



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Ανάλυση, σχεδιασμός και υλοποίηση εκπαιδευτικού πληροφοριακού συστήματος για τη μεταδευτεροβάθμια εκπαίδευση Analysis, design and development of an educational information system for post secondary education
Όνοματεπώνυμο Φοιτητή	Θρασύβουλος Βοΐλας
Πατρώνυμο	Κωνσταντίνος
Αριθμός Μητρώου	ΜΠΣΠ/ 13017
Επιβλέπων	Ευθύμιος Αλέπης, Επίκουρος καθηγητής

Ημερομηνία Παράδοσης **9ος 2016**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Ευθύμιος Αλέπης
Επίκουρος Καθηγητής

Μαρία Βίρβου
Καθηγήτρια

Γεώργιος Τσιχριντζής
Καθηγητής

Copyright © 2016, Θρασύβουλος Βοΐλας

Με την επιφύλαξη παντός δικαιώματος. Δεν επιτρέπεται η αναπαραγωγή κανενός τμήματος αυτής της εργασίας, σε οποιαδήποτε μορφή, με τη χρήση ηλεκτρονικών ή μηχανικών μέσων. Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ' ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό χωρίς την έγγραφη άδεια του συγγραφέα. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για μη κερδοσκοπικό σκοπό ερευνητικής ή εκπαιδευτικής φύσης, με τον όρο να αναφέρεται η πηγή προέλευσης.

Οι απόψεις και τα συμπεράσματα που παρουσιάζονται στην παρούσα εργασία εκφράζουν αποκλειστικά το συγγραφέα και δεν αποτελούν απαραίτητα επίσημες θέσεις της τριμελούς εξεταστικής επιτροπής, του τμήματος Πληροφορικής και του Πανεπιστημίου Πειραιώς.

Θρασύβουλος Βοΐλας
Ηλεκτρολόγος Μηχανικός &
Μηχανικός Τεχνολογίας Υπολογιστών
Πτυχιούχος Πολυτεχνικής Σχολής
Πανεπιστημίου Πατρών

Πρόλογος-Ευχαριστίες

Η εργασία αυτή εκπονήθηκε στα πλαίσια του μεταπτυχιακού προγράμματος σπουδών Προηγμένα Συστήματα Πληροφορικής (ΜΠΣΠ) του Πανεπιστημίου Πειραιώς κατά το ακαδημαϊκό έτος 2015-2016. Το αποτέλεσμα αυτής της προσπάθειας ήταν η υλοποίηση του εκπαιδευτικού λογισμικού iTestU.

Κατ' αρχάς θα ήθελα να ευχαριστήσω τον επίκουρο καθηγητή κο Ευθύμιο Αλέπη για τη βοήθεια και τις εύστοχες παρατηρήσεις κατά τη διάρκεια επίβλεψης της παρούσας μεταπτυχιακής διατριβής.

Θα ήθελα να ευχαριστήσω θερμά την καθηγήτρια και πρόεδρο του τμήματος πληροφορικής κα Μαρία Βίρβου και τον καθηγητή και πρόεδρο του μεταπτυχιακού προγράμματος σπουδών κο Γεώργιο Τσιχριντζή για την εμπιστοσύνη που έδειξαν στο πρόσωπό μου κατά τη διαδικασία επιλογής στο πρόγραμμα μεταπτυχιακό σπουδών.

Επίσης θα ήθελα να ευχαριστήσω όλους τους καθηγητές, το διδακτικό και ερευνητικό προσωπικό για την προσπάθεια που κατέβαλαν να μας μεταδώσουν τις γνώσεις και την εμπειρία τους καθ' όλη τη διάρκεια φοίτησης στο πρόγραμμα.

Θα ήθελα να ευχαριστήσω τους σπουδαστές του Δημοσίου Ινστιτούτου Επαγγελματικής Κατάρτισης - ΔΙΕΚ Σαλαμίνας και τους σπουδαστές του Κέντρου Ελευθέρων Σπουδών & Πληροφορικής Eurolab Καλλιθέας για τη συνεισφορά και τις εύστοχες παρατηρήσεις τους κατά το στάδιο ελέγχου της εφαρμογής.

Τέλος, δε θα μπορούσα να μην πω, ένα μεγάλο ευχαριστώ και να αφιερώσω την παρούσα μεταπτυχιακή διατριβή στην οικογένειά μου για την πολύτιμη υλική και ηθική αρωγή τους, χωρίς την οποία η εργασία αυτή δε θα είχε ολοκληρωθεί ποτέ.

Περίληψη

Η παρούσα μεταπτυχιακή διατριβή αφορά στην ανάλυση, το σχεδιασμό και την υλοποίηση ενός λογισμικού που θα μπορούσε να χρησιμοποιηθεί ως εκπαιδευτικό βοήθημα για βασικές και προχωρημένες γνώσεις χειρισμού εφαρμογών γραφείου και γενικότερα εννοιών πληροφορικής. Το λογισμικό iTestU σχεδιάστηκε με διπλό στόχο, να αποτελέσει ένα βοηθητικό εργαλείο για τον σπουδαστή που θα δοκιμάσει τις γνώσεις του πάνω σε πρακτικά και θεωρητικά θέματα πληροφορικής, αλλά και για τον καθηγητή που θα μπορέσει με τη βοήθεια της εφαρμογής να αξιολογήσει το επίπεδο γνώσεων των σπουδαστών του. Το λογισμικό iTestU παρέχει τη δυνατότητα στον σπουδαστή να παρακολουθήσει τα αποτελέσματά του στα επιμέρους αντικείμενα εξέτασης, μέσα από ένα απλό και φιλικό περιβάλλον και στο τέλος να τα εκτυπώσει με την μορφή αναφοράς. Αντίστοιχα ο καθηγητής μέσα από το περιβάλλον της εφαρμογής μπορεί να έχει μια γενικότερη εικόνα για τις επιδόσεις όλων των σπουδαστών, που έχουν εκτελέσει κάποια εξέταση, και να εκτυπώσει μια συγκριτική αναφορά. Τέλος να επισημάνουμε ότι το λογισμικό iTestU παρέχει στον καθηγητή ένα εύκολο και γρήγορο περιβάλλον για να δημιουργήσει θεωρητικές ερωτήσεις πολλαπλής επιλογής χωρίς τη συγγραφή κώδικα.

Η ανάλυση, ο σχεδιασμός και η υλοποίηση της εφαρμογής έγιναν ακολουθώντας τη μεθοδολογία Rational Unified Process - RUP. Για την μοντελοποίηση της εφαρμογής χρησιμοποιήσαμε τη γλώσσα UML και για την ανάπτυξη το περιβάλλον Visual Studio 2012 με τη γλώσσα προγραμματισμού VB.NET. Η υλοποίηση της βάσης δεδομένων έγινε σε περιβάλλον SQL Server και Access 2010.

Abstract

This master thesis concerns the analysis, design and implementation of a windows-based application, that could be used as a computer based training tool for basic and advanced office applications and information technology theory. The iTestU software was designed with a dual aim, in mind. First of all, to become an auxiliary tool for students to test their practical and theoretical information technology knowledge, as well as for the teacher to evaluate students' progress. In a simple and friendly environment, students have the opportunity to select a theoretical or practical exam, monitor their results for each exam category and finally export a report. On the other hand, teacher through the application's environment can take an overall idea of all students' performance and print a comparative report for all students that have performed an exam. Furthermore, iTestU application provides teacher with the ability to create multiple-choice questions in an easy and friendly interface, without writing a single line of code.

The analysis, design and implementation of the application was made following the Rational Unified Process - RUP methodology. The Unified Modeling Language - UML was the preferential graphical modeling language that used to model our system. The development took place on the Visual Studio 2012 integrated development environment with VB.NET programming language. The database's implementation was made in SQL Server and Access 2010 environment.

Πίνακας Περιεχομένων

1	Εισαγωγή	1
1.1	Σύντομη περιγραφή	1
1.2	Διάρθρωση της εργασίας	2
1.3	Παρουσίαση της μεθοδολογίας RUP	2
1.4	Παρουσίαση της γλώσσας μοντελοποίησης UML	4
2	Φάση Έναρξης - Σύλληψη Ιδέας	6
2.1	Η Φάση Έναρξης / Σύλληψης (Inception Phase)	6
2.2	Σύλληψη των απαιτήσεων	7
3	Φάση Εκπόνηση Μελέτης	9
3.1	Η Φάση Εκπόνηση Μελέτης (Elaboration Phase)	9
3.2	Περιπτώσεις Χρήσης - Use Cases	10
3.3	Η αρχιτεκτονική του συστήματος	22
4	Φάση Κατασκευής	24
4.1	Η Φάση της Κατασκευής (Construction Phase)	24
4.2	Ανάλυση και Σχεδιασμός	25
4.2.1	Διαγράμματα Συνεργασίας - Collaboration Diagrams	25
4.2.2	Διαγράμματα Ακολουθίας - Sequence Diagrams	29
4.2.3	Διαγράμματα Δραστηριοτήτων - Activity Diagrams	35
4.3	Υλοποίηση	38
4.3.1	Διαγράμματα Εξαρτημάτων - Component Diagram	38
4.3.2	Διαγράμματα Διανομής - Deployment Diagram	38
4.3.3	Ανάπτυξη του κώδικα	39
4.4	Έλεγχος της εφαρμογής	41
5	Η Βάση Δεδομένων	42
5.1	Εισαγωγή	42
5.1.1	Τι είναι μια Βάση Δεδομένων	42
5.1.2	Σχεσιακές Βάσεις Δεδομένων	42
5.2	Διερεύνηση των πινάκων της ΒΔ	43
5.2.1	Ο πίνακας tblMembers	43
5.2.2	Ο πίνακας tblUsers	44
5.2.3	Ο πίνακας tblSession	45
5.2.4	Ο πίνακας Execution	45
5.2.5	Ο πίνακας tblQuestionCategory	46
5.2.6	Ο πίνακας Questions	47
5.2.7	Ο πίνακας TheoryAnswers	47
5.3	Διερεύνηση των ερωτημάτων της ΒΔ	48
5.3.1	Το ερώτημα QueryResults	48
5.3.2	Το ερώτημα QuerySessionTotalResults	49
5.3.3	Το ερώτημα QueryMemberNumQuestions	50
6	Φάση Μετάβασης	52

6.1	Η Φάση Μετάβασης (Transition Phase)	52
6.2	Έλεγχος - Beta Testing	53
6.3	Εγχειρίδιο χρήσης απλού χρήστη	54
6.3.1	Εισαγωγικές οθόνες σύνδεσης και εγγραφής.....	54
6.3.2	Κεντρικό παράθυρο επιλογών χρήστη.....	56
6.3.3	Έναρξη εξέτασης.....	57
6.3.4	Αποτελέσματα εξετάσεων.....	61
6.3.5	Διαχείριση λογαριασμού.....	63
6.3.6	Βοήθεια της εφαρμογής.....	63
6.4	Εγχειρίδιο χρήσης διαχειριστή	64
6.4.1	Κεντρικό παράθυρο επιλογών διαχειριστή.....	64
6.4.2	Διαχείριση ερωτήσεων.....	65
6.4.3	Γενικά αποτελέσματα.....	67
6.4.4	Διαχείριση Χρηστών.....	68
6.4.5	Στοιχεία Χρηστών.....	70
7	Συμπεράσματα	72
8	Βιβλιογραφικές - Διαδικτυακές Πηγές	73
8.1	Βιβλιογραφία	73
8.2	Διαδίκτυο	73
	Παράρτημα	74
	Εργαλεία και τεχνολογίες	74
	Microsoft Visual Studio Ultimate 2012.....	74
	Microsoft SQL Server 2012.....	74
	IBM Rational Rose Enterprise Edition.....	75
	Microsoft Visio 2010.....	75
	Flexera InstallShield.....	76
	Microsoft Access 2010.....	76
Ο	κώδικας της εφαρμογής	77

Πίνακας Εικόνων

Εικόνα 1. Οι φάσεις της μεθοδολογίας RUP	3
Εικόνα 2. Διάγραμμα χρήσης απλού χρήστη	11
Εικόνα 3. Διάγραμμα χρήσης διαχειριστή	11
Εικόνα 4. Αρχιτεκτονική πληροφοριακού συστήματος iTestU	22
Εικόνα 5. Διάγραμμα συνεργασίας για τη σύνδεση χρήστη	25
Εικόνα 6. Διάγραμμα συνεργασίας για την αποσύνδεση χρήστη	25
Εικόνα 7. Διάγραμμα συνεργασίας για την εγγραφή χρήστη	26
Εικόνα 8. Διάγραμμα συνεργασίας για υποβολή της απάντησης	26
Εικόνα 9. Διάγραμμα συνεργασίας για αγνόηση ερώτησης	26
Εικόνα 10. Διάγραμμα συνεργασίας για αναίρεση ερώτησης	27
Εικόνα 11. Διάγραμμα συνεργασίας για την εμφάνιση αποτελεσμάτων εξέτασης	27
Εικόνα 12. Διάγραμμα συνεργασίας για διαχείριση λογαριασμού χρήστη	27
Εικόνα 13. Διάγραμμα συνεργασίας για δημιουργία χρήστη	28
Εικόνα 14. Διάγραμμα συνεργασίας για αλλαγή στοιχείων χρήστη	28
Εικόνα 15. Διάγραμμα συνεργασίας για δημιουργία νέας ερώτησης	28
Εικόνα 16. Διάγραμμα συνεργασίας για επεξεργασία ερώτησης	29
Εικόνα 17. Διάγραμμα συνεργασίας για την βοήθεια της εφαρμογής	29
Εικόνα 18. Διάγραμμα ακολουθίας εγγραφής χρήστη	29
Εικόνα 19. Διάγραμμα ακολουθίας σύνδεσης χρήστη	30
Εικόνα 20. Διάγραμμα ακολουθίας αποσύνδεση χρήστη	30
Εικόνα 21. Διάγραμμα ακολουθίας έναρξη εξέτασης και υποβολή απάντησης	31
Εικόνα 22. Διάγραμμα ακολουθίας έναρξη εξέτασης και αγνόηση ερώτησης	31
Εικόνα 23. Διάγραμμα ακολουθίας έναρξη εξέτασης και αναίρεση ερώτησης	32
Εικόνα 24. Διάγραμμα ακολουθίας διαχείριση λογαριασμού χρήστη	32
Εικόνα 25. Διάγραμμα ακολουθίας για την βοήθεια της εφαρμογής	32
Εικόνα 26. Διάγραμμα ακολουθίας για την εμφάνιση αποτελεσμάτων εξέτασης	33
Εικόνα 27. Διάγραμμα ακολουθίας για δημιουργία χρήστη	33
Εικόνα 28. Διάγραμμα ακολουθίας για αλλαγή στοιχείων χρήστη	34
Εικόνα 29. Διάγραμμα ακολουθίας για δημιουργία νέας ερώτησης	34
Εικόνα 30. Διάγραμμα ακολουθίας για επεξεργασία ερώτησης	35
Εικόνα 31. Διάγραμμα δραστηριοτήτων για εγγραφή χρήστη	35
Εικόνα 32. Διάγραμμα δραστηριοτήτων για σύνδεση χρήστη	36
Εικόνα 33. Διάγραμμα δραστηριοτήτων για έναρξη εξέτασης	36
Εικόνα 34. Διάγραμμα δραστηριοτήτων για εμφάνιση αποτελεσμάτων εξετάσεων	36
Εικόνα 35. Διάγραμμα δραστηριοτήτων διαχείρισης λογαριασμού	37
Εικόνα 36. Διάγραμμα δραστηριοτήτων διαχείρισης ερωτήσεων	37
Εικόνα 37. Διάγραμμα δραστηριοτήτων διαχείρισης χρηστών	37
Εικόνα 38. Διάγραμμα εξαρτημάτων εφαρμογής iTestU	38
Εικόνα 39. Διάγραμμα διανομής συστήματος	39
Εικόνα 40. Η ΒΔ της εφαρμογής	43
Εικόνα 41. Στιγμιότυπο του DataGridView από την εκτέλεση του ερωτήματος QueryResults	49
Εικόνα 42. Στιγμιότυπο του DataGridView από την εκτέλεση του ερωτήματος	50
Εικόνα 43. Στιγμιότυπο από την εκτέλεση του ερωτήματος	50
Εικόνα 44. Εισαγωγική οθόνη της εφαρμογής	54
Εικόνα 45. Παράθυρο προειδοποιητικού μηνύματος	54
Εικόνα 46. Παράθυρο για τη σύνδεση χρήστη	55
Εικόνα 47. Επιτυχημένη σύνδεση χρήστη	55
Εικόνα 48. Αποτυχημένη σύνδεση χρήστη	55
Εικόνα 49. Παράθυρο για την εγγραφή	56
Εικόνα 50. Επιτυχημένη εγγραφή χρήστη	56
Εικόνα 51. Προειδοποίηση για κενά πεδία	56
Εικόνα 52. Κεντρικό παράθυρο επιλογών απλού χρήστη	57
Εικόνα 53. Επιλογή αντικειμένου και κατηγορίας εξέτασης	57

Εικόνα 54. Περιβάλλον εξέτασης	58
Εικόνα 55. Εκφώνηση και χειριστήρια της ερώτησης	58
Εικόνα 56. Σωστή απάντηση	59
Εικόνα 57. Λάθος απάντηση	59
Εικόνα 58. Παράθυρο αναλυτικών αποτελεσμάτων	60
Εικόνα 59. Βοηθητικά εργαλεία και κουμπιά ενεργειών	60
Εικόνα 60. Παράθυρο συγκεντρωτικών αποτελεσμάτων	61
Εικόνα 61. Ιστορικό αποτελεσμάτων χρήση	62
Εικόνα 62. Αρχείο εξαγωγής Excel	63
Εικόνα 63. Παράθυρο διαχείριση λογαριασμού	64
Εικόνα 64. Κεντρικό παράθυρο επιλογών διαχειριστή	65
Εικόνα 65. Παράθυρο δημιουργίας ερώτησης	65
Εικόνα 66. Πεδίο αναζήτησης ερώτησης	66
Εικόνα 67. Παράθυρο επεξεργασίας ερώτησης	66
Εικόνα 68. Γενικό ιστορικό αποτελεσμάτων	67
Εικόνα 69. Παράθυρο δημιουργίας νέου χρήστη	68
Εικόνα 70. Αναζήτηση χρήστη	69
Εικόνα 71. Παράθυρο επεξεργασίας χρήστη	70
Εικόνα 72. Παραμετρικό ερώτημα Crystal Report	71
Εικόνα 73. Αναφορά με στοιχεία χρηστών	71

1 Εισαγωγή

Στην πρώτη ενότητα παρουσιάζουμε περιληπτικά το λογισμικό που αναπτύξαμε. Στις επόμενες ενότητες ακολουθούν μια σύντομη περιγραφή της μεθοδολογίας Rational Unified Process και της γλώσσας μοντελοποίησης Unified Modeling Language.

1.1 Σύντομη περιγραφή

Στην παρούσα μεταπτυχιακή διατριβή αναπτύξαμε ένα εκπαιδευτικό λογισμικό που απευθύνεται τόσο σε σπουδαστές όσο και καθηγητές της μεταδευτεροβάθμιας εκπαίδευσης. Η αρχική ιδέα ήταν να δημιουργήσουμε ένα εκπαιδευτικό λογισμικό που θα γίνει μέρος της εκπαιδευτικής διαδικασίας σε Ινστιτούτα Επαγγελματικής Κατάρτισης (IEK) αλλά και σε Κέντρα Ελευθέρων Σπουδών (ΚΕΣ). Το εκπαιδευτικό λογισμικό iTestU σχεδιάστηκε με σκοπό να αποτελέσει ένα βοηθητικό εργαλείο για τους σπουδαστές, να δοκιμάσουν τις γνώσεις τους και να εντοπίσουν τις αδυναμίες τους σε θέματα που σχετίζονται με βασικές έννοιες της πληροφορικής πριν δώσουν τις εξετάσεις πιστοποίησης του ΕΟΠΠΕΠ (Εθνικός Οργανισμός Πιστοποίησης Προσόντων και Επαγγελματικού Προσανατολισμού).

Ο σπουδαστής, μέσα από το κεντρικό παράθυρο της εφαρμογής, μπορεί να επιλέξει από ένα πλήθος λειτουργιών όπως να ξεκινήσει μια θεωρητική ή πρακτική εξέταση, να παρακολουθήσει το ιστορικό των αποτελεσμάτων του, να διαχειριστεί τον προσωπικό του λογαριασμό ή να πάρει ηλεκτρονική βοήθεια για τις λειτουργίες της εφαρμογής iTestU. Αναμφίβολα η σημαντικότερη λειτουργία που μπορεί να επιλέξει ένας σπουδαστής είναι η λειτουργία της εξέτασης.

Η διαδικασία της εξέτασης περιλαμβάνει δύο μεγάλα αντικείμενα εξέτασης, το θεωρητικό και το πρακτικό μέρος. Το θεωρητικό περιλαμβάνει ερωτήσεις πολλαπλής επιλογής, όπου ο σπουδαστής καλείται να απαντήσει σε θέματα από τη θεωρία της πληροφορικής. Το πρακτικό μέρος αποτελείται από τρεις κατηγορίες εξετάσεων και του δίνει τη δυνατότητα να εκτελέσει ασκήσεις στο πραγματικό περιβάλλον κάποιου λογισμικού. Οι τρεις κατηγορίες εξετάσεων είναι επεξεργασία κειμένου με το λογισμικό Microsoft Word, υπολογιστικά φύλλα με το λογισμικό Microsoft Excel και παρουσιάσεις με το λογισμικό Microsoft PowerPoint.

Το λογισμικό iTestU έχει τη δυνατότητα να υποβάλει ερωτήσεις στον εξεταζόμενο σπουδαστή και να αξιολογήσει τις απαντήσεις του τόσο στο θεωρητικό όσο και στο πρακτικό μέρος της εξέτασης. Ο σπουδαστής μετά το πέρας της εξέτασης μπορεί να δει αναλυτικά αποτελέσματα για κάθε ερώτηση που απάντησε σωστά, λανθασμένα ή αγνόησε. Τα αποτελέσματα αυτά μπορεί να τα εξάγει και να τα αποθηκεύσει στον υπολογιστή του ή να τα εκτυπώσει με τη μορφή μιας αναφοράς. Μέσω του ιστορικού αποτελεσμάτων ένας σπουδαστής μπορεί να αναζητήσει αποτελέσματα παλαιότερων εξετάσεων, εφαρμόζοντας κατάλληλα κριτήρια και να εξαγάγει ή να εκτυπώσει συγκριτικές αναφορές. Μέσα από τις αναφορές αυτές μπορεί να έχει μια συνολική εικόνα για την επίδοσή του.

Κάθε σπουδαστής που χρησιμοποιεί την εφαρμογή iTestU έχει το δικό του προσωπικό λογαριασμό. Με αυτό τον τρόπο διασφαλίζεται η αρχή της προστασίας των προσωπικών δεδομένων του. Ο κάθε σπουδαστής έχει πρόσβαση μόνο στα δικά του προσωπικά δεδομένα. Η λειτουργία της διαχείρισης του προσωπικού λογαριασμού δίνει τη δυνατότητα στο σπουδαστή να επεξεργαστεί και να τροποποιήσει τα προσωπικά του στοιχεία όποτε το επιθυμεί χωρίς να χρειάζεται να απευθυνθεί στο διαχειριστή της εφαρμογής.

Το λογισμικό iTestU σχεδιάστηκε και αναπτύχθηκε με σκοπό να αποτελέσει ένα βοηθητικό εργαλείο και για τον καθηγητή. Μέσα από το περιβάλλον του λογισμικού του δίνεται η δυνατότητα να δημιουργήσει νέες ερωτήσεις πολλαπλής επιλογής. Η διαδικασία δημιουργίας ερωτήσεων είναι μια απλή διαδικασία δύο βημάτων που μπορεί να γίνει γρήγορα χωρίς να απαιτούνται εξειδικευμένες γνώσεις. Τις ερωτήσεις αυτές μπορεί να τις χρησιμοποιήσει κατά τη

διάρκεια του μαθήματος ως μέρος της εκπαιδευτικής διαδικασίας ή ως εξέταση για να αξιολογήσει το επίπεδο των σπουδαστών του.

Ο καθηγητής έχει πρόσβαση σε λειτουργίες διαχείρισης ιστορικού αποτελεσμάτων εξετάσεων. Σε αντίθεση με τον σπουδαστή που μπορεί να παρακολουθήσει μόνο το δικό του ιστορικό αποτελεσμάτων, ο καθηγητής έχει πρόσβαση στο ιστορικό όλων των σπουδαστών του. Αυτό θα μπορούσε να αποτελέσει ένα σημαντικό εργαλείο για να βγάλει σημαντικά συμπεράσματα για τις επιδόσεις των σπουδαστών του. Μέσα από το ιστορικό μπορεί να φιλτράρει τα αποτελέσματα των εξετάσεων εφαρμόζοντας κριτήρια και τελικά να εξαγάγει την πληροφορία που τον ενδιαφέρει. Τα αρχεία εξαγωγής μπορεί να τα αποθηκεύσει στον υπολογιστή του σε διάφορες μορφές ή να τα εκτυπώσει με τη μορφή αναφοράς. Μια ακόμα σημαντική λειτουργία για τον καθηγητή είναι ότι μπορεί να δημιουργήσει ο ίδιος τους λογαριασμούς των σπουδαστών του και να τους επεξεργαστεί.

Για την ανάπτυξη του λογισμικού χρησιμοποιήσαμε την αντικειμενοστραφή προσέγγιση σε όλες τις φάσεις της ανάπτυξης. Για την ανάλυση και το σχεδιασμό του λογισμικού χρησιμοποιήσαμε την γλώσσα μοντελοποίησης UML και για την υλοποίηση τη γλώσσα προγραμματισμού VB.NET. Η ανάπτυξη και ο αρχικός έλεγχος έγινε στο ολοκληρωμένο περιβάλλον Visual Studio 2012 χρησιμοποιώντας το .NET framework 4.5. Ο τελικός έλεγχος της εφαρμογής έγινε χρησιμοποιώντας πραγματικά σενάρια χρήσης τόσο από τον υποφαινόμενο όσο και από πραγματικούς σπουδαστές που δοκίμασαν την εφαρμογή. Για τη δημιουργία του τελικού πακέτου διανομής της εφαρμογής iTestU χρησιμοποιήσαμε το λογισμικό Install Shield.

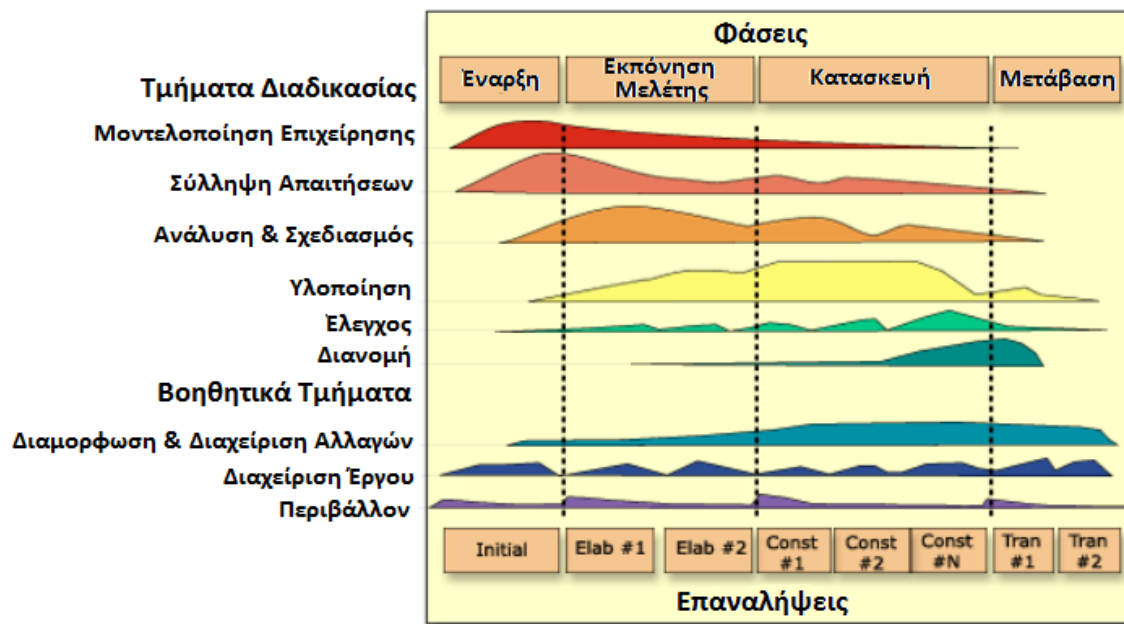
1.2 Διάρθρωση της εργασίας

Η διπλωματική διατριβή αποτελείται από έξι μέρη - κεφάλαια. Το πρώτο κεφάλαιο ξεκινά με μια σύντομη περιγραφή του προτύπου Rational Unified Process - RUP που αποτέλεσε τη βασική μεθοδολογία που ακολουθήσαμε σε όλα τα στάδια του κύκλου ζωής της εφαρμογής από την έναρξη μέχρι την ολοκλήρωση. Στο ίδιο κεφάλαιο υπάρχει και μια σύντομη αναφορά στη γλώσσα μοντελοποίησης Unified Modeling Language - UML, που χρησιμοποιήθηκε στα διάφορα στάδια του κύκλου ζωής της εφαρμογής για να περιγράψουμε λειτουργίες της εφαρμογής. Στο δεύτερο κεφάλαιο γίνεται μια περιγραφή της φάσης της έναρξης, όπου παρουσιάζουμε τους στόχους που θέσαμε για την εφαρμογή, μέσω της σύλληψης απαιτήσεων. Στο τρίτο κεφάλαιο, που αποτελεί τη φάση εκπόνησης μελέτης, γίνεται μια παρουσίαση των περιπτώσεων χρήσης της εφαρμογής iTestU, με τη χρήση της γλώσσας UML και παρουσιάζεται το μοντέλο αρχιτεκτονικής στο οποίο καταλήξαμε. Στο τέταρτο κεφάλαιο, τη φάση της κατασκευής, στο τμήμα ανάλυσης και σχεδιασμού παρουσιάζουμε τα διαγράμματα συνεργασίας, ακολουθίας και δραστηριοτήτων της UML, ενώ στο τμήμα της διαδικασίας υλοποίησης περιγράφουμε τα διαγράμματα εξαρτημάτων και διανομής που καταλήξαμε κατά την υλοποίηση της εφαρμογής. Στο πέμπτο κεφάλαιο που επίσης αποτελεί τμήμα της διαδικασίας υλοποίησης παρουσιάζεται η βάση δεδομένων της εφαρμογής κάνοντας μια αναλυτική αναφορά σε πίνακες και ερωτήματα που την απαρτίζουν. Στο έκτο κεφάλαιο γίνεται μια αναφορά στη φάση της μετάβασης που ακολουθεί η μεθοδολογία RUP και παρουσιάζεται το τελικό προϊόν μέσα από τα εγχειρίδια χρήσης για τον απλό χρήστη και τον διαχειριστή της εφαρμογής. Η διατριβή ολοκληρώνεται με τα συμπεράσματα και τη βιβλιογραφία. Στο παράρτημα γίνεται αναφορά στα εργαλεία που χρησιμοποιήθηκαν στα διάφορα στάδια του έργου και παρουσιάζεται ο κώδικας όπως αναπτύχθηκε κατά το στάδιο της υλοποίησης.

1.3 Παρουσίαση της μεθοδολογίας RUP

Για την περιγραφή του κύκλου ζωής του πληροφοριακού συστήματος iTestU επιλέξαμε το αντικειμενοστραφές μοντέλο Rational Unified Process (RUP) που βασίζεται εν μέρει στο μοντέλο του καταρράκτη, ωστόσο θεωρεί ότι η ανάλυση απαιτήσεων, ο σχεδιασμός, η υλοποίηση και ο έλεγχος δε συμπίπτουν με χρονικές φάσεις, αλλά αντιπροσωπεύουν τμήματα διαδικασίας που

λαμβάνουν χώρα σε διάφορες χρονικές φάσεις. Η μέθοδος RUP δηλαδή εισάγει και δεύτερη διάσταση που αφορά στο χρόνο. Ένα ακόμα χαρακτηριστικό της είναι η επαναληπτική της φύση. Ο κύκλος ζωής λογισμικού είναι επαναληπτικός και η ανάπτυξη προχωράει μέσα από μια σειρά επαναλήψεων μέχρι να πάρουμε το τελικό προϊόν.



Εικόνα 1. Οι φάσεις της μεθοδολογίας RUP

Γενικά, η μέθοδος RUP είναι δομημένη σε δύο διαστάσεις: χρόνο και τμήματα διαδικασίας όπως παρουσιάζεται στην Εικόνα 1. Ο κύκλος της ζωής του λογισμικού σύμφωνα με τη RUP διαιρείται, ως προς το χρόνο, σε τέσσερις διαδοχικές φάσεις, καθεμία από τις οποίες ολοκληρώνεται σε ένα σημείο - ορόσημο (milestone). Ειδικότερα, η δόμηση ενός έργου σε σχέση με το χρόνο ακολουθεί τις παρακάτω τέσσερις φάσεις:

- *Έναρξη / Σύλληψη (Inception phase):*
Στην πρώτη φάση γίνεται ο καθορισμός της προοπτικής του έργου και η ανάλυση εφικτότητας
- *Εκπόνηση Μελέτης / Επεξεργασία (Elaboration phase):*
Περιλαμβάνει σχεδιασμό των απαιτούμενων δραστηριοτήτων και πόρων, καθορισμό των χαρακτηριστικών και σχεδιασμό της αρχιτεκτονικής του συστήματος
- *Κατασκευή (Construction phase):*
Το στάδιο αυτό περιλαμβάνει την ανάπτυξη του προϊόντος σε μια σειρά βημάτων που επαναλαμβάνονται
- *Μετάβαση (Transition phase):*
Στο τελευταίο στάδιο γίνεται η διάθεση του προϊόντος στους τελικούς χρήστες

Κάθε φάση είναι, ουσιαστικά, μια περίοδος του χρόνου μεταξύ δύο ορόσημων. Στο τέλος κάθε φάσης πραγματοποιείται αξιολόγηση, η οποία καθορίζει εάν οι στόχοι της εκάστοτε φάσης έχουν επιτευχθεί. Ένα ικανοποιητικό πόρισμα αξιολόγησης δίνει την άδεια να προχωρήσει το έργο στην επόμενη φάση. Ο κύκλος ζωής του λογισμικού, σύμφωνα με τη διάσταση των τμημάτων διαδικασίας περιλαμβάνει τα εξής βασικά τμήματα:

1. Σύλληψη απαιτήσεων (Business Modeling & Requirements)

Περιλαμβάνει μια ανάλυση του τι πρέπει να κάνει το σύστημα

2. Ανάλυση και σχεδιασμός (*Analysis & Design*)

Περιλαμβάνει μια περιγραφή του πώς θα υλοποιηθεί το σύστημα

3. Υλοποίηση (*Implementation*)

Παραγωγή του κώδικα

4. Έλεγχος (*Testing*)

Επαλήθευση του συστήματος

Ένα πέρασμα μέσω των τεσσάρων φάσεων είναι ένας *κύκλος ανάπτυξης* (*development cycle*). Κάθε πέρασμα μέσω των τεσσάρων φάσεων αποτελεί μια *γενιά* (*generation*) του λογισμικού. Αν το προϊόν δεν "πεθάνει", θα εξελιχθεί στην επόμενη γενιά του μέσα από την ίδια ακολουθία φάσεων Έναρξης, Επεξεργασίας, Κατασκευής και Μετάβασης, αλλά αυτή τη φορά δίνοντας διαφορετική έμφαση στις διάφορες φάσεις. Αυτοί οι επόμενοι κύκλοι καλούνται *κύκλοι εξέλιξης* (*evolution cycles*). Καθώς το προϊόν περνά από διάφορους κύκλους, δημιουργούνται οι νέες γενιές.

Η αφορμή για την έναρξη ενός νέου κύκλου εξέλιξης μπορεί να δοθεί από προτάσεις των χρηστών για βελτιώσεις, αλλαγές στην τεχνολογία, ανταπόκριση στον ανταγωνισμό και άλλα. Οι κύκλοι εξέλιξης, συνήθως, έχουν πολύ πιο σύντομες φάσεις έναρξης και επεξεργασίας, δεδομένου ότι ο βασικός ορισμός και η αρχιτεκτονική του προϊόντος καθορίζονται από τους προγενέστερους κύκλους ανάπτυξης. Εξαιρέσεις σε αυτόν τον κανόνα αποτελούν οι κύκλοι εξέλιξης, στους οποίους προκύπτει ένα σημαντικό προϊόν ή ένας αρχιτεκτονικός επαναπροσδιορισμός λαμβάνει χώρα.

1.4 Παρουσίαση της γλώσσας μοντελοποίησης UML

Κύριο χαρακτηριστικό της μεθοδολογίας RUP είναι ότι καθοδηγείται από περιπτώσεις χρήσης (*use cases*) και χρησιμοποιεί την ενοποιημένη γλώσσα μοντελοποίησης-σχεδιασμού (*Unified Modeling Language - UML*) για την ανάπτυξη λογισμικού. Η UML είναι μία γραφική γλώσσα για την οπτική αναπαράσταση, τη διαμόρφωση προδιαγραφών και την τεκμηρίωση συστημάτων. Είναι μία πλήρης, πλούσια και πρότυπη γλώσσα μοντελοποίησης στη μηχανική λογισμικού και μπορεί να χρησιμοποιηθεί σε όλη τη διάρκεια του έργου, από την ανάλυση των απαιτήσεων ως τον έλεγχο το παραδοτέου λογισμικού/πληροφοριακού συστήματος. Αποτελείται από ένα σύνολο προσυμφωνημένων όρων, συμβόλων και διαγραμμάτων που επιτρέπουν την εμφάνιση των ορίων ενός συστήματος και των βασικών λειτουργιών του, χρησιμοποιώντας περιπτώσεις χρήσης (*use cases*), τη χρονική αλληλεπίδραση μεταξύ των αντικειμένων χρησιμοποιώντας «διαγράμματα αλληλεπίδρασης», την αναπαράσταση μιας στατικής δομής ενός συστήματος χρησιμοποιώντας «διαγράμματα κλάσεων», τη μοντελοποίηση της συμπεριφοράς των αντικειμένων με «διαγράμματα καταστάσεων», την περιγραφή της υλοποίησης της αρχιτεκτονικής με «διαγράμματα συστατικών».

Η UML περιλαμβάνει τρία βασικά στοιχεία:

- Οντότητες
- Σχέσεις
- Διαγράμματα

Οντότητα (*entity*) είναι ένα αντικείμενο ενδιαφέροντος του πραγματικού κόσμου. Μπορεί να είναι άνθρωποι, μέρη, αντικείμενα, γεγονότα, έννοιες κλπ. *Στιγμιότυπο* (*instance*) μιας οντότητας είναι μια συγκεκριμένη περίπτωση ενός τύπου οντότητας.

Σχέση (*relationship*) είναι η σύνδεση δύο ή περισσότερων τύπων οντοτήτων που παρουσιάζει ενδιαφέρον για το σχεδιασμό. Με συσχετίσεις μπορούν να συνδέονται και χαρακτηριστικά οντοτήτων.

Διάγραμμα (*diagram*) είναι μια απλοποιημένη και δομημένη οπτική παρουσίαση εννοιών, ιδεών, κατασκευών, σχέσεων, στατιστικών δεδομένων, ανατομίας κλπ. Χρησιμοποιείται σε όλες τις

ανθρώπινες δραστηριότητες για να παρουσιάσει, απλοποιήσει και γενικά να κάνει κατανοητό το θέμα με το οποίο σχετίζεται.

Για να αναπαράσχει τις διαφορετικές απόψεις μοντελοποίησης η γλώσσα UML ορίζει 9 είδη διαγραμμάτων:

- 1) Διαγράμματα τάξεων (Class diagrams): Αναπαριστούν τη στατική δομή όσον αφορά στις τάξεις και τις σχέσεις τους.
- 2) Διαγράμματα αντικειμένων (Object diagrams): Αναπαριστούν αντικείμενα και τις σχέσεις τους και αντιστοιχούν σε απλοποιημένα διαγράμματα συνεργασίας που δεν αναπαριστούν μετάδοση μηνυμάτων.
- 3) Διαγράμματα συνεργασίας (Collaboration diagrams): Η αναπαράσταση των αντικειμένων, συνδέσεων και αλληλεπιδράσεων.
- 4) Διαγράμματα σειράς/ακολουθίας (Sequence diagrams): Χρονική αναπαράσταση των αντικειμένων και των αλληλεπιδράσεών τους.
- 5) Διαγράμματα καταστάσεων (State chart diagrams): Αναπαριστούν τη συμπεριφορά της τάξης, όσον αφορά στις καταστάσεις της.
- 6) Διαγράμματα δραστηριοτήτων (Activity diagrams): Αναπαριστούν τη συμπεριφορά μιας λειτουργίας ως σύνολο ενεργειών.
- 7) Διαγράμματα εξαρτημάτων (Component diagrams): Αναπαριστούν τα φυσικά εξαρτήματα μιας εφαρμογής.
- 8) Διαγράμματα διανομής (Deployment diagrams): Αναπαριστούν τη διανομή των εξαρτημάτων σε συγκεκριμένα τεμάχια του υλικού.
- 9) Διαγράμματα περιπτώσεων χρήσης (Use case diagrams): Αναπαριστούν τις λειτουργίες ενός συστήματος από την οπτική γωνία του χρήστη.

Στα επόμενα κεφάλαια παρουσιάζονται τα σημαντικότερα από τα διαγράμματα αυτά που σχεδιάσαμε για να μοντελοποιήσουμε την εφαρμογή iTestU. Για την δημιουργία των διαγραμμάτων της UML χρησιμοποιήθηκε το CASE εργαλείο της IBM, Rational Rose Enterprise Edition version 7.0

2 Φάση Έναρξης - Σύλληψη Ιδέας

Στην πρώτη φάση γίνεται η σύλληψη της ιδέας, ο καθορισμός της προοπτικής του έργου και η ανάλυση εφικτότητας. Στην επόμενη ενότητα γίνονται περισσότερο κατανοητές οι έννοιες αυτές μέσα από την αναλυτική περιγραφή αυτής της φάσης. Στην ενότητα 2.2 παρουσιάζουμε με την μορφή προτάσεων μερικά από τα σενάρια λειτουργίας του συστήματος όπως τα οραματιστήκαμε πριν τα υλοποιήσουμε. Να σημειώσουμε εδώ ότι αν και κάποια από τα σενάρια αυτά τροποποιήθηκαν ή εμπλουτίστηκαν σε επόμενα στάδια, ωστόσο αποτέλεσαν το βασικό πλαίσιο για να κινηθούμε κατά την υλοποίηση της εφαρμογής.

2.1 Η Φάση Έναρξης / Σύλληψης (Inception Phase)

Ο στόχος σε αυτήν τη φάση είναι να επιτευχθεί συμφωνία μεταξύ όλων των ενδιαφερόμενων (stakeholders) σχετικά με τους στόχους του έργου. Η φάση Έναρξης είναι σημαντική κυρίως για τις νέες προσπάθειες ανάπτυξης, στις οποίες υπάρχουν σημαντικοί κίνδυνοι που πρέπει να ξεταστούν, προτού μπορέσει να προχωρήσει το έργο. Για τα έργα που στοχεύουν σε βελτιώσεις ενός υπάρχοντος συστήματος, η φάση Έναρξης είναι πιο σύντομη, αλλά εξακολουθεί να επικεντρώνεται στο αν το έργο αξίζει και είναι εφικτό να γίνει.

Οι κύριοι στόχοι της φάσης Έναρξης περιλαμβάνουν:

- Τον προσδιορισμό των στόχων και των ορίων του έργου, μαζί με ένα λειτουργικό όραμα, τα κριτήρια αποδοχής καθώς και με το τι προορίζεται να είναι στο προϊόν και τι όχι.
- Τη διάκριση των κρίσιμων περιπτώσεων χρήσης (use cases) του συστήματος, δηλαδή των κύριων σεναρίων λειτουργίας που θα κατευθύνουν τις σημαντικότερες αποφάσεις σχεδιασμού.
- Την έκθεση και την επίδειξη τουλάχιστον μιας υποψήφιας αρχιτεκτονικής βάσει μερικών από τα κύρια σενάρια.
- Την εκτίμηση του συνολικού κόστους και του χρονοδιαγράμματος για την ολοκλήρωση του έργου (και πιο λεπτομερείς εκτιμήσεις για τη φάση Εκπόνησης Μελέτης που θα ακολουθήσει αμέσως μετά)
- Την εκτίμηση των ενδεχόμενων κινδύνων.
- Την προετοιμασία του περιβάλλοντος για την υποστήριξη του έργου.

Οι σημαντικές δραστηριότητες στη φάση Έναρξης περιλαμβάνουν:

- Τη διαμόρφωση του πεδίου του έργου. Αυτό περιλαμβάνει τη συλλογή των σημαντικότερων απαιτήσεων και περιορισμών σε τέτοιο βαθμό, ώστε να μπορούν να προκύψουν τα κριτήρια αποδοχής του τελικού προϊόντος.
- Τον προγραμματισμό και την προετοιμασία μιας επιχειρηματικής περίπτωσης (business case). Αξιολόγηση των εναλλακτικών λύσεων για τη διαχείριση κινδύνου, τη στελέχωση, το πλάνο του έργου και την εξισορρόπηση κόστους / χρονοδιαγράμματος / αποδοτικότητας.
- Τη σύνθεση μιας υποψήφιας αρχιτεκτονικής, αξιολογώντας τους συμβιβασμούς στο σχεδιασμό και την κατασκευή, έτσι ώστε το κόστος, το πρόγραμμα και οι πόροι να μπορούν να υπολογιστούν. Ο στόχος, εδώ, είναι να καταδειχθεί αν το έργο μπορεί ή όχι να υλοποιηθεί. Η εργασία αυτή μπορεί να λάβει τη μορφή ενός μοντέλου που προσομοιώνει τι απαιτείται, ή ενός αρχικού πρωτότυπου που εξερευνά τις περιοχές υψηλού κινδύνου. Η προσπάθεια κατασκευής πρωτοτύπων κατά τη διάρκεια της Έναρξης πρέπει να περιοριστεί στην πεποίθηση ότι μια λύση είναι δυνατή – η λύση πραγματοποιείται κατά τη διάρκεια της Επεξεργασίας και της Κατασκευής.

- Την προετοιμασία του περιβάλλοντος για το έργο, αξιολογώντας το έργο και την οργάνωση, επιλέγοντας τα εργαλεία και αποφασίζοντας ποια μέρη της διαδικασίας πρέπει να βελτιωθούν.

Στο τέλος της φάσης Έναρξης έχουμε το πρώτο ορόσημο των στόχων του κύκλου ζωής (Life cycle Objectives Milestone). Σε αυτό το σημείο, εξετάζονται οι στόχοι του κύκλου ζωής του έργου και αποφασίζεται αν θα συνεχιστεί ή αν θα ακυρωθεί το έργο. Λαμβάνεται υπόψη η συμφωνία των ενδιαφερόμενων ως προς τους στόχους, τις εκτιμήσεις κόστους / χρονοδιαγράμματος, τις απαιτήσεις, τις εκτιμήσεις δαπανών / χρονοδιαγράμματος, τις προτεραιότητες, τους κινδύνους. Λαμβάνεται ακόμα υπόψη αν η διαδικασία ανάπτυξης είναι η κατάλληλη, αν οι κίνδυνοι έχουν αναγνωριστεί και αν υπάρχει μια στρατηγική μετριασμού για τον καθένα. Εάν το έργο αποτύχει να φθάσει σε αυτό το ορόσημο μπορεί να ανακληθεί ή να αναθεωρηθεί σημαντικά.

2.2 Σύλληψη των απαιτήσεων

Το στάδιο αυτό είναι πολύ σημαντικό κυρίως για τις νέες προσπάθειες ανάπτυξης ενός πληροφοριακού συστήματος, όπως άλλωστε ήταν και το δικό μας. Σε αυτό το στάδιο έπρεπε να καθορίσουμε τους στόχους σχετικά με το τι θα θέλαμε να κάνει το λογισμικό, χωρίς να μας απασχολήσει ο τρόπος με τον οποίο θα το κάνει. Απλά καταγράψαμε με τη μορφή προτάσεων τις ιδέες που είχαμε γι' αυτό το λογισμικό. Στη γέννηση ιδεών συνέβαλλαν παρόμοια λογισμικά που υπάρχουν ήδη στην αγορά για χρόνια και σημειώνουν μεγάλη επιτυχία. Αυτό που κάναμε ήταν να εστιάσουμε στα πλεονεκτήματα τους και στις αδυναμίες τους, να προσθέσουμε και τις δικές μας ιδέες και τελικά να καταλήξουμε στις παρακάτω προτάσεις. Οι προτάσεις αυτές αποτέλεσαν την τροφοδότηση για τα επόμενα στάδια. Ουσιαστικά αποτέλεσαν το πλαίσιο μέσα στο οποίο κινηθήκαμε για να σχεδιάσουμε και να υλοποιήσουμε το εκπαιδευτικό πληροφοριακό σύστημα iTestU.

- Θέλουμε ένα λογισμικό όπου ο χρήστης-σπουδαστής θα μπορεί να συνδεθεί σε ένα προσωπικό του λογαριασμό.
- Ο χρήστης να μπορεί μέσα από το περιβάλλον της εφαρμογής να δημιουργήσει τον προσωπικό του λογαριασμό.
- Ο χρήστης να έχει τη δυνατότητα αφού συνδεθεί στο λογαριασμό του να αλλάξει να επεξεργαστεί και να συμπληρώσει στοιχεία από τον προσωπικό του λογαριασμό.
- Κάθε χρήστης να έχει πρόσβαση μόνο σε πληροφορία που σχετίζεται με τον ίδιο. Για παράδειγμα ένας χρήστης να μπορεί να έχει πρόσβαση μόνο στα δικά του προσωπικά δεδομένα.
- Ο χρήστης μέσα στην εφαρμογή θα έχει την δυνατότητα να απαντήσει σε θεωρητικές ερωτήσεις με τη μορφή ερωτήσεων πολλαπλής επιλογής.
- Ο χρήστης μέσα από το περιβάλλον της εφαρμογής θα έχει την δυνατότητα να εκτελέσει πρακτικές ασκήσεις σε πραγματικά περιβάλλοντα άλλων εφαρμογών όπως για παράδειγμα εφαρμογές γραφείου (πχ Microsoft Word, Excel, PowerPoint), σχεδιαστικά προγράμματα (πχ Autocad) και επεξεργασίας εικόνων (πχ Gimp, Photoshop) ή ακόμα και να γράψει κώδικα για κάποια γλώσσα προγραμματισμού. Δεν θέλαμε σε καμία περίπτωση το περιβάλλον της εξέτασης να είναι περιβάλλον εξομοίωσης αλλά το πραγματικό περιβάλλον του λογισμικού που εξετάζεται.
- Το λογισμικό σε κάθε περίπτωση είτε πρόκειται για θεωρητικές ερωτήσεις ή πρακτικές ασκήσεις να έχει τη δυνατότητα να αξιολογήσει τις απαντήσεις και να αποφανθεί αν είναι σωστές ή λάθος.
- Το λογισμικό να έχει την δυνατότητα να εμφανίζει ιστορικό των εξετάσεων του μαθητή στο οποίο μπορεί να ανατρέξει και να συγκρίνει τα αποτελέσματα μεταξύ τους.

