενες ανάγκες των πελατών", "η συλλογή των χαρακτηριστικών σχεδιασμού, κατασκευής και συντήρησης, διαμέσου των οποίων το προϊόν κατά τη χρήση του θα εκπληρώσει τις προσδοκίες των πελατών", "η συμμόρφωση με τυποποιημένες προδιαγραφές που περιγράφουν τα βασικά χαρακτηριστικά του προϊόντος και

έχουν βασιστεί στις ανάγκες και προσδοκίες των πελατών". Ο κοινός παρανομαστής των ορισμών είναι η ικανοποίηση των απαιτήσεων του πελάτη. Η εξασφάλιση της ποιότητας είναι άμεσα συνδεδεμένη με τις μετρήσεις, ενώ σαν μέτρηση ορίζεται η διαδικασία με την οποία αριθμοί ή σύμβολα αντιστοιχίζονται σε ιδιότητες οντοτήτων του πραγματικού κόσμου έτσι ώστε να τις περιγράφουν σύμφωνα με καθορισμένους κανόνες. Δεν μετράμε το λογισμικό, αλλά ιδιότητες του λογισμικού. Η εξασφάλιση ποιότητας και η χρήση μετρήσεων έδωσε μεγάλη ώθηση στη βιομηχανία τις τελευταίες δεκαετίες. Οι βασικές λειτουργίες του προγράμματος εξασφάλισης ποιότητας στη μαζική παραγωγή υλικών προϊόντων είναι ο καθορισμός των χαρακτηριστικών που θα μετρηθούν και των επιθυμητών ορίων, ο καθορισμός των διαδικασιών μέτρησης, ο εντοπισμός και η απόρριψη των προϊόντων που δεν ικανοποιούν τις ποιοτικές προδιαγραφές και η βελτίωση της διαδικασίας παραγωγής, ώστε να ελαχιστοποιηθεί ο αριθμός των προϊόντων που απορρίπτονται. Αν και οι βασικές αρχές της εξασφάλισης ποιότητας λογισμικού είναι ίδιες με την εξασφάλιση ποιότητας κάθε άλλου προϊόντος, το πρόβλημα στο λογισμικό έγκειται στην έλλειψη μετρήσιμων στόχων και διαδικασιών μέτρησης. Δεν είναι εύκολο να μετρήσουμε την συντηρησιμότητα, την επεκτασιμότητα, την μεταφερισιμότητα ή την αξιοπιστία του λογισμικού. Η εξασφάλιση ποιότητας έχει προταθεί σαν θεραπεία στην γνωστή «χρόνια πάθηση» του λογισμικού (γνωστή και σαν «κρίση») μέσα από διεθνή στάνταρτ όπως το ISO9000, το CMM και άλλα. Αυτά καθορίζουν πώς πρέπει να εφαρμοστεί ένα πρόγραμμα εξασφάλισης ποιότητας στη διαδικασία παραγωγής λογισμικού, αν και αφήνουν αρκετά ανοικτά θέματα. Ενθαρρύνουν την εφαρμογή μετρήσεων, αλλά δεν προτείνουν συγκεκριμένες λύσεις για τις μετρήσεις. Βασικό πρόβλημα στην εξασφάλιση ποιότητας λογισμικού είναι η έλλειψη δυνατότητας άμεσης μέτρησης χαρακτηριστικών για τα οποία να υπάρχει απτή αντίληψη του τί ακριβώς επιθυμούμε να μετρήσουμε. Έχουμε αντίληψη του τί σημαίνει φιλικό user interface αλλά πώς το μετράμε; Επειδή η έννοια «ποιότητα λογισμικού» είναι αρκετά αφηρημένη, ώστε να μπορούν να τεθούν μετρήσιμοι στόχοι, έγινε φανερή η ανάγκη επιμερισμού της «ποιότητας» σε χαρακτηριστικά τα οποία θα συνθέτουν την ποιότητα. Αυτά ονομάζονται «ποιοτικά χαρακτηριστικά» και ορίζονται σαν τα χαρακτηριστικά τα οποία συνθέτουν την ποιότητα ενός προϊόντος, έχουν την ελάχιστη δυνατή επικάλυψη μεταξύ τους και είναι επαρκή για την σύνθεση της ποιότητας. Δεδομένου ότι η «ποιότητα» και οι ποιοτικοί «παράγοντες» είναι εξαιρετικά αφηρημένες έννοιες, προτάθηκε από τον McCall η τμηματοποίηση των «παραγόντων» σε «κριτήρια» που βρίσκονται σε χαμηλότερο επίπεδο αφάιρησης και τα οποία μπορούν να μετρηθούν άμεσα με «μετρικές». Έτσι προέκυψε το μοντέλο παράγοντες-κριτήρια-μετρικές. Μερικοί από τους 11 παράγοντες ποιότητας που πρότεινε ο McCall είναι: *Ορθότητα, συντηρησιμότητα, μεταφερισιμότητα, ακεραιότητα και επαναχρησιμοποιησιμότητα*. Για τους παράγοντες αυτούς προτάθηκαν 23 κριτήρια και 41 μετρικές. Η διαδικασία διάσπασης της ποιότητας σε ποιοτικά χαρακτηριστικά και ο εντοπισμός των ποιοτικών χαρακτηριστικών που ενδιαφέρουν άμεσα κάποιο λογισμικό, με στόχο την αντιστοίχιση μετρικών για τον έλεγχο των επιθυμητών χαρακτηριστικών, είναι μια βασική διαδικασία κάθε προγράμματος εξασφάλισης ποιότητας. Δηλαδή, το περιεχόμενο του Προγράμματος Εξασφάλισης Ποιότητας (ΠΕΠ) είναι διαφορετικό για διαφορετικού τύπου εφαρμογές. Για παράδειγμα τα ποιοτικά χαρακτηριστικά τα οποία περιλαμβάνονται στο ΠΕΠ για ένα εκπαιδευτικό λογισμικό τύπου drill and practice δεν θα είναι ακριβώς τα ίδια με εκείνα που περιλαμβάνονται στο αντίστοιχο ΠΕΠ για ένα εκπαιδευτικό λογισμικό τύπου educational game. Στη συνέχεια θα αναφερθούμε στα εσωτερικά και εξωτερικά χαρακτηριστικά του λογισμικού (και κατ' επέκταση και του εκπαιδευτικού λογισμικού), χωρίς να χάνουμε αναφορά σε συγκεκριμένες μετρικές οι οποίες καταγράφονται αλλού. Σαν εσωτερικά χαρακτηριστικά του λογισμικού ορίζονται τα χαρακτηριστικά για τα οποία υπάρχει απτή και φυσική αντίληψη και άμεσος τρόπος μέτρησης τους (π.χ. γραμμές τεκμηρίωσης). Αυτά είναι χαρακτηριστικά τα οποία μετρούνται εύκολα, αλλά δεν προσφέρουν πληροφορίες υψηλού επιπέδου που σχετίζονται με την ποιότητα του λογισμικού. Σαν εξωτερικά χαρακτηριστικά ορίζονται τα χαρακτηριστικά με υψηλό επίπεδο αφάιρησης τα οποία συνθέτουν την ποιότητα του

λογισμικού (π.χ. μεταφερσιμότητα). Τα χαρακτηριστικά αυτά είναι δύσκολο έως αδύνατο να μετρηθούν άμεσα. Αντίστοιχα με τα εσωτερικά και εξωτερικά χαρακτηριστικά ορίζονται και οι εσωτερικές και εξωτερικές μετρικές μέσω των οποίων 'μετρούνται' τα χαρακτηριστικά. Οι εσωτερικές μετρικές δεν μας ενδιαφέρουν ιδιαίτερα στην εργασία αυτή μια και αναφέρονται εκτενώς στη βιβλιογραφία και δεν θα τις εξετάσουμε περισσότερο. Οι εξωτερικές μετρικές επειδή δεν μπορούν να βασιστούν σε μετρήσεις φυσικών ποσοτήτων, βασίζονται σε 'έρευνες της γνώμης' των 'πελατών' βασισμένες κυρίως σε ερωτηματολόγια και συνεντεύξεις. Έτσι, όσον αφορά το εκπαιδευτικό λογισμικό, οι διάφοροι τύποι αξιολόγησης του που αναφέρθηκαν στην αρχή της εργασίας ουσιαστικά συνιστούν τύπους εξωτερικών μετρικών. Τίθενται επομένως τα ερωτήματα: ποιός είναι ο «πελάτης» και ποιά είναι τα προβλήματα των εξωτερικών μετρικών που αφορούν το εκπαιδευτικό λογισμικό. Στο μεν πρώτο ερώτημα η απάντηση είναι ότι πελάτης είναι ο σπουδαστής, ο εκπαιδευτικός και το πρόγραμμα σπουδών. Φυσικά αναφερόμαστε σε όλες τις βαθμίδες της εκπαίδευσης. Στην περίπτωση της επαγγελματικής κατάρτισης μπορούμε να ορίσουμε τον «πελάτη» αντίστοιχα. Στο δεύτερο ερώτημα, τα βασικά προβλήματα της μέτρησης της γνώμης του πελάτη με χρήση εξωτερικών μετρικών είναι πως τέτοιου είδους μετρήσεις βασίζονται σε υποκειμενικά δεδομένα και είναι αντικείμενο συχνών λαθών. Ένα άλλο πρόβλημα των εξωτερικών μετρικών είναι ότι απαιτούν έντονη συμμετοχή ανθρώπινου δυναμικού με αποτέλεσμα το αυξημένο κόστος. Από την άλλη πλευρά οι εξωτερικές μετρικές δίνουν αποτελέσματα που είναι άμεσα αξιοποιήσιμα μια και μετρούν απευθείας τα ζητούμενα εξωτερικά ποιοτικά χαρακτηριστικά του λογισμικού. Έτσι, οι εξωτερικές μετρικές μπορούν να αποτελέσουν τον κορμό ενός Προγράμματος Εξασφάλισης Ποιότητας, με τις εσωτερικές μετρικές (οι οποίες μπορούν να είναι αυτοματοποιημένες με τη βοήθεια προγραμματιστικών εργαλείων) να χρησιμοποιούνται για την εκτίμηση της ποιότητας των τμημάτων του λογισμικού και τις εξωτερικές να μετρούν τα καθαρά ποιοτικά χαρακτηριστικά αυτού.

2. Παρούσα Κατάσταση

Το Εκπαιδευτικό Λογισμικό άρχισε να γνωρίζει μεγάλη άνθιση κυρίως από τις αρχές της δεκαετίας του '90. Ένας απέραντος αριθμός τίτλων αναπτύχθηκε και κυκλοφόρησε από τα μέσα της δεκαετίας του '90 και μετά, οι οποίοι στόχευαν πρώτιστα στην εκπαίδευση των μικρότερων παιδιών εξ αποστάσεως. Έτσι, άρχισαν να αναπτύσσονται και πολλά εκπαιδευτικά παιχνίδια καθώς κέντριζαν το ενδιαφέρον των παιδιών και συγχρόνως τα βοηθούσαν να μαθαίνουν με έναν πιο ευχάριστο και αποδοτικό τρόπο. Η χρησιμοποίηση και παραγωγή εκπαιδευτικών προγραμμάτων στην Ελλάδα έχει αυξηθεί περισσότερο τα τελευταία χρόνια. Στη συνέχεια θα παραθέσουμε κάποιες χαρακτηριστικές περιπτώσεις εκπαιδευτικού λογισμικού και εκπαιδευτικών παιχνιδιών.

i) Ποιος θέλει να γίνει πτυχιούχος; (www.ptyxiouxos.net)

Το εκπαιδευτικό αυτό παιχνίδι αποτελεί παράφραση του πασίγνωστου τηλεπαιχνιδιού «ποιος θέλει να γίνει εκατομμυριούχος». Είναι μία αμιγώς ελληνική ιστοσελίδα και αφορά μαθητές δημοτικού, γυμνασίου και λυκείου, που μέσα από διάφορα εκπαιδευτικά παιχνίδια

διδάσκει στους μαθητές διάφορα ενδοσχολικά μαθήματα, όπως μαθηματικά, γεωγραφία, ιστορία κτλ.

The screenshot shows the homepage of the website www.ptyxiouchos.net. The page is designed for students and features several key sections:

- Navigation Bar:** Includes links for 'Προσθήκη στα αγαπημένα', 'Η ομάδα', 'Επικοινωνία', 'F. A. Q', and 'Διαχείριση'.
- Main Header:** 'ΠΟΙΟΣ ΘΕΛΕΙ ΝΑ ΓΙΝΕΙ ΠΤΥΧΙΟΥΧΟΣ; ΜΑΘΗΣΗ ΜΕΣΑ ΑΠΟ ΕΚΠΑΙΔΕΥΤΙΚΑ ΠΑΙΧΝΙΔΙΑ'.
- Navigation Menu:** 'ΔΙΑΚΡΙΣΕΙΣ', 'ΖΩΝΤΑΝΗ ΣΥΝΟΜΙΑΙΑ', 'ΒΙΒΛΙΟ ΕΠΙΣΚΕΠΤΩΝ', 'ΣΤΑΤΙΣΤΙΚΑ', 'FORUM'.
- ΣΥΝΔΕΣΗ (Login):** Fields for 'Όνομα χρήστη' and 'Κωδικός', with options for 'Αυτόματη είσοδος', 'Είσοδος', 'Νέος χρήστης', and 'Ξέχασα τον κωδικό μου'.
- ΚΑΛΩΣΟΡΙΣΤΕ! (Welcome):** A message encouraging users to use educational games to learn, mentioning that the site is free and supported by donations.
- ΝΕΑ ΚΑΤΑΧΩΡΙΣΗ (New Registrations):** A list of registration options: 'Εισαγωγή ερώτησης', 'Εισαγωγή προορισμού', 'Εισαγωγή διαγωνίσματος', and 'Εισαγωγή ΜγΠτυχιούχος'.
- ΕΝΗΜΕΡΩΣΗ (News):** A section titled 'GRedu' with the text 'Curated by Greek School Network' and a quote: '«Η Εκπαίδευση για την αειφορία στα προγράμματα των καινοτόμων δράσεων»'.
- ΕΚΠΑΙΔΕΥΤΙΚΑ ΠΑΙΧΝΙΔΙΑ (Educational Games):** A list of game categories: 'Πτυχιούχος Λυκείου', 'Πτυχιούχος Γυμνασίου', 'Πτυχιούχος Δημοτικού', and 'Online Διαγωνίσματα'.
- ΚΟΡΥΦΑΙΟΙ ΑΠΟΣΤΟΛΕΙΣ (Top Scorers):** A table listing top scorers with their names and scores.

10.	Γεωργία Πετραλιφή	600
11.	ΚΩΔΩΝΑΚΗΣ ΓΙΩΡΓΟΣ	540
12.	Αντώνης Μπούρας	500
13.	Γεωργία Καλύβα	490
14.	Βαγγέλης Τσακαλώφας	480
15.	...	470
- ΚΑΛΥΤΕΡΟΙ ΠΑΙΚΤΕΣ (Best Players):** A section for 'ΚΑΛΥΤΕΡΟΙ ΠΑΙΚΤΕΣ'.

Εικόνα 2.1 Ιστοσελίδα του «ποιος θέλει να γίνει πτυχιούχος» (www.ptyxiouchos.net).

ii) Jele (www.jele.gr)

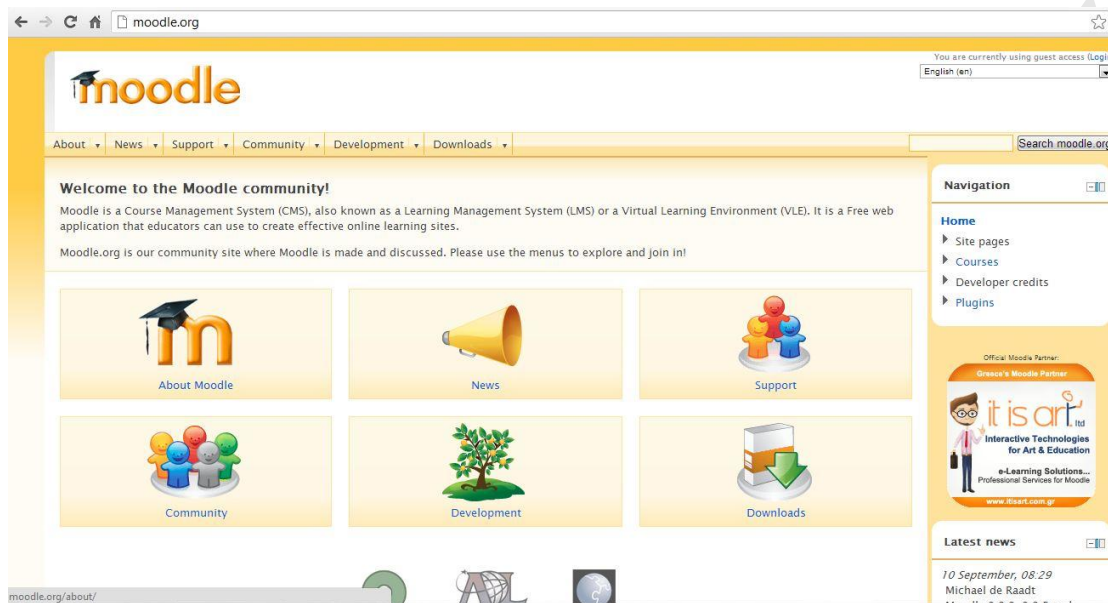
Άλλο ένα εκπαιδευτικό παιχνίδι ελληνικής προέλευσης, που απευθύνεται σε μαθητές των πρώτων τριών τάξεων δημοτικού και το οποίο με ιδιαίτερα χαριτωμένο τρόπο διδάσκει τους μαθητές γλώσσα και μαθηματικά στις τάξεις της πρώτης και δευτέρας δημοτικού, ενώ στους μαθητές της τρίτης δημοτικού προστίθεται και το μάθημα της ιστορίας. Το χαρακτηριστικό αυτής της εφαρμογής είναι ότι χρησιμοποιεί πολλούς χαρακτήρες (agents) για να κάνει την εκμάθηση πιο ευχάριστη και δημιουργική.



Εικόνα 2.2 Ιστοσελίδα Jele (www.jele.gr).

iii) M.O.O.D.L.E. (Modular Object Oriented Dynamic Learning Environment) (www.moodle.org)

Το MOODLE είναι μία διαδικτυακή πλατφόρμα ανοικτού λογισμικού για τη διαχείριση εκπαιδευτικού περιεχομένου (Course Management System). Χρησιμοποιείται σε 205 χώρες ανά τον κόσμο και είναι μεταφρασμένο σε περίπου 80 γλώσσες. Μπορεί να εγκατασταθεί σε servers που έχουν λειτουργικό Windows, Linux ή Macintosh και έχει γραφτεί σε PHP. Διαθέτει αρκετές αλληλεπιδραστικές δραστηριότητες, όπως υποβολή εργασίας, η ζωντανή συνομιλία (chat), οι ψηφοφορίες, οι ομάδες συζήτησης, το γλωσσάρι ορολογιών μαθήματος, τα κουίζ, η συλλογική συγγραφή κειμένων, οι έρευνες και τα παιχνίδια. Τα εκπαιδευτικά παιχνίδια που υποστηρίζει το MOODLE είναι η κρεμάλα, το σταυρόλεξο, το κρυπτόλεξο, ο εκατομμυριούχος, το Sudoku, την «κρυμμένη» εικόνα και το φιδάκι.



Εικόνα 2.3 Ιστοσελίδα MOODLE (www.moodle.org).

3. Ανάλυση και Σχεδιασμός

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

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

Συντήρηση των προγραμμάτων, έτσι ώστε να ανταποκρίνονται στις μεταβαλλόμενες ανάγκες των πελατών.

3.1 Κύκλος ζωής ανάπτυξης λογισμικού

Υπάρχουν διάφορα μοντέλα για την περιγραφή του κύκλου ζωής ανάπτυξης του λογισμικού. Ενδεικτικά θα αναφέρουμε το μοντέλο καταρράκτη, την εξελικτική ανάπτυξη και την τεχνολογία λογισμικού βάσει συστατικών στοιχείων [2].

3.1.1 Μοντέλο Καταρράκτη

Το μοντέλο του Καταρράκτη (waterfall ή linear sequential model) αναπτύχθηκε από τον Royce, το 1970 και περιλαμβάνει 5 διακριτές φάσεις. Ήταν το πρώτο μοντέλο που δημιουργήθηκε και έγινε ευρέως αποδεκτό, ενώ ακόμα παραμένει δημοφιλές ιδιαίτερα για μικρά ή μεσαία μεγέθη εφαρμογών αφού συμβάλλει στην επιτυχή κατασκευή αξιόπιστων προϊόντων σε μικρό χρονικό διάστημα. Στο μοντέλο αυτό οι διάφορες φάσεις διαχωρίζονται και ακολουθούνται σειριακά. Η κάθε φάση παράγει ενδιάμεσα αποτελέσματα τα οποία χρησιμοποιούνται από τις επόμενες φάσεις και κορυφώνεται από μια διαδικασία επικύρωσης ή επαλήθευσης των προϊόντων που παράγονται, με σκοπό να απαλειφθούν τυχόν σφάλματα. Οι εργασίες που γίνονται σε κάθε στάδιο του κύκλου ζωής είναι οι εξής:

1. Ανάλυση Απαιτήσεων

Μέσω συμβουλευτικής διαδικασίας με τους χρήστες του συστήματος προσδιορίζονται οι υπηρεσίες που θα παρέχει το σύστημα, οι περιορισμοί και οι στόχοι του.

2. Σχεδιασμός συστήματος και λογισμικού

Η διαδικασία σχεδιασμού του συστήματος χωρίζει τις απαιτήσεις σε απαιτήσεις υλικού και απαιτήσεις λογισμικού. Εγκαθιδρύει μία γενική αρχιτεκτονική συστήματος. Ο σχεδιασμός του λογισμικού περιλαμβάνει την αναγνώριση και την περιγραφή των σημαντικών μερών του συστήματος λογισμικού και των σχέσεων μεταξύ τους.

3. Υλοποίηση και έλεγχος ενοτήτων

Κατά τη διάρκεια αυτού του σταδίου, ο σχεδιασμός του λογισμικού γίνεται αντιληπτός ως ένα σύνολο προγραμμάτων ή ενοτήτων προγραμμάτων. Ο έλεγχος των ενοτήτων περιλαμβάνει την επαλήθευση του γεγονότος ότι κάθε ενότητα εκπληρώνει τις προδιαγραφές της.

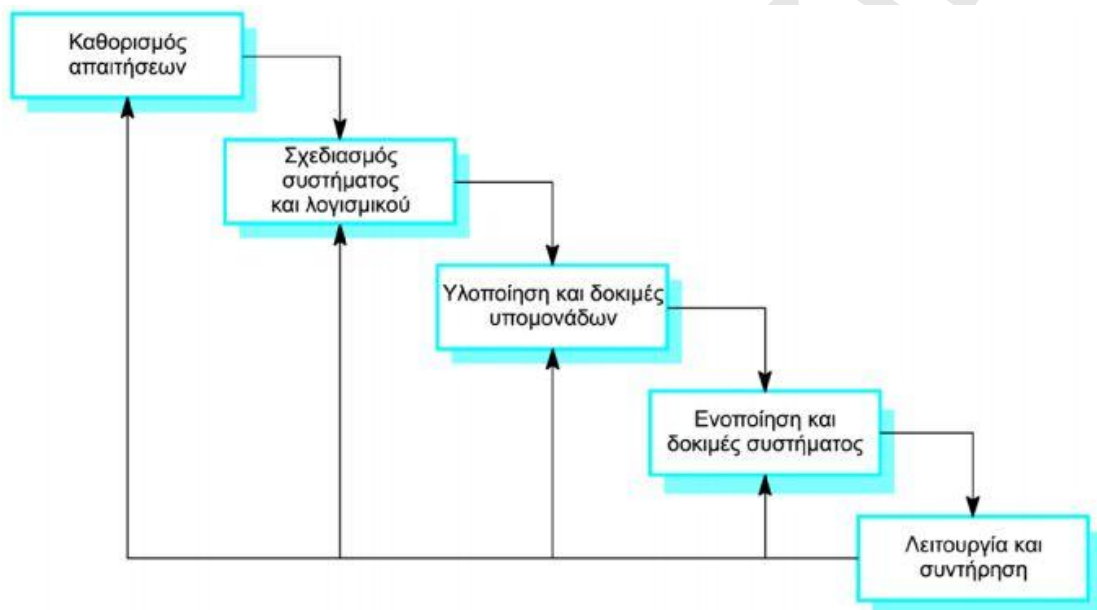
4. Ενσωμάτωση και έλεγχος συστήματος

Οι ανεξάρτητες ενότητες των προγραμμάτων ή τα ίδια τα διαφορετικά προγράμματα συγχωνεύονται και ελέγχονται σαν ένα ολοκληρωμένο σύστημα για να διασφαλιστεί ότι έχουν εκπληρωθεί οι απαιτήσεις του λογισμικού. Μετά τον έλεγχο, το σύστημα λογισμικού παραδίδεται στον πελάτη.

5. Λειτουργία και συντήρηση

Κανονικά, αλλά όχι απαραίτητα, αυτή είναι η πιο χρονοβόρα φάση του κύκλου ζωής. Το σύστημα εγκαθίσταται και μπαίνει σε χρήση στην πράξη. Η συντήρηση περιλαμβάνει τη διόρθωση λαθών που δεν έγιναν αντιληπτά σε νωρίτερα στάδια του κύκλου ζωής,

βελτιώνοντας την εφαρμογή των διαφόρων ενότητων του συστήματος και τις υπηρεσίες που προσφέρει το σύστημα, αφού ανακαλύπτονται νέες απαιτήσεις.

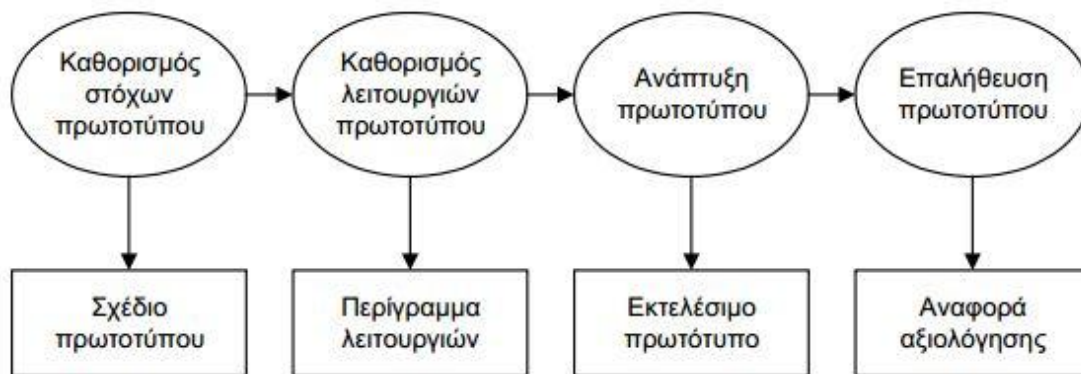


Εικόνα 3.1.1.1 Μοντέλο Καταρράκτη.

Τα μειονεκτήματα που παρουσιάζει το μοντέλο του καταρράκτη εντοπίζονται στα εξής: α) στο ότι γνωρίζουμε αν θα είναι ικανοποιημένος ο χρήστης μόνο στο τελικό στάδιο (ουσιαστικά αργά), β) στο ότι δεν προβλέπει επαναχρησιμοποίηση του λογισμικού που πιθανά υπάρχει, και γ) στο ότι είναι απαιτητικό τόσο σε χρόνο όσο και σε κόστος και αρκετά δύσκαμπτο μοντέλο.

3.1.2 Μοντέλο Πρωτοτυποποίησης

Ένας τρόπος να δούμε την πρωτοτυποποίηση είναι ως μια τεχνική για μείωση του κινδύνου. Ο πιο σημαντικός κίνδυνος στην ανάπτυξη λογισμικού είναι τα λάθη και περισσότερο οι παραλείψεις που προκύπτουν από μη σαφείς απαιτήσεις των χρηστών για το τελικό σύστημα. Το κόστος της διόρθωσης αυτών των λαθών και παραλείψεων σε επόμενα στάδια μπορεί να είναι πολύ υψηλό. Είναι προφανές ότι η δημιουργία ενός πρωτοτύπου μπορεί να μειώσει τον αριθμό των προβλημάτων των απαιτήσεων και ως εκ τούτου να μειώσει το συνολικό κόστος ανάπτυξης. Παρακάτω βλέπουμε τη σχηματική αναπαράσταση του μοντέλου πρωτοτυποποίησης.



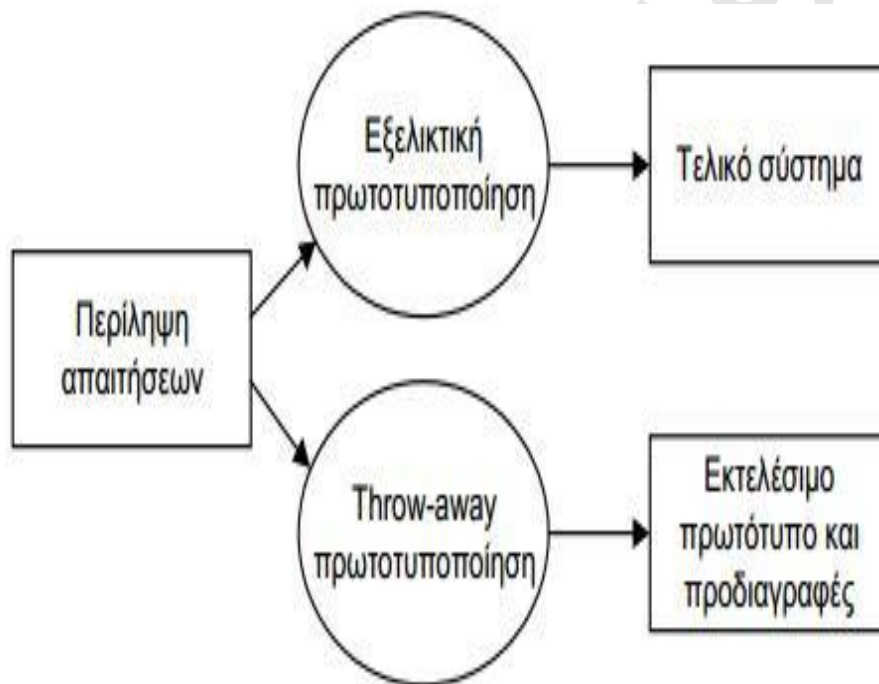
Εικόνα 3.1.2.1 Μοντέλο Πρωτοτυποποίησης.

Αρχικά, θα πρέπει να καθοριστούν επακριβώς οι στόχοι του πρωτότυπου. Το πρωτότυπο σύστημα μπορεί να αφορά τη διεπαφή χρήστη ή να περιέχει τις λειτουργίες εκείνες που θεωρούνται περισσότερο κρίσιμες. Είναι προφανές ότι ένα πρωτότυπο δεν μπορεί να καλύπτει όλες τις απαιτήσεις του συστήματος. Για τον λόγο αυτό κάθε φορά θα πρέπει να ορίζονται πλήρως οι απαιτήσεις που αυτό θα καλύπτει, αλλιώς τελικά μπορεί να μην λάβουμε τα πλεονεκτήματα που μας προσφέρει η μέθοδος αυτή. Το επόμενο στάδιο αφορά στην επιλογή των λειτουργιών εκείνων από το σύνολο των λειτουργιών του τελικού συστήματος που θα συμπεριληφθούν στο προς δημιουργία πρωτότυπο (άρα και αυτών που δεν θα συμπεριληφθούν). Ο λόγος για την επιλογή αυτή είναι, γιατί προφανώς έχει πολύ υψηλό κόστος εάν το πρωτότυπο δημιουργηθεί με όλες τις λειτουργίες του τελικού συστήματος. Βέβαια θα μπορούσε να αποφασιστεί να περιλαμβάνονται όλες οι λειτουργίες που έχει αποφασιστεί αλλά σε μειωμένο επίπεδο (πχ. χωρίς διαχείριση λαθών). Η τελευταία φάση, μετά την ανάπτυξη του πρωτότυπου, είναι η επαλήθευση του πρωτότυπου και είναι ίσως η πιο σημαντική φάση. Θα πρέπει να καταγραφούν συμπεράσματα για το πως νιώθουν οι χρήστες με το σύστημα, αν γίνεται κατανοητό το περιβάλλον και η λειτουργία του και να βρεθούν τυχόν λάθη και προβλήματα. Τα πλεονεκτήματα της χρήσης πρωτότυπου είναι ότι ανακαλύπτονται και διορθώνονται:

- Παρεξηγήσεις μεταξύ των χρηστών και των δημιουργών.
- Παραλειπόμενες υπηρεσίες στο σύστημα
- Δυσκολίες στη χρήση
- Ασυνέχειες και κενά στις προδιαγραφές

Η πρωτοτυποποίηση μπορεί να χρησιμοποιηθεί και για άλλους σκοπούς, όπως στην εκπαίδευση των χρηστών, δηλαδή το πρωτότυπο να χρησιμοποιηθεί ως εκπαιδευτικό εργαλείο για την εκμάθηση του τελικού συστήματος, αλλά ταυτόχρονα είναι και ένας τρόπος μείωσης του κινδύνου, αφού περιορίζονται τα λάθη και οι παραλείψεις. Αν τα λάθη αφεθούν για διόρθωση στις τελευταίες φάσεις του κύκλου ζωής το κόστος αυξάνεται κατακόρυφα. Το βασικό μειονέκτημα του μοντέλου της πρωτοτυποποίησης είναι ότι το κόστος ανάπτυξής του αποτελεί ένα μεγάλο μέρος του συνολικού κόστους του συστήματος που αναπτύσσεται. Πολλές φορές είναι οικονομικά πιο συμφέρον να μεταβληθεί το τελικό προϊόν από το να δημιουργηθεί ένα πρωτότυπο. Είναι προφανές, ότι είναι πολύ δύσκολο να προβλεφτεί ποιες ακριβώς δυσκολίες θα αντιμετωπίσει ο τελικός χρήστης από την καθημερινή χρήση ενός νέου συστήματος λογισμικού. Ιδιαίτερα εάν αναφερόμαστε σε μεγάλα συστήματα λογισμικού η δυσκολία αυτή μπορεί να καταφανεί μόνο όταν το ολοκληρωμένο σύστημα αναπτυχθεί και τεθεί σε λειτουργία.

Για να αντιμετωπιστεί αυτή η δυσκολία μπορεί να χρησιμοποιηθεί η εξελικτική (evolutionary) πρωτοτυποποίηση. Κατά τη προσέγγιση αυτή δημιουργείται μια περιορισμένη (ατελής) έκδοση του συστήματος πάνω στην οποία γίνονται οι διορθώσεις και οι προσθήκες καθώς απαιτήσεις είτε διευκρινίζονται είτε ανακαλύπτονται είτε βελτιώνονται μέχρι να καταλήξουμε σε μια έκδοση που να είναι επαρκής και ικανοποιητική. Το βασικό πρόβλημα του μοντέλου της εξελικτικής πρωτοτυποποίησης είναι ότι με τις συνεχείς διορθωτικές και προσθετικές παρεμβολές παράγεται «μπλεγμένος» κώδικας (spaghetti code) που είναι πολύ δύσκολα συντηρήσιμος ιδιαίτερα όταν δεν τηρούνται κανόνες ποιότητας λογισμικού (Ξένος, 2003). Εναλλακτικά και προκειμένου να αποφύγουμε το φαινόμενο του «μπλεγμένου κώδικα», θα μπορούσε να δημιουργηθεί ένα πρωτότυπο το οποίο θα είχε αποκλειστικό σκοπό την αποσαφήνιση των απαιτήσεων και την παροχή πληροφοριών για την εκτίμηση του κινδύνου του τελικού συστήματος. Μετά την αξιολόγηση το πρωτότυπο σύστημα «πετιέται» και δεν χρησιμοποιείται για την ανάπτυξη του συστήματος (throw-away prototype).

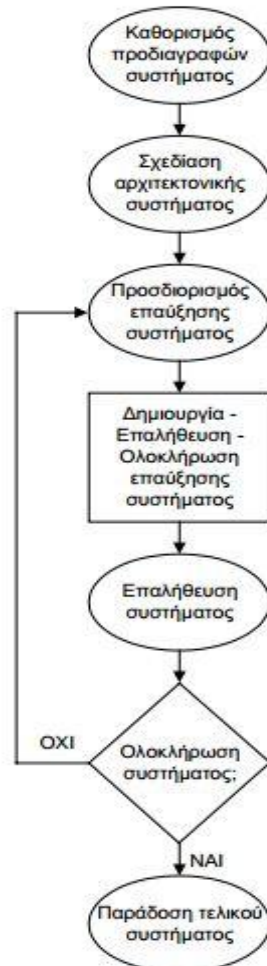


Εικόνα 3.1.2.2 Εξελικτική και throw-away Πρωτοτυποποίηση.

3.1.3 Μοντέλο Λειτουργικής Επαύξεσης

Μια εναλλακτική διαδικασία που συνδυάζει τα πλεονεκτήματα της εξελικτικής προσέγγισης με τον έλεγχο που απαιτείται για μεγάλα συστήματα είναι η λειτουργική επαύξηση (incremental development). Σύμφωνα με το μοντέλο αυτό αρχικά αναπτύσσεται μια έκδοση του συστήματος που περιέχει τις περισσότερες σημαντικές και κρίσιμες λειτουργίες. Από την χρησιμοποίηση αυτής της έκδοσης κερδίζεται εμπειρία η οποία χρησιμοποιείται για την βελτίωσή της. Στην συνέχεια γίνεται μια προσαύξηση που επεκτείνει την προηγούμενη έκδοση και περιέχει και άλλες λειτουργίες. Η νέα έκδοση εκλεπτύνεται και προσαυξάνεται με την σειρά της με τον ίδιο τρόπο έως ότου κατασκευαστεί η τελική έκδοση. Το μοντέλο ενδείκνυται στις περιπτώσεις που υπάρχει σαφής γνώση και πολύ μικρή ή καθόλου μεταβλητότητα των απαιτήσεων του υπό ανάπτυξη εκπαιδευτικού λογισμικού. Άρα πρόκειται για μοντέλο που

χρησιμοποιείται σε λίγες περιπτώσεις μια και το βασικό πρόβλημα της ανάπτυξης εκπαιδευτικού λογισμικού είναι η ασάφεια (σε μικρότερο ή μεγαλύτερο βαθμό) των απαιτήσεων του συστήματος.



Εικόνα 3.1.3.1 Λειτουργική Επαύξηση.

Με αυτό το μοντέλο αποφεύγονται προβλήματα που προκύπτουν από τις συνεχείς αλλαγές, όπως στην εξελικτική πρωτοτυποποίηση. Η αρχιτεκτονική του συστήματος καθορίζεται σχετικά νωρίς, και λειτουργεί ως πλαίσιο. Τα μέρη που αποτελούν το σύστημα αναπτύσσονται με επαυξήσεις και παραδίδονται με αυτό τον τρόπο.

3.1.4 Εναλλακτικά Μοντέλα Ανάπτυξης Εκπαιδευτικού Λογισμικού

α) Λειτουργικό Μοντέλο

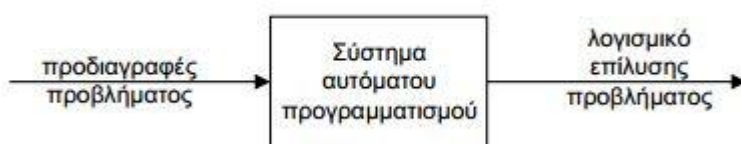
Το Λειτουργικό Μοντέλο (operational model) χρησιμοποιεί αποκλειστικά τις προδιαγραφές που είναι λειτουργικές και περιγράφουν τι θα κάνει το σύστημα έμμεσα. Έτσι, δημιουργείται μια περιγραφή η οποία δείχνει πως αυτό θα λειτουργεί, η οποία υλοποιείται σε

κάποια γλώσσα ώστε να μπορεί να εκτελεστεί. Έτσι μπορεί να γίνει φανερή η συμπεριφορά του τελικού συστήματος και ως εκ τούτου να αξιολογηθεί και να βελτιωθεί εάν απαιτείται. Οι λειτουργικές προδιαγραφές που χρησιμοποιούνται σύμφωνα με το μοντέλο αυτό είναι ένα είδος πρωτότυπου στο οποίο είναι εμφανής όλη η λειτουργική συμπεριφορά του συστήματος, χρησιμοποιώντας διαφορετικά μέσα από αυτά που θα χρησιμοποιούσε το τελικό σύστημα. Αξιολογώντας τη συμπεριφορά του συστήματος, οι χρήστες μπορούν να κάνουν παρατηρήσεις και αλλαγές στις λειτουργικές προδιαγραφές. Ο κύκλος αυτός αξιολόγησης – αλλαγών επαναλαμβάνεται έως ότου θεωρηθεί ότι το σύστημα έχει την επιθυμητή λειτουργικότητα. Έτσι ολοκληρώνεται η φάση των απαιτήσεων και στην συνέχεια μπορεί να χρησιμοποιηθεί για παράδειγμα το μοντέλο του καταρράκτη από την φάση της σχεδίασης και κάτω. Βασικό πρόβλημα του μοντέλου αποτελεί το γεγονός ότι χρησιμοποιεί εκτελέσιμες γλώσσες προδιαγραφών (executable specification languages) που είναι αυστηρά τυπικές (formal) και απαιτούν ιδιαίτερες γνώσεις από την ομάδα ανάπτυξης αλλά και εργαλεία υλοποίησης (compilers – interpreters, specification languages).

