



Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης
Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής &
Πολυμέσων



Πτυχιακή Εργασία

**Υλοποίηση του παιχνιδιού ‘Ποιος θέλει να γίνει εκατομμυριούχος’
ως διαδικτυακή εφαρμογή**

Λαμπράκης Κωνσταντίνος (ΑΜ:426)

Επιβλέπων Καθηγητής: Δρ. Παπαδάκης Νίκος

Επιτροπή Αξιολόγησης: Δρ. Μανιφάβας Χαράλαμπος
Δρ. Παναγιωτάκης Σπυρίδων
Δρ. Παπαδάκης Νικόλαος

Ημερομηνία Παρουσίασης: Τρίτη 1 Απριλίου 2014

Ευχαριστίες

Θα ήθελα να ευχαριστήσω αρχικά τον επιβλέπων καθηγητή μου, Δρ. Νίκο Παπαδάκη, ο οποίος με βοήθησε στην επιλογή του θέματος της πτυχιακής εργασίας που μου ταίριαζε περισσότερο και που μέσω αυτής μου δόθηκε η ευκαιρία να ασχοληθώ με ένα τόσο ενδιαφέρον και σημαντικό ζήτημα, όπως αυτό της κατασκευής των διαδικτυακών εκπαιδευτικών εφαρμογών. Θα ήθελα επίσης να τον ευχαριστήσω, γιατί καθ' όλη την διάρκεια της εκπόνησης της πτυχιακής εργασίας μου προσέφερε βοήθεια με τις γνώσεις του και με καθοδήγησε σωστά για την επιτυχή αποπεράτωσή της .

Τέλος, θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες πρωτίστως στην οικογένειά μου αλλά και στους φίλους μου, που μου έδειξαν αμέριστη συμπαράσταση.

Abstract

Technological developments and new training paradigms have a strong influence on society. Information technology (IT) is evolving toward an integrated communication and information technology (CIT). Training is evolving from a separately planned external activity toward an integrated learning-working activity. Education and the teaching profession have to take such developments into account. As a consequence the teaching profession is evolving from an emphasis on delivering information to an emphasis on creating learning environments. The main purpose of this essay is to create a web educational application which can be even used for learners' assessment.

Σύνοψη

Από την πληθώρα των μέσων που προσφέρει σήμερα η εκπαιδευτική τεχνολογία, σημαντική θέση κατέχει ο ηλεκτρονικός υπολογιστής, του οποίου η εισαγωγή και χρήση στην εκπαίδευση όσον αφορά τη διδακτική υποστήριξη διαφόρων αντικειμένων, βρίσκεται ακόμη σε πρώιμα στάδια. Επομένως ο χώρος προσφέρεται για δημιουργική ενασχόληση και ειδικότερα για την διερεύνηση των δυνατοτήτων συμβολής της πληροφορικής στην εποπτικοποίηση της διδακτικής πράξης, γεγονός που αποτελεί συστηματική επίδιωξη της σύγχρονης διδακτικής. Βασικός στόχος της συγκεκριμένης πτυχιακής εργασίας είναι ο σχεδιασμός και η υλοποίηση μιας διαδικτυακής εκπαιδευτικής εφαρμογής, η οποία εκτός μέσο εκπαίδευσης μπορεί να χρησιμοποιηθεί και ως μέσο αξιολόγησης.

Πίνακας Περιεχομένων

1	Εισαγωγή	1
1.1	Περίληψη	1
1.2	Κίνητρο για τη Διεξαγωγή της Εργασίας	1
1.3	Σκοπός και Στόχοι Εργασίας	1
1.4	Δομή Εργασίας	2
2	Οι Θεωρίες Μάθησης και η Σημασία της χρήσης των Νέων Τεχνολογιών στην Εκπαίδευση	3
2.1	Η διαδικασία της μάθησης	3
2.2	Ο εκπαιδευτικός και η αξιοποίηση των σύγχρονων θεωριών μάθησης	3
2.3	Η εκπαιδευτική Αξιολόγηση	4
2.4	Διδακτική και Νέες Τεχνολογίες	5
3	Μεθοδολογία Υλοποίησης	8
3.1	Μέθοδος Ανάλυσης και Ανάπτυξης της Πτυχιακής	8
3.1.1	Συστήματα Διαχείρισης Βάσεων Δεδομένων	8
3.1.2	Γλώσσες Προγραμματισμού	9
4	Σχέδιο Δράσης για την εκπόνηση της εργασίας	11
4.1	State of the Art	11
4.1.1	Το μοντέλο Client Server	12
4.2	Περιπτώσεις Χρήσης	13
4.2.1	Περίπτωση Χρήσης 1: Είσοδος Χρήστη	14
4.2.2	Περίπτωση Χρήσης 2: Εγγραφή Χρήστη	14
4.2.3	Περίπτωση Χρήσης 3: Απεικόνιση Ιστορικού Χρήστη	15
4.2.4	Περίπτωση Χρήσης 4: Απεικόνιση των Καλύτερων Αποτελεσμάτων	15
4.2.5	Περίπτωση Χρήσης 5: Απεικόνιση Πρόσφατων Αποτελεσμάτων	15
4.2.6	Περίπτωση Χρήσης 6: Αλλαγή Κωδικού Ασφαλείας	16
4.2.7	Περίπτωση Χρήσης 7: Διαγραφή του Λογαριασμού του Χρήστη	17
4.2.8	Περίπτωση Χρήσης 8: Έξοδος από το Σύστημα	17
4.2.9	Περίπτωση Χρήσης 9: Εισαγωγή Ερώτησης	18
4.2.10	Περίπτωση Χρήσης 10: Τροποποίηση Ερώτησης	19
4.2.11	Περίπτωση Χρήσης 11: Διαγραφή Ερώτησης	21
4.2.12	Περίπτωση Χρήσης 12: Προσθήκη Διαχειριστή	23
4.2.13	Περίπτωση Χρήσης 13: Αφαίρεση Διαχειριστή	24
4.2.14	Περίπτωση Χρήσης 14: Διαγραφή Χρήστη	25
4.2.15	Περίπτωση Χρήσης 15: Εκκίνηση του Παιχνιδιού	25
4.3	Το Διάγραμμα Κλάσεων του Συστήματος	27
4.3.1	Η κλάση ServerAccess	27
4.3.2	Η κλάση UserManagement	28
4.3.3	Η κλάση QuestionManagement	30
4.3.4	Η κλάση HistoryManagement	31
4.3.5	Η κλάση UserBean	32
4.3.6	Η κλάση GameBean	33
4.3.7	Η κλάση GameLogic	34
4.4	Ο Σχεδιασμός της Βάσης Δεδομένων	35
4.4.1	Ο πίνακας των Χρηστών	35
4.4.2	Ο πίνακας του Ιστορικού	36
4.4.3	Ο πίνακας των Ερωτήσεων	37
4.5	Σημαντικοί Στόχοι για την Ολοκλήρωση της Πτυχιακής	37
4.6	Χρονοδιάγραμμα Gantt	38
5	Κύριο Μέρος Πτυχιακής	39
5.1	Ανάλυση Προβλήματος	39
5.1.1	Απαιτήσεις Συστήματος	39

5.2	Σχεδιασμός Υλοποίησης	39
5.2.1	Η φόρμα εισαγωγής του χρήστη	39
5.2.2	Η φόρμα εγγραφής χρήστη στο σύστημα	41
5.2.3	Η κεντρική οθόνη του χρήστη	44
5.2.4	Έξοδος χρήστη από το σύστημα	45
5.2.5	Φόρμα αλλαγής κωδικού ασφαλείας	45
5.2.6	Φόρμα διαγραφής χρήστη από το σύστημα	47
5.2.7	Απεικόνιση των βαθμολογιών του χρήστη	48
5.2.8	Απεικόνιση των πιο πρόσφατων βαθμολογιών	49
5.2.9	Απεικόνιση των καλύτερων βαθμολογιών	50
5.2.10	Η κεντρική οθόνη του διαχειριστή	51
5.2.11	Η φόρμα προσθήκης ερώτησης	52
5.2.12	Η φόρμα επεξεργασίας ερώτησης	53
5.2.13	Η φόρμα διαγραφής ερώτησης	54
5.2.14	Η φόρμα προσθήκης διαχειριστή	55
5.2.15	Η φόρμα αφαίρεσης δικαιωμάτων διαχειριστή	56
5.2.16	Η φόρμα για την διαγραφή χρήστη από τον διαχειριστή	57
5.2.17	Η σελίδα του παιχνιδιού	58
5.2.18	Ο χρήστης κερδίζει το παιχνίδι	59
5.2.19	Ο χρήστης χάνει στο παιχνίδι	60
5.2.20	Ο χρήστης επιλέγει την έξοδό του από το παιχνίδι	61
6	Αποτελέσματα	63

Πίνακας Πινάκων

Πίνακας	Τίτλος	Σελ.
1	Περίπτωση Χρήσης 1: Είσοδος Χρήστη	14
2	Περίπτωση Χρήσης 2: Εγγραφή Χρήστη	14-15
3	Περίπτωση Χρήσης 3: Απεικόνιση Ιστορικού Χρήστη	15
4	Περίπτωση Χρήσης 4: Απεικόνιση των Καλύτερων Αποτελεσμάτων	15
5	Περίπτωση Χρήσης 5: Απεικόνιση Πρόσφατων Αποτελεσμάτων	16
6	Περίπτωση Χρήσης 6: Αλλαγή Κωδικού Ασφαλείας	16-17
7	Περίπτωση Χρήσης 7: Διαγραφή του Λογαριασμού του Χρήστη	17
8	Περίπτωση Χρήσης 8: Έξοδος από το Σύστημα	18
9	Περίπτωση Χρήσης 9: Εισαγωγή Ερώτησης	18-19
10	Περίπτωση Χρήσης 10: Τροποποίηση Ερώτησης	19-21
11	Περίπτωση Χρήσης 11: Διαγραφή Ερώτησης	21-23
12	Περίπτωση Χρήσης 12: Προσθήκη Διαχειριστή	23-24
13	Περίπτωση Χρήσης 13: Αφαίρεση Διαχειριστή	24-25
14	Περίπτωση Χρήσης 14: Διαγραφή Χρήστη	25
15	Περίπτωση Χρήσης 15: Εκκίνηση του Παιχνιδιού	26-27
16	Διάγραμμα της κλάσης ServerAccess	28
17	Διάγραμμα της κλάσης UserManagement	28-29
18	Διάγραμμα της κλάσης QuestionManagement	30
19	Διάγραμμα της κλάσης HistoryManagement	31
20	Διάγραμμα της κλάσης UserBean	32
21	Διάγραμμα της κλάσης GameBean	33
22	Διάγραμμα της κλάσης GameLogic	34
23	Ο πίνακας των Χρηστών	35-36
24	Ο πίνακας του Ιστορικού	36
25	Ο πίνακας των Ερωτήσεων	37
26	Στόχοι για την Ολοκλήρωση της Πτυχιακής	37-38
27	Χρονοδιάγραμμα Gantt	38

Πίνακας Εικόνων

Εικόνα	Τίτλος	Σελ.
1	Η φόρμα εισαγωγής του χρήστη	40
2	Εισαγωγή λάθος ονόματος χρήστη ή λάθος κωδικού	41
3	Η φόρμα εγγραφής χρήστη	42
4	Εισαγωγή απαγορευμένου ονόματος χρήστη ή κωδικού	43
5	Εισαγωγή κενού ονόματος χρήστη και κωδικού	44
6	Η κεντρική οθόνη του χρήστη	45
7	Φόρμα αλλαγής κωδικού ασφαλείας	46
8	Ασυμφωνία κωδικών	47
9	Φόρμα διαγραφής χρήστη από το σύστημα	48
10	Απεικόνιση των βαθμολογιών του χρήστη	49
11	Απεικόνιση των πιο πρόσφατων βαθμολογιών	50
12	Απεικόνιση των καλύτερων βαθμολογιών	51
13	Η κεντρική οθόνη του διαχειριστή	52
14	φόρμα προσθήκης ερώτησης	53
15	Η φόρμα επεξεργασίας ερώτησης	54
16	Η φόρμα διαγραφής ερώτησης	55
17	Η φόρμα προσθήκης διαχειριστή	56
18	Η φόρμα αφαίρεσης δικαιωμάτων διαχειριστή	57
19	Η φόρμα για την διαγραφή χρήστη από τον διαχειριστή	58
20	Η σελίδα του παιχνιδιού	59
21	Ο χρήστης κερδίζει το παιχνίδι	60
22	Ο χρήστης χάνει στο παιχνίδι	61
23	Ο χρήστης επιλέγει την έξοδο του από το παιχνίδι	62

1.Εισαγωγή

Οι διαδικτυακές εφαρμογές όπως για παράδειγμα η παρούσα που αφορά την υλοποίηση του γνωστού παιχνιδιού «Ποιος θέλει να γίνει Εκατομμυριούχος», μπορεί να αποδειχτεί πολύτιμο εκπαιδευτικό εργαλείο κατά τη διάρκεια της εκπαιδευτικής διαδικασίας. Τη συγκεκριμένη διαδικτυακή εφαρμογή μπορεί να χρησιμοποιήσουν: οι εκπαιδευτικοί στο σχολείο, οι εκπαιδευόμενοι στο οικία τους αλλά και οποιοσδήποτε άτομο επιθυμεί να εμπλουτίσει τις γνώσεις τους. Με τον τρόπο αυτό η γνώση μεταδίδεται εύκολα και διασκεδαστικά. Η παρούσα πτυχιακή υλοποιήθηκε για να παρέχει καθοδήγηση στην υλοποίηση μιας τέτοιας διαδικτυακής εφαρμογής η οποία θα χρησιμοποιηθεί για εκπαιδευτικούς σκοπούς.

1.1 Περίληψη

Σκοπός της παρούσας μελέτης ήταν η μελέτη, η σχεδίαση και η ανάπτυξη μιας διαδικτυακής εφαρμογής βασισμένης στο γνωστό παιχνίδι «Ποιος θέλει να γίνει Εκατομμυριούχος». Μέσα από τη συγκεκριμένη μελέτη καθορίστηκαν οι ανάγκες σχεδίασης νέων διαδικτυακών εφαρμογών εκπαιδευτικού χαρακτήρα, οι οποίες έδωσαν τη δυνατότητα στη διαδικτυακή εφαρμογή να υποστηρίξει την παρουσίαση και τη διάχυση πληροφοριών που αφορούν την εκπαιδευτική διαδικασία.

Πιο συγκεκριμένα για τις ανάγκες υλοποίησης της εφαρμογής χρησιμοποιήθηκε ένας server και μια βάση δεδομένων ώστε να καταγράφονται οι χρήστες και οι ερωτήσεις του παιχνιδιού. Από κατασκευαστικής άποψης χρησιμοποιήθηκε server-side scripting. Καθώς η εφαρμογή είναι απομακρυσμένη και δεν εγκαθίσταται από το χρήστη απαιτείται υψηλή ταχύτητα ανταπόκρισης. Είναι μια εφαρμογή server-side, η οποία δέχεται τα δεδομένα του χρήστη, τα επεξεργάζεται και του επιστρέφει τα τελικά αποτελέσματα. Αναλυτικά: η εφαρμογή δέχεται την απάντηση που επέλεξε ο χρήστης, ελέγχει εάν είναι η σωστή απάντηση για την συγκεκριμένη ερώτηση και επιστρέφει το τελικό αποτέλεσμα. Οι γλώσσες που επιλέχθηκαν ήταν οι: JSP, Java και MySQL.

Η διαδικτυακή εφαρμογή που αναπτύχθηκε, με κεντρικό θέμα την εκπαίδευση, δίνει τη δυνατότητα στο χρήστη να εγγραφεί σε αυτή και να μεταβεί στο παιχνίδι. Κατά τη διάρκεια του παιχνιδιού ο χρήστης καλείται να απαντήσει σε 15 ερωτήσεις με κλιμακούμενο επίπεδο δυσκολίας. Κάθε ερώτηση, αντιστοιχεί σε ένα συγκεκριμένο χρηματικό ποσό. Ο χρήστης καλείται να επιλέξει ανάμεσα από 4 διαφορετικές πιθανές απαντήσεις την σωστή. Εάν απαντήσει σωστά, ο παίκτης κερδίζει το αντίστοιχο ποσό χρημάτων και προχωράει στην επόμενη ερώτηση. Με κάθε σωστή απάντηση ο παίκτης μπορεί να κερδίσει και μεγαλύτερο ποσό ή να επιλέξει να αποχωρίσει και να κρατήσει το ποσό που έχει κερδίσει ως τότε. Όσο το ποσό των χρημάτων αυξάνει, τόσο ο βαθμός δυσκολίας των ερωτήσεων γίνεται μεγαλύτερος. Επίσης υπάρχουν 2 ερωτήσεις ,η 5^η και η 10^η, που όταν απαντηθούν σωστά εξασφαλίζουν στον παίκτη το κέρδος ενός ελάχιστου ποσού. Το ποσό των χρημάτων έχει το ρόλο της ανταμοιβής για τη σωστή ανταπόκριση στις ερωτήσεις.

1.2 Κίνητρο για την Διεξαγωγή της Εργασίας

Είναι συχνά δύσκολο ως και ακατόρθωτο να κατασκευασθεί μια διαδικτυακή εφαρμογή για εκπαιδευτικούς λόγους με απλή ανάγνωση. Με το ίδιο σκεπτικό είναι δύσκολο να θα εξηγηθούν όλοι οι τρόποι κατασκευής προτύπων, καθώς και η μέθοδος σχεδιασμού μιας πλήρους multimedia εφαρμογής. Η παρούσα πτυχιακή στοχεύει στην επίλυση των παραπάνω προβλημάτων με την κατασκευή μιας αντίστοιχης διαδικτυακής εφαρμογής και καταγραφή των βημάτων που ακολουθήθηκαν για να κατασκευαστεί.

1.3 Σκοπός και Στόχοι Εργασίας

Βασικός σκοπός της παρούσας εργασίας είναι η δημιουργία μιας διαδικτυακής εφαρμογής, εκπαιδευτικού χαρακτήρα, η οποία μπορεί να χρησιμοποιηθεί ακόμα και για αξιολόγηση των εκπαιδευομένων.

Στόχοι εργασίας:

- Να παρουσιαστεί και να αναλυθεί το στάδιο της σύλληψης της δημιουργίας της διαδικτυακής εκπαιδευτικής εφαρμογής «Ποιος θέλει να γίνει Εκατομμυριούχος»

- Να παρουσιαστεί και να αναλυθεί το στάδιο της επεξεργασίας ώστε να μετριαστούν τα προβλήματα που προκύπτουν από την ανάλυση του έργου.
- Να υλοποιηθεί η διαδικτυακή εκπαιδευτική εφαρμογή.

1.4 Δομή Εργασίας

Το πρώτο κεφάλαιο της εργασίας αποτελεί και την εισαγωγή της συγκεκριμένης πτυχιακής εργασίας η οποία περιλαμβάνει: την περίληψη, την παρουσίαση του σκοπού και των στόχων της πτυχιακής καθώς και τη διάρθρωση των κεφαλαίων.

Στο δεύτερο κεφάλαιο παρουσιάζονται οι Θεωρίες Μάθησης καθώς και η σημασία που έχει η χρησιμοποίηση των Νέων Τεχνολογιών στην εκπαίδευση και στην αξιολόγηση.

Στο τρίτο κεφάλαιο παρουσιάζεται η Μεθοδολογία Υλοποίησης στην οποία περιλαμβάνονται: τα Συστήματα Διαχείρισης Βάσεων Δεδομένων και οι Γλώσσες Προγραμματισμού.

Στο τέταρτο κεφάλαιο παρουσιάζεται και αναλύεται το Σχέδιο Δράσης για την εκπόνηση της εργασίας στο οποίο περιλαμβάνεται: το State of the Art, τους σημαντικούς στόχους για την ολοκλήρωση της πτυχιακής και το χρονοδιάγραμμα Gantt.

Στο πέμπτο κεφάλαιο αναλύεται το πρόβλημα και οι απαιτήσεις του συστήματος ενώ παρέχονται όλες οι λεπτομέρειες και τα στοιχεία του Σχεδιασμού ενώ τέλος περιλαμβάνει το πλήρες εγχειρίδιο χρήσης της διαδικτυακής εκπαιδευτικής εφαρμογής.

Στο έκτο κεφάλαιο παρουσιάζονται τα αποτελέσματα, τα οφέλη καθώς και τους σκοπούς και στόχους που θα εξυπηρετήσει.

2. Οι Θεωρίες Μάθησης και η Σημασία της χρήσης των Νέων Τεχνολογιών στην Εκπαίδευση

2.1 Η διαδικασία της μάθησης

Σύμφωνα με τη θεωρητική παραδοχή, η ατομική πορεία της μάθησης και της ανάπτυξης των παιδιών συντελείται μέσω της αλληλεπίδρασης με τους άλλους μαθητές. Κάτω από ένα τέτοιο πλαίσιο οι εκπαιδευτικοί ωθούν τους μαθητές σε δραστηριότητες όπως: η συνεργασία, η συζήτηση και η διαλεκτική αντιπαράθεση. Η μη ενεργή συμμετοχή των μαθητών στο μάθημα έχει ως αποτέλεσμα την εμφάνιση διαμαρτυριών για τον τρόπο που τους αντιμετωπίζουν μέσα στο μάθημα.

Ο διαλεκτικός τρόπος διδασκαλίας κατείχε σημαντική θέση στο παλιό σχολείο, όμως σύντομα αμφισβητήθηκε από τη σύγχρονη διδακτική. Μέσω της διάλεξης, ο εκπαιδευτικός προσφέρει, περιγράφει και μεταδίδει γνώσεις μόνο με τον προφορικό λόγο χωρίς την ενεργό συμμετοχή των μαθητών. Σύμφωνα με τις σύγχρονες προσεγγίσεις μάθησης, ο ρόλος του μαθητή είναι πιο ενεργός.

Σύμφωνα με τον *Piaget* η ψυχοπνευματική ανάπτυξη του παιδιού είναι εξελικτική διαδικασία, η οποία διαμορφώνεται μέσα από διαφορετικά στάδια. Σε κάθε στάδιο το παιδί έχει συγκεκριμένες διανοητικές δυνατότητες. Η μάθηση είναι μια διαδικασία ενεργή. Το να κάνει λάθη ο μαθητής και να προσπαθεί να επιλύσει προβλήματα συντελούν στη μαθησιακή διαδικασία. Η γνώση των μαθητών κατασκευάζεται καθώς εκείνοι αλληλεπιδρούν με λογικούς τρόπους με τον κόσμο γύρω τους.

Ο γνωστικός ψυχολόγος της μάθησης *Bruner* θεωρούσε ότι η μάθηση μπορεί να προέλθει μέσω της κατανόησης των δομών ενός γνωστικού αντικειμένου αλλά και της καθοδηγούμενης ανακάλυψης η οποία οδηγεί στην ανάπτυξη εσωτερικών κινήτρων μάθησης από το μαθητή. Το σχολικό περιβάλλον παρέχει πολιτισμικά πρότυπα αλλά σε κάθε περίπτωση θα πρέπει να λαμβάνεται υπόψη και το διαφορετικό πολιτιστικό και κοινωνικό περιβάλλον που προέρχεται κάθε μαθητής. Υποστηρίζει επίσης ότι η γνώση είναι αποτέλεσμα κυρίως της διαδικασίας και όχι μόνο το αποτέλεσμα της ενώ δίνει ιδιαίτερη έμφαση στην έννοια της ανακάλυψης. Το φυσικό περιβάλλον θα πρέπει να εξασφαλίζει στους μαθητές την απαραίτητη αυτονομία ώστε να είναι σε θέση να μουν σε μια διαδικασία ανακάλυψης.

Η *Θεωρία του Κονστρουκτιβισμού* σήμερα αποτελεί μια από τις πιο δημοφιλείς θεωρίες μάθησης. Ο κάθε μαθητής κατασκευάζει τη γνώση του, γεγονός που συνεπάγεται πως η μάθηση είναι προσωπική υπόθεση. Οι μαθητές οικοδομούν τις γνώσεις τους πάνω στις προηγούμενες εμπειρίες τους και στις προϋπάρχουσες γνωστικές δομές και σχήματα. Η μαθησιακή διαδικασία πραγματοποιείται μέσα σε συγκεκριμένο κοινωνικό πλαίσιο όπου η συνεργασία του εκπαιδευτικού και των μαθητών είναι βασικό στοιχείο της μάθησης. Ο εκπαιδευτικός δεν μεταφέρει απευθείας τις πληροφορίες και τις γνώσεις στα παιδιά αλλά τα διευκολύνει διαμορφώνοντας ένα περιβάλλον πλούσιο σε πνευματικές προκλήσεις και ερεθίσματα. Για τον Κονστρουκτιβισμό σημασία έχει η διαδικασία της μάθησης και όχι μόνο το τελικό αποτέλεσμα. Η μάθηση γίνεται βιωματική, δηλαδή η διαδικασία μέσα από την οποία μαθαίνουν και εκπαιδεύονται οι μαθητές γίνεται μέσω της δράσης, της ανακάλυψης και του προβληματισμού.

Σύμφωνα με τη *συνεργατική μάθηση* δίνεται ιδιαίτερη έμφαση στις συνεργατικές και στις συμμετοχικές διαδικασίες μάθησης. Η τάξη οργανώνεται σε μικρές ομάδες μέσω των οποίων επιτυγχάνεται η δημιουργική συνεργασία των παιδιών. Τα μέλη κάθε ομάδας επιδιώκουν το δυνατότερο αποτέλεσμα για την ομάδα αλλά και για τους ίδιους, ενώ με τον τρόπο αυτό επιτυγχάνουν να ξεπεράσουν τα ατομικά όρια των πράξεων και των σκέψεών τους. Το εκπαιδευτικό περιβάλλον θα πρέπει να είναι διαμορφωμένο με τέτοιο τρόπο ώστε να επιτρέπει και να δίνει τη δυνατότητα για τη δημιουργία ομάδων συνεργασίας μεταξύ των παιδιών.

2.2 Ο εκπαιδευτικός και η αξιοποίηση των σύγχρονων θεωριών μάθησης

Αξιοποιώντας ο εκπαιδευτικός τις σύγχρονες θεωρίες μάθησης θα μπορούσε να εστιάζει πάνω σε κάθε μαθητή, λόγω του ότι κάθε μαθητής παρουσιάζει διαφορετική αναπτυξιακή πορεία σε σχέση με τους άλλους μαθητές. Ο εκπαιδευτικός είναι εκείνος ο οποίος είναι υπεύθυνος για το τι θα διδαχτεί αλλά και το πώς θα διδαχτεί, καθώς επίσης και για την ποικιλία των δραστηριοτήτων την οποία προσφέρει στο μαθητή. Η γνώση δε θα πρέπει να μεταφέρεται στον μαθητή μέσω του μονολόγου αλλά θα πρέπει και ο ίδιος ο μαθητής να την ανακαλύπτει μέσα από τις δραστηριότητες που του θέτει ο εκπαιδευτικός. Η πραγματική γνώση μπορεί να επιτευχθεί μέσω της ανακάλυψης από τον ίδιο το

μαθητή αλλά και της ενεργητικής συνεργασίας με τον ίδιο τον εκπαιδευτικό. Ένα μάθημα πλούσιο σε ανάθεση εργασιών τύπου project θα ενίσχυε την ανακάλυψη της γνώσης από τη μεριά του μαθητή αλλά και η συμμετοχή του εκπαιδευτικού θα ήταν σημαντική στην καθοδήγηση της συγκεκριμένης δραστηριότητα.

Σύμφωνα με τον J. Bruner υποστηρικτή της ανακαλυπτικής μάθησης, κάθε άτομο θα πρέπει να οδηγείται στην ανακάλυψη νέων γνώσεων και εννοιών ώστε στη συνέχεια να τις μετασχηματίζει, να τις εκτιμά και να τις ελέγχει. Ο ρόλος του εκπαιδευτικού εστιάζεται στο να βοηθήσει τους μαθητές να μάθουν να σκέφτονται, να μετέχουν δηλαδή στην διαδικασία της γνώσης και όχι να μεταφέρει μόνο γνώσεις. Η γνώση αναπαρίσταται με τρεις τρόπους: μέσω της πραξιακής αναπαράστασης, της εικονικής αναπαράστασης και της συμβολικής αναπαράστασης. Σύμφωνα με τον Bruner και τα τρία στάδια χρησιμοποιούνται συνεχώς από το άτομο και όχι διαδοχικά όπως υποστηρίζει ο Piaget. Για το λόγο αυτό οι δραστηριότητες που βάζει ο εκπαιδευτικός στους μαθητές του θα πρέπει να ικανοποιούν τα τρία αυτά στάδια, για παράδειγμα δραστηριότητες όπως η επίλυση προβλημάτων και εργασιών. Ιδιαίτερη σημασία θα πρέπει να δίνεται στην εποπτικότητα κάθε μαθήματος καθώς επίσης στη γενική δομή και τις βασικές δομές και αρχές κάθε μαθήματος. Ο εκπαιδευτικός θα πρέπει να βοηθάει στη σταδιακή ανακάλυψη της γνώσης, ενώ κρίνεται ότι η ενίσχυση των μαθητών του είναι περιττή.

2.3 Η εκπαιδευτική αξιολόγηση

Ένα από τα πιο σημαντικά στάδια της διδασκαλίας είναι η εκπαιδευτική αξιολόγηση. Χωρίς τα αποτελέσματα της εκπαιδευτικής αξιολόγησης δεν μπορεί να σχεδιαστεί σωστά η επόμενη διδασκαλία. Ουσιαστικά η εκπαιδευτική αξιολόγηση αναφέρεται στη συστηματική συλλογή πληροφοριών και στην οργάνωσή τους η οποία οδηγεί στο σχεδιασμό διδασκαλίας η οποία θα περιλαμβάνει: επιλογή περιεχομένου διδασκαλίας, μεθόδου διδασκαλίας και κατάλληλου εκπαιδευτικού υλικού, και όχι στην περιστασιακή και τυχαία συλλογή πληροφοριών η οποία δεν μπορεί να τεκμηριώσει στη συνέχεια σχεδιασμό κατάλληλης διδασκαλίας

Κατά τη διαδικασία της εκπαιδευτικής αξιολόγησης κάθε εκπαιδευτικός λειτουργεί ως ερευνητής. Η διαδικασία αποτελείται από τέσσερα στάδια: από την παρατήρηση της μαθησιακής συμπεριφοράς, τη διατύπωση των υποθέσεων για τις μαθησιακές δυνατότητες αλλά και αδυναμίες του μαθητή, την αξιολόγηση του περιεχομένου της διδασκαλίας, της οργάνωσης, των υλικών, των πρακτικών που ακολουθούνται, των στρατηγικών αλλά και των κινήτρων των μαθητών και τέλος από την επιβεβαίωση ή τη μη επιβεβαίωση των αρχικών υποθέσεων.

Οι τρόποι αξιολόγησης χωρίζονται σε δυο μεγάλες κατηγορίες: τους παραδοσιακούς τρόπους αξιολόγησης και τους εναλλακτικούς τρόπους αξιολόγησης. Ανάμεσα στους παραδοσιακούς τρόπους αξιολόγησης περιλαμβάνονται: η *προφορική εξέταση*, που αφορά κυρίως το μάθημα της ημέρας και δίνει τη δυνατότητα στον εκπαιδευτικό να αξιολογήσει πέρα από τις γνωστικές ικανότητες του μαθητή και στοιχεία της προσωπικότητάς του, οι *εργασίες για το σπίτι*, οι οποίες αποσκοπούν στην ανάπτυξη δεξιοτήτων αναζήτησης πληροφοριών, καλλιέργεια της αυτοπειθαρχίας και διαχείρισης χρόνου και τέλος τις *γραπτές εξετάσεις*, μέσω των οποίων εξετάζονται οι επιδόσεις των μαθητών.

Οι παραδοσιακοί τρόποι αξιολόγησης παρουσιάζουν τόσο πλεονεκτήματα όσο και μειονεκτήματα. Ανάμεσα στα πλεονεκτήματα είναι ότι τα ερωτήματα που τίθενται προσφέρουν τη δυνατότητα στο μαθητή να εκφράζει την προσωπικότητά του. Κάθε απάντηση απαιτεί από το μαθητή οργάνωση, συνδυασμό εννοιών, σύνθεση γνώσεων και παρουσίαση τους. Τα σημαντικότερα μειονεκτήματα που εμφανίζονται είναι: ότι ο εκπαιδευτικός δεν έχει την απόλυτη αντικειμενικότητα για να βαθμολογήσει τους μαθητές, στα θέματα ανάπτυξης παρουσιάζεται αναξιοπιστία της βαθμολογίας, ο μαθητής υποχρεώνεται να προβεί σε απομνημόνευση και τέλος όταν οι γραπτές εξετάσεις γίνονται από το μαθητή αυτοσκοπός υποβαθμίζεται η εκπαιδευτική διαδικασία.

Ανάμεσα στους εναλλακτικούς τρόπους αξιολόγησης των μαθητών περιλαμβάνονται: η *παρατήρηση της καθημερινής εργασίας*, μέσω της οποίας ο εκπαιδευτικός είναι σε θέση να αξιολογήσει τα χαρακτηριστικά της ατομικότητας του μαθητή, της συμπεριφοράς του και της προσπάθειας του, η *αξιολόγηση πρότζεκτ*, μέσω της οποίας αξιοποιείται η δημιουργία και η δράση του μαθητή, η *αξιολόγηση μέσω φακέλου (portfolio)*, η οποία περιλαμβάνει στοιχεία που αφορούν τη σφαιρικότερη εκπαιδευτική αξιολόγηση, η *μαθητική αυτό αξιολόγηση και έτερο- αξιολόγηση*, κατά την οποία η συμμετοχή των μαθητών είναι ενεργή, η *κλίμακα των γραμμάτων*, όπου οι αριθμοί αντικαθίστανται

από γράμματα, η περιγραφική αξιολόγηση, μέσω της οποίας οι επιδόσεις των μαθητών περιγράφονται με περισσότερη σαφήνεια και πληρότητα και τέλος το σύστημα της πολλαπλής βαθμολόγησης, κατά το οποίο υπάρχει η αριθμητική βαθμολογία αλλά βαθμολογούνται και άλλοι παράγοντες όπως: η προσπάθεια και η συμμετοχή στο μάθημα. Τα τελευταία δέκα πέντε χρόνια ως μέθοδος αξιολόγησης χρησιμοποιείται και η χρήση ειδικών εφαρμογών πληροφορικής, οι οποίες προκαλούν το έντονο ενδιαφέρον των μαθητών.

Οι εναλλακτικοί τρόποι αξιολόγησης παρουσιάζουν επίσης έναν αριθμό πλεονεκτημάτων και μειονεκτημάτων. Για παράδειγμα κατά την αξιολόγηση με παρατήρηση ο εκπαιδευτικός θα πρέπει να είναι ικανός να διαχωρίζει τα ζωντανά προϊόντα της μάθησης από την απλή αναμετάδοση γνώσεων. Όσον αφορά τα πλεονεκτήματα η αξιολόγηση μέσω πρότζεκτ αναδεικνύεται τη συνεργασία ανάμεσα στους μαθητές, ενώ η αξιολόγηση βάσει φακέλου (portfolio) μπορεί να βοηθήσει τον εκπαιδευτικό να διαμορφώσει την προφορική βαθμολογία του μαθητή με συνεκτίμηση του προφορικού βαθμού και του περιεχομένου του. Με τη μαθητική αυτό-αξιολόγηση και την έτερο-αξιολόγηση αναπτύσσονται τα εσωτερικά κίνητρα ενώ με το σύστημα της πολλαπλής βαθμολόγησης ενθαρρύνει ιδιαίτερα τους πιο αδύναμους μαθητές να αποκτήσουν καλύτερους βαθμούς. Μέσα από τις ειδικές εφαρμογές πληροφορικής, οι μαθητές μπορούν να αυτό-αξιολογηθούν και να διασκεδάσουν ταυτόχρονα.

Για να μπορέσει ένας εκπαιδευτικός να προβεί σε εποικοδομητική και σωστή αξιολόγηση ενός μαθητή θα πρέπει να περιλάβει κατά τη διάρκεια της διαδικασίας της αξιολόγησης τα εξής στοιχεία: την αξιολόγηση των γνωστικών και μεταγνωστικών στρατηγικών του μαθητή, την αξιολόγηση των γνώσεων και των δεξιοτήτων του, πληροφορίες για τα ενδιαφέροντά του, τα κίνητρα του αλλά και τα χαρακτηριστικά από το οικογενειακό περιβάλλον του. Το βασικότερο εργαλείο για μια επιτυχή εκπαιδευτική αξιολόγηση είναι η επιλογή του κατάλληλου εργαλείου, η οποία θα παράσχει στον εκπαιδευτικό μεγάλο όγκο πληροφοριών ώστε να ερμηνεύσει και να βελτιώσει το περιεχόμενο και τον τρόπο διδασκαλίας που θα επιλέξει.

Για την καλύτερη αξιολόγηση των μαθητών το ιδανικότερο είναι ο εκπαιδευτικός να συνδυάσει τους παραδοσιακούς τρόπους αξιολόγησης με τους εναλλακτικούς τρόπους αξιολόγησης. Ο συνδυασμός των μεθόδων προσφέρει τα εξής πλεονεκτήματα: *έκφραση της προσωπικότητας του, αποτέλεσμα των παραδοσιακών μεθόδων αξιολόγησης, ανάπτυξη εσωτερικών κινήτρων, συνεργασία μεταξύ μαθητών, βελτίωση επίδοσης, δημιουργικότητα, κριτική σκέψη, αποτέλεσμα των εναλλακτικών μεθόδων αξιολόγησης.* Μέσω μιας τέτοιας συνδυαστικής διαδικασίας ο εκπαιδευτικός συλλέγει μεγάλο πλήθος πληροφοριών: για τους μαθητές, για το περιεχόμενο της διδασκαλίας, τις μεθόδους διδασκαλίας και το εκπαιδευτικό υλικό ώστε να μπορεί να προβεί σε βελτίωση της ποιότητας της εκπαιδευτικής διαδικασίας μέσα στην τάξη.

2.4 Διδακτική και Νέες Τεχνολογίες

Ένα από τα βασικά ερωτήματα που απασχολεί τις τελευταίες δεκαετίες την παιδαγωγική έρευνα, καθώς και την εκπαιδευτική κοινότητα γενικότερα, αφορά στο πώς ευνοείται η οικοδόμηση των γνώσεων στο πλαίσιο ατομικών ή συλλογικών καταστάσεων διδασκαλίας. Με άλλα λόγια, πρόκειται για ένα ερώτημα το οποίο σχετίζεται άμεσα με το πρόβλημα της σχολικής μάθησης και της Διδακτικής.

Σε αυτό το ερευνητικό πλαίσιο αναπτύχθηκε η *Διδακτική των Επιστημών*, η οποία εξετάζει τις διαδικασίες μετάδοσης, πρόσκτησης και οικοδόμησης των γνώσεων με απώτερο στόχο τη βελτίωση αυτών των διαδικασιών. Μελετά, με άλλα λόγια, τις συνθήκες μέσα στις οποίες οι μαθητές μαθαίνουν ή δεν μαθαίνουν και εστιάζει την προσοχή της στα ιδιαίτερα προβλήματα που ανακύπτουν από το περιεχόμενο των γνώσεων όσο και των δεξιοτήτων που πρέπει να προσκτηθούν.

Υπό το πρίσμα αυτό, η *Διδακτική* ενδιαφέρεται για τους τρόπους με τους οποίους ευνοείται η οικοδόμηση των γνώσεων στο πλαίσιο ατομικών ή συλλογικών καταστάσεων διδασκαλίας. Ωστόσο, μέχρι σήμερα δεν υπάρχει μια θεωρία γενικής Διδακτικής και, συνακόλουθα, γίνεται χρήση των εννοιών και των θεωρητικών πλαισίων που αναπτύσσονται στις διδακτικές των διαφόρων μαθημάτων και κυρίως αυτών που έχουν διαμορφωθεί στη Διδακτική των Επιστημών.

Η πρόοδος στη *Διδακτική των Επιστημών* είναι ραγδαία κατά την τελευταία εικοσαετία και συνιστάται κυρίως στη δημιουργία ενός θεωρητικού πλαισίου που βασίζεται στις εποικοδομητικές και κοινωνικοπολιτισμικές ψυχολογικές θεωρίες. Οι πρώτες προσπάθειες δημιουργίας ενός θεωρητικού πλαισίου που να ξεφεύγει από τον εμπειρισμό ξεκίνησαν από το χώρο της διδακτικής των

μαθηματικών. Το θεωρητικό αυτό πλαίσιο γνώρισε αρκετές διακυμάνσεις, αλλά τελικά κατάφερε να παγιώσει μια σειρά από βασικές έννοιες και μια μεθοδολογική προσέγγιση που είναι πλέον αποδεκτές στις μέρες μας από το σύνολο σχεδόν των ερευνητών στο χώρο της Διδακτικής των Επιστημών.

Μπορούμε να διακρίνουμε δύο μεγάλους άξονες αυτού του θεωρητικού πλαισίου:

- Την επιστημολογία των γνώσεων
- Τη γένεση και την πρόσκτηση των γνώσεων από τα υποκείμενα που βρίσκονται σε κατάσταση μάθησης και την τοποθέτηση αυτής της γένεσης μέσα σε πραγματικές σχολικές καταστάσεις.

Βασικό πεδίο μελέτης της διδακτικής συνιστά καταρχήν ο τομέας της ανάπτυξης των περιεχομένων. Ο τομέας αυτός μελετάται κάτω από ποικίλα πρίσματα. Αφενός κάτω από το πρίσμα του διδακτικού μετασχηματισμού και των κοινωνικών πρακτικών αναφοράς και αφετέρου κάτω από το πρίσμα των διαδικασιών συγκρότησης σχετικών με το πλαίσιο προγράμματος σπουδών και το αναλυτικό πρόγραμμα, όπως επίσης και με το παραγόμενο διδακτικό υλικό δηλαδή τα σχολικά βιβλία και εγχειρίδια, το βιβλίο καθηγητή καθώς και το εκπαιδευτικό υλικό και λογισμικό.

Στη συνέχεια, βασικό άξονα μελέτης αποτελεί ο τομέας των στρατηγικών της οικοδόμησης των γνώσεων και της κατανόησης των διαδικασιών της μάθησης, ο οποίος κατά κανόνα αφορά το πλαίσιο σχολικών συνθηκών. Τόσο η κατανόηση των στρατηγικών, όσο και η μελέτη των διαδικασιών μάθησης, απαιτούν τη μελέτη των αναπαραστάσεων, της εννοιολογικής αλλαγής, και των διαδικασιών επίλυσης προβλημάτων. Σημαντικός τομέας είναι, επίσης, αυτός της οικοδόμησης διδακτικών καταστάσεων που αφορά στο διδακτικό συμβόλαιο, στην επιμόρφωση των εκπαιδευτικών, αλλά και στη συγκρότηση διδακτικού υλικού.

Ο τομέας των διδακτικών αλληλεπιδράσεων, όπως η διδακτική βοήθεια, η γνωστική και η κοινωνικογνωστική σύγκρουση, καθώς και ο τομέας των χρησιμοποιούμενων μέσων που διαμεσολαβούν αυτές τις αλληλεπιδράσεις είναι άλλος ένας σημαντικός άξονας μελέτης της διδακτικής. Κάτω από το πρίσμα αυτό, η ύπαρξη ενός τεχνολογικού μέσου όπως ο υπολογιστής φαίνεται να δρα καταλυτικά κατά τη διδασκαλία της πληροφορικής ή κατά τη διδασκαλία με χρήση των τεχνολογιών της πληροφορίας και των επικοινωνιών.

Ο τομέας αυτός, μάλιστα, είναι ιδιαίτερα σημαντικός στην περιοχή της διδακτικής της πληροφορικής, αφού είναι γνωστό ότι κύριο ρόλο στη διδακτική πράξη διαδραματίζει η αλληλεπίδραση του μαθητή με το τεχνικό μέσο και οι παρεμβάσεις που παρέχει ο εκπαιδευτικός.

Τέλος, δύο σημαντικοί άξονες μελέτης αποτελούν αφενός τα σύγχρονα προγράμματα σπουδών και αφετέρου οι στάσεις και αντιλήψεις των εκπαιδευτικών πληροφορικής. Ο τομέας ανάπτυξης σύγχρονου προγράμματος σπουδών και κατάλληλου εκπαιδευτικού υλικού αποτελεί βασικό πεδίο μελέτης της διδακτικής. Τόσο ο σχεδιασμός των προγραμμάτων σπουδών, όσο και του εκπαιδευτικού υλικού, πρέπει να λαμβάνουν υπόψη τους και να βασίζονται στα αποτελέσματα των ερευνών που αφορούν όλες τις προηγούμενες κατηγορίες.

Ιδιαίτερα σημαντικός είναι επίσης ο τομέας των στάσεων και των αντιλήψεων των εκπαιδευτικών Πληροφορικής, αλλά και όλων των εκπαιδευτικών που χρησιμοποιούν τις ΤΠΕ στην εκπαιδευτική τους πρακτική, αφού άπτεται άμεσα της οργάνωσης των διδακτικών παρεμβάσεων και των σχολικών πρακτικών.

Ωστόσο, στους χώρους της διδακτικής και της εκπαιδευτικής ψυχολογίας δεν υπάρχει μια ενιαία θεωρία που να περιγράφει τις διαδικασίες μάθησης.

Υπάρχουν τρία μεγάλα ψυχολογικά ρεύματα που έχουν διαμορφώσει αντίστοιχα θεωρητικά μοντέλα να περιγράψουν ή και να εξηγήσουν, στον έναν ή στον άλλο βαθμό, την σχολική μάθηση και τις αντίστοιχες διδακτικές παρεμβάσεις:

- Ο συμπεριφορισμός (behaviorism), που θεωρεί ότι η γνώση μεταδίδεται
- Ο εποικοδομητισμός (constructivism), που θεωρεί ότι η γνώση είναι προϊόν προσωπικής οικοδόμησης μέσω της αλληλεπίδρασης με το περιβάλλον
- Και οι κοινωνικοπολιτισμικές (sociocultural) ή ιστορικοπολιτισμικές (historicocultural) προσεγγίσεις, που αντιλαμβάνονται την γνώση και την μάθηση ως παράγωγα της κοινωνικής αλληλεπίδρασης και του πολιτισμού.

Στη διδακτική, ο εποικοδομισμός αφορά τις εκπαιδευτικές διαδικασίες οι οποίες τοποθετούν το μαθητή στο κέντρο της σχολικής μάθησης. Κάτω από το πρίσμα αυτό, το εποικοδομητικό μοντέλο αντιτίθεται στο μοντέλο μετάδοσης της γνώσης. Βασική διδακτική θέση του εποικοδομητισμού είναι

ότι οι γνώσεις δεν μεταδίδονται. Οφείλουν να οικοδομηθούν ή να αναδομηθούν από το μαθητή, ο οποίος, είναι υπεύθυνος για τη μάθηση του. Ο μαθητής οικοδομεί τις γνώσεις του με βάση μια διερεύνηση της πραγματικότητας, η οποία συμπεριλαμβάνει και τη γνώση με τις διαφορές μορφές μετάδοσης της.

Τη γνώση αυτή την ιδιοποιείται με μη γραμμικό τρόπο, μέσω διαφοροποιήσεων, γενικεύσεων και ρήξεων. Η ιδιοποίηση αυτή της γνώσης βασίζεται πάνω σε εξατομικευμένες οικοδομήσεις αλλά και σε συλλογικές καταστάσεις τάξης, όπου εμφανίζονται γνωστικές συγκρούσεις, ικανές να προχωρήσουν την οικοδόμηση των γνώσεων.

3. Μεθοδολογία Υλοποίησης

Για το σχεδιασμό της διαδικτυακής εκπαιδευτικής εφαρμογής μελετήθηκαν τα βασικά στοιχεία του παιχνιδιού οι κανόνες και οι ερωτήσεις. Από εδώ και πέρα θα αναφέρεται το σύνολο των κανόνων του παιχνιδιού και οι ερωτήσεις σαν εφαρμογή ενώ ο παίκτης θα αναφέρεται σαν χρήστη. Το πεδίο εφαρμογής του παρόντος έργου αφορά την ανάπτυξη της σε πειραματικό / δοκιμαστικό επίπεδο, γεγονός που θα μεταφραστεί σε: περιορισμένο πλήθος χρηστών, μικρό αριθμό ταυτόχρονων χρήσεων και μικρό αριθμό διαθέσιμων ερωτήσεων.

Ο χρήστης θα πρέπει να μπορεί να παίζει το παιχνίδι, να δει το σκορ που έχει επιτύχει κάθε φορά και να δει σε ποια θέση βρίσκεται σε σχέση με τους άλλους χρήστες. Η εφαρμογή από την άλλη θα πρέπει να είναι σε θέση να αναγνωρίζει τους χρήστες, να διαχειριστεί τις ερωτήσεις και να προβάλει ερωτήσεις διαφορετικού επιπέδου δυσκολίας ανάλογα με το πόσο έχει προχωρήσει ο χρήστης στο παιχνίδι.

Η εφαρμογή θα ακολουθήσει το γενικό μοντέλο server-client με τον server να αναλαμβάνει το μεγαλύτερο μέρος του φόρτου εργασίας της εφαρμογής και με τον client να αναλαμβάνει την απεικόνιση των στοιχείων που στέλνει ο server, στον χρήστη.

