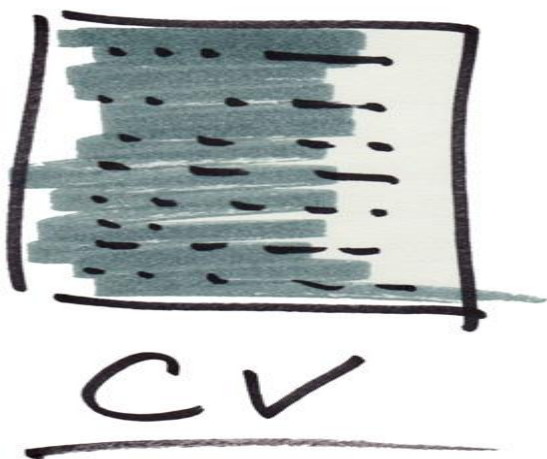




**Αλεξάνδρειο Τεχνολογικό Εκπαιδευτικό Ίδρυμα
Θεσσαλονίκης
Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Πληροφορικής**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Ανάπτυξη συστήματος διαχείρισης Βιογραφικών
υποψηφίων μελών ΕΠ**



Του φοιτητή
Μασούρα Μιχάλη
Αρ.Μητρώου:2416

Επιβλέπων Καθηγητής
Ιγνάτιος Δεληγιάννης

Θεσσαλονίκη 2011

ΠΡΟΛΟΓΟΣ

Η αντικειμενοστρεφής ανάλυση και σχεδίαση (Object Oriented Analysis and Design-OOAD) είναι μια προσέγγιση τεχνολογίας λογισμικού που διαμορφώνει ένα σύστημα ως μια ομάδα αντικειμένων που αλληλεπιδρούν. Κάθε αντικείμενο αντιπροσωπεύει κάποια οντότητα ενδιαφέροντος για το σύστημα που μοντελοποιείται, και χαρακτηρίζεται από την τάξη του, την κατάσταση του(δεδομένα), και τη συμπεριφορά του.

Διάφορα μοντέλα μπορούν να δημιουργηθούν για να παρουσιάσουν τη στατική δομή, τη δυναμική συμπεριφορά, και την απόκριση σε χρόνο εκτέλεσης αυτών των συνεργαζόμενων αντικειμένων. Υπάρχουν διάφορες σημειογραφίες για την αναπαράσταση αυτών των μοντέλων, όπως η ενοποιημένη γλώσσα μοντελοποίησης (UML).

Η αντικειμενοστρεφής ανάλυση (Object Oriented Analysis - OOA) εφαρμόζει τις τεχνικές μοντελοποίησης των αντικειμένων για να αναλυθούν οι λειτουργικές απαιτήσεις για ένα σύστημα. Ο αντικειμενοστρεφής σχεδιασμός (Object Oriented Design - OOD) διαμορφώνει τα μοντέλα ανάλυσης για να παράγει τις προδιαγραφές υλοποίησης. Η αντικειμενοστρεφής ανάλυση εστιάζει στο τι κάνει το σύστημα, ενώ ο αντικειμενοστρεφής σχεδιασμός στον τρόπο με τον οποίο το κάνει.

Ένα αντικειμενοστρεφές σύστημα αποτελείται από αντικείμενα. Η συμπεριφορά του συστήματος προκύπτει από τη συνεργασία αυτών των αντικειμένων. Η συνεργασία μεταξύ των αντικειμένων περιλαμβάνει την αποστολή μηνυμάτων μεταξύ τους. Η αποστολή ενός μηνύματος διαφέρει από την κλήση μιας συνάρτησης στο ότι όταν ένα αντικείμενο στόχος λαμβάνει ένα μήνυμα, το ίδιο αποφασίζει ποια λειτουργία θα καλέσει για να εξυπηρετήσει το μήνυμα. Το ίδιο μήνυμα μπορεί να εφαρμοστεί από πολλές διαφορετικές μεθόδους, αυτή που επιλέγεται εξαρτάται από την κατάσταση του αντικειμένου-στόχου (Object Oriented Analysis and Design,n.d.) .

Το αντικείμενο της παρούσας πτυχιακής είναι αφενός η μελέτη της αντικειμενοστρεφούς ανάλυσης και σχεδίασης και της φιλοσοφίας που αυτή πρεσβεύει και αφετέρου η σχεδίαση και η ανάπτυξη ενός αντικειμενοστρεφούς συστήματος διαχείρισης βιογραφικών υποψηφίων μελών ΕΠ, που θα παρέχει τις δυνατότητες καταχώρηση και ενημέρωση βιογραφικού για τους υποψηφίους, προβολή και ταξινόμηση βιογραφικών για την επιτροπή και εγγραφή υποψηφίων για τη γραμματεία.

ΠΕΡΙΛΗΨΗ

Αντικείμενο αυτής της εργασίας είναι η ανάλυση, ο σχεδιασμός και η υλοποίηση ενός αντικειμενοστρεφούς συστήματος διαχείρισης βιογραφικών υποψηφίων μελών ΕΠ, που θα διαχειρίζεται την προϋπηρεσία των υποψηφίων, υπολογίζοντας χρονικά τις επικαλύψεις, το δημοσιευμένο έργο των υποψηφίων και θα ταξινομεί τα διαθέσιμα στοιχεία με βάση μερικών κριτηρίων.

Στα πλαίσια της θεωρητικής μελέτης γίνεται μια επισκόπηση της αντικειμενοστρεφούς ανάλυσης και σχεδίασης λογισμικού, και του πώς αυτή εφαρμόζεται μέσα από τη γλώσσα μοντελοποίησης UML. Στη συνέχεια περιγράφονται οι τεχνολογίες με τις οποίες υλοποιήθηκε η εφαρμογή, Java, JDBC και SQLite.

Το κεφάλαιο του σχεδιασμού περιλαμβάνει την περιγραφή των περιπτώσεων χρήσης του συστήματος καθώς και τα διαγράμματα των περιπτώσεων χρήσης, τα διαγράμματα ακολουθίας και συνεργασίας και την απεικόνιση του εννοιολογικού μοντέλου και του μοντέλου υλοποίησης. Ακόμα στο κεφάλαιο αυτό απεικονίζεται το εννοιολογικό διάγραμμα Οντοτήτων-Συσχετίσεων της βάσης του συστήματος.

Στο κεφάλαιο της Υλοποίησης περιγράφονται οι λειτουργίες του συστήματος, η βάση δεδομένων και παρατίθενται κάποια τμήματα του κώδικα της υλοποίησης, και στιγμιότυπα από την εκτέλεση του προγράμματος. Τέλος, αναφέρεται ποιες λειτουργίες του αρχικού σχεδιασμού υλοποιήθηκαν τελικά καθώς και κάποιες ιδέες για μελλοντικές επεκτάσεις και βελτιώσεις.

ABSTRACT

Object of this work is the analysis, the design and the implementation of an object oriented system for the management of teaching candidates CVs, that will store and manage the previous (work/teaching/research) experience of the candidates, calculating the coverings, the published work of the candidates and will sort the available data based on certain criteria.

The theoretical study contains a review of Object Oriented Analysis and Design (OOAD) of software, and how this is applied through the Unified Modeling Language (UML). Afterwards are described the technologies that were used for the implementation of the system, such as Java, JDBC and Derby.

The Design chapter includes the description of use cases of the system as well as the use cases diagrams, the sequence and communication diagrams and the domain and implementation model diagrams. This chapter also contains the Entity-Relationship diagram for the system CV database.

In the Implementation chapter are described the operations of system, the CV database, and are quoted some parts of the implementation code, and snapshots from the application user interface. Finally, it is reported which functionalities of initial design implemented eventually, as well as certain ideas for future extensions and improvements.

ΕΥΧΑΡΙΣΤΙΕΣ (προαιρετικά)

Περιεχόμενα

ΠΡΟΛΟΓΟΣ.....	2
ΠΕΡΙΛΗΨΗ.....	3
ABSTRACT	4
ΕΥΧΑΡΙΣΤΙΕΣ (προαιρετικά)	5
ΕΥΡΕΤΗΡΙΟ ΕΙΚΟΝΩΝ.....	9
ΕΥΡΕΤΗΡΙΟ ΠΙΝΑΚΩΝ.....	10
ΕΙΣΑΓΩΓΗ	10
ΚΕΦΑΛΑΙΟ 1.....	11
ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΑΝΑΛΥΣΗ ΚΑΙ ΣΧΕΔΙΑΣΗ	11
ΕΙΣΑΓΩΓΗ.....	11
1.1 ΠΡΟΣΑΝΑΤΟΛΙΣΜΟΣ ΣΕ ΑΝΤΙΚΕΙΜΕΝΑ.....	12
1.2 ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΑΝΑΛΥΣΗ ΚΑΙ ΣΧΕΔΙΑΣΗ.....	13
1.3 ΚΑΤΑΓΡΑΦΗ ΑΠΑΙΤΗΣΕΩΝ	14
1.4 ΤΟ ΜΟΝΤΕΛΟ ΤΟΥ ΠΕΔΙΟΥ ΠΡΟΒΛΗΜΑΤΟΣ.....	15
1.5 ΤΟ ΜΟΝΤΕΛΟ ΤΩΝ ΠΕΡΙΠΤΩΣΕΩΝ ΧΡΗΣΗΣ	16
1.6 ΔΥΝΑΜΙΚΑ ΜΟΝΤΕΛΑ.....	18
1.7 ΔΙΑΓΡΑΜΜΑΤΑ ΑΛΛΗΛΕΠΙΔΡΑΣΗΣ.....	19
1.8 ΔΙΑΓΡΑΜΜΑΤΑ ΚΛΑΣΕΩΝ.....	20
ΕΠΙΛΟΓΟΣ.....	22
ΚΕΦΑΛΑΙΟ 2.....	22
ΓΛΩΣΣΑ ΜΟΝΤΕΛΟΠΟΙΗΣΗΣ UML.....	22
ΕΙΣΑΓΩΓΗ.....	22
2.1 ΜΕΘΟΔΟΛΟΓΙΕΣ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ	23
2.2 ΤΥΠΟΙ ΔΙΑΓΡΑΜΜΑΤΩΝ ΤΗΣ UML	24
ΕΠΙΛΟΓΟΣ.....	25
ΚΕΦΑΛΑΙΟ 3.....	26
ΤΕΧΝΟΛΟΓΙΕΣ: JAVA & SQLite Administrator	26
ΕΙΣΑΓΩΓΗ.....	26
3.1 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΗΣ JAVA	27
3.1.1 Η ΕΙΚΟΝΙΚΗ ΜΗΧΑΝΗ ΤΗΣ JAVA	28
3.1.2 Ο ΣΥΛΛΕΚΤΗΣ ΑΠΟΡΡΙΜΑΤΩΝ (GARBAGE COLLECTOR)	29
3.1.3 ΕΠΙΔΟΣΕΙΣ.....	29
3.1.4 ΕΡΓΑΛΕΙΑ ΑΝΑΠΤΥΞΗΣ	30
3.2 ΟΛΟΚΛΗΡΩΜΕΝΟ ΠΕΡΙΒΑΛΛΟΝ ΑΝΑΠΤΥΞΗΣ (IDE).....	30

3.2.1 Η ΠΛΑΤΦΟΡΜΑ NETBEANS	30
3.2.2 NETBEANS IDE	31
3.3 JDBC.....	32
3.3.1 ΛΕΙΤΟΥΡΓΙΑ.....	32
3.3.2 ΠΑΡΑΔΕΙΓΜΑΤΑ	33
3.4 SQLite Administrator DATABASE	34
3.5 JAVA DB	34
3.5.1 ΠΑΡΑΔΕΙΓΜΑΤΑ	35
ΕΠΙΛΟΓΟΣ.....	37
ΚΕΦΑΛΑΙΟ 4.....	38
ΣΧΕΔΙΑΣΜΟΣ	38
ΕΙΣΑΓΩΓΗ.....	38
4.1 ΠΕΡΙΠΤΩΣΕΙΣ ΧΡΗΣΕΙΣ.....	38
4.2 ΔΙΑΓΡΑΜΜΑΤΑ ΠΕΡΙΠΤΩΣΕΩΝ ΧΡΗΣΗΣ	50
4.2.1 ΔΙΑΓΡΑΜΜΑ ΠΕΡΙΠΤΩΣΕΩΝ ΧΡΗΣΗΣ ΣΥΣΤΗΜΑΤΟΣ.....	50
4.2.2 ΔΙΑΓΡΑΜΜΑ ΠΕΡΙΠΤΩΣΕΩΝ ΧΡΗΣΗΣ ΥΠΟΨΗΦΙΟΥ.....	51
4.2.3 ΔΙΑΓΡΑΜΜΑ ΠΕΡΙΠΤΩΣΕΩΝ ΧΡΗΣΗΣ ΓΡΑΜΜΑΤΕΑ	52
4.2.4 ΔΙΑΓΡΑΜΜΑ ΠΕΡΙΠΤΩΣΕΩΝ ΧΡΗΣΗΣ ΕΠΙΤΡΟΠΗΣ	53
4.2.5 ΔΙΑΓΡΑΜΜΑ ΠΕΡΙΠΤΩΣΕΩΝ ΧΡΗΣΗΣ ΔΙΑΧΕΙΡΙΣΤΗ	54
4.3 ΠΙΝΑΚΑΣ ΧΡΗΣΤΩΝ ΣΤΟΧΩΝ	55
4.4 ΚΑΤΑΛΟΓΟΣ ΣΧΕΣΕΩΝ ΕΝΝΟΙΩΝ.....	56
4.5 ΚΑΤΑΛΟΓΟΣ ΕΝΝΟΙΩΝ	57
4.6 ΕΝΝΟΙΟΛΟΓΙΚΟ ΜΟΝΤΕΛΟ (DOMAIN MODEL).....	58
4.7 ΔΙΑΓΡΑΜΜΑΤΑ ΑΚΟΛΟΥΘΙΑΣ ΣΥΣΤΗΜΑΤΟΣ	59
4.7.1 ΕΙΣΟΔΟΣ ΣΤΟ ΣΥΣΤΗΜΑ	59
4.7.2 ΔΗΜΙΟΥΡΓΙΑ ΚΩΔΙΚΩΝ.....	59
4.7.3 ΠΡΟΒΟΛΗ ΒΙΟΓΡΑΦΙΚΩΝ.....	60
4.7.4 ΚΑΤΑΧΩΡΗΣΗ ΣΤΟΙΧΕΙΩΝ.....	60
4.7.5 ΕΝΗΜΕΡΩΣΗ ΣΤΟΙΧΕΙΩΝ.....	61
4.7.6 ΔΙΑΓΡΑΦΗ ΥΠΟΨΗΦΙΩΝ.....	61
4.8 ΔΙΑΓΡΑΜΜΑΤΑ ΑΚΟΛΟΥΘΙΑΣ	62
4.8.1 ΕΙΣΟΔΟΣ ΣΤΟ ΣΥΣΤΗΜΑ	62
4.8.2 ΔΗΜΙΟΥΡΓΙΑ ΚΩΔΙΚΩΝ	62
4.8.3 ΕΝΗΜΕΡΩΣΗ ΣΤΟΙΧΕΙΩΝ.....	63
4.8.4 ΔΙΑΓΡΑΦΗ ΥΠΟΨΗΦΙΩΝ.....	63
4.8.5 ΠΡΟΒΟΛΗ ΒΙΟΓΡΑΦΙΚΩΝ.....	64

4.8.6 ΚΑΤΑΧΩΡΗΣΗ ΣΤΟΙΧΕΙΩΝ.....	64
4.9 ΔΙΑΓΡΑΜΜΑΤΑ ΣΥΝΕΡΓΑΣΙΑΣ.....	65
4.9.1 ΕΙΣΟΔΟΣ ΣΤΟ ΣΥΣΤΗΜΑ	65
4.9.2 ΕΝΗΜΕΡΩΣΗ ΣΤΟΙΧΕΙΩΝ.....	66
4.9.3 ΠΡΟΒΟΛΗ ΒΙΟΓΡΑΦΙΚΩΝ.....	67
4.9.4 ΔΙΑΓΡΑΦΗ ΥΠΟΨΗΦΙΩΝ.....	67
4.9.5 ΚΑΤΑΧΩΡΗΣΗ ΣΤΟΙΧΕΙΩΝ	68
4.10 ΔΙΑΓΡΑΜΜΑ ΚΛΑΣΕΩΝ.....	68
4.11 ΔΙΑΓΡΑΜΜΑ ΟΝΤΟΤΗΤΩΝ ΣΥΣΧΕΤΙΣΕΩΝ	69
ΕΠΙΛΟΓΟΣ.....	69
ΚΕΦΑΛΑΙΟ 5.....	69
ΥΛΟΠΟΙΗΣΗ	69
ΕΙΣΑΓΩΓΗ.....	69
5.1 ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ.....	70
5.2 ΕΙΣΟΔΟΣ ΣΤΟ ΣΥΣΤΗΜΑ	71
5.3 ΚΑΤΑΧΩΡΗΣΗ/ΕΝΗΜΕΡΩΣΗ ΒΙΟΓΡΑΦΙΚΩΝ	72
5.4 ΠΡΟΒΟΛΗ /ΔΙΑΓΡΑΦΗ ΒΙΟΓΡΑΦΙΚΩΝ.....	77
5.5 ΕΓΓΡΑΦΗ ΥΠΟΨΗΦΙΩΝ.....	80
Κατά τον υπολογισμό της προ우πηρεσίας, εκτελείται ο παρακάτω κώδικας:	82
ΕΠΙΛΟΓΟΣ.....	83
ΣΥΜΠΕΡΑΣΜΑΤΑ.....	83
ΑΡΧΙΚΟΣ ΣΤΟΧΟΣ	83
ΤΕΛΙΚΗ ΥΛΟΠΟΙΗΣΗ	83
ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ ΚΑΙ ΒΕΛΤΙΩΣΕΙΣ.....	83
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	84
ΓΛΩΣΣΑΡΙΟ	85
ΟΔΗΓΟΣ ΧΡΗΣΗΣ ΛΟΓΙΣΜΙΚΟΥ.....	86
ΥΠΟΨΗΦΙΟΣ	86
ΕΠΙΤΡΟΠΗ	86
ΓΡΑΜΜΑΤΕΑΣ	86

ΕΥΡΕΤΗΡΙΟ ΕΙΚΟΝΩΝ

Εικόνα 1 “Παράδειγμα: Μοντέλο πεδίου Τραπεζικού Συστήματος”	16
Εικόνα 2 “Παράδειγμα: Διάγραμμα Περιπτώσεων Χρήσης διεργασίας πώλησης”	18
Εικόνα 3 “Παράδειγμα διαγράμματος ακολουθίας για μια διεργασία online κράτησης”	20
Εικόνα 4 “Παράδειγμα Διαγράμματος Κλάσεων για μία διεργασία πώλησης”	21
Εικόνα 5 "Το λογότυπο της Java"	26
Εικόνα 6 "Διάγραμμα Περιπτώσεων Χρήσης Συστήματος Διαχείρισης Βιογραφικών"	50
Εικόνα 7 "Διάγραμμα Περιπτώσεων Χρήσης Υποψηφίου"	51
Εικόνα 8 "Διάγραμμα Περιπτώσεων Χρήσης Γραμματέα"	52
Εικόνα 9 "Διάγραμμα Περιπτώσεων Χρήσης Επιτροπής"	53
Εικόνα 10 "Διάγραμμα Περιπτώσεων Χρήσης Διαχειριστή"	54
Εικόνα 11"Εννοιολογικό Μοντέλο Πεδίου Προβλήματος(Domain Model)"	58
Εικόνα 12 "Διάγραμμα Ακολουθίας Συστήματος για την Π.Χ: Είσοδος στο Σύστημα"	59
Εικόνα 13 "Διάγραμμα Ακολουθίας Συστήματος για την Π.Χ: Δημιουργία Κωδικών"	60
Εικόνα 14 "Διάγραμμα Ακολουθίας Συστήματος για την Π.Χ: Προβολή Βιογραφικών"	60
Εικόνα 15 "Διάγραμμα Ακολουθίας Συστήματος για την Π.Χ: Καταχώρηση Στοιχείων"	61
Εικόνα 16 "Διάγραμμα Ακολουθίας Συστήματος για την Π.Χ: Ενημέρωση Στοιχείων"	61
Εικόνα 17"Διάγραμμα Ακολουθίας Συστήματος για την Π.Χ: Διαγραφή Υποψηφίων"	61
Εικόνα 18 "Διάγραμμα Ακολουθίας για την Π.Χ: Είσοδος στο Σύστημα"	62
Εικόνα 19 "Διάγραμμα Ακολουθίας για την Π.Χ: Δημιουργία Κωδικών"	63
Εικόνα 20 "Διάγραμμα Ακολουθίας για την Π.Χ: Ενημέρωση Στοιχείων"	63
Εικόνα 21 "Διάγραμμα Ακολουθίας για την Π.Χ: Διαγραφή Υποψηφίων"	64
Εικόνα 22 "Διάγραμμα Ακολουθίας για την Π.Χ: Προβολή Βιογραφικών"	64
Εικόνα 23 "Διάγραμμα Ακολουθίας για την Π.Χ: Καταχώρηση Στοιχείων"	64
Εικόνα 24 "Διάγραμμα Συνεργασίας για την Π.Χ: Είσοδος στο Σύστημα"	65
Εικόνα 25 "Διάγραμμα Συνεργασίας για την Π.Χ: Ενημέρωση Στοιχείων"	66
Εικόνα 26 "Διάγραμμα Συνεργασίας για την Π.Χ: Προβολή Βιογραφικών"	67
Εικόνα 27 "Διάγραμμα Συνεργασίας για την Π.Χ: Διαγραφή Υποψηφίων"	67
Εικόνα 28 "Διάγραμμα Συνεργασίας για την Π.Χ: Καταχώρηση Στοιχείων"	68
Εικόνα 29 "Διάγραμμα Κλάσεων Συστήματος Διαχείρισης Βιογραφικών"	68
Εικόνα 30 "Εννοιολογικό Διάγραμμα Οντοτήτων Συσχετίσεων της βάσης του συστήματος"	69
Εικόνα 31 "Παράθυρο εισαγωγής όνομα χρήστη και κωδικού"	71
Εικόνα 32 "Παράθυρο μηνύματος σφάλματος"	73
Εικόνα 33 "Φόρμα Καταχώρησης Βιογραφικού-Υποψήφιος"	78
Εικόνα 34 "Φόρμα Εγγραφής Υποψηφίου-Γραμματεία"	78
Εικόνα 35 "Φόρμα Προβολής Βιογραφικών-Επιτροπή"	80
Εικόνα 36 "Φόρμα Επεξεργασίας Χρηστών, Μαθημάτων Και E-mail-Διαχειριστής"	77
Εικόνα 37 "Φόρμα Ενημέρωσης Βιογραφικού"	78
Εικόνα 38 "Φόρμα Αναζήτησης Βιογραφικού"	79
Εικόνα 39 "Επιλογή Διαγραφής Υποψηφίου"	79
Εικόνα 4 "Μήνυμα Σφάλματος Διαγραφής Υποψηφίου"	80
Εικόνα 44 "Παράθυρο Εγγραφής Υποψηφίου"	81

ΕΥΡΕΤΗΡΙΟ ΠΙΝΑΚΩΝ

Πίνακας 1 "Πίνακας Χρηστών Στόχων"	55
Πίνακας 2 "Κατάλογος Σχέσεων Εννοιών"	56
Πίνακας 3 "Κατάλογος Εννοιών"	57

ΕΙΣΑΓΩΓΗ

Τα τελευταία χρόνια η κυρίαρχη προσέγγιση για την ανάπτυξη λογισμικού είναι η αντικειμενοστρεφής προσέγγιση, σύμφωνα με την οποία η ανάπτυξη των συστημάτων βασίζεται σε αντικείμενα. Μια κλάση μπορεί να θεωρηθεί ως η περιγραφή της δομής και της συμπεριφοράς των αντικειμένων που ανήκουν σε αυτή, τα οποία είναι τα συστατικά στοιχεία λογισμικού που έχουν τη δομή και εκδηλώνουν τη συμπεριφορά. Μερικά από τα πλεονεκτήματα της αντικειμενοστρεφούς ανάπτυξης συστημάτων λογισμικού είναι η απλότητα στην ανάπτυξη, η καλύτερη ποιότητα και η αξιοπιστία των παραγόμενων συστημάτων, και τέλος η δυνατότητα επαναχρησιμοποίησης των συστατικών αυτών.

Στα πλαίσια αυτά η Ενοποιημένη Γλώσσα Μοντελοποίησης (UML) αποτελεί την πλέον δημοφιλή και πρότυπη γλώσσα για την οπτικοποίηση, προσδιορισμό και τεκμηρίωση συστημάτων λογισμικού. Σήμερα χιλιάδες εταιρίες χρησιμοποιούν τη γλώσσα UML στην ανάπτυξη των συστημάτων τους.

Είναι βέβαιο λοιπόν ότι η UML αποτελεί απαραίτητο εφόδιο για τον επαγγελματία πληροφορικό-αναλυτή, σχεδιαστή, προγραμματιστή- ή οποιονδήποτε άλλον επιθυμεί να αναπτύξει λογισμικό. Επιπλέον η χρησιμότητα της γλώσσας UML δεν περιορίζεται στο χώρο της πληροφορικής μια και η ίδια γλώσσα μπορεί να χρησιμοποιηθεί για τη μοντελοποίηση επιχειρηματικών διαδικασιών και συνεπώς η χρησιμότητά της επεκτείνεται και στο χώρο διοίκησης επιχειρήσεων (Γερογιάννης, Κακαρόντζας, Καμέας, Σταμέλος, Φιτσιλής, 2006).

Το θέμα που πραγματεύεται η συγκεκριμένη πτυχιακή εργασία είναι η ανάλυση, ο σχεδιασμός και η υλοποίηση ενός αντικειμενοστρεφούς συστήματος διαχείρισης βιογραφικών υποψηφίων μελών ΕΠ το οποίο θα διαχειρίζεται την προύπηρεσία των υποψηφίων, υπολογίζοντας χρονικά τις επικαλύψεις, το δημοσιευμένο έργο των υποψηφίων και θα ταξινομεί τα διαθέσιμα στοιχεία με βάση μερικών κριτηρίων. Το σύστημα θα έχει 4 κατηγορίες χρηστών: ΥΠΟΨΗΦΙΟΣ, ΕΠΙΤΡΟΠΗ,

ΓΡΑΜΜΑΤΕΙΑ και ΔΙΑΧΕΙΡΙΣΤΗΣ. Ο χρήστης ΥΠΟΨΗΦΙΟΣ θα έχει τη δυνατότητα να εγγράφεται στο σύστημα για να καταχωρεί και να ενημερώνει το βιογραφικό του. Ο χρήστης ΕΠΙΤΡΟΠΗ θα έχει τη δυνατότητα να διαχειρίζεται τα βιογραφικά των υποψηφίων, όπως αναφέρθηκε παραπάνω, ο χρήστης ΓΡΑΜΜΑΤΕΙΑ θα δημιουργεί κωδικούς για τους χρήστες που επιθυμούν να εγγραφούν στο σύστημα και ο χρήστης ΔΙΑΧΕΙΡΙΣΤΗΣ θα προσθέτει και θα διαγράφει χρήστες, θα επεξεργάζεται μαθήματα και θα καταχωρεί τις επιλογές του e-mail server .

Το 1^ο κεφάλαιο της πτυχιακής αναλύει τις έννοιες της αντικειμενοστρεφούς ανάλυσης και σχεδίασης, των περιπτώσεων χρήσης και των διαγραμμάτων αλληλεπίδρασης, καθώς και του εννοιολογικού μοντέλου και του μοντέλου υλοποίησης. Στο 2^ο κεφάλαιο περιγράφεται η γλώσσα μοντελοποίησης UML, αναλύονται οι μεθοδολογίες ανάπτυξης λογισμικού δίνοντας έμφαση στην Ενοποιημένη Προσέγγιση (UP) και περιγράφονται όλοι οι τύποι διαγραμμάτων της UML. Το κεφάλαιο 3 παρουσιάζει τις τεχνολογίες Java και SQLite και τις πλατφόρμες που χρησιμοποιήθηκαν για την ανάπτυξη του συστήματος, όπως η πλατφόρμα NetBeans και η JDBC.