β) Μοντέλο Αυτόματου Προγραμματισμού

Το μοντέλο αυτόματου προγραμματισμού (automatic programming model) βασίζεται στην ιδέα της δημιουργίας ενός συστήματος που να μπορεί να δημιουργήσει λογισμικό αυτόματα αφού πρώτα του δοθούν οι προδιαγραφές του προβλήματος. Η αυτοματοποιημένη δημιουργία λογισμικού είναι μια παλιά ιδέα που χρησιμοποιήθηκε και για την δημιουργία μεταφραστών γλωσσών προγραμματισμού (μετα-μεταφραστές / meta-translators). Το μοντέλο αυτόματου προγραμματισμού ενδείκνυται για τις περιπτώσεις που οι απαιτήσεις του συστήματος είναι σαφώς καθορισμένες ώστε να μπορούν να

περιγραφούν με ένα πολύ τυπικό τρόπο όπως είναι οι γραμματικές χωρίς συμφραζόμενα κλπ. Τα εργαλεία που χρησιμοποιούνται (μετα-μεταφραστές και οι αντίστοιχες γραμματικές) απαιτούν υψηλές γνώσεις πληροφορικής και επομένως πολύ εξειδικευμένη και έμπειρη ομάδα ανάπτυξης.



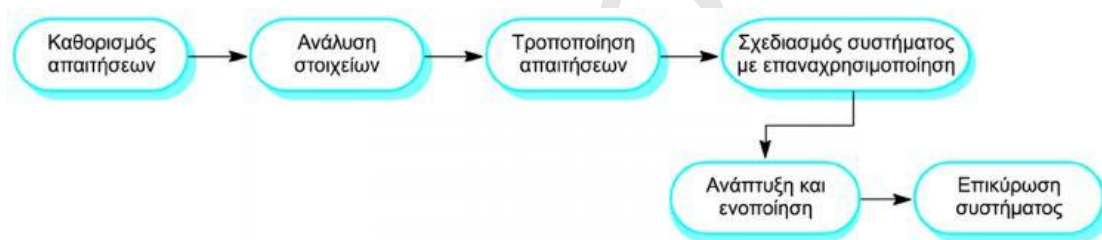
Εικόνα 3.1.4.1 Σύστημα Αυτόματου Προγραμματισμού.

γ) Μοντέλο Επαναχρησιμοποίησης Λογισμικού

Με το μοντέλο επαναχρησιμοποίησης λογισμικού (software reusability model) γίνεται χρήση ήδη υπάρχοντος και δοκιμασμένου λογισμικού, σχεδίων και κώδικα. Οι υπάρχουσες ψηφίδες λογισμικού (με ελεγμένη ορθότητα) ενσωματώνονται σε νέα προϊόντα λογισμικού. Η διαδικασία αυτή δεν είναι εύκολη, αφού παρουσιάζονται δυσκολίες, λόγω της ανυπαρξίας εργαλείων και τεχνικών καταλλήλων για αυτή τη δουλειά, αλλά και της έλλειψης προτύπων κατασκευής ψηφίδων λογισμικού που να μπορούν να επαναχρησιμοποιηθούν. Τα βασικά πλεονεκτήματα του μοντέλου είναι η συντόμευση του χρονικού διαστήματος κατασκευής λογισμικού αλλά και η βελτίωση της αξιοπιστίας του αφού στηρίζεται σε έτοιμα, δοκιμασμένα και

άρα αξιόπιστα τμήματα λογισμικού. Τα συστατικά λογισμικού που θα επαναχρησιμοποιηθούν μπορεί να είναι πολλών και διαφόρων μεγεθών, όπως για παράδειγμα :

- Επαναχρησιμοποίηση ολόκληρων συστημάτων εφαρμογών (application system reuse) τα οποία είτε ενσωματώνονται στο καινούριο σύστημα χωρίς αλλαγή είτε δημιουργούνται ολόκληρες οικογένειες εφαρμογών που μπορούν να τρέξουν σε διαφορετικές πλατφόρμες ώστε να ικανοποιήσουν συγκεκριμένες ανάγκες.
- Επαναχρησιμοποίηση ψηφίδων (component reuse) όπου διάφορα 'συστατικά' μιας εφαρμογής από ένα υποσύστημα μέχρι ένα μεμονωμένο αντικείμενο μπορούν να επαναχρησιμοποιηθούν. Για παράδειγμα αντικείμενο τύπου 'χρονόμετρο' μπορεί να επαναχρησιμοποιηθεί πολλές φορές στην ανάπτυξη ενός εκπαιδευτικού λογισμικού φυσικής, χημείας κλπ.
- Επαναχρησιμοποίηση κάποιας λειτουργίας ή συνάρτησης (function reuse) όπου συστατικά (συναρτήσεις, διαδικασίες) που υλοποιούν μια απλή λειτουργία όπως μια μαθηματική συνάρτηση επαναχρησιμοποιούνται σε άλλες εφαρμογές. Αυτού του είδους η επαναχρησιμοποίηση που βασίζεται στις βιβλιοθήκες είναι και η πιο διαδεδομένη. Το προφανές κέρδος από την επαναχρησιμοποίηση είναι η μείωση του κόστους ανάπτυξης δεδομένου ότι λιγότερα συστατικά του συστήματος χρειάζεται να προσδιοριστούν, να σχεδιαστούν, να υλοποιηθούν και να αξιολογηθούν.



Εικόνα 3.1.4.2 Επαναχρησιμοποίηση Λογισμικού.

δ) Αντικειμενοστραφές Μοντέλο

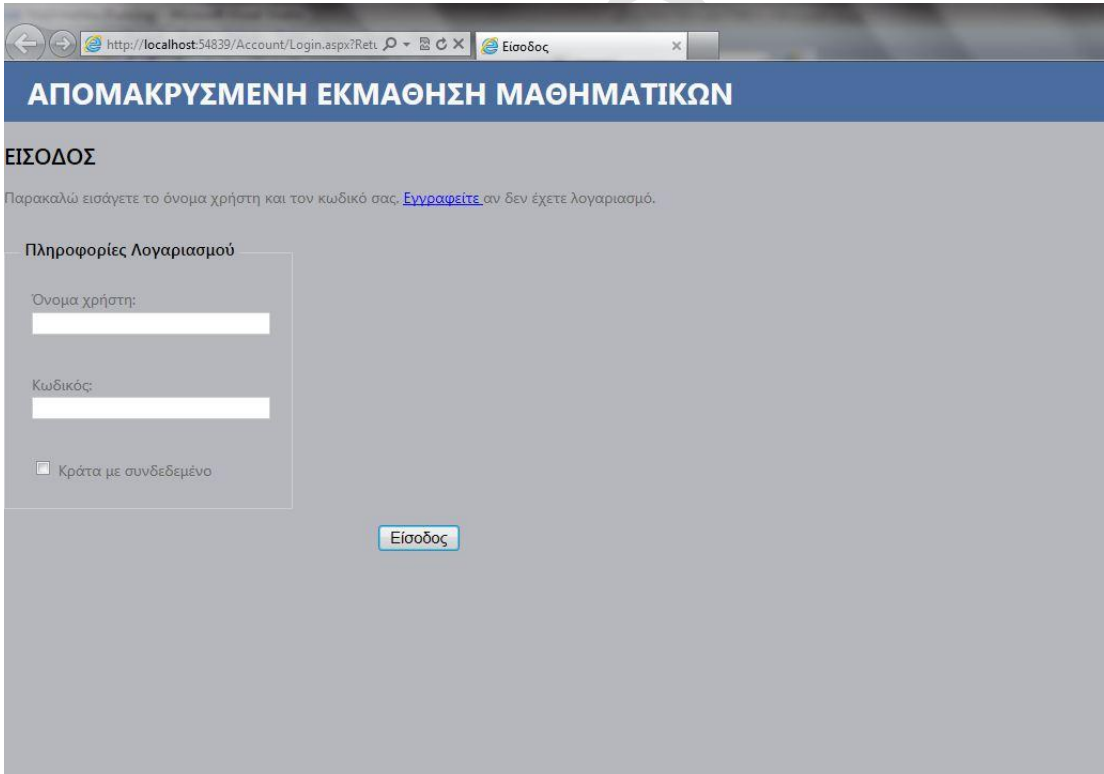
Το αντικειμενοστραφές μοντέλο (object-oriented model) βασίζεται στον αντικειμενοστραφή προγραμματισμό (βλέπε και Βεσκούκης, 2000). Αναπτύσσεται με τρόπο παρόμοιο με το μοντέλο του καταρράκτη, αλλά διαφέρει σε δύο βασικά σημεία:

- Οι διάφορες φάσεις υπερκαλύπτονται μεταξύ τους
- Η ανάπτυξη του, αν χρειαστεί οπισθοδρομεί στην προηγούμενη φάση, εκτός από την τελευταία που οπισθοδρομεί στην αρχή.

Το κύριο πλεονέκτημα του μοντέλου είναι ότι κάνει χρήση επαναχρησιμοποιήσιμων μονάδων και με αυτό τον τρόπο συντομεύεται τόσο η φάση της ανάπτυξης όσο και η φάση της συντήρησης.

4. Παρουσίαση εφαρμογής

Όταν ένας χρήστης προσπαθεί να εισέλθει για πρώτη φορά στην ιστοσελίδα, τότε αυτός θα συναντήσει την σελίδα εισόδου (Login Page) Login.aspx (Εικόνα 4.1).



The screenshot shows a web browser window with the URL `http://localhost:54839/Account/Login.aspx?Retn`. The page title is "Είσοδος". The main heading is "ΑΠΟΜΑΚΡΥΣΜΕΝΗ ΕΚΜΑΘΗΣΗ ΜΑΘΗΜΑΤΙΚΩΝ". Below this, the section is titled "ΕΙΣΟΔΟΣ". A message reads: "Παρακαλώ εισάγετε το όνομα χρήστη και τον κωδικό σας. [Εγγραφείτε](#) αν δεν έχετε λογαριασμό." The form is titled "Πληροφορίες Λογαριασμού" and contains two input fields: "Όνομα χρήστη:" and "Κωδικός:". Below the fields is a checkbox labeled "Κράτα με συνδεδεμένο". At the bottom of the form is a button labeled "Είσοδος".

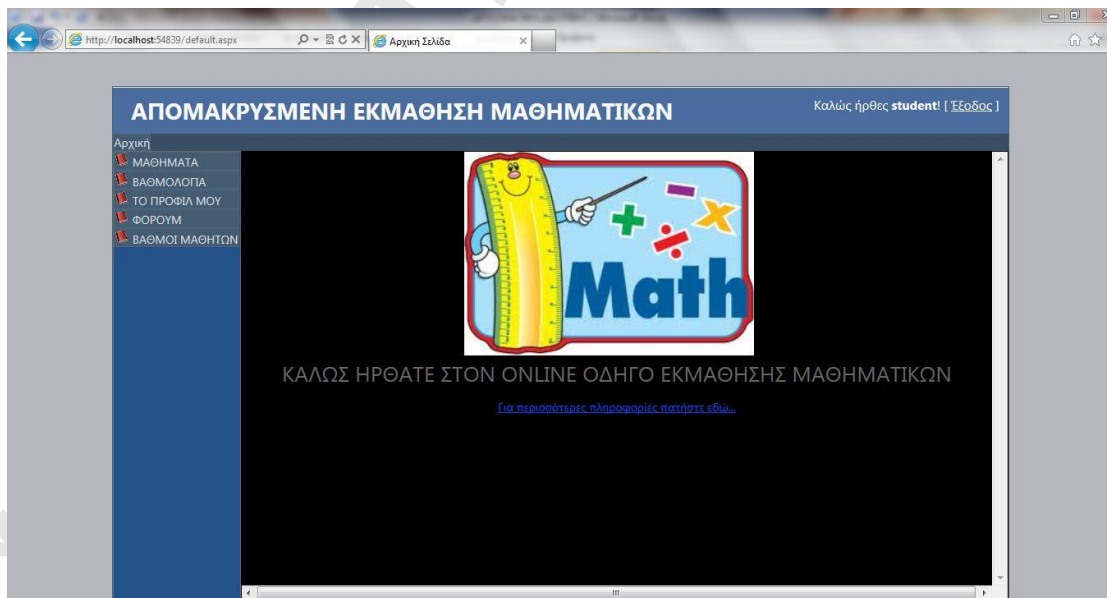
Εικόνα 4.1 Σελίδα εισόδου.

Εκεί μιάς και δεν έχει ξαναμπεί στο σύστημα θα πρέπει να κάνει εγγραφή, ειδάλλως το σύστημα θα του πει ότι ο συγκεκριμένος χρήστης δεν υπάρχει. Έτσι πηγαίνοντας στη σελίδα εγγραφής (Registration Page) Register.aspx (Εικόνα 4.2) προσθέτει τις κατάλληλες πληροφορίες και εφόσον τις υποβάλει τότε η εγγραφή είναι επιτυχής. Να σημειωθεί πως όλα τα πεδία στη σελίδα εγγραφής είναι υποχρεωτικά και ο κωδικός θα πρέπει να έχει μήκος μεγαλύτερο ή ίσο του 6, να

ταιριάζει με την επαλήθευσή του και να μην υπάρχει βέβαια χρήστης με το ίδιο όνομα. Για όλους αυτούς τους περιορισμούς το σύστημα ενημερώνει κατάλληλα το χρήστη. Στη σελίδα εγγραφής ο χρήστης μπορεί να επιλέξει να κατέχει ρόλο καθηγητή ή μαθητή.

Εικόνα 4.2. Σελίδα εγγραφής.

Μετά την επιτυχή εγγραφή του συμπληρώνει τις πληροφορίες στη σελίδα εισόδου και με τικάρισμα του checkbox «Κράτα με συνδεδεμένο» μπορεί σε επόμενη είσοδό του να μπει απευθείας στην αρχική σελίδα (Εικόνα 4.3) χωρίς να ξανασυμπληρώσει τα πεδία της σελίδας εισόδου.

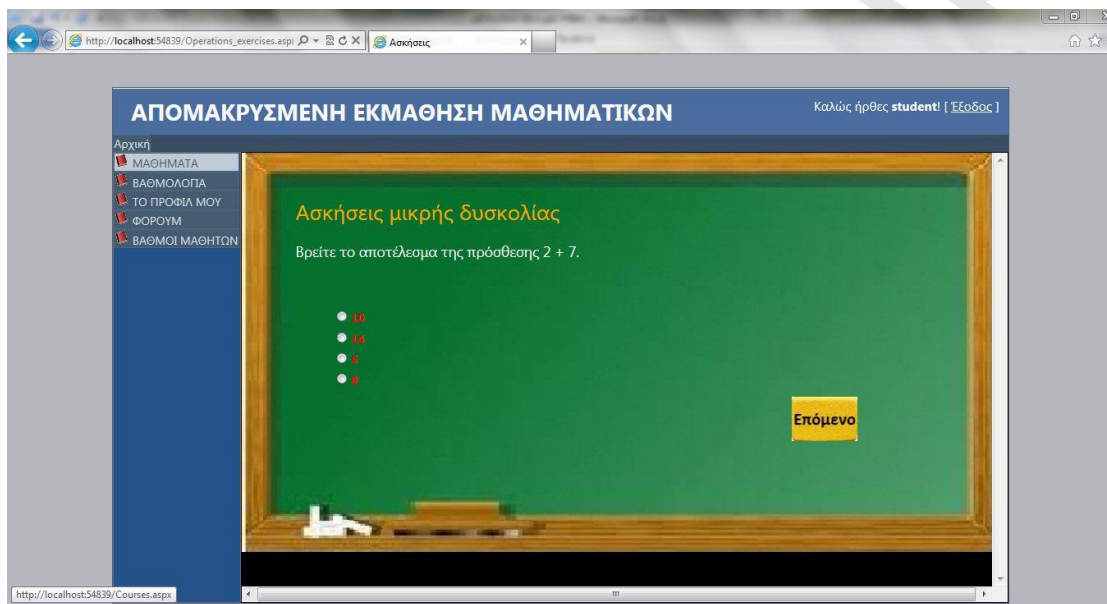


Εικόνα 4.3 Αρχική Σελίδα.

4.1 Επίπεδα χρηστών και είδη ασκήσεων

Όταν γίνεται μία εγγραφή στο σύστημα ο χρήστης μπαίνει αυτόματα στο επίπεδο του αρχάριου (Beginner). Κρίνεται απαραίτητο να μην χρησιμοποιηθούν ερωτηματολόγια κατάταξης των χρηστών σε ένα από τα 3 επίπεδα χρηστών (Αρχάριος, Μέτριος, Καλός), (Beginner, Intermediate, Good). Αυτό γιατί το κοινό που απευθύνεται η ιστοσελίδα είναι παιδιά μικρών τάξεων του δημοτικού και επομένως είναι σχεδόν αδύνατο να έχουν προηγούμενη γνώση σχετικά με τα μαθηματικά. Η μεταπήδηση των χρηστών από το ένα επίπεδο στο άλλο, γίνεται όταν ο μαθητής λύνει ασκήσεις και συγκεκριμένα είναι συνάρτηση του πλήθους των σωστών απαντήσεων και του πλήθους όλων των απαντήσεων. Συγκεκριμένα όταν το κλάσμα σωστές_απαντήσεις / συνολικές_απαντήσεις είναι μικρότερο του 60% τότε ο χρήστης βρίσκεται στο επίπεδο του Αρχάριου (Beginner). Αν το κλάσμα είναι μεταξύ 60% και 80% χωρίς το τελευταίο τότε ο χρήστης ανήκει στο επίπεδο του Μέτριου. Τέλος, αν ο χρήστης έχει κλάσμα μεγαλύτερο ή ίσο του 80% τότε αυτόματα μπαίνει στο επίπεδο του Καλού (Good). Αξίζει να επισημανθεί ότι δεν πρέπει να συγχέεται το επίπεδο του χρήστη με το αποτέλεσμα που έφερε σε μία άσκηση. Για παράδειγμα, αν κάποιος χρήστης είχε προσπαθήσει 3 φορές πριν να φέρει 60% και πάνω και δεν τα κατάφερε και στην τέταρτη φορά έφερε 70% δε σημαίνει ότι το επίπεδό του θα είναι Μέτριος, δηλαδή ότι αποτέλεσμα έφερε την τελευταία φορά. Συνυπολογίζονται εκεί και οι προηγούμενες αποτυχημένες προσπάθειές του. Έτσι, λοιπόν είναι πολύ πιο δύσκολο να αλλάξει το επίπεδο ενός χρήστη που έχει χρησιμοποιήσει το σύστημα αρκετές φορές. Το σύνολο των απαντήσεων και οι σωστές απαντήσεις κρατούνται αποθηκευμένες στη Βάση Δεδομένων.

Σε κάθε επίπεδο, κάθε χρήστης έχει συγκεκριμένο τρόπο εμφάνισης των ασκήσεων. Υπάρχουν 3 είδη ασκήσεων: i) οι εύκολες, ii) οι μέτριες και iii) οι δύσκολες. Οι εύκολες ασκήσεις (Εικόνα 4.1.1) είναι ασκήσεις πολλαπλής επιλογής και είναι το μόνο είδος ασκήσεων που μπορεί να επεξεργαστεί ο καθηγητής, δηλαδή να διαγράψει, να προσθέσει νέα ή να ενημερώσει. Οι μέτριες και οι δύσκολες ασκήσεις, είναι ασκήσεις που δεν παρεμβαίνει ο καθηγητής, αλλά μπαίνουν τυχαία από το σύστημα. Συγκεκριμένα, οι μέτριες (Εικόνα 4.1.2) ασκήσεις είναι ασκήσεις συμπλήρωσης κενού. Οι δύσκολες ασκήσεις (Εικόνα 4.1.3) είναι ασκήσεις πολλαπλής συμπλήρωσης κενού. Ο αρχάριος χρήστης λύνει ασκήσεις και των 3 ειδών, πρώτα τις εύκολες που είναι 5 στον αριθμό, μετά τις μέτριες που είναι και αυτές 5 και τέλος τις δύσκολες που και αυτές είναι 5. Συνολικά δηλαδή ο αρχάριος απαντά σε 15 ερωτήσεις. Ο μέτριος χρήστης λύνει ασκήσεις 2 ειδών, μέτριες και δύσκολες, 5 ερωτήσεων σε κάθε είδος, άρα συνολικά επιλύει 10 απαντήσεις και ο καλός χρήστης λύνει μόνο δύσκολες ασκήσεις 5 τον αριθμό. Όπως καταλαβαίνουμε αυτό είναι εύλογο μιας και ο αρχάριος χρήστης πρέπει να λύνει περισσότερες ασκήσεις και μάλιστα από το κατώτερο επίπεδο σταδιακά προς το ανώτερο. Ο καλός χρήστης δε χρειάζεται να λύνει τις ασκήσεις όλων των επιπέδων μιας και αν έλυσε και εύκολες ασκήσεις θα είχε ως αποτέλεσμα να του προκαλέσει δυσανασχέτηση ως προς τον τρόπο που τον αντιμετωπίζει το σύστημα. Δηλαδή θα ένιωθε σαν να τον αντιμετωπίζει όπως και τους αρχάριους μαθητές. Επομένως, το σύστημα κάνει σαφή διαχωρισμό των χρηστών, μοντελοποιώντας τους ανάλογα με το επίπεδό τους και κάνοντας έτσι καλύτερη διαχείριση ως προς την καθοδήγησή τους.



Εικόνα 4.1.1. Εύκολες ασκήσεις.



Εικόνα 4.1.2. Μέτριες ασκήσεις.



Εικόνα 4.1.3. Δύσκολες ασκήσεις.

4.2 Ο αλγόριθμος

Η παράγραφος αυτή θα μπορούσαμε να πούμε ότι έχει και τη μεγαλύτερη βαρύτητα σε σχέση με τα υπόλοιπα και αποτελεί τον πυρήνα της ιστοσελίδας μας. Θα αναφερθούμε στον αλγόριθμο ο οποίος έχει υλοποιηθεί για να γίνει η βαθμολόγηση των χρηστών, καθώς και η καταγραφή των ειδών λαθών. Όταν ένας χρήστης μπαίνει στη σελίδα των μαθημάτων Operations.aspx τότε προσπαθεί να επιλύσει διάφορα είδη ασκήσεων, ανάλογα πάντα με το επίπεδο του όπως αναφέραμε σε προηγούμενη παράγραφο.

4.2.1 Είδη λαθών

Το σύστημα μπορεί να καταγράψει συνολικά 8 είδη λαθών και αυτά έχουν ως εξής:

- Σύγχυση μεταξύ πρόσθεσης και αφαίρεσης
- Σύγχυση μεταξύ πρόσθεσης και πολλαπλασιασμού
- Σύγχυση μεταξύ πρόσθεσης και διαίρεσης
- Σύγχυση μεταξύ αφαίρεσης και πολλαπλασιασμού
- Σύγχυση μεταξύ αφαίρεσης και διαίρεσης
- Σύγχυση μεταξύ πολλαπλασιασμού και διαίρεσης
- Λάθη κρατουμένου
- Άσχετα λάθη (Σοβαρά λάθη)

Εξηγούμε τι σημαίνει το καθένα. Για παράδειγμα έστω ότι ο χρήστης θέλει να προσθέσει $5 + 4$ και δίνει αποτέλεσμα 1, τότε το σύστημα τον ενημερώνει ότι έχει μπερδέψει την πρόσθεση με την αφαίρεση. Αντίστοιχα, αν δώσει 20, θα του πει ότι το μπερδέψε με τον πολλαπλασιασμό. Το ίδιο ισχύει και για τα άλλα είδη συγχύσεων. Τα λάθη κρατουμένου εμφανίζονται στις δύσκολες

ασκήσεις που υπάρχει πολλαπλή συμπλήρωση κενού. Έστω για παράδειγμα ότι προστίθενται οι αριθμοί 183 και 286 (Εικόνα 4.2.1).

Κρατούμενα	<input type="text"/>	<input type="text"/>
	1	8 3
	+	2 8 6
	<input type="text"/>	<input type="text"/>

Εικόνα 4.2.1 Δύσκολη άσκηση.

Έστω ότι ο χρήστης προσθέτει σωστά το 6 και το 3 και βάζει στο δεξί κάτω κουτάκι 9 και στο δεξί πάνω 0, μιας και δεν υπάρχει κρατούμενο. Έστω ότι βάζει σωστά στο δεύτερο από δεξιά κάτω κουτάκι 6 και κρατούμενο στο αριστερά πάνω 1, αφού $8 + 8 = 16$. Αν στα 2 τελευταία κουτάκια βάλει 0 αριστερά και 3 δεξιά αντί για 4 που είναι το σωστό, τότε έχει γίνει λάθος κρατουμένου (Εικόνα 4.2.2).

Κρατούμενα	1	0
	1	8 3
	+	2 8 6
	0	3 6 9

Εικόνα 4.2.2. Λάθος κρατουμένου.

Τα άσχετα λάθη είναι σοβαρά λάθη τα οποία δεν προέρχονται από σύγχυση κάποιας πράξης με κάποια άλλη, ή είναι κάποιο λάθος κρατουμένου. Δηλαδή, για παράδειγμα έστω ότι ο χρήστης στην πρόσθεση $5 + 3$ ο χρήστης δίνει 7. Το 7 δεν είναι ούτε γινόμενο, ούτε υπόλοιπο, ούτε διαφορά των 5 και 3 για να συγχυστεί με κάποια άλλη πράξη η πρόσθεση.

Στη βάση δεδομένων μας τα λάθη αυτά είναι καταγεγραμμένα στον πίνακα Results.dbo. Ο πίνακας αυτός περιλαμβάνει κάποιες στήλες που είναι οι εξής:

- username
- total_errors
- total_answers
- sum-sub_error_index
- sum-mult_error_index
- sum-div_error_index

- sub-mult_error_index
- sub-dib_error_index
- mult-div_error_index
- curry-error_index
- serious_error_index
- sum-sub_errors
- sum-mult_errors
- sum-div_errors
- sub-mult_errors
- sub-dib_errors
- mult-div_errors
- curry-errors
- serious_errors
- addition_total_rate
- subtraction_total_rate
- multiplication_total_rate
- division_total_rate

Το username είναι το όνομα του χρήστη και τα total_errors, total_answers είναι τα συνολικά λάθη και συνολικές απαντήσεις αντίστοιχα. Τα addition_total_rate, subtraction_total_rate, multiplication_total_rate, division_total_rate είναι οι βαθμοί που πήραν οι χρήστες όταν εξετάστηκαν αντίστοιχα στην πρόσθεση, την αφαίρεση, τον πολλαπλασιασμό και τη διαίρεση. Θα κάνουμε αναφορά σε επόμενη παράγραφο για αυτά. Όσον αφορά τα sum-sub, sum-mult, sum-div, sub-mult, sub-div, mult-div, curry και serious error indexes αυτά βοηθούν ουσιαστικά το σύστημα στον τρόπο βαθμολόγησης. Το ίδιο και τα αντίστοιχα τους errors. Έστω ότι στο σύστημα μπαίνει ένας αρχάριος. Τότε αυτός θα λύσει 5 εύκολες, 5 μέτριες και 5 δύσκολες ασκήσεις. Σε κάθε άσκηση όταν ο χρήστης πάρει το 100% από την άσκηση θα προσθέσει μία μονάδα στο μερικό άθροισμα. Ο βαθμός που θα προκύψει από τις συνολικά 15 απαντήσεις θα είναι το μερικό αυτό άθροισμα διὰ του 15 επί 100 %. Όταν σε μία άσκηση στο κεφάλαιο της πρόσθεσης, έστω σε μέτρια, όπου υπάρχει απλή συμπλήρωση κενού το σύστημα εντοπίζει ότι έχει γίνει σύγχυση μεταξύ πρόσθεσης και πολλαπλασιασμού, το σύστημα θα εντοπίζει στη βάση δεδομένων του το αντίστοιχο index, στη συγκεκριμένη περίπτωση το sum-mult_error_index και στο μερικό άθροισμα θα προσθέσει την ποσότητα $1/(2 + \text{error_index})$, όπου εδώ το error_index μπορεί να είναι ένα από τα 9 error_indexes εκτός του serious_error_index. Μόνο όταν ο χρήστης απαντήσει σωστά θα πάρει άσσο και αν απαντήσει λάθος θα πάρει μηδέν σαν μερικό άθροισμα. Επομένως, όταν ο χρήστης συγχέει μία πράξη με μία άλλη δεν χάνει όλη την ερώτηση, δηλαδή δεν παίρνει μηδέν, αλλά ένα ποσοστό του άσσου. Επιπρόσθετα, το σύστημα θα αυξήσει το αντίστοιχο error_index, εδώ το sum-mult_error_index και το αντίστοιχο error, εδώ το sum-mult_error κατά ένα. Γιατί, όμως χρησιμοποιούνται 2 είδη error για κάθε μία σύγχυση πράξης με άλλη και για τα λάθη κρατουμένου και για τα άσχετα λάθη; Η απάντηση έγκειται στο εξής. Στη βαθμολόγηση χρησιμοποιούνται τα error_indexes, διότι έστω ότι έχει γίνει κάποιος αριθμός λαθών μέχρι κάποιο σημείο για ένα χρήστη. Όταν ο χρήστης απαντήσει σωστά στην επόμενη ερώτηση τότε τα αντίστοιχα error_indexes μειώνονται κατά ένα και μπορούν να πάρουν μέχρι και την τιμή μηδέν. Τα αντίστοιχα errors σε σωστή απάντηση θα παραμείνουν αναλλοίωτα. Σε λάθος απάντηση όμως θα αυξηθούν και τα 2 κατά ένα. Αυτή η χρήση των error_indexes έγκειται στο γεγονός ότι, όταν ένας χρήστης αρχίζει και απαντά σωστά σε κάποιες συνεχόμενες ερωτήσεις, τότε στο επόμενο λάθος που θα κάνει θα πρέπει να έχει μεγαλύτερο μερικό άθροισμα, από το μερικό άθροισμα που είχε την τελευταία φορά που έκανε λάθος. Και αυτό διότι είχε αρκετές σωστές απαντήσεις ενδιάμεσα. Επομένως, οι σωστές απαντήσεις επηρεάζουν θετικά τα error_indexes. Τα errors δεν τα επηρεάζει. Που

χρησιμεύουν όμως τα errors; Τα errors χρησιμεύουν μόνο και μόνο για να είναι αποθηκευμένο ένα αρχείο λαθών για την παρουσίασή τους στο χρήστη στη βαθμολογία του, αλλά και για την επιλογή από το σύστημα των διφορούμενων λαθών που πιθανόν συμβαίνουν σε άσκηση πολλαπλής συμπλήρωσης κενού, που θα αναφέρουμε σε επόμενη παράγραφο. Επομένως, ο τύπος για το μερικό άθροισμα σε μία άσκηση είναι ο εξής:

$$\sigma_i = \begin{cases} \frac{1}{1 + \text{error_index}_i} & , \text{αν είναι λάθος.} \\ 0 & , \text{αν είναι άσχετο λάθος.} \\ 1 & , \text{αν είναι σωστό.} \end{cases}$$

error_index_i : είδος λάθους
mark: βαθμός κεφαλαίου
σ_i : μερικό άθροισμα εύκολης και μέτριας άσκησης
n: πλήθος ασκήσεων

$$\text{mark} = \sum_{i=1}^n \sigma_i / n$$

Να σημειωθεί ότι το n για τους αρχάριους είναι 15, για τους μέτριους είναι 10 και για τους καλούς είναι 5.

4.2.2 Πολλαπλά είδη λαθών

Τα πολλαπλά είδη λαθών αφορούν ασκήσεις μεγάλης δυσκολίας. Όταν αναφερόμαστε σε πολλαπλά είδη λαθών αυτό σημαίνει ότι σε κάθε φάση μιας άσκησης μπορεί να έχουν γίνει 2 ή περισσότερα λάθη (το πολύ 3). Όταν λέμε φάση μιας άσκησης εννοούμε το εξής. Χωρίς βλάβη της γενικότητας στην πράξη της πρόσθεσης και της αφαίρεσης έχουμε πάρει 2 τριψήφιους αριθμούς. Στην πράξη του πολλαπλασιασμού έχουμε έναν τριψήφιο με έναν μονοψήφιο αριθμό. Σε αυτές τις 3 πράξεις κάθε άσκηση χωρίζεται σε 3 φάσεις. Έστω στην πρόσθεση (Εικόνα 4.2.1.1), η πρώτη φάση της πράξης εξετάζει το αποτέλεσμα των 2 δεξιά ψηφίων των αριθμών (κάτω δεξιά κουτάκι) καθώς και το κρατούμενό του (πάνω δεξιά κουτάκι). Η δεύτερη φάση που έχει και περισσότερη επεξεργασία, υπολογίζει το κρατούμενο στο δεξί πάνω κουτάκι που έδωσε η προηγούμενη μερική πρόσθεση, υπολογίζει το αποτέλεσμα που δίνουν οι 2 μεσαίοι αριθμοί και το κρατούμενο (δεύτερο από δεξιά κάτω κουτάκι) και βρίσκει και το κρατούμενο στο αριστερά πάνω κουτάκι. Η τρίτη φάση υπολογίζει το αποτέλεσμα των τελευταίων ψηφίων στα 2 αριστερά κάτω κουτάκια (καθώς η πρόσθεση θα δώσει το πολύ τετραψήφιο αποτέλεσμα) και υπολογίζει σε αυτό και το πάνω αριστερά κρατούμενο. Στην αφαίρεση (Εικόνα 4.2.1.2) γίνεται η ίδια διαδικασία, μόνο που στην τρίτη φάση το αποτέλεσμα τοποθετείται σε ένα κουτάκι και όχι σε 2 (γιατί σίγουρα θα δώσει το πολύ διψήφιο αποτέλεσμα). Στον πολλαπλασιασμό (Εικόνα 4.2.1.3) οι φάσεις είναι όπως και στην πρόσθεση, μόνο που πολλαπλασιάζονται τα ψηφία του πρώτου αριθμού με τον μονοψήφιο (ένα ψηφίο του πρώτου αριθμού σε κάθε φάση).



Εικόνα 4.2.2.1 Φάσεις πρόσθεσης.



Εικόνα 4.2.2.2 Φάσεις αφαίρεσης.



Εικόνα 4.2.2.3 Φάσεις πολλαπλασιασμού.

Στη διαίρεση έχουμε 2 ειδών φάσεις. Καταρχάς, χωρίς βλάβη της γενικότητας έχουμε επιλέξει να διαιρούνται ένας διψήφιος με ένα μονοψήφιο. Όταν ο διαιρέτης χωρά στο πρώτο ψηφίο του διαιρετέου, τότε έχουμε 7 φάσεις για επεξεργασία. Στην πρώτη φάση το σύστημα τσεκάρει αν το πρώτο ψηφίο του πηλίκου είναι το σωστό, δηλαδή αν ο διαιρέτης χωρά στο πρώτο ψηφίο του διαιρετέου όσο λέει το πρώτο ψηφίο του πηλίκου. Στη δεύτερη φάση το σύστημα τσεκάρει αν το γινόμενο είναι το σωστό σε σχέση με το πρώτο από αριστερά κουτάκι της πρώτης σειράς text boxes. Στην τρίτη φάση το σύστημα ελέγχει αν η διαφορά του πρώτου ψηφίου του διαιρετέου από το αποτέλεσμα του προηγούμενου πολλαπλασιασμού είναι ίδια με το αποτέλεσμα του αριστερού κουτιού της δεύτερης σειράς text boxes. Στην τέταρτη φάση το σύστημα ελέγχει αν το δεύτερο ψηφίο του διαιρετέου είναι ίδιο με το δεξί κουτί της δεύτερης σειράς text boxes, δηλαδή αυτό που «κατεβάζει» ο χρήστης. Στην πέμπτη φάση το σύστημα τσεκάρει αν το δεύτερο ψηφίο του πηλίκου είναι ίδιο με τον αριθμό των φερών που χωρά ο διαιρέτης στον διψήφιο αριθμό της δεύτερης σειράς text boxes. Στην έκτη φάση το σύστημα τσεκάρει αν ο πολλαπλασιασμός του δεύτερου ψηφίου του πηλίκου με το διαιρέτη είναι ίδιος με τον διψήφιο (ή μονοψήφιο) που προκύπτει από την τρίτη σειρά text boxes. Τέλος, στην έβδομη φάση ελέγχεται αν η διαφορά της τρίτης σειράς από τη δεύτερη σειρά είναι ίδια με το τελικό υπόλοιπο που προκύπτει. Όλα αυτά φαίνονται στην παρακάτω εικόνα (Εικόνα 4.2.1.4).



Εικόνα 4.2.2.4. Φάσεις διαίρεσης όταν ο διαιρέτης χωρά στο πρώτο ψηφίο από αριστερά του διαιρετέου.

Αντίστοιχα, όταν ο διαιρέτης δε χωρά στο πρώτο ψηφίο του διαιρετέου, αλλά σε όλο το διαιρετέο έχουμε 3 φάσεις. Στην πρώτη φάση το σύστημα ελέγχει αν το πηλίκο είναι ίδιο με τον αριθμό των φορών που χωρά ο διαιρέτης στον διαιρετέο. Στη δεύτερη φάση το σύστημα τσεκάρει αν ο πολλαπλασιασμός πηλίκου με διαιρέτη είναι ίδιος με το αποτέλεσμα που τοποθετείται στα 2 κουτάκια της πρώτης σειράς text boxes. Στην τρίτη φάση το σύστημα ελέγχει αν η διαφορά του αποτελέσματος του προηγούμενου πολλαπλασιασμού με το διαιρετέο είναι το ίδιο με το τελευταίο κουτάκι, δηλαδή το υπόλοιπο (Εικόνα 4.2.1.5).



Εικόνα 4.2.2.5 Φάσεις διαίρεσης όταν ο διαιρέτης δε χωρά στο πρώτο ψηφίο από αριστερά του διαιρετέου.

Όταν τώρα προκύπτει πολλαπλό λάθος σε μία φάση αυτό σημαίνει ότι έγινε σύγχυση μιας πράξης με μία άλλη, σε συνδυασμό με λάθος κρατουμένου. Στα πολλαπλά λάθη πάντα το ένα

λάθος αφορά λάθος κρατούμενου. Παράδειγμα, έστω ότι προστίθεται το 789 με το 456. Στα πρώτα ζεύγη αριθμών 9 και 6 έστω ότι ο χρήστης δε βάζει το σωστό αποτέλεσμα που θα ήταν 5 στο κουτάκι των αποτελεσμάτων και 1 σαν κρατούμενο και βάζει 4 σαν αποτέλεσμα και μηδέν σαν κρατούμενο. Το σύστημα αυτόματα θα καταλάβει ότι ο χρήστης εννοούσε ότι ήθελε να κάνει πολλαπλασιασμό του 6 με το 9, δηλαδή 54 και ταυτόχρονα ξέχασε να βάλει κρατούμενο το 4 και έβαλε μηδέν. Επομένως, ο χρήστης έκανε 2πλό λάθος, δηλαδή υπήρξε σύγχυση πρόσθεσης-πολλαπλασιασμού και ταυτόχρονα λάθος κρατούμενου. Έστω σε ένα άλλο παράδειγμα ότι ο χρήστης προσθέτει πάλι το 789 με το 456 και στην αρχή βάζει 5 στο κουτάκι αποτελεσμάτων (κάτω δεξιά) και 1 σαν κρατούμενο (πάνω δεξιά). Στη δεύτερη φάση όμως αντί να βάλει το σωστό που θα ήταν 4 στο κουτάκι αποτελεσμάτων και 1 σαν κρατούμενο ($8 + 5 + 1 = 14$), να βάλει 0 στο κουτάκι αποτελεσμάτων και 0 και στο κρατούμενο. Το σύστημα αυτόματα θα καταλάβει ότι έγινε τριπλό λάθος ως εξής. Θα καταλάβει ότι πολλαπλασίασε το 5 με το 8 και βρήκε 40, επομένως έβαλε το 0 στο κουτάκι αποτελεσμάτων και δεν υπολόγισε το κρατούμενο από πριν και ταυτόχρονα έβαλε και 0 στο επόμενο κρατούμενο, ενώ θα έπρεπε να βάλει το 4. Επομένως, έκανε 2 λάθη κρατουμένου και ένα λάθος σύγχυσης πρόσθεσης-πολλαπλασιασμού. Να σημειωθεί ότι τριπλά λάθη μπορούν να γίνουν μόνο στη δεύτερη φάση της πρόσθεσης, αφαίρεσης και πολλαπλασιασμού και όταν γίνονται αυτά τα 2 θα αφορούν λάθη κρατουμένου, ενώ το εναπομείναν θα είναι λάθος σύγχυσης πράξεων. Στο μερικό άθροισμα θα προστεθεί το λάθος με τη μεγαλύτερη συχνότητα. Δηλαδή, έστω στο τριπλό λάθος που αναφέραμε πριν, θα προστεθεί στο μερικό άθροισμα της βαθμολογίας το λάθος με τη μεγαλύτερη συχνότητα. Δηλαδή, αν η συχνότητα λαθών κρατουμένου είναι μεγαλύτερη από τη συχνότητα λαθών σύγχυσης πρόσθεσης-πολλαπλασιασμού, στο μερικό άθροισμα θα συμβάλει μόνο η μεριά των λαθών κρατουμένου. Όταν γίνεται ένα άσχετο λάθος σε μία φάση τότε δεν είναι δυνατό αυτό να είναι μέρος πολλαπλού λάθους. Τα άσχετα λάθη είναι απλά λάθη. Δε συμπεριλαμβάνονται σε πολλαπλά λάθη. Παρακάτω φαίνεται ένας πίνακας σχετικά με τη βαθμολογία στις δύσκολες ασκήσεις και συγκεκριμένα τον υπολογισμό των μερικών αθροισμάτων των δύσκολων ασκήσεων.

$$\varphi_i = \begin{cases} \min_{j=1}^m \left\{ \frac{1}{1 + \text{error_index}_j} \right\} & , \text{αν είναι λάθος.} \\ 0 & , \text{αν είναι άσχετο λάθος.} \\ 1 & , \text{αν είναι σωστό.} \end{cases}$$

$$\sigma_i = \sum_{j=1}^n \varphi_j / n$$

m : πλήθος πολλαπλών λαθών.

error_index_i : οι δείκτες πολλαπλών λαθών της φάσης i .