3.1 Μέθοδος Ανάλυσης & Ανάπτυξης Πτυχιακής

Η μεθοδολογία που χρησιμοποιείται στο μοντέλο της Ενοποιητικής Διαδικασίας,(Unified Process) κρίθηκε ως η καταλληλότερη για τη συγκεκριμένη διαδικτυακή εκπαιδευτική εφαρμογή. Η μέθοδος αναλύεται σε 4 βασικά στάδια:

- *Στάδιο της σύλληψης.* Το στάδιο της σύλληψης θεσπίζει το επιχειρησιακό κομμάτι του συστήματος, δηλαδή για ποιόν λόγο πρέπει να αναλάβουμε αυτό το εγχείρημα, ποιο το πεδίο εφαρμογής του, ποιες είναι οι βασικές απαιτήσεις του και με ποιους τρόπους μπορεί να υλοποιηθεί.
- *Στάδιο της επεξεργασίας.* Ο κύριος στόχος είναι να μετριάσει τα βασικά στοιχεία των κινδύνων που προσδιορίζονται από την ανάλυση του έργου έως το τέλος αυτού του σταδίου. Στο στάδιο επεξεργασίας είναι εκεί που το έργο παίρνει μορφή. Σε αυτό το στάδιο γίνεται η ανάλυση του τομέα των προβλημάτων και η αρχιτεκτονική του έργου παίρνει την βασική της μορφή.
- *Στάδιο της υλοποίησης.* Η υλοποίηση είναι το μεγαλύτερο στάδιο του έργου. Σε αυτό το στάδιο το υπόλοιπο του έργου 'χτίζεται' πάνω στα θεμέλια που έβαλε το στάδιο της επεξεργασίας. Τα στοιχεία του συστήματος υλοποιούνται με μία σειρά από σύντομες και χρονικά περιορισμένες επαναλήψεις. Από κάθε επανάληψη προκύπτει μια λειτουργική έκδοση του προγράμματος. Είναι σύνηθες να γράφονται πλήρη σενάρια χρήσης κατά την διάρκεια του σταδίου της υλοποίησης τα οποία στη συνέχεια γίνονται η αρχή μιας νέας επανάληψης.
- *Στάδιο της μετάβασης.* Στην φάση της μετάβασης το έργο παραδίδεται στους χρήστες. Τα σχόλια και οι παρατηρήσεις των χρηστών από αυτή την αρχική παράδοση έχουν σαν αποτέλεσμα την περαιτέρω βελτίωση του έργου σε βάθος πολλαπλών επαναλήψεων αυτής της φάσης. Η φάση μετάβασης επίσης περιλαμβάνει τυχών μετατροπές του έργου και την εκπαίδευση των χρηστών

3.1.1 Συστήματα Διαχείρισης Βάσεων Δεδομένων

Λόγο της φύσης της εφαρμογής υποχρεωτικά θα υπάρχει ένας συνδυασμός αρχιτεκτονικών. Συγκεκριμένα υπάρχουν τρεις κύριες κατηγορίες:

- Η αρχιτεκτονική της βάσης δεδομένων που θα διαχειρίζεται τις ερωτήσεις με τις απαντήσεις τους, τα στοιχεία των χρηστών και κάποια γενικά στατιστικά.
- Η αρχιτεκτονική της γλώσσας προγραμματισμού που θα χρησιμοποιηθεί για την υλοποίηση της λογικής του παιχνιδιού και τέλος
- Η γλώσσα προγραμματισμού που θα αναλάβει την απεικόνιση των στοιχείων της εφαρμογής στον χρήστη.

Για την βάση δεδομένων της εφαρμογής υπάρχουν τα ακόλουθα υποψήφια συστήματα διαχείρισης βάσεων δεδομένων:

- Η MySQL, η οποία είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων που μετρά περισσότερες από 11 εκατομμύρια εγκαταστάσεις. Το πρόγραμμα τρέχει έναν εξυπηρετητή (server) παρέχοντας πρόσβαση πολλών χρηστών σε ένα σύνολο βάσεων δεδομένων. Ο κωδικός του εγχειρήματος είναι διαθέσιμος μέσω της GNU General Public License, καθώς και μέσω ορισμένων ιδιόκτητων συμφωνιών. Η MySQL είναι δημοφιλής βάση δεδομένων για διαδικτυακά προγράμματα και ιστοσελίδες.
- Το Microsoft SQL Server, το οποίο είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων που αναπτύχθηκε από την Microsoft. Σαν βάση δεδομένων είναι ένα λογισμικό που σαν κύρια λειτουργία του είναι η αποθήκευση και η ανάκτηση δεδομένων με βάση τις αιτήσεις άλλων λογισμικών, είτε αυτά βρίσκονται στον ίδιο υπολογιστή είτε σε άλλους που επικοινωνούν μέσω κάποιου δικτύου. Υπάρχουν πολλές διαφορετικές εκδόσεις του Microsoft SQL Server που η κάθε μια στοχεύει σε διαφορετικό κοινό και σε διαφορετικές απαιτήσεις φόρτου εργασίας. Οι κύριες γλώσσες επικοινωνίας του είναι η T-SQL και η ANSI SQL.
- Η Microsoft Access, γνωστή και ως Microsoft Office Access, η οποία είναι ένα σύστημα διαχείρισης βάσεων δεδομένων από την Microsoft που συνδυάζει την σχεσιακή Microsoft Jet Database Engine με ένα γραφικό περιβάλλον για τον χρήστη και εργαλεία για ανάπτυξη λογισμικού. Είναι μέλος της οικογένειας λογισμικού του Microsoft Office, που συμπεριλαμβάνετε από την έκδοση Professional και πάνω ή πωλείται ξεχωριστά. Η Microsoft Access αποθηκεύει τα δεδομένα με δικό της πρωτόκολλο που στηρίζεται πάνω στην Access Jet Database Engine. Μπορεί ακόμα να εισαγάγει ή να συνδεθεί με δεδομένα από άλλα λογισμικά και από άλλες βάσεις δεδομένων. Οι σχεδιαστές λογισμικού και οι σχεδιαστές δεδομένων μπορούν να χρησιμοποιήσουν την Microsoft Access για να αναπτύξουν λογισμικά και οι έμπειροι χρήστες μπορούν να την χρησιμοποιήσουν για να φτιάξουν εφαρμογές.

3.1.2 Γλώσσες Προγραμματισμού

Για τον προγραμματισμό της λογικής της εφαρμογής υπάρχουν τρεις κύριες υποψήφιες γλώσσες προγραμματισμού, η Java, η C και η C++.

- Η C είναι μια διαδικαστική γλώσσα προγραμματισμού γενικής χρήσης η οποία αναπτύχθηκε στις αρχές της δεκαετίας 1970-1980 από τον Ντένις Ρίτσι στα εργαστήρια Bell Labs για να χρησιμοποιηθεί για την ανάπτυξη του λειτουργικού συστήματος UNIX. Από τότε χρησιμοποιείται ευρύτατα, και ιδιαίτερα για ανάπτυξη προγραμμάτων συστήματος (system software) αλλά και για απλές εφαρμογές. Οι λόγοι της ραγδαίας ανάπτυξης της συγκεκριμένης γλώσσας προγραμματισμού είναι η ταχύτητα της, καθώς και το γεγονός ότι είναι διαθέσιμη στα περισσότερα σημερινά λειτουργικά συστήματα.
- Η C++ είναι μια γενικού σκοπού γλώσσα προγραμματισμού. Θεωρείται μέσου επιπέδου γλώσσα, καθώς περιλαμβάνει έναν συνδυασμό χαρακτηριστικών από γλώσσες υψηλού και χαμηλού επιπέδου. Υποστηρίζει δομημένο, αντικειμενοστραφή και γενικό προγραμματισμό. Η γλώσσα αναπτύχθηκε από τον Μπιάρνε Στρούστρουπ το 1979 στα εργαστήρια Bell της AT&T, ως βελτίωση της ήδη υπάρχουσας γλώσσας προγραμματισμού C, και αρχικά ονομάστηκε "C with Classes", δηλαδή C με Κλάσεις. Μετονομάστηκε σε C++ το 1983. Οι βελτιώσεις ξεκίνησαν με την προσθήκη κλάσεων, και ακολούθησαν, μεταξύ άλλων, εικονικές συναρτήσεις, υπερφόρτωση τελεστών, πολλαπλή κληρονομικότητα, πρότυπα κ.α.
- Η Java είναι μια γενικής χρήσης, αντικειμενοστραφής γλώσσα προγραμματισμού που έχει σχεδιαστεί ειδικά για να έχει όσο το δυνατόν λιγότερες εξαρτήσεις για την εκτέλεση της. Σκοπός της είναι να επιτρέπει στους προγραμματιστές να γράφουν κώδικα που μπορεί να εκτελεστεί σε οποιαδήποτε πλατφόρμα χωρίς καμία τροποποίηση. Οι εφαρμογές της συνήθως συντάσσονται σε αρχεία class που μπορούν στην συνέχεια να εκτελεστούν από οποιαδήποτε εικονική μηχανή Java (JVM), ανεξάρτητα από την αρχιτεκτονική του υπολογιστή. Η Java αναπτύχθηκε αρχικά από τον James Gosling της Sun Microsystems (η οποία έκτοτε συγχωνεύονται σε Oracle Corporation) και κυκλοφόρησε το 1995 ως βασικό στοιχείο της πλατφόρμας Java της Sun Microsystems. Μεγάλο μέρος του συντακτικού της γλώσσας

προέρχεται από την C και την C++, αλλά έχει λιγότερες παροχές χαμηλού επιπέδου από ό,τι οποιαδήποτε από αυτές.

Τέλος για τον προγραμματισμό της διεπαφής του χρήστη, οι υποψήφιες γλώσσες προγραμματισμού είναι οι:

- Η ASP (Active Server Pages), ήταν η πρώτη γλώσσα δέσμης ενεργειών από την πλευρά του διακομιστή (server-side script engine) της Microsoft για την δημιουργία δυναμικών ιστοσελίδων. Η αρχική της διάθεση ήταν σαν πρόσθετο των Υπηρεσιών Πληροφοριών του Ίντερνετ (Internet Information Services) στο προαιρετικό πακέτο των Windows NT4.0 και στην συνέχεια συμπεριλαμβανόταν σαν μέρος των Windows Server (από την έκδοση των Windows 2000 Server και ύστερα). Ιστοσελίδες με την κατάληξη .asp χρησιμοποιούν την ASP, παρόλα αυτά μερικές ιστοσελίδες αποκρύπτουν ποια γλώσσα χρησιμοποιούν για λόγους ασφαλείας χρησιμοποιώντας τις πιο κοινές καταλήξεις .htm και .html.
- Η ASP.NET είναι ένα δικτυακό πλαίσιο εφαρμογών από την πλευρά του διακομιστή που έχει σχεδιαστεί για την παραγωγή δυναμικών ιστοσελίδων. Αναπτύσσεται από την Microsoft για να επιτρέψει στους προγραμματιστές να κατασκευάζουν δυναμικές ιστοσελίδες, δικτυακές εφαρμογές και υπηρεσίες δικτύου. Κυκλοφόρησε για πρώτη φορά τον Ιανουάριο του 2002 με την έκδοση 1.0 του .NET Framework, και είναι ο διάδοχος της τεχνολογίας ASP της Microsoft.
- Η JSP (Java Server Pages) είναι μια τεχνολογία που βοηθά τους προγραμματιστές να δημιουργήσουν δυναμικές ιστοσελίδες που βασίζονται πάνω στην HTML, XML, ή άλλους τύπους εγγράφων. Έχοντας κυκλοφορήσει το 1999 από την Sun Microsystems, είναι παρόμοια με την PHP αλλά χρησιμοποιεί τη γλώσσα προγραμματισμού Java. Αρχιτεκτονικά, η JSP μπορεί να θεωρηθεί ως ένα υψηλό επίπεδο αφαίρεσης των Java servlets. Η JSP μεταφράζεται σε servlets κατά την εκτέλεση και κάθε servlet αποθηκεύεται και ξαναχρησιμοποιείται μέχρι να τροποποιηθεί το αρχικό JSP αρχείο. Η JSP μπορεί να χρησιμοποιηθεί ανεξάρτητη ή σαν τη διεπαφή μιας εφαρμογής στην πλευρά του διακομιστή.
- Η PHP είναι μια γλώσσα δέσμης ενεργειών από την πλευρά του διακομιστή για την ανάπτυξη δικτυακών εφαρμογών, αλλά χρησιμοποιείται επίσης και ως μια γενικής χρήσης γλώσσα προγραμματισμού. Αρχικά δημιουργήθηκε από τον Rasmus Lerdorf το 1995, η εφαρμογή αναφοράς της PHP παράγεται πλέον από το «The PHP Group». Ενώ αρχικά PHP σήμαινε «Personal Home Page» (Προσωπική Σελίδα), πλέον σημαίνει PHP: Hypertext Preprocessor (PHP: Προ-επεξεργαστής υπερκειμένου). Ο κώδικας της PHP ερμηνεύεται από τον διακομιστή μέσω ειδικού λογισμικού το οποίο παράγει στην συνέχεια μια ιστοσελίδα. Εντολές της PHP μπορούν να ενσωματωθούν απευθείας μέσα σε ένα αρχείο HTML.

4. Σχέδιο Δράσης για την εκπόνηση της εργασίας

Ο πρωταρχικός σκοπός του σταδίου της επεξεργασίας είναι ο μετριασμός των κυρίων παραγόντων κίνδυνου που θα αναγνωριστούν από την ανάλυση που εκτελείτε έως το τέλος αυτού του σταδίου. Είναι το σημείο όπου το έργο αρχίζει να παίρνει σχήμα.

Σκοπός του συγκεκριμένου σταδίου είναι να αναλυθεί σε μεγαλύτερο βάθος το πεδίο προβλήματος και να κατασκευασθεί μια πρώτη καλή αρχιτεκτονική, σταθερή και οργανωμένη, όσο αναφορά στην εμφάνιση της ιστοσελίδας. Είναι ο πρώτος «σκελετός» της ιστοσελίδας, με την οργανωτική δομή που θα περιέχει αυτή. Ποια είναι τα στοιχεία που θα εμπεριέχονται σε αυτήν. Κάθε στοιχείο αποτελεί μια επιμέρους περίπτωση χρήσης του γενικότερου πεδίου προβλήματος. Οποιαδήποτε αλλαγή πάνω στην βασική αρχιτεκτονική προστίθεται αργότερα ως πρότυπο.

4.1 State of the Art

Η εφαρμογή μας θα αποτελείται από τρία κυρίως τμήματα, την Βάση Δεδομένων όπου θα αποθηκεύονται όλες οι πληροφορίες σχετικά με τους Χρήστες, της Ερωτήσεις και το Ιστορικό των Χρηστών, την λογική της εφαρμογής η οποία θα εκτελείται στον εξυπηρετητή που θα φιλοξενεί την εφαρμογή μας και θα αναλαμβάνει το μεγαλύτερο μέρος της επεξεργασίας, αν όχι όλο, των δεδομένων από τους χρήστες και την διεπαφή της εφαρμογής μας, που στην προκειμένη περίπτωση θα είναι διαδραστικές ιστοσελίδες, η οποία θα αναλαμβάνει την απεικόνιση των δεδομένων, που θα στέλνει ο εξυπηρετητής, μέσω κάποιου προγράμματος περιήγησης του διαδικτύου στον υπολογιστή του χρήστη.

Από τα παραπάνω είναι προφανές ότι οι λύσεις που θα επιλέξουμε για κάθε κομμάτι του συστήματος θα πρέπει να μπορούν να συνεργαστούν τουλάχιστον με ένα από τα άλλα δύο. Από τις υποψήφιες τεχνολογίες μπορούμε εύκολα να παραβλέψουμε και από μία υποψήφια επιλογή σε κάθε τμήμα λόγο παλαιότητας και λόγο εύρους χρήσης. Θα προσπεράσουμε την γλώσσα προγραμματισμού C μιας και αποτελεί υποσύνολο της C++ όπως θα παραβλέψουμε και την ASP γιατί έχει αντικατασταθεί πλέον από την ASP.NET. Τέλος όσο αφορά τις υποψηφιότητες για την βάση δεδομένων θα πρέπει να παραβλέψουμε την Microsoft Access γιατί και οι δύο άλλες υποψήφιες τεχνολογίες μπορούν να διαχειριστούν πολύ μεγαλύτερους όγκους δεδομένων και πολλές περισσότερες συνδέσεις χρηστών ταυτόχρονα.

Έχοντας περιορίσει πλέον της υποψήφιες λύσεις μας σε δύο για την βάση δεδομένων, δύο για την γλώσσα προγραμματισμού της λογικής του συστήματος και τρεις για τον προγραμματισμό της διεπαφής του χρήστη είναι πλέον ευκολότερο να καταλήξουμε στις τεχνολογίες που θα χρησιμοποιήσουμε για την υλοποίηση του συστήματος. Για την επιλογή του συστήματος διαχείρισης της βάσης δεδομένων τα πράγματα είναι σχετικά εύκολα από την στιγμή που και οι δύο λύσεις μας προσφέρουν περίπου τα ίδια πράγματα αλλά έχουν και μερικές σημαντικές διαφορές. Η MySQL είναι κυρίως δωρεάν και μπορεί να εγκατασταθεί σε όλα τα βασικά λειτουργικά συστήματα (Windows, Linux και Mac OS). Από την άλλη το Microsoft SQL Server πρέπει κυρίως να αγοραστεί και μπορεί να εγκατασταθεί μόνο στο λειτουργικό περιβάλλον των Microsoft Windows. Ως εκ τούτου για το σύστημα διαχείρισης της βάσης δεδομένων μας θα επιλέξουμε την MySQL και ποιο συγκεκριμένα την MySQL Community Server επειδή διατίθεται δωρεάν και μπορεί να εγκατασταθεί σε, πρακτικά, οποιοδήποτε λειτουργικό σύστημα.

Με την επιλογή του συστήματος της βάσης δεδομένων που θα χρησιμοποιήσουμε, μας μένει να προσδιορίσουμε τις γλώσσες που θα χρησιμοποιηθούν για τον προγραμματισμό της λογικής της εφαρμογής και της διεπαφής του χρήστη. Από τα πιθανά ζεύγη λύσεων, το ζεύγος Java και JSP ξεχωρίζει γιατί στην ουσία πρόκειται για την ίδια γλώσσα προγραμματισμού κάτι που θα μας διευκολύνει στην πορεία γιατί θα έχουμε να ασχοληθούμε με την λογική και της ιδιοτροπίες μόνο μίας γλώσσας έναντι δύο.

4.1.1 Το μοντέλο Client Server

Το ευρύτερα διαδεδομένο μοντέλο ανάπτυξης δικτυακών εφαρμογών είναι το μοντέλο του πελάτη - εξυπηρετητή (client - server). Ο εξυπηρετητής είναι μια διεργασία, η οποία εκτελείται σε έναν υπολογιστή και αναμένει να συνδεθεί σε αυτήν κάποιο πρόγραμμα – ο πελάτης, όπως ονομάζεται, για να του παράσχει υπηρεσίες. Ένα τυπικό σενάριο που ακολουθείται συνήθως, είναι το εξής :

Η διεργασία - εξυπηρετητής αρχίζει να εκτελείται σε κάποιον υπολογιστή. Μετά την αρχικοποίηση της, πέφτει σε “λήθαργο”, αναμένοντας μία διεργασία - πελάτη να επικοινωνήσει μαζί της και να της ζητήσει κάποια υπηρεσία.

Μία διεργασία - πελάτης αρχίζει να εκτελείται, είτε στο ίδιο σύστημα, είτε σε κάποιο απομακρυσμένο, το οποίο συνδέεται με τον υπολογιστή στον οποίο “τρέχει” ο εξυπηρετητής μέσω δικτύου. Η διεργασία πελάτης στέλνει μια αίτηση, μέσω του δικτύου, στον εξυπηρετητή, ζητώντας του κάποιου είδους υπηρεσία (π.χ. μεταφορά αρχείου, απομακρυσμένη εκτύπωση, ανάγνωση και αποστολή mail και άλλες).

Ταυτόχρονα με την εξυπηρέτηση κάποιου πελάτη, ο server έχει την δυνατότητα να δέχεται και αιτήσεις άλλων πελατών προς εξυπηρέτηση. Όταν ο εξυπηρετητής τελειώσει με όλους τους πελάτες, τότε ξαναπέφτει σε “λήθαργο”, περιμένοντας για μια καινούργια αίτηση και η διαδικασία ξαναρχίζει από την αρχή.

Ορίζουμε ως σύνδεση, τον επικοινωνιακό δίαυλο μεταξύ των δύο διεργασιών. Την σύνδεση μπορούμε να την περιγράψουμε ως εξής: {πρωτόκολλο, τοπική διεύθυνση, τοπική διεργασία, απομακρυσμένη διεύθυνση, απομακρυσμένη διεργασία}.

Το πρωτόκολλο αναφέρεται στο σύνολο των κανόνων που διέπουν την επικοινωνία. Η τοπική διεύθυνση και απομακρυσμένη διεύθυνση, προσδιορίζουν την ταυτότητα των υποδικτύων και των υπολογιστών, στους οποίους εκτελούνται οι επικοινωνούσες διεργασίες. Η τοπική-διεργασία και απομακρυσμένη-διεργασία, προσδιορίζουν την ταυτότητα των διεργασιών που θα επικοινωνούν, καθώς σε έναν υπολογιστή, μπορούν να εκτελούνται περισσότερες της μιας διεργασίες. Κάθε μία από αυτές τις διεργασίες που εκτελούνται στον ίδιο host και που χρειάζονται επικοινωνία μέσω δικτύου, λαμβάνει έναν ακέραιο αριθμό εύρους δεκαέξι bit, ο οποίος αναπαριστά την θύρα (port number) της διεργασίας και κατ’ επέκταση, της υπηρεσίας.

Το υπολογιστικό μοντέλο πελάτη-εξυπηρετητή έχει τη δομή μιας κατανεμημένης εφαρμογής οι οποία τμηματοποιεί εργασίες ή υπολογιστικά φορτία ανάμεσα στους παρόχους ενός πόρου ή μιας υπηρεσίας, τους οποίους ονομάζουμε εξυπηρετητές, και σε αυτούς που αιτούνται μια υπηρεσία, τους οποίους ονομάζουμε πελάτες. Συχνά πελάτες και εξυπηρετητές επικοινωνούν μέσω ενός υπολογιστικού δικτύου όταν βρίσκονται σε ξεχωριστά μηχανήματα, αν και πελάτες και εξυπηρετητές μπορούν να υπάρχουν και οι δύο στο ίδιο σύστημα. Το μηχάνημα ενός εξυπηρετητή είναι ένα σύστημα στο οποίο τρέχουν ένα και παραπάνω προγράμματα εξυπηρετητές τα οποία μοιράζονται τους πόρους τους με τα προγράμματα πελάτη. Ένας πελάτης δεν μοιράζεται κανέναν από τους πόρους του, αλλά ζητά περιεχόμενο ή λειτουργίες από τον εξυπηρετητή. Για αυτό οι πελάτες είναι αυτοί που αρχίζουν την επικοινωνία με τους εξυπηρετητές οι οποίοι περιμένουν για εισερχόμενα αιτήματα.

Το χαρακτηριστικό client-server χαρακτηρίζει τη σχέση μεταξύ των συνεργαζόμενων προγραμμάτων σε μια εφαρμογή. Το στοιχείο του εξυπηρετητή προσφέρει μια λειτουργία ή μια υπηρεσία σε έναν ή παραπάνω πελάτες, οι οποίοι στέλνουν αιτήματα για αυτές τις υπηρεσίες.

Λειτουργίες όπως η πρόσβαση σε βάσεις δεδομένων, είναι φτιαγμένες πάνω στο μοντέλο client-server. Για παράδειγμα, ένας περιηγητής (browser) είναι ένα πρόγραμμα πελάτης το οποίο τρέχει στον υπολογιστή ενός χρήστη το οποίο μπορεί να προσπελάσει πληροφορίες αποθηκευμένες σε έναν εξυπηρετητή στο διαδίκτυο. Χρήστες οι οποίοι έχουν πρόσβαση σε τραπεζικές υπηρεσίες από τον υπολογιστή τους χρησιμοποιούν έναν περιηγητή ως πελάτη για να στείλουν αιτήματα σε έναν εξυπηρετητή κάποιας τράπεζας. Αυτό το πρόγραμμα μπορεί με τη σειρά του να προωθήσει το αίτημα στο δικό του πρόγραμμα πελάτη για τη βάση δεδομένων το οποίο στέλνει αιτήματα σε έναν εξυπηρετητή βάσης δεδομένων σε κάποιον άλλο υπολογιστή της τράπεζας ώστε να πάρει τις πληροφορίες ενός λογαριασμού. Το υπόλοιπο του λογαριασμού επιστρέφεται στον πελάτη βάσης δεδομένων, ο οποίος με τη σειρά του εξυπηρετεί τον πελάτη περιηγητή ο οποίος προβάλλει τα αποτελέσματα στον χρήστη. Το μοντέλο client-server έχει γίνει μια από τις κεντρικές ιδέες στα υπολογιστικά δίκτυα. Κάθε στιγμιότυπο του λογισμικού πελάτη μπορεί να στείλει αιτήματα για

δεδομένα σε έναν ή πολλούς συνδεδεμένους εξυπηρετητές. Με τη σειρά τους, οι εξυπηρετητές μπορούν να δεχτούν τα αιτήματα, να τα επεξεργαστούν, και να επιστρέψουν την αιτηθείσα πληροφορία στον πελάτη. Αν και αυτή η διαδικασία μπορεί να εφαρμοστεί για πολλούς λόγους σε πολλά διαφορετικά είδη εφαρμογών, η θεμελιώδης αρχιτεκτονική παραμένει η ίδια.

Ο πιο βασικός τύπος αρχιτεκτονικής client-server χρησιμοποιεί μόνο δύο τύπους συστημάτων: πελάτες και εξυπηρετητές. Αυτό το είδος αρχιτεκτονικής αναφέρετε μερικές φορές ως αρχιτεκτονική δύο επιπέδων. Επιτρέπει σε συσκευές να μοιράζονται αρχεία και πόρους. Αρχιτεκτονική δύο επιπέδων σημαίνει ότι ο πελάτης δρα ως το πρώτο επίπεδο και η εφαρμογή σε συνδυασμό με τον εξυπηρετητή ως το δεύτερο επίπεδο. Στο διαδίκτυο χρησιμοποιείτε όλο και περισσότερο η αρχιτεκτονική τριών επιπέδων. Σε αυτήν, η πλευρά του εξυπηρετητή αποτελείται από έναν εξυπηρετητή εφαρμογής, όπως ένας εξυπηρετητής ιστοσελίδων, και έναν εξυπηρετητή βάσης δεδομένων. Έτσι τα τρία επίπεδα είναι τα εξής: Πελάτης, Εξυπηρετητής Εφαρμογής και Βάση Δεδομένων. Και τα τρία επίπεδα είναι σχετικά ανεξάρτητα, για παράδειγμα μπορεί να γίνει αλλαγή σε διαφορετικό εξυπηρετητή ιστοσελίδων ενώ διατηρείτε η ακεραιότητα του μοντέλου.

Σίγουρα στην εφαρμογή υπάρχουν οι απλοί χρήστες και ο διαχειριστής. Και οι δύο μπορούν να παίξουν το παιχνίδι, επομένως είναι και οι δύο παίκτες. Η μόνη διαφορά τους είναι στα δικαιώματα που έχουν επάνω στα στοιχεία της Βάσης Δεδομένων. Για παράδειγμα, ο διαχειριστής μπορεί να εισάγει, τροποποιήσει, διαγράψει μία ερώτηση και τα στοιχεία του, ενώ ο απλός χρήστης μπορεί να εισάγει, τροποποιήσει, διαγράψει μόνο τα δικά του στοιχεία και κανενός άλλου.

Από τα παραπάνω προκύπτει ότι υπάρχει μια οντότητα ΠΑΙΚΤΗΣ, της οποίας τα χαρακτηριστικά κληρονομούν ο απλός χρήστης και ο διαχειριστής. Ένας παίκτης έχει ένα όνομα και έναν κωδικό.

Ακόμη, κάθε παίκτης έχει και ένα Ιστορικό, στο οποίο σημειώνονται οι νίκες, η τελευταία ημερομηνία που έπαιξε το παιχνίδι και τα χρήματα που κέρδισε τις συγκεκριμένες ημερομηνίες. Κάθε φορά που τερματίζεται το παιχνίδι, αυτά τα στοιχεία θα καταχωρούνται στην Βάση Δεδομένων του Ιστορικού. Σε οποιαδήποτε στιγμή, ο παίκτης μπορεί να δει το Ιστορικό του παιχνιδιού, ενώ δεν μπορεί να κάνει καμία τροποποίηση πάνω σ' αυτό.

Ο διαχειριστής μπορεί να δει και να διαγράψει τους λογαριασμούς όλων των απλών χρηστών, αλλά δεν μπορεί να τους τροποποιήσει (για λόγους προστασίας δεδομένων χρηστών). Ακόμη, κάθε απλός χρήστης μπορεί να τροποποιήσει μόνο τα δικά του στοιχεία, ενώ δεν μπορεί να τροποποιήσει ούτε να διαγράψει λογαριασμούς άλλων χρηστών.

Τέλος, έχουμε την οντότητα ΕΡΩΤΗΣΗ, η οποία αποτελείται από ένα αύξων αριθμό-κλειδί, την ερώτηση, την σωστή απάντηση, τις υπόλοιπες τρεις εναλλακτικές απαντήσεις και το επίπεδο δυσκολίας κάθε ερώτησης. Είναι ενόητο πως όσο ο παίκτης ανεβαίνει βαθμίδα στα χρηματικά ποσά, τόσο οι ερωτήσεις θα δυσκολεύουν.

Ο διαχειριστής μπορεί να τροποποιήσει με οποιαδήποτε τρόπο τις ερωτήσεις, ενώ ο απλός χρήστης μπορεί μόνο να τις διαβάσει.

Το πολύ γενικό πλαίσιο της εφαρμογής, είναι ένας χρήστης που στέλνει αιτήσεις στον εξυπηρετητή. Ο εξυπηρετητής έχει πρόσβαση στη βάση δεδομένων, επεξεργάζεται τα στοιχεία και τα δεδομένα του χρήστη και του αποστέλλει τις κατάλληλες απαντήσεις.

4.2 Περιπτώσεις Χρήσης

Οι περιπτώσεις χρήσης είναι μία ακολουθία βημάτων που περιγράφουν μία αλληλεπίδραση ανάμεσα στο χρήστη και στο Σύστημα. Η περίπτωση χρήσης μας βοηθάει να εντοπίσουμε τις απαιτήσεις του συστήματος.

Χωρίζουμε έναν πίνακα σε τρεις στήλες. Η πρώτη στήλη πραγματεύεται τις ενέργειες του χρήστη, η δεύτερη διευκρινίζει τις υποπεριπτώσεις και η τρίτη στήλη τις αποκρίσεις του Συστήματος. Έτσι θα έχουμε πιο σαφή εικόνα των λειτουργιών της εφαρμογής.

4.2.1 Περίπτωση Χρήσης 1: Είσοδος Χρήστη

Η διάδραση χρήστη/συστήματος στην περίπτωση που ο πρώτος θέλει να εισέλθει στο σύστημα.

ΧΡΗΣΤΗΣ	Περίπτώσεις	ΣΥΣΤΗΜΑ
		1. Εμφάνιση πεδίου εισαγωγής username
		2. Εμφάνιση πεδίου εισαγωγής password
3. Εισάγει username		
4. Εισάγει password		
5. Επιλέγει Log In		
		6. Έλεγχος στοιχείων
	Περίπτωση λάθους στοιχείων	
		6.1 Καταχώρηση λάθους ανεπιτυχούς πρόσβασης
		6.2 Επιστροφή στο βήμα 1 και εμφάνιση μηνύματος λάθους
	Περίπτωση σωστών στοιχείων	
		6.3 Προώθηση των στοιχείων του χρήστη στην κεντρική σελίδα του παιχνιδιού
		6.4 Προώθηση χρήστη στην κεντρική σελίδα του παιχνιδιού

Πίνακας 1: Περίπτωση Χρήσης 1: Είσοδος Χρήστη

4.2.2 Περίπτωση Χρήσης 2: Εγγραφή Χρήστη

Η διάδραση χρήστη/συστήματος στην περίπτωση που ο πρώτος θέλει να κάνει εγγραφή στο σύστημα.

ΧΡΗΣΤΗΣ	Περίπτώσεις	ΣΥΣΤΗΜΑ
1. Επιλέγει Sign Up		
		2. Εμφάνιση φόρμας εγγραφής
		3. Εμφάνιση πεδίου εισαγωγής username
		4. Εμφάνιση πεδίου εισαγωγής password
5. Εισάγει username		
6. Εισάγει password		
		7. Έλεγχος στοιχείων
	Περίπτωση που το όνομα υπάρχει	
		7.1 Εμφάνιση μηνύματος επανεισαγωγής username
7.2. Επανεισάγει username		
		7.3 Έλεγχος στοιχείων
		(Επανάληψη βημάτων 7.1-7.3 έως ότου εισαχθεί ελεύθερο username)
	Περίπτωση που το όνομα δεν υπάρχει	

		8. Εμφάνιση κεντρικής οθόνης του παιχνιδιού
--	--	---

Πίνακας 2: Περίπτωση Χρήσης 2: Εγγραφή Χρήστη

4.2.3 Περίπτωση Χρήσης 3: Απεικόνιση Ιστορικού Χρήστη

Η διάδραση χρήστη/συστήματος στην περίπτωση που ο πρώτος θέλει να δει το ιστορικό των βαθμολογιών του στο παιχνίδι.

Προαπαιτήσεις :

- Να είναι εγγεγραμμένος χρήστης (03.02.03 Περίπτωση Χρήσης 02: Εγγραφή Χρήστη)
- Να έχει κάνει είσοδο στην εφαρμογή (03.02.02 Περίπτωση Χρήσης 01: Είσοδος Χρήστη)

ΧΡΗΣΤΗΣ	Περιπτώσεις	ΣΥΣΤΗΜΑ
1. Επιλέγει "View your History"		
		2. Ελέγχει το Όνομα Χρήστη
		3. Διαβάζει από την Βάση Δεδομένων του Ιστορικού όλες της καταχωρίσεις με αυτό το Όνομα Χρήστη
		4. Εμφάνιση στον Χρήστη των καταχωρίσεων

Πίνακας 3: Περίπτωση Χρήσης 3: Απεικόνιση Ιστορικού Χρήστη

4.2.4 Περίπτωση Χρήσης 4: Απεικόνιση των Καλύτερων Αποτελεσμάτων

Η διάδραση χρήστη/συστήματος στην περίπτωση που ο πρώτος θέλει να δει το ιστορικό των καλύτερων βαθμολογιών στο παιχνίδι.

Προαπαιτήσεις :

- Να είναι εγγεγραμμένος χρήστης (4.2.2 Περίπτωση Χρήσης 2: Εγγραφή Χρήστη)
- Να έχει κάνει είσοδο στην εφαρμογή (4.2.1 Περίπτωση Χρήσης 1: Είσοδος Χρήστη)

ΧΡΗΣΤΗΣ	Περιπτώσεις	ΣΥΣΤΗΜΑ
1. Επιλέγει "View High Scores"		
		2. Ταξινομεί την Βάση Δεδομένων του Ιστορικού κατά το υψηλότερο Score
		3. Διαβάζει από την Βάση Δεδομένων τις πρώτες 100 καταχωρίσεις
		4. Εμφάνιση στον Χρήστη των καταχωρίσεων

Πίνακας 4: Περίπτωση Χρήσης 4: Απεικόνιση των Καλύτερων Αποτελεσμάτων

4.2.5 Περίπτωση Χρήσης 5: Απεικόνιση Πρόσφατων Αποτελεσμάτων

Η διάδραση χρήστη/συστήματος στην περίπτωση που ο πρώτος θέλει να δει τις πιο πρόσφατες βαθμολογίες στο παιχνίδι.

Προαπαιτήσεις :

- Να είναι εγγεγραμμένος χρήστης (4.2.2 Περίπτωση Χρήσης 2: Εγγραφή Χρήστη)
- Να έχει κάνει είσοδο στην εφαρμογή (4.2.1 Περίπτωση Χρήσης 1: Είσοδος Χρήστη)

ΧΡΗΣΤΗΣ	Περιπτώσεις	ΣΥΣΤΗΜΑ
1. Επιλέγει "View All Recent History"		
		2. Ταξινομεί την Βάση Δεδομένων του Ιστορικού με βάση την ημερομηνία
		3. Διαβάζει από την Βάση Δεδομένων του Ιστορικού τις πρώτες 200 καταχωρίσεις
		4. Εμφάνιση στον Χρήστη των καταχωρίσεων

Πίνακας 5: Περίπτωση Χρήσης 5: Απεικόνιση Πρόσφατων Αποτελεσμάτων

4.2.6 Περίπτωση Χρήσης 6: Αλλαγή Κωδικού Ασφαλείας

Η διάδραση χρήστη/συστήματος στην περίπτωση που ο πρώτος θέλει να αλλάξει τον κωδικό πρόσβασής του στην εφαρμογή.

Προαπαιτήσεις :

- Να είναι εγγεγραμμένος χρήστης (4.2.2 Περίπτωση Χρήσης 2: Εγγραφή Χρήστη)
- Να έχει κάνει είσοδο στην εφαρμογή (4.2.1 Περίπτωση Χρήσης 1: Είσοδος Χρήστη)

ΧΡΗΣΤΗΣ	Περιπτώσεις	ΣΥΣΤΗΜΑ
1. Επιλέγει "Change your Password"		
		2. Εμφάνιση σελίδας αλλαγής Κωδικού Ασφαλείας
3. Εισάγει τον παλιό Κωδικό Ασφαλείας		
4. Εισάγει τον καινούργιο Κωδικό Ασφαλείας		
5. Επαληθεύει τον καινούργιο Κωδικό Ασφαλείας		
		6. Ελέγχει αν ο καινούργιος Κωδικός Ασφαλείας ταιριάζει με την επαλήθευση από τον χρήστη
	Περίπτωση λάθους στον καινούριο κωδικό	
		7. Επιστροφή στο βήμα 2 και εμφάνιση μηνύματος λάθους
	Περίπτωση επαλήθευσης καινούριων Κωδικών	
		8. Ελέγχει αν ο παλιός Κωδικός Ασφαλείας ταιριάζει με αυτόν που είναι καταχωρημένος
	Περίπτωση λάθους στον παλιό κωδικό	
		9. Αυτόματη έξοδος του χρήστη από το σύστημα και επιστροφή στην σελίδα εισόδου χρήστη
	Περίπτωση επαλήθευσης παλιού Κωδικού	

		10. Αλλαγή του παλιού κωδικού με τον καινούριο
		11. Επιστροφή του χρήστη στην κεντρική σελίδα του παιχνιδιού

Πίνακας 6: Περίπτωση Χρήσης 6: Αλλαγή Κωδικού Ασφαλείας

4.2.7 Περίπτωση Χρήσης 7: Διαγραφή του Λογαριασμού του Χρήστη

Η διάδραση χρήστη/συστήματος στην περίπτωση που ο πρώτος θέλει να διαγράψει τον λογαριασμό του από την εφαρμογή.

Προαπαιτήσεις :

- Να είναι εγγεγραμμένος χρήστης (4.2.2 Περίπτωση Χρήσης 2: Εγγραφή Χρήστη)
- Να έχει κάνει είσοδο στην εφαρμογή (4.2.1 Περίπτωση Χρήσης 1: Είσοδος Χρήστη)

ΧΡΗΣΤΗΣ	Περιπτώσεις	ΣΥΣΤΗΜΑ
1. Επιλέγει "Delete your Account"		
		2. Εμφάνιση σελίδας διαγραφής Χρήστη
3. Εισάγει το όνομα Χρήστη του		
4. Εισάγει τον Κωδικό Ασφαλείας του		
		5. Ελέγχει αν το όνομα και ο κωδικός ασφαλείας ταιριάζουν με τον τρέχων χρήστη
	Περίπτωση ασυμφωνίας ονόματος χρήστη ή κωδικού ασφαλείας	
		6. Αυτόματη έξοδος του χρήστη από το σύστημα και επιστροφή στην σελίδα εισόδου χρήστη
	Περίπτωση επαλήθευσης Ονόματος Χρήστη και Κωδικού Ασφαλείας	
		7. Καθαρισμός ιστορικού χρήστη
		8. Διαγραφή χρήστη
		9. Εμφάνιση ενημερωτικής σελίδας για την επιτυχία της διαγραφής
		10. Επιστροφή στην σελίδα εισόδου χρήστη

Πίνακας 7: Περίπτωση Χρήσης 7: Διαγραφή του Λογαριασμού του Χρήστη

4.2.8 Περίπτωση Χρήσης 8: Έξοδος από το Σύστημα

Η διάδραση χρήστη/συστήματος στην περίπτωση που ο πρώτος θέλει να εξέλθει από την εφαρμογή.

Προαπαιτήσεις :

- Να είναι εγγεγραμμένος χρήστης (4.2.2 Περίπτωση Χρήσης 2: Εγγραφή Χρήστη)

- Να έχει κάνει είσοδο στην εφαρμογή (4.2.1 Περίπτωση Χρήσης 1: Είσοδος Χρήστη)

ΧΡΗΣΤΗΣ	Περιπτώσεις	ΣΥΣΤΗΜΑ
1. Επιλέγει "Log Off"		
		2. Καθαρισμός των στοιχείων Χρήστη από την μνήμη του συστήματος
		3. Εμφάνιση σελίδας Εισόδου Χρήστη

Πίνακας 8: Περίπτωση Χρήσης 8: Έξοδος από το Σύστημα

4.2.9 Περίπτωση Χρήσης 9: Εισαγωγή Ερώτησης

Η διάδραση χρήστη/συστήματος στην περίπτωση που ο πρώτος θέλει να δημιουργήσει καινούρια ερώτηση στην Βάση Δεδομένων των Ερωτήσεων στην εφαρμογή.

Προαπαιτήσεις :

- Να είναι εγγεγραμμένος χρήστης (4.2.2 Περίπτωση Χρήσης 2: Εγγραφή Χρήστη)
- Να έχει κάνει είσοδο στην εφαρμογή (4.2.1 Περίπτωση Χρήσης 1: Είσοδος Χρήστη)
- Ο χρήστης να είναι Διαχειριστής της εφαρμογής

ΧΡΗΣΤΗΣ	Περιπτώσεις	ΣΥΣΤΗΜΑ
1. Επιλέγει "Add Question"		
		2. Εμφάνιση σελίδας προσθήκης ερώτησης
3. Εισαγωγή Βαθμού δυσκολίας		
4. Εισαγωγή Ερώτησης		
5. Εισαγωγή Ορθής Απάντησης		
6. Εισαγωγή Πρώτης Λάθους Απάντησης		
7. Εισαγωγή Δεύτερης Λάθους Απάντησης		
8. Εισαγωγή Τρίτης Λάθους Απάντησης		
	Περίπτωση που ο χρήστης πατήσει "Cancel"	
		9. Αρχικοποίηση όλων των πεδίων της σελίδας
	Περίπτωση που ο χρήστης πατήσει "Confirm"	
		10. Διάβασμα του πεδίου της ερώτησης
		11. Διάβασμα του πεδίου της Σωστής Απάντησης
		12. Διάβασμα του πεδίου της Πρώτης Λάθους Απάντησης
		13. Διάβασμα του πεδίου της Δεύτερης Λάθους Απάντησης
		14. Διάβασμα του πεδίου της Τρίτης Λάθους Απάντησης
		15. Διάβασμα του πεδίου του Βαθμού δυσκολίας

		16. Καταχώριση στην Βάση Δεδομένων των Ερωτήσεων τα στοιχεία που έχουν διαβαστεί
		17. Αρχικοποίηση όλων των πεδίων της σελίδας και αναμονή για την επόμενη καταχώριση

Πίνακας 9: Περίπτωση Χρήσης 9: Εισαγωγή Ερώτησης

4.2.10 Περίπτωση Χρήσης 10: Τροποποίηση Ερώτησης

Η διάδραση χρήστη/συστήματος στην περίπτωση που ο πρώτος θέλει να τροποποιήσει κάποια ερώτηση στην Βάση Δεδομένων των Ερωτήσεων της εφαρμογής.

Προαπαιτήσεις :

- Να είναι εγγεγραμμένος χρήστης (4.2.2 Περίπτωση Χρήσης 2: Εγγραφή Χρήστη)
- Να έχει κάνει είσοδο στην εφαρμογή (4.2.1 Περίπτωση Χρήσης 1: Είσοδος Χρήστη)
- Ο χρήστης να είναι Διαχειριστής της εφαρμογής

ΧΡΗΣΤΗΣ	Περίπτώσεις	ΣΥΣΤΗΜΑ
1. Επιλέγει "Edit Question"		
		2. Εμφάνιση σελίδας
		3. Διάβασμα από την Βάση Δεδομένων των Ερωτήσεων της Πρώτης Ερώτησης
		4. Εμφάνιση του Αύξοντα Αριθμού της Ερώτησης
		5. Εμφάνιση του βαθμού Δυσκολίας της Ερώτησης
		6. Εμφάνιση της Ερώτησης
		7. Εμφάνιση της Σωστής Απάντησης
		8. Εμφάνιση της Πρώτης Λάθους Απάντησης
		9. Εμφάνιση της Δεύτερης Λάθους Απάντησης
		10. Εμφάνιση της Τρίτης Λάθους Απάντησης
	Περίπτωση "Next/Previous"	
11. Χρησιμοποιώντας τα διαθέσιμα Κουμπιά (Next/Previous) Πλοηγείται στην ερώτηση που θέλει να επεξεργαστεί		
		12. Διάβασμα από την Βάση Δεδομένων των Ερωτήσεων της Επόμενης/Προηγούμενης Ερώτησης
		13. Επανάληψη των βημάτων 4 έως 10
	Περίπτωση που δεν υπάρχει Επόμενη/Προηγούμενη Ερώτηση	
		14. Επανεμφάνιση της

		τελευταίας Ερώτησης
	Περίπτωση επιλογής Ερώτησης με τον Αύξοντα Αριθμό	
15. Πληκτρολογεί στο πεδίο "QID" τον επιθυμητό Αύξοντα Αριθμό της ερώτησης που θέλει να επεξεργαστεί		
16. Πατάει το πλήκτρο "Enter" ή το κουμπί "Go To Question"		
		17. Διάβασμα από την Βάση Δεδομένων των Ερωτήσεων της Ερώτησης με τον συγκεκριμένο Αύξοντα Αριθμό
		18. Επανάληψη των βημάτων 4 έως 10
	Περίπτωση που ο Επιλεγμένος Αριθμός είναι μεγαλύτερος από τον τελευταίο έγκυρο Αύξοντα Αριθμό	
		19. Διάβασμα από την Βάση Δεδομένων των Ερωτήσεων της Ερώτησης με τον μεγαλύτερο Αύξοντα Αριθμό
		20. Επανάληψη των βημάτων 4 έως 10
	Περίπτωση που ο Επιλεγμένος Αριθμός είναι μικρότερος από τον Πρώτο έγκυρο Αύξοντα Αριθμό	
		21. Διάβασμα από την Βάση Δεδομένων των Ερωτήσεων της Ερώτησης με τον μικρότερο Αύξοντα Αριθμό
		22. Επανάληψη των βημάτων 4 έως 10
	Περίπτωση Επιλογής Ενδιάμεσου Μη Υπαρκτού Αύξοντος Αριθμού	
		23. Διάβασμα από την Βάση Δεδομένων των Ερωτήσεων της Ερώτησης με τον αμέσως μεγαλύτερο Αύξοντα Αριθμό
		24. Επανάληψη των βημάτων 4 έως 10
25. Τροποποιεί το ή τα πεδία που επιθυμεί από την Ερώτηση		
	Περίπτωση Επιλογής "Cancel"	

		26. Διάβασμα από την Βάση Δεδομένων των Ερωτήσεων της Ερώτησης με τον Τρέχων Αύξοντα Αριθμό
		27. Επανάληψη των βημάτων 4 έως 10
	Περίπτωση Επιλογής "Confirm"	
		28. Διάβασμα από την Σελίδα του Τρέχων Αύξοντος Αριθμού
		29. Διάβασμα από την Σελίδα της Ερώτησης
		30. Διάβασμα από την Σελίδα της Σωστής Απάντησης
		31. Διάβασμα από την Σελίδα της Πρώτης Λάθος Απάντησης
		32. Διάβασμα από την Σελίδα της Δεύτερης Λάθος Απάντησης
		33. Διάβασμα από την Σελίδα της Τρίτης Λάθος Απάντησης
		34. Διάβασμα από την Σελίδα του βαθμού δυσκολίας της Ερώτησης
		35. Αποθήκευση στην Βάση Δεδομένων των Ερωτήσεων Των καινούριων Στοιχείων για την Ερώτηση

Πίνακας 10: Περίπτωση Χρήσης 10: Τροποποίηση Ερώτησης

4.2.11 Περίπτωση Χρήσης 11: Διαγραφή Ερώτησης

Η διάδραση χρήστη/συστήματος στην περίπτωση που ο πρώτος θέλει να διαγράψει κάποια ερώτηση από την Βάση Δεδομένων των Ερωτήσεων της εφαρμογής.

Προαπαιτήσεις :

- Να είναι εγγεγραμμένος χρήστης (4.2.2 Περίπτωση Χρήσης 2: Εγγραφή Χρήστη)
- Να έχει κάνει είσοδο στην εφαρμογή (4.2.1 Περίπτωση Χρήσης 1: Είσοδος Χρήστη)
- Ο χρήστης να είναι Διαχειριστής της εφαρμογής

ΧΡΗΣΤΗΣ	Περίπτώσεις	ΣΥΣΤΗΜΑ
1. Επιλέγει "Delete Question"		
		2. Εμφάνιση σελίδας
		3. Διάβασμα από την Βάση Δεδομένων των Ερωτήσεων της Πρώτης Ερώτησης
		4. Εμφάνιση του Αύξοντα Αριθμού της Ερώτησης
		5. Εμφάνιση του βαθμού Δυσκολίας της Ερώτησης
		6. Εμφάνιση της Ερώτησης

		7. Εμφάνιση της Σωστής Απάντησης
		8. Εμφάνιση της Πρώτης Λάθους Απάντησης
		9. Εμφάνιση της Δεύτερης Λάθους Απάντησης
		10. Εμφάνιση της Τρίτης Λάθους Απάντησης
	Περίπτωση "Next/Previous"	
11. Χρησιμοποιώντας τα διαθέσιμα Κουμπιά (Next/Previous) Πλοηγείται στην ερώτηση που θέλει να διαγράψει		
		12. Διάβασμα από την Βάση Δεδομένων των Ερωτήσεων της Επόμενης/Προηγούμενης Ερώτησης
		13. Επανάληψη των βημάτων 4 έως 10
	Περίπτωση που δεν υπάρχει Επόμενη/Προηγούμενη Ερώτηση	
		14. Επανεμφάνιση της τελευταίας Ερώτησης
	Περίπτωση επιλογής Ερώτησης με τον Αύξοντα Αριθμό	
15. Πληκτρολογεί στο πεδίο "QID" τον επιθυμητό Αύξοντα Αριθμό της Ερώτησης που θέλει να διαγράψει		
16. Πατάει το πλήκτρο "Enter" ή το κουμπί "Go To Question"		
		17. Διάβασμα από την Βάση Δεδομένων των Ερωτήσεων της Ερώτησης με τον συγκεκριμένο Αύξοντα Αριθμό
		18. Επανάληψη των βημάτων 4 έως 10
	Περίπτωση που ο Επιλεγμένος Αριθμός είναι μεγαλύτερος από τον τελευταίο έγκυρο Αύξοντα Αριθμό	
		19. Διάβασμα από την Βάση Δεδομένων των Ερωτήσεων της Ερώτησης με τον μεγαλύτερο Αύξοντα Αριθμό
		20. Επανάληψη των βημάτων 4 έως 10

	Περίπτωση που ο Επιλεγμένος Αριθμός είναι μικρότερος από τον Πρώτο έγκυρο Αύξοντα Αριθμό	
		21. Διάβασμα από την Βάση Δεδομένων των Ερωτήσεων της Ερώτησης με τον μικρότερο Αύξοντα Αριθμό
		22. Επανάληψη των βημάτων 4 έως 10
	Περίπτωση Επιλογής Ενδιάμεσου Μη Υπαρκτού Αύξοντος Αριθμού	
		23. Διάβασμα από την Βάση Δεδομένων των Ερωτήσεων της Ερώτησης με τον αμέσως μεγαλύτερο Αύξοντα Αριθμό
		24. Επανάληψη των βημάτων 4 έως 10
	Περίπτωση Επιλογής "Cancel"	
		25. Επιστροφή του Χρήστη στην Κεντρική Σελίδα της Εφαρμογής
	Περίπτωση Επιλογής "Confirm"	
		26. Διάβασμα από την Βάση Δεδομένων των Ερωτήσεων της Πρώτης Ερώτησης
		27. Επανάληψη των βημάτων 4 έως 10

Πίνακας 11: Περίπτωση Χρήσης 11: Διαγραφή Ερώτησης

4.2.12 Περίπτωση Χρήσης 12: Προσθήκη Διαχειριστή

Η διάδραση χρήστη/συστήματος στην περίπτωση που ο πρώτος θέλει να κάνει κάποιο χρήστη διαχειριστή της εφαρμογής.