Το κεφάλαιο 4 περιλαμβάνει την ανάλυση και το σχεδιασμό του συστήματος με τη γλώσσα UML. Παρουσιάζονται οι περιπτώσεις χρήσης του συστήματος, διαγραμματικά και λεκτικά, τα διαγράμματα ακολουθίας συστήματος, τα διαγράμματα ακολουθίας και συνεργασίας, τα διαγράμματα εννοιολογικού μοντέλου και μοντέλου υλοποίησης-κλάσεων καθώς και το εννοιολογικό μοντέλο της βάσης δεδομένων (Entity-Relationship model –ER). Το κεφάλαιο 5 εστιάζει στην ανάπτυξη του συστήματος, περιγράφοντας όλες τις λειτουργίες του συστήματος καθώς και κάποια τμήματα της υλοποίησής του. Στο κεφάλαιο Συμπεράσματα αναφέρονται ο αρχικός στόχος και η τελική υλοποίηση του συστήματος καθώς και κάποιες ιδέες για μελλοντικές επεκτάσεις και βελτιώσεις. Τέλος, παρουσιάζονται κάποιες οδηγίες για την εγκατάσταση και χρήση του συστήματος.

ΚΕΦΑΛΑΙΟ 1

ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΑΝΑΛΥΣΗ ΚΑΙ ΣΧΕΔΙΑΣΗ

ΕΙΣΑΓΩΓΗ

Η Ανάλυση δίνει έμφαση στην διερεύνηση του προβλήματος και των απαιτήσεων παρά στην επίλυση. Για παράδειγμα, αν πρόκειται να υλοποιηθεί ένα νέο πληροφοριακό σύστημα βιβλιοθήκης, πώς θα χρησιμοποιείται; Η Ανάλυση είναι ένας ευρύς όρος, που εκφράζεται καλύτερα στην ανάλυση απαιτήσεων (μια

διερεύνηση των απαιτήσεων) ή την ανάλυση αντικειμένων (μια διερεύνηση των κύριων αντικειμένων).

Η Σχεδίαση επικεντρώνεται σε μία εννοιολογική επίλυση που ικανοποιεί τις απαιτήσεις, παρά στην υλοποίησή της. Για παράδειγμα, η περιγραφή ενός σχήματος βάσεων δεδομένων και των αντικειμένων λογισμικού. Τελικά τα σχέδια μπορούν να υλοποιηθούν.

Η ανάλυση απαιτήσεων μπορεί να περιλαμβάνει μια περιγραφή των συσχετιζόμενων διεργασιών του πεδίου. Αυτές μπορούν να γραφούν ως περιπτώσεις χρήσης. Οι περιπτώσεις χρήσης δεν είναι ένα αντικειμενοστρεφές τεχνούργημα. Είναι απλά καταγραμμένες ιστορίες. Ωστόσο είναι ένα δημοφιλές εργαλείο στην ανάλυση απαιτήσεων και ένα σημαντικό κομμάτι της ενοποιημένης προσέγγισης (Unified Process-UP). Για παράδειγμα, παρατίθεται μια σύντομη έκδοση της περίπτωσης χρήσης του παιχνιδιού Ρίξιμο Ζαριού: Ένας παίκτης μαζεύει και ρίχνει τα ζάρια. Αν τα ζάρια κάνουν άθροισμα επτά κερδίζει, αλλιώς χάνει.

Η αντικειμενοστρεφής ανάλυση σχετίζεται με τη δημιουργία μιας περιγραφής του πεδίου από την οπτική της κατηγοριοποίησης από αντικείμενα. Μία αποσύνθεση του πεδίου περιλαμβάνει την αναγνώριση των εννοιών, των χαρακτηριστικών και των συσχετίσεων που αναγνωρίζονται. Το αποτέλεσμα μπορεί να εκφραστεί ως ένα εννοιολογικό μοντέλο το οποίο αποτυπώνεται σε ένα σύνολο διαγραμμάτων που δείχνουν τις έννοιες του πεδίου ή αντικείμενα.

Επιπρόσθετα στη δυναμική απεικόνιση που προσφέρουν τα διαγράμματα αλληλεπίδρασης, είναι χρήσιμο να δημιουργείται μια στατική απεικόνιση των ορισμών των κλάσεων με ένα διάγραμμα σχεδιασμού κλάσεων. Αυτό απεικονίζει τις ιδιότητες και τις μεθόδους των κλάσεων (Larman,2002).

1.1 ΠΡΟΣΑΝΑΤΟΛΙΣΜΟΣ ΣΕ ΑΝΤΙΚΕΙΜΕΝΑ

Ο προσανατολισμός σε αντικείμενα (object orientation) είναι μια προσέγγιση στην ανάπτυξη λογισμικού που οργανώνει το εκάστοτε πρόβλημα και τη λύση του με τη μορφή συλλογής διακριτών αντικειμένων. Τόσο η δομή των δεδομένων όσο και η συμπεριφορά του αντικειμένου περιλαμβάνονται στην αναπαράσταση. Μπορούμε να ξεχωρίσουμε μια αντικειμενοστρεφή αναπαράσταση από 7 χαρακτηριστικά της: ταυτότητα, αφαίρεση, κατηγοριοποίηση, ενθυλάκωση, κληρονομικότητα, πολυμορφισμό και διατήρηση. Η ταυτότητα (identity) αναφέρεται στο γεγονός ότι τα δεδομένα οργανώνονται σε διακριτές, ξεχωριστές οντότητες που ονομάζονται αντικείμενα. Ένα απλό αντικείμενο διαθέτει καταστάσεις και συμπεριφορές που σχετίζονται μαζί του. Η αφαίρεση είναι θεμελιώδης για την κατασκευή οπουδήποτε συστήματος είτε αυτό είναι αντικειμενοστρεφές είτε όχι. Οι αφαιρέσεις σε ένα αντικειμενοστρεφές σύστημα βοηθούν στην αναπαράσταση των διαφορετικών οπτικών γωνιών που ενσωματώνονται στο σύστημα που αναπτύσσεται. Ο

προσανατολισμός σε αντικείμενα χρησιμοποιεί την κατηγοριοποίηση για να ομαδοποιήσει αντικείμενα που διαθέτουν κοινά χαρακτηριστικά και συμπεριφορές. Μπορούμε να οργανώσουμε ιεραρχικά τις κλάσεις ανάλογα με τις ομοιότητες και διαφορές που παρουσιάζουν. Αυτή η ιεραρχία αναπαριστά τη δομή κληρονομικότητας των αντικειμενοστρεφών κλάσεων.

Η συμπεριφορά είναι μια ενέργεια ή μετασχηματισμός που ένα αντικείμενο εκτελεί ή στην οποία υπόκειται. Μερικές φορές η ίδια συμπεριφορά μπορεί να επιδειχθεί με διαφορετικό τρόπο σε διαφορετικές κλάσεις ή υποκλάσεις, μια ιδιότητα που ονομάζεται πολυμορφισμός. Μια συγκεκριμένη υλοποίηση μιας συμπεριφοράς για κάποια κλάση ονομάζεται μέθοδος. Σε ένα σύστημα με πολυμορφισμό, σε μια λειτουργία μπορούν να αντιστοιχούν περισσότερες από μία μεθόδους που την υλοποιούν.

Η διατήρηση (persistence) είναι η δυνατότητα του ονόματος, της κατάστασης, και των συμπεριφορών ενός αντικειμένου να υπερβαίνουν το χώρο ή το χρόνο. Με άλλα λόγια το όνομα, η κατάσταση και οι συμπεριφορές του αντικειμένου διατηρούνται καθώς μετασχηματίζεται το αντικείμενο (Pfleeger, 2001).

1.2 ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΑΝΑΛΥΣΗ ΚΑΙ ΣΧΕΔΙΑΣΗ

Η Αντικειμενοστρεφής Ανάλυση αναφέρεται στο πεδίο του προβλήματος με στόχο να παράγει ένα εννοιολογικό μοντέλο των πληροφοριών που υπάρχουν στην περιοχή που αναλύεται. Τα μοντέλα ανάλυσης δε λαμβάνουν υπόψη περιορισμούς υλοποίησης που μπορεί να υπάρχουν όπως ταυτοχρονισμός, κατανομή, διατήρηση, ή πώς πρέπει να υλοποιηθεί το σύστημα. Οι περιορισμοί υλοποίησης λαμβάνονται υπόψη κατά τη διάρκεια του Αντικειμενοστρεφούς Σχεδιασμού. Η Ανάλυση γίνεται πριν το Σχεδιασμό.

Οι πηγές για την ανάλυση μπορεί να είναι καταγραμμένες απαιτήσεις, ένα τυπικό οπτικό έγγραφο, συνεντεύξεις με τους πελάτες ή άλλα ενδιαφερόμενα μέρη. Ένα σύστημα μπορεί να διαιρεθεί σε πολλαπλά πεδία αναπαριστώντας διαφορετικές διεργασίες, τεχνολογικές ή άλλες, κάθε μία από τις οποίες αναλύεται ξεχωριστά.

Το αποτέλεσμα της Αντικειμενοστρεφούς Ανάλυσης είναι μια περιγραφή του τι το σύστημα απαιτείται να κάνει στα πλαίσια ενός εννοιολογικού μοντέλου. Αυτό τυπικά αναπαρίσταται ως ένα σύνολο περιπτώσεων χρήσης, ενός ή περισσότερων UML διαγραμμάτων, και ενός αριθμού διαγραμμάτων αλληλεπίδρασης. Μπορεί επίσης να περιλαμβάνει κάποια πρότυπη διεπιφάνεια χρήστη. Ο σκοπός της Αντικειμενοστρεφούς Ανάλυσης είναι να αναπτύξει ένα μοντέλο που να περιγράφει τη λειτουργία ενός λογισμικού ώστε να ικανοποιεί ένα σύνολο προκαθορισμένων απαιτήσεων.

Ο Αντικειμενοστρεφής Σχεδιασμός μετατρέπει το εννοιολογικό μοντέλο που παράγεται στην Αντικειμενοστρεφή Ανάλυση ώστε να λαμβάνει υπόψη τους

περιορισμούς που τίθενται από την επιλεγμένη αρχιτεκτονική και κάθε μη λειτουργική, τεχνολογική ή περιβαλλοντική απαίτηση, όπως τα αποτελέσματα συναλλαγών, ο χρόνος απόκρισης, το περιβάλλον ανάπτυξης ή η προγραμματιστική γλώσσα. Οι έννοιες της Αντικειμενοστρεφούς Ανάλυσης αντιστοιχίζονται σε κλάσεις υλοποίησης και διασυνδέσεις. Το αποτέλεσμα είναι ένα μοντέλο επίλυσης του πεδίου, μια λεπτομερής περιγραφή του πώς το σύστημα θα υλοποιηθεί (Object Oriented Analysis and Design).

1.3 ΚΑΤΑΓΡΑΦΗ ΑΠΑΙΤΗΣΕΩΝ

Η καταγραφή και η ανάλυση των απαιτήσεων έχει ως στόχο να απαντήσει στη βασική ερώτηση «τι πρέπει να κάνει το σύστημα». Πρόκειται για μια διαδικασία ιδιαίτερα επίπονη και δύσκολη, καθώς πρέπει να κατανοήσουμε τις πραγματικές ανάγκες των χρηστών από το σύστημα και στη συνέχεια να τις καταγράψουμε. Ορισμένα από τα χαρακτηριστικά γνωρίσματα της καταγραφής των απαιτήσεων είναι:

- Στις περισσότερες περιπτώσεις, το σύστημα απευθύνεται σε πολλούς χρήστες, οι οποίοι έχουν γνώση ενός μικρού μόνο μέρους της συνολικής εργασίας που καλείται να υποστηρίξει το πληροφοριακό σύστημα.
- Κάθε έργο ανάπτυξης λογισμικού είναι μοναδικό. Η μοναδικότητα αυτή προέρχεται από τη διαφορετικότητα του πεδίου προβλήματος, τη διαφορετικότητα του οργανισμού που θα χρησιμοποιεί το σύστημα κ.τ.λ.
- Το σημείο εκκίνησης για την καταγραφή των απαιτήσεων είναι διαφορετικό για κάθε έργο αφού μπορεί να ξεκινάει από μια πολύ σύντομη περιγραφή της γενικής ιδέας ή από την αναλυτική περιγραφή των αναγκών της επιχείρησης.

Είναι προφανές λοιπόν ότι η καταγραφή των απαιτήσεων είναι μια διαδικασία σύνθετη, η οποία διαφέρει από έργο σε έργο. Στη γενική περίπτωση, η καταγραφή και η ανάλυση των απαιτήσεων περιλαμβάνει τα ακόλουθα βήματα:

1. Προσδιορισμός του μοντέλου του πεδίου προβλήματος. Το μοντέλο του προβλήματος περιγράφει τις βασικές οντότητες-έννοιες που αποτελούν το σύστημα.
2. Προσδιορισμός του αντικείμενου εργασιών του συστήματος (χάρτης εμβέλειας, *scope map*). Προσδιορίζει με επιγραμματικό τρόπο τις απαιτήσεις του συστήματος.
3. Προσδιορισμός του μοντέλου των περιπτώσεων χρήσης. Προσδιορίζει τον τρόπο με τον οποίο χρησιμοποιείται το σύστημα.
4. Οργάνωση των περιπτώσεων χρήσης σε πακέτα. Ομαδοποιεί τις περιπτώσεις χρήσεις σε λογικά πακέτα.
5. Ανάπτυξη πρωτότυπων του συστήματος με σκοπό την επίδειξη στο χρήστη βασικών παραμέτρων (Γερογιάννης et al., 2006).

1.4 ΤΟ ΜΟΝΤΕΛΟ ΤΟΥ ΠΕΔΙΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

Το μοντέλο του πεδίου προβλήματος είναι ένα από τα βασικά μοντέλα της UML. Πρωταρχικός στόχος στη δημιουργία του μοντέλου του πεδίου του προβλήματος είναι ο προσδιορισμός των αφαιρέσεων του πραγματικού κόσμου που απαιτούνται για την κατασκευή του συστήματος-δηλαδή τα κύρια εννοιολογικά αντικείμενα που πρόκειται να συμμετέχουν σε αυτό το σύστημα. Η προσέγγιση αυτή απορρέει από τη γενικότερη αρχή δημιουργίας συστημάτων με αντικειμενοστρεφή τρόπο, δηλαδή ότι το λογισμικό πρέπει να βασίζεται στα φυσικά αντικείμενα του πεδίου προβλήματος. Η αρχή αυτή βασίζεται στη θεωρία ότι ο πραγματικός κόσμος αλλάζει λιγότερο συχνά από ότι οι προδιαγραφές ενός συστήματος λογισμικού.

Ο ορισμός του μοντέλου του πεδίου προβλήματος προηγείται των περιπτώσεων χρήσης καθώς οι περιπτώσεις χρήσης πρέπει να γραφούν έτσι ώστε να λαμβάνουν υπόψη τα εννοιολογικά αντικείμενα που καθορίζονται στο εννοιολογικό μοντέλο του προβλήματος. Επιπλέον με την προσέγγιση αυτή μπορούμε να συνδέσουμε το στατικό και το δυναμικό μοντέλο του συστήματος. Συνοψίζοντας, το μοντέλο του πεδίου προβλήματος αποτελεί ένα αρχικό λεξικό όρων και εννοιών του προς ανάπτυξη συστήματος.

Κατά τον προσδιορισμό των εννοιολογικών αντικειμένων πρέπει επίσης να προσδιοριστούν και οι σχέσεις μεταξύ τους. Οι πιο σημαντικοί τύποι σχέσεων είναι τρεις:

- Η σχέση γενίκευσης/ειδίκευσης (generalization/specialization) με την οποία περιγράφουμε την κληρονομικότητα,
- Η σχέση συναρμολόγησης (aggregation) με την οποία περιγράφουμε τη σύνθεση, και τέλος,
- Οι απλές συσχετίσεις που περιγράφουν την ανάγκη ύπαρξης άλλων αντικειμένων με σκοπό την υλοποίηση μιας συμπεριφοράς.

Μια από τις πρώτες ενέργειες που πρέπει να γίνουν για τη δημιουργία του στατικού μοντέλου είναι να βρεθούν οι κατάλληλες κλάσεις που αντιπροσωπεύουν ακριβώς τις αφαιρέσεις που παρουσιάζει η περιοχή προβλήματος. Οι κλάσεις μπορεί να είναι είτε φυσικές οντότητες, είτε να αναπαριστούν γενικότερες ιδέες.

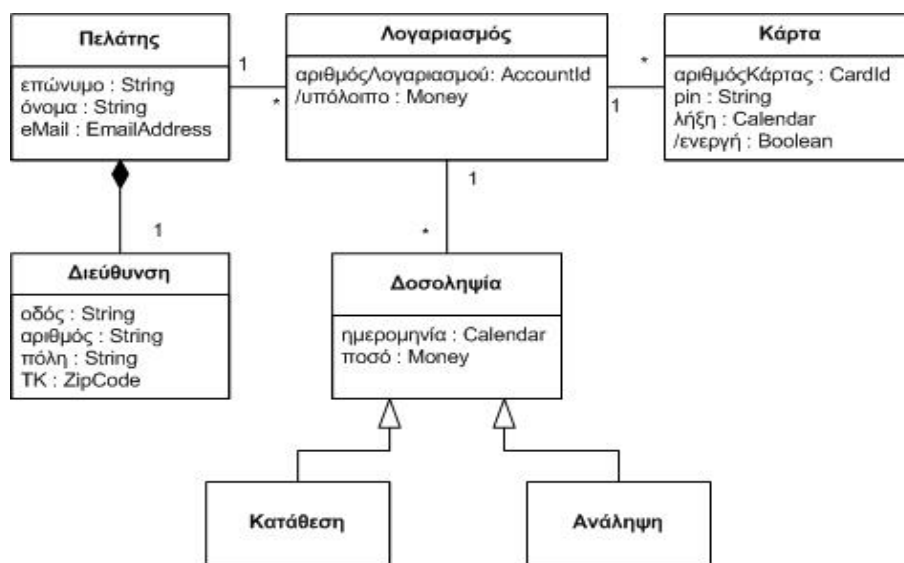
Μερικοί βασικοί κανόνες για την εύρεση των κλάσεων είναι:

- Τα ουσιαστικά και οι ονομαστικές κλάσεις αντιστοιχούν συνήθως σε κλάσεις ή πεδία κλάσεων.
- Τα ρήματα και οι ρηματικές φράσεις γίνονται συνήθως μέθοδοι κλάσεων ή συνδέσεις (associations).
- Οι κτητικές φράσεις δείχνουν ότι το ουσιαστικό μάλλον αναπαριστά μια ιδιότητα κλάσης παρά μια κλάση.

Έχοντας εντοπίσει τις βασικές κλάσεις του συστήματος, εξετάζουμε για κάθε μία από αυτές τις ακόλουθες ερωτήσεις:

- Είναι πράγμα ή είναι οικογένεια/κατηγορία πραγμάτων;
- Είναι μέρος του προβλήματος;
- Έχει δεδομένα και συμπεριφορά;
- Είναι λεπτομέρεια υλοποίησης ή όχι;
- Είναι ένα συμβάν/γεγονός;
- Είναι ιδιότητα ενός πράγματος;

Με βάση την παραπάνω ανάλυση και μετά από προσεκτική εξέταση της κάθε περίπτωσης, θα καταλήξουμε στις κλάσεις του πεδίου του προβλήματος, όπως φαίνεται στο παράδειγμα (Γερογιάννης et al., 2006):



Εικόνα 1 “Παράδειγμα: Μοντέλο πεδίου Τραπεζικού Συστήματος”

1.5 ΤΟ ΜΟΝΤΕΛΟ ΤΩΝ ΠΕΡΙΠΤΩΣΕΩΝ ΧΡΗΣΗΣ

Το μοντέλο των περιπτώσεων χρήσης του συστήματος είναι από τα σπουδαιότερα στην ανάπτυξη του συστήματος, αφού περιγράφει για ποιον αναπτύσσεται και τι πρέπει να κάνει το σύστημα. Ίσως είναι το πιο γνωστό μοντέλο της UML. Αν και το μοντέλο των περιπτώσεων χρήσης βασίζεται στο διάγραμμα περιπτώσεων χρήσης, η κύρια περιγραφή του συστήματος γίνεται με την αναλυτική περιγραφή των περιπτώσεων χρήσης και όχι απαραίτητα με τη χρήση του διαγράμματος.

Μια περίπτωση χρήσης είναι ένας διαφορετικός τρόπος με τον οποίο ένας χρήστης μπορεί πραγματικά να χρησιμοποιήσει το σύστημα. Η εκφραστική δύναμη των περιπτώσεων χρήσης βασίζεται κατά κύριο λόγο στην απλότητα και στον τρόπο οργάνωσης: προσδιορίζουμε και οργανώνουμε τις περιπτώσεις χρήσης έτσι ώστε να μπορούμε να έχουμε μια σαφέστερη εικόνα των δυνατοτήτων και των λειτουργιών του συστήματος από την οπτική γωνία του χρήστη. Χρησιμοποιώντας το μοντέλο των περιπτώσεων χρήσης μπορούμε να

παρουσιάσουμε αυτή τη σαφή εικόνα στους πελάτες και στους χρήστες του συστήματος-στοιχείο το οποίο βοηθάει στη βελτίωση της επικοινωνίας και αποτελεί τη βάση μιας αναλυτικής συζήτησης για το τι θα πρέπει, τελικά, να κάνει το σύστημα.

Οι περιπτώσεις χρήσης μπορούν να χρησιμοποιηθούν με πολλούς διαφορετικούς τρόπους και σε διαφορετικές περιπτώσεις όπως:

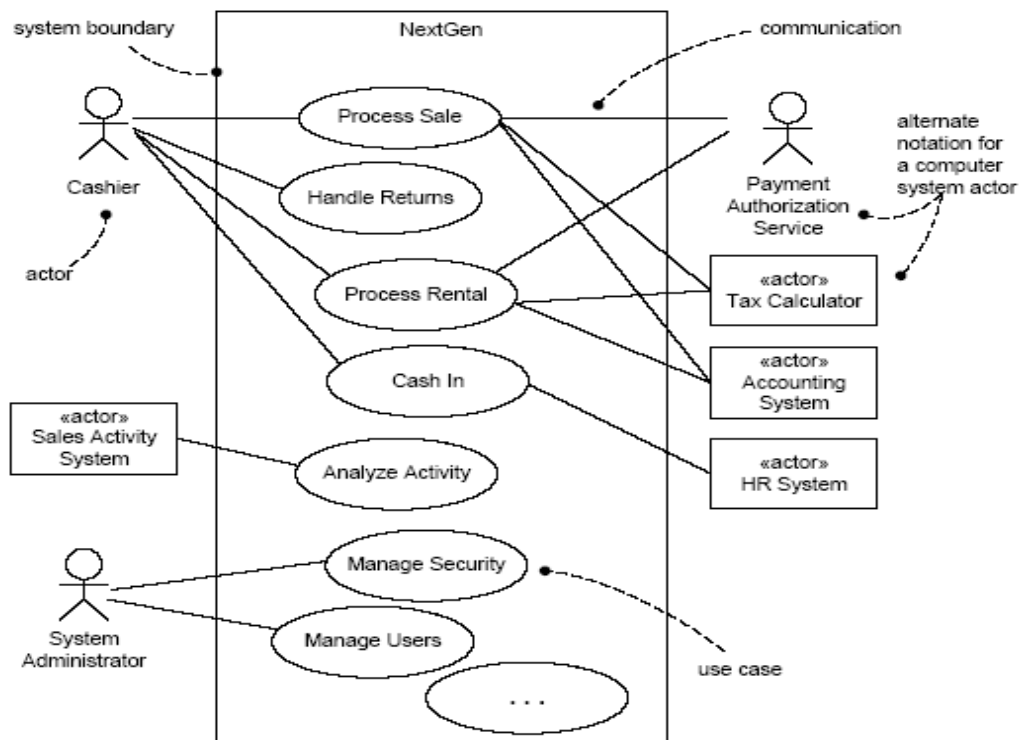
- Για την περιγραφή επιχειρηματικών διαδικασιών,
- Για την περιγραφή των προδιαγραφών του συστήματος,
- Για τον προσδιορισμό των απαιτήσεων,
- Για την τεκμηρίωση της λειτουργικότητας του συστήματος.

Η διαδικασία ανάπτυξης του μοντέλου περιπτώσεων χρήσης συνίσταται στον προσδιορισμό των εξής:

- Ποιος χρησιμοποιεί το σύστημα;
- Πώς χρησιμοποιείται το σύστημα;
- Ποιο είναι το περιβάλλον και ποια τα όρια του συστήματος;
- Πώς οργανώνεται στο σύστημα σε πακέτα;

Επιπλέον, η ανάπτυξη του μοντέλου των περιπτώσεων χρήσης είναι μια επαναληπτική διαδικασία. Η επαναληπτικότητα είναι απαραίτητη αφού δε γίνεται να προσδιορίσουμε εξ αρχής όλες τις περιπτώσεις χρήσης του συστήματος και με όλες τις απαραίτητες λεπτομέρειες. Για το λόγο αυτό, σε κάθε επανάληψη επιλέγουμε ένα σύνολο περιπτώσεων χρήσης όπου και εστιάζουμε την προσοχή μας. Η επιλογή αυτών των περιπτώσεων χρήσης γίνεται με κριτήρια: τη σπουδαιότητά τους, τη διαθεσιμότητα της πληροφορίας, τον τρόπο οργάνωσης της εργασίας σε ομάδες κ.λπ.

Η περίπτωση χρήσης αναπαριστά ένα στόχο για έναν εξωτερικό χειριστή (actor) του συστήματος. Οι χειριστές ενός συστήματος μπορεί να είναι άνθρωποι ή και εξωτερικά συστήματα. Το σύμβολο για μια περίπτωση χρήσης είναι η έλλειψη μέσα στην οποία αναγράφεται το όνομα της περίπτωσης χρήσης. Μία περίπτωση χρήσης περιλαμβάνει συνήθως πολλά εναλλακτικά σενάρια τα οποία ονομάζονται επεκτάσεις (extensions). Η ιδέα είναι ότι μια περίπτωση χρήσης έχει ένα κύριο σενάριο όπου όλα πάνε καλά (happy path) και αρκετές επεκτάσεις στις οποίες κάτι δεν πάει καλά ή προκύπτει μια εξαίρεση (Γερογιάννης et al., 2006). Στην εικόνα φαίνεται ένα διάγραμμα περιπτώσεων χρήσης ενός καταστήματος για τη διεργασία μιας πώλησης.



Εικόνα 2 “Παράδειγμα: Διάγραμμα Περιπτώσεων Χρήσης διεργασίας πώλησης”

1.6 ΔΥΝΑΜΙΚΑ ΜΟΝΤΕΛΑ

Τα δυναμικά μοντέλα απεικονίζουν τη δυναμική συμπεριφορά του συστήματος, για παράδειγμα πώς αποκρίνεται στις ενέργειες των χρηστών ή σε άλλα εξωτερικά ερεθίσματα και πώς διαμορφώνεται η εσωτερική του κατάσταση κατά τη λειτουργία του. Τα δυναμικά μοντέλα απεικονίζουν την αλληλεπίδραση μεταξύ αντικειμένων και κατά την ανάπτυξή τους ενδέχεται να προκύψουν βελτιώσεις της στατικής δομής.

Σε ένα αντικειμενοστρεφές σύστημα χρησιμοποιούνται οι εξής τύποι μοντέλων για τη μοντελοποίηση της συμπεριφοράς:

- Διάγραμμα καταστάσεων (state diagram)
- Διάγραμμα δραστηριοτήτων (activity diagram)
- Διαγράμματα αλληλεπίδρασης (interaction diagrams)
 - Διάγραμμα ακολουθίας (sequence diagram)
 - Διάγραμμα επικοινωνίας (communication diagram)

Τα διαγράμματα αλληλεπίδρασης περιέχουν και απεικονίζουν την ίδια πληροφορία που καθορίζεται από την αλληλεπίδραση και είναι συμπληρωματικά. Έχουν διαφορετική μορφή δίνοντας έμφαση σε διαφορετική οπτική γωνία και όψη της αλληλεπίδρασης.