φ_i : το μερικό άθροισμα της φάσης i στις δύσκολες ασκήσεις.

n : το πλήθος των φάσεων. Για την πρόσθεση την αφαίρεση, τον πολλαπλασιασμό και τη διαίρεση, όταν ο διαιρέτης δε χωρά στο πρώτο ψηφίο του διαιρετέου είναι 3. Για τη διαίρεση όταν ο διαιρέτης χωρά στο πρώτο ψηφίο του διαιρετέου είναι 7.

σ_i : το μερικό άθροισμα της δύσκολης άσκησης.

4.2.3 Διφορούμενα λάθη

Ο αλγόριθμος εκτός από το γεγονός ότι ελέγχει για πολλαπλά λάθη σε κάθε φάση της πράξης είναι δυνατόν να βρει και διφορούμενα λάθη. Αξίζει να τονιστεί ιδιαίτερα ότι τα διφορούμενα λάθη σε μία φάση δεν είναι απαραίτητα 2 ή περισσότερα διαφορετικά είδη λαθών, αλλά και 2 ή περισσότερα διαφορετικά είδη πολλαπλών λαθών. Έστω για παράδειγμα ότι προστίθενται οι αριθμοί 687 με 458. Ο χρήστης αρχικά έστω ότι προσθέτει σωστά το 7 και το 8 στην πρώτη φάση και βάζει στο κουτάκι αποτελεσμάτων το 5 και στα κρατούμενα το 1. Ύστερα όμως στη δεύτερη φάση έστω ότι βάζει 3 στο κουτάκι αποτελεσμάτων και 0 στα κρατούμενα. Το σύστημα εδώ αντιμετωπίζει μία ασάφεια. Θα θεωρήσει ότι έκανε ένα διπλό λάθος κρατουμένου δηλαδή πρόσθεσε 8 και 5 χωρίς να υπολογίσει το κρατούμενο από πριν και ξεχνώντας και το κρατούμενο μετά ή θα θεωρήσει ότι έκανε διπλό λάθος με σύγχυση πρόσθεσης-αφαίρεσης και ένα λάθος κρατουμένου, ότι δηλαδή αφείρεσε το 5 από το 8 χωρίς πρώτα να προσθέσει το προηγούμενο κρατούμενο (το 1) στο 5; Σε αυτό το σημείο το σύστημα χρησιμοποιεί το ιστορικό των λαθών του χρήστη και συγκεκριμένα τα errors του (sum-sub_errors, sum-mult_errors, sum-div_errors, sub-mult_errors, sub-div_errors, mult-div_errors, curry-errors). Τα serious_errors, δε λαμβάνονται υπόψη στα διφορούμενα λάθη. Όταν το σύστημα εντοπίσει διφορούμενα λάθη, τότε επιλέγει τα λάθη με το μεγαλύτερο error_index. Αν μάλιστα πρόκειται για πολλαπλά λάθη σε ένα διφορούμενο λάθος τότε από τη μία πλευρά του διφορούμενου λάθους θα επιλεγεί το λάθος (απλό ή πολλαπλό) με τη μεγαλύτερη συχνότητα. Αν έχουν και τα 2 την ίδια συχνότητα, τότε ο αλγόριθμος επιλέγει τυχαία (randomly) ένα από τα διφορούμενα λάθη. Παρακάτω φαίνεται και ο συγκεντρωτικός πίνακας για τη βαθμολόγηση στις δύσκολες ασκήσεις.

$$\varphi_i = \begin{cases} 0 & \left\{ \prod_{p=1}^k \left\{ \prod_{w=1}^m \left\{ \frac{1}{1 + \text{error_index}_{w,j}} \right\} \right\} \right\}, \text{ αν είναι άσχετο λάθος.} \\ 1 & \text{, αν είναι σωστό.} \end{cases}$$

$$\sigma_i = \sum_{p=1}^n \varphi_i / n$$

m: πλήθος πολλαπλών λαθών.
w: δείκτης πολλαπλών λαθών.
k: πλήθος διφορούμενων λαθών.
j: δείκτης διφορούμενων λαθών.
i: δείκτης φάσης.
error_index_i: οι δείκτες πολλαπλών λαθών της φάσης i.
φ_i: το μερικό άθροισμα της φάσης i στις δύσκολες ασκήσεις.
n: το πλήθος των φάσεων. Για την πρόσθεση την αφαίρεση, τον πολλαπλασιασμό και τη διαίρεση, όταν ο διαιρέτης δε χωρά στο πρώτο ψηφίο του διαιρετέου είναι 3. Για τη διαίρεση όταν ο διαιρέτης χωρά στο πρώτο ψηφίο του διαιρετέου είναι 7.
σ_i: το μερικό άθροισμα της δύσκολης άσκησης.

4.3 Βάση Δεδομένων

Η Βάση Δεδομένων που υποστηρίζει την ιστοσελίδα είναι υλοποιημένη σε SQL Server 2012. Η Βάση έχει όνομα Mathimatika και περιέχει 6 πίνακες, τους: Comments.dbo, Easy_exercises.dbo, Rates.dbo, Results.dbo, Subjects.dbo, Xristis.dbo. Οι πίνακες και τα χαρακτηριστικά τους έχουν ως εξής:

- 1) Comments(comment_id, subject_id, comment, username):

Ο πίνακας αυτός περιέχει πληροφορίες για τα σχόλια που έκαναν οι χρήστες σε ένα θέμα που ανάρτησε κάποιος χρήστης.

- `comment_id`: Ο αύξων αριθμός του σχολίου.
- `subject_id`: Ο αύξων αριθμός του θέματος στο οποίο γίνεται το σχόλιο.
- `comment`: Το σχόλιο καθεαυτό.
- `username`: Το όνομα χρήστη ο οποίος έκανε το σχόλιο.

2) `Easy_exercises(exercise_id, chapter, description, answer1, answer2, answer3, answer4, correct_answer)`:

Ο πίνακας αυτός περιέχει τις εύκολες ασκήσεις.

- `exercise_id`: Ο αύξων αριθμός της άσκησης.
- `chapter`: Το κεφάλαιο στο οποίο ανήκει η άσκηση. Για τιμή 1 είναι εύκολη άσκηση πρόσθεσης, για 2 είναι αφαίρεσης, για 3 είναι πολλαπλασιασμού και για 4 είναι διαίρεσης.
- `description`: Η εκφώνηση της άσκησης.
- `answer1`: Η πρώτη πιθανή απάντηση.
- `answer2`: Η δεύτερη πιθανή απάντηση.
- `answer3`: Η τρίτη πιθανή απάντηση.
- `answer4`: Η τέταρτη πιθανή απάντηση.
- `correct_answer`: Η σωστή απάντηση.

3) `Rates(addition_last_rate, subtraction_last_rate, multiplication_last_rate, division_last_rate, username)`:

Ο πίνακας αυτός περιέχει τα σκορ που έχει ο κάθε χρήστης σε κάθε κεφάλαιο.

- `addition_last_rate`: Ο βαθμός τελευταίας προσπάθειας στην πρόσθεση.
- `subtraction_last_rate`: Ο βαθμός τελευταίας προσπάθειας στην αφαίρεση.
- `multiplication_last_rate`: Ο βαθμός τελευταίας προσπάθειας στον πολλαπλασιασμό.
- `division_last_rate`: Ο βαθμός τελευταίας προσπάθειας στη διαίρεση.
- `username`: Το όνομα χρήστη.

4) `Results(username, total_errors, total_answers, addition_total_rate, subtraction_total_rate, multiplication_total_rate, division_total_rate, sum-sub_error_index, sum-mult_error_index, sum-div_error_index, sub-mult_error_index, sub-div_error_index, mult-div_error_index, curry-error_index, serious_error_index, sum-sub_errors, sum-mult_errors, sum-div_errors, sub-mult_errors, sub-div_errors, mult-div_errors, curry-errors, serious_errors)`:

Ο πίνακας αυτός περιέχει τα είδη λαθών των χρηστών καθώς και τις βαθμολογίες τους σε όλα τα κεφάλαια, όπως και τα συνολικά λάθη και συνολικές απαντήσεις. Ο λόγος

που βάλαμε τα είδη λαθών errors με τους δείκτες λαθών error_indexes έχει ήδη σχολιαστεί επομένως δε θα κάνουμε αναφορά.

- username: όνομα χρήστη.
- total_errors: συνολικά λάθη.
- total_answers: συνολικές απαντήσεις.
- addition_total_rate: βαθμός στην πρόσθεση.
- subtraction_total_rate: βαθμός στην αφαίρεση.
- multiplication_total_rate: βαθμός στον πολλαπλασιασμό.
- division_total_rate: βαθμός στη διαίρεση.

Αυτό που πρέπει να τονίσουμε εδώ είναι γιατί να χρησιμοποιήσουμε τα total_rates για τις 4 πράξεις, ενώ χρησιμοποιούνται ήδη στον πίνακα Rates ως last_rates; Ο λόγος είναι ο εξής: Λόγω του γεγονότος ότι ένας χρήστης αν μπορεί να επιλύσει ένα κεφάλαιο έστω τη διαίρεση και προσπαθεί να επιλύσει τις ασκήσεις της πρόσθεσης και δεν περάσει το κεφάλαιο της πρόσθεσης, τότε τα αντίστοιχα total_rates τίθενται σε -1, ενώ τα last_rates μένουν αμετάβλητα. Πιο πολύ για προγραμματιστικούς λόγους λοιπόν προστέθηκαν αυτές οι στήλες του πίνακα.

5) Subjects(subject_id, subject, username):

Ο πίνακας αυτός περιέχει τα θέματα που προσθέτουν οι χρήστες στο forum.

- subject_id: Ο αύξων αριθμός του θέματος.
- subject: Το θέμα
- username: Το όνομα χρήστη.

6) Xristis(username, level, e-mail, role, password, currentCourse)

Ο πίνακας αυτός περιέχει το προφίλ του χρήστη.

- username: Το όνομα χρήστη.
- level: Το επίπεδο του χρήστη.
- e-mail: Το email του χρήστη.
- role: Ο ρόλος του χρήστη.
- password: Ο κωδικός του χρήστη.
- currentCourse: Το τρέχον μάθημα του χρήστη.

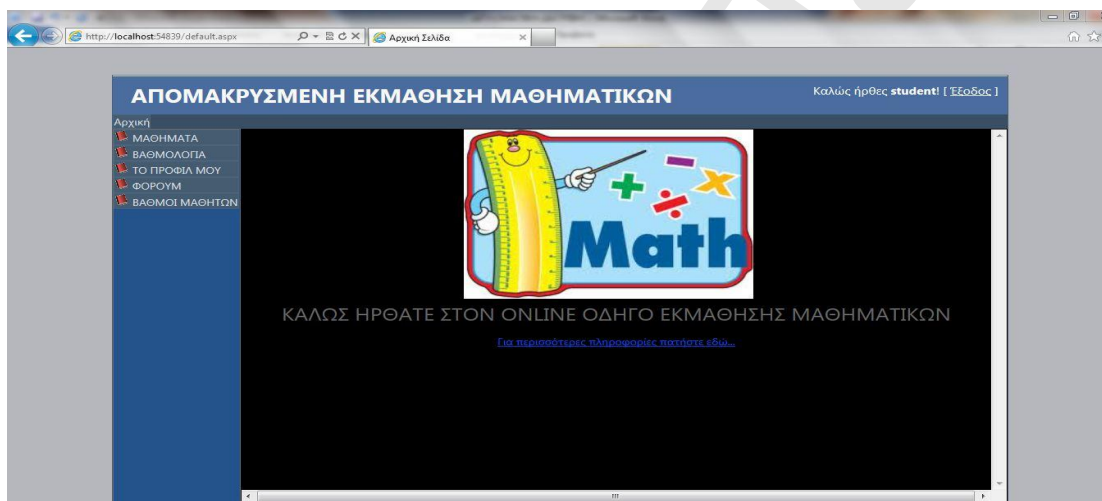
Αξίζει να σημειωθεί ότι το χαρακτηριστικό currentCourse, προστέθηκε για να δείξει ποια είναι η κατάσταση του χρήστη σχετικά με τα μαθήματα που έχει ξεκλειδώσει. Έτσι αν το currentCourse πάρει την τιμή "Subtraction", αυτό σημαίνει ότι ο χρήστης έχει ξεκλειδώσει μέχρι και τις ασκήσεις της αφαίρεσης και μπορεί να προχωρήσει στην επίλυσή τους.

4.4 Σενάρια Εκτέλεσης

Σε αυτό το κεφάλαιο θα παραθέσουμε ένα σενάριο εκτέλεσης και για την πλευρά του μαθητή και για την πλευρά του καθηγητή.

4.4.1 Σενάριο Εκτέλεσης χρήστη με ρόλο μαθητή

Στην παράγραφο αυτή θα κάνουμε εκτενή αναφορά στην πλοήγηση ενός μαθητή στην ιστοσελίδα που δημιουργήθηκε. Το κομμάτι της εγγραφής και την εισόδου αναλύθηκε σε προηγούμενη παράγραφο και επομένως θα ξεκινήσουμε όταν ένας υποτιθέμενος χρήστης με username “student” μπαίνει στην αρχική σελίδα μετά την είσοδό του (Εικόνα 4.4.1.1).

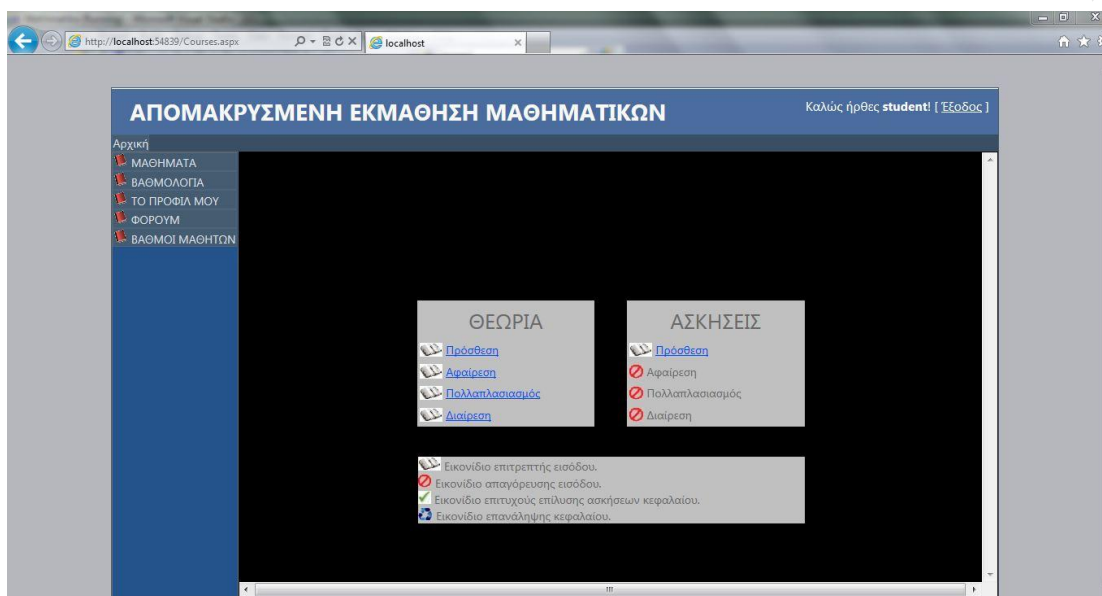


Εικόνα 4.4.1.1 Αρχική Σελίδα.





Όπως παρατηρούμε το σύστημα πάνω δεξιά καλωσορίζει το χρήστη και του δίνει τη δυνατότητα να αποσυνδεθεί, έτσι ώστε την επόμενη φορά που θα μπει κάποιος άλλος χρήστης από το τερματικό του student να μη μπει με το προφίλ του student. Αριστερά υπάρχει ένα μενού το οποίο βρίσκεται σε σταθερή θέση καθ' όλη την πλοήγηση του χρήστη μέσα στην ιστοσελίδα. Υπάρχουν οι εξής υπερσύνδεσμοι: «ΜΑΘΗΜΑΤΑ», «ΒΑΘΜΟΛΟΓΙΑ», «ΤΟ ΠΡΟΦΙΛ ΜΟΥ», «ΦΟΡΟΥΜ» και «ΒΑΘΜΟΣ ΜΑΘΗΤΩΝ». Οι σύνδεσμοι αυτοί οδηγούν στις εξής ιστοσελίδες αντίστοιχα: Courses.aspx, Marks.aspx, Profile.aspx, Forum.aspx, ViewMarks.aspx. Στο κέντρο περίπου της αρχικής βρίσκεται ένας σύνδεσμος που δίνει χρήσιμες πληροφορίες στο χρήστη σχετικά με τη λειτουργία της ιστοσελίδας και τις ενέργειες που μπορεί να κάνει.

Σελίδα Courses.aspx

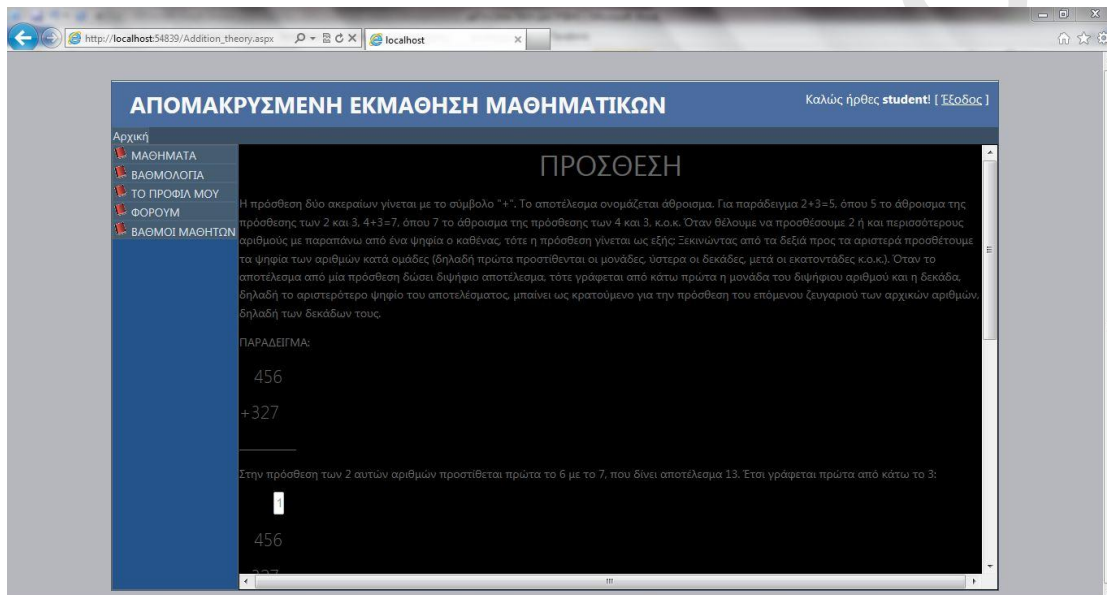
Όταν ο χρήστης μπαίνει στη σελίδα αυτή βλέπει 2 λίστες (Εικόνα 4.4.1.2). Η αριστερή λίστα περιέχει τη θεωρία για τις 4 βασικές πράξεις σε 4 διαφορετικούς συνδέσμους και δεξιά βρίσκονται οι ασκήσεις που μπορεί να λύσει ο μαθητής για τα αντίστοιχα κεφάλαια, δηλαδή της πρόσθεσης, της αφαίρεσης, του πολλαπλασιασμού και της διαίρεσης.



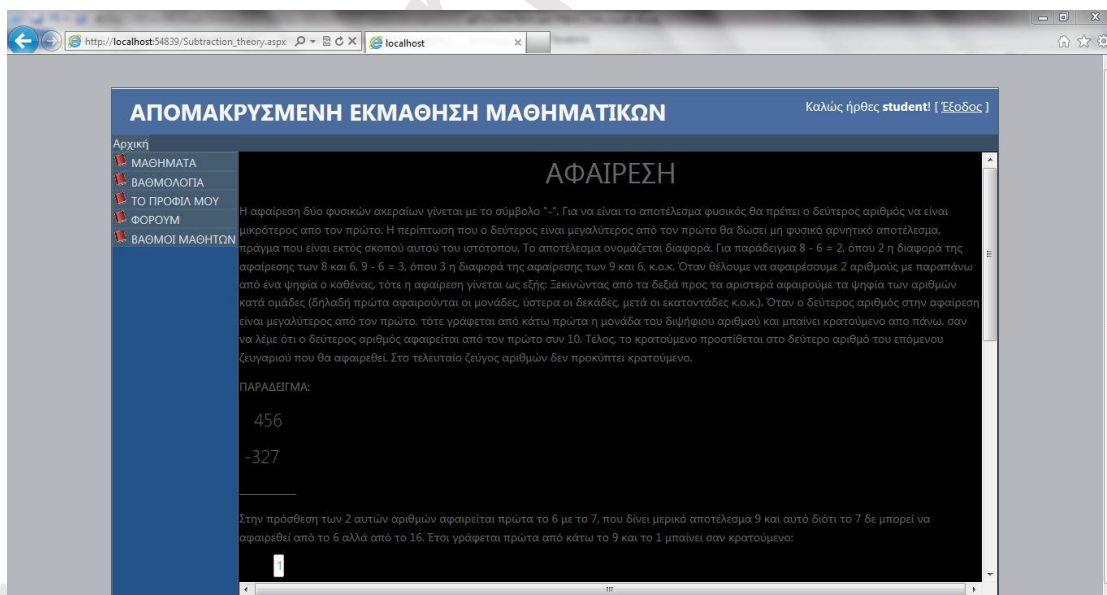
Εικόνα 4.4.1.2 Σελίδα μαθημάτων.

Όπως παρατηρούμε και στην εικόνα ο μαθητής έχει τη δυνατότητα να διαβάσει τη θεωρία από όλα τα κεφάλαια (πρόσθεσης, αφαίρεσης, πολλαπλασιασμού, διαίρεσης). Όμως, δε μπορεί να λύσει τις ασκήσεις όλων των κεφαλαίων. Υπάρχει μία ιεραρχία στην επίλυση ασκήσεων. Πρώτα θα λύσει τις ασκήσεις της πρόσθεσης, μετά της αφαίρεσης, μετά του πολλαπλασιασμού και τέλος της διαίρεσης. Αυτό πρακτικά σημαίνει ότι όσο προχωράμε από την πρόσθεση προς τη διαίρεση, τόσο πιο πολύ δυσκολεύουν τα κεφάλαια και ότι το ένα κεφάλαιο εξαρτάται από κάποια προηγούμενά του (Για παράδειγμα στην πράξη του πολλαπλασιασμού χρειάζεται η πρόσθεση και στην πράξη της διαίρεσης χρειάζεται η πράξη του πολλαπλασιασμού και της αφαίρεσης). Χρησιμοποιούνται χαρακτηριστικά εικονίδια για να δει ο μαθητής την παρούσα κατάστασή του αναφορικά με τις ασκήσεις. Συγκεκριμένα το εικονίδιο του ανοιχτού βιβλίου () υποδηλώνει ότι ο χρήστης είναι έτοιμος να λύσει τις ασκήσεις του κεφαλαίου αυτού. Το εικονίδιο της απαγόρευσης (), απαγορεύει στο μαθητή την επίλυση ασκήσεων αυτού του κεφαλαίου. Το εικονίδιο της επανάληψης () δείχνει στο μαθητή ότι δεν πέρασε ένα κεφάλαιο και πρέπει να κάνει επανάληψη τη θεωρία και να ξαναλύσει τις ασκήσεις του κεφαλαίου αυτού για να περάσει και να λύσει ασκήσεις επόμενων κεφαλαίων. Το εικονίδιο του τικαρίσματος () υποδηλώνει ότι ο μαθητής πέρασε το συγκεκριμένο κεφάλαιο. Όπως είναι φυσικό κάθε νέος χρήστης που μπαίνει στη σελίδα των μαθημάτων βλέπει ότι μπορεί να λύσει μόνο τις ασκήσεις της πρόσθεσης, δηλαδή του πρώτου κεφαλαίου. Τα υπόλοιπα κεφάλαια έχουν απαγορευτικό και επομένως ο χρήστης πρέπει να επιλύσει σωστά τις ασκήσεις της πρόσθεσης για να ξεκλειδώσει τις ασκήσεις της αφαίρεσης. Για να ξεκλειδωθεί ένα κεφάλαιο πρέπει να έχουν απαντηθεί σωστά, δηλαδή από 60% και πάνω, οι ασκήσεις των προηγούμενων κεφαλαίων. Αν για κάποιο λόγο ο μαθητής ενώ μπορεί ας πούμε για παράδειγμα να λύσει τις ασκήσεις της διαίρεσης και θέλει να κάνει αναβαθμολόγηση στην πρόσθεση και πάρει κάτω από 60% στην πρόσθεση, τότε αυτόματα του απαγορεύεται να λύσει τις ασκήσεις των κεφαλαίων μετά την πρόσθεση. Αυτό συμβαίνει υπό την έννοια ότι για να λύσει σωστά ένα κεφάλαιο ο μαθητής θα πρέπει να έχει λύσει σωστά και όλα τα υπόλοιπα κεφάλαια. Σε αυτό το σημείο επομένως το σύστημα κρίνεται λίγο αυστηρό για τους μαθητές. Όταν ο μαθητής πατήσει πάνω στη θεωρία κάποιου κεφαλαίου, τότε αυτόματα πλοηγείται στη θεωρία του αντίστοιχου κεφαλαίου. Στη σελίδα Addition_theory.aspx για την πρόσθεση, στη σελίδα Subtraction_theory.aspx για την

αφαίρεση, στη σελίδα Multiplication_theory.aspx για τον πολλαπλασιασμό και στη σελίδα Division_theory.aspx για τη διαίρεση (Εικόνες 4.4.1.3, 4.4.1.4, 4.4.1.5, 4.4.1.6).

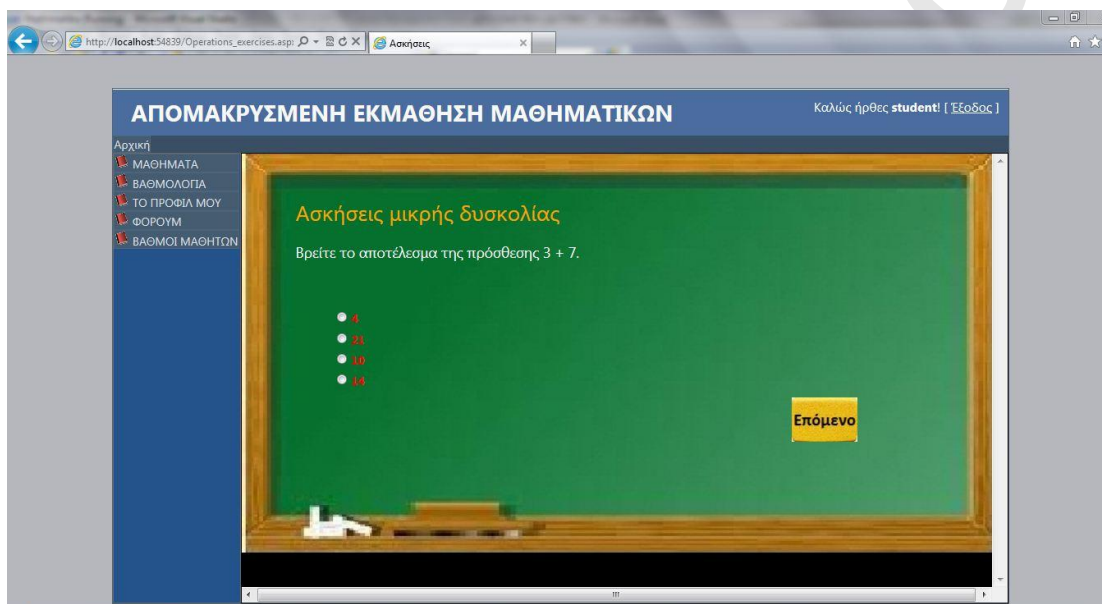


Εικόνα 4.4.1.3. Θεωρία πρόσθεσης.



Εικόνα 4.4.1.4. Θεωρία αφαίρεσης.

εύκολες, 5 μέτριες και 5 δύσκολες ασκήσεις. Αρχικά, επιλύει τις εύκολες ασκήσεις που όπως είπαμε είναι πολλαπλής επιλογής. Οι ασκήσεις αυτές έχουν προστεθεί από τον καθηγητή (θα αναφερθούμε λεπτομερέστερα στο σενάριο εκτέλεσης του καθηγητή) και είναι το μόνο είδος όπου παρεμβαίνει ο καθηγητής. Οι πολλαπλές απαντήσεις είναι κατάλληλα επιλεγμένες, έτσι ώστε ο μαθητής να μπερδεύεται περισσότερο και να υπάρχει πιθανότητα να μπλέξει μερικές πράξεις μεταξύ τους. Αυτό βέβαια είναι ανάλογο στην κρίση του καθηγητή, το πόσο θέλει να μπερδέψει το μαθητή. Πριν επιλύσει ο μαθητής τις ασκήσεις, όλα τα είδη λαθών και οι συνολικές απαντήσεις και τα συνολικά λάθη είναι μηδέν. Μόνο τα `total_rates` κάθε πράξης είναι `-1`, όπου το `-1` υποδηλώνει ότι ο χρήστης δεν έχει προσπελάσει το αντίστοιχο κεφάλαιο. Ο χρήστης περνά επομένως στην πρώτη ερώτηση (Εικόνα 4.4.1.7).

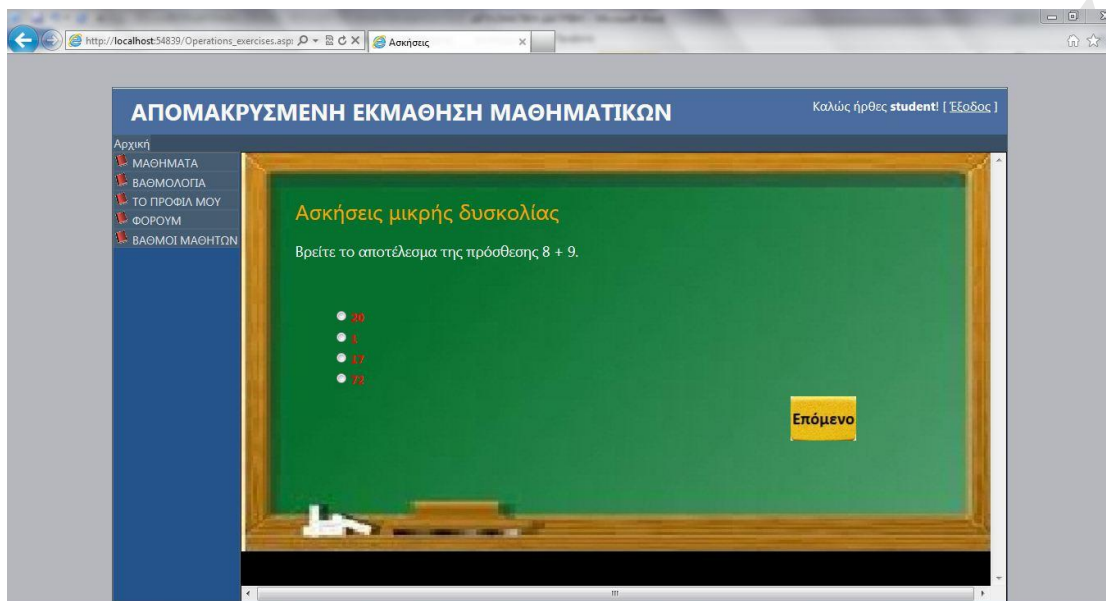


Εικόνα 4.4.1.7. Ο χρήστης βλέπει την πρώτη ερώτηση.

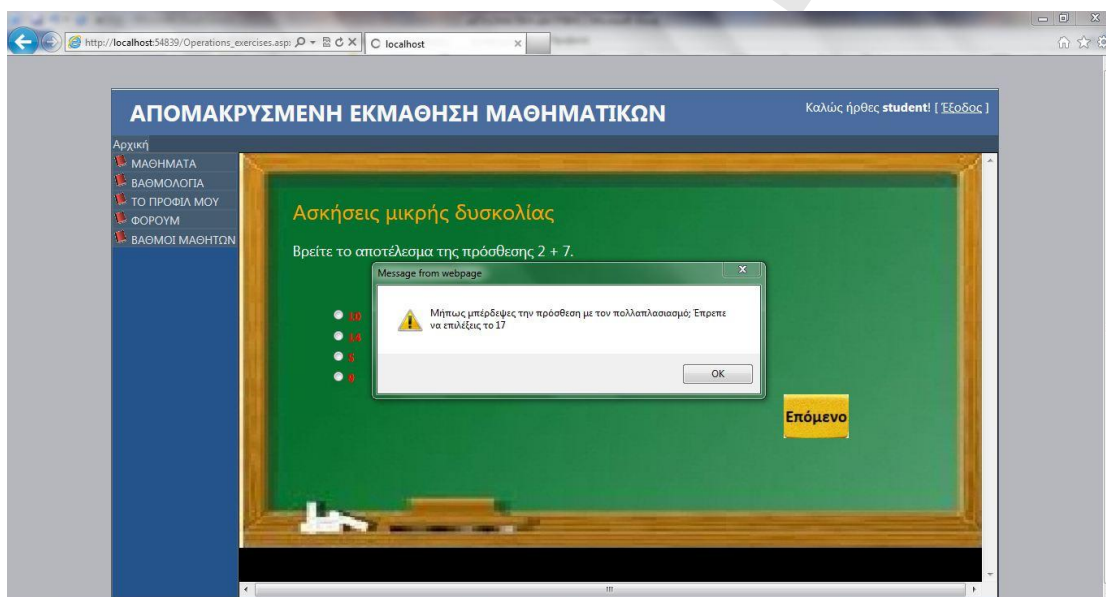
Το σύστημα τον ρωτάει πόσο κάνει $3 + 7$. Ο χρήστης έχει 4 πιθανές απαντήσεις: 4, 21, 10, 14. Τα αποτελέσματα αυτά είναι κατάλληλα επιλεγμένα, εφόσον το 4 είναι η διαφορά των 3 και 7, το 21 είναι το γινόμενό τους, το 10 η σωστή απάντηση και το 14 είναι μία άσχετη απάντηση. Σημειώνεται ότι καθ' όλη τη διάρκεια της εκτέλεσης υπάρχει μία μεταβλητή, η `result`, η οποία αποθηκεύει το σκορ του μαθητή, η οποία αρχικά είναι μηδέν. Έστω ότι ο χρήστης απαντά σωστά επομένως επιλέγει το 10. Τότε η `result` θα γίνει ίση με 1, οι συνολικές απαντήσεις θα είναι 1 και θα καταγραφούν στη βάση και τίποτα άλλο δε θα καταγραφεί στη βάση πλέον. Επομένως, το σύστημα τον προχωρά στην ερώτηση νούμερο 2. Αν ο χρήστης επιχειρήσει να πατήσει το κουμπί χωρίς να επιλέξει κάτι το σύστημα τον υποχρεώνει να πράξει ανάλογα.

Το σύστημα τον ρωτά πόσο κάνει $8 + 9$ (Εικόνα 4.4.1.8). Οι πιθανές απαντήσεις είναι 20, 1, 17, 72. Έστω ότι ο χρήστης απαντά 72. Τότε ένα σύνολο από ενέργειες γίνονται. Το σύστημα θα τον πληροφορήσει κατάλληλα ότι μπερδεψε την πρόσθεση με τον πολλαπλασιασμό και ότι θα έπρεπε να επιλέξει το 17 που είναι και η σωστή απάντηση (Εικόνα 4.4.1.9). Παράλληλα οι λάθος απαντήσεις θα γίνουν 1, οι συνολικές απαντήσεις θα γίνουν 2 και η εγγραφή στη βάση `sum-mult_error_index` και `sum-mult_errors` θα αυξηθούν κατά ένα, δηλαδή θα είναι ίσες με 1. Η μεταβλητή `result` τώρα που είναι και το πιο ενδιαφέρον σημείο, μέσω των τύπων που παρουσιάσαμε θα έχει ως εξής:

$result = result + 1/(2 + mult_error_index)$. Επομένως, θα γίνει ίση με $1 + 1/3$, δηλαδή $4/3$.



Εικόνα 4.4.1.8. Ο χρήστης θα επιλέξει την απάντηση 72.



Εικόνα 4.4.1.9. Μήνυμα που πληροφορεί το χρήστη πως έπρεπε να είχε πράξει.

Έπειτα προχωρά στην επόμενη ερώτηση, νούμερο 3. Εκεί το σύστημα τον ρωτά πόσο κάνει $2 + 7$. Έστω ότι απαντά σωστά το 9 ο χρήστης. Τότε γίνεται πάλι μία σειρά ενεργειών. Οι συνολικές απαντήσεις θα αυξηθούν κατά 1 (αυτό γίνεται σε κάθε ερώτηση), η μεταβλητή `results` θα αυξηθεί κατά ένα, δηλαδή θα γίνει $4/3 + 1 = 7/3$ και το πιο σημαντικό στη διαδικασία είναι ότι τα `error_indexes` όλων των συγχύσεων των πράξεων μεταξύ τους, όπως και τα `curry-error_index` και `serious_error_index` θα μειωθούν κατά ένα, μέχρι να γίνουν μηδέν. Αρνητικά αποτελέσματα δε μπορούν να πάρουν. Έτσι, εφόσον όλα τα υπόλοιπα `error_indexes` είναι μηδέν το `sum_mult_error_index` θα γίνει μηδέν από 1 που ήταν. Τα αντίστοιχα `errors` παραμένουν αμετάβλητα. Η λογική της μείωσης των `error_indexes` είναι όπως αναφέραμε και σε προηγούμενη παράγραφο, ότι όσο ο χρήστης δίνει σωστές απαντήσεις, το επόμενο λάθος που θα κάνει δε θα του στοιχίσει τόσο, όσο θα του στοιχίζε αν ήταν σε μία σειρά λαθών. Με μαθηματικό τρόπο αυτό σημαίνει πως το ποσοστό που θα πάρει από την ερώτηση στην πρώτη περίπτωση θα είναι

μεγαλύτερο από το ποσοστό που θα έπαιρνε στη δεύτερη. Αυτό είναι και μία λογική σκέψη από την άποψη του πώς θα αντιμετώπιζε ένας πραγματικός καθηγητής ένα μαθητή. Το δύσκολο είναι πώς αυτή η σκέψη μπορεί να γίνει μαθηματικό μοντέλο. Τις επόμενες 3 ερωτήσεις θα τις σχολιάσουμε σύντομα, μιας και δε γίνεται κάτι καινούργιο από πλευράς αλγορίθμου.

Η επόμενη ερώτηση ρωτά πόσο κάνει $4 + 5$. Έστω ότι ο μαθητής απαντά 12, αποτέλεσμα που του λέει ότι είναι εντελώς άσχετο με κάποια πιθανή σύγχυση με άλλη πράξη που θα μπορούσε να είχε κάνει. Επομένως, η `result` παραμένει ως έχει, δηλαδή $7/3$, αλλά οι συνολικές απαντήσεις αυξάνονται κατά ένα και γίνονται συνολικά 4. Η στήλη `total_errors` αυξάνει επίσης κατά ένα και γίνεται ίση με 2. Επίσης, αυξάνονται και οι εγγραφές τις βάσης δεδομένων κατά 1 και συγκεκριμένα, η `serious_error_index` και η `serious_errors` και θα γίνουν ίσες με 1. Το σύστημα τον συμβουλεύει ότι θα έπρεπε να είχε επιλέξει το 9 που είναι και το σωστό.

Η τελευταία ερώτηση του λέει πόσο κάνει $2 + 6$. Τότε ο χρήστης απαντά 3 και το σύστημα τον συμβουλεύει ότι έχει μπερδέψει την πρόσθεση με τη διαίρεση και ότι έπρεπε να είχε επιλέξει το 8 που είναι το σωστό. Η στήλη `total_errors` αυξάνει κατά ένα και γίνεται ίση με 3. Οι συνολικές απαντήσεις θα γίνουν 4 και οι στήλες `sum-div_error_index` και `sum-div_errors` θα αυξηθούν κατά ένα και θα γίνουν ένα και οι 2. Η `result` τώρα θα γίνει ίση με $7/3 + 1/3 = 8/3$. Συνοψίζοντας λοιπόν από τις εύκολες ερωτήσεις:

`result = 8/3`, `total_answers = 5`, `total_errors = 3`, `serious_error_index = 1`, `sum-div_error_index = 1`, `sum-mult_errors = 1`, `serious_errors = 1`, `sum-div_errors = 1`. Όλα τα άλλα `error_indexes` και `errors` είναι 0.

Ο χρήστης προχωρά στην επίλυση ασκήσεων μέτριας δυσκολίας. Στις ασκήσεις μέτριας δυσκολίας το σύστημα είναι αυτό που παράγει από μόνο του τους αριθμούς και δεν έχουν εισαχθεί από κάποιο καθηγητή. Οι αριθμοί είναι μονοψήφιοι. Το αποτέλεσμα μπορεί να είναι μέχρι και 2 ψηφία. Για αυτό το κενό `text box` μπορεί να δεχθεί μέχρι 2 χαρακτήρες. Σε αυτή τη φάση το σύστημα του ζητά να επιλέξει έναν αριθμό το πού διψήφιο ως αποτέλεσμα της πρόσθεσης. Συγκεκριμένα τον ρωτά πόσο κάνει $9 + 4$. Αν ο χρήστης δε δώσει καμία απάντηση το σύστημα τον υποχρεώνει να βάλει κάτι (Εικόνα 4.4.1.10). Συγκεκριμένα τον πληροφορεί ότι στο `text box` πρέπει να μπει οπωσδήποτε μία τιμή, είναι δηλαδή υποχρεωτικό πεδίο. Επίσης, αν δεν προσθέσει μία αριθμητική τιμή στο κενό πεδίο και βάλει ως πούμε οτιδήποτε άλλο (π.χ. ένα χαρακτήρα-γράμμα), το σύστημα ανάλογα θα τον προτρέψει να βάλει οπωσδήποτε μία αριθμητική τιμή (Εικόνα 4.4.1.11). Έστω ότι ο χρήστης δίνει την απάντησή του και βάζει 5 στο πεδίο. Τότε το σύστημα θα τον συμβουλέψει ότι έχει μπερδέψει την πρόσθεση με την αφαίρεση και ότι θα έπρεπε να είχε βάλει την τιμή 13. Επίσης, οι στήλες `sum-sub_error_index` και `sum-sub_errors` θα αυξηθούν κατά ένα και θα γίνουν ένα. Η `total_answers` θα γίνει 6, η `total_errors` θα γίνει 4 και η `results` θα γίνει $8/3 + 1/3 = 9/3 = 3$.