Προαπαιτήσεις :

- Να είναι εγγεγραμμένος χρήστης (4.2.2 Περίπτωση Χρήσης 2: Εγγραφή Χρήστη)
- Να έχει κάνει είσοδο στην εφαρμογή (4.2.1 Περίπτωση Χρήσης 1: Είσοδος Χρήστη)
- Ο χρήστης να είναι Διαχειριστής της εφαρμογής

ΧΡΗΣΤΗΣ	Περίπτώσεις	ΣΥΣΤΗΜΑ
1. Επιλέγει "Make Admin"		
		2. Διαβάζει από την Βάση Δεδομένων των Χρηστών όλα τα ονόματα των χρηστών που δεν είναι διαχειριστές
		3. Εμφάνιση σελίδας για την προσθήκη διαχειριστή
		4. Εμφάνιση των ονομάτων σε μορφή "drop down menu"

5. Επιλέγει από την λίστα το επιθυμητό όνομα χρήστη για να γίνει διαχειριστής		
	Ο χρήστης πατάει το κουμπί "Make Administrator"	
		6. Διαβάζει το επιλεγμένο όνομα από την λίστα και το ορίζει σαν διαχειριστή
		7. Επανάληψη των βημάτων 2 έως 4
	Ο χρήστης πατάει το κουμπί "Cancel"	
		8. Επιστροφή του Χρήστη στην Κεντρική Σελίδα της Εφαρμογής

Πίνακας 12: Περίπτωση Χρήσης 12: Προσθήκη Διαχειριστή

4.2.13 Περίπτωση Χρήσης 13: Αφαίρεση Διαχειριστή

Η διάδραση χρήστη/συστήματος στην περίπτωση που ο πρώτος θέλει να διαγράψει κάποιο χρήστη από διαχειριστή της εφαρμογής.

Προαπαιτήσεις :

- Να είναι εγγεγραμμένος χρήστης (4.2.2 Περίπτωση Χρήσης 2: Εγγραφή Χρήστη)
- Να έχει κάνει είσοδο στην εφαρμογή (4.2.1 Περίπτωση Χρήσης 1: Είσοδος Χρήστη)
- Ο χρήστης να είναι Διαχειριστής της εφαρμογής

ΧΡΗΣΤΗΣ	Περίπτώσεις	ΣΥΣΤΗΜΑ
1. Επιλέγει "Delete Admin"		
		2. Διαβάζει από την Βάση Δεδομένων των Χρηστών όλα τα ονόματα των χρηστών που είναι διαχειριστές
		3. Εμφάνιση σελίδας για την προσθήκη διαχειριστή
		4. Εμφάνιση των ονομάτων σε μορφή "drop down menu" με εξαίρεση το όνομα του τρέχοντος χρήστη
5. Επιλέγει από την λίστα το επιθυμητό όνομα χρήστη για να διαγραφεί από διαχειριστής		
	Ο χρήστης πατάει το κουμπί "Delete Administrator!!"	
		6. Διαβάζει το επιλεγμένο όνομα από την λίστα και το διαγράφει από τους διαχειριστές
		7. Επανάληψη των βημάτων 2 έως 4
	Ο χρήστης πατάει το κουμπί "Cancel"	
		8. Επιστροφή του Χρήστη στην Κεντρική Σελίδα της

		Εφαρμογής
--	--	-----------

Πίνακας 13: Περίπτωση Χρήσης 13: Αφαίρεση Διαχειριστή

4.2.14 Περίπτωση Χρήσης 14: Διαγραφή Χρήστη

Η διάδραση χρήστη/συστήματος στην περίπτωση που ο πρώτος θέλει να διαγράψει κάποιο άλλο χρήστη (εκτός του εαυτού του) από την εφαρμογή.

Προαπαιτήσεις :

- Να είναι εγγεγραμμένος χρήστης (4.2.2 Περίπτωση Χρήσης 2: Εγγραφή Χρήστη)
- Να έχει κάνει είσοδο στην εφαρμογή (4.2.1 Περίπτωση Χρήσης 1: Είσοδος Χρήστη)
- Ο χρήστης να είναι Διαχειριστής της εφαρμογής

ΧΡΗΣΤΗΣ	Περίπτώσεις	ΣΥΣΤΗΜΑ
1. Επιλέγει "Delete User"		
		2. Διαβάζει από την Βάση Δεδομένων των Χρηστών όλα τα ονόματα των χρηστών
		3. Εμφάνιση σελίδας για την προσθήκη διαχειριστή
		4. Εμφάνιση των ονομάτων σε μορφή "drop down menu" με εξαίρεση το όνομα του τρέχοντος χρήστη
5. Επιλέγει από την λίστα το επιθυμητό όνομα χρήστη για να διαγραφεί		
	Ο χρήστης πατάει το κουμπί "Delete User!!"	
		6. Διαβάζει το επιλεγμένο όνομα από την λίστα
		7. Διαγράφει όλα στα στατιστικά στοιχεία του επιλεγμένου χρήστη από την Βάση Δεδομένων των Ιστορικών Στοιχείων
		8. Διαγράφει τον επιλεγμένο χρήστη από την Βάση Δεδομένων των Χρηστών
		9. Επανάληψη των βημάτων 2 έως 4
	Ο χρήστης πατάει το κουμπί "Cancel"	
		10. Επιστροφή του Χρήστη στην Κεντρική Σελίδα της Εφαρμογής

Πίνακας 14: Περίπτωση Χρήσης 14: Διαγραφή Χρήστη

4.2.15 Περίπτωση Χρήσης 15: Εκκίνηση του Παιχνιδιού

Η διάδραση χρήστη/συστήματος στην περίπτωση που ο πρώτος θέλει να παίξει το παιχνίδι.

Προαπαιτήσεις :

- Να είναι εγγεγραμμένος χρήστης (4.2.2 Περίπτωση Χρήσης 2: Εγγραφή Χρήστη)
- Να έχει κάνει είσοδο στην εφαρμογή (4.2.1 Περίπτωση Χρήσης 1: Είσοδος Χρήστη)

ΧΡΗΣΤΗΣ	Περιπτώσεις	ΣΥΣΤΗΜΑ
1. Επιλέγει "Play -=The Game=-"		
		2. Διαβάζει 5 τυχαίες ερωτήσεις με της απαντήσεις τους από την Βάση Δεδομένων των Ερωτήσεων με μικρό βαθμό δυσκολίας
		3. Διαβάζει 5 τυχαίες ερωτήσεις με της απαντήσεις τους από την Βάση Δεδομένων των Ερωτήσεων με μεσαίο βαθμό δυσκολίας
		4. Διαβάζει 5 τυχαίες ερωτήσεις με της απαντήσεις τους από την Βάση Δεδομένων των Ερωτήσεων με μεγάλο βαθμό δυσκολίας
		5. Δημιουργεί ένα πίνακα οδηγό για την σειρά που θα εμφανίζονται οι απαντήσεις
		6. Θέτει τον δείκτη τρέχουσας ερώτησης στην πρώτη ερώτηση
		7. Εμφάνιση της σελίδας του παιχνιδιού
		8. Φόρτωση στο κεντρικό πεδίο της τρέχουσας ερώτησης
		9. Φόρτωση στο πρώτο υποπεδίο της απάντησης που υποδεικνύει ο πίνακας οδηγός γι' αυτό το υποπεδίο
		10. επανάληψη του βήματος 9 και για τα υπόλοιπα 3 υποπεδία
	Περίπτωση που ο Χρήστης επιλέξει "Cancel"	
		11. Ακύρωση του τρέχοντος παιχνιδιού
		12. Καθαρισμός της μνήμης του συστήματος
		13. Επιστροφή στην κεντρική σελίδα της εφαρμογής
14. Επιλέγει μια απάντηση		
		15. Ενεργοποιεί το κουμπί "Confirm Answer"
16. Πατάει το κουμπί "Confirm Answer"		
		17. Ελέγχει την ορθότητα της απάντησης
	Περίπτωση που έχει επιλεγεί λάθος απάντηση	

		18. Εμφάνιση ενδιάμεσης ενημερωτικής σελίδας που πληροφορεί τον χρήστη για την πρόοδο του στο παιχνίδι, το τελικό του σκορ και ποια ήταν η σωστή απάντηση
		19. Εμφάνιση επιλογών για να ξαναπαίξει ή να επιστρέψει στην κεντρική σελίδα
	Περίπτωση που έχει επιλεγεί σωστή απάντηση	
		20. Προωθεί τον δείκτη της τρέχουσας ερώτησης στην επόμενη ερώτηση
		21. Επαναλαμβάνει τα βήματα 7 έως 10
	Περίπτωση που έχει επιλεγεί σωστή απάντηση στην τελευταία ερώτηση	
		22. Εμφάνιση συγχαρητήριας σελίδας που ενημερώνει τον χρήστη ότι έχει ολοκληρώσει επιτυχώς το παιχνίδι, του λέει το σκορ του και του δίνει την δυνατότητα να ξαναπαίξει ή να επιστρέψει στην κεντρική σελίδα

Πίνακας 15: Περίπτωση Χρήσης 15: Εκκίνηση του Παιχνιδιού

4.3 Το Διάγραμμα Κλάσεων του Συστήματος

Στην τρέχουσα ενότητα θα παραθέσουμε έναν αριθμό από σχεδιαγράμματα που θα απεικονίζουν τα δομικά στοιχεία της λογικής του συστήματος και τις υπηρεσίες που θα προσφέρει το κάθε ένα από αυτά. Αυτά θα αποτελέσουν το κέλυφος – οδηγό πάνω στο οποίο θα αναπτυχτεί ο κώδικας για όλο το σύστημα. Τα δομικά αυτά στοιχεία θα περιέχουν ρουτίνες διαχωρισμένες με βάση την λειτουργικότητα τους ούτως ώστε να απλοποιούν το έργο της συγγραφής τους και του ελέγχου σωστής λειτουργίας τους. Επίσης διευκολύνουν κάποιον μελλοντικό προγραμματιστή που θα ασχοληθεί με το σύστημα να καταλάβει ευκολότερα τον τρόπο λειτουργίας του και την τροποποίησή του. Ο διαχωρισμός της λειτουργικότητας ενός συστήματος σε κλάσεις είναι βασική αρχή στον προγραμματισμό στην γλώσσα Java.

4.3.1 Η κλάση ServerAccess

Η συγκεκριμένη κλάση θα αποτελέσει τον πυρήνα της επικοινωνίας του συστήματος μας με την Βάση Δεδομένων. Η δημιουργία της είναι απαραίτητη γιατί θα μας παρέχει ότι χρειαζόμαστε για να «μιλήσουμε» με την βάση δεδομένων κάτι που εναλλακτικά θα έπρεπε να γίνετε ξεχωριστά από κάθε μια από τις υπόλοιπες κλάσεις και θα καθιστούσε την συντήρηση και τροποποίηση του σχετικού κώδικα δύσκολη και επίπονη.

ServerAccess
<u>-JDBC_DRIVER: String</u> <u>-SERVER_ADDRESS: String</u> <u>-MYSQL_USER: String</u> <u>-MYSQL_PASS: String</u>
+connectMySQL(): Connection +makeQuery(command: String, con: Connection): ResultSet +makeChange(command: String, con: Connection): Void +closeMySQLConnection(con: Connection): Void

Πίνακας 16: Διάγραμμα της κλάσης ServerAccess

Μεταβλητές και αντικείμενα της κλάσης ServerAccess:

- JDBC_DRIVER : αντικείμενο τύπου «String» όπου καταχωρείται το όνομα του οδηγού για την σύνδεση με τη Βάση Δεδομένων.
- SERVER_ADDRESS : αντικείμενο τύπου «String» όπου καταχωρείται η διεύθυνση και το όνομα της Βάσης Δεδομένων που θα χρησιμοποιήσουμε.?
- MYSQL_USER : αντικείμενο τύπου «String» όπου καταχωρείται το όνομα χρήστη της Βάσης Δεδομένων που έχει δικαιώματα για την τροποποίηση των στοιχείων στην Βάση Δεδομένων μας.
- MYSQL_PASS : αντικείμενο τύπου «String» όπου καταχωρείται ο κωδικός ασφαλείας του προαναφερθέντος χρήστη.

Ρουτίνες της κλάσης ServerAccess:

- connectMySQL : Η ρουτίνα αυτή παίρνει τα στοιχεία που χρειάζεται από τις μεταβλητές της κλάσης και ανοίγει έναν διάλογο επικοινωνίας με την Βάση Δεδομένων τον οποίο και επιστρέφει στην ρουτίνα που την κάλεσε με την μορφή ενός αντικειμένου τύπου «connection».
- makeQuery : Η ρουτίνα αυτή δέχεται σαν όρισμα από την ρουτίνα που την κάλεσε μία μεταβλητή τύπου «String» όπου περιέχετε η εντολή προς τη Βάση Δεδομένων και το αντικείμενο που περιέχει τον διάλογο επικοινωνίας με αυτή. Σκοπός της ρουτίνας είναι η αποστολή εντολών στην Βάση Δεδομένων για το διάβασμα διαφόρων στοιχείων της. Επιστρέφει στην ρουτίνα που την κάλεσε το αντικείμενο που περιέχει τα αποτελέσματα του διαβάσματος.
- makeChange : Η ρουτίνα αυτή δέχεται σαν όρισμα από την ρουτίνα που την κάλεσε μία μεταβλητή τύπου «String» όπου περιέχετε η εντολή προς τη Βάση Δεδομένων και το αντικείμενο που περιέχει τον διάλογο επικοινωνίας με αυτή. Σκοπός της ρουτίνας είναι η αποστολή εντολών στην Βάση Δεδομένων για την προσθήκη ή τροποποίηση ή διαγραφή διαφόρων στοιχείων της.
- closeMySQLConnection : Η ρουτίνα αυτή δέχεται σαν όρισμα από την ρουτίνα που την κάλεσε το αντικείμενο που περιέχει τον διάλογο επικοινωνίας με τη Βάση Δεδομένων. Σκοπός της ρουτίνας είναι το κλείσιμο του διαύλου επικοινωνίας με τη Βάση Δεδομένων.

4.3.2 Η κλάση UserManagement

Η κλάση «UserManagement» συγκεντρώνει όλες τις ρουτίνες που έχουν σαν στόχο την διαχείριση των λογαριασμών των χρηστών. Μέσω αυτής της κλάσης υλοποιούνται και οι ρουτίνες που είναι υπεύθυνες για την είσοδο και έξοδο κάποιου χρήστη από το σύστημα.

UserManagement
-dbtable: String -serverAcc: ServerAccess
+createUser(username: String, password: String): Boolean

```

+loginUser(username: String, password: String): Boolean[]
+logoutUser(username: String): Void
+changeUser(username: String, oldPwd: String, newPwd1: String, newPwd2: String): String
+makeAdmin(username: String): Void
+delAdmin(username: String): Void
+listUsers(selectAdmins: int): String[]
+delAccount(username: String, password: String): Boolean
+delUser(username: String): Void
+genHash(password: String): String

```

Πίνακας 17: Διάγραμμα της κλάσης UserManagement

Μεταβλητές και αντικείμενα της κλάσης UserManagement:

- dbtable : αντικείμενο τύπου «String» το οποίο περιέχει το όνομα της Βάσης Δεδομένων και το όνομα του πίνακα που απευθύνεται η κλάση UserManagement.
- serverAcc : αντικείμενο τύπου «ServerAccess» που ενσωματώνει την κλάση ServerAccess και είναι υπεύθυνο για την επικοινωνία της κλάσης με την Βάση Δεδομένων.

Ρουτίνες της κλάσης UserManagement:

- createUser : Ρουτίνα που είναι υπεύθυνη για την δημιουργία του λογαριασμού ενός καινούριου χρήστη. Από την ρουτίνα που την κάλεσε παίρνει σαν ορίσματα το «όνομα χρήστη» (username) και τον «κωδικό χρήστη» (password) και επιστρέφει αληθής εάν η δημιουργία του χρήστη εκτελέστηκε με επιτυχία ή ψευδής εάν απέτυχε.
- loginUser : ρουτίνα που είναι υπεύθυνη για την είσοδο των χρηστών στο σύστημα. Από την ρουτίνα που την κάλεσε παίρνει σαν ορίσματα το «όνομα χρήστη» (username) και τον «κωδικό χρήστη» (password) και επιστρέφει ένα δυαδικό πίνακα τριών στοιχείων με το πρώτο να δηλώνει εάν η είσοδος του χρήστη ήταν επιτυχημένη, το δεύτερο να δηλώνει εάν πρόκειται για διαχειριστή ή απλό χρήστη και το τρίτο εάν έγινε κάποιο λάθος στην προσπάθεια προσπέλασης της Βάσης Δεδομένων.
- logoutUser : ρουτίνα που εκτελεί την λειτουργία της εξόδου του χρήστη από το σύστημα και δέχεται σαν όρισμα το όνομα του χρήστη που είναι να εξέλθει από το σύστημα.
- changeUser : η ρουτίνα αυτή είναι υπεύθυνη για την αλλαγή του κωδικού ασφαλείας του χρήστη. Δέχεται σαν ορίσματα το «όνομα χρήστη» (username), τον παλιό «κωδικό χρήστη» (oldPwd), τον καινούριο «κωδικό χρήστη» (newPwd1), και την επαλήθευση του καινούριου «κωδικού χρήστη» (newPwd2). Η ρουτίνα επιστρέφει μια συμβολοσειρά όπου αναφέρει το αποτέλεσμα της.
- makeAdmin : σκοπός της ρουτίνας είναι ο ορισμός ενός χρήστη σε διαχειριστή της εφαρμογής. Δέχεται σαν όρισμα το όνομα του χρήστη (username) που είναι να γίνει διαχειριστής.
- delAdmin : σκοπός της ρουτίνας είναι η κατάργηση ενός χρήστη από διαχειριστή της εφαρμογής. Δέχεται σαν όρισμα το όνομα του χρήστη (username) που θα διαγραφεί από διαχειριστής.
- listUsers : ρουτίνα που επιστρέφει τα ονόματα των χρηστών που είναι εγγεγραμμένοι στο σύστημα. Δέχεται σαν όρισμα μια μεταβλητή τύπου «int» (selectAdmins) όπου ανάλογα την τιμή της επιλέγει την κατηγορία χρηστών που θα επιστρέψει. Αν η τιμή της μεταβλητής είναι '0' θα επιστρέψει μόνο τα ονόματα των χρηστών που δεν είναι διαχειριστές, αν είναι '1' θα επιστρέψει μόνο τα ονόματα των χρηστών που είναι διαχειριστές ενώ αν είναι οποιοδήποτε άλλο νούμερο θα επιστρέψει μια λίστα με τα ονόματα όλων των χρηστών. Η ρουτίνα επιστρέφει έναν πίνακα συμβολοσειρών (τύπου «String») με μήκος όσο και το πλήθος των ονομάτων που περιέχει.
- delAccount : ρουτίνα που χρησιμοποιεί ο χρήστης για να διαγράψει τον λογαριασμό του στο σύστημα. Δέχεται σαν όρισμα το όνομα χρήστη (username) και τον κωδικό χρήστη (password) και ελέγχει αν επαληθεύονται από το σύστημα. Αν επαληθεύονται βγάζει τον

χρήστη από το σύστημα και διαγράφει τα στοιχεία του λογαριασμού του ενώ αν δεν επαληθευτούν βγάζει τον χρήστη από το σύστημα και τον ειδοποιεί για την αποτυχία της διαγραφής του.

- delUser : ρουτίνα που χρησιμοποιεί ο διαχειριστής της εφαρμογής για να διαγράψει άλλους χρήστες. Δέχεται σαν όρισμα το όνομα χρήστη (username) που είναι να διαγραφεί και στην συνέχεια διαγράφει τα στοιχεία του λογαριασμού του.
- genHash : ρουτίνα που χρησιμοποιείτε μόνο από το σύστημα για την «κωδικοποίηση» των κωδικών των χρηστών προτού αποθηκευτούν στον πίνακα της βάσης δεδομένων με τα στοιχεία του λογαριασμού τους. Δέχεται σαν όρισμα τον κωδικό του χρήστη από τον οποίο εφαρμόζει τον “CRC32” αλγόριθμο και επιστρέφει το αποτέλεσμα σαν συμβολοσειρά για να αποθηκευτεί στην βάση δεδομένων. Σκοπός της ρουτίνας είναι η απόκρυψη από τον διαχειριστή του συστήματος των κωδικών των χρηστών.

4.3.3 Η κλάση QuestionManagement

Η κλάση «QuestionManagement» συγκεντρώνει όλες τις ρουτίνες που έχουν σαν στόχο την διαχείριση των ερωτήσεων του παιχνιδιού που υλοποιεί το σύστημα. Μέσω αυτής της κλάσης υλοποιούνται οι ρουτίνες που είναι υπεύθυνες για την δημιουργία, τροποποίηση, διαγραφή και ανάγνωση των ερωτήσεων του παιχνιδιού.

QuestionManagement
-dbtable: String -serverAcc: ServerAccess
+createQuestion(question: String, correct_answer: String, alternative1: String, alternative2: String, alternative3: String, level: Int): Void +editQuestion(x: Int, question: String, correct_answer: String, alternative1: String, alternative2: String, alternative3: String, level: Int): Void +readQuestion(x: Int): String[] +readQuestionByLevel(level: String): String[][] +delQuestion(x: Int): Void

Πίνακας 18: Διάγραμμα της κλάσης QuestionManagement

Μεταβλητές και αντικείμενα της κλάσης QuestionManagement:

- dbtable : αντικείμενο τύπου «String» το οποίο περιέχει το όνομα της Βάσης Δεδομένων και το όνομα του πίνακα που απευθύνεται η κλάση QuestionManagement.
- serverAcc : αντικείμενο τύπου «ServerAccess» που ενσωματώνει την κλάση ServerAccess και είναι υπεύθυνο για την επικοινωνία της κλάσης με την Βάση Δεδομένων.

Ρουτίνες της κλάσης QuestionManagement:

- createQuestion : η ρουτίνα αυτή είναι υπεύθυνη για την δημιουργία και την αποθήκευση των ερωτήσεων στην βάση δεδομένων. Δέχεται σαν ορίσματα την ερώτηση, την σωστή απάντηση, τις τρεις λάθος απαντήσεις και το επίπεδο δυσκολίας της ερώτησης. Στην συνέχεια τα εισαγάγει στον πίνακα των ερωτήσεων της βάσης δεδομένων με αύξων αριθμό, κλειδί, που ορίζει αυτόματα το σύστημα διαχείρισης της βάσης δεδομένων που έχουμε επιλέξει.
- editQuestion : η ρουτίνα αυτή είναι υπεύθυνη για την τροποποίηση των ερωτήσεων στην βάση δεδομένων. Δέχεται σαν ορίσματα τον αύξων αριθμό της ερώτησης, την ερώτηση, την σωστή απάντηση, τις τρεις λάθος απαντήσεις και το επίπεδο δυσκολίας της ερώτησης. Στην συνέχεια αντικαθιστά την ήδη υπάρχουσα ερώτηση στην βάση δεδομένων με την ‘καινούρια’ ερώτηση που έχει δεχτεί από τα ορίσματά της.
- readQuestion : ρουτίνα που διαβάζει μια ερώτηση από τον πίνακα ερωτήσεων της βάσης δεδομένων. Δέχεται σαν όρισμα τον αύξων αριθμό της ερώτησης που είναι να διαβαστεί. Επιστρέφει έναν πίνακα συμβολοσειρών με τα στοιχεία της ερώτησης.

- `readQuestionByLevel` : ρουτίνα που διαβάζει όλες τις ερωτήσεις ενός συγκεκριμένου επιπέδου δυσκολίας από τον πίνακα ερωτήσεων της βάσης δεδομένων. Δέχεται σαν όρισμα το επίπεδο δυσκολίας των ερωτήσεων που θέλουμε να πάρουμε. Επιστρέφει έναν πίνακα πινάκων συμβολοσειρών με τα στοιχεία όλων των ερωτήσεων με το συγκεκριμένο επίπεδο δυσκολίας.
- `delQuestion` : ρουτίνα που διαγράφει μια συγκεκριμένη ερώτηση από τον πίνακα ερωτήσεων της βάσης δεδομένων βάσει του αύξων αριθμού της και στην συνέχεια τροποποιεί όλους τους αύξοντες αριθμούς, κλειδιά, ούτως ώστε να είναι συνεχόμενα. Δέχεται σαν όρισμα τον αύξων αριθμό της ερώτησης προς διαγραφή.

4.3.4 Η κλάση `HistoryManagement`

Η κλάση «`HistoryManagement`» συγκεντρώνει όλες τις ρουτίνες που έχουν σαν στόχο την διαχείριση του ιστορικού των χρηστών στο παιχνίδι που υλοποιεί το σύστημα. Μέσω αυτής της κλάσης υλοποιούνται οι ρουτίνες που είναι υπεύθυνες για την ενημέρωση, απεικόνιση και διαγραφή του ιστορικού των χρηστών του παιχνιδιού.

HistoryManagement
-dbtable: String
-serverAcc: ServerAccess
+updateHistory(username: String, DateTime: String, questions_answered: Int, money_gained: String): Void
+showPlayerHistory(username: String): Object[][]
+showAllHistory(): Object[][]
+showHighScores(): Object[][]
+delPlayerHistory(username: String): Void

Πίνακας 19: Διάγραμμα της κλάσης `HistoryManagement`

Μεταβλητές και αντικείμενα της κλάσης `HistoryManagement`:

- `dbtable` : αντικείμενο τύπου «`String`» το οποίο περιέχει το όνομα της Βάσης Δεδομένων και το όνομα του πίνακα που απευθύνεται η κλάση `HistoryManagement`.
- `serverAcc` : αντικείμενο τύπου «`ServerAccess`» που ενσωματώνει την κλάση `ServerAccess` και είναι υπεύθυνο για την επικοινωνία της κλάσης με την Βάση Δεδομένων.

Ρουτίνες της κλάσης `HistoryManagement`:

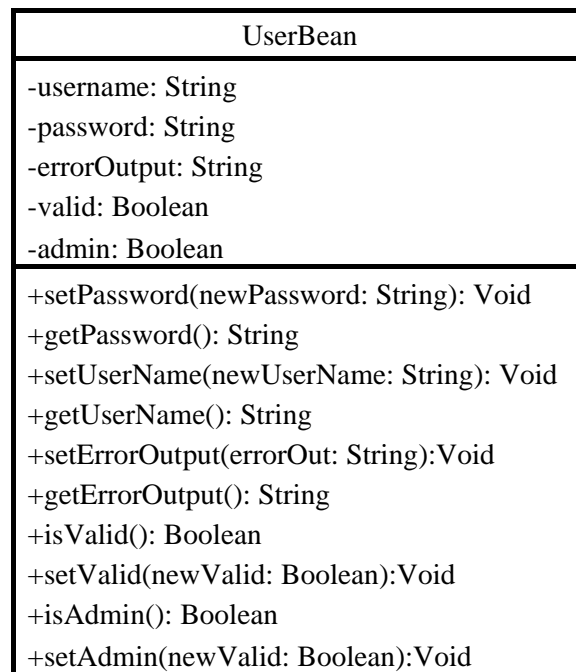
- `updateHistory` : ρουτίνα που σκοπό έχει την προσθήκη καινούριας καταχώρισης στον πίνακα ιστορικού της βάσης δεδομένων με το αποτέλεσμα του παιχνιδιού που μόλις ολοκλήρωσε ο χρήστης. Δέχεται σαν ορίσματα το όνομα του χρήστη, την ώρα και ημερομηνία της καταχώρισης, το πόσες ερωτήσεις απαντήθηκαν με επιτυχία και το ποσό που κέρδισε ο χρήστης. Όταν η ρουτίνα καλείται πρώτα ελέγχει για κενά στους αύξοντες αριθμούς, κλειδιά, του ιστορικού και εάν εντοπίσει κάποιο αποθηκεύει εκεί την καινούρια καταχώριση αλλιώς αφήνει το σύστημα διαχείρισης της βάσης δεδομένων να την τοποθετήσει στο τέλος του πίνακα του ιστορικού της βάσης δεδομένων. Σκοπός του αρχικού ελέγχου είναι η αποφυγή της αδικαιολόγητης αύξησης του αύξοντος αριθμού, κλειδιού, των καταχωρίσεων.
- `showPlayerHistory` : ρουτίνα που επιστρέφει όλο το ιστορικό του χρήστη. Δέχεται σαν όρισμα το όνομα χρήστη και επιστρέφει ένα πίνακα από πίνακες που περιέχουν όλες τις καταχωρίσεις του χρήστη στον πίνακα του ιστορικού της βάσης δεδομένων της εφαρμογής. Η πρώτη καταχώριση είναι η πιο πρόσφατη και οι υπόλοιπες είναι ταξινομημένες με φθίνουσα σειρά.
- `showAllHistory` : ρουτίνα που επιστρέφει το ιστορικό του παιχνιδιού. Διαβάζει από τον πίνακα του ιστορικού της βάσης δεδομένων της εφαρμογής τις 200 πιο πρόσφατες καταχωρίσεις και τις επιστρέφει σαν έναν πίνακα δύο διαστάσεων φθίνουσα ταξινομημένες.
- `showHighScores` : ρουτίνα που επιστρέφει τις μεγαλύτερες βαθμολογίες του παιχνιδιού. Διαβάζει από τον πίνακα του ιστορικού της βάσης δεδομένων της εφαρμογής τις 100

μεγαλύτερες βαθμολογίες και τις επιστρέφει σαν έναν πίνακα δύο διαστάσεων φθίνουσα ταξινομημένες.

- delPlayerHistory : ρουτίνα που διαγράφει τις βαθμολογίες που έχει επιτύχει κάποιος χρήστης στο παιχνίδι. Δέχεται σαν όρισμα το όνομα του χρήστη του οποίου θέλουμε να καθαρίσουμε το ιστορικό. Χρησιμοποιείτε κατά την αφαίρεση κάποιου χρήστη από το σύστημα.

4.3.5 Η κλάση UserBean

Η κλάση «UserBean» χρησιμοποιείτε για την δημιουργία ενός αντικειμένου που περιέχει συγκεντρωμένα τα στοιχεία του χρήστη κατά την διάρκεια που αυτός θα είναι συνδεδεμένος στο σύστημα. Σκοπός της είναι η εύκολη πρόσβαση του συστήματος στα στοιχεία του χρήστη χωρίς να χρειάζεται να διαβάζει συνέχεια την βάση δεδομένων. Το αντικείμενο δημιουργείτε όταν ο χρήστης «ανοίξει» την σελίδα εισόδου στο σύστημα και καταργείται με την έξοδό του από αυτό.



Πίνακας 20: Διάγραμμα της κλάσης UserBean

Μεταβλητές και αντικείμενα της κλάσης UserBean:

- username : αντικείμενο τύπου «String», για την αποθήκευση του ονόματος του χρήστη.
- password : αντικείμενο τύπου «String», για την αποθήκευση του κωδικού ασφαλείας του χρήστη.
- errorOutput : αντικείμενο τύπου «String», όπου αποθηκεύονται τα μηνύματα λάθους της εφαρμογής για την μετέπειτα απεικόνισή τους.
- valid : μεταβλητή τύπου «Boolean», που χρησιμοποιείτε για την ένδειξη εγκυρότητας του χρήστη.
- admin : μεταβλητή τύπου «Boolean», που χρησιμοποιείτε για την ένδειξη ότι ο χρήστης είναι διαχειριστής.

Ρουτίνες της κλάσης UserBean:

- setPassword : ρουτίνα που αποθηκεύει τον κωδικό ασφαλείας του χρήστη. Δέχεται σαν όρισμα τον κωδικό ασφαλείας.
- getPassword : ρουτίνα που επιστρέφει τον κωδικό ασφαλείας του χρήστη.
- setUserName : ρουτίνα που αποθηκεύει το όνομα του χρήστη. Δέχεται σαν όρισμα το όνομα του χρήστη.
- getUserName : ρουτίνα που επιστρέφει το όνομα του χρήστη.

- setErrorOutput : ρουτίνα που αποθηκεύει τυχών μήνυμα λάθους που δημιουργήθηκε κατά την εκτέλεση των διαφόρων ρουτινών του συστήματος.
- getErrorOutput : ρουτίνα που επιστρέφει το μήνυμα λάθους που έχει αποθηκευτεί.
- isValid : ρουτίνα που επιστρέφει το αποτέλεσμα του ελέγχου εγκυρότητας του χρήστη.
- setValid : ρουτίνα που ορίζει την εγκυρότητα του χρήστη.
- isAdmin : ρουτίνα που επιστρέφει την ένδειξη αν ο χρήστης είναι διαχειριστής.
- setAdmin : ρουτίνα που ορίζει την ένδειξη αν ο χρήστης είναι διαχειριστής.

4.3.6 Η κλάση GameBean

Η κλάση «GameBean» χρησιμοποιείτε για την δημιουργία ενός αντικειμένου που περιέχει συγκεντρωμένα τα στοιχεία που αφορούν το παιχνίδι που παίζει ο χρήστης. Το αντικείμενο αυτό δημιουργείται όταν ο χρήστης αρχίζει να παίζει το παιχνίδι και καταργείται όταν ο χρήστης χάσει ή τερματίσει στο παιχνίδι.

GameBean
-gameQuestions: Object[][]
-answersOrder: Object[][]
-currentQuestion: Integer
-moneyTable: Integer[]
+setGameQuestions(x: Object[][]): Void
+getGameQuestions(): Object[][]
+getGameQuestion(x: Integer): Object[]
+setAnswersOrder(x: Object[][]): Void
+getAnswersOrder(): Object[][]
+getAnswerOrder(x: Integer): Object[]
+setCurrentQuestion(x: String): Void
+setCurrentQuestion(x: Integer): Void
+getCurrentQuestion(): String
+getCurrentLevel(): String

Πίνακας 21: Διάγραμμα της κλάσης GameBean

Μεταβλητές και αντικείμενα της κλάσης GameBean:

- gameQuestions : δυσδιάστατος πίνακας αντικειμένων τύπου «Object» όπου αποθηκεύονται οι δεκαπέντε ερωτήσεις που θα κληθεί να απαντήσει ο χρήστης στο παιχνίδι.
- answersOrder : δυσδιάστατος πίνακας αντικειμένων τύπου «Object» όπου αποθηκεύεται η σειρά με την οποία θα εμφανιστούν οι απαντήσεις σε κάθε μία από τις ερωτήσεις που θα κληθεί να απαντήσει ο χρήστης.
- currentQuestion : αντικείμενο τύπου «Integer» όπου αποθηκεύεται το νούμερο της τρέχουσας ερώτησης που βρίσκεται ο χρήστης.
- moneyTable : πίνακας αντικειμένων τύπου «Integer» όπου έχει καταχωρηθεί η βαθμολογία που αντιστοιχεί σε κάθε μία από τις ερωτήσεις.

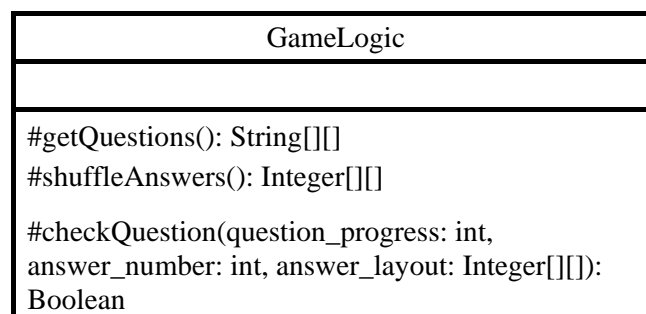
Ρουτίνες της κλάσης GameBean:

- setGameQuestions : ρουτίνα που ορίζει τις ερωτήσεις που θα κληθεί ο χρήστης να απαντήσει στο παιχνίδι. Δέχεται σαν όρισμα τον πίνακα με τις ερωτήσεις
- getGameQuestions : ρουτίνα που επιστρέφει τις ερωτήσεις που θα κληθεί ο χρήστης να απαντήσει στο παιχνίδι.
- getGameQuestion : ρουτίνα που επιστρέφει μια συγκεκριμένη ερώτηση από αυτές που έχουν οριστεί. Δέχεται σαν όρισμα έναν αριθμό από το 0 έως και το 14 και επιστρέφει την αντίστοιχη ερώτηση.

- `setAnswersOrder` : ρουτίνα που ορίζει τον πίνακα που υποδεικνύει την σειρά με την οποία θα εμφανίζονται οι απαντήσεις των ερωτήσεων στον χρήστη. Δέχεται σαν όρισμα τον πίνακα με την σειρά των απαντήσεων.
- `getAnswersOrder` : ρουτίνα που επιστρέφει τον πίνακα που υποδεικνύει την σειρά με την οποία θα εμφανίζονται οι απαντήσεις των ερωτήσεων στον χρήστη.
- `getAnswerOrder` : ρουτίνα που επιστρέφει μια συγκεκριμένη σειρά απαντήσεων από αυτές που έχουν οριστεί. Δέχεται σαν όρισμα έναν αριθμό από το 0 έως και το 14 και επιστρέφει την αντίστοιχη σειρά.
- `setCurrentQuestion` : ρουτίνα που ορίζει την τρέχουσα ερώτηση. Δέχεται σαν όρισμα έναν αριθμό από το 0 έως και το 14 σε μορφή συμβολοσειράς.
- `setQuestion` : ρουτίνα που ορίζει την τρέχουσα ερώτηση. Δέχεται σαν όρισμα έναν αριθμό από το 0 έως και το 14.
- `getCurrentQuestion` : ρουτίνα που επιστρέφει τον αριθμό της τρέχουσας ερώτησης που βρίσκεται ο χρήστης.
- `getCurrentLevel` : ρουτίνα που επιστρέφει την τρέχουσα βαθμολογία του χρήστη.

4.3.7 Η κλάση GameLogic

Η κλάση «GameLogic» συγκεντρώνει όλες τις ρουτίνες που έχουν σαν στόχο την υλοποίηση της λογικής του παιχνιδιού που υλοποιεί το σύστημα. Μέσω αυτής της κλάσης υλοποιούνται οι ρουτίνες που είναι υπεύθυνες για την ανάγνωση των ερωτήσεων, το ανακάτεμα της σειράς των απαντήσεων και τον έλεγχο ορθότητας της κάθε απάντησης του χρήστη.



Πίνακας 22: Διάγραμμα της κλάσης GameLogic

Ρουτίνες της κλάσης GameLogic:

- `getQuestions` : ρουτίνα που επιστρέφει τις απαραίτητες ερωτήσεις από τον πίνακα των ερωτήσεων στην Βάση Δεδομένων. Η ρουτίνα στην αρχή διαβάζει όλες τις ερωτήσεις του χαμηλότερου επιπέδου δυσκολίας από τον πίνακα των ερωτήσεων στην Βάση Δεδομένων. Από αυτές επιλέγει πέντε διαφορετικές τυχαίες ερωτήσεις και τις αποθηκεύει στον πίνακα ερωτήσεων που θα επιστρέψει μετά την εκτέλεσή της. Στην συνέχεια διαβάζει όλες τις ερωτήσεις για το μεσαίο επίπεδο δυσκολίας, επιλέγει άλλες πέντε τυχαίες ερωτήσεις και τις αποθηκεύει στον πίνακα. Ομοίως και για το υψηλό επίπεδο δυσκολίας. Αφού έχει συμπληρώσει τον πίνακα με τις ερωτήσεις που πρέπει να επιστρέψει, τον επιστρέφει στην ρουτίνα που την κάλεσε και τερματίζει την λειτουργία της. Επιστρέφει έναν πίνακα συμβολοσειρών με 15 σειρές και 7 στήλες.
- `shuffleAnswers` : ρουτίνα που ανακατεύει την σειρά που θα εμφανιστούν οι απαντήσεις στον χρήστη. Η ρουτίνα δημιουργεί έναν πίνακα αριθμών 15 σειρών και 4 στηλών και συμπληρώνει την κάθε σειρά με τους αριθμούς 1 έως 4 με τυχαία σειρά. Με την συμπλήρωση του πίνακα τον επιστρέφει στην ρουτίνα που την κάλεσε και τερματίζει την λειτουργία της.
- `checkQuestion` : ρουτίνα που ελέγχει την ορθότητα της απάντησης του χρήστη. Η ρουτίνα δέχεται σαν όρισμα τον πίνακα που ορίζει την σειρά των απαντήσεων, το νούμερο της τρέχουσας ερώτησης και το νούμερο της απάντησης του χρήστη. Στην συνέχεια ελέγχει αν για την συγκεκριμένη ερώτηση η απάντηση του χρήστη αντιστοιχεί στην ορθή απάντηση και επιστρέφει αληθείς ή ψευδής αναλόγως το αποτέλεσμα.

4.4 Ο Σχεδιασμός της Βάσης Δεδομένων

Η Βάση Δεδομένων μας δημιουργείτε με την εντολή στο σύστημα διαχείρισης της βάσης δεδομένων «CREATE DATABASE WhoWantsToBeAMillionaire;». Η οποία δημιουργεί μια κενή βάση δεδομένων με όνομα ‘WhoWantsToBeAMillionaire’.

Στις παραπάνω ενότητες έχουμε ήδη αναφέρει τους πίνακες που απαρτίζουν την Βάση Δεδομένων μας και σε κάποια από τα στοιχεία τους αλλά στην παρούσα ενότητα θα τους ορίσουμε με όλες τους τις λεπτομέρειες.

Πριν όμως προχωρήσουμε στον ορισμό των πινάκων της Βάσης Δεδομένων μας θα πρέπει να αναφερθούμε σε κάποιους από τους τύπους δεδομένων που μπορεί να περιέχει το κάθε πεδίο των πινάκων μιας βάσης δεδομένων στην MySQL.

- Πεδίο τύπου BIT. Το μέγεθος του πεδίου μπορεί να είναι από 1 έως 64 bit.
- Πεδίο τύπου TINYINT. Είναι ένας πολύ μικρός ακέραιος αριθμός. Το εύρος τιμών που μπορεί να πάρει είναι από το -128 έως το 127 ή από το 0 έως το 255.
- Πεδίο τύπου SMALLINT. Είναι ένας μικρός ακέραιος αριθμός. Το εύρος τιμών που μπορεί να λάβει είναι από το -32,768 έως το 32,767 ή από το 0 έως το 65,535.
- Πεδίο τύπου MEDIUMINT. Είναι ένας μεσαίος ακέραιος αριθμός. Το εύρος τιμών που μπορεί να λάβει είναι από το -8,388,608 έως το 8,388,607 ή από το 0 έως το 16,777,215.
- Πεδίο τύπου INT. Είναι ένας ακέραιος αριθμός. Το εύρος τιμών που μπορεί να λάβει είναι από το -2,147,483,648 έως το 2,147,483,647 ή από το 0 έως το 4,294,967,295.
- Πεδίο τύπου BIGINT. Είναι ένας μεγάλος ακέραιος αριθμός. Το εύρος τιμών που μπορεί να λάβει είναι από το -9,223,372,036,854,775,808 έως το 9,223,372,036,854,775,807 ή από το 0 έως το 18,446,744,073,709,551,615.
- Πεδίο τύπου DATE. Είναι μια ημερομηνία. Το εύρος των ημερομηνιών που υποστηρίζει είναι από το ‘1000-01-01’ έως το ‘9999-12-31’. Η MySQL απεικονίζει τις ημερομηνίες στην μορφοποίηση ‘Έτος – Μήνας – Μέρα’ (YYYY-MM-DD).
- Πεδίο τύπου DATETIME. Είναι συνδυασμός ημερομηνίας και ώρας. Το εύρος τιμών που υποστηρίζει είναι από το ‘1000-01-01 00:00:00’ έως το ‘9999-12-31 23:59:59’. Η MySQL απεικονίζει αυτό τον τύπο με την μορφοποίηση ‘Έτος – Μήνας – Μέρα Ώρα : Λεπτά : Δευτερόλεπτα’ (YYYY-MM-DD HH:MM:SS).
- Πεδίο τύπου CHAR. Είναι μια συμβολοσειρά σταθερού μήκους από 0 έως 255 χαρακτήρες. Κατά την αποθήκευση συμπληρώνετε από αριστερά με τον χαρακτήρα του κενού μέχρι το μήκος που έχει οριστεί.
- Πεδίο τύπου VARCHAR. Είναι μια συμβολοσειρά μεταβλητού μήκους, με μέγιστο μήκος έως 65,535 χαρακτήρες. Τα πεδία αυτού του τύπου διαφέρουν από τα πεδία τύπου ‘CHAR’ στο γεγονός ότι ενώ ένα πεδίο τύπου CHAR[20] θα είναι πάντα είκοσι χαρακτήρες σε μήκος, ένα πεδίο VARCHAR[20] θα είναι έως είκοσι χαρακτήρες σε μήκος με αποτέλεσμα να χρειάζεται λιγότερο χώρο για να αποθηκευτεί.

4.4.1 Ο πίνακας των Χρηστών

Στον πίνακα των χρηστών αποθηκεύονται οι πληροφορίες που απαιτεί το σύστημα για να αναγνωρίσει κάποιο χρήστη και της ιδιότητες του. Ο πίνακας αποτελείται από τρία πεδία:

- Το πεδίο του ονόματος του χρήστη, που έχει μέγεθος 50 στοιχείων, είναι τύπου ‘VarChar’ και έχει όνομα ‘Pusername’. Επίσης αυτό είναι το πεδίο κλειδί του πίνακα αναγκάζοντας έτσι το κάθε όνομα χρήστη να είναι διαφορετικό.
- Το πεδίο του κωδικού του χρήστη, με μέγεθος 50 στοιχείων, τύπου ‘VarChar’ και ονόματος ‘Ppwd’.
- Το πεδίο αναγνώρισης διαχειριστή, με μέγεθος 1 στοιχείου, τύπου ‘bit’ και ονόματος ‘PisAdmin’.

Κανένα από τα πεδία του πίνακα δεν μπορεί να αφεθεί κενό.

player		
Όνομα Πεδίου	Τύπος Πεδίου	Μέγεθος Πεδίου

PuserName	VARCHAR	50
Ppwd	VARCHAR	50
PisAdmin	BIT	1

Πίνακας 23: Ο πίνακας των Χρηστών

Η εντολή που χρησιμοποιήσαμε για την δημιουργία του πίνακα στην βάση δεδομένων είναι:

```
CREATE TABLE player (
Pusername VARCHAR(50) NOT NULL,
Ppwd VARCHAR(50) NOT NULL,
PisAdmin BIT(1) NOT NULL,
PRIMARY KEY (Pusername)
);
```

4.4.2 Ο πίνακας του Ιστορικού

Στον πίνακα του Ιστορικού αποθηκεύονται οι πληροφορίες που είναι σχετικές με την πορεία των χρηστών στο παιχνίδι. Δηλαδή ποιος έπαιξε, πότε έπαιξε, πόσες ερωτήσεις απάντησε και πόση ήταν η βαθμολογία του. Ο πίνακας αποτελείται από τα ακόλουθα πέντε πεδία:

- Το πεδίο αύξοντος αριθμού των καταχωρίσεων του πίνακα, με μέγεθος 20 στοιχείων, τύπου 'BIGINT' και με όνομα 'id'. Αυτό το πεδίο είναι το πεδίο κλειδί του πίνακα.
- Το πεδίο του ονόματος του χρήστη, που έχει μέγεθος 50 στοιχείων, είναι τύπου 'VarChar' και έχει όνομα 'Pusername'.
- Το πεδίο που καταχωρεί την ημερομηνία και ώρα που ο χρήστης τελείωσε το παιχνίδι. Το πεδίο έχει μέγεθος 19 στοιχεία τύπου 'char' και έχει όνομα 'PlayedAt'. Η ώρα και η ημερομηνία παίρνονται από τον διακομιστή την εφαρμογής και όχι από το σύστημα του χρήστη.
- Το πεδίο που καταχωρεί το πλήθος των ερωτήσεων που απάντησε ο χρήστης. Το πεδίο έχει μέγεθος 5 στοιχείων τύπου 'Int' και έχει όνομα 'qstAnswd'.
- Το πεδίο που καταχωρεί την βαθμολογία του χρήστη. Έχει μέγεθος 9 στοιχείων τύπου 'MediumInt' και όνομα 'moneyGained'.