- Το λογισμικό να έχει τη δυνατότητα να εξαγάγει τα αποτελέσματα των εξετάσεων εφόσον το θελήσει ο χρήστης σε διάφορες μορφές.
- Να υπάρχουν χρήστες με διαφορετικά δικαιώματα χρήσης της εφαρμογής. Για παράδειγμα απλοί χρήστες με περιορισμένα δικαιώματα και χρήστες - διαχειριστές με τη δυνατότητα να εκτελέσουν προχωρημένες λειτουργίες.
- Να υπάρχει η δυνατότητα μέσα από το περιβάλλον της εφαρμογής να προστεθούν νέες ερωτήσεις και να εμπλουτιστούν-τροποποιηθούν οι παλιότερες. Αυτό θα ήταν επιθυμητό τόσο για τις θεωρητικές όσο και για τις πρακτικές ασκήσεις. Εναλλακτικά θα μπορούσε να υλοποιηθεί ένα βοηθητικό εργαλείο γι' αυτή τη λειτουργία.
- Ο χρήστης-διαχειριστής να έχει τη δυνατότητα να επεξεργαστεί τους λογαριασμούς των απλών χρηστών μέσα από το περιβάλλον της εφαρμογής. Δηλαδή να τροποποιήσει στοιχεία, να διαγράψει ένα λογαριασμό ή να δημιουργήσει ένα νέο λογαριασμό χρήστη.
- Θα θέλαμε η εφαρμογή να τρέχει διαδικτυακά μέσω ενός προγράμματος περιήγησης ή να είναι μια αυτόνομη εφαρμογή στο λειτουργικό σύστημα windows.

3 Φάση Εκπόνηση Μελέτης

Η δεύτερη φάση περιλαμβάνει σχεδιασμό των απαιτούμενων δραστηριοτήτων και πόρων, καθορισμό των χαρακτηριστικών και σχεδιασμό της αρχιτεκτονικής του συστήματος. Στην ενότητα 3.1 αναλύουμε λεπτομερέστερα τις ενέργειες που πρέπει να γίνουν κατά τη διάρκεια ενός έργου, στην ενότητα 3.2 περιγράφουμε και αναλύουμε τα σενάρια χρήσης για το δικό μας έργο και τέλος στην ενότητα 3.3 παρουσιάζουμε την αρχιτεκτονική στην οποία καταλήξαμε για να υλοποιήσουμε το πληροφοριακό σύστημα.

3.1 Η Φάση Εκπόνηση Μελέτης (Elaboration Phase)

Στόχος της φάσης Εκπόνηση Μελέτης είναι να θεμελιώσει την αρχιτεκτονική του συστήματος για να παρέχει μια σταθερή βάση στην προσπάθεια σχεδιασμού και υλοποίησης της φάσης Κατασκευής. Η αρχιτεκτονική εξελίσσεται λαμβάνοντας υπόψη τις σημαντικότερες απαιτήσεις (εκείνες που ασκούν μεγάλη επιρροή στην αρχιτεκτονική του συστήματος) και τις εκτιμήσεις του κινδύνου. Η σταθερότητα της αρχιτεκτονικής αξιολογείται μέσω ενός ή περισσότερων αρχιτεκτονικών πρωτοτύπων.

Οι κύριοι στόχοι της φάσης Εκπόνηση Μελέτης είναι:

- Να εξασφαλιστεί ότι η αρχιτεκτονική, οι απαιτήσεις και τα σχέδια είναι αρκετά σταθερά και οι κίνδυνοι αρκετά μετριασμένοι προκειμένου να προβλεφθεί το κόστος και το χρονοδιάγραμμα για την ολοκλήρωση της ανάπτυξης.
- Να αντιμετωπιστούν όλοι οι αρχιτεκτονικά σημαντικοί κίνδυνοι του έργου.
- Να οριστεί μια βασική αρχιτεκτονική που θα προκύπτει από την εξέταση των σημαντικών, από αρχιτεκτονικής άποψης, σεναρίων, τα οποία, συνήθως, εκθέτουν τους κορυφαίους τεχνικούς κινδύνους του έργου.
- Να δημιουργηθεί ένα εξελικτικό πρωτότυπο (evolutionary prototype), καθώς και ένα ή περισσότερα διερευνητικά (exploratory), ενδεχομένως, πρόχειρα πρωτότυπα για να μετριαστούν συγκεκριμένοι κίνδυνοι, όπως η αντιστάθμιση σχεδίου / απαιτήσεων, η επαναχρησιμοποίηση τμημάτων και η δυνατότητα πραγματοποίησης ή μη του έργου.
- Να αποδειχτεί ότι η βασική αρχιτεκτονική θα υποστηρίξει τις απαιτήσεις του συστήματος με ένα λογικό κόστος και σε ένα λογικό χρόνο.
- Να οριστεί ένα περιβάλλον υποστήριξης.

Για την επίτευξη αυτών των κύριων στόχων είναι εξίσου σημαντική η δημιουργία του περιβάλλοντος για την υποστήριξη του έργου, δηλαδή η δημιουργία μιας περίπτωσης ανάπτυξης (development case), η προετοιμασία προτύπων, οδηγιών και η κατασκευή εργαλείων.

Οι σημαντικές δραστηριότητες της φάσης Εκπόνηση Μελέτης περιλαμβάνουν:

- Τον προσδιορισμό, την επικύρωση και τη θεμελίωση της αρχιτεκτονικής τόσο θεωρητικά όσο και πρακτικά.
- Τη βελτίωση του Οράματος (Vision) για το έργο, βάσει των νέων πληροφοριών που συλλέγονται κατά τη διάρκεια της φάσης, ενισχύοντας την κατανόηση των κρισιμότερων περιπτώσεων χρήσης που κατευθύνουν τις αποφάσεις αρχιτεκτονικής και προγραμματισμού.
- Τη δημιουργία και τη θεμελίωση λεπτομερών σχεδίων επανάληψης για τη επόμενη φάση, τη φάση της Κατασκευής.
- Την εκλέπτυνση της περίπτωσης της ανάπτυξης (development case) και τοποθέτηση του περιβάλλοντος της ανάπτυξης, συμπεριλαμβανομένης της διαδικασίας, των εργαλείων και της αυτοματοποίησης που απαιτείται για την υποστήριξη της ομάδας κατασκευής.
- Τη βελτίωση της αρχιτεκτονικής και την επιλογή των συστατικών μερών. Τα πιθανά συστατικά αξιολογούνται και οι αποφάσεις για την κατασκευή, την αγορά και την

επαναχρησιμοποίησή τους κατανοούνται, έτσι ώστε να καθοριστεί το κόστος και το χρονοδιάγραμμα της φάσης Κατασκευής με εμπιστοσύνη. Τα επιλεγμένα αρχιτεκτονικά τμήματα ολοκληρώνονται και αξιολογούνται σύμφωνα με τα κύρια σενάρια. Τα συμπεράσματα από αυτές τις δραστηριότητες μπορούν να οδηγήσουν σε έναν επανασχεδιασμό της αρχιτεκτονικής, λαμβάνοντας υπόψη εναλλακτικά σχέδια ή την επανεξέταση των απαιτήσεων.

Στο τέλος της φάσης Εκπόνηση Μελέτης έχουμε το δεύτερο ορόσημο, το ορόσημο της Αρχιτεκτονικής του Κύκλου Ζωής του έργου. Σε αυτό το σημείο, εξετάζονται οι λεπτομερείς στόχοι του συστήματος, η επιλογή της αρχιτεκτονικής και η λύση των σημαντικότερων κινδύνων. Εξετάζεται εάν το προϊόν-όραμα (γιατί ακόμα δεν υπάρχει προϊόν) και οι απαιτήσεις είναι σταθερά, εάν η αρχιτεκτονική είναι σταθερή, εάν οι βασικές προσεγγίσεις που χρησιμοποιούνται στον έλεγχο και την αξιολόγηση είναι αποδεδειγμένες, εάν ο έλεγχος και η αξιολόγηση των εκτελέσιμων πρωτοτύπων έχουν καταδείξει ότι τα σημαντικότερα στοιχεία κινδύνου έχουν αντιμετωπιστεί και έχουν επιλυθεί αξιόπιστα, αν τα σχέδια επανάληψης για τη φάση Κατασκευής είναι ικανοποιητικά λεπτομερή και αξιόπιστα, ώστε να προχωρήσει η εργασία. Ακόμα, θα πρέπει τα σχέδια επανάληψης για τη φάση Κατασκευής να υποστηρίζονται με αξιόπιστες εκτιμήσεις, οι ενδιαφερόμενοι να συμφωνούν ότι το τρέχον όραμα μπορεί να γίνει πραγματικότητα, εάν το τρέχον σχέδιο εκτελεστεί για να αναπτύξει το πλήρες σύστημα, στα πλαίσια της τρέχουσας αρχιτεκτονικής. Τέλος, εξετάζεται εάν οι πραγματικές δαπάνες για πόρους έναντι των προγραμματισμένων δαπανών είναι αποδεκτές. Το έργο μπορεί να ανακληθεί ή να αναθεωρηθεί σημαντικά, εάν αποτύχει να φθάσει σε αυτό το ορόσημο.

Η προτεινόμενη διαδικασία για την ανάπτυξη πληροφοριακών έργων δίνει στον Αρχιτέκτονα Λογισμικού (Software Architect) και στο Διαχειριστή του Έργου (Project Manager) την ελευθερία να κατασκευάσουν πρωτότυπα (prototypes) διάφορων τύπων ως στρατηγική μείωσης του κινδύνου. Μερικά από αυτά τα πρωτότυπα μπορούν να είναι καθαρά διερευνητικά (exploratory), οπότε απορρίπτονται στη συνέχεια. Είναι πιθανό όμως, κυρίως για μεγαλύτερα συστήματα, η αρχιτεκτονική να έχει κατασκευαστεί σαν μια σειρά εξελικτικών πρωτοτύπων (evolutionary prototypes), καλύπτοντας διαφορετικά ζητήματα, καθώς η εκπόνηση μελέτης προχωρά. Είναι επιθυμητό στο τέλος του σταδίου αυτού, να έχουμε καταλήξει σε μια ολοκληρωμένη και σταθερή αρχιτεκτονική.

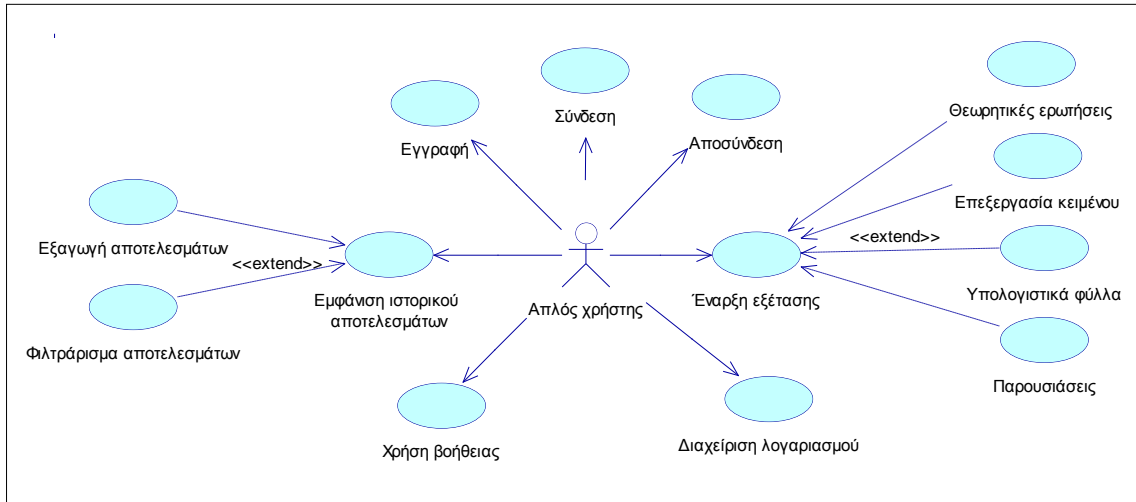
3.2 Περιπτώσεις Χρήσης - Use Cases

Τα διαγράμματα περιπτώσεων χρήσης περιγράφουν την συμπεριφορά του συστήματος από την πλευρά του χρήστη και με τον τρόπο αυτό καθορίζονται τα όρια του συστήματος και του περιβάλλοντός του. Κάθε περίπτωση χρήσης αντανακλά σε μια συγκεκριμένη λειτουργία του συστήματος. Το μοντέλο περιπτώσεων χρήσης περιλαμβάνει:

- Τις περιπτώσεις χρήσης (Use Cases)
- Τους ενεργοποιούς (Actors)

Με την έννοια «ενεργοποιός» εννοούμε το ρόλο που παίζει ένα άτομο που αλληλεπιδρά με το σύστημα. Οι ενεργοποιοί μπορούν να προσδιοριστούν εύκολα αν εντοπίσουμε τους χρήστες του συστήματος καθώς και κάθε άλλο εξωτερικό σύστημα που αλληλεπιδρά με αυτό που αναπτύσσεται. Για το συμβολισμό στο διάγραμμα χρησιμοποιούμε ένα ανθρωπάκι για να αναπαραστήσουμε τον ενεργοποιό και μια έλλειψη για να για να δηλώσουμε την περίπτωση χρήσης.

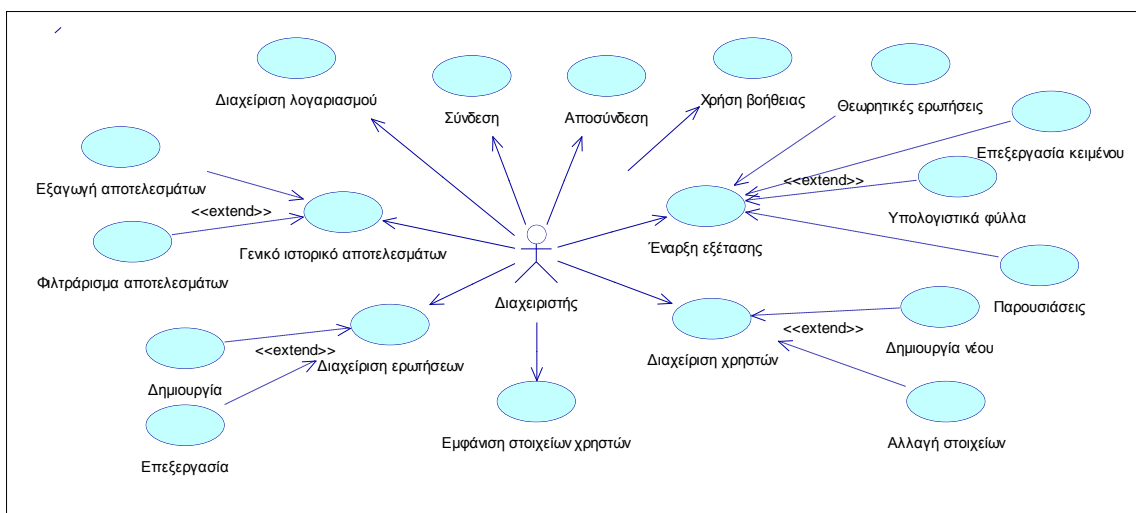
Στην ενότητα αυτή θα παρουσιάσουμε τις περιπτώσεις χρήσης και τα διαγράμματα περιπτώσεων χρήσης για την εκπαιδευτική εφαρμογή iTestU. Οι ενεργοποιοί του συστήματος μπορούν να μοντελοποιηθούν χρησιμοποιώντας έναν από τους ακόλουθους δύο ρόλους, το ρόλο του απλού χρήστη που μπορεί για παράδειγμα να είναι ένας εξεταζόμενος σπουδαστής και το ρόλο του διαχειριστή που μπορεί να είναι ο καθηγητής του.



Εικόνα 2. Διάγραμμα χρήσης απλού χρήστη

Περιπτώσεις χρήσης για τον απλό χρήστη της εφαρμογής είναι:

- Μελέτη περίπτωσης 1 - «Εγγραφή χρήστη στην εφαρμογή iTestU»
- Μελέτη περίπτωσης 2 - «Σύνδεση χρήστη στην εφαρμογή iTestU»
- Μελέτη περίπτωσης 3 - «Αποσύνδεση χρήστη από την εφαρμογή iTestU»
- Μελέτη περίπτωσης 4 - «Επιλογή λειτουργίας Έναρξη Εξέτασης»
- Μελέτη περίπτωσης 5 - «Εκκίνηση θεωρητικών ερωτήσεων»
- Μελέτη περίπτωσης 6 - «Εκκίνηση ερωτήσεων επεξεργασίας κειμένου»
- Μελέτη περίπτωσης 7 - «Εκκίνηση ερωτήσεων υπολογιστικών φύλλων»
- Μελέτη περίπτωσης 8 - «Εκκίνηση ερωτήσεων παρουσιάσεων»
- Μελέτη περίπτωσης 9 - «Διαχείριση λογαριασμού»
- Μελέτη περίπτωσης 10 - «Επιλογή λειτουργίας Χρήση Βοήθειας»
- Μελέτη περίπτωσης 11 - «Εμφάνιση ιστορικού αποτελεσμάτων»
- Μελέτη περίπτωσης 12 - «Φιλτράρισμα αποτελεσμάτων»
- Μελέτη περίπτωσης 13 - «Εξαγωγή αποτελεσμάτων»



Εικόνα 3. Διάγραμμα χρήσης διαχειριστή

Περιπτώσεις χρήσης για το διαχειριστή της εφαρμογής είναι:

- Μελέτη περίπτωσης 14 - «Διαχείριση χρηστών-δημιουργία νέου χρήστη»
- Μελέτη περίπτωσης 15 - «Διαχείριση χρηστών-αλλαγή στοιχείων χρήστη»
- Μελέτη περίπτωσης 16 - «Εμφάνιση στοιχείων χρηστών»
- Μελέτη περίπτωσης 17 - «Διαχείριση ερωτήσεων-δημιουργία νέας ερώτησης»
- Μελέτη περίπτωσης 18- «Διαχείριση ερωτήσεων-επεξεργασία παλιάς ερώτησης»
- Μελέτη περίπτωσης 19 - «Εμφάνιση γενικού ιστορικού αποτελεσμάτων»
- Μελέτη περίπτωσης 20 - «Φιλτράρισμα γενικών αποτελεσμάτων»
- Μελέτη περίπτωσης 21- «Εξαγωγή γενικών αποτελεσμάτων»

Μελέτη περίπτωσης 1 - «Εγγραφή χρήστη στην εφαρμογή iTestU»

Χειριστής: Απλός χρήστης

Περιγραφή: Εγγράφεται ένας νέος χρήστης για να μπορεί να εκτελέσει τις βασικές λειτουργίες της εφαρμογής.

Προαπαιτούμενο: Να έχει εκκινήσει την εφαρμογή.

Μετά-συνθήκες: Δημιουργείται μια νέα εγγραφή στη ΒΔ και ένα μήνυμα επιβεβαίωσης της εγγραφής εμφανίζεται στην οθόνη.

Βασική ροή γεγονότων:

1. Ο χρήστης επιλέγει να πραγματοποιήσει εγγραφή.
2. Εμφανίζεται η φόρμα εγγραφής χρήστη
3. Συμπληρώνει τα πεδία με τα στοιχεία του
4. Υποβάλλει την φόρμα εγγραφής
5. Ενημερώνεται η ΒΔ με τα στοιχεία του χρήστη
6. Εμφανίζεται μήνυμα επιβεβαίωσης για την επιτυχημένη εγγραφή του χρήστη

Εναλλακτική ροή γεγονότων (exception):

Εναλλακτική ροή Α: Ο χρήστης δεν συμπληρώνει όλα τα υποχρεωτικά πεδία και υποβάλλει τη φόρμα εγγραφής χρήστη.

1. Η εφαρμογή εμφανίζει προειδοποιητικό μήνυμα
2. Εμφανίζεται ξανά το ίδιο παράθυρο διαλόγου για την συμπλήρωση των στοιχείων που δεν συμπληρώθηκαν
3. Ο χρήστης συμπληρώνει τα στοιχεία και υποβάλλει εκ νέου τη φόρμα

Μελέτη περίπτωσης 2 - «Σύνδεση χρήστη στην εφαρμογή iTestU»

Χειριστής: Απλός χρήστης, διαχειριστής

Περιγραφή: Ο χειριστής εισάγει όνομα χρήστη και κωδικό πρόσβασης για να συνδεθεί στην εφαρμογή. Συνδέεται ανάλογα με το ρόλο του στο κεντρικό παράθυρο επιλογών της εφαρμογής.

Προαπαιτούμενο: Να έχει εκκινήσει την εφαρμογή και να έχει κάνει εγγραφή.

Μετά-συνθήκες: Ο χρήστης συνδέεται στο κεντρικό παράθυρο επιλογών της εφαρμογής για να χρησιμοποιήσει κάποια από τις λειτουργίες της.

Βασική ροή γεγονότων:

1. Ο χρήστης εκκινεί την εφαρμογή και επιλέγει να πραγματοποιήσει σύνδεση.
2. Εμφανίζεται η φόρμα σύνδεσης του χρήστη
3. Συμπληρώνει τα πεδία όνομα χρήστη και κωδικό πρόσβασης

4. Υποβάλλει την φόρμα ενεργοποιώντας το κουμπί Σύνδεση
5. Ελέγχεται αν υπάρχει εγγραφή στη ΒΔ με τα στοιχεία αυτά
6. Επιβεβαιώνεται η ταυτότητα του χρήστη
7. Ο χρήστης συνδέεται στο κεντρικό παράθυρο επιλογών

Εναλλακτική ροή γεγονότων (exception):

Εναλλακτική ροή Α: Ο χρήστης δεν συμπληρώνει σωστά τα πεδία όνομα χρήστη ή/και κωδικό πρόσβασης και υποβάλλει την φόρμα.

1. Η εφαρμογή εμφανίζει προειδοποιητικό μήνυμα
2. Εμφανίζεται ξανά το παράθυρο διαλόγου σύνδεσης χρήστη
3. Ο χρήστης συμπληρώνει τα στοιχεία και υποβάλλει τη φόρμα
4. Επιβεβαιώνεται η ταυτότητα του χρήστη
5. Ο χρήστης συνδέεται στο κεντρικό παράθυρο επιλογών

Εναλλακτική ροή Β: Ο χρήστης δεν συμπληρώνει τα πεδία ταυτοποίησης ή συμπληρώνει το ένα από αυτά και υποβάλλει τη φόρμα.

1. Η εφαρμογή εμφανίζει προειδοποιητικό μήνυμα
2. Εμφανίζεται ξανά το παράθυρο διαλόγου σύνδεσης χρήστη
3. Ο χρήστης συμπληρώνει τα στοιχεία και υποβάλλει τη φόρμα
4. Επιβεβαιώνεται η ταυτότητα του χρήστη
5. Ο χρήστης συνδέεται στο κεντρικό παράθυρο επιλογών

Μελέτη περίπτωσης 3 - «Αποσύνδεση χρήστη από την εφαρμογή iTestU»

Χειριστής: Απλός χρήστης, διαχειριστής

Περιγραφή: Ο χρήστης επιλέγει να τερματίσει την εφαρμογή.

Προσ απαιτούμενο: Να έχει εκκινήσει την εφαρμογή.

Μετά-συνθήκες: Ο χρήστης επιστρέφει στο περιβάλλον του λειτουργικού συστήματος.

Βασική ροή γεγονότων:

1. Ο χρήστης επιλέγει το κουμπί τερματισμού
2. Η εφαρμογή εμφανίζει προειδοποιητικό μήνυμα
3. Ο χρήστης επιβεβαιώνει τον τερματισμό
4. Η εφαρμογή iTestU τερματίζεται
5. Αποδεσμεύονται οι πόροι που είχαν δεσμευτεί από την εφαρμογή

Μελέτη περίπτωσης 4 - «Επιλογή λειτουργίας Έναρξη Εξέτασης»

Χειριστής: Απλός χρήστης, διαχειριστής

Περιγραφή: Ο χρήστης επιλέγει τη λειτουργία έναρξη εξέτασης.

Προσ απαιτούμενο: Να έχει εκκινήσει την εφαρμογή και να έχει συνδεθεί στον λογαριασμό του.

Μετά-συνθήκες: Εμφανίζεται η πρώτη θεωρητική ή πρακτική ερώτηση εξετάσεων

Βασική ροή γεγονότων:

1. Ο χρήστης συνδέεται στον προσωπικό του λογαριασμό
2. Η εφαρμογή iTestU εμφανίζει το κεντρικό παράθυρο επιλογών
3. Ο χρήστης επιλέγει τη λειτουργία Έναρξη εξέτασης
4. Ο χρήστης επιλέγει αντικείμενο και κατηγορία εξέτασης

5. Ο χρήστης ενεργοποιεί το κουμπί Συνέχεια
6. Η πρώτη ερώτηση εμφανίζεται

Εναλλακτική ροή γεγονότων (exception):

Εναλλακτική ροή Α: Ο χρήστης δεν ενεργοποιεί το κουμπί Συνέχεια αλλά το κουμπί Έξοδος.

1. Η εφαρμογή εμφανίζει προειδοποιητικό μήνυμα
2. Ο χρήστης επιβεβαιώνει ή απορρίπτει τον τερματισμό
3. Η εφαρμογή τερματίζεται ή συνεχίζει τη ροή της αντίστοιχα

Μελέτη περίπτωσης 5 - «Εκκίνηση θεωρητικών ερωτήσεων»

Χειριστής: Απλός χρήστης, διαχειριστής

Περιγραφή: Ο χρήστης απαντάει ερωτήσεις θεωρίας με τη μορφή πολλαπλών επιλογών και το σύστημα τον αξιολογεί.

Προσ απαιτούμενο: Να έχει επιλέξει το θεωρητικό αντικείμενο εξέτασης.

Μετά-συνθήκες: Ολοκληρώνεται η εξέταση και εμφανίζεται το παράθυρο αναλυτικών αποτελεσμάτων της εξέτασης

Βασική ροή γεγονότων Α: Ο χρήστης απαντάει στην ερώτηση και επιλέγει υποβολή

1. Η εκφώνηση της ερώτησης και οι 4 εναλλακτικές απαντήσεις φορτώνονται από τη ΒΔ και εμφανίζονται στο παράθυρο της εξέτασης
2. Ο χρήστης επιλέγει μια από τις απαντήσεις
3. Ο χρήστης υποβάλλει την απάντηση στο σύστημα
4. Η εφαρμογή iTestU αξιολογεί την απάντηση του χρήστη
5. Εμφανίζεται κατάλληλο μήνυμα ανάλογα με το αν είναι σωστή ή λάθος απάντηση
6. Το λογισμικό καταγράφει στη ΒΔ τα στοιχεία της ερώτησης και το βαθμό
7. Εμφανίζεται η επόμενη ερώτηση

Βασική ροή γεγονότων Β: Ο χρήστης επιλέγει να αναιρέσει την ερώτηση

1. Η εκφώνηση της ερώτησης και οι 4 εναλλακτικές απαντήσεις φορτώνονται από τη ΒΔ και εμφανίζονται στο παράθυρο της εξέτασης
2. Ο χρήστης επιλέγει μια από τις απαντήσεις
3. Ο χρήστης αναιρεί την απάντηση
4. Η εφαρμογή iTestU επιστρέφει την ερώτηση στην αρχική της μορφή
5. Ο χρήστης επιλέγει μια από τις απαντήσεις
6. Ο χρήστης υποβάλλει την απάντηση στο σύστημα
7. Η εφαρμογή iTestU αξιολογεί την απάντηση του χρήστη
8. Εμφανίζεται κατάλληλο μήνυμα ανάλογα με το αν είναι σωστή ή λάθος απάντηση
9. Εμφανίζεται η επόμενη ερώτηση

Βασική ροή γεγονότων Γ: Ο χρήστης επιλέγει να αγνοήσει την ερώτηση

1. Η εκφώνηση της ερώτησης και οι 4 εναλλακτικές απαντήσεις φορτώνονται από τη ΒΔ και εμφανίζονται στο παράθυρο της εξέτασης
2. Ο χρήστης αγνοεί την ερώτηση
3. Το λογισμικό καταγράφει στη ΒΔ τα στοιχεία της ερώτησης και το βαθμό
4. Εμφανίζεται η επόμενη ερώτηση

Βασική ροή γεγονότων Δ: Ο χρήστης επιλέγει να τερματίσει την εξέταση

1. Η εκφώνηση της ερώτησης και οι 4 εναλλακτικές απαντήσεις φορτώνονται από τη ΒΔ και εμφανίζονται στο παράθυρο της εξέτασης
2. Ο χρήστης επιλέγει τερματισμό για την ερώτηση
3. Το λογισμικό καταγράφει στη ΒΔ τα στοιχεία της ερώτησης και το βαθμό
4. Εμφανίζεται το παράθυρο των αναλυτικών αποτελεσμάτων

Εναλλακτική ροή γεγονότων (exception):

Εναλλακτική ροή Δ: Ο χρήστης επιλέγει να τερματίσει την εξέταση αλλά, στην εμφάνιση του προειδοποιητικού μηνύματος για τερματισμό, δεν αποδέχεται την ενέργεια

1. Η εφαρμογή εμφανίζει προειδοποιητικό μήνυμα για επιβεβαίωση τερματισμού
2. Ο χρήστης απορρίπτει τον τερματισμό της εξέτασης
3. Η εφαρμογή συνεχίζει τη βασική ροή της

Μελέτη περίπτωσης 6,7,8 - «Εκκίνηση πρακτικών ερωτήσεων»

Οι μελέτες περιπτώσεων 6,7,8 είναι ίδιες γι' αυτό και ενσωματώνεται η περιγραφή τους σε μια.

Χειριστής: Απλός χρήστης, διαχειριστής

Περιγραφή: Ο χρήστης απαντάει ερωτήσεις από το πρακτικό αντικείμενο εξέτασης σε μια από τις κατηγορίες εξετάσεων όπως επεξεργασία κειμένου, υπολογιστικά φύλλα, παρουσιάσεις και το σύστημα τον αξιολογεί.

Προαπαιτούμενο: Να έχει επιλέξει το πρακτικό αντικείμενο εξέτασης.

Μετά-συνθήκες: Ολοκληρώνεται η εξέταση με αποδέσμευση των χρησιμοποιούμενων πόρων και εμφανίζεται το παράθυρο αναλυτικών αποτελεσμάτων της εξέτασης

Βασική ροή γεγονότων Α: Ο χρήστης απαντάει στην ερώτηση και επιλέγει υποβολή

1. Η εκφώνηση της ερώτησης φορτώνεται από τη ΒΔ και ταυτόχρονα ανοίγει το κατάλληλο λογισμικό με το αρχείο που θα επεξεργαστεί, στο παράθυρο της εξέτασης
2. Ο χρήστης εκτελεί ενέργειες πάνω στο αρχείο επεξεργασίας
3. Ο χρήστης υποβάλλει την απάντηση στο σύστημα
4. Η εφαρμογή iTestU αξιολογεί την απάντηση του χρήστη
5. Εμφανίζεται κατάλληλο μήνυμα ανάλογα με το αν είναι σωστή ή λάθος απάντηση
6. Το λογισμικό καταγράφει στη ΒΔ τα στοιχεία της ερώτησης και το βαθμό
7. Εμφανίζεται το παράθυρο της επόμενης ερώτησης

Βασική ροή γεγονότων Β: Ο χρήστης επιλέγει να αναιρέσει την ερώτηση

1. Η εκφώνηση της ερώτησης φορτώνεται από τη ΒΔ και ταυτόχρονα ανοίγει το κατάλληλο λογισμικό με το αρχείο που θα επεξεργαστεί, στο παράθυρο της εξέτασης
2. Ο χρήστης εκτελεί ενέργειες πάνω στο αρχείο επεξεργασίας
3. Ο χρήστης αναιρεί την απάντηση
4. Η εφαρμογή iTestU επιστρέφει το αρχείο επεξεργασίας στην αρχική του μορφή
4. Η εφαρμογή συνεχίζει τη βασική ροή της

Βασική ροή γεγονότων Γ: Ο χρήστης επιλέγει να αγνοήσει την ερώτηση

1. Η εκφώνηση της ερώτησης φορτώνεται από τη ΒΔ και ταυτόχρονα ανοίγει το κατάλληλο λογισμικό με το αρχείο που θα επεξεργαστεί, στο παράθυρο της εξέτασης
2. Ο χρήστης αγνοεί την ερώτηση
3. Το λογισμικό καταγράφει στη ΒΔ τα στοιχεία της ερώτησης και το βαθμό
4. Εμφανίζεται το παράθυρο της επόμενης ερώτησης

Βασική ροή γεγονότων Δ: Ο χρήστης επιλέγει να τερματίσει την εξέταση

1. Η εκφώνηση της ερώτησης φορτώνεται από τη ΒΔ και ταυτόχρονα ανοίγει το κατάλληλο λογισμικό με το αρχείο που θα επεξεργαστεί, στο παράθυρο της εξέτασης
2. Ο χρήστης επιλέγει τερματισμό της εξέτασης μέσα από την ερώτηση
3. Το λογισμικό καταγράφει στη ΒΔ τα στοιχεία της ερώτησης και το βαθμό
4. Ολοκληρώνεται η εξέταση με αποδέσμευση των χρησιμοποιούμενων πόρων
5. Εμφανίζεται το παράθυρο των αναλυτικών αποτελεσμάτων

Εναλλακτική ροή γεγονότων (exception):

Εναλλακτική ροή Δ: Ο χρήστης επιλέγει να τερματίσει την εξέταση αλλά, στην εμφάνιση του προειδοποιητικού μηνύματος για τερματισμό, δεν αποδέχεται την ενέργεια

1. Η εφαρμογή εμφανίζει προειδοποιητικό μήνυμα για επιβεβαίωση τερματισμού
2. Ο χρήστης απορρίπτει τον τερματισμό της εξέτασης
3. Η εφαρμογή συνεχίζει τη βασική ροή της

Μελέτη περίπτωσης 9 - «Διαχείριση λογαριασμού»

Χειριστής: Απλός χρήστης, διαχειριστής

Περιγραφή: Ο χρήστης μπαίνει στο παράθυρο διαχείρισης λογαριασμού και επεξεργάζεται τα προσωπικά του στοιχεία.

Προσ απαιτούμενο: Να έχει συνδεθεί στην εφαρμογή iTestU.

Μετά-συνθήκες: Τα στοιχεία του προσωπικού του λογαριασμού ενημερώνονται.

Βασική ροή γεγονότων:

1. Ο χρήστης συνδέεται στο κεντρικό παράθυρο επιλογών
2. Επιλέγει τη λειτουργία Διαχείριση Λογαριασμού
3. Εμφανίζεται το παράθυρο με τα στοιχεία του χρήστη
4. Ο χρήστης τροποποιεί τα στοιχεία του
5. Ενεργοποιεί το κουμπί Αλλαγή
6. Ενημερώνεται η ΒΔ της εφαρμογής για τις αλλαγές που πραγματοποιήθηκαν
7. Εμφανίζεται μήνυμα πληροφόρησης χρήστη για επιτυχημένη ενημέρωση στοιχείων

Εναλλακτική ροή γεγονότων (exception):

Εναλλακτική ροή Α: Ο χρήστης δεν συμπληρώνει τα υποχρεωτικά πεδία.

1. Εμφανίζεται από την εφαρμογή προειδοποιητικό μήνυμα
2. Επιστρέφει η ροή γεγονότων στο παράθυρο Διαχείριση Λογαριασμού

Μελέτη περίπτωσης 10 - «Επιλογή λειτουργίας Χρήση Βοήθειας»

Χειριστής: Απλός χρήστης, διαχειριστής

Περιγραφή: Ο χρήστης ανοίγει το ηλεκτρονικό εγχειρίδιο χρήσης της εφαρμογής.

Προσ απαιτούμενο: Να έχει συνδεθεί στην εφαρμογή iTestU.

Μετά-συνθήκες: Εμφανίζεται το ηλεκτρονικό αρχείο.

Βασική ροή γεγονότων:

1. Ο χρήστης συνδέεται στο κεντρικό παράθυρο επιλογών
2. Επιλέγει τη λειτουργία Χρήση Βοήθειας
3. Εμφανίζεται το ηλεκτρονικό αρχείο της Βοήθειας

Εναλλακτική ροή γεγονότων (exception):

Εναλλακτική ροή A: Το ηλεκτρονικό αρχείο δεν ανοίγει γιατί δεν υπάρχει εγκατεστημένο στον Η/Υ του χρήστη κατάλληλο λογισμικό όπως Acrobat Reader.

1. Εμφανίζεται προειδοποιητικό μήνυμα
2. Επιστρέφει η ροή γεγονότων στο κεντρικό παράθυρο επιλογών

Μελέτη περίπτωσης 11 - «Εμφάνιση ιστορικού αποτελεσμάτων»

Χειριστής: Απλός χρήστης, διαχειριστής

Περιγραφή: Ο χρήστης επιλέγει τη λειτουργία Αποτελέσματα Εξετάσεων από το κεντρικό παράθυρο επιλογών και εμφανίζει το ιστορικό με τα αποτελέσματα στις εξετάσεις του.

Προαπαιτούμενο: Να έχει συνδεθεί στην εφαρμογή iTestU.

Μετά-συνθήκες: Εμφανίζεται το ιστορικό αποτελεσμάτων.

Βασική ροή γεγονότων:

1. Ο χρήστης συνδέεται στο κεντρικό παράθυρο επιλογών
2. Επιλέγει τη λειτουργία Αποτελέσματα Εξετάσεων
3. Η εφαρμογή εμφανίζει το παράθυρο με τα αποτελέσματα

Εναλλακτική ροή γεγονότων (exception):

Εναλλακτική ροή A: Ο χρήστης δεν έχει εκτελέσει καμία εξέταση.

1. Η εφαρμογή εμφανίζει το ιστορικό αποτελεσμάτων κενό
2. Επιστρέφει η ροή γεγονότων στο παράθυρο αποτελεσμάτων

Μελέτη περίπτωσης 12 - «Φιλτράρισμα ιστορικού αποτελεσμάτων»

Χειριστής: Απλός χρήστης, διαχειριστής

Περιγραφή: Ο χρήστης εμφανίζει στο ιστορικό αποτελεσμάτων εγγραφές εφαρμόζοντας κριτήρια ημερομηνίας και κριτήρια για το αντικείμενο εξέτασης.

Προαπαιτούμενο: Να έχει συνδεθεί στην λειτουργία Αποτελέσματα Εξετάσεων.

Μετά-συνθήκες: Εμφανίζονται τα φιλτραρισμένα δεδομένα.

Βασική ροή γεγονότων:

1. Ο χρήστης επιλέγει ένα ή περισσότερα κριτήρια και ενεργοποιεί το κουμπί εφαρμογή φίλτρου
2. Η εφαρμογή εμφανίζει το παράθυρο με τα φιλτραρισμένα αποτελέσματα

Εναλλακτική ροή γεγονότων (exception):

Εναλλακτική ροή A: Ο χρήστης ενεργοποιεί το κουμπί διαγραφή φίλτρου.

1. Ο χρήστης επιλέγει ένα ή περισσότερα κριτήρια και ενεργοποιεί το κουμπί διαγραφή φίλτρου
2. Η εφαρμογή εμφανίζει το παράθυρο με τα αρχικά αφιλτράριστα αποτελέσματα

Μελέτη περίπτωσης 13 - «Εξαγωγή αποτελεσμάτων»

Χειριστής: Απλός χρήστης, διαχειριστής

Περιγραφή: Ο χρήστης παίρνει μια αναφορά σε εξωτερικό αρχείο (excel ή pdf) με τα αποτελέσματα των εξετάσεών του.

Προαπαιτούμενο: Να έχει συνδεθεί στην λειτουργία Αποτελέσματα Εξετάσεων.

Μετά-συνθήκες: Δημιουργείται αρχείο με το όνομά του στην επιφάνεια εργασίας του Η/Υ.

Βασική ροή γεγονότων:

1. Ο χρήστης επιλέγει τον τύπο του αρχείου εξαγωγής και ενεργοποιεί το κουμπί Εξαγωγή.
2. Η εφαρμογή δημιουργεί το αρχείο εξαγωγής
3. Το αρχείο εξαγωγής αποθηκεύεται αυτόματα στην επιφάνεια εργασίας

Εναλλακτική ροή γεγονότων (exception):

Εναλλακτική ροή A: Ο χρήστης ενεργοποιεί το κουμπί Εξαγωγή χωρίς να έχει επιλέξει τον τύπο του αρχείου εξαγωγής.

1. Η εφαρμογή εμφανίζει προειδοποιητικό μήνυμα για την επιλογή του τύπου του αρχείου εξαγωγής.
2. Επιστρέφει η ροή γεγονότων στο παράθυρο αποτελεσμάτων

Μελέτη περίπτωσης 14 - «Διαχείριση χρηστών – δημιουργία νέου χρήστη»

Χειριστής: Διαχειριστής

Περιγραφή: Ο διαχειριστής δημιουργεί ένα νέο χρήστη για την εφαρμογή iTestU.

Προσ απαιτούμενο: Να έχει συνδεθεί στην εφαρμογή με λογαριασμό διαχειριστή και να βρίσκεται στο κεντρικό παράθυρο επιλογών.

Μετά-συνθήκες: Δημιουργείται μια νέα εγγραφή στη ΒΔ στον πίνακα tblMembers

Βασική ροή γεγονότων:

1. Ο διαχειριστής από το κεντρικό παράθυρο επιλογών ενεργοποιεί τη λειτουργία Διαχείριση χρηστών
2. Το παράθυρο της διαχείρισης χρηστών εμφανίζεται
3. Ο διαχειριστής επιλέγει την καρτέλα Δημιουργία Νέου Χρήστη
4. Το παράθυρο της δημιουργίας νέου χρήστη γίνεται το ενεργό παράθυρο
5. Ο διαχειριστής εισάγει τα στοιχεία του νέου χρήστη
6. Ο διαχειριστής ενεργοποιεί το κουμπί Εγγραφή
7. Μια νέα εγγραφή δημιουργείται στη ΒΔ της εφαρμογής
8. Η εφαρμογή εμφανίζει μήνυμα ότι η διαδικασία ολοκληρώθηκε με επιτυχία

Εναλλακτική ροή γεγονότων (exception):

Εναλλακτική ροή A: Ο διαχειριστής δεν συμπληρώνει όλα τα υποχρεωτικά πεδία.

1. Εμφανίζεται από την εφαρμογή προειδοποιητικό μήνυμα
2. Ο χρήστης συμπληρώνει όλα τα υποχρεωτικά πεδία και ενεργοποιεί το κουμπί Εγγραφή
3. Η ροή γεγονότων συνεχίζεται κανονικά

Μελέτη περίπτωσης 15 - «Διαχείριση χρηστών – αλλαγή Στοιχείων χρήστη»

Χειριστής: Διαχειριστής

Περιγραφή: Ο διαχειριστής αλλάζει τα στοιχεία κάποιου χρήστη της εφαρμογής iTestU.

Προσ απαιτούμενο: Να έχει συνδεθεί στην εφαρμογή με λογαριασμό διαχειριστή και να βρίσκεται στο κεντρικό παράθυρο επιλογών.

Μετά-συνθήκες: Ενημερώνεται μια υπάρχουσα εγγραφή στη ΒΔ στον πίνακα tblMembers

Βασική ροή γεγονότων:

1. Ο διαχειριστής από το κεντρικό παράθυρο επιλογών ενεργοποιεί τη λειτουργία Διαχείριση χρηστών
2. Το παράθυρο της διαχείρισης χρηστών εμφανίζεται
3. Ο διαχειριστής επιλέγει την καρτέλα Αλλαγή Στοιχείων Χρήστη

4. Το παράθυρο της επεξεργασίας των στοιχείων του χρήστη γίνεται το ενεργό παράθυρο
5. Ο διαχειριστής αναζητά το χρήστη που θα επεξεργαστεί με βάση το επίθετό του
6. Ο διαχειριστής ενεργοποιεί το κουμπί της αναζήτησης εγγραφής
7. Τα στοιχεία του χρήστη εντοπίζονται από τη ΒΔ και εμφανίζονται στη φόρμα
8. Τα πεδία της φόρμας γίνονται ενεργά και μπορούν να επεξεργαστούν
9. Ο διαχειριστής επεξεργάζεται ένα ή περισσότερα πεδία της φόρμας
10. Ο διαχειριστής ενεργοποιεί το κουμπί Αλλαγή
11. Η εγγραφή του πίνακα tblMembers στη ΒΔ ενημερώνεται με τα νέα στοιχεία
12. Εμφανίζεται μήνυμα ότι η διαδικασία της ενημέρωσης ολοκληρώθηκε με επιτυχία

Εναλλακτική ροή γεγονότων (exception):

Εναλλακτική ροή Α: Ο διαχειριστής δεν συμπληρώνει όλα τα υποχρεωτικά πεδία.

1. Εμφανίζεται από την εφαρμογή προειδοποιητικό μήνυμα
2. Ο χρήστης συμπληρώνει όλα τα υποχρεωτικά πεδία και ενεργοποιεί το κουμπί Εγγραφή
3. Η ροή γεγονότων συνεχίζεται κανονικά

Μελέτη περίπτωσης 16 - «Εμφάνιση στοιχείων χρηστών»

Χειριστής: Διαχειριστής

Περιγραφή: Ο διαχειριστής δημιουργεί και εμφανίζει μια αναφορά με τα στοιχεία ενός ή περισσότερων χρηστών της εφαρμογής iTestU.

Προαπαιτούμενο: Να έχει συνδεθεί στην εφαρμογή με λογαριασμό διαχειριστή και να βρίσκεται στο κεντρικό παράθυρο επιλογών.

Μετά-συνθήκες: Εμφανίζεται η αναφορά

Βασική ροή γεγονότων:

1. Ο διαχειριστής από το κεντρικό παράθυρο επιλογών ενεργοποιεί τη λειτουργία Στοιχεία χρηστών
2. Εμφανίζεται το παράθυρο για την επιλογή των επιθέτων με τη λίστα επιθέτων
3. Ο διαχειριστής επιλέγει ένα ή περισσότερα επίθετα και ενεργοποιεί το κουμπί OK
4. Η πρόσθετη εφαρμογή Crystal Report δημιουργεί την αναφορά

Εναλλακτική ροή γεγονότων (exception):

Εναλλακτική ροή Α: Ο διαχειριστής δεν επιλέγει κανένα επίθετο από την λίστα.

1. Η εφαρμογή Crystal Report δημιουργεί την αναφορά με όλα τους χρήστες
2. Η ροή γεγονότων συνεχίζεται κανονικά

Μελέτη περίπτωσης 17 - «Διαχείριση ερωτήσεων - δημιουργία νέας ερώτησης»

Χειριστής: Διαχειριστής

Περιγραφή: Ο διαχειριστής δημιουργεί μια νέα ερώτηση για την εφαρμογή iTestU.

Προαπαιτούμενο: Να έχει συνδεθεί στην εφαρμογή με λογαριασμό διαχειριστή και να βρίσκεται στο κεντρικό παράθυρο επιλογών.

Μετά-συνθήκες: Δημιουργείται μια νέα εγγραφή στη ΒΔ στον πίνακα Questions και tblAnswers

Βασική ροή γεγονότων:

1. Ο διαχειριστής από το κεντρικό παράθυρο επιλογών ενεργοποιεί τη λειτουργία Διαχείριση Ερωτήσεων
2. Το παράθυρο της διαχείρισης ερωτήσεων εμφανίζεται

3. Ο διαχειριστής επιλέγει την καρτέλα Δημιουργία
4. Το παράθυρο της δημιουργίας νέας ερώτησης γίνεται το ενεργό παράθυρο
5. Ο διαχειριστής εισάγει στα πεδία της φόρμας τα στοιχεία της ερώτησης
6. Ο διαχειριστής ενεργοποιεί το κουμπί Δημιουργία
7. Μια νέα εγγραφή δημιουργείται στη ΒΔ της εφαρμογής
8. Εμφανίζεται μήνυμα ότι η διαδικασία δημιουργίας της ερώτησης ολοκληρώθηκε

Εναλλακτική ροή γεγονότων (exception):

Εναλλακτική ροή Α: Ο διαχειριστής δε συμπληρώνει όλα τα πεδία με τα στοιχεία της ερώτησης.

1. Εμφανίζεται από την εφαρμογή προειδοποιητικό μήνυμα
2. Ο χρήστης συμπληρώνει όλα τα υποχρεωτικά πεδία και ενεργοποιεί το κουμπί Εγγραφή
3. Η ροή γεγονότων συνεχίζεται κανονικά

Μελέτη περίπτωσης 18 - «Διαχείριση χρηστών – επεξεργασία παλιάς ερώτησης»

Χειριστής: Διαχειριστής

Περιγραφή: Ο διαχειριστής επεξεργάζεται το περιεχόμενο κάποιας υπάρχουσας ερώτησης.

Προαπαιτούμενο: Να έχει συνδεθεί στην εφαρμογή με λογαριασμό διαχειριστή και να βρίσκεται στο κεντρικό παράθυρο επιλογών.

Μετά-συνθήκες: Ενημερώνεται μια υπάρχουσα εγγραφή στον πίνακα Questions και μια εγγραφή στον πίνακα tblAnswers της βάσης δεδομένων

Βασική ροή γεγονότων:

1. Ο διαχειριστής από το κεντρικό παράθυρο επιλογών ενεργοποιεί τη λειτουργία Διαχείριση Ερωτήσεων
2. Το παράθυρο της διαχείρισης ερωτήσεων εμφανίζεται
3. Ο διαχειριστής επιλέγει την καρτέλα Επεξεργασία
4. Το παράθυρο της επεξεργασίας των ερωτήσεων γίνεται το ενεργό παράθυρο
5. Ο διαχειριστής αναζητά την ερώτηση που θα επεξεργαστεί με βάση τον κωδικό της
6. Ο διαχειριστής ενεργοποιεί το κουμπί της αναζήτησης της ερώτησης
7. Τα στοιχεία της ερώτησης εντοπίζονται στη ΒΔ και εμφανίζονται στη φόρμα
8. Τα πεδία της φόρμας γίνονται ενεργά και μπορούν πλέον να επεξεργαστούν
9. Ο διαχειριστής επεξεργάζεται ένα ή περισσότερα πεδία της φόρμας
10. Ο διαχειριστής ενεργοποιεί το κουμπί Τροποποίηση
11. Ενημερώνονται οι εγγραφές της ερώτησης στους πίνακες Questions και tblAnswers
12. Εμφανίζεται μήνυμα ότι η διαδικασία επεξεργασίας της ερώτησης ολοκληρώθηκε και η ερώτηση ενημερώθηκε με επιτυχία

Εναλλακτική ροή γεγονότων (exception):

Εναλλακτική ροή Α: Ο διαχειριστής δεν συμπληρώνει όλα τα πεδία της ερώτησης.