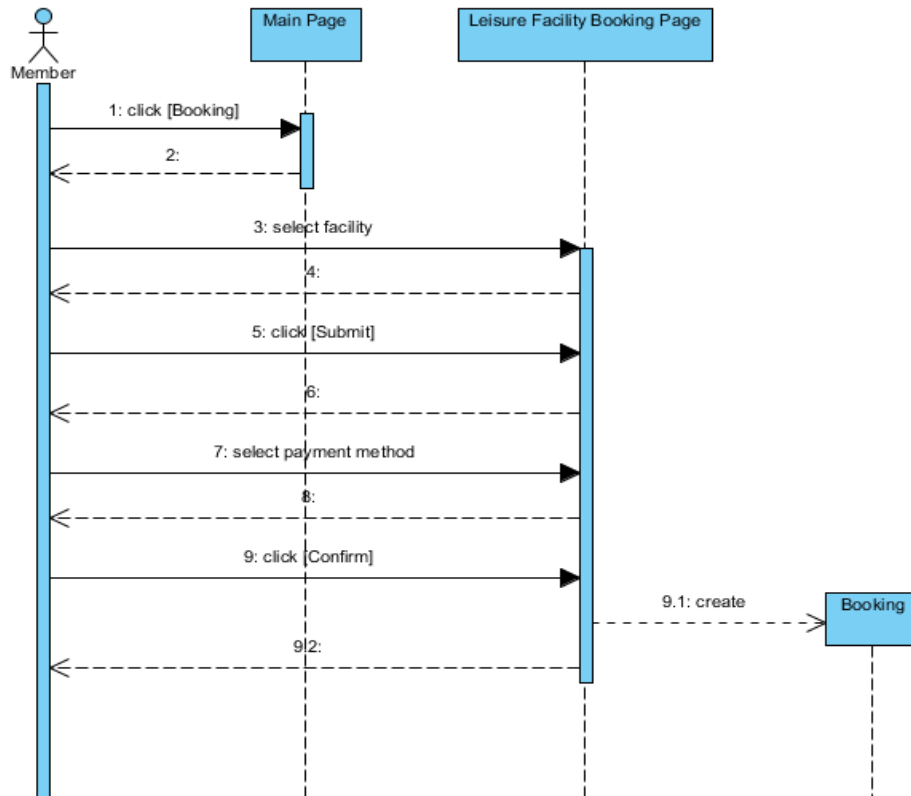
1.7 ΔΙΑΓΡΑΜΜΑΤΑ ΑΛΛΗΛΕΠΙΔΡΑΣΗΣ

Τα διαγράμματα αλληλεπίδρασης περιγράφουν τον τρόπο με τον οποίο μια ομάδα αντικειμένων συνεργάζεται. Ένα διάγραμμα συνήθως περιγράφει τη συμπεριφορά σε μια περίπτωση χρήσης. Δείχνει παραδείγματα αντικειμένων που εμπλέκονται και τα μηνύματα που ανταλλάσσονται μεταξύ τους κατά τη διάρκεια της περίπτωσης χρήσης. Η αλληλεπίδραση αποτελεί το σύνολο των μηνυμάτων που ανταλλάσσονται μεταξύ των αντικειμένων. Ένα μήνυμα μπορεί να είναι μια μονόδρομη επικοινωνία μεταξύ δύο αντικειμένων, μια ροή ελέγχου με πληροφορία από αποστολέα προς αποδέκτη ή μπορεί να μεταφέρει τιμές μέσω παραμέτρων. Η ακολουθία των μηνυμάτων παρουσιάζεται:

- ❖ Είτε με διαγράμματα ακολουθίας που εστιάζουν στη χρονική ακολουθία των μηνυμάτων
- ❖ Είτε με διαγράμματα επικοινωνίας που εστιάζουν στις σχέσεις μεταξύ των αντικειμένων που ανταλλάσσουν μηνύματα.

Το διάγραμμα ακολουθίας (sequence diagram) αναπαριστά τη χρονική σειρά εκτέλεσης των γεγονότων που λαμβάνουν χώρα κατά τη διάρκεια της αλληλεπίδρασης, ενώ το διάγραμμα επικοινωνίας αναπαριστά και τις σχέσεις μεταξύ των αντικειμένων που ωφελεί στην κατανόηση των επιπτώσεων σε ένα στιγμιότυπο/αντικείμενο. Το διάγραμμα επικοινωνίας απαιτεί λιγότερο σχεδιαστικό χώρο μια και η διάταξη των αντικειμένων είναι ελεύθερη στο χώρο και επομένως προτιμάται όταν έχουμε πολλά αντικείμενα σε μια αλληλεπίδραση ή όταν σχεδιάζουμε ένα διάγραμμα με το χέρι.

Στο διάγραμμα ακολουθίας, σε κάθε αντικείμενο αντιστοιχεί μια κάθετη γραμμή που ονομάζεται γραμμή ζωής (lifeline). Για όσο χρόνο ένα αντικείμενο υφίσταται η γραμμή αυτή είναι διακεκομμένη ενώ για όσο χρόνο μια διαδικασία του εν λόγω αντικειμένου είναι ενεργή, η γραμμή ζωής σχεδιάζεται ως μία διπλή γραμμή (με activation box). Ένα μήνυμα συμβολίζεται ως μια ακμή από τη γραμμή ζωής ενός αντικειμένου προς τη γραμμή ζωής του άλλου. Η θέση των ακμών αντιστοιχεί στην τοποθέτησή τους σε σχέση με το χρόνο (μια ακμή χαμηλότερα από μια άλλη αντιστοιχεί σε μεταγενέστερο χρόνο). Οι απαντήσεις στα μηνύματα υποδηλώνονται ως οριζόντιες διακεκομμένες γραμμές (Πρέντζα, n.d.).



Εικόνα 3 “Παράδειγμα διαγράμματος ακολουθίας για μια διεργασία online κράτησης”

1.8 ΔΙΑΓΡΑΜΜΑΤΑ ΚΛΑΣΕΩΝ

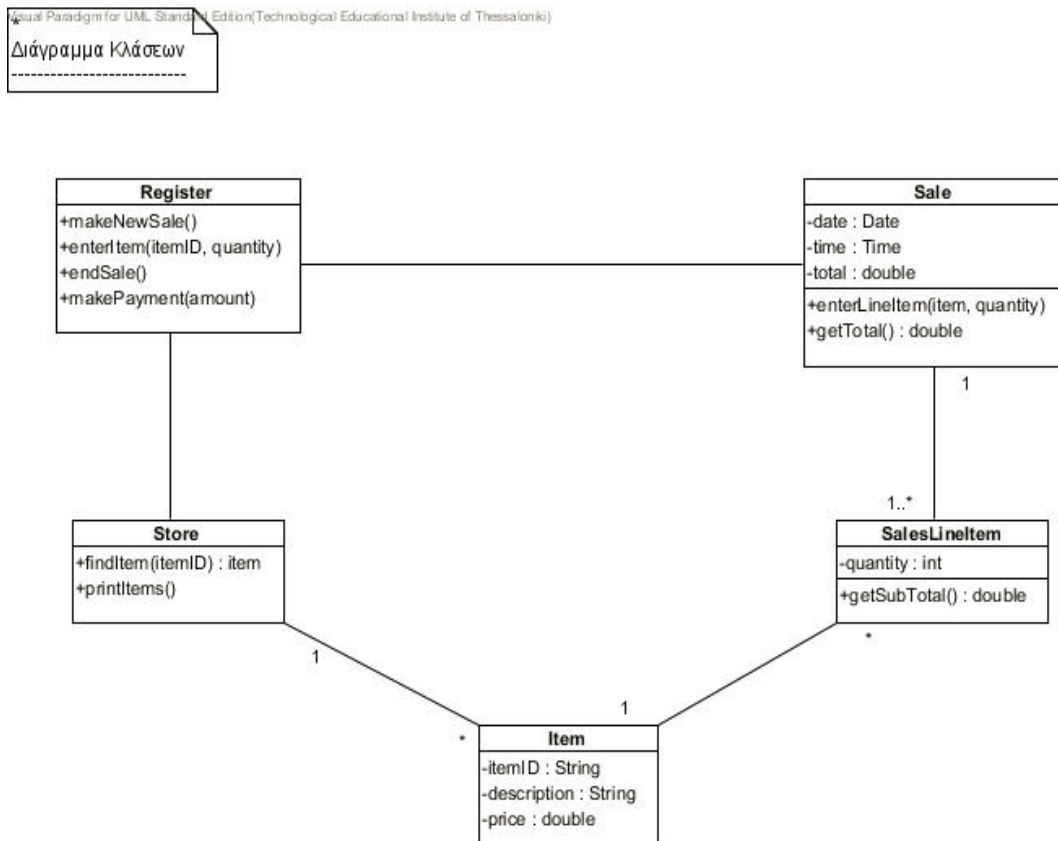
Τα αντικειμενοστρεφή συστήματα λειτουργούν ως μια συλλογή συνεργαζόμενων αντικειμένων. Τα αντικείμενα αποτελούν στιγμιότυπα κλάσεων. Η κλάση είναι ένας τύπος από τον οποίο δημιουργούνται κατά τη διάρκεια της εκτέλεσης του προγράμματος αντικείμενα που ανήκουν στον τύπο αυτό. Είναι λοιπόν χρήσιμο να καταγράφονται οι κλάσεις που ανήκουν σε ένα σύστημα και οι μεταξύ τους συσχετίσεις. Αυτό ακριβώς επιτυγχάνεται με ένα διάγραμμα κλάσεων.

Κατά τη διαδικασία της ανάλυσης προδιαγραφών οι κατασκευαστές αρχίζουν να αποκτούν γνώση του πεδίου προβλήματος του συστήματος. Αυτή η αρχική κατανόηση των εννοιών του πεδίου προβλήματος καταγράφεται σε ένα διάγραμμα κλάσεων το οποίο ονομάζεται μοντέλο του πεδίου προβλήματος (problem domain model). Στο μοντέλο αυτό καταγράφονται ως κλάσεις οι έννοιες του πεδίου προβλήματος χωρίς να υπάρχει δέσμευση πως αυτές οι ίδιες κλάσεις θα υπάρχουν και στο λογισμικό που θα κατασκευαστεί. Κατά τη φάση της επεξεργασίας, το μοντέλο του πεδίου του προβλήματος θα αποτελέσει πηγή έμπνευσης για το επίπεδο του πεδίου προβλήματος του λογισμικού, με τις περισσότερες κλάσεις να υπάρχουν και εκεί. Φυσικά οι λειτουργίες και οι συσχετίσεις μπορεί να έχουν αλλάξει. Οι λειτουργίες των κλάσεων του λογισμικού αποτελούν μεθόδους των αντικειμένων αυτών των κλάσεων, οι οποίες καλούνται

από άλλα αντικείμενα, άλλων συνεργαζόμενων κλάσεων, στο πλαίσιο της υλοποίησης ενός σεναρίου μιας περίπτωσης χρήσης. Η εύρεση των μεθόδων γίνεται με την καθοδήγηση των διαγραμμάτων αλληλεπίδρασης.

Οι κλάσεις αποτελούν τη βάση κατασκευής οποιουδήποτε αντικειμενοστρεφούς συστήματος. Ενσωματώνουν τα δεδομένα καθώς και τις λειτουργίες που επενεργούν στα δεδομένα αυτά. Συμβολίζονται με ένα παραλληλόγραμμο που έχει 3 διαμερίσματα. Στο πάνω διαμέρισμα αναγράφεται το όνομα της κλάσης, στο μεσαίο διαμέρισμα οι ιδιότητές της, και στο κάτω διαμέρισμα οι λειτουργίες της.

Μια συσχέτιση (association) μεταξύ δύο κλάσεων απεικονίζει μια στατική σχέση μεταξύ των 2 κλάσεων. Η ανάγκη μιας συσχέτισης προκύπτει από τη διαπίστωση πως για τη λειτουργία μιας κλάσης απαιτείται η συνεργασία της με μία ή περισσότερες άλλες κλάσεις. Για την αναπαράσταση μιας συσχέτισης χρησιμοποιούμε μια γραμμή μεταξύ των 2 κλάσεων (Γερογιάννης et al., 2006).



Εικόνα 4 "Παράδειγμα Διαγράμματος Κλάσεων για μία διεργασία πώλησης"

ΕΠΙΛΟΓΟΣ

Σκοπός αυτού του κεφαλαίου ήταν να κάνει μια εισαγωγή στις έννοιες της αντικειμενοστρεφούς ανάλυσης και σχεδίασης λογισμικού και να περιγράψει μεθόδους αντικειμενοστρεφούς ανάλυσης όπως η καταγραφή απαιτήσεων, οι περιπτώσεις χρήσης, τα διαγράμματα αλληλεπίδρασης και κλάσεων και το μοντέλο πεδίου προβλήματος. Στο επόμενο κεφάλαιο παρουσιάζεται η γλώσσα μοντελοποίησης UML, που αποτελεί την κυρίαρχη γλώσσα απεικόνισης αντικειμενοστρεφών συστημάτων. Παρουσιάζονται οι μεθοδολογίες ανάπτυξης λογισμικού δίνοντας έμφαση στην Ενοποιημένη Προσέγγιση (UP) και περιγράφονται οι τύποι διαγραμμάτων της UML.

ΚΕΦΑΛΑΙΟ 2

ΓΛΩΣΣΑ ΜΟΝΤΕΛΟΠΟΙΗΣΗΣ UML

ΕΙΣΑΓΩΓΗ

Η UML αποτελεί μια γλώσσα απεικόνισης ή μοντελοποίησης ενός πληροφοριακού συστήματος βασισμένου σε αντικείμενα (αντικειμενοστρεφούς συστήματος). Όπως σε όλα τα σύνθετα έργα έτσι και στα έργα πληροφορικής η ανάγκη της μοντελοποίησης πριν την κατασκευή του συστήματος είναι επιτακτική. Η μοντελοποίηση ενός συστήματος παρέχει τη δυνατότητα της αφαίρεσης των ασήμαντων για αυτό λεπτομερειών και της εστίασης στις σημαντικές λεπτομέρειες του συστήματος που είναι απαραίτητο να κατανοηθούν πριν την κατασκευή του. Επίσης, δίνει τη δυνατότητα του πειραματισμού με διαφορετικές λύσεις ή προσεγγίσεις για το ίδιο πρόβλημα. Καθιστά εφικτή τη δυνατότητα ανάλυσης, σχεδιασμού, καταγραφής και παρακολούθησης της προόδου ενός έργου πληροφορικής. Τέλος προσφέρει μια κοινή γλώσσα για την επικοινωνία όσων εμπλέκονται στην κατασκευή του συστήματος. Χωρίς ένα μοντέλο δεν είναι δυνατόν να προσεγγίσει κανείς την πολυπλοκότητα των σύγχρονων πληροφοριακών συστημάτων.

Φυσικά η μοντελοποίηση των συστημάτων πληροφορικής δεν είναι κάτι νέο. Μοντέλα συστημάτων πληροφορικής κατασκευάζονται εδώ και δεκαετίες, με διαφορετικές όμως τεχνικές. Ο τρόπος με τον οποίο σχεδιάζουμε ένα σύστημα είναι άρρηκτα συνδεδεμένος με την τεχνολογία που θα χρησιμοποιήσουμε για να το αναπτύξουμε. Έτσι από τα διαγράμματα ροής προγράμματος περάσαμε με την εμφάνιση του δομημένου προγραμματισμού σε πιο προχωρημένες τεχνικές αναπαράστασης και σχεδίασης συστημάτων που περιλάμβαναν πιο σύνθετες διαγραμματικές τεχνικές για τον έλεγχο των πιο πολύπλοκων σύγχρονων

συστημάτων. Αυτές οι τεχνικές περιλάμβαναν λεξικά δεδομένων, διαγράμματα ροής δεδομένων, πίνακες αποφάσεων κ.λπ. Σήμερα τα συστήματα που αναπτύσσονται είναι σχεδόν στο σύνολό τους αντικειμενοστρεφή, και υπάρχουν καλοί λόγοι για αυτό:

- Το αντικειμενοστρεφές λογισμικό είναι ευκολότερο στην αρχική του σύλληψη, αφού τα αντικείμενα είναι οντότητες του υπαρκτού κόσμου του πεδίου προβλήματος
- Το αντικειμενοστρεφές λογισμικό είναι ευκολότερο στην εξέλιξή του, με τη χρήση διασυνδέσεων (interfaces).
- Η αντικειμενοστρεφής προσέγγιση επιτρέπει τη δημιουργία λογισμικού με βάση τα συστατικά (components).
- Σύγχρονες τεχνολογίες κατασκευής κατανεμημένων συστημάτων προσανατολισμένων στη σύνδεση επιχειρήσεων (B2B) και στη σύνδεση επιχειρηματικών εφαρμογών όπως οι υπηρεσίες του παγκόσμιου ιστού (Web Services) έχουν σαν τεχνολογικό υπόβαθρο αντικειμενοστρεφείς γλώσσες προγραμματισμού (Java,C#).

Συμπερασματικά, είναι αδύνατο να ασχοληθεί σήμερα κανείς με την ανάπτυξη λογισμικού χωρίς να γνωρίζει να αναπτύσσει αντικειμενοστρεφή συστήματα. Ως εκ τούτου είναι αναγκαία η γνώση της UML για την ανάπτυξη αυτών των συστημάτων, αφού η UML αποτελεί την πρότυπη γλώσσα μοντελοποίησης αντικειμενοστρεφών συστημάτων. Η UML αποτελεί πρότυπο του OMG(Object Management Group). Ο OMG είναι ένας διεθνής μη κερδοσκοπικός οργανισμός που παράγει και διαχειρίζεται πρότυπα για τη διαλειτουργικότητα των επιχειρηματικών εφαρμογών λογισμικού μεταξύ των οποίων και η UML. Ο OMG έχει μέλη σχεδόν όλες τις μεγάλες εταιρίες παραγωγής λογισμικού και εκατοντάδες μικρότερες. Η προτυποποίηση της UML από τον OMG, η γενικότερη αποδοχή της από τη βιομηχανία λογισμικού, καθώς και το γεγονός πως έχει περάσει αρκετός χρόνος από το 1997 που υιοθετήθηκε το πρώτο πρότυπο της UML από τον OMG, προσδίδουν στη UML το σημαντικό χαρακτηριστικό της διάρκειας στο χρόνο, κάτι που είναι πολύ σημαντικό για τη βιομηχανία λογισμικού που εξελίσσεται με ραγδαίους ρυθμούς (Γερογιάννης et al., 2006).

2.1 ΜΕΘΟΔΟΛΟΓΙΕΣ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ

Η UML είναι μια γλώσσα μοντελοποίησης και όχι μια μεθοδολογία ανάπτυξης έργων λογισμικού. Μια μεθοδολογία ανάπτυξης λογισμικού παρέχει μια συστηματική προσέγγιση στη διαδικασία ανάλυσης, σχεδίασης, κατασκευής και εξέλιξης ενός έργου πληροφορικής. Μια μεθοδολογία είναι ουσιαστικά μια σειρά σταδίων που περιγράφουν συγκεκριμένες εργασίες. Η UML είναι ουδέτερη σε σχέση με τις μεθοδολογίες, χωρίς να επιβάλλει κάποια συγκεκριμένη, και μπορεί να χρησιμοποιηθεί με διάφορους τρόπους. Υπάρχουν πολλές μεθοδολογίες ανάπτυξης λογισμικού, όπως οι Unified Process, eXtreme Programming, Catalysis,

Syntropy κ.ο.κ. Κάθε μία από αυτές θεωρείται πως είναι πιο κατάλληλη για κάποιο συγκεκριμένο τύπο λογισμικού.

Ίσως η πιο διαδεδομένη μεθοδολογία είναι η Ενοποιημένη Προσέγγιση (Unified Process-UP). Η UP αναπτύχθηκε αρχικά από την εταιρία Rational. Τα βασικά στάδια της UP είναι τα εξής:

- Η σύλληψη (inception) είναι η πρώτη φάση της ενοποιημένης προσέγγισης όπου παρουσιάζεται η αρχική ιδέα του συστήματος.
- Η λεπτομερής επεξεργασία (elaboration) είναι η δεύτερη φάση όπου περιγράφεται ο σκοπός του συστήματος καθώς και η υψηλού επιπέδου αρχιτεκτονική του. Σε αυτή τη φάση προσδιορίζονται οι απαιτήσεις του συστήματος.
- Η κατασκευή (construction) είναι η Τρίτη φάση όπου σχεδιάζεται και κατασκευάζεται το λογισμικό.
- Η μετάβαση (transition) είναι η τέταρτη φάση της διαδικασίας όπου το λογισμικό υπόκειται σε έλεγχο και τελικά παραδίδεται στους χρήστες. Η φάση της μετάβασης σηματοδοτεί την έναρξη της φάσης της συντήρησης λογισμικού και όχι το τέλος της διαδικασίας ανάπτυξης (Γερογιάννης et al., 2006) .

2.2 ΤΥΠΟΙ ΔΙΑΓΡΑΜΜΑΤΩΝ ΤΗΣ UML

Τα βασικά διαγράμματα που υπάρχουν στη UML είναι τα ακόλουθα:

- Διάγραμμα Περιπτώσεων Χρήσης (Use Case Diagram): Αποτυπώνει τις προδιαγραφές του υπό κατασκευή συστήματος.
- Διάγραμμα Κλάσεων (Class Diagram) : Αποτυπώνει τις κλάσεις του συστήματος καθώς και τις στατικές σχέσεις μεταξύ τους.
- Διαγράμματα Συμπεριφοράς (Behavior Diagrams): Αποτυπώνουν τη δυναμική συμπεριφορά του συστήματος. Στα διαγράμματα συμπεριφοράς ανήκουν τα εξής διαγράμματα:
 - Διάγραμμα καταστάσεων (Statechart Diagram): Περιγράφει τις καταστάσεις ενός αντικειμένου του συστήματος και τον τρόπο με τον οποίο το αντικείμενο αυτό αλλάζει κατάσταση ως ανάδραση σε συμβάντα.
 - Διάγραμμα δραστηριοτήτων (Activity Diagram): Παρέχει ένα τρόπο για την αποτύπωση των δραστηριοτήτων που λαμβάνουν χώρα στο σύστημα, συμπεριλαμβανομένων και των παράλληλων δραστηριοτήτων που μπορεί να συμβαίνουν σε αυτό.
 - Διάγραμμα αλληλεπίδρασης (Interaction Diagrams): Δύο ισοδύναμοι τύποι διαγραμμάτων που χρησιμοποιούνται για την αποτύπωση της αλληλεπίδρασης των αντικειμένων του συστήματος:

- Διάγραμμα ακολουθίας (Sequence Diagram): Δίνει έμφαση στη χρονική ακολουθία της αλληλεπίδρασης μεταξύ των αντικειμένων και απεικονίζει τις ανταλλαγές των μηνυμάτων που λαμβάνουν χώρα μεταξύ αυτών για την επίτευξη κάποιου συγκεκριμένου στόχου στο πλαίσιο μιας περίπτωσης χρήσης.
- Διάγραμμα συνεργασίας (Collaboration Diagram): Το διάγραμμα αυτό είναι ισοδύναμο με το διάγραμμα ακολουθίας με τη έννοια ότι αποτυπώνονται σε αυτό οι ίδιες πληροφορίες. Η διαφορά είναι ότι απεικονίζονται οι σύνδεσμοι μεταξύ των αντικειμένων που αλληλεπιδρούν και δεν είναι τόσο ξεκάθαρη η αποτύπωση της χρονικής σειράς των μηνυμάτων.
- Διαγράμματα υλοποίησης (Implementation Diagrams): Δύο τύποι διαγραμμάτων σχετικοί με την υλοποίηση του συστήματος:
 - Διάγραμμα συστατικών (Component Diagram): Δίνει τη δυνατότητα να εμφανίσουμε συστατικά με τις δημόσιες διασυνδέσεις που αυτά παρέχουν.
 - Διάγραμμα διάταξης (Deployment Diagram): Δείχνει τη διάταξη των συστατικών ενός συστήματος κατά το χρόνο εκτέλεσης και τους κόμβους στους οποίους είναι τοποθετημένα αυτά τα συστατικά. Είναι αρκετά σημαντικό για την περιγραφή σύνθετων καταμεμημένων συστημάτων που έχουν διάφορα υποσυστήματα διεσπαρμένα στο χώρο.

Για την περιγραφή ενός συστήματος χρησιμοποιούμε ένα σύνολο διαγραμμάτων διαφορετικού τύπου, το καθένα από τα οποία περιγράφει μια διαφορετική οπτική γωνία του συστήματος. Έτσι χρησιμοποιούμε τα διαγράμματα κλάσεων για την περιγραφή των στατικών σχέσεων μεταξύ των κλάσεων, τα διαγράμματα συμπεριφοράς (κατάστασης, ακολουθίας κ.τ.λ.) για την περιγραφή της δυναμικής συμπεριφοράς του συστήματος και τα διαγράμματα υλοποίησης (συστατικών, διάταξης) για την καταγραφή των λεπτομερειών υλοποίησης του συστήματος. Τα διαγράμματα περιπτώσεων χρήσης είναι μια ειδική περίπτωση διαγραμμάτων, με την έννοια πως δεν έχουν άμεση σχέση με αντικείμενα ή αντικειμενοστρεφή συστήματα, αλλά αποτελούν ένα μηχανισμό καταγραφής των προδιαγραφών του συστήματος (Γερογιάννης et al., 2006).

ΕΠΙΛΟΓΟΣ

Σκοπός αυτού του κεφαλαίου ήταν να περιγράψει την Ενοποιημένη Γλώσσα Μοντελοποίησης (UML) εστιάζοντας στην ανάγκη μοντελοποίησης των πληροφοριακών συστημάτων, στους λόγους που τα συστήματα που αναπτύσσονται σήμερα είναι στο σύνολό τους αντικειμενοστρεφή και η UML είναι σήμερα η δημοφιλέστερη γλώσσα μοντελοποίησης συστημάτων λογισμικού. Έπειτα αναλύεται η πιο διαδεδομένη μεθοδολογία ανάπτυξης λογισμικού, η UP

που χρησιμοποιεί τη UML, και τέλος περιγράφονται οι τύποι διαγραμμάτων της UML. Σύμφωνα με τις παραπάνω τεχνικές έγινε η ανάλυση και η σχεδίαση του συστήματος διαχείρισης βιογραφικών υποψηφίων μελών ΕΠ. Το επόμενο κεφάλαιο περιγράφει τις τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση του συστήματος.

ΚΕΦΑΛΑΙΟ 3

ΤΕΧΝΟΛΟΓΙΕΣ: JAVA & SQLite Administrator

ΕΙΣΑΓΩΓΗ

Η Java είναι μια αντικειμενοστρεφής γλώσσα προγραμματισμού που σχεδιάστηκε από την εταιρεία πληροφορικής *Sun Microsystems*.

Στις αρχές του 1991, η *Sun* αναζητούσε το κατάλληλο εργαλείο για να αποτελέσει την πλατφόρμα ανάπτυξης λογισμικού σε μικρο-συσσκευές (έξυπνες οικιακές συσκευές έως πολύπλοκα συστήματα παραγωγής γραφικών). Τα εργαλεία της εποχής ήταν γλώσσες όπως η C++ και η C. Μετά από διάφορους πειραματισμούς προέκυψε το συμπέρασμα ότι οι υπάρχουσες γλώσσες δεν μπορούσαν να καλύψουν τις ανάγκες τους. Ο "πατέρας" της Java, James Gosling, που εργαζόταν εκείνη την εποχή για την *Sun*, έκανε ήδη πειραματισμούς πάνω στη C++ και είχε παρουσιάσει κατά καιρούς κάποιες πειραματικές γλώσσες (C++ ++) ως πρότυπα για το νέο εργαλείο που αναζητούσαν στην *Sun*. Τελικά μετά από λίγο καιρό κατέληξαν με μια πρόταση για το επιτελείο της εταιρίας, η οποία ήταν η γλώσσα *Oak*. Το όνομά της το πήρε από το ομώνυμο δένδρο (βελανιδιά) το οποίο ο Gosling είχε έξω από το γραφείο του και έβλεπε κάθε μέρα.