Κανένα από τα πεδία του πίνακα δεν μπορεί να αφεθεί κενό.

history		
Όνομα Πεδίου	Τύπος Πεδίου	Μέγεθος Πεδίου
id	BIGINT	20
PuserName	VARCHAR	50
PlayedAt	CHAR	19
qstAnswd	INT	5
moneyGained	MEDIUMINT	9

Πίνακας 24: Ο πίνακας του Ιστορικού

Η εντολή που χρησιμοποιήσαμε για την δημιουργία του πίνακα στην βάση δεδομένων είναι:

```
CREATE TABLE history (
id BIGINT(20) NOT NULL AUTO_INCREMENT,
Pusername VARCHAR(50) NOT NULL,
PlayedAt CHAR(19) NOT NULL,
qstAnswd INT(5) NOT NULL,
moneyGained MEDIUMINT(9) NOT NULL,
PRIMARY KEY (id)
)
AUTO_INCREMENT=1;
```

4.4.3 Ο πίνακας των Ερωτήσεων

Στον πίνακα των Ερωτήσεων αποθηκεύονται οι ερωτήσεις μαζί με τις απαντήσεις τους, τον βαθμό δυσκολίας του και τον αύξων αριθμό της εκάστοτε ερώτησης. Ο πίνακας αποτελείται από τα ακόλουθα επτά πεδία:

- Το πεδίο αύξοντος αριθμού των καταχωρίσεων του πίνακα, με μέγεθος 6 στοιχείων, τύπου 'SmallInt' και με όνομα 'qid'. Αυτό το πεδίο είναι το πεδίο κλειδί του πίνακα.
- Το πεδίο που καταχωρεί την ερώτηση. Το πεδίο έχει μέγεθος 1000 στοιχείων τύπου 'VarChar' και έχει όνομα 'qst'.
- Το πεδίο που καταχωρεί την σωστή απάντηση της ερώτησης. Το πεδίο έχει μέγεθος 500 στοιχείων τύπου 'VarChar' και έχει όνομα 'CorAns'.
- Το πεδίο που καταχωρεί την πρώτη εναλλακτική απάντηση της ερώτησης. Το πεδίο έχει μέγεθος 500 στοιχείων τύπου 'VarChar' και έχει όνομα 'Ans1'.
- Το πεδίο που καταχωρεί την δεύτερη εναλλακτική απάντηση της ερώτησης. Το πεδίο έχει μέγεθος 500 στοιχείων τύπου 'VarChar' και έχει όνομα 'Ans2'.
- Το πεδίο που καταχωρεί την τρίτη εναλλακτική απάντηση της ερώτησης. Το πεδίο έχει μέγεθος 500 στοιχείων τύπου 'VarChar' και έχει όνομα 'Ans3'.
- Το πεδίο που καταχωρεί το επίπεδο δυσκολίας της ερώτησης. Το πεδίο έχει μέγεθος 6 στοιχείων τύπου 'VarChar' και έχει όνομα 'dlevel'.

Κανένα από τα πεδία του πίνακα δεν μπορεί να αφεθεί κενό.

questions		
Όνομα Πεδίου	Τύπος Πεδίου	Μέγεθος Πεδίου
qid	SMALLINT	6
qst	VARCHAR	1000
CorAns	VARCHAR	500
Ans1	VARCHAR	500
Ans2	VARCHAR	500
Ans3	VARCHAR	500
dlevel	VARCHAR	6

Πίνακας 25: Ο πίνακας των Ερωτήσεων

Η εντολή που χρησιμοποιήσαμε για την δημιουργία του πίνακα στην βάση δεδομένων είναι:

```
CREATE TABLE question (  
qid SMALLINT(6) NOT NULL AUTO_INCREMENT,  
qst VARCHAR(1000) NOT NULL,  
CorAns VARCHAR(500) NOT NULL,  
ans1 VARCHAR(500) NOT NULL,  
ans2 VARCHAR(500) NOT NULL,  
ans3 VARCHAR(500) NOT NULL,  
dlevel VARCHAR(6) NOT NULL,  
PRIMARY KEY (qid)  
)  
AUTO_INCREMENT=1;
```

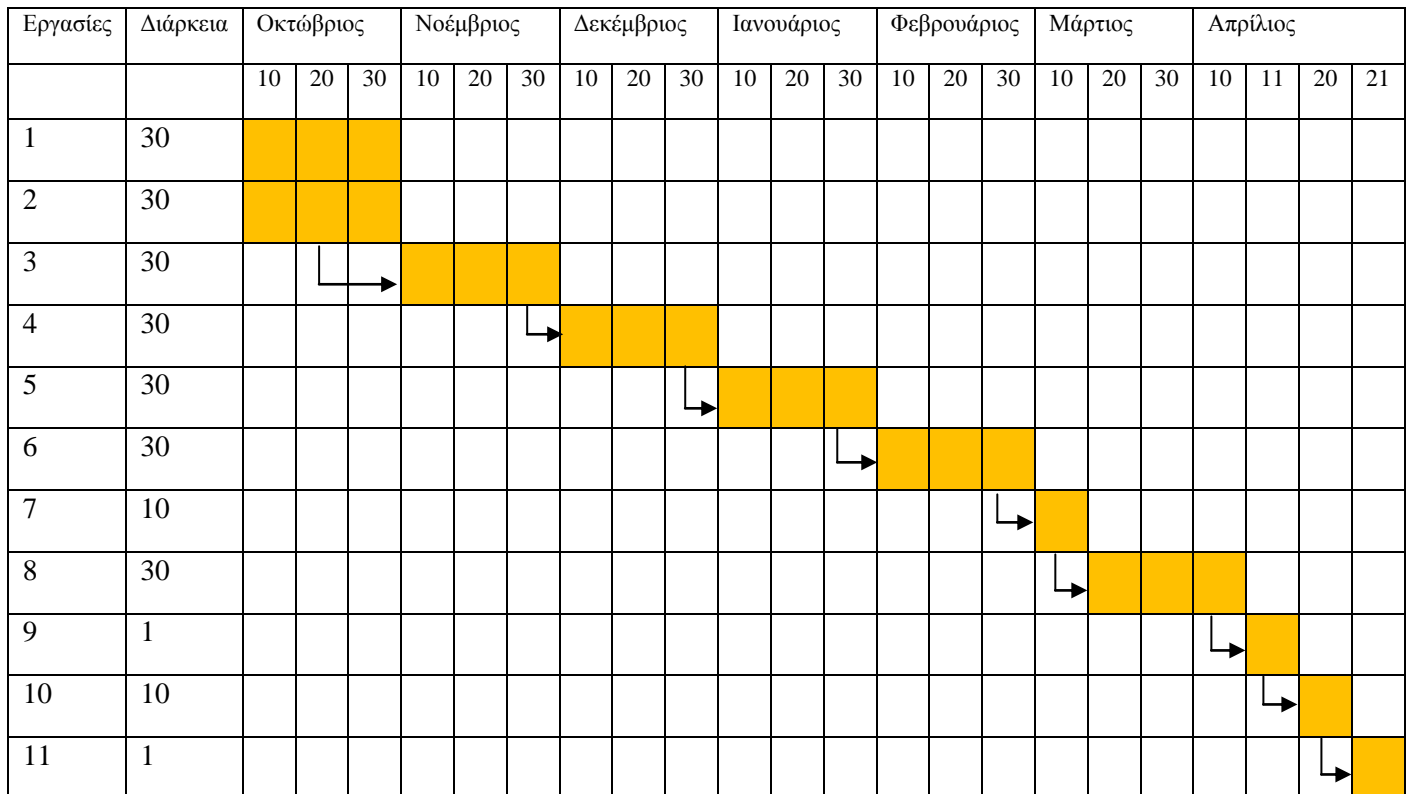
4.5 Σημαντικοί Στόχοι για την Ολοκλήρωση της Πτυχιακής

A/A	Στόχοι	Ημέρες
1	Ολοκλήρωση της έρευνας State of the Art	30
2	Ολοκλήρωση της ανάλυσης του προβλήματος	30
3	Ολοκλήρωση του σχεδιασμού ανάπτυξης της πτυχιακής	30
4	Υλοποίηση του 1 ^{ου} πιλότου του τεχνικού μέρους της πτυχιακής	30
5	Υλοποίηση του 2 ^{ου} πιλότου του τεχνικού μέρους της πτυχιακής	30

6	Ολοκλήρωση της υλοποίησης του τεχνικού μέρους της πτυχιακής εργασίας	30
7	Έλεγχος Λειτουργίας τεχνικού μέρους	10
8	Συγγραφή Αναφοράς Εργασίας	30
9	Υποβολή Αίτησης Αξιολόγησης Εργασίας	1
10	Προετοιμασία Παρουσίασης Αναφοράς	10
11	Παρουσίαση Αναφοράς	1

Πίνακας 26: Στόχοι για την Ολοκλήρωση της Πτυχιακής

4.6 Χρονοδιάγραμμα Gantt



Πίνακας 27: Χρονοδιάγραμμα Gantt

5. Κύριο Μέρος Πτυχιακής

5.1 Ανάλυση Προβλήματος

Στο στάδιο της υλοποίησης συνεχίζουμε την ανάπτυξη του συστήματος όπου και θα το τροποποιήσουμε για να λειτουργεί αρμονικά με το υπόλοιπο σύστημα που θα υλοποιηθεί σε αυτό το στάδιο. Συγκεκριμένα σε αυτό το στάδιο θα υλοποιηθεί η διεπαφή του συστήματός μας με τον χρήστη και θα γίνουν οι απαραίτητες τροποποιήσεις και στην διεπαφή και στην ήδη υπάρχουσα υποδομή του συστήματος ώστε να μπορούν να συνεργαστούν αρμονικά μεταξύ τους αυτά τα δύο κομμάτια.

Ο λόγος που διαλέξαμε αυτή την τακτική είναι ούτως ώστε να μπορούμε να ασχοληθούμε με μόνο ένα αντικείμενο κάθε φορά και να μην έχουμε να επιλύσουμε περισσότερα από έναν τύπο προβλήματος σε κάθε βήμα της ανάπτυξης του συστήματος.

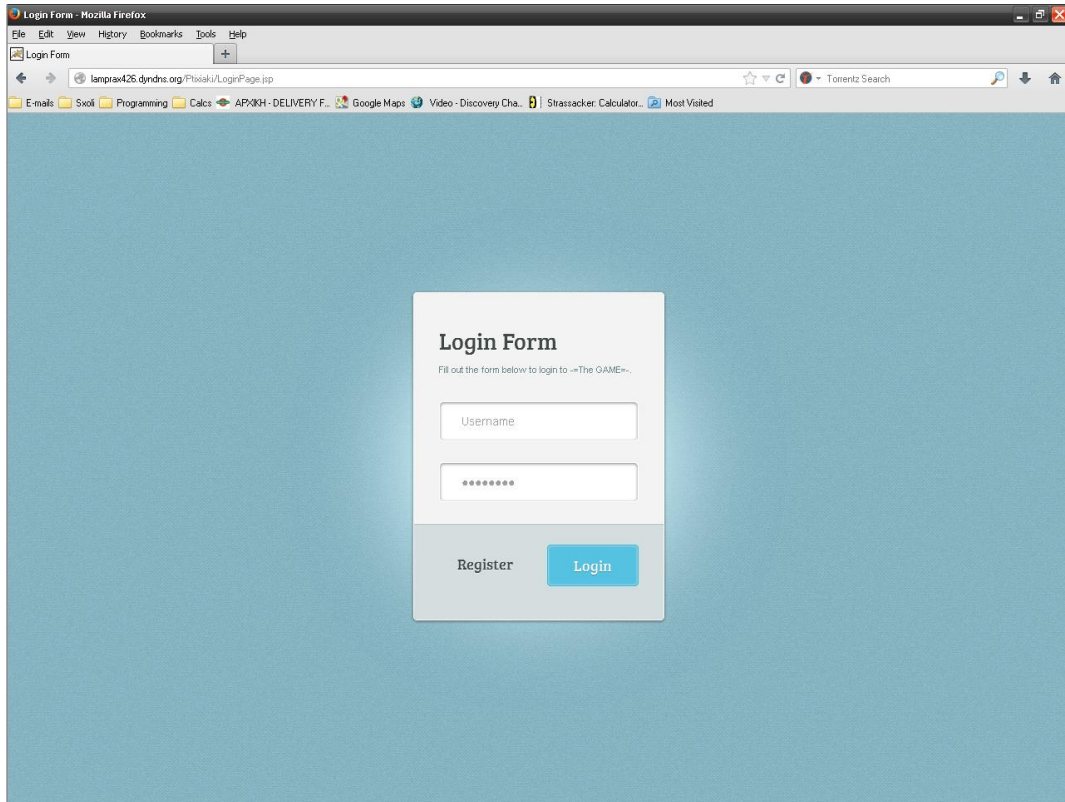
5.1.1 Απαιτήσεις Συστήματος

Πιο συγκεκριμένα για τις ανάγκες υλοποίησης της εφαρμογής χρησιμοποιήθηκε ένας server και μια βάση δεδομένων ώστε να καταγράφονται οι χρήστες και οι ερωτήσεις του παιχνιδιού. Από κατασκευαστικής άποψης χρησιμοποιήθηκε server-side scripting. Καθώς η εφαρμογή είναι απομακρυσμένη και δεν εγκαθίσταται από το χρήστη απαιτείται υψηλή ταχύτητα ανταπόκρισης. Είναι μια εφαρμογή server-side, η οποία δέχεται τα δεδομένα του χρήστη, τα επεξεργάζεται και του επιστρέφει τα τελικά αποτελέσματα. Αναλυτικά: η εφαρμογή δέχεται την απάντηση που επέλεξε ο χρήστης, ελέγχει εάν είναι η σωστή απάντηση για την συγκεκριμένη ερώτηση και επιστρέφει το τελικό αποτέλεσμα. Οι γλώσσες που επιλέχθηκαν ήταν οι: JSP, Java και MySQL.

5.2 Σχεδιασμός Υλοποίησης

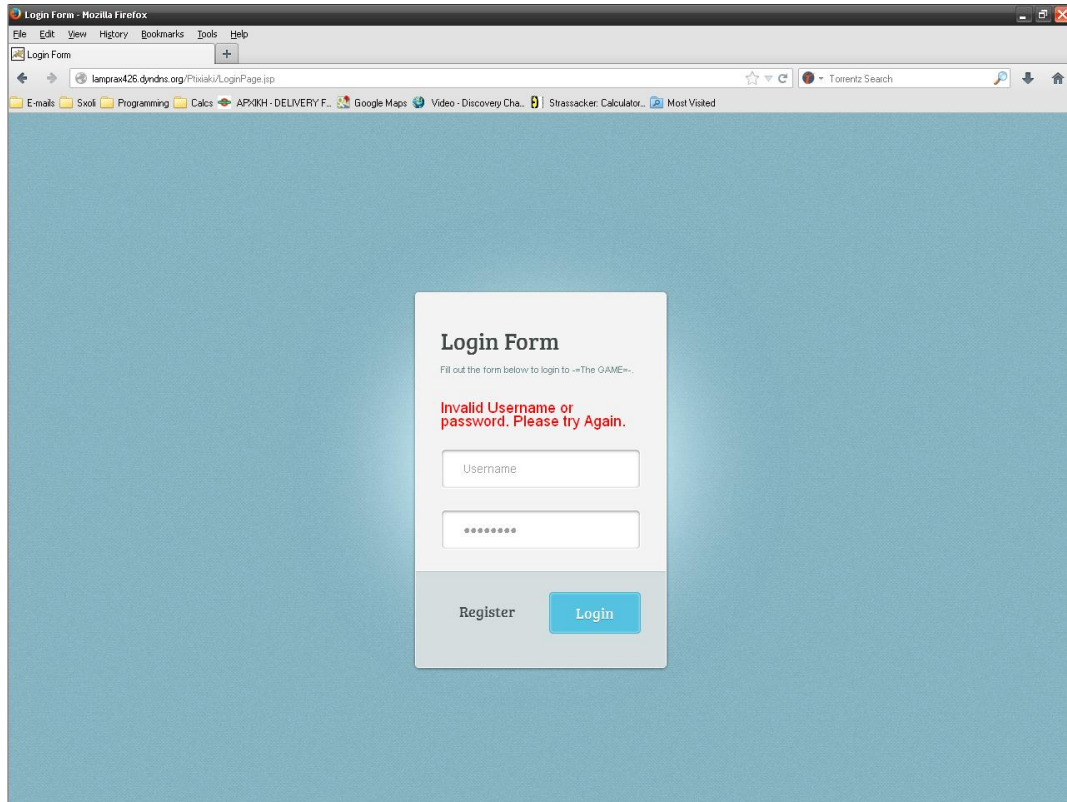
5.2.1 Η φόρμα εισαγωγής του χρήστη

Όταν ο χρήστης θελήσει να μπει στην εφαρμογή θα χρειαστεί να εισαγάγει στην γραμμή διευθύνσεων του περιηγητή διαδικτύου της προτίμησης του την διεύθυνση της εφαρμογής (στο παράδειγμα παρακάτω χρησιμοποιήθηκε η διεύθυνση <http://lamprax426.dyndns.org/Ptixiaki/>) και το σύστημα θα του εμφανίσει την ακόλουθη φόρμα.



Εικόνα 1: Η φόρμα εισαγωγής του χρήστη

Στην παραπάνω φόρμα ο χρήστης θα χρειαστεί να εισαγάγει στα αντίστοιχα πεδία το όνομα χρήστη του, τον κωδικό του και στην συνέχεια να πατήσει το πλήκτρο 'Enter' ή να πατήσει με το ποντίκι του το κουμπί 'Login' που βρίσκετε στην κάτω δεξιά πλευρά της φόρμας εισαγωγής χρήστη. Στην περίπτωση που ο χρήστης εισαγάγει λάθος όνομα χρήστη ή λάθος κωδικό το σύστημα θα του εμφανίσει την ακόλουθη οθόνη.

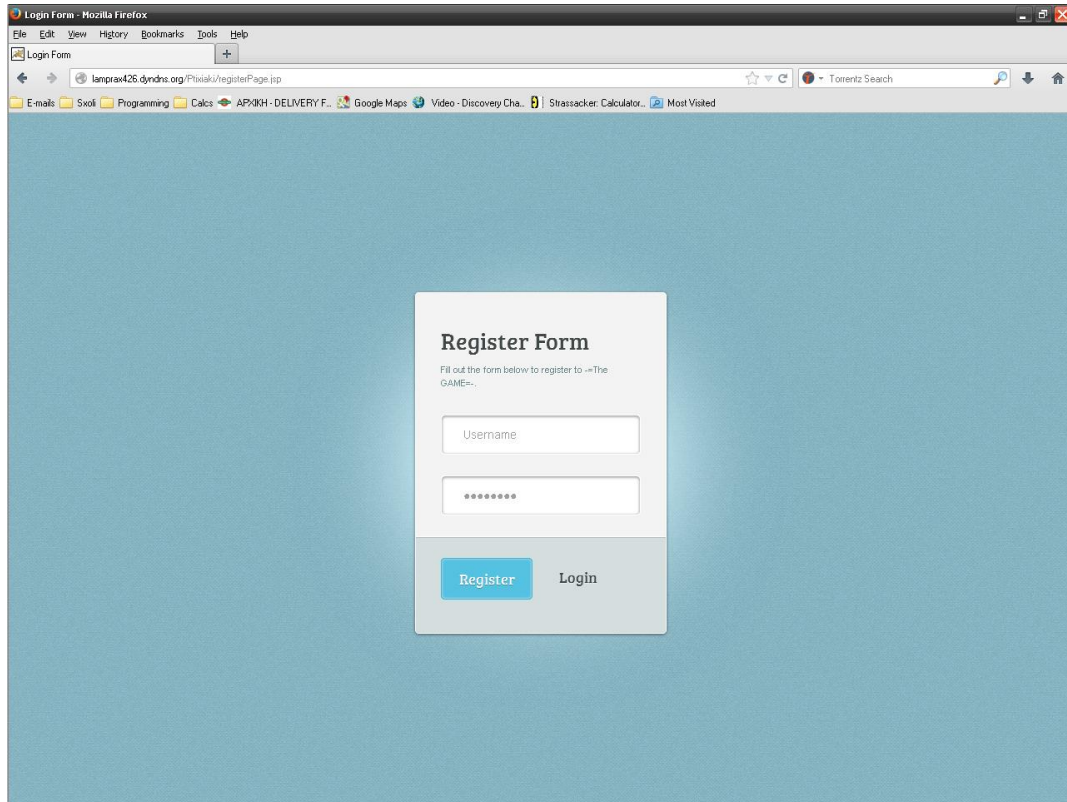


Εικόνα 2: Εισαγωγή λάθος ονόματος χρήστη ή λάθος κωδικού

Ο σκοπός του κοινού μηνύματος και για τις δύο περιπτώσεις είναι να μην μπορεί κάποιος με κακόβουλες προθέσεις να εντοπίσει το όνομα χρήστη κάποιου άλλου και να εισέλθει στην εφαρμογή με ψευδή στοιχεία.

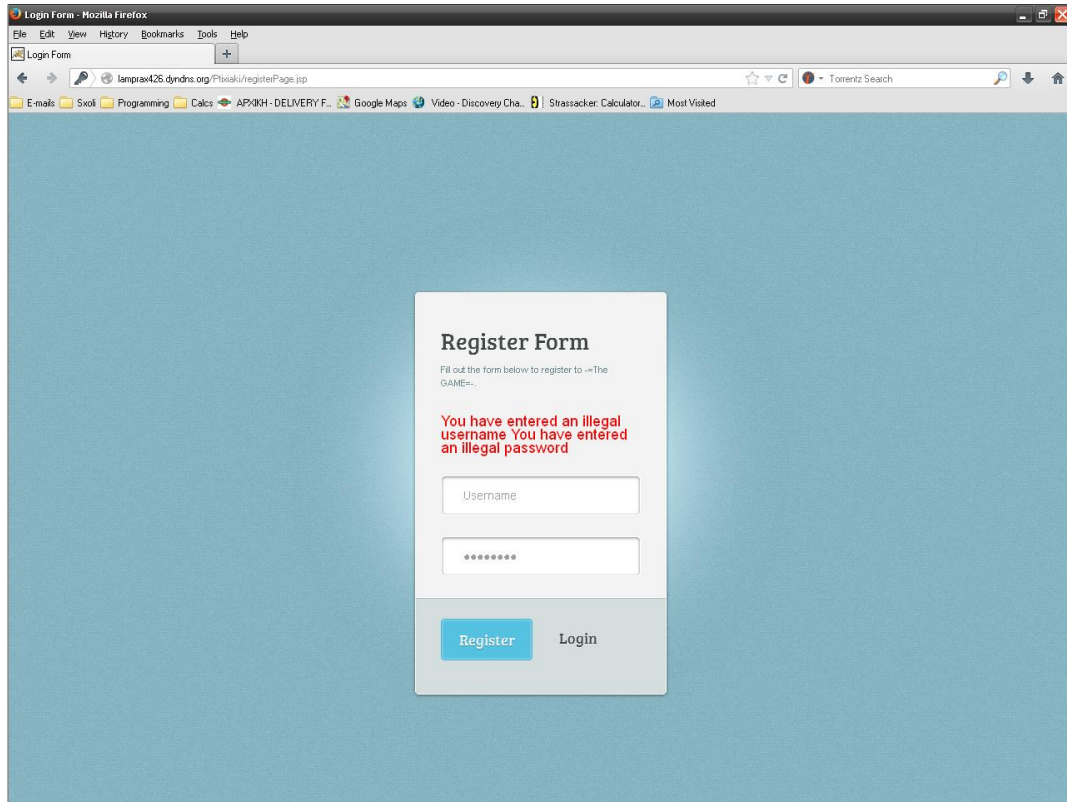
5.2.2 Η φόρμα εγγραφής χρήστη στο σύστημα

Όταν ο χρήστης θελήσει να εγγραφεί στην εφαρμογή θα χρειαστεί να εισαγάγει στην γραμμή διευθύνσεων του περιηγητή διαδικτύου της προτίμησής του την διεύθυνση της εφαρμογής (στο παράδειγμα παρακάτω χρησιμοποιήθηκε η διεύθυνση <http://lamprax426.dyndns.org/Ptixiaki/>) και από την φόρμα εισόδου στο σύστημα (5.2.1 Η φόρμα εισαγωγής χρήστη στο σύστημα) να επιλέξει με το ποντίκι του την επιλογή 'Register', που βρίσκεται στο κάτω αριστερό άκρο της φόρμας, ώστε να του εμφανιστεί η φόρμα εγγραφής καινούργιου χρήστη.



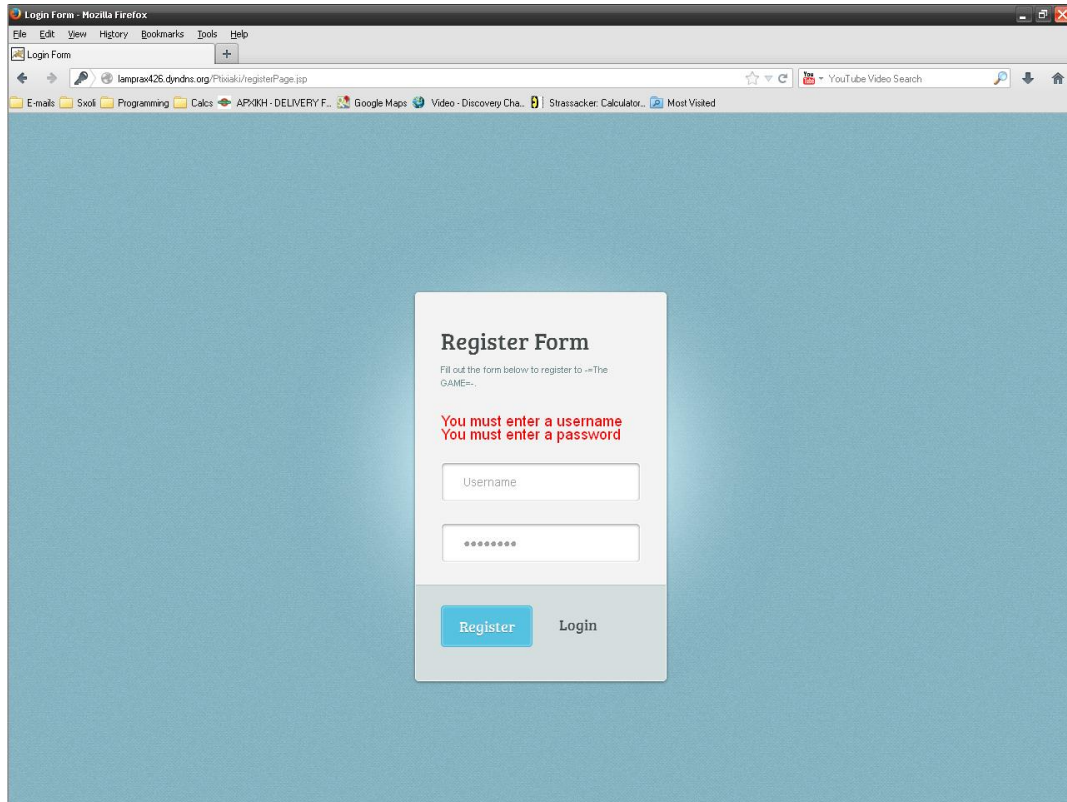
Εικόνα 3: Η φόρμα εγγραφής χρήστη

Στην φόρμα εγγραφής θα χρειαστεί να συμπληρώσει το όνομα χρήστη και κωδικό της επιλογής του και στην συνέχεια να πατήσει το πλήκτρο 'Enter' ή να πατήσει με το ποντίκι του το κουμπί 'Register' που βρίσκετε στην κάτω αριστερή πλευρά της φόρμας εγγραφής χρήστη. Εν συνεχεία το σύστημα θα ελέγξει τα στοιχεία που συμπλήρωσε ο χρήστης για την ορθότητά τους, τα πεδία για το όνομα και τον κωδικό του χρήστη δεν γίνεται να είναι κενά και δεν μπορούν να περιέχουν κάποιες λέξεις κλειδιά που θα μπορούσαν να προκαλέσουν βλάβες στο σύστημα (όπως χιουμοριστικά παρουσιάζεται στο δικτυακό κόμικς 'xkcd' στην σελίδα: '<http://xkcd.com/327/>'). Στην περίπτωση που τα στοιχεία δεν περάσουν από αυτό τον έλεγχο εμφανίζεται μήνυμα λάθους στον χρήστη που τον πληροφορεί για το είδος του σφάλματος.



Εικόνα 4: Εισαγωγή απαγορευμένου ονόματος χρήστη ή κωδικού

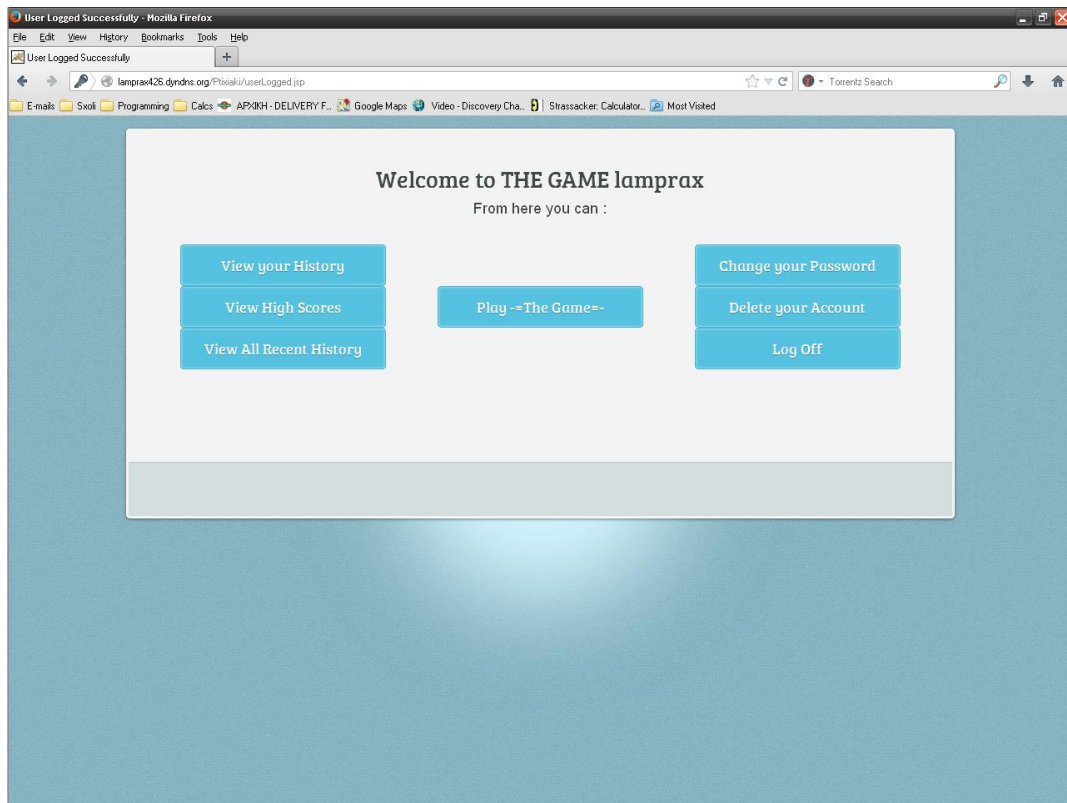
Μετά από τον έλεγχο νομιμότητας των στοιχείων που έχει συμπληρώσει ο χρήστης, το σύστημα ελέγχει στην βάση δεδομένων των χρηστών για να επιβεβαιώσει ότι το όνομα χρήστη που έχει επιλεγεί είναι διαθέσιμο. Εφ' όσον η διαθεσιμότητα του ονόματος επαληθευτεί το σύστημα προχωρεί στην εγγραφή του καινούργιου χρήστη και τον παραπέμπει στην κεντρική οθόνη της εφαρμογής. Σε αντίθετη περίπτωση το σύστημα θα εμφανίσει στον υποψήφιο χρήστη ότι το όνομα που έχει επιλέξει είναι ήδη καταχωρημένο σε άλλον και θα τον προτρέψει να εισαγάγει ένα διαφορετικό όνομα χρήστη.



Εικόνα 5: Εισαγωγή κενού ονόματος χρήστη και κωδικού

5.2.3 Η κεντρική οθόνη του χρήστη

Μετά την επιτυχή εισαγωγή ή την εγγραφή καινούριου χρήστη (υποκεφάλαια [5.2.1](#) και [5.2.2](#) αντίστοιχα) το σύστημα θα προωθήσει τον χρήστη στην κεντρική οθόνη της εφαρμογής όπου το σύστημα τον υποδέχεται και του παρουσιάζονται οι δυνατότητες που έχει στην επιλογή του.



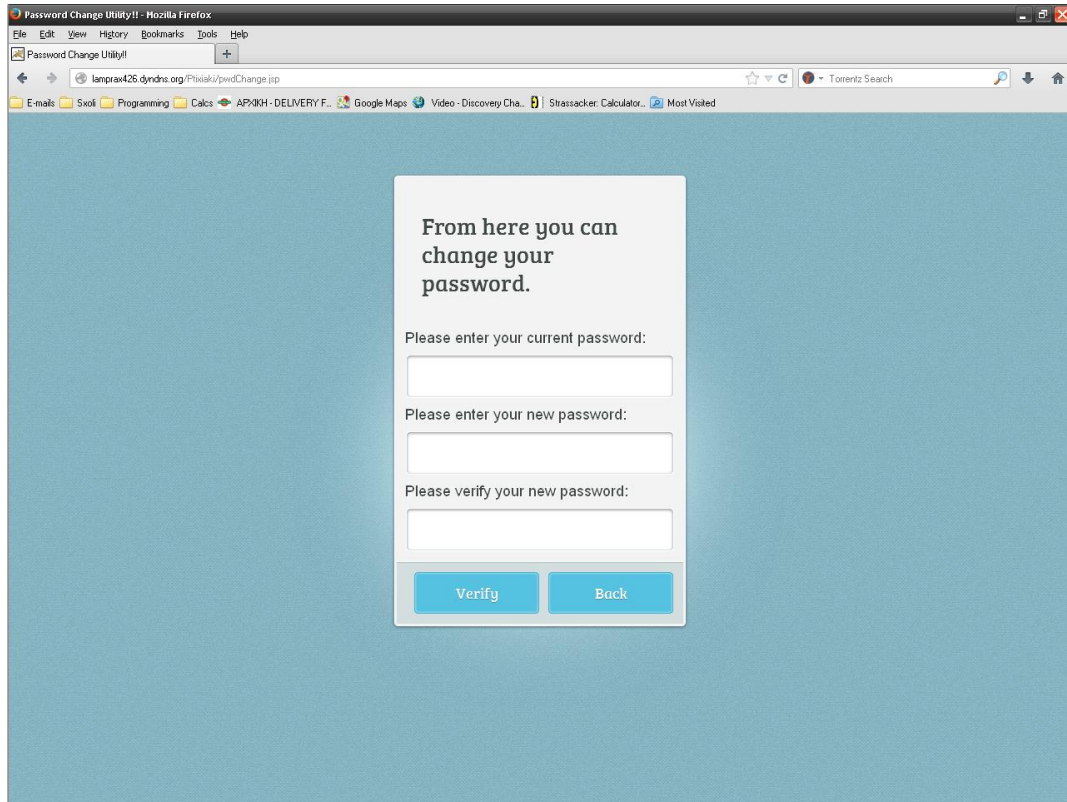
Εικόνα 6: Η κεντρική οθόνη του χρήστη

5.2.4 Έξοδος χρήστη από το σύστημα

Στην περίπτωση που ο χρήστης επιθυμήσει την έξοδό του από το σύστημα θα χρειαστεί να επιλέξει την επιλογή 'Log Off' από την κεντρική οθόνη της εφαρμογής και το σύστημα θα εκτελέσει τις απαιτούμενες ενέργειες. Όταν ολοκληρωθεί η διαδικασία το σύστημα θα επιστρέψει τον χρήστη στην φόρμα εισαγωγής χρήστη (5.2.1). Σαν δικλείδα ασφαλείας το σύστημα θα αποβάλλει αυτόματα κάποιον χρήστη που έχει παραμείνει σε αδράνεια για περισσότερο από ένα προκαθορισμένο χρονικό διάστημα και θα τον επιστρέψει στην φόρμα εισαγωγής χρήστη.

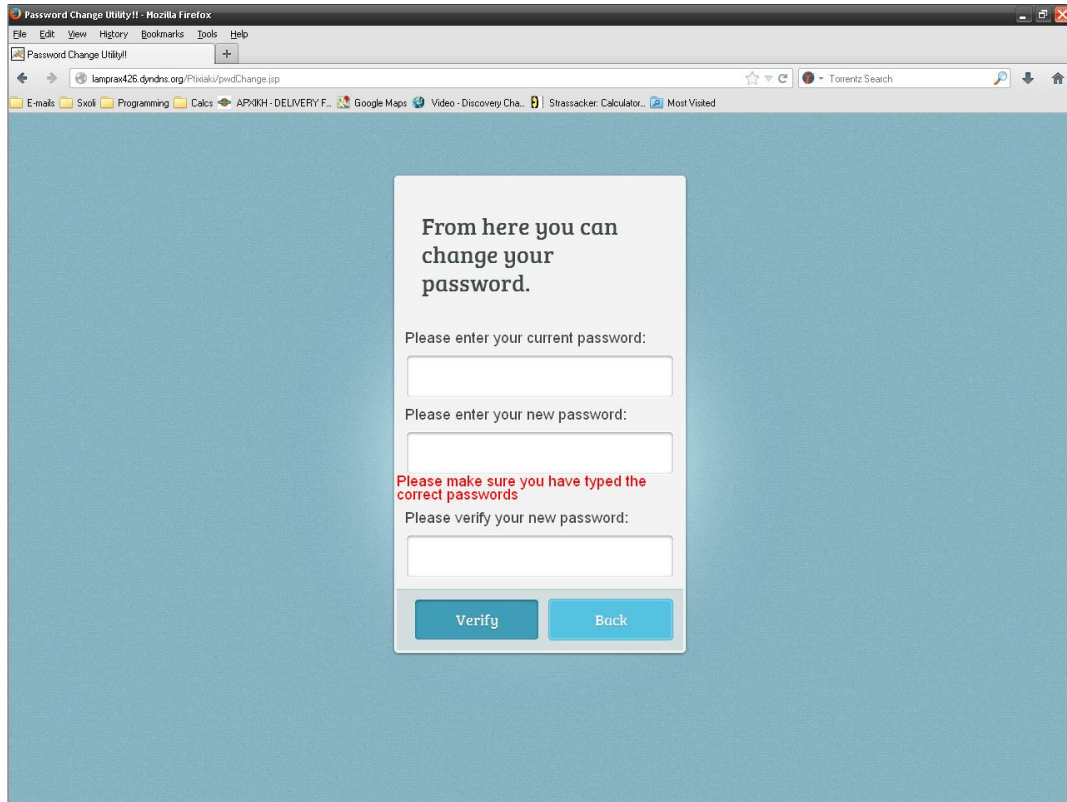
5.2.5 Φόρμα αλλαγής κωδικού ασφαλείας

Το σύστημα παρέχει την δυνατότητα στους χρήστες του της αλλαγής του κωδικού ασφαλείας τους. Εφόσον ο χρήστης επιθυμεί να αλλάξει τον κωδικό ασφαλείας του θα χρειαστεί να επιλέξει το κουμπί 'Change your Password' από την κεντρική οθόνη της εφαρμογής και θα μεταφερθεί στην φόρμα αλλαγής του κωδικού ασφαλείας.



Εικόνα 7: Φόρμα αλλαγής κωδικού ασφαλείας

Εκεί θα του ζητηθεί να επαληθεύσει τον τρέχων κωδικό ασφαλείας του, να εισαγάγει τον καινούριο κωδικό ασφαλείας και να τον επαληθεύσει. Στην συνέχεια το σύστημα θα ελέγξει αν ο τρέχων κωδικός ασφαλείας είναι ίδιος με αυτόν που είναι αποθηκευμένος στην βάση δεδομένων των χρηστών. Αν υπάρξει ασυμφωνία των κωδικών τότε θα υποθέσει ότι ο χειριστής της εφαρμογής δεν είναι εγγεγραμμένος χρήστης και θα προχωρήσει στην έξοδο του χρήστη από το σύστημα και την μεταφορά του στην φόρμα εισόδου χρήστη στην εφαρμογή. Στην περίπτωση που επαληθευτεί ο τρέχων κωδικός χρήστη το σύστημα θα προχωρήσει στον έλεγχο ταυτοποίησης μεταξύ του καινούριου κωδικού και στην επαλήθευση του από τον χρήστη. Στην περίπτωση ασυμφωνίας των δύο καινούριων κωδικών το σύστημα θα εμφανίσει μήνυμα σφάλματος στον χρήστη και θα τον παροτρύνει να επαναλάβει την διαδικασία προσέχοντας οι δύο καινούριοι κωδικοί να ταιριάζουν μεταξύ τους.

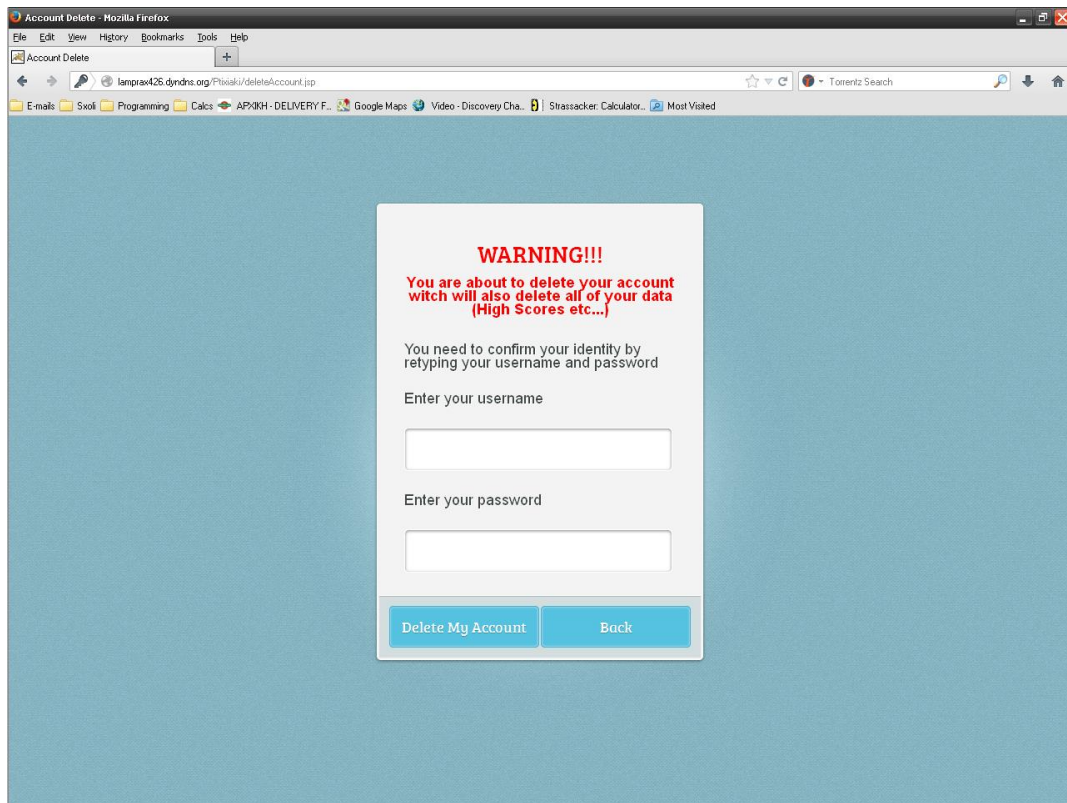


Εικόνα 8: Ασυμφωνία κωδικών

Στην περίπτωση που οι δύο καινούριοι κωδικοί συμφωνούν μεταξύ τους το σύστημα θα προχωρήσει στην καταχώριση του καινούριου κωδικού στην βάση δεδομένων των χρηστών και θα μεταβιβάσει τον χρήστη στην κεντρική σελίδα της εφαρμογής.

5.2.6 Φόρμα διαγραφής χρήστη από το σύστημα

Στην περίπτωση που ο χρήστης θελήσει να διαγραφεί από το σύστημα θα χρειαστεί να επιλέξει το κουμπί 'Delete your Account' που βρίσκετε τοποθετημένο στην μέση της δεξιάς στήλης κουμπιών στην κεντρική του οθόνη. Αυτό θα τον μεταφέρει στην παρακάτω εικονιζόμενη φόρμα.

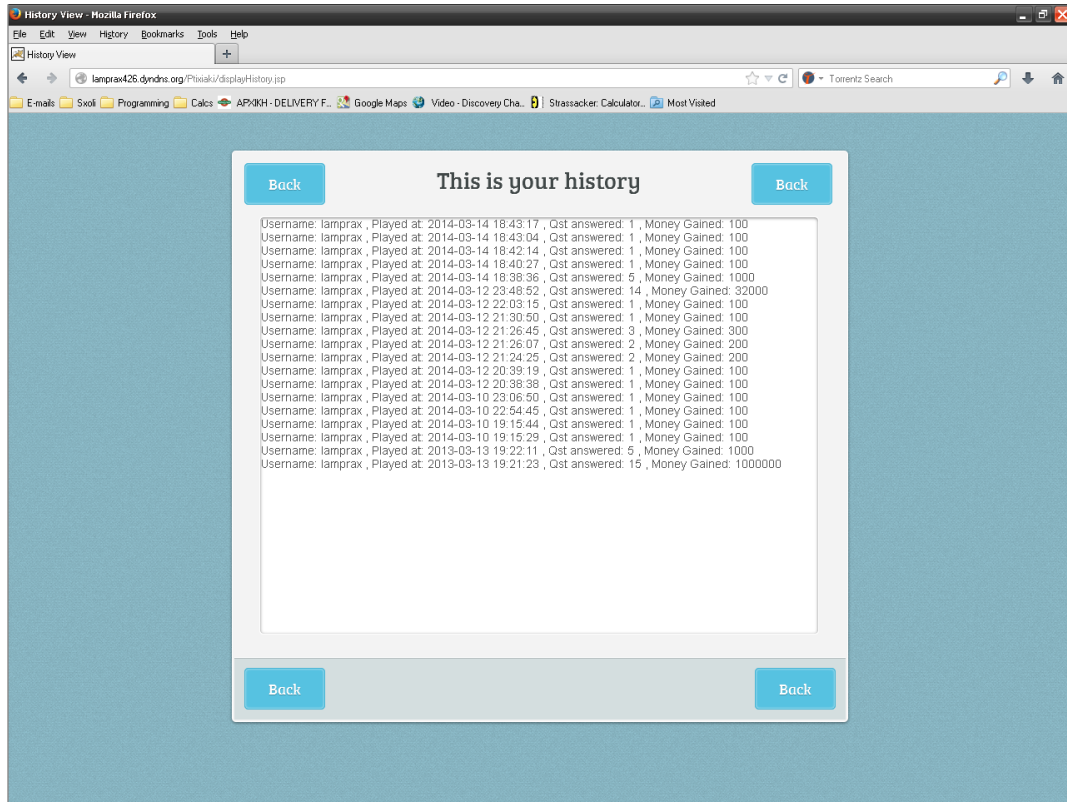


Εικόνα 9: Φόρμα διαγραφής χρήστη από το σύστημα

Σε αυτή την φόρμα ζητείται από τον χρήστη να ξανά εισαγάγει τα στοιχεία του για την επαλήθευση της ταυτότητάς του. Στην συνέχεια το σύστημα ελέγχει αν τα στοιχεία που έχουν εισαχθεί ταιριάζουν με τα στοιχεία του χρήστη που είναι εισαγμένος στο σύστημα. Αν διαπιστωθεί ασυμφωνία των στοιχείων το σύστημα αποβάλλει αυτόματα τον χρήστη και τον επιστρέφει στην φόρμα εισόδου χρήστη (5.2.1). Αν τα στοιχεία συμβαδίζουν με αυτά του συστήματος τότε ξεκινάει η διαδικασία διαγραφής χρήστη όπως αυτή περιγράφετε στην περίπτωση χρήσης «Διαγραφή του λογαριασμού του χρήστη».

5.2.7 Απεικόνιση των βαθμολογιών του χρήστη

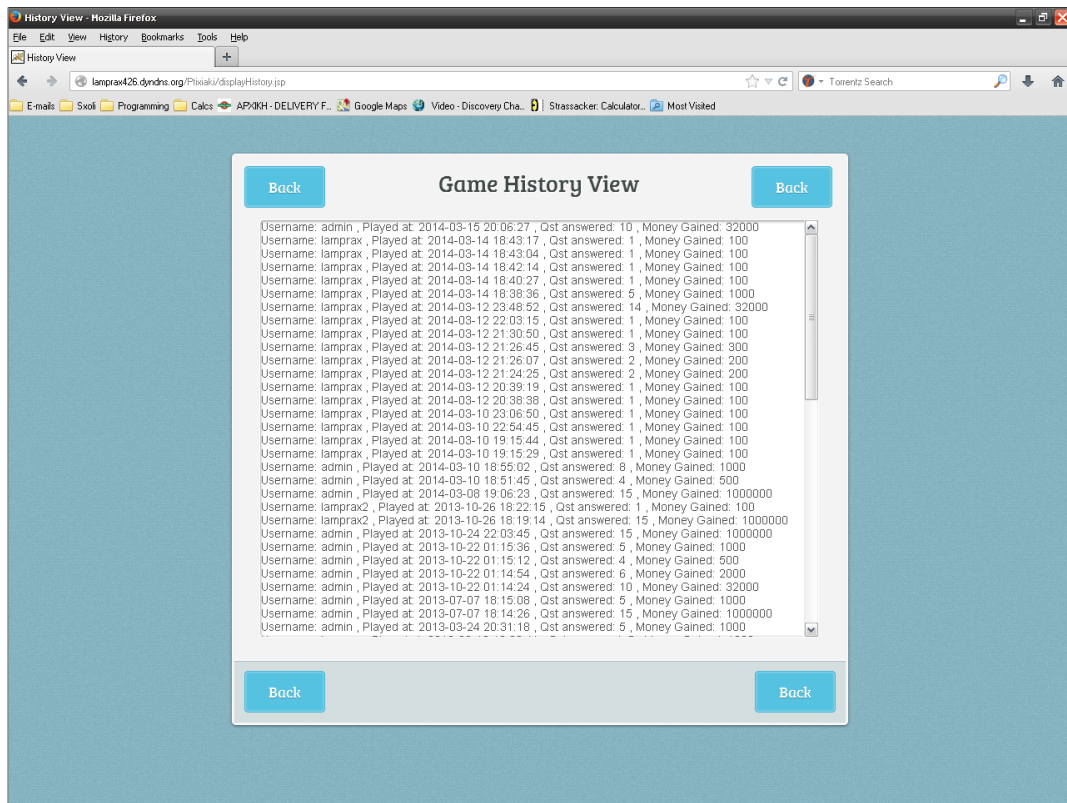
Όταν ο χρήστης επιθυμήσει να δει το ιστορικό της βαθμολογίας του θα χρειαστεί να επιλέξει από την κεντρική οθόνη το κουμπί με τον τίτλο 'View your History' που βρίσκετε στην πάνω γωνία της αριστερής στήλης κουμπιών. Αυτή η ενέργεια θα τον μεταβιβάσει στην σελίδα που απεικονίζεται παρακάτω και η οποία του απεικονίζει όλο το ιστορικό των βαθμολογιών που έχει επιτύχει στην εφαρμογή με φθίνουσα χρονολογική σειρά.