1. Εμφανίζεται από την εφαρμογή προειδοποιητικό μήνυμα
2. Ο χρήστης συμπληρώνει όλα τα πεδία και ενεργοποιεί το κουμπί Τροποποίηση
3. Η ροή γεγονότων συνεχίζεται κανονικά

Μελέτη περίπτωσης 19 - «Εμφάνιση γενικού ιστορικού αποτελεσμάτων»

Χειριστής: Διαχειριστής

Περιγραφή: Ο διαχειριστής επιλέγει τη λειτουργία Γενικά Αποτελέσματα από το κεντρικό παράθυρο επιλογών και εμφανίζει το ιστορικό με τα αποτελέσματα όλων των χρηστών.

Προαπαιτούμενο: Να έχει συνδεθεί στην εφαρμογή iTestU με λογαριασμό διαχειριστή.

Μετά-συνθήκες: Εμφανίζεται το γενικό ιστορικό αποτελεσμάτων.

Βασική ροή γεγονότων:

1. Ο διαχειριστής συνδέεται στο κεντρικό παράθυρο επιλογών
2. Επιλέγει τη λειτουργία Γενικά Αποτελέσματα
3. Η εφαρμογή εμφανίζει το γενικό ιστορικό αποτελεσμάτων

Εναλλακτική ροή γεγονότων (exception):

Εναλλακτική ροή Α: Κανένας χρήστης δεν έχει εκτελέσει εξετάσεις.

1. Η εφαρμογή εμφανίζει το γενικό ιστορικό αποτελεσμάτων κενό
2. Επιστρέφει η ροή γεγονότων στο παράθυρο αποτελεσμάτων

Μελέτη περίπτωσης 20 - «Φιλτράρισμα γενικού ιστορικού αποτελεσμάτων»

Χειριστής: Διαχειριστής

Περιγραφή: Ο διαχειριστής εφαρμόζει κριτήρια ημερομηνίας και κριτήρια για το αντικείμενο εξέτασης και εμφανίζει στο γενικό ιστορικό αποτελεσμάτων εγγραφές.

Προαπαιτούμενο: Να έχει συνδεθεί στην λειτουργία Γενικά Αποτελέσματα.

Μετά-συνθήκες: Εμφανίζονται τα φιλτραρισμένα δεδομένα.

Βασική ροή γεγονότων:

1. Ο χρήστης επιλέγει ένα ή περισσότερα κριτήρια και ενεργοποιεί το κουμπί εφαρμογή φίλτρου
2. Η εφαρμογή εμφανίζει το παράθυρο με τα φιλτραρισμένα αποτελέσματα

Εναλλακτική ροή γεγονότων (exception):

Εναλλακτική ροή Α: Ο χρήστης ενεργοποιεί το κουμπί διαγραφή φίλτρου.

1. Ο χρήστης επιλέγει ένα ή περισσότερα κριτήρια και ενεργοποιεί το κουμπί διαγραφή φίλτρου
2. Η εφαρμογή εμφανίζει το παράθυρο με τα αρχικά αφιλτράριστα αποτελέσματα

Μελέτη περίπτωσης 21 - «Εξαγωγή γενικών αποτελεσμάτων»

Χειριστής: Διαχειριστής

Περιγραφή: Ο χρήστης παίρνει μια αναφορά σε εξωτερικό αρχείο (excel ή pdf) με τα αποτελέσματα των εξετάσεων ενός ή περισσότερων χρηστών.

Προαπαιτούμενο: Να έχει συνδεθεί στην λειτουργία Γενικά Αποτελέσματα.

Μετά-συνθήκες: Δημιουργείται αρχείο γενικών αποτελεσμάτων στην επιφάνεια εργασίας.

Βασική ροή γεγονότων:

1. Ο χρήστης επιλέγει τον τύπο του αρχείου εξαγωγής και ενεργοποιεί το κουμπί Εξαγωγή.
2. Η εφαρμογή δημιουργεί το αρχείο εξαγωγής
3. Το αρχείο εξαγωγής αποθηκεύεται αυτόματα στην επιφάνεια εργασίας

Εναλλακτική ροή γεγονότων (exception):

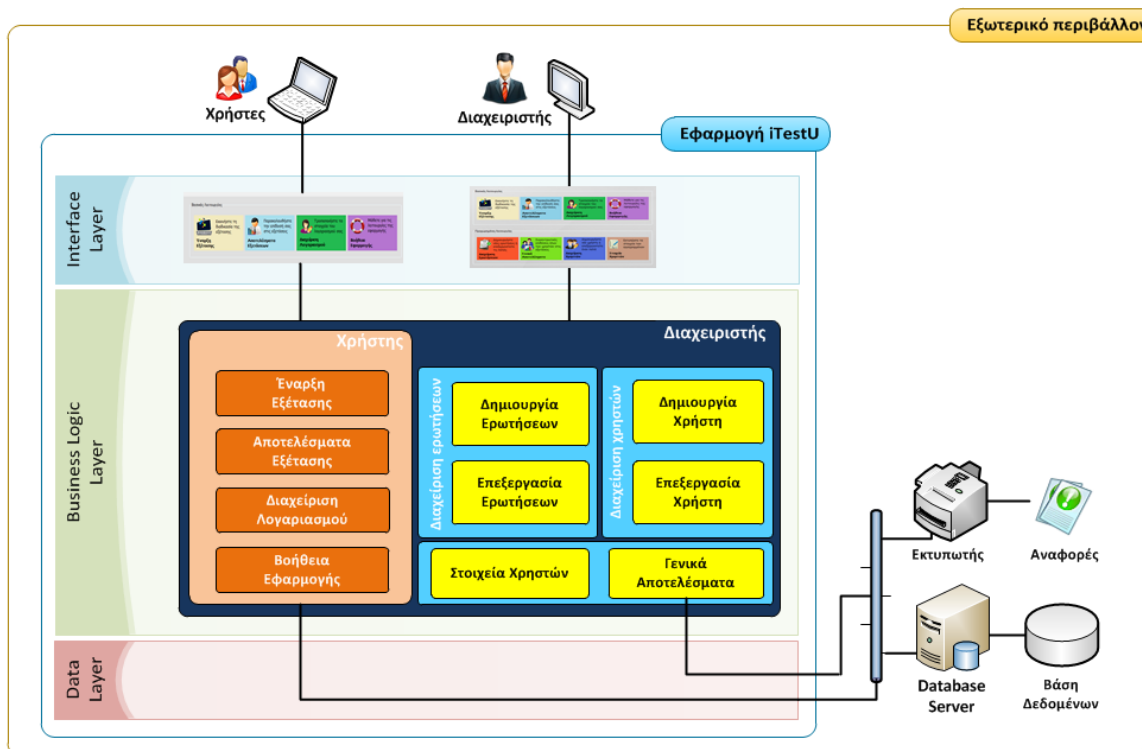
Εναλλακτική ροή Α: Ο χρήστης ενεργοποιεί το κουμπί Εξαγωγή χωρίς να έχει επιλέξει τον τύπο του αρχείου εξαγωγής.

1. Η εφαρμογή εμφανίζει προειδοποιητικό μήνυμα για την επιλογή του τύπου του αρχείου εξαγωγής.
2. Επιστρέφει η ροή γεγονότων στο παράθυρο αποτελεσμάτων

3.3 Η αρχιτεκτονική του συστήματος

Πριν προχωρήσουμε στην επόμενη φάση, την φάση της κατασκευής, θα περιγράψουμε το αρχιτεκτονικό μοντέλο στο οποίο καταλήξαμε να υλοποιήσουμε για το εκπαιδευτικό πληροφοριακό σύστημα iTestU. Το μοντέλο αυτό μπορεί να διαχωριστεί σε τρία επίπεδα. Τα επίπεδα αυτά είναι:

- Το επίπεδο της επικοινωνίας/διεπαφής των χρηστών (interface layer) με την εφαρμογή. Σε αυτό το επίπεδο παρουσιάζεται ο τρόπος επικοινωνίας του διαχειριστή και των απλών χρηστών με την κεντρική οθόνη επιλογών του πληροφοριακού συστήματος. Να σημειώσουμε ότι η οθόνη αυτή εμφανίζεται διαφορετικά για τον διαχειριστή σε σχέση με έναν απλό χρήστη.
- Το μεσαίο επίπεδο ή αλλιώς το επίπεδο της επιχειρησιακής λογικής (business logic layer) όπου εκτελούνται οι λειτουργίες της εφαρμογής.
- Το επίπεδο των δεδομένων (data layer) που αποτελεί τη διασύνδεση της εφαρμογής με τη βάση δεδομένων.



Εικόνα 4. Αρχιτεκτονική πληροφοριακού συστήματος iTestU

Στην (Εικόνα 4) παρουσιάζεται η αρχιτεκτονική του πληροφοριακού συστήματος iTestU μαζί με το εξωτερικό περιβάλλον του. Το βασικότερο τμήμα της αρχιτεκτονικής αποτελεί το επίπεδο της επιχειρησιακής λογικής όπου παρουσιάζονται οι λειτουργίες του συστήματος. Ανάλογα με τα δικαιώματα του χρήστη που συνδέεται στην εφαρμογή, μπορούμε να διακρίνουμε δύο κατηγορίες λειτουργιών. Τις βασικές λειτουργίες, που εκτελούνται από τους απλούς χρήστες ενώ ταυτόχρονα αποτελούν τμήμα των ενεργειών που μπορεί να εκτελέσει ο

διαχειριστής και τις προχωρημένες λειτουργίες που εκτελούνται αποκλειστικά από τον διαχειριστή.

Οι βασικές λειτουργίες της εφαρμογής είναι:

- Η *έναρξη εξέτασης*, όπου μέσα από αυτήν τη λειτουργία ο χρήστης μπορεί να εκκινήσει τη διαδικασία θεωρητικής ή πρακτικής εξέτασης.
- Η λειτουργία *αποτελέσματα εξετάσεων* που δίνει στο χρήστη τη δυνατότητα να αναζητήσει το ιστορικό των αποτελεσμάτων των εξετάσεων που έχει πραγματοποιήσει.
- Η *διαχείριση λογαριασμού* μέσα από την οποία ο χρήστης μπορεί να επεξεργαστεί τον προσωπικό του λογαριασμό.
- Η *βοήθεια εφαρμογής*, που είναι ένα ηλεκτρονικό εγχειρίδιο που εξηγεί τον τρόπο που δουλεύει η εφαρμογή.

Οι προχωρημένες λειτουργίες της εφαρμογής είναι:

- Η *διαχείριση ερωτήσεων*, που δίνει τη δυνατότητα στον διαχειριστή να δημιουργήσει νέες θεωρητικές ερωτήσεις πολλαπλής επιλογής καθώς και να επεξεργαστεί παλιές ερωτήσεις.
- Η *διαχείριση χρηστών*, μέσα από την οποία ο διαχειριστής μπορεί να δημιουργήσει νέους χρήστες και να επεξεργαστεί τα στοιχεία των παλιών χρηστών.
- Η λειτουργία *γενικά αποτελέσματα*, όπου δίνεται στον διαχειριστή η δυνατότητα να παρακολουθήσει το ιστορικό των αποτελεσμάτων των εξετάσεων όλων των χρηστών της εφαρμογής.
- Τα *στοιχεία χρηστών*, όπου ο διαχειριστής μπορεί να δημιουργήσει αναφορές με τα στοιχεία των εγγεγραμμένων μελών.

Το μεσαίο επίπεδο επικοινωνεί άμεσα με το επίπεδο των δεδομένων (data layer). Η επικοινωνία αυτή περιλαμβάνει ανάκτηση δεδομένων και τροφοδότηση με δεδομένα του επιπέδου της επιχειρησιακής λογικής και αποθήκευση των δεδομένων που προέρχονται από το επίπεδο της επιχειρησιακής λογικής στο επίπεδο δεδομένων.

Στο επίπεδο των δεδομένων χρησιμοποιούμε κατάλληλα συστήματα διαχείρισης βάσεων δεδομένων που αναλαμβάνουν να κάνουν την επικοινωνία με την χρήση ερωτημάτων.

4 Φάση Κατασκευής

Το στάδιο αυτό περιλαμβάνει την ανάπτυξη του προϊόντος σε μια σειρά βημάτων που επαναλαμβάνονται. Στην ενότητα 4.1 γίνεται μια θεωρητική ανάλυση για το τι περιλαμβάνει η φάση αυτή, στην ενότητα 4.2 γίνεται η ανάλυση και ο σχεδιασμός της εφαρμογής, στην ενότητα 4.3 ακολουθεί η υλοποίηση-κωδικοποίηση και τέλος στην ενότητα 4.4 περιγράφεται η διαδικασία που ακολουθήσαμε για τον έλεγχο της εφαρμογής.

4.1 Η Φάση της Κατασκευής (Construction Phase)

Ο στόχος της φάσης Κατασκευής είναι η διευκρίνιση των υπόλοιπων απαιτήσεων και η ολοκλήρωση της ανάπτυξης του συστήματος βάσει της βασικής αρχιτεκτονικής. Η φάση Κατασκευής είναι, κατά μια έννοια, μια διαδικασία, όπου η έμφαση δίνεται στη διαχείριση των πόρων και στον έλεγχο των λειτουργιών, ώστε να βελτιστοποιηθούν οι δαπάνες, τα χρονοδιαγράμματα και η ποιότητα. Υπό αυτήν την έννοια, η νοοτροπία της διαχείρισης υποβάλλεται σε μια μετάβαση από την ανάπτυξη της πνευματικής ιδιοκτησίας κατά τη διάρκεια της Έναρξης και της Επεξεργασίας, στην ανάπτυξη προϊόντων που μπορούν να επεκταθούν (deployable products) κατά τη διάρκεια της Κατασκευής και της Μετάβασης.

Οι κύριοι στόχοι της φάσης Κατασκευής είναι:

- Να μειωθούν οι δαπάνες ανάπτυξης βελτιστοποιώντας τους πόρους και αποφεύγοντας τις περιττές εργασίες και τις επαναλήψεις.
- Να επιτευχθεί επαρκής ποιότητα τόσο γρήγορα όσο και πρακτικά.
- Να δημιουργηθούν χρήσιμες εκδόσεις (alpha, beta και άλλες εκδόσεις ελέγχου) τόσο γρήγορα όσο και πρακτικά.
- Να ολοκληρωθεί η ανάλυση, το σχέδιο, η ανάπτυξη και η δοκιμή όλων των απαραίτητων λειτουργιών.
- Να αναπτυχθεί επαναληπτικά και σταδιακά ένα πλήρες προϊόν, έτοιμο για μετάβαση στους χρήστες του. Αυτό προϋποθέτει την περιγραφή των υπόλοιπων περιπτώσεων χρήσης και των άλλων απαιτήσεων, τη συμπλήρωση του σχεδίου, την ολοκλήρωση της υλοποίησης και τον έλεγχο του λογισμικού.
- Να αποφασιστεί εάν το λογισμικό και οι χρήστες είναι όλοι έτοιμοι για την εφαρμογή που πρόκειται να επεκταθεί.
- Να επιτευχθεί η παράλληλη εργασία των ομάδων ανάπτυξης μέχρι κάποιο βαθμό. Ακόμη και στα μικρότερα έργα, συνήθως υπάρχουν τμήματα (components) που μπορούν να αναπτυχθούν ανεξάρτητα το ένα από το άλλο, κάτι που επιτρέπει στις ομάδες να δουλεύουν παράλληλα, εφόσον οι πόροι το επιτρέπουν. Αυτή η παράλληλη εργασία μπορεί να επιταχύνει σημαντικά τις δραστηριότητες ανάπτυξης, αλλά ταυτόχρονα αυξάνει την πολυπλοκότητα της διαχείρισης πόρων και του συγχρονισμού της ροής της εργασίας.

Οι σημαντικές δραστηριότητες της φάσης Κατασκευής περιλαμβάνουν:

- Τη διαχείριση των πόρων, τη βελτιστοποίηση του ελέγχου και της διαδικασίας.
- Την πλήρη ανάπτυξη και τον έλεγχο των τμημάτων (components) σύμφωνα με τα ορισμένα κριτήρια αξιολόγησης.
- Την αξιολόγηση των εκδόσεων του προϊόντος βάσει των κριτηρίων αποδοχής.

Στο τέλος της φάσης Κατασκευής έχουμε ένα ακόμα ορόσημο, το ορόσημο της Αρχικής Ικανότητας Λειτουργίας (Initial Operational Capability Milestone): Σ' αυτό το ορόσημο, το προϊόν είναι έτοιμο να παραδοθεί στην ομάδα Μετάβασης. Όλες οι λειτουργίες έχουν αναπτυχθεί και ο έλεγχος άλφα (alpha testing) έχει ολοκληρωθεί. Εκτός από το λογισμικό, έχει δημιουργηθεί ένα εγχειρίδιο για τους χρήστες και υπάρχει μια περιγραφή της τρέχουσας κυκλοφορίας του προϊόντος.

Κριτήρια αξιολόγησης για το εάν πληρείται το ορόσημο είναι: αν το προϊόν είναι αρκετά σταθερό και ώριμο, ώστε να παραδοθεί στην κοινότητα των χρηστών και αν οι πραγματικές δαπάνες για πόρους έναντι των προγραμματισμένων είναι μέσα στα αρχικά πλαίσια πρόβλεψης και είναι ακόμα αποδεκτές. Η φάση Μετάβασης μπορεί να αναβληθεί, εάν το έργο αποτύχει να φθάσει σε αυτό το ορόσημο.

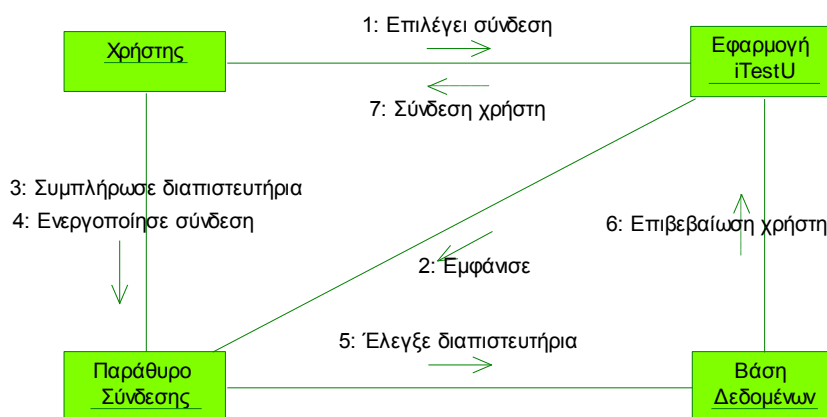
4.2 Ανάλυση και Σχεδιασμός

Σε αυτή την ενότητα θα παρουσιάσουμε τα υπόλοιπα διαγράμματα της Unified Modeling Language - UML. Συγκεκριμένα θα δούμε τα διαγράμματα συνεργασίας, τα διαγράμματα ακολουθίας ή σειράς, τα διαγράμματα δραστηριοτήτων, το διάγραμμα εξαρτημάτων και τέλος το διάγραμμα διανομής.

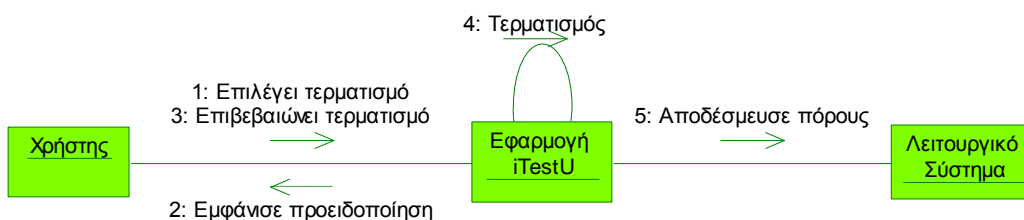
4.2.1 Διαγράμματα Συνεργασίας - Collaboration Diagrams

Στα συστήματα αντικειμενοστραφούς προγραμματισμού τα αντικείμενα δεν είναι στατικά αλλά αλληλεπιδρούν μεταξύ τους ανταλλάσσοντας μηνύματα. Η αναπαράσταση της αλληλεπίδρασης γίνεται με τα διαγράμματα αλληλεπίδρασης (interaction diagrams). Με τον όρο αλληλεπίδραση εννοούμε τον τρόπο με τον οποίο ανταλλάσσονται τα μηνύματα μεταξύ των αντικειμένων. Τα διαγράμματα αλληλεπίδρασης τα συναντάμε με δύο μορφές, τα διαγράμματα συνεργασίας που θα δούμε σε αυτή την ενότητα και τα διαγράμματα ακολουθίας που θα αναλύσουμε στην επόμενη ενότητα.

Τα διαγράμματα συνεργασίας χρησιμοποιούν μια χωρική στατική δομή για να αναπαραστήσουν τη συνεργασία μεταξύ των αντικειμένων. Για να απεικονίσουν την ακολουθία των μηνυμάτων που ανταλλάσσουν τα αντικείμενα χρησιμοποιείται αρίθμηση. Ακολουθούν τα διαγράμματα για τη σύνδεση και την αποσύνδεση του χρήστη από την εφαρμογή.

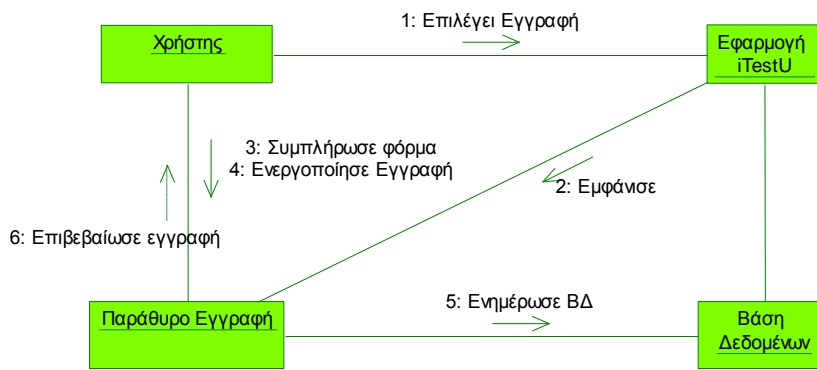


Εικόνα 5. Διάγραμμα συνεργασίας για τη σύνδεση χρήστη

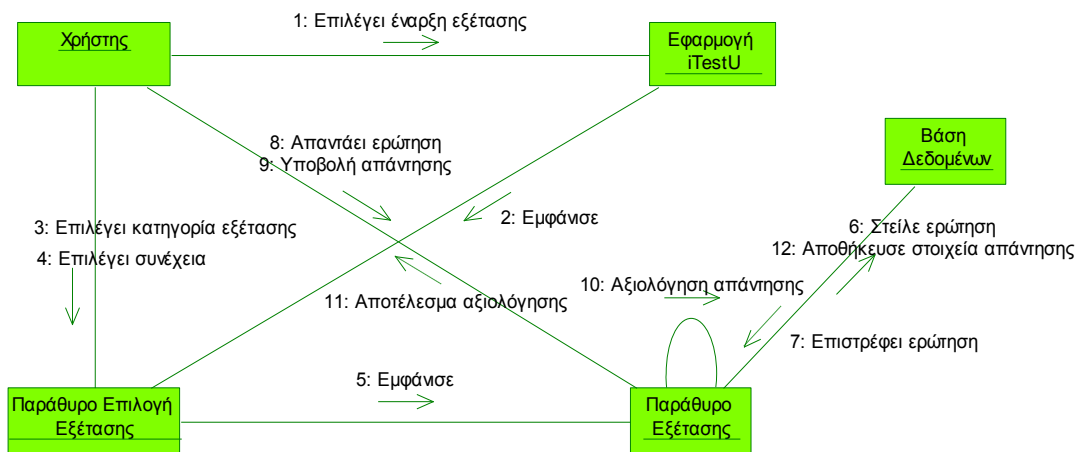


Εικόνα 6. Διάγραμμα συνεργασίας για την αποσύνδεση χρήστη

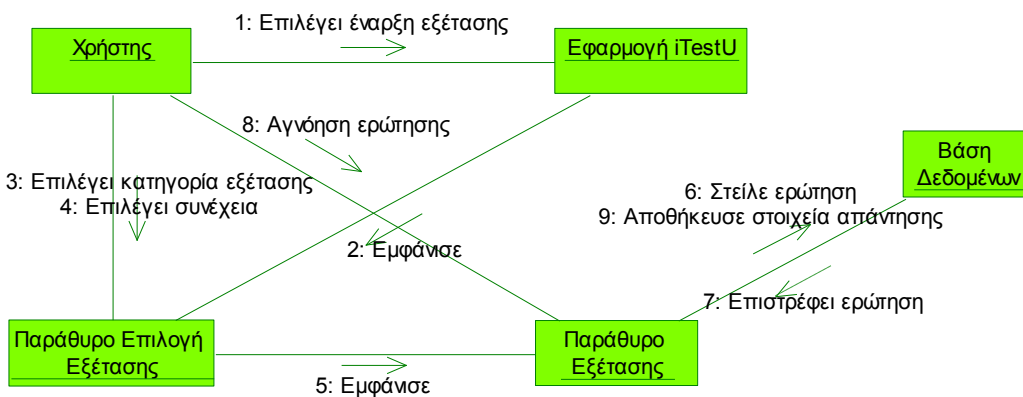
Στα επόμενα διαγράμματα συνεργασίας μπορούμε να δούμε την αλληλεπίδραση των αντικειμένων για να περιγράψουμε την εγγραφή του χρήστη στην εφαρμογή (Εικόνα 7) καθώς και την περίπτωση της υποβολής απάντησης σε ερώτηση εξετάσεων (Εικόνα 8), της αγνόησης ερώτησης (Εικόνα 9) και της αναίρεσης ερώτησης κατά τη διαδικασία της εξέτασης (Εικόνα 10). Η Εικόνα 11 δείχνει τη συνεργασία των αντικείμενων για την εμφάνιση των αποτελεσμάτων από τις εξετάσεις που έχει εκτελέσει ο χρήστης.



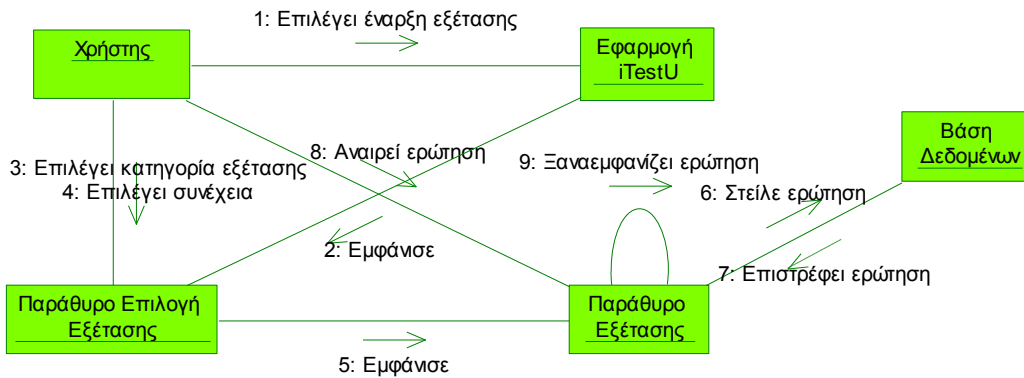
Εικόνα 7. Διάγραμμα συνεργασίας για την εγγραφή χρήστη



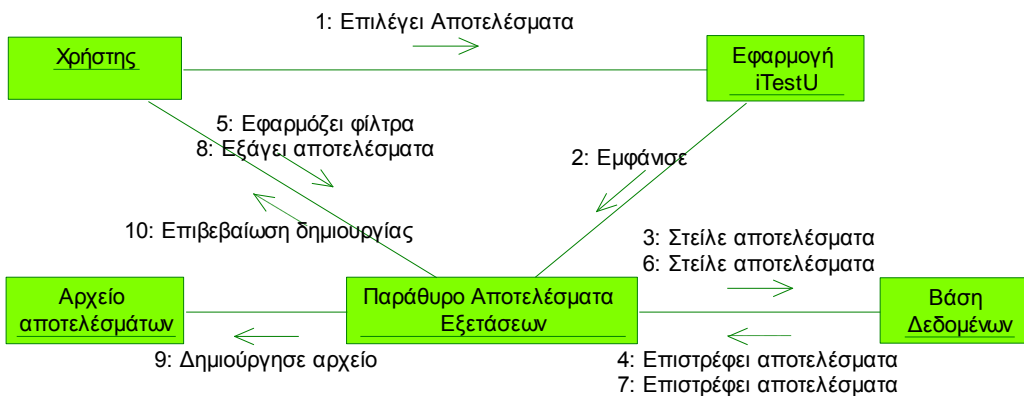
Εικόνα 8. Διάγραμμα συνεργασίας για υποβολή της απάντησης



Εικόνα 9. Διάγραμμα συνεργασίας για αγνόηση ερώτησης

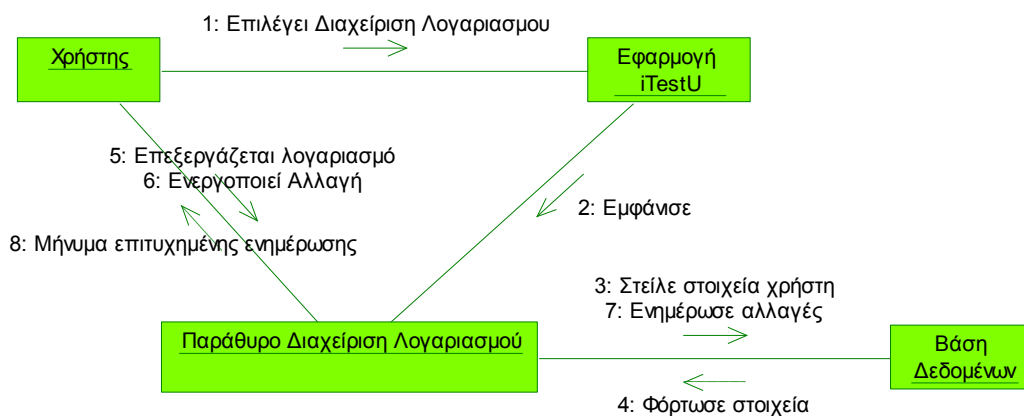


Εικόνα 10. Διάγραμμα συνεργασίας για αναίρεση ερώτησης



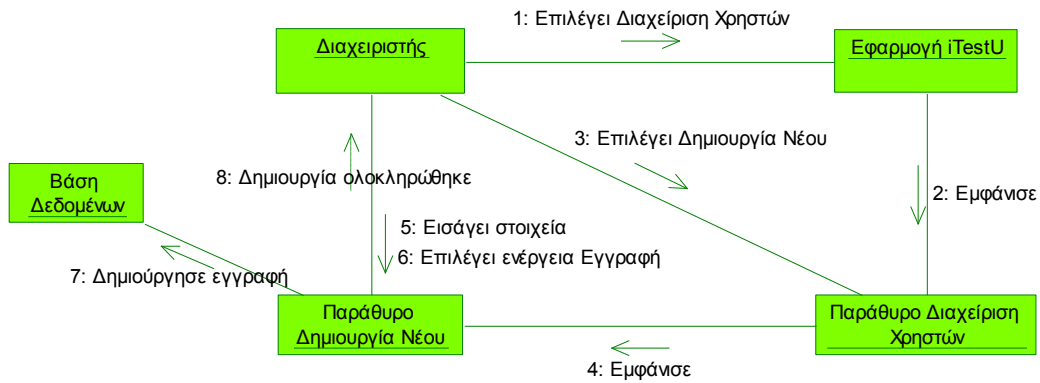
Εικόνα 11. Διάγραμμα συνεργασίας για την εμφάνιση αποτελεσμάτων εξέτασης

Στη συνέχεια με στο διάγραμμα συνεργασίας (Εικόνα 12) μπορούμε να δούμε την επικοινωνία των αντικειμένων κατά την λειτουργία διαχείρισης του λογαριασμού χρήστη, ενώ στις οι περιπτώσεις της δημιουργίας νέου χρήστη στην εφαρμογή και της αλλαγής των στοιχείων του παρουσιάζονται στα διαγράμματα (Εικόνα 13) και (Εικόνα 14) αντίστοιχα.

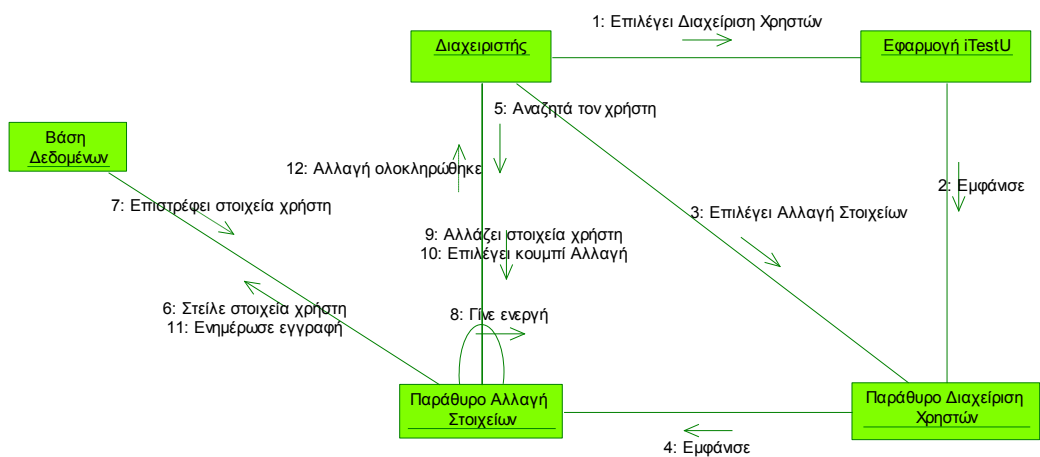


Εικόνα 12. Διάγραμμα συνεργασίας για διαχείριση λογαριασμού χρήστη

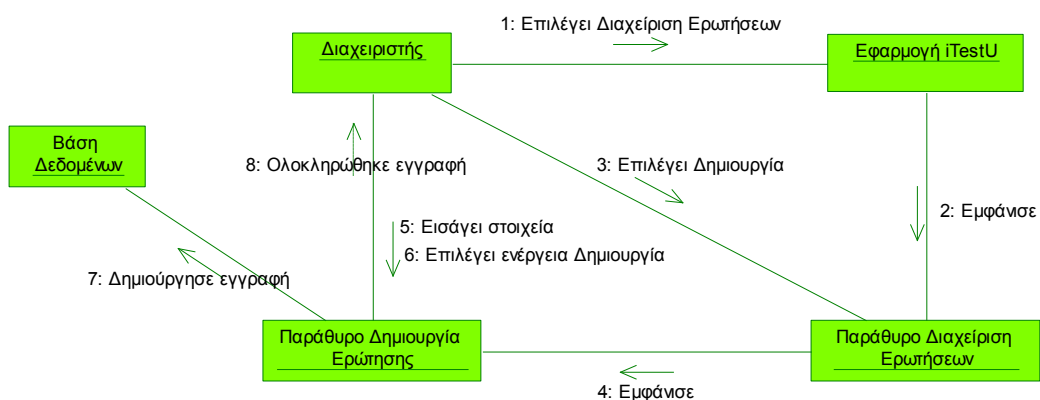
Στα διαγράμματα (Εικόνα 15) και (Εικόνα 16) παρουσιάζεται η αλληλεπίδραση των αντικειμένων κατά τη λειτουργία της δημιουργίας νέας ερώτησης από το διαχειριστή και κατά την επεξεργασία μιας υπάρχουσας ερώτησης αντίστοιχα.



Εικόνα 13. Διάγραμμα συνεργασίας για δημιουργία χρήστη



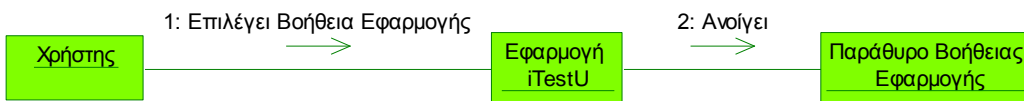
Εικόνα 14. Διάγραμμα συνεργασίας για αλλαγή στοιχείων χρήστη



Εικόνα 15. Διάγραμμα συνεργασίας για δημιουργία νέας ερώτησης



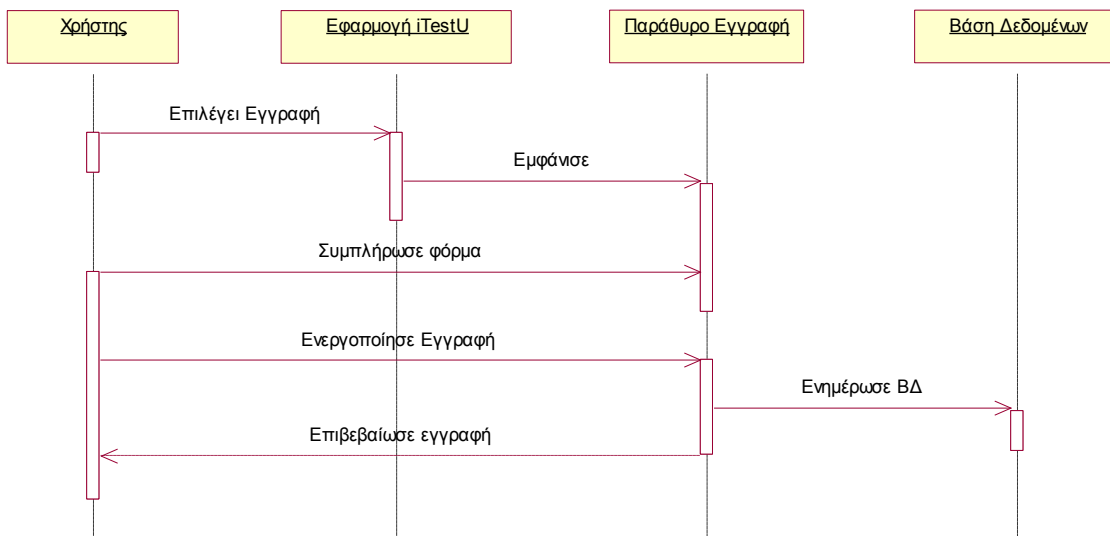
Εικόνα 16. Διάγραμμα συνεργασίας για επεξεργασία ερώτησης



Εικόνα 17. Διάγραμμα συνεργασίας για την βοήθεια της εφαρμογής

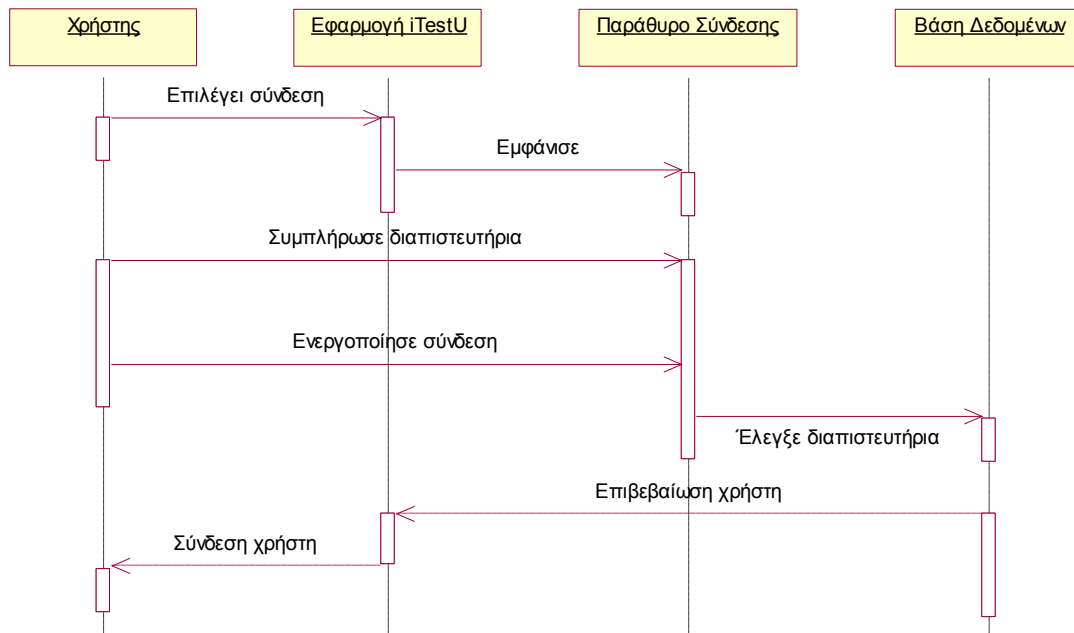
4.2.2 Διαγράμματα Ακολουθίας - Sequence Diagrams

Το διάγραμμα ακολουθίας ανήκει στην κατηγορία των διαγραμμάτων αλληλεπίδρασης (interaction diagrams), σκοπός του είναι να παρουσιάσει τον τρόπο που διαφορετικά αντικείμενα συνεργάζονται μεταξύ τους σε μια χρονική ακολουθία. Ένα τέτοιο διάγραμμα περιλαμβάνει: αντικείμενα, σχέσεις αντικειμένων, μηνύματα και τη διάρκεια ζωής κάθε αντικειμένου. Στη συνέχεια (Εικόνα 18), παραθέτουμε το διάγραμμα ακολουθίας που περιγράφει την εγγραφή ενός χρήστη στην εφαρμογή.

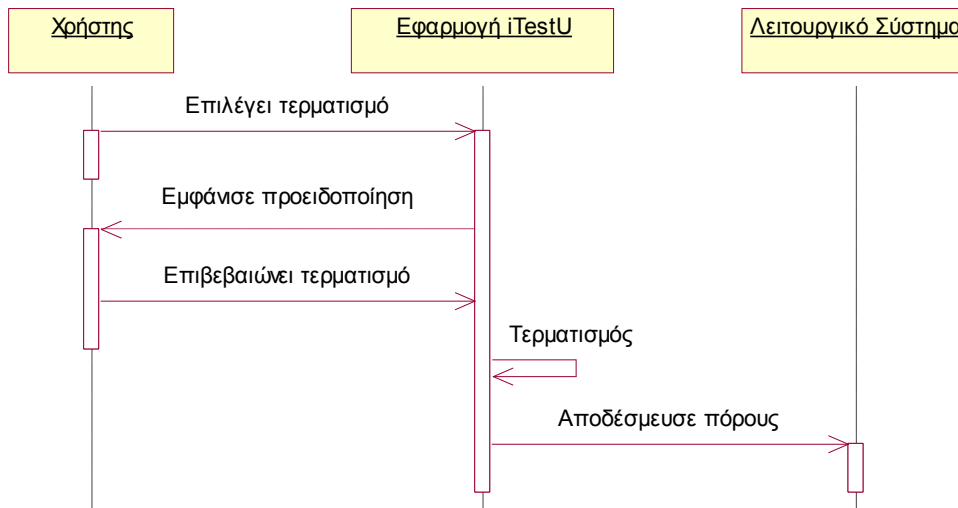


Εικόνα 18. Διάγραμμα ακολουθίας εγγραφής χρήστη

Στα επόμενα δύο διαγράμματα ακολουθίας στην Εικόνα 19 και Εικόνα 20 απεικονίζονται οι λειτουργίες της σύνδεσης και της αποσύνδεσης κάποιου χρήστη από την εφαρμογή. Στα διαγράμματα φαίνεται η αλληλεπίδραση του αντικειμένου *Χρήστης* με τα αντικείμενα *Εφαρμογή*, *Παράθυρο Εφαρμογής* και το αντικείμενο *Βάση Δεδομένων*. Η επικοινωνία μεταξύ των αντικειμένων γίνεται με ανταλλαγή κατάλληλων μηνυμάτων.



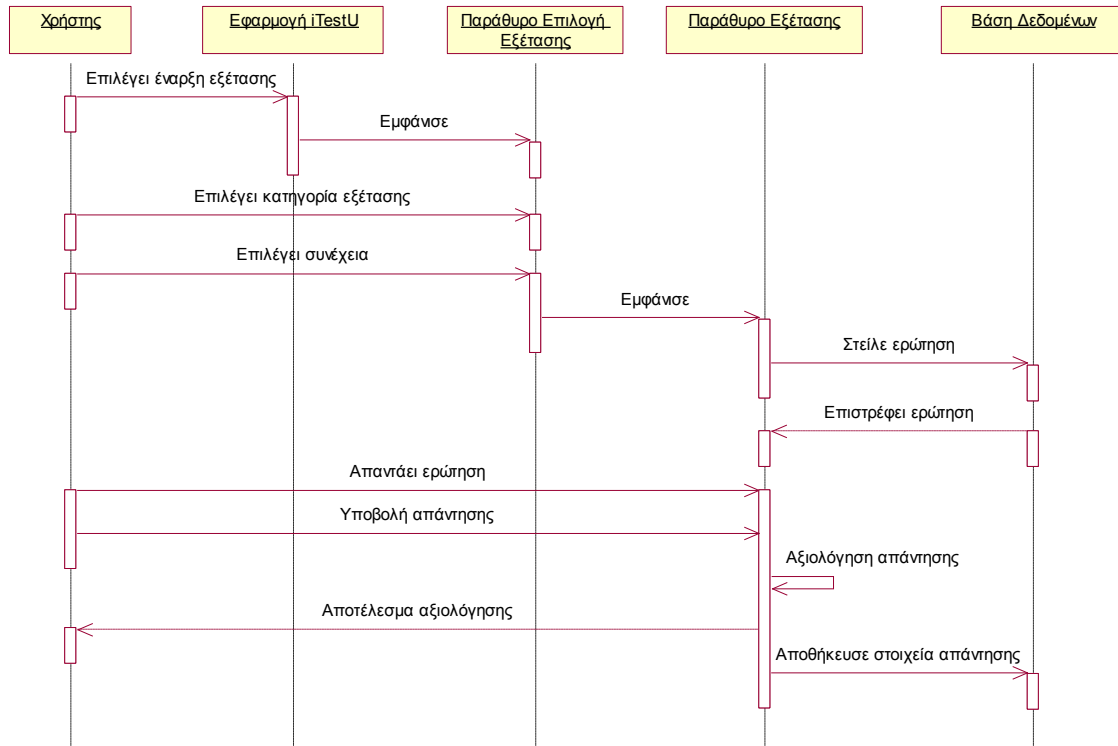
Εικόνα 19. Διάγραμμα ακολουθίας σύνδεσης χρήστη



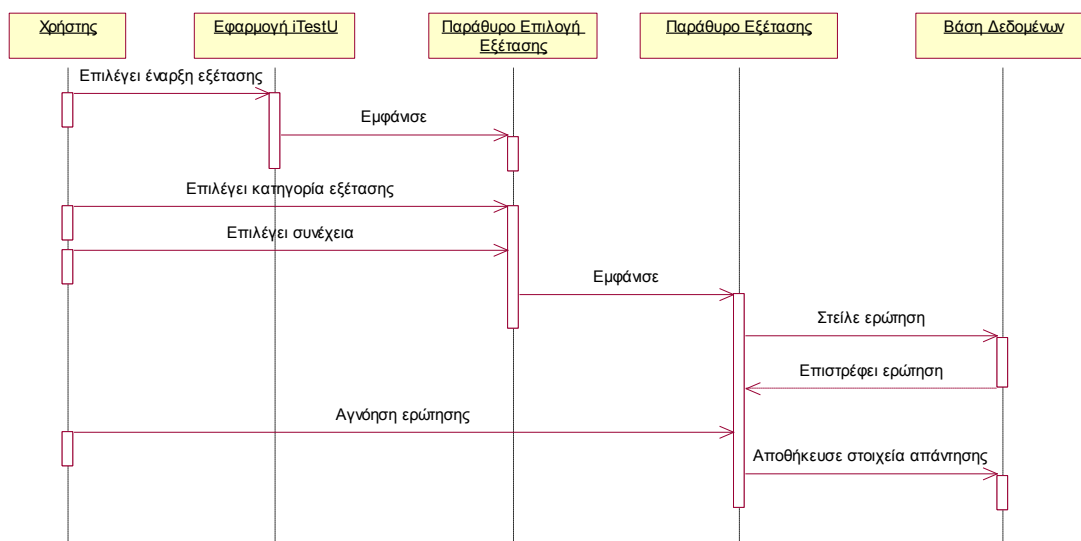
Εικόνα 20. Διάγραμμα ακολουθίας αποσύνδεση χρήστη

Στα επόμενα τρία διαγράμματα ακολουθίας παρουσιάζεται η λειτουργία της έναρξης και επιλογής κατηγορίας εξέτασης από τον χρήστη της εφαρμογής. Στα διαγράμματα φαίνεται η χρονική αλληλεπίδραση του αντικειμένου *Χρήστης* με τα αντικείμενα *Εφαρμογή iTestU*, τα αντικείμενα *Παράθυρο Εφαρμογής* και το αντικείμενο *Βάση Δεδομένων*. Παρουσιάζονται οι αλληλεπιδράσεις για την περίπτωση που ο χρήστης απαντάει στην ερώτηση και την υποβάλλει (Εικόνα 21), για την περίπτωση που επιλέγει να αγνοήσει την ερώτηση χωρίς να την απαντήσει (Εικόνα 22) και

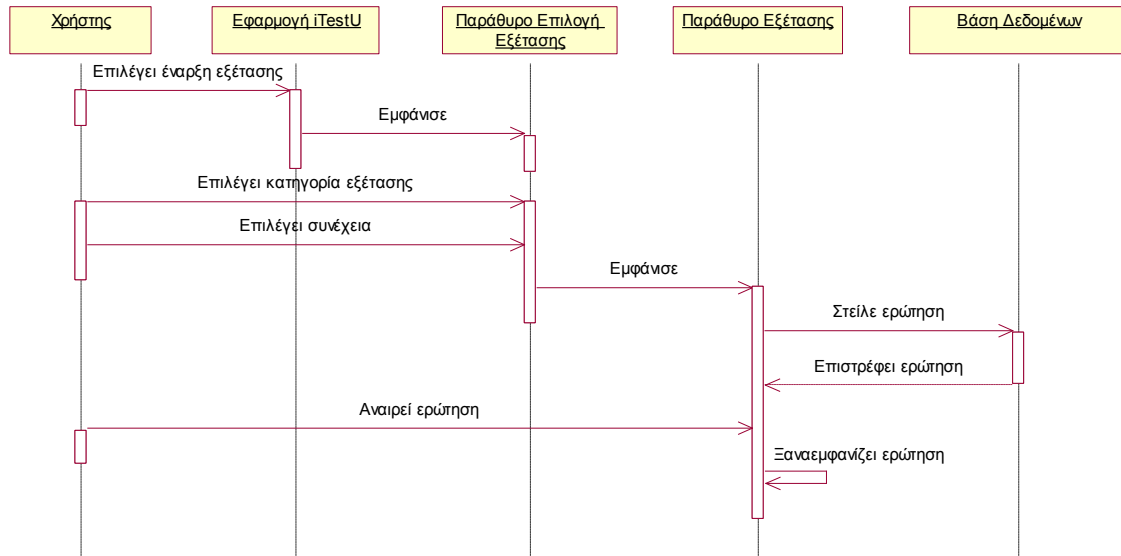
τέλος για την περίπτωση που αναιρεί την ερώτηση (Εικόνα 23). Στο διάγραμμα στην Εικόνα 24 παρουσιάζεται η αλληλεπίδραση των αντικειμένων κατά τη διαχείριση λογαριασμού του χρήστη, ενώ στην Εικόνα 25 εμφανίζεται η ανταλλαγή μηνυμάτων κατά τη χρήσης της βοήθειας της εφαρμογής.