Εικόνα 5 "Το λογότυπο της Java"

Η *Oak* ήταν μία γλώσσα που διατηρούσε μεγάλη συγγένεια με την C++. Παρόλα αυτά είχε πολύ πιο έντονο αντικειμενοστρεφή (*object oriented*) χαρακτήρα σε σχέση με την C++ και χαρακτηριζόταν για την απλότητα της. Σύντομα οι υπεύθυνοι ανάπτυξης της νέας γλώσσας ανακάλυψαν ότι το όνομα *Oak* ήταν ήδη κατοχυρωμένο οπότε κατά την διάρκεια μιας εκ των πολλών συναντήσεων σε κάποιο τοπικό καφέ αποφάσισαν να μετονομάσουν το νέο τους δημιούργημα σε Java που εκτός των άλλων ήταν το όνομα αγαπητού καφέ για τους δημιουργούς της (*Java* στην αγγλική γλώσσα είναι το φυτό που βγάζει τον καφέ). Η επίσημη εμφάνιση της *Java* αλλά και του *HotJava* (πλοηγός με υποστήριξη *Java*) στη βιομηχανία της πληροφορικής έγινε το Μάρτιο του 1995 όταν η *Sun* την ανακοίνωσε στο συνέδριο *Sun World 1995*. Ο πρώτος μεταγλωττιστής (*compiler*) της ήταν γραμμένος στη γλώσσα C από τον James Gosling. Το 1994, ο A. Van Hoff ξαναγράφει τον μεταγλωττιστή της γλώσσας σε *Java*, ενώ το Δεκέμβριο του 1995 πρώτες οι IBM, Borland, Mitsubishi Electronics, Sybase και Symantec ανακοινώνουν σχέδια να χρησιμοποιήσουν τη *Java* για την δημιουργία λογισμικού. Από εκεί και πέρα η *Java* ακολουθεί μία ανοδική πορεία και είναι πλέον μία από τις πιο δημοφιλείς γλώσσες στον χώρο της πληροφορικής. Στις 13 Νοεμβρίου του 2006 η *Java* έγινε πλέον μια γλώσσα ανοιχτού κώδικα (GPL) όσον αφορά το μεταγλωττιστή (*javac*) και το πακέτο ανάπτυξης (*JDK, Java Development Kit*).

Στις 27 Απριλίου 2010 η εταιρία λογισμικού Oracle Corporation ανακοίνωσε ότι μετά από πολύμηνες συζητήσεις ήρθε σε συμφωνία για την εξαγορά της Sun Microsystems και των τεχνολογιών (πνευματικά δικαιώματα/ πατέντες) που η δεύτερη είχε στην κατοχή της ή δημιουργήσει. Η συγκεκριμένη συμφωνία θεωρείται σημαντική για το μέλλον της *Java* και του γενικότερου οικοσυστήματος τεχνολογιών γύρω από αυτή μιας και ο έμμεσος έλεγχος της τεχνολογίας και η εξέλιξη της περνάει σε άλλα χέρια (*Java, n.d.*).

3.1 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΗΣ JAVA

Ένα από τα βασικά πλεονεκτήματα της *Java* έναντι των περισσότερων άλλων γλωσσών είναι η ανεξαρτησία του λειτουργικού συστήματος και πλατφόρμας. Τα προγράμματα που είναι γραμμένα σε *Java* τρέχουν ακριβώς το ίδιο σε Windows, Linux, Unix και Macintosh (σύντομα θα τρέχουν και σε

Playstation καθώς και σε άλλες κονσόλες παιχνιδιών) χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση (compiling) ή να αλλάξει ο πηγαίος κώδικας για κάθε διαφορετικό λειτουργικό σύστημα. Για να επιτευχθεί όμως αυτό χρειαζόταν κάποιος τρόπος έτσι ώστε τα προγράμματα γραμμένα σε Java να μπορούν να είναι «κατανοητά» από κάθε υπολογιστή ανεξάρτητα του είδους επεξεργαστή (Intel x86, IBM, Sun SPARC, Motorola) αλλά και λειτουργικού συστήματος (Windows, Unix, Linux, BSD, MacOS). Ο λόγος είναι ότι κάθε κεντρική μονάδα επεξεργασίας κατανοεί διαφορετικό κώδικα μηχανής. Ο συμβολικός (*assembly*) κώδικας που μεταφράζεται και εκτελείται σε Windows είναι διαφορετικός από αυτόν που μεταφράζεται και εκτελείται σε έναν υπολογιστή Macintosh. Η λύση δόθηκε με την ανάπτυξη της *Εικονικής Μηχανής (Virtual Machine ή VM ή EM στα ελληνικά)*.

3.1.1 Η ΕΙΚΟΝΙΚΗ ΜΗΧΑΝΗ ΤΗΣ JAVA

Αφού γραφεί κάποιο πρόγραμμα σε Java, στη συνέχεια μεταγλωττίζεται μέσω του μεταγλωττιστή `javac`, ο οποίος παράγει έναν αριθμό από αρχεία `.class` (κώδικας `byte` ή `bytecode`). Ο κώδικας `byte` είναι η μορφή που παίρνει ο πηγαίος κώδικας της Java όταν μεταγλωττιστεί. Όταν πρόκειται να εκτελεστεί η εφαρμογή σε ένα μηχάνημα, το Java Virtual Machine που πρέπει να είναι εγκατεστημένο σε αυτό θα αναλάβει να διαβάσει τα αρχεία `.class`. Στη συνέχεια τα μεταφράζει σε γλώσσα μηχανής που να υποστηρίζεται από το λειτουργικό σύστημα και τον επεξεργαστή, έτσι ώστε να εκτελεστεί (να σημειωθεί εδώ ότι αυτό συμβαίνει με την παραδοσιακή Εικονική Μηχανή (Virtual Machine)). Πιο σύγχρονες εφαρμογές της εικονικής Μηχανής μπορούν και μεταγλωττίζουν εκ των προτέρων τμήματα `bytecode` απευθείας σε κώδικα μηχανής (εγγενή κώδικα ή `native code`) με αποτέλεσμα να βελτιώνεται η ταχύτητα). Χωρίς αυτό δε θα ήταν δυνατή η εκτέλεση λογισμικού γραμμένου σε Java. Πρέπει να σημειωθεί ότι η JVM είναι λογισμικό που εξαρτάται από την πλατφόρμα, δηλαδή για κάθε είδος λειτουργικού συστήματος και αρχιτεκτονικής επεξεργαστή υπάρχει διαφορετική έκδοση του. Έτσι υπάρχουν διαφορετικές JVM για Windows, Linux, Unix, Macintosh, κινητά τηλέφωνα, παιχνιδιομηχανές κλπ.

Οτιδήποτε θέλει να κάνει ο προγραμματιστής (ή ο χρήστης) γίνεται μέσω της εικονικής μηχανής. Αυτό βοηθάει στο να υπάρχει μεγαλύτερη ασφάλεια στο σύστημα γιατί η εικονική μηχανή είναι υπεύθυνη για την επικοινωνία χρήστη - υπολογιστή. Ο προγραμματιστής δεν μπορεί να γράψει κώδικα ο οποίος θα έχει καταστροφικά αποτελέσματα για τον υπολογιστή γιατί η εικονική μηχανή θα τον ανιχνεύσει και δε θα επιτρέψει να εκτελεστεί. Από την άλλη μεριά ούτε ο χρήστης μπορεί να κατεβάσει «κακό» κώδικα από το δίκτυο και να τον εκτελέσει. Αυτό είναι

ιδιαίτερα χρήσιμο για μεγάλα καταναμημένα συστήματα όπου πολλοί χρήστες χρησιμοποιούν το ίδιο πρόγραμμα συγχρόνως.

3.1.2 Ο ΣΥΛΛΕΚΤΗΣ ΑΠΟΡΡΙΜΑΤΩΝ (GARBAGE COLLECTOR)

Ακόμα μία ιδέα που βρίσκεται πίσω από τη *Java* είναι η ύπαρξη του συλλέκτη απορριμμάτων (*Garbage Collector*). Συλλογή απορριμμάτων είναι μία κοινή ονομασία που χρησιμοποιείται στον τομέα της πληροφορικής για να δηλώσει την ελευθέρωση τμημάτων μνήμης από δεδομένα που δε χρειάζονται και δε χρησιμοποιούνται άλλο. Αυτή η απελευθέρωση μνήμης στη *Java* είναι αυτόματη και γίνεται μέσω του συλλέκτη απορριμμάτων. Υπεύθυνη για αυτό είναι και πάλι η εικονική μηχανή η οποία μόλις «καταλάβει» ότι ο σωρός (heap) της μνήμης (στη *Java* η συντριπτική πλειοψηφία των αντικειμένων αποθηκεύονται στο σωρό σε αντίθεση με τη *C++* όπου αποθηκεύονται κυρίως στη στοίβα) κοντεύει να γεμίσει ενεργοποιεί το συλλέκτη απορριμμάτων. Έτσι ο προγραμματιστής δε χρειάζεται να ανησυχεί για το πότε και αν θα ελευθερώσει ένα συγκεκριμένο τμήμα της μνήμης, ούτε και για σφάλματα δεικτών. Αυτό είναι ιδιαίτερα σημαντικό γιατί είναι κοινά τα σφάλματα προγραμμάτων που οφείλονται σε λανθασμένο χειρισμό της μνήμης.

3.1.3 ΕΠΙΔΟΣΕΙΣ

Παρόλο που η εικονική μηχανή προσφέρει όλα αυτά (και όχι μόνο) τα πλεονεκτήματα, η *Java* αρχικά ήταν πιο αργή σε σχέση με άλλες προγραμματιστικές γλώσσες υψηλού επιπέδου (high-level) όπως η *C* και η *C++*. Εμπειρικές μετρήσεις στο παρελθόν είχαν δείξει ότι η *C++* μπορούσε να είναι αρκετές φορές γρηγορότερη από την *Java*. Ωστόσο γίνονται προσπάθειες από τη *Sun* για τη βελτιστοποίηση της εικονικής μηχανής, ενώ υπάρχουν και άλλες υλοποιήσεις της εικονικής μηχανής από διάφορες εταιρίες (όπως της *IBM*), οι οποίες μπορεί σε κάποια σημεία να προσφέρουν καλύτερα και σε κάποια άλλα χειρότερα αποτελέσματα. Επιπλέον με την καθιέρωση των μεταγλωττιστών JIT (Just In Time), οι οποίοι μετατρέπουν τον κώδικα byte απευθείας σε γλώσσα μηχανής, η διαφορά ταχύτητας από τη *C++* έχει μικρύνει κατά πολύ.

Οι τελευταίες εκδόσεις του *javac* με τη χρήση της τεχνολογίας Hot Spot έχουν καταφέρει αξιόλογες επιδόσεις που πλησιάζουν ή και ξεπερνούν σε μερικές περιπτώσεις τον εγγενή κώδικα.

3.1.4 ΕΡΓΑΛΕΙΑ ΑΝΑΠΤΥΞΗΣ

Όλα τα εργαλεία που χρειάζεται κάποιος για να γράψει Java προγράμματα έρχονται δωρεάν, από το περιβάλλον ανάπτυξης μέχρι εργαλεία *build* όπως το Apache Ant και βιβλιοθήκες, ενώ υπάρχουν πολλές διαφορετικές υλοποιήσεις της *Εικονικής Μηχανής* και του *μεταγλωττιστή* (πχ the GNU Compiler for Java) της *Java*.

Πολλά εργαλεία και τεχνολογίες σε Java μπορούν να βρεθούν στο Apache Software Foundation αλλά και στο Jakarta Project (Java, n.d.).

3.2 ΟΛΟΚΛΗΡΩΜΕΝΟ ΠΕΡΙΒΑΛΛΟΝ ΑΝΑΠΤΥΞΗΣ (IDE)

Ένα ολοκληρωμένο περιβάλλον ανάπτυξης (*integrated development environment*, IDE) είναι μία σουίτα λογισμικού που βοηθάει στην ανάπτυξη προγραμμάτων υπολογιστή. Συνήθως ένα IDE περιλαμβάνει κάποιον επεξεργαστή πηγαίου κώδικα, έναν μεταγλωττιστή, εργαλεία αυτόματης παραγωγής κώδικα, αποσφαλματωτή, συνδότη, σύστημα ελέγχου εκδόσεων και εργαλεία κατασκευής γραφικών διασυνδέσεων χρήστη για τις υπό ανάπτυξη εφαρμογές.

3.2.1 Η ΠΛΑΤΦΟΡΜΑ NETBEANS

Το NetBeans αποτελεί ένα ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment – IDE) και μια πλατφόρμα για την ανάπτυξη εφαρμογών για το διαδίκτυο (χρησιμοποιώντας Java, JavaScript, PHP, Python, Ruby, Groovy, C, και C++). Η πλατφόρμα NetBeans επιτρέπει στις εφαρμογές να αναπτυχθούν από ένα σύνολο τμημάτων λογισμικού που λέγονται ενότητες (modules). Μια ενότητα είναι ένα αρχείο της Java που περιέχει τάξεις της Java οι οποίες γράφονται για να αλληλεπιδράσουν με το NetBeans Open API και ένα αρχείο manifest που το προσδιορίζει ως ενότητα. Οι εφαρμογές που στηρίζονται σε ενότητες μπορούν να επεκταθούν με την προσθήκη νέων ενοτήτων. Δεδομένου ότι οι ενότητες μπορούν να αναπτυχθούν ανεξάρτητα, οι εφαρμογές που βασίζονται στην πλατφόρμα NetBeans μπορούν να επεκταθούν από προγραμματιστές τρίτου μέρους.

Το NetBeans άρχισε το 1997 ως ένα φοιτητικό project υπό την καθοδήγηση της σχολής των Μαθηματικών και της Φυσικής του Charles University στην Πράγα. Ο Roman Stanek δημιούργησε αργότερα μια επιχείρηση γύρω από το πρόγραμμα και παρήγαγε τις εμπορικές εκδόσεις του NetBeans IDE έως ότου αγοράστηκε από τη Sun Microsystems το 1999. Η Sun έκανε το NetBeans ανοικτού λογισμικού τον Ιούνιο του επόμενου έτους. Η κοινότητα NetBeans από τότε συνεχίζει να

μεγαλώνει, χάρη σε άτομα και εταιρίες που συνεισφέρουν στην ανάπτυξη του προγράμματος και το χρησιμοποιούν.

Η πλατφόρμα NetBeans είναι ένα επαναχρησιμοποιήσιμο πλαίσιο για την ανάπτυξη εφαρμογών υπολογιστών γραφείου. Όταν μια εφαρμογή βασισμένη στην πλατφόρμα NetBeans τρέχει, εκτελείται η κύρια τάξη (Main) της πλατφόρμας. Οι διαθέσιμες ενότητες τοποθετούνται σε μία ιεραρχία μνήμης, και εκτελούνται οι διεργασίες αρχικοποίησης των ενοτήτων. Γενικά, ο κώδικας μιας ενότητας φορτώνεται στη μνήμη μόνο όταν απαιτείται.

Οι εφαρμογές μπορούν να εγκαταστήσουν τις ενότητες δυναμικά. Οποιαδήποτε εφαρμογή μπορεί να συμπεριλάβει την ενότητα Update Center για να επιτρέψει στους χρήστες της εφαρμογής να μεταφορτώσουν ψηφιακά-υπογεγραμμένες βελτιώσεις και τα νέα χαρακτηριστικά γνωρίσματα άμεσα στην τρέχουσα εφαρμογή. Η επανατοποθέτηση μιας βελτίωσης ή μιας νέας έκδοσης δεν αναγκάζει τους χρήστες να κατεβάσουν πάλι ολόκληρη την εφαρμογή. Η πλατφόρμα προσφέρει υπηρεσίες κοινές στις εφαρμογές γραφείου, που επιτρέπουν στους υπεύθυνους για την ανάπτυξη να εστιάσουν στη λογική, ειδικά για την εφαρμογή τους. Μεταξύ των χαρακτηριστικών γνωρισμάτων της πλατφόρμας είναι:

- Διαχείριση διασύνδεσης χρήστη (μενού και γραμμές εργαλείων)
- Διαχείριση ρυθμίσεων χρήστη
- Διαχείριση αποθήκευσης (αποθήκευση και φόρτωση κάθε είδους δεδομένων)
- Διαχείριση παραθύρων
- Wizard framework (υποστηρίζει βήμα προς βήμα διαλόγους)

3.2.2 NETBEANS IDE

Το NetBeans IDE είναι ένα ενοποιημένο ανοικτού λογισμικού περιβάλλον ανάπτυξης γραμμένο εξ ολοκλήρου σε Java που χρησιμοποιεί την πλατφόρμα NetBeans. Το NetBeans IDE υποστηρίζει την ανάπτυξη όλων των τύπων εφαρμογών της Java (Java SE, Ιστού, EJB και κινητές εφαρμογές). Μεταξύ άλλων τα χαρακτηριστικά γνωρίσματα είναι ένα σύστημα προγραμμάτων βασισμένο σε Ant, έλεγχος εκδόσεων και refactoring.

Όλες οι λειτουργίες του IDE παρέχονται από ενότητες (modules). Κάθε ενότητα παρέχει μια καλά καθορισμένη λειτουργία, όπως υποστήριξη για τη γλώσσα Java, επεξεργασία, υποστήριξη για το CVS σύστημα ελέγχου και SVN. Το NetBeans

περιέχει όλες τις ενότητες που χρειάζονται για την ανάπτυξη εφαρμογών Java σε μία μόνο μεταφόρτωση, επιτρέποντας στο χρήστη να αρχίσει άμεση εργασία. Οι ενότητες επιτρέπουν επίσης στο Netbeans να επεκταθεί. Νέα χαρακτηριστικά, όπως υποστήριξη για άλλες προγραμματιστικές γλώσσες, μπορούν να προστεθούν με την εγκατάσταση των κατάλληλων ενοτήτων. Για παράδειγμα οι Sun Studio, Sun Java Studio Enterprise, και Sun Java Studio Creator από τη Sun Microsystems, βασίζονται στο NetBeans IDE (NetBeans, n.d.).

3.3 JDBC

Η Συνδετικότητα Βάσης Δεδομένων JAVA (Java Database Connectivity-JDBC) είναι μία διεπαφή προγραμματισμού εφαρμογών (API) για την γλώσσα προγραμματισμού Java η οποία ορίζει πώς ένας χρήστης μπορεί να έχει πρόσβαση σε μια βάση δεδομένων. Παρέχει συναρτήσεις για εξαγωγή, πρόσθεση, ανανέωση ή διαγραφή δεδομένων σε μια βάση.

Η πλατφόρμα 2 της Java, Standard έκδοση 1.4 (**J2SE**) εμπεριέχει την Συνδετικότητα Βάσης Δεδομένων (JDBC) 3.0 API μαζί με την υλοποίηση αναφοράς της γέφυρας JDBC-to-ODBC, επιτρέποντας συνδέσεις σε ODBC προσβάσιμες πηγές δεδομένων στο περιβάλλον της JVM. Αυτή η γέφυρα είναι πρωτογενής κλειστός κώδικας (όχι Java), και μόνο κατάλληλος για πειραματισμό και για συνθήκες στις οποίες κανένας άλλος οδηγός δεν είναι διαθέσιμος (JDBC, n.d.).

3.3.1 ΛΕΙΤΟΥΡΓΙΑ

Η JDBC επιτρέπει πολλαπλές υλοποιήσεις να υπάρχουν και να χρησιμοποιούνται από την ίδια εφαρμογή. Το API παρέχει έναν μηχανισμό για δυναμική φόρτωση των σωστών πακέτων της Java και την ρύθμισή τους στο JDBC Driver Manager. Ο Driver Manager χρησιμοποιείται ως εργοστάσιο σύνδεσης για τη δημιουργία συνδέσεων JDBC.

Οι συνδέσεις JDBC υποστηρίζουν τη δημιουργία και την εκτέλεση δηλώσεων. Αυτές μπορούν να είναι δηλώσεις ενημέρωσης όπως της SQL CREATE, INSERT, UPDATE και DELETE, ή μπορούν να είναι δηλώσεις ερώτησης όπως SELECT. Επιπλέον, οι αποθηκευμένες διαδικασίες μπορούν να κληθούν μέσω μιας σύνδεσης JDBC. Η JDBC αναπαριστά τις δηλώσεις χρησιμοποιώντας μια από τις ακόλουθες κατηγορίες:

- Statement: Η δήλωση αποστέλλεται στο server κάθε φορά
- PreparedStatement: Η δήλωση εναποθηκεύεται και έπειτα η πορεία εκτέλεσης προκαθορίζεται στον κεντρικό server της βάσης επιτρέποντας της να εκτελείται πολλές φορές κατά τρόπο αποδοτικό.
- CallableStatement: Χρησιμοποιείται για την εκτέλεση αποθηκευμένων διαδικασιών στη βάση δεδομένων.

3.3.2 ΠΑΡΑΔΕΙΓΜΑΤΑ

Η μέθοδος `Class.forName(String)` χρησιμοποιείται για να φορτώσει τον JDBC Driver.

```
Class.forName( "com.somejdbcvender.TheirJdbcDriver" );
```

Όταν μια τάξη Driver φορτώνεται δημιουργεί μια οντότητα του εαυτού του και την εγγράφει στο DriverManager. Αυτό γίνεται συμπεριλαμβάνοντας τον αναγκαίο κώδικα. Όταν απαιτείται μια σύνδεση μία από τις μεθόδους `DriverManager.getConnection()` χρησιμοποιείται για να δημιουργήσει μια σύνδεση JDBC.

```
Connection conn = DriverManager.getConnection(
    "jdbc:somejdbcvender:other data needed by some
jdbc vendor",
    "myLogin",
    "myPassword" );
try {
    /* you use the connection here */
} finally {
    //It's important to close the connection when
you are done with it
    try { conn.close(); } catch (Throwable ignore) {
/* Propagate the original exception
instead of this one that you may want just logged */
    }
}
```

Το URL που χρησιμοποιείται εξαρτάται κυρίως από το συγκεκριμένο JDBC driver. Θα αρχίσει πάντα με το «jdbc:» πρωτόκολλο, αλλά το υπόλοιπο εξαρτάται από τον ιδιαίτερο προμηθευτή. Μόλις καθιερωθεί μια σύνδεση, μια δήλωση μπορεί να δημιουργηθεί (Java Database Connectivity).

```
Statement stmt = conn.createStatement();
try {
    stmt.executeUpdate( "INSERT INTO MyTable( name ) VALUES
( 'my name' ) " );
} finally {
    //It's important to close the statement when you are
done with it
    try { stmt.close(); } catch (Throwable ignore) { /*
Propagate the original exception
instead of this one that you may want just logged */ }
}
```

3.4 SQLite Administrator DATABASE

Το SQLite είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων της Java που μπορεί να ενσωματωθεί στα προγράμματα της Java και να χρησιμοποιηθεί για σε απευθείας σύνδεση συναλλαγές. Συγκεκριμένα είναι ένα ισχυρό εργαλείο που μας βοηθάει να δημιουργήσουμε εύκολα, να σχεδιάσουμε ή να διαχειριστούμε τα αρχεία βάσεων δεδομένων SQLite. Ο συντάκτης κώδικα SQL μας βοηθά να γράψουμε γρήγορα τις ερωτήσεις SQL με τα χαρακτηριστικά γνωρίσματα όπως η ολοκλήρωση και ο τονισμός κώδικα.

3.5 JAVA DB

Η Java DB είναι η διανομή της ανοικτής βάσης δεδομένων Apache Derby που υποστηρίζεται από τη Sun. Η ευκολία στη χρήση, η συμμόρφωση με τα πρότυπα, το πλήρες σύνολο χαρακτηριστικών γνωρισμάτων την κάνει την ιδανική βάση δεδομένων για τους προγραμματιστές της Java . Η Java DB γράφεται στη γλώσσα προγραμματισμού Java, παρέχοντας φορητότητα. Μπορεί να ενσωματωθεί στις εφαρμογές της Java, χωρίς να απαιτεί διαχείριση από τον υπεύθυνο για την ανάπτυξη ή το χρήστη. Μπορεί επίσης να χρησιμοποιηθεί σε μορφή πελάτη-εξυπηρετητή. Η Java DB είναι πλήρως συναλλαγματική και παρέχει μια τυποποιημένη διεπαφή καθώς επίσης και ένα JDBC 4.0 SQL συμβατό οδηγό (Java DB, n.d.).

3.5.1 ΠΑΡΑΔΕΙΓΜΑΤΑ

Δημιουργία Σύνδεσης στη Βάση

```
private static final String DIAXEIRISI_CV_DB_DIRECTORY = "DiaxeirisiCV";  
  
private static final String DIAXEIRISI_CV_DB_FILE = "DiaxeirisiCV.db";  
  
public void connectToDatabase()  
{  
    if ( checkFileIntegrityOfDatabase(DIAXEIRISI_CV_DB_FILE) )  
    {  
        System.out.println("Integrity of the database is verified");  
        initDB(DIAXEIRISI_CV_DB_DIRECTORY + "/" + DIAXEIRISI_CV_DB_FILE, "", "");  
    }  
    else  
    {  
        deleteDatabase();  
        System.out.println("Now creating the database");  
        createDatabaseDirectory();  
        initDB(DIAXEIRISI_CV_DB_DIRECTORY + "/" + DIAXEIRISI_CV_DB_FILE, "", "");  
        createDatabase();  
    }  
    if (managedToConnect)  
    {  
        System.out.println("The database successfully loaded");  
    }  
    else  
    {
```

Δημιουργία Πίνακα

```
private void createUsersTable()
{
    String query = null;
    try
    {
        if (conn != null)
        {
            Statement stmt = conn.createStatement();

            query = "CREATE TABLE " + USERS_TABLE_NAME + " ("
                + USERS_TABLE_COLUMN_NAMES[0] + " INTEGER PRIMARY
KEY AUTOINCREMENT,"
                + USERS_TABLE_COLUMN_NAMES[1] + " VARCHAR (20),"
                + USERS_TABLE_COLUMN_NAMES[2] + " VARCHAR (20),"
```

Εισαγωγή Γραμμής σε Πίνακα

```
public boolean insertIntoUsersTable(String username, String password, int type)
{
    boolean success = true;
    String query = null;
    try
    {
        if (conn != null)
        {
            Statement stmt = conn.createStatement();

            query = "INSERT INTO " + USERS_TABLE_NAME + " ("
```

E

```
public boolean getUserData(ArrayList<User> userList)
{
    boolean success = true;
    String query = null;
    try
    {
        if (conn != null)
        {
            Statement stmt = conn.createStatement();
            ResultSet rslt = null;

            query = "SELECT * FROM " + USERS_TABLE_NAME;
            rslt = stmt.executeQuery(query);
        }
    }
}
```

ΕΠΙΛΟΓΟΣ

Το κεφάλαιο αυτό είχε ως στόχο να περιγράψει τις κύριες τεχνολογίες με τις οποίες υλοποιήθηκε το σύστημα διαχείρισης βιογραφικών υποψηφίων μελών ΕΠ. Αρχικά αναφέρθηκε η ιστορία της Java μέχρι σήμερα και κάποια ιδιαίτερα χαρακτηριστικά της. Στη συνέχεια περιγράφεται το Java Applet και δίνεται ένα παράδειγμα υλοποίησης ενός απλού applet. Έπειτα περιγράφονται η πλατφόρμα Netbeans πάνω στην οποία υλοποιήθηκε η εφαρμογή, η συνδετικότητα βάσης δεδομένων JAVA (JDBC) και το σύστημα διαχείρισης βάσεων δεδομένων Derby (JavaDB) με το οποίο υλοποιήθηκε η βάση δεδομένων του συστήματος. Τέλος, παρατίθενται κάποια παραδείγματα σύνδεσης και ερωτημάτων στη βάση δεδομένων Derby από κώδικα Java μέσω JDBC. Το επόμενο κεφάλαιο περιγράφει τα βήματα της ανάλυσης του συστήματος, από την καταγραφή των περιπτώσεων χρήσης, μέχρι την δημιουργία του διαγράμματος κλάσεων του συστήματος.