Εικόνα 10: Απεικόνιση των βαθμολογιών του χρήστη

5.2.8 Απεικόνιση των πιο πρόσφατων βαθμολογιών

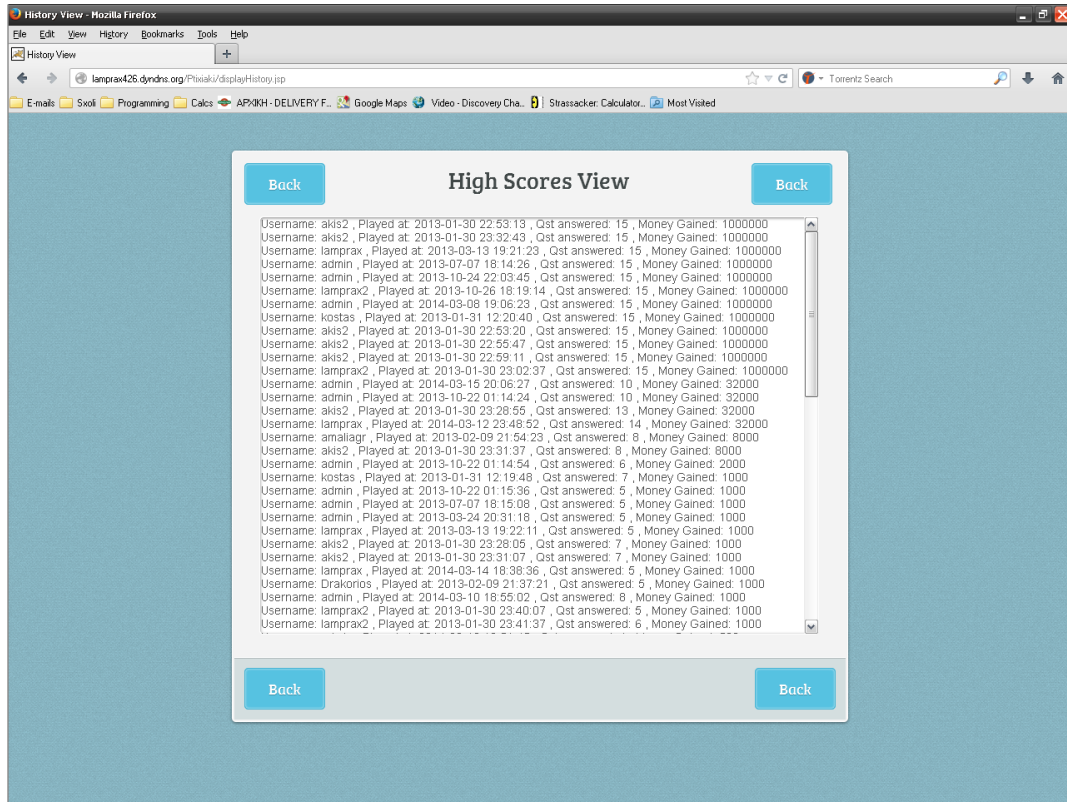
Όταν ο χρήστης επιθυμήσει να δει το ιστορικό των βαθμολογιών που έχουν επιτύχει όλοι οι χρήστες της εφαρμογής θα χρειαστεί να επιλέξει από την κεντρική οθόνη το κουμπί με τον τίτλο 'View all Recent History' που βρίσκεται στο κέντρο της αριστερής στήλης κουμπιών. Αυτή η ενέργεια θα τον μεταβιβάσει στην σελίδα που απεικονίζεται παρακάτω και η οποία του απεικονίζει όλο το ιστορικό των βαθμολογιών στην εφαρμογή με φθίνουσα χρονολογική σειρά.



Εικόνα 11: Απεικόνιση των πιο πρόσφατων βαθμολογιών

5.2.9 Απεικόνιση των καλύτερων βαθμολογιών

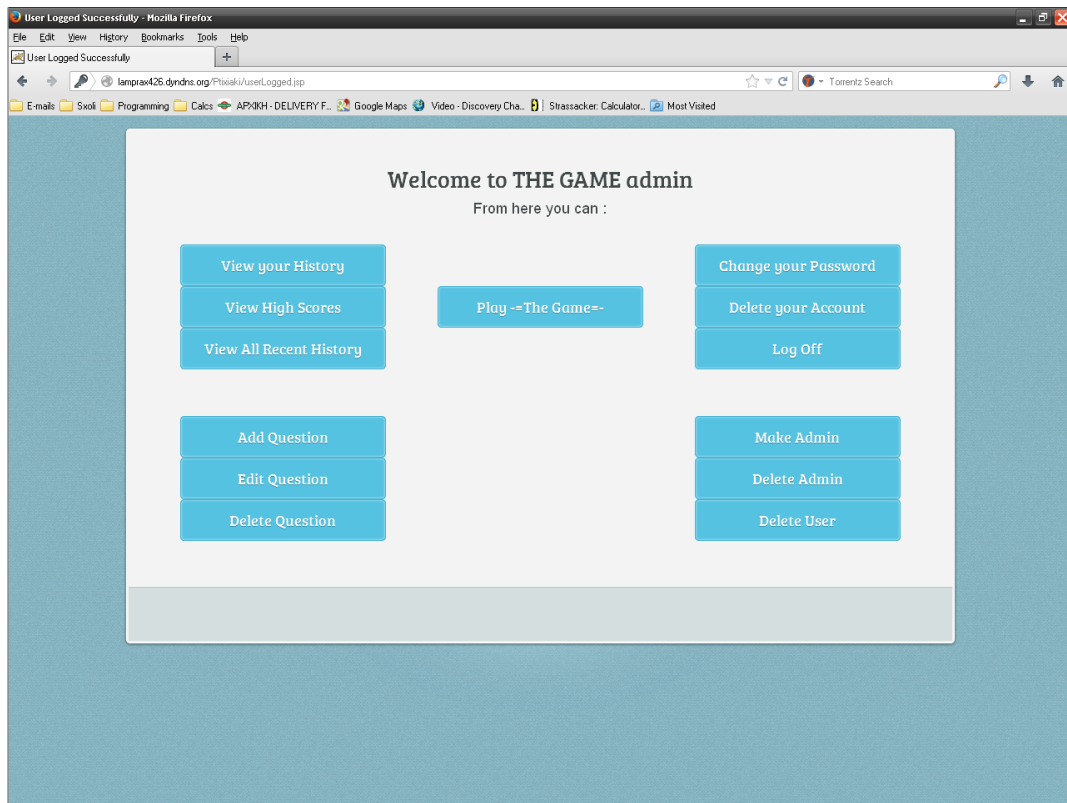
Όταν ο χρήστης επιθυμήσει να δει το ιστορικό των βαθμολογιών που έχουν επιτύχει όλοι οι χρήστες της εφαρμογής θα χρειαστεί να επιλέξει από την κεντρική οθόνη το κουμπί με τον τίτλο 'View High Scores' που βρίσκεται στην κάτω γωνία της αριστερής στήλης κουμπιών. Αυτή η ενέργεια θα τον μεταβιβάσει στην σελίδα που απεικονίζεται παρακάτω και η οποία του απεικονίζει τις καλύτερες βαθμολογίες στην εφαρμογή με φθίνουσα σειρά.



Εικόνα 12: Απεικόνιση των καλύτερων βαθμολογιών

5.2.10 Η κεντρική οθόνη του διαχειριστή

Μετά την επιτυχή εισαγωγή διαχειριστή το σύστημα θα τον προωθήσει στην κεντρική οθόνη της εφαρμογής όπου τον υποδέχεται και του παρουσιάζονται οι δυνατότητες που έχει στην επιλογή του. Όπως είναι εμφανές η κεντρική οθόνη του διαχειριστή είναι επέκταση της κεντρικής οθόνης του απλού χρήστη και αυτό βασίζεται στο γεγονός ότι και οι δύο έχουν τις ίδιες βασικές ιδιότητες απλώς ο διαχειριστής έχει και κάποιες επιπλέον δυνατότητες. Να σημειωθεί εδώ πως ό,τι είπαμε παραπάνω ισχύουν και στην περίπτωση που ο χρήστης είναι διαχειριστής.

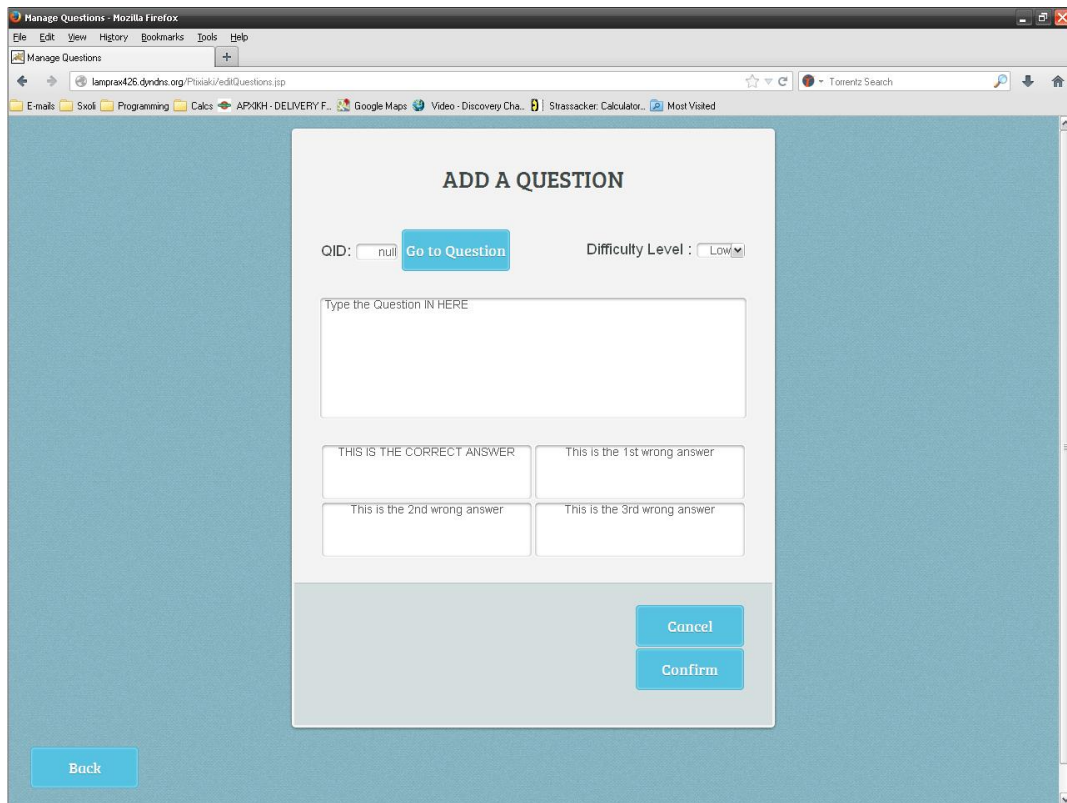


Εικόνα 13: Η κεντρική οθόνη του διαχειριστή

Εν συνεχεία θα δούμε πιο αναλυτικά τις παραπάνω επιλογές που έχει στην διάθεσή του ένας διαχειριστής της εφαρμογής.

5.2.11 Η φόρμα προσθήκης ερώτησης

Όταν κάποιος διαχειριστής θελήσει να προσθέσει κάποια ερώτηση στην βάση δεδομένων των ερωτήσεων θα πρέπει από την κεντρική οθόνη της εφαρμογής να πατήσει το κουμπί “Add Question” που βρίσκεται στην κορυφή της κάτω αριστερής στήλης κουμπιών. Αυτή η ενέργεια θα τον μεταβιβάσει στην φόρμα που απεικονίζεται παρακάτω.

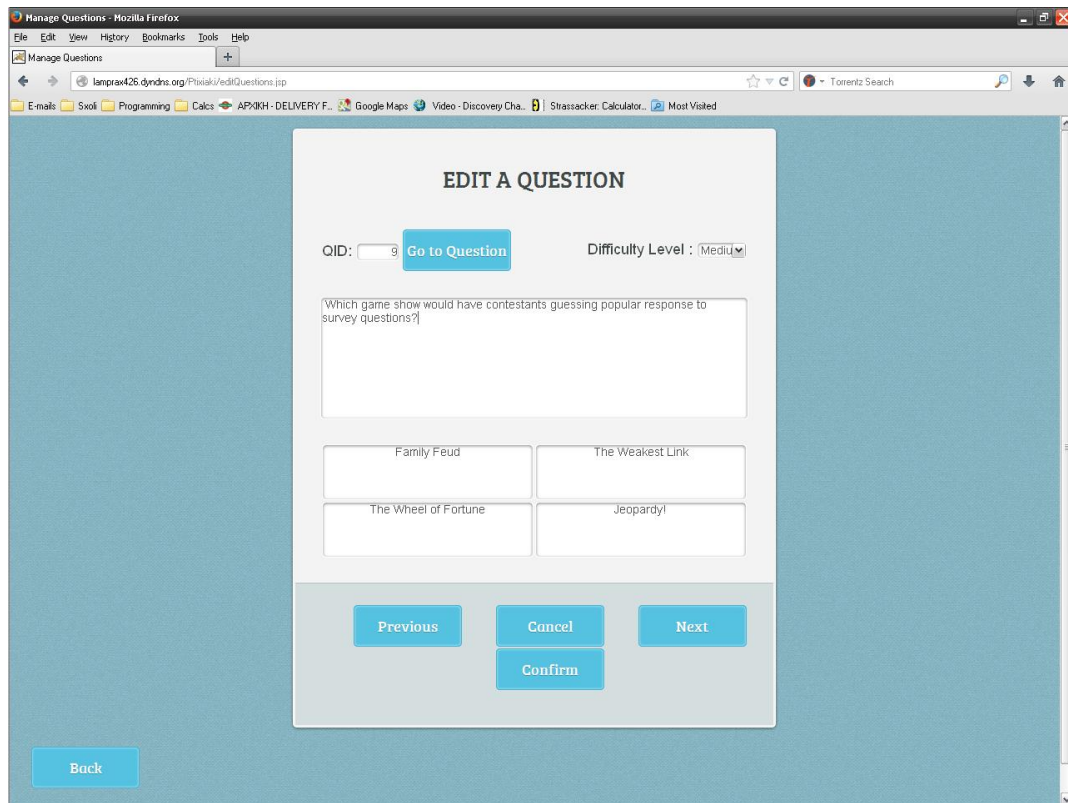


Εικόνα 14: φόρμα προσθήκης ερώτησης

Εδώ θα του ζητηθεί να συμπληρώσει την ερώτηση, την σωστή απάντηση και τρεις λάθος απαντήσεις για την ερώτηση. Επίσης του ζητείται να επιλέξει τον βαθμό δυσκολίας της ερώτησης. Όταν ολοκληρώσει την συμπλήρωση των πεδίων μπορεί να πατήσει το κουμπί “Confirm” για την καταχώριση της ερώτησης. Στην συνέχεια, το σύστημα διαβάζει τα πεδία της φόρμας και τα καταχωρεί στην βάση δεδομένων των ερωτήσεων. Όταν ολοκληρωθεί αυτή η διαδικασία το σύστημα επιστρέφει τον διαχειριστή στην φόρμα προσθήκης ερωτήσεων. Αν ο διαχειριστής επιλέξει το κουμπί “Cancel” η εφαρμογή αρχικοποιεί τα πεδία της φόρμας. Αν ο διαχειριστής επιλέξει το κουμπί “Back” θα επιστρέψει στην κεντρική του οθόνη.

5.2.12 Η φόρμα επεξεργασίας ερώτησης

Όταν κάποιος διαχειριστής θελήσει να επεξεργαστεί ή να διορθώσει κάποια ερώτηση στην βάση δεδομένων των ερωτήσεων θα πρέπει από την κεντρική οθόνη της εφαρμογής να πατήσει το κουμπί “Edit Question” που βρίσκεται στην μέση της κάτω αριστερής στήλης κουμπιών. Αυτή η ενέργεια θα τον μεταβιβάσει στην φόρμα που απεικονίζεται παρακάτω.

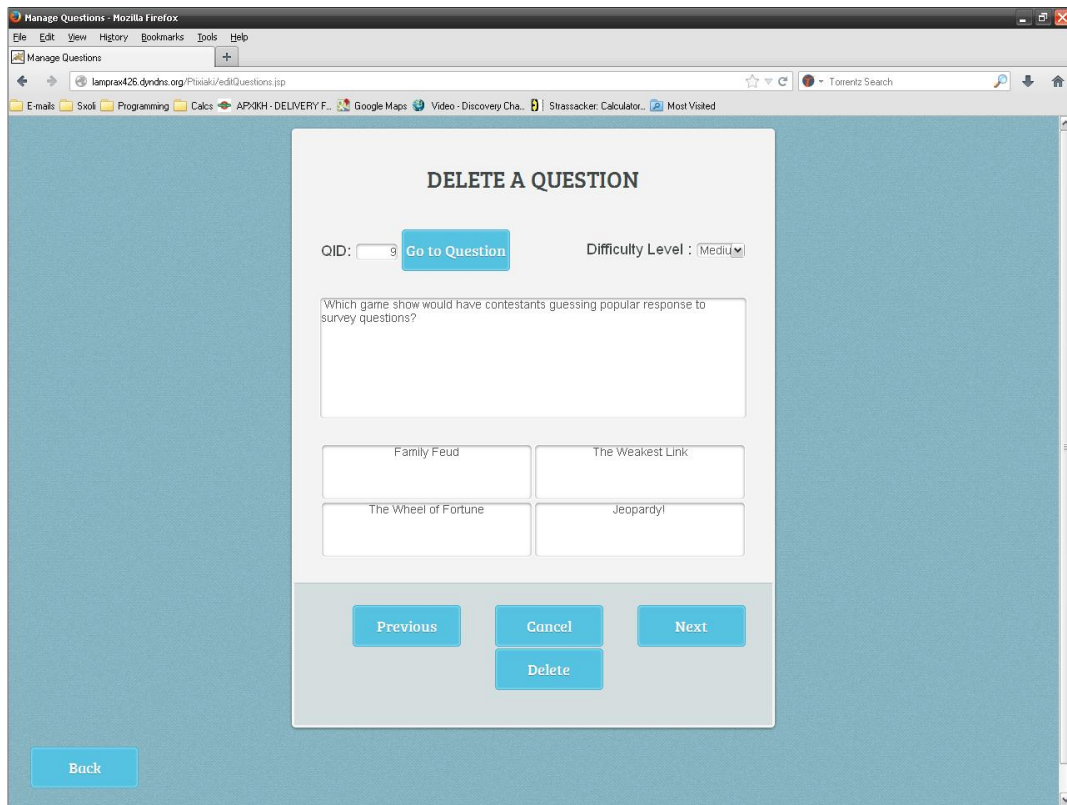


Εικόνα 15: Η φόρμα επεξεργασίας ερώτησης

Στην φόρμα που θα του εμφανιστεί έχουν ήδη διαβαστεί και του απεικονίζονται τα περιεχόμενα της πρώτης ερώτησης που είναι αποθηκευμένη στη βάση δεδομένων των ερωτήσεων. Ο διαχειριστής μπορεί εάν το επιθυμεί να πλοηγηθεί στις ερωτήσεις με την χρήση των κουμπιών “Previous” και “Next” για να προχωρήσει στην προηγούμενη ή στην επόμενη ερώτηση αντιστοίχως μέχρι να του απεικονιστεί η ερώτηση που θέλει να τροποποιήσει. Εναλλακτικά εάν ξέρει τον αύξων αριθμό της ερώτησης μπορεί να τον συμπληρώσει απ’ ευθείας στο πεδίο “QID” και να πατήσει το κουμπί “Go to Question”. Όταν εντοπίσει την ερώτηση που θέλει να τροποποιήσει θα μπορέσει να αλλάξει οποιοδήποτε πεδίο της ερώτησης με εξαίρεση τον αύξων αριθμό. Για να καταχωρίσει τις αλλαγές του θα χρειαστεί να πατήσει το κουμπί “Confirm”. Σε αντίθετη περίπτωση θα χρειαστεί να πατήσει το κουμπί “Cancel” το οποίο θα αρχικοποιήσει τα περιεχόμενα της φόρμας απορρίπτοντας οποιοδήποτε αλλαγές και θα επανεμφανίσει τα πεδία της τρέχουσας ερώτησης. Αν ο διαχειριστής επιλέξει το κουμπί “Back” θα επιστρέψει στην κεντρική του οθόνη.

5.2.13 Η φόρμα διαγραφής ερώτησης

Όταν κάποιος διαχειριστής θελήσει να διαγράψει κάποια ερώτηση από βάση δεδομένων των ερωτήσεων θα πρέπει από την κεντρική οθόνη της εφαρμογής να πατήσει το κουμπί “Delete Question” που βρίσκεται στην βάση της κάτω αριστερής στήλης κουμπιών. Αυτή η ενέργεια θα τον μεταβιβάσει στην φόρμα που απεικονίζεται παρακάτω.

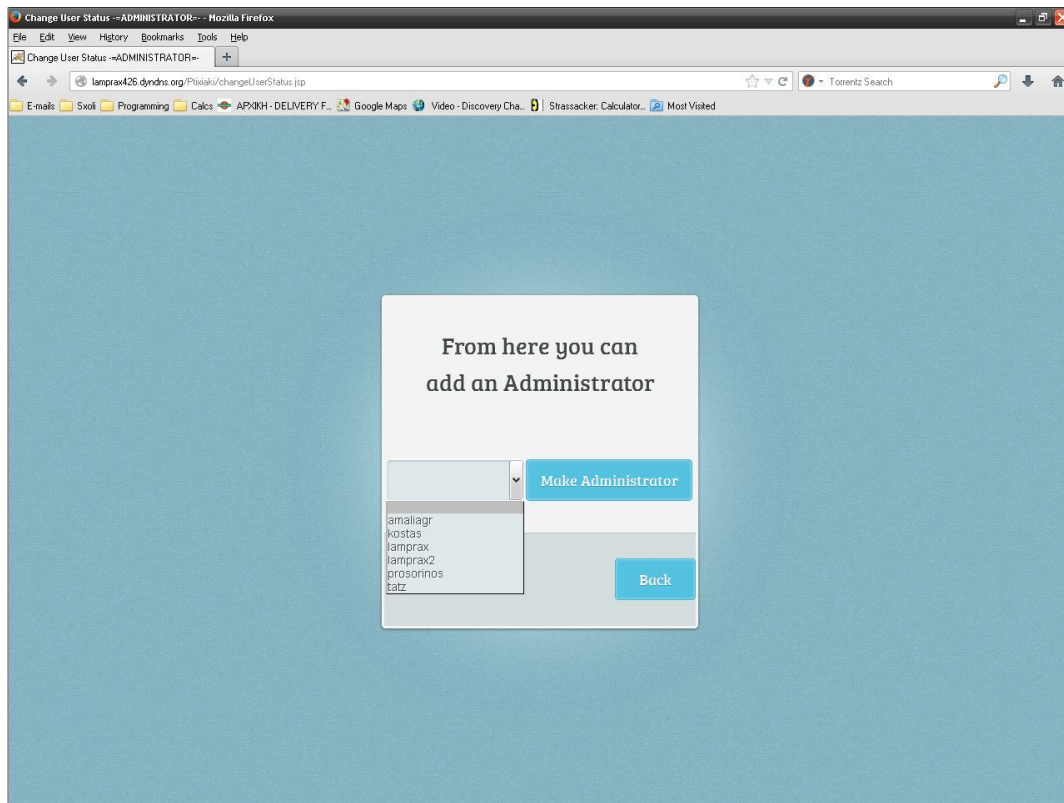


Εικόνα 16: Η φόρμα διαγραφής ερώτησης

Στην φόρμα που θα του εμφανιστεί έχουν ήδη διαβαστεί και του απεικονίζονται τα περιεχόμενα της πρώτης ερώτησης που είναι αποθηκευμένη στη βάση δεδομένων των ερωτήσεων. Ο διαχειριστής μπορεί εάν το επιθυμεί να πλοηγηθεί στις ερωτήσεις με την χρήση των κουμπιών “Previous” και “Next” για να προχωρήσει στην προηγούμενη ή στην επόμενη ερώτηση αντιστοίχως μέχρι να του απεικονιστεί η ερώτηση που θέλει να διαγράψει. Εναλλακτικά εάν ξέρει τον αύξων αριθμό της ερώτησης μπορεί να τον συμπληρώσει απ’ ευθείας στο πεδίο “QID” και να πατήσει το κουμπί “Go to Question”. Όταν εντοπίσει την ερώτηση που θέλει να διαγράψει θα χρειαστεί να πατήσει το κουμπί “Delete”. Σε αντίθετη περίπτωση θα χρειαστεί να πατήσει το κουμπί “Cancel” το οποίο θα τον επαναφέρει στην κεντρική οθόνη της εφαρμογής. Αν ο διαχειριστής επιλέξει το κουμπί “Back” θα επιστρέψει στην κεντρική του οθόνη.

5.2.14 Η φόρμα προσθήκης διαχειριστή

Όταν κάποιος διαχειριστής θελήσει να προάγει έναν χρήστη σε διαχειριστή της εφαρμογής θα χρειαστεί να πατήσει το κουμπί “Make Admin” που βρίσκεται στην κορυφή της κάτω δεξιάς στήλης κουμπιών της κεντρικής οθόνης της εφαρμογής. Αυτή η ενέργεια θα τον μεταβιβάσει στην φόρμα που απεικονίζεται παρακάτω.

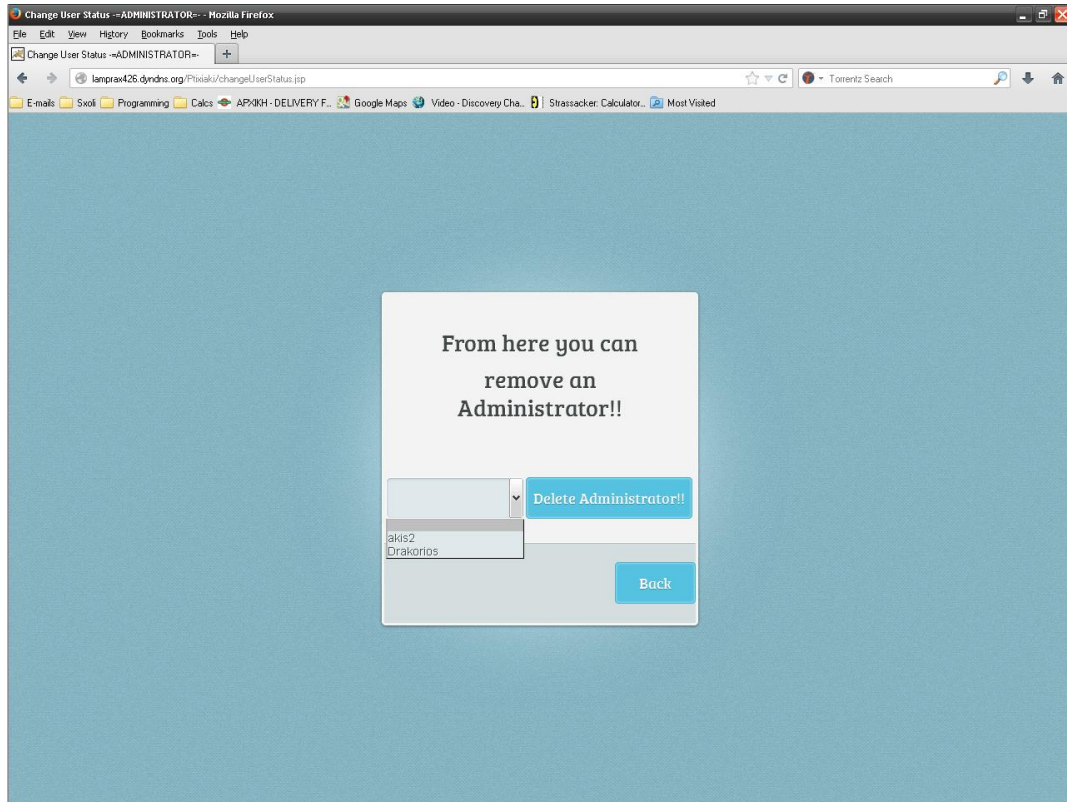


Εικόνα 17: Η φόρμα προσθήκης διαχειριστή

Εδώ θα του ζητηθεί να επιλέξει από την αναπτυσσόμενη λίστα το όνομα του χρήστη που θέλει να προάγει σε διαχειριστή της εφαρμογής. Η λίστα συμπληρώνεται από το σύστημα με τα ονόματα των διαθέσιμων χρηστών που δεν είναι διαχειριστές της εφαρμογής. Αφού επιλέξει το όνομα θα χρειαστεί να πατήσει το κουμπί “Make Administrator” για να επιβεβαιώσει την επιλογή του όπου και το σύστημα θα μετατρέψει τον επιλεγμένο χρήστη σε διαχειριστή. Μετά το πέρας της μετατροπής το σύστημα θα επιστρέψει τον διαχειριστή στη ίδια φόρμα ώστε να ορίσει και άλλους διαχειριστές αν το επιθυμεί. Στην περίπτωση που ο διαχειριστής πατήσει το κουμπί “Back” η εφαρμογή θα τον επιστρέψει στην κεντρική οθόνη της εφαρμογής.

5.2.15 Η φόρμα αφαίρεσης δικαιωμάτων διαχειριστή

Όταν κάποιος διαχειριστής θελήσει να αφαιρέσει τα διαχειριστικά δικαιώματα κάποιου άλλου διαχειριστή της εφαρμογής θα χρειαστεί να πατήσει το κουμπί “Delete Admin” που βρίσκεται στην μέση της κάτω δεξιάς στήλης κουμπιών της κεντρικής οθόνης της εφαρμογής. Αυτή η ενέργεια θα τον μεταβιβάσει στην φόρμα που απεικονίζεται παρακάτω.

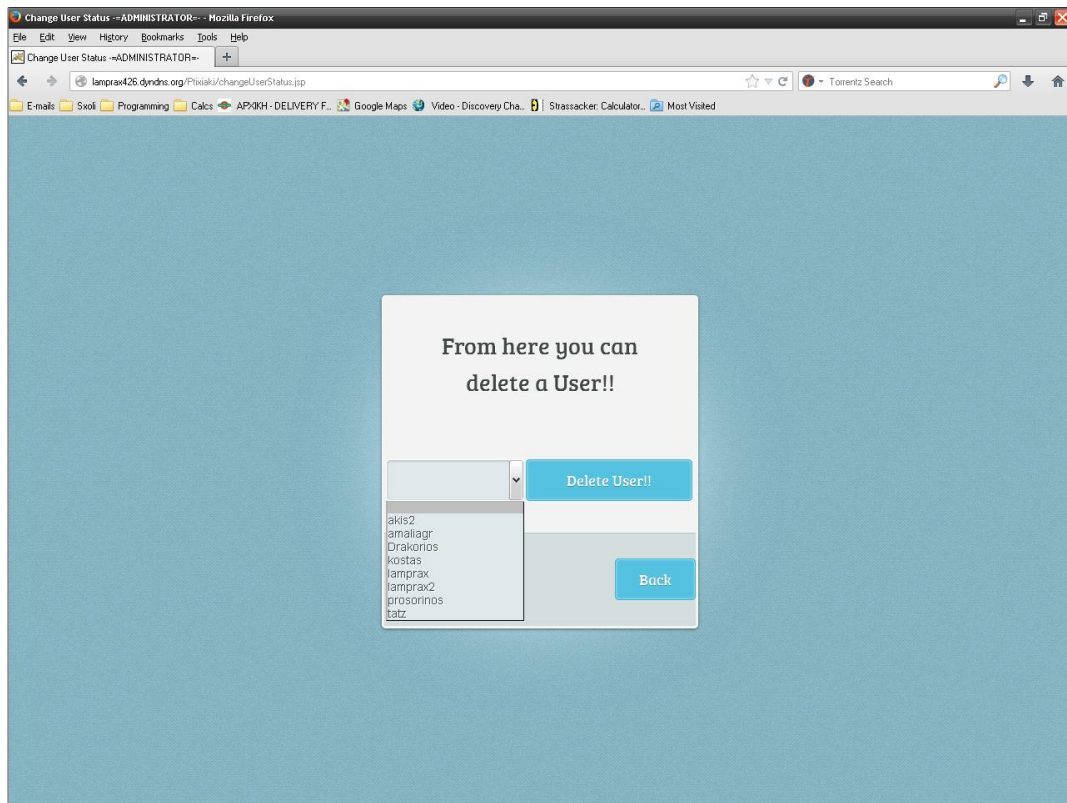


Εικόνα 18: Η φόρμα αφαίρεσης δικαιωμάτων διαχειριστή

Εδώ θα του ζητηθεί να επιλέξει από την αναπτυσσόμενη λίστα το όνομα του χρήστη που θέλει να διαγράψει από διαχειριστή της εφαρμογής. Η λίστα συμπληρώνεται από το σύστημα με τα ονόματα των διαθέσιμων χρηστών που είναι διαχειριστές της εφαρμογής με εξαίρεση τον τρέχων. Αφού επιλέξει το όνομα θα χρειαστεί να πατήσει το κουμπί “Delete Administrator!!” για να επιβεβαιώσει την επιλογή του όπου και το σύστημα θα μετατρέψει τον επιλεγμένο διαχειριστή σε χρήστη. Μετά το πέρας της μετατροπής το σύστημα θα επιστρέψει τον διαχειριστή στη ίδια φόρμα ώστε να διαγράψει και άλλους διαχειριστές αν το επιθυμεί. Στην περίπτωση που ο διαχειριστής πατήσει το κουμπί “Back” η εφαρμογή θα τον επιστρέψει στην κεντρική οθόνη της εφαρμογής.

5.2.16 Η φόρμα για την διαγραφή χρήστη από τον διαχειριστή

Όταν κάποιος διαχειριστής θελήσει να διαγράψει κάποιον χρήστη της εφαρμογής θα χρειαστεί να πατήσει το κουμπί “Delete User” που βρίσκεται στην βάση της κάτω δεξιάς στήλης κουμπιών της κεντρικής οθόνης της εφαρμογής. Αυτή η ενέργεια θα τον μεταβιβάσει στην φόρμα που απεικονίζεται παρακάτω.

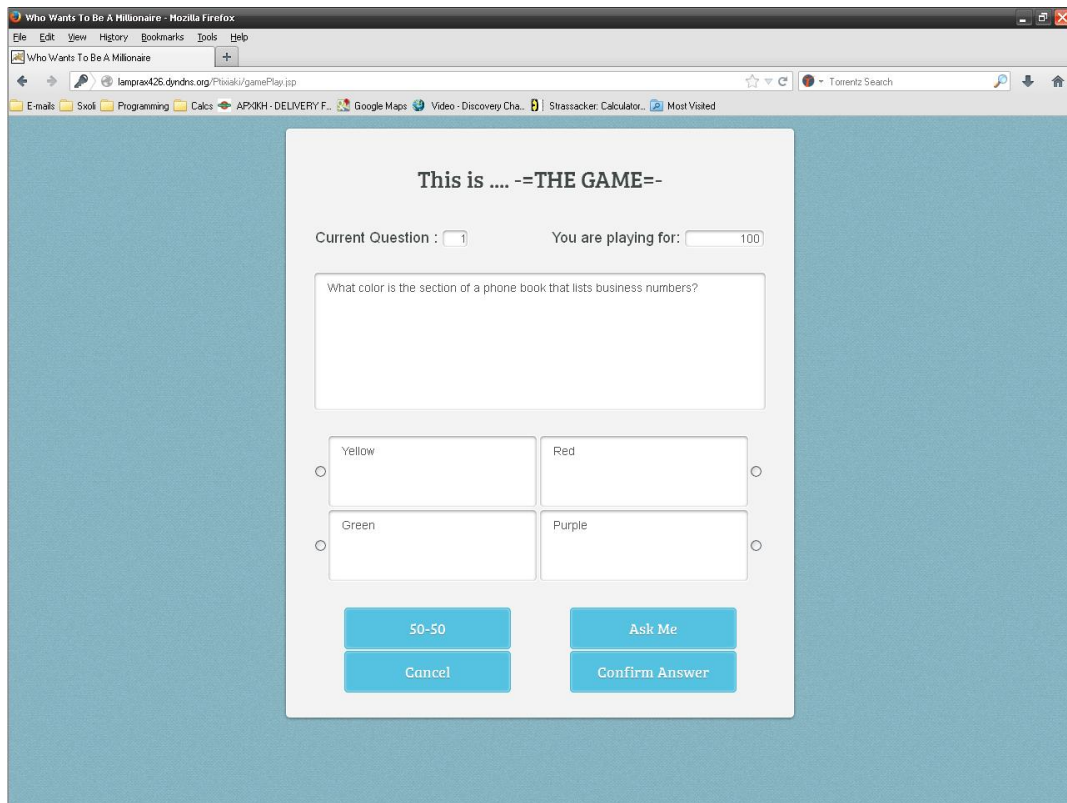


Εικόνα 19: Η φόρμα για την διαγραφή χρήστη από τον διαχειριστή

Εδώ θα του ζητηθεί να επιλέξει από την αναπτυσσόμενη λίστα το όνομα του χρήστη που θέλει να διαγράψει από την εφαρμογή. Η λίστα συμπληρώνεται από το σύστημα με τα ονόματα των διαθέσιμων χρηστών και διαχειριστών της εφαρμογής με εξαίρεση τον τρέχων. Αφού επιλέξει το όνομα θα χρειαστεί να πατήσει το κουμπί “Delete User!!” για να επιβεβαιώσει την επιλογή του όπου και το σύστημα θα διαγράψει το ιστορικό του επιλεγμένου χρήστη από την βάση δεδομένων του ιστορικού και στην συνέχεια θα διαγράψει τον χρήστη από την βάση δεδομένων των χρηστών. Μετά το πέρας της μετατροπής το σύστημα θα επιστρέψει τον διαχειριστή στη ίδια φόρμα ώστε να διαγράψει και άλλους χρήστες ή διαχειριστές αν το επιθυμεί. Στην περίπτωση που ο διαχειριστής πατήσει το κουμπί “Back” η εφαρμογή θα τον επιστρέψει στην κεντρική οθόνη της εφαρμογής.

5.2.17 Η σελίδα του παιχνιδιού

Όταν κάποιος χρήστης της εφαρμογής θελήσει να παίξει το παιχνίδι θα χρειαστεί να πατήσει το κουμπί “Play -=The Game=-” που βρίσκεται στην μέση της κεντρικής σελίδας της εφαρμογής. Αυτή η ενέργεια θα τον μεταβιβάσει στην φόρμα του παιχνιδιού που απεικονίζεται παρακάτω.

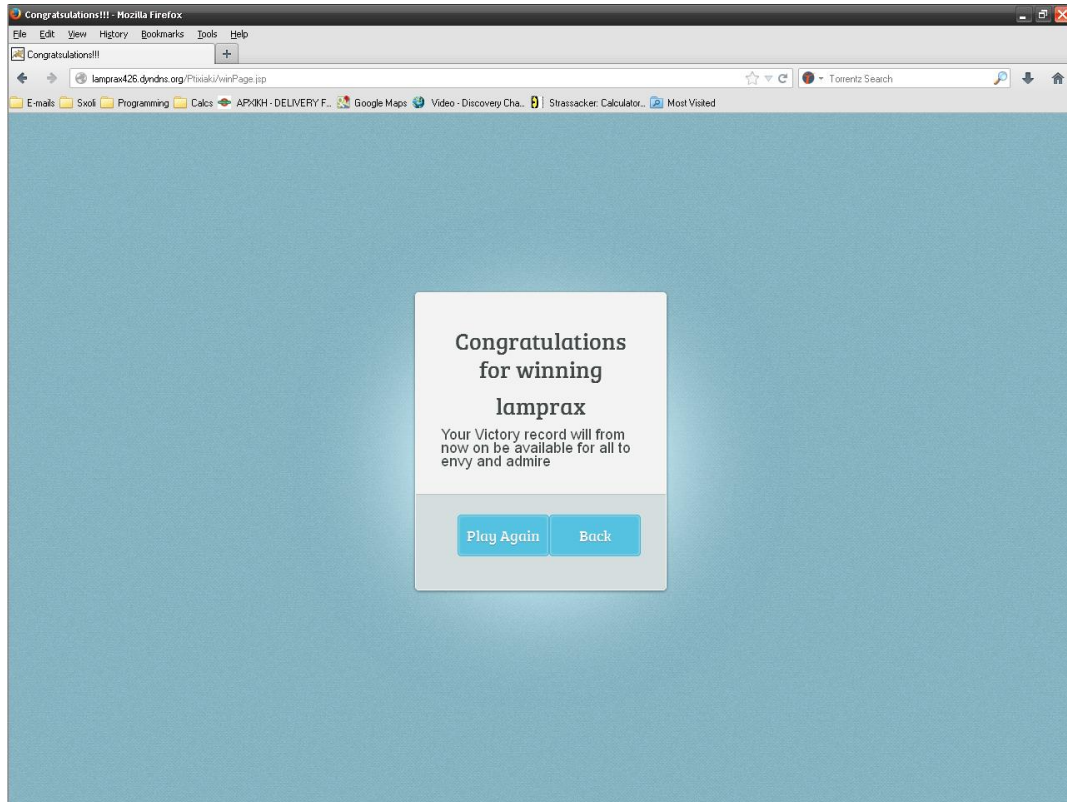


Εικόνα 20: Η σελίδα του παιχνιδιού

Σε αυτή τη φόρμα του εμφανίζονται στα αντίστοιχα πεδία ο αριθμός της τρέχουσας ερώτησης στο παιχνίδι, η βαθμολογία την οποία διεκδικεί, η ερώτηση που καλείτε να απαντήσει και οι πιθανές απαντήσεις της ερώτησης. Επίσης στην φόρμα υπάρχουν δύο κουμπιά ελέγχου, το “Confirm Answer” που θα δώσει την εντολή στο σύστημα να ελέγξει την απάντηση του χρήστη και το “Cancel” που θα παραιτήσει τον χρήστη από το παιχνίδι και θα τον προωθήσει στην σελίδα παραίτησης του παιχνιδιού.

5.2.18 Ο χρήστης κερδίζει το παιχνίδι

Όταν ο χρήστης απαντήσει σωστά σε δεκαπέντε ερωτήσεις τότε κερδίζει το παιχνίδι και προωθείτε στην παρακάτω φόρμα που τον ενημερώνει και συγχαίρει για την νίκη του.

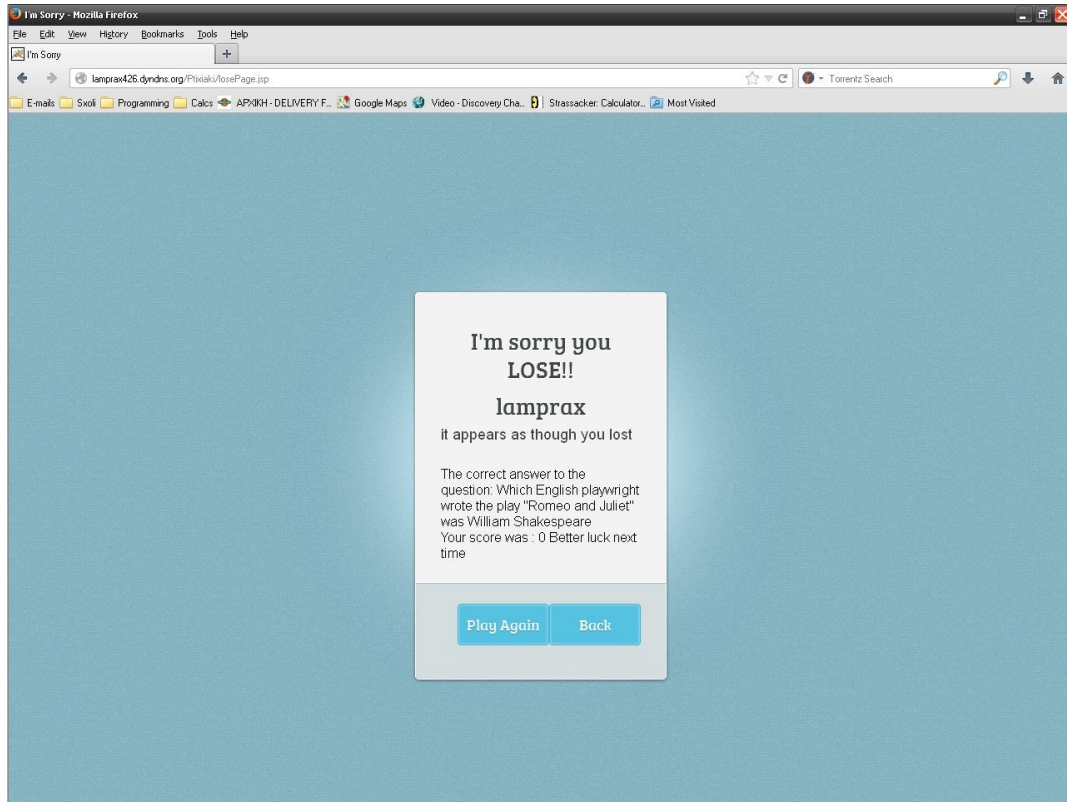


Εικόνα 21: Ο χρήστης κερδίζει το παιχνίδι

Σε αυτή την φόρμα δίνετε στον χρήστη η δυνατότητα να ξαναπαίξει το παιχνίδι ή να επιστρέψει στην κεντρική σελίδα της εφαρμογής πατώντας τα κουμπιά “Play Again” και “Back” αντιστοίχως.

5.2.19 Ο χρήστης χάνει στο παιχνίδι

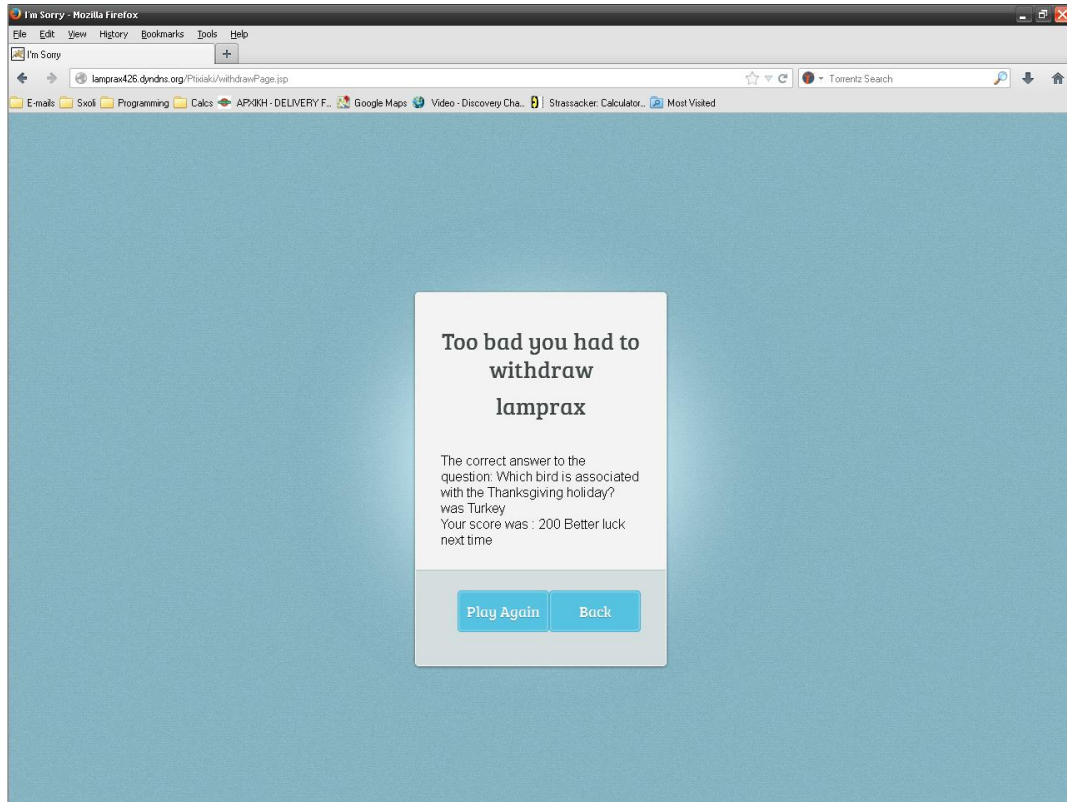
Στην περίπτωση που ο χρήστης απαντήσει λάθος σε κάποια από τις ερωτήσεις, χάνει το παιχνίδι και θα προωθηθεί αυτόματα στη παρακάτω φόρμα η οποία θα τον ενημερώσει για την ήττα του, την σωστή απάντηση στην ερώτηση που έκανε λάθος και την βαθμολογία που κέρδισε από το παιχνίδι που έπαιξε. Σε αυτή την φόρμα δίνετε στον χρήστη η δυνατότητα να ξαναπαίξει το παιχνίδι ή να επιστρέψει στην κεντρική σελίδα της εφαρμογής πατώντας τα κουμπιά “Play Again” και “Back” αντιστοίχως.



Εικόνα 22: Ο χρήστης χάνει στο παιχνίδι

5.2.20 Ο χρήστης επιλέγει την έξοδο του από το παιχνίδι

Αν κατά την διάρκεια του παιχνιδιού ο χρήστης πατήσει το κουμπί “Cancel” θα μεταβεί στην φόρμα που απεικονίζεται παρακάτω φόρμα η οποία θα τον ενημερώσει για την παραίτησή του, την σωστή απάντηση στην ερώτηση που παραιτήθηκε και την βαθμολογία που κέρδισε από το παιχνίδι που έπαιξε. Σε αυτή την φόρμα δίνετε στον χρήστη η δυνατότητα να ξαναπαίξει το παιχνίδι ή να επιστρέψει στην κεντρική σελίδα της εφαρμογής πατώντας τα κουμπιά “Play Again” και “Back” αντιστοίχως.