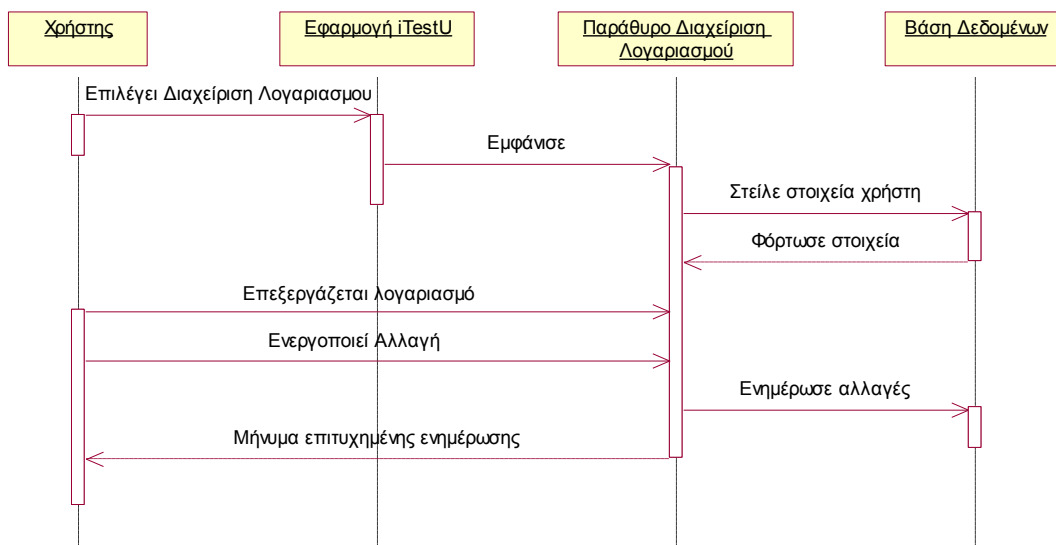
Εικόνα 21. Διάγραμμα ακολουθίας έναρξη εξέτασης και υποβολή απάντησης



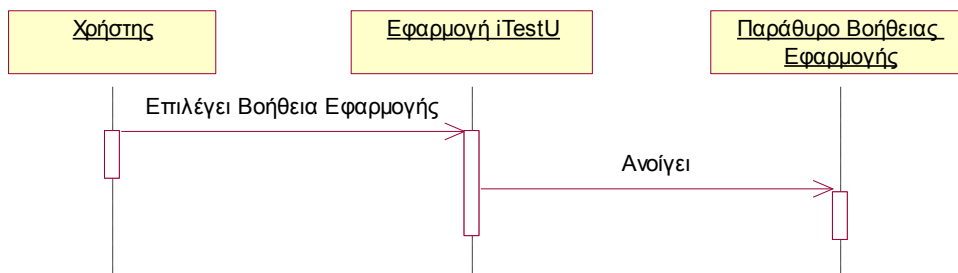
Εικόνα 22. Διάγραμμα ακολουθίας έναρξη εξέτασης και αγνόηση ερώτησης



Εικόνα 23. Διάγραμμα ακολουθίας έναρξη εξέτασης και αναίρεση ερώτησης

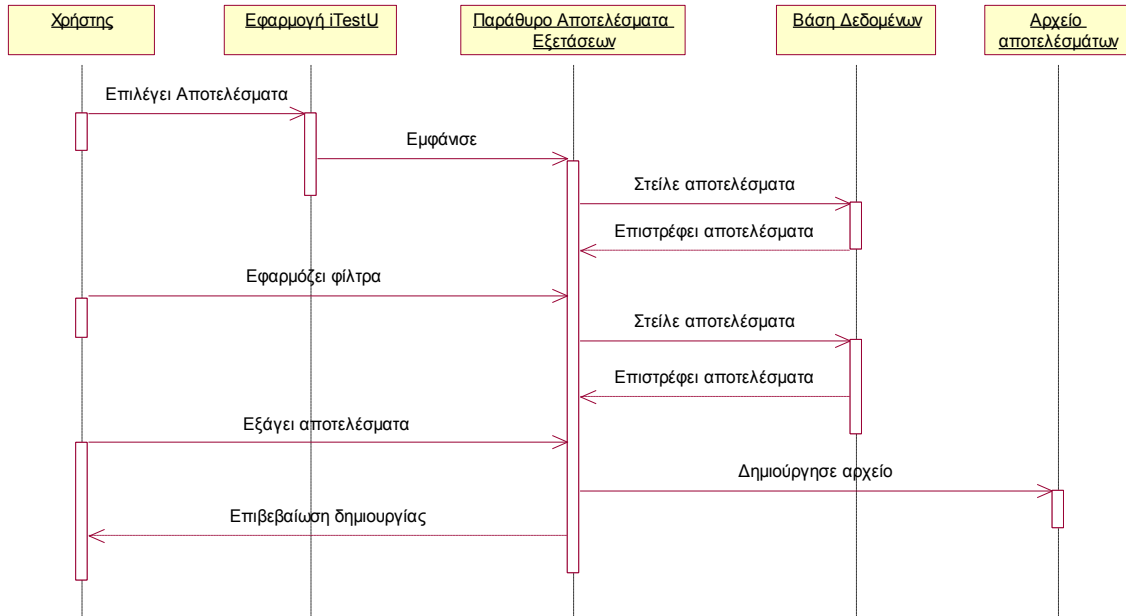


Εικόνα 24. Διάγραμμα ακολουθίας διαχείριση λογαριασμού χρήστη

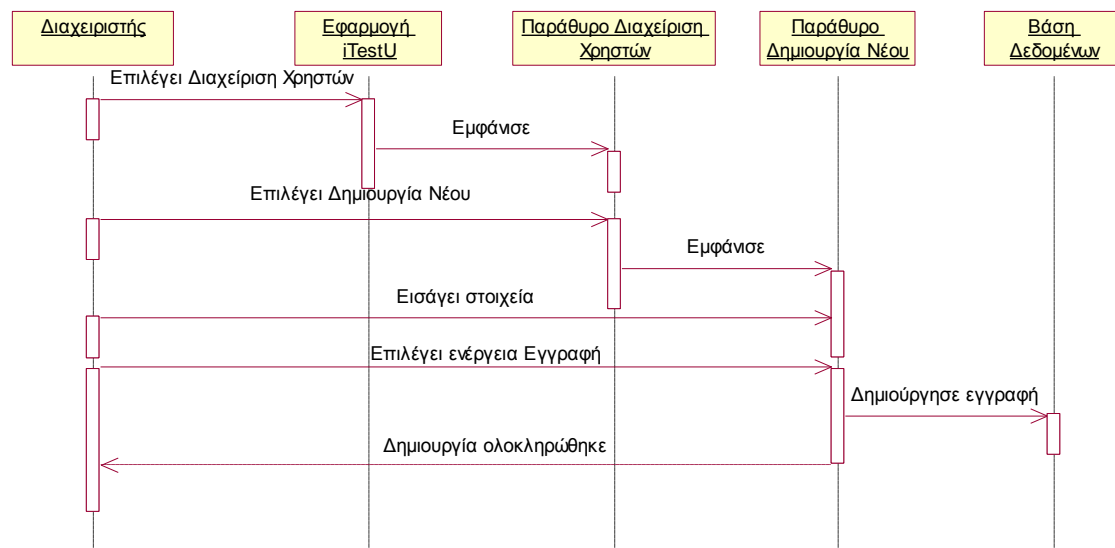


Εικόνα 25. Διάγραμμα ακολουθίας για την βοήθεια της εφαρμογής

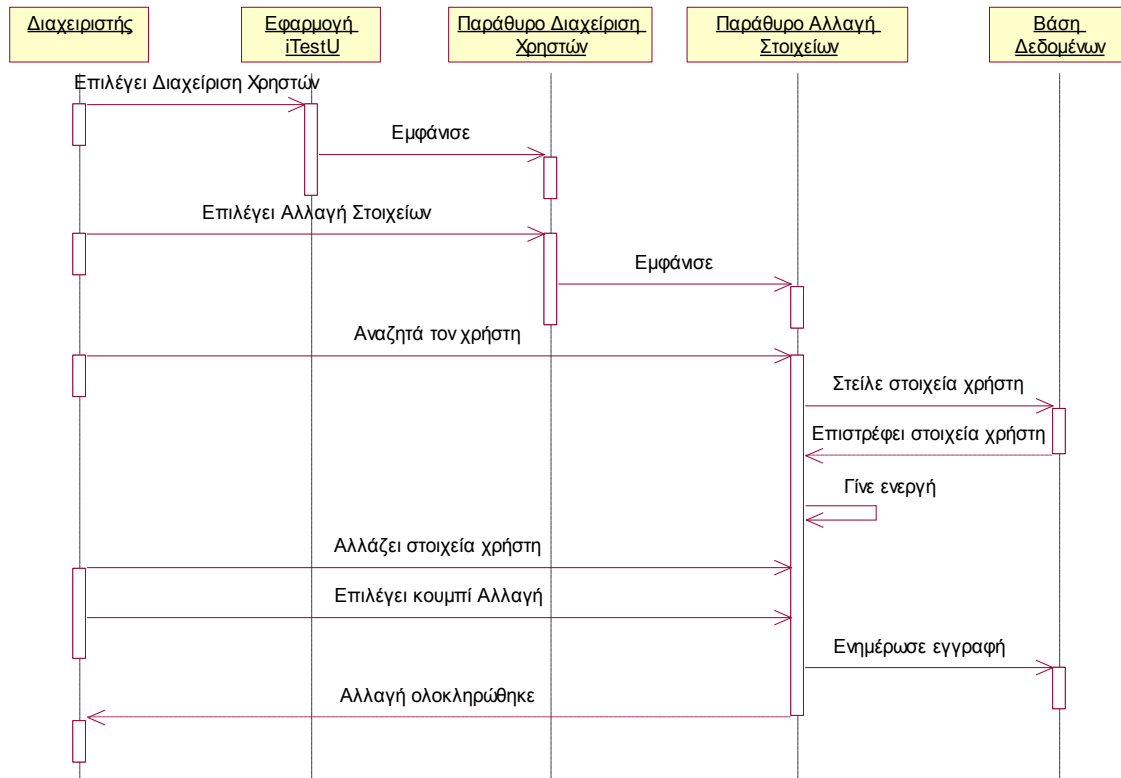
Στην Εικόνα 26 παρουσιάζονται οι αλληλεπιδράσεις των αντικειμένων για την περίπτωση που ο χρήστης επιλέγει τα αποτελέσματα της εξέτασης, να φιλτράρει τα αποτελέσματα εφαρμόζοντας κριτήρια και τέλος να εξάγει σε κάποιο αρχείο μια αναφορά. Στο διάγραμμα ακολουθίας στην Εικόνα 27 φαίνεται η ανταλλαγή μηνυμάτων ανάμεσα στο διαχειριστή, τα παράθυρα της εφαρμογής και τη ΒΔ κατά την διαδικασία δημιουργίας ενός νέου χρήστη, ενώ στην Εικόνα 28 κατά την επεξεργασία των στοιχείων του χρήστη εφαρμογή.



Εικόνα 26. Διάγραμμα ακολουθίας για την εμφάνιση αποτελεσμάτων εξέτασης

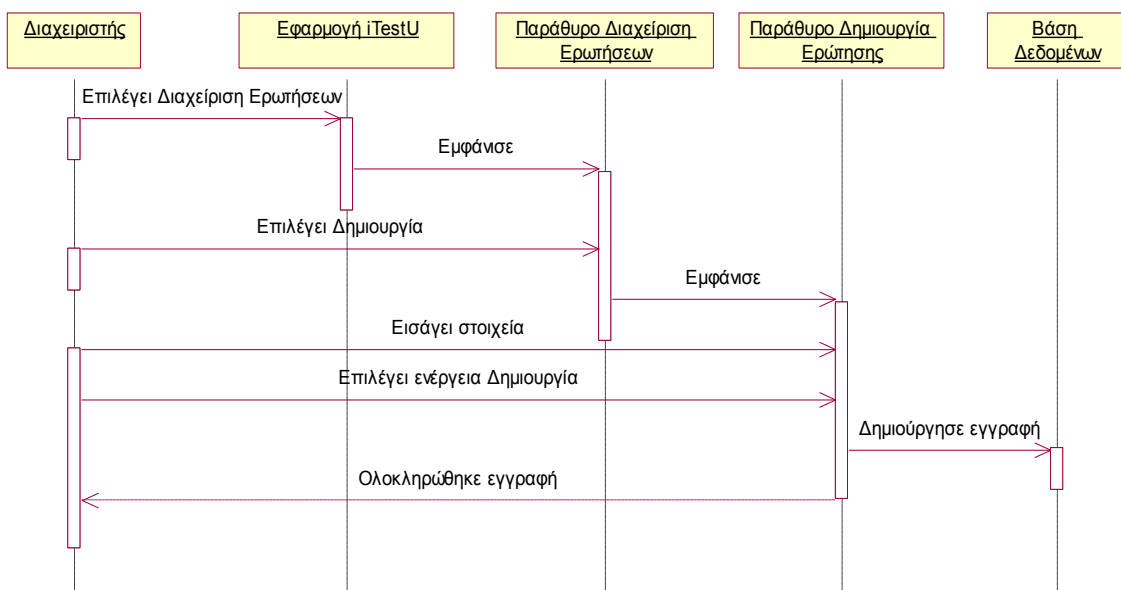


Εικόνα 27. Διάγραμμα ακολουθίας για δημιουργία χρήστη

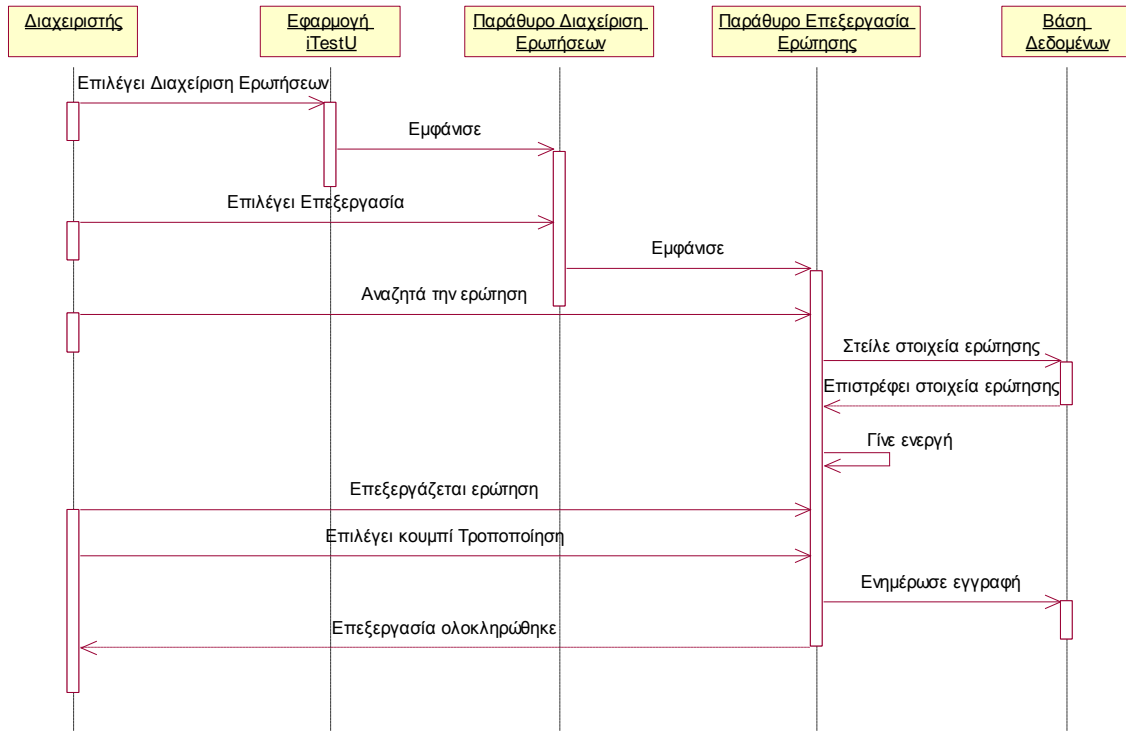


Εικόνα 28. Διάγραμμα ακολουθίας για αλλαγή στοιχείων χρήστη

Η Εικόνα 29 παρουσιάζει τις χρονικές αλληλεπιδράσεις των αντικειμένων κατά τη διαδικασία δημιουργίας μια νέας ερώτησης στο πληροφοριακό σύστημα, ενώ η Εικόνα 30 κατά την επεξεργασία μιας ερώτησης από τον διαχειριστή.



Εικόνα 29. Διάγραμμα ακολουθίας για δημιουργία νέας ερώτησης

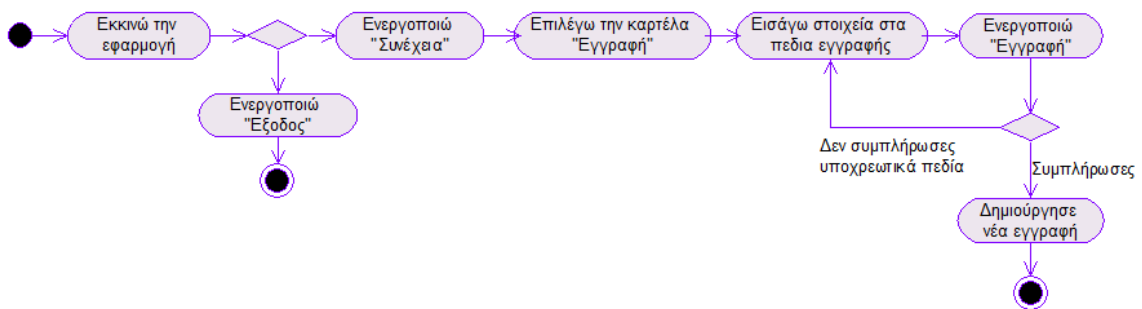


Εικόνα 30. Διάγραμμα ακολουθίας για επεξεργασία ερώτησης

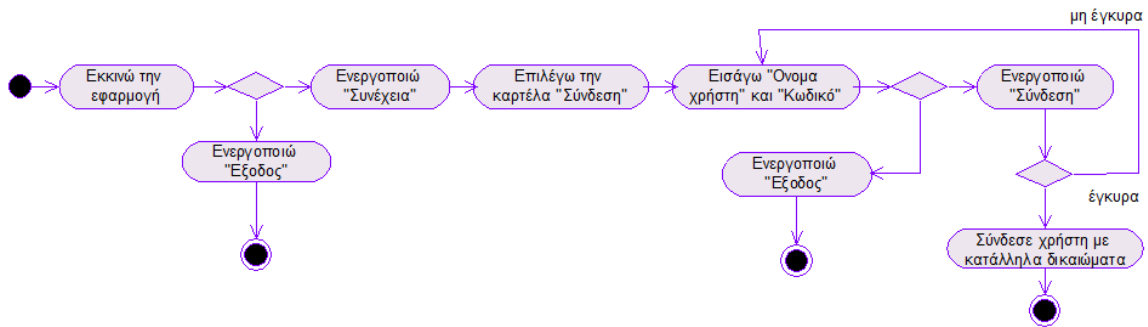
4.2.3 Διαγράμματα Δραστηριοτήτων - Activity Diagrams

Τα διαγράμματα δραστηριοτήτων αναπαριστούν τη συμπεριφορά μιας λειτουργίας ως σύνολο ενεργειών. Τα διαγράμματα αυτά χρησιμοποιούνται για την μοντελοποίηση των περιπτώσεων χρήσης. Πιο συγκεκριμένα μπορούν να απεικονίσουν τις ενέργειες ενός χαρακτήρα για μια περίπτωση χρήσης.

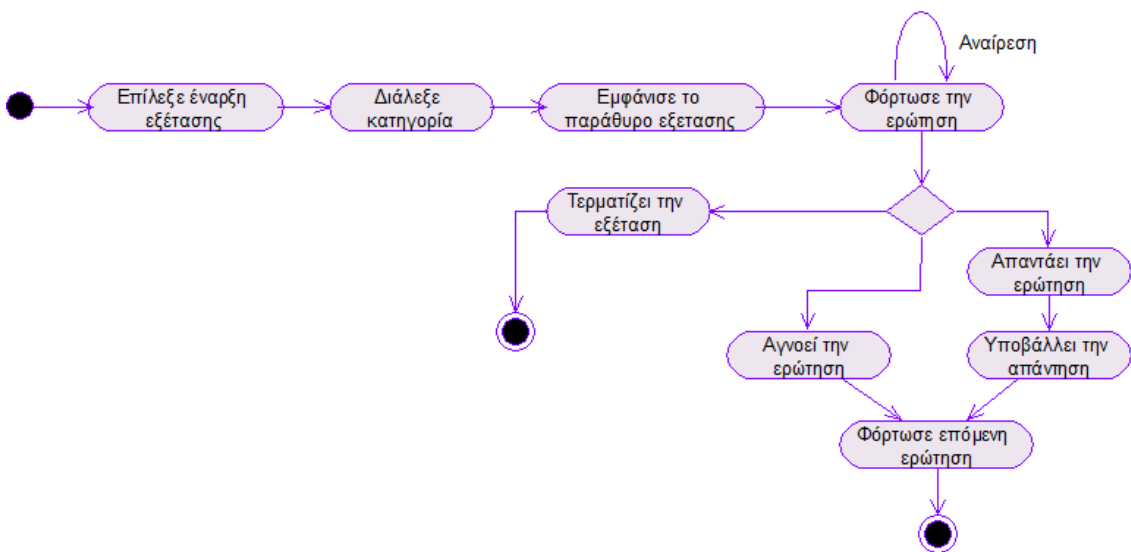
Στη συνέχεια παρουσιάζουμε τα διαγράμματα δραστηριοτήτων, για τις επιμέρους λειτουργίες της εφαρμογής, ξεκινώντας με την εγγραφή (Εικόνα 31) και τη σύνδεση του χρήστη (Εικόνα 32). Ακολουθούν τα διαγράμματα που σχετίζονται με τις λειτουργίες της εξεταστικής διαδικασίας και συγκεκριμένα με την εκκίνηση της εξέτασης (Εικόνα 33) και την εμφάνιση και έκδοση αποτελεσμάτων (Εικόνα 34), τη διαχείριση του προσωπικού λογαριασμού του χρήστη (Εικόνα 35). Ολοκληρώνουμε με τα διαγράμματα δραστηριοτήτων του διαχειριστή και συγκεκριμένα τη διαχείριση ερωτήσεων (Εικόνα 36) και τη διαχείριση χρηστών (Εικόνα 37).



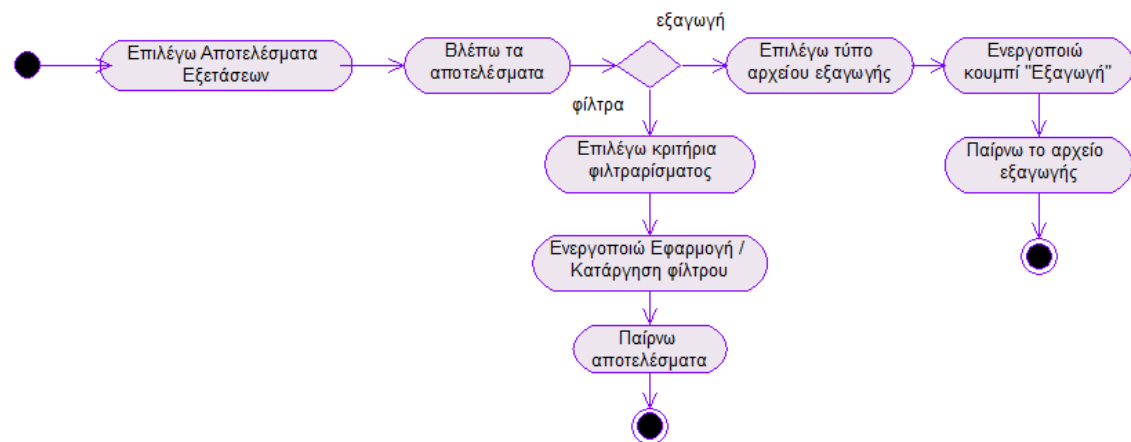
Εικόνα 31. Διάγραμμα δραστηριοτήτων για εγγραφή χρήστη



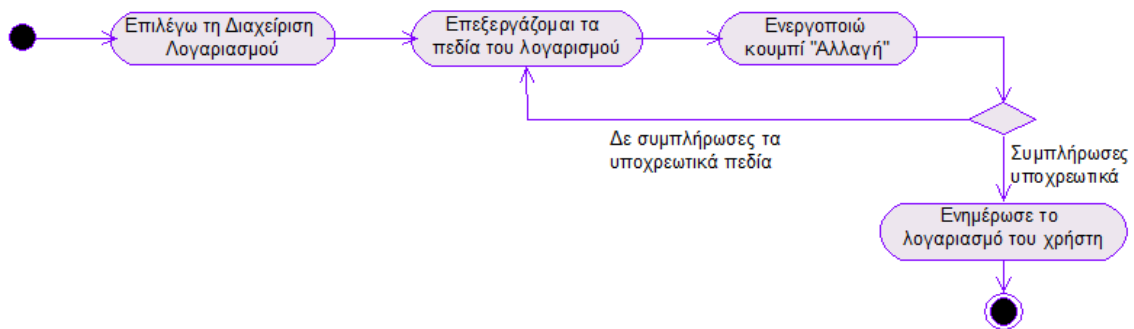
Εικόνα 32. Διάγραμμα δραστηριοτήτων για σύνδεση χρήστη



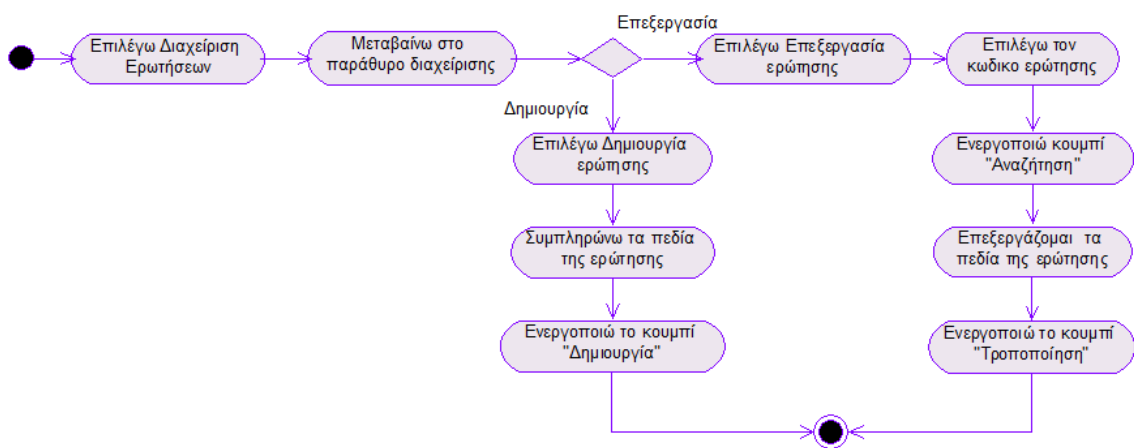
Εικόνα 33. Διάγραμμα δραστηριοτήτων για έναρξη εξέτασης



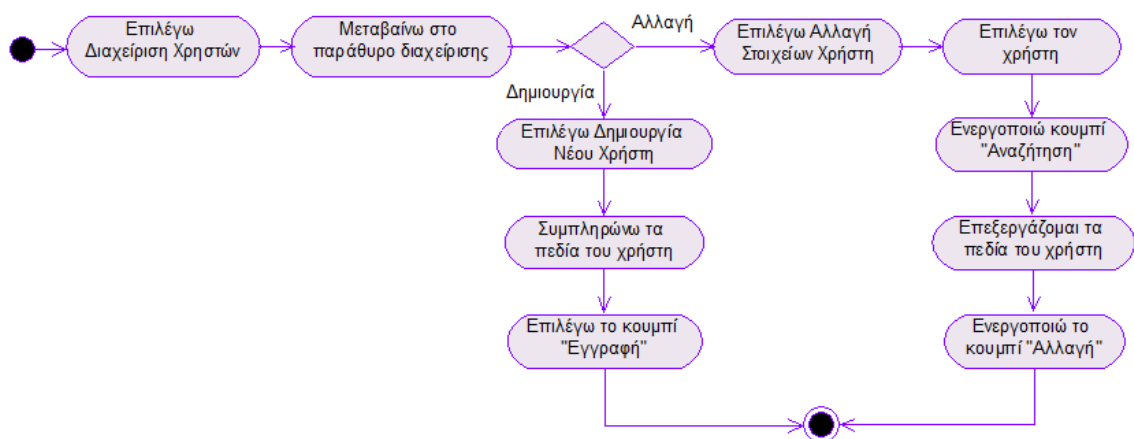
Εικόνα 34. Διάγραμμα δραστηριοτήτων για εμφάνιση αποτελεσμάτων εξετάσεων



Εικόνα 35. Διάγραμμα δραστηριοτήτων διαχείρισης λογαριασμού



Εικόνα 36. Διάγραμμα δραστηριοτήτων διαχείρισης ερωτήσεων



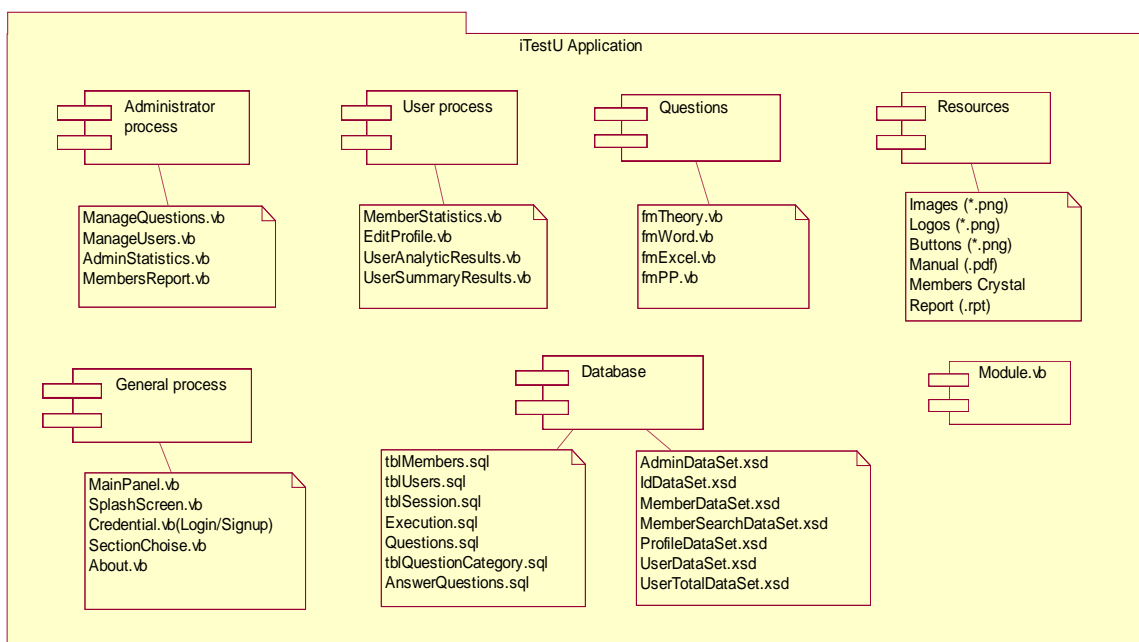
Εικόνα 37. Διάγραμμα δραστηριοτήτων διαχείρισης χρηστών

4.3 Υλοποίηση

Στην ενότητα αυτή αρχικά θα παρουσιάσουμε τα διαγράμματα υλοποίησης και στη συνέχεια τις βασικές αρχές προγραμματισμού που ακολουθήσαμε για να παράξουμε τον πηγαίο κώδικα της εφαρμογής. Τα διαγράμματα υλοποίησης παρουσιάζουν όψεις της υλοποίησης, συμπεριλαμβανομένης της δομής του κώδικα και της δομής υλοποίησης. Υπάρχουν δύο κατηγορίες διαγραμμάτων υλοποίησης, τα διαγράμματα εξαρτημάτων (component diagrams) και τα διαγράμματα διανομής (deployment diagrams) που παρουσιάζουν τη δομή του συστήματος. Ο κώδικας που αναπτύξαμε κατά το στάδιο υλοποίησης, παρουσιάζεται στο Παράρτημα της μεταπτυχιακής διατριβής και είναι εξ' ολοκλήρου διαθέσιμος στο συνοδευτικό CD.

4.3.1 Διαγράμματα Εξαρτημάτων - Component Diagram

Τα διαγράμματα εξαρτημάτων χρησιμοποιούνται για να περιγράψουν πως τα εξαρτήματα-συστατικά χρησιμοποιούνται για τη δημιουργία ενός συστήματος. Ως εξάρτημα (component) θεωρούμε κάποιο αρχείο κώδικα, μια ΒΔ και άλλους πόρους όπως για παράδειγμα εικόνες που χρησιμοποιήθηκαν.



Εικόνα 38. Διάγραμμα εξαρτημάτων εφαρμογής iTestU

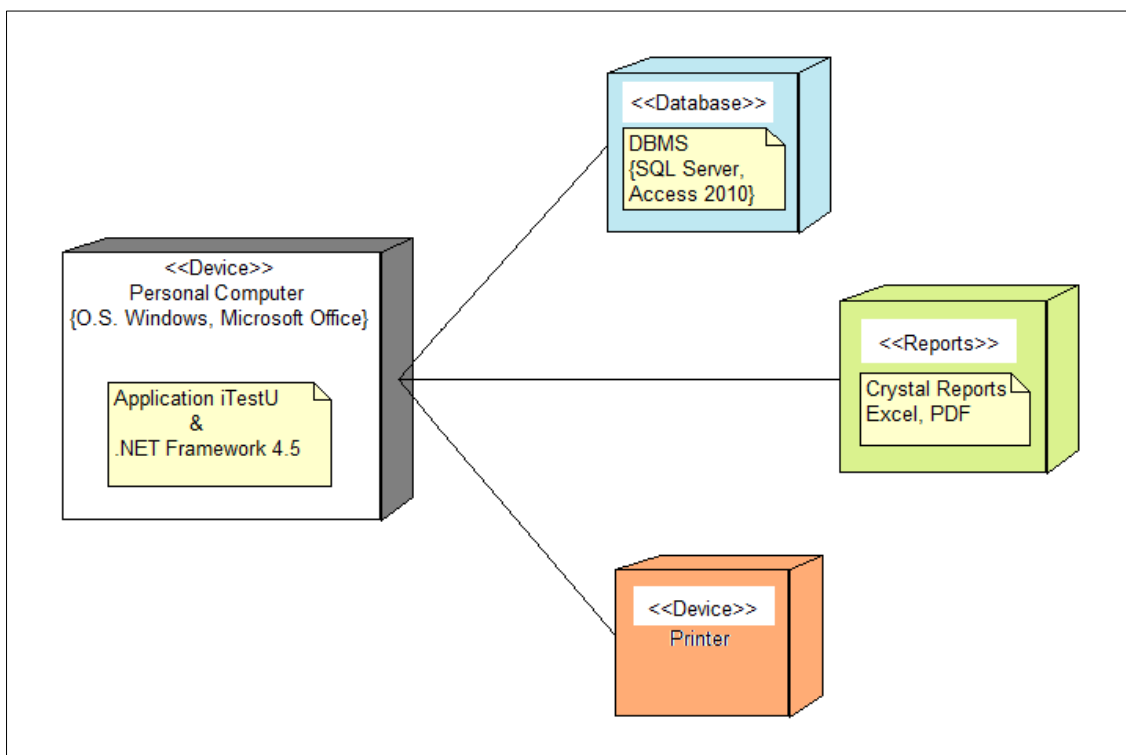
4.3.2 Διαγράμματα Διανομής - Deployment Diagram

Τα διαγράμματα διανομής αναπαριστούν τη φυσική διάταξη των διαφόρων εξαρτημάτων υλικού, που απαρτίζουν το σύστημα καθώς και τη διανομή των εκτελέσιμων προγραμμάτων πάνω σε αυτό το υλικό. Κάθε πόρος του υλικού αναπαρίσταται, στο διάγραμμα αυτό, με έναν κόμβο. Οι κόμβοι του συστήματος διασυνδέονται μεταξύ τους με γραμμές που δηλώνουν τα κανάλια επικοινωνίας, που μπορεί να είναι μονής ή διπλής κατεύθυνσης.

Για το εκπαιδευτικό πληροφοριακό σύστημα που υλοποιήσαμε το διάγραμμα διανομής παρουσιάζεται στην Εικόνα 39. Το διάγραμμα διανομής αποτελείται από τον βασικό κόμβο που εκτελείται η εφαρμογή iTestU. Ο κόμβος αυτός είναι ένας προσωπικός υπολογιστής (PC) με εγκατεστημένο λειτουργικό σύστημα Windows. Στον υπολογιστή αυτό πρέπει να είναι εγκατεστημένη η σουίτα του Office για να μπορούν να εκτελεστούν οι εξετάσεις στα αντικείμενα

της επεξεργασίας κειμένου, των υπολογιστικών φύλλων και των παρουσιάσεων. Τέλος πρέπει να υπάρχει εγκατεστημένο το .NET Framework 4.5 και φυσικά η εφαρμογή iTestU.

Οι υπόλοιποι κόμβοι του διαγράμματος αναπαριστούν τη ΒΔ της εφαρμογής που μπορεί να βρίσκεται στον ίδιο προσωπικό υπολογιστή που βρίσκεται εγκατεστημένη και η εφαρμογή iTestU ή σε διαφορετικό υπολογιστή που επικοινωνεί με τον κεντρικό υπολογιστή μέσω κατάλληλης υποδομής δικτύου. Το ΣΔΒΔ μπορεί να είναι η εφαρμογή Access 2010 ή στην περίπτωση που αυξηθούν οι απαιτήσεις του πληροφοριακού συστήματος ο SQL Server. Ο επόμενος κόμβος που παρουσιάζεται στο διάγραμμα είναι ένας εκτυπωτής, για να εκτυπώνουμε τις αναφορές αποτελεσμάτων εξετάσεων ή άλλα στοιχεία των χρηστών της εφαρμογής. Ο εκτυπωτής μπορεί να είναι άμεσα συνδεδεμένος στον κεντρικό υπολογιστή ή να είναι εκτυπωτής δικτύου. Ο τελευταίος κόμβος αναπαριστάνει τα αρχεία που μπορούμε να εξαγάγουμε μέσα από την εφαρμογή. Τα αρχεία εξαγωγής που είναι αρχεία αναλυτικών και συγκεντρωτικών αποτελεσμάτων εξετάσεων, ιστορικό εξετάσεων και στοιχεία των χρηστών μπορούμε να τα πάρουμε σε μορφή Excel και σε μορφή PDF.



Εικόνα 39. Διάγραμμα διανομής συστήματος

4.3.3 Ανάπτυξη του κώδικα

Στην ενότητα αυτή θα αναφέρουμε μερικά από τα χαρακτηριστικά του πηγαίου κώδικα που είναι επιθυμητά για να εξασφαλίσουμε ποιότητα στο λογισμικό που αναπτύσσουμε. Κατά τη φάση της κωδικοποίησης δεν αρκεί μόνο να παραχθεί ο πηγαίος κώδικας που έτσι και αλλιώς είναι το αυτονόητο, αλλά πρέπει να λάβουμε σοβαρά μερικούς παράγοντες ακόμα, ώστε ο κώδικας που θα παραχθεί να διαθέτει τα παρακάτω χαρακτηριστικά. Στη φάση της υλοποίησης - κωδικοποίησης της εφαρμογής iTestU, πρώτο μας μέλημα ήταν να γράψουμε κώδικα που θα δούλευε σωστά σύμφωνα με τις προδιαγραφές της ανάλυσης και της σχεδίασης αλλά ταυτόχρονα θα ικανοποιήσουμε και τα περισσότερα από τα χαρακτηριστικά που αναφέρονται στη συνέχεια.

Επάρκεια

Αποτελεί αναμφίβολα το πιο βασικό χαρακτηριστικό του πηγαίου κώδικα. Ο κώδικας μπορεί να χαρακτηριστεί επαρκής όταν το υλοποιημένο πληροφοριακό σύστημα λειτουργεί σωστά, χωρίς σφάλματα, βάσει των προδιαγραφών που έχουν καθοριστεί, κάνει σωστή χρήση των διαθέσιμων πόρων του περιβάλλοντος μέσα στο οποίο εκτελείται και είναι διαθέσιμο όποτε απαιτηθεί. Η επάρκεια του πηγαίου κώδικα διασφαλίζεται κατά τη ελέγχου του λογισμικού που κατά το αντικειμενοστραφές μοντέλο RUP γίνεται τόσο στη φάση εκπόνησης της μελέτης (elaboration), κυρίως στη φάση της κατασκευής (construction) όσο και κατά το τελευταίο στάδιο της μετάβασης (transition)

Επιδόσεις

Στα περισσότερα πληροφοριακά συστήματα η επίδοση/ταχύτητα (performance) παίζει σημαντικό ρόλο. Όταν αναφερόμαστε στην επίδοση ενός συστήματος στην πραγματικότητα εννοούμε την ταχύτητα εκτέλεσης των επιμέρους λειτουργιών και τις απαιτήσεις σε πόρους όπως υπολογιστική ισχύ και μνήμη. Η επιλογή της κατάλληλης γλώσσας προγραμματισμού, η χρήση των κατάλληλων προγραμματιστικών τεχνικών και η χρήση βέλτιστων αλγορίθμων αποτελούν καθοριστικούς παράγοντες στην βελτίωση των επιδόσεων του τελικού προϊόντος.

Αναγνωσιμότητα

Με τον όρο αναγνωσιμότητα (readability) εννοούμε την ευκολία κατανόησης του πηγαίου κώδικα μέσω της ανάγνωσής του. Υπάρχουν κάποιοι πρακτικοί κανόνες που βοηθάνε στην αύξηση της αναγνωσιμότητας του κώδικα όπως:

- Η απλότητα
- Η επιλογή περιγραφικών ονομάτων
- Η χρήση σχολίων
- Η σωστή στοίχιση

Με τον όρο απλότητα εννοούμε ότι οι προγραμματιστές πρέπει να αποφεύγουν την περιττή πολυπλοκότητα κατά τη συγγραφή του πηγαίου κώδικα. Πάντα πρέπει να έχουν ως στόχο την αναζήτηση απλών και κομψών λύσεων κωδικοποίησης. Επιπλέον, ο απλούστερος κώδικας εκτός ότι επιτυγχάνει το μέγιστο βαθμό αναγνωσιμότητας, είναι και λιγότερο επιρρεπής σε προγραμματιστικά λάθη.

Η επιλογή των κατάλληλων ονομάτων, για τα αναγνωριστικά του προγράμματος, κατά τη φάση συγγραφής κώδικα είναι εξαιρετικής σημασίας. Η επιλογή περιγραφικών ονομάτων για τις μεταβλητές, τις σταθερές, τις συναρτήσεις και τις μεθόδους κάνει τον κώδικα ευανάγνωστο τόσο για τον ίδιο τον προγραμματιστή όσο και για τους μελλοντικούς ανθρώπους που θα συντηρήσουν ή θα επεκτείνουν την εφαρμογή.

Η χρήση σχολίων πρέπει να αποτελεί αναπόσπαστο τμήμα του πηγαίου κώδικα. Τα σχόλια πρέπει να γράφονται τη στιγμή που παράγεται και ο πηγαίος κώδικας και να ενημερώνονται σε κάθε μελλοντική αλλαγή του. Τις περισσότερες φορές η έλλειψη επαρκούς σχολιασμού δυσχεραίνει ή καθιστά αδύνατη την κατανόηση ενός προγράμματος. Δεν είναι λίγες οι φορές που δυσκολευόμαστε να κατανοήσουμε κώδικα που έχουμε δημιουργήσει εμείς οι ίδιοι και τον διαβάζουμε μετά από καιρό. Τα σχόλια είναι καλό να γράφονται τόσο στην αρχή των τμημάτων πηγαίου κώδικα όσο και ενδιάμεσα όπου αυτό κρίνεται απαραίτητο.

Η στοίχιση του κώδικα είναι μια παράμετρος που βοηθάει πολύ της αύξησης της αναγνωσιμότητας του πηγαίου κώδικα. Η μορφή της στοίχισης πολλές φορές είναι υποκειμενική υπόθεση, αλλά στις περιπτώσεις που υπάρχει ομάδα έργου που έχει αναλάβει την κωδικοποίηση πρέπει να αποφασίζεται από κοινού. Τα σύγχρονα περιβάλλοντα ανάπτυξης λογισμικού στοιχίζουν τις περισσότερες φορές τον πηγαίο κώδικα αυτόματα οπότε ο προγραμματιστής δεν χρειάζεται να ανησυχεί γι' αυτό.

Επαρκής Τεκμηρίωση

Είναι συχνά αναγκαίο να τεκμηριώνεται ο πηγαίος κώδικας, με την βοήθεια συνοδευτικών εγγράφων γραμμένων σε κατάλληλη γλώσσα. Συνηθίζεται η τεκμηρίωση (documentation) του πηγαίου κώδικα να γίνεται με φυσική γλώσσα που αποσκοπεί να επεξηγήσει τα πολύπλοκα σημεία όταν αυτό δεν είναι εφικτό μόνο με τον σχολιασμό. Σε άλλες περιπτώσεις μαζί με την φυσική γλώσσα υπάρχουν και συνοδευτικά διαγράμματα.

Μεταφερσιμότητα

Με τον όρο αυτό μεταφερσιμότητα (portable) εννοούμε την ιδιότητα του πηγαίου κώδικα να μπορεί να εκτελεστεί σε διαφορετικά περιβάλλοντα όπως είναι εκδόσεις λειτουργικών συστημάτων ή τύπο υπολογιστή, από αυτά που αρχικά σχεδιάστηκε. Για να το επιτύχει αυτό ο προγραμματιστής πρέπει να προσέξει κατά την φάση της κωδικοποίησης να μη χρησιμοποιήσει συστατικά της γλώσσας που υποστηρίζονται μόνο από συγκεκριμένες υλοποιήσεις.

Επαναχρησιμοποιησιμότητα

Με την δυνατότητα επαναχρησιμοποίησης (reusability) εννοούμε ότι τα διάφορα μέρη του προγράμματος πρέπει να σχεδιάζονται έτσι ώστε να μπορούν να χρησιμοποιούνται και σε άλλα προγράμματα. Μέρη του λογισμικού που ικανοποιούν την αρχή της επαναχρησιμοποιησιμότητας είναι διάφοροι αλγόριθμοι (ρουτίνες και βιβλιοθήκες) και δομές δεδομένων που χρησιμοποιούμε στα προγράμματα.

4.4 Έλεγχος της εφαρμογής

Στη φάση των δοκιμών γίνονται οι απαραίτητοι έλεγχοι για να διαπιστωθεί κατά πόσο το λογισμικό που έχει παραχθεί βρίσκεται μέσα στις προδιαγραφές που είχαμε καθορίσει σε προηγούμενα στάδια του κύκλου ζωής λογισμικού. Έλεγχοι πραγματοποιήθηκαν σε όλη τη φάση της κατασκευής κατά το στάδιο συγγραφής του κώδικα. Συγκεκριμένα το στάδιο του ελέγχου εκτελούνταν παράλληλα με την κωδικοποίηση της εφαρμογής. Κάθε τμήμα κώδικα που ολοκληρώνονταν ελέγχονταν μεμονωμένα για σφάλματα. Αφού επιβεβαιώναμε τη σωστή λειτουργία του, ενσωματώνονταν στο υπόλοιπο σύστημα και ακολουθούσε νέα δοκιμή (testing). Για τον έλεγχο της εφαρμογής χρησιμοποιήσαμε τα εργαλεία εκσφαλμάτωσης (debugging tools) που μας παρείχε το περιβάλλον ανάπτυξης Visual Studio 2012.

5 Η Βάση Δεδομένων

Σε αυτό το κεφάλαιο θα αναλύσουμε τη Βάση Δεδομένων (ΒΔ) που αναπτύξαμε για να υποστηρίξουμε τα δεδομένα της εφαρμογής. Θα ξεκινήσουμε το κεφάλαιο με μια αναφορά σε βασικές έννοιες ΒΔ και στη συνέχεια θα περιγράψουμε αναλυτικά τη δομή της ΒΔ, του πίνακες και τα ερωτήματα που χρησιμοποιήθηκαν για να υποστηρίξουν την εφαρμογή iTestU.

5.1 Εισαγωγή

Στην ενότητα αυτή κάνουμε μια σύντομη περιγραφή για το τι καλείται βάση δεδομένων, τι είναι σχεσιακή βάση δεδομένων, τι είναι τα συστήματα διαχείρισης βάσεων δεδομένων και τι είναι πίνακας.

5.1.1 Τι είναι μια Βάση Δεδομένων

Μια Βάση Δεδομένων είναι μια οργανωμένη και δομημένη συλλογή από αποθηκευμένα δεδομένα τα οποία σχετίζονται με ένα συγκεκριμένο θέμα. Για παράδειγμα έχουμε τη ΒΔ για μια βιβλιοθήκη για να οργανώσουμε βιβλία και δανειζόμενους, τη ΒΔ μιας εμπορικής επιχείρησης για να οργανώσουμε τα προϊόντα, τους πελάτες και τους προμηθευτές ή τη ΒΔ για μια εκπαιδευτική εφαρμογή όπως το iTestU για να οργανώσουμε ερωτήσεις, εξεταζόμενους και βαθμολογίες.