ΚΕΦΑΛΑΙΟ 4

ΣΧΕΔΙΑΣΜΟΣ

ΕΙΣΑΓΩΓΗ

Το κεφάλαιο αυτό περιέχει την αντικειμενοστρεφή ανάλυση και σχεδίαση του συστήματος με τη γλώσσα μοντελοποίησης UML. Αρχικά δίνεται η λεπτομερής περιγραφή των περιπτώσεων χρήσης του συστήματος οι οποίες στη συνέχεια παρουσιάζονται διαγραμματικά, και παρατίθενται ο πίνακας χρηστών στόχων, ο κατάλογος εννοιών και ο κατάλογος σχέσεων-εννοιών. Έπειτα παρουσιάζονται το εννοιολογικό μοντέλο, τα διαγράμματα ακολουθίας και συνεργασίας του συστήματος, και τέλος το διάγραμμα κλάσεων και το διάγραμμα οντοτήτων-συσχετίσεων της βάσης.

4.1 ΠΕΡΙΠΤΩΣΕΙΣ ΧΡΗΣΕΙΣ

ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ ΠΧ1:ΕΙΣΟΔΟΣ ΣΤΟ ΣΥΣΤΗΜΑ

Κύριος Χρήστης: ΓΡΑΜΜΑΤΕΑΣ,ΥΠΟΨΗΦΙΟΙ,ΕΠΙΤΡΟΠΗ,ΔΙΑΧΕΙΡΙΣΤΗΣ

Εμπλεκόμενοι και Ενδιαφέροντα:

Γραμματέας: Εισάγεται στο σύστημα για να καταχωρίσει τα στοιχεία κάποιου υποψηφίου ώστε να εγγραφεί στο σύστημα.

Υποψήφιος: Εισάγεται στο σύστημα για να καταχωρίσει ή να ενημερώσει τα στοιχεία του βιογραφικού του.

Επιτροπή: Εισάγεται στο σύστημα για να εξετάσει την καταλληλότητα των υποψηφίων σύμφωνα με κάποια κριτήρια.

Διαχειριστής: Εισάγεται στο σύστημα για να επεξεργαστεί τους χρήστες του συστήματος όπως να προσθέσει και να διαγράψει κάποιο χρήστη,να προσθέσει διαθέσιμα μαθήματα και να καταχωρίσει τις επιλογές του email server.

ΒΑΣΙΚΗ ΡΟΗ:

1. Το σύστημα παρουσιάζει τη φόρμα εισαγωγής σε αυτό, με την οποία ζητάει από το χρήστη να δώσει το όνομα χρήστη (user name) και τον κωδικό πρόσβασης (password).

2. Ο χρήστης εισάγει το όνομα χρήστη και τον κωδικό πρόσβασης και πατάει το κουμπί «Είσοδος».
3. Το σύστημα ανατρέχει στη βάση δεδομένων χρηστών και ταυτοποιεί τον χρήστη.
4. Ο χρήστης εισέρχεται στο σύστημα αποκτώντας πρόσβαση στη λειτουργικότητα του ρόλου που έχει ο χρήστης (Υποψήφιος, Γραμματέας, Επιτροπή ή Διαχειριστής).

ΕΝΑΛΛΑΚΤΙΚΗ ΡΟΗ 1:

1. Το σύστημα ανατρέχει στη βάση δεδομένων και δεν εντοπίζει το χρήστη.
2. Το σύστημα εμφανίζει μήνυμα και επιλογή ΟΚ.
3. Ο χρήστης επιλέγει ΟΚ.
4. Η οθόνη του μηνύματος κλείνει.
5. Η περίπτωση χρήσης συνεχίζει από το βήμα 1 της βασικής ροής.

ΜΗ ΛΕΙΤΟΥΡΓΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ

Δεν υπάρχουν.

Κατάσταση Εισόδου:

Δεν υπάρχει.

Κατάσταση Εξόδου:

1. Ο χρήστης ταυτοποιήθηκε επιτυχημένα από το σύστημα και ξεκίνησε να χρησιμοποιεί το σύστημα.
2. Το σύστημα απέτυχε να ταυτοποιήσει το χρήστη και εμφανίζει την αρχική οθόνη εισόδου.

ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ ΠΧ2: ΔΗΜΙΟΥΡΓΙΑ ΚΩΔΙΚΩΝ ΥΠΟΨΗΦΙΩΝ ΕΕΠ

Κύριος Χρήστης: Γραμματέας

Εμπλεκόμενοι και Ενδιαφέροντα:

Γραμματέας: Θέλει να γίνεται σωστά η εγγραφή του υποψηφίου και να αποστέλλεται άμεσα στο e-mail του το όνομα χρήστη και ο κωδικός του.

Υποψήφιος: Θέλει να γίνεται γρήγορα η εγγραφή του ώστε να μπορεί να καταχωρίσει στο σύστημα το βιογραφικό του.

Επιτροπή: Θέλει να γίνεται σωστά η εγγραφή των υποψηφίων ώστε να καταχωρούν άμεσα τα στοιχεία τους στο σύστημα και να γίνεται η εξέτασή τους.

Προϋποθέσεις: Ο υποψήφιος έχει κάνει αίτηση για εγγραφή στην γραμματεία.

ΒΑΣΙΚΗ ΡΟΗ:

1. Το σύστημα εμφανίζει την οθόνη καταχώρησης των στοιχείων του υποψηφίου όπως όνομα, επίθετο, username και e-mail.
2. Ο χρήστης εισάγει τα πιο πάνω στοιχεία του υποψηφίου.
3. Το σύστημα στέλνει στο e-mail του υποψηφίου ένα μήνυμα με το όνομα χρήστη και τον κωδικό του.
4. Το σύστημα εμφανίζει μήνυμα επιτυχούς αποστολής.
5. Ο χρήστης επιλέγει OK.
6. Η οθόνη του μηνύματος κλείνει.

ΕΝΑΛΛΑΚΤΙΚΗ ΡΟΗ 1:

1. Ο χρήστης επιλέγει «Ακύρωση».
2. Το σύστημα επιστρέφει στο βασικό μενού.

ΕΝΑΛΛΑΚΤΙΚΗ ΡΟΗ 2:

1. Το σύστημα αναγνώρισε ότι ο χρήστης έχει εισάγει λανθασμένη ή άκυρη πληροφορία.
2. Το σύστημα παρουσιάζει στο χρήστη το σχετικό μήνυμα λάθους μαζί με τη λανθασμένη πληροφορία.
3. Το σύστημα προτρέπει το χρήστη να επανεισάγει την πληροφορία.
4. Ο χρήστης εισάγει εκ νέου την πληροφορία.
5. Το σύστημα ελέγχει ξανά αν η πληροφορία είναι σωστή.
6. Αν η πληροφορία είναι σωστή το σύστημα συνεχίζει με το βήμα 4 της κανονικής ροής.
7. Αν η πληροφορία που δόθηκε δεν είναι σωστή το σύστημα επανέρχεται στο βήμα 2 της εναλλακτικής ροής 2.

Λανθασμένη πληροφορία μπορεί να είναι:

- Σε κάποιο πεδίο δεν έχει εισαχθεί πληροφορία.
- Ο τρόπος γραφής του e-mail είναι λανθασμένος.
- Ο τρόπος γραφής των στοιχείων του βιογραφικού δεν είναι σωστός.

ΜΗ ΛΕΙΤΟΥΡΓΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ

1. Ο χρόνος αποστολής των στοιχείων λογαριασμού του υποψηφίου στο e-mail του δεν πρέπει να ξεπερνάει τα 30 δευτερόλεπτα.

Κατάσταση Εισόδου: ο χρήστης έχει συνδεθεί στο σύστημα και έχει επιλέξει την προβολή της εγγραφής χρήστη.

Κατάσταση Εξόδου: Το σύστημα έχει αποστείλει τα στοιχεία λογαριασμού στο e-mail του υποψηφίου που έχει καταχωρηθεί.

Η αποστολή ακυρώθηκε είτε επειδή ο χρήστης απέτυχε να εισάγει τη σωστή πληροφορία, είτε επειδή ακύρωσε την αποστολή.

ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ ΠΧ3: ΚΑΤΑΧΩΡΗΣΗ ΣΤΟΙΧΕΙΩΝ ΥΠΟΨΗΦΙΟΥ

Κύριος χρήστης: Υποψήφιος

Εμπλεκόμενοι και Ενδιαφέροντα:

Υποψήφιος: Θέλει να μπορεί να καταχωρήσει όλα τα στοιχεία του βιογραφικού του με τέτοιο τρόπο ώστε να προβάλλει όλα τα προσόντα του (σπουδές, προϋπηρεσία, ερευνητική δραστηριότητα κ.τ.λ.).

Επιτροπή: Θέλει να καταχωρούνται από τους υποψηφίους όλα τα απαραίτητα στοιχεία που χρειάζονται για την εξέταση της υποψηφιότητάς τους σε σύγκριση με άλλους υποψηφίους.

Προϋποθέσεις: Ο χρήστης έχει ταυτοποιηθεί από το σύστημα.

ΒΑΣΙΚΗ ΡΟΗ:

1. Ο χρήστης επιλέγει τα μενού για να καταχωρίσει τα ανάλογα στοιχεία, όπως εκπαίδευση, δημοσιεύσεις, προϋπηρεσίας και ίδρυση εταιρίας.
2. Το σύστημα εμφανίζει την αντίστοιχη οθόνη εισαγωγής στοιχείων του υποψηφίου.
3. Ο χρήστης εισάγει τα στοιχεία: Προσωπικά Στοιχεία (όνομα, επώνυμο, διεύθυνση, τηλέφωνο), Πτυχία, Ξένες γλώσσες, Μαθήματα που μπορεί να διδάξει, Πίνακα με ώρες που δεν μπορεί να διδάξει, Εταιρία που έχει ιδρύσει (όνομα, αντικείμενο, χρονολογία), Προϋπηρεσία (επαγγελματική, διδακτική, ερευνητική) και Δημοσιεύσεις.
4. Ο χρήστης επιλέγει «Αποθήκευση».
5. Το σύστημα εμφανίζει μήνυμα επιτυχούς καταχώρησης.
6. Το σύστημα εμφανίζει το βασικό μενού.

ΕΝΑΛΛΑΚΤΙΚΗ ΡΟΗ 1:

1. Ο χρήστης επιλέγει «Ακύρωση».
2. Το σύστημα διακόπτει την καταχώρηση και επιστρέφει στο βασικό μενού.

ΕΝΑΛΛΑΚΤΙΚΗ ΡΟΗ 2:

1. Το σύστημα αναγνώρισε ότι ο χρήστης έχει εισάγει λανθασμένη ή άκυρη πληροφορία.
2. Το σύστημα παρουσιάζει στο χρήστη το σχετικό μήνυμα λάθους μαζί με τη λανθασμένη πληροφορία.
3. Το σύστημα προτρέπει το χρήστη να επανεισάγει την πληροφορία.
4. Ο χρήστης εισάγει εκ νέου την πληροφορία.
5. Το σύστημα ελέγχει ξανά αν η πληροφορία είναι σωστή.
6. Αν η πληροφορία είναι σωστή το σύστημα συνεχίζει με το βήμα 5 της κανονικής ροής.
7. Αν η πληροφορία που δόθηκε δεν είναι σωστή το σύστημα επανέρχεται στο βήμα 2 της εναλλακτικής ροής 2.

Λανθασμένη πληροφορία μπορεί να είναι:

- Σε κάποιο πεδίο δεν έχει εισαχθεί πληροφορία.
- Ο τρόπος γραφής του e-mail είναι λανθασμένος.
- Ο τρόπος γραφής των στοιχείων του βιογραφικού δεν είναι σωστός.

ΜΗ ΛΕΙΤΟΥΡΓΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ

Δεν υπάρχουν.

Κατάσταση Εισόδου:

Ο χρήστης έχει συνδεθεί στο σύστημα με επιτυχία.

Κατάσταση Εξόδου:

1. Η καταχώρηση των στοιχείων του υποψηφίου πραγματοποιήθηκε επιτυχώς.
2. Η καταχώρηση δεν πραγματοποιήθηκε επειδή ο χρήστης ακύρωσε τη διαδικασία ή απέτυχε να εισάγει τη σωστή πληροφορία.

ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ ΠΧ4: ΕΝΗΜΕΡΩΣΗ ΣΤΟΙΧΕΙΩΝ ΥΠΟΨΗΦΙΟΥ

Κύριος Χρήστης: Υποψήφιος

Εμπλεκόμενοι και Ενδιαφέροντα:

Υποψήφιος: Ο υποψήφιος θέλει να έχει τη δυνατότητα να ενημερώνει το βιογραφικό του όταν προκύπτουν νέα στοιχεία όπως δημοσιεύσεις, προϋπηρεσία, δημιουργία εταιρίας κ.τ.λ.

Επιτροπή: Θέλει να υπάρχει η δυνατότητα ενημέρωσης ή αλλαγής των στοιχείων των υποψηφίων ώστε να γίνεται σωστά η αξιολόγησή τους.

Προϋποθέσεις: Ο χρήστης έχει ταυτοποιηθεί από το σύστημα.

ΒΑΣΙΚΗ ΡΟΗ:

1. Ο χρήστης επιλέγει τα μενού με τα ήδη καταχωρημένα στοιχεία του και στη συνέχεια επιλέγει το μενού επεξεργασία για να ενημερώσει τα στοιχεία του.
2. Το σύστημα εμφανίζει παράθυρο με τα στοιχεία του χρήστη.
3. Ο χρήστης αλλάζει τα πεδία που επιθυμεί και πατάει «Αποθήκευση».
4. Το σύστημα παρουσιάζει τα νέα στοιχεία που έχουν εισαχθεί και ζητάει από το χρήστη να επιβεβαιώσει ότι είναι σωστά.
5. Ο χρήστης επιβεβαιώνει ότι ο λογαριασμός του πρέπει να τροποποιηθεί σύμφωνα με τα στοιχεία που του παρουσιάζονται από το σύστημα.
6. Το σύστημα ενημερώνει τη βάση με τα νέα στοιχεία για το συγκεκριμένο χρήστη και εμφανίζει μήνυμα επιτυχούς αποθήκευσης.

ΕΝΑΛΛΑΚΤΙΚΗ ΡΟΗ 1:

1. Ο χρήστης επιλέγει «Ακύρωση».
2. Το σύστημα διακόπτει την ενημέρωση και δεν πραγματοποιείται καμία αλλαγή στα δεδομένα.
3. Το σύστημα επιστρέφει στο βασικό μενού.

ΕΝΑΛΛΑΚΤΙΚΗ ΡΟΗ 2:

1. Το σύστημα αναγνώρισε ότι ο χρήστης έχει εισάγει λανθασμένη ή άκυρη πληροφορία.
2. Το σύστημα παρουσιάζει στο χρήστη το σχετικό μήνυμα λάθους μαζί με τη λανθασμένη πληροφορία.
3. Το σύστημα προτρέπει το χρήστη να επανεισάγει την πληροφορία.
4. Ο χρήστης εισάγει εκ νέου την πληροφορία.
5. Το σύστημα ελέγχει ξανά αν η πληροφορία είναι σωστή.
6. Αν η πληροφορία είναι σωστή το σύστημα συνεχίζει με το βήμα 4 της κανονικής ροής.
7. Αν η πληροφορία που δόθηκε δεν είναι σωστή το σύστημα επανέρχεται στο βήμα 2 της εναλλακτικής ροής 2.

Λανθασμένη πληροφορία μπορεί να είναι:

- Σε κάποιο πεδίο δεν έχει εισαχθεί πληροφορία.
- Ο τρόπος γραφής του e-mail είναι λανθασμένος.
- Ο τρόπος γραφής των στοιχείων του βιογραφικού δεν είναι σωστός.

ΜΗ ΛΕΙΤΟΥΡΓΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ

1. Η ενημέρωση των δεδομένων της βάσης δε θα πρέπει να ξεπερνάει τα 5 δευτερόλεπτα.

Κατάσταση εισόδου: Ο χρήστης έχει συνδεθεί στο σύστημα με επιτυχία.

Κατάσταση εξόδου:

1. Ο λογαριασμός του υποψηφίου τροποποιήθηκε επιτυχημένα και ο χρήστης συνεχίζει να χρησιμοποιεί το σύστημα.
2. Ο λογαριασμός δεν τροποποιήθηκε είτε επειδή ο χρήστης απέτυχε να εισάγει τη σωστή πληροφορία είτε επειδή ακύρωσε τη διαδικασία τροποποίησης του λογαριασμού.

ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ ΠΧ5:ΔΙΑΓΡΑΦΗ ΥΠΟΨΗΦΙΩΝ

Κύριος Χρήστης: Επιτροπή

Εμπλεκόμενοι και ενδιαφέροντα:

Επιτροπή: Θέλει να έχει τον πλήρη έλεγχο των δεδομένων της βάσης.

Προϋποθέσεις: Ο χρήστης έχει ταυτοποιηθεί από το σύστημα.

ΒΑΣΙΚΗ ΡΟΗ:

1. Ο χρήστης επιλέγει από το μενού «Εμφάνιση Όλων».
2. Ο χρήστης μπορεί να επιλέξει αναζήτηση υποψηφίων με βάση κάποιων κριτηρίων.
3. Το σύστημα εμφανίζει τα στοιχεία του υποψηφίου και την επιλογή διαγραφής.
4. Ο χρήστης επιλέγει «Διαγραφή».
5. Ο χρήστης επιλέγει ΟΚ και τα στοιχεία του υποψηφίου διαγράφονται από τη βάση.

ΕΝΑΛΛΑΚΤΙΚΗ ΡΟΗ 1:

1. Το σύστημα δεν εντόπισε τον υποψήφιο με το συγκεκριμένο ονοματεπώνυμο.
2. Το σύστημα εμφανίζει το σχετικό μήνυμα στο χρήστη.
3. Το σύστημα προτρέπει το χρήστη να επανεισάγει τα στοιχεία αναζήτησης.

4. Ο χρήστης εισάγει εκ νέου την πληροφορία που ζητάει το σύστημα.
5. Το σύστημα ελέγχει ξανά αν η πληροφορία είναι σωστή.
6. Αν η πληροφορία είναι σωστή το σύστημα συνεχίζει από το βήμα 3 της βασικής ροής.
7. Αν η πληροφορία δεν είναι σωστή το σύστημα επανέρχεται στο Βήμα 2 της εναλλακτικής ροής 1.

ΕΝΑΛΛΑΚΤΙΚΗ ΡΟΗ 2:

1. Από το βήμα 4 της βασικής ροής: Ο χρήστης επιλέγει «Ακύρωση».
2. Το σύστημα επιστρέφει στο βασικό μενού.

Λανθασμένη πληροφορία μπορεί να είναι:

- Σε κάποιο πεδίο δεν έχει εισαχθεί πληροφορία.
- Ο τρόπος γραφής του e-mail είναι λανθασμένος.
- Ο τρόπος γραφής των στοιχείων του βιογραφικού δεν είναι σωστός.

ΜΗ ΛΕΙΤΟΥΡΓΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ

Δεν υπάρχουν.

Κατάσταση εισόδου: Ο χρήστης έχει συνδεθεί στο σύστημα επιτυχία και έχουν ανακτηθεί από τη βάση τα στοιχεία του υποψηφίου που έχει επιλέξει.

Κατάσταση εξόδου: Η βάση δεδομένων ενημερώνεται με τα νέα στοιχεία.

ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ ΠΧ6: ΠΡΟΒΟΛΗ ΣΤΟΙΧΕΙΩΝ ΥΠΟΨΗΦΙΩΝ

Κύριος Χρήστης: Επιτροπή

Εμπλεκόμενοι και ενδιαφέροντα:

Υποψήφιος: Ο υποψήφιος θέλει να εμφανίζονται τα προσόντα του στις αναζητήσεις της επιτροπής ώστε να αυξάνονται οι πιθανότητές του να προσληφθεί.

Επιτροπή: Η επιτροπή θέλει να βλέπει τα προσόντα όλων των υποψηφίων που είναι σύμφωνα με τα κριτήρια της αναζήτησης.

Προϋποθέσεις: Ο χρήστης έχει ταυτοποιηθεί από το σύστημα.

ΒΑΣΙΚΗ ΡΟΗ:

1. Ο χρήστης επιλέγει από το μενού «Εμφάνιση Όλων».
2. Ο χρήστης μπορεί να επιλέξει αναζήτηση υποψηφίων με βάση κάποιων κριτηρίων.
3. Το σύστημα εμφανίζει στην οθόνη τα αποτελέσματα της αναζήτησης με βάση τα κριτήρια που έδωσε ο χρήστης.
4. Ο χρήστης επιλέγει «ΟΚ».
5. Το σύστημα επιστρέφει στο βασικό μενού επιλογών.

ΕΝΑΛΛΑΚΤΙΚΗ ΡΟΗ 1:

1. Ο χρήστης επιλέγει «Ακύρωση».
2. Το σύστημα επιστρέφει στο βασικό μενού επιλογών.

ΕΝΑΛΛΑΚΤΙΚΗ ΡΟΗ 2:

1. Το σύστημα αναγνώρισε ότι ο χρήστης έχει εισάγει λανθασμένη ή άκυρη πληροφορία.
2. Το σύστημα παρουσιάζει στο χρήστη το σχετικό μήνυμα λάθους μαζί με τη λανθασμένη πληροφορία.
3. Το σύστημα προτρέπει το χρήστη να επανεισάγει την πληροφορία.
4. Ο χρήστης εισάγει εκ νέου την πληροφορία.
5. Το σύστημα ελέγχει ξανά αν η πληροφορία είναι σωστή.
6. Αν η πληροφορία είναι σωστή το σύστημα συνεχίζει με το βήμα 3 της κανονικής ροής.
7. Αν η πληροφορία που δόθηκε δεν είναι σωστή το σύστημα επανέρχεται στο βήμα 2 της εναλλακτικής ροής 2.

Λανθασμένη πληροφορία μπορεί να είναι:

- Σε κάποιο πεδίο δεν έχει εισαχθεί πληροφορία.
- Ο τρόπος γραφής του e-mail είναι λανθασμένος.
- Ο τρόπος γραφής των στοιχείων του βιογραφικού δεν είναι σωστός.

ΜΗ ΛΕΙΤΟΥΡΓΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ

1. Η ανάκτηση των στοιχείων από τη βάση θα πρέπει να γίνεται μέσα σε 5 δευτερόλεπτα.

Κατάσταση Εισόδου: Ο χρήστης έχει συνδεθεί στο σύστημα με επιτυχία.

Κατάσταση Εξόδου:

1. Εμφανίζονται στην οθόνη τα βιογραφικά σύμφωνα με τα κριτήρια που έθεσε ο χρήστης.
2. Εμφανίζεται κενή οθόνη αν δεν υπάρχουν στοιχεία σύμφωνα με τα κριτήρια.
3. Η αναζήτηση δεν πραγματοποιήθηκε επειδή ο χρήστης απέτυχε να εισάγει τη σωστή πληροφορία είτε ακύρωσε την αναζήτηση.

ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ ΠΧ7: ΔΙΑΧΕΙΡΙΣΗ ΧΡΗΣΤΩΝ

Κύριος Χρήστης: Διαχειριστής

Εμπλεκόμενοι και ενδιαφέροντα:

Διαχειριστής: Θέλει να διαχειριστεί, να εισάγει, να διαγράψει και να τροποποιήσει τα στοιχεία των χρηστών του συστήματος με ευκολία, ταχύτητα και ασφάλεια.

Γραμματέας: Μόνο μετά την εισαγωγή των στοιχείων του στο σύστημα από το Διαχειριστή θα έχει το δικαίωμα να χρησιμοποιήσει το σύστημα με τα δικαιώματα της ιδιότητάς του.

Επιτροπή: Μόνο μετά την εισαγωγή των στοιχείων του στο σύστημα από το Διαχειριστή θα μπορεί να χρησιμοποιήσει το σύστημα με τα αυξημένα δικαιώματα που προκύπτουν από την ιδιότητά του ως μέλος επιτροπής.

Προϋποθέσεις: Ο Διαχειριστής θα πρέπει να αναγνωριστεί και αυθεντικοποιηθεί.

ΒΑΣΙΚΗ ΡΟΗ 1:

1. Ο Διαχειριστής δημιουργεί μια νέα εγγραφή στη βάση δεδομένων χρηστών στον πίνακα Users.
2. Εισάγει τα στοιχεία του χρήστη όπως username, e-mail και διαλέγει το τύπο χρήστη.
3. Τα δεδομένα αποθηκεύονται στη βάση.

ΕΝΑΛΛΑΚΤΙΚΗ ΡΟΗ 1:

1. Ο χρήστης έχει εισάγει λανθασμένη ή άκυρη πληροφορία σε κάποιο πεδίο του πίνακα.
2. Το σύστημα εμφανίζει στο διαχειριστή το κατάλληλο μήνυμα λάθους.
3. Ο διαχειριστής εισάγει εκ νέου την πληροφορία.
4. Αν η πληροφορία είναι σωστή το σύστημα συνεχίζει με το βήμα 3 της βασικής ροής.

5. Αν η πληροφορία που δόθηκε δεν είναι σωστή το σύστημα επανέρχεται στο βήμα 2 της εναλλακτικής ροής 2.

ΜΗ ΛΕΙΤΟΥΡΓΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ

Δε υπάρχουν.

Κατάσταση Εισόδου: Δεν υπάρχει.

Κατάσταση Εξόδου:

1. Ένας νέος χρήστης με την ιδιότητα Υποψήφιος, Γραμματέας ή Επιτροπή προστίθεται στο σύστημα.

ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ ΠΧ8: ΔΙΑΧΕΙΡΙΣΗ ΒΑΣΗΣ ΒΙΟΓΡΑΦΙΚΩΝ

Κύριος Χρήστης: Διαχειριστής

Εμπλεκόμενοι και ενδιαφέροντα:

Διαχειριστής: Θέλει να διαχειρίζεται και να επεξεργάζεται τα δεδομένα της βάσης που αφορούν τα βιογραφικά των χρηστών.

Υποψήφιος: Τα δεδομένα του βιογραφικού του θα διατηρούνται στη βάση από τον διαχειριστή.

Επιτροπή: Θα ενημερώνει το διαχειριστή για τυχόν αλλαγές που επιθυμεί να γίνουν στη βάση, όπως η αλλαγή κάποιων πινάκων, η δημιουργία νέων κ.τ.λ.

Προϋποθέσεις: Ο Διαχειριστής θα πρέπει να αναγνωριστεί και αυθεντικοποιηθεί.

ΒΑΣΙΚΗ ΡΟΗ 1:

1. Ο Διαχειριστής θέλει να προσθέσει ένα νέο πίνακα στη βάση.
2. Δημιουργεί τον πίνακα και ορίζει τα πεδία του.
3. Ενημερώνει το σύστημα ώστε να ζητά να συμπληρωθούν τα νέα πεδία κατά την καταχώρηση των στοιχείων του υποψηφίου.
4. Ο νέος πίνακας αποθηκεύεται στη βάση.

ΕΝΑΛΛΑΚΤΙΚΗ ΡΟΗ 1:

1. Ο Διαχειριστής θέλει να προσθέσει ένα νέο πίνακα στη βάση.
2. Δημιουργεί τον πίνακα και ορίζει τα πεδία του.

3. Εισάγει τις κατάλληλες τιμές στα πεδία του πίνακα.
4. Η βάση αποθηκεύει τα νέα δεδομένα.

ΒΑΣΙΚΗ ΡΟΗ 2:

1. Ο διαχειριστής θέλει να διαγράψει κάποιον πίνακα από τη βάση.
2. Επιλέγει να διαγραφεί ο πίνακας.
3. Η βάση ενημερώνεται με τα νέα δεδομένα.

ΒΑΣΙΚΗ ΡΟΗ 3:

1. Βελτιστοποίηση Βάσης Δεδομένων του συστήματος. Θα γίνεται σε μέρα και ώρα που καθορίζει ο Διαχειριστής. Στην πραγματικότητα πρόκειται για μια επαναδιαμόρφωση των διαφορών δέντρων που διατρέχουν τη βάση έτσι ώστε οι αναζητήσεις να γίνονται πιο γρήγορα.
2. Χειροκίνητη συντήρηση της βάσης. Αυτή γίνεται από τον Διαχειριστή και αφορά όλους τους πίνακες της βάσης δεδομένων.

ΜΗ ΛΕΙΤΟΥΡΓΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ

Δεν υπάρχουν.