Εικόνα 4.4.1.10. Το σύστημα υποχρεώνει μέσω validation controls ότι πρέπει να συμπληρωθεί το κενό πεδίο.



Εικόνα 4.4.1.11. Το σύστημα προτρέπει το χρήστη να προσθέσει μία αριθμητική τιμή.

Η επόμενη ερώτηση λέει στο χρήστη πόσο κάνει $9 + 6$. Έστω ότι ο χρήστης βάζει την τιμή 54. Τότε το σύστημα θα του πει ότι μπέρδεψε την πρόσθεση με τον πολλαπλασιασμό και ότι θα έπρεπε να είχε βάλει την τιμή 15. Αυτόματα η `total_answers` γίνεται 7, η `total_errors` γίνεται 5, η `sum-mult_error_index` και η `sum-mult_errors` θα αυξηθούν κατά ένα και έτσι η `sum-mult_error_index` θα γίνει 1 και η `sum-mult_errors` θα γίνει ίση με 2. Επομένως, η μεταβλητή `result` θα γίνει ίση με $3 + 1/3 = 10/3$.

Η επόμενη ερώτηση του συστήματος ρωτά το χρήστη πόσο κάνει $0 + 8$. Έστω ότι ο χρήστης απαντά 0. Εδώ το σύστημα θα αυξήσει τη στήλη `total_errors` της βάσης κατά 1 και θα

γίνει 6 και η `total_answers` θα γίνει 8. Όμως, εδώ το σύστημα έχει να αντιμετωπίσει ένα διαφορετικό λάθος. Ο χρήστης μπέρδεψε πρόσθεση με πολλαπλασιασμό και βρήκε 0, επειδή $0 * 8 = 0$, ή μπέρδεψε την πρόσθεση με τη διαίρεση, εφόσον $0 / 8 = 0$; Η λύση δίδεται όταν το σύστημα τσεκάρει τα `errors` και βλέπει ότι η μεταβλητή `sum-mult_errors` που είναι 2 είναι μεγαλύτερη από τη `sum-div_errors` που είναι 1. Επομένως, θα επιλέξει ότι έγινε σύγχυση με τον πολλαπλασιασμό και η `sum-mult_errors` θα γίνει ίση με 3, ενώ η `sum-mult_error_index` θα γίνει ίση με 2. Έτσι το σύστημα θα συμβουλέψει το χρήστη ότι μπέρδεψε πρόσθεση με πολλαπλασιασμό και ότι έπρεπε να είχε βάλει την τιμή 8. Έτσι η μεταβλητή `result` θα γίνει ίση με $10/3 + 1/(2 + 2) = 43/12$.

Η επόμενη ερώτηση ρωτά το χρήστη πόσο κάνει $7 + 1$. Έστω ότι ο χρήστης απαντά και βάζει 6. Τότε το σύστημα του λέει ότι μπέρδεψε την πρόσθεση με την αφαίρεση και ότι έπρεπε να είχε βάλει την τιμή 8. Επομένως, η `total_answers` γίνεται ίση με 9, η `total_errors` ίση με 7 και η `sum-sub_error_index` όπως και η `sum-sub_errors` θα αυξηθούν κατά ένα, όπου και οι δύο γίνονται ίσες με 2. Η `result` θα γίνει ίση με $43/12 + 1/4 = 46/12$.

Η τελευταία ερώτηση από τις μέτριας δυσκολίας ασκήσεις λέει στο χρήστη πόσο κάνει $7 + 5$. Έστω ότι ο χρήστης απαντά σωστά το αποτέλεσμα 12, επομένως η `total_answers` θα γίνει ίση με 10, η `total_errors` θα παραμείνει ίδια, δηλαδή ίση με 7 και όλα τα `error_indexes` θα μειωθούν κατά ένα, δηλαδή η `sum-sub_error_index` θα γίνει ίση με 1, η `sum-mult_error_index` ίση με 1 και οι `sum-div_error_index` και `serious_error_index` θα μειωθούν κατά ένα και θα γίνουν 0. Τα υπόλοιπα `error_indexes` θα παραμείνουν 0 επειδή ήταν 0. Η `result` θα γίνει ίση με $46/12 + 1 = 58/12$.

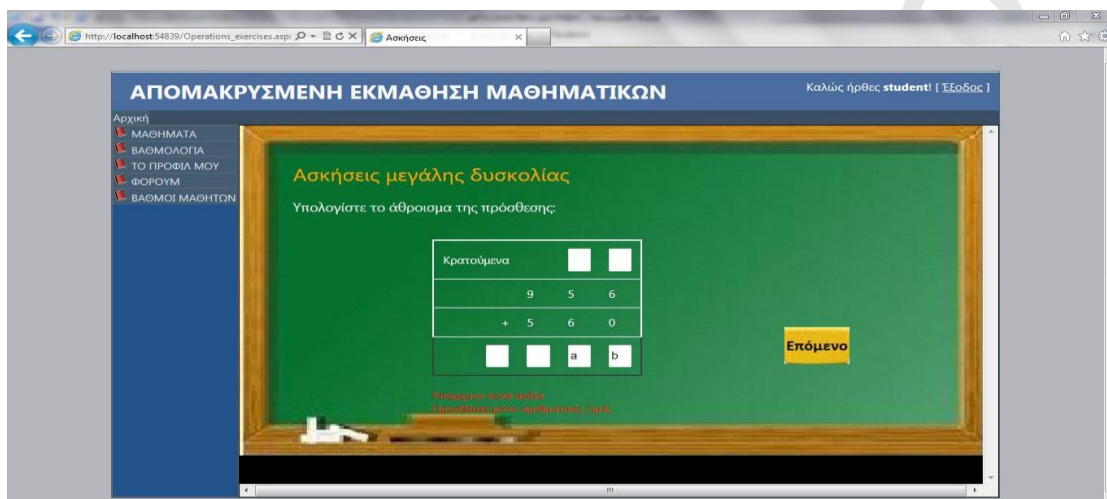
Επομένως στο σημείο αυτό έχουμε: `total_answers` = 10, `total_errors` = 7, `sum-mult_error_index` = 1, `sum-sub_error_index` = 1, `sum-mult_errors` = 3, `serious_errors` = 1, `sum-div_errors` = 1, `sum-sub_errors` = 2. Όλα τα υπόλοιπα `errors` και `error_indexes` είναι 0.

Από το σημείο αυτό και έπειτα ο χρήστης είναι έτοιμος να λύσει τις δύσκολες ασκήσεις. Οι δύσκολες ασκήσεις είναι ασκήσεις πολλαπλής συμπλήρωσης κενού και σαφώς είναι πιο δύσκολες στην επεξεργασία τους και επομένως και στον αλγόριθμο. Όταν ο χρήστης απαντήσει και την τελευταία μέτριας δυσκολίας ερώτηση προχωρά στις ασκήσεις μεγάλης δυσκολίας (Εικόνα 4.4.1.12).



Εικόνα 4.4.1.12. Ασκήσεις μεγάλης δυσκολίας.

Όπως παρατηρούμε ο χρήστης δεν πληκτρολογεί μόνο το αποτέλεσμα, αλλά πρέπει να γράψει και τα κρατούμενα που δίνει αυτή η πρόσθεση. Έτσι το σύστημα λαμβάνει σοβαρά υπόψη του όχι μόνο το αποτέλεσμα, αλλά και τα κρατούμενα που γεννήθηκαν. Συγκεκριμένα θα δούμε ότι το σύστημα είναι πιο ευέλικτο στον τρόπο βαθμολόγησης και δε βαθμολογεί το αποτέλεσμα καθεαυτό, όσο πιο πολύ τον τρόπο και τη συλλογιστική πορεία που ακολούθησε ο μαθητής για να δώσει τις απαντήσεις του. Όπως και στη απλή συμπλήρωση κενού, έτσι και στη πολλαπλή, αν κάποια πεδία είναι κενά ή δεν περιέχουν αριθμούς, το σύστημα τον πληροφορεί να πράξει ανάλογα (Εικόνα 4.4.1.13).

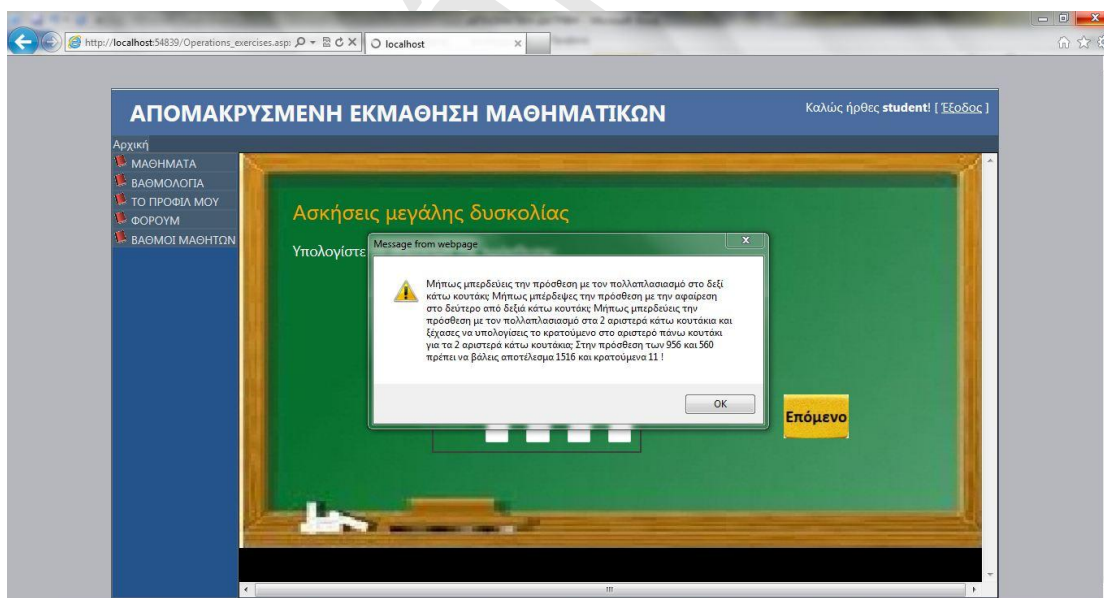


Εικόνα 4.4.1.13. Το σύστημα προτρέπει το χρήστη να συμπληρώσει όλα τα κενά πεδία και μάλιστα με αριθμητικές τιμές.

Όπως βλέπουμε οι αριθμοί που εξετάζονται είναι 3ψήφιοι. Το σύστημα όπως και στις μέτριες δυσκολίας ασκήσεις παράγει αυτούς τους αριθμούς τυχαία, δεν τους εισάγουν οι καθηγητές. Ο πίνακας κρατουμένων αφορά μόνο τα πρώτα 2 μόνο ζεύγη ψηφίων και επομένως θα έχουμε 2 κουτάκια. Το αποτέλεσμα είναι λογικό να υπάρχει σε 4 κουτάκια καθώς μπορεί να προκύψει αριθμός κατά ένα μεγαλύτερο ψηφίο από τους αριθμούς που προστίθενται. Επίσης αξίζει να προστεθεί ότι όταν γίνεται λάθος σε μία φάση αυτό δεν επηρεάζει την κρίση του συστήματος για τις απαντήσεις που θα δώσει ο χρήστης στις επόμενες φάσεις. Όταν γίνεται λάθος σε μία φάση το σύστημα θα κρίνει τις επόμενες φάσεις με βάση το λάθος που έγινε πριν να ήταν σωστό. Υπό αυτή την έννοια το σύστημα κρίνει πιο πολύ τον τρόπο συλλογιστικής που ακολούθησε ο χρήστης παρά το αποτέλεσμα καθεαυτό.

Στην πρώτη άσκηση από τις μεγάλης δυσκολίας ασκήσεις το σύστημα ρωτά το χρήστη πόσο κάνει $956 + 560$. Ο χρήστης έστω ότι δίνει την απάντηση 4590 στη σειρά αποτελεσμάτων και στα κρατούμενα δίνει τις τιμές 0 και 1 από αριστερά προς τα δεξιά. Όταν ο χρήστης πατήσει το κουμπί της επόμενης άσκησης γίνεται μία πολύπλοκη διαδικασία. Η πρόσθεση χωρίζεται σε 3 φάσεις. Στην πρώτη φάση εξετάζονται κατά ζεύγη οι μονάδες, δηλαδή το 6 και το 0. Ο χρήστης έβαλε στις μονάδες του αποτελέσματος 0 και στο πρώτο από τα κρατούμενα, το δεξί, 0. Αυτό σημαίνει ότι μπορεί να μπέρδεψε την πρόσθεση με τον πολλαπλασιασμό ή με τη διαίρεση. Επομένως, έχουμε διαφορετικό λάθος. Για τον ίδιο λόγο όμως που έγινε και πριν, το σύστημα θα επιλέξει ότι η σύγχυση που έγινε ήταν μεταξύ πρόσθεσης και πολλαπλασιασμού, εφόσον το `sum-mult_errors` είναι μεγαλύτερο από το `sum-div_errors`. Έτσι το `sum-mult_error_index` και το `sum-mult_errors` θα αυξηθούν κατά 1 με το πρώτο να γίνεται 2 και το δεύτερο 4, η `total_answers` θα γίνει ίση με 11 και η `total_errors` ίση με 8. Αξίζει να αναφέρουμε κάτι σε αυτό το σημείο. Καταρχάς εδώ ξεχνάμε προς το παρόν λίγο τη μεταβλητή `result` και θα

αναφερθούμε σε μία άλλη μεταβλητή, την *partial_result*. Η *partial_result* κρατά τα συνολικά μερικά αποτελέσματα από κάθε φάση. Αυτή στο τέλος θα διαιρεθεί διά του 3 και μετά θα προστεθεί στο μερικό άθροισμα της *result*. Επίσης, βλέπουμε ότι σε κάθε άσκηση μεγάλης δυσκολίας η μεταβλητή *total_answers* θα αυξάνεται κατά 3, διότι σε κάθε άσκηση η *total_answers* αυξάνεται κατά ένα σε κάθε φάση και επειδή είναι 3 οι φάσεις τότε αυτή από άσκηση σε άσκηση θα αυξάνει κατά 3. Η *partial_result* στην πρώτη φάση θα είναι ίση με $1/4$. Στη δεύτερη φάση ο χρήστης έβαλε στον πίνακα αποτελεσμάτων κάτω από το 5 και το 6, το 9 και σαν κρατούμενο το 1. Εδώ το σύστημα κάνει τον εξής συλλογισμό. Το 9 μπήκε με τη λογική ότι έγινε αφαίρεση ως εξής: το 6 δε μπορούσε να αφαιρεθεί από το 5. Επομένως, μπήκε κρατούμενο και αφαιρέθηκε από το 15, όπου και $15 - 6 = 9$. Το σύστημα δε μπορεί να βρει άλλου είδους συλλογιστικές για να υπάρξει διαφορετικό λάθος για σύγκριση της πρόσθεσης με άλλη πράξη. Επομένως, είναι ξεκάθαρο ότι η πρόσθεση συγχύστηκε με την αφαίρεση. Έτσι η *total_answers* γίνεται ίση με 12, η *total_errors* ίση με 9, η *sum-sub_error_index* και η *sum-sub_errors* αυξάνονται κατά ένα και γίνεται 2 η πρώτη και 3 η δεύτερη. Η μεταβλητή *partial_result* θα γίνει ίση με $1/4 + 1/4 = 1/2$. Στην τρίτη φάση ο χρήστης κάτω από τους αριθμούς 5 και 9 έδωσε 45. Εδώ προκύπτει πολλαπλό λάθος και για την ακρίβεια διπλό. Από τη μία ο χρήστης μπέρδεψε την πρόσθεση με τον πολλαπλασιασμό και δεν έλαβε κατά δεύτερον υπόψη του το κρατούμενο που γεννήθηκε από πριν. Έτσι έχουμε και σύγκριση πρόσθεσης με πολλαπλασιασμό και λάθος κρατουμένου. Έτσι η *total_errors* αυξάνεται κατά 2 και από 9 γίνεται 11, η *total_answers* γίνεται 13, η *sum-mult_error_index* και η *sum-mult_errors* αυξάνονται κατά 1, με την πρώτη να γίνεται ίση με 3 και η δεύτερη ίση με 5 και τα *curry-error_index*, *curry-errors* να αυξάνονται επίσης κατά ένα και να γίνονται και τα 2 ίσα με 1. Στην *partial_result* θα συμβάλει το λάθος με τη μεγαλύτερη συχνότητα (με το μεγαλύτερο *error* όχι *error_index*), δηλαδή στη συγκεκριμένη περίπτωση ο πολλαπλασιασμός και έτσι η *partial_result* θα γίνει $1/2 + 1/(2 + 3) = 7/10$. Επομένως, η *result* θα γίνει $58/12 + \text{partial_result} / 3 = 58/12 + 7/30 = 152/30$. Το σύστημα θα του βγάλει ένα συγκεντρωτικό μήνυμα για την ερώτηση συμβουλευόντας το χρήστη για το τι θα έπρεπε να είχε απαντήσει σε κάθε κουτάκι και τι είδους λάθη έκανε (Εικόνα 4.4.1.14).



Εικόνα 4.4.1.14. Συγκεντρωτικό μήνυμα λαθών.

Στην επόμενη ερώτηση το σύστημα ρωτά το χρήστη πόσο κάνει $465 + 230$. Έστω ότι ο χρήστης βάζει ως αποτέλεσμα 0290 και σαν κρατούμενα 00. Στην πρώτη φάση το σύστημα ελέγχει τους προσθετέους(5,0) και το αποτέλεσμα (0) καθώς και το κρατούμενο (0). Το σύστημα

αντιμετωπίζει πάλι διφορούμενο λάθος, συγκεκριμένα σύγχυση πρόσθεσης-πολλαπλασιασμού και πρόσθεσης-διαίρεσης, αλλά όπως είπαμε και πριν θα επιλέξει τη σύγχυση πρόσθεσης-πολλαπλασιασμού. Επομένως, η `total_answers` θα γίνει ίση με 14, η `total_errors` ίση με 12 και η `sum-mult_error_index` και `sum-mult_errors` θα αυξηθούν κατά 1 και θα γίνουν 4 και 6 αντίστοιχα. Επομένως, η `partial_result` θα γίνει ίση με $1/6$. Στη δεύτερη φάση το σύστημα βλέπει ότι κάτω από το 6 και 3 έχει μπει το 9 και από πάνω κρατούμενο 0. Επομένως, θεωρεί ότι είναι σωστή η απάντηση και επομένως η `partial_result` θα γίνει ίση με $7/6$, η `total_answers` ίση με 15, η `total_errors` θα παραμείνει σταθερή, ενώ θα αλλάξουν και τα `error_indexes` των λαθών. Συγκεκριμένα, η `sum-mult_error_index` θα γίνει 3, η `sum-sub_error_index` γίνεται 1, το `curry-error_index` θα γίνει 0 και όλα τα υπόλοιπα `error_indexes` θα παραμείνουν 0. Στην τρίτη φάση το σύστημα παρατηρεί ότι κάτω από τα 4 και 2 έχει μπει το 02. Επομένως, το σύστημα θεωρεί ότι υπάρχει διφορούμενο λάθος, πρόσθεσης-αφαίρεσης και πρόσθεσης-διαίρεσης. Το σύστημα θα επιλέξει τη σύγχυση πρόσθεσης-αφαίρεσης, επειδή είναι πιο συχνή και επομένως η `sum-sub_errors` θα γίνει ίση με 4, η `sum-sub_error_index` ίση με 2, η `total_answers` ίση με 16 και η `total_errors` ίση με 13. Η `partial_result` θα γίνει ίση με $7/6 + 1/4 = 17/12$. Έτσι η `result` θα γίνει $152/30 + 17/(12 * 3) = 997/180$. Το σύστημα θα του προβάλει σχετικό μήνυμα με τα λάθη που έκανε και τί θα έπρεπε να είχε βάλει στα κουτάκια. Επομένως συνοψίζοντας έχουμε:

`total_answers` = 16, `total_errors` = 13 `sum-sub_error_index` = 2, `sum-mult_error_index` = 2, όλα τα άλλα `error_indexes` είναι 0 και `sum-sub_errors` = 4, `sum-mult_errors` = 6, `sum-div_errors` = 1, `curry-errors` = 1, `serious_errors` = 1 και όλα τα άλλα `errors` είναι 0.

Στην επόμενη ερώτηση το σύστημα ρωτά το χρήστη πόσο κάνει $721 + 336$. Έστω ότι ο χρήστης απαντά και βάζει 1037 σαν αποτέλεσμα και 00 σαν κρατούμενα. Το σύστημα βλέπει ότι ο χρήστης δεν έκανε κανένα λάθος και επομένως η `total_answers` θα αυξηθεί κατά 3 (λόγω των 3 φάσεων) και θα γίνει `total_answers` = 19, όλα τα `error_indexes` θα μηδενιστούν (γιατί τα μόνα μη μηδενικά ήταν τα `sum-sub_error_index` = 2 και `sum-mult_error_index` = 2). Η `result` γίνεται $997/180 + 1 = 1177/180$. Συνοψίζοντας λοιπόν: `total_answers` = 19, `total_errors` = 13 όλα τα `error_indexes` είναι 0 και `sum-sub_errors` = 4, `sum-mult_errors` = 6, `sum-div_errors` = 1, `curry-errors` = 1, `serious_errors` = 1 και όλα τα άλλα `errors` είναι 0.

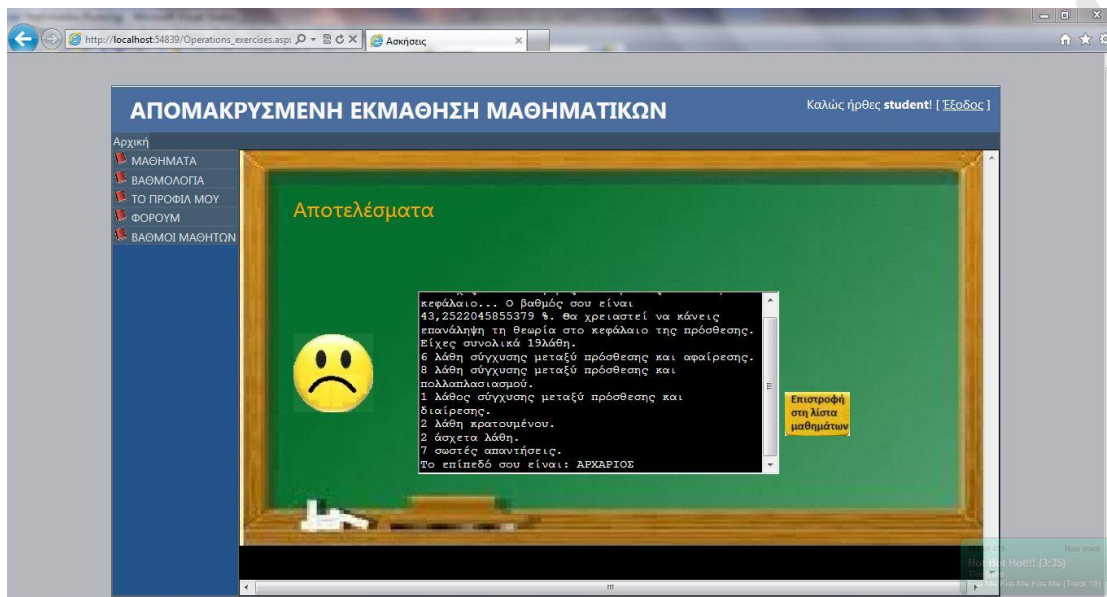
Στην επόμενη ερώτηση το σύστημα ρωτά πόσο κάνει $996 + 335$. Έστω ότι ο χρήστης απαντά 1271 σαν αποτέλεσμα και 20 σαν κρατούμενα. Στην πρώτη φάση το σύστημα βλέπει ότι στην πρόσθεση των 6 και 5 ο χρήστης έβαλε ως αποτέλεσμα 1 και κρατούμενο μηδέν. Το σύστημα βρίσκει ότι υπάρχουν 2 διφορούμενα και αυτό ισχύει ως εξής. Στη μία περίπτωση το σύστημα θεωρεί ότι ο χρήστης μπερδέψε πρόσθεση με αφαίρεση, αφού $6-5=1$ και στην άλλη ότι έκανε λάθος κρατουμένου, εφόσον $6+5=11$ και απλά ξέχασε τον άσσο στα κρατούμενα. Το σύστημα θα επιλέξει ότι έγινε σύγχυση πρόσθεσης-αφαίρεσης για τους λόγους που έχουμε αναφέρει επανειλημμένως. Επομένως, η `total_answers` θα γίνει ίση με 20, η `total_errors` ίση με 14, η `sum-sub_error_index` θα γίνει 1 και η `sum-sub_errors` ίση με 5. Η `partial_result` θα γίνει ίση με $1/3$. Στη δεύτερη φάση το σύστημα παρατηρεί ότι ο χρήστης κάτω από τα 9 και 3 έβαλε 7 και σαν κρατούμενο έβαλε 2. Άρα είναι προφανές ότι ο χρήστης μπερδέψε πρόσθεση με πολλαπλασιασμό. Έτσι, η `total_answers` = 21, `total_errors` = 15, `sum-mult_error_index` = 1, `sum-mult_errors` = 7. Η `partial_result` θα γίνει $1/3 + 1/3 = 2/3$. Στην τρίτη φάση το σύστημα βλέπει ότι ο χρήστης κάτω από τα 9 και 3 έβαλε 12 αλλά από πριν υπάρχει το κρατούμενο 2. Επομένως ξέχασε το κρατούμενο και έκανε λάθος κρατουμένου. Έτσι η `total_answers` γίνεται 22, η `total_errors` γίνεται 16 και η `curry-error_index` ίση με 1 και η `curry-errors` ίση με 2. Η `partial_result` θα γίνει ίση με $2/3 + 1/3 = 1$. Η `result` τότε θα γίνει ίση με $1177/180 + 1/3 = 1237/180$. Το σύστημα θα βγάλει προειδοποιητικό μήνυμα για το χρήστη για τα λάθη του και τί θα έπρεπε να είχε απαντήσει. Συνοψίζοντας:

total_answers = 22, total_errors = 16, sum-sub_error_index = 1, sum-mult_error_index = 1, curry-error_index = 1, sum-sub_errors = 5, sum-mult_errors = 7, sum-div_errors = 1, curry-errors = 2, serious-errors = 1.

Στην τελευταία ερώτηση το σύστημα ρωτά το χρήστη πόσο κάνει $589 + 182$. Ο χρήστης έστω ότι απαντά, 0517 σαν αποτέλεσμα και 00 σαν κρατούμενα. Στην πρώτη φάση το σύστημα παρατηρεί ότι ο χρήστης κάτω από τα 9 και 2 έβαλε 7 και σαν κρατούμενο 0. Επομένως θεωρεί αυτονόητο ότι μπέρδεψε πρόσθεση με αφαίρεση. Έτσι, η total_answers θα γίνει ίση με 23, η total_errors ίση με 17, η sum-sub_error_index ίση με 2 και η sum-sub_errors ίση με 6. Η partial_result θα γίνει ίση με $1/4$. Στη δεύτερη φάση το σύστημα παρατηρεί ότι κάτω από τα 8 και 8 ο χρήστης έβαλε 1 και σαν κρατούμενο 0. Αυτό σημαίνει αυτόματα ότι ο χρήστης μπέρδεψε εδώ πρόσθεση με διαίρεση. Επομένως, η total_answers θα γίνει 24, η total_errors ίση με 18 και η sum-div_error_index ίση με 1 και η sum-div_errors ίση με 2. Η partial_result θα γίνει $1/4 + 1/3 = 7/12$. Τέλος, παρατηρείται στην τρίτη φάση ότι κάτω από τα 5 και 1 ο χρήστης έβαλε 05. Εδώ υπάρχει διφορούμενο λάθος, σύγχυση πρόσθεσης-πολλαπλασιασμού και πρόσθεσης-διαίρεσης. Το σύστημα θα επιλέξει τελικά ότι υπάρχει σύγχυση πρόσθεσης-πολλαπλασιασμού και επομένως η total_answers θα γίνει ίση με 25, η total_errors ίση με 19 και η sum-mult_error_index ίση με 2 και η sum-mult_errors ίση με 8. Η partial_result θα γίνει ίση με $7/12 + 1/4 = 10/12 = 5/6$. Επομένως, η result θα γίνει ίση με $1237/180 + 5/(6*3) = 1287/180$. Τέλος, η result θα διαιρεθεί διά του πλήθους των ερωτήσεων που ήταν 15 συνολικά σε όλα τα επίπεδα δυσκολίας και θα βρεθεί ο τελικός βαθμός για το μάθημα της πρόσθεσης: $(1287 / (180 * 15)) * 100\% = 47,67\%$. Συνοψίζοντας έχουμε:

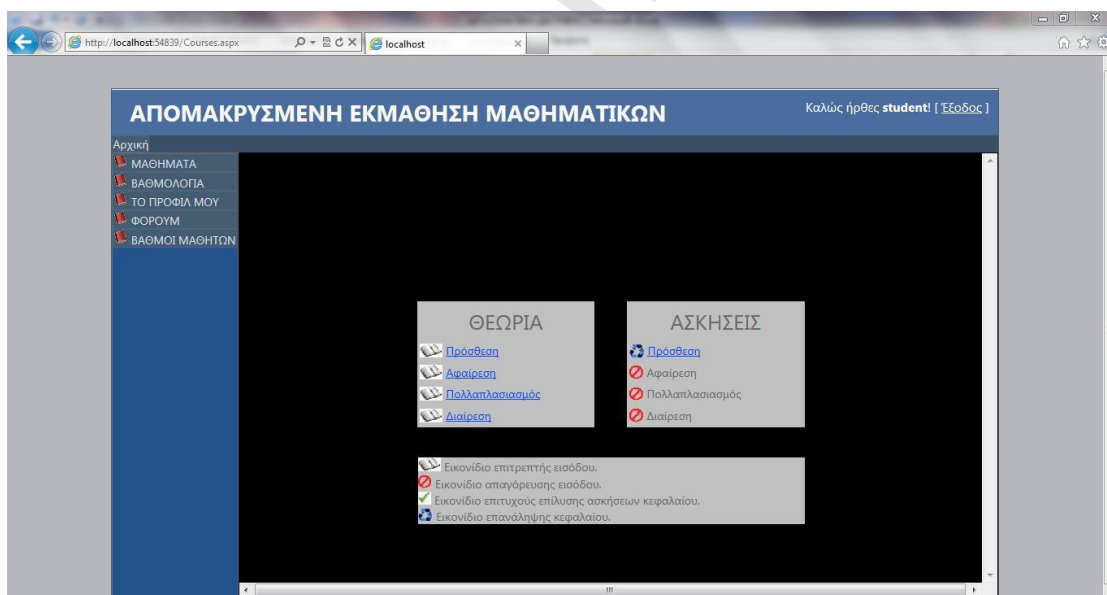
total_answers = 25, total_errors = 19, sum-sub_error_index = 2, sum-mult_error_index = 2, sum-div_error_index = 1, curry-error_index = 1, sum-sub_errors = 6, sum-mult_errors = 8, sum-div_errors = 2, curry-errors = 2, serious-errors = 1.

Αυτό φαίνεται και στον συγκεντρωτικό πίνακα που βγαίνει μετά το τέλος των ερωτήσεων από το σύστημα στο χρήστη για να του δείξει μία πλήρη εικόνα και με στατιστικά για το τί έκανε (Εικόνα 4.4.1.15). Με ποσοστό 47,67% (προσεγγιστικά πάντα) ο μαθητής δεν περνά το μάθημα, γιατί θέλει πάνω από 60% για να το περάσει, το επίπεδό του παραμένει αρχάριος, διότι το επίπεδο υπολογίζεται ως $((total_answers - total_errors) / total_answers) * 100\%$ και για να περάσει επίπεδο θα πρέπει το κλάσμα αυτό να είναι 60% έως 80% για να ναι μέτριος, ενώ για να είναι καλός πρέπει να έχει από 80% και πάνω. Στο συγκεκριμένο παράδειγμα ο χρήστης έχει $((25 - 19) / 25) * 100\% = (6/25) * 100\% = 24\%$. Όταν ο χρήστης δεν περνά το μάθημα ένα χαρακτηριστικό ανθρωπάκι με λυπημένη φάτσα παρουσιάζεται.



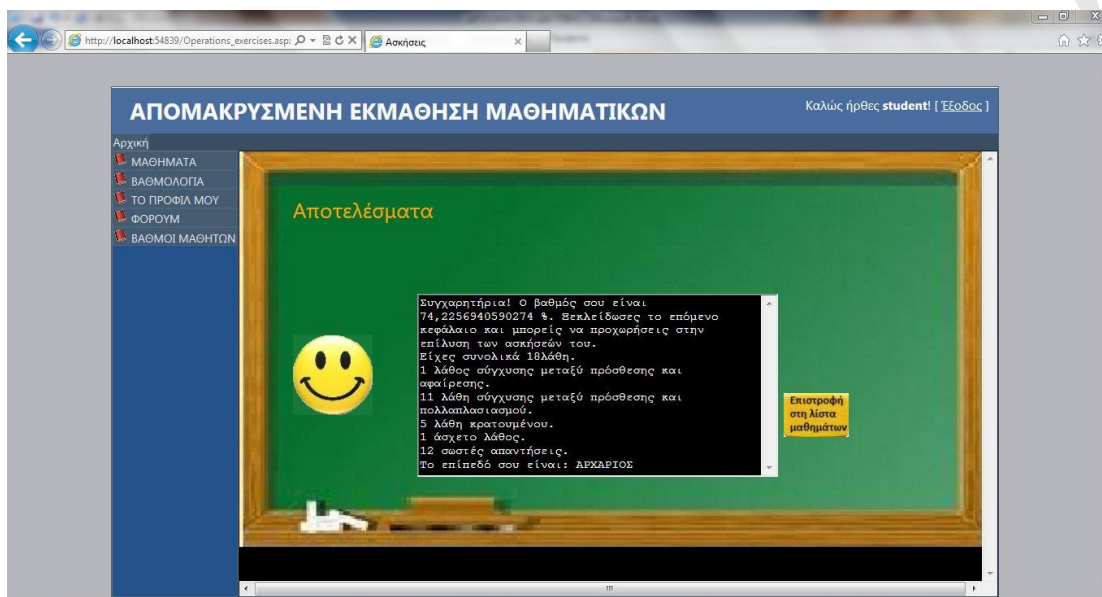
Εικόνα 4.4.1.15 Αποτελέσματα απαντήσεων του χρήστη για την πρόσθεση.

Όταν ο χρήστης επιστρέψει στη λίστα μαθημάτων θα δει ένα χαρακτηριστικό εικονίδιο επανάληψης στο μάθημα των μαθηματικών, που σημαίνει ότι για να το περάσει πρέπει να κάνει επανάληψη στη θεωρία και να λύσει εκ νέου ασκήσεις (Εικόνα 4.4.1.16).



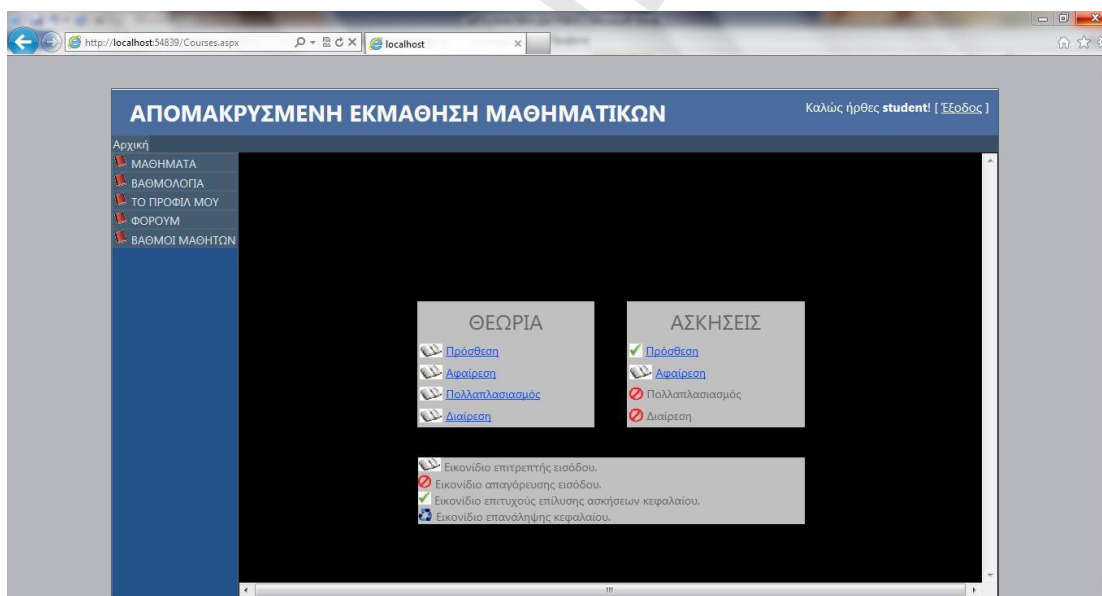
Εικόνα 4.4.1.16. Λίστα μαθημάτων με το εικονίδιο επανάληψης να εμφανίζεται για το κεφάλαιο της «πρόσθεσης».

Όταν ο χρήστης ξαναλύσει τις ασκήσεις της πρόσθεσης επιτυχώς, τότε στον πίνακα αποτελεσμάτων θα προβληθεί το σχετικό μήνυμα, με μία χαμογελαστή φάτσα και θα δει ακριβώς τα είδη λαθών του, το ποσοστό του και το επίπεδό του, αν άλλαξε ή όχι (Εικόνα 4.4.1.17).



Εικόνα 4.4.1.17. Ο χρήστης απαντά σωστά στις ασκήσεις με ποσοστό περίπου 74%, αλλά το επίπεδό του παραμένει αρχάριος.

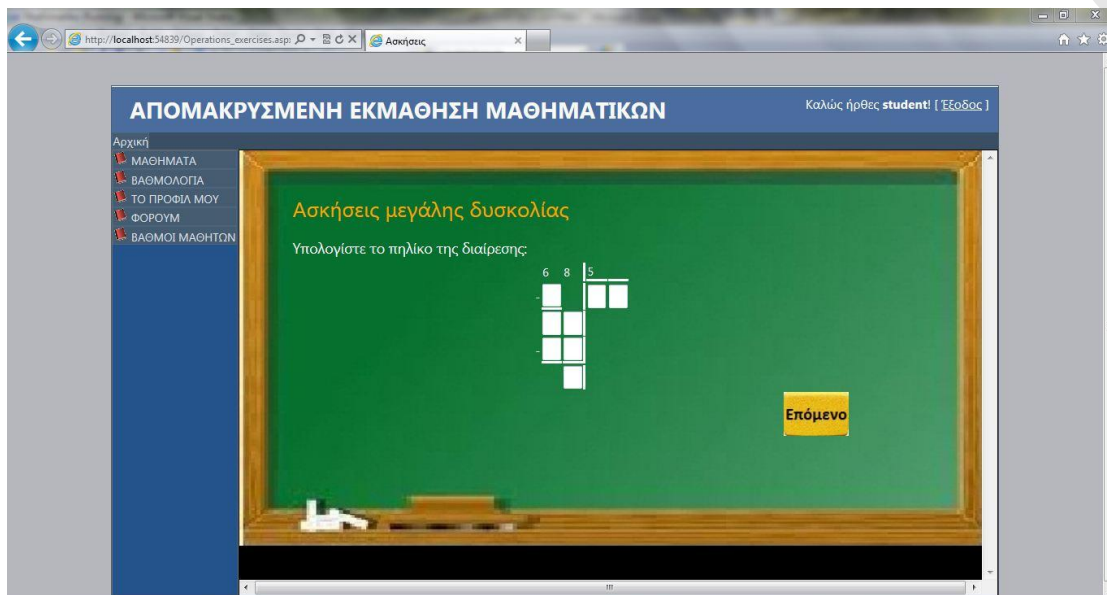
Όταν δε ο χρήστης επιστρέψει στη λίστα μαθημάτων θα δει ότι το κεφάλαιο της πρόσθεσης είναι τικαρισμένο, που σημαίνει ότι πέρασε το κεφάλαιο αυτό και το κεφάλαιο της αφαίρεσης, το επόμενο δηλαδή, έχει εικονίδιο ένα ανοιχτό βιβλίο, που σημαίνει ότι ξεκλείδωσε το κεφάλαιο της αφαίρεσης (Εικόνα 4.4.2.18).