Στις μέρες μας η πλειονότητα των πληροφοριακών συστημάτων που αναπτύσσονται είτε πρόκειται για διαδικτυακές εφαρμογές είτε για αυτόνομες εφαρμογές ή ακόμα και ιστοσελίδες ή εφαρμογές κινητών τηλεφώνων στηρίζονται σε κάποια ΒΔ για να αποθηκεύσουν τα δεδομένα που χρειάζονται πριν από τη φάση της επεξεργασίας, αλλά και μετά όταν θα τα παρουσιάσουν. Αυτό που καθιστά μια ΒΔ μοναδική και τόσο απαραίτητη σε μια εφαρμογή είναι ότι μας παρέχει εύκολη πρόσβαση και διαχείριση των δεδομένων μας. Ο σκοπός μιας ΒΔ δεν είναι μόνο η αποθήκευση των πληροφοριών αλλά η αποθήκευση με τέτοιο τρόπο ώστε να είναι γρήγορη η ανάκτησή τους, όταν τα χρειαστούμε.

Οι ΒΔ δημιουργούνται, οργανώνονται και συντηρούνται από κατάλληλα προγράμματα που ονομάζονται Συστήματα Διαχείρισης Βάσεων Δεδομένων - ΣΔΒΔ (Data Base Management Systems - DBMS) και συγκαταλέγονται στις πολυπλοκότερες εφαρμογές. Τέτοια συστήματα - εφαρμογές είναι το SQL Server, η Microsoft Access, το MySQL ή Oracle και πολλά ακόμα. Ένα θεμελιώδες χαρακτηριστικό ενός ΣΔΒΔ είναι ότι απομονώνει μέρος της πολυπλοκότητας της ΒΔ από τον προγραμματιστή. Αυτό σημαίνει ότι ανεξάρτητα από τον τρόπο με τον οποίο κάθε ΣΔΒΔ αποθηκεύει τα δεδομένα, βλέπουμε τα δεδομένα οργανωμένα σε πίνακες, με σχέσεις μεταξύ των πινάκων.

Για να προσπελάσουμε ή να ενημερώσουμε τα δεδομένα που είναι αποθηκευμένα στη ΒΔ χρησιμοποιούμε μια ειδική γλώσσα, τη Γλώσσα Δομημένων Ερωτημάτων (Structured Query Language - SQL). Σε αντίθεση με άλλες περιοχές προγραμματισμού, η SQL είναι μια πραγματικά ενιαία γλώσσα που όλα τα βασικά ΣΔΒΔ την υποστηρίζουν.

5.1.2 Σχεσιακές Βάσεις Δεδομένων

Το μοντέλο αναπαράστασης δεδομένων που χρησιμοποιείται συχνότερα για την περιγραφή μιας ΒΔ είναι το σχεσιακό. Το μοντέλο αυτό αφορά τόσο τη λογική οργάνωση και συσχέτιση των δεδομένων, όσο και τον τρόπο αποθήκευσης και διαχείρισης των πληροφοριών. Οι ΒΔ που είναι σχεδιασμένες σύμφωνα με το μοντέλο αυτό ονομάζονται σχεσιακές ΒΔ (Relational DB) επειδή βασίζονται στις σχέσεις μεταξύ των δεδομένων που περιέχουν.

Από τη στιγμή που θα καθορίσουμε τις οντότητες σε ένα σύστημα μπορούμε να προχωρήσουμε στη δημιουργία των πινάκων. Τα δεδομένα που αποθηκεύονται στους πίνακες σχετίζονται μεταξύ ως προς την οντότητα που ανήκουν. Ένας πίνακας είναι μια ορθογώνια, γραμμική διάταξη στοιχείων σε οριζόντια και κατακόρυφη μορφή, δηλαδή σε γραμμές και

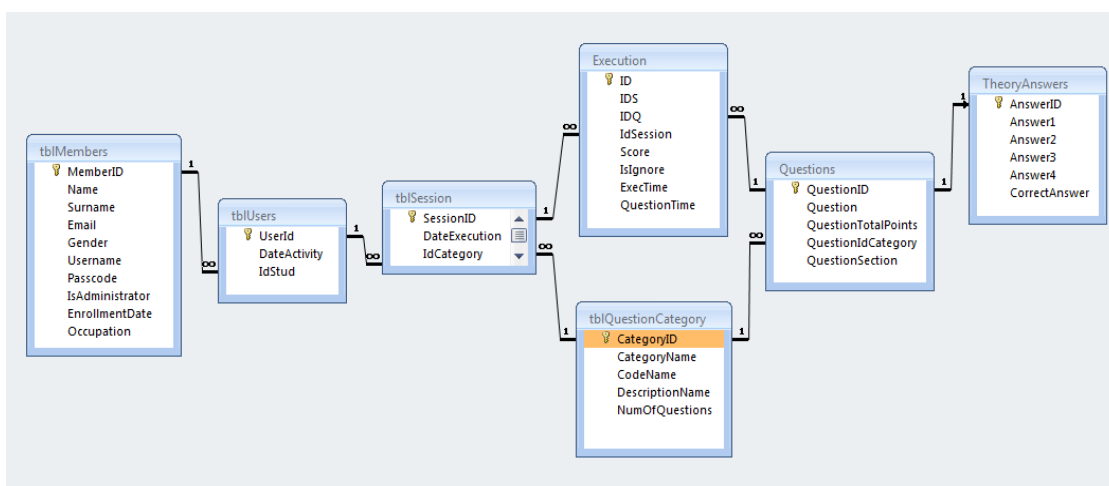
στήλης. Κάθε στήλη αντιστοιχεί σε κάποια ιδιότητα ή χαρακτηριστικό της οντότητας που αναπαριστά και ονομάζεται πεδίο (field). Κάθε γραμμή του πίνακα είναι ένα σύνολο διαφορετικών πεδίων που περιέχει όλες τις πληροφορίες για ένα στιγμιότυπο της οντότητας και ονομάζεται πλειάδα (tuple) ή εγγραφή.

Ο λόγος για την διάσπαση των πληροφοριών, που θέλουμε να αποθηκεύσουμε σε μια ΒΔ, σε ξεχωριστούς πίνακες είναι για να αποφύγουμε την επανάληψη των πληροφοριών. Αυτό αποτελεί βασική σχεδιαστική αρχή στη σχεδίαση μιας ΒΔ και ήταν κάτι που λάβαμε σοβαρά υπόψη στη ΒΔ της εφαρμογής μας. Η επανάληψη των πληροφοριών πρέπει να αποφεύγεται γιατί αργά ή γρήγορα θα οδηγήσει σε ανακολουθίες μέσα στη ΒΔ. Η διαδικασία διαχωρισμού των δεδομένων σε σχετικούς πίνακες που εξαλείφουν όλες τις ενδεχόμενες μορφές αναπαραγωγής πληροφοριών ονομάζεται κανονικοποίηση. Για την κανονικοποίηση μιας ΒΔ υπάρχουν συγκεκριμένοι κανόνες που πρέπει να ακολουθηθούν, τους κανόνες αυτούς ακολουθήσαμε στη φάση ανάλυσης και σχεδιασμού της ΒΔ.

5.2 Διερεύνηση των πινάκων της ΒΔ

Για την υλοποίηση του εκπαιδευτικού πληροφοριακού συστήματος iTestU σχεδιάσαμε και υλοποιήσαμε μια ΒΔ που αποτελείται από 7 πίνακες συνολικά. Οι πίνακες αυτοί είναι οι tblMembers, tblUsers, tblSession, Execution, Questions, tblQuestionCategory και TheoryAnswers. Η υλοποίηση της ΒΔ έγινε στο σύστημα διαχείρισης ΒΔ Access 2010. Να σημειώσουμε εδώ ότι πραγματοποιήθηκε και μια δεύτερη υλοποίηση της ΒΔ σε SQL Server 2012. Ο λόγος της δεύτερης δοκιμαστικής υλοποίησης είναι για μια πιθανή μετεγκατάσταση σε περιβάλλον SQL Server, σε μια επόμενη έκδοση της εφαρμογής, στην περίπτωση που υπάρξουν αυξημένες ανάγκες αποθήκευσης δεδομένων. Η μετεγκατάσταση έγινε μέσα από το περιβάλλον της Access 2010, εναλλακτικά θα μπορούσαμε να χρησιμοποιήσουμε κάποιο βοηθητικό εργαλείο όπως το Microsoft SQL Server Migration Assistant (SSMA) for Access.

Στις επόμενες ενότητες παρουσιάζουμε μια σύντομη περιγραφή για τους πίνακες της ΒΔ και τον SQL κώδικα που χρησιμοποιήθηκε για την υλοποίηση του κάθε πίνακα. Στην Εικόνα 40 παρουσιάζεται το διάγραμμα σχέσεων πινάκων της ΒΔ όπως έχει υλοποιηθεί για την εφαρμογή μέσα από το περιβάλλον της Access 2010.



Εικόνα 40. Η ΒΔ της εφαρμογής

5.2.1 Ο πίνακας tblMembers

Στον πίνακα tblMembers αποθηκεύονται τα στοιχεία για όλα τα μέλη που έχουν δημιουργήσει και διατηρούν λογαριασμό στην εφαρμογή. Οι χρήστες αυτοί μπορεί να είναι οι απλοί χρήστες -

εξεταζόμενοι που έχουν περιορισμένα δικαιώματα χρήσης στην εφαρμογή ή οι διαχειριστές της εφαρμογής με πλήρη δικαιώματα χρήσης. Ο πίνακας για να αποθηκεύσει τα στοιχεία των μελών χρησιμοποιεί τα πεδία MemberID, Name, Surname, Email, Gender, Username, Passcode, IsAdministrator, EnrollmentDate και το πεδίο Occupation.

Το πεδίο MemberID αποτελεί το αναγνωριστικό πεδίο για κάθε εγγραφή του πίνακα και επειδή δέχεται μοναδικές τιμές αποτελεί και το πρωτεύον κλειδί του πίνακα. Το πεδίο αυτό δεν μπορεί να είναι κενό και έχει οριστεί με τύπο δεδομένων ακέραιος μεγάλου μεγέθους. Οι τιμές του πεδίου ενημερώνονται αυτόματα και βηματικά (αυτόματη αρίθμηση). Τα πεδία Name και Surname αποθηκεύουν το όνομα και το επίθετο της οντότητας μέλος. Ο τύπος δεδομένων των δύο πεδίων είναι κείμενο και έχουν ρυθμιστεί να δέχονται μέχρι 30 αλφαριθμητικούς χαρακτήρες. Τα πεδία αυτά δε μπορεί να είναι κενά κατά τη δημιουργία μιας νέας εγγραφής. Το πεδίο Email κρατάει τη διεύθυνση ηλεκτρονικού ταχυδρομείου του μέλους και δεν είναι υποχρεωτικό πεδίο. Στο πεδίο Gender αποθηκεύουμε το φύλο του μέλους που επιλέγεται μεταξύ 2 διακριτών τιμών (άρρεν ή θήλυ) γι' αυτό και περιορίσαμε το μέγεθος του πεδίου μέχρι 5 χαρακτήρες, ενώ μπορεί να παραμείνει κενό. Στα πεδία Username και Passcode αποθηκεύονται πληροφορίες που σχετίζονται με το όνομα χρήστη και τον κωδικό πρόσβασης που διατηρεί το μέλος στην εφαρμογή. Είναι και τα δύο υποχρεωτικά πεδία με μέγεθος μέχρι 20 χαρακτήρες το καθένα. Το πεδίο IsAdministrator κρατάει πληροφορία που σχετίζεται με το ρόλο του μέλους στην εφαρμογή. Είναι τύπου boolean, δηλαδή παίρνει τιμές true ή false και δεν είναι υποχρεωτικό. Η προεπιλεγμένη τιμή του είναι false που σημαίνει ότι το μέλος διατηρεί δικαιώματα απλού χρήστη. Η τιμή του πεδίου δεν μπορεί να αλλάξει από τον ίδιο το χρήστη παρά μόνο από το διαχειριστή του συστήματος. Το πεδίο EnrollmentDate διατηρεί την ημερομηνία και την ώρα εγγραφής του μέλους στην εφαρμογή. Κάθε φορά που δημιουργείται μια νέα εγγραφή μέλους το πεδίο αυτό ενημερώνεται αυτόματα σύμφωνα με το ρολόι του συστήματος. Δεν υπάρχει η δυνατότητα το πεδίο αυτό να αλλάξει τιμή μέσα από την εφαρμογή. Το πεδίο Occupation κρατάει πληροφορία που έχει να κάνει με την ιδιότητα του μέλους ή την ενασχόληση του στην προσωπική του ζωή. Δέχεται αλφαριθμητικούς χαρακτήρες με μέγεθος μέχρι 50 και δεν είναι υποχρεωτικό. Το μέλος μπορεί να εισάγει πληροφορία στο πεδίο αυτό τη στιγμή που δημιουργεί το λογαριασμό του ή αργότερα μέσα από την διαχείριση του προσωπικού του λογαριασμού. Το πεδίο αυτό μέσα στην εφαρμογή εμφανίζεται ως αναπτυσσόμενη λίστα (combobox) όπου το μέλος επιλέγει μια από τις επιλογές ή πληκτρολογεί μια νέα τιμή.

```
/*SQL script για τη δημιουργία του πίνακα tblMembers */
CREATE TABLE [dbo].[tblMembers](
    [MemberID] [int] NOT NULL,
    [Name] [nvarchar](30) NOT NULL,
    [Surname] [nvarchar](30) NOT NULL,
    [Email] [nvarchar](255) NULL,
    [Gender] [nvarchar](5) NULL,
    [Username] [nvarchar](20) NOT NULL,
    [Passcode] [nvarchar](20) NOT NULL,
    [IsAdministrator] [bit] NOT NULL,
    [EnrollmentDate] [datetime] NOT NULL,
    [Occupation] [nvarchar](50) NULL,
    PRIMARY KEY CLUSTERED ([MemberID] ASC));
```

5.2.2 Ο πίνακας tblUsers

Ο πίνακας tblUsers χρησιμοποιείται για να κρατάει πληροφορίες σχετικά με τους χρήστες που συνδέονται στην εφαρμογή. Κάθε φορά που ένας απλός χρήστης ή διαχειριστής της εφαρμογής συνδέεται, δημιουργείται μια εγγραφή στον πίνακα με την ημερομηνία και ώρα σύνδεσης. Ο πίνακας αυτός είναι διασυνδεδεμένος με τον πίνακα tblMembers άρα γνωρίζουμε με αυτό τον τρόπο τα πλήρη στοιχεία του μέλους που συνδέθηκε.

Ο πίνακας χρησιμοποιεί τρία πεδία για να αποθηκεύσει την πληροφορία του. Το πεδίο UserId αποτελεί το αναγνωριστικό πεδίο για κάθε εγγραφή του πίνακα και είναι το πρωτεύον

κλειδί. Το πεδίο αυτό δεν μπορεί να είναι κενό και έχει οριστεί με τύπο δεδομένων ακέραιος μεγάλο μεγέθους. Οι τιμές του πεδίου ενημερώνονται αυτόματα. Το δεύτερο πεδίο DateActivity, διατηρεί την ημερομηνία και την ώρα σύνδεσης των χρηστών στην εφαρμογή. Κάθε φορά που πραγματοποιείται μια νέα σύνδεση μέλους το πεδίο αυτό ενημερώνεται αυτόματα σύμφωνα με το ρολόι του συστήματος. Η τιμή του πεδίου δεν μπορεί να αλλάξει χειροκίνητα μέσα από την εφαρμογή ούτε από τον απλό χρήστη ούτε από τον διαχειριστή. Το τελευταίο πεδίο IdStud είναι το ξένο κλειδί του πίνακα tblMembers και έχει δηλωθεί με τύπο δεδομένων ακέραιο αριθμό. Ακολουθεί ο SQL κώδικας που περιγράφει τη δομή του πίνακα tblUsers.

```
/*SQL script για τη δημιουργία του πίνακα tblUsers */
CREATE TABLE [dbo].[tblUsers](
    [UserId] [int] NOT NULL,
    [DateActivity] [datetime] NULL,
    [IdStud] [int] NULL,
    PRIMARY KEY CLUSTERED ([UserId] ASC),
    FOREIGN KEY ([IdStud]) REFERENCES [dbo].[tblMembers] ([MemberID]));
```

5.2.3 Ο πίνακας tblSession

Ο πίνακας tblSession χρησιμοποιείται για να αποθηκεύει τα δεδομένα που σχετίζονται με την έναρξη κάθε νέας εξέτασης που εκτελείται. Κάθε φορά που ένας απλός χρήστης ή διαχειριστής της εφαρμογής συνδέεται και ξεκινάει μια νέα εξέταση δημιουργείται στον πίνακα tblSession μια εγγραφή με την ημερομηνία και ώρα έναρξης της εξέτασης. Ο πίνακας αυτός είναι διασυνδεδεμένος με τον πίνακα tblUsers κατά συνέπεια γνωρίζουμε τα πλήρη στοιχεία του χρήστη που εκκίνησε την εξέταση.

Ο πίνακας διαθέτει τέσσερα πεδία τα οποία είναι το SessionID, το DateExecution, το IdCategory και το IDS. Το πεδίο SessionID είναι το πρωτεύον κλειδί του πίνακα και έχει οριστεί ως ακέραιος αριθμός με αυτόματη αρίθμηση. Οι τιμές που λαμβάνει είναι μοναδικές και δεν επιτρέπεται να είναι κενές. Το πεδίο DateExecution, διατηρεί την ημερομηνία και την ώρα έναρξης μιας νέας συνόδου, δηλαδή μιας έναρξης εξέτασης από κάποιο διασυνδεδεμένο χρήστη. Το πεδίο έχει δηλωθεί με τύπο δεδομένων ημερομηνία/ώρα και δέχεται μη κενές τιμές. Η τιμή του δεν μπορεί να αλλάξει χειροκίνητα μέσα από την εφαρμογή. Στο πεδίο IdCategory αποθηκεύεται ο κωδικός του αντικειμένου της εξέτασης ως ακέραιος αριθμός. Για παράδειγμα ο κωδικός 1 αντιστοιχεί σε εξετάσεις θεωρίας, το 2 σε εξετάσεις επεξεργασίας κειμένου κ.ο.κ. Το τελευταίο πεδίο του πίνακα είναι το IDS αποτελεί το ξένο κλειδί του πίνακα tblUsers και έχει δηλωθεί με τύπο δεδομένων ακέραιο αριθμό. Παρακάτω παρουσιάζουμε τον SQL κώδικα που χρησιμοποιήσαμε για τη δημιουργία του πίνακα.

```
/* SQL script για τη δημιουργία του πίνακα tblSession*/
CREATE TABLE [dbo].[tblSession](
    [SessionID] [int] NOT NULL,
    [DateExecution] [datetime] NOT NULL,
    [IdCategory] [int] NOT NULL,
    [IDS] [int] NOT NULL,
    PRIMARY KEY CLUSTERED ([SessionID] ASC),
    FOREIGN KEY ([IDS]) REFERENCES [dbo].[tblUsers] ([UserId]));
```

5.2.4 Ο πίνακας Execution

Ο πίνακας αυτός είναι ο σημαντικότερος πίνακας της ΒΔ. Σε αυτόν καταγράφονται δεδομένα που σχετίζονται με το χρήστη που εξετάζεται, τον κωδικό της εξέτασης που τρέχει, τον αριθμό της ερώτησης που απαντάει, το βαθμό που παίρνει σε κάθε απάντηση και το χρόνο εκτέλεσης κάθε ερώτησης. Αποτελείται από οκτώ πεδία και διασυνδέεται με τους πίνακες tblSession και Questions.

Αναλυτικότερα το πεδίο ID είναι το πρωτεύον κλειδί του πίνακα και έχει οριστεί ως ακέραιος αριθμός με αυτόματη αρίθμηση. Το πεδίο IDQ είναι ο μοναδικός αναγνωριστικός κωδικός κάθε ερώτησης και εδώ αποτελεί το ξένο κλειδί στη διασύνδεση με τον πίνακα Questions και το πεδίο IdSession είναι το ξένο κλειδί στη διασύνδεση με τον πίνακα tblSession που περιγράψαμε προηγουμένως. Στο πεδίο Score αποθηκεύεται ο βαθμός που λαμβάνει ο εξεταζόμενος σε κάθε ερώτηση που απαντάει. Ο βαθμός για κάθε ερώτηση μπορεί να πάρει τις τιμές 0 εφόσον δεν απαντηθεί σωστά η ερώτηση και 1 εφόσον απαντηθεί σωστά. Για την περίπτωση κάποιων ερωτήσεων με 2 υποερωτήματα, η σωστή απάντηση λαμβάνει 2 βαθμούς και η μερική απάντηση λαμβάνει 1 βαθμό. Η ερώτηση που αγνοείται λαμβάνει 0 βαθμούς. Με βάση τα παραπάνω επιλέξαμε τον τύπο δεδομένων του πεδίου να είναι ακέραιος αριθμός μη κενός. Το πεδίο IsIgnore σχεδιάστηκε για να καταγράφουμε μια ερώτηση ως αγνοημένη. Είναι τύπου boolean και κατά συνέπεια δέχεται τις διακριτές τιμές true/false. Αν ο χρήστης αγνοήσει μια ερώτηση κατά την εκτέλεση της εξέτασης το πεδίο αυτό παίρνει για την τρέχουσα εγγραφή της ερώτησης την τιμή true. Στο πεδίο ExecTime καταγράφουμε το χρόνο εκτέλεσης μιας ερώτησης με τη μορφή ακέραιου αριθμού ενώ στο πεδίο QuestionTime ο χρόνος αποθηκεύεται με τη μορφή αλφαριθμητικού της μορφής mm:ss όπου m είναι λεπτά και s είναι δευτερόλεπτα.

```
/* SQL script για τη δημιουργία του πίνακα Execution*/
CREATE TABLE [dbo].[Execution](
    [ID] [int] NOT NULL,
    [IDS] [int] NOT NULL,
    [IDQ] [int] NOT NULL,
    [IdSession] [int] NOT NULL,
    [Score] [int] NOT NULL,
    [IsIgnore] [bit] NOT NULL,
    [ExecTime] [int] NOT NULL,
    [QuestionTime] [nvarchar](10) NOT NULL,
PRIMARY KEY CLUSTERED ([ID] ASC),
FOREIGN KEY([IDQ])REFERENCES [dbo].[Questions] ([QuestionID]),
FOREIGN KEY([IdSession])REFERENCES [dbo].[tblSession] ([SessionID]));
```

5.2.5 Ο πίνακας tblQuestionCategory

Ο πίνακας tblQuestionCategory χρησιμοποιείται για να κρατάει δεδομένα σχετικά με τις κατηγορίες των ερωτήσεων. Ο πίνακας αποτελείται από έξι πεδία και διασυνδέεται με σχέση «ένα προς πολλά» με τον πίνακα tblSession και τον πίνακα Questions.

Το πεδίο CategoryID αποτελεί το πρωτεύον κλειδί του πίνακα, είναι τύπου ακέραιος και δέχεται έναν αριθμό που αντιστοιχεί στην κατηγορία ερωτήσεων. Συγκεκριμένα τις θεωρητικές ερωτήσεις τις χαρακτηρίζουμε με κωδικό 1, τις ερωτήσεις επεξεργασίας κειμένου με τον κωδικό 2, την κατηγορία ερωτήσεων υπολογιστικά φύλλα με κωδικό 3 και τις ερωτήσεις στις παρουσιάσεις με κωδικό 4. Το πεδίο CategoryName και CodeName αποτελούν κωδικές ονομασίες για την κάθε κατηγορία ερωτήσεων. Για παράδειγμα στην κατηγορία επεξεργασία κειμένου έχουμε CategoryName το λεκτικό word και CodeName την τιμή 200. Τα στοιχεία αυτά δεν τα συναντάει ο χρήστης στην εφαρμογή αλλά είναι εσωτερικές πληροφορίες που χρησιμοποιούνται από την εφαρμογή. Το πεδίο DescriptionName αποτελεί το περιγραφικό όνομα της κάθε κατηγορίας εξετάσεων. Για το προηγούμενο παράδειγμα που αναφέραμε, το CategoryName είναι word και το DescriptionName είναι επεξεργασία κειμένου. Το περιγραφικό όνομα εμφανίζεται σε όλα τα παράθυρα αποτελεσμάτων μέσα στην εφαρμογή. Το πεδίο NumOfQuestions είναι τύπου ακέραιος αριθμός και κρατάει το πλήθος των ερωτήσεων για κάθε κατηγορία ερωτήσεων. Στη συνέχεια παρουσιάζουμε τον SQL κώδικα που χρησιμοποιήσαμε για τη δημιουργία του πίνακα.

```
/* SQL script για τη δημιουργία του πίνακα tblQuestionCategory */
CREATE TABLE [dbo].[tblQuestionCategory](
    [CategoryID] [int] NOT NULL,
    [CategoryName] [nvarchar](20) NULL,
```



```

[CodeName] [int] NULL,
[DescriptionName] [nvarchar](50) NOT NULL,
[NumOfQuestions] [int] NOT NULL,
PRIMARY KEY CLUSTERED([CategoryID] ASC));

```

5.2.6 Ο πίνακας Questions

Ο πίνακας Questions χρησιμοποιείται για να κρατάει δεδομένα σχετικά με τις ερωτήσεις των εξετάσεων. Ο πίνακας αυτός έχει πέντε πεδία και διασυνδέεται με σχέση «ένα προς πολλά» με τον πίνακα Execution που περιγράψαμε προηγουμένως και με σχέση «ένα προς ένα» με τον πίνακα TheoryAnswers.

Το πεδίο QuestionID είναι ο αναγνωριστικός κωδικός για κάθε ερώτηση που τρέχει στις εξετάσεις. Το πεδίο αυτό αποτελεί πρωτεύον κλειδί για τον πίνακα Questions και σε καμία περίπτωση δε μπορεί να παραμείνει κενό. Το QuestionID λαμβάνει τιμές από τον διαχειριστή του συστήματος όταν δημιουργεί τις ερωτήσεις και ο τύπος δεδομένων του είναι ακέραιος αριθμός. Στο πεδίο Question αποθηκεύονται οι εκφωνήσεις των ερωτήσεων και είναι τύπου αλφαριθμητικού μέγιστου μήκους. Το πεδίο αυτό είναι υποχρεωτικό στην καταχώριση δεδομένων και λαμβάνει δεδομένα από τον διαχειριστή όταν δημιουργεί την ερώτηση. Στο πεδίο QuestionTotalPoints αποθηκεύεται ο βαθμός που πιάνει κάθε ερώτηση και μπορεί να πάρει τις διακριτές τιμές 1 ή 2. Είναι στην ευχέρεια του διαχειριστή να καθορίσει το βαθμό κάθε ερώτησης, αλλά συνηθίζεται να παίρνουν 2 βαθμούς οι ερωτήσεις που αποτελούνται από πολλά υποερωτήματα. Το πεδίο QuestionIdCategory αποθηκεύει τον κωδικό της κατηγορίας που ανήκει η ερώτηση και είναι τύπου ακέραιος αριθμός. Αποτελεί ξένο κλειδί στη διασύνδεση με τον πίνακα tblQuestionCategory που περιγράψαμε στην προηγούμενη ενότητα. Το τελευταίο πεδίο του πίνακα, το πεδίο QuestionSection χρησιμοποιείται για να κρατάμε σχόλια για κάθε κατηγορία ερωτήσεων. Έχει δημιουργηθεί με τύπο δεδομένων αλφαριθμητικό και δεν είναι υποχρεωτικό στην καταχώριση δεδομένων κατά τη δημιουργία μιας ερώτησης.

```

/* SQL script για τη δημιουργία του πίνακα Questions*/
CREATE TABLE [dbo].[Questions](
    [QuestionID] [int] NOT NULL,
    [Question] [nvarchar](max) NOT NULL,
    [QuestionTotalPoints] [int] NOT NULL,
    [QuestionIdCategory] [int] NOT NULL,
    [QuestionSection] [nvarchar](255) NULL,
    PRIMARY KEY CLUSTERED([QuestionID] ASC),
    FOREIGN KEY([QuestionIdCategory])
    REFERENCES [dbo].[tblQuestionCategory]([CategoryID]));

```

5.2.7 Ο πίνακας TheoryAnswers

Ο πίνακας TheoryAnswers χρησιμοποιείται για να κρατάει δεδομένα σχετικά με τις απαντήσεις των θεωρητικών ερωτήσεων. Ο πίνακας αυτός αποτελείται από έξι πεδία και διασυνδέεται με σχέση «ένα προς ένα» με τον πίνακα Questions.

Το πεδίο AnswerID είναι το αναγνωριστικό πεδίο για κάθε εγγραφή του πίνακα και κατά συνέπεια και το πρωτεύον κλειδί. Οι κωδικοί του πεδίου αυτού είναι ίδιοι με τους κωδικούς των ερωτήσεων που σχετίζονται, έτσι ώστε σε κάθε ερώτηση του πίνακα Questions να αντιστοιχούν οι απαντήσεις του πίνακα TheoryAnswers. Απαντήσεις μπορούμε να βρούμε μόνο για τις θεωρητικές ερωτήσεις πολλαπλής επιλογής. Τα πεδία Answer1, Answer2, Answer3 και Answer4 αντιστοιχούν στις τέσσερις εναλλακτικές απαντήσεις κάθε θεωρητικής ερώτησης και είναι υποχρεωτικά πεδία. Ο τύπος δεδομένων τους είναι αλφαριθμητικό μεγάλου μήκους. Το τελευταίο πεδίο του πίνακα, το πεδίο CorrectAnswer χρησιμοποιείται για να αποθηκεύσει τη σωστή απάντηση μεταξύ των τεσσάρων εναλλακτικών απαντήσεων. Είναι υποχρεωτικό πεδίο, τύπου δεδομένων ακέραιος και λαμβάνει τιμές στο διάστημα τιμών {1,2,3,4}.

```

/* SQL script για τη δημιουργία του πίνακα TheoryAnswers */
CREATE TABLE [dbo].[TheoryAnswers](
    [AnswerID] [int] NOT NULL,
    [Answer1] [nvarchar](max) NOT NULL,
    [Answer2] [nvarchar](max) NOT NULL,
    [Answer3] [nvarchar](max) NOT NULL,
    [Answer4] [nvarchar](max) NOT NULL,
    [CorrectAnswer] [int] NOT NULL,
PRIMARY KEY CLUSTERED([AnswerID] ASC));

```

5.3 Διερεύνηση των ερωτημάτων της ΒΔ

Η Γλώσσα Δομημένων Ερωτημάτων (Structured Query Language - SQL) είναι μια αυτόνομη γλώσσα που χρησιμοποιείται για τον χειρισμό πινάκων σε μια ΒΔ. Σχεδόν όλα τα γνωστά εμπορικά ΣΔΒΔ την υποστηρίζουν. Η συγγραφή προτάσεων SQL μπορεί να πραγματοποιηθεί με τη χρήση οπτικών εργαλείων ή απλά γράφοντας κώδικα SQL σε ένα συντάκτη κειμένου. Τα περισσότερα σύγχρονα ΣΔΒΔ διαθέτουν τέτοια οπτικά εργαλεία συμπεριλαμβανομένου της Access και του SQL Server μέσω του SQL Server Management Studio. Επίσης το Visual Studio διαθέτει το περιβάλλον του Query Designer που είναι ένας οπτικός επεξεργαστής για προτάσεις SQL που μας επιτρέπει να τις εκτελούμε και να βλέπουμε τα αποτελέσματά τους. Αν και τα οπτικά εργαλεία εμφανίζουν ένα φιλικό και εύχρηστο περιβάλλον για τη δημιουργία προτάσεων SQL αυτό δεν σημαίνει ότι θα υποκαταστήσουν την ίδια τη γλώσσα.

Οι προτάσεις SQL χωρίζονται σε δύο βασικές κατηγορίες, που στην πραγματικότητα θεωρούνται ως δύο ξεχωριστές γλώσσες. Τις προτάσεις SQL για τον χειρισμό, που αποτελούν τη Γλώσσα Χειρισμού Δεδομένων (Data Manipulation Language - DML) και τις προτάσεις για τον ορισμό των αντικείμενων των ΒΔ, που σχηματίζουν τη Γλώσσα Ορισμού Δεδομένων (Data Definition Language - DDL). Η γλώσσα DML της γλώσσας SQL είναι επίσης γνωστή και ως ερωτήματα. Υπάρχουν δύο τύποι ερωτημάτων: τα ερωτήματα επιλογής και τα ερωτήματα ενεργειών. Τα ερωτήματα επιλογής χρησιμοποιούνται για την ανάκτηση πληροφοριών από τη ΒΔ, επιστρέφοντας ένα σύνολο εγγραφών. Τα πεδία που χρησιμοποιούμε για να σχεδιάσουμε το ερώτημα μπορεί να προέρχονται από έναν ή περισσότερους πίνακες. Τα ερωτήματα ενεργειών τροποποιούν τα αντικείμενα της ΒΔ ή δημιουργούν νέα αντικείμενα και τα προσθέτουν στη βάση δεδομένων. Στις επόμενες ενότητες θα παρουσιάσουμε τα ερωτήματα SQL που σχεδιάσαμε για να ανακτήσουμε δεδομένα από την ΒΔ της εφαρμογής.

5.3.1 Το ερώτημα QueryResults

Το ερώτημα αυτό χρησιμοποιείται για να εξάγει τα αποτελέσματα των εξετάσεων κάθε χρήστη μετά το τέλος της εξέτασης. Στα αποτελέσματα θα πάρουμε πληροφορίες που έχουν να κάνουν με την ημερομηνία και την ώρα εκτέλεσης της εξέτασης, την κατηγορία της εξέτασης, τον κωδικό της ερώτησης, την εκφώνηση της ερώτησης, τους κερδισμένους βαθμούς από τον χρήστη και τους συνολικούς βαθμούς της ερώτησης, αν τυχόν αγνοήθηκε η ερώτηση από τον χρήστη και το χρόνο που δαπάνησε για να την απαντήσει. Τα αποτελέσματα του ερωτήματος αυτού προβάλλονται σε ένα αντικείμενο DataGridView στην κλάση DataAnalyticsResults και εμφανίζονται στο παράθυρο της εφαρμογής *Αναλυτικά Αποτελέσματα Χρήστη*. Το αντικείμενο DataGridView χρησιμοποιεί ένα αντικείμενο DataSet, το UserDataSet.xsd του μηχανισμού ADO.NET για να συμπληρώσει τα δεδομένα. Στην συνέχεια παρουσιάζεται ο SQL κώδικας που εκτελείται για να πάρουμε το παράθυρο αποτελεσμάτων που εμφανίζεται στην Εικόνα 41.

```

/*Εμφανίζει αποτελέσματα για τους χρήστες και τις εξετάσεις που έχουν τρέξει*/
SELECT Questions.QuestionID, Questions.Question, Execution.Score,
        Questions.QuestionTotalPoints, Execution.IsIgnore, Execution.ExecTime,
        Execution.QuestionTime, tblUsers.UserId,
tblQuestionCategory.CategoryName
FROM (tblStudents INNER JOIN tblUsers ON tblStudents.StudentID = tblUsers.IdStud)

```

```
INNER JOIN ((tblQuestionCategory INNER JOIN tblSession
    ON tblQuestionCategory.CategoryID = tblSession.IdCategory)
INNER JOIN (Questions INNER JOIN Execution ON Questions.QuestionID = Execution.IDQ)
    ON tblSession.SessionID = Execution.IdSession) ON (tblUsers.UserId =
tblSession.IDS) AND (tblUsers.UserId = Execution.IDS);
```

	Κωδ.	Ερώτηση	Βαθμός απάντησης	Βαθμός ερώτησης	Αγνοήθηκε	Χρόνος
▶	101	Ποιο είναι το παλαιότερο λειτουργικό σύστημα;	1	1	<input type="checkbox"/>	00:14
	102	Ποιο από τα παραπάνω δεν είναι λειτουργικό σύστημα για κινητά ...	1	1	<input type="checkbox"/>	00:07
	103	Ένα διαδικτυακό ημερολόγιο στο οποίο ο διαχειριστής καταγράφε...	1	1	<input type="checkbox"/>	00:12
	104	Ποιο από τα παραπάνω δεν αποτελεί έκδοση του λειτουργικού συ...	1	1	<input type="checkbox"/>	00:14
	105	Ποια από τις προτάσεις περιγράφει καλύτερα τον όρο της πληροφ...	0	1	<input checked="" type="checkbox"/>	00:13
	106	Το τείχος προστασίας (firewall) είναι ...	1	1	<input type="checkbox"/>	00:20
	107	Τι δεν ισχύει σχετικά με την κρυπτογράφηση των δεδομένων στο...	1	1	<input type="checkbox"/>	00:08
	108	Το ακρωνύμιο HTTP προέρχεται από τα αρχικά των λέξεων:	0	1	<input type="checkbox"/>	00:08
	109	Ποιο από τα παραπάνω δεν είναι πρόγραμμα περιήγησης στο Διαδ...	1	1	<input type="checkbox"/>	00:05
	110	Μια Βάση Δεδομένων (ΒΔ) είναι...	1	1	<input type="checkbox"/>	00:10
	111	Ποια από τις παραπάνω πρότασεις περιγράφει καλύτερα το περιε...	1	1	<input type="checkbox"/>	00:11
	112	Ποιο από τα παραπάνω είναι μέσο μόνιμης αποθήκευσης δεδομέν...	1	1	<input type="checkbox"/>	00:07

Εικόνα 41. Στιγμιότυπο του DataGridView από την εκτέλεση του ερωτήματος QueryResults

Επειδή η εκτέλεση του ερωτήματος θα είχε ως αποτέλεσμα να εμφανίσει πληροφορίες για τα αποτελέσματα των εξετάσεων όλων των χρηστών, αλλά εμείς χρειαζόμαστε μόνο τα αποτελέσματα του ενεργού χρήστη, χρησιμοποιώντας την μέθοδο Fill του αντικείμενου TableAdapter καταφέραμε να παραμετροποιήσουμε το παραπάνω DataSet χρησιμοποιώντας ως παράμετρο το πεδίο του κωδικού χρήστη που είναι το ids. Παρακάτω φαίνεται η πρόταση που παραμετροποιεί το επιστρεφόμενο DataSet

```
'Εγγραφές που φορτώνει από το ερώτημα QueryResults
Me.QueryResultsTableAdapter.Fill(Me.UserDataSet.QueryResults, CType(ids, Integer))
```

Με μια δεύτερη μέθοδο που δημιουργήσαμε, την FillbyCategory μας δίνεται η δυνατότητα να επιλέξουμε την εμφάνιση των αποτελεσμάτων ανά κατηγορία εξέτασης. Τη μέθοδο αυτή τη χρησιμοποιούμε για να φιλτράρουμε τα αποτελέσματα.

```
'Φιλτράρισμα δεδομένων DataGridView ανά κατηγορία
Me.QueryResultsTableAdapter.FillByCategory(Me.UserDataSet.QueryResults,
    CType(ids, Integer), cmbCategory.Text)
```

5.3.2 Το ερώτημα QuerySessionTotalResults

Το ερώτημα αυτό χρησιμοποιείται για να εξάγει τα συγκεντρωτικά αποτελέσματα των εξετάσεων κάθε χρήστη. Τα συγκεντρωτικά αποτελέσματα περιλαμβάνουν την κατηγορία εξέτασης, τον βαθμό του χρήστη ως % ποσοστό στην κλίμακα 0-100 και την ημερομηνία και ώρα διεξαγωγής της εξέτασης. Τα αποτελέσματα του ερωτήματος αυτού προβάλλονται σε ένα αντικείμενο DataGridView στην κλάση DataSummaryResults και εμφανίζονται στο παράθυρο της εφαρμογής *Συγκεντρωτικά Αποτελέσματα Χρήστη*. Το UserTotalDataSet.xsd του μηχανισμού ADO.NET είναι το DataSet που χρησιμοποιείται για να συμπληρώσει με δεδομένα το αντικείμενο DataGridView όπως παρουσιάζεται στην Εικόνα 42.

	Κατηγορία εξέτασης	Βαθμός (%)	Ημερομηνία εξέτασης
▶	Επεξεργασία Κειμένου	50,00%	01/09/2016 12:53
	Θεωρία Πληροφορικής	83,33%	01/09/2016 12:49

Εικόνα 42. Στιγμιότυπο του DataGridView από την εκτέλεση του ερωτήματος

Στην συνέχεια παρουσιάζεται ο SQL κώδικας του ερωτήματος QuerySessionTotalResults

```
/*Εμφανίζει τα συγκεντρωτικά αποτελέσματα των χρηστών*/
SELECT tblMembers.Name, tblMembers.Surname, tblSession.DateExecution,
tblUsers.UserId,
Sum(Execution.Score) AS ΑθροισμαScore,
Sum(Questions.QuestionTotalPoints) AS ΑθροισμαQuestionTotalPoints,
tblQuestionCategory.CategoryName, tblQuestionCategory.DescriptionName,
tblSession.SessionID
FROM (tblMembers INNER JOIN tblUsers ON tblMembers.MemberID=tblUsers.IdStud)
INNER JOIN ((tblQuestionCategory INNER JOIN ((Questions INNER JOIN Execution
ON Questions.QuestionID=Execution.IDQ) LEFT JOIN TheoryAnswers
ON Questions.QuestionID=TheoryAnswers.AnswerID)
ON tblQuestionCategory.CategoryID=Questions.QuestionIdCategory) INNER JOIN tblSession
ON (tblQuestionCategory.CategoryID=tblSession.IdCategory) AND
(tblSession.SessionID=Execution.IdSession)) ON tblUsers.UserId=tblSession.IDS
GROUP BY tblMembers.Name, tblMembers.Surname, tblSession.DateExecution,
tblUsers.UserId, tblQuestionCategory.CategoryName,
tblQuestionCategory.DescriptionName, tblSession.SessionID
ORDER BY tblSession.DateExecution DESC;
```

5.3.3 Το ερώτημα QueryMemberNumQuestions

Αυτό το ερώτημα χρησιμοποιείται για να εξαγάγει μέσα από τη ΒΔ το πλήθος των ερωτήσεων που έχει απαντήσει κάθε χρήστης της εφαρμογής σε καθεμία από τις κατηγορίες εξετάσεων. Τα πεδία που χρησιμοποιούνται στο ερώτημα είναι το όνομα, το επίθετο και ο αριθμός μητρώου του εγγεγραμμένου μέλους και οι κατηγορίες των εξετάσεων. Για να υπολογίσουμε το πλήθος των ερωτήσεων κάθε κατηγορίας χρησιμοποιήσαμε τη συνάρτηση Count() και μετρήσαμε τους κωδικούς των ερωτήσεων. Τα αποτελέσματα εμφανίζονται στο παράθυρο της εφαρμογής *Ιστορικό Αποτελεσμάτων* που είναι το παράθυρο μέσα από το οποίο κάθε χρήστης βλέπει όλες τις εξετάσεις που έχει εκτελέσει. Στην Εικόνα 43 βλέπουμε ένα στιγμιότυπο της εκτέλεσης του ερωτήματος όπως αυτό εμφανίζεται μέσα στην εφαρμογή. Παρακάτω μπορούμε να δούμε τον κώδικα SQL που τρέχει κατά την εκτέλεση του ερωτήματος.

The screenshot shows a user interface with the following elements:

- Αριθμός μητρώου: 4
- Όνομα: Μαρία
- Επίθετο: Οικονόμου
- Πλήθος ερωτήσεων που απαντήθηκαν:
- Table with 2 columns: Category Name and Count.

Κατηγορία	Αριθμός
Επεξεργασία Κειμένου	19
Θεωρία Πληροφορικής	41

Εικόνα 43. Στιγμιότυπο από την εκτέλεση του ερωτήματος

```
/*Εμφανίζει το πλήθος των ερωτήσεων που έχει εκτελέσει κάθε χρήστης*/
SELECT tblMembers.Name, tblMembers.Surname, Count(Questions.QuestionID)
AS ΠλήθοςΤουQuestionID, tblQuestionCategory.DescriptionName, tblMembers.MemberID
FROM (tblMembers INNER JOIN tblUsers ON tblMembers.MemberID=tblUsers.IdStud)
INNER JOIN ((tblQuestionCategory INNER JOIN ((Questions INNER JOIN Execution
ON Questions.QuestionID=Execution.IDQ) LEFT JOIN TheoryAnswers
ON Questions.QuestionID=TheoryAnswers.AnswerID)
```

```
ON tblQuestionCategory.CategoryID=Questions.QuestionIdCategory) INNER JOIN tblSession
ON (tblQuestionCategory.CategoryID=tblSession.IdCategory) AND
(tblSession.SessionID=Execution.IdSession))
ON tblUsers.UserId=tblSession.IDS
GROUP BY tblMembers.Name, tblMembers.Surname, tblQuestionCategory.DescriptionName,
tblMembers.MemberID;
```

6 Φάση Μετάβασης

Η φάση της μετάβασης αποτελεί την τελευταία φάση του κύκλου ζωής του λογισμικού κατά την αντικειμενοστραφή προσέγγιση. Στην ενότητα 6.1 παρουσιάζουμε τις ενέργειες που πρέπει να γίνουν στη φάση αυτή και στην ενότητα 6.2 περιγράφουμε τον τρόπο με τον οποίο πραγματοποιήσαμε beta έλεγχο στο πλαίσιο ανάπτυξης της εφαρμογής. Στην ενότητα 6.3 παρουσιάζουμε τα εγχειρίδια χρήσης για τον απλό χρήστη και το διαχειριστή της εφαρμογής. Τα εγχειρίδια αυτά θα αποτελέσουν ένα πρώτο τρόπο για να εκπαιδευτούν οι χρήστες στο περιβάλλον του λογισμικού iTestU.

6.1 Η Φάση Μετάβασης (Transition Phase)

Η φάση Μετάβασης εστιάζει στο να διασφαλιστεί ότι το λογισμικό είναι διαθέσιμο για τους τελικούς χρήστες του. Η φάση Μετάβασης μπορεί να χρειαστεί πολλές επαναλήψεις για να ολοκληρωθεί και περιλαμβάνει τον έλεγχο του προϊόντος που πρόκειται να κυκλοφορήσει και μικρές ρυθμίσεις, οι οποίες βασίζονται στην ανατροφοδότηση πληροφοριών (feedback) από τους χρήστες. Σε αυτό το σημείο του κύκλου ζωής, οι πληροφορίες από τους χρήστες πρέπει να επικεντρώνονται κυρίως στη σωστή ρύθμιση του προϊόντος, σε ζητήματα διαμόρφωσης¹, εγκατάστασης και δυνατότητας χρησιμοποίησης. Όλα τα σημαντικά ζητήματα που αφορούν τη δομή πρέπει να έχουν επιλυθεί πολύ νωρίτερα στον κύκλο ζωής του έργου.

Στο τέλος της φάσης Μετάβασης, οι στόχοι του κύκλου ζωής πρέπει να έχουν επιτευχθεί και το έργο να είναι έτοιμο να ολοκληρωθεί. Σε μερικές περιπτώσεις, το τέλος του τρέχοντος κύκλου ζωής μπορεί να συμπέσει με την έναρξη ενός άλλου κύκλου ζωής για το ίδιο προϊόν, που θα οδηγήσει στην επόμενη γενιά ή στην επόμενη έκδοση του προϊόντος. Για άλλα έργα, το τέλος της μετάβασης μπορεί να συμπέσει με την παράδοση των «τεχνημάτων» (artifacts)² σε έναν τρίτο (third party), ο οποίος μπορεί να είναι υπεύθυνος για τις λειτουργίες, τη συντήρηση και τις βελτιώσεις του παραδοθέντος συστήματος.

Η φάση Μετάβασης μπορεί να είναι από απλή μέχρι εξαιρετικά σύνθετη, ανάλογα με το είδος του προϊόντος. Μια νέα κυκλοφορία ενός υπάρχοντος προϊόντος για εφαρμογές γραφείου μπορεί να είναι πολύ απλή, ενώ αντίθετα η αντικατάσταση του συστήματος ελέγχου εναέριας κυκλοφορίας μιας χώρας μπορεί να είναι υπερβολικά περίπλοκη.

Οι δραστηριότητες που εκτελούνται κατά τη διάρκεια μιας επανάληψης στη φάση Μετάβασης εξαρτώνται από το στόχο. Παραδείγματος χάρη, κατά τη διόρθωση σφαλμάτων (bugs), η εφαρμογή και ο έλεγχος είναι συνήθως αρκετά. Εάν, όμως, πρέπει να προστεθούν νέα χαρακτηριστικά, η επανάληψη θα είναι παρόμοια με αυτή στη φάση Κατασκευής και θα απαιτεί Ανάλυση και Σχεδιασμό.

Η είσοδος στη φάση Μετάβασης πραγματοποιείται όταν προϊόν είναι αρκετά ώριμο, ώστε να επεκταθεί στο χώρο των τελικών χρηστών. Αυτό τυπικά απαιτεί την ολοκλήρωση κάποιου χρησιμοποιήσιμου υποσύνολου του συστήματος με αποδεκτά επίπεδα ποιότητας και τα έγγραφα για το χρήστη (user documentation), έτσι ώστε η μετάβαση στο χρήστη να έχει θετικά αποτελέσματα για όλα τα συμβαλλόμενα μέρη.

¹ Διαμόρφωση (Configuration) = Οι απαιτήσεις, ο σχεδιασμός και η υλοποίηση που καθορίζουν μια συγκεκριμένη έκδοση ενός συστήματος ή ενός τμήματος του συστήματος.

² Artifact = Πληροφορία, η οποία: 1) παράγεται, τροποποιείται ή χρησιμοποιείται από μια διαδικασία, 2) ορίζει μια περιοχή αρμοδιότητας και 3) υπόκειται σε έλεγχο έκδοσης (version control). Artifact μπορεί να είναι ένα μοντέλο, ένα στοιχείο ενός μοντέλου ή ένα έγγραφο. Ένα έγγραφο μπορεί να περιλαμβάνει άλλα έγγραφα. Τα artifacts είναι η είσοδος και η έξοδος των δραστηριοτήτων και αποτελούν το μηχανισμό με τον οποίο οι πληροφορίες διαδίδονται μεταξύ των δραστηριοτήτων.

Οι κύριοι στόχοι της φάσης Μετάβασης περιλαμβάνουν:

- Βήτα έλεγχο (beta testing) για την επικύρωση του νέου συστήματος βάσει των προσδοκιών των χρηστών.
- Μετατροπή των λειτουργικών βάσεων δεδομένων.
- Εκπαίδευση χρηστών και συντηρητών.
- Την αρχική παρουσία του προϊόντος (roll-out) στο χώρο του μάρκετινγκ, της διανομή και των πωλήσεων
- Το σχεδιασμό για τη συγκεκριμένη ανάπτυξη, την εμπορική συσκευασία και την παραγωγή, το ξεκίνημα των πωλήσεων, την εκπαίδευση προσωπικού
- Το συντονισμό των δραστηριοτήτων, όπως η διόρθωση σφαλμάτων, η βελτίωση της απόδοσης και της δυνατότητας χρησιμοποίησης
- Εκτίμηση των βασικών γραμμών της ανάπτυξης με βάση το πλήρες όραμα και τα κριτήρια αποδοχής του προϊόντος
- Επίτευξη συμφωνίας των ενδιαφερόμενων στο ότι οι βασικές γραμμές της ανάπτυξης είναι πλήρεις
- Επίτευξη συμφωνίας των ενδιαφερόμενων στο ότι οι βασικές γραμμές της ανάπτυξης είναι συνεπείς με τα κριτήρια αξιολόγησης του αρχικού οράματος για το προϊόν.