Εικόνα 23: Ο χρήστης επιλέγει την έξοδο του από το παιχνίδι

6. Αποτελέσματα

Η εκπαίδευση ως διαδικασία δεν είναι στατική, όπως δεν είναι στατικό και το κοινωνικό περιβάλλον. Στην ιστορική της διαδρομή ως θεσμού επηρεάζει αλλά και επηρεάζεται από αλλαγές που προκύπτουν στα επιστημονικά, κοινωνικά, πολιτικά και οικονομικά δεδομένα, με πιο πρόσφατο παράδειγμα τις αλλαγές που οφείλονται στην εισαγωγή και χρήση των νέων μέσων ή ΤΠΕ, όπως ονομάζονται: των ηλεκτρονικών υπολογιστών και του διαδικτύου. Σύμφωνα με τον Anthony Giddens τα μέσα επικοινωνίας ασκούν σημαντικές επιδράσεις «γιατί είναι τα μέσα πρόσβασης στην γνώση από την οποία εξαρτώνται πολλές κοινωνικές μας δραστηριότητες». Η πρόσβαση στη γνώση γίνεται επιτακτική ανάγκη λόγω των έντονων αλλαγών που συντελούνται στην κοινωνική και επαγγελματική ζωή των πολιτών. Η διά βίου μάθηση (ΔΒΜ) εξυπηρετεί τη διαρκή επικαιροποίηση των δεξιοτήτων, την αναβάθμιση των γνώσεων, την επιμόρφωση, κατάρτιση ή εξειδίκευση των επαγγελματιών. Αλλά δε θα μπορούσε να γίνει εφικτή στο εύρος που συναντάται σήμερα αν δεν υπήρχε η κατάλληλη επικοινωνιακή τεχνολογία να την στηρίξει. Η έλευση και διάδοση των ΤΠΕ, λοιπόν, στήριξε το βασικό σκέλος της ΔΒΜ, την εκπαίδευση από απόσταση, που δημιουργεί νέα δεδομένα στην εκπαίδευση και επιφέρει τεράστιες αλλαγές στο ρόλο, τη δομή, τη λειτουργία, την απήχηση και συμβολή της στην κοινωνία.

Η καθοριστική επίδραση των νέων τεχνολογιών στην εκπαίδευση γίνεται εμφανής αν αναλογιστούμε ότι το διαδίκτυο, εκτός από το αμερικανικό Πεντάγωνο, ήταν αρχικά διαδεδομένο και σε πανεπιστήμια των θετικών επιστημών, κυρίως, για μεγάλο χρονικό διάστημα. Όσο το διαδίκτυο γινόταν κτήμα ολοένα και περισσότερων πολιτών ανά τον κόσμο, τόσο διαπιστωνόταν και η ανάγκη να αναθεωρηθεί ο ρόλος της τεχνολογίας, ώστε η γνώση να διαχέεται ευκολότερα, ταχύτερα και αποτελεσματικότερα. Οι δυνατότητες που παρείχαν τα νέα μέσα ήταν πλέον απεριόριστες, καθιστώντας τη μάθηση δυνατή σε συντομότερο χρονικό διάστημα, με μικρότερο κόστος, με δυνατότητα επιλογής μεταξύ πολλών μαθησιακών εργαλείων, διδακτικών μοντέλων και προγραμμάτων σπουδών, και μάλιστα από έγκυρους φορείς και εκπαιδευτικά ιδρύματα όλων των βαθμίδων.

Σήμερα, η εκπαίδευση με Νέες Τεχνολογίες έχει ενταχθεί στη ζωή μας, καθιστώντας τη μάθηση εφικτή έξω από το χώρο, το χρόνο, την ηλικία, την οικογενειακή ή επαγγελματική κατάσταση, το επίπεδο σπουδών, την πνευματική ή σωματική αρτιμέλεια ή ιδιαιτερότητα, το φύλο ή τη φυλή, τα ενδιαφέροντα. Παρέχει ευκαιρίες για εκπαίδευση και μόρφωση σε όλους, ανεξάρτητα από περιορισμούς και διακρίσεις. Αποδεσμεύει την εκπαιδευτική κοινότητα από τα ασφυκτικά χωροχρονικά πλαίσια διδασκαλίας, δίνοντας την ευκαιρία στον καθένα να καθορίσει ο ίδιος το ρυθμό, το χρόνο, το χώρο, τον τρόπο μάθησης που τον εξυπηρετεί. Συμβάλλει στην εξοικονόμηση πόρων και για τους φορείς αλλά και για τους πολίτες, καθώς τα μέσα και τα υλικά που απαιτούνται είναι ελάχιστα, εύχρηστα και συχνά αρκετά πιο οικονομικά. Προωθεί την ανάπτυξη των ιδιαίτερων ενδιαφερόντων και κλίσεων μέσα από τη δυνατότητα επιλογής μιας πολύ μεγάλης γκάμας εκπαιδευτικών προγραμμάτων. Σέβεται την ατομικότητα ενώ παράλληλα προωθεί την ενίσχυση της συνεργασίας και της αλληλεπίδρασης των συμμετεχόντων. Συμπληρώνει και εμπλουτίζει τις παραδοσιακές μεθόδους εκπαίδευσης, προσδίδοντας ευελιξία και δυναμικότητα στο μαθησιακό περιβάλλον.

Η εκπαίδευση με Νέες Τεχνολογίες ήδη χαρακτηρίζεται από αρκετούς θεωρητικούς ως η εκπαίδευση του μέλλοντος, που θα βοηθήσει τους πολίτες όχι μόνο να επιτυγχάνουν βασικούς γνωστικούς στόχους αλλά και να εντάσσονται αρμονικά στην κοινωνία και να ανταποκρίνονται με επιτυχία στις μεταβαλλόμενες συνθήκες ζωής και εργασίας. Υπάρχει, βέβαια, μερίδα θεωρητικών που αμφισβητούν τα παραπάνω πλεονεκτήματα και υποστηρίζουν ότι είναι ορατός ο κίνδυνος μιας ταξικής διαίρεσης νέου τύπου στην κοινωνία. Ότι, δηλαδή, η ηλεκτρονική μάθηση προϋποθέτει πρώτα απ' όλα την δυνατότητα πρόσβασης στα τεχνολογικά μέσα, κάτι που ακόμα δεν αποτελεί πραγματικότητα για όλο τον κόσμο. Υπάρχει, λένε, ο κίνδυνος να περιθωριοποιηθούν όσοι παραμένουν για διάφορους λόγους τεχνολογικά αναλόγητοι και να καταλήξουμε στον ψηφιακό διίσμο της κοινωνίας. Για να αποτραπεί αυτός ο κίνδυνος, ωστόσο, προέκυψαν σημαντικές αλλαγές στα εκπαιδευτικά προγράμματα πολλών χωρών, ώστε οι ΤΠΕ να ενταχθούν στην πρωτοβάθμια εκπαίδευση και οι πολίτες του αύριο να είναι από μικρή ηλικία εξοικειωμένοι με τις τεχνολογικές αλλαγές.

Συμπερασματικά, διαπιστώνεται η επιβεβαίωση της θεωρίας της τεχνολογικής νομοτέλειας στην περίπτωση της εκπαίδευσης με Νέες Τεχνολογίες. Κάθε τεχνολογική μεταβολή που οδηγούσε στην εμφάνιση ενός νέου επικοινωνιακού μέσου προκαλούσε διαφοροποιήσεις στη δομή και τη λειτουργία του επικοινωνιακού μοντέλου και αυτές με τη σειρά τους σημαντικές μεταβολές στην εκπαίδευση με Νέες Τεχνολογίες. Η ταχύτατη εξάπλωση της τεχνολογίας των υπολογιστών και του διαδικτύου, που στην παρούσα υπόθεση εργασίας επιλέχθηκε ως ανεξάρτητη μεταβλητή, σαφέστατα διαφοροποίησε την εκπαιδευτική πρακτική και πολιτική στο χώρο της εκπαίδευσης με Νέες Τεχνολογίες (ενδιάμεση μεταβλητή), που με τη σειρά της επέφερε σημαντικές ανακατατάξεις στην κοινωνία (εξαρτημένη μεταβλητή), αναφορικά με τη διάχυση και την προσβασιμότητα της γνώσης, τη διάνοιξη οριζόντων σε επαγγελματικό επίπεδο και κατά συνέπεια την ετοιμότητα και τη δυνατότητα άμεσης προσαρμογής και ανταπόκρισης των πολιτών στις αυξημένες απαιτήσεις της παγκοσμιοποίησης. Τέλος, αξίζει να επισημανθεί ότι αν και υπάρχει σαφής αλληλεπίδραση τεχνολογίας – κοινωνίας και η σχέση είναι αμφίδρομη, στην παρούσα υπόθεση εργασίας καταδεικνύεται ότι οι τεχνολογικές μεταβολές είναι εκείνες που προηγούνται και υπερέχουν στην ταχύτητα και την ένταση επίδρασής τους έναντι των κοινωνικών αλλαγών και επιδράσεων.

Η παρούσα εργασία επιδέχεται βελτιώσεις και αλλαγές, όμως είναι η βάση για μια μελλοντική αξιοποίηση μελλοντικών διαδικτυακών εφαρμογών που μπορούν να χρησιμοποιηθούν τόσο για εκπαίδευση όσο και για αξιολόγηση των σπουδαστών.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- 1) Κόμης Β. (2000), Η έννοια του διδακτικού μετασχηματισμού στη διδακτική της Πληροφορικής, Η Βάση, 2, 23-34.
- 2) Κόμης Β. (2001), Μελέτη βασικών εννοιών του προγραμματισμού στο πλαίσιο μιας οικοδομηστικής διδακτικής προσέγγισης, Θέματα στην Εκπαίδευση, 2/3, 243-270.
- 3) Κόμης Β. & Μικρόπουλος Τ. Α. (2001), Πληροφορική στην Εκπαίδευση, Ελληνικό Ανοικτό Πανεπιστήμιο.
- 4) http://en.wikipedia.org/wiki/Who_Wants_to_Be_a_Millionaire%3F : Πληροφορίες για το Παιχνίδι
- 5) http://en.wikipedia.org/wiki/C_%28programming_language%29 : Πληροφορίες για την C
- 6) <http://en.wikipedia.org/wiki/C%2B%2B> : Πληροφορίες για την C++
- 7) http://en.wikipedia.org/wiki/Java_%28programming_language%29 : Πληροφορίες για την Java
- 8) http://en.wikipedia.org/wiki/Active_Server_Pages : Πληροφορίες για την ASP
- 9) <http://en.wikipedia.org/wiki/ASP.NET> : Πληροφορίες για την ASP.net
- 10) http://en.wikipedia.org/wiki/JavaServer_Pages : Πληροφορίες για την JSP
- 11) <http://en.wikipedia.org/wiki/PHP> : Πληροφορίες για την PHP
- 12) http://en.wikipedia.org/wiki/Unified_Process : Πληροφορίες για την Ενοποιητική Διαδικασία 1/2
- 13) http://en.wikipedia.org/wiki/IBM_Rational_Unified_Process : Πληροφορίες για την Ενοποιητική Διαδικασία 2/2
- 14) <http://xkcd.com/327/> : Υπενθύμιση για τον έλεγχο εισαγόμενων στοιχείων από τον χρήστη.
- 15) <http://www.w3schools.org> : Βοήθεια στην χρήση της HTML
- 16) <http://www.daniweb.com> : Βοήθεια στην εκμάθηση της JSP
- 17) <http://www.roseindia.net> : Βοήθεια στην χρήση της JSP
- 18) <http://www.jsptut.com> : Βοήθεια στην χρήση της JSP
- 19) <http://stackoverflow.com> : Βοήθεια στην χρήση της Java
- 20) <http://docs.oracle.com/javaee/5/api/index.html> : Το εγχειρίδιο της JavaEE
- 21) <http://docs.oracle.com/javase/1.5.0/docs/api/index.html> : Το εγχειρίδιο της JavaSE

Παράρτημα

Η διαδικτυακή εφαρμογή «Ποιος θέλει να γίνει εκατομμυριούχος;» ως μέσο εκπαίδευσης και αξιολόγησης των μαθητών

Λαμπράκης Κωνσταντίνος

Περίληψη

Από την πληθώρα των μέσων που προσφέρει σήμερα η εκπαιδευτική τεχνολογία, σημαντική θέση κατέχει ο ηλεκτρονικός υπολογιστής, του οποίου η εισαγωγή και χρήση στην εκπαίδευση όσον αφορά τη διδακτική υποστήριξη διαφόρων αντικειμένων, βρίσκεται ακόμη σε πρώιμα στάδια. Επομένως ο χώρος προσφέρεται για δημιουργική ενασχόληση και ειδικότερα για την διερεύνηση των δυνατοτήτων συμβολής της πληροφορικής στην εποπτικοποίηση της διδακτικής πράξης, γεγονός που αποτελεί συστηματική επιδίωξη της σύγχρονης διδακτικής. Βασικός στόχος της συγκεκριμένης πτυχιακής εργασίας είναι ο σχεδιασμός και η υλοποίηση μιας διαδικτυακής εκπαιδευτικής εφαρμογής, η οποία εκτός μέσο εκπαίδευσης μπορεί να χρησιμοποιηθεί και ως μέσο αξιολόγησης.

Λέξεις Κλειδιά: «Ποιος θέλει να γίνει εκατομμυριούχος», διαδικτυακή εφαρμογή, εποπτικοποίηση διδακτικής πράξης

Εισαγωγή

Σκοπός της παρούσας μελέτης ήταν η μελέτη, η σχεδίαση και η ανάπτυξη μιας διαδικτυακής εφαρμογής βασισμένης στο γνωστό παιχνίδι «Ποιος θέλει να γίνει Εκατομμυριούχος». Μέσα από τη συγκεκριμένη μελέτη καθορίστηκαν οι ανάγκες σχεδίασης νέων διαδικτυακών εφαρμογών εκπαιδευτικού χαρακτήρα, οι οποίες έδωσαν τη δυνατότητα στη διαδικτυακή εφαρμογή να υποστηρίξει την παρουσίαση και τη διάχυση πληροφοριών που αφορούν την εκπαιδευτική διαδικασία.

Πιο συγκεκριμένα για τις ανάγκες υλοποίησης της εφαρμογής χρησιμοποιήθηκε ένας server και μια βάση δεδομένων ώστε να καταγράφονται οι χρήστες και οι ερωτήσεις του παιχνιδιού. Από κατασκευαστικής άποψης χρησιμοποιήθηκε server-side scripting. Καθώς η εφαρμογή είναι απομακρυσμένη και δεν εγκαθίσταται από το χρήστη απαιτείται υψηλή ταχύτητα ανταπόκρισης. Είναι μια εφαρμογή server-side, η οποία δέχεται τα δεδομένα του χρήστη, τα επεξεργάζεται και του επιστρέφει τα τελικά αποτελέσματα. Αναλυτικά: η εφαρμογή δέχεται την απάντηση που επέλεξε ο χρήστης, ελέγχει εάν είναι η σωστή απάντηση για την συγκεκριμένη ερώτηση και επιστρέφει το τελικό αποτέλεσμα. Οι γλώσσες που επιλέχθηκαν ήταν οι: ASP, Java, MySQL.

Η διαδικτυακή εφαρμογή που αναπτύχθηκε, με κεντρικό θέμα την εκπαίδευση, δίνει τη δυνατότητα στο χρήστη να εγγραφεί σε αυτή και να μεταβεί στο παιχνίδι. Κατά τη διάρκεια του παιχνιδιού ο χρήστης καλείται να απαντήσει σε 15 ερωτήσεις με κλιμακούμενο επίπεδο δυσκολίας. Κάθε ερώτηση, αντιστοιχεί σε ένα συγκεκριμένο χρηματικό ποσό. Ο χρήστης καλείται να επιλέξει ανάμεσα από 4 διαφορετικές πιθανές απαντήσεις την σωστή. Εάν απαντήσει σωστά, ο παίκτης κερδίζει το αντίστοιχο ποσό χρημάτων και προχωράει στην επόμενη ερώτηση. Με κάθε σωστή απάντηση ο παίκτης μπορεί να κερδίσει και μεγαλύτερο ποσό ή να επιλέξει να αποχωρίσει και να κρατήσει το ποσό που έχει κερδίσει ως τότε. Όσο το ποσό των χρημάτων αυξάνει, τόσο ο βαθμός δυσκολίας των ερωτήσεων γίνεται μεγαλύτερος. Επίσης υπάρχουν 2 ερωτήσεις, η 5^η και η 10^η, που όταν απαντηθούν σωστά εξασφαλίζουν στον παίκτη την εξασφάλιση ενός συγκεκριμένου ποσού. Το ποσό των χρημάτων έχει το ρόλο της ανταμοιβής για τη σωστή ανταπόκριση στις ερωτήσεις.

Διδακτική και Νέες Τεχνολογίες

Η πρόοδος στη *Διδακτική των Επιστημών* είναι ραγδαία κατά την τελευταία εικοσαετία και συνιστάται κυρίως στη δημιουργία ενός θεωρητικού πλαισίου που βασίζεται στις εποικοδομητικές και κοινωνικοπολιτισμικές ψυχολογικές θεωρίες. Οι πρώτες προσπάθειες δημιουργίας ενός θεωρητικού πλαισίου που να ξεφεύγει από τον εμπειρισμό ξεκίνησαν από το χώρο της διδακτικής των μαθηματικών. Το θεωρητικό αυτό πλαίσιο γνώρισε αρκετές διακυμάνσεις, αλλά τελικά κατάφερε να παγιώσει μια σειρά από βασικές έννοιες και μια μεθοδολογική προσέγγιση που είναι πλέον αποδεκτές στις μέρες μας από το σύνολο σχεδόν των ερευνητών στο χώρο της Διδακτικής των Επιστημών.

Μπορούμε να διακρίνουμε δύο μεγάλους άξονες αυτού του θεωρητικού πλαισίου:

- Την επιστημολογία των γνώσεων
- Τη γένεση και την πρόσκτηση των γνώσεων από τα υποκείμενα που βρίσκονται σε κατάσταση μάθησης και την τοποθέτηση αυτής της γένεσης μέσα σε πραγματικές σχολικές καταστάσεις.

Βασικό πεδίο μελέτης της διδακτικής συνιστά καταρχήν ο τομέας της ανάπτυξης των περιεχομένων. Ο τομέας αυτός μελετάται κάτω από ποικίλα πρίσματα. Αφενός κάτω από το πρίσμα του διδακτικού μετασχηματισμού και των κοινωνικών πρακτικών αναφοράς και αφετέρου κάτω από το πρίσμα των διαδικασιών συγκρότησης σχετικών με το πλαίσιο προγράμματος σπουδών και το αναλυτικό πρόγραμμα, όπως επίσης και με το παραγόμενο διδακτικό υλικό δηλαδή τα σχολικά βιβλία και εγχειρίδια, το βιβλίο καθηγητή καθώς και το εκπαιδευτικό υλικό και λογισμικό.

Στη συνέχεια, βασικό άξονα μελέτης αποτελεί ο τομέας των στρατηγικών της οικοδόμησης των γνώσεων και της κατανόησης των διαδικασιών της μάθησης, ο οποίος κατά κανόνα αφορά το πλαίσιο σχολικών συνθηκών. Τόσο η κατανόηση των στρατηγικών, όσο και η μελέτη των διαδικασιών μάθησης, απαιτούν τη μελέτη των αναπαραστάσεων, της εννοιολογικής αλλαγής, και των διαδικασιών επίλυσης προβλημάτων. Σημαντικός τομέας είναι, επίσης, αυτός της οικοδόμησης διδακτικών καταστάσεων που αφορά στο διδακτικό συμβόλαιο στην επιμόρφωση των εκπαιδευτικών αλλά και στη συγκρότηση του διδακτικού υλικού.

Ο τομέας των διδακτικών αλληλεπιδράσεων, όπως η διδακτική βοήθεια, η γνωστική και η κοινωνικογνωστική σύγκρουση, καθώς και ο τομέας των χρησιμοποιούμενων μέσων που διαμεσολαβούν αυτές τις αλληλεπιδράσεις είναι άλλος ένας σημαντικός άξονας μελέτης της διδακτικής. Κάτω από το πρίσμα αυτό, η ύπαρξη ενός τεχνολογικού μέσου όπως ο υπολογιστής φαίνεται να δρα καταλυτικά κατά τη διδασκαλία της πληροφορικής ή κατά τη διδασκαλία με χρήση των τεχνολογιών της πληροφορίας και των επικοινωνιών.

Ο τομέας αυτός, μάλιστα, είναι ιδιαίτερα σημαντικός στην περιοχή της διδακτικής της πληροφορικής, αφού είναι γνωστό ότι κύριο ρόλο στη διδακτική πράξη διαδραματίζει η αλληλεπίδραση του μαθητή με το τεχνικό μέσο και οι παρεμβάσεις που παρέχει ο εκπαιδευτικός.

Τέλος, δύο σημαντικοί άξονες μελέτης αποτελούν αφενός τα σύγχρονα προγράμματα σπουδών και αφετέρου οι στάσεις και αντιλήψεις των εκπαιδευτικών πληροφορικής. Ο τομέας ανάπτυξης σύγχρονου προγράμματος σπουδών και κατάλληλου εκπαιδευτικού υλικού αποτελεί βασικό πεδίο μελέτης της διδακτικής. Τόσο ο σχεδιασμός των προγραμμάτων σπουδών, όσο και του εκπαιδευτικού υλικού, πρέπει να λαμβάνουν υπόψη τους και να βασίζονται στα αποτελέσματα των ερευνών που αφορούν όλες τις προηγούμενες κατηγορίες.

Ιδιαίτερα σημαντικός είναι επίσης ο τομέας των στάσεων και των αντιλήψεων των εκπαιδευτικών Πληροφορικής, αλλά και όλων των εκπαιδευτικών που χρησιμοποιούν τις ΤΠΕ στην εκπαιδευτική τους πρακτική, αφού άπτεται άμεσα της οργάνωσης των διδακτικών παρεμβάσεων και των σχολικών πρακτικών.

Ωστόσο, στους χώρους της διδακτικής και της εκπαιδευτικής ψυχολογίας δεν υπάρχει μια ενιαία θεωρία που να περιγράφει τις διαδικασίες μάθησης.

Υπάρχουν τρία μεγάλα ψυχολογικά ρεύματα που έχουν διαμορφώσει αντίστοιχα θεωρητικά μοντέλα ικανά να περιγράψουν ή και να εξηγήσουν, στον έναν ή στον άλλο βαθμό, την σχολική μάθηση και τις αντίστοιχες διδακτικές παρεμβάσεις:

- Ο συμπεριφορισμός (behaviorism), που θεωρεί ότι η γνώση μεταδίδεται

- Ο εποικοδομητισμός (constructivism), που θεωρεί ότι η γνώση είναι προϊόν προσωπικής οικοδόμησης μέσω της αλληλεπίδρασης με το περιβάλλον
- Και οι κοινωνικοπολιτισμικές (sociocultural) ή ιστορικοπολιτισμικές (historicocultural) προσεγγίσεις, που αντιλαμβάνονται την γνώση και την μάθηση ως παράγωγα της κοινωνικής αλληλεπίδρασης και του πολιτισμού.

Στη διδακτική, ο εποικοδομητισμός αφορά τις εκπαιδευτικές διαδικασίες οι οποίες τοποθετούν το μαθητή στο κέντρο της σχολικής μάθησης. Κάτω από το πρίσμα αυτό, το εποικοδομητικό μοντέλο αντιτίθεται στο μοντέλο μετάδοσης της γνώσης. Βασική διδακτική θέση του εποικοδομητισμού είναι ότι οι γνώσεις δεν μεταδίδονται. Οφείλουν να οικοδομηθούν ή να αναδομηθούν από το μαθητή, ο οποίος, είναι υπεύθυνος για τη μάθηση του. Ο μαθητής οικοδομεί τις γνώσεις του με βάση μια διερεύνηση της πραγματικότητας, η οποία συμπεριλαμβάνει και τη γνώση με τις διαφορές μορφές μετάδοσης της.

Τη γνώση αυτή την ιδιοποιείται με μη γραμμικό τρόπο, μέσω διαφοροποιήσεων, γενικεύσεων και ρήξεων. Η ιδιοποίηση αυτή της γνώσης βασίζεται πάνω σε εξατομικευμένες οικοδομήσεις αλλά και σε συλλογικές καταστάσεις τάξης, όπου εμφανίζονται γνωστικές συγκρούσεις, ικανές να προχωρήσουν την οικοδόμηση των γνώσεων.

Μεθοδολογία Υλοποίησης της εφαρμογής

Για το σχεδιασμό της διαδικτυακής εκπαιδευτικής εφαρμογής μελετήθηκαν τα βασικά στοιχεία του παιχνιδιού οι κανόνες και οι ερωτήσεις. Από εδώ και πέρα θα αναφέρεται το σύνολο των κανόνων του παιχνιδιού και οι ερωτήσεις σαν εφαρμογή ενώ ο παίκτης θα αναφέρεται σαν χρήστη. Το πεδίο εφαρμογής του παρόντος έργου αφορά την ανάπτυξη της σε πειραματικό / δοκιμαστικό επίπεδο, γεγονός που θα μεταφραστεί σε: περιορισμένο πλήθος χρηστών, μικρό αριθμό ταυτόχρονων χρήσεων και μικρό αριθμό διαθέσιμων ερωτήσεων.

Ο χρήστης θα πρέπει να μπορεί να παίξει το παιχνίδι, να δει το σκορ που έχει επιτύχει κάθε φορά και να δει σε ποια θέση βρίσκεται σε σχέση με τους άλλους χρήστες. Η εφαρμογή από την άλλη θα πρέπει να είναι σε θέση να αναγνωρίσει τους χρήστες, να διαχειριστεί τις ερωτήσεις και να προβάλει ερωτήσεις διαφορετικού επιπέδου δυσκολίας ανάλογα με το πόσο έχει προχωρήσει ο χρήστης στο παιχνίδι.

Η εφαρμογή θα ακολουθήσει το γενικό μοντέλο server-client με τον server να αναλαμβάνει το μεγαλύτερο μέρος του φόρτου εργασίας της εφαρμογής και με τον client να αναλαμβάνει την απεικόνιση των στοιχείων που στέλνει ο server, στον χρήστη.

Η μεθοδολογία που χρησιμοποιείται στο μοντέλο της Ενοποιητικής Διαδικασίας, (Unified Process) κρίθηκε ως η καταλληλότερη για τη συγκεκριμένη διαδικτυακή εκπαιδευτική εφαρμογή. Η μέθοδος αναλύεται σε 4 βασικά στάδια:

- *Στάδιο της σύλληψης.* Το στάδιο της σύλληψης θεσπίζει το επιχειρησιακό κομμάτι του συστήματος, δηλαδή για ποιόν λόγο πρέπει να αναλάβουμε αυτό το εγχείρημα, ποιο το πεδίο εφαρμογής του, ποιες είναι οι βασικές απαιτήσεις του και με ποιους τρόπους μπορεί να υλοποιηθεί.
- *Στάδιο της επεξεργασίας.* Ο κύριος στόχος είναι να μετριάσει τα βασικά στοιχεία των κινδύνων που προσδιορίζονται από την ανάλυση του έργου έως το τέλος αυτού του σταδίου. Στο στάδιο επεξεργασίας είναι εκεί που το έργο παίρνει μορφή. Σε αυτό το στάδιο γίνεται η ανάλυση του τομέα των προβλημάτων και η αρχιτεκτονική του έργου παίρνει την βασική της μορφή.
- *Στάδιο της υλοποίησης.* Η υλοποίηση είναι το μεγαλύτερο στάδιο του έργου. Σε αυτό το στάδιο το υπόλοιπο του έργου 'χτίζεται' πάνω στα θεμέλια που έβαλε το στάδιο της επεξεργασίας. Τα στοιχεία του συστήματος υλοποιούνται με μία σειρά από σύντομες και χρονικά περιορισμένες επαναλήψεις. Από κάθε επανάληψη προκύπτει μια λειτουργική έκδοση του προγράμματος. Είναι σύνηθες να γράφονται πλήρη σενάρια χρήσης κατά την

διάρκεια του σταδίου της υλοποίησης τα οποία στη συνέχεια γίνονται η αρχή μιας νέας επανάληψης.

- **Στάδιο της μετάβασης.** Στην φάση της μετάβασης το έργο παραδίδεται στους χρήστες. Τα σχόλια και οι παρατηρήσεις των χρηστών από αυτή την αρχική παράδοση έχουν σαν αποτέλεσμα την περαιτέρω βελτίωση του έργου σε βάθος πολλαπλών επαναλήψεων αυτής της φάσης. Η φάση μετάβασης επίσης περιλαμβάνει τυχόν μετατροπές του έργου και την εκπαίδευση των χρηστών

Λόγο της φύσης της εφαρμογής υποχρεωτικά θα υπάρχει ένας συνδυασμός αρχιτεκτονικών. Συγκεκριμένα υπάρχουν τρεις κύριες κατηγορίες:

- Η αρχιτεκτονική της βάσης δεδομένων που θα διαχειρίζεται τις ερωτήσεις με τις απαντήσεις τους, τα στοιχεία των χρηστών και κάποια γενικά στατιστικά.
- Η αρχιτεκτονική της γλώσσας προγραμματισμού που θα χρησιμοποιηθεί για την υλοποίηση της λογικής του παιχνιδιού και τέλος
- Η γλώσσα προγραμματισμού που θα αναλάβει την απεικόνιση των στοιχείων της εφαρμογής στον χρήστη.

Σχέδιο Δράσης

Η εφαρμογή μας αποτελείται από τρία κυρίως τμήματα, την Βάση Δεδομένων όπου αποθηκεύονται όλες οι πληροφορίες σχετικά με τους Χρήστες, της Ερωτήσεις και το Ιστορικό των Χρηστών, την λογική της εφαρμογής η οποία εκτελείται στον εξυπηρετητή που φιλοξενεί την εφαρμογή μας και αναλαμβάνει το μεγαλύτερο μέρος της επεξεργασίας, αν όχι όλο, των δεδομένων από τους χρήστες και την διεπαφή της εφαρμογής μας, που στην προκειμένη περίπτωση είναι διαδραστικές ιστοσελίδες, η οποία αναλαμβάνει την απεικόνιση των δεδομένων, που στέλνει ο εξυπηρετητής, μέσο κάποιου προγράμματος περιήγησης του διαδικτύου στον υπολογιστή του χρήστη.

Από τα παραπάνω είναι προφανές ότι οι λύσεις που επιλέγουμε για κάθε κομμάτι του συστήματος πρέπει να μπορεί να συνεργαστεί τουλάχιστον με ένα από τα άλλα δύο. Από τις υποψήφιες τεχνολογίες μπορούμε εύκολα να παραβλέψουμε και από μία υποψήφια επιλογή σε κάθε τμήμα λόγω παλαιότητας και λόγω εύρους χρήσης. Προσπερνάμε την γλώσσα προγραμματισμού C μιας και αποτελεί υποσύνολο της C++ όπως παραβλέπουμε και την ASP γιατί έχει αντικατασταθεί πλέον από την ASP.NET. Τέλος όσο αφορά τις υποψηφιότητες για την βάση δεδομένων πρέπει να παραβλέψουμε την Microsoft Access γιατί και οι δύο άλλες υποψήφιες τεχνολογίες μπορούν να διαχειριστούν πολύ μεγαλύτερους όγκους δεδομένων και πολλές περισσότερες συνδέσεις χρηστών ταυτόχρονα.

Έχοντας περιορίσει πλέον της υποψήφιες λύσεις μας σε δύο για την βάση δεδομένων, δύο για την γλώσσα προγραμματισμού της λογικής του συστήματος και τρεις για τον προγραμματισμό της διεπαφής του χρήστη είναι πλέον ευκολότερο να καταλήξουμε στις τεχνολογίες που θα χρησιμοποιήσουμε για την υλοποίηση του συστήματος. Για την επιλογή του συστήματος διαχείρισης της βάσης δεδομένων τα πράγματα είναι σχετικά εύκολα από την στιγμή που και οι δύο λύσεις μας προσφέρουν περίπου τα ίδια πράγματα αλλά έχουν και μερικές σημαντικές διαφορές. Η MySQL είναι κυρίως δωρεάν και μπορεί να εγκατασταθεί σε όλα τα βασικά λειτουργικά συστήματα (Windows, Linux και Mac OS). Από την άλλη το Microsoft SQL Server πρέπει κυρίως να αγοραστεί και μπορεί να εγκατασταθεί μόνο στο λειτουργικό περιβάλλον των Microsoft Windows. Ως εκ τούτου για το σύστημα διαχείρισης της βάσης δεδομένων μας επιλέγουμε την MySQL και ποιο συγκεκριμένα την MySQL Community Server επειδή διατίθεται δωρεάν και μπορεί να εγκατασταθεί σε, πρακτικά, οποιοδήποτε λειτουργικό σύστημα.

Με την επιλογή του συστήματος της βάσης δεδομένων που χρησιμοποιούμε, μας μένει να προσδιορίσουμε τις γλώσσες που χρησιμοποιούνται για τον προγραμματισμό της λογικής της εφαρμογής και της διεπαφής του χρήστη. Από τα πιθανά ζεύγη λύσεων, το ζεύγος Java και JSP ξεχωρίζει γιατί στην ουσία πρόκειται για την ίδια γλώσσα προγραμματισμού κάτι που μας

διευκολύνει στην πορεία γιατί έχουμε να ασχοληθούμε με την λογική και της ιδιοτροπίες μόνο μίας γλώσσας έναντι δύο.

Αποτελέσματα

Σήμερα, η εκπαίδευση με Νέες Τεχνολογίες έχει ενταχθεί στη ζωή μας, καθιστώντας τη μάθηση εφικτή έξω από το χώρο, το χρόνο, την ηλικία, την οικογενειακή ή επαγγελματική κατάσταση, το επίπεδο σπουδών, την πνευματική ή σωματική αρτιμέλεια ή ιδιαιτερότητα, το φύλο ή τη φυλή, τα ενδιαφέροντα. Παρέχει ευκαιρίες για εκπαίδευση και μόρφωση σε όλους, ανεξάρτητα από περιορισμούς και διακρίσεις. Αποδεσμεύει την εκπαιδευτική κοινότητα από τα ασφυκτικά χωροχρονικά πλαίσια διδασκαλίας, δίνοντας την ευκαιρία στον καθένα να καθορίσει ο ίδιος το ρυθμό, το χρόνο, το χώρο, τον τρόπο μάθησης που τον εξυπηρετεί. Συμβάλλει στην εξοικονόμηση πόρων και για τους φορείς αλλά και για τους πολίτες, καθώς τα μέσα και τα υλικά που απαιτούνται είναι ελάχιστα, εύχρηστα και συχνά αρκετά πιο οικονομικά. Προωθεί την ανάπτυξη των ιδιαίτερων ενδιαφερόντων και κλίσεων μέσα από τη δυνατότητα επιλογής μιας πολύ μεγάλης γκάμας εκπαιδευτικών προγραμμάτων. Σέβεται την ατομικότητα ενώ παράλληλα παρωθεί στην ενίσχυση της συνεργασίας και της αλληλεπίδρασης των συμμετεχόντων. Συμπληρώνει και εμπλουτίζει τις παραδοσιακές μεθόδους εκπαίδευσης, προσδίδοντας ευελιξία και δυναμικότητα στο μαθησιακό περιβάλλον.

Η εκπαίδευση με Νέες Τεχνολογίες ήδη χαρακτηρίζεται από αρκετούς θεωρητικούς ως η εκπαίδευση του μέλλοντος, που θα βοηθήσει τους πολίτες όχι μόνο να επιτυγχάνουν βασικούς γνωστικούς στόχους αλλά και να εντάσσονται αρμονικά στην κοινωνία και να ανταποκρίνονται με επιτυχία στις μεταβαλλόμενες συνθήκες ζωής και εργασίας. Υπάρχει, βέβαια, μερίδα θεωρητικών που αμφισβητούν τα παραπάνω πλεονεκτήματα και υποστηρίζουν ότι είναι ορατός ο κίνδυνος μιας ταξικής διαίρεσης νέου τύπου στην κοινωνία. Ότι, δηλαδή, η ηλεκτρονική μάθηση προϋποθέτει πρώτα απ' όλα την δυνατότητα πρόσβασης στα τεχνολογικά μέσα, κάτι που ακόμα δεν αποτελεί πραγματικότητα για όλο τον κόσμο. Υπάρχει, λένε, ο κίνδυνος να περιθωριοποιηθούν όσοι παραμένουν για διάφορους λόγους τεχνολογικά αναλφάβητοι και να καταλήξουμε στον ψηφιακό δυϊσμό της κοινωνίας. Για να αποτραπεί αυτός ο κίνδυνος, ωστόσο, προέκυψαν σημαντικές αλλαγές στα εκπαιδευτικά προγράμματα πολλών χωρών, ώστε οι ΤΠΕ να ενταχθούν στην πρωτοβάθμια εκπαίδευση και οι πολίτες του αύριο να είναι από μικρή ηλικία εξοικειωμένοι με τις τεχνολογικές αλλαγές.

Συμπερασματικά, διαπιστώνεται η επιβεβαίωση της θεωρίας της τεχνολογικής νομοτέλειας στην περίπτωση της εκπαίδευσης με Νέες Τεχνολογίες. Κάθε τεχνολογική μεταβολή που οδηγούσε στην εμφάνιση ενός νέου επικοινωνιακού μέσου προκαλούσε διαφοροποιήσεις στη δομή και τη λειτουργία του επικοινωνιακού μοντέλου και αυτές με τη σειρά τους σημαντικές μεταβολές στην εκπαίδευση με Νέες Τεχνολογίες. Η ταχύτατη εξάπλωση της τεχνολογίας των υπολογιστών και του διαδικτύου, που στην παρούσα υπόθεση εργασίας επιλέχθηκε ως ανεξάρτητη μεταβλητή, σαφέστατα διαφοροποίησε την εκπαιδευτική πρακτική και πολιτική στο χώρο της εκπαίδευσης με Νέες Τεχνολογίες (ενδιάμεση μεταβλητή), που με τη σειρά της επέφερε σημαντικές ανακατατάξεις στην κοινωνία (εξαρτημένη μεταβλητή), αναφορικά με τη διάχυση και την προσβασιμότητα της γνώσης, τη διάνοιξη οριζόντων σε επαγγελματικό επίπεδο και κατά συνέπεια την ετοιμότητα και τη δυνατότητα άμεσης προσαρμογής και ανταπόκρισης των πολιτών στις αυξημένες απαιτήσεις της παγκοσμιοποίησης. Τέλος, αξίζει να επισημανθεί ότι αν και υπάρχει σαφής αλληλεπίδραση τεχνολογίας – κοινωνίας και η σχέση είναι αμφίδρομη, στην παρούσα υπόθεση εργασίας καταδεικνύεται ότι οι τεχνολογικές μεταβολές είναι εκείνες που προηγούνται και υπερέχουν στην ταχύτητα και την ένταση επίδρασής τους έναντι των κοινωνικών αλλαγών και επιδράσεων.

Η παρούσα εργασία επιδέχεται βελτιώσεις και αλλαγές, όμως είναι η βάση για μια μελλοντική αξιοποίηση μελλοντικών διαδικτυακών εφαρμογών που μπορούν να χρησιμοποιηθούν τόσο για εκπαίδευση όσο και για αξιολόγηση των σπουδαστών.

Βιβλιογραφία

- 1) Κόμης Β. (2000), Η έννοια του διδακτικού μετασχηματισμού στη διδακτική της Πληροφορικής, Η Βάση, 2, 23-34.
- 2) Κόμης Β. (2001), Μελέτη βασικών εννοιών του προγραμματισμού στο πλαίσιο μιας οικοδομηστικής διδακτικής προσέγγισης, Θέματα στην Εκπαίδευση, 2/3, 243-270.
- 3) Κόμης Β. & Μικρόπουλος Τ. Α. (2001), Πληροφορική στην Εκπαίδευση, Ελληνικό Ανοικτό Πανεπιστήμιο.
- 4) http://en.wikipedia.org/wiki/Who_Wants_to_Be_a_Millionaire%3F : Πληροφορίες για το Παιχνίδι
- 5) http://en.wikipedia.org/wiki/C_%28programming_language%29 : Πληροφορίες για την C
- 6) <http://en.wikipedia.org/wiki/C%2B%2B> : Πληροφορίες για την C++
- 7) http://en.wikipedia.org/wiki/Java_%28programming_language%29 : Πληροφορίες για την Java
- 8) http://en.wikipedia.org/wiki/Active_Server_Pages : Πληροφορίες για την ASP
- 9) <http://en.wikipedia.org/wiki/ASP.NET> : Πληροφορίες για την ASP.net
- 10) http://en.wikipedia.org/wiki/JavaServer_Pages : Πληροφορίες για την JSP
- 11) <http://en.wikipedia.org/wiki/PHP> : Πληροφορίες για την PHP
- 12) http://en.wikipedia.org/wiki/Unified_Process : Πληροφορίες για την Ενοποιητική Διαδικασία 1/2
- 13) http://en.wikipedia.org/wiki/IBM_Rational_Unified_Process : Πληροφορίες για την Ενοποιητική Διαδικασία 2/2
- 14) <http://xkcd.com/327/> : Υπενθύμιση για τον έλεγχο εισαγόμενων στοιχείων από τον χρήστη.
- 15) <http://www.w3schools.org> : Βοήθεια στην χρήση της HTML
- 16) <http://www.daniweb.com> : Βοήθεια στην εκμάθηση της JSP
- 17) <http://www.roseindia.net> : Βοήθεια στην χρήση της JSP
- 18) <http://www.jsptut.com> : Βοήθεια στην χρήση της JSP
- 19) <http://stackoverflow.com> : Βοήθεια στην χρήση της Java
- 20) <http://docs.oracle.com/javase/5/api/index.html> : JavaTM 2 Platform Enterprise Edition, v 5.0 API Specifications : Το εγχειρίδιο της JavaEE
- 21) <http://docs.oracle.com/javase/1.5.0/docs/api/index.html> : JavaTM 2 Platform Standard Edition 5.0 API Specification : Το εγχειρίδιο της JavaSE



Υλοποίηση του παιχνιδιού «Ποιος θέλει να γίνει εκατομμυριούχος;» ως διαδικτυακή εφαρμογή

Το θέμα επιμελήθηκε ο σπουδαστής:
Λαμπράκης Κωνσταντίνος

Σκοπός

Σήμερα, η εκπαίδευση με Νέες Τεχνολογίες έχει ενταχθεί στη ζωή μας, καθιστώντας τη μάθηση εφικτή έξω από το χώρο και το χρόνο του παραδοσιακού εκπαιδευτηρίου.

Οι νέες τεχνολογίες συμπληρώνουν και εμπλουτίζουν τις παραδοσιακές μεθόδους εκπαίδευσης, προσδίδοντας ευελιξία και δυναμικότητα στο μαθησιακό περιβάλλον.

Βασικός στόχος της συγκεκριμένης πτυχιακής εργασίας είναι ο σχεδιασμός και η υλοποίηση μιας διαδικτυακής εκπαιδευτικής εφαρμογής, η οποία μέσω του παιχνιδιού μπορεί εκτός από μέσο εκπαίδευσης να χρησιμοποιηθεί και ως μέσο αξιολόγησης.

Τι χρησιμοποιήθηκε στην εφαρμογή

Για τις ανάγκες υλοποίησης της εφαρμογής χρησιμοποιήθηκε μια βάση δεδομένων ώστε να καταγράφονται:

- οι χρήστες,
- οι ερωτήσεις του παιχνιδιού
- το ιστορικό των χρηστών
- ένας server

Από κατασκευαστικής άποψης χρησιμοποιήθηκε server side scripting.

Η εφαρμογή δέχεται την ενέργεια που επέλεξε ο χρήστης, την επεξεργάζεται και του επιστρέφει το τελικό αποτέλεσμα.

Οι τεχνολογίες που επιλέχθηκαν ήταν οι: **MySQL**, **Java** και **JSP**.

Περιεχόμενο διαδικτυακής εφαρμογής

Η διαδικτυακή εφαρμογή αναπτύχθηκε με κεντρικό σκοπό να δώσει την δυνατότητα στον χρήστη να παίξει μια έκδοση του παιχνιδιού «Ποιος θέλει να γίνει εκατομμυριούχος».

Κατά τη διάρκεια του παιχνιδιού ο χρήστης καλείται να απαντήσει σε 15 ερωτήσεις με κλιμακούμενο επίπεδο δυσκολίας.

Κάθε ερώτηση, αντιστοιχεί σε ένα συγκεκριμένο χρηματικό ποσό. Ο χρήστης καλείται να επιλέξει ανάμεσα από 4 διαφορετικές πιθανές απαντήσεις την σωστή. Εάν απαντήσει σωστά, ο παίκτης κερδίζει το αντίστοιχο ποσό και προχωράει στην επόμενη ερώτηση. Με κάθε σωστή απάντηση ο παίκτης μπορεί να κερδίσει και μεγαλύτερο ποσό ή να αποχωρίσει και να κρατήσει το ποσό που έχει κερδίσει ως τότε.

Όσο το ποσό των χρημάτων αυξάνει, τόσο ο βαθμός δυσκολίας των ερωτήσεων γίνεται μεγαλύτερος. Επίσης υπάρχουν 2 ερωτήσεις ,η 5^η και η 10^η, που όταν απαντηθούν σωστά εξασφαλίζουν στον παίκτη την εξασφάλιση ενός ελάχιστου ποσού.

Μεθοδολογία Εργασίας

Η μεθοδολογία που περιγράφεται στο μοντέλο της Ενοποιητικής Διαδικασίας (Unified Process) κρίθηκε ως η καταλληλότερη για τη συγκεκριμένη διαδικτυακή εφαρμογή

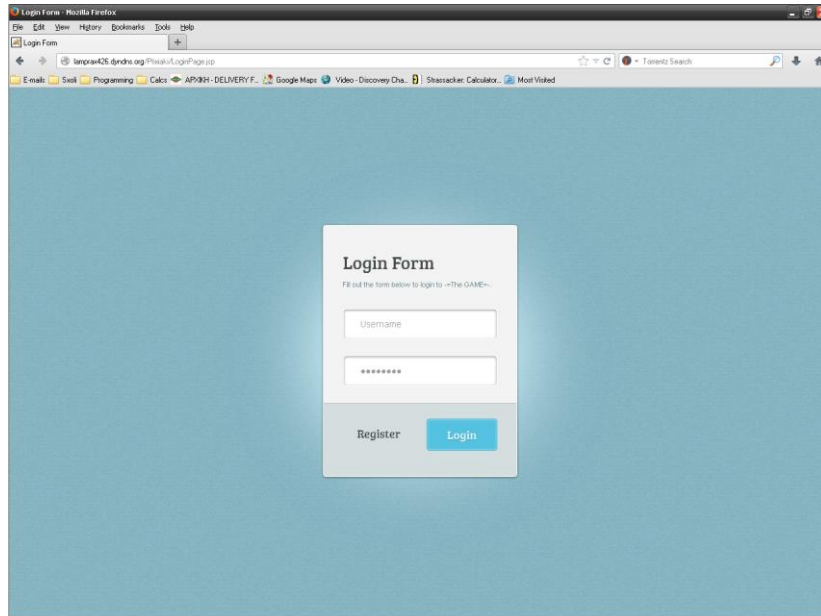
- **Στάδιο της σύλληψης**. Το οποίο θεσπίζει το επιχειρησιακό κομμάτι του συστήματος,.
- **Στάδιο της επεξεργασίας**. Ο κύριος στόχος είναι να μετριάσει τα βασικά στοιχεία των κινδύνων που προσδιορίζονται από την ανάλυση του έργου έως το τέλος αυτού του σταδίου. Σε αυτό το στάδιο γίνεται η ανάλυση του τομέα των προβλημάτων και η αρχιτεκτονική του έργου παίρνει την βασική της μορφή.
- **Στάδιο της υλοποίησης**. Σε αυτό το στάδιο το υπόλοιπο του έργου 'χτίζεται' πάνω στα θεμέλια που έβαλε το στάδιο της επεξεργασίας.
- **Στάδιο της μετάβασης**. Στο στάδιο αυτό το έργο παραδίδεται στους χρήστες. Τα σχόλια και οι παρατηρήσεις των χρηστών από αυτή την αρχική παράδοση έχουν σαν αποτέλεσμα την περαιτέρω βελτίωση του έργου σε βάθος πολλαπλών επαναλήψεων αυτού του σταδίου.

Τα τμήματα της εφαρμογής

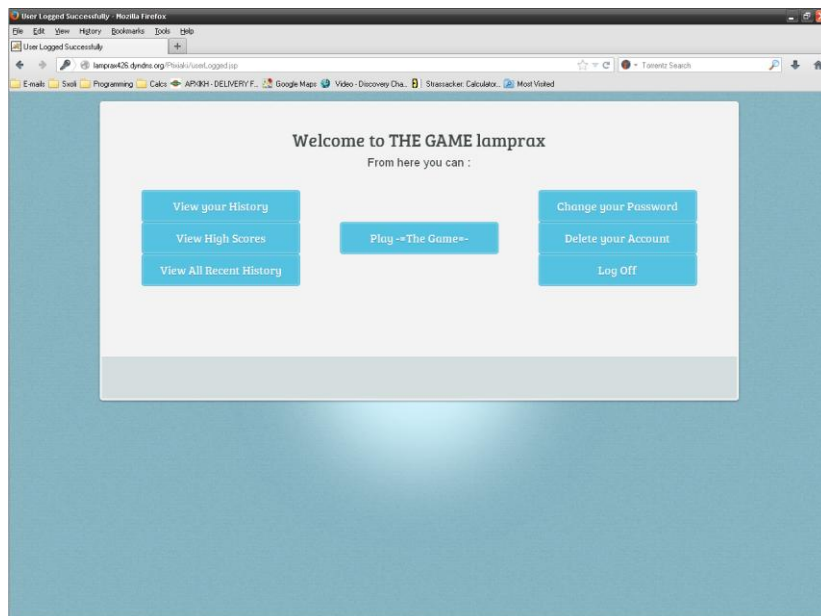
Η εφαρμογή μας αποτελείται από τρία κυρίως τμήματα:

- **Την Βάση Δεδομένων** όπου αποθηκεύονται όλες οι πληροφορίες σχετικά με τους Χρήστες, τις Ερωτήσεις και το Ιστορικό των Χρηστών,
- **Την λογική της εφαρμογής** η οποία εκτελείται στον εξυπηρετητή που φιλοξενεί την εφαρμογή μας και αναλαμβάνει το μεγαλύτερο μέρος της επεξεργασίας των δεδομένων από τους χρήστες και την βάση δεδομένων και
- **Την διεπαφή της εφαρμογής** μας, που στην προκειμένη περίπτωση είναι διαδραστικές ιστοσελίδες, η οποία αναλαμβάνει την απεικόνιση των δεδομένων από την λογική της εφαρμογής και την στέλνει μέσω του εξυπηρετητή σε όποιο πρόγραμμα περιήγησης του διαδικτύου χρησιμοποιεί ο χρήστης.