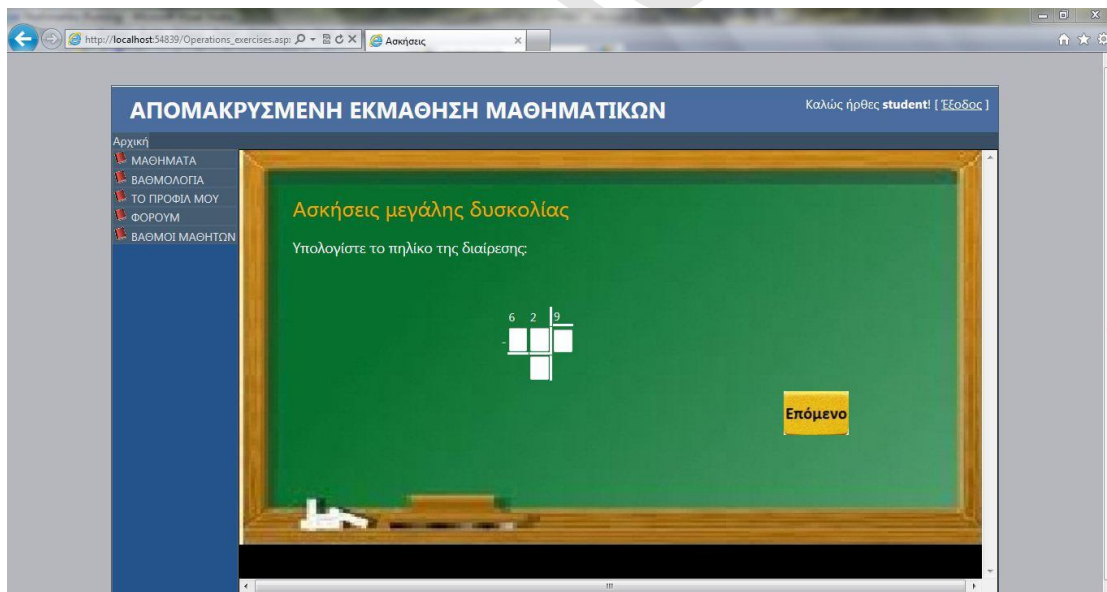
Εικόνα 4.4.1.17. Ο χρήστης ξεκλειδώνει το κεφάλαιο της αφαίρεσης.

Αντίστοιχα σε επόμενα κεφάλαια αν επιλύσει σωστά τις ασκήσεις ενός κεφαλαίου ξεκλειδώνει τα επόμενα κεφάλαια. Επειδή οι διαδικασία αξιολόγησης είναι η ίδια και για τα επόμενα κεφάλαια δε θα κάνουμε αναφορά περαιτέρω. Στην αφαίρεση έχουμε επίσης 2 τριψήφιους αριθμούς που αφαιρούνται. Στον πολλαπλασιασμό έχουμε ένα τριψήφιο που πολλαπλασιάζεται με ένα μονοψήφιο αριθμό. Το μόνο που αλλάζει ως προς τον τρόπο παρουσίασης είναι στις δύσκολες ασκήσεις της διαίρεσης. Εκεί ένας 2ψήφιος διαιρείται από ένα μονοψήφιο, απλά ο τρόπος

παρουσίασης αλλάζει ως προς το αν ο μονοψήφιος είναι μικρότερος ή όχι από το πιο σημαντικό ψηφίο, το δεύτερο, του διψήφιου (Εικόνες 4.4.1.18, 4.4.1.19).



Εικόνα 4.4.1.18. Πρώτη μορφή δύσκολων ασκήσεων διαίρεσης, όταν ο διαιρέτης χωρά στο πρώτο ψηφίο του διαιρετέου.



Εικόνα 4.4.1.19. Δεύτερη μορφή δύσκολων ασκήσεων διαίρεσης, όταν ο διαιρέτης δε χωρά στο πρώτο ψηφίο του διαιρετέου.

Έστω ότι ο χρήστης που μπορεί να λύσει ασκήσεις διαίρεσης και επομένως τις έχει ξεκλειδώσει, θέλει να κάνει αναβαθμολόγηση στην πρόσθεση. Αν επιλύσει ασκήσεις πρόσθεσης και δε μπορέσει να φέρει πάνω ή ίσο με 60% τότε κλειδώνουν αυτόματα όλες οι ασκήσεις των κεφαλαίων που είναι επόμενα της πρόσθεσης, με την έννοια ότι εφόσον ο χρήστης δεν πήγε καλά σε ένα κεφάλαιο είναι υποχρεωμένος να ξαναδουλέψει πάλι και τα επόμενα που τα είχε περάσει μιας και τα επόμενα κεφάλαια εξαρτώνται από τα παλαιότερα.

Η σελίδα Marks.aspx

Ο χρήστης όταν πατήσει στο σύνδεσμο «ΒΑΘΜΟΛΟΓΙΑ» θα οδηγηθεί στη σελίδα Marks.aspx. Εκεί μπορεί να δει τη βαθμολογία του σε όλα τα μαθήματα καθώς και τα είδη των λαθών που κάνει συνολικά, συγκεκριμένα τα ποσοστά που έχει το καθένα είδος επί του συνόλου των λαθών (Εικόνα 4.4.1.20).

The screenshot shows a web browser window displaying the 'ΑΠΟΜΑΚΡΥΣΜΕΝΗ ΕΚΜΑΘΗΣΗ ΜΑΘΗΜΑΤΙΚΩΝ' page. The page title is 'ΒΑΘΜΟΛΟΓΙΑ ΤΕΛΕΥΤΙΑΣ ΠΡΟΣΠΕΛΑΣΗΣ (%)'. The browser address bar shows 'http://localhost:54839/Marks.aspx'. The page content includes a navigation menu on the left and two main data tables.

ΒΑΘΜΟΛΟΓΙΑ ΤΕΛΕΥΤΙΑΣ ΠΡΟΣΠΕΛΑΣΗΣ (%)				
Όνομα Χρήστη	Πρόσθεση	Αφαίρεση	Πολλαπλασιασμός	Διαίρεση
student	43	-1	-1	-1

ΕΙΔΗ ΛΑΘΩΝ							
Σύγκριση Πρόσθεσης με Αφαίρεση	Σύγκριση Πρόσθεσης με Πολλαπλασιασμό	Σύγκριση Πρόσθεσης με Διαίρεση	Σύγκριση Αφαίρεσης με Πολλαπλασιασμό	Σύγκριση Αφαίρεσης με Διαίρεση	Σύγκριση Πολλαπλασιασμού με Διαίρεση	Κρατουμένου	Άσχετα
33.33 %	44.44 %	11.11 %	0 %	0 %	0 %	11.11 %	5.56 %

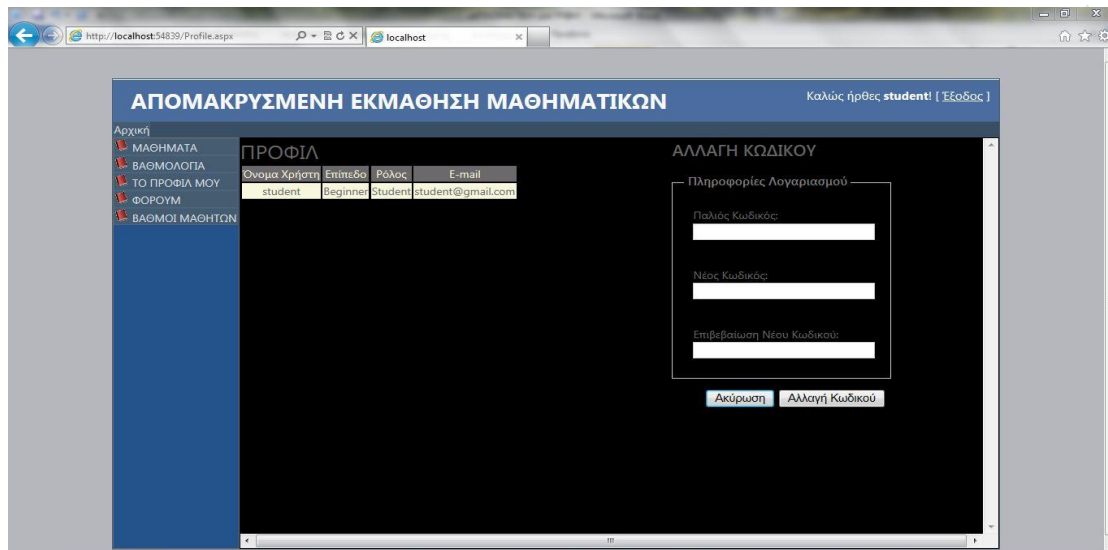
Βαθμολογία -1 σημαίνει ότι ο εκάστοτε χρήστης δεν έχει εξεταστεί στο συγκεκριμένο κεφάλαιο.

Εικόνα 4.4.1.20. Ο χρήστης βλέπει τη βαθμολογία του και τα είδη των λαθών του.

Όπως βλέπουμε και στην εικόνα ο χρήστης μπορεί να δει τη βαθμολογία του στον πρώτο πίνακα και τα είδη των λαθών του στο δεύτερο πίνακα. Σε όσα πεδία του πρώτου πίνακα υπάρχει η τιμή -1 σημαίνει ότι ο μαθητής δεν έχει εξεταστεί στο συγκεκριμένο κεφάλαιο.

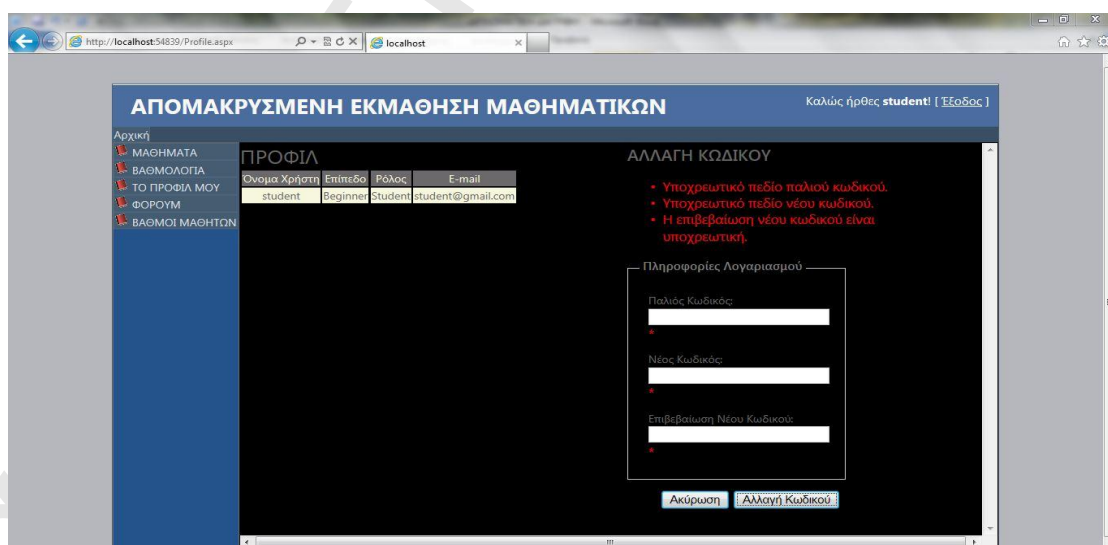
Η σελίδα Profile.aspx

Για να μπει στη σελίδα Profile.aspx ο χρήστης πρέπει να πατήσει στο σύνδεσμο «ΤΟ ΠΡΟΦΙΛ ΜΟΥ». Στη σελίδα αυτή ο χρήστης έχει τη δυνατότητα να δει το προφίλ του και να αλλάξει αν θέλει τον κωδικό του (Εικόνα 4.4.1.21).

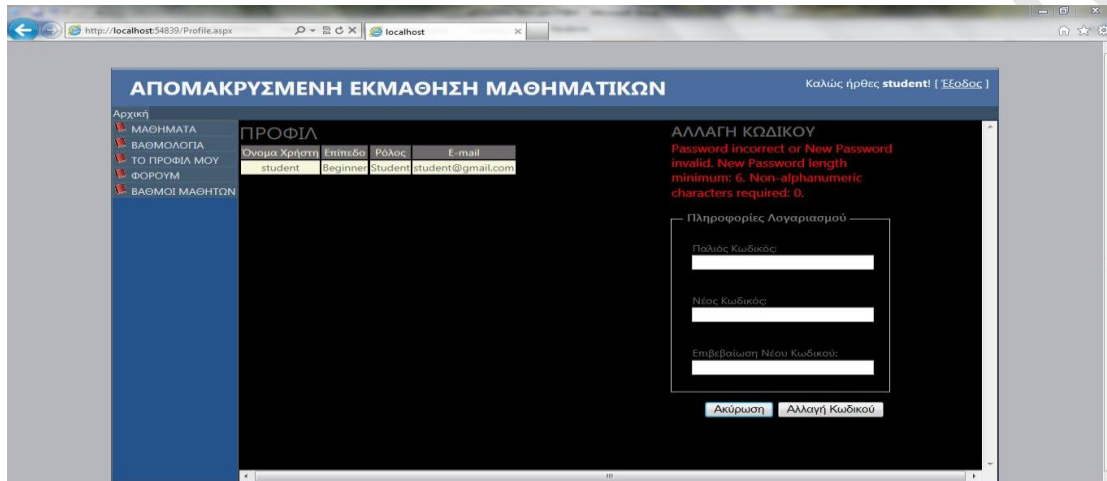


Εικόνα 4.4.1.21. Σελίδα προφίλ χρήστη.

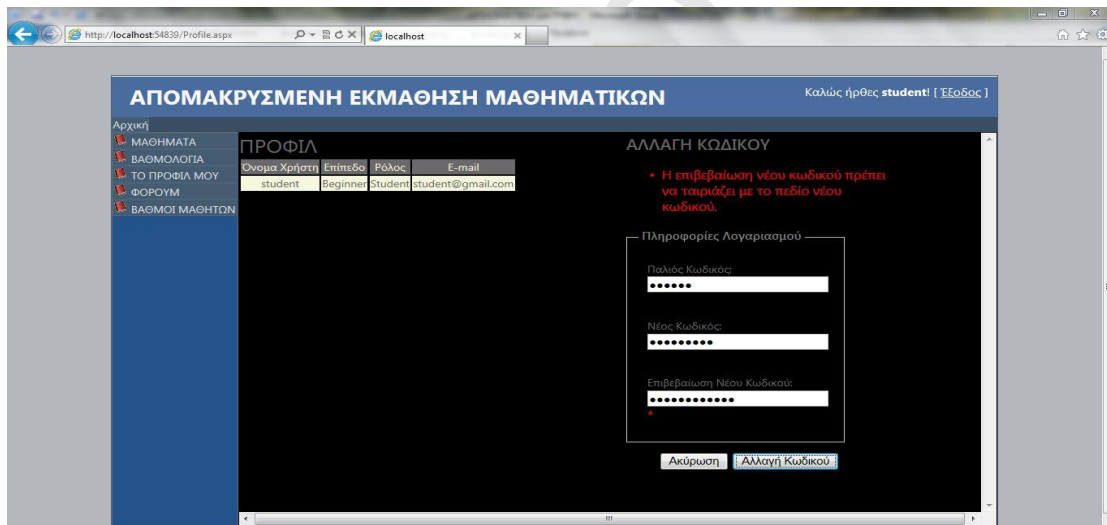
Συγκεκριμένα μπορεί να δει το username του, το επίπεδό του, το e-mail του και το ρόλο του. Στο δεξί panel βλέπουμε ότι μπορεί να αλλάξει και τον κωδικό του. Στην αρχή πρέπει να βάλει στο πρώτο πεδίο τον παλιό κωδικό, στο μεσαίο πεδίο τον νέο κωδικό και στο τέλος να επαληθεύσει τον νέο κωδικό. Αν δε βάλει τίποτα πουθενά το σύστημα θα του πει ότι είναι και τα 3 υποχρεωτικά πεδία και θα πρέπει να προσθέσει κατάλληλα τα κενά (Εικόνα 4.4.1.22). Αν στην αρχή βάλει άλλο κωδικό από αυτόν που ήδη έχει το σύστημα θα του πει ότι δεν είναι ο σωστός (Εικόνα 4.4.1.23). Επίσης, θα τον πληροφορήσει ότι ο νέος κωδικός θα πρέπει να έχει τουλάχιστον 6 χαρακτήρες σε μήκος και ότι δεν είναι υποχρεωτικό να βάλει μη αφαιρετικό χαρακτήρα σε αυτόν. Επίσης, αν βάλει σωστά τον παλιό κωδικό, βάλει ένα νέο κωδικό σύμφωνα με τις προαναφερθείσες προδιαγραφές, αλλά στην επιβεβαίωση κωδικού βάλει άλλο κωδικό από το νέο που έβαλε στο δεύτερο πεδίο, το σύστημα θα του πει ότι ο νέος κωδικός και η επιβεβαίωσή του θα πρέπει να ταιριάζουν (Εικόνα 4.4.1.24).



Εικόνα 4.4.1.22. Υποχρεωτικά πεδία εισαγωγής.



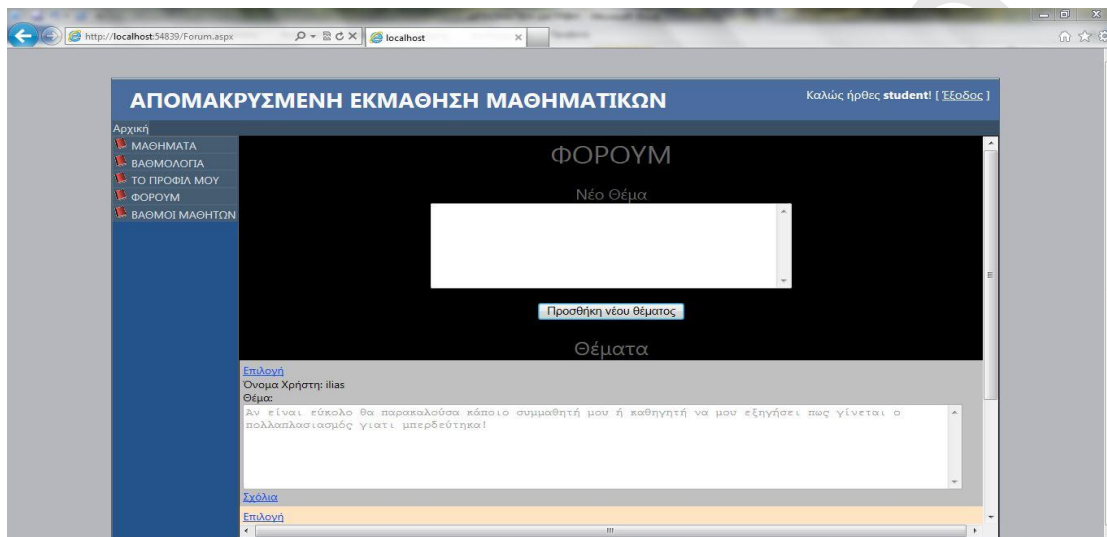
Εικόνα 4.4.1.23. Λάθος κωδικός.



4.4.1.24. Λάθος επιβεβαίωσης κωδικού.

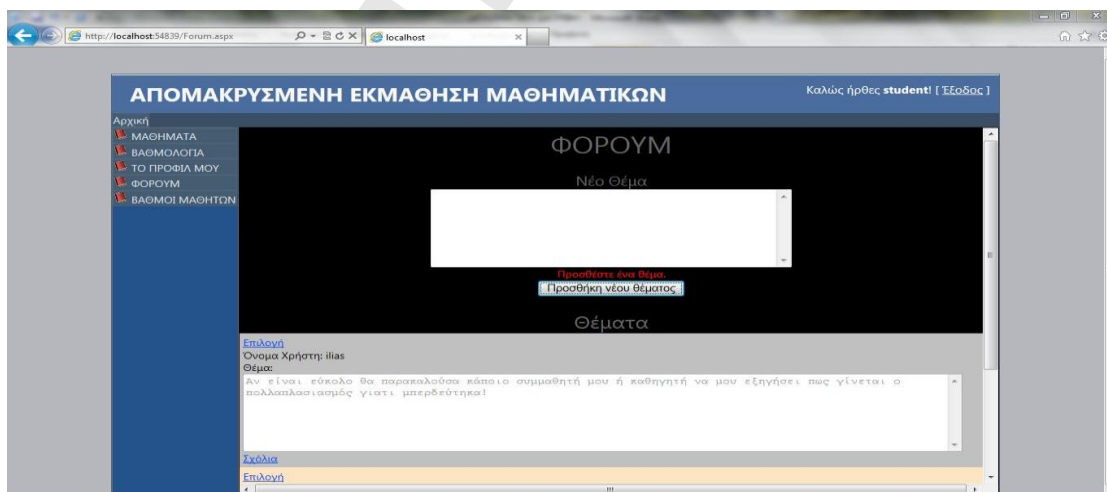
Η σελίδα Forum.aspx

Για να μπει στη σελίδα αυτή ο χρήστης πρέπει να πατήσει στο σύνδεσμο «ΦΟΡΟΥΜ». Στη σελίδα αυτή ο χρήστης μπορεί να ανταλλάσσει απόψεις και ιδέες με άλλους μαθητές ή καθηγητές, να παίρνει κάποια βοήθεια και να συμβουλευτείται από κάποιο καθηγητή κτλ. Είναι μία σελίδα φόρουμ, όπου οι χρήστες (μαθητές ή καθηγητές) ποστάρουν διάφορα θέματα και σε κάθε θέμα ανοίγει ένα thread από σχόλια. Ο χρήστης μπορεί να επεξεργαστεί τα θέματα και τα σχόλια που έγραψε ο ίδιος, δηλαδή να τα διαγράψει να προσθέσει ένα νέο ή να τα ανανεώσει (Εικόνα 4.4.1.25).



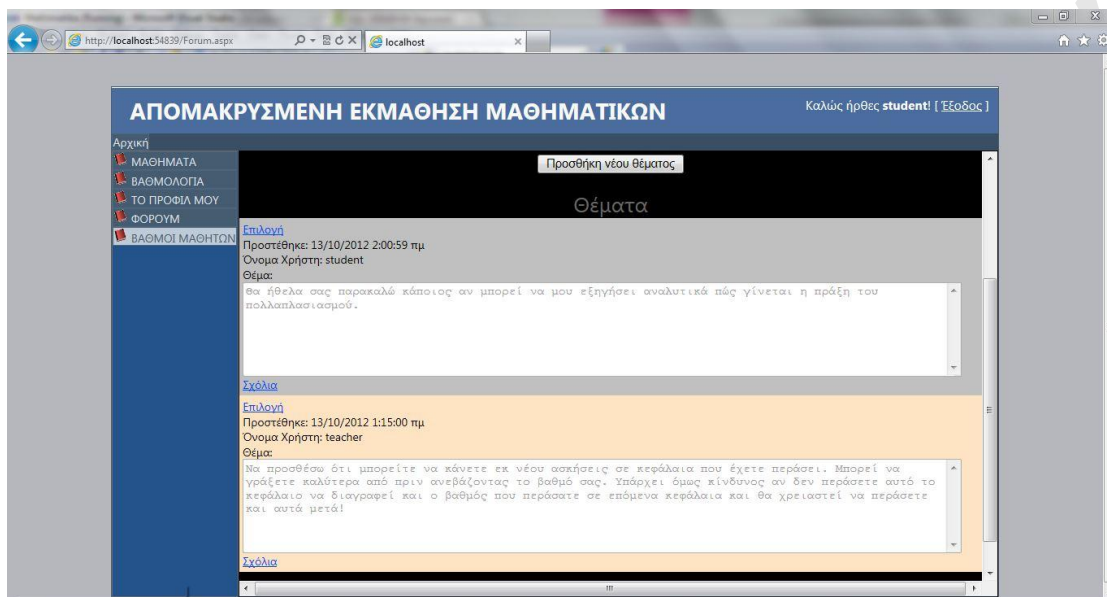
Εικόνα 4.4.1.25. Σελίδα φόρουμ.

Έστω ότι ο χρήστης θέλει να προσθέσει ένα νέο θέμα. Τότε θα πρέπει να προσθέσει οποιοδήποτε γεγονός τον απασχολεί μέσα στο μεγάλο κενό πεδίο και να πατήσει το κουμπί προσθήκης του νέου θέματος. Αν δεν πατήσει τίποτα τότε το σύστημα θα τον προτρέψει να εισάγει οπωσδήποτε κάτι (Εικόνα 4.4.1.26).



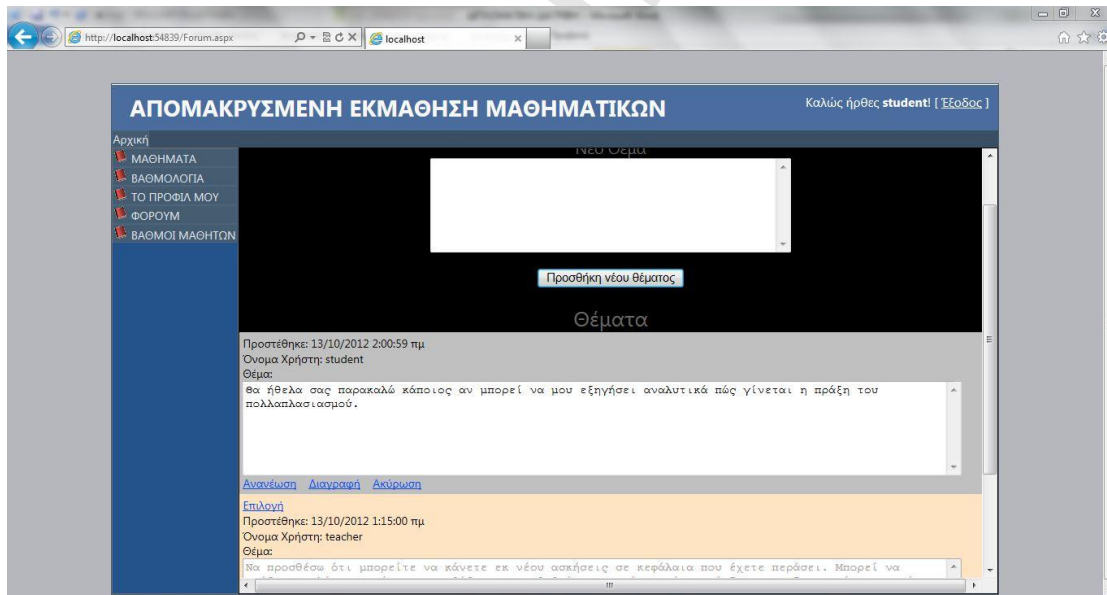
Εικόνα 4.4.1.26. Υποχρεωτική προσθήκη θέματος.

Έστω ότι ο χρήστης «student» προσθέτει ένα νέο θέμα. Τότε αφού πατήσει το κουμπί εισαγωγής θα το δει να προβάλλεται στα θέματα πρώτο στη σειρά (Εικόνα 4.4.1.27).



Εικόνα 4.4.1.27. Εισαγωγή νέου θέματος από τον χρήστη student.

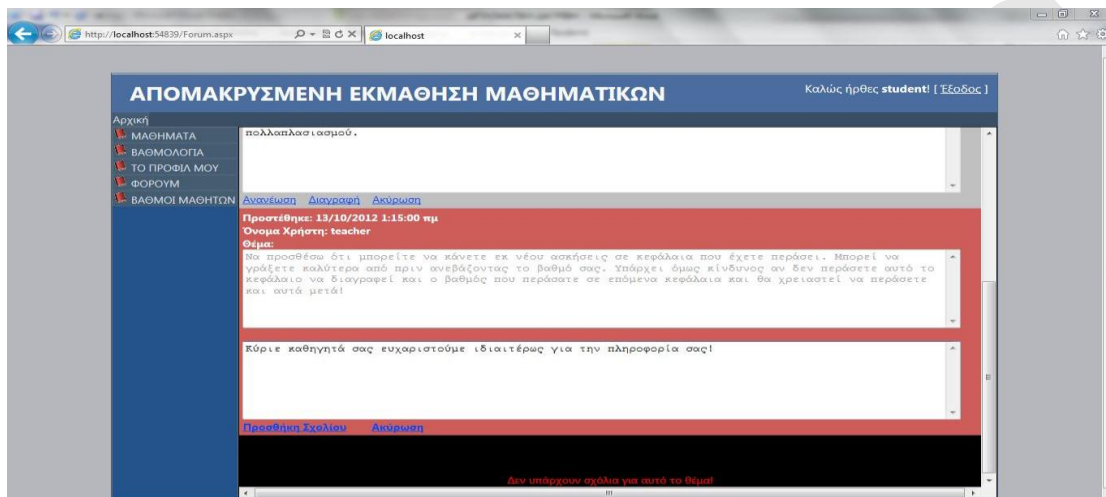
Παρατηρούμε ότι το τελευταίο θέμα που προστέθηκε μπαίνει στην κορυφή της λίστας. Στο θέμα που προστέθηκε φαίνεται το θέμα που πρόσθεσε ένας χρήστης, το του όνομα χρήστη, η ημερομηνία και η ώρα που προστέθηκε το θέμα καθώς και άλλα 2 κουμπάκια. Το κουμπί που λέει «επιλογή» δίνει τη δυνατότητα στο χρήστη να διαγράψει το θέμα, να ανανεώσει το θέμα, ή να ακυρώσει την επιλογή του αυτή (Εικόνα 4.4.1.28).



Εικόνα 4.4.1.28. Ο χρήστης μπορεί να διαγράψει, ανανεώσει, ή να ακυρώσει ένα δικό του θέμα.

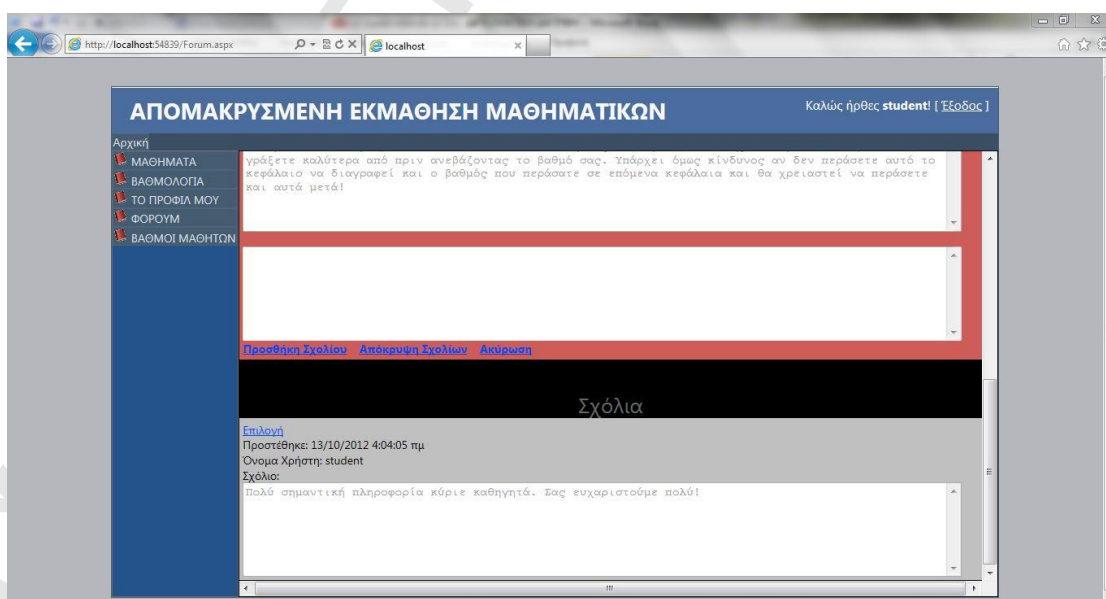
Εδώ πρέπει να προσθέσουμε ότι ο κάθε χρήστης μπορεί να επεξεργαστεί μόνο το δικό του θέμα. Θέματα άλλων δε μπορεί να τα επεξεργαστεί. Επίσης, ο χρήστης μπορεί να σχολιάσει σε όλα τα θέματα που έχουν τεθεί, ακόμα και στο δικό του. Επομένως, έστω ότι πάει και σχολιάζει στο θέμα που έθεσε ο καθηγητής «teacher». Αυτό το κάνει ως εξής: Στο θέμα που θέλει να σχολιάσει πατάει το κουμπί «σχόλια». Όταν το επιλέξει του εμφανίζεται από κάτω από το θέμα ένα κενό πεδίο που του επιτρέπει να προσθέσει το σχολίο του. Υπάρχουν από κάτω άλλα 2

κουμπιά, όταν προσθέτει το σχόλιό του. Το ένα λέει «προσθήκη σχολίου» το οποίο εισάγει το σχόλιο στη βάση και το άλλο ακυρώνει την εισαγωγή σχολίου. Παρατηρούμε επίσης πως όταν δεν υπάρχουν σχόλια προς ένα θέμα το σύστημα πληροφορεί κατάλληλα το χρήστη (Εικόνα 4.4.1.29).



Εικόνα 4.4.1.29. Προσθήκη σχολίου από το χρήστη. Για το συγκεκριμένο θέμα δεν υπάρχουν άλλα σχόλια.

Μετά το πάτημα του κουμπιού και την προσθήκη του σχολίου για το αντίστοιχο θέμα, εμφανίζεται αμέσως στα σχόλια το τελευταίο σχόλιο που πρόσθεσε ο χρήστης student (Εικόνα 4.4.1.30). Παράλληλα στο θέμα στο οποίο έκανε σχόλιο εμφανίζεται άλλο ένα κουμπί που λέει «απόκρυψη σχολίων». Με το συγκεκριμένο κουμπί ο χρήστης μπορεί να αποκρύψει τα σχόλια για το συγκεκριμένο θέμα. Στα σχόλια που κάνει ο εκάστοτε χρήστης μπορεί όπως και στα δικά του θέματα να παρέμβει. Συγκεκριμένα όταν κάνει ένα σχόλιο, υπάρχει πάνω από το σχόλιο ένα κουμπάκι που λέει «επιλογή». Με το συγκεκριμένο κουμπάκι ο χρήστης μπορεί να διαγράψει, να ανανεώσει ένα σχόλιό του, ή να ακυρώσει την επιλογή του. Ο χρήστης έχει εξουσιοδότηση να παρεμβαίνει μόνο στα δικά του σχόλια και όχι σε άλλων χρηστών.

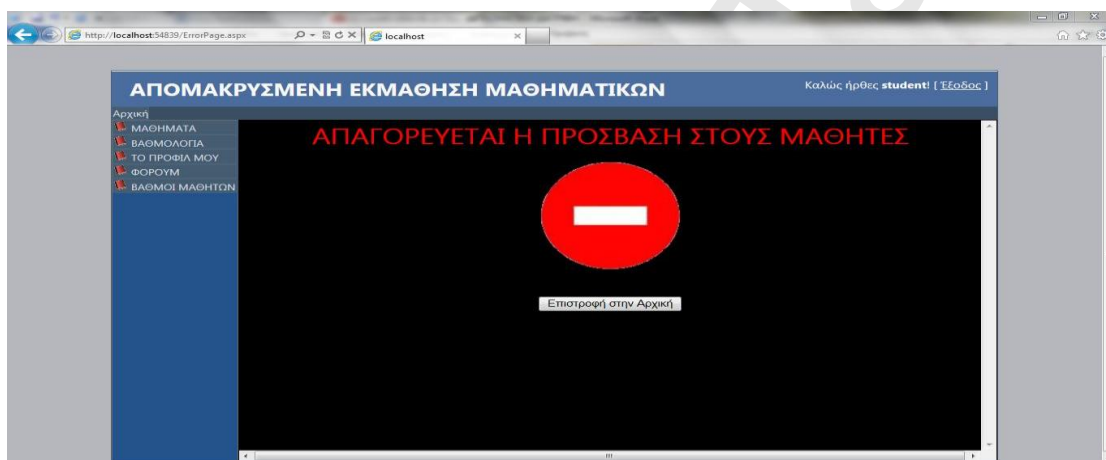


Εικόνα 4.4.1.30. Ο χρήστης student προσθέτει ένα σχόλιο σε ένα θέμα που έθεσε ο καθηγητής teacher.

Τέλος, πρέπει να πούμε ότι όταν ένας χρήστης διαγράψει ένα θέμα που πρόσθεσε τότε διαγράφονται αυτόματα και όλα τα σχόλια που είχαν γίνει για το συγκεκριμένο θέμα.

Η σελίδα *ViewMarks.aspx*

Για να μπει στη σελίδα αυτή ο μαθητής πρέπει να πατήσει στο σύνδεσμο «ΒΑΘΜΟΙ ΜΑΘΗΤΩΝ». Οποιοσδήποτε μαθητής όμως μπει σε αυτή τη σελίδα, έτσι και ο student, θα δει ότι δεν είναι εξουσιοδοτημένος να προσπελάσει τη συγκεκριμένη σελίδα, καθώς είναι σελίδα δημιουργημένη μόνο για καθηγητές (Εικόνα 4.4.1.31).



Εικόνα 4.4.1.31. Σελίδα παρακολούθησης βαθμολογιών μαθητών. Εξουσιοδοτημένοι να την προσπελάσουν είναι μόνο οι καθηγητές.

4.4.2 Σενάριο Εκτέλεσης χρήστη με ρόλο καθηγητή

Για το σενάριο εκτέλεση με χρήστη καθηγητή, θα αναφερθούμε λεπτομερέστερα μόνο στην προσθήκη και προβολή ασκήσεων, καθώς και στην προβολή βαθμολογιών των μαθητών. Όλες οι υπόλοιπες λειτουργίες της ιστοσελίδας είναι όμοιες για καθηγητή και μαθητή και επομένως δε θα κάνουμε αναφορά σε αυτές, μιας και έχουμε ήδη κάνει λεπτομερή αναφορά στο ρόλο του μαθητή. Ο ρόλος του καθηγητή διαφοροποιείται από αυτόν του μαθητή σε 3 περιπτώσεις: i) Στην προβολή των ασκήσεων, ii) στην προσθήκη νέων ασκήσεων, iii) στην προβολή των βαθμολογιών των μαθητών. Θα αναφερθούμε αναλυτικά σε καθένα από αυτά. Έστω λοιπόν ότι στο σύστημα έχει εγγραφεί ο καθηγητής με όνομα χρήστη «teacher» και μπαίνει στην κεντρική σελίδα.

Προβολή των ασκήσεων

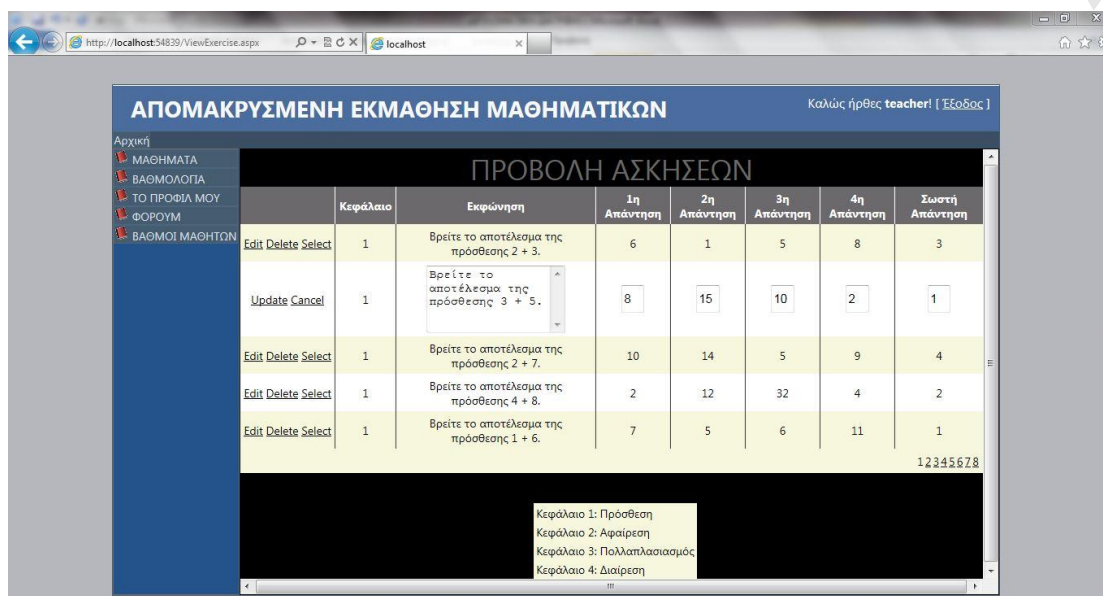
Ο καθηγητής όπως είπαμε και σε προηγούμενη παράγραφο, έχει τη δυνατότητα να δει τις ασκήσεις που θα εισάγει. Συγκεκριμένα, έχει τη δυνατότητα να παρακολουθήσει και να παρέμβει μόνο στις εύκολες ασκήσεις. Οι υπόλοιπες ασκήσεις παράγονται τυχαία από το σύστημα, δηλαδή οι αριθμοί που προκύπτουν και για τις 4 πράξεις βγαίνουν τυχαία. Ο καθηγητής teacher για να προβάλει τις ασκήσεις πρέπει να πατήσει στο σύνδεσμο «ΜΑΘΗΜΑΤΑ», όπου και θα οδηγηθεί στη λίστα μαθημάτων και για θεωρία και για ασκήσεις. Πάνω από τη λίστα των ασκήσεων υπάρχουν 2 άλλοι σύνδεσμοι. Ο ένας λέει «Προσθήκη νέας άσκησης» και ο άλλος «Προβολή των ασκήσεων». Ο χρήστης πατάει στην προβολή των ασκήσεων και οδηγείται στη σελίδα ViewExercise.aspx (Εικόνα 4.4.2.1).