Οι σημαντικές δραστηριότητες της φάσης Μετάβασης περιλαμβάνουν:

- Την εκτέλεση σχεδίων ανάπτυξης
- Την οριστικοποίηση του υλικού υποστήριξης των τελικών χρηστών
- Τη δοκιμή του προϊόντος στον τόπο ανάπτυξης
- Τη δημιουργία μιας κυκλοφορίας του προϊόντος
- Την εκμετάλλευση της ανατροφοδότησης πληροφοριών (feedback) από τους χρήστες
- Την σωστή ρύθμιση του προϊόντος βάσει της ανατροφοδότησης αυτής
- Τη διάθεση του προϊόντος στους τελικούς χρήστες

Στο τέλος της φάσης Μετάβασης βρίσκεται το τέταρτο ορόσημο του έργου, που είναι το ορόσημο της Κυκλοφορίας του Προϊόντος (Product Release Milestone). Σε αυτό το σημείο, αποφασίζεται εάν οι στόχοι επιτεύχθηκαν και εάν πρέπει να αρχίσει άλλος κύκλος ανάπτυξης. Σε μερικές περιπτώσεις, αυτό το ορόσημο μπορεί να συμπίπτει με το τέλος της φάσης Έναρξης για τον επόμενο κύκλο. Η Κυκλοφορία του Προϊόντος είναι το αποτέλεσμα της αναθεώρησης και της αποδοχής από τον πελάτη των προϊόντων του έργου.

Κριτήρια Αξιολόγησης είναι εάν ο χρήστης είναι ικανοποιημένος και εάν οι πραγματικές δαπάνες για πόρους έναντι των προγραμματισμένων δαπανών είναι αποδεκτές. Στο σημείο αυτό, το προϊόν είναι στην παραγωγή και ξεκινάει ο κύκλος της συντήρησης μετά την κυκλοφορία.

6.2 Έλεγχος - Beta Testing

Αν και πραγματοποιήθηκαν εκτενείς έλεγχοι σε όλο το στάδιο της κωδικοποίησης και υπήρξε και συνολικός έλεγχος του λογισμικού (Alpha testing) μετά την ολοκλήρωσή του, αυτό δεν σημαίνει ότι θα είναι απαλλαγμένο από λάθη. Μάλιστα μπορεί να είναι αρκετά ασταθές και να προκαλέσει προβλήματα κατά τη χρήση του. Ένα τέτοιο σενάριο ασταθούς λειτουργίας αντιμετωπίσαμε μετά την ολοκλήρωση του λογισμικού αλλά αντιμετωπίστηκε με επιτυχία.

Για να μειώσουμε την πιθανότητα εμφάνισης και άλλων τέτοιων λαθών μελλοντικά, πήραμε την απόφαση να προχωρήσουμε σε πιο δραστικούς και εξαντλητικούς ελέγχους. Για τους ελέγχους αυτούς χρησιμοποιήθηκαν ως εθελοντές (beta testers) σπουδαστές Δημοσίων Ινστιτούτων Επαγγελματικής Κατάρτισης (ΔΙΕΚ) και Κέντρων Ελευθέρων Σπουδών. Η ανάδραση που λάβαμε από τις εύστοχες παρατηρήσεις τους βοήθησαν στο σωστότερο επανασχεδιασμό ενός μικρού τμήματος της εφαρμογής. Συγκεκριμένα βρέθηκε και διορθώθηκε ένα σοβαρό

πρόβλημα με την λειτουργία της επανεξέτασης μέσα στο παράθυρο των γενικών αποτελεσμάτων του χρήστη.

Γενικά θα μπορούσαμε να πούμε ότι ο έλεγχος beta επειδή λαμβάνει χώρα στο χώρο των τελικών χρηστών είναι πολύ σημαντικός για να επικυρώσει τη χρηστικότητα, τη λειτουργικότητα, και την αξιοπιστία της εφαρμογής.

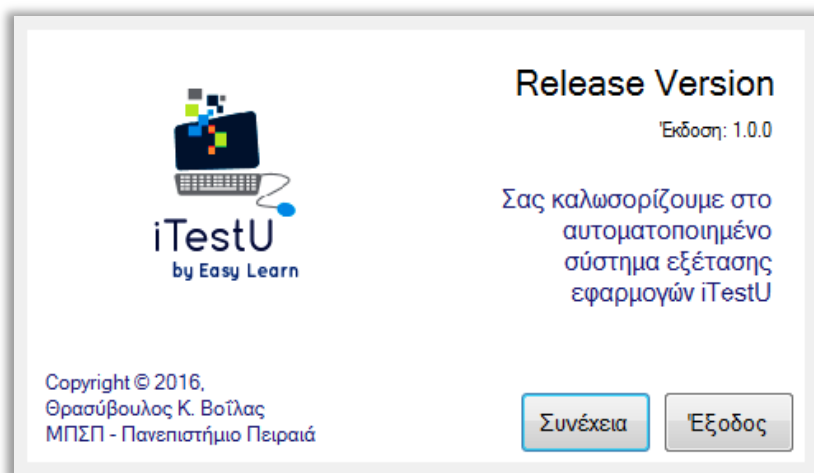
6.3 Εγχειρίδιο χρήσης απλού χρήστη

Η εφαρμογή iTestU εμφανίζει διαφορετικές λειτουργίες ανάλογα με το χρήστη που συνδέεται. Μπορούμε να διακρίνουμε δύο βασικούς ρόλους, τον απλό χρήστη και τον διαχειριστή. Πολλές από τις λειτουργίες είναι κοινές, ενώ προφανώς ο διαχειριστής έχει αυξημένο αριθμό λειτουργιών σε σχέση με τον απλό χρήστη.

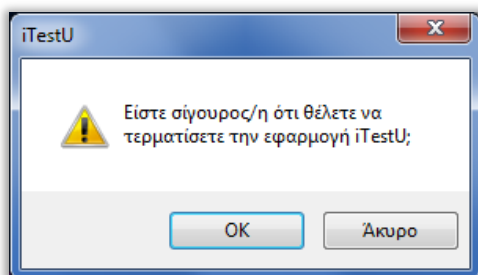
Στην ενότητα αυτή γίνεται μια παρουσίαση των λειτουργιών της εφαρμογής από την πλευρά του απλού χρήστη.

6.3.1 Εισαγωγικές οθόνες σύνδεσης και εγγραφής

Όταν ο χρήστης εκκινήσει την εφαρμογή εμφανίζεται η οθόνη που φαίνεται στην Εικόνα 44. Ο χρήστης μπορεί να συνεχίσει με την εφαρμογή ενεργοποιώντας το κουμπί *Συνέχεια* ή να διακόψει τη λειτουργία της πατώντας στο κουμπί *Έξοδος*. Σε αυτήν την περίπτωση εμφανίζεται προειδοποιητικό μήνυμα (Εικόνα 45) μέσα από το οποίο μπορεί να επιβεβαιώσει τον τερματισμό πατώντας στο κουμπί *OK* ή να επιστρέψει στην εισαγωγική οθόνη.

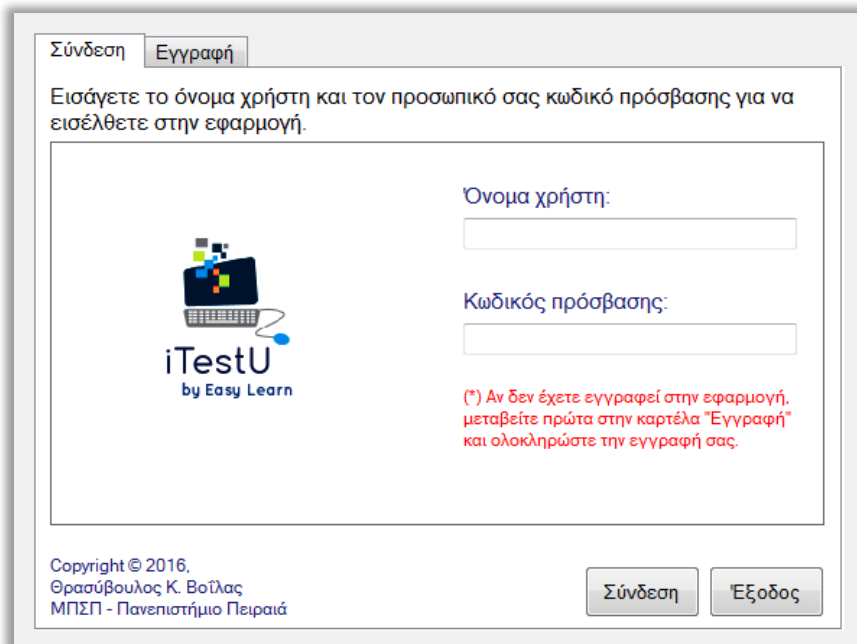


Εικόνα 44. Εισαγωγική οθόνη της εφαρμογής



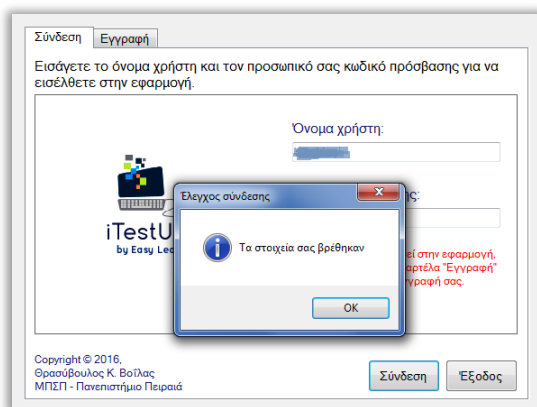
Εικόνα 45. Παράθυρο προειδοποιητικού μηνύματος

Αν ο χρήστης επιλέξει να συνεχίσει, η επόμενη οθόνη που εμφανίζεται είναι η οθόνη μέσα από την οποία μπορεί να συνδεθεί ή να εγγραφεί στην εφαρμογή (Εικόνα 46).

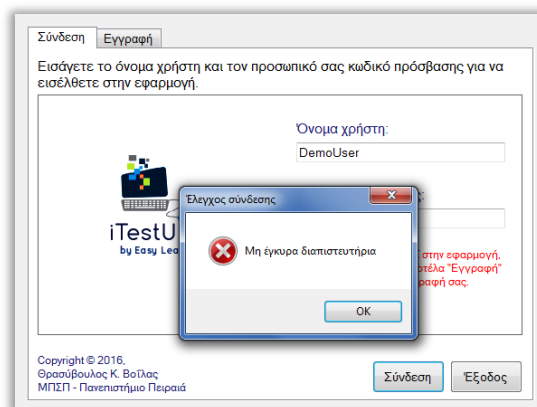


Εικόνα 46. Παράθυρο για τη σύνδεση χρήστη

Εφόσον ο χρήστης διαθέτει λογαριασμό στην εφαρμογή πληκτρολογεί στα αντίστοιχα πεδία το όνομα χρήστη και τον κωδικό πρόσβασης που διαθέτει και στην συνέχεια ενεργοποιεί το κουμπί *Σύνδεση*. Αν τα στοιχεία που εισήγαγε είναι σωστά το σύστημα θα τον αναγνωρίσει (Εικόνα 47) και θα συνδεθεί στην εφαρμογή, διαφορετικά εμφανίζεται μήνυμα σφάλματος και αποτρέπεται η σύνδεση του χρήστη (Εικόνα 48).



Εικόνα 47. Επιτυχημένη σύνδεση χρήστη



Εικόνα 48. Αποτυχημένη σύνδεση χρήστη

Στην περίπτωση που ο χρήστης δε διαθέτει λογαριασμό μπορεί να εγγραφεί στην εφαρμογή ενεργοποιώντας την καρτέλα *Εγγραφή* και να συμπληρώσει τα πεδία (Εικόνα 49). Για να ολοκληρώσει την εγγραφή πρέπει να συμπληρώσει όλα τα υποχρεωτικά πεδία και να πατήσει στο κουμπί *Εγγραφή* (Εικόνα 50) διαφορετικά εμφανίζεται προειδοποιητικό μήνυμα (Εικόνα 51) προτρέποντας το χρήστη να συμπληρώσει τα πεδία που έχουν παραμείνει κενά. Τα στοιχεία της εγγραφής μπορούν να τροποποιηθούν μετά την ολοκλήρωση της εγγραφής. Επιπλέον μπορούν

να συμπληρωθούν μη υποχρεωτικά πεδία που είχαν παραμείνει κενά κατά την εγγραφή. Αφού ολοκληρωθεί η εγγραφή μπορεί να συνδεθεί στην εφαρμογή ή να την τερματίσει.

Με το κουμπί *Απαλοιφή*, ο χρήστης διαγράφει ταυτόχρονα το περιεχόμενο όλων των πεδίων του παραθύρου.

Σύνδεση Εγγραφή

Εισάγετε τα στοιχεία σας για να εγγραφείτε στην εφαρμογή.

Όνομα: * Όνομα χρήστη: *

Επίθετο: * Κωδικός πρόσβασης: *

Διεύθυνση email: Ιδιότητα: --- Επιλέξτε ή εισάγετε ---

Φύλο
 Άρρεν Θήλυ

(*) Υποχρεωτικό πεδίο κατά την εγγραφή.

Copyright © 2016.
 Θρασύβουλος Κ. Βοΐλας
 ΜΠΣΠ - Πανεπιστήμιο Πειραιά

Εγγραφή Απαλοιφή

Εικόνα 49. Παράθυρο για την εγγραφή

Σύνδεση Εγγραφή

Εισάγετε τα στοιχεία σας για να εγγραφείτε στην εφαρμογή.

Όνομα: * Όνομα χρήστη: *

Αναστασία nansy

Επίθετο: * Κωδικός πρόσβασης: *

Οικονόμου

Διεύθυνση email: Ιδιότητα: --- Επιλέξτε ή εισάγετε ---

anastasia@gmail.com

Φύλο
 Άρρεν Θήλυ

Copyright © 2016.
 Θρασύβουλος Κ. Βοΐλας
 ΜΠΣΠ - Πανεπιστήμιο Πειραιά

Εγγραφή Απαλοιφή

Ολοκλήρωση εγγραφής

Η εγγραφή σας ολοκληρώθηκε, για να συνδεθείτε επιστρέψτε στην καρτέλα Σύνδεση

OK

Εικόνα 50. Επιτυχημένη εγγραφή χρήστη

Σύνδεση Εγγραφή

Εισάγετε τα στοιχεία σας για να εγγραφείτε στην εφαρμογή.

Όνομα: * Όνομα χρήστη: *

Αναστασία Κωδικός πρόσβασης: *

Επίθετο: * Κωδικός πρόσβασης: *

Οικονόμου

Διεύθυνση email: Ιδιότητα: --- Επιλέξτε ή εισάγετε ---

anastasia@gmail.com

Φύλο
 Άρρεν Θήλυ

Copyright © 2016.
 Θρασύβουλος Κ. Βοΐλας
 ΜΠΣΠ - Πανεπιστήμιο Πειραιά

Εγγραφή Απαλοιφή

Σφάλμα εγγραφής

Πρέπει να συμπληρώσετε όλα τα υποχρεωτικά πεδία της φόρμας

OK

Εικόνα 51. Προειδοποίηση για κενά πεδία

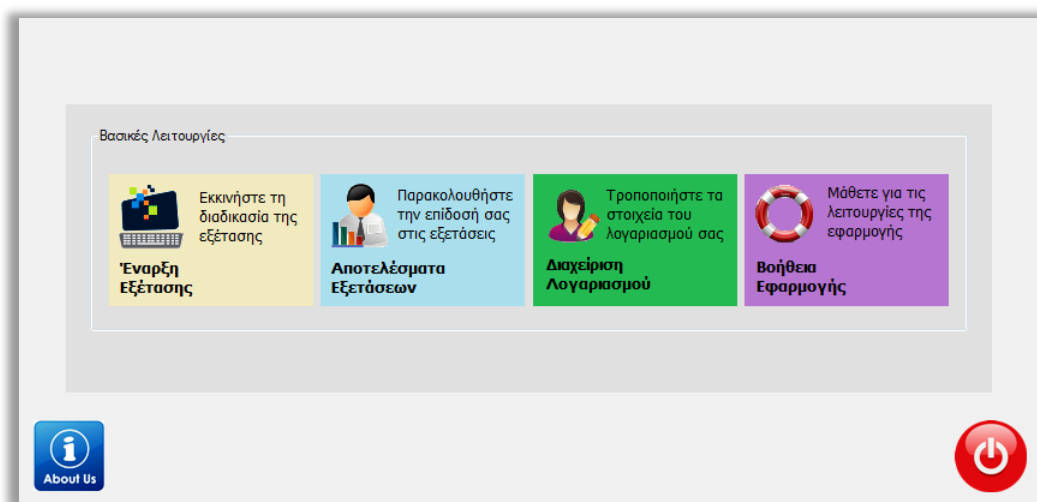
Να σημειώσουμε εδώ ότι ένας χρήστης μπορεί να έχει λογαριασμό στην εφαρμογή χωρίς να έχει εγγραφεί ο ίδιος στο παρελθόν, με την προϋπόθεση ότι την εγγραφή την έχει κάνει ο διαχειριστής του συστήματος και τον έχει προμηθεύσει με το όνομα χρήστη και τον κωδικό πρόσβασης. Η λειτουργία αυτή περιγράφεται σε επόμενη ενότητα.

6.3.2 Κεντρικό παράθυρο επιλογών χρήστη

Εφόσον ο χρήστης ταυτοποιηθεί και συνδεθεί, εμφανίζεται το κεντρικό παράθυρο επιλογών με τις βασικές λειτουργίες της εφαρμογής iTestU (Εικόνα 52).

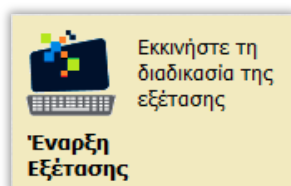
Οι λειτουργίες που μπορεί να εκτελέσει είναι:

- Έναρξη Εξέτασης
- Αποτελέσματα Εξετάσεων
- Διαχείριση Λογαριασμού
- Βοήθεια Εφαρμογής



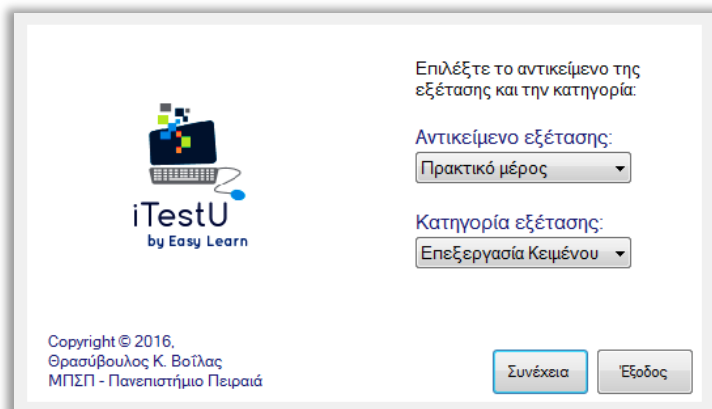
Εικόνα 52. Κεντρικό παράθυρο επιλογών απλού χρήστη

6.3.3 Έναρξη εξέτασης

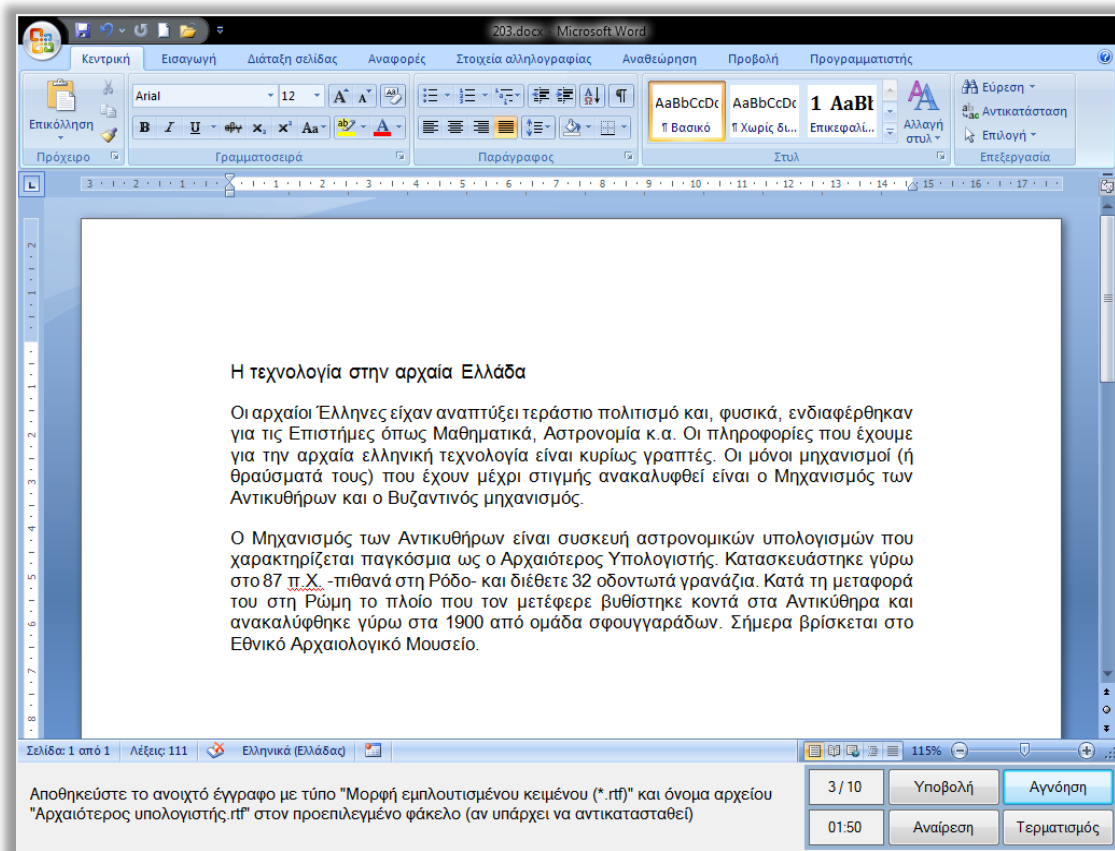


Η λειτουργία αυτή είναι η σημαντικότερη της εφαρμογής iTestU. Μέσα από αυτή ο χρήστης εκκινεί θεωρητικές εξετάσεις με την μορφή ερωτήσεων πολλαπλής επιλογής ή πρακτικές εξετάσεις χρησιμοποιώντας κατάλληλα λογισμικά του πακέτου Office σε πραγματικό περιβάλλον. Η επιλογή της κατηγορίας εξετάσεων γίνεται μέσα από το παράθυρο επιλογών (Εικόνα 53).

Εφόσον ο χρήστης επιλέξει αντικείμενο και κατηγορία εξέτασης μπορεί να ενεργοποιήσει το κουμπί *Συνέχεια* για να ξεκινήσει το περιβάλλον της εξέτασης (Εικόνα 54). Το παράθυρο του περιβάλλοντος εξέτασης αποτελείται από δύο περιοχές. Στο επάνω μέρος είναι το εξεταζόμενο αντικείμενο και στο κάτω μέρος η εκφώνηση της ερώτησης και τα κουμπιά ενεργειών.



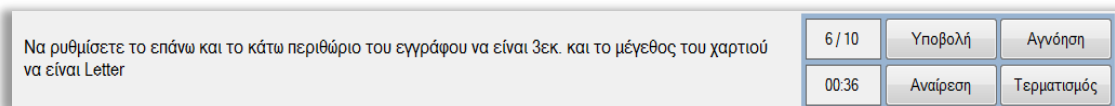
Εικόνα 53. Επιλογή αντικειμένου και κατηγορίας εξέτασης



Εικόνα 54. Περιβάλλον εξέτασης

Για το εξεταζόμενο αντικείμενο, στην περίπτωση των θεωρητικών ερωτήσεων, έχουμε 4 εναλλακτικές απαντήσεις εκ των οποίων μόνο μία είναι η σωστή, ενώ στην εξέταση του πρακτικού αντικειμένου ανοίγει κάποιο από τα λογισμικά του πακέτου Microsoft Office. Συγκεκριμένα για την εξέταση στην επεξεργασία κειμένου ανοίγει το Microsoft Word, για τα υπολογιστικά φύλλα το Excel και για τις παρουσιάσεις το PowerPoint. Τα λογισμικά αυτά τρέχουν μέσα σε πραγματικό περιβάλλον και κατά συνέπεια ο χρήστης μπορεί να εκτελέσει όποια λειτουργία επιθυμεί.

Στο κάτω μέρος του περιβάλλοντος εξέτασης μπορούμε να διακρίνουμε στα αριστερά την εκφώνηση της ερώτησης που ουσιαστικά είναι η ενέργεια που πρέπει να εκτελέσουμε και στη δεξιά πλευρά τα χειριστήρια της ερώτησης (Εικόνα 55).



Εικόνα 55. Εκφώνηση και χειριστήρια της ερώτησης

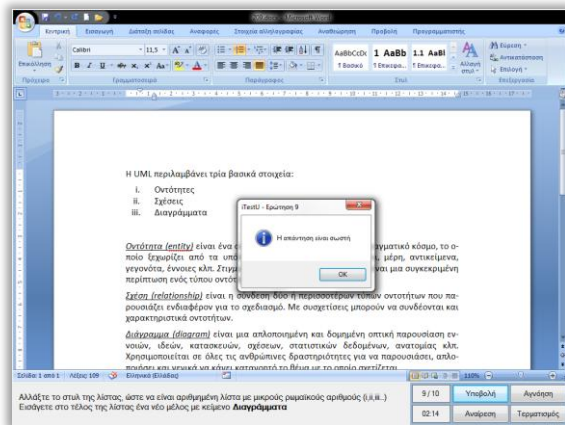
Πιο αναλυτικά, στο δεξί τμήμα των χειριστηρίων μπορούμε να διακρίνουμε τον αριθμό της τρέχουσας ερώτησης και τον συνολικό αριθμό ερωτήσεων, το συνολικό χρόνο που έχουμε δαπανήσει στην εξέταση και τα κουμπιά ενεργειών που αναλύονται στη συνέχεια.

Το κουμπί ενέργειας *Υποβολή*, υποβάλλει το αποτέλεσμα των ενεργειών, που έχει εκτελέσει ο χρήστης στο περιβάλλον του λογισμικού, προς αξιολόγηση. Ανάλογα με τις ενέργειες που έχουν εκτελεστεί μια ερώτηση μπορεί να χαρακτηριστεί ως σωστή (Εικόνα 56), ως μερικής

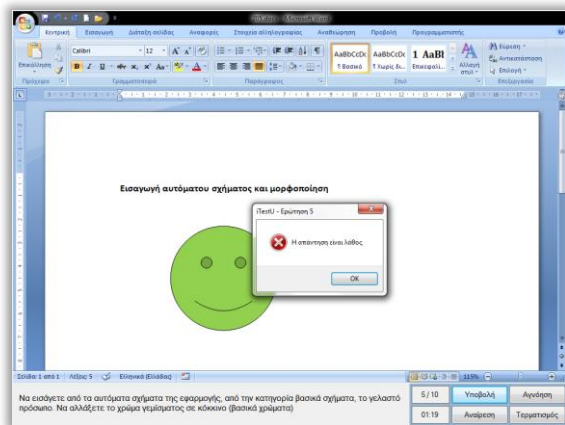
απαντημένη ή ως λανθασμένη (Εικόνα 57). Αντίστοιχα κατάλληλα μηνύματα εμφανίζονται στην οθόνη του χρήστη που τον πληροφορούν για το αποτέλεσμα της απάντησης.

Το κουμπί ενέργειας *Αγνόηση*, προσπερνάει την τρέχουσα ερώτηση και οδηγεί στην εμφάνιση της επόμενης ερώτησης. Στην περίπτωση που βρισκόμαστε στην τελευταία ερώτηση ανοίγει το παράθυρο των αναλυτικών αποτελεσμάτων. Οι αγνοημένες ερωτήσεις στην τελική αξιολόγηση προσμετρούνται ως λανθασμένες.

Το κουμπί *Αναίρεση*, καθαρίζει την ερώτηση από τις ενέργειες που έχει εκτελέσει ο χρήστης μέχρι εκείνη τη στιγμή. Η ερώτηση εμφανίζεται στην αρχική της κατάσταση και ο χρήστης μπορεί να εκτελέσει νέα προσπάθεια για απάντηση. Δεν υπάρχει περιορισμός στον αριθμό των αναιρέσεων που μπορεί να εκτελέσει ο χρήστης σε μια ερώτηση.



Εικόνα 56. Σωστή απάντηση



Εικόνα 57. Λάθος απάντηση

Το κουμπί *Τερματισμός*, ολοκληρώνει τη διαδικασία της τρέχουσας εξέτασης και ανοίγει το παράθυρο των αναλυτικών αποτελεσμάτων. Οι ερωτήσεις που δεν απαντήθηκαν όταν ενεργοποιήθηκε το κουμπί Τερματισμός βαθμολογούνται με μηδέν βαθμούς. Πριν τον τερματισμό της εξέτασης εμφανίζεται κατάλληλο προειδοποιητικό μήνυμα για να αποδεχτεί ή να απορρίψει ο χρήστης τον τερματισμό της διαδικασίας εξέτασης.

Μετά την ολοκλήρωση της εξέτασης εμφανίζεται το παράθυρο με τα αναλυτικά αποτελέσματα της εξέτασης (Εικόνα 58). Το παράθυρο αυτό μας δίνει αναλυτικά, με τη μορφή πίνακα, στοιχεία όπως είναι ο κωδικός της ερώτησης, η εκφώνηση της ερώτησης, το βαθμό που πήραμε σε κάθε απάντηση και το συνολικό βαθμό που έπιασε η ερώτηση, ποιες ερωτήσεις αγνοήθηκαν και το χρόνο που δαπανήσαμε σε κάθε ερώτηση.

Κωδ.	Ερώτηση	Βαθμός απάντησης	Βαθμός ερώτησης	Αγνοήθηκε	Χρόνος
101	Ποιο είναι το παλαιότερο λειτουργικό σύστημα;	1	1	<input type="checkbox"/>	00:14
102	Ποιο από τα παραπάνω δεν είναι λειτουργικό σύστημα για κινητά ...	1	1	<input type="checkbox"/>	00:07
103	Ένα διαδικτυακό ημερολόγιο στο οποίο ο διαχειριστής καταγράφε...	1	1	<input type="checkbox"/>	00:12
104	Ποιο από τα παραπάνω δεν αποτελεί έκδοση του λειτουργικού συ...	1	1	<input type="checkbox"/>	00:14
105	Ποια από τις προτάσεις περιγράφει καλύτερα τον όρο της πληροφ...	0	1	<input checked="" type="checkbox"/>	00:13
106	Το τείχος προστασίας (firewall) είναι ...	1	1	<input type="checkbox"/>	00:20
107	Τι δεν ισχύει σχετικά με την κρυπτογράφηση των δεδομένων στο...	1	1	<input type="checkbox"/>	00:08
108	Το ακρωνύμιο HTTP προέρχεται από τα αρχικά των λέξεων:	0	1	<input type="checkbox"/>	00:08
109	Ποιο από τα παραπάνω δεν είναι πρόγραμμα περιήγησης στο Διαδ...	1	1	<input type="checkbox"/>	00:05
110	Μια Βάση Δεδομένων (ΒΔ) είναι...	1	1	<input type="checkbox"/>	00:10
111	Ποια από τις παραπάνω προτάσεις περιγράφει καλύτερα το περιε...	1	1	<input type="checkbox"/>	00:11
112	Ποιο από τα παραπάνω είναι μέσο μόνιμης αποθήκευσης δεδομέν...	1	1	<input type="checkbox"/>	00:07

Εικόνα 58. Παράθυρο αναλυτικών αποτελεσμάτων

Στο κάτω μέρος του παραθύρου αναλυτικών αποτελεσμάτων υπάρχουν εργαλεία πλοήγησης μεταξύ των γραμμών του πίνακα και βοηθητικά εργαλεία για φιλτράρισμα των δεδομένων, την εμφάνιση συγκεντρωτικών αποτελεσμάτων και την εξαγωγή των δεδομένων. Επίσης μπορούμε να διακρίνουμε κουμπιά ενεργειών για νέα προσπάθεια εξέτασης, για επιστροφή στο κεντρικό παράθυρο επιλογών και το κουμπί τερματισμού της εφαρμογής.

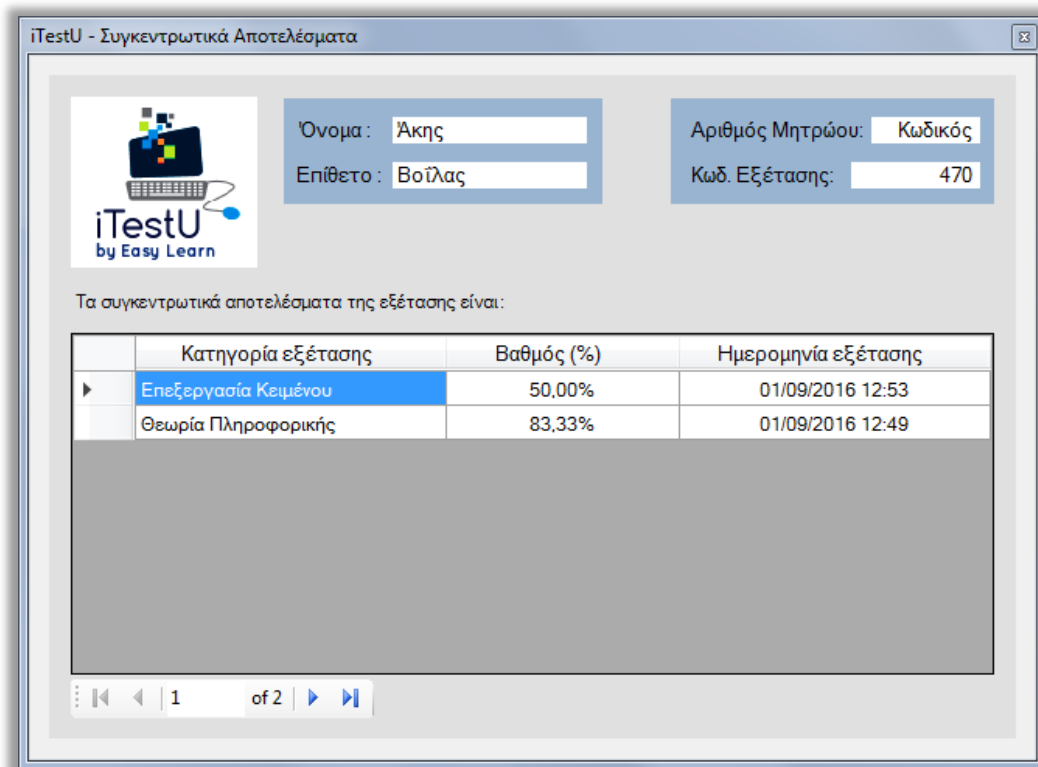
Εικόνα 59. Βοηθητικά εργαλεία και κουμπιά ενεργειών

Μέσα από την ομάδα εντολών *Φίλτρα* ο χρήστης μπορεί να επιλέξει από την πτυσσόμενη λίστα (combo box) για ποια κατηγορία εξέτασης θέλει να προβάλει αναλυτικά αποτελέσματα. Με την ενεργοποίηση της επιλογής αυτόματα ενημερώνεται ο πίνακας αποτελεσμάτων. Για να προβάλλουμε όλα τα αποτελέσματα ανεξάρτητα από την κατηγορία εξέτασης, ενεργοποιούμε το κουμπί κατάργηση φίλτρου.

Από την ομάδα εντολών *Εξαγωγή* ο χρήστης μπορεί να εξαγάγει τα αναλυτικά αποτελέσματα σε αρχείο Excel που αποθηκεύεται αυτόματα στην επιφάνεια εργασίας του υπολογιστή. Εναλλακτικά μπορεί να εκτυπώσει το αρχείο αποτελεσμάτων ή να το αποθηκεύσει με την μορφή αναφοράς τύπου portable document format – pdf. Σε κάθε περίπτωση ο χρήστης επιλέγει τη μορφή εξαγωγής και ενεργοποιεί το διπλανό κουμπί.

Το κουμπί *Συγκεντρωτικά* εμφανίζει σε νέο παράθυρο τα αποτελέσματα της εξέτασης με τον τελικό βαθμό ως % ποσοστό στη κλίμακα 0 έως 100. Στο παράθυρο αυτό εμφανίζονται

πληροφορίες με τα στοιχεία του χρήστη και τον κωδικό εξέτασης, η κατηγορία εξέτασης, και η ημερομηνία και ώρα εκκίνησης της εξέτασης (Εικόνα 60).

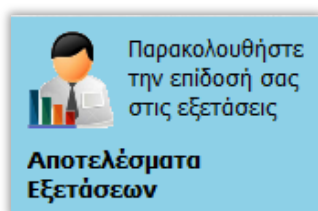


Εικόνα 60. Παράθυρο συγκεντρωτικών αποτελεσμάτων

Το κουμπι ενέργειας *Νέα προσπάθεια* δίνει την δυνατότητα στο χρήστη να επιλέξει με γρήγορο τρόπο το επόμενο αντικείμενο και κατηγορία εξέτασης που θέλει να εξεταστεί.

Το κουμπι *Παράθυρο επιλογών* δίνει τη δυνατότητα στο χρήστη να επιστρέψει στο κεντρικό παράθυρο επιλογών για να επιλέξει κάποια άλλη λειτουργία του λογισμικού iTestU.

6.3.4 Αποτελέσματα εξετάσεων



Η λειτουργία αυτή δίνει στο χρήστη τη δυνατότητα να παρακολουθήσει το ιστορικό των αποτελεσμάτων των εξετάσεων που έχει πραγματοποιήσει (Εικόνα 61). Μέσα από τη λειτουργία μπορεί να έχει μια πιο ασφαλή και ολοκληρωμένη εικόνα για τις επιδόσεις του. Να σημειώσουμε εδώ ότι κάθε χρήστης μπορεί να έχει πρόσβαση μόνο στα δικά του αποτελέσματα και όχι άλλων χρηστών της εφαρμογής.

Στο επάνω μέρος του παραθύρου εμφανίζονται πληροφορίες με τα στοιχεία του χρήστη όπως είναι ο μοναδικός αριθμός μητρώου, το όνομα και το επίθετο και κάποια στατιστικά στοιχεία που σχετίζονται με το συνολικό αριθμό των ερωτήσεων που έχει απαντήσει ανά κατηγορία εξέτασης. Ακολουθεί ένας πίνακας με το ιστορικών των αποτελεσμάτων όπου εμφανίζονται η κατηγορία της εξέτασης, ο βαθμός και η ημερομηνία που πραγματοποιήθηκε η κάθε εξέταση.

Αριθμός μητρώου : 4
 Όνομα : Μαρία
 Επώνυμο : Οικονόμου

Πλήθος ερωτήσεων που απαντήθηκαν
 Επεξεργασία Κειμένου : 19
 Θεωρία Πληροφορικής : 41

Αποτελέσματα

Κατηγορία εξέτασης	Βαθμός (%)	Ημερομηνία εξέτασης
Θεωρία Πληροφορικής	83,33%	30/08/2016 11:36
Επεξεργασία Κειμένου	75,00%	25/08/2016 22:01
Θεωρία Πληροφορικής	100,00%	25/08/2016 21:57
Παρουσιάσεις	91,67%	22/08/2016 23:20
Υπολογιστικά Φύλλα	91,67%	22/08/2016 23:13
Θεωρία Πληροφορικής	83,33%	22/08/2016 19:39
Επεξεργασία Κειμένου	66,67%	22/08/2016 19:32
Θεωρία Πληροφορικής	41,67%	20/08/2016 15:32

Φίλτρα
 Ανά κατηγορία :
 Από: 20/08/2016 00:00
 Εως: 22/08/2016 23:59

Εξαγωγή
 Excel (*.xlsx)
 Report (*.pdf)

Εικόνα 61. Ιστορικό αποτελεσμάτων χρήστη

Ο πίνακας αυτός είναι διαμορφώσιμος, αφού μπορεί να τροποποιηθεί αλλάζοντας τα κριτήρια ταξινόμησης σε αύξουσα ή φθίνουσα ταξινόμηση, κάνοντας απλό κλικ στον τίτλο κάθε στήλης. Η προεπιλογή είναι να εμφανίζει τα αποτελέσματα κατά φθίνουσα χρονολογική σειρά. Επίσης ο χρήστης μπορεί να αλλάξει τη σειρά εμφάνισης των πεδίων και να διαμορφώσει το πλάτος των πεδίων. Με τα εργαλεία της γραμμής πλοήγησης μπορεί να επιλέξει και να εστιάσει σε κάποια συγκεκριμένη εγγραφή.

Στο κάτω μέρος του παραθύρου υπάρχουν βοηθητικά εργαλεία για το φιλτράρισμα και την εξαγωγή των δεδομένων. Το φιλτράρισμα μπορεί να γίνει με κριτήριο την κατηγορία εξέτασης και την ημερομηνία διεξαγωγής της εξέτασης. Τα κριτήρια μπορεί να είναι και συνδυαστικά. Σε κάθε περίπτωση αφού καθορίσουμε τα κριτήρια φιλτραρίσματος, ενεργοποιούμε το κουμπί εφαρμογή φίλτρου για να πάρουμε τα αποτελέσματα. Για να καταργήσουμε τα κριτήρια απλά ενεργοποιούμε το κουμπί κατάργηση φίλτρου.

Η ομάδα εντολών *Εξαγωγή* χρησιμοποιείται για να εξαγάγουμε το ιστορικό αποτελεσμάτων σε αρχείο Excel ή σε αναφορά τύπου pdf. Πρώτα επιλέγουμε τον τύπο του αρχείου και στη συνέχεια ενεργοποιούμε το κουμπί εξαγωγή που βρίσκεται δίπλα από τις επιλογές. Στην περίπτωση του αρχείου Excel αποθηκεύεται αυτόματα ένα αρχείο με το όνομα του χρήστη στην επιφάνεια εργασίας του υπολογιστή του σαν αυτό που εμφανίζεται παρακάτω (Εικόνα 62). Αντίστοιχο αρχείο δημιουργείται και στην περίπτωση που επιλέξουμε αναφορά τύπου pdf. Το αρχείο που δημιουργείται είναι έτοιμο για εκτύπωση ή μπορεί ο χρήστης να το αποθηκεύσει και να το εκτυπώσει αργότερα.

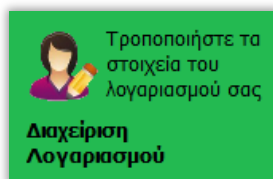
iTestU - Συγκριτικά αποτελέσματα 1/9/2016 4:39:22 μμ

Όνοματεπώνυμο: Μαρία Οικονόμου		Αρ. Μητρώου: 4
Κατηγορία εξέτασης	Βαθμός (%)	Ημερομηνία εξέτασης
Θεωρία Πληροφορικής	83,33%	30/8/2016 11:36
Επεξεργασία Κειμένου	75,00%	25/8/2016 22:01
Θεωρία Πληροφορικής	100,00%	25/8/2016 21:57
Παρουσιάσεις	91,67%	22/8/2016 23:20
Υπολογιστικά Φύλλα	91,67%	22/8/2016 23:15
Θεωρία Πληροφορικής	83,33%	22/8/2016 19:39
Επεξεργασία Κειμένου	66,67%	22/8/2016 19:32
Θεωρία Πληροφορικής	41,67%	20/8/2016 15:32

Σελίδα 1 από 1

Εικόνα 62. Αρχείο εξαγωγής Excel

6.3.5 Διαχείριση λογαριασμού

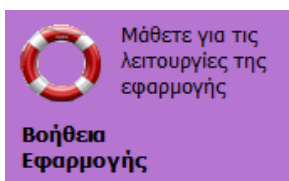


Η λειτουργία αυτή δίνει στο χρήστη τη δυνατότητα να διαχειριστεί τα προσωπικά στοιχεία του λογαριασμού που διατηρεί στην εφαρμογή. Το παράθυρο της διαχείρισης λογαριασμού αποτελείται από δύο περιοχές, την περιοχή *Βασικά στοιχεία* και την περιοχή *Δευτερεύοντα στοιχεία* (Εικόνα 63).

Στην περιοχή των βασικών στοιχείων εμφανίζονται το όνομα και το επίθετο του χρήστη, το όνομα χρήστη και ο κωδικός πρόσβασης που αποτελούν τα στοιχεία ταυτοποίησης στην εφαρμογή. Όλα τα παραπάνω πεδία είναι υποχρεωτικά και πρέπει να είναι συμπληρωμένα. Στην αντίθετη περίπτωση που κάποιο από αυτά τα πεδία παραμείνει κενό εμφανίζεται προειδοποιητικό μήνυμα. Αν ο χρήστης θέλει να κάνει οποιαδήποτε τροποποίηση των πεδίων αυτών, απλά πληκτρολογεί τα νέα στοιχεία και στο τέλος ενεργοποιεί το κουμπί *Αλλαγή*.

Στην περιοχή των δευτερευόντων στοιχείων εμφανίζονται τα πεδία της διεύθυνσης ηλεκτρονικού ταχυδρομείου, το φύλο και η ιδιότητά του. Τα πεδία αυτά είναι προαιρετικά και μπορεί να τα συμπληρώσει ο χρήστης όποτε το επιθυμεί. Στο πεδίο *Ιδιότητα* που είναι ένα αναπτυσσόμενο πλαίσιο μπορεί να επιλέξει μια ιδιότητα από τη λίστα ή και να εισαγάγει μια νέα ιδιότητα πληκτρολογώντας την.

6.3.6 Βοήθεια της εφαρμογής



Με την λειτουργία αυτή ο χρήστης μπορεί να ανοίξει το εγχειρίδιο χρήσης της εφαρμογής και να αναζητήσει βοήθεια για όλες τις λειτουργίες. Το εγχειρίδιο χρήσης ανοίγει ως αρχείο pdf, άρα για να το διαβάσετε πρέπει να διαθέτετε εγκατεστημένο το λογισμικό Acrobat Reader ή άλλο παρόμοιο.

Εικόνα 63. Παράθυρο διαχείριση λογαριασμού

6.4 Εγχειρίδιο χρήσης διαχειριστή

Στην ενότητα αυτή γίνεται μια παρουσίαση των λειτουργιών της εφαρμογής από την πλευρά του διαχειριστή. Προφανώς οι λειτουργίες που είναι κοινές με τον απλό χρήστη αναλύθηκαν στην προηγούμενη ενότητα. Στην ενότητα αυτή θα περιγράψουμε μόνο τις λειτουργίες του διαχειριστή που δεν είδαμε στην περίπτωση του απλού χρήστη.

6.4.1 Κεντρικό παράθυρο επιλογών διαχειριστή

Εφόσον ο χρήστης με δικαιώματα διαχειριστή ταυτοποιηθεί και συνδεθεί, εμφανίζεται το κεντρικό παράθυρο επιλογών με τις βασικές και προχωρημένες λειτουργίες της εφαρμογής iTestU (Εικόνα 64).

Οι βασικές λειτουργίες που μπορεί να εκτελέσει είναι:

- Έναρξη Εξέτασης
- Αποτελέσματα Εξετάσεων
- Διαχείριση Λογαριασμού
- Βοήθεια Εφαρμογής

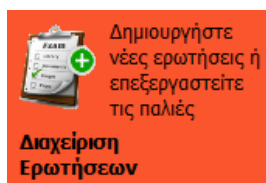
Οι προχωρημένες λειτουργίες που μπορεί να εκτελέσει είναι:

- Διαχείριση Ερωτήσεων
- Γενικά Αποτελέσματα
- Διαχείριση Χρηστών
- Στοιχεία Χρηστών

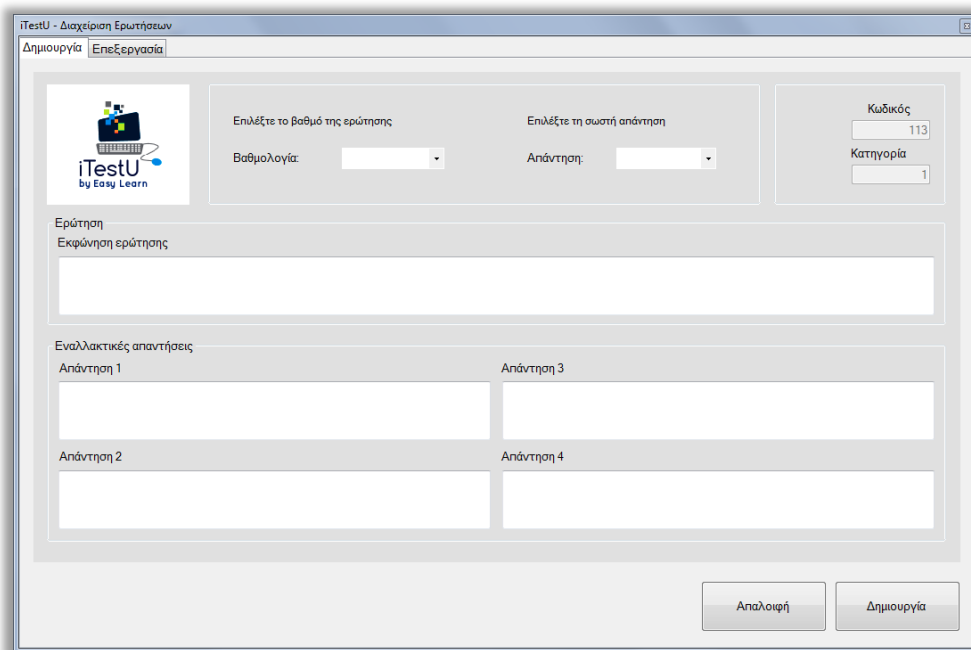


Εικόνα 64. Κεντρικό παράθυρο επιλογών διαχειριστή

6.4.2 Διαχείριση ερωτήσεων



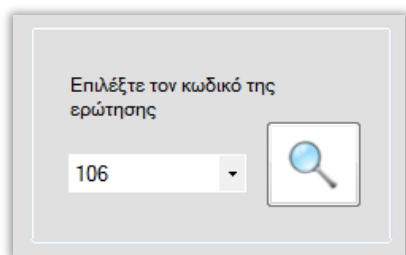
Η λειτουργία αυτή δίνει τη δυνατότητα στον διαχειριστή να δημιουργήσει νέες θεωρητικές ερωτήσεις πολλαπλής επιλογής καθώς και να επεξεργαστεί παλιές ερωτήσεις. Δεν υπάρχει δυνατότητα για επεξεργασία των πρακτικών ερωτήσεων. Για να δημιουργήσει νέες ερωτήσεις πρέπει να είναι ενεργοποιημένη η καρτέλα *Δημιουργία* (Εικόνα 65).