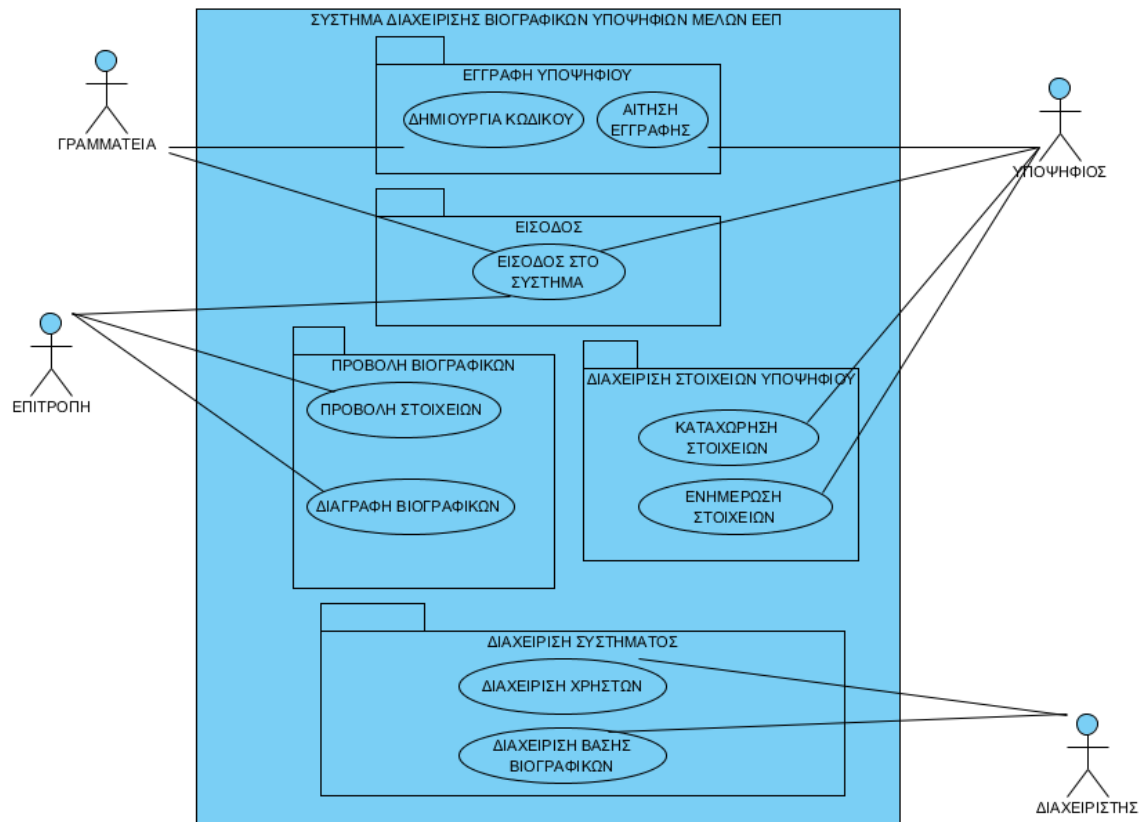
Κατάσταση Εισόδου: Δεν υπάρχει.

Κατάσταση Εξόδου:

1. Η βάση ενημερώνεται με τα νέα δεδομένα.
2. Η βάση βελτιστοποιείται.

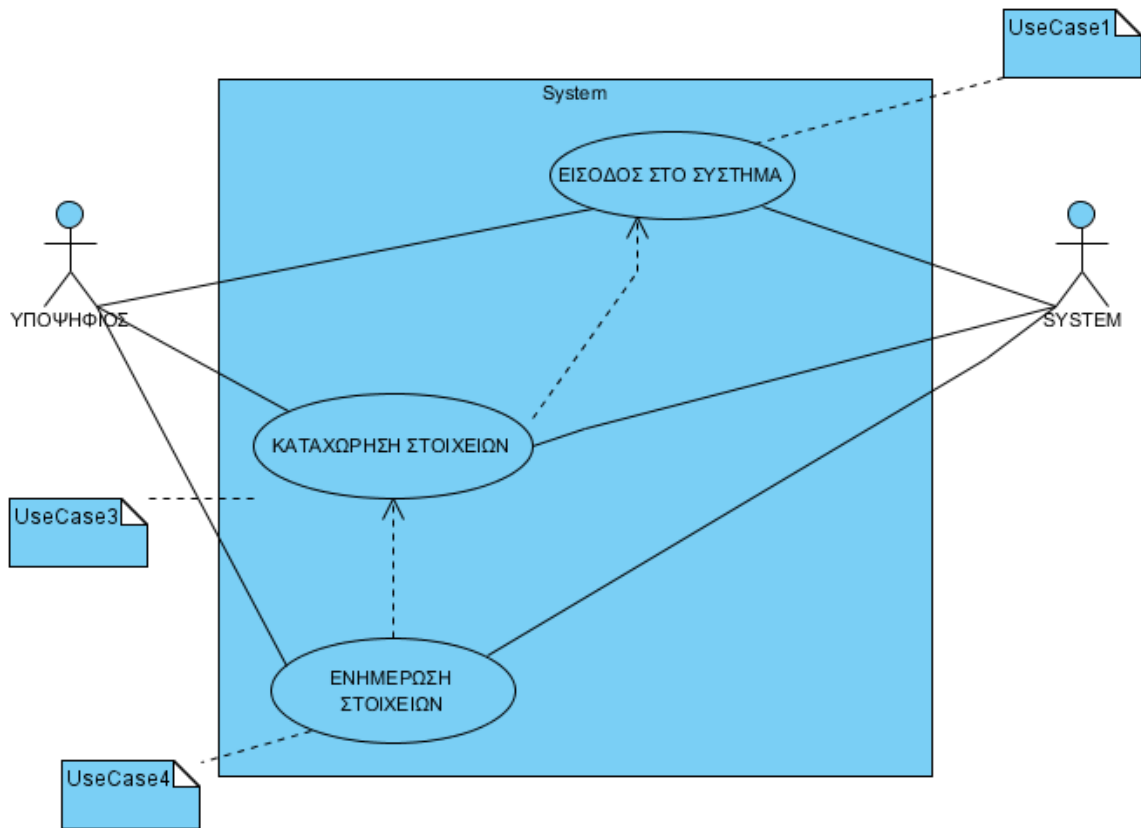
4.2 ΔΙΑΓΡΑΜΜΑΤΑ ΠΕΡΙΠΤΩΣΕΩΝ ΧΡΗΣΗΣ

4.2.1 ΔΙΑΓΡΑΜΜΑ ΠΕΡΙΠΤΩΣΕΩΝ ΧΡΗΣΗΣ ΣΥΣΤΗΜΑΤΟΣ



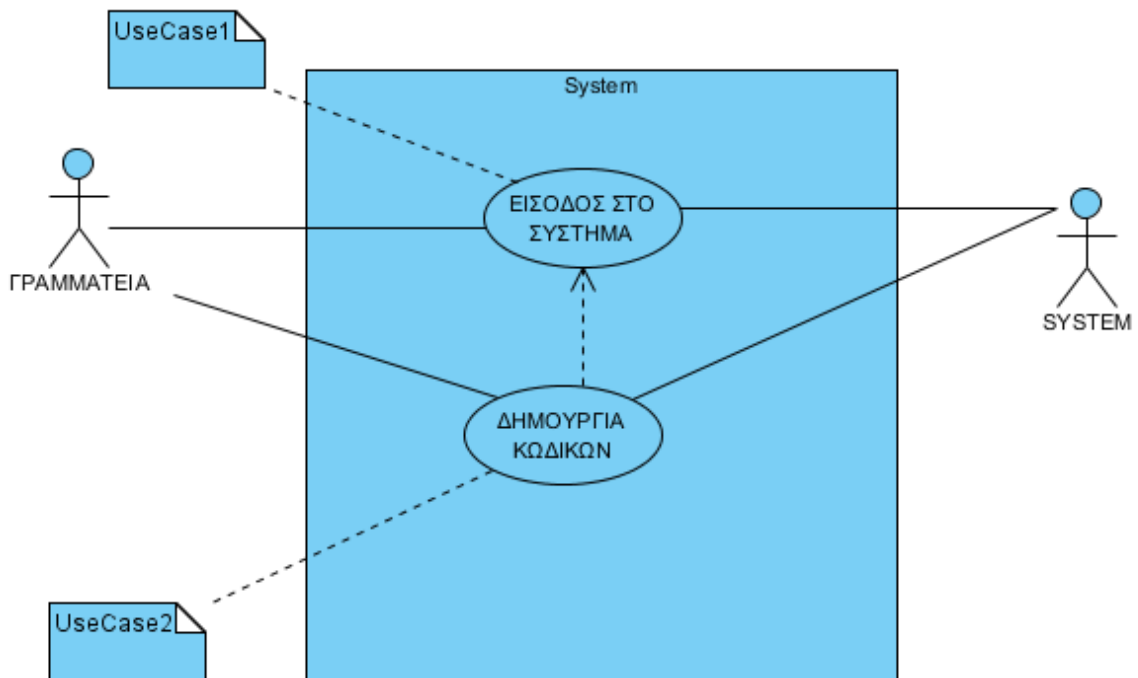
Εικόνα 6 "Διάγραμμα Περιπτώσεων Χρήσης Συστήματος Διαχείρισης Βιογραφικών"

4.2.2 ΔΙΑΓΡΑΜΜΑ ΠΕΡΙΠΤΩΣΕΩΝ ΧΡΗΣΗΣ ΥΠΟΨΗΦΙΟΥ



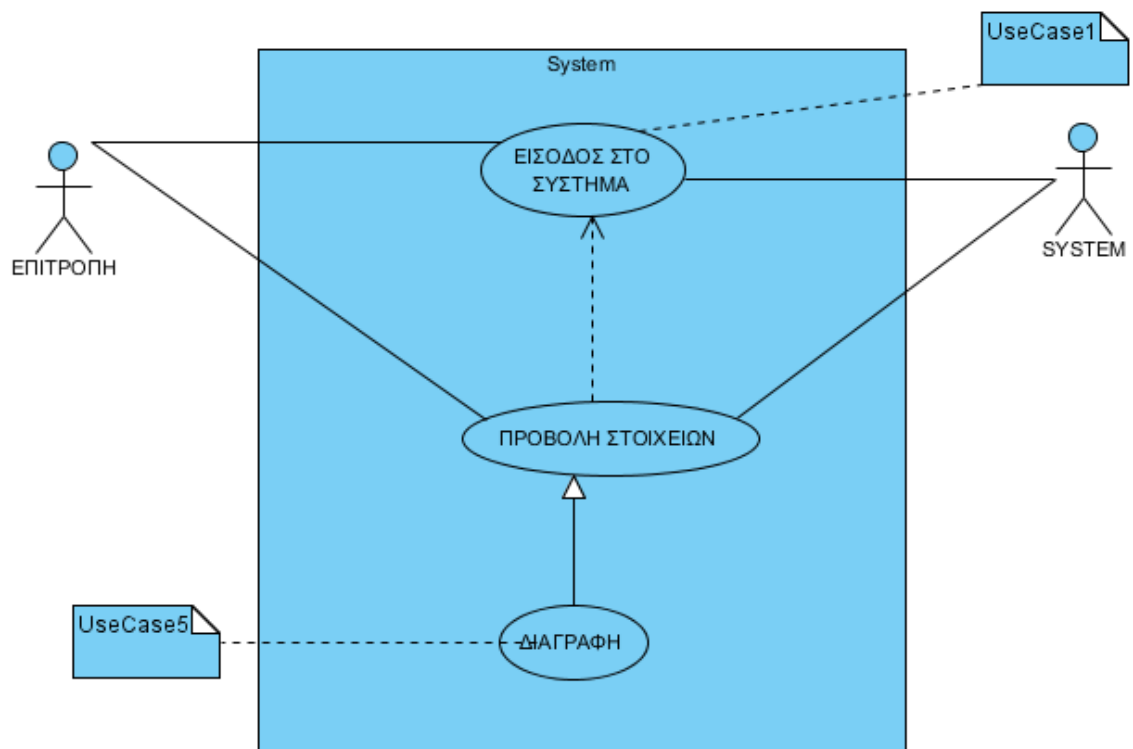
Εικόνα 7 "Διάγραμμα Περιπτώσεων Χρήσης Υποψηφίου"

4.2.3 ΔΙΑΓΡΑΜΜΑ ΠΕΡΙΠΤΩΣΕΩΝ ΧΡΗΣΗΣ ΓΡΑΜΜΑΤΕΑ



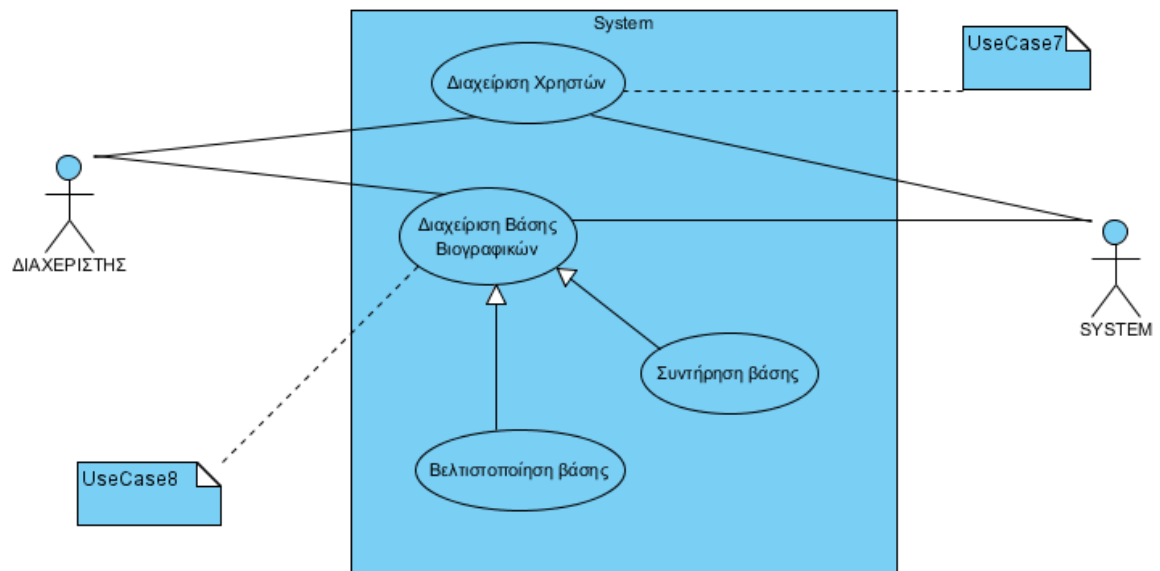
Εικόνα 8 "Διάγραμμα Περιπτώσεων Χρήσης Γραμματέα"

4.2.4 ΔΙΑΓΡΑΜΜΑ ΠΕΡΙΠΤΩΣΕΩΝ ΧΡΗΣΗΣ ΕΠΙΤΡΟΠΗΣ



Εικόνα 9 "Διάγραμμα Περιπτώσεων Χρήσης Επιτροπής"

4.2.5 ΔΙΑΓΡΑΜΜΑ ΠΕΡΙΠΤΩΣΕΩΝ ΧΡΗΣΗΣ ΔΙΑΧΕΙΡΙΣΤΗ



Εικόνα 10 "Διάγραμμα Περιπτώσεων Χρήσης Διαχειριστή"

4.3 ΠΙΝΑΚΑΣ ΧΡΗΣΤΩΝ ΣΤΟΧΩΝ

Χρήστης	Στόχος	Προτεραιότητα	Εκτίμηση προσπάθειας
Γραμματεία	Είσοδος Στο Σύστημα	1	
	Δημιουργία Κωδικών Υποψηφίων ΕΕΠ	1	
Υποψήφιο ΕΕΠ	Αίτηση Εγγραφής	1	
	Είσοδος Στο Σύστημα	1	
	Καταχώρηση Στοιχείων Βιογραφικού	1	
	Ενημέρωση Στοιχείων Βιογραφικού	2	
Επιτροπή	Είσοδος Στο Σύστημα	1	
	Προβολή Βιογραφικών	1	
	Διαγραφή Βιογραφικών Από Το Σύστημα	3	
Διαχειριστής	Διαχείριση Χρηστών του Συστήματος	1	
	Διαχείριση Βάσης Βιογραφικών	2	
	Βελτιστοποίηση Της Βάσης	1	
	Συντήρηση Της Βάσης	3	

Πίνακας 1 "Πίνακας Χρηστών Στόχων"

4.4 ΚΑΤΑΛΟΓΟΣ ΣΧΕΣΕΩΝ ΕΝΝΟΙΩΝ

Κατηγορία	Παραδείγματα
Το Α είναι φυσικό μέρος του Β	
Το Α είναι λογικό μέρος του Β	Μάθημα-Κατάλογος
Το Α περιέχεται φυσικά στο Β	
Το Α περιέχεται λογικά στο Β Το Α είναι μια περιγραφή για το Β	Ονοματεπώνυμο-Βιογραφικό
Το Α είναι ένα αντικείμενο γραμμής μιας συναλλαγής του Β	Βιογραφικό-Βάση Δεδομένων
Το Α αναφέρεται στο Β	
Το Α είναι μέλος του Β	Γραμματέας-Διαχείριση
Το Α είναι υποενότητα του Β Το Α χρησιμοποιεί ή χειρίζεται το Β	Επιτροπή-Σύστημα Διαχείρισης
Το Α επικοινωνεί με το Β	Γραμματέας-Υποψήφιος Επιτροπή-Υποψήφιος
Το Α σχετίζεται με μια συναλλαγή Β	Υποψήφιος-Καταχώριση Επιτροπή- Προβολή/Ταξινόμηση/Διαγραφή
Το Α είναι μια συναλλαγή που σχετίζεται με μια άλλη συναλλαγή Β	Προβολή-Διαγραφή Καταχώριση-Ενημέρωση
Το Α είναι δίπλα στο Β	Βιογραφικό-Βιογραφικό
Το Α ανήκει στο Β	
Το Α είναι ένα γεγονός που σχετίζεται με το Β	Δημιουργία Κωδικών-Γραμματέας Αίτηση Εγγραφής-Υποψήφιος

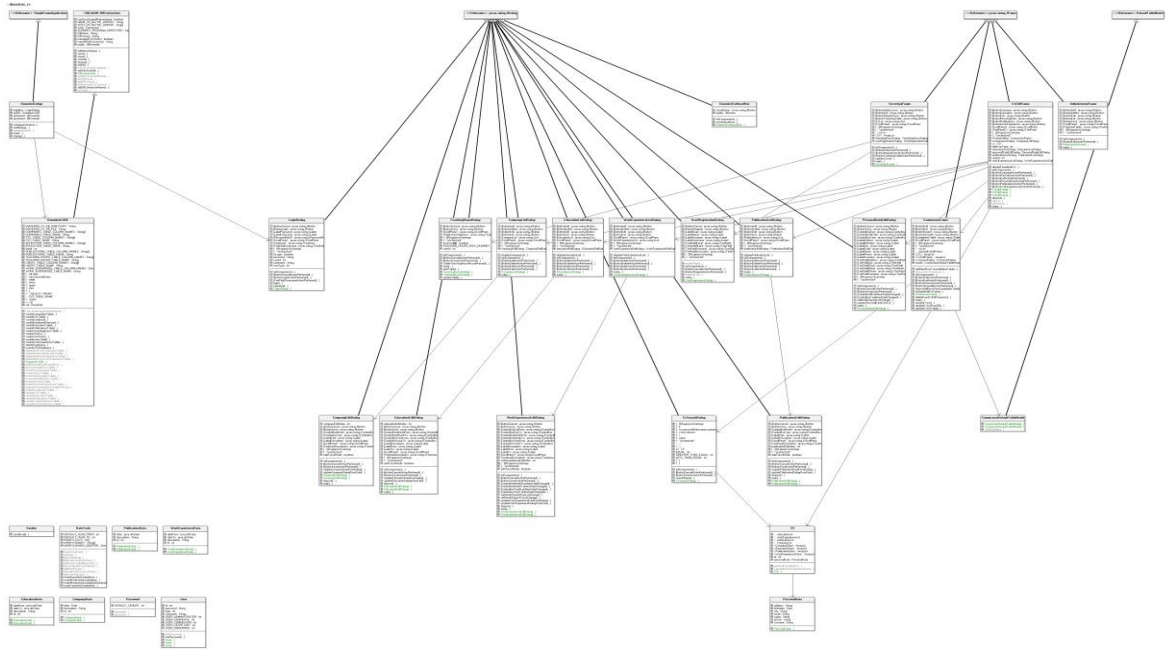
Πίνακας 2 "Κατάλογος Σχέσεων Εννοιών"

4.5 ΚΑΤΑΛΟΓΟΣ ΕΝΝΟΙΩΝ

Εννοιολογικές Κλάσεις	Παραδείγματα
Φυσικά αντικείμενα	
Προδιαγραφές/Περιγραφές	
Μέρη	
Συναλλαγές	Καταχώρηση, Ενημέρωση, Προβολή, Διαγραφή
Ρόλοι Χρηστών	Γραμματέας, Υποψήφιος, Επιτροπή
Εξωτερικά Συστήματα	Βάση Δεδομένων
Αφηρημένες έννοιες	
Γεγονότα	Είσοδος, Αίτηση Εγγραφής, Δημιουργία Κωδικών
Κανόνες και Πολιτικές	
Κατάλογοι	Μαθήματα
Διαδικασίες	Διαχείριση βιογραφικών
Εγχειρίδια	User manual

Πίνακας 3 "Κατάλογος Εννοιών"

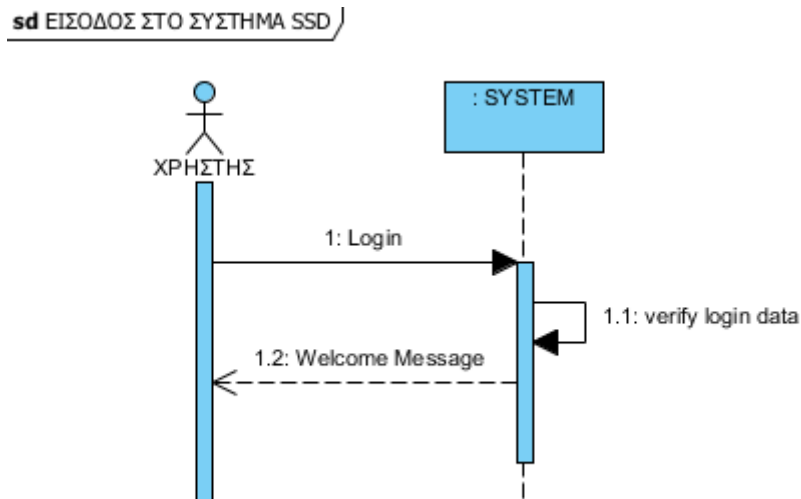
4.6 ΕΝΝΟΙΟΛΟΓΙΚΟ ΜΟΝΤΕΛΟ (DOMAIN MODEL)



Εικόνα 11 "Εννοιολογικό Μοντέλο Πεδίου Προβλήματος(Domain Model)"

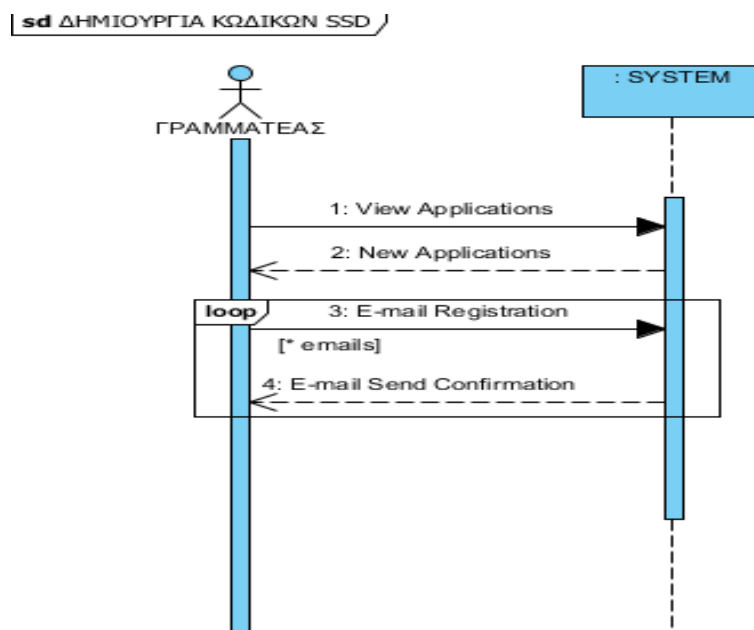
4.7 ΔΙΑΓΡΑΜΜΑΤΑ ΑΚΟΛΟΥΘΙΑΣ ΣΥΣΤΗΜΑΤΟΣ

4.7.1 ΕΙΣΟΔΟΣ ΣΤΟ ΣΥΣΤΗΜΑ



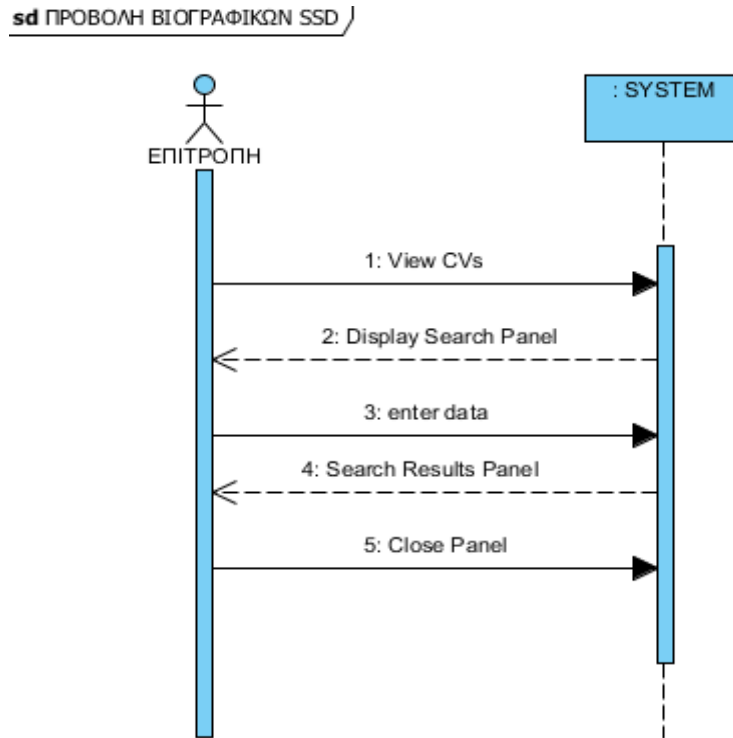
Εικόνα 12 "Διάγραμμα Ακολουθίας Συστήματος για την Π.Χ: Είσοδος στο Σύστημα"

4.7.2 ΔΗΜΙΟΥΡΓΙΑ ΚΩΔΙΚΩΝ



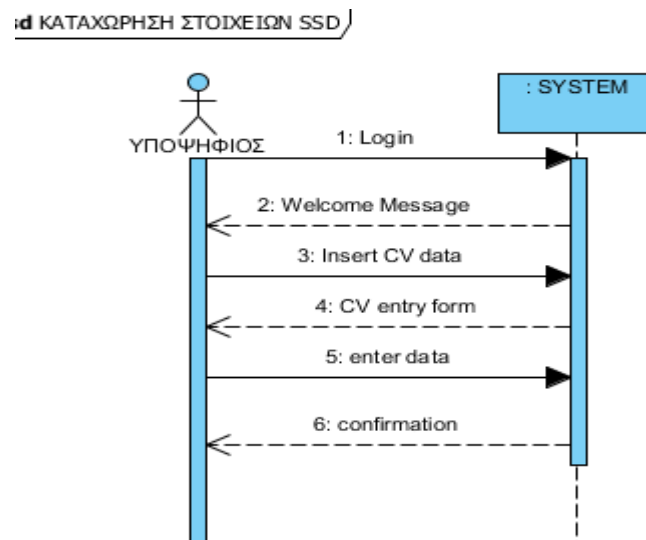
Εικόνα 13 "Διάγραμμα Ακολουθίας Συστήματος για την Π.Χ: Δημιουργία Κωδικών"