ΑΠΟΜΑΚΡΥΣΜΕΝΗ ΕΚΜΑΘΗΣΗ ΜΑΘΗΜΑΤΙΚΩΝ		Καλώς ήρθες teacher! [Έξοδος]						
ΠΡΟΒΟΛΗ ΑΣΚΗΣΕΩΝ								
	Κεφάλαιο	Εκφώνηση	1η Απάντηση	2η Απάντηση	3η Απάντηση	4η Απάντηση	Σωστή Απάντηση	
Edit Delete Select	1	Βρείτε το αποτέλεσμα της πρόσθεσης 2 + 3.	6	1	5	8	3	
Edit Delete Select	1	Βρείτε το αποτέλεσμα της πρόσθεσης 3 + 5.	8	15	10	2	1	
Edit Delete Select	1	Βρείτε το αποτέλεσμα της πρόσθεσης 2 + 7.	10	14	5	9	4	
Edit Delete Select	1	Βρείτε το αποτέλεσμα της πρόσθεσης 4 + 8.	2	12	32	4	2	
Edit Delete Select	1	Βρείτε το αποτέλεσμα της πρόσθεσης 1 + 6.	7	5	6	11	1	12345678

Κεφάλαιο 1: Πρόσθεση
 Κεφάλαιο 2: Αφαίρεση
 Κεφάλαιο 3: Πολλαπλασιασμός
 Κεφάλαιο 4: Διαιρέση

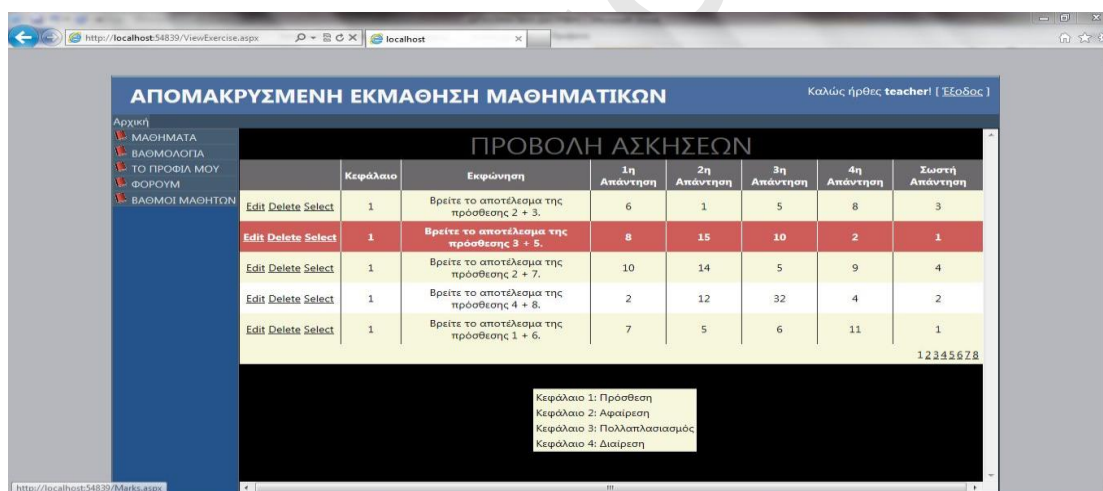
Εικόνα 4.4.2.1. Σελίδα προβολής ασκήσεων για καθηγητές.

Πρέπει να πούμε ότι οι 2 αυτοί σύνδεσμοι που μόλις αναφέραμε είναι ορατοί μόνο στο ρόλο του καθηγητή. Όταν ο καθηγητής οδηγείται στη σελίδα αυτή, μπορεί να δει τις εύκολες ασκήσεις που έχει ήδη εισάγει, να τις τροποποιήσει, να τις διαγράψει και να τις επιλέξει. Συγκεκριμένα όταν πατήσει πάνω στο κουμπί «Edit» τότε η συγκεκριμένη γραμμή που επέλεξε μπαίνει αυτόματα σε Edit mode, δηλαδή μπορεί ο χρήστης να ανανεώσει το περιεχόμενο της γραμμής. Συγκεκριμένα μπορεί να τροποποιήσει την εκφώνηση της άσκησης, τις πιθανές απαντήσεις και τη σωστή απάντηση, όχι όμως και το κεφάλαιο στο οποίο ανήκει αυτή. Όταν ο χρήστης εισάγει τα νέα δεδομένα μπορεί να πατήσει το κουμπί «Update» για να τα εισάγει στη βάση και για να ακυρώσει την επιλογή του αυτή πατάει το κουμπί «Cancel» (Εικόνα 4.4.2.2).



Εικόνα 4.4.2.2. Ο χρήστης προσπαθεί να ανανεώσει μία άσκηση.

Όταν ο χρήστης πατήσει το κουμπί «Select» τότε απλώς η άσκηση που επιλέχθηκε τονίζεται με κόκκινο χρώμα (Εικόνα 4.4.2.3).

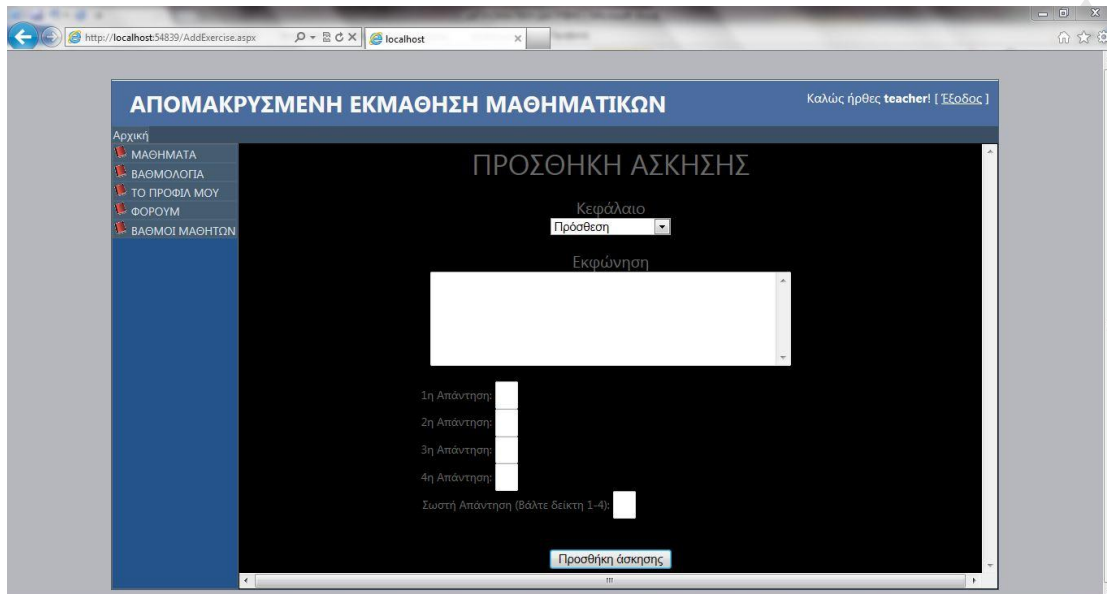


Εικόνα 4.4.2.3. Ο χρήστης επιλέγει μία συγκεκριμένη άσκηση.

Επίσης παρατηρούμε ότι οι ασκήσεις προβάλλονται με δεικτοδότηση του πίνακα για να μην πιάνει αρκετό χώρο στη σελίδα ο πίνακας. Έτσι όταν ο χρήστης πατήσει ένα από τα νούμερα κάτω δεξιά του πίνακα μπορεί να προβάλει και άλλες ασκήσεις. Τέλος, υπάρχει μία υπόδειξη κάτω από τον πίνακα που του λέει ότι στη στήλη «Κεφάλαιο» οι αριθμοί 1,2,3,4 αντιστοιχούν στα κεφάλαια Πρόσθεση, Αφαίρεση, Πολλαπλασιασμός, Διαίρεση αντίστοιχα.

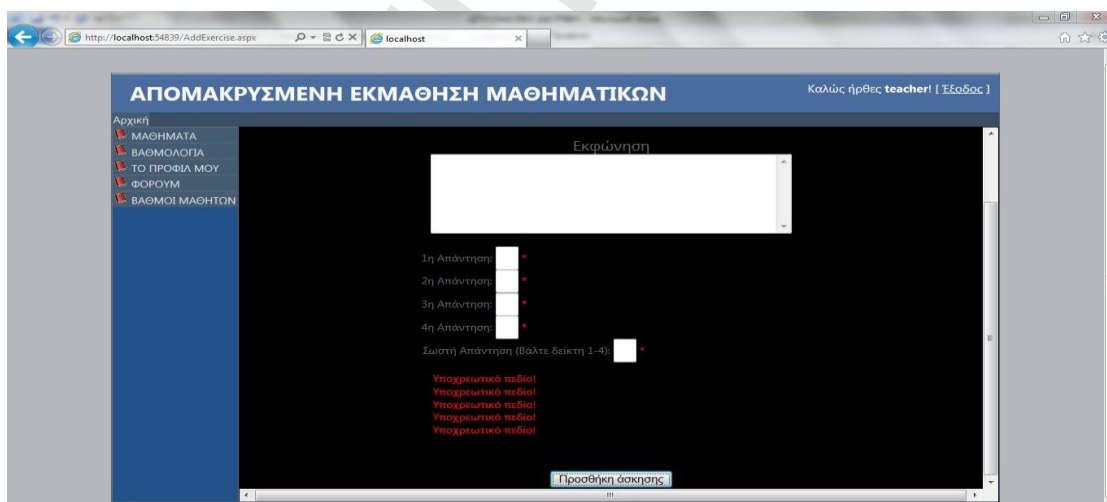
Προσθήκη ασκήσεων

Όταν ο καθηγητής θέλει να εισάγει μία νέα άσκηση τότε από τη σελίδα Courses.aspx με τις λίστες των μαθημάτων πρέπει να πατήσει το κουμπί «Προσθήκη νέας άσκησης». Τότε οδηγείται στη σελίδα AddExercise.aspx (Εικόνα 4.4.2.4).



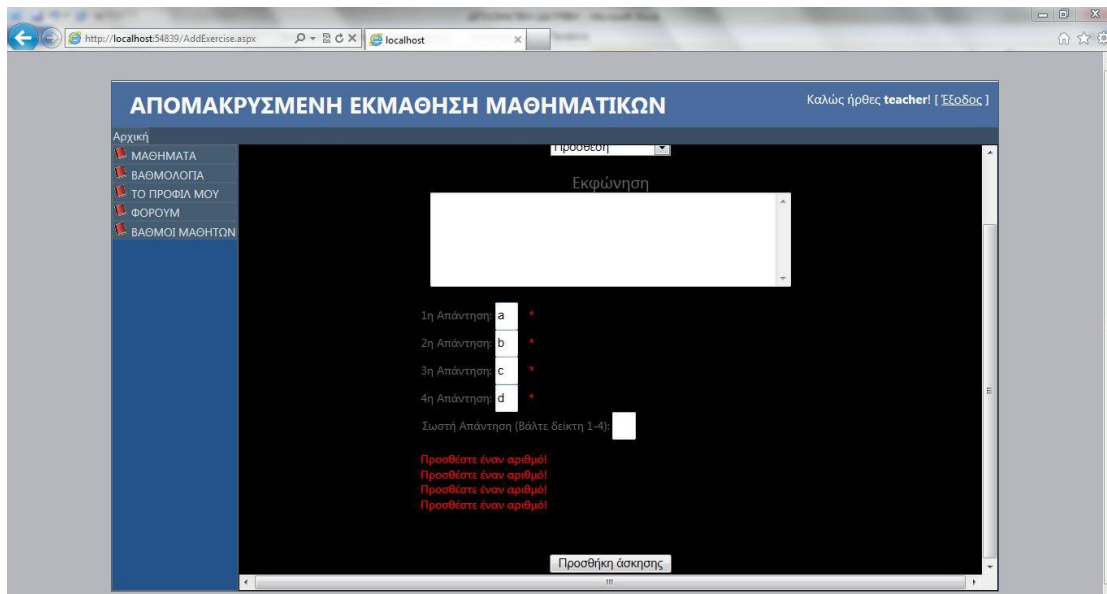
Εικόνα 4.4.2.4. Σελίδα προσθήκης ασκήσεων (Μόνο για καθηγητές).

Στη σελίδα αυτή ο καθηγητής μπορεί να κάνει προσθήκη μιας νέας άσκησης. Όταν θέλει να προσθέσει μία νέα άσκηση τότε μπορεί στο μεγάλο κενό πεδίο να προσθέσει την εκφώνηση της άσκησης, στα 4 κουτάκια από κάτω τις 4 πιθανές απαντήσεις για την πολλαπλή επιλογή, στο 5^ο κουτάκι να βάλει τη σωστή απάντηση και στη μπάρα που βρίσκεται στην κορυφή να επιλέξει το κεφάλαιο στο οποίο θα προστεθεί αυτή η άσκηση. Η προσθήκη τέλος γίνεται με το κουμπί «Προσθήκη άσκησης». Ο χρήστης αν δεν βάλει τίποτα στο κενό πεδίο το σύστημα θα τον παροτρύνει να πράξει ανάλογα. Το ίδιο συμβαίνει και με τα υπόλοιπα κενά πεδία, όπως και με τη μπάρα επιλογής κεφαλαίου (Εικόνα 4.4.2.5).



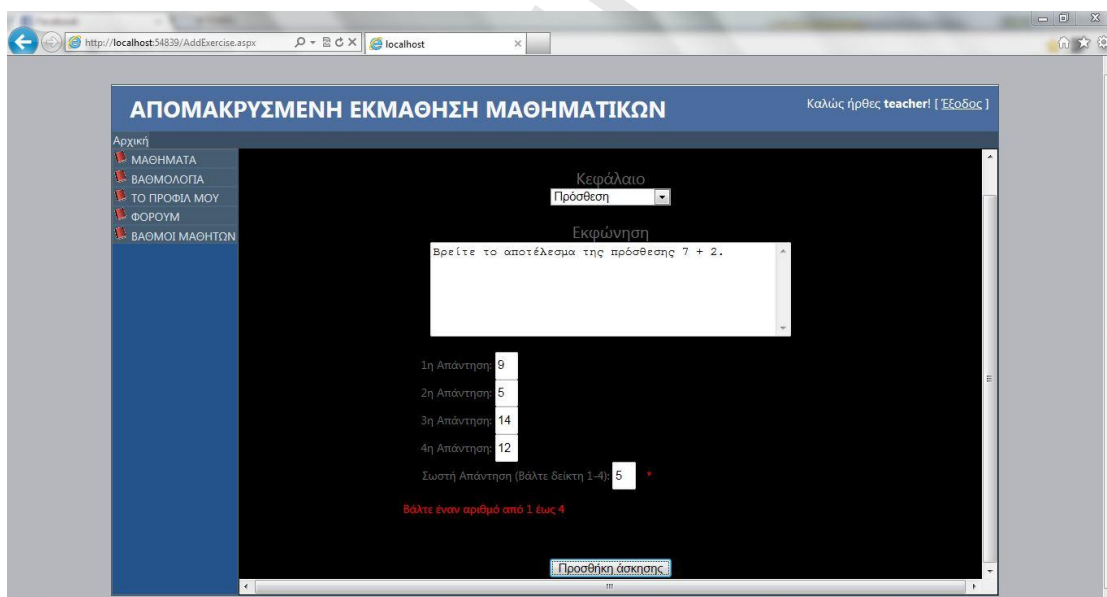
Εικόνα 4.4.2.5. Ο χρήστης δεν έχει προσθέσει τίποτα και πατά το κουμπί «Προσθήκη άσκησης». Το σύστημα βγάζει μηνύματα λάθους και τον παροτρύνει να πράξει ανάλογα.

Επίσης, όταν ο χρήστης προσθέσει μία μη αριθμητική τιμή στα πεδία των απαντήσεων, τότε το σύστημα θα τον προτρέψει να βάλει οπωσδήποτε αριθμητική τιμή (Εικόνα 4.4.2.6).



Εικόνα 4.4.2.6. Ο χρήστης δεν έχει εισάγει αριθμητικές τιμές και το σύστημα τον παροτρύνει να πράξει ανάλογα.

Επίσης, αν ο χρήστης στη σωστή απάντηση δε βάλει μία από τις τιμές 1,2,3,4 το σύστημα θα του πετάξει μήνυμα λάθους και θα τον παροτρύνει να πράξει ανάλογα (Εικόνα 4.4.2.7).



Εικόνα 4.4.2.7. Ο χρήστης στην προσθήκη νέας άσκησης έβαλε αριθμό στη σωστή απάντηση εκτός του επιτρεπόμενου διαστήματος. Το σύστημα τον παροτρύνει να πράξει ανάλογα.

Τέλος, ο χρήστης προσθέτει τη νέα άσκηση με τα σωστά δεδομένα και πατώντας το κουμπί την εισάγει στη βάση δεδομένων. Έστω ότι προσθέτει την άσκηση με εκφώνηση: «βρείτε το αποτέλεσμα της πρόσθεσης $7 + 2$ » και με πιθανές απαντήσεις 9,5,14,12 και σωστή απάντηση βέβαια το 1, μιας και η σωστή απάντηση βρίσκεται στην 1^η επιλογή. Ο χρήστης όταν προσθέσει αυτή την άσκηση μπορεί να τη δει στην προβολή ασκήσεων που είπαμε νωρίτερα. Κάθε νέα άσκηση που προστίθεται μπαίνει στο τέλος των ασκήσεων του αντίστοιχου κεφαλαίου. Έτσι ο

χρήστης θα δει την άσκηση που πρόσθεσε στο τέλος των ασκήσεων της πρόσθεσης (Εικόνα 4.4.2.8).

Κεφάλαιο	Εκφώνηση	1η Απάντηση	2η Απάντηση	3η Απάντηση	4η Απάντηση	Σωστή Απάντηση
1	Βρείτε το αποτέλεσμα της πρόσθεσης $7 + 2$.	9	5	14	12	1
2	Βρείτε το αποτέλεσμα της αφαίρεσης $7 - 3$.	10	4	21	17	2
2	Βρείτε το αποτέλεσμα της αφαίρεσης $9 - 4$.	5	36	13	21	1
2	Βρείτε το αποτέλεσμα της αφαίρεσης $10 - 5$.	15	5	50	2	2
2	Βρείτε το αποτέλεσμα της αφαίρεσης $8 - 6$.	48	14	20	2	4

Κεφάλαιο 1: Πρόσθεση
Κεφάλαιο 2: Αφαίρεση
Κεφάλαιο 3: Πολλαπλασιασμός
Κεφάλαιο 4: Διάρθρωση

Εικόνα 4.4.2.8. Ο χρήστης βλέπει την άσκηση που πρόσθεσε στο τέλος των ασκήσεων του αντίστοιχου κεφαλαίου.

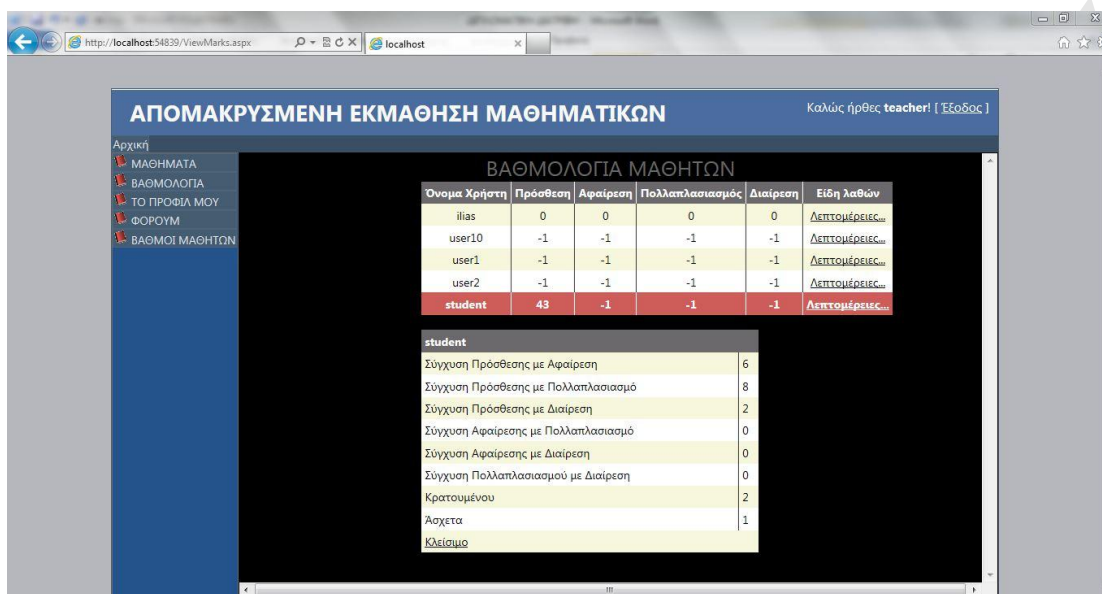
Προβολή βαθμολογιών μαθητών

Ο καθηγητής όταν θέλει να δει τη βαθμολογία των μαθητών του πρέπει να πατήσει στο σύνδεσμο «ΒΑΘΜΟΙ ΜΑΘΗΤΩΝ». Εκεί ο καθηγητής έχει τη δυνατότητα να δει τους βαθμούς όλων των μαθητών του για όλα τα μαθήματα σε ένα πίνακα. Βαθμοί με περιεχόμενο -1 σημαίνει ότι ο μαθητής δεν έχει λύσει ασκήσεις από αυτό το κεφάλαιο. Αυτό γίνεται για να διαφοροποιηθεί από την περίπτωση που ο βαθμός είναι 0 και σημαίνει ότι όντως ο μαθητής έχει όντως απαντήσει παντού λάθος με άσχετες απαντήσεις (Εικόνα 4.4.2.9). Όταν για κάποιο μαθητή ο καθηγητής πατήσει το κουμπί «Λεπτομέρειες...», έχει τη δυνατότητα να δει και τα είδη λαθών που έχει κάνει ο μαθητής (4.4.2.10).

Όνομα Χρήστη	Πρόσθεση	Αφαίρεση	Πολλαπλασιασμός	Διάρθρωση	Είδη λαθών
ilias	0	0	0	0	Δεπιτομέρειες...
user10	-1	-1	-1	-1	Δεπιτομέρειες...
user1	-1	-1	-1	-1	Δεπιτομέρειες...
user2	-1	-1	-1	-1	Δεπιτομέρειες...
student	43	-1	-1	-1	Δεπιτομέρειες...

Κεφάλαιο 1: Πρόσθεση
Κεφάλαιο 2: Αφαίρεση
Κεφάλαιο 3: Πολλαπλασιασμός
Κεφάλαιο 4: Διάρθρωση

Εικόνα 4.4.2.9. Ο καθηγητής παρακολουθεί τη βαθμολογία των μαθητών.



Εικόνα 4.4.2.10. Ο καθηγητής «teacher» βλέπει τα είδη λαθών του μαθητή «student».

Όταν βλέπει λεπτομέρειες για τα είδη λαθών κάποιου συγκεκριμένου μαθητή, η γραμμή για το συγκεκριμένο μαθητή στον πάνω πίνακα έχει τονιστεί με κόκκινο χρώμα. Τέλος, αν ο καθηγητής θέλει να κλείσει τον πίνακα λεπτομερειών με τα είδη λαθών του κάθε μαθητή, πατά το κουμπί «κλείσιμο».

5. Σύγκριση με άλλες εφαρμογές

Η εφαρμογή που δημιουργήσαμε έχει ομοιότητες με πολλές εφαρμογές απομακρυσμένης εκμάθησης μαθηματικών που υπάρχουν, όμως θα αναφερθεί μία σημαντική ομοιότητα με μία παλιά εφαρμογή που είχε αναπτυχθεί το 1975 από τον Brown και ονομάζεται BUGGY. Η ομοιότητα που έχει το BUGGY με την εφαρμογή μας είναι κυρίως έγκειται στο ότι και οι 2 εφαρμογές αναφέρονται στην αριθμητική, αλλά πολύ περισσότερο στο ότι και οι 2 εφαρμογές αναδεικνύουν στο μαθητή τα είδη λαθών που αυτός έχει κάνει. Συγκεκριμένα, η εφαρμογή μας δείχνει στο χρήστη σε κάθε άσκηση που επιλύει τι είδους λάθος ή λάθη έχει κάνει και του επισημαίνει τη σωστή απάντηση που θα έπρεπε να είχε δώσει. Στην εφαρμογή BUGGY δεν επισημαίνεται το σωστό. Εκτός όμως από ομοιότητες τα 2 συστήματα παρουσιάζουν και διαφορές. Η εφαρμογή μας έχει a priori κάποια πρότυπα είδη λαθών στο σύστημα τα οποία αυτή εξετάζει απέναντι στις απαντήσεις των χρηστών. Συγκεκριμένα, η εφαρμογή παρουσιάζει 8 είδη λαθών: Σύγχυση πρόσθεσης-αφαίρεσης, σύγχυση πρόσθεσης-πολλαπλασιασμού, σύγχυση πρόσθεσης-διαίρεσης, σύγχυση αφαίρεσης-πολλαπλασιασμού, σύγχυση αφαίρεσης-διαίρεσης, σύγχυση πολλαπλασιασμού-διαίρεσης, λάθη κρατούμενου και σοβαρά λάθη. Απεναντίας, η εφαρμογή BUGGY δε χρησιμοποιεί κάποια πρότυπα είδη λαθών που λαμβάνει υπόψη της απέναντι σε απαντήσεις χρηστών σε ασκήσεις, αλλά παρακολουθεί μέσα από την επίλυση ασκήσεων των μαθητών, τον τρόπο που σκέφτεται ο μαθητής και καταγράφει διάφορα είδη λαθών που προκύπτουν, περισσότερα σε αριθμό από την εφαρμογή μας. Αυτό μπορεί να το πετύχει και μέσω επίλυσης ασκήσεων από τους μαθητές και σε αλληλεπίδραση και με τους καθηγητές, όπου αυτοί μετά από «διάλογο» με το σύστημα μπορούν να καταγράφουν νέα είδη λαθών μέσα από παραδείγματα-ερωτήματα που τους θέτει το σύστημα και το BUGGY να τα βρίσκει. Αυτές οι διαφορές είναι λογικό να υπάρχουν καθώς το BUGGY έχει αναπτυχθεί ειδικά,

για ανεύρεση λανθασμένων συμπεριφορών σε πράξεις αριθμητικής, ενώ η εφαρμογή μας πιο πολύ εξετάζει τις 4 βασικές αριθμητικές πράξεις παρά είναι εξειδικευμένη στο να καταγράφει συμπεριφορές μαθητών και επομένως να κάνει και καταγραφή νέων λαθών. Παρακάτω, φαίνεται συνοπτικά και ένας πίνακας με τις ομοιότητες και τις διαφορές των 2 εφαρμογών.

Ομοιότητες ιστοσελίδας και BUGGY

- Και οι 2 εφαρμογές αναφέρονται σε αριθμητικές πράξεις.
- Κάνουν αναφορά στα είδη λαθών που κάνει ο μαθητής.

Διαφορές ιστοσελίδας και BUGGY

- Το BUGGY καταγράφει νέα είδη λαθών που προκύπτουν από αλληλεπίδραση με τους μαθητές μέσω ασκήσεων και από αλληλεπίδραση με τους καθηγητές, μέσω ερωταποκρίσεων.
- Η ιστοσελίδα μας έχει προκαθορισμένα 8 είδη λαθών απέναντι στα οποία εξετάζει τον τρόπο σκέψης των μαθητών στην επίλυση ασκήσεων.

6. Συμπεράσματα

Η εφαρμογή μας, εκμάθηση Μαθηματικών από απόσταση για παιδιά Α' και Β' Δημοτικού βοηθά τα παιδιά στην εκμάθηση των 4 βασικών πράξεων. Είναι μία εφαρμογή η οποία μέσω των διαφορετικών ειδών ασκήσεων, του συστήματος πληροφόρησης για τα είδη λαθών του και της αλληλεπίδρασης του χρήστη με άλλους μέσα από το φόρουμ, μπορεί να δώσει αρκετή γνώση στο μαθητή σχετικά με τον τρόπο που θα επιλύει βασικές μαθηματικές πράξεις. Η θεωρία που παρέχεται είναι απλή και κατανοητή με παραδείγματα και ο τρόπος επίλυσης των διαφορετικών ειδών ασκήσεων, μέσω των επιπέδων των χρηστών και του τρόπου που το σύστημα πληροφορεί το χρήστη για τα λάθη του και τον καθοδηγεί για το πώς έπρεπε να κινηθεί για να δώσει τη σωστή απάντηση, καθώς και ο τρόπος βαθμολόγησης των χρηστών, καθιστά το σύστημα αρκετά αξιόπιστο και αντικειμενικό και με χαρακτήρα αρκετά συμβουλευτικό. Βεβαίως, έχει πολλά περιθώρια βελτίωσης με πολλές προσθήκες. Οι προσθήκες αυτές θα μπορούσαν να είναι οι εξής:

- Δημιουργία καλύτερων γραφικών για την καλαισθησία της εφαρμογής.
- Εισαγωγή παιχνιδιών με τη μορφή εκπαιδευτικού χαρακτήρα ως επιβράβευση του μαθητή και για να μην καθιστούν το σύστημα ανιαρό.
- Εισαγωγή κάποιων πρακτόρων (agents), ώστε η καθοδήγηση του χρήστη να είναι πιο αποτελεσματική και ευχάριστη.
- Εισαγωγή περαιτέρω κεφαλαίων σχετικά με τα Μαθηματικά. Αυτό θα είχε ως αποτέλεσμα να αυξήσει και το φάσμα των μαθητών στο οποίο η ιστοσελίδα θα μπορούσε να φιλοξενήσει. Θα μπορούσε να υπάρχει μία μεγάλη γκάμα κεφαλαίων των Μαθηματικών και αυτά να αφορούσαν για παράδειγμα, μαθητές Δημοτικού, Γυμνασίου και Λυκείου. Χαρακτηριστικά παραδείγματα είναι, να προστεθούν κεφάλαια όπως η σύγκριση αριθμών, πράξεις με δεκαδικούς ή κλάσματα για μαθητές Δημοτικού, λύσεις εξισώσεων και ανισώσεων (πρωτοβάθμιων ή δευτεροβάθμιων), πράξεις με ρητούς και άρρητους αριθμούς για μαθητές Γυμνασίου, απόλυτες τιμές, λογαρίθμους και ανάλυση για μαθητές Λυκείου.

7. Βιβλιογραφία

1. Π. Πιντέλας, Εκπαιδευτικό Λογισμικό: Μία πρόταση για την εξασφάλιση της ποιότητάς του. <http://www.etpe.gr/files/proceedings/uploads/e3.pdf>

Μ.Κορδάκη, (2002) Σχεδιασμός και αξιολόγηση εκπαιδευτικού λογισμικού. http://www.ct.aegean.gr/people/t.daradoumis/4ETDE104/Paradeigmata%20Ekpaideytikoy%20Ylikou/3-Kefalaio_me%20sxolia.pdf

Παράρτημα Α - User Manual

Εγγραφή

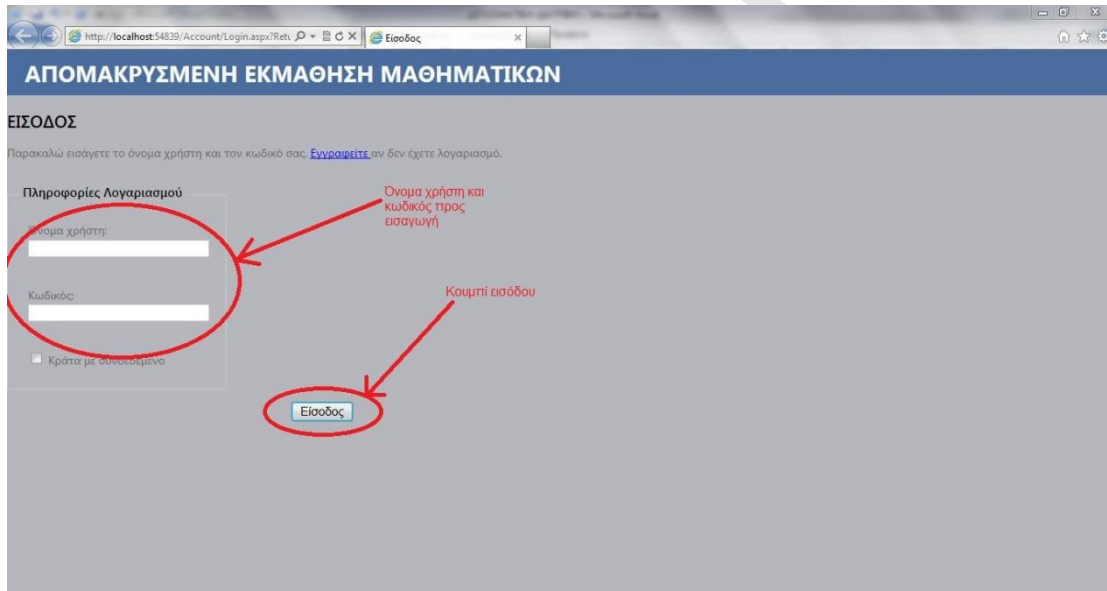
Για να αποκτήσετε πρόσβαση στην ιστοσελίδα, θα πρέπει να κάνετε μία εγγραφή πρώτα. Αυτό το πετυχαίνετε αν από τη σελίδα εισόδου (Login.aspx), πατήσετε πάνω στο κουμπί-σύνδεσμο «Εγγραφείτε» (Εικόνα 1). Αυτό θα σας οδηγήσει στη σελίδα Register.aspx.



Εικόνα 1. Για να εγγραφεί ο χρήστης θα πρέπει να πατήσει στο σύνδεσμο που δείχνει το κόκκινο βελάκι.

Στη σελίδα Register.aspx υπάρχουν κάποια κενά πεδία για να προσθέσετε τις πληροφορίες σας. Καταρχάς πρέπει να επιλέξετε ως τι είδους χρήστη θέλετε να κάνετε εγγραφή, καθηγητής ή μαθητής. Έπειτα σας ζητείται να προσθέσετε ένα όνομα χρήστη, ένα e-mail, τον επιθυμητό κωδικό σας και την επαλήθευσή του. Σημειώνουμε ότι ο κωδικός πρέπει να έχει μήκος τουλάχιστον 6 χαρακτήρες και δεν υπάρχει κανένας περιορισμός στο τι χαρακτήρα θα βάλετε (αλφαριθμητικό και μη). Η επαλήθευση κωδικού επίσης πρέπει να είναι ακριβώς ίδια με τον

κωδικό. Τέλος, ενδέχεται το όνομα χρήστη να υπάρχει ήδη στην εφαρμογή, οπότε επιλέξτε ένα διαφορετικό όνομα χρήστη, μέχρι το σύστημα να το κρίνει έγκυρο. Όταν πατήσετε το κουμπί «Εγγραφή» τότε ανάλογα με το αποτέλεσμα το σύστημα θα σας πληροφορήσει αν η εγγραφή ήταν επιτυχημένη ή όχι. Αν είναι επιτυχημένη τότε οδηγήστε στη σελίδα εισόδου Login.aspx μέσω του κουμπιού «Σελίδα Εισόδου». Όταν οδηγηθείτε στη σελίδα Login.aspx, τότε θα πρέπει να εισάγετε στα αντίστοιχα πεδία τα στοιχεία σας και να πατήσετε το κουμπί «Είσοδος» (Εικόνα 2). Το κουτάκι «Κράτα με συνδεδεμένο» αν τικαριστεί, σημαίνει πως σε μελλοντική είσοδο στο σύστημα δε θα χρειαστεί να εισάγετε πάλι τα στοιχεία σας, αλλά θα μπειτε κατευθείαν στην κεντρική σελίδα.

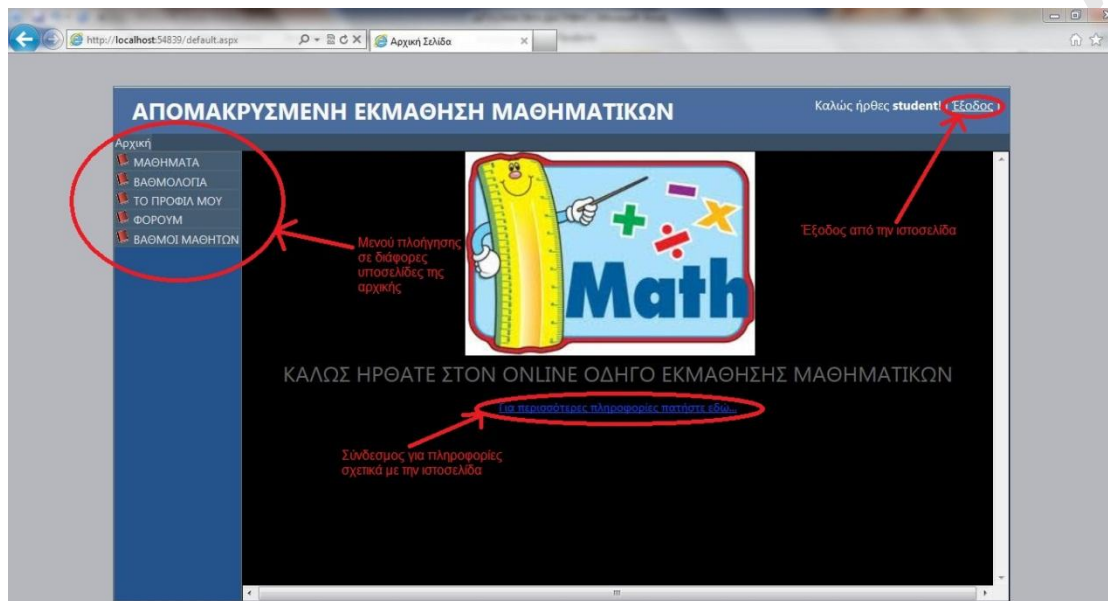


Εικόνα 2. Ο χρήστης πρέπει να εισάγει τα δεδομένα του (όνομα χρήστη και κωδικό) στα κενά πεδία και μετά να πατήσει το κουμπί «Είσοδος».

Εφόσον πατηθεί το κουμπί μπαίνετε αυτόματα στην κεντρική σελίδα.

Κεντρική Σελίδα (Default.aspx)

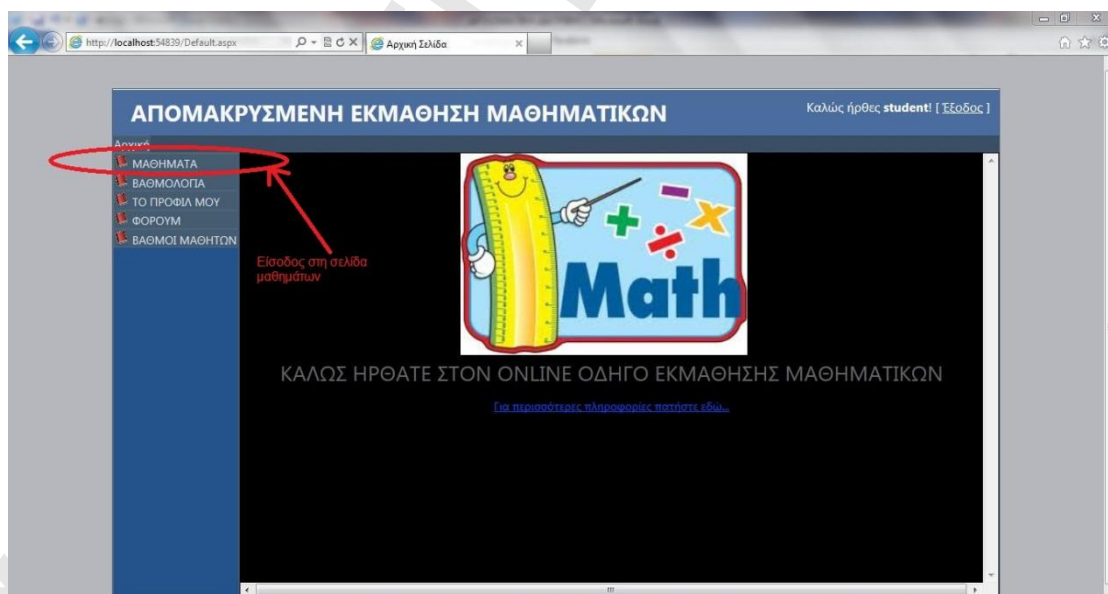
Ο χρήστης στην κεντρική σελίδα μπορεί να δει πληροφορίες σχετικά με τη σελίδα στο σύνδεσμο με μπλε στο κάτω μέρος της σελίδας, «Για περισσότερες πληροφορίες πατήστε εδώ» και να κάνει έξοδο από αυτό μέσω του κουμπιού «Έξοδος» πάνω δεξιά. Επίσης μπορεί να πλοηγηθεί στις διάφορες υποσελίδες μέσω ενός μενού που υπάρχει αριστερά καθ' όλη τη διάρκεια της πλοήγησής του στην ιστοσελίδα (Εικόνα 3).



Εικόνα 3. Κεντρική Σελίδα. Ο χρήστης μπορεί να δει πληροφορίες από τον κάτω υπερσύνδεσμο (με μπλε), να κάνει έξοδο από το σύστημα πάνω δεξιά, ή να πλοηγηθεί σε διάφορες σελίδες μέσω του μενού αριστερά.

Σελίδα μαθημάτων (Θεωρία και Ασκήσεις) (Courses.aspx)

Όταν ο χρήστης στην κεντρική σελίδα για να διαβάσει θεωρία και να επιλύσει ασκήσεις θα πρέπει να πατήσει στο σύνδεσμο «ΜΑΘΗΜΑΤΑ» στο μενού που βρίσκεται πάντα αριστερά του καθ' όλη την περιήγησή του στην ιστοσελίδα (Εικόνα 4). Εκεί θα μεταφερθεί στη σελίδα Courses.aspx.