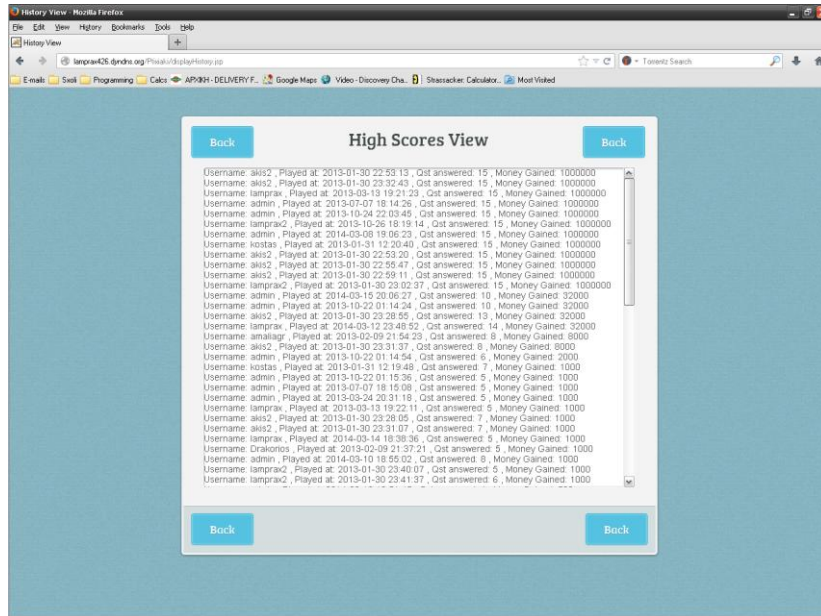
Είσοδος του χρήστη



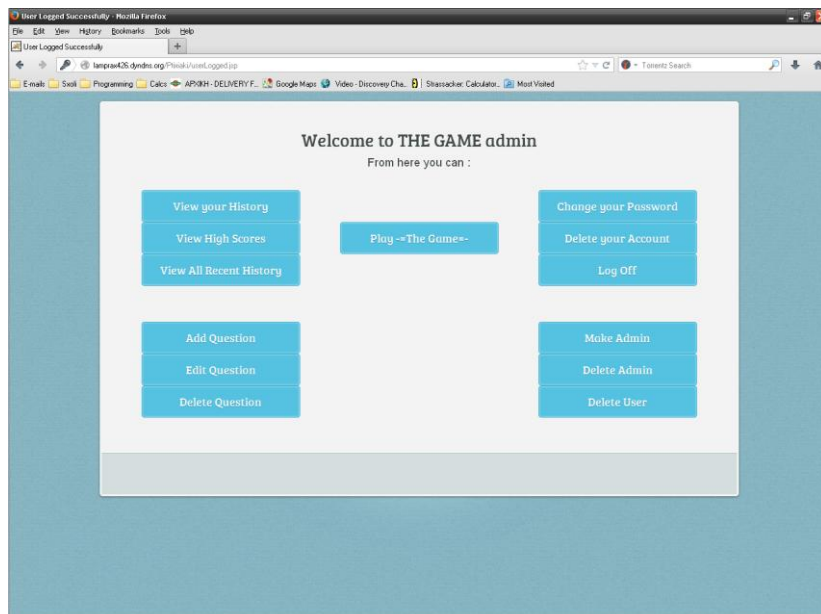
Κεντρική οθόνη χρήστη



Απεικόνιση καλύτερων βαθμολογιών



Κεντρική οθόνη διαχειριστή



Φόρμα προσθήκης Ερώτησης

ADD A QUESTION

OID: Difficulty Level:

Type the Question IN HERE

THIS IS THE CORRECT ANSWER This is the 1st wrong answer

This is the 2nd wrong answer This is the 3rd wrong answer

Φόρμα Επεξεργασίας Ερώτησης

EDIT A QUESTION

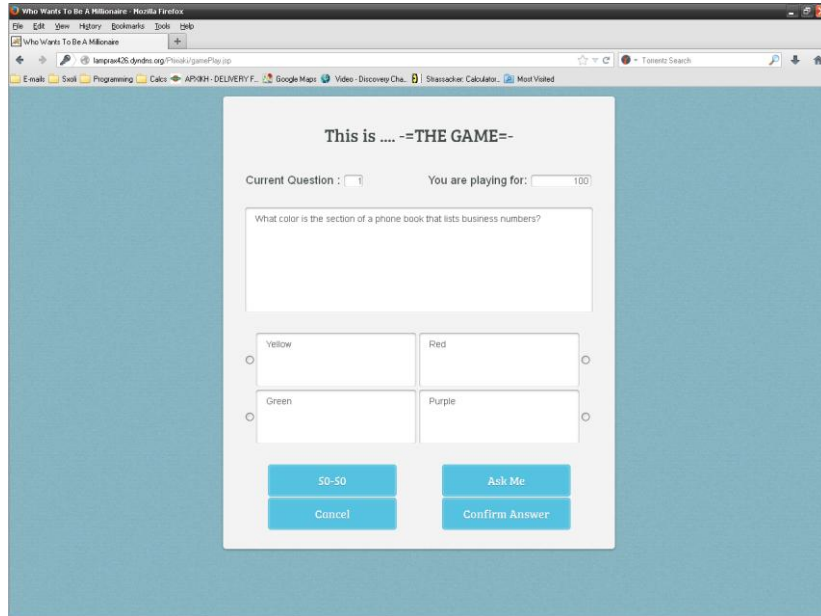
OID: Difficulty Level:

Which game show would have contestants guessing popular response to survey questions?

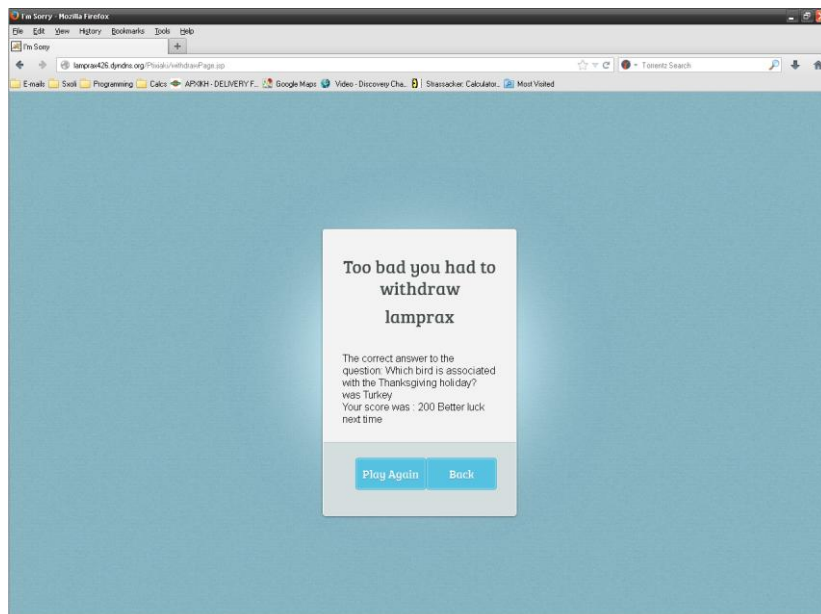
Family Feud The Weakest Link

The Wheel of Fortune Jeopardy!

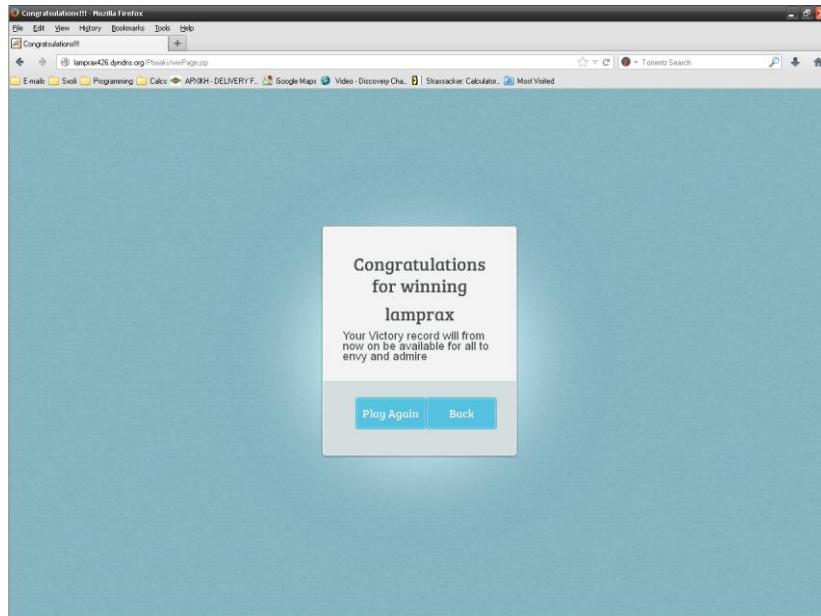
Σελίδα του παιχνιδιού



Έξοδος από το παιχνίδι



Ο παίχτης κερδίζει το παιχνίδι



Συμπεράσματα

- Με λίγες αλλαγές γίνετε να εκτελεσθούν άλλες λειτουργίες
- Δεν περιοριζόμαστε μόνο στο κείμενο
 - Μπορούμε να έχουμε εικόνες
 - Ήχους
 - Ή ακόμα και βίντεο
- Παράδειγμα είναι η θεωρητική εξέταση των υποψηφίων για δίπλωμα οδήγησης

Σας ευχαριστώ
για τον χρόνο σας

Πηγαίος Κώδικας Java

GameBean.java

```
package ptixiaki;
```

```
public class GameBean {
    private Object[][] gameQuestions;
    private Object[][] answersOrder;
    private Integer currentQuestion=null;
    private Integer[] answersFromHelp=null;
    private Boolean[] helpStatus={true, true};
    private Boolean[] fromHelp={false, false};
    private final static Integer[] moneyTable={0, 100, 200, 300, 500, 1000, 2000, 4000, 8000, 16000, 32000, 64000,
125000, 250000, 500000, 1000000};

    public void setGameQuestions(Object[][] x){
        gameQuestions=x;
    }

    public Object[][] getGameQuestions(){
        return gameQuestions;
    }

    public Object[] getGameQuestion(int x){
        return gameQuestions[x];
    }

    public void setAnswersOrder(Object[][] x){
        answersOrder=x;
    }

    public Object[][] getAnswersOrder(){
```

```

        return answersOrder;
    }

    public Object[] getAnswerOrder(int x){
        return answersOrder[x];
    }

    public void setCurrentQuestion(String x){
        currentQuestion=Integer.parseInt(x);
    }

    public void setCurrentQuestion(int x){
        currentQuestion=x;
    }

    public String getCurrentQuestion(){
        return Integer.toString(currentQuestion);
    }

    public String getCurrentLevel(){
        return Integer.toString(moneyTable[currentQuestion]);
    }

    public void setHelpStatus(Boolean[] x){
        helpStatus=x;
    }

    public Boolean[] getHelpStatus(){
        return helpStatus;
    }

    public void setFromHelp(Boolean[] x){
        fromHelp=x;
    }

```

```
public Boolean[] getFromHelp(){
    return fromHelp;
}

public void setAnswersFromHelp(Integer[] x){
    answersFromHelp=x;
}

public Integer[] getAnswersFromHelp(){
    return answersFromHelp;
}
}
```

GameLogic.java

```
package ptixiaki;

import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.sql.ResultSet;
import java.sql.Connection;
import java.sql.SQLException;
import java.security.SecureRandom;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collections;
import java.util.List;
import java.util.Random;
import java.util.Date;
import java.text.DateFormat;
import java.text.SimpleDateFormat;

public class GameLogic extends HttpServlet {
    String dbtable="emotherearth.Question";

    @Override
    public void init(ServletConfig config) throws ServletException{
        super.init(config);
    }

    public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException,
    java.io.IOException {

        String redirectPage="userLogged.jsp"; // Default redirect page
        String originID=request.getParameter("originID"); // Metavliti gia na 3exorizo tis formes meta3i
        tous

        String buttonID=request.getParameter("buttonID"); // Metavliti gia na 3exorizo ta koumpia meta3i tous
        request.getSession().setAttribute("originID",originID);
    }
}
```

```

if (originID.equals("Play Again")){
    originID="startGamePlay";
}

if (originID.equals("startGamePlay")){
    GameBean gameB=new GameBean();
    gameB.setGameQuestions(getQuestions());
    gameB.setAnswersOrder(shuffleAnswers());
    gameB.setCurrentQuestion(1);
    request.getSession().removeAttribute("currentGame");
    request.getSession().setAttribute("currentGame",gameB);

    gameB=null;
    redirectPage="gamePlay.jsp";
}

if (originID.equals("gamePlay")){
    String abc=request.getParameter("currentQuestion");
    Integer currentQuestion=Integer.parseInt(abc);
    if (buttonID.equals("Confirm Answer")){
        Integer playerChoise=Integer.parseInt(request.getParameter("playerChoise"));
        Integer[][]
answer_layout=(Integer[][])((GameBean)request.getSession().getAttribute("currentGame")).getAnswersOrder();

        int qid_temp=0;
        Object[][]
gameQuestions=((GameBean)request.getSession().getAttribute("currentGame")).getGameQuestions();
        qid_temp=Integer.parseInt((String)gameQuestions[currentQuestion-1][0]); //New

        if (checkQuestion(currentQuestion-1, playerChoise-1,answer_layout,qid_temp)){
            if (currentQuestion<15){
                ((GameBean)request.getSession().getAttribute("currentGame")).setCurrentQuestion(currentQuestion+1);
                redirectPage="gamePlay.jsp";
            }else{
                DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
                Date datetime = new Date();
                HistoryManagement historyM=new HistoryManagement();
            }
        }
    }
}

```

```

        historyM.updateHistory(((UserBean) request.getSession().getAttribute("currentSessionUser")).getUsername(), dateF
ormat.format(datetime), 15, "1000000");
            historyM=null;
            dateFormat=null;
            datetime=null;

            redirectPage="winPage.jsp";
        }
    } else {
        currentQuestion=currentQuestion-1;

        ((GameBean) request.getSession().getAttribute("currentGame")).setCurrentQuestion(currentQuestion);

        if(currentQuestion>=10){

            ((GameBean) request.getSession().getAttribute("currentGame")).setCurrentQuestion(10);
            DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
            Date datetime = new Date();
            HistoryManagement historyM=new HistoryManagement();

            historyM.updateHistory(((UserBean) request.getSession().getAttribute("currentSessionUser")).getUsername(), dateF
ormat.format(datetime), currentQuestion, ((GameBean) request.getSession().getAttribute("currentGame")).getCurrentLevel(
));

            historyM=null;
            dateFormat=null;
            datetime=null;
        }else if(currentQuestion>=5){

            ((GameBean) request.getSession().getAttribute("currentGame")).setCurrentQuestion(5);
            DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
            Date datetime = new Date();
            HistoryManagement historyM=new HistoryManagement();

            historyM.updateHistory(((UserBean) request.getSession().getAttribute("currentSessionUser")).getUsername(), dateF
ormat.format(datetime), currentQuestion, ((GameBean) request.getSession().getAttribute("currentGame")).getCurrentLevel(
));

            historyM=null;
            dateFormat=null;
            datetime=null;

```



```

        }
        redirectPage="losePage.jsp";
    }
    answer_layout=null;
    playerChoise=null;
    currentQuestion=null;
    gameQuestions=null;
}else if(buttonID.equals("50-50")){
    GameBean currentGame = (GameBean)request.getSession().getAttribute("currentGame"); //ola ta
stoixeia tou paixnidou
    Boolean[] check=currentGame.getHelpStatus(); //mevliti gia ton elegxo isxios tis voi8eias
    Boolean[] fromHelp={true,false};
    Integer[] answersOrder=(Integer[])currentGame.getAnswerOrder(currentQuestion-1); //o pinakas
pou exei tin seira apeikonisis ton erotiseon stin selida
    Integer[] targetAnswers={0,0};

    Random ran = new Random();
    int secondRandom = ran.nextInt(3) + 2; // enas tixaios ari8mos meta3i tou 2 kai tou 4

    targetAnswers[0]=1;
    targetAnswers[1]=secondRandom;

    check[0]=false;
    currentGame.setAnswersFromHelp(targetAnswers);
    currentGame.setFromHelp(fromHelp);
    currentGame.setHelpStatus(check);
    request.getSession().removeAttribute("currentGame");
    request.getSession().setAttribute("currentGame",currentGame);

    currentGame=null;
    check=null;
    ran=null;
    fromHelp=null;
    answersOrder=null;
    targetAnswers=null;
    redirectPage="gamePlay.jsp";
}else if(buttonID.equals("Ask Me")){
    GameBean currentGame = (GameBean)request.getSession().getAttribute("currentGame");
    Boolean[] check=currentGame.getHelpStatus();
    Boolean[] fromHelp={false,true};

```

```

String qid=(String)currentGame.getGameQuestion(currentQuestion-1)[0];

Integer[] targetAnswers=audienceHelp(Integer.parseInt(qid));

currentGame.setAnswersFromHelp(targetAnswers);
check[1]=false;

currentGame.setFromHelp(fromHelp);
currentGame.setHelpStatus(check);
request.getSession().removeAttribute("currentGame");
request.getSession().setAttribute("currentGame",currentGame);

currentGame=null;
check=null;
fromHelp=null;
qid=null;
targetAnswers=null;
redirectPage="gamePlay.jsp";
}else{
//The "Cancel" Button was pushed
if (currentQuestion>1){
currentQuestion=currentQuestion-1;

((GameBean)request.getSession().getAttribute("currentGame")).setCurrentQuestion(currentQuestion);
DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
Date datetime = new Date();
HistoryManagement historyM=new HistoryManagement();

historyM.updateHistory(((UserBean)request.getSession().getAttribute("currentSessionUser")).getUsername(),dateFormat.format(datetime),currentQuestion,((GameBean)request.getSession().getAttribute("currentGame")).getCurrentLevel());

historyM=null;
dateFormat=null;
datetime=null;
}
redirectPage="withdrawPage.jsp";
}
currentQuestion=null;
}

```

```

        if (originID.equalsIgnoreCase("back")) {
            request.getSession().removeAttribute("currentGame");
            ((UserBean) request.getSession().getAttribute("currentSessionUser")).setErrorOutput("");
            redirectPage="userLogged.jsp";
        }

        originID=null;
        response.sendRedirect(redirectPage);
    }

    protected String[][] getQuestions(){
        QuestionManagement mpla=new QuestionManagement();
        SecureRandom randomGen = new SecureRandom();
        String[][] Questions= new String[15][11];
        String[][] q_tmp=null;
        int q_t_size=0;
        Integer[] seed=new Integer[5];
        boolean flag=true;
        int ii=0;
        for(int jj=1;jj<=3;jj++){
            if(jj==1){
                q_tmp=mpla.readQuestionByLevel("low");
            }else if (jj==2){
                q_tmp=mpla.readQuestionByLevel("medium");
            }else if (jj==3){
                q_tmp=mpla.readQuestionByLevel("high");
            }
            q_t_size=q_tmp.length;
            seed[0]=randomGen.nextInt(q_t_size);
            for (int i = 1; i<5; i++){
                flag=true;
                while(flag){
                    flag=false;
                    ii = randomGen.nextInt(q_t_size);
                    for (int j=0;j<i;j++){
                        if (seed[j]==ii){
                            flag=true;
                        }
                    }
                }
            }
        }
    }

```

```

        seed[i]=ii;
    }
    for (int i=(5*(jj-1)); i<(5*jj); i++){
        Questions[i]=q_tmp[seed[i-(5*(jj-1))]];
    }
}

    q_tmp=null;
    seed=null;
    randomGen=null;
    mpla=null;

    return Questions;
}

protected Integer[][] shuffleAnswers(){

    Integer[][] Answers=new Integer[15][4];
    SecureRandom randomGen = new SecureRandom();

    List<Integer> intList = new ArrayList<Integer>(Arrays.asList(1, 2, 3, 4));
    List<Integer> intList2= new ArrayList<Integer>();

    for(int j=0;j<15;j++){
        intList2=intList;
        Collections.shuffle(intList2, randomGen);
        for(int i=0;i<4;i++){
            Answers[j][i]=intList2.get(i);
        }
    }
    intList=null;
    intList2=null;
    randomGen=null;

    return Answers;
}

protected Boolean checkQuestion(int question_progress, int answer_number, Integer[][] answer_layout,int
qID_temp){
    int number_selected =answer_layout[question_progress][answer_number];

```

```

String scoreSelect=new String();

if (number_selected==1){
    scoreSelect="CorAnsScore";
}else if (number_selected==2){
    scoreSelect="ans1Score";
}else if (number_selected==3){
    scoreSelect="ans2Score";
}else if (number_selected==4){
    scoreSelect="ans3Score";
}

ServerAccess serverAcc=new ServerAccess();
Connection con=serverAcc.connectMySQL();

String command="UPDATE "+dbtable+" SET "+scoreSelect+"="+scoreSelect+"+1 WHERE qid='"+qID_temp+"'";
serverAcc.makeChange(command,con);

serverAcc.closeMySQLConnection(con);
scoreSelect=null;
con=null;
serverAcc=null;
command=null;
answer_layout=null;

if (number_selected==1){
    return true;
}else{
    return false;
}
}

protected Integer[] audienceHelp(int qID){

    ResultSet result=null;
    Integer[] res=new Integer[4];
    Integer sumTotal=0;
    ServerAccess serverAcc=new ServerAccess();
    Connection con=serverAcc.connectMySQL();
    String command="select * from "+dbtable+" where qid='"+qID+"'";

```

```

result=serverAcc.makeQuery(command, con);
try{
    if(result.first()){
        res[0]=Integer.parseInt(result.getString("CorAnsScore"));
        res[1]=Integer.parseInt(result.getString("ans1Score"));
        res[2]=Integer.parseInt(result.getString("ans2Score"));
        res[3]=Integer.parseInt(result.getString("ans3Score"));
    }

    result.close();
} catch(SQLException e){
    System.out.println("SQLException\n"+e);
}
serverAcc.closeMySQLConnection(con);
result=null;
con=null;
serverAcc=null;
command=null;

sumTotal=res[0]+res[1]+res[2]+res[3]; //get te total answers given

res[0]=(res[0]*100)/sumTotal; //calc the persentage that the correct answer was given
res[1]=(res[1]*100)/sumTotal; //calc the persentage that the first wrong answer was given
res[2]=(res[2]*100)/sumTotal; //calc the persentage that the second wrong answer was given
res[3]=(res[3]*100)/sumTotal; //calc the persentage that the third wrong answer was given

sumTotal=null;

return res;
}
}

```

HistoryManagement.java

```
package ptixiaki;

import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.sql.ResultSet;
import java.sql.Connection;
import java.sql.SQLException;

public class HistoryManagement extends HttpServlet {
    String dbtable="emotherearth.History2";
    ServerAccess serverAcc=new ServerAccess();

    @Override
    public void init(ServletConfig config) throws ServletException{
        super.init(config);
    }

    public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException,
    java.io.IOException {
        // Metavliti gia na 3exorizo ta koumpia meta3i tous
        String originID=request.getParameter("originID");
        String redirectPage="userLogged.jsp";

        if (originID.equals("View your History")){
            Object
historyObject=showPlayerHistory(((UserBean) request.getSession().getAttribute("currentSessionUser")).getUsername());
            request.getSession().setAttribute("historyObject",historyObject);
            request.getSession().setAttribute("originID",originID);
            historyObject=null;
        }
    }
}
```

```

        redirectPage="displayHistory.jsp";
    }

    if (originID.equals("View High Scores")){
        request.getSession().setAttribute("historyObject", showHighScores());
        request.getSession().setAttribute("originID", originID);
        redirectPage="displayHistory.jsp";
    }

    if (originID.equals("View All Recent History")){
        request.getSession().setAttribute("historyObject", showAllHistory());
        request.getSession().setAttribute("originID", originID);
        redirectPage="displayHistory.jsp";
    }

    if (originID.equals("Back")){
        request.getSession().removeAttribute("historyObject");
        request.getSession().removeAttribute("buttonID");

        ((UserBean) request.getSession().getAttribute("currentSessionUser")).setErrorOutput("");

        redirectPage="userLogged.jsp";
    }

    originID=null;
    response.sendRedirect(redirectPage);
}

public void updateHistory(String Pusername, String DateTime, int questions_answered, String money_gained){

    Integer result=null;
    ResultSet res=null;
    Connection con=serverAcc.connectMySQL();

```



```

String command="select id from "+dbtable+"";
res=serverAcc.makeQuery(command,con);
int tempest=0;
int tempest1=0;
boolean flag=true;
try{
    res.last();
    int j=res.getRow();
    for (int i=1;i<=j&&flag;i++){
        res.absolute(i);
        tempest1=Integer.parseInt(res.getString("id"));
        if (tempest+1!=tempest1){
            tempest=tempest+1;
            flag=false;
        }else{
            tempest=tempest1;
        }
    }
    res.close();
}catch(SQLException e){
    System.out.println("SQLException\n"+e);
}
if (flag==false){
    command="insert into "+dbtable+" values
("+tempest+", '"+Pusername+"', '"+DateTime+"', '"+questions_answered+"', '"+money_gained+'');"
}else{
    command="insert into "+dbtable+" values
(0, '"+Pusername+"', '"+DateTime+"', '"+questions_answered+'', '"+money_gained+'');"
}
result=serverAcc.makeChange(command,con);
serverAcc.closeMySQLConnection(con);

Pusername=null;
DateTime=null;

```

```

    money_gained=null;
    result=null;
    res=null;
    con=null;
    command=null;
}

public Object[][] showPlayerHistory(String username) {
    ResultSet result=null;
    Object[][] res=null;
    Connection con=serverAcc.connectMySQL();
    String command="select * from "+dbtable+" where Pusername='"+username+"' order by PlayedAt desc;";
    result=serverAcc.makeQuery(command,con);
    try{
        result.last();
        int j=result.getRow();
        res=new Object[j][5];
        for (int i=0;i<j;i++){
            result.absolute(i+1);
            res[i][0]=Integer.parseInt(result.getString("id"));
            res[i][1]=result.getString("Pusername");
            res[i][2]=result.getString("PlayedAt");
            res[i][3]=Integer.parseInt(result.getString("qstAnswd"));
            res[i][4]=Integer.parseInt(result.getString("moneyGained"));
        }
        result.close();
    } catch (SQLException e) {
        System.out.println("SQLException\n"+e);
    }
    serverAcc.closeMySQLConnection(con);

    username=null;
    result=null;
}

```

```

        con=null;
        command=null;

        return res;
    }

    public Object[][] showAllHistory() {
        ResultSet result=null;
        Object[][] res=null;
        Connection con=serverAcc.connectMySQL();
        String command="select * from "+dbtable+" order by PlayedAt desc limit 200;";
        result=serverAcc.makeQuery(command,con);
        try{
            result.last();
            int j=result.getRow();
            res=new Object[j][5];
            for (int i=0;i<j;i++){
                result.absolute(i+1);
                res[i][0]=Integer.parseInt(result.getString("id"));
                res[i][1]=result.getString("Pusername");
                res[i][2]=result.getString("PlayedAt");
                res[i][3]=Integer.parseInt(result.getString("qstAnswd"));
                res[i][4]=Integer.parseInt(result.getString("moneyGained"));
            }
            result.close();
        } catch(SQLException e){
            System.out.println("SQLException\n"+e);
        }
        serverAcc.closeMySQLConnection(con);

        result=null;
        con=null;
        command=null;
    }

```

```

        return res;
    }

    public Object[][] showHighScores() {
        ResultSet result=null;
        Object[][] res=null;
        Connection con=serverAcc.connectMySQL();
        String command="select * from "+dbtable+" order by moneyGained desc limit 100;";
        result=serverAcc.makeQuery(command,con);
        try{
            result.last();
            int j=result.getRow();
            res=new Object[j][5];
            for (int i=0;i<j;i++){
                result.absolute(i+1);
                res[i][0]=Integer.parseInt(result.getString("id"));
                res[i][1]=result.getString("Pusername");
                res[i][2]=result.getString("PlayedAt");
                res[i][3]=Integer.parseInt(result.getString("qstAnswd"));
                res[i][4]=Integer.parseInt(result.getString("moneyGained"));
            }
            result.close();
        } catch(SQLException e){
            System.out.println("SQLException\n"+e);
        }
        serverAcc.closeMySQLConnection(con);

        result=null;
        con=null;
        command=null;

        return res;
    }
}

```

```
public void delPlayerHistory(String username) {
    Integer result=null;
    Connection con=serverAcc.connectMySQL();
    String command="delete from "+dbtable+" where Pusername='"+username+"'";
    result=serverAcc.makeChange(command,con);
    serverAcc.closeMySQLConnection(con);

    username=null;
    result=null;
    con=null;
    command=null;
}
}
```

QuestionManagement.java

```
package ptixiaki;

import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.sql.ResultSet;
import java.sql.Connection;
import java.sql.SQLException;

public class QuestionManagement extends HttpServlet {

    String dbtable="emotherearth.Question";
    ServerAccess serverAcc=new ServerAccess();

    @Override
    public void init(ServletConfig config) throws ServletException{
        super.init(config);
    }

    public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException,
    java.io.IOException {
        String originID=request.getParameter("originID");
        String redirectPage="userLogged.jsp";
        String buttonID=request.getParameter("buttonID");

        request.getSession().setAttribute("buttonID",buttonID);

        if (originID.equals("Add Question")){
            request.getSession().setAttribute("originID",originID);
            redirectPage="editQuestions.jsp";
        }
    }
}
```

```

}

if (originID.equals("addQuestion")){
    if(buttonID.equals("Confirm")){
        String questionBox=request.getParameter("questionBox");
        String answer1=request.getParameter("answer1");
        String answer2=request.getParameter("answer2");
        String answer3=request.getParameter("answer3");
        String answer4=request.getParameter("answer4");
        String difficultyLevel=request.getParameter("difficultyLevel");
        String CorAnsScore=null;
        String ans1Score=null;
        String ans2Score=null;
        String ans3Score=null;

        if (difficultyLevel.equals("Low")){
            CorAnsScore="50";
            ans1Score="20";
            ans2Score="20";
            ans3Score="10";
        }else if (difficultyLevel.equals("Medium")){
            CorAnsScore="28";
            ans1Score="24";
            ans2Score="24";
            ans3Score="24";
        }else if (difficultyLevel.equals("High")){
            CorAnsScore="20";
            ans1Score="30";
            ans2Score="30";
            ans3Score="20";
        }

        createQuestion(questionBox,answer1,answer2,answer3,answer4,difficultyLevel,CorAnsScore,ans1Score,ans2Score,ans3Score);
    }
}

```

```

        questionBox=null;
        answer1=null;
        answer2=null;
        answer3=null;
        answer4=null;
        difficultyLevel=null;
        CorAnsScore=null;
        ans1Score=null;
        ans2Score=null;
        ans3Score=null;

    }
    request.getSession().setAttribute("originID",originID);
    redirectPage="editQuestions.jsp";
}

if (originID.equals("Edit Question")){
    request.getSession().setAttribute("questionsObject",readQuestion(1));
    request.getSession().setAttribute("originID",originID);
    redirectPage="editQuestions.jsp";
}

if (originID.equals("editQuestion")){
    Integer qID=new Integer(request.getParameter("qIDText"));

    if (buttonID.equals("Next")){
        request.getSession().setAttribute("questionsObject", readQuestion(qID+1));

        qID=null;
        redirectPage="editQuestions.jsp";
    }else if(buttonID.equals("Previous")){
        if (qID>1){
            request.getSession().setAttribute("questionsObject", readQuestion(qID-1));

```



```

    }
    qID=null;
    redirectPage="editQuestions.jsp";
}else if(buttonID.equals("Go to Question")){
    if (qID<1){
        qID=1;
    }
    request.getSession().setAttribute("questionsObject", readQuestion(qID));
    qID=null;
    redirectPage="editQuestions.jsp";
}else if(buttonID.equals("Cancel")){
    request.getSession().setAttribute("questionsObject", readQuestion(qID));
    request.getSession().setAttribute("originID",originID);
    qID=null;
    redirectPage="editQuestions.jsp";
}else if(buttonID.equals("Confirm")){
    String questionBox=request.getParameter("questionBox");
    String answer1=request.getParameter("answer1");
    String answer2=request.getParameter("answer2");
    String answer3=request.getParameter("answer3");
    String answer4=request.getParameter("answer4");
    String difficultyLevel=request.getParameter("difficultyLevel");

    editQuestion(qID,questionBox,answer1,answer2,answer3,answer4,difficultyLevel);
    request.getSession().setAttribute("questionsObject", readQuestion(qID));

    qID=null;
    questionBox=null;
    answer1=null;
    answer2=null;
    answer3=null;
    answer4=null;
    difficultyLevel=null;
    redirectPage="editQuestions.jsp";
}

```

```

    }

    qID=null;
    request.getSession().setAttribute("originID",originID);
}

if (originID.equals("Delete Question")){
    request.getSession().setAttribute("questionsObject", readQuestion(1));
    request.getSession().setAttribute("originID",originID);
    redirectPage="editQuestions.jsp";
}

if (originID.equals("deleteQuestion")){
    Integer qID=new Integer(request.getParameter("qIDText"));

    if (buttonID.equals("Next")){
        request.getSession().setAttribute("questionsObject", readQuestion(qID+1));
        qID=null;
        redirectPage="editQuestions.jsp";
    }else if(buttonID.equals("Previous")){
        if (qID>1){
            request.getSession().setAttribute("questionsObject", readQuestion(qID-1));
        }
        qID=null;
        redirectPage="editQuestions.jsp";
    }else if(buttonID.equals("Go to Question")){
        if (qID<1){
            qID=1;
        }
        request.getSession().setAttribute("questionsObject", readQuestion(qID));

        qID=null;
        redirectPage="editQuestions.jsp";
    }else if(buttonID.equals("Cancel")){

```

```

        originID="Back";
    }else if(buttonID.equals("Delete")){
        delQuestion(qID);
        request.getSession().setAttribute("questionsObject", readQuestion(1));
        qID=null;
        redirectPage="editQuestions.jsp";
    }

    qID=null;
    request.getSession().setAttribute("originID",originID);
    redirectPage="editQuestions.jsp";
}

if (originID.equals("Back")){
    request.getSession().removeAttribute("userList");
    request.getSession().removeAttribute("historyObject");
    request.getSession().removeAttribute("buttonID");
    request.getSession().removeAttribute("questionsObject");

    ((UserBean) request.getSession().getAttribute("currentSessionUser")).setErrorOutput("");

    redirectPage="userLogged.jsp";
}

buttonID=null;
originID=null;
response.sendRedirect(redirectPage);
}

```

```

public void createQuestion(String question, String correct_answer, String alternative1, String alternative2,
String alternative3, String level, String CorAnsScore, String ans1Score, String ans2Score, String ans3Score){
    Integer result=null;
    Connection con=serverAcc.connectMySQL();

```

```

        String command="insert into "+dbtable+" values
(0,'" +question+"', '"+correct_answer+"', '"+alternative1+"', '"+alternative2+"', '"+alternative3+"', '"+level.toLowerCase
()+'', '"+CorAnsScore+"', '"+ans1Score+"', '"+ans2Score+"', '"+ans3Score+'');"
        result=serverAcc.makeChange (command, con);
        serverAcc.closeMySQLConnection (con);

        question=null;
        correct_answer=null;
        alternative1=null;
        alternative2=null;
        alternative3=null;
        level=null;
        CorAnsScore=null;
        ans1Score=null;
        ans2Score=null;
        ans3Score=null;

        result=null;
        con=null;
        command=null;
}

```

```

public void editQuestion(int x,String question, String correct_answer, String alternative1, String
alternative2, String alternative3, String level){
    Integer res=null;
    ResultSet result=null;
    Connection con=serverAcc.connectMySQL();

    Integer[] answerScores=new Integer[4];
    String command="select * from "+dbtable+" where qid='"+x+"'";
    result=serverAcc.makeQuery (command, con);
    try{
        if(result.first()){

```

```

        answerScores[0]=Integer.parseInt(result.getString("CorAnsScore"));
        answerScores[1]=Integer.parseInt(result.getString("ans1Score"));
        answerScores[2]=Integer.parseInt(result.getString("ans2Score"));
        answerScores[3]=Integer.parseInt(result.getString("ans3Score"));
    }

    result.close();
} catch(SQLException e){
    System.out.println("SQLException\n"+e);
}
result=null;

command="delete from "+dbtable+" where qid='"+x+"'";
res=serverAcc.makeChange(command,con);
command="insert into "+dbtable+" values
('+x+', '"+question+"', '"+correct_answer+"', '"+alternative1+"', '"+alternative2+"', '"+alternative3+"', '"+level.toLowerCase()+
"', '"+answerScores[0]+'', '"+answerScores[1]+'', '"+answerScores[2]+'', '"+answerScores[3]+'')";
res=serverAcc.makeChange(command,con);
serverAcc.closeMySQLConnection(con);

question=null;
correct_answer=null;
alternative1=null;
alternative2=null;
alternative3=null;
answerScores=null;

level=null;
res=null;
con=null;
command=null;
}

public String[] readQuestion(int x){

```

```

String[] question=null;
ResultSet result=null;
Connection con=serverAcc.connectMySQL();
String command="select * from "+dbtable+"";
result=serverAcc.makeQuery(command,con);
try{
    result.last();
    int j=result.getRow();

    if (x>j){
        x=j;
    }
    question=new String[7];
    result.absolute(x);
    question[0]=result.getString("qid");
    question[1]=result.getString("qst");
    question[2]=result.getString("CorAns");
    question[3]=result.getString("ans1");
    question[4]=result.getString("ans2");
    question[5]=result.getString("ans3");
    question[6]=result.getString("dlevel");

    result.close();
}catch(SQLException e){
    System.out.println("SQLException\n"+e);
}
serverAcc.closeMySQLConnection(con);
result=null;
con=null;
command=null;

return question;
}

```

```

public String[][] readQuestionByLevel(String level){
    ResultSet result=null;
    String[][] question_table=null;
    Connection con=serverAcc.connectMySQL();
    String command="select * from "+dbtable+" where dlevel='"+level.toLowerCase()+"'";
    result=serverAcc.makeQuery(command,con);
    try{
        result.last();
        int j=result.getRow();
        question_table=new String[j][11];
        for (int i=0;i<j;i++){
            result.absolute(i+1);
            question_table[i][0]=result.getString("qid");
            question_table[i][1]=result.getString("qst");
            question_table[i][2]=result.getString("CorAns");
            question_table[i][3]=result.getString("ans1");
            question_table[i][4]=result.getString("ans2");
            question_table[i][5]=result.getString("ans3");
            question_table[i][6]=result.getString("dlevel");
            question_table[i][7]=result.getString("CorAnsScore");
            question_table[i][8]=result.getString("ans1Score");
            question_table[i][9]=result.getString("ans2Score");
            question_table[i][10]=result.getString("ans3Score");
        }
        result.close();
    }catch(SQLException e){
        System.out.println("SQLException\n"+e);
    }
    serverAcc.closeMySQLConnection(con);

    level=null;
    result=null;
    con=null;
    command=null;
}

```

```

        return question_table;
    }

    public void delQuestion(int x){
        Integer result=null;
        ResultSet result2=null;
        Connection con=serverAcc.connectMySQL();
        String command="delete from "+dbtable+" where qid='"+x+"'";

        result=serverAcc.makeChange(command,con);

        try{
            command="select qid from "+dbtable+" order by qid asc;";
            result2=serverAcc.makeQuery(command,con);
            result2.last();
            int j=result2.getRow();
            String[] oldKeys=new String[j];
            for (int i=0;i<j;i++){
                result2.absolute(i+1);
                oldKeys[i]=(String)result2.getString("qid");
            }
            result2.close();
            for(int i=1;i<=j;i++){
                command="update "+dbtable+" set qid='"+i+"' where qid='"+oldKeys[i-1]+'";
                result=serverAcc.makeChange(command,con);
            }
            oldKeys=null;
        }catch(SQLException e){
            System.out.println("SQLException\n"+e);
        }

        serverAcc.closeMySQLConnection(con);
        result=null;
    }

```



```
result2=null;  
con=null;  
command=null;
```

```
}
```

```
}
```

ServerAccess.java

```
package ptixiaki;

import java.sql.*;
import com.mysql.jdbc.Driver;

public class ServerAccess{
    //orismos tou onomatos tou driver pou 8a xrisimopoi8ei gia tin epikinonia me tin basi dedomenon
    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    //orismos tou url kai tou onomatos tis vasis dedomenon pou 8a xrisimopoiisoume
    static final String SERVER_ADDRESS = "jdbc:mysql://127.0.0.1:3306/emotherearth";
    //orismos tou username tou xristi pou exei dikaiomata stin basi dedomenon
    static final String MYSQL_USER = "lamprax";
    //orismos tou password tou xristi
    static final String MYSQL_PASS = "6942682888";

    //routina pou dimiourgei to "object" me tin sindesi stin vasi dedomenon
    public Connection connectMySQL(){
        //orismos tou "object" pou 8a xrisimopoi8ei gia tin epikinonia me tin vasi dedomenon
        Connection con=null;
        try{
            //arxikopoiisi tou odigou tis vasis dedomenon
            Class.forName(JDBC_DRIVER).newInstance();
            //dimiouria tis sindesis me tin vasi dedomenon
            con = DriverManager.getConnection(SERVER_ADDRESS, MYSQL_USER, MYSQL_PASS);
        }catch(ClassNotFoundException e){ // Diaxirisi sfalmaton
            System.out.println("Couldn't load database driver: \n" + e);
        }catch(Exception e){
            System.out.println( e );
        }

        return con;
    }
}
```

```

//routina pou stelnei erotimata stin vasi dedomenon
public ResultSet makeQuery(String command,Connection con){
    ResultSet result=null;
    try{
        Statement st = con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_UPDATABLE);
        result = st.executeQuery(command);
        //st=null;
    }catch(SQLException e){
        System.out.println("SQLException\n"+e);
    }
    command=null;
    con=null;
    return result;
}

public Integer makeChange(String command,Connection con){
    Integer res=null;
    try{
        Statement st = con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_UPDATABLE);
        res = st.executeUpdate(command);
        //st=null;
    }catch(SQLException e){
        System.out.println("SQLException\n"+e);
    }
    command=null;
    con=null;
    return res;
}

public void closeMySQLConnection(Connection con){
    try {
        if (con!= null){
            con.close();
        }
    }
}

```

```
        }  
    }catch (SQLException ignored){  
        System.out.println( ignored );  
    }  
    con=null;  
}  
  
}
```

UserBean.java

```
package ptixiaki;

public class UserBean {
    private String username;
    private String password;
    private String errorOutput="";
    private String originID;
    private String buttonPushed;
    private boolean valid;
    private boolean admin;

    public void setPassword(String newPassword) {
        password = newPassword;
    }

    public String getPassword() {
        return password;
    }

    public void setUsername(String newUsername) {
        username = newUsername;
    }

    public String getUsername() {
        return username;
    }

    public void setErrorOutput(String errorOut){
        errorOutput=errorOut;
    }

    public String getErrorOutput(){
```

```
        return errorOutput;
    }

    public boolean isValid() {
        return valid;
    }

    public void setValid(boolean newValid) {
        valid = newValid;
    }

    public boolean isAdmin() {
        return admin;
    }

    public void setAdmin(boolean newValid) {
        admin = newValid;
    }

    public void setOriginID(String orID){
        originID=orID;
    }

    public String getOriginID(){
        return originID;
    }

    public void setButtonPushed(String bP){
        buttonPushed=bP;
    }

    public String getButtonPushed(){
        return buttonPushed;
    }
}
```

UserManagement.java

```
package ptixiaki;

import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import java.sql.ResultSet;
import java.sql.Connection;
import java.sql.SQLException;
import java.util.regex.Pattern;
import java.util.ArrayList;
import java.util.zip.CRC32;

public class UserManagement extends HttpServlet {
    String dbtable="emotherearth.Player";
    ServerAccess serverAcc=new ServerAccess();

    @Override
    public void init(ServletConfig config) throws ServletException{
        super.init(config);
    }

    public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException,
    java.io.IOException {
        // Default redirect page
        String redirectPage="LoginPage.jsp";
        // Metavliti pou mou leei pio koumpi pati8ike
        String buttonPushed=request.getParameter("submit");
        // Object pou apo8ikeuei ta dedomena tou trexontos xristi
        UserBean user=new UserBean();
    }
}
```

```

if(buttonPushed.equals("Login")||buttonPushed.equals("Register")){
    if (request.getParameter("userName")==null ){
        user.setUserName("");
        buttonPushed="Login";
        redirectPage="LoginPage.jsp";
    }else{
        user.setUserName(request.getParameter("userName"));
    }

    if (request.getParameter("passWord")==null ){
        user.setPassword("");
        buttonPushed="Login";
        redirectPage="LoginPage.jsp";
    }else{
        user.setPassword(request.getParameter("passWord"));
    }
}

if (buttonPushed.equals("Register")){
    redirectPage="registerPage.jsp";
    Boolean[] getUserStatus=new Boolean[3];

    user.setAdmin(false);
    Boolean usernameCheck=createUser(user.getUsername(),user.getPassword(),request);
    // if new user registration was succesfull GOTO login routine
    if(usernameCheck){
        buttonPushed="Login";
    }else {
        // Name taken
        //redirectPage="LoginPage.jsp";
    }
    usernameCheck=null;
}

```



```

if (buttonPushed.equals("Login")){
    redirectPage="LoginPage.jsp";
    Boolean[] getUserStatus=new Boolean[3];
    try{
        getUserStatus=loginUser(request); // try to login the user
        //getUserStatus[0] user login succes flag
        //getUserStatus[1] admin flag
        //getUserStatus[2] sql exception flag
        if (!getUserStatus[2]){ // if there was no SQL error
            if(getUserStatus[0]){ // if the user login was succesfull
                redirectPage="userLogged.jsp";
            }
        }
        getUserStatus=null;
    }catch(Throwable e){
        response.sendError(500, "Something went wrong..... "+e);
    }
}

if (buttonPushed.equals("Delete your Account")){
    redirectPage="deleteAccount.jsp";
}

if (buttonPushed.equals("Delete My Account")){
    Boolean verify=delAccount(request);
    if (verify){
        user=null;
        redirectPage="deleteNotice.jsp";
    } else {
        buttonPushed="Log Off";
    }
}
verify=null;

```

```

}

if (buttonPushed.equals("Change your Password")){
    redirectPage="pwdChange.jsp";
}

if (buttonPushed.equals("Verify")){
    String verify=changeUser(request);
    if(verify.equals("ok")){
        redirectPage="userLogged.jsp";
    }else if(verify.equals("Please make sure you have typed the correct passwords")){
        redirectPage="pwdChange.jsp";
    }else if(verify.equals("Wrong password!!")){
        buttonPushed="Log Off";
    }else if(verify.equals("SQLException")){
        redirectPage="pwdChange.jsp";
    }else if(verify.equals("false")){
        redirectPage="pwdChange.jsp";
    }else{
        redirectPage="pwdChange.jsp";
    }
    verify=null;
}

if (buttonPushed.equals("Log Off")){
    logoffUser(request);
    user=null;
    redirectPage="LoginPage.jsp";
}

if (buttonPushed.equals("Delete User")){
    listUsers(request,0);
    request.getSession().setAttribute("submit",buttonPushed);
    redirectPage="changeUserStatus.jsp";
}

```

```

}

if (buttonPushed.equals("Delete User!!")) {
    delUser(request);
    listUsers(request, 0);
    request.getSession().setAttribute("submit", buttonPushed);
    redirectPage="changeUserStatus.jsp";
}

if (buttonPushed.equals("Make Admin")) {
    listUsers(request, 1);
    request.getSession().setAttribute("submit", buttonPushed);
    redirectPage="changeUserStatus.jsp";
}

if (buttonPushed.equals("Make Administrator")) {
    makeAdmin((String) request.getParameter("userSelected"));
    listUsers(request, 1);
    request.getSession().setAttribute("submit", buttonPushed);
    redirectPage="changeUserStatus.jsp";
}

if (buttonPushed.equals("Delete Admin")) {
    listUsers(request, 2);
    request.getSession().setAttribute("submit", buttonPushed);
    redirectPage="changeUserStatus.jsp";
}

if (buttonPushed.equals("Delete Administrator!!")) {
    delAdmin((String) request.getParameter("userSelected"));
    listUsers(request, 2);
    request.getSession().setAttribute("submit", buttonPushed);
    redirectPage="changeUserStatus.jsp";
}

```

```

    if (buttonPushed.equals("Back")){
        request.getSession().removeAttribute("userList");
        request.getSession().removeAttribute("historyObject");
        request.getSession().removeAttribute("buttonID");
        request.getSession().removeAttribute("questionsObject");

        ((UserBean) request.getSession().getAttribute("currentSessionUser")).setErrorOutput("");

        redirectPage="userLogged.jsp";
    }

    buttonPushed=null;
    response.sendRedirect(redirectPage);
}

public Boolean createUser(String username, String password, HttpServletRequest request){
    UserBean user=new UserBean(); // Object pou apo8ikeuei ta dedomena tou trexontos xristi
    user.setErrorOutput("Something WEIRD happend!!!");
    Boolean errorFlag=false;
    Boolean ret=false;
    String errorMessage=new String();

    if((!(username==null))&&username.length()==0){
        errorFlag=true;
        errorMessage="You must enter a username\n";
    }

    if((!(password==null))&&password.length()==0){
        errorFlag=true;
        errorMessage=errorMessage+"You must enter a password\n";
    }

    Pattern illegalChars = Pattern.compile("[^a-zA-Z0-9]");

```

```

if(illegalChars.matcher(username).find()||illegalChars.matcher(password).find()){
    errorFlag=true;
    errorMessage=errorMessage+"Only english alphanumeric characters are accepted\n";
}

//catch the crackers!! credits go to the comic page xkcd.com/327 from XKCD!!!!
if(username.toLowerCase().contains("drop")){
    errorFlag=true;
    errorMessage=errorMessage+"You have entered an illegal username\n";
}
if(password.toLowerCase().contains("drop")){
    errorFlag=true;
    errorMessage=errorMessage+"You have entered an illegal password \n";
}
if(username.toLowerCase().contains("username")){
    errorFlag=true;
    errorMessage=errorMessage+"You have entered an illegal username \n";
}
if(password.toLowerCase().contains("password")){
    errorFlag=true;
    errorMessage=errorMessage+"You have entered an illegal password \n";
}

user.setErrorOutput(errorMessage);

if(!errorFlag){
    ResultSet result=null;
    Integer res=null;
    password=genHash(password);
    Connection con=serverAcc.connectMySQL();
    String command="select Pusername from "+dbtable+" where Pusername='"+username+"'";
    result=serverAcc.makeQuery(command,con);
    try{

```

```

        if(result.first()==false){
            result.close();
            command="insert into "+dbtable+" values ('"+username+"','"+password+"',0)";
            res=serverAcc.makeChange(command,con);
            ret=true;
            user.setErrorOutput("Registration Successfull");
        } else {
            ret=false;
            user.setErrorOutput("The username is already taken");
        }
    }catch(SQLException e){
        user.setErrorOutput("There was an error in the database.\n Please try again later");
    }
    serverAcc.closeMySQLConnection(con);
    con=null;
    result=null;
    res=null;
    command=null;
}

HttpSession session=request.getSession();
session.setAttribute("currentSessionUser",user);
errorFlag=null;
errorMessage=null;
password=null;
username=null;
session=null;
user=null;

return ret;
}

public Boolean[] loginUser(HttpServletRequest request){
    Boolean[] ret=new Boolean[3];