4.7.3 ΠΡΟΒΟΛΗ ΒΙΟΓΡΑΦΙΚΩΝ



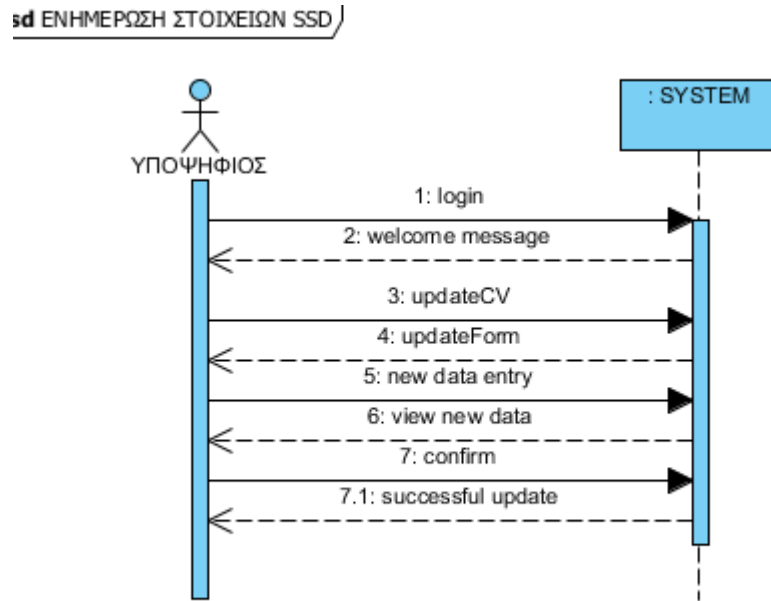
Εικόνα 14 "Διάγραμμα Ακολουθίας Συστήματος για την Π.Χ: Προβολή Βιογραφικών"

4.7.4 ΚΑΤΑΧΩΡΗΣΗ ΣΤΟΙΧΕΙΩΝ



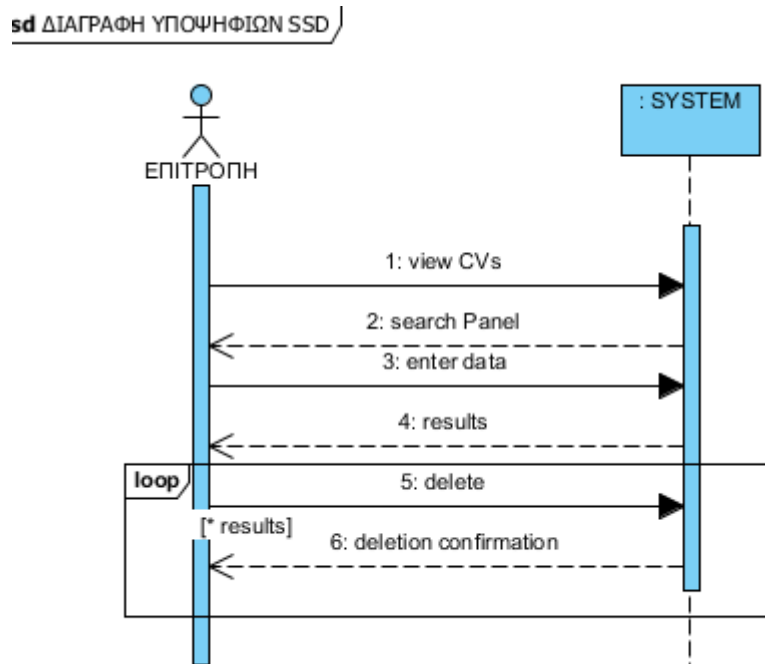
Εικόνα 15 "Διάγραμμα Ακολουθίας Συστήματος για την Π.Χ: Καταχώρηση Στοιχείων"

4.7.5 ΕΝΗΜΕΡΩΣΗ ΣΤΟΙΧΕΙΩΝ



Εικόνα 16 "Διάγραμμα Ακολουθίας Συστήματος για την Π.Χ: Ενημέρωση Στοιχείων"

4.7.6 ΔΙΑΓΡΑΦΗ ΥΠΟΨΗΦΙΩΝ

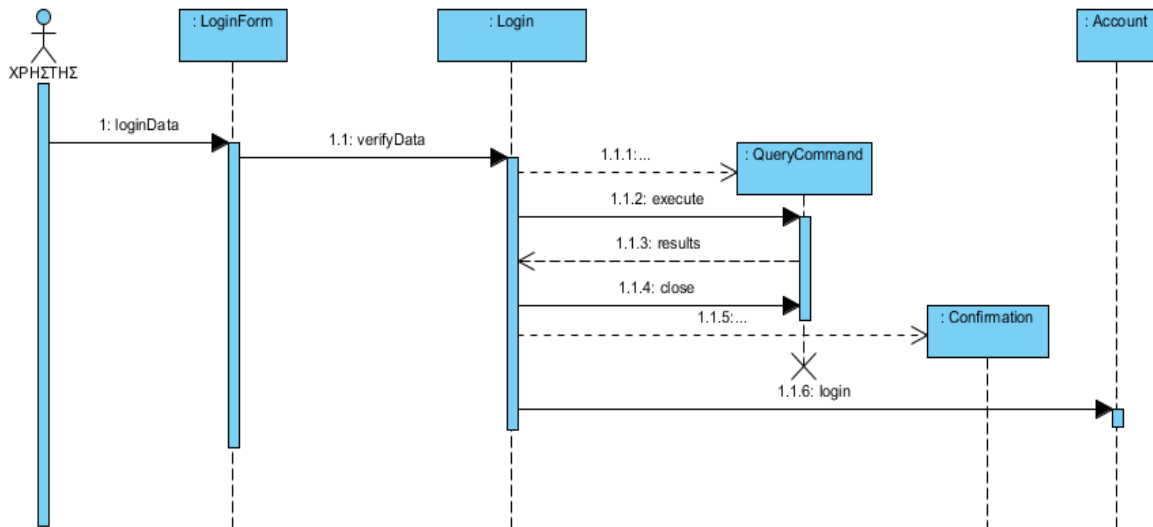


Εικόνα 14 "Διάγραμμα Ακολουθίας Συστήματος για την Π.Χ: Διαγραφή Υποψηφίων"

4.8 ΔΙΑΓΡΑΜΜΑΤΑ ΑΚΟΛΟΥΘΙΑΣ

4.8.1 ΕΙΣΟΔΟΣ ΣΤΟ ΣΥΣΤΗΜΑ

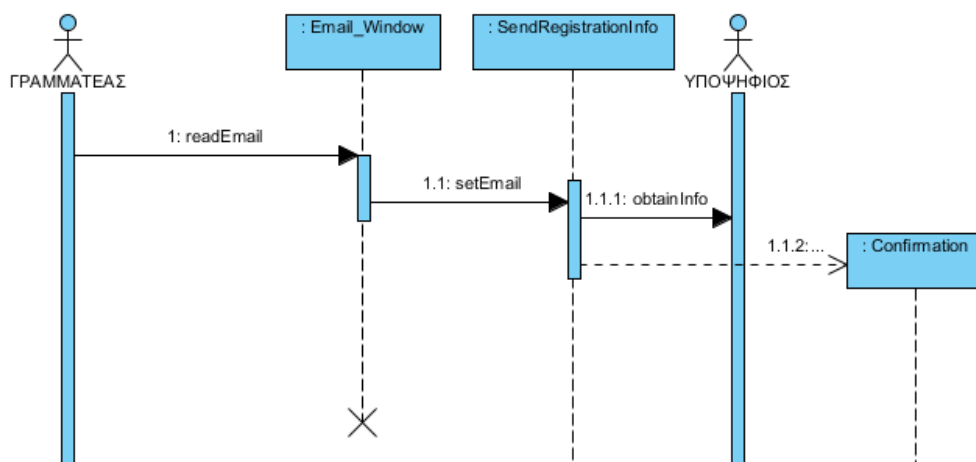
sd Eisodos sto Sistima /



Εικόνα 15 "Διάγραμμα Ακολουθίας για την Π.Χ: Είσοδος στο Σύστημα"

4.8.2 ΔΗΜΙΟΥΡΓΙΑ ΚΩΔΙΚΩΝ

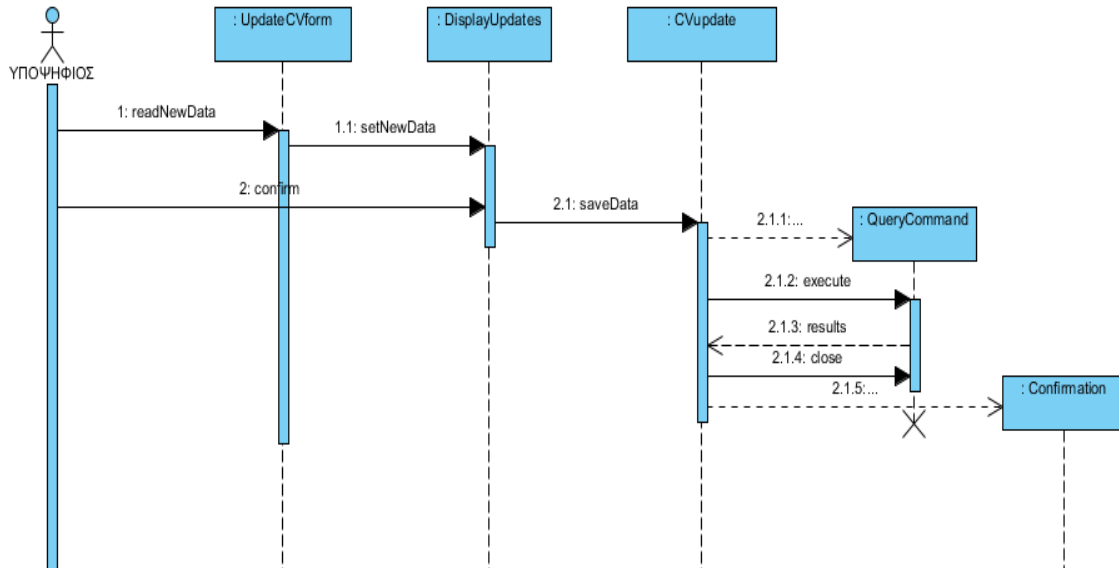
sd Dimiourgia Kodikon /



Εικόνα19 "Διάγραμμα Ακολουθίας για την Π.Χ: Δημιουργία Κωδικών"

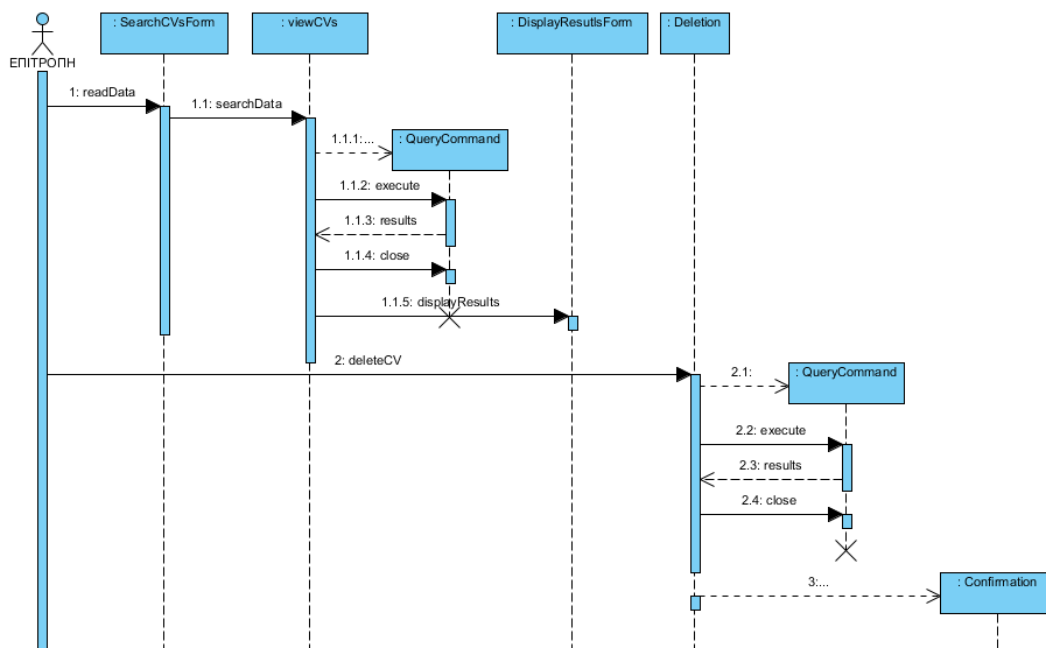
4.8.3 ΕΝΗΜΕΡΩΣΗ ΣΤΟΙΧΕΙΩΝ

sd Enimerosi Stoiheion



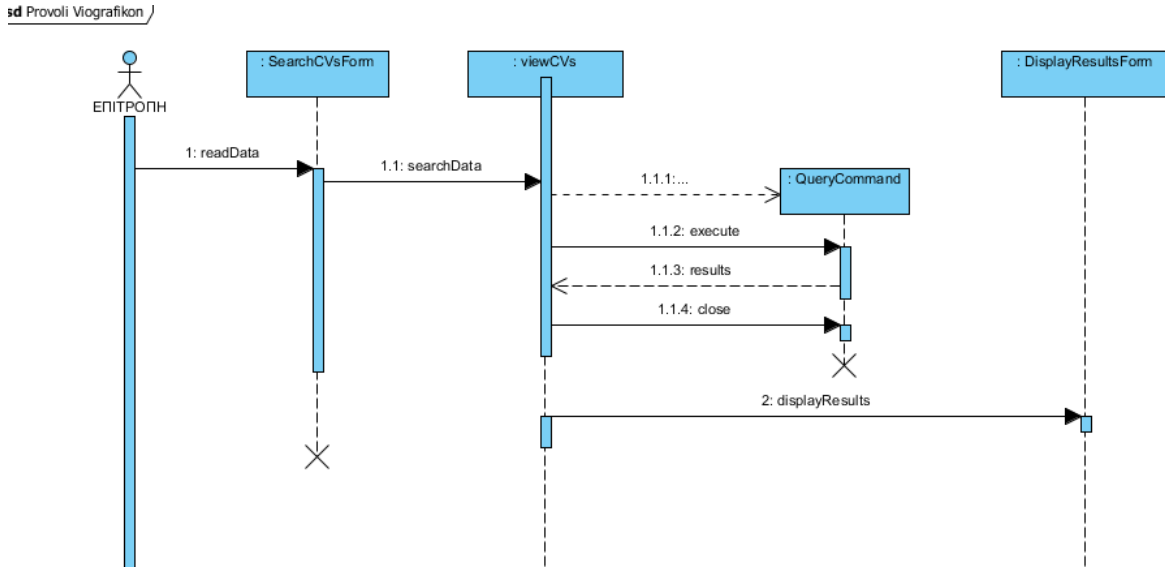
Εικόνα 20 "Διάγραμμα Ακολουθίας για την Π.Χ: Ενημέρωση Στοιχείων"

4.8.4 ΔΙΑΓΡΑΦΗ ΥΠΟΨΗΦΙΩΝ



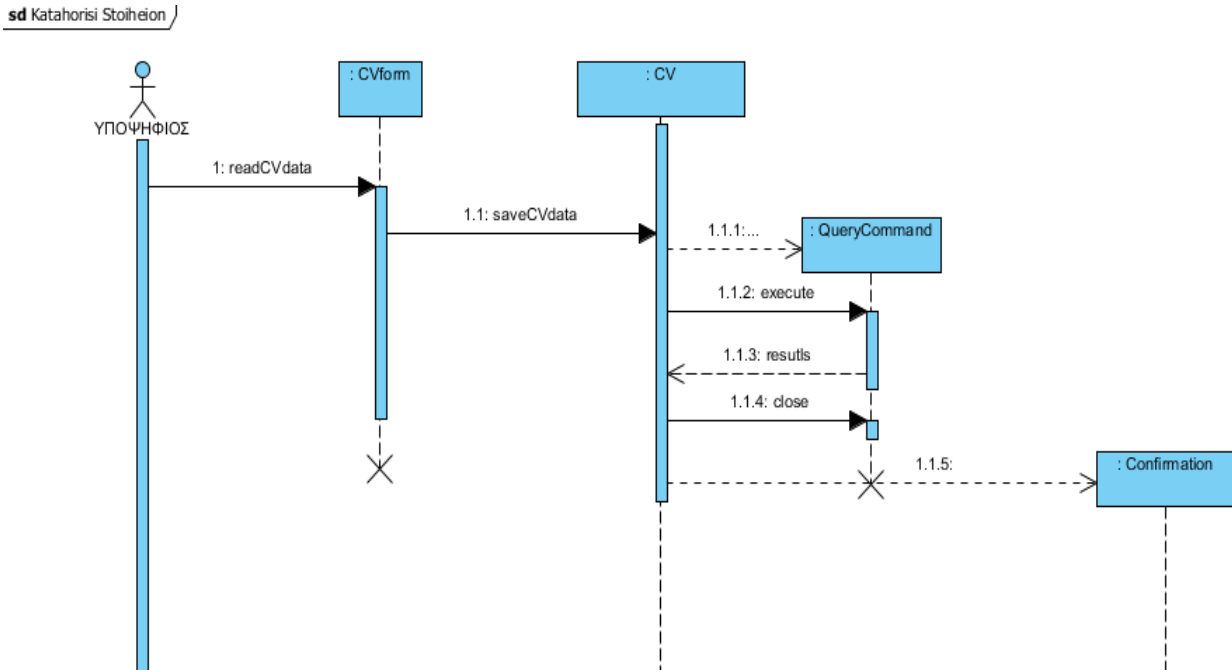
Εικόνα 16 "Διάγραμμα Ακολουθίας για την Π.Χ: Διαγραφή Υποψηφίων"

4.8.5 ΠΡΟΒΟΛΗ ΒΙΟΓΡΑΦΙΚΩΝ



Εικόνα 17 "Διάγραμμα Ακολουθίας για την Π.Χ: Προβολή Βιογραφικών"

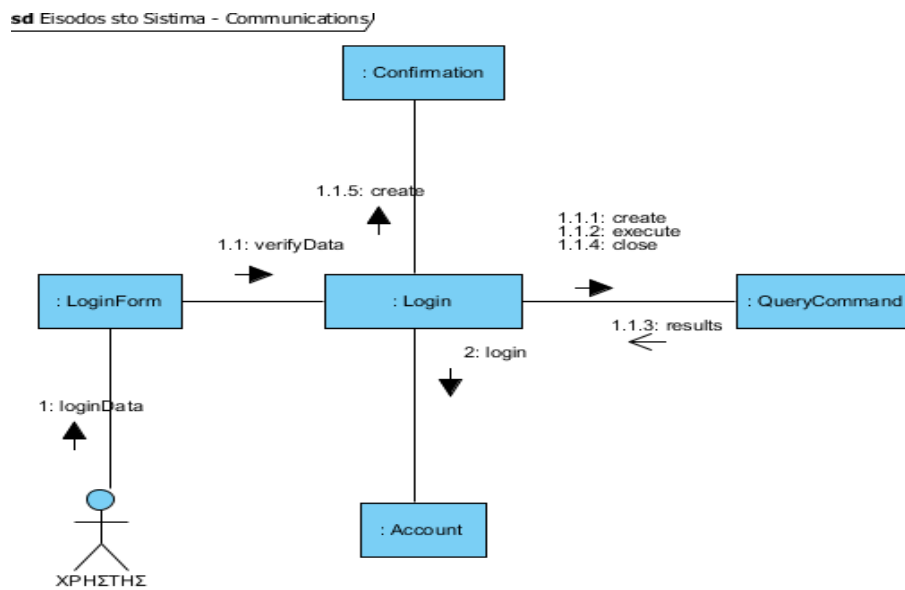
4.8.6 ΚΑΤΑΧΩΡΗΣΗ ΣΤΟΙΧΕΙΩΝ



Εικόνα 23 "Διάγραμμα Ακολουθίας για την Π.Χ: Καταχώρηση Στοιχείων"

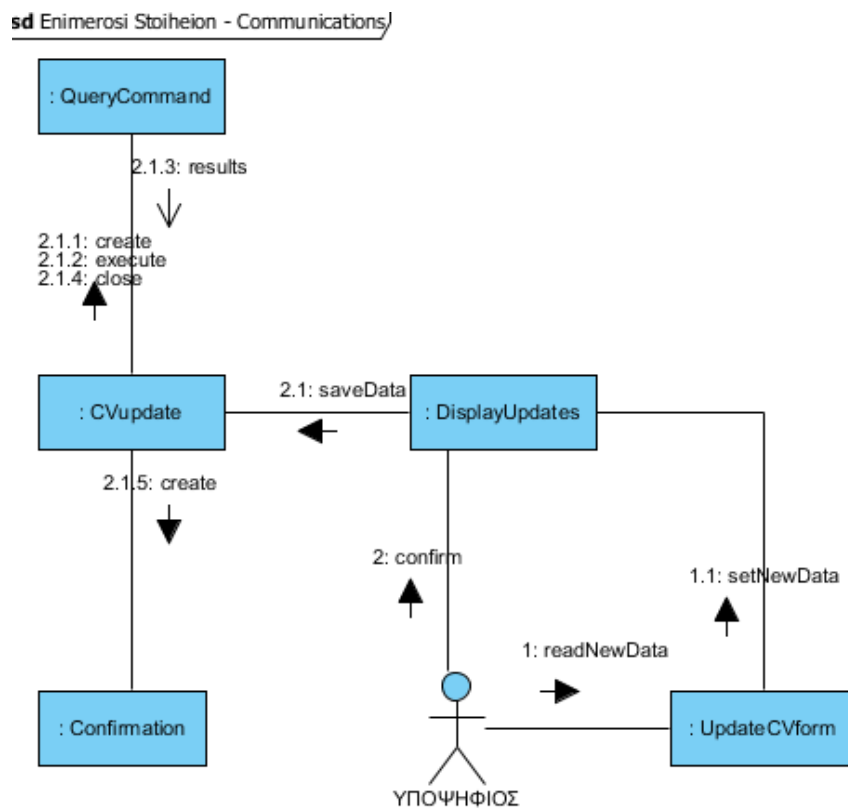
4.9 ΔΙΑΓΡΑΜΜΑΤΑ ΣΥΝΕΡΓΑΣΙΑΣ

4.9.1 ΕΙΣΟΔΟΣ ΣΤΟ ΣΥΣΤΗΜΑ



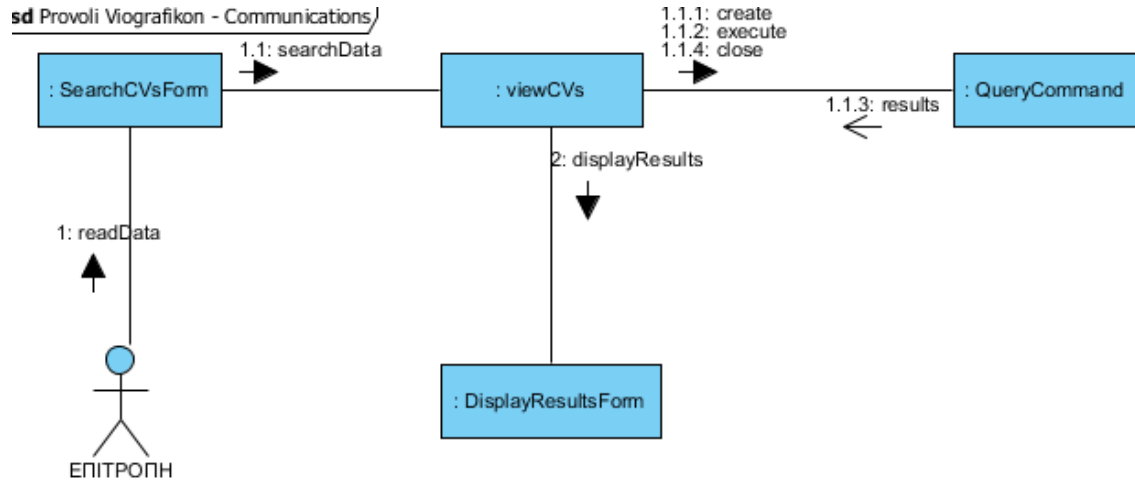
Εικόνα 24 "Διάγραμμα Συνεργασίας για την Π.Χ: Είσοδος στο Σύστημα"

4.9.2 ΕΝΗΜΕΡΩΣΗ ΣΤΟΙΧΕΙΩΝ



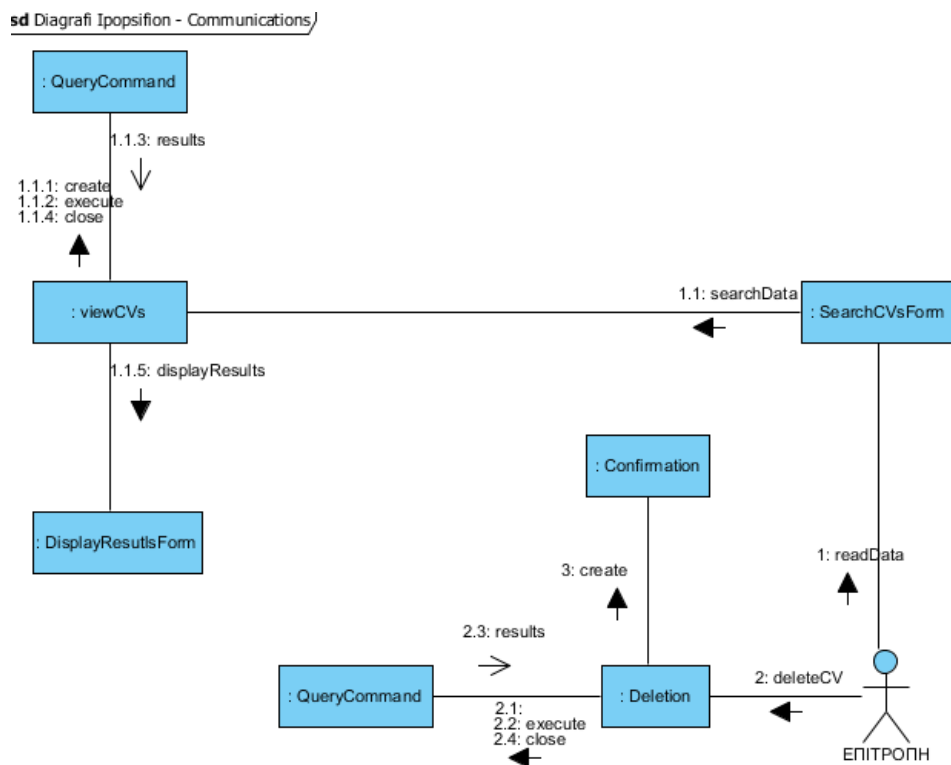
Εικόνα 18 "Διάγραμμα Συνεργασίας για την Π.Χ: Ενημέρωση Στοιχείων"

4.9.3 ΠΡΟΒΟΛΗ ΒΙΟΓΡΑΦΙΚΩΝ



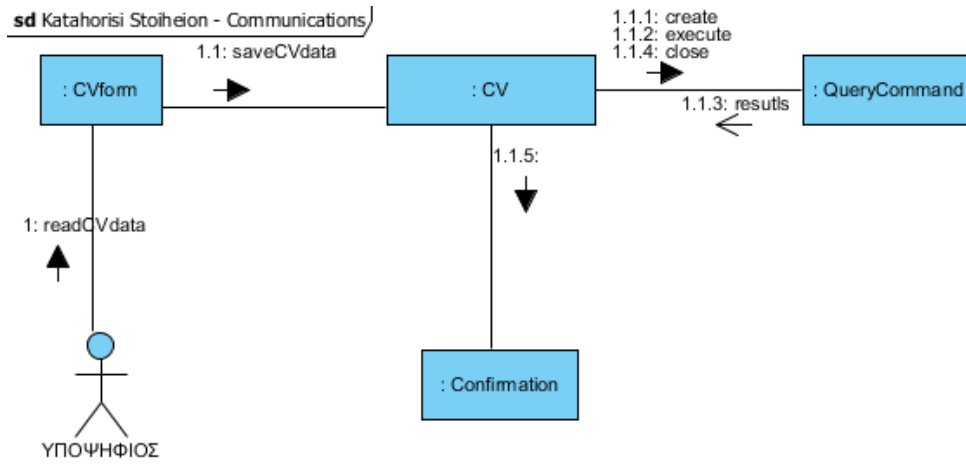
Εικόνα 26 "Διάγραμμα Συνεργασίας για την Π.Χ: Προβολή Βιογραφικών"