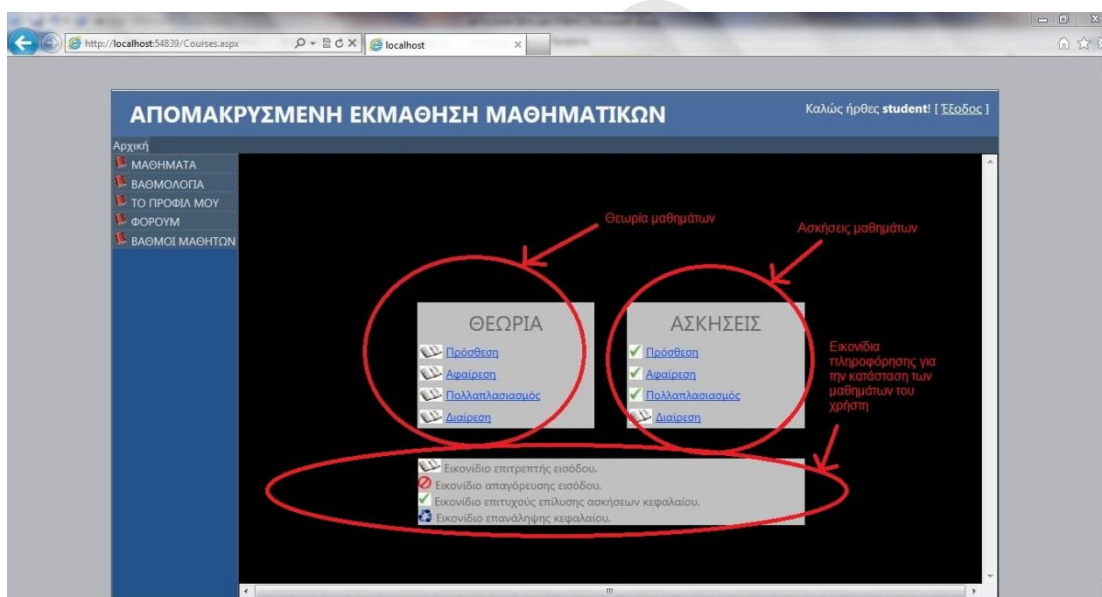


Εικόνα 4. Για να μπει ο χρήστης στη σελίδα μαθημάτων θα πρέπει να πατήσει στο σύνδεσμο «ΜΑΘΗΜΑΤΑ» στο μενού στα αριστερά της ιστοσελίδας.

Πρέπει να αναφέρουμε ότι ο τρόπος παρουσίασης της σελίδας Courses.aspx διαφέρει ανάλογα το ρόλο. Έτσι θα εξετάσουμε 2 διαφορετικές περιπτώσεις, μία για το ρόλο του μαθητή και μία για το ρόλο του καθηγητή.

i) Για το μαθητή:

Όταν εισέλθει στη σελίδα μαθημάτων, τότε για να διαβάσει τη θεωρία πρέπει να πατήσει στην αριστερή λίστα «ΘΕΩΡΙΑ» τους συνδέσμους που οδηγούν στη θεωρία των αντίστοιχων κεφαλαίων (Πρόσθεση, Αφαίρεση, Πολλαπλασιασμός, Διαίρεση). Στη δεξιά λίστα υπάρχουν οι ασκήσεις των αντίστοιχων κεφαλαίων που εξετάζονται. Στο κάτω μέρος βρίσκεται ένας επεξηγηματικός πίνακας σχετικά με τα εικονίδια που θα εμφανίζονται στα αντίστοιχα κεφάλαια για τις ασκήσεις. Συγκεκριμένα το εικονίδιο ανοιχτού βιβλίου, σημαίνει ότι ο χρήστης μπορεί να προσπελάσει και να επιλύσει τις ασκήσεις αυτού του κεφαλαίου. Το εικονίδιο με το απαγορευτικό σημαίνει ότι το κεφάλαιο είναι κλειδωμένο και δε μπορεί ο χρήστης να το προσπελάσει. Το εικονίδιο τικαρίσματος σημαίνει ότι ο χρήστης μόλις έλυσε επιτυχώς το εκάστοτε κεφάλαιο. Το εικονίδιο της ανακύκλωσης σημαίνει ότι ο χρήστης πρέπει να κάνει επανάληψη τη θεωρία για το εκάστοτε κεφάλαιο και πρέπει να επιλύσει ξανά τις ασκήσεις του κεφαλαίου για να το περάσει (Εικόνα 5).



Εικόνα 5. Στην αριστερή λίστα βρίσκονται οι θεωρίες των μαθημάτων. Στη δεξιά λίστα βρίσκονται τα μαθήματα που μπορεί να επιλύσει ο χρήστης. Κάτω βρίσκεται μια επεξηγηματική λίστα για τα εικονίδια που θα εμφανίζονται κατά την επίλυση των ασκήσεων στα αντίστοιχα κεφάλαια.

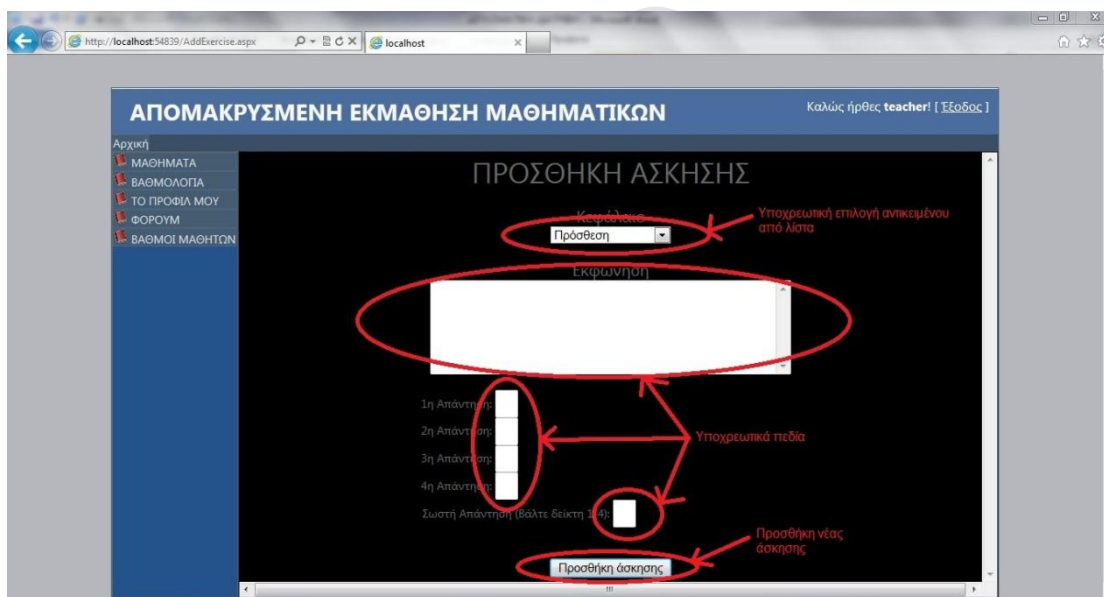
Όταν ο χρήστης μπει στη θεωρία της πρόσθεσης οδηγείται στη σελίδα Addition_theory.aspx, όταν μπει στη θεωρία της αφαίρεσης οδηγείται στη σελίδα Subtraction_theory.aspx, όταν μπει στη θεωρία του πολλαπλασιασμού οδηγείται στη σελίδα Multiplication_theory.aspx, όταν μπει στη θεωρία της διαίρεσης οδηγείται στη σελίδα Division_theory.aspx. Όταν πατήσει ένα από τους συνδέσμους της δεξιάς λίστας οδηγείται στη σελίδα Operations_exercises.aspx.

ii) Για τον καθηγητή:

Όταν εισέλθει στη σελίδα μαθημάτων ο καθηγητής μπορεί να δει ακριβώς ό, τι και ο μαθητής, με τη διαφορά ότι βλέπει και 2 άλλους συνδέσμους. Τους «Προσθήκη νέας άσκησης» και «Προβολή των ασκήσεων».

Σελίδα προσθήκης νέας άσκησης (AddExercise.aspx)

Όταν ο χρήστης πατήσει το κουμπί «Προσθήκη νέας άσκησης» μπορεί να δει τις εύκολες ασκήσεις που έχει προσθέσει. Να υπενθυμίσουμε εδώ ότι οι μέτριες και δύσκολες ασκήσεις παράγονται τυχαία από το σύστημα. Όταν πατήσει αυτό το κουμπί οδηγείται στη σελίδα AddExercise.aspx. Εκεί ο καθηγητής έχει τη δυνατότητα να προσθέσει μία νέα εύκολη άσκηση. Όλα τα πεδία είναι υποχρεωτικά, όπως και η επιλογή ενός μαθήματος από τη λίστα που βρίσκεται στην κορυφή και στα 5 τελευταία κενά κουτάκια πρέπει να προσθέσει αριθμητικές τιμές, ενώ στο τελευταίο εξ αυτών πρέπει να προσθέσει την τιμή 1 έως 4 για τη σωστή απάντηση που θα αφορά αντίστοιχα μία από τις 4 επιλογές. Ο χρήστης για να προσθέσει μία νέα άσκηση θα πρέπει να πατήσει το κουμπί «Προσθήκη άσκησης» για να καταχωρηθεί αυτή στη βάση δεδομένων (Εικόνα 6).



Εικόνα 6. Σελίδα προσθήκης άσκησης. Τα κενά πεδία όπως και η επιλογή αντικειμένου από λίστα είναι υποχρεωτικά. Στα 5 τελευταία πεδία πρέπει να μπει αριθμητική τιμή και στο τελευταίο συγκεκριμένα αριθμός από 1 έως 4. Το κουμπί «Προσθήκη νέας άσκησης» καταχωρεί την άσκηση στη βάση δεδομένων.

Σελίδα προβολής ασκήσεων (ViewExercise.aspx)

Ο καθηγητής για να παρατηρήσει τις ασκήσεις που έχει προσθέσει πρέπει να πατήσει το κουμπί «Προβολή των ασκήσεων». Τότε θα οδηγηθεί στη σελίδα ViewExercise.aspx. Εκεί θα δει ένα συγκεντρωτικό πίνακα με όλες τις εύκολες ασκήσεις που έχει προσθέσει για όλα τα κεφάλαια και μπορεί να πλοηγηθεί σε όλες αυτές μέσω δεικτοδότησης του πίνακα, όπου οι

δείκτες φαίνονται κάτω δεξιά. Σε κάθε δείκτη παρουσιάζονται 5 ασκήσεις, εκτός ίσως από τον τελευταίο δείκτη. Συγκεκριμένα το κεφάλαιο στο οποίο αναφέρεται η εκάστοτε άσκηση γίνεται αντιληπτό από τον αριθμό που υπάρχει στη στήλη «Κεφάλαιο». Από κάτω υπάρχει ένας επεξηγηματικός πίνακας για το τι σημαίνει κάθε νόμμερο. Ο καθηγητής μπορεί να επεξεργαστεί τις ασκήσεις και συγκεκριμένα να τροποποιήσει το περιεχόμενό τους ή να τις διαγράψει. Για να τις τροποποιήσει πρέπει να πατήσει το κουμπί «Edit». Τότε ο πίνακας γίνεται από read only, editable, αλλάζει το χρώμα στο φόντο από απαλό κίτρινο, σε απαλό κόκκινο και μπορεί ο χρήστης να διορθώσει μία εκφώνηση ή και να προσθέσει νέες απαντήσεις πολλαπλής επιλογής και σωστής απάντησης (Εικόνα 7). Αν πατήσει το κουμπί «Update» τότε η άσκηση τροποποιείται με το νέο περιεχόμενο, αλλιώς αν πατήσει «Cancel», τότε ακυρώνει την επιλογή του και ο πίνακας γίνεται πάλι read only.

	Κεφάλαιο	Εκφώνηση	1η Απάντηση	2η Απάντηση	3η Απάντηση	4η Απάντηση	Σωστή Απάντηση
Edit Delete Select	1	Βρείτε το αποτέλεσμα της πρόσθεσης 2 + 3.	6	1	5	8	3
Update Cancel	1	Βρείτε το αποτέλεσμα της πρόσθεσης 3 + 5.	8	15	10	2	1
Edit Delete Select	1	Βρείτε το αποτέλεσμα της πρόσθεσης 2 + 7.	10	14	5	9	4
Edit Delete Select	1	Βρείτε το αποτέλεσμα της πρόσθεσης 4 + 8.	2	12	32	4	2
Edit Delete Select	1	Βρείτε το αποτέλεσμα της πρόσθεσης 1 + 6.	7	5	6	11	1

Κεφάλαιο 1: Πρόσθεση
Κεφάλαιο 2: Αφαίρεση
Κεφάλαιο 3: Πολλαπλασιασμός
Κεφάλαιο 4: Διαίρεση

Εικόνα 7. Ο καθηγητής έχει πατήσει το κουμπί «Edit» και τροποποιεί το περιεχόμενο της εκάστοτε άσκησης.

Αν πατήσει το κουμπί «Delete» τότε η άσκηση στην οποία πάτησε το αντίστοιχο κουμπί θα διαγραφεί. Το κουμπί «Select» απλώς επιλέγει και κάνει highlight την αντίστοιχη άσκηση, όπου αλλάζει το χρώμα στο φόντο από απαλό κίτρινο, σε απαλό κόκκινο (Εικόνα 8).

	Κεφάλαιο	Εκφώνηση	1η Απάντηση	2η Απάντηση	3η Απάντηση	4η Απάντηση	Σωστή Απάντηση
Edit Delete Select	1	Βρείτε το αποτέλεσμα της πρόσθεσης 2 + 3.	6	1	5	8	3
Edit Delete Select	1	Βρείτε το αποτέλεσμα της πρόσθεσης 3 + 5.	8	15	10	2	1
Edit Delete Select	1	Βρείτε το αποτέλεσμα της πρόσθεσης 2 + 7.	10	14	5	9	4
Edit Delete Select	1	Βρείτε το αποτέλεσμα της πρόσθεσης 4 + 8.	2	12	32	4	2
Edit Delete Select	1	Βρείτε το αποτέλεσμα της πρόσθεσης 1 + 6.	7	5	6	11	1

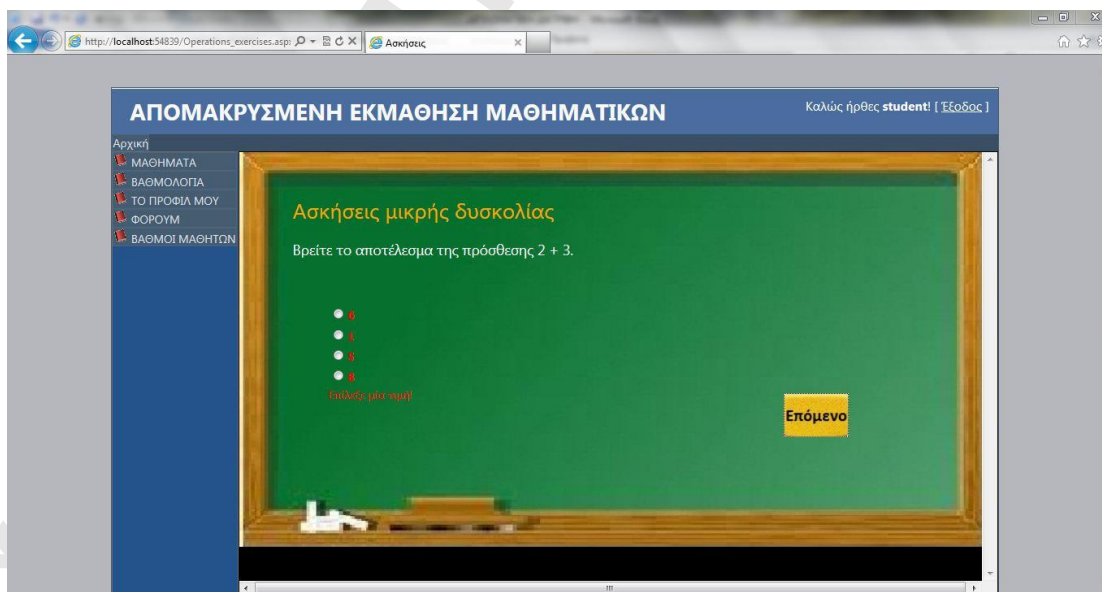
1 2 3 4 5 6 7 8 9

Κεφάλαιο 1: Πρόσθεση
Κεφάλαιο 2: Αφαίρεση
Κεφάλαιο 3: Πολλαπλασιασμός
Κεφάλαιο 4: Διαίρεση

Εικόνα 8. Επιλογή άσκησης.

Σελίδα ασκήσεων μαθημάτων (*Operations_exercises.aspx*)

Ανάλογα το επίπεδο του χρήστη και το τρέχον μάθημα στο οποίο εξετάζεται, η σελίδα αυτή έχει διαφορετικό περιεχόμενο κάθε φορά. Έτσι, αν ο χρήστης είναι αρχάριος θα λύσει 5 ασκήσεις απλές, 5 μέτριες και 5 δύσκολες σε όλα τα κεφάλαια. Αν είναι μέτριος θα λύσει 5 μέτριες και 5 δύσκολες σε όλα τα κεφάλαια. Αν είναι καλός θα λύσει μόνο 5 δύσκολες σε όλα τα κεφάλαια. Στις εύκολες ασκήσεις ο χρήστης θα πρέπει να επιλέξει οπωσδήποτε μία από τις 4 τιμές πολλαπλής επιλογής και μετά να πατήσει στο κουμπί «Επόμενο» ή «Ασκήσεις Μέτριας Δυσκολίας», ανάλογα αν είναι η 5^η ερώτηση ή όχι. Ειδάλλως, το σύστημα θα τον προτρέψει να πράξει αναλόγως (Εικόνα 9). Κάθε φορά ο χρήστης για να πάει σε επόμενη ερώτηση θα πρέπει να πατήσει πάνω στο κουμπί-σφουγγάρι.

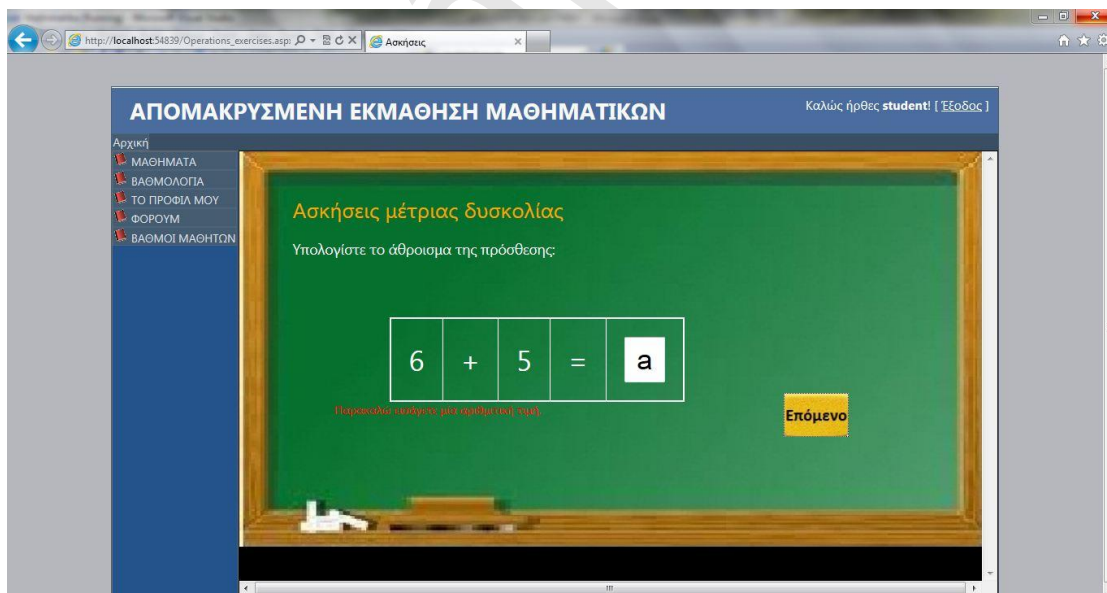


Εικόνα 9. Ο χρήστης πρέπει οπωσδήποτε να επιλέξει μία τιμή.

Στις μέτριες ασκήσεις ο χρήστης θα πρέπει να βάλει οπωσδήποτε μία τιμή στο κενό πεδίο και μάλιστα αριθμητική, ειδάλλως το σύστημα θα τον προτρέψει να πράξει ανάλογα (Εικόνες 10, 11).



Εικόνα 10. Ο χρήστης πρέπει να προσθέσει μία τιμή οπωσδήποτε.



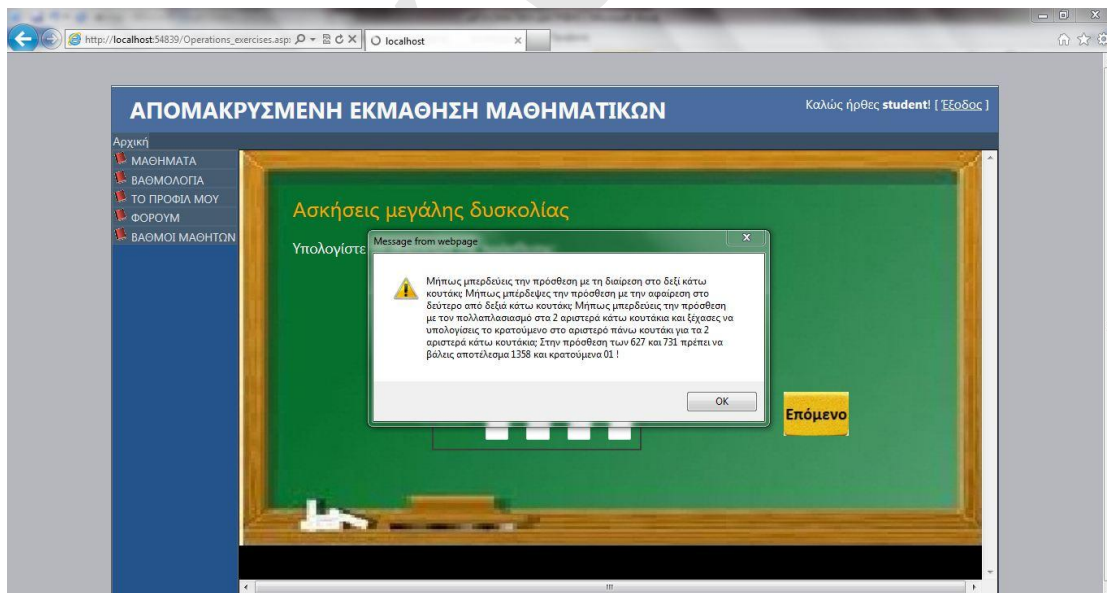
Εικόνα 11. Ο χρήστης πρέπει να προσθέσει μία αριθμητική τιμή.

Στις δύσκολες ασκήσεις όπως και στις μέτριες ο χρήστης πρέπει να συμπληρώσει όλα τα κενά πεδία υποχρεωτικά, ακόμα και αν αυτά είναι μηδέν και μάλιστα αριθμητικές τιμές (Εικόνα 12).



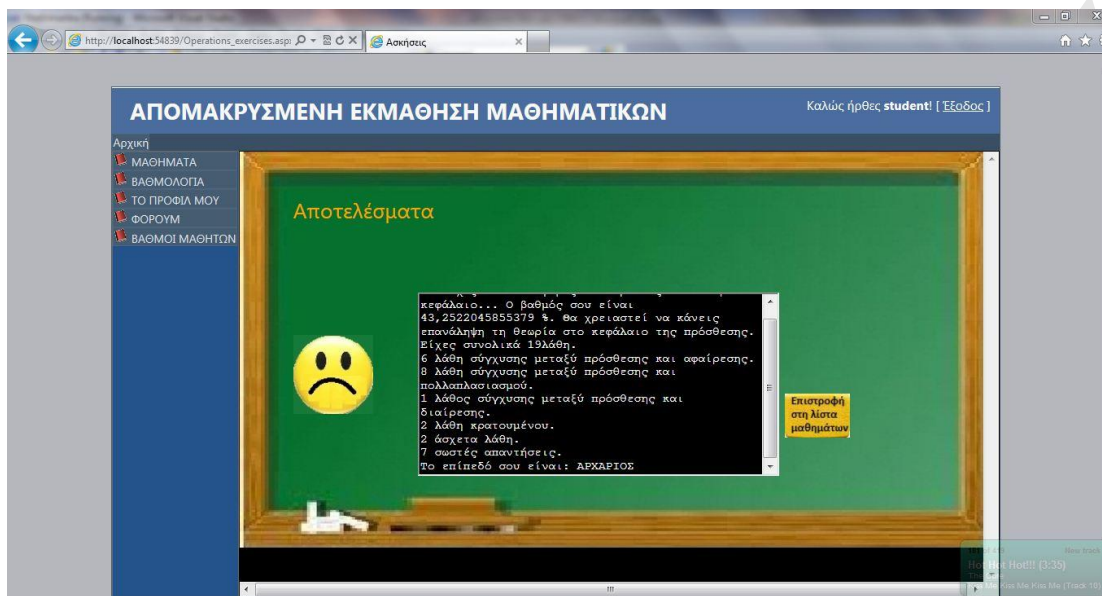
Εικόνα 12. Ο χρήστης έχει αφήσει κάποια κενά πεδία και σε άλλα έχει προσθέσει μη αριθμητικές τιμές.

Κατά τη διάρκεια της επίλυσης ασκήσεων το σύστημα θα καθοδηγεί το χρήστη για το αν έκανε σωστά ή λάθος σε κάποια ερώτηση. Συγκεκριμένα όταν η απάντηση είναι σωστή το σύστημα δε θα προβάλει κάποιο μήνυμα, ενώ όταν είναι λάθος η απάντηση τότε το σύστημα θα του βγάλει σχετικό μήνυμα με το ότι έκανε λάθος, το είδος λάθους που έκανε και το τι θα έπρεπε να είχε βάλει ή επιλέξει (Εικόνα 13).



Εικόνα 13. Το σύστημα πληροφορεί το χρήστη για το είδος λάθους που έκανε και για το τι θα έπρεπε να είχε προσθέσει στα κουτάκια.

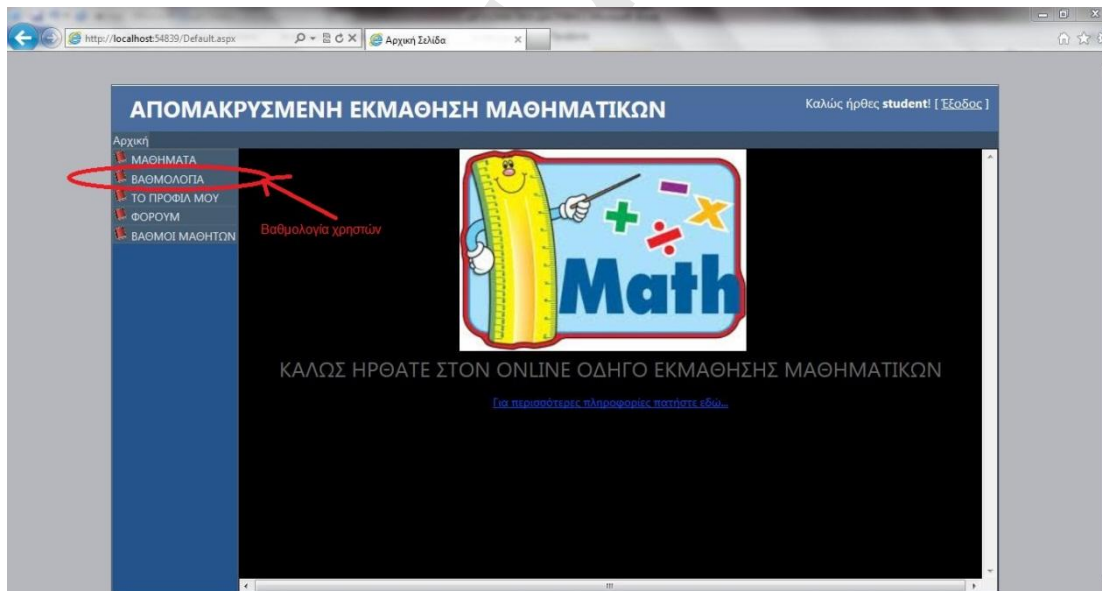
Στο τέλος των ασκήσεων παρουσιάζονται τα αποτελέσματα στο χρήστη και συγκεκριμένα το σκορ του, τα είδη λαθών που έκανε και με αριθμούς και το επίπεδό του (Εικόνα 14).



Εικόνα 14. Πίνακας αποτελεσμάτων.

Σελίδα Βαθμολογίας (Marks.aspx)

Για να παρακολουθήσει ένας χρήστης τη βαθμολογία του καθώς και τα είδη λαθών του θα πρέπει στο μενού αριστερά να πατήσει στο σύνδεσμο «ΒΑΘΜΟΛΟΓΙΑ» (Εικόνα 15).

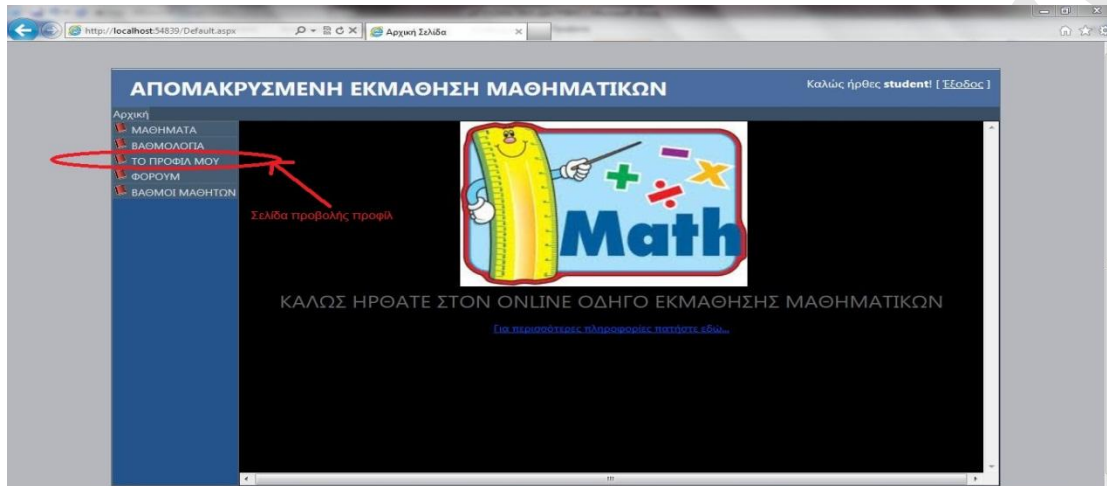


Εικόνα 15. Ο χρήστης για να μπει στη σελίδα βαθμολογίας του θα πρέπει να πατήσει το σύνδεσμο «ΒΑΘΜΟΛΟΓΙΑ».

Τότε ο χρήστης θα πλοηγηθεί στη σελίδα Marks.aspx. Εκεί ο χρήστης έχει τη δυνατότητα να δει τα μαθήματα που έχει περάσει καθώς και τους βαθμούς που έχει πάρει σε καθένα από αυτά, όπως και τα είδη λαθών που έχει κάνει σε ποσοστιαίες μονάδες. Μάθημα με βαθμό -1 σημαίνει πως ο χρήστης δεν έχει εξεταστεί στο μάθημα αυτό.

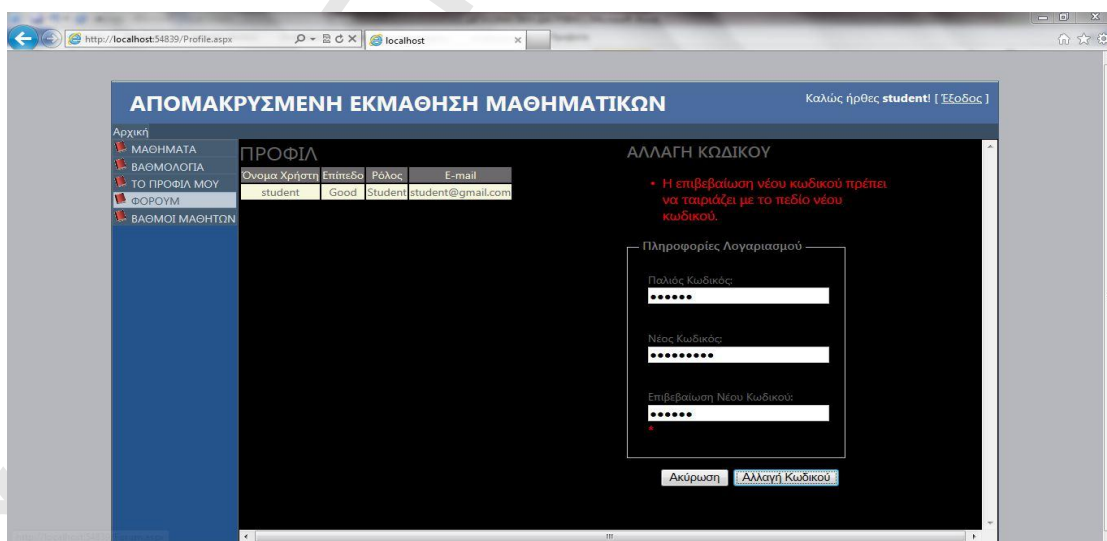
Σελίδα προβολής προφίλ (Profile.aspx)

Για να δει ο χρήστης το προφίλ του θα πρέπει στο μενού αριστερά να πατήσει στο σύνδεσμο «ΤΟ ΠΡΟΦΙΛ ΜΟΥ» (Εικόνα 16).



Εικόνα 16. Ο χρήστης για να μπει στη σελίδα για προβολή του προφίλ του θα πρέπει να πατήσει στο σύνδεσμο «ΤΟ ΠΡΟΦΙΛ ΜΟΥ».

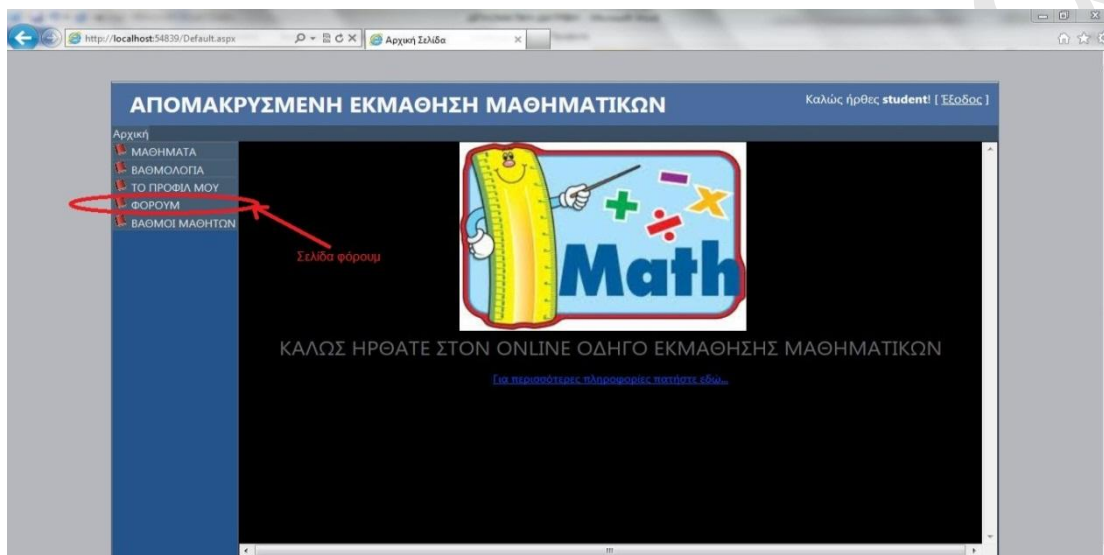
Όταν πατήσει το σύνδεσμο αυτό οδηγείται στη σελίδα προβολής του προφίλ του (Profile.aspx). Στη σελίδα αυτή μπορεί στο αριστερό μέρος να δει πληροφορίες για το προφίλ του και συγκεκριμένα το όνομα χρήστη του, το e-mail του, το ρόλο που έχει στην ιστοσελίδα (καθηγητής ή μαθητής) και το επίπεδό του. Στο δεξί μέρος μπορεί αν θέλει να αλλάξει τον κωδικό του. Συγκεκριμένα για να το κάνει αυτό θα πρέπει να βάλει τον παλιό του κωδικό σωστά, να πληκτρολογήσει το νέο του κωδικό σύμφωνα με τους κανόνες που υπήρχαν και στην εγγραφή του, δηλαδή να είναι ο κωδικός τουλάχιστον 6 χαρακτήρες σε μήκος με κανένα περιορισμό ως προς το είδος των χαρακτήρων και τέλος η επιβεβαίωση του κωδικού του να είναι ταυτόσημη με το νέο του κωδικό. Στην εικόνα παρακάτω φαίνεται ένα παράδειγμα που ο χρήστης δεν πρόσθεσε σωστά τα πεδία (Εικόνα 17).



Εικόνα 17. Ο χρήστης βάζει διαφορετική επιβεβαίωση από το νέο του κωδικό.

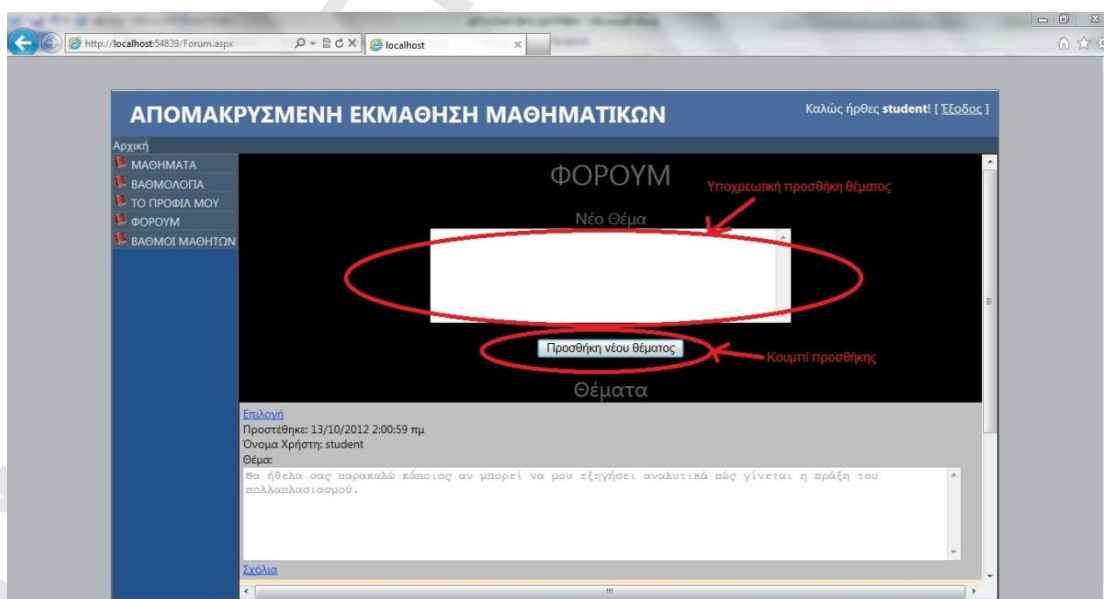
Σελίδα φόρουμ (Forum.aspx)

Ο χρήστης για να μπει στο φόρουμ πρέπει να πατήσει στο σύνδεσμο «ΦΟΡΟΥΜ» στο μενού αριστερά της ιστοσελίδας (Εικόνα 18). Όταν το πατήσει τότε θα οδηγηθεί στη σελίδα φόρουμ (Forum.aspx).



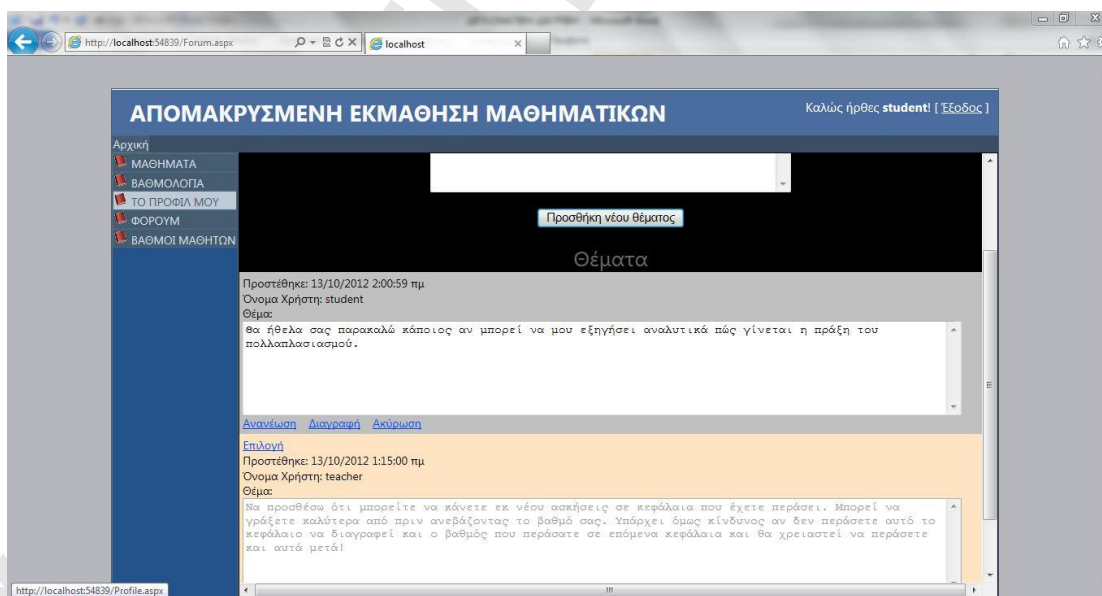
Εικόνα 18. Σύνδεσμος για είσοδο στη σελίδα φόρουμ.

Όταν εισέλθει στη σελίδα φόρουμ, μπορεί να συμμετέχει ενεργά σε αυτό μαζί με άλλους χρήστες και να ανταλλάξουν μηνύματα. Συγκεκριμένα ο τρόπος συμμετοχής του είναι να προσθέτει ένα καινούργιο θέμα προς συζήτηση, ή να σχολιάζει θέματα άλλων χρηστών. Για να προσθέσει ένα νέο θέμα θα πρέπει να προσθέσει την σκέψη του στο κενό πεδίο που βρίσκεται πάνω και κεντρικά της σελίδας. Για να γίνει η καταχώρηση στη βάση θα πρέπει να πατήσει το κουμπί «Προσθήκη νέου θέματος» (Εικόνα 19). Τότε αυτόματα προβάλλεται στο χρήστη το νέο θέμα στη λίστα με τα θέματα που υπάρχουν από κάτω.