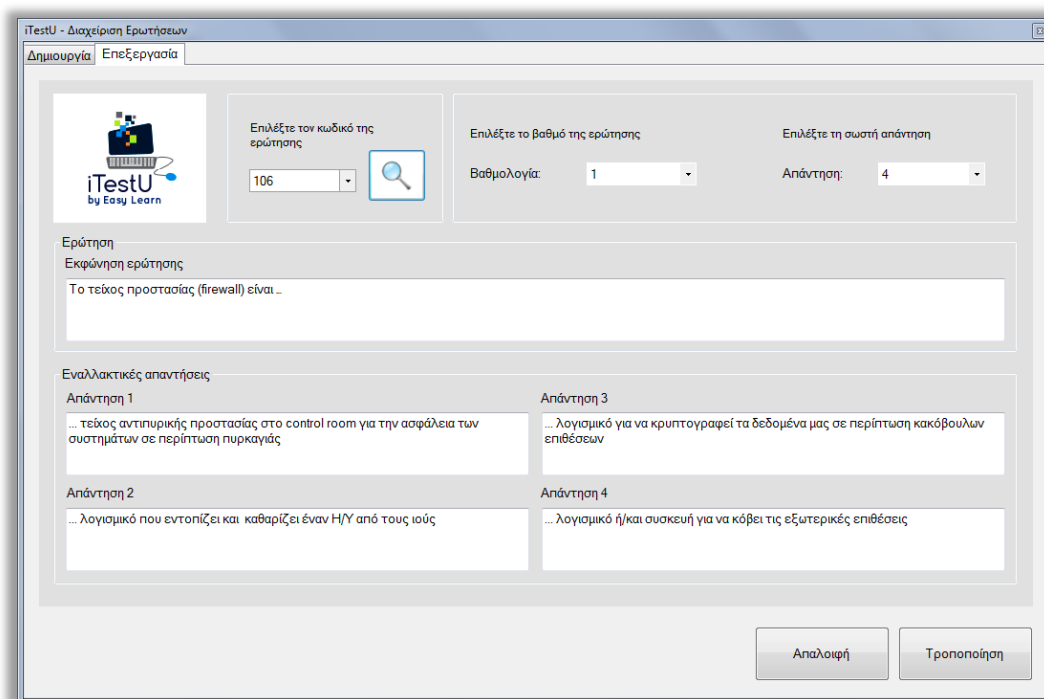
Εικόνα 65. Παράθυρο δημιουργίας ερώτησης

Ο διαχειριστής πρέπει να συμπληρώσει υποχρεωτικά όλα τα πεδία του παραθύρου και στη συνέχεια να ενεργοποιήσει το κουμπί *Δημιουργία*. Συγκεκριμένα πρέπει να συμπληρώσει τα πεδία με την εκφώνηση της ερώτησης, τα 4 πεδία με τις εναλλακτικές απαντήσεις, να επιλέξει το βαθμό που πιάνει η ερώτηση και να επιλέξει τη σωστή απάντηση. Τα πεδία του κωδικού της ερώτησης και της κατηγορίας είναι κλειδωμένα και ενημερώνονται αυτόματα από την εφαρμογή.

Στην περίπτωση που ο διαχειριστής θέλει να επεξεργαστεί μια υπάρχουσα ερώτηση πρέπει να μεταβεί στην καρτέλα Επεξεργασία του παραθύρου Διαχείριση Ερωτήσεων. Όλα τα πεδία του παραθύρου είναι ανενεργά εκτός από το πεδίο αναζήτησης της ερώτησης με βάση τον κωδικό (Εικόνα 66). Στο πεδίο αυτό, επιλέγει τον κωδικό της ερώτησης, που είναι μοναδικός για κάθε ερώτηση, και ενεργοποιεί το κουμπί αναζήτησης με το μεγεθυντικό φακό. Η ερώτηση εντοπίζεται από την εφαρμογή και φορτώνονται τα πεδία της φόρμας (Εικόνα 67).



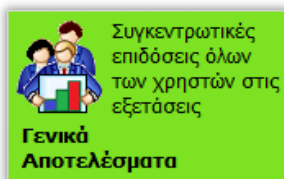
Εικόνα 66. Πεδίο αναζήτησης ερώτησης



Εικόνα 67. Παράθυρο επεξεργασίας ερώτησης

Ο διαχειριστής το μόνο που απομένει να κάνει είναι να επεξεργαστεί ένα ή περισσότερα από τα διαθέσιμα πεδία της ερώτησης και να ενεργοποιήσει το κουμπί *Τροποποίηση*. Είναι υποχρεωτικό να είναι συμπληρωμένα όλα τα πεδία της ερώτησης για να ολοκληρωθεί η διαδικασία. Στην αντίθετη περίπτωση εμφανίζεται προειδοποιητικό μήνυμα για την συμπλήρωση των πεδίων. Τέλος με το κουμπί Απαλοιφή μπορεί να διαγράψει το περιεχόμενο όλων των πεδίων της φόρμας.

6.4.3 Γενικά αποτελέσματα



Ο διαχειριστής με τη λειτουργία αυτή έχει τη δυνατότητα να παρακολουθήσει το ιστορικό των αποτελεσμάτων των εξετάσεων όλων των χρηστών της εφαρμογής. Μέσα από το γενικό ιστορικό αποτελεσμάτων μπορεί να έχει μια ολοκληρωμένη εικόνα για τις επιδόσεις όλων των χρηστών ανά κατηγορία εξέτασης (Εικόνα 68).

Όνομα	Επίθετο	Κατηγορία εξέτασης	Βαθμός (%)	Ημερομηνία
test	test	Θεωρία Πληροφορικής	0,00%	24/08/2016 16:09
test	test	Θεωρία Πληροφορικής	16,67%	24/08/2016 16:07
Θρασύβουλος	Βοΐλας	Θεωρία Πληροφορικής	66,67%	24/08/2016 16:03
Θρασύβουλος	Βοΐλας	Θεωρία Πληροφορικής	25,00%	24/08/2016 16:01
Δοκίμιος	Δοκιμόπουλος	Θεωρία Πληροφορικής	100,00%	24/08/2016 11:55
Λησής	Βοΐλας	Θεωρία Πληροφορικής	91,67%	23/08/2016 11:10
Μαρία	Οικονόμου	Θεωρία Πληροφορικής	83,33%	22/08/2016 19:39
Λησής	Βοΐλας	Θεωρία Πληροφορικής	33,33%	21/08/2016 20:18
Δοκίμιος	Δοκιμόπουλος	Θεωρία Πληροφορικής	25,00%	21/08/2016 18:51
Λησής	Βοΐλας	Θεωρία Πληροφορικής	100,00%	21/08/2016 14:38

Φίλτρα: Ανά κατηγορία: Θεωρία Πληροφορικής, Από: 21/08/2016 00:00, Έως: 24/08/2016 23:59

Εξαγωγή: Excel (*.xlsx), Crystal Report, Report (*.pdf)

Εικόνα 68. Γενικό ιστορικό αποτελεσμάτων

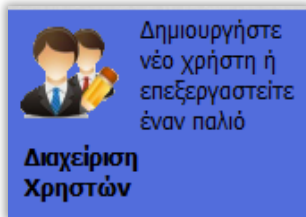
Τα αποτελέσματα εμφανίζονται με τη μορφή πίνακα, που είναι διαμορφώσιμος, αφού μπορεί να τροποποιηθεί αλλάζοντας τα κριτήρια ταξινόμησης για κάθε πεδίο σε αύξουσα ή φθίνουσα ταξινόμηση. Επίσης ο διαχειριστής μπορεί να αλλάξει τη σειρά εμφάνισης των πεδίων και να διαμορφώσει το πλάτος αυτών. Με τα εργαλεία της γραμμής πλοήγησης μπορεί να επιλέξει και να εστιάσει σε κάποια συγκεκριμένη εγγραφή. Στο παραπάνω στιγμιότυπο είναι επιλεγμένη η εγγραφή 6 και μπορούμε να πάρουμε πέρα από τις πληροφορίες που εμφανίζονται στον πίνακα επιπλέον πληροφορίες όπως είναι ο κωδικός χρήστη με τον οποίο είχε συνδεθεί ο εξεταζόμενος και ο κωδικός της εξέτασης. Οι πληροφορίες αυτές είναι μοναδικές και ταυτοποιούν μια εξέταση στην περίπτωση που ο διαχειριστής θελήσει να βρει επιπλέον πληροφορίες στη βάση δεδομένων της εφαρμογής.

Στο κάτω μέρος του παραθύρου υπάρχουν βοηθητικά εργαλεία για το φιλτράρισμα και την εξαγωγή των δεδομένων. Το φιλτράρισμα μπορεί να γίνει με κριτήριο την κατηγορία εξέτασης και την ημερομηνία διεξαγωγής της εξέτασης. Τα κριτήρια μπορεί να είναι και συνδυαστικά. Σε κάθε περίπτωση αφού καθορίσουμε τα κριτήρια φιλτραρίσματος, ενεργοποιούμε το κουμπί

εφαρμογή φίλτρου για να πάρουμε τα αποτελέσματα. Για να καταργήσουμε τα κριτήρια απλά ενεργοποιούμε το κουμπί κατάργηση φίλτρου. Στο παραπάνω στιγμιότυπο έχουμε φιλτράρει τα δεδομένα και παίρνουμε αποτελέσματα μόνο για την κατηγορία θεωρία πληροφορικής στο διάστημα ημερομηνιών 21/8/2016 έως 24/8/2016

Η ομάδα εντολών *Εξαγωγή* χρησιμοποιείται για να εξαγάγουμε τα αποτελέσματα σε αρχείο Excel ή σε αναφορά τύπου pdf. Πρώτα επιλέγουμε τον τύπο του αρχείου και στη συνέχεια ενεργοποιούμε το κουμπί εξαγωγή που βρίσκεται δίπλα από τις επιλογές.

6.4.4 Διαχείριση Χρηστών



Η λειτουργία αυτή δίνει τη δυνατότητα στον διαχειριστή να δημιουργήσει νέους απλούς χρήστες ή διαχειριστές της εφαρμογής. Επιπλέον δίνει τη δυνατότητα να επεξεργαστεί και να αλλάξει τα στοιχεία παλιών χρηστών. Για να δημιουργήσει ένα νέο χρήστη πρέπει να είναι ενεργοποιημένη η καρτέλα *Δημιουργία Νέου* και να εισαγάγει τα στοιχεία του χρήστη στα πεδία της φόρμας (Εικόνα 69).

The screenshot shows a web application window titled "iTestU - Διαχείριση Χρηστών". It has two tabs: "Δημιουργία Νέου" (selected) and "Αλλαγή Στοιχείων". The main content area contains the iTestU logo and a text box: "Εισάγετε τα παρακάτω στοιχεία για να εγγράψετε ένα νέο χρήστη στην εφαρμογή. Ο χρήστης που θα δημιουργήσετε μπορεί να τροποποιήσει τα στοιχεία αυτά μόλις συνδεθεί." Below this are two columns of form fields:

- Βασικά στοιχεία:**
 - Όνομα: *
 - Επίθετο: *
 - Όνομα χρήστη: *
 - Κωδικός πρόσβασης: *
- Δευτερεύοντα στοιχεία:**
 - Διεύθυνση email:
 - Ιδιότητα: (dropdown menu with "Επιλέξτε ή εισάγετε")
 - Φύλο:
 - Άρρεν
 - Θήλυ
 - Είναι Διαχειριστής;

At the bottom left, there is a red note: "(*) Υποχρεωτικό πεδίο". At the bottom right, there are two buttons: "Απαλοιφή" and "Εγγραφή".

Εικόνα 69. Παράθυρο δημιουργίας νέου χρήστη

Η φόρμα αποτελείται από δύο περιοχές, την περιοχή βασικών στοιχείων όπου εμφανίζονται το όνομα και το επίθετο του χρήστη, το πεδίο όνομα χρήστη και ο κωδικός πρόσβασης που αποτελούν τα στοιχεία ταυτοποίησης του χρήστη στην εφαρμογή. Όλα τα παραπάνω πεδία πρέπει να συμπληρωθούν υποχρεωτικά κατά την εγγραφή. Αν κάποιο από αυτά τα πεδία παραμείνει κενό η διαδικασία της εγγραφής δεν ολοκληρώνεται, αλλά εμφανίζεται προειδοποιητικό μήνυμα με τα πεδία που πρέπει να συμπληρωθούν.

Στην περιοχή των δευτερευόντων στοιχείων εμφανίζονται τα πεδία της διεύθυνσης ηλεκτρονικού ταχυδρομείου, το φύλο, η ιδιότητά του χρήστη και το πεδίο για δικαιώματα διαχειριστή. Τα πεδία αυτά είναι προαιρετικά και μπορούν να συμπληρωθούν αργότερα. Ειδικά τα πρώτα 3 πεδία μπορούν να συμπληρωθούν και τόσο από το διαχειριστή όσο και από τον χρήστη, όταν συνδεθεί στο λογαριασμό του, μέσα από την λειτουργία *Διαχείριση Λογαριασμού*. Ειδικά όμως, το τελευταίο πεδίο με τα δικαιώματα διαχειριστή μπορεί να επεξεργαστεί μόνο από κάποιον που έχει συνδεθεί στην εφαρμογή ως διαχειριστής και όχι από τον χρήστη.

Για να ολοκληρωθεί η διαδικασία της εγγραφής πρέπει στο τέλος να ενεργοποιηθεί το κουμπί *Εγγραφή*. Το σύστημα θα εμφανίσει μήνυμα για την επιτυχημένη ολοκλήρωση της διαδικασίας εγγραφής.

Εικόνα 70. Αναζήτηση χρήστη

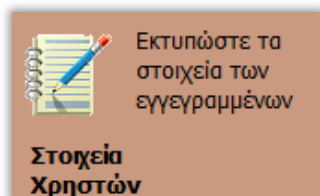
Στην περίπτωση που ο διαχειριστής θέλει να επεξεργαστεί μια υπάρχουσα εγγραφή χρήστη πρέπει να μεταβεί στην καρτέλα *Αλλαγή Στοιχείων* του παραθύρου Διαχείριση Χρηστών. Αρχικά όλα τα πεδία του παραθύρου είναι ανενεργά εκτός από το πεδίο αναζήτησης του χρήστη. Η αναζήτηση γίνεται με βάση τον κωδικό (Εικόνα 70). Ο διαχειριστής μπορεί να αναζητήσει το χρήστη που θέλει να επεξεργαστεί απευθείας στη λίστα με τα επίθετα ή να πληκτρολογήσει τα πρώτα γράμματα του επιθέτου και με αυτόν τον τρόπο να εντοπίσει το επίθετο που θέλει. Για κάθε επίθετο που επιλέγεται, στη διπλανή περιοχή, εμφανίζονται περισσότερες πληροφορίες όπως είναι ο αριθμός μητρώου του χρήστη, το μικρό του όνομα και το επίθετο. Με τον τρόπο αυτό διασφαλίζεται ότι έχουμε επιλέξει τον σωστό χρήστη. Από τη στιγμή που ο διαχειριστής εντοπίσει την επιθυμητή εγγραφή, ενεργοποιεί το κουμπί αναζήτησης με το μεγεθυντικό φακό. Η εγγραφή εντοπίζεται από την εφαρμογή και τα στοιχεία της φορτώνονται τα πεδία της φόρμας (Εικόνα 71). Τα πεδία της φόρμας παύουν να είναι ανενεργά οπότε ο διαχειριστής μπορεί να προχωρήσει με την επεξεργασία των στοιχείων.

Η φόρμα επεξεργασίας των στοιχείων έχει την ίδια μορφή που είχαμε συναντήσει και κατά τη δημιουργία του χρήστη. Συγκεκριμένα αποτελείται από 2 διακριτές περιοχές, την περιοχή στα αριστερά με τα υποχρεωτικά πεδία και την περιοχή με τα δευτερεύοντα μη υποχρεωτικά πεδία. Ο διαχειριστής αφού επεξεργαστεί όποια πεδία χρειάζεται, για να ολοκληρώσει την διαδικασία πρέπει να ενεργοποιήσει το κουμπί *Αλλαγή*. Σε αυτή την περίπτωση, εφόσον δεν υπάρχουν κενά υποχρεωτικά πεδία, θα εμφανιστεί μήνυμα που μας πληροφορεί ότι η εγγραφή ενημερώθηκε με επιτυχία.

Με το κουμπί *Απαλοιφή*, ο διαχειριστής μπορεί να καθарίσει ταυτόχρονα όλα τα πεδία της φόρμας. Δεν υπάρχει η δυνατότητα αναιρέσης της απαλοιφής οπότε πρέπει κάποιος να είναι προσεκτικός με τη χρήση αυτής της ενέργειας.

Εικόνα 71. Παράθυρο επεξεργασίας χρήστη

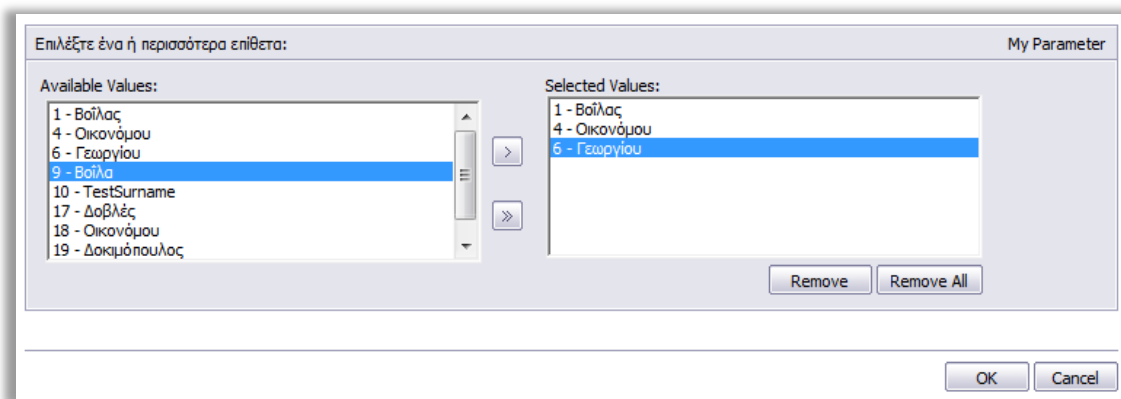
6.4.5 Στοιχεία Χρηστών



Η λειτουργία αυτή δίνει τη δυνατότητα στον διαχειριστή να εμφανίσει τα προσωπικά στοιχεία των εγγεγραμμένων χρηστών της εφαρμογής. Το αποτέλεσμα της λειτουργίας είναι μια αναφορά Crystal Report που μπορεί να εκτυπωθεί ή να αποθηκευτεί σε διάφορες μορφές.

Από τη στιγμή που ο διαχειριστής επιλέξει τη λειτουργία αυτή εμφανίζεται ένα παράθυρο διαλόγου για να επιλέξει τα επίθετα των χρηστών (Εικόνα 72). Το παράθυρο αυτό αποτελείται από δύο βασικές περιοχές. Την αριστερή περιοχή όπου είναι μια λίστα με όλα τις διαθέσιμες τιμές και τη δεξιά περιοχή όπου εμφανίζονται όλες οι επιλεγμένες τιμές. Για να επιλέξει κάποια τιμή απλά μαρκάρει το επίθετο που θέλει και με το μονό βελάκι αυτό εμφανίζεται και στη δεξιά στήλη. Υπάρχει η δυνατότητα να επιλέξει ταυτόχρονα πολλαπλές τιμές ή με το διπλό βελάκι να επιλέξει όλες τις τιμές. Αντίστοιχα με τα κουμπιά Remove και Remove All μπορεί να διαγράψει μία ή όλες τις τιμές από τη δεξιά λίστα. Το αποτέλεσμα που θα πάρει ο διαχειριστής

ενεργοποιώντας το κουμπί Ok είναι μια αναφορά με τα στοιχεία των χρηστών που είναι διατηρούν λογαριασμό στην εφαρμογή iTestU (Εικόνα 73).



Εικόνα 72. Παραμετρικό ερώτημα Crystal Report

Κωδ.	Όνομα	Επίθετο	Διεύθυνση email	Φύλο	Διαχειριστής	Ημερομηνία εγγραφής
1	Άκης	Βοΐλας	akisvoilas@yahoo.com	Άρρεν	<input checked="" type="checkbox"/>	15/7/2016 11:01:40πμ
4	Μαρία	Οικονόμου	mary@yahoo.com	Θήλυ	<input type="checkbox"/>	4/8/2016 10:32:58πμ
6	Ιωάννης	Γεωργίου	johngeo@gmail.gr	Άρρεν	<input type="checkbox"/>	4/8/2016 11:12:26πμ
9	Ειρήνη	Βοΐλα	irene@yahoo.gr	Θήλυ	<input type="checkbox"/>	6/8/2016 6:17:03μμ
10	TestName	TestSurname	test@test.com	Άρρεν	<input type="checkbox"/>	
17	Παύλος	Δοβλές		Άρρεν	<input type="checkbox"/>	
18	Αναστασία	Οικονόμου	anastasia@gmail.gr	Θήλυ	<input type="checkbox"/>	
19	Δοκίμιος	Δοκιμόπουλας	test@test.com	Άρρεν	<input type="checkbox"/>	20/8/2016 9:17:52μμ
20	Γιώργος	Νικολάου	geonik@diek.gr	Άρρεν	<input type="checkbox"/>	2/9/2016 12:33:44μμ

Εικόνα 73. Αναφορά με στοιχεία χρηστών

7 Συμπεράσματα

Στην μεταπτυχιακή αυτή διατριβή παρουσιάστηκε η ανάλυση και ο σχεδιασμός ενός εκπαιδευτικού λογισμικού που θα μπορούσε να χρησιμοποιηθεί τόσο από σπουδαστές όσο και από καθηγητές. Το όραμά μας είναι να αποτελέσει ένα βοηθητικό εργαλείο που θα γίνει μέρος της εκπαιδευτικής διαδικασίας σε Ινστιτούτα Επαγγελματικής Κατάρτισης (ΙΕΚ) αλλά και σε Κέντρα Ελευθέρων Σπουδών (ΚΕΣ).

Το εκπαιδευτικό λογισμικό iTestU σχεδιάστηκε με σκοπό να αποτελέσει ένα βοηθητικό εργαλείο για σπουδαστές της μεταδευτεροβάθμιας εκπαίδευσης να δοκιμάσουν τη γνώση τους σε θέματα που σχετίζονται με βασικές έννοιες της πληροφορικής πριν δώσουν τις εξετάσεις πιστοποίησης του ΕΟΠΠΕΠ (Εθνικός Οργανισμός Πιστοποίησης Προσόντων και Επαγγελματικού Προσανατολισμού). Μέσα από το περιβάλλον εξέτασης έχουν τη δυνατότητα να αξιολογηθούν τόσο σε πρακτικά θέματα όσο και σε θεωρητικές έννοιες. Πιο συγκεκριμένα με τη βοήθεια του εκπαιδευτικού λογισμικού iTestU τους δίνεται η δυνατότητα να εκτελέσουν σε πραγματικό περιβάλλον ασκήσεις επεξεργασίας κειμένου με το λογισμικό Microsoft Word, ασκήσεις υπολογιστικών φύλλων με το λογισμικό Microsoft Excel και ασκήσεις παρουσιάσεων με το λογισμικό PowerPoint. Επιπλέον έχουν τη δυνατότητα να απαντήσουν σε θεωρητικές ερωτήσεις πολλαπλής επιλογής.

Το λογισμικό iTestU σχεδιάστηκε και αναπτύχθηκε με σκοπό να αποτελέσει ένα βοήθημα και για τον καθηγητή. Μέσα από το λογισμικό θα μπορούσε δημιουργήσει εξετάσεις αξιολόγησης για τους σπουδαστές του, να τους εξετάσει, να τους αξιολογήσει και να καταγράψει την πρόοδο τους.

Για την ανάπτυξη του λογισμικού χρησιμοποιήσαμε την αντικειμενοστραφή προσέγγιση σε όλες τις φάσεις και συγκεκριμένα στην έναρξη του έργου, κατά τη φάση της εκπόνησης μελέτης, κατά τη φάση υλοποίησης και τέλος στη φάση μετάβασης. Για την ανάλυση και το σχεδιασμό του λογισμικού χρησιμοποιήσαμε την γλώσσα μοντελοποίησης UML και για την υλοποίηση τη γλώσσα προγραμματισμού VB.NET. Η ανάπτυξη και ο αρχικός έλεγχος έγινε στο ολοκληρωμένο περιβάλλον Visual Studio 2012 χρησιμοποιώντας το .NET framework 4.5. Ο τελικός έλεγχος της εφαρμογής έγινε χρησιμοποιώντας πραγματικά σενάρια χρήσης τόσο από τον υποφαινόμενο όσο και από πραγματικούς σπουδαστές που δοκίμασαν την εφαρμογή. Για τη δημιουργία του τελικού πακέτου διανομής της εφαρμογής iTestU χρησιμοποιήσαμε το λογισμικό Install Shield.

Το τελικό παραδοτέο είναι η παρούσα μεταπτυχιακή διατριβή και ένα CD με τον κώδικα που αναπτύξαμε για την υλοποίηση της εφαρμογής.

8 Βιβλιογραφικές - Διαδικτυακές Πηγές

8.1 Βιβλιογραφία

Βεσκούκης Βασίλειος. *Σημειώσεις για το μάθημα Τεχνολογία Λογισμικού.* Πειραιάς: Πανεπιστημιακές σημειώσεις, 2000

Βίρβου Μαρία. *Ειδικά Θέματα Τεχνολογίας Λογισμικού.* Πανεπιστημιακές σημειώσεις, 2013.

Halvorson Michael. *Microsoft Visual Basic 2010 Step by Step.* Redmond Washington: Microsoft Press, 2010

Petroutsos Evangelos. *Πλήρες Εγχειρίδιο της Microsoft Visual Basic.* Αθήνα: Εκδόσεις Μ. Γκιούρδας, 2009

Stephens Rod. *Visual Basic .NET Ανάπτυξη Εφαρμογών Βάσεων Δεδομένων.* Αθήνα: Εκδόσεις Μ. Γκιούρδας, 2003

IBM Rational Unified Process. *Best Practises for Software Development Teams.* Rational Software White paper, 1998

8.2 Διαδίκτυο

<https://msdn.microsoft.com>

<https://www.edrawsoft.com/embed-word-vbnet.php>

http://www.tutorialspoint.com/vb.net/vb.net_excel_sheet.htm

<https://notesbyparth.wordpress.com/2014/02/13/ms-powerpoint-automation-vb-net-or-create-powerpoint-presentation-programatically-vb-net/>

<https://siddharthrouth.wordpress.com/vb-net-and-excel/>

<http://www.c-sharpcorner.com/UploadFile/cb88b2/simple-steps-to-create-setup-file-using-install-shield-le-fr/>

Παράρτημα

Στο πρώτο μέρος του παραρτήματος γίνεται μια αναφορά και περιγραφή των εργαλείων που χρησιμοποιήθηκαν για να αναπτυχθεί το εκπαιδευτικό πληροφοριακό σύστημα iTestU. Στο δεύτερο μέρος παρουσιάζεται ο κώδικας της εφαρμογής όπως αναπτύχθηκε στη γλώσσα προγραμματισμού VB.NET.

Εργαλεία και τεχνολογίες

Microsoft Visual Studio Ultimate 2012

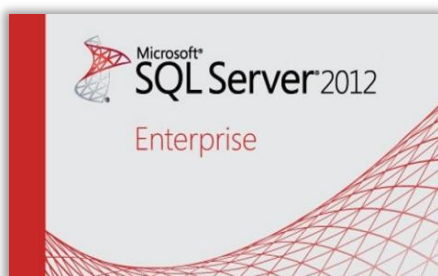


Το Visual Studio Ultimate 2012 είναι ένα ενοποιημένο περιβάλλον ανάπτυξης (Integrated Development Environment - IDE) που χρησιμοποιείται τόσο στην ανάπτυξη διαδικτυακών εφαρμογών όσο και εφαρμογών γραφείου καθώς και διαδικτυακών υπηρεσιών (web services). Κύριο χαρακτηριστικό στο περιβάλλον ανάπτυξης του VS 2012 είναι ότι παρέχει ένα περιβάλλον που είναι κοινό σε όλες τις γλώσσες και τεχνολογίες.

Το περιβάλλον IDE παρέχει εργαλεία για την σχεδίαση, την εκτέλεση και την ανίχνευση λαθών για τις εφαρμογές που αναπτύσσονται. Μπορούμε να αναπτύξουμε μια λειτουργική εφαρμογή γραφείου οδηγούμενη από δεδομένα ή αντίστοιχα μια διαδικτυακή εφαρμογή, γράφοντας ελάχιστες γραμμές κώδικα. Αυτό το επιτυγχάνεται εύκολα γιατί απλοποιεί ορισμένες εργασίες με τη χρήση οδηγών. Επίσης παρέχει εργαλεία για τη σχεδίαση του σχήματος μια βάσης δεδομένων αλλά και για το χειρισμό της. Τέλος δεν μπορούμε να παραλείψουμε το μεγάλο πλήθος έτοιμων αντικειμένων που μας παρέχεται για την ανάπτυξη διεπαφών χρήστη (user interface).

Οι βασικές ενσωματωμένες γλώσσες προγραμματισμού στο VS 2012 είναι η C#, η VB.NET και η ASP.NET. Ωστόσο υποστηρίζει και πλήθος άλλων γλωσσών όπως C++, C, F#, Python και Ruby.

Microsoft SQL Server 2012



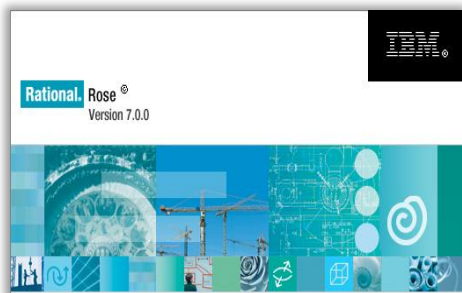
Το Microsoft SQL Server είναι το κορυφαίο σύστημα διαχείρισης βάσεων δεδομένων της Microsoft. Ο SQL Server χρησιμοποιεί έναν κεντρικό διακομιστή (server) βάσεων δεδομένων για την πρόσβαση και τη διαχείριση της ΒΔ.

Αντίθετα με την Microsoft Access δεν υπόκειται σε περιορισμό στο μέγεθος της ΒΔ. Ο διακομιστής παρέχει καλύτερες δυνατότητες κλειδώματος των αρχείων σε σχέση με την Access.

Αυτό σημαίνει ότι ο SQL Server μπορεί να χειριστεί οποιονδήποτε αριθμό ταυτόχρονων χρηστών με ελάχιστες συγκρούσεις.

Ο SQL Server χρησιμοποιεί μια παραλλαγή της γλώσσας SQL για την επικοινωνία του με την εφαρμογή που ονομάζεται Transact SQL (T-SQL). Επίσης περιλαμβάνει μια σειρά πρόσθετων υπηρεσιών και εργαλείων πάνω από το κεντρικό ΣΔΒΔ, όπως είναι τα Replication Services, Analysis Services, Integration Services, Reporting Services και πολλά ακόμα.

IBM Rational Rose Enterprise Edition



Ένα ακόμα λογισμικό που χρησιμοποιήθηκε, κατά τη διάρκεια ανάπτυξης του εκπαιδευτικού πληροφοριακού συστήματος, είναι το IBM Rational Rose. Το λογισμικό χρησιμοποιήθηκε στις φάσεις της εκπόνησης μελέτης και της κατασκευής για το σχεδιασμό των διαγραμμάτων της Unified Modeling Language - UML. Το IBM Rational Rose θεωρείται ένα από τα πιο διαδεδομένα CASE εργαλεία στην μοντελοποίηση συστημάτων λογισμικού και όχι μόνο.

Υποστηρίζει όλους τους τύπους διαγραμμάτων για την έκδοση 2 της UML και ακόμα και την ελεύθερη σχεδίαση διαγραμμάτων. Ένα ακόμα σημαντικό χαρακτηριστικό του, είναι η συγχώνευση διαγραμμάτων, στοιχείο το οποίο επιτρέπει την παράλληλη ανάπτυξη μοντέλων από διαφορετικές ομάδες σχεδίασης. Τέλος, δεν μπορεί να παραληφθεί η δυνατότητα για αυτόματη παραγωγή κώδικα από τα μοντέλα σχεδίασης. Το IBM Rational Rose υποστηρίζει τη μετατροπή ενός μοντέλου UML στις γλώσσες Java , C++ και CORBA. Επιπρόσθετα υποστηρίζει και την αντιστροφή διαδικασία από C# σε UML.

Microsoft Visio 2010



Το Microsoft Visio είναι μια εφαρμογή διανυσματικής σχεδίασης που παρέχει σύγχρονα σχήματα και πρότυπα για την δημιουργία διαγραμμάτων. Το λογισμικό αυτό αποτελεί μέρος της σουίτας εφαρμογών γραφείου Microsoft Office. Παρέχει δεκάδες βιβλιοθήκες με γραφικά ομαδοποιημένα σε θεματικές κατηγορίες που σχετίζονται με διάφορους κλάδους και τεχνολογίες της πληροφορικής, τη μοντελοποίηση, τη σχεδίαση UI, της δόμηση και την αρχιτεκτονική, τη διαχείριση ανθρωπίνου δυναμικού, τη διαχείριση έργου και άλλα.

Flexera InstallShield



Το λογισμικό InstallShield είναι ένα λογισμικό για την δημιουργία πακέτων εγκατάστασης και διανομής λογισμικού. Χρησιμοποιείται κυρίως για τη δημιουργία πακέτων εγκατάστασης εφαρμογών σε desktop υπολογιστές με λειτουργικό σύστημα Windows αλλά και σε πλήθος φορητών και κινητών συσκευών. Το χρησιμοποιήσαμε ως πρόσθετο εργαλείο στο Visual Studio 2012 κατά τη διάρκεια της φάσης της μετάβασης για να δημιουργήσουμε το πακέτο εγκατάστασης για το λογισμικό iTestU

Microsoft Access 2010



Η Microsoft Access είναι ένα από τα πιο δημοφιλή συστήματα διαχείρισης βάσεων δεδομένων. Το λογισμικό αυτό αποτελεί τμήμα της σουίτας εφαρμογών γραφείου Microsoft Office και παρέχει τη δυνατότητα εύκολης και γρήγορης δημιουργίας σχεσιακών ΒΔ.

Οι ΒΔ της Access είναι σχετικά απλές και κάθε ΒΔ αποθηκεύεται σε ένα ξεχωριστό αρχείο. Αυτός είναι ο λόγος που υπόκεινται σε ορισμένους περιορισμούς οι οποίοι τίθενται από το σύστημα αρχείων του υπολογιστή στο οποίο τρέχουν.

Το βασικό περιβάλλον της Microsoft Access περιέχει καρτέλες με τα αντικείμενα που μπορούμε να δημιουργήσουμε μέσα στη ΒΔ. Τέτοια αντικείμενα είναι οι πίνακες (tables), τα ερωτήματα (queries), οι φόρμες (forms) και οι εκθέσεις (reports). Επίσης το περιβάλλον διαχείρισης παρέχει εργαλεία για την δημιουργία μακροεντολών (macros) και λειτουργικών μονάδων (modules) για τη συγγραφή κώδικα στη γλώσσα VBA.

Οι δυνατότητες της Microsoft Access αν και είναι περιορισμένες σε σχέση με άλλα ανταγωνιστικά ΣΔΒΔ που κυκλοφορούν στην αγορά, εξυπηρετούν πλήρως τις ανάγκες για ανάπτυξη εφαρμογών μικρής ή μεσαίας κλίμακας. Σε κάποιες περιπτώσεις μάλιστα προβάλλει και ως ιδανική λύση εξαιτίας των παρακάτω λόγων:

- Παρέχει ένα εύχρηστο και φιλικό περιβάλλον
- Είναι εύκολη στην εκμάθηση λόγω του γραφικού περιβάλλοντος
- Έχει περιορισμένες απαιτήσεις σε πόρους
- Η συντήρηση δεν απαιτεί εξειδικευμένες γνώσεις
- Συνεργάζεται με άλλα λογισμικά του Office και τεχνολογίες όπως XML, ODBC

Ο κώδικας της εφαρμογής

```

-----
'Project:      Μεταπτυχιακή Διατριβή
'Filename:     Splash.vb
'Author:       Άκης Βοΐλας
'Version:      1.0.0
'Date:         21/5/2016
'Description:  Αποτελεί την splash screen, την πρώτη οθόνη της εφαρμογής
-----

Public Class Splash
    'Εμφανίζει το παράθυρο εισαγωγής διαπιστευτηρίων (σύνδεση/εγγραφή)
    Private Sub btnEntryApp_Click(sender As Object, e As EventArgs) Handles btnEntryApp.Click
        Credential.Show()
        Me.Hide()
    End Sub

    'Τερματίζει την εφαρμογή και αποδεσμεύει πόρους από τη Διαχείριση εργασιών Windows
    Private Sub btnExitApp_Click(sender As Object, e As EventArgs) Handles btnExitApp.Click
        Dim result As Integer = MsgBox("Είστε σίγουρος/η ότι θέλετε να" + vbNewLine +
"τερματίσετε την εφαρμογή iTestU;", MsgBoxStyle.OkCancel Or MsgBoxStyle.Exclamation, "iTestU")
        'Αν πατηθεί Ok τερματίζεται αν πατηθεί cancel επιστρέφω στην οθόνη
        If result = DialogResult.OK Then
            Me.Close()
            Process.GetCurrentProcess().Kill()
        End If
    End Sub
End Class

-----
'Project:      Μεταπτυχιακή Διατριβή
'Filename:     Credential.vb
'Author:       Άκης Βοΐλας
'Version:      1.0.0
'Date:         4/8/2016
'Description:  Είναι η φόρμα για σύνδεση/εγγραφή του χρήστη
-----

'Για τη σύνδεση με ΒΔ
Imports System.Data.OleDb
Public Class Credential
    Dim Mycn As OleDbConnection
    Dim Command As OleDbCommand
    Dim icount As Integer
    Dim SQLstr, SQLstr2, SQLstr3 As String
    Dim SQLstrName, SQLstrSurname, SQLstrIsAdmin As String

    'Εγγραφή χρήστη στην εφαρμογή
    Private Sub btnSignUp_Click(sender As Object, e As EventArgs) Handles btnSignUp.Click
        'Αποθηκεύει το φύλο
        Dim gender As String = "Αρρεν"
        If rbWoman.Checked = True Then
            gender = "Θήλυ"
        End If
        'Ελέγχει αν έχουν συμπληρωθεί τα υποχρεωτικά πεδία
        If (txtName.Text <> "" And txtSurname.Text <> "" And txtUsername.Text <> ""
            And txtPassword.Text <> "") Then

            Try
                Mycn = New OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;
                    Data Source=C:\Users\Akis\Documents\Test2U.accdb")

                Mycn.Open()
                'Τα δεδομένα της φόρμας εγγραφής αποθηκεύονται στα πεδία του πίνακα tblMembers
                SQLstr = "INSERT INTO tblMembers
                    (Name,Surname,Email,Gender,Username,Passcode,Occupation)
                    VALUES ('" & txtName.Text & "', '" & txtSurname.Text & "',
                    '" & txtEmail.Text & "', '" & gender & "', '" & txtUsername.Text & "',

```

```

        "" & txtPassword.Text & "',' & cbOccupation.Text & ''")
    Command = New OleDbCommand(SQLstr, Mycn)
    icount = Command.ExecuteNonQuery
    MessageBox.Show("Η εγγραφή σας ολοκληρώθηκε," & vbNewLine & "για να συνδεθείτε
    επιστρέψτε" & vbNewLine & "στην καρτέλα Σύνδεση",
    "Ολοκλήρωση εγγραφής", MessageBoxButtons.OK, MessageBoxIcon.Information)

    'Αν συμβεί πρόβλημα με τη σύνδεση της ΒΔ
Catch ex As Exception
    MsgBox(ex)

Finally
    'Κλείνω τη σύνδεση με τη ΒΔ και καθαρίζω τα πεδία της φόρμας εγγραφής
    Mycn.Close()
    ClearFormFields()
End Try

Else
    MessageBox.Show("Πρέπει να συμπληρώσετε όλα τα" & vbNewLine & "υποχρεωτικά πεδία της
    φόρμας", "Σφάλμα εγγραφής", MessageBoxButtons.OK, MessageBoxIcon.Warning)
End If
End Sub

'Σύνδεση χρήστη στην εφαρμογή
Private Sub btnLogin_Click(sender As Object, e As EventArgs) Handles btnLogin.Click
    Dim response As Integer

    Try
        Mycn = New OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;
        Data Source=C:\Users\Akis\Documents\Test2U.accdb")

        Mycn.Open()
        'Αν έχω δώσει έγκυρα διαπιστευτήρια επιστρέφω το μοναδικό MemberID
        SQLstr = "SELECT tblMembers.MemberID
        FROM tblMembers
        WHERE (tblMembers.Username='" & Me.txtUser.Text & "' and
        tblMembers.Passcode='" & Me.txtPass.Text & "' )"
        Command = New OleDbCommand(SQLstr, Mycn)

        If (Command.ExecuteScalar = vbNullString) Then
            MessageBox.Show("Μη έγκυρα διαπιστευτήρια", "Έλεγχος σύνδεσης",
            MessageBoxButtons.OK, MessageBoxIcon.Error)

        Else
            response = MsgBox("Τα στοιχεία σας βρέθηκαν",
            MsgBoxStyle.Information, "Έλεγχος σύνδεσης")
            value = Int32.Parse(Command.ExecuteScalar) 'MemberID

            'Ο χρήστης έχει συνδεθεί, να καταγράψω την σύνδεση στον πίνακα tblUsers
            If response = MsgBoxResult.Ok Then

                'Ελέγγω αν αυτός που συνδέθηκε έχει δικαιώματα administrator
                SQLstrIsAdmin = "SELECT tblMembers.IsAdministrator
                FROM tblMembers
                WHERE (tblMembers.MemberID=" & value & ")"
                Command = New OleDbCommand(SQLstrIsAdmin, Mycn)
                IsAdmin = CInt(Command.ExecuteScalar) ' αν επιστρέψει -1 είναι admin

                'Κρατώ το μικρό όνομα αυτού που συνδέθηκε
                SQLstrName = "SELECT tblMembers.Name
                FROM tblMembers
                WHERE (tblMembers.MemberID=" & value & ")"
                Command = New OleDbCommand(SQLstrName, Mycn)
                memberName = CStr(Command.ExecuteScalar)

                'Κρατώ το επίθετο αυτού που συνδέθηκε
                SQLstrSurname = "SELECT tblMembers.Surname
                FROM tblMembers
                WHERE (tblMembers.MemberID=" & value & ")"

```



```

Command = New OleDbCommand(SQLstrSurname, Mycn)
memberSurname = CStr(Command.ExecuteScalar)

'Dημιουργώ μια νέα εγγραφή στον πίνακα tblUsers για τον χρήστη που συνδέθηκε
SQLstr2 = "INSERT INTO tblUsers(IdStud)VALUES(" & value & ")"
Command = New OleDbCommand(SQLstr2, Mycn)
icount = Command.ExecuteNonQuery

'Παίρνω το UserId του ενεργού χρήστη που τρέχει την εφαρμογή
SQLstr3 = "SELECT tblUsers.UserId
          FROM tblUsers
          WHERE (tblUsers.IdStud=" & value & ")
          ORDER BY tblUsers.DateActivity DESC"
Command = New OleDbCommand(SQLstr3, Mycn)
ids = Int32.Parse(Command.ExecuteScalar) 'επιστρέφει το UserId ενεργού

    Me.Hide()
    MainPanel.Show()
End If
End If

Catch ex As Exception
    MsgBox(ex)
Finally
    'Κλείνω τη συνδεση με τη ΒΔ και καθαρίζω τα στοιχεία της φόρμας
    Mycn.Close()
    txtUsername.Text = ""
    txtPassword.Text = ""
End Try
End Sub

'Τερματίζει την εφαρμογή και αποδεσμεύει πόρους απο τη Διαχείριση εργασιών Windows
Private Sub btnExitApp_Click(sender As Object, e As EventArgs) Handles btnExitApp.Click
    Dim result As Integer = MsgBox("Είστε σίγουρος/η ότι θέλετε να" + vbNewLine +
        "τερματίσετε την εφαρμογή iTestU;", MsgBoxStyle.OkCancel
        Or MsgBoxStyle.Exclamation, "iTestU")
    'Αν πατηθεί Ok τερματίζεται αν πατηθεί cancel επιστρέφω στην οθόνη
    If result = DialogResult.OK Then
        Me.Close()
        Process.GetCurrentProcess().Kill()
    End If
End Sub

'Καθαρίζει τα πεδία της φόρμας εγγραφής
Private Sub btnClear_Click(sender As Object, e As EventArgs) Handles btnClear.Click
    ClearFormFields()
End Sub

'Υπορουτίνα αρχικοποίησης πεδίων της φόρμας εγγραφής
Sub ClearFormFields()
    txtName.Text = vbNullString
    txtSurname.Text = vbNullString
    txtEmail.Text = vbNullString
    txtUsername.Text = vbNullString
    txtPassword.Text = vbNullString
    cbOccupation.Text = "--- Επιλέξτε ή εισάγετε ---"
End Sub

End Class

```

```

'-----
'Project:      Μεταπτυχιακή Διατριβή
'Filename:     MainPanel.vb
'Author:       Άκης Βοΐλας
'Version:      1.0.0
'Date:         30/8/2016
'Description:  Το κεντρικό παράθυρο επιλογών της εφαρμογής
'-----
Public Class MainPanel
    Private Sub MainPanel_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        'Αν έχεις δικαιώματα Admin ενεργοποίησε επιπλέον λειτουργίες
        If IsAdmin = -1 Then
            GroupBoxAdmin.Visible = True
            GroupBoxAdmin.Enabled = True
        Else
            Me.Height = 415
            Panel1.Height = 230
        End If
    End Sub

    'Εκκινεί το παράθυρο με τις επιλογές εξέτασης
    Private Sub PictureStartExams_Click(sender As Object, e As EventArgs) Handles
PictureStartExams.Click
        SectionChoise.Show()
        Me.Hide()
        Me.Close()
    End Sub
    Private Sub PictureStartExams_MouseHover(sender As Object, e As EventArgs) Handles
PictureStartExams.MouseHover
        PictureStartExams.Image = My.Resources.start2
    End Sub
    Private Sub PictureStartExams_MouseLeave(sender As Object, e As EventArgs) Handles
PictureStartExams.MouseLeave
        PictureStartExams.Image = My.Resources.start1
    End Sub

    'Ανοίγει το παράθυρο Στατιστικά Χρήστη
    Private Sub PictureUserStats_Click(sender As Object, e As EventArgs) Handles
PictureUserStats.Click
        MemberStatistics.ShowDialog()
    End Sub
    Private Sub PictureUserStats_MouseHover(sender As Object, e As EventArgs) Handles
PictureUserStats.MouseHover
        PictureUserStats.Image = My.Resources.UserStat2
    End Sub
    Private Sub PictureUserStats_MouseLeave(sender As Object, e As EventArgs) Handles
PictureUserStats.MouseLeave
        PictureUserStats.Image = My.Resources.UserStat1
    End Sub

    'Ανοίγει το παράθυρο Διαχείριση Λογαριασμού
    Private Sub PictureEditProfile_Click(sender As Object, e As EventArgs) Handles
PictureEditProfile.Click
        EditProfile.Dispose() 'για να τρέχει πάντα η μέθοδος Load()
        EditProfile.ShowDialog()
    End Sub
    Private Sub PictureEditProfile_MouseHover(sender As Object, e As EventArgs) Handles
PictureEditProfile.MouseHover
        PictureEditProfile.Image = My.Resources.EditProfile2
    End Sub
    Private Sub PictureEditProfile_MouseLeave(sender As Object, e As EventArgs) Handles
PictureEditProfile.MouseLeave
        PictureEditProfile.Image = My.Resources.EditProfile1
    End Sub

    'Ανοίγει το παράθυρο της Βοήθειας για την εφαρμογή
    Private Sub PictureHelp_Click(sender As Object, e As EventArgs) Handles PictureHelp.Click
        MsgBox("υπό κατασκευή!!!")
    End Sub
End Class

```

```

Private Sub PictureHelp_MouseHover(sender As Object, e As EventArgs) Handles
PictureHelp.MouseHover
    PictureHelp.Image = My.Resources.Help2
End Sub
Private Sub PictureHelp_MouseLeave(sender As Object, e As EventArgs) Handles
PictureHelp.MouseLeave
    PictureHelp.Image = My.Resources.Help1
End Sub

'Ανοίγει τη Διαχείριση Ερωτήσεων
Private Sub PictureEditExam_Click(sender As Object, e As EventArgs) Handles
PictureEditExam.Click
    ManageQuestion.ShowDialog()
End Sub
Private Sub PictureEditExam_MouseHover(sender As Object, e As EventArgs) Handles
PictureEditExam.MouseHover
    PictureEditExam.Image = My.Resources.exam2
End Sub
Private Sub PictureEditExam_MouseLeave(sender As Object, e As EventArgs) Handles
PictureEditExam.MouseLeave
    PictureEditExam.Image = My.Resources.exam1
End Sub

'Ανοίγει τα Στατιστικά του Διαχειριστή
Private Sub PictureAdmin_Click(sender As Object, e As EventArgs) Handles PictureAdmin.Click
    AdminStatistics.ShowDialog()
End Sub
Private Sub PictureAdmin_MouseHover(sender As Object, e As EventArgs) Handles
PictureAdmin.MouseHover
    PictureAdmin.Image = My.Resources.AdminStats2
End Sub
Private Sub PictureAdmin_MouseLeave(sender As Object, e As EventArgs) Handles
PictureAdmin.MouseLeave
    PictureAdmin.Image = My.Resources.AdminStats1
End Sub

'Ανοίγει τη Διαχείριση Χρηστών
Private Sub PictureEditUsers_Click(sender As Object, e As EventArgs) Handles
PictureEditUsers.Click
    ManageUsers.ShowDialog()
End Sub
Private Sub PictureEditUsers_MouseHover(sender As Object, e As EventArgs) Handles
PictureEditUsers.MouseHover
    PictureEditUsers.Image = My.Resources.EditUsers2
End Sub
Private Sub PictureEditUsers_MouseLeave(sender As Object, e As EventArgs) Handles
PictureEditUsers.MouseLeave
    PictureEditUsers.Image = My.Resources.EditUsers1
End Sub

'Εμφανίζει το παράθυρο Σχετικά...
Private Sub PictureAbout_Click(sender As Object, e As EventArgs) Handles PictureAbout.Click
    About.ShowDialog()
End Sub
Private Sub PictureAbout_MouseHover(sender As Object, e As EventArgs) Handles
PictureAbout.MouseHover
    PictureAbout.Image = My.Resources.AboutUs2
End Sub
Private Sub PictureAbout_MouseLeave(sender As Object, e As EventArgs) Handles
PictureAbout.MouseLeave
    PictureAbout.Image = My.Resources.AboutUs
End Sub

'Εμφανίζει αναφορά με τα προσωπικά στοιχεία των χρηστών
Private Sub PictureUserInfo_Click(sender As Object, e As EventArgs) Handles
PictureUserInfo.Click
    MembersReport.ShowDialog()
End Sub