```

```

/*
ret[0] = Succesfull login
ret[1] = User / Admin
ret[2] = SQLException
*/
ResultSet result=null;
UserBean user=new UserBean(); // Object pou apo8ikeuei ta dedomena tou trexontos xristi
user.setUsername(request.getParameter("userName"));
user.setPassword(request.getParameter("passWord"));
user.setAdmin(false);
user.setValid(false);
user.setOriginID(request.getParameter("originID"));
user.setButtonPushed(request.getParameter("submit"));
user.setErrorOutput("Invalid Username or password.\n Please try Again.");

Connection con=serverAcc.connectMySQL();

String command="select PisAdmin from "+dbtable+" where Pusername='"+user.getUsername()+"' and
Ppwd='"+genHash(user.getPassword())+"'";

result=serverAcc.makeQuery(command, con);
//if result is not NULL then the login was succesfull
//if result is '0' then the user is a simple player
//if it is '1' then the user is an administrator
try{
    ret[0]=false;
    ret[1]=false;
    ret[2]=false;
    if(result.first()){
        if(result.getString("PisAdmin").compareTo("1")==0){
            ret[1]=true;
            user.setErrorOutput("Welcome Home Admin");
        }else{
            ret[1]=false;

```

```

        user.setErrorOutput("Welcome Home User");
    }
    ret[0]=true;
    ret[2]=false;
}
result.close();
}catch(SQLException e){
    ret[0]=false;
    ret[1]=false;
    ret[2]=true;
    user.setErrorOutput("There was an error in the database.\n Please try again later");
}
serverAcc.closeMySQLConnection(con);

user.setValid(ret[0]);
user.setAdmin(ret[1]);
HttpSession session=request.getSession();
session.setAttribute("currentSessionUser",user);
session.setMaxInactiveInterval(600);

con=null;
result=null;
command=null;
user=null;
session=null;

return ret;
}

public void logoffUser(HttpServletRequest request){
    request.getSession().invalidate();
}

public String changeUser(HttpServletRequest request){

```



```

UserBean user=(UserBean)request.getSession().getAttribute("currentSessionUser");

Integer res=null;
String ret="false";
//If both new passwords match the try to change the password
if (request.getParameter("newPassword1").equals(request.getParameter("newPassword2"))){
    Connection con=serverAcc.connectMySQL();
    String command="select Pusername from "+dbtable+" where Pusername='"+user.getUsername()+"' and
Ppwd='"+genHash(request.getParameter("oldPassword"))+"'";
    ResultSet result=serverAcc.makeQuery(command,con);
    try{
        if(result.first()){
            result.close();
            command="update "+dbtable+" set
Ppwd='"+genHash(request.getParameter("newPassword1"))+"' where Pusername='"+user.getUsername()+"'";
            res=serverAcc.makeChange(command,con);
            ret="ok";
        } else {
            ret="Wrong password!!!";
        }
    }catch(SQLException e){
        ret="SQLException";
    }
    serverAcc.closeMySQLConnection(con);

    con=null;
    command=null;
    res=null;
    result=null;
}else {
    ret="Please make sure you have typed the correct passwords";
}

UserBean newU=(UserBean) request.getSession().getAttribute("currentSessionUser");

```

```

        newU.setErrorOutput (ret);
        HttpSession session=request.getSession();
        session.removeAttribute("currentSessionUser");
        session.setAttribute("currentSessionUser", newU);
        res=null;

        return ret;
    }

    public void makeAdmin(String username){
        Integer result=null;
        Connection con=serverAcc.connectMySQL();
        String command="update "+dbtable+" set PisAdmin=1 where Pusername='"+username+"'";
        result=serverAcc.makeChange (command, con);
        serverAcc.closeMySQLConnection (con);
        username=null;
        result=null;
        con=null;
        command=null;
    }

    public void delAdmin(String username){
        Integer result=null;
        Connection con=serverAcc.connectMySQL();
        String command="update "+dbtable+" set PisAdmin=0 where Pusername='"+username+"'";
        result=serverAcc.makeChange (command, con);
        serverAcc.closeMySQLConnection (con);
        username=null;
        result=null;
        con=null;
        command=null;
    }

    @SuppressWarnings ("unchecked")

```

```

public void listUsers(HttpServletRequest request,int selectAdmins){
    ResultSet result=null;
    ArrayList results_string=null;
    Connection con=serverAcc.connectMySQL();
    String command=new String();
    if(selectAdmins==0){
        command="select Pusername from "+dbtable+" ";
    }else if (selectAdmins==1){
        command="select Pusername from "+dbtable+" where PisAdmin=0;";
    }else if (selectAdmins==2){
        command="select Pusername from "+dbtable+" where PisAdmin=1;";
    }

    result=serverAcc.makeQuery(command,con);

    try{
        result.last();
        int j=result.getRow();
        results_string=new ArrayList(j);
        for (int i=0;i<j;i++){
            result.absolute(i+1);
            results_string.add((String)result.getString("Pusername"));
        }
        result.close();
    } catch(SQLException e){
        System.out.println("SQLException\n"+e);
    }
    serverAcc.closeMySQLConnection(con);

    result=null;
    con=null;
    command=null;

    HttpSession session=request.getSession();
}

```

```

        session.removeAttribute("userList");
        session.setAttribute("userList",results_string);
    }

    public Boolean delAccount(HttpServletRequest request){
        Boolean ret=false;
        UserBean user=(UserBean) request.getSession().getAttribute("currentSessionUser");

        if( user.getUsername().equals(request.getParameter("userName")) &&
user.getPassword().equals(request.getParameter("passWord")) ){
            HistoryManagement tmp=new HistoryManagement();
            tmp.delPlayerHistory(user.getUsername());
            tmp=null;

            Connection con=serverAcc.connectMySQL();

            String command="delete from "+dbtable+" where Pusername='"+user.getUsername()+"' and
Ppwd='"+genHash(user.getPassword())+"'";
            Integer result=serverAcc.makeChange(command,con);
            serverAcc.closeMySQLConnection(con);

            result=null;
            con=null;
            command=null;

            request.getSession().invalidate();

            ret=true;
        }else{
            request.getSession().invalidate();
            ret=false;
        }

        user=null;
    }

```

```

        return ret;
    }

    public void delUser(HttpServletRequest request){
        String username=(String)request.getParameter("userSelected");
        if (!username.equals("")){
            HistoryManagement tmp=new HistoryManagement();
            tmp.delPlayerHistory(username);
            tmp=null;
            Integer result=null;
            Connection con=serverAcc.connectMySQL();
            String command="delete from "+dbtable+" where Pusername='"+username+"'";
            result=serverAcc.makeChange(command,con);
            serverAcc.closeMySQLConnection(con);

            result=null;
            con=null;
            command=null;
        }

        username=null;
    }

    public String genHash(String password){
        CRC32 temp=new CRC32();
        temp.update(password.getBytes(),0,password.length());
        Long pass=temp.getValue(); //CRC32 generation
        temp=null;
        password=null;
        return pass.toString();
    }
}

```

Πηγαίος Κώδικας JSP

LoginPage.jsp

```
<%@ page language="java"
    contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"
    import="ptixiaki.UserBean"
%>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html>
    <head>

        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
        <title>Login Form</title>

        <link href="css/style.css" rel="stylesheet" type="text/css" />

        <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.2.6/jquery.min.js"></script>
        <script type="text/javascript">
            $(document).ready(function() {
                $(".username").focus(function() {
                    $(".user-icon").css("left", "-48px");
                });
                $(".username").blur(function() {
                    $(".user-icon").css("left", "0px");
                });

                $(".password").focus(function() {
                    $(".pass-icon").css("left", "-48px");
                });
                $(".password").blur(function() {
                    $(".pass-icon").css("left", "0px");
                });
            });
        </script>
    </head>
</html>
```

```

function redirect() {
    window.location = "../registerPage.jsp"
}
</script>

<!--<%
    UserBean currentUser=new UserBean();
    currentUser=(UserBean)session.getAttribute("currentSessionUser");
    String errorMessage=new String();
    if (session.getAttribute("currentSessionUser")!=null) {
        errorMessage=
            "<div class=\"content\">"+
                "<span>"+
                    "<font color=red>"+
                        currentUser.getErrorOutput()+
                    "</font>"+
                "</span>"+
            "</div>";
    }
%>-->

</head>
<body>
    <div id="wrapper1">
        <div class="user-icon"></div>
        <div class="pass-icon"></div>
        <form name="game-form" class="game-form" action="UserManagement" method="post">
            <div class="header">
                <h1>Login Form</h1>
                <span>Fill out the form below to login to --The GAME--.</span>
            </div>
            <%=errorMessage%>
            <div class="content">
                <input name="userName" type="text" class="input username" value="Username"
onfocus="this.value=''" />
                <input name="passWord" type="password" class="input password" value="Password"
onfocus="this.value=''" />
                <input type="hidden" name="originID" value="login" />
            </div>
        </form>
    </div>

```

```

                <div class="footer">
                    <input type="submit" name="submit" value="Login" class="button"
style="width:110px;"/>
                    <input type="button" name="submit" value="Register" class="register"
onclick="redirect()" style="width:110px;"/>
                </div>
            </form>
        </div>
        <div class="gradient"></div>
        <!--<%
            if (session.getAttribute("currentSessionUser")!=null) {
                currentUser.setErrorOutput("");
                session.setAttribute("currentSessionUser",currentUser);
            }
        %>-->
    </body>
</html>

```


registerPage.jsp

```
<%@ page language="java"
    contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"
    import="ptixiaki.UserBean"
%>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
        <title>Login Form</title>
        <link href="css/style.css" rel="stylesheet" type="text/css" />
        <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.2.6/jquery.min.js"></script>
        <script type="text/javascript">
            $(document).ready(function() {
                $(".username").focus(function() {
                    $(".user-icon").css("left", "-48px");
                });
                $(".username").blur(function() {
                    $(".user-icon").css("left", "0px");
                });

                $(".password").focus(function() {
                    $(".pass-icon").css("left", "-48px");
                });
                $(".password").blur(function() {
                    $(".pass-icon").css("left", "0px");
                });
            });

            function redirect() {
                window.location = "../LoginPage.jsp"
            }
        </script>
```

```

<!--<%
    UserBean currentUser=new UserBean();
    currentUser=(UserBean)session.getAttribute("currentSessionUser");
    String errorMessage=new String();
    if (session.getAttribute("currentSessionUser")!=null){
        errorMessage=
            "<div class=\"content\">"+
                "<span>"+
                    "<font color=red>"+
                        currentUser.getErrorOutput()+
                    "</font>"+
                "</span>"+
            "</div>";
    }
%>-->

</head>
<body>
    <div id="wrapper1">
        <div class="user-icon"></div>
        <div class="pass-icon"></div>
        <form name="game-form" class="game-form" action="UserManagement" method="post">
            <div class="header">
                <h1>Register Form</h1>
                <span>Fill out the form below to register to --The GAME--.</span>
            </div>
            <%=errorMessage%>
            <div class="content">
                <input name="userName" type="text" class="input username" value="Username"
onfocus="this.value=''" />
                <input name="passWord" type="password" class="input password" value="Password"
onfocus="this.value=''" />
                <input type="hidden" name="originID" value="login" />
            </div>
            <div class="footer">
                <input type="button" name="submit" value="Login" class="register" onclick="redirect()"
style="width:110px;"/>
                <input type="submit" name="submit" value="Register" class="button"
style="width:110px;"/>
            </div>
        </form>
    </div>

```

```
        </div>
    </form>
</div>
<div class="gradient"></div>
<%
    if (session.getAttribute("currentSessionUser") != null) {
        currentUser.setErrorOutput("");
        session.setAttribute("currentSessionUser", currentUser);
    }
%>
</body>
</html>
```

userLogged.jsp

```
<%@ page language="java"
    contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"
    import="ptixiaki.UserBean"
%>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">

<html>
  <head>

    <!--META-->
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>User Logged Successfully</title>
    <!--STYLESHEETS-->
    <link href="css/style.css" rel="stylesheet" type="text/css" />
    <!--SCRIPTS-->
    <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.2.6/jquery.min.js"></script>
    <%
        int timeout=session.getMaxInactiveInterval();
        response.setHeader("Refresh", timeout + ";URL = LoginPage.jsp");
    %>

  </head>
  <body>
    <!--<%
        UserBean currentUser=(UserBean)session.getAttribute("currentSessionUser");
        String adminPanel=new String();

        if(currentUser.isAdmin()){
            adminPanel=
                "<form action=\"QuestionManagement\" method=\"POST\">"+
                "<input type=\"submit\" name=\"originID\" value=\"Add Question\"
class=\"button\" style=\"text-align: center; height: 50px; width: 247px;\">"+
```

```

                "<input type=\"submit\" name=\"originID\" value=\"Edit Question\"
class=\"button\" style=\"text-align: center; height: 50px; width: 247px;\">"+
                "<input type=\"submit\" name=\"originID\" value=\"Delete Question\"
class=\"button\" style=\"text-align: center; height: 50px; width: 247px;\">"+
                "</form>"+
            "</td>"+
            "<td style=\"width:33%\">"+
            "</td>"+
            "<td class=\"content\" style=\"width:33%\">"+
                "<form action=\"UserManagement\" method=\"POST\">"+
                    "<input type=\"submit\" name=\"submit\" value=\"Make Admin\" class=\"button\"
style=\"text-align: center; height: 50px; width: 247px;\">"+
                    "<input type=\"submit\" name=\"submit\" value=\"Delete Admin\" class=\"button\"
style=\"text-align: center; height: 50px; width: 247px;\">"+
                    "<input type=\"submit\" name=\"submit\" value=\"Delete User\" class=\"button\"
style=\"text-align: center; height: 50px; width: 247px;\">"+
                "</form>";
        }
%>-->
<table name="game-form" class="game-form" id="wrapper" style="margin-bottom: 5%;">
    <tr>
        <td>
            <div class="header" style="text-align: center;">
                <h1>
                    Welcome to THE GAME
                    <%= currentUser.getUsername() %>
                </h1>
                <span2>
                    From here you can :
                </span2>
            </div>
        </td>
    </tr>
    <tr>
    </tr>
    <tr>
        <td>
            <table border="0" class="content">
                <tr>

```

```

        <td class="content">
            <form action="HistoryManagement" method="POST">
                <input type="submit" name="originID" value="View your History"
class="button" style="text-align: center; height: 50px; width: 247px;">
                <input type="submit" name="originID" value="View High Scores"
class="button" style="text-align: center; height: 50px; width: 247px;">
                <input type="submit" name="originID" value="View All Recent
History" class="button" style="text-align: center; height: 50px; width: 247px;">
            </form>
        </td>
        <td class="content">
            <form action="GameLogic" method="POST">
                <input type="hidden" name="originID" value="startGamePlay" />
                <input type="submit" name="originID" value="Play -=The Game=-"
class="button" style="text-align: center; height: 50px; width: 247px;">
            </form>
        </td>
        <td class="content">
            <form action="UserManagement" method="POST">
                <input type="submit" name="submit" value="Change your
Password" class="button" style="text-align: center; height: 50px; width: 247px;">
                <input type="submit" name="submit" value="Delete your Account"
class="button" style="text-align: center; height: 50px; width: 247px;">
                <input type="submit" name="submit" value="Log Off"
class="button" style="text-align: center; height: 50px; width: 247px;">
            </form>
        </td>
    </tr>
</table>
</td>
</tr>
<tr>
<td>
    <table border="0" class="content">
        <tr>
            <td class="content" style="width:33%">
                <%= adminPanel %>
            </td>
        </tr>
    </table>

```

```
        </td>
    </tr>
    <tr>
        <td class="footer" style="text-align: center;">
        </td>
    </tr>
</table>
<div class="gradient"></div>
</body>
</html>
```

changeUserStatus.jsp

```
<!--<%@ page language="java"
    contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"
    import="ptixiaki.UserBean"
    import="java.util.ArrayList"
%>-->

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title> Change User Status -=ADMINISTRATOR=- </title>

    <!--STYLESHEETS-->
    <link href="css/style.css" rel="stylesheet" type="text/css" />

    <!--SCRIPTS-->
    <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.2.6/jquery.min.js"></script>
    <!--<%
        int timeout=session.getMaxInactiveInterval();
        response.setHeader("Refresh", timeout + ";URL = LoginPage.jsp");
    %>-->

  </head>
  <body>
    <!--<%
        UserBean currentUser=(UserBean)session.getAttribute("currentSessionUser");
        String buttonPushed=(String)session.getAttribute("submit");
        String pageList=new String();

        ArrayList userList=(ArrayList)session.getAttribute("userList");
        int numberOfUsers= userList.size();
        for (int i=0;i<numberOfUsers;i++){
            if (!currentUser.getUsername().equals((String)userList.get(i))){
```



```

        pageList=pageList+"<option value=\""+(String)userList.get(i)+"\"
>"+(String)userList.get(i)+"</option>\n";
    }
}

String pageButton=new String();
String useCase=new String();

if (buttonPushed.equals("Delete User")||buttonPushed.equals("Delete User!!")){
    pageButton="Delete User!!";
    useCase="delete a User!!";
}

if (buttonPushed.equals("Make Admin")||buttonPushed.equals("Make Administrator")){
    pageButton="Make Administrator";
    useCase="add an Administrator";
}

if (buttonPushed.equals("Delete Admin")||buttonPushed.equals("Delete Administrator!!")){
    pageButton="Delete Administrator!!";
    useCase="remove an Administrator!!";
}

%>-->
<br>
<br>
<table border="0" name="game-form" class="game-form" id="wrapper1" style="width: 380px; height:400px;
margin-left: -190px;">
    <form class="content" action="UserManagement" method="POST">
        <tr style="height:150px;">
            <th class="header" style="text-align: center;">
                <h1>From here you can </h1>
                <h1><%=useCase %></h1>
            </th>
        </tr>
        <tr style="text-align: center;">
            <td>
                <table border="0" style="width: 370px;">
                    <tr>

```

```

                                <td class="content" style="text-align: right; height:50px; padding:
0 0px 0px 0px;">
                                <span>
                                    <select class="input" name="userSelected" style="text-
align: left; width:166px; height:50px; padding: 0px 0px 0px 25px;">
                                        <option value="" selected></option>
                                        <%=pageList %>
                                    </select>
                                </span>
                                </td>
                                <td style="text-align: right; width:150px; height:50px;">
                                    <div class="content" style="padding: 0 0px 0px 0px;">
                                        <input type="submit" class="button" name="submit"
style="text-align: center; width: 200px;" value="<%=pageButton %>">
                                        </div>
                                </td>
                            </tr>
                        </table>
                    </td>
                </tr>
                <tr>
                    <td class="footer" style="padding: 0 0px 0px 0px;">
                        <input type="submit" class="button" name="submit" value="Back">
                    </td>
                </tr>
            </form>
        </table>
        <div class="gradient"></div>
    </body>
</html>

```

deleteAccount.jsp

```
<!--<%@ page language="java"
    contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"
    import="ptixiaki.UserBean"
%>-->

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Account Delete</title>

    <!--STYLESHEETS-->
    <link href="css/style.css" rel="stylesheet" type="text/css" />

    <!--SCRIPTS-->
    <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.2.6/jquery.min.js"></script>
    <!--<%
        int timeout=session.getMaxInactiveInterval();
        response.setHeader("Refresh", timeout + ";URL = LoginPage.jsp");
%>-->
  </head>

  <body>

    <table border="0" name="game-form" class="game-form" id="wrapper" style="width:320px; max-width:320;
margin-top: 7%; margin-bottom: 7%">
      <form action="UserManagement" method="POST">
        <tr>
          <th class="header" style="text-align: center;">

            <h1>
              <font color=red>WARNING!!!</font>
            </h1>
          </th>
        </tr>
      </form>
    </table>
  </body>
</html>
```

```

        <span2><font color=red>
            You are about to delete your account <br>
            witch will also delete all of your data (High Scores etc...)
        </font></span2>
    </th>
</tr>
<tr>
    <td class="content">
        <span>
            You need to confirm your identity by retying your username and password
        </span>
    </td>
</tr>
<tr>
    <td class="content">
        <span>
            Enter your username
        </span>
    </td>
</tr>
<tr>
    <td class="content">
        <input type="text" class="input" name="userName" style="padding: 0px 10px
0px 10px; text-align: left; width:300px; height:50px;"/>
    </td>
</tr>
<tr>
    <td class="content">
        <span>
            Enter your password
        </span>
    </td>
</tr>
<tr>
    <td class="content">
        <input type="password" class="input" name="passWord" style="padding: 0px 10px
0px 10px; text-align: left; width:300px; height:50px;"/>
    </td>
</tr>

```

```

        <tr>
            <td class="footer" style="padding: 0px 0px 0px 0px;">
                <table border="0" style="width: 344px; padding: 10px 10px 10px 10px;" >
                    <tr>
                        <td>
                            <input type="submit" class="button" name="submit" style="text-align: center; width: 180px;" value="Delete My Account"/>
                        </td>
                        <td>
                            <input type="submit" class="button" name="submit" style="text-align: center; width: 180px;" value="Back">
                        </td>
                    </tr>
                </table>
            </td>
        </tr>
    </form>
</table>
<div class="gradient"></div>
</body>
</html>

```

deleteNotice.jsp

```
<!--<%@ page language="java"
    contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"
%>-->

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Account Delete</title>
    <link href="css/style.css" rel="stylesheet" type="text/css" />
    <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.2.6/jquery.min.js"></script>

    <script type="text/javascript">
      function redirect1() {
        window.location = "./LoginPage.jsp"
      }

      function redirect2() {
        window.location = "./registerPage.jsp"
      }

    </script>
  </head>

  <body>
    <div id="wrapper1">
      <form name="game-form" class="game-form" action="GameLogic" method="post">
        <div class="header">
          <center><h1>Your account is no more!!!</h1></center>
        </div>
        <div class="content">
          <span>From here on you can either return to the real world</span2>
          <span> or choose one of the options below. </span2>
        </div>
      </form>
    </div>
  </body>
</html>
```

```

        </div>
        <div class="footer">
            <table border="0">
                <tr>
                    <td>
                        <input type="submit" value="Register" class="button"
onclick="redirect2()" style="width:117px;"/>
                    </td>
                    <td>
                        <input type="submit" value="Log In" class="button"
onclick="redirect1()" style="width:117px;"/>
                    </td>
                </tr>
            </table>
        </div>
    </form>
</div>
<div class="gradient"></div>
</body>
</html>

```

displayHistory.jsp

```
<!--<%@ page language="java"
    contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"
    import="ptixiaki.UserBean"
    import="java.util.ArrayList"
%>-->

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1256">
    <title> History View </title>

    <!--STYLESHEETS-->
    <link href="css/style.css" rel="stylesheet" type="text/css" />

    <!--SCRIPTS-->
    <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.2.6/jquery.min.js"></script>
    <!--<%
        int timeout=session.getMaxInactiveInterval();
        response.setHeader("Refresh", timeout + ";URL = LoginPage.jsp");
    %>-->
  </head>
  <body>
    <!--<%
        String useCase=new String();
        Object[][] historyObject=(Object[][] )session.getAttribute("historyObject");

        String historyDecoded=new String();

        if (session.getAttribute("originID").equals("View your History")){
            // Player History View
            useCase="This is your history";
        }
    %>-->
  </body>
</html>
```



```

if (session.getAttribute("originID").equals("View High Scores")){
    // High Scores View
    useCase="High Scores View";
}
if (session.getAttribute("originID").equals("View All Recent History")){
    // Game History View
    useCase="Game History View";
}

for(int i=0;i<historyObject.length;i++){
    //
                                Pusername
                                PlayedAt
                                qstAnswd
                                moneyGained
    historyDecoded=historyDecoded+"Username: "+historyObject[i][1]+" , Played at:
"+historyObject[i][2]+" , Qst answered: "+historyObject[i][3]+" , Money Gained: "+historyObject[i][4]+" \n";
}

session.removeAttribute("historyObject");

%>-->

<table border="0" name="game-form" class="game-form" id="wrapper" style="width:740px; max-width:760;
margin-top: 30px; margin-bottom: 30px% padding: 10px 10px 10px 10px;">
<tr>
<td>
<table class="header" style="width:720px; max-width:720; padding: 10px 0px 10px
10px;">
<tr>
<td style="width:10%;">
<form action="HistoryManagement" method="POST">
<input type="submit" class="button" name="originID"
value="Back">
</form>
</td>
<th style="width:80%; text-align: center;">
<h1>
<%=useCase %>
</h1>

```

```

                </th>
                <td style="width:10%;">
                    <form action="HistoryManagement" method="POST">
                        <input type="submit" class="button" name="originID"
value="Back">
                    </form>
                </td>
            </tr>
        </table>
    </td>
</tr>
<tr>
    <td style="text-align: left; height: 150px; max-height: 151px; width: 100%;"
class="content">
        <textarea readonly class="input_game" name="questionBox" style="text-align: justify;
resize: none; width: 670px; height:500px" ><%=historyDecoded %></textarea>
    </td>
</tr>
<tr>
    <td>
        <table border="0" class="footer" style="text-align:center; width:100%; max-width:100%;
padding: 10px 10px 10px 10px;">
            <tr>
                <td style="width:10%;">
                    <form action="HistoryManagement" method="POST">
                        <input type="submit" class="button" name="originID"
value="Back">
                    </form>
                </td>
                <td style="width:80%;">
                    <!-- EMPTY -->
                </td>
                <td style="width:10%;">
                    <form action="HistoryManagement" method="POST">
                        <input type="submit" class="button" name="originID"
value="Back">
                    </form>
                </td>
            </tr>
        </table>

```

```
        </td>
    </tr>
</table>
<div class="gradient"></div>
</body>
</html>
```

editQuestions.jsp

```
<!--<%@ page language="java"
    contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"
    import="ptixiaki.UserBean"
    import="ptixiaki.GameBean"
    import="java.util.ArrayList"
    import="java.lang.Integer"
%>-->

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=windows-1256">
        <title>Manage Questions</title>

        <!--STYLESHEETS-->
        <link href="css/style.css" rel="stylesheet" type="text/css" />
        <!--SCRIPTS-->
        <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.2.6/jquery.min.js"></script>
        <!--<%
            int timeout=session.getMaxInactiveInterval();
            response.setHeader("Refresh", timeout + ";URL = LoginPage.jsp");
        %>-->
    </head>

    <body>
        <!--<%

                String useCase=new String();
                String originID=(String)session.getAttribute("originID");
                String readOnlySwitch=new String();
                String actionButton=new String();
                String previousButton=new String();
```

```

String nextButton=new String();
String valueFill=new String();
Integer qID=null;
String question=new String();
String ans1=new String();
String ans2=new String();
String ans3=new String();
String ans4=new String();
String difficultyLevel=new String();
Object[] questionsObject=(Object[])session.getAttribute("questionsObject");

readOnlySwitch="";

%>-->

<!--<%
    if (!(originID.equals("Add Question"))&&!(originID.equals("addQuestion"))){
        previousButton="<input type=\"submit\" class=\"button\" name=\"buttonID\" value=\"Previous\"
style=\"text-align: center; height: 50px; width: 130px;\>";
        nextButton="<input type=\"submit\" class=\"button\" name=\"buttonID\" value=\"Next\"
style=\"text-align: center; height: 50px; width: 130px;\>";

        if ((String)questionsObject[6].equalsIgnoreCase("Low")){
            valueFill="<option value=\"Low\" selected>Low</option>\n";
        }else{
            valueFill="<option value=\"Low\">Low</option>\n";
        }

        if ((String)questionsObject[6].equalsIgnoreCase("Medium")){
            valueFill=valueFill+"<option value=\"Medium\" selected>Medium</option>\n";
        }else{
            valueFill=valueFill+"<option value=\"Medium\">Medium</option>\n";
        }

        if ((String)questionsObject[6].equalsIgnoreCase("High")){
            valueFill=valueFill+"<option value=\"High\" selected>High</option>\n";
        }else{
            valueFill=valueFill+"<option value=\"High\">High</option>\n";
        }
}

```

```

    }
%>-->

<!--<%
    if ( originID.equals("Add Question") || originID.equals("addQuestion") ){
        useCase="ADD A QUESTION";
        actionButton="Confirm";
        originID="addQuestion";

        question="Type the Question IN HERE";
        ans1="THIS IS THE CORRECT ANSWER";
        ans2="This is the 1st wrong answer";
        ans3="This is the 2nd wrong answer";
        ans4="This is the 3rd wrong answer";
        valueFill="<option value=\"Low\" selected>Low</option>\n";
        valueFill=valueFill+"<option value=\"Medium\">Medium</option>\n";
        valueFill=valueFill+"<option value=\"High\">High</option>\n";
    }
%>-->

<!--<%
    if ( originID.equals("Edit Question") || originID.equals("editQuestion") ){
        useCase="EDIT A QUESTION";
        actionButton="Confirm";
        originID="editQuestion";

        qID=Integer.parseInt( (String) questionsObject[0] );
        question=(String) questionsObject[1];
        ans1=(String) questionsObject[2];
        ans2=(String) questionsObject[3];
        ans3=(String) questionsObject[4];
        ans4=(String) questionsObject[5];
        difficultyLevel=(String) questionsObject[6];
    }
%>-->

<!--<%
    if ( originID.equals("Delete Question") ||originID.equals("deleteQuestion") ) {
        useCase="DELETE A QUESTION";
        actionButton="Delete";
    }
%>-->

```

```

originID="deleteQuestion";
readOnlySwitch="readonly";

qID=Integer.parseInt((String)questionsObject[0]);
question=(String)questionsObject[1];
ans1=(String)questionsObject[2];
ans2=(String)questionsObject[3];
ans3=(String)questionsObject[4];
ans4=(String)questionsObject[5];
difficultyLevel=(String)questionsObject[6];
}
%>-->
<form action="QuestionManagement" method="POST">
    <table border="1" name="game-form" class="game-form" id="wrapper" style="text-align: center;
height: 602px; width: 580px;">
        <tr>
            <th class="header">
                <h1><%=useCase %></h1>
            </th>
        </tr>
        <tr style="text-align: center;">
            <td>
                <table border="1" >
                    <tr>
                        <td style="text-align: left; width:282px;" class="content">
                            <span>QID: <input class="input_game" type="text"
name="qIDText" style="text-align: right; width: 50px; height:20px;" value="<%=qID%>"/></span>
                            <input type="submit" class="button" name="buttonID" value="Go
to Question" style="text-align: center; height: 50px; width: 130px;">
                        </td>
                        <td style="text-align: right; width:270px;" class="content">
                            <span>Difficulty Level :
                            <select name="difficultyLevel" class="input_game" style="text-
align: right; width:60px; height:20px;">
                                <%=valueFill %>
                            </select></span>
                        </td>
                    </tr>
                </table>
            </td>
        </tr>
    </table>
</td>

```

```

        </tr>
        <tr>
            <td style="text-align: left; height: 150px; max-height: 151px; width: 100%;"
class="content">
                <textarea class="input_game" rows="8" cols="64" name="questionBox" style="text-
align: left; resize: none; width: 512px;" <%=readOnlySwitch %> > <%=question %></textarea>
            </td>
        </tr>
        <tr style="text-align: center; height: 100px; max-height: 105px; width: 100%;">
            <td>
                <table border="0" style="text-align: center; height: 104px; max-height: 105px;
width: 574px;" class="content">
                    <tr>
                        <td>
                            <textarea class="input_game" rows="3" cols="29" name="answer1"
style="text-align: center; resize: none; width: 252px;" <%=readOnlySwitch %> ><%=ans1 %></textarea>
                        </td>
                        <td>
                            <textarea class="input_game" rows="3" cols="29" name="answer2"
style="text-align: center; resize: none; width: 252px;" <%=readOnlySwitch %> ><%=ans2 %></textarea>
                        </td>
                    </tr>
                    <tr>
                        <td>
                            <textarea class="input_game" rows="3" cols="29" name="answer3"
style="text-align: center; resize: none; width: 252px;" <%=readOnlySwitch %> ><%=ans3 %></textarea>
                        </td>
                        <td>
                            <textarea class="input_game" rows="3" cols="29" name="answer4"
style="text-align: center; resize: none; width: 252px;" <%=readOnlySwitch %> ><%=ans4 %></textarea>
                        </td>
                    </tr>
                </table>
            </td>
        </tr>
        <tr style="text-align: center; height: 74px; width: 100%;">
            <td style="height: 60px; width: 100%;">
                <table border="0" style="text-align: center; height: 70px; width: 100%;"
class="footer">
                    <tr>

```


gamePlay.jsp

```
<!--<%@ page language="java"
    contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"
    import="ptixiaki.UserBean"
    import="ptixiaki.GameBean"
    import="java.util.ArrayList"
    import="java.lang.Integer"
%>-->

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
        <title> Who Wants To Be A Millionaire </title>

        <link href="css/style.css" rel="stylesheet" type="text/css" />

        <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.2.6/jquery.min.js"></script>
        <!--<%
            int timeout=session.getMaxInactiveInterval();
            response.setHeader("Refresh", timeout + ";URL = LoginPage.jsp");
        %>-->

    </head>

    <body>

        <!--<%
            String currentQuestion=null;
            String currentLevel=null;
            String question=new String();
            String ans1=new String();
            String ans2=new String();
```

```

String ans3=new String();
String ans4=new String();

GameBean gameQuestion=(GameBean) session.getAttribute("currentGame");
currentQuestion=gameQuestion.getCurrentQuestion();
currentLevel=gameQuestion.getCurrentLevel();

question=(String) gameQuestion.getGameQuestion(Integer.parseInt(currentQuestion)-1)[1];
ans1=(String) gameQuestion.getGameQuestion(Integer.parseInt(currentQuestion)-
1)[((Integer) gameQuestion.getAnswerOrder(Integer.parseInt(currentQuestion)-1)[0])+1];
ans2=(String) gameQuestion.getGameQuestion(Integer.parseInt(currentQuestion)-
1)[((Integer) gameQuestion.getAnswerOrder(Integer.parseInt(currentQuestion)-1)[1])+1];
ans3=(String) gameQuestion.getGameQuestion(Integer.parseInt(currentQuestion)-
1)[((Integer) gameQuestion.getAnswerOrder(Integer.parseInt(currentQuestion)-1)[2])+1];
ans4=(String) gameQuestion.getGameQuestion(Integer.parseInt(currentQuestion)-
1)[((Integer) gameQuestion.getAnswerOrder(Integer.parseInt(currentQuestion)-1)[3])+1];

%>-->
<!--<%
Boolean[] helpStatus=gameQuestion.getHelpStatus();
Boolean[] fromHelp=gameQuestion.getFromHelp();

String disable50="";
String disableStats="";
String disableChoise1="";
String disableChoise2="";
String disableChoise3="";
String disableChoise4="";

if(helpStatus[0]==true){
    disable50="";
}else{
    disable50="DISABLED";
}

if(helpStatus[1]==true){
    disableStats="";
}else{
    disableStats="DISABLED";
}

```

```

        if (fromHelp[0]==true){
            Integer[]
answerOrder=(Integer[]) gameQuestion.getAnswerOrder(Integer.parseInt(currentQuestion)-1);
            Integer[] answersFromHelp=gameQuestion.getAnswersFromHelp();

            if((answerOrder[0]!=answersFromHelp[0]) && (answerOrder[0]!=answersFromHelp[1])){
                ans1="";
                disableChoise1="DISABLED";
            };
            if((answerOrder[1]!=answersFromHelp[0]) && (answerOrder[1]!=answersFromHelp[1])){
                ans2="";
                disableChoise2="DISABLED";
            };
            if((answerOrder[2]!=answersFromHelp[0]) && (answerOrder[2]!=answersFromHelp[1])){
                ans3="";
                disableChoise3="DISABLED";
            };
            if((answerOrder[3]!=answersFromHelp[0]) && (answerOrder[3]!=answersFromHelp[1])){
                ans4="";
                disableChoise4="DISABLED";
            };

            fromHelp[0]=false;
            gameQuestion.setFromHelp(fromHelp);
            request.getSession().removeAttribute("currentGame");
            request.getSession().setAttribute("currentGame",gameQuestion);
        }

%>-->
<!--<%
        if (fromHelp[1]==true){
            Integer[]
answerOrder=(Integer[]) gameQuestion.getAnswerOrder(Integer.parseInt(currentQuestion)-1);
            String[] tmp=(String[]) gameQuestion.getGameQuestion(Integer.parseInt(currentQuestion)-1);

            int ans1Stats=Integer.parseInt(tmp[answerOrder[0]+6]);
            int ans2Stats=Integer.parseInt(tmp[answerOrder[1]+6]);
            int ans3Stats=Integer.parseInt(tmp[answerOrder[2]+6]);

```

```

int ans4Stats=Integer.parseInt(tmp[answerOrder[3]+6]);

ans1=ans1+"\n\n"+ans1Stats;
ans2=ans2+"\n\n"+ans2Stats;
ans3=ans3+"\n\n"+ans3Stats;
ans4=ans4+"\n\n"+ans4Stats;

fromHelp[1]=false;
gameQuestion.setFromHelp(fromHelp);
request.getSession().removeAttribute("currentGame");
request.getSession().setAttribute("currentGame",gameQuestion);
}
%>-->

<center>
    Debug Info:<br>
    Button Pressed: <%=session.getAttribute("buttonID") %><br>
</center>

<script type="text/javascript">
    function disable(id){
        document.getElementById(id).disabled = 'disabled';
    }
    function enable(id){
        document.getElementById(id).disabled = '';
    }
</script>

<form action="GameLogic" method="POST">
    <table border="1" name="game-form" class="game-form" id="wrapper" style="text-align: center;
height: 396px; max-height: 480px; width: 580px;">
        <tr>
            <td class="header">
                <h1>
                    This is .... -=THE GAME=-
                </h1>
            </td>
        </tr>
    </table>

```

```

<tr style="text-align: center;">
  <td>
    <table border="1" >
      <tr>
        <td style="text-align: left; width: 270px;" class="content">
          <span>Current Question : </span>
          <input class="input_game" type="text" name="currentQuestion"
readonly style="text-align: right; width: 30px; height: 20px;" value="<%=currentQuestion %>"/>
        </td>
        <td style="text-align: right; width:380px;" class="content">
          <span>You are playing for: </span>
          <input class="input_game" type="text" name="currentLevel"
readonly style="text-align: right; width: 95px; height: 20px;" value=" <%=currentLevel %> " />
        </td>
      </tr>
    </table>
  </td>
</tr>
<tr>
  <td style="text-align: left; height: 150px; max-height: 151px; width: 100%;"
class="content">
    <textarea class="input_game" rows="8" cols="64" name="questionBox"
style="padding: 10px 15px; text-align: justify; resize: none; width: 512px; " readonly ><%=question %></textarea>
  </td>
</tr>
<tr style="text-align: center; height: 100px; max-height: 105px; width: 100%;">
  <td>
    <table border="0" class="content" style="text-align: center; height: 104px; max-
height: 105px; width: 574px;">
      <tr>
        <td style="height: 100%; width: 10px;">
          <input type="radio" name="playerChoise" <%=disableChoise1 %>
value="1" onclick='enable("Confirm")' >
        </td>
        <td>
          <textarea class="input_game" rows="3" cols="35" name="answer1"
style="padding: 10px 15px; text-align: justify; resize: none; width: 220px" readonly ><%=ans1 %></textarea>
        </td>
      </tr>
    </table>
  </td>
</tr>

```

```

                                <textarea class="input_game" rows="3" cols="35" name="answer2"
style="padding: 10px 15px; text-align: justify; resize: none; width: 220px" readonly ><%=ans2 %></textarea>
                                </td>
                                <td style="height: 100%; width: 10px;">
                                <input type="radio" name="playerChoise" <%=disableChoise2 %>
value="2" onclick='enable("Confirm")' >
                                </td>
                                </tr>
                                <tr>
                                <td style="height: 100%; width: 10px;">
                                <input type="radio" name="playerChoise" <%=disableChoise3 %>
value="3" onclick='enable("Confirm")' >
                                </td>
                                <td>
                                <textarea class="input_game" rows="3" cols="35" name="answer3"
style="padding: 10px 15px; text-align: justify; resize: none; width: 220px" readonly ><%=ans3 %></textarea>
                                </td>
                                <td>
                                <textarea class="input_game" rows="3" cols="35" name="answer4"
style="padding: 10px 15px; text-align: justify; resize: none; width: 220px" readonly ><%=ans4 %></textarea>
                                </td>
                                <td style="height: 100%; width: 10px;">
                                <input type="radio" name="playerChoise" <%=disableChoise4 %>
value="4" onclick='enable("Confirm")' >
                                </td>
                                </tr>
                                </table>
                                </td>
                                </tr>
                                <tr style="text-align: center; height: 74px; width: 100%;">
                                <td style="height: 60px; width: 100%;">
                                <table border="0" style="text-align: center; height: 70px; width: 100%;"
class="content">
                                <tr>
                                <td>
                                <input type="submit" class="button" <%=disable50 %>
name="buttonID" value="50-50" style="text-align: center; height: 50px; width: 200px;">
                                </td>
                                <td>

```

```

                                <input type="submit" class="button" <%=disableStats %>
name="buttonID" value="Ask Me" style="text-align: center; height: 50px; width: 200px;">
                                </td>
                                </tr>
                                <tr>
                                <td>
                                <input type="submit" class="button" name="buttonID"
value="Cancel" style="text-align: center; height: 50px; width: 200px;">
                                </td>
                                <td>
                                <input type="submit" class="button" DISABLED id="Confirm"
name="buttonID" value="Confirm Answer" style="text-align: center; height: 50px; width: 200px;">
                                </td>
                                </tr>
                                </table>
                                </td>
                                </tr>
                                </table>
                                <input type="hidden" name="originID" value="gamePlay" />
                                </form>
                                <div class="gradient"></div>
                                </body>
</html>

```


losePage.jsp

```
<!--<%@ page language="java"
    contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"
    import="ptixiaki.UserBean"
    import="ptixiaki.GameBean"
    import="java.util.ArrayList"
    import="java.lang.Integer"
%>-->

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
        <title>I'm Sorry</title>
        <link href="css/style.css" rel="stylesheet" type="text/css" />
        <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.2.6/jquery.min.js"></script>

        <!--<%
            GameBean gameQuestion=(GameBean) session.getAttribute("currentGame");
            String currentUser=((UserBean) session.getAttribute("currentSessionUser")).getUsername();
            Integer currentScore=Integer.parseInt((String)gameQuestion.getCurrentLevel());
            String currentQuestion=gameQuestion.getCurrentQuestion();
            String question=new String();
            String ans=new String();
            question=(String)gameQuestion.getGameQuestion(Integer.parseInt(currentQuestion))[1];
            ans=(String)gameQuestion.getGameQuestion(Integer.parseInt(currentQuestion))[2];

            if (currentScore<1000){
                currentScore=0;
            }else if (currentScore<32000){
                currentScore=1000;
            }else{
                currentScore=32000;
            }
        %>-->
```

```

        int timeout=session.getMaxInactiveInterval();
        response.setHeader("Refresh", timeout + ";URL = LoginPage.jsp");
    %>-->
</head>
<body>
    <div id="wrapper1">
        <form name="game-form" class="game-form" action="GameLogic" method="post">
            <div class="header">
                <center><h1>I'm sorry you LOSE!!</h1></center>
                <center><h1> <%=currentUser%> </h1></center>
                <span2 style="text-align:justify;">it appears as though you lost</span2>
            </div>
            <div class="content">
                <span2>The correct answer to the question:</span2>
                <span2><%=question%></span2>
                <span2>was</span2>
                <span2><%=ans%></span2>
                <span2><br></span2>
                <span2>Your score was : <%=currentScore%></span2>
                <span2>Better luck next time</span2>
            </div>
            <div class="footer">
                <input type="submit" name="originID" value="Back" class="button"
style="width:110px;" />
                <input type="submit" name="originID" value="Play Again" class="button"
style="width:110px;" />
            </div>
        </form>
    </div>
    <div class="gradient"></div>
</body>
</html>

```

pwdChange.jsp

```
<%@ page language="java"
    contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"
    import="ptixiaki.UserBean"
%>
<jsp:useBean id="UserBean" class="ptixiaki.UserBean" scope="request"/>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Password Change Utility!!</title>

    <!--STYLESHEETS-->
    <link href="css/style.css" rel="stylesheet" type="text/css" />

    <!--SCRIPTS-->
    <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.2.6/jquery.min.js"></script>

    <!--<%
        UserBean currentUser=(UserBean)session.getAttribute("currentSessionUser");

        int timeout=session.getMaxInactiveInterval();
        response.setHeader("Refresh", timeout + ";URL = LoginPage.jsp");
    %>-->

  </head>

  <body>

    <table border="0" name="game-form" class="game-form" id="wrapper1" style="width: 350px; height:200px;
margin-left: -175px; margin-top: -340px;">
      <form action="UserManagement" method="POST" id=pwdChange name=pwdChange >
        <tr>
```

```

        <th class="header">
            <h1>From here you can change your password.</h1>
        </th>
    </tr>
    <tr>
        <td class="content" style="padding: 0px 0px 0px 10px;">
            <span>Please enter your current password:</span>
        </td>
    </tr>
    <tr>
        <td class="content" style="padding: 10px 0px 10px 0px;">
            <center>
                <input type="password" style="text-align: left; width:320px; height:50px;"
class="input_game" name="oldPassword" value="" />
            </center>
        </td>
    </tr>
    <tr>
        <td class="content" style="padding: 0px 0px 0px 10px;">
            <span>Please enter your new password:</span>
        </td>
    </tr>
    <tr>
        <td class="content" style="padding: 10px 0px 10px 0px;">
            <center>
                <input type="password" style="text-align: left; width:320px; height:50px;"
class="input_game" name="newPassword1" value="" />
            </center>
            <span>
                <font color=red><%= currentUser.getErrorOutput() %></font>
            </span>
        </td>
    </tr>
    <tr>
        <td class="content" style="padding: 0px 0px 0px 10px;">
            <span>Please verify your new password:</span>
        </td>
    </tr>
    <tr>
        <td class="content" style="padding: 10px 0px 10px 0px;">

```

```

                <center>
                    <input type="password" style="text-align: left; width:320px; height:50px;"
class="input_game" name="newPassword2" value="" />
                </center>
            </td>
        </tr>
        <tr>
            <td class="footer" style=" padding: 0px 0px 0px 0px;">
                <table border="0" style="width: 344px; padding: 10px 10px 10px 10px;" >
                    <tr>
                        <td>
                            <input type="submit" class="button" name="submit"
value="Verify" style="text-align: center; width: 150px;">
                        </td>
                        <td>
                            <input type="submit" class="button" name="submit"
value="Back" style="text-align: center; width: 150px;">
                        </td>
                    </tr>
                </table>
            </td>
        </tr>
    </form>
</table>
<div class="gradient"></div>
</body>
</html>

```

winPage.jsp

```
<!--<%@ page language="java"
    contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"
    import="ptixiaki.UserBean"
%>-->

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html>
  <head>

    <!--META-->
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Congratsulations!!!</title>

    <!--STYLESHEETS-->
    <link href="css/style.css" rel="stylesheet" type="text/css" />

    <!--SCRIPTS-->
    <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.2.6/jquery.min.js"></script>

    <!--<%
        int timeout=session.getMaxInactiveInterval();
        response.setHeader("Refresh", timeout + ";URL = LoginPage.jsp");

        String currentUser=((UserBean) session.getAttribute("currentSessionUser")).getUsername();
    %>-->

  </head>
  <body>
    <div id="wrapper1">
      <div class="user-icon"></div>
      <div class="pass-icon"></div>
      <form name="game-form" class="game-form" action="GameLogic" method="post">
        <div class="header">
          <center><h1>Congratulacions for winning</h1></center>
        </div>
      </form>
    </div>
  </body>
</html>
```

```

                <center><h1> <%=currentUser%> </h1></center>
                <span2 style="text-align:justify;">Your Victory record will from now on be available
for all to envy and admire</span2>
            </div>
            <div class="footer">
                <input type="submit" name="originID" value="Back" class="button"
style="width:110px;" />
                <input type="submit" name="originID" value="Play Again" class="button"
style="width:110px;" />
            </div>
        </form>
    </div>
    <div class="gradient"></div>
</body>
</html>

```

withdrawPage.jsp

```
<!--<%@ page language="java"
    contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"
    import="ptixiaki.UserBean"
    import="ptixiaki.GameBean"
    import="java.util.ArrayList"
    import="java.lang.Integer"
%>-->

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html>
    <head>

        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
        <title>I'm Sorry</title>

        <!--STYLESHEETS-->
        <link href="css/style.css" rel="stylesheet" type="text/css" />

        <!--SCRIPTS-->
        <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.2.6/jquery.min.js"></script>

        <!--<%
            GameBean gameQuestion=(GameBean) session.getAttribute("currentGame");
            String currentUser=((UserBean) session.getAttribute("currentSessionUser")).getUsername();
            Integer currentScore=Integer.parseInt((String) gameQuestion.getCurrentLevel());
            String currentQuestion=gameQuestion.getCurrentQuestion();
            String question=new String();
            String ans=new String();
            question=(String) gameQuestion.getGameQuestion(Integer.parseInt(currentQuestion))[1];
            ans=(String) gameQuestion.getGameQuestion(Integer.parseInt(currentQuestion))[2];

            int timeout=session.getMaxInactiveInterval();
            response.setHeader("Refresh", timeout + ";URL = LoginPage.jsp");
%>-->
```



```

</head>
<body>
  currentQuestion=<%=currentQuestion%>

  <div id="wrapper1">
    <div class="user-icon"></div>
    <div class="pass-icon"></div>
    <form name="game-form" class="game-form" action="GameLogic" method="post">
      <div class="header">
        <center><h1>Too bad you had to withdraw</h1></center>
        <center><h1> <%=currentUser%> </h1></center>
      </div>
      <div class="content">
        <span2>The correct answer to the question:</span2>
        <span2><%=question%></span2>
        <span2>was</span2>
        <span2><%=ans%></span2>
        <span2><br></span2>
        <span2>Your score was : <%=currentScore%></span2>
        <span2>Better luck next time</span2>
      </div>
      <div class="footer">
        <input type="submit" name="originID" value="Back" class="button"
style="width:110px;"/>
        <input type="submit" name="originID" value="Play Again" class="button"
style="width:110px;"/>
      </div>
    </form>
  </div>
  <div class="gradient"></div>
</body>
</html>

```