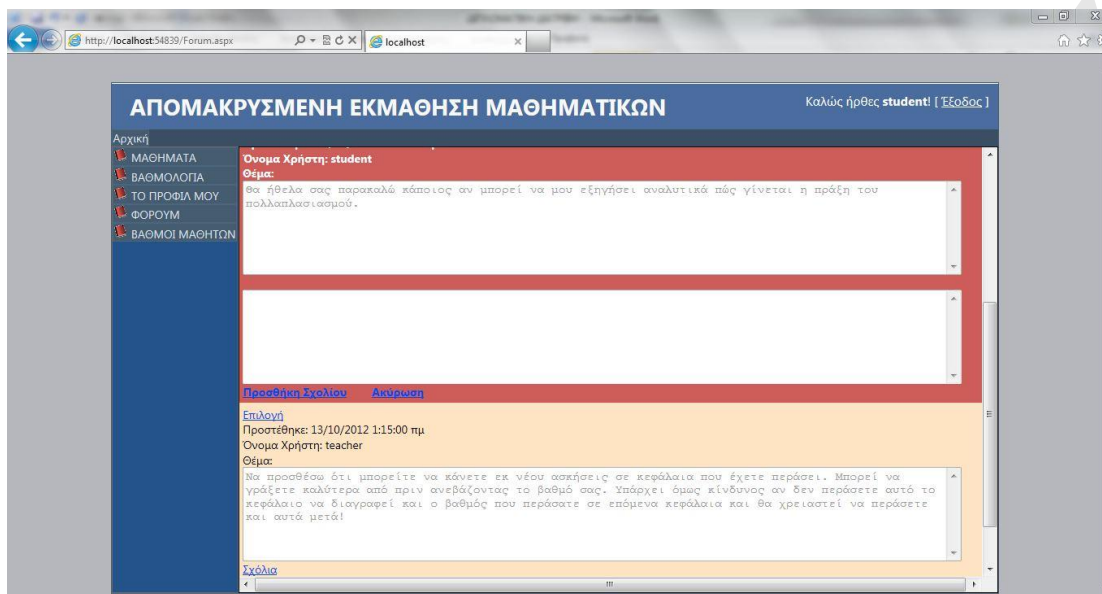


Εικόνα 19. Ο χρήστης για να προσθέσει ένα νέο θέμα πρέπει να προσθέσει υποχρεωτικά κάτι στο κενό πεδίο. Με το κουμπί «Προσθήκη νέου θέματος» καταχωρείται το θέμα στη βάση. Από κάτω βρίσκεται η λίστα θεμάτων.

Στο θέμα που πρόσθεσε είναι καταγεγραμμένο το όνομα χρήστη που το έγραψε, η ημερομηνία και ώρα που έγινε αυτό, το θέμα που προστέθηκε και 2 κουμπιά, τα «Επιλογή» και «Σχόλια» (Εικόνα 20). Ο χρήστης μπορεί να επεξεργαστεί τα υπάρχοντα θέματα που έχει προσθέσει. Δε μπορεί να επεξεργαστεί θέματα που έχουν προσθέσει άλλοι χρήστες. Αυτό μπορεί να το κάνει με το κουμπί «Επιλογή». Συγκεκριμένα όταν πατηθεί αυτό το κουμπί ο χρήστης μπορεί να διαγράψει το θέμα ή να το τροποποιήσει με άλλα 2 κουμπιά που εμφανίζονται κάτω από το θέμα, τα «Ανανέωση» και «Διαγραφή». Το κουμπί «Ακύρωση» απλά ακυρώνει την επιλογή του. Όταν διαγράφεται ένα θέμα, τότε διαγράφονται όπως είναι λογικό και τα σχόλια που έχουν γραφτεί για αυτό. Το κουμπί «Σχόλια» επιτρέπει στο χρήστη να προσθέσει ένα νέο σχόλιο προς το θέμα στο οποίο πάτησε το συγκεκριμένο κουμπί. Συγκεκριμένα όταν πατηθεί αυτό το κουμπί το αντίστοιχο θέμα αλλάζει χρώμα (κόκκινο) και 2 άλλα κουμπιά εμφανίζονται στη θέση του κουμπιού «Σχόλια», τα «Προσθήκη Σχολίου» και «Ακύρωση» (Εικόνα 21). Επίσης κάτω από το υπό σχολιασμό θέμα προβάλλεται προς το χρήστη ένα μεγάλο κενό πεδίο στο οποίο ο χρήστης μπορεί να προσθέσει τα δικά του σχόλια προς το υπό συζήτηση θέμα. Με το πάτημα του κουμπιού «Προσθήκη Σχολίου» το σχόλιο καταχωρείται στη βάση και παρουσιάζεται ακριβώς από κάτω στη λίστα σχολίων. Όταν θέλει ο χρήστης να προβάλλει τα σχόλια από ένα συγκεκριμένο θέμα με το κουμπί «Σχόλια», τότε τα αντίστοιχα σχόλια για το θέμα αυτό εμφανίζονται. Με το κουμπί «Ακύρωση» ο χρήστης ακυρώνει την επιλογή εισαγωγής σχολίου. Επίσης, αν δεν υπάρχουν σχόλια για ένα θέμα ένα σχετικό μήνυμα εμφανίζεται κάτω από τη λίστα θεμάτων.



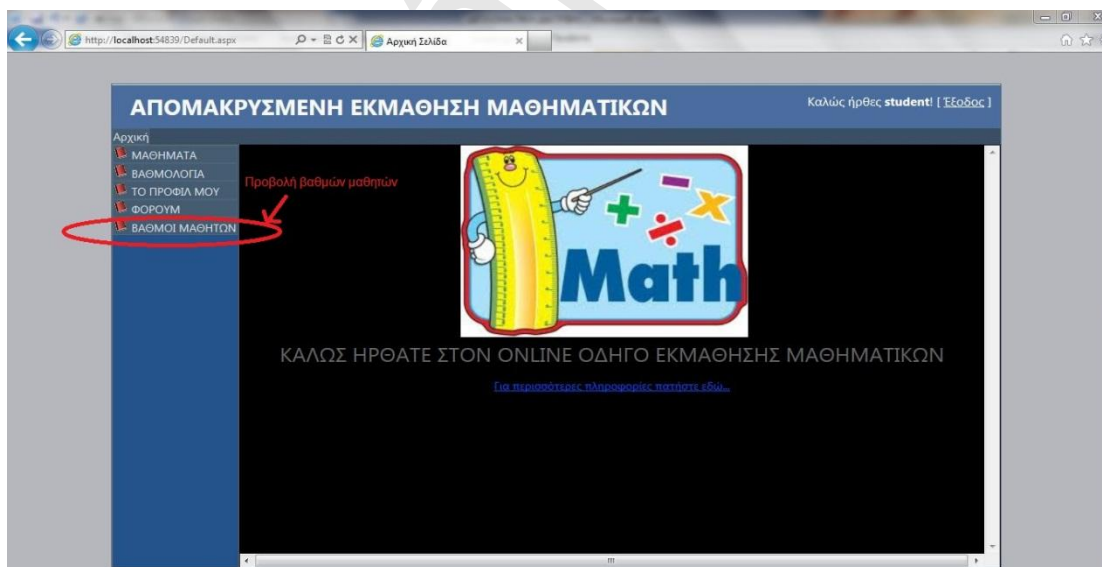
Εικόνα 20. Όταν ο χρήστης πατήσει το κουμπί «Επιλογή» τότε μπορεί να επεξεργαστεί το εκάστοτε θέμα που έχει προσθέσει από πριν.



Εικόνα 21. Όταν ο χρήστης πατήσει το κουμπί «Σχόλια» τότε μπορεί εμφανίζονται τα σχετικά προς το θέμα σχόλια και μπορεί να κάνει προσθήκη ενός νέου σχολίου ή να ακυρώσει την επιλογή του.

Σελίδα προβολής βαθμολογίας μαθητών (ViewMarks.aspx)

Για να μπει ο χρήστης στη σελίδα προβολής των βαθμών των μαθητών θα πρέπει να πατήσει στο σύνδεσμο «ΒΑΘΜΟΙ ΜΑΘΗΤΩΝ», στο αριστερό μενού της ιστοσελίδας (Εικόνα 22).

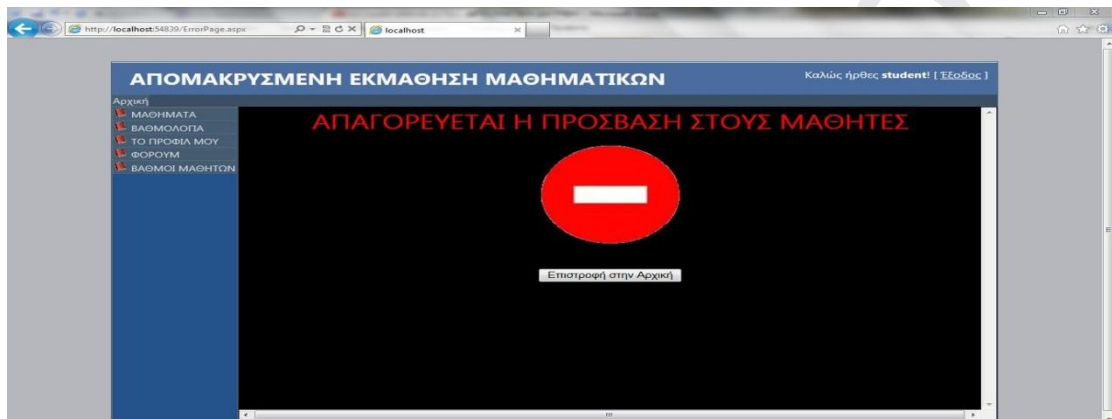


Εικόνα 22. Σύνδεσμος προβολής βαθμολογίας μαθητών.

Εδώ πρέπει να αναφέρουμε ότι η συγκεκριμένη σελίδα έχει διαφορετικό τρόπο παρουσίασης στο ρόλο του μαθητή και στο ρόλο του καθηγητή. Για αυτό θα εξετάσουμε διαφορετικά τις 2 περιπτώσεις.

i) Για το μαθητή:

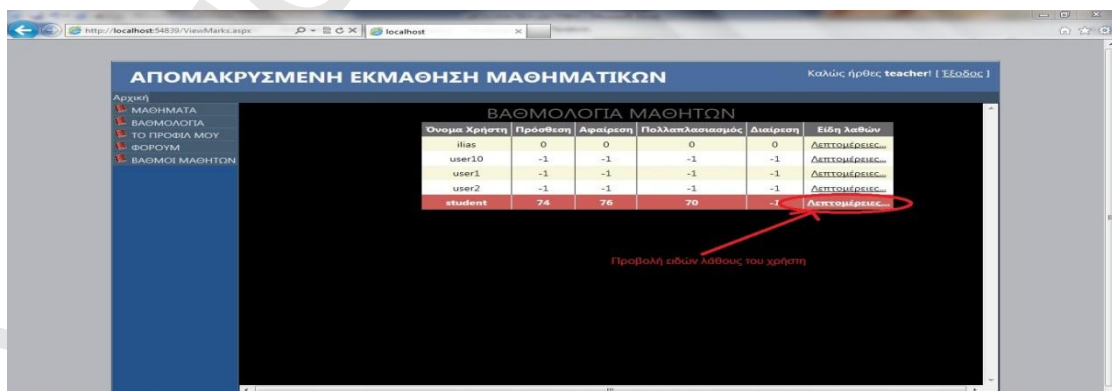
Όταν ο μαθητής πατήσει το κουμπί «ΒΑΘΜΟΛΟΓΙΑ» τότε εισέρχεται στη σελίδα ErrorPage.aspx η οποία τον πληροφορεί ότι η συγκεκριμένη σελίδα που προσπαθεί να προσπελάσει (η ViewMarks.aspx), αφορά μόνο καθηγητές. Παρακάτω φαίνεται η σελίδα ErrorPage.aspx (Εικόνα 23).



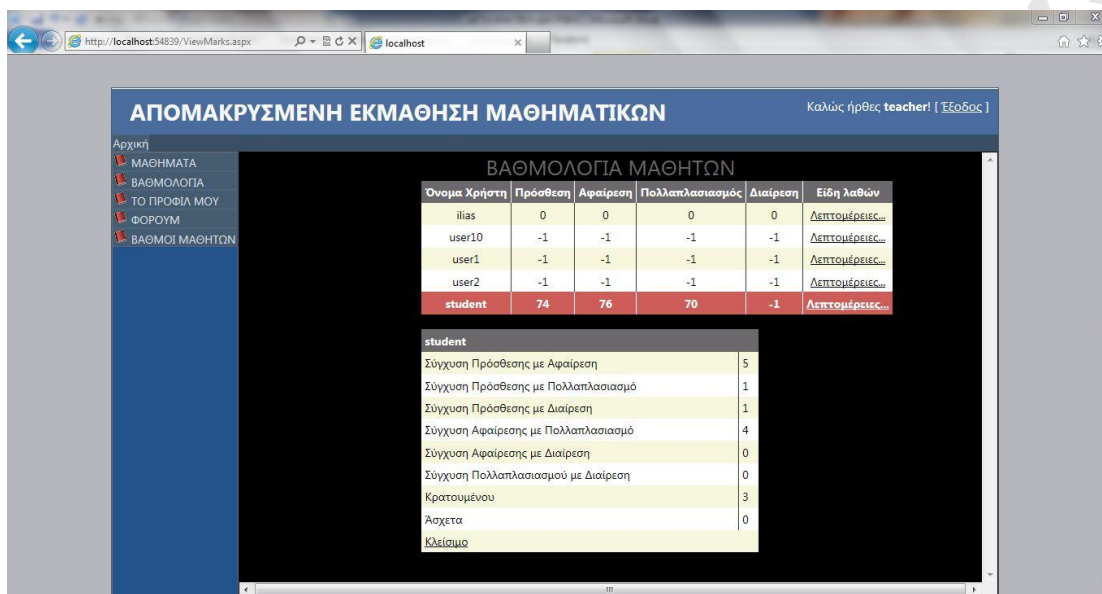
Εικόνα 23. Σελίδα εσφαλμένης εισόδου για τους μαθητές.

ii) Για τον καθηγητή:

Όταν πατήσει το σύνδεσμο «ΒΑΘΜΟΙ ΜΑΘΗΤΩΝ» ο καθηγητής, τότε οδηγείται στη σελίδα ViewMarks.aspx. Στη σελίδα αυτή ο καθηγητής έχει τη δυνατότητα να παρατηρήσει τη βαθμολογία των μαθητών του σε όλα τα μαθήματα, καθώς να δει και τα είδη λαθών που αυτός έχει κάνει με το πάτημα του κουμπιού «Λεπτομέρειες» (Εικόνες 24,25). Αν ο χρήστης θέλει να κλείσει τον πίνακα με τα είδη λάθους πρέπει να πατήσει το κουμπί «Ακύρωση».



Εικόνα 24. Σελίδα προβολής βαθμολογιών μαθητών. Ο καθηγητής για να δει τα είδη λαθών του κάθε μαθητή πρέπει να πατήσει το αντίστοιχο κουμπί «Λεπτομέρειες...».



Εικόνα 25. Προβολή ειδών λάθους για το χρήστη «student».

ΠΑΡΑΡΤΗΜΑ Β - Τμήματα Κώδικα

Στο παράρτημα αυτό θα παραθέσουμε τα πιο σημαντικά σημεία του κώδικά μας. Τα αποσπάσματα του κώδικα που θα ακολουθήσουν είναι γραμμένα σε C# και η server side γλώσσα που χρησιμοποιήθηκε είναι η ASP.NET.

Η κλάση *Operations_exercises*

Στην κλάση αυτή υπάρχει πληθώρα μεθόδων και θα εξηγηθεί κάθε μία από αυτές:

`void Operations_exercises.launch_exercise():`

Η μέθοδος αυτή χρησιμοποιείται για την εισαγωγή των ασκήσεων. Μέσω αυτής καλούνται οι μέθοδοι `launch_easy_exercises()`, `launch_medium_exercises()` και `launch_difficult_exercises()`, αλλά κάθε φορά καλούνται με διαφορετικό τρόπο αυτές, ανάλογα το επίπεδο του χρήστη, μέσω ενός μιας εντολής `switch`. Συγκεκριμένα αν ο χρήστης είναι «Αρχάριος» τότε καλείται 5 φορές η `launch_easy_exercises()`, άλλες 5 η `launch_medium_exercises()` και άλλες 5 τέλος η `launch_difficult_exercises()`. Αν ο χρήστης είναι «Μέτριος» τότε καλείται 5 φορές η `launch_medium_exercises()` και άλλες 5 η `launch_difficult_exercises()`. Αν ο χρήστης είναι «Καλός», τότε καλείται 5 φορές η `launch_difficult_exercises()`. Το επίπεδο του χρήστη βρίσκεται από τη συνάρτηση `FindLevel()`.

`void Operations_exercises.launch_easy_exercises():`

Η μέθοδος αυτή χρησιμοποιείται για την εισαγωγή των εύκολων ασκήσεων στη σελίδα `Operations_exercises.aspx` και για την προβολή των αποτελεσμάτων στο τέλος κάθε κεφαλαίου. Συγκεκριμένα καλεί τη μέθοδο `CurrentCourses.FetchCurrentCourse()` για να εξετάσει ποιο είναι το τωρινό υπό εξέταση μάθημα και ανάλογα την απάντηση, τότε ανακτά τις εύκολες ασκήσεις του τωρινού κεφαλαίου με τυχαίο τρόπο. Ο τυχαίος αυτός τρόπος επιτυγχάνεται με τη μέθοδο

`int Randomize(int , int [])` που παίρνει σαν πρώτο όρισμα το μήκος του δεύτερου ορίσματος που είναι το σύνολο των εύκολων ασκήσεων για το τωρινό κεφάλαιο. Επιστρέφει έναν ακέραιο που είναι ο τυχαίος δείκτης που θα εφαρμοστεί πάνω στη μήτρα που είναι το δεύτερο όρισμα της μεθόδου. Μετά καταγράφει στη στατική μήτρα ιστορικού ασκήσεων `history[]` την άσκηση που επιλέχθηκε για να εμφανιστεί. Επικοινωνεί με τη βάση, εισάγει τα δεδομένα της επιλεγείσας άσκησης και αυξάνει το δείκτη ιστορικού `history_array_counter` κατά ένα. Όταν θέλει να προβάλει τα αποτελέσματα στο τέλος κάθε κεφαλαίου τότε καλεί την `ExportResults(double, int[])`.

`void Operations_exercises.launch_medium_exercises():`

Η μέθοδος αυτή καλεί επίσης την `CurrentCourses.FetchCurrentCourse()` για να ελέγξει το τωρινό κεφάλαιο και ανάλογα την απάντηση παράγει τυχαίους αριθμούς ως τελεστές για πρόσθεση, αφαίρεση, πολλαπλασιασμό , ή διαίρεση με τη βοήθεια της έτοιμης συνάρτησης `Random.Next(int, int)`.

`void Operations_exercises.launch_difficult_exercises():`

Η μέθοδος αυτή όπως ακριβώς και η προηγούμενη καλεί την `CurrentCourses.FetchCurrentCourse()` για να ελέγξει το τωρινό κεφάλαιο και ανάλογα την απάντηση παράγει τυχαίους αριθμούς ως τελεστές για πρόσθεση, αφαίρεση, πολλαπλασιασμό , ή διαίρεση με τη βοήθεια της έτοιμης συνάρτησης `Random.Next(int, int)`.

`int Operations_exercises.Randomize(int, int[]):`

Η μέθοδος αυτή που προαναφέρθηκε και πριν είναι μία αναδρομική συνάρτηση. Δέχεται ως πρώτο όρισμα το μήκος της μήτρας που μπαίνει σε δεύτερο όρισμα, η οποία εκφράζει τις (εύκολες) ασκήσεις του κεφαλαίου από τις οποίες θα επιλεγεί σε κάθε κλήση μία. Η μέθοδος μέσω ενός βρόχου `for` ελέγχει τα στοιχεία αυτά της μήτρας με τη μήτρα ιστορικού `if(array[random_number - 1] != history[i])` και αν το δεδομένο στοιχείο που επιλέχθηκε της μήτρας `array (random_number - 1)` είναι διάφορο όλων των στοιχείων του ιστορικού τότε επιλέγεται αυτό το `random_number`, ειδώς ξανακαλείται αναδρομικά η `Random` παράγοντας άλλο αριθμό. Όπως καταλαβαίνουμε αυτή η συνάρτηση μπορεί να προκαλέσει `stack overflow`.

`string Operations_exercises.FindLevel(string):`

Η μέθοδος αυτή βρίσκει το επίπεδο κάποιου χρήστη. Σαν όρισμα παίρνει το όνομα χρήστη του χρήστη που αλληλεπιδρά άμεσα με το σύστημα (`User.Identity.Name`). Επιστρέφει σε μορφή `string` το επίπεδό του.

`void Operations_exercises.ImageButton1_Click():`

Η μέθοδος αυτή είναι και η πιο πολύπλοκη της εφαρμογής. Θα επεξηγηθεί αναλυτικά. Είναι ουσιαστικά η μέθοδος η οποία καλείται όταν ο χρήστης πατήσει το κουμπί «Επόμενο» ή «Μέτριες Ασκήσεις» ή «Δύσκολες Ασκήσεις» και αξιολογεί την απάντηση του χρήστη. Οδηγός σε όλη τη μέθοδο είναι η στατική μεταβλητή `click_count` που μετρά τον αριθμό των φορών που έχει πατηθεί το κουμπί που είπαμε και επομένως τον αριθμό των ασκήσεων που έχει λύσει ο χρήστης μέχρι εκείνη τη στιγμή. Ανάλογα τη μεταβλητή αυτή και το επίπεδο του χρήστη παίρνουμε διαφορετικές περιπτώσεις.

Όταν ο χρήστης εξετάζεται στις εύκολες ερωτήσεις, τότε τσεκάρεται το αποτέλεσμα που έδωσε με το αποτέλεσμα των 2 τελεστών, `if (RadioButtonList1.SelectedValue == (operand1 op`

`operand2).ToString()`), όπου `op` ένα εκ των `+`, `-`, `X`, `/`. Αν η παραπάνω συνθήκη είναι αληθής τότε η απάντηση είναι σωστή για την αντίστοιχη πράξη. Τότε αυξάνεται η τιμή του σκορ κατά ένα (`result++`), αυξάνεται ο αριθμός των ερωτήσεων κατά ένα (`UpdateAnswer()`) και καλείται και η μέθοδος `ReduceConfusedOperatorErrorIndexes(string, string, string, string, string)`, η οποία ουσιαστικά μειώνει τα `error_indexes` των άλλων λαθών εκτός του λάθους κρατουμένου, διότι λάθος κρατουμένου στις εύκολες ασκήσεις δε μπορεί να υπάρξει. Αν όμως η απάντηση είναι λανθασμένη τότε η τιμή του `result` θα γίνει ίση με `result + 1/(2 + error_index)`, όπου το `error_index` μπορεί να είναι μία από τις τιμές, `sum-sub_error_index`, `sum-mult_error_index`, `sum-div_error_index`, `sub-mult_error_index`, `sub-div_error_index`, `mult-div_error_index`, `serious_error_index`. Την ποσότητα αυτή θα την πάρει μέσω κλήσης της μεθόδου `ConfusedOperators(string, bool, int, string)`. Δηλαδή, ο χρήστης παίρνει ένα ποσοστό της μονάδας, ανάλογα με το είδος λάθους που έχει κάνει. Καλείται η `UpdateWrongAnswer()`, η οποία αυξάνει το συνολικό αριθμό των λαθών κατά 1 και εκτυπώνει μήνυμα λάθους στην οθόνη `MessageBox()`. Είτε είναι σωστή πάντως είτε λάθος η απάντηση υπάρχει μία στατική μήτρα, η `current_errors[]` η οποία περιέχει όλα τα είδη λαθών, καθώς και τις σωστές απαντήσεις και χρησιμοποιείται στην προβολή των αποτελεσμάτων για την προβολή των τρεχόντων λαθών που έκανε στις ασκήσεις του κεφαλαίου. Όταν ο χρήστης εξετάζεται στις μέτριες απαντήσεις ακολουθείται η ίδια διαδικασία που εκτελείται και στις εύκολες ερωτήσεις.

Στις δύσκολες απαντήσεις τώρα, ακολουθείται μία λίγο διαφορετική λογική. Σε κάθε είδους κεφάλαιο και σε κάθε φάση της δύσκολης ερώτησης (κάθε φάση εξετάζεται μέσα σε ένα βρόχο `for`), εξετάζεται ένα σύνολο από κουτάκια διαφορετικό σε κάθε φάση και περνά μέσα από ένα σύνολο εντολών `if` για να διαπιστωθεί αν έγιναν συγκεκριμένα είδη λαθών. Όταν ο κώδικας μπει σε περισσότερα του ενός `if`, αυτό σημαίνει ότι υπάρχουν διφορούμενα λάθη. Κάθε φορά που μπαίνει μέσα στη συνθήκη ενός `if`, αυξάνεται η τιμή ενός πίνακα στην αντίστοιχη θέση (`error_index[]` για την πρώτη φάση, `second_box_error_index[]` για τη δεύτερη φάση, `third_box_error_index[]` για την τρίτη φάση, `div_error_index[]` για τη διαίρεση ξεχωριστά φάσεων). Όταν έχουν προκύψει διφορούμενα λάθη τότε η μήτρα με τα υποψήφια λάθη αποθηκεύει τους δείκτες από την εκάστοτε `error_index[]`. Αξίζει να επισημανθεί ότι ο δείκτης της εκάστοτε `error_index[]` έχει διαφορετική σε κάθε φάση ερμηνεία, δηλαδή για παράδειγμα μπορεί στη φάση 1 για την πρόσθεση ο δείκτης 0 της `error_index[]` να αφορά λάθος σύγχυσης πρόσθεσης-αφαίρεσης και ο δείκτης 5 να αφορά λάθος κρατουμένου, ενώ σε άλλη φάση ο δείκτης 0 να αφορά λάθος για παράδειγμα σύγχυσης πρόσθεσης-πολλαπλασιασμού και λάθος κρατουμένου ταυτόχρονα και ο δείκτης 5 να είναι η σωστή απάντηση. Στην πρόσθεση, στην αφαίρεση και τον πολλαπλασιασμό χρησιμοποιείται η `error_index[]` για την πρώτη φάση, η `second_box_error_index[]` στη δεύτερη φάση και η `third_box_error_index[]` στην τρίτη φάση. Μετά μέσω μιας εντολής `switch` καταχωρούνται για κάθε υποψήφιο λάθος το μερικό άθροισμα που θα εφαρμοζόταν αν επιλεγόταν το συγκεκριμένο λάθος στη μήτρα `error_frequency[]` με κλήση της `ConfusedOperators(string, bool, int, string)` και μέσω της `CorrectAnswers()` για τη σωστή απάντηση. Ύστερα, μία κρίσιμης σημασίας μέθοδος καλείται, η `ChooseIndex(double, int)`, στην οποία ουσιαστικά επιλέγεται το είδος λάθους με τη μεγαλύτερη συχνότητα, άρα με τη μικρότερη καταχώρηση στον πίνακα `error_frequency[]`, το οποίο μπαίνει σαν πρώτο όρισμα στην `ChooseIndex()` και σα δεύτερο όρισμα μπαίνει η `candidate_index[]`. Επιστρέφεται ένας πίνακας `error_list[]` με τα λάθη τα οποία συνέβησαν παράλληλα αλλά έχουν τη μεγαλύτερη συχνότητα σαν ιστορικό από οποιοδήποτε άλλο λάθος που συνέβηκε παράλληλα σε αυτή τη φάση. Ενδέχεται να υπάρχουν περισσότερα του ενός τέτοια λάθη, τα οποία έχουν την ίδια ακριβώς συχνότητα. Τέλος, από τα πολλά αυτά λάθη (αν υπάρχουν) με την ίδια συχνότητα επιλέγεται τελικά ένα με τυχαία επιλογή. Αν η απάντηση είναι σωστή τότε η μεταβλητή `partial_result` αυξάνεται κατά 1, καλείται η `ReduceConfusedOperatorErrorIndexes(string, string, string, string)`, η μήτρα `current_errors[]` αυξάνεται κατά ένα στην αντίστοιχη θέση που αφορά τη σωστή απάντηση και μία μήτρα για τις υποδείξεις λαθών, η `alert[]` παίρνει την τιμή `false`, για να μη δώσει κάποιο λάθος μήνυμα στη φάση αυτή. Η `alert[]` είναι μία μήτρα η οποία είναι υπεύθυνη για τα μηνύματα λάθους που θα δείχνει το σύστημα μετά από κάθε άσκηση. Σα δείκτες χρησιμοποιεί τις φάσεις. Έτσι, όταν είναι `true` για κάποιους δείκτες σημαίνει ότι στην αντίστοιχη φάση συνέβη λάθος. Αν είναι `false` σημαίνει ότι στη φάση εκείνη ήταν σωστό το αποτέλεσμα. Αν η απάντηση είναι λάθος τότε η μήτρα `current_errors[]` αυξάνεται κατά ένα στον αντίστοιχο δείκτη όπου συνέβη το λάθος από τα διαδοχικά `if`, καλείται η `UpdateWrongAnswer()` η `alert[]` γίνεται `true` και το `alert_string` παίρνει την τιμή θα τυπώσει το αντίστοιχο μήνυμα στην οθόνη μέσω της `MessageBox()`. Η μεταβλητή `partial_result` θα γίνει ίση με `partial_result + 1/(2 +`

`error_index`), όπου `error_index` είναι ο δείκτης λάθους που επιλέχθηκε από τον αλγόριθμο. Σε κάθε είτε σωστή είτε λάθος φάση καλείται η `UpdateAnswer()` που αυξάνει τις συνολικές ερωτήσεις. Μετά και την τρίτη φάση για τις πράξεις της πρόσθεσης, αφαίρεσης και πολλαπλασιασμού διαιρείται η `partial_result` διά του 3 και το αποτέλεσμα προστίθεται στη μεταβλητή `result` που κρατά το σκορ από τις προηγούμενες ασκήσεις. Όταν τελειώνουν οι ασκήσεις του κεφαλαίου η `result` διαιρείται διά του 15 αν οι συνολικές ασκήσεις ήταν 15 (Αρχάριος μαθητής), διά του 10 αν ήταν 10 (Μέτριος μαθητής) και διά του 5 αν ήταν 5 (Καλός μαθητής) και βγαίνει το τελικό αποτέλεσμα, το οποίο μέσω της μεθόδου `EnterRate(double, string, string)` εισάγεται στη βάση. Επίσης, καλείται η `UpdateLevel()`, όταν τελειώσουν οι ασκήσεις για ένα κεφάλαιο, η οποία ανανεώνει το επίπεδο χρήστη. Τέλος, ξανακαλείται η `launch_exercise()` για την επόμενη άσκηση, ή για την προβολή αποτελεσμάτων.

`double Operations_exercises.CorrectAnswers():`

Η μέθοδος αυτή επιστρέφει τον αντίστροφο του αριθμού των σωστών απαντήσεων.

`double Operations_exercises.ConfusedOperators(string error_index, bool IsDoubleMistake, double result, string error):`

Η μέθοδος αυτή έχει διπλή έννοια. Όταν το boolean όρισμά του είναι `true`, τότε χρησιμοποιείται απλά για να επιστρέψει την ποσότητα $1/(2 + error_index)$, όπου `error_index` είναι το πρώτο όρισμα. Χρησιμοποιείται κυρίως στην `ChooseIndex()` για την επιλογή του κατάλληλου λάθους. Αν το boolean όρισμα είναι `false`, τότε ανάλογα το λάθος προσθέτει στη `result` που είναι το 3^ο όρισμα την ποσότητα $1/(2 + error_index)$ αν δεν είναι σοβαρό λάθος και παράλληλα αυξάνει τις τιμές `error_index` και `error` στη βάση δεδομένων. Επιστρέφει την ανανεωμένη (ή μη) τιμή της `result`.

`void Operations_exercises.UpdateAnswer():`

Η μέθοδος αυτή αυξάνει την τιμή των συνολικών ασκήσεων στη βάση κατά ένα.

`void Operations_exercises.UpdateWrongAnswer():`

Η μέθοδος αυτή αυξάνει την τιμή των λαθών στη βάση κατά ένα.

`int[] Operations_exercises.ChooseIndex(double [] array, int [] candidate_index):`

Όπως αναφέραμε ελάχιστα λίγο πριν η μέθοδος αυτή είναι που επιλέγει το λάθος το οποίο θα εφαρμοστεί. Χρησιμοποιεί απλά στατιστική μέθοδο. Παίρνει σαν ορίσματα τη μήτρα `array[]` που περιέχει τις ποσότητες $1/(2 + error_index)$ των υποψήφιων λαθών που βρίσκονται στον πίνακα `candidate_index` ως δεύτερο όρισμα. Συγκεκριμένα, η μέθοδος βρίσκει το υποψήφιο ή τα υποψήφια λάθη τα οποία θα επιστραφούν ως πίνακας τα οποία συμβαίνουν συχνότερα και τα οποία συνέβησαν παράλληλα ως διαφορούμενο λάθος σε κάποια φάση, δηλαδή με τη μικρότερη ποσότητα $1/(2 + error_index)$. Η μέθοδος επιστρέφει ένα πίνακα με τα υποψήφια λάθη που συμβαίνουν συχνότερα, από τα διαφορούμενα λάθη που έλαβαν χώρα.

`void Operations_exercises.UpdateLevel():`

Η μέθοδος αυτή ανανεώνει το επίπεδο του χρήστη, το οποίο έχει ως εξής. Αν το κλάσμα (συνολικές απαντήσεις – συνολικά λάθη) / συνολικές απαντήσεις, είναι μικρότερο του 60% τότε

ο χρήστης γίνεται «Αρχάριος», αν είναι μεγαλύτερο ή ίσο του 60% και μικρότερο του 80% τότε ο χρήστης γίνεται «Μέτριος», ενώ αν είναι μεγαλύτερο ή ίσο του 80% τότε γίνεται «Καλός».

`void Operations_exercises.EnterRate(double rate, string str, string last):`

Η μέθοδος αυτή προσθέτει το τελικό αποτέλεσμα `rate` και στον πίνακα `Results` της βάσης και στον πίνακα `Rates`, σχετικά με το πόσο πέρασε ή όχι ένα κεφάλαιο. Αν το αποτέλεσμα είναι μικρότερο του 60% για ένα κεφάλαιο τότε το αποτέλεσμα στα επόμενα κεφάλαια από αυτό τίθενται σε -1, μόνο στον πίνακα `Rates`. Αυτό γιατί υποτίθεται ότι όταν δεν πηγαίνεις καλά σε ένα κεφάλαιο δε γίνεται να γνωρίζεις καλά και τα επόμενα. Ή το αντίστροφο, αν κάποιος έχει πάει καλά σε ένα κεφάλαιο, τότε θα πρέπει να έχει πάει και στα προηγούμενά του.

`void Operations_exercises.ReduceConfusedOperatorErrorIndexes(string arg1, string arg2, string arg3, string arg4, string arg5):`

Η μέθοδος αυτή μπορεί να πάρει μέχρι 5 ορίσματα, που δηλώνουν 5 διαφορετικά `error_indexes`. Αυτή η μέθοδος καλείται όταν το αποτέλεσμα σε μία άσκηση είναι σωστό και η μέθοδος τότε μειώνει κατά ένα τα `error_indexes` των λαθών που έχουν περάσει σαν ορίσματα στη μέθοδο. Αυτό γίνεται για να δείχτεί ότι όταν ο χρήστης κάνει ένα επόμενο λάθος θα στοιχίσει λιγότερο από το αν το έκανε μετά από κάποια λάθος απάντηση, δείχνει δηλαδή «την τάση» που έχει ο χρήστης να δίνει σωστές ή λάθος απαντήσεις.

`string Operations_exercises.ExportResults(double rate, int [] errors):`

Η μέθοδος αυτή εκτυπώνει τα αποτελέσματα στην οθόνη του χρήστη στο τέλος κάθε κεφαλαίου. Παίρνει σαν όρισμα τη μεταβλητή `rate` που περιέχει το συνολικό σκορ του χρήστη που πέτυχε σε ένα κεφάλαιο, τη στατική μήτρα λαθών, για να εκτυπώσει ανάλογα μηνύματα για τον αριθμό των λαθών που έκανε στην τελευταία εκτέλεση ασκήσεων του κεφαλαίου και επιστρέφει ένα `string`, το οποίο αποτελεί το συνολικό μήνυμα στο χρήστη.

`void Operations_exercises.MessageBox(string msg):`

Η μέθοδος αυτή εκτυπώνει μηνύματα κατά τη διάρκεια των ασκήσεων έτσι ώστε να δείξει στο χρήστη τι είδους λάθη έκανε και πως θα έπρεπε να είχε καθοδηγηθεί και να απαντήσει σωστά. Παίρνει σαν όρισμα το μήνυμα που θα εκτυπωθεί στην οθόνη.

Η κλάση `CurrentCourses`

Η κλάση αυτή περιέχει τις εξής μεθόδους: `void LoadCurrentCourse(string str)` και `string FetchCurrentCourse()`.

`void CurrentCourses.LoadCurrentCourse(string str):`

Η μέθοδος αυτή ανανεώνει στη βάση το πεδίο `currentCourse` στον πίνακα του χρήστη, ανάλογα με το σύνδεσμο τον οποίο πάτησε ο χρήστης από τη δεξιά λίστα κεφαλαίων τελευταία φορά για να επιλύσει ασκήσεις.

`string CurrentCourses.FetchCurrentCourse():`

Η μέθοδος αυτή κάνει ανάκτηση του τωρινού κεφαλαίου του χρήστη, το οποίο επιλύει.

Η κλάση *Courses*

Η κλάση αυτή περιέχει 2 σημαντικές μεθόδους: `double SelectRate(string str)` και `bool isExecuted(string str)`.

`double Courses.SelectRate(string str)`:

Η μέθοδος αυτή επιλέγει από τη βάση δεδομένων το σκορ του χρήστη για το κεφάλαιο που περιέχεται ως όρισμα στη μέθοδο (`str`). Το τσεκάρισμα γίνεται στον πίνακα `Results` της βάσης.

`bool Courses.isExecuted(string str)`:

Η μέθοδος αυτή τσεκάρει αν ένα κεφάλαιο έχει εκτελεστεί, συγκεκριμένα αν έχει την τιμή `-1` (δεν έχει εκτελεστεί) ή `0` (έχει εκτελεστεί). Το τσεκάρισμα γίνεται στον πίνακα `Results` της βάσης.

Η κλάση *Forum*

Η κλάση `Forum` περιέχει κάποιες σημαντικές μεθόδους: `BindSubjects()`, `BindComments()`, `DataList1_ItemCommand()`, `DataList2_ItemCommand()`.

`void Forum.BindSubjects()`:

Η μέθοδος αυτή παρουσιάζει στο χρήστη τα θέματα του φόρουμ.

`void Forum.BindComments()`:

Η μέθοδος αυτή παρουσιάζει στο χρήστη τα σχόλια για κάποιο θέμα.

`void Forum.DataList1_ItemCommand()`:

Η μέθοδος αυτή έχει ποικίλες λειτουργίες. Έτσι σύμφωνα με το αντικείμενο `e.CommandName`, (όπου το `e` είναι αντικείμενο τύπου `DataListCommandEventArgs`) στη μέθοδο αυτή μπορεί να γίνει, προβολή σχολίων, επιλογή θέματος, προσθήκη σχολίου στο θέμα αυτό, ακύρωση, ανανέωση, διαγραφή και απόκρυψη σχολίων.

`void Forum.DataList2_ItemCommand()`:

Όπως και στη `DataList1` έτσι και εδώ μπορεί να γίνει ανάλογα την τιμή του `e.CommandName` επιλογή, ανανέωση, διαγραφή και ακύρωση.

Η κλάση *Marks*

Η κλάση αυτή περιέχει 2 σημαντικές μεθόδους, τις: `BindLastGrid()` `BindErrorTable()`.

`void Marks.BindLastGrid()`:

Η μέθοδος αυτή παρουσιάζει στο χρήστη τη βαθμολογία του σε όλα τα μαθήματα, την τελευταία φορά που τα προσπέρασε. Η μέθοδος ανακτά τα δεδομένα αυτά από τον πίνακα `Rates` της βάσης.

`void Marks.BindErrorTable()`:

Η μέθοδος αυτή παρουσιάζει στο χρήστη τη είδη λαθών του. Η μέθοδος ανακτά τα δεδομένα αυτά από τον πίνακα `Results` της βάσης.

Η κλάση *Profile*

Η κλάση αυτή περιέχει μία σημαντική μέθοδο, την `BindDetails()`.

`void Profile.BindDetails()`:

Η μέθοδος αυτή παρουσιάζει στο χρήστη τις πληροφορίες του προφίλ του.

Η κλάση ViewExercise

Η κλάση αυτή περιλαμβάνει μία σημαντική μέθοδο, τη View().

void ViewExercise.View():

Η μέθοδος αυτή εμφανίζει στον καθηγητή τις εύκολες ασκήσεις από όλα τα κεφάλαια. Καλείται σχεδόν από όλα τα γεγονότα (events) του GridView1, τις RowEditing, RowUpdating, RowDeleting, RowCanceling, PageIndexChanged.

Η κλάση ViewMarks

Η κλάση αυτή περιλαμβάνει μερικές σημαντικές μεθόδους, τις: ShowRole() , BindGrid(), BindDetails().

string ViewMarks.ShowRole():

Η μέθοδος αυτή επιστρέφει από τη βάση δεδομένων, το ρόλο του τωρινά συνδεδεμένου χρήστη.

void ViewMarks.BindGrid():

Η μέθοδος αυτή παρουσιάζει στον καθηγητή τα σκορ που πέτυχαν τελευταία φορά, όλοι οι μαθητές.

void ViewMarks.BindDetails():

Η μέθοδος αυτή παρουσιάζει στον καθηγητή τα είδη λαθών από κάποιον από τους μαθητές που επιλέχτηκε.