```

```

Private Sub PictureUserInfo_MouseHover(sender As Object, e As EventArgs) Handles
PictureUserInfo.MouseHover
    PictureUserInfo.Image = My.Resources.InfoUser2
End Sub

Private Sub PictureUserInfo_MouseLeave(sender As Object, e As EventArgs) Handles
PictureUserInfo.MouseLeave
    PictureUserInfo.Image = My.Resources.InfoUser1
End Sub

'Τερματίζει την εφαρμογή και αποδεσμεύει πόρους από τη Διαχείριση εργασιών Windows
Private Sub PictureExit_Click(sender As Object, e As EventArgs) Handles PictureExit.Click
    Dim result As Integer = MsgBox("Είστε σίγουρος/η ότι θέλετε να" + vbNewLine +
"τερματίσετε την εφαρμογή iTestU;", MsgBoxStyle.OkCancel Or MsgBoxStyle.Exclamation, "iTestU -
Έξοδος")
    'Αν πατηθεί Ok τερματίζεται αν πατηθεί cancel επιστρέφω στην οθόνη
    If result = DialogResult.OK Then
        Me.Close()
        Process.GetCurrentProcess().Kill()
    End If
End Sub

Private Sub PictureExit_MouseHover(sender As Object, e As EventArgs) Handles
PictureExit.MouseHover
    PictureExit.Image = My.Resources.Exodus2
End Sub

Private Sub PictureExit_MouseLeave(sender As Object, e As EventArgs) Handles
PictureExit.MouseLeave
    PictureExit.Image = My.Resources.Exodus1
End Sub
End Class

```

```

-----
'Project:      Μεταπτυχιακή Διατριβή
'Filename:     SectionChoise.vb
'Author:       Άκης Βοΐλας
'Version:     1.0.0
'Date:        22/5/2016
'Description:  Παράθυρο από το οποίο επιλέγουμε το αντικείμενο και κατηγορία εξέτασης
-----

```

```

'Για τη σύνδεση με ΒΔ
Imports System.Data.OleDb

```

```

Public Class SectionChoise
    Dim Mycn As OleDbConnection
    Dim Command As OleDbCommand
    Dim icount As Integer
    Dim SQLstr, SQLstr2, SQLstr3 As String
    Dim SelectionCategory As Integer 'Η κατηγορία ερωτήσεων πχ 2~>word, 3~>excel κτλ

    'Αρχικοποίηση των πλαισίων επιλογής
    Private Sub SectionChoise_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        ComboBox1.Focus() 'Εστίαση στο Combobox
        ComboBox1.SelectedIndex = 0
        ComboBox2.SelectedIndex = 0
    End Sub

    'Ρυθμίζω τι εμφανίζει το δεύτερο πλαίσιο σε κάθε περίπτωση
    Private Sub ComboBox1_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
ComboBox1.SelectedIndexChanged
        ComboBox2.Items.Clear()
        If ComboBox1.SelectedIndex = 0 Then
            ComboBox2.Items.Add("Επεξεργασία Κειμένου")
            ComboBox2.Items.Add("Υπολογιστικά Φύλλα")
            ComboBox2.Items.Add("Παρουσιάσεις")
            ComboBox2.SelectedIndex = 0
        Else

```

```

        ComboBox2.Items.Add("Θεωρία Υπολογιστών")
        ComboBox2.SelectedIndex = 0
    End If
End Sub

'Επιλογή αντικειμένου/ενότητας για εξέταση
Private Sub btnStartTest_Click(sender As Object, e As EventArgs) Handles btnStartTest.Click

    Dim valueOfCombo As String = ComboBox2.SelectedItem.ToString()

    Select Case valueOfCombo
        Case "Θεωρία Υπολογιστών"
            SelectionCategory = 1

            'Κλήση υπορουτίνας για την εγγραφή στον πίνακα tblSession
            InsertRecordTotblSession()

            fmTheory.Show()
            Me.Hide()

        Case "Επεξεργασία Κειμένου"
            SelectionCategory = 2

            'Κλήση υπορουτίνας για την εγγραφή στον πίνακα tblSession
            InsertRecordTotblSession()

            fmWord1.Show()
            Me.Hide()

        Case "Υπολογιστικά Φύλλα"
            SelectionCategory = 3 'Επιλογή της κατηγορίας Υπολογιστικά φύλλα

            'Κλήση υπορουτίνας για την εγγραφή στον πίνακα tblSession
            InsertRecordTotblSession()

            fmExcel1.Show()
            Me.Hide()

        Case "Παρουσιάσεις"
            SelectionCategory = 4 'Επιλογή της κατηγορίας Παρουσιάσεις

            'Κλήση υπορουτίνας για την εγγραφή στον πίνακα tblSession
            InsertRecordTotblSession()

            fmPP1.Show()
            Me.Hide()
        Case Else
            MsgBox("Επιλέξτε μια ενότητα", MsgBoxStyle.Exclamation, "iTestU - Επιλογή
Ενότητας")
    End Select
End Sub

'Τερματίζει την εφαρμογή και αποδεσμεύει πόρους από τη Διαχείριση εργασιών Windows
Private Sub btnExitApp_Click(sender As Object, e As EventArgs) Handles btnExitApp.Click
    Dim result As Integer = MsgBox("Είστε σίγουρος/η ότι θέλετε να" + vbNewLine
        + "τερματίσετε την εφαρμογή iTestU;", MsgBoxStyle.OkCancel
        Or MsgBoxStyle.Exclamation, "iTestU")
    'Αν πατηθεί Ok τερματίζεται αν πατηθεί cancel επιστρέφω στην οθόνη
    If result = DialogResult.OK Then
        Me.Close()
        Process.GetCurrentProcess().Kill()
    End If
End Sub

'Υπορουτίνα που γράφει στον πίνακα tblSession την κατηγορία και το αναγνωριστικό του χρήστη
Sub InsertRecordTotblSession()
    Try
        Mycn = New OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;
            Data Source=C:\Users\Akis\Documents\Test2U.accdb")

        Mycn.Open()
    End Try
End Sub

```

```

'Εισάγω επιλογή ενότητας και χρήστη στα πεδία του πίνακα tblSession
SQLstr = "INSERT INTO tblSession(IdCategory,IDS)
        VALUES(" & SelectionCategory & ", " & ids & ")"
Command = New OleDbCommand(SQLstr, Mycn)
icount = Command.ExecuteNonQuery

'Παίρνω το τελευταίο SessionID που δημιουργήθηκε από την επιλογή της ενότητας
SQLstr2 = "SELECT LAST(tblSession.SessionID) FROM tblSession"
Command = New OleDbCommand(SQLstr2, Mycn)
idszn = Int32.Parse(Command.ExecuteScalar) ' επιστρέφει το SessionID ως ακέραιο

'Αν συμβεί πρόβλημα με τη σύνδεση της ΒΔ
Catch ex As Exception
    MsgBox(ex)

Finally
    'Κλείνω τη σύνδεση με τη ΒΔ
    Mycn.Close()
End Try
End Sub
End Class

'-----
'Project:      Μεταπτυχιακή Διατριβή
'Filename:     fmTheory.vb
'Author:       Άκης Βοΐλας
'Version:      1.0.0
'Date:         18/8/2016
'Description:  Είναι η φόρμα για θεωρητικές ερωτήσεις
'-----

'Για τη σύνδεση με ΒΔ
Imports System.Data.OleDb

Public Class fmTheory
    Dim correct As Integer
    Dim myanswer As Integer 'κρατάει την απάντηση του χρήστη
    Dim activeQID As Integer 'κρατάω ID της ενεργής ερώτησης
    Dim LastQuestionID As Integer = 101 'κρατάει το τελευταίο ID θεωρητικής ερώτησης
    Dim NumQuestion As Integer = 1 'αριθμός στο label της φόρμας
    Dim NumOfQs As Integer 'πλήθος ερωτήσεων

    Dim Mycn As OleDbConnection
    Dim Command As OleDbCommand
    Dim icount As Integer

    Dim SQLstrLastQuestionID As String
    Dim SQLstr, SQLstrCorrect As String 'ερώτηση, σωστή απάντηση
    Dim SQLstr1, SQLstr2, SQLstr3, SQLstr4 As String '4 απαντήσεις

    Dim qTime As String 'χρόνος ερώτησης ως κείμενο για εμφάνιση στη ΒΔ και το DataGridView
    Dim iSpan2 As TimeSpan 'προσωρινή μεταβλητή για μετατροπή χρόνου σε μορφή mm:ss
    Dim previous_counter As Long = 0 'προσωρινή μεταβλητή για το χρονόμετρο

    'Αρχικοποίηση της φόρμας ερώτησης 1 για το Word
    Private Sub fmTheory_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        activeQID = 101 'η πρώτη θεωρητική ερώτηση

        'Κλήση υπορουτίνας για να απενεργοποιήσω τα κουμπάκια Tick
        HideTickButtons()

        'Κλήση υπορουτίνας για να πάρω ερωτήσεις και απαντήσεις
        RetrieveQuestionAndAnswers()
        NumOfQs = LastQuestionID - 100
        lblNum.Text = NumQuestion & "/" & NumOfQs '1/12 για την πρώτη ερώτηση κτλ

        'Ενεργοποιώ το χρονόμετρο
        Timer1.Start()
    End Sub

```

```

'Υποβολή ερώτησης
Private Sub btnSubmit_Click(sender As Object, e As EventArgs) Handles btnSubmit.Click
    ignoreQuestion = 0 'γράψω false επειδή δεν αγνοήθηκε

    'Σταματάω το χρονόμετρο
    Timer1.Stop()

    'Ελέγχει την ορθότητα της απάντησης
    If (myanswer = correct) Then
        MsgBox("Η απάντηση είναι σωστή", MsgBoxStyle.Information, "iTestU - Ερώτηση
Θεωρίας")
        pointsThQuestion = 1 'παίρνω 1 βαθμό
    Else
        MsgBox("Η απάντηση είναι λάθος", MsgBoxStyle.Critical, "iTestU - Ερώτηση Θεωρίας")
        pointsThQuestion = 0 'παίρνω 0 πόντους
    End If

    'Εδώ γιατί αλλιώς αλλάζει ο αριθμός με το ok του MsgBox()
    NumQuestion += 1
    lblNum.Text = NumQuestion & "/" & NumOfQs
    'Κλήση υπορουτίνας για να απενεργοποιήσω τα κουμπάκια Tick
    HideTickButtons()

    'Αποθηκεύω το χρόνο παραμονής στην ερώτηση
    execTimeTheory = (counter - 1) - previous_counter
    previous_counter += execTimeTheory

    'Μετατροπή του χρόνου παραμονής στην ερώτηση σε μορφή mm:ss ως κείμενο για παρουσίαση
στο GridView
    iSpan2 = TimeSpan.FromSeconds(execTimeTheory)
    qTime = iSpan2.Minutes.ToString.PadLeft(2, "0"c) & ":" &
iSpan2.Seconds.ToString.PadLeft(2, "0"c)

    'Εκκινώ το χρονόμετρο
    Timer1.Start()

    'Κλήση υπορουτίνας για να γράψω πληροφορίες για την ερώτηση
    InsertQuestionInfoToDatabase()

    'Ελεγχος αν υπάρχει άλλη θεωρητική ερώτηση
    activeQID += 1
    If (activeQID <= LastQuestionID) Then
        'φορτώσε επόμενη ερώτηση
        'Κλήση υπορουτίνας για να πάρω ερωτήσεις και απαντήσεις
        RetrieveQuestionAndAnswers()
    Else
        'δεν υπάρχει, τερματίζω και εμφανίζω αποτελέσματα
        UserAnalyticResults.Show() 'ανοίγει τη φόρμα αποτελεσμάτων
        Me.Hide()
    End If
End Sub

'Αγνόηση ερώτησης
Private Sub btnIgnore_Click(sender As Object, e As EventArgs) Handles btnIgnore.Click
    NumQuestion += 1
    lblNum.Text = NumQuestion & "/" & NumOfQs

    'Κλήση υπορουτίνας για να απενεργοποιήσω τα κουμπάκια Tick
    HideTickButtons()

    ignoreQuestion = -1 'γράψω true επειδή αγνοήθηκε
    pointsThQuestion = 0 'η αγνόηση παίρνει 0 βαθμούς

    'Αποθηκεύω το χρόνο παραμονής στην ερώτηση
    execTimeTheory = (counter - 1) - previous_counter
    previous_counter += execTimeTheory

```

```

'Μετατροπή του χρόνου παραμονής στην ερώτηση σε μορφή mm:ss ως κείμενο για παρουσίαση
στο GridView
iSpan2 = TimeSpan.FromSeconds(execTimeTheory)
qTime = iSpan2.Minutes.ToString.PadLeft(2, "0"c) & ":" &
iSpan2.Seconds.ToString.PadLeft(2, "0"c)

'Κλήση υπορουτίνας για να γράψω πληροφορίες για την ερώτηση
InsertQuestionInfoToDatabase()

'ελεγχος αν υπάρχει άλλη θεωρητική ερώτηση
activeQID += 1
If (activeQID <= LastQuestionID) Then
    'φορτωσε επόμενη ερώτηση
    'Κλήση υπορουτίνας για να πάρω ερωτήσεις και απαντήσεις
    RetrieveQuestionAndAnswers()

Else
    'δεν υπάρχει, τερματίζω και εμφανίζω αποτελέσματα
    UserAnalyticResults.Show() 'ανοίγει τη φόρμα αποτελεσμάτων
    Me.Hide()
End If

End Sub

'Αναίρεση ερώτησης
Private Sub btnUndo_Click(sender As Object, e As EventArgs) Handles btnUndo.Click
    HideTickButtons()
End Sub

'Τερματισμός ερώτησης
Private Sub btnExit_Click(sender As Object, e As EventArgs) Handles btnExit.Click
    ignoreQuestion = -1 'γράψω true τερματισμός σημαίνει ότι αγνοήθηκε
    pointsThQuestion = 0 'ο τερματισμός παίρνει 0 βαθμούς γιατί θεωρώ ως αγνοημένη
    Dim result As Integer = MsgBox("Είστε σίγουρος/η ότι θέλετε να" + vbNewLine +
"τερματίσετε την πρακτική εξάσκηση;", MsgBoxStyle.OkCancel Or MsgBoxStyle.Exclamation, "iTestU")
    'Αν πατηθεί Ok τερματίζεται αν πατηθεί cancel επιστρέφω στην ερώτηση
    If result = DialogResult.OK Then

        'Αποθηκεύω το χρόνο παραμονής στην ερώτηση
        execTimeTheory = (counter - 1) - previous_counter
        previous_counter += execTimeTheory

        'Μετατροπή του χρόνου παραμονής στην ερώτηση σε μορφή mm:ss
        iSpan2 = TimeSpan.FromSeconds(execTimeTheory)
        qTime = iSpan2.Minutes.ToString.PadLeft(2, "0"c) & ":" &
iSpan2.Seconds.ToString.PadLeft(2, "0"c)

        'Σταματάω το χρονόμετρο
        Timer1.Stop()
        'Κλήση υπορουτίνας για να γράψω πληροφορίες για την ερώτηση
        InsertQuestionInfoToDatabase()
        UserAnalyticResults.Show() 'ανοίγει τη φόρμα αποτελεσμάτων
        Me.Hide()
    End If
End Sub

'Επιλέγω την πρώτη απάντηση
Private Sub lbAnswer1_Click(sender As Object, e As EventArgs) Handles lbAnswer1.Click
    HideTickButtons()
    pbTickAnswer1.Visible = True
    myanswer = 1
End Sub

'Επιλέγω τη δεύτερη απάντηση
Private Sub lbAnswer2_Click(sender As Object, e As EventArgs) Handles lbAnswer2.Click
    HideTickButtons()
    pbTickAnswer2.Visible = True
    myanswer = 2
End Sub

```



```

'Επιλέγω την τρίτη απάντηση
Private Sub lbAnswer3_Click(sender As Object, e As EventArgs) Handles lbAnswer3.Click
    HideTickButtons()
    pbTickAnswer3.Visible = True
    myanswer = 3
End Sub
'Επιλέγω την τέταρτη απάντηση
Private Sub lbAnswer4_Click(sender As Object, e As EventArgs) Handles lbAnswer4.Click
    HideTickButtons()
    pbTickAnswer4.Visible = True
    myanswer = 4
End Sub
'Χρονόμετρο της ερώτησης
Private Sub Timer1_Tick_1(sender As Object, e As EventArgs) Handles Timer1.Tick
    Dim iSpan As TimeSpan = TimeSpan.FromSeconds(counter)
    counter += 1
    'Εμφανίζω το χρόνο με την μορφή mm:ss πάνω στη φόρμα
    lblTime.Text = iSpan.Minutes.ToString.PadLeft(2, "0"c) & ":" &
iSpan.Seconds.ToString.PadLeft(2, "0"c)
End Sub

'Υπορουτίνα ανάκτησης ερωτήσεων από πίνακα Questions και απαντήσεων από πίνακα TheoryAnswers
Sub RetrieveQuestionAndAnswers()
    'Συνδέομαι στη ΒΔ
    Try
        Mycn = New OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;
Data Source=C:\Users\Akis\Documents\Test2U.accdb")
        Mycn.Open()

        'Παίρνω το ID της τελευταίας θεωρητικής ερώτησης που έχει προστεθεί
        SQLstrLastQuestionID = "SELECT LAST (TheoryAnswers.AnswerID) FROM TheoryAnswers"
        Command = New OleDbCommand(SQLstrLastQuestionID, Mycn)
        LastQuestionID = CInt(Command.ExecuteScalar) 'ακέραιος

        'Παίρνω την εκφώνηση της ερώτησης
        SQLstr = "SELECT Questions.Question FROM Questions
WHERE (Questions.QuestionID=" & activeQID & ")"
        Command = New OleDbCommand(SQLstr, Mycn)
        lbQuestion.Text = CStr(Command.ExecuteScalar) 'string

        'Παίρνω την απάντηση 1 από τον πίνακα TheoryAnswers
        SQLstr1 = "SELECT TheoryAnswers.Answer1 FROM TheoryAnswers
WHERE (TheoryAnswers.AnswerID=" & activeQID & ")"
        Command = New OleDbCommand(SQLstr1, Mycn)
        lbAnswer1.Text = CStr(Command.ExecuteScalar) 'string που πάει στη φόρμα :-

        'Παίρνω την απάντηση 2
        SQLstr2 = "SELECT TheoryAnswers.Answer2 FROM TheoryAnswers
WHERE (TheoryAnswers.AnswerID=" & activeQID & ")"
        Command = New OleDbCommand(SQLstr2, Mycn)
        lbAnswer2.Text = CStr(Command.ExecuteScalar) 'string

        'Παίρνω την απάντηση 3
        SQLstr3 = "SELECT TheoryAnswers.Answer3 FROM TheoryAnswers
WHERE (TheoryAnswers.AnswerID=" & activeQID & ")"
        Command = New OleDbCommand(SQLstr3, Mycn)
        lbAnswer3.Text = CStr(Command.ExecuteScalar) 'string

        'Παίρνω την απάντηση 4
        SQLstr4 = "SELECT TheoryAnswers.Answer4 FROM TheoryAnswers
WHERE (TheoryAnswers.AnswerID=" & activeQID & ")"
        Command = New OleDbCommand(SQLstr4, Mycn)
        lbAnswer4.Text = CStr(Command.ExecuteScalar) 'string

        'Παίρνω την σωστή απάντηση
        SQLstrCorrect = "SELECT TheoryAnswers.CorrectAnswer FROM TheoryAnswers
WHERE (TheoryAnswers.AnswerID=" & activeQID & ")"
        Command = New OleDbCommand(SQLstrCorrect, Mycn)
        correct = CInt(Command.ExecuteScalar)
    End Try
End Sub

```

```

        'Αν συμβεί πρόβλημα με τη σύνδεση της ΒΔ
    Catch ex As Exception
        MsgBox(ex)

    Finally
        'Κλείνω τη σύνδεση με τη ΒΔ
        Mycn.Close()
    End Try
End Sub

'Υπορουτίνα που γράφει στη βάση στον πίνακα Execution στοιχεία της ερώτησης όπως το Session,
'User, βαθμό, χρόνο εκτέλεσης κτλ
Sub InsertQuestionInfoToDatabase()
    'Εισαγωγή δεδομένων στη ΒΔ
    Try
        Mycn = New OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;
                                     Data Source=C:\Users\Akis\Documents\Test2U.accdb")

        Mycn.Open()
        'Τα δεδομένα της ερώτησης εισάγονται στα πεδία του πίνακα Execution στη ΒΔ
        SQLstr = "INSERT INTO
                  Execution(IDS,IDQ,IdSession,Score,IsIgnore,ExecTime,QuestionTime)
                  VALUES(" & ids & "," & activeQID & "," & idszn & "," & pointsThQuestion & ",
                           " & ignoreQuestion & "," & execTimeTheory & "," & qTime & ")"

        'Εκτέλεση ερωτήματος εισαγωγής δεδομένων
        Command = New OleDbCommand(SQLstr, Mycn)
        icount = Command.ExecuteNonQuery

        'Αν συμβεί πρόβλημα με τη σύνδεση της ΒΔ
    Catch ex As Exception
        MsgBox(ex)

    Finally
        'Κλείνω τη σύνδεση με τη ΒΔ
        Mycn.Close()
    End Try
End Sub

'Υπορουτίνα που κρύβει τα κουμπάκια με το Tick και αρχικοποιεί την απάντηση
Sub HideTickButtons()
    pbTickAnswer1.Visible = False
    pbTickAnswer2.Visible = False
    pbTickAnswer3.Visible = False
    pbTickAnswer4.Visible = False
    myanswer = 0 'δεν έχω απαντήσει τίποτα
End Sub
End Class

'-----
'Project:      Μεταπτυχιακή Διατριβή
'Filename:     EditProfile.vb
'Author:      Άκης Βοΐλας
'Version:     1.0.0
'Date:
'Description:  Είναι το παράθυρο για την διαχείριση του προσωπικού λογαριασμού
'-----
'Για τη σύνδεση με ΒΔ
Imports System.Data.OleDb

Public Class EditProfile
    Dim Mycn As OleDbConnection
    Dim Command As OleDbCommand
    Dim icount As Integer
    Dim SQLstrUpdate As String

    Private Sub EditProfile_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Me.tblMembersTableAdapter.Fill(Me.ProfileDataSet.tblMembers, CType(value, Integer))
    End Sub
End Class

```

```

txtMName.Text = NameLabel1.Text
txtMSurname.Text = SurnameLabel1.Text
txtMUsername.Text = UsernameLabel2.Text
txtMPassword.Text = PasscodeLabel2.Text
txtMEmail.Text = EmailLabel2.Text
cbMOccupation.Text = OccupationLabel2.Text
ReturnGender() 'φύλο
'Αν είναι διαχειριστής
End Sub

'Υπορουτίνα για να εντοπίζω το φύλο
Sub ReturnGender()
If GenderLabel2.Text = "Αρρεν" Then
    rbMMan.Checked = True
    rbMWoman.Checked = False
ElseIf GenderLabel2.Text = "Θήλυ" Then
    rbMMan.Checked = False
    rbMWoman.Checked = True
Else
    rbMMan.Checked = False
    rbMWoman.Checked = False
End If
End Sub

Private Sub btnModifyUser_Click(sender As Object, e As EventArgs) Handles
btnModifyUser.Click
    'Αποθηκεύει το φύλο
    Dim gender As String = "Αρρεν"
    If rbMWoman.Checked = True Then
        gender = "Θήλυ"
    End If

    If (txtMName.Text <> "" And txtMSurname.Text <> "" And txtMUsername.Text <> "" And
txtMPassword.Text <> "") Then
        'Συνδέομαι στη ΒΔ
        Try
            Mycn = New OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;
Data Source=C:\Users\Akis\Documents\Test2U.accdb")
            Mycn.Open()

            'Ενημερώνω στον πίνακα Questions τα πεδία Question και QuestionTotalPoints
            SQLstrUpdate = "UPDATE tblMembers
SET tblMembers.Name='" & txtMName.Text &
"',tblMembers.Surname='" & txtMSurname.Text & "',tblMembers.Username='" & txtMUsername.Text &
"',tblMembers.Passcode='" & txtMPassword.Text & "',tblMembers.Email='" & txtMEmail.Text &
"',tblMembers.Occupation='" & cbMOccupation.Text & "',tblMembers.Gender='" & gender & "'
WHERE tblMembers.MemberID=" & value & " "
            Command = New OleDbCommand(SQLstrUpdate, Mycn)
            icount = Command.ExecuteNonQuery
            MessageBox.Show("Ενημερώθηκαν επιτυχώς τα" & vbNewLine & "προσωπικά στοιχεία
σας", "iTestU - Ενημέρωση", MessageBoxButtons.OK, MessageBoxIcon.Information)

            'Αν συμβεί πρόβλημα με τη σύνδεση της ΒΔ
            Catch ex As Exception
                MsgBox(ex)

            Finally
                'Κλείνω τη σύνδεση με τη ΒΔ
                Mycn.Close()
            End Try

        Else
            MessageBox.Show("Πρέπει να συμπληρώσετε όλα τα" & vbNewLine & "υποχρεωτικά πεδία της
φόρμας", "Σφάλμα εγγραφής", MessageBoxButtons.OK, MessageBoxIcon.Warning)
        End If
    End Sub

```

```

Private Sub btnModifyClear_Click(sender As Object, e As EventArgs) Handles
btnModifyClear.Click
    txtMName.Text = vbNullString
    txtMSurname.Text = vbNullString
    txtMEmail.Text = vbNullString
    txtMUsername.Text = vbNullString
    txtMPassword.Text = vbNullString
    cbMOccupation.Text = "--- Επιλέξτε ή εισάγετε ---"
    rbMMan.Checked = False
    rbMWoman.Checked = False
End Sub
End Class

'-----
'Project:      Μεταπτυχιακή Διατριβή
'Filename:     Administrator/ManageQuestion.vb
'Author:       Άκης Βοΐλας
'Version:      1.0.0
'Date:         20/8/2016
'Description:  0 διαχειριστής μπορεί να δημιουργήσει ή/και να τροποποιήσει ερωτήσεις
'-----
'Για τη σύνδεση με ΒΔ
Imports System.Data.OleDb

Public Class ManageQuestion
    Dim SearchID As Integer 'Ο κωδικός της ερώτησης που ψάχνω να τροποποιήσω
    Dim LastQID As Integer 'κρατάει ID της τελευταίας θεωρητικής ερώτησης
    Dim NewQID As Integer 'κρατάει ID της νέας θεωρητικής ερώτησης
    Dim QIDcat As Integer = 1 'κρατάει τον κωδικό της κατηγορία 1~>θεωρητικές
    Dim QPoints As Integer = 1 'κρατάει βαθμό ερώτησης
    Dim QcorrectAnswer As Integer

    Dim Mycn As OleDbConnection
    Dim Command As OleDbCommand
    Dim icount As Integer

    'Στην δημιουργία ερώτησης
    Dim SQLstrLastAnswerID As String
    Dim SQLstr, SQLstr2, SQLstrNum As String
    'Στην τροποποίηση ερώτησης
    Dim SQLstrQ, SQLstrPoints, SQLstrAns1, SQLstrAns2, SQLstrAns3, SQLstrAns4, SQLstrCorrect As
String
    Dim SQLstrModQ, SQLstrModAns As String

    Private Sub ManageQuestion_Load(sender As Object, e As EventArgs) Handles MyBase.Load

        Me.TheoryAnswersTableAdapter.Fill(Me.IdDataSet.TheoryAnswers)

        'Συνδέομαι στη ΒΔ
        Try
            Mycn = New OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;
                                     Data Source=C:\Users\Akis\Documents\Test2U.accdb")
            Mycn.Open()

            'Παίρνω το ID της τελευταίας ερώτησης που δημιουργήθηκε, από πίνακα TheoryAnswers
            SQLstrLastAnswerID = "SELECT LAST(TheoryAnswers.AnswerID) FROM TheoryAnswers"
            Command = New OleDbCommand(SQLstrLastAnswerID, Mycn)
            LastQID = CInt(Command.ExecuteScalar) 'τελευταίο AnswerID

            'Αν συμβεί πρόβλημα με τη σύνδεση της ΒΔ
            Catch ex As Exception
                MsgBox(ex)

            Finally
                'Κλείνω τη σύνδεση με τη ΒΔ
                Mycn.Close()
            End Try
        End Try
    End Sub
End Class

```

```

NewQID = LastQID + 1
'Tα εμφανίζω στη φόρμα και είναι κλειδωμένα (Disable)
txtNewQID.Text = NewQID
txtQIDcat.Text = QIDcat
End Sub

'Δημιουργία ερώτησης
Private Sub btnCreate_Click(sender As Object, e As EventArgs) Handles btnCreate.Click
    Dim NumOfQ As Integer = 12
    Dim CatID As Integer = 1 'κατηγορία θεωρητικών
    NumOfQ = NewQID - 100 'νέο πλήθος ερωτήσεων
    If (txtQuestion.Text <> "" And txtAnswer1.Text <> "" And txtAnswer2.Text <> "" And
        txtAnswer3.Text <> "" And txtAnswer4.Text <> "" And cmbCorrectAnswer.Text <> "" And
        cmbPoints.Text <> "") Then
        QPoints = CInt(cmbPoints.Text)
        QcorrectAnswer = CInt(cmbCorrectAnswer.Text) 'από combobox το μετατρέπει σε ακέραιο

        'Συνδέομαι στη ΒΔ
        Try
            Mycn = New OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;
                Data Source=C:\Users\Akis\Documents\Test2U.accdb")
            Mycn.Open()

            'Τα δεδομένα της φόρμας εισάγονται στα πεδία του πίνακα Questions
            SQLstr = "INSERT INTO Questions(QuestionID, Question, QuestionTotalPoints,
                QuestionIdCategory)
                VALUES(" & NewQID & ",'" & txtQuestion.Text & "'," & QPoints & ",
                " & QIDcat & ")"
            Command = New OleDbCommand(SQLstr, Mycn)
            icount = Command.ExecuteNonQuery

            'Τα δεδομένα της φόρμας εισάγονται στα πεδία του πίνακα TheoryAnswers
            SQLstr2 = "INSERT INTO TheoryAnswers(AnswerID, Answer1, Answer2, Answer3,
                Answer4, CorrectAnswer)
                VALUES(" & NewQID & ",'" & txtAnswer1.Text & "',
                '" & txtAnswer2.Text & "'," & txtAnswer3.Text & "',
                '" & txtAnswer4.Text & "'," & QcorrectAnswer & ")"
            Command = New OleDbCommand(SQLstr2, Mycn)
            icount = Command.ExecuteNonQuery

            'Ενημερώνω τον πίνακα tblQuestionCategory για το νέο πλήθος των ερωτήσεων
            SQLstrNum = "UPDATE tblQuestionCategory
                SET tblQuestionCategory.NumOfQuestions=" & NumOfQ & "
                WHERE tblQuestionCategory.CategoryID=" & CatID & " "
            Command = New OleDbCommand(SQLstrNum, Mycn)
            icount = Command.ExecuteNonQuery
            MessageBox.Show("Δημιουργήθηκε μια νέα" & vbCrLf & "ερώτηση με κωδικό " &
                NewQID, "iTestU", MessageBoxButtons.OK, MessageBoxIcon.Information)

            'Αν συμβεί πρόβλημα με τη σύνδεση της ΒΔ
            Catch ex As Exception
                MsgBox(ex)
            Finally
                'Κλείνω τη σύνδεση με τη ΒΔ
                Mycn.Close()
            End Try

            NewQID = NewQID + 1 'επόμενο QID
            'Τα εμφανίζω στη φόρμα
            txtNewQID.Text = NewQID
            txtQIDcat.Text = QIDcat
            'Κλήση υπορουτίνας αρχικοποίησης
            ClearFormFields()
        Else
            MessageBox.Show("Πρέπει να συμπληρώσετε όλα" & vbCrLf & "τα πεδία της φόρμας",
                "Σφάλμα εγγραφής", MessageBoxButtons.OK, MessageBoxIcon.Warning)
        End If
    End Sub
    'Καθαρίζει τα πεδία της φόρμας

```

```

Private Sub btnClear_Click(sender As Object, e As EventArgs) Handles btnClear.Click
    ClearFormFields()'Κλήση υπορουτίνας αρχικοποίησης
End Sub

'Εύρεση της ερώτησης από τον κωδικό της και εμφάνιση όλων των δεδομένων της
Private Sub btnFindQuestion_Click(sender As Object, e As EventArgs) Handles
btnFindQuestion.Click
    SearchID = CInt(cmbSearch2ID.Text)

    'Συνδέομαι στη ΒΔ
    Try
        Mycn = New OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;
            Data Source=C:\Users\Akis\Documents\Test2U.accdb")

        Mycn.Open()

        'Εμφανίζω την εκφώνηση της ερώτησης
        SQLstrQ = "SELECT Questions.Question FROM Questions
            WHERE (Questions.QuestionID=" & SearchID & ")"
        Command = New OleDbCommand(SQLstrQ, Mycn)
        txtModQuestion.Text = CStr(Command.ExecuteScalar)

        'Εμφανίζω τη βαθμολογία της σωστής απάντησης
        SQLstrPoints = "SELECT Questions.QuestionTotalPoints FROM Questions
            WHERE (Questions.QuestionID=" & SearchID & ")"
        Command = New OleDbCommand(SQLstrPoints, Mycn)
        cmbModPoints.Text = CInt(Command.ExecuteScalar)

        'Εμφανίζω την απάντηση 1
        SQLstrAns1 = "SELECT TheoryAnswers.Answer1 FROM TheoryAnswers
            WHERE (TheoryAnswers.AnswerID=" & SearchID & ")"
        Command = New OleDbCommand(SQLstrAns1, Mycn)
        txtModAns1.Text = CStr(Command.ExecuteScalar)

        'Εμφανίζω την απάντηση 2
        SQLstrAns2 = "SELECT TheoryAnswers.Answer2 FROM TheoryAnswers
            WHERE (TheoryAnswers.AnswerID=" & SearchID & ")"
        Command = New OleDbCommand(SQLstrAns2, Mycn)
        txtModAns2.Text = CStr(Command.ExecuteScalar)

        'Εμφανίζω την απάντηση 3
        SQLstrAns3 = "SELECT TheoryAnswers.Answer3 FROM TheoryAnswers
            WHERE (TheoryAnswers.AnswerID=" & SearchID & ")"
        Command = New OleDbCommand(SQLstrAns3, Mycn)
        txtModAns3.Text = CStr(Command.ExecuteScalar)

        'Εμφανίζω την απάντηση 4
        SQLstrAns4 = "SELECT TheoryAnswers.Answer4 FROM TheoryAnswers
            WHERE (TheoryAnswers.AnswerID=" & SearchID & ")"
        Command = New OleDbCommand(SQLstrAns4, Mycn)
        txtModAns4.Text = CStr(Command.ExecuteScalar)

        'Εμφανίζω τη σωστή απάντηση
        SQLstrCorrect = "SELECT TheoryAnswers.CorrectAnswer FROM TheoryAnswers
            WHERE (TheoryAnswers.AnswerID=" & SearchID & ")"
        Command = New OleDbCommand(SQLstrCorrect, Mycn)
        cmbModCorrectAnswer.Text = CInt(Command.ExecuteScalar)

        'Αν συμβεί πρόβλημα με τη σύνδεση της ΒΔ
    Catch ex As Exception
        MsgBox(ex)
    Finally
        'Κλείνω τη σύνδεση με τη ΒΔ
        Mycn.Close()
    End Try
    GroupBox6.Enabled = True
    GroupBox7.Enabled = True
    GroupBox8.Enabled = True
End Sub

'Τροποποίηση ερώτησης - Ενημέρωση ΒΔ

```

```

Private Sub btnModify_Click(sender As Object, e As EventArgs) Handles btnModify.Click

    If (txtModQuestion.Text <> "" And txtModAns1.Text <> "" And txtModAns2.Text <> "" And
        txtModAns3.Text <> "" And txtModAns4.Text <> "" And cmbModCorrectAnswer.Text <> "" And
        cmbModPoints.Text <> "") Then
        'Συνδέομαι στη ΒΔ
        Try
            Mycn = New OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;
                Data Source=C:\Users\Akis\Documents\Test2U.accdb")
            Mycn.Open()

            'Ενημερώνω στον πίνακα Questions τα πεδία Question και QuestionTotalPoints
            SQLstrModQ = "UPDATE Questions SET Questions.QuestionID=" & txtModQuestion.Text &
                "' ,Questions.QuestionTotalPoints=" & cmbModPoints.Text & " WHERE Questions.QuestionID=" &
                SearchID & " "
            Command = New OleDbCommand(SQLstrModQ, Mycn)
            icount = Command.ExecuteNonQuery

            'Ενημερώνω τον πίνακα TheoryAnswers τα πεδία
            Answer1,Answer2,Answer3,Answer4,CorrectAnswer
            SQLstrModAns = "UPDATE TheoryAnswers SET TheoryAnswers.Answer1=" &
                txtModAns1.Text & "' , TheoryAnswers.Answer2=" & txtModAns2.Text & "' , TheoryAnswers.Answer3=" &
                & txtModAns3.Text & "' , TheoryAnswers.Answer4=" & txtModAns4.Text &
                "' , TheoryAnswers.CorrectAnswer=" & cmbModCorrectAnswer.Text & " WHERE TheoryAnswers.AnswerID=" &
                SearchID & " "
            Command = New OleDbCommand(SQLstrModAns, Mycn)
            icount = Command.ExecuteNonQuery

            'Αν συμβεί πρόβλημα με τη σύνδεση της ΒΔ
            MessageBox.Show("Ενημερώθηκε επιτυχώς" & vbNewLine & "η ερώτηση με κωδικό " &
                SearchID, "iTestU", MessageBoxButtons.OK, MessageBoxIcon.Information)

        Catch ex As Exception
            MsgBox(ex)

        Finally
            'Κλείνω τη σύνδεση με τη ΒΔ
            Mycn.Close()
        End Try

    Else
        MessageBox.Show("Πρέπει να συμπληρώσετε όλα" & vbNewLine & "τα πεδία της φόρμας",
            "Σφάλμα εγγραφής", MessageBoxButtons.OK, MessageBoxIcon.Warning)
    End If

End Sub
'Αρχικοποίηση της φόρμας τροποποίηση
Private Sub btnModClear_Click(sender As Object, e As EventArgs) Handles btnModClear.Click
    txtModQuestion.Text = vbNullString
    txtModAns1.Text = vbNullString
    txtModAns2.Text = vbNullString
    txtModAns3.Text = vbNullString
    txtModAns4.Text = vbNullString
    cmbModPoints.Text = vbNullString
    cmbModCorrectAnswer.Text = vbNullString
End Sub

'Υπορουτίνα αρχικοποίησης πεδίων της φόρμας εγγραφής
Sub ClearFormFields()
    txtQuestion.Text = vbNullString
    txtAnswer1.Text = vbNullString
    txtAnswer2.Text = vbNullString
    txtAnswer3.Text = vbNullString
    txtAnswer4.Text = vbNullString
    cmbPoints.Text = vbNullString
    cmbCorrectAnswer.Text = vbNullString
End Sub
End Class

```

```

'-----
'Project:      Μεταπτυχιακή Διατριβή
'Filename:     Statistics/ AdminStatistics.vb
'Author:       Άκης Βοΐλας
'Version:     1.0.0
'Date:        25/8/2016
'Description:  0 διαχειριστής παρακολουθεί το ιστορικό των εξετάσεων
'-----
Imports Excel = Microsoft.Office.Interop.Excel

Public Class AdminStatistics

    Private Sub AdminStatistics_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        'Εμφανίζει όλες τις εγγραφές από το ερώτημα QueryAdminPercentage της ΒΔ
        Me.QueryAdminPercentageTableAdapter.Fill(Me.AdminDataSet.QueryAdminPercentage)
    End Sub

    'Δημιουργία παραμετρικών ερωτημάτων στο AdminDataSet
    Private Sub btnApplyFilter_Click(sender As Object, e As EventArgs) Handles
btnApplyFilter.Click
        If ckboxCategory.Checked = False And ckboxDate.Checked = False Then
            'Εμφανίζει όλες τις εγγραφές
            Me.QueryAdminPercentageTableAdapter.Fill(Me.AdminDataSet.QueryAdminPercentage)
        ElseIf ckboxCategory.Checked = True And ckboxDate.Checked = False Then
            'Εμφανίζει τις εγγραφές με παράμετρο την κατηγορία που δίνουμε
            Me.QueryAdminPercentageTableAdapter.FillByCategory(Me.AdminDataSet.QueryAdminPercentage,
cmbCategory.Text)
        ElseIf ckboxCategory.Checked = False And ckboxDate.Checked = True Then
            'Εμφανίζει τις εγγραφές με παράμετρο το διάστημα ημερομηνιών που δίνουμε
            Me.QueryAdminPercentageTableAdapter.FillByDate(Me.AdminDataSet.QueryAdminPercentage,
DateFrom.Value, DateTo.Value)
        Else
            'Εμφανίζει τις εγγραφές με παράμετρο την κατηγορία και το διάστημα ημερομηνιών
            Me.QueryAdminPercentageTableAdapter.FillByAll(Me.AdminDataSet.QueryAdminPercentage,
cmbCategory.Text, DateFrom.Value, DateTo.Value)
        End If
    End Sub

    'Απενεργοποίηση του φίλτρου
    Private Sub btnDisableFilter_Click(sender As Object, e As EventArgs) Handles
btnDisableFilter.Click
        ckboxCategory.Checked = False
        ckboxDate.Checked = False
        Me.QueryAdminPercentageTableAdapter.Fill(Me.AdminDataSet.QueryAdminPercentage)
    End Sub

    'Εξαγωγή αποτελεσμάτων DataGridView σε Excel ή εκτύπωση/PDF
    Private Sub btnExportResults_Click(sender As Object, e As EventArgs) Handles
btnExportResults.Click
        If rbExportExcel.Checked = True Then
            ExportToExcel() 'κλήση υπορουτίνας excel
        ElseIf rbPrintPDF.Checked = True Then
            PrintDocument1.Print() 'κλήση μεθόδου pdf/εκτύπωση
        Else
            MsgBox("Επιλέξτε πρώτα τον τύπο" & vbCrLf & "του αρχείου εξαγωγής",
MsgBoxStyle.Exclamation, "iTestU - Εξαγωγή αρχείου")
        End If
    End Sub

    'Δημιουργία της μορφής του αρχείου
    Private Sub PrintDocument1_PrintPage(ByVal sender As System.Object, ByVal e As
System.Drawing.Printing.PrintPageEventArgs) Handles PrintDocument1.PrintPage
        ' Turn on antialias for text
        e.Graphics.TextRenderingHint = Drawing.Text.TextRenderingHint.AntiAlias

        'Μορφοποιημένο κείμενο στην κορυφή
        Dim pFont As Font

```



```

pFont = New Font("Arial", 11, FontStyle.Italic)
e.Graphics.DrawString(Date.Today, pFont, Brushes.Black, 720, 30)
e.Graphics.DrawString("iTESTU - Συγκεντρωτικά αποτελέσματα χρηστών", pFont,
Brushes.Black, 40, 50)

Dim Lmargin, Rmargin, Tmargin, Bmargin As Integer
With PrintDocument1.DefaultPageSettings.Margins
    Lmargin = .Left
    Rmargin = .Right
    Tmargin = .Top
    Bmargin = .Bottom
End With

'Υπολογίζει τις διαστάσεις της περιοχής εκτύπωσης
Dim PrintWidth, PrintHeight As Integer
With PrintDocument1.DefaultPageSettings.PaperSize
    PrintWidth = .Width - Lmargin - Rmargin
    PrintHeight = .Height - Tmargin - Bmargin
End With

'Mορφοποιώ το DataGridView για την εκτύπωση bitmap
QueryAdminPercentageDataGridView.ScrollBars = ScrollBars.None 'κρύβω γραμμές κύλισης
QueryAdminPercentageDataGridView.RowHeadersVisible = False 'κρύβω τον οδηγό γραμμών
QueryAdminPercentageDataGridView.BackgroundColor = Color.White
'Εκτυπώνω το DataGridView ως bitmap
Dim bm As New Bitmap(Me.QueryAdminPercentageDataGridView.Width,
Me.QueryAdminPercentageDataGridView.Height)
QueryAdminPercentageDataGridView.DrawToBitmap(bm, New Rectangle(0, 0,
Me.QueryAdminPercentageDataGridView.Width, Me.QueryAdminPercentageDataGridView.Height))
e.Graphics.DrawImage(bm, 40, 120) '40 οριζόντια 100 κατακόρυφα
End Sub

'Επαναφέρω το GridView στην αρχική του μορφή πριν την εκτύπωση
Private Sub btnExportResults_MouseLeave(sender As Object, e As EventArgs) Handles
btnExportResults.MouseLeave
    QueryAdminPercentageDataGridView.ScrollBars = ScrollBars.Vertical
    QueryAdminPercentageDataGridView.RowHeadersVisible = True
    QueryAdminPercentageDataGridView.BackgroundColor = SystemColors.AppWorkspace
End Sub

'Εξαγωγή αποτελεσμάτων DataGridView σε φύλλο Excel
Sub ExportToExcel()
    Dim xlApp As Excel.Application
    Dim xlWorkBook As Excel.Workbook
    Dim xlWorkSheet As Excel.Worksheet
    Dim misValue As Object = System.Reflection.Missing.Value

    Dim i As Int16, j As Int16
    'Ανοίγει την εφαρμογή Excel
    xlApp = CreateObject("Excel.Application")
    xlApp.Visible = True

    xlWorkBook = xlApp.Workbooks.Add(misValue)
    'Ορίζω να είναι το φύλλο1 του βιβλίου
    xlWorkSheet = xlWorkBook.Sheets(1)

    'Διαμόρφωση σελίδας
    With xlWorkSheet.PageSetup
        'Κατακόρυφος προσανατολισμός
        .Orientation = Excel.XlPageOrientation.xlPortrait
        'Κλίμακα
        .Zoom = False
        .FitToPagesWide = 1
        .FitToPagesTall = False
        'Κεφαλίδα και Υποσέλιδο
        .LeftHeader = "iTestU - Συγκεντρωτικά αποτελέσματα"
        .RightHeader = Date.Now
        .CenterFooter = "Σελίδα &P από &N"
    End With
End Sub

```

```

        'Φύλλο εκτύπωση πλέγματος
        .PrintGridlines = True
        'Επανάληψη γραμμών στην κορυφή
        .PrintTitleRows = "$1:$2"
    End With

    'Όνομα φύλλου
    xlWorkSheet.Name = "Συγκεντρωτικά αποτελέσματα"

    'Συγχωνεύσεις κελιών
    With xlWorkSheet
        .Range("A1:F1").MergeCells = True
    End With

    With xlWorkSheet
        'Πλάτη των στηλών
        .Range("A2").ColumnWidth = 15
        .Range("B2").ColumnWidth = 16
        .Range("C2").ColumnWidth = 23
        .Range("D2").ColumnWidth = 15
        .Range("E2").ColumnWidth = 18
        .Range("F2").ColumnWidth = 8

        'Υψη γραμμών
        .Range("A1").RowHeight = 24
        .Range("A2").RowHeight = 21
    End With

    'Στοιχίσεις κελιών
    With xlWorkSheet.Range("A1")
        .HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter
        .VerticalAlignment = Excel.XlVAlign.xlVAlignCenter
    End With

    With xlWorkSheet.Range("D2:F2000")
        .HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter
        .VerticalAlignment = Excel.XlVAlign.xlVAlignCenter
    End With

    'Μορφοποιήσεις γραμματοσειράς
    xlWorkSheet.Range("A1").Interior.Color = Color.LightGray
    xlWorkSheet.Range("A1").Font.Color = Color.DarkBlue
    xlWorkSheet.Range("A1").Font.Size = 14
    xlWorkSheet.Range("A2:F2").Font.Size = 12
    xlWorkSheet.Range("A1:F2").Font.Bold = True

    'Μορφοποιήσεις αριθμών
    xlWorkSheet.Range("D3:D2000").NumberFormat = "0.00%"

    'Τίτλος πρώτης γραμμής
    With xlWorkSheet
        .Range("A1").Value = "Συγκεντρωτικά Αποτελέσματα Εξετάσεων"
    End With

    'Γράφω τίτλους μέχρι την έκτη στήλη του DataGridView
    For j = 0 To 5
        xlWorkSheet.Cells(2, j + 1) = QueryAdminPercentageDataGridView.Columns(j).HeaderText
    Next

    'Γράφω δεδομένα από τη γραμμή 3 στο Excel, προσοχή διαβάζεται DGV(j,i) από DGV(0,0)
    For i = 0 To QueryAdminPercentageDataGridView.RowCount - 1
        For j = 0 To QueryAdminPercentageDataGridView.ColumnCount - 3
            xlWorkSheet.Cells(i + 3, j + 1) = QueryAdminPercentageDataGridView(j, i).Value
        Next
    Next

    xlWorkSheet.Range("F2").Value = "Κωδ. ΕΞ."

```

```

'Το αποθηκεύω στην επιφάνεια εργασίας με το επίθετο του χρήστη
Dim SaveFilename, Savepath As String
SaveFilename = "Συγκεντρωτικό Διαχειριστή.xlsx" 'όνομα αρχείου
Savepath = "C:\Users\Akis\Desktop\" & SaveFilename 'θέση αποθήκευσης
xlWorkbook.SaveAs(Savepath, Excel.XlFileFormat.xlWorkbookDefault, misValue, misValue,
misValue, misValue, _
Excel.XlSaveAsAccessMode.xlExclusive, misValue, misValue, misValue, misValue, misValue)
xlWorkbook.Close(True, misValue, misValue)
xlApp.Quit()
'απελευθέρωση δεσμευμένων πόρων
releaseObject(xlWorkSheet)
releaseObject(xlWorkbook)
releaseObject(xlApp)
MsgBox("Δημιουργήθηκε το αρχείο" & vbNewLine & "στην επιφάνεια εργασίας",
MsgBoxStyle.Information, "iTestU - Εξαγωγή Excel")
End Sub

'Κλήση από την δημιουργία του Excel για απελευθέρωση πόρων
Private Sub releaseObject(ByVal obj As Object)
Try
System.Runtime.InteropServices.Marshal.ReleaseComObject(obj)
obj = Nothing
Catch ex As Exception
obj = Nothing
MessageBox.Show("Exception Occured while releasing object " + ex.ToString())
Finally
GC.Collect()
End Try
End Sub
End Class

```