4.9.4 ΔΙΑΓΡΑΦΗ ΥΠΟΨΗΦΙΩΝ



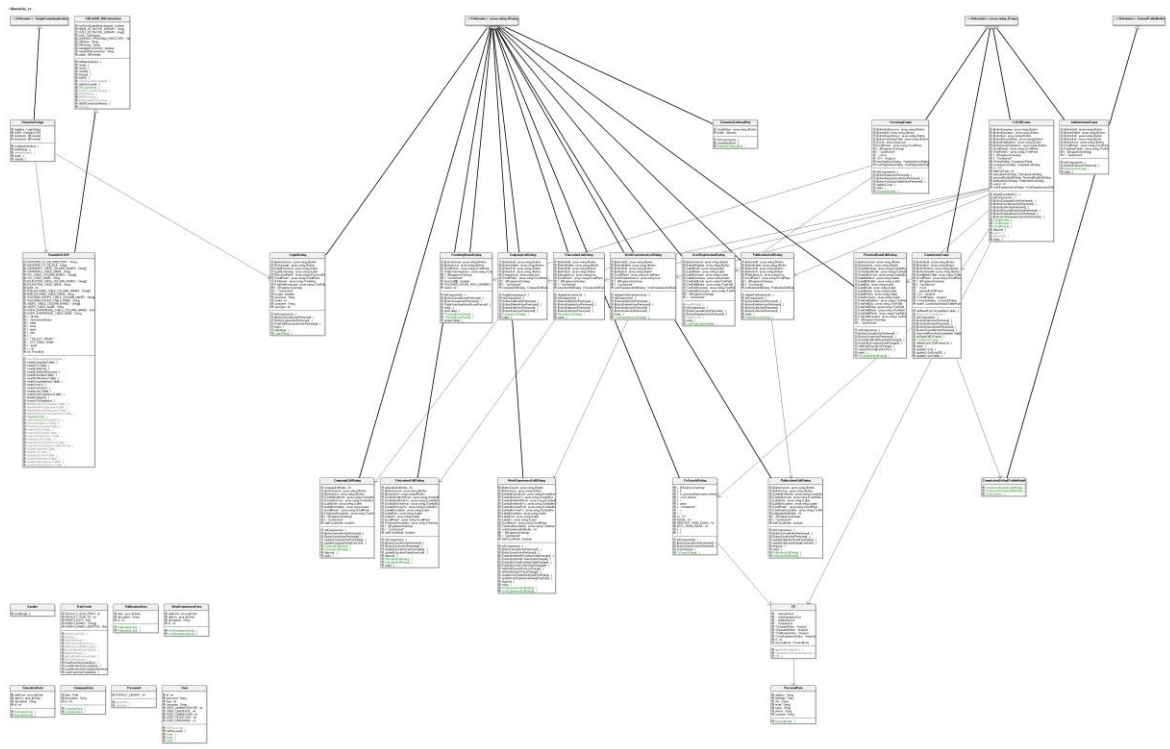
Εικόνα 27 "Διάγραμμα Συνεργασίας για την Π.Χ: Διαγραφή Υποψηφίων"

4.9.5 ΚΑΤΑΧΩΡΗΣΗ ΣΤΟΙΧΕΙΩΝ



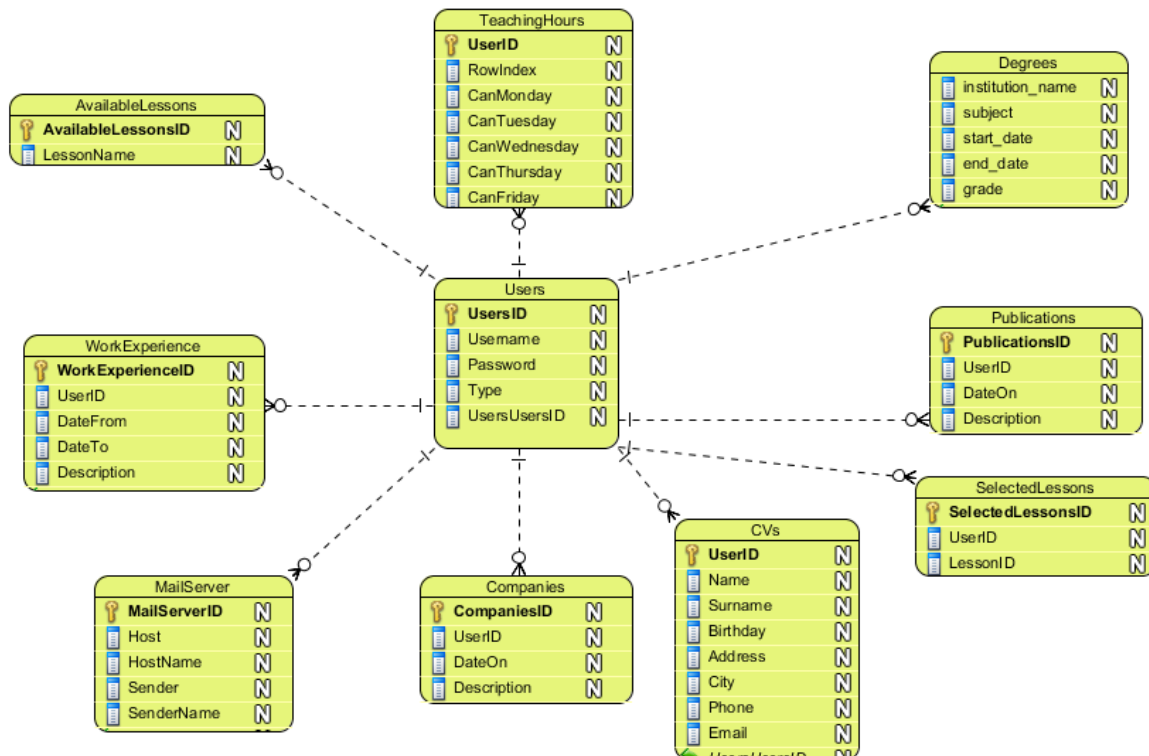
Εικόνα 19 "Διάγραμμα Συνεργασίας για την Π.Χ: Καταχώρηση Στοιχείων"

4.10 ΔΙΑΓΡΑΜΜΑ ΚΛΑΣΕΩΝ



Εικόνα 209 "Διάγραμμα Κλάσεων Συστήματος Διαχείρισης Βιογραφικών"

4.11 ΔΙΑΓΡΑΜΜΑ ΟΝΤΟΤΗΤΩΝ ΣΥΣΧΕΤΙΣΕΩΝ



Εικόνα 30 "Εννοιολογικό Διάγραμμα Οντοτήτων Συσχετίσεων της βάσης του συστήματος"

ΕΠΙΛΟΓΟΣ

Στο κεφάλαιο αυτό παρουσιάστηκαν οι αναλυτικές περιπτώσεις χρήσεις του συστήματος, λεκτικά και διαγραμματικά όπως επίσης και τα διαγράμματα ακολουθίας και συνεργασίας για κάθε περίπτωση χρήσης. Ακόμα παρουσιάστηκαν το εννοιολογικό μοντέλο του συστήματος, το διάγραμμα κλάσεων και το μοντέλο οντοτήτων-συσχετίσεων της βάσης δεδομένων. Στο επόμενο κεφάλαιο περιγράφονται οι κύριες λειτουργίες του συστήματος, και δίνονται κάποια στιγμιότυπα της εκτέλεσης κάθε λειτουργίας όπως επίσης και κάποια τμήματα κώδικα.

ΚΕΦΑΛΑΙΟ 5

ΥΛΟΠΟΙΗΣΗ

ΕΙΣΑΓΩΓΗ

Στο κεφάλαιο αυτό περιγράφονται οι κύριες λειτουργίες του συστήματος διαχείρισης ειδικού προσωπικού και παρατίθενται κάποιες λεπτομέρειες της

υλοποίησής τους. Αυτές οι λειτουργίες είναι η είσοδος στο σύστημα, η καταχώρηση και ενημέρωση βιογραφικού, η δημιουργία κωδικών, η προβολή και ταξινόμηση των βιογραφικών, και η διαγραφή ενός υποψηφίου από το σύστημα. Επίσης περιγράφονται οι πίνακες και το σχήμα της βάσης δεδομένων.

5.1 ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

Η βάση δεδομένων του συστήματος ΔΙΑΧΕΙΡΙΣΙΣCV περιέχει τους ακόλουθους πίνακες:

```
AvailableLessons (AvailableLessonsID INTEGER PRIMARY KEY
AUTOINCREMENT, LessonName VARCHAR(40))

CVs (UserID INTEGER, Name VARCHAR (40), Surname VARCHAR (40), Birthdate
DATE, Address VARCHAR (255), City VARCHAR (20), Phone VARCHAR
(20), Email VARCHAR (30), FOREIGN KEY(UserID) REFERENCES
Users(UserID))

Companies (CompaniesID INTEGER PRIMARY KEY AUTOINCREMENT, UserID
INTEGER, DateOn DATE, Description VARCHAR (255), FOREIGN KEY(UserID)
REFERENCES Users(UserID))

Education (EducationID INTEGER PRIMARY KEY AUTOINCREMENT, UserID
INTEGER, DateFrom DATE, DateTo DATE, Description VARCHAR (255), FOREIGN
KEY(UserID) REFERENCES Users(UserID))

MailServer (MailServerID INTEGER PRIMARY KEY AUTOINCREMENT, Host
VARCHAR (255), HostName VARCHAR (255), Sender VARCHAR (255), SenderName
VARCHAR (255))

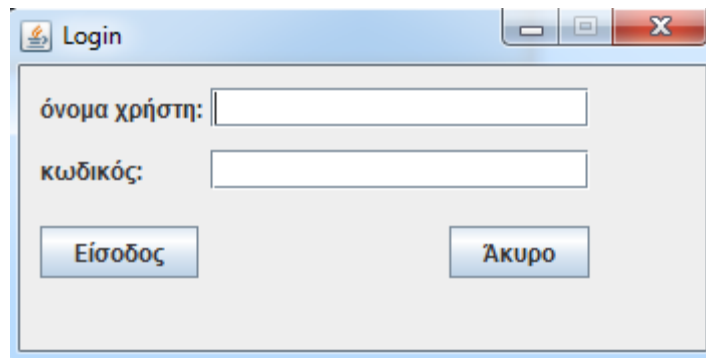
Publications (PublicationsID INTEGER PRIMARY KEY
AUTOINCREMENT, UserID INTEGER, DateOn DATE, Description VARCHAR
(255), FOREIGN KEY(UserID) REFERENCES Users(UserID))

SelectedLessons (SelectedLessonsID INTEGER PRIMARY KEY
AUTOINCREMENT, UserID INTEGER, LessonID INTEGER, FOREIGN KEY(UserID)
REFERENCES Users(UserID), FOREIGN KEY(LessonID) REFERENCES
AvailableLessons(AvailableLessonsID))

TeachingHours (UserID INTEGER, RowIndex INTEGER, CanMonday
BOOLEAN, CanTuesday BOOLEAN, CanWednesday BOOLEAN, CanThursday
BOOLEAN, CanFriday BOOLEAN, FOREIGN KEY(UserID) REFERENCES Users(UserID))
```

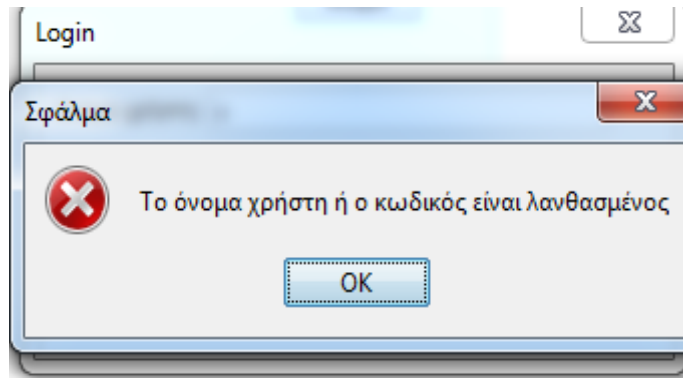
5.2 ΕΙΣΟΔΟΣ ΣΤΟ ΣΥΣΤΗΜΑ

Στο σύστημα διαχείρισης ειδικού προσωπικού υπάρχουν τέσσερις κατηγορίες χρηστών: υποψήφιος, γραμματέας, επιτροπή και διαχειριστής. Οι χρήστες αποθηκεύονται στη βάση δεδομένων στον πίνακα Users από το διαχειριστή του συστήματος. Ο χρήστης υποψήφιος εισάγεται με τα στοιχεία που αποστέλλονται στο e-mail που έχει καταχωρίσει στη γραμματεία. Κατά την εκκίνηση η εφαρμογή παρουσιάζει το παράθυρο εισόδου στο σύστημα. Αν δε γίνει είσοδος ο χρήστης δε μπορεί να χρησιμοποιήσει το σύστημα. Αν ο χρήστης είναι υποψήφιος και δεν έχει εγγραφεί στο σύστημα πρέπει να καταχωρίσει τα στοιχεία του στη γραμματεία.



Εικόνα 31 "Παράθυρο εισαγωγής όνομα χρήστη και κωδικού"

Μόλις ο χρήστης πιάσει «Είσοδος» εκτελείται ο παρακάτω κώδικας. Στη βάση εκτελείται το ερώτημα «Επέλεξε όνομα χρήστη, κωδικό και κατηγορία χρήστη από Users όπου όνομα χρήστη = <είσοδος χρήστη> και κωδικός = <είσοδος χρήστη>». Αν υπάρχει τέτοιος χρήστης, αν είναι υποψήφιος ενεργοποιείται το παράθυρο του για την καταχώρηση του βιογραφικού του. Αν είναι επιτροπή ενεργοποιείται το παράθυρο για την προβολή και διαγραφή των βιογραφικών. Αν είναι γραμματεία ενεργοποιείται το παράθυρο για την εγγραφή χρήστη, την επεξεργασία του πίνακα διδασκαλίας και για την επιλογή των μαθημάτων και αν είναι διαχειριστής ενεργοποιείται το παράθυρο για την προσθήκη και διαγραφή χρήστη, την επεξεργασία μαθημάτων και τις επιλογές του e-mail server. Αν ο χρήστης δεν υπάρχει στη βάση, εμφανίζεται μήνυμα λάθους και ο χρήστης δεν εισάγεται στο σύστημα.



Εικόνα 32 "Παράθυρο μηνύματος σφάλματος"

```
private void login()
{
    User usr = DiaxeirisiCvApp.myDB.getFromUsersTable(username, password);

    userId = usr.id;
    userType = usr.type;

    if (userType == User.USER_UNDEFINED)
        isLogin = false;

    else
        isLogin = true;
}
```


Διαχείριση Βιογραφικών Υποψηφίων Εκπαιδευτικών - Επεξεργασία Βιογραφικού

Προσωπικά Στοιχεία

Εκπαίδευση

Δημοσιεύσεις

Προϋπηρεσία

Ίδρυση Εταιρίας

Έξοδος

Mike1 Mixos1

Ημερομηνία Γέννησης: 23 Νοεμβρίου 1981
Διεύθυνση: Ελευθερίου 16 - Αρκαδία 53000
Πόλη: Θεσσαλονίκη1
Τηλέφωνο: 2310300400
Email: mike@hotmail.com

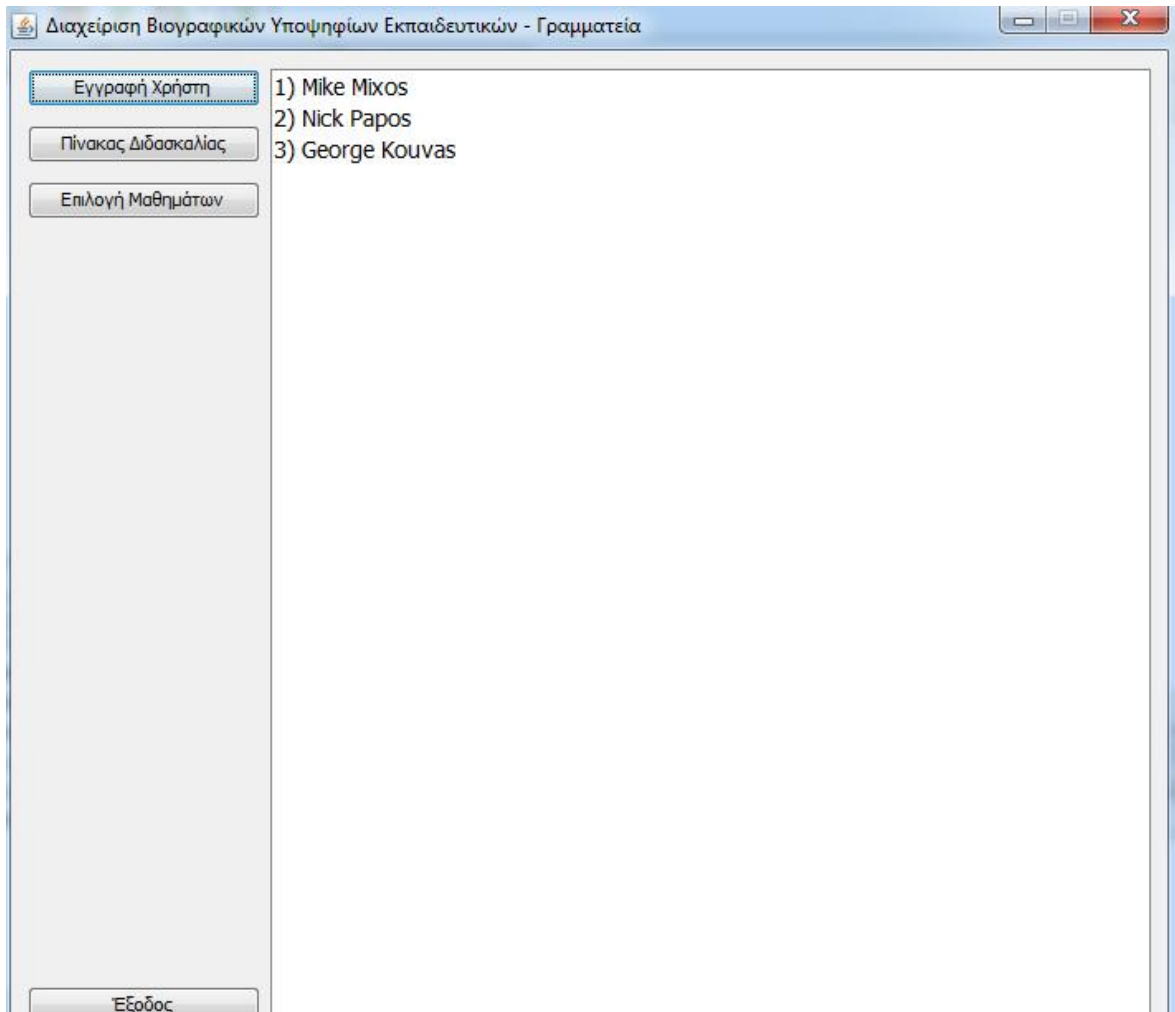
Εκπαίδευση

- *Ιανουάριος 1990 - Δεκέμβριος 1993*
 - Test123 University - BSc in Theoretical Theories
- *Σεπτέμβριος 1994 - Σεπτέμβριος 1996*
 - Theical University - MSc in Theical Theistry

Δημοσιεύσεις

- *Σεπτέμβριος 1999*
 - Data Reconstructor Technologies in Tabular Data Approximating Telematic Devices Error Logs - PHd Unreal University

Εικόνα 33 "Φόρμα Καταχώρησης Βιογραφικού-Υποψήφιος"



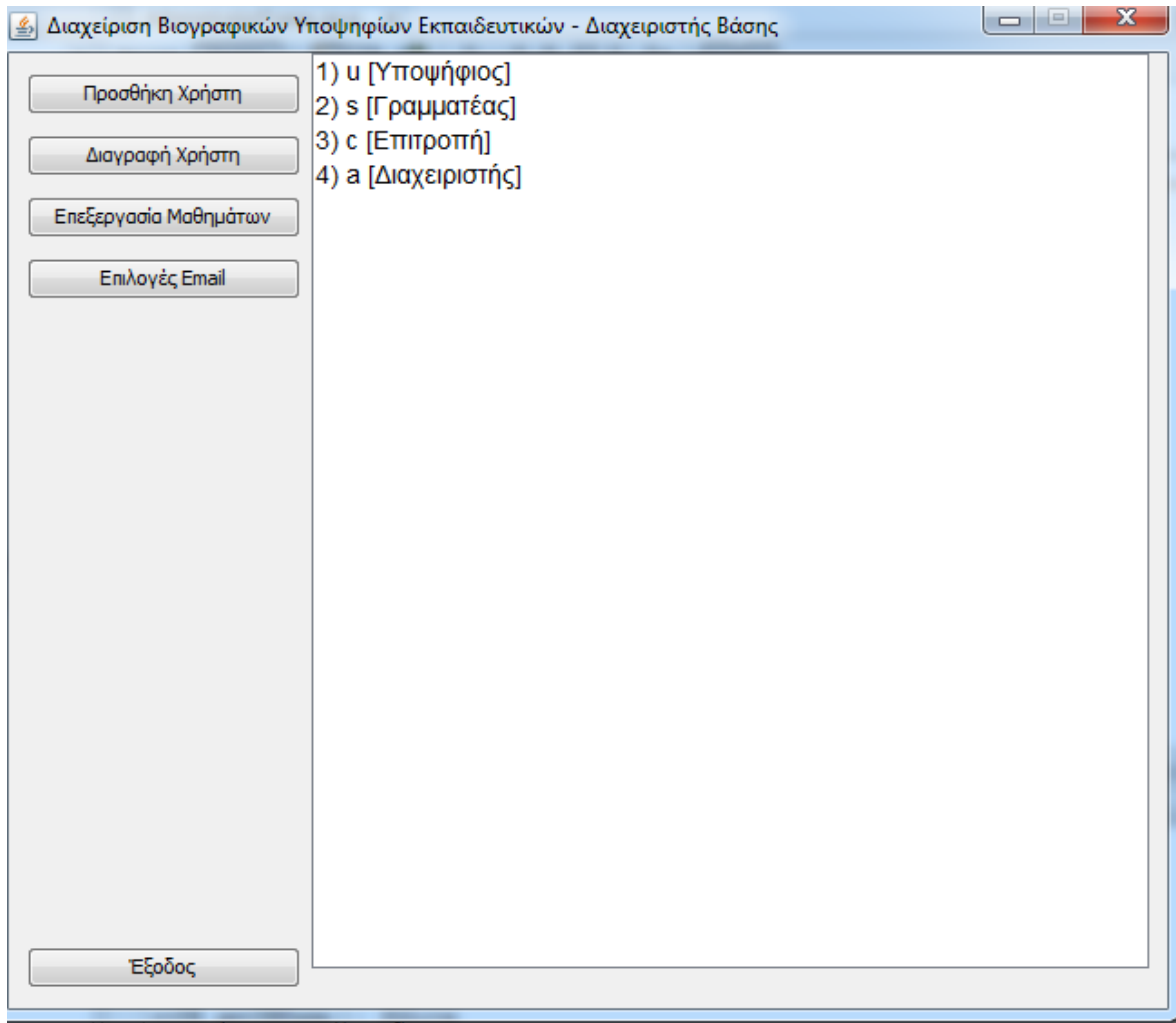
Εικόνα 34"Φόρμα Εγγραφής Υποψηφίου-Γραμματεία"

Πτυχιακή εργασία του φοιτητή Μασούρα Μιχάλη

Επεξεργασία	Ονοματεπώνυμο	Ηλικία	Τόπος Διαμονής	Αριθμός Δημοσιεύσεων	Αριθμός Εταιριών	Μέρες Προϋπηρεσίας
Διαγραφή	Mike Mixos	29	Θεσσαλονίκη	1	1	827
Αναζήτηση	Nick Papos	28	Ιωάννινα	3	1	633
Εμφάνιση Όλων	George Kouvas	42	Αθήνα	2	1	305

Έξοδος

Εικόνα 35 "Φόρμα Προβολής Βιογραφικών-Επιτροπή "



Εικόνα 36"Φόρμα Επεξεργασίας Χρηστών,Μαθημάτων Και E-mail-Διαχειριστής"

Ο χρήστης πρέπει να καταχωρήσει τα προσωπικά στοιχεία και την εκπαίδευση του,την προϋπηρεσία του, τις δημοσιεύσεις του και το όνομα της εταιρίας αν έχει ιδρύσει.Μόλις επιλέξει αποθήκευση τα δεδομένα που έχει εισάγει αποθηκεύονται στους πίνακες της βάσης. Παρακάτω φαίνεται η SQL command συνοπτικά και η εισαγωγή των προσωπικών στοιχείων του υποψηφίου στον πίνακα CVs:

```
INSERT INTO Cvs (UserID, Name, Surname, Birthdate, Address, City, Phone, Email) VALUES (userId, name, surname, DateTools.getSqlDateFromJavaDate(birthdate), address, city, phone, email)
```

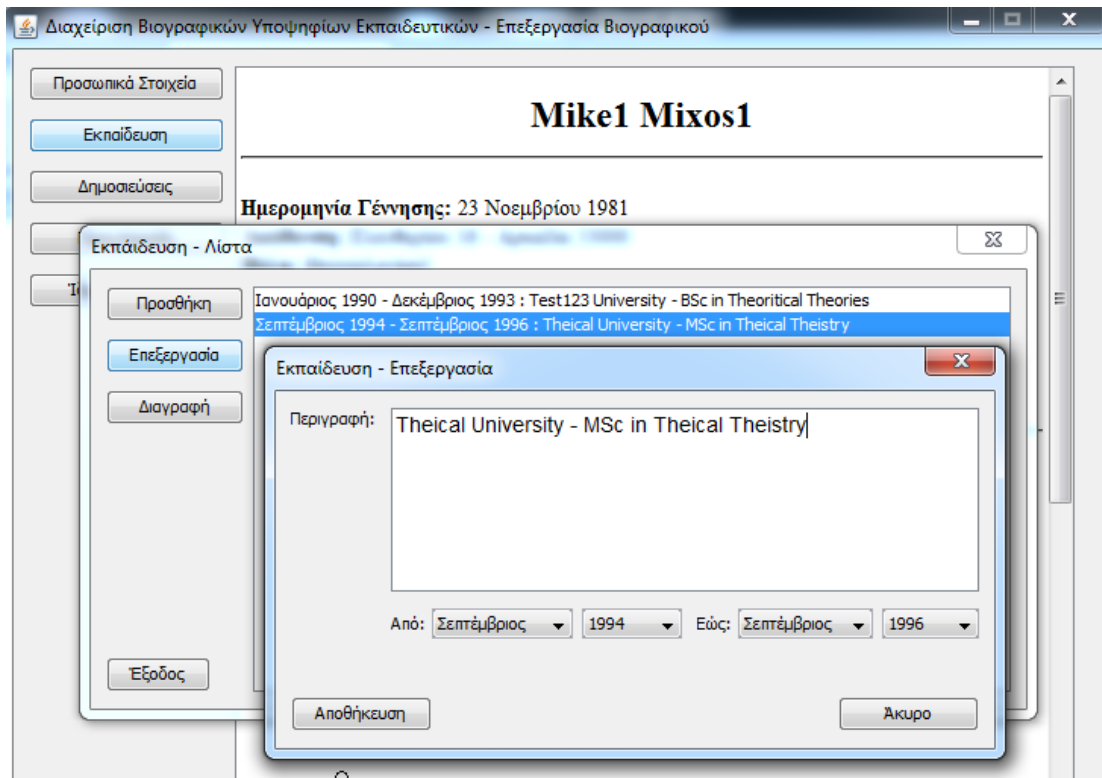
```
public boolean insertIntoCVsTable(int userId, String name, String surname, java.util.Date birthdate, String address, String city, String phone, String email)
```

```
{
```

```
boolean success = true;
```

```
String query = null;
```

Κατά την ενημέρωση εμφανίζεται το ίδιο frame με τα στοιχεία του χρήστη συμπληρωμένα. Ο υποψήφιος μπορεί να συμπληρώσει τα υπόλοιπα πεδία που θέλει, τα οποία θα προστεθούν στο βιογραφικό του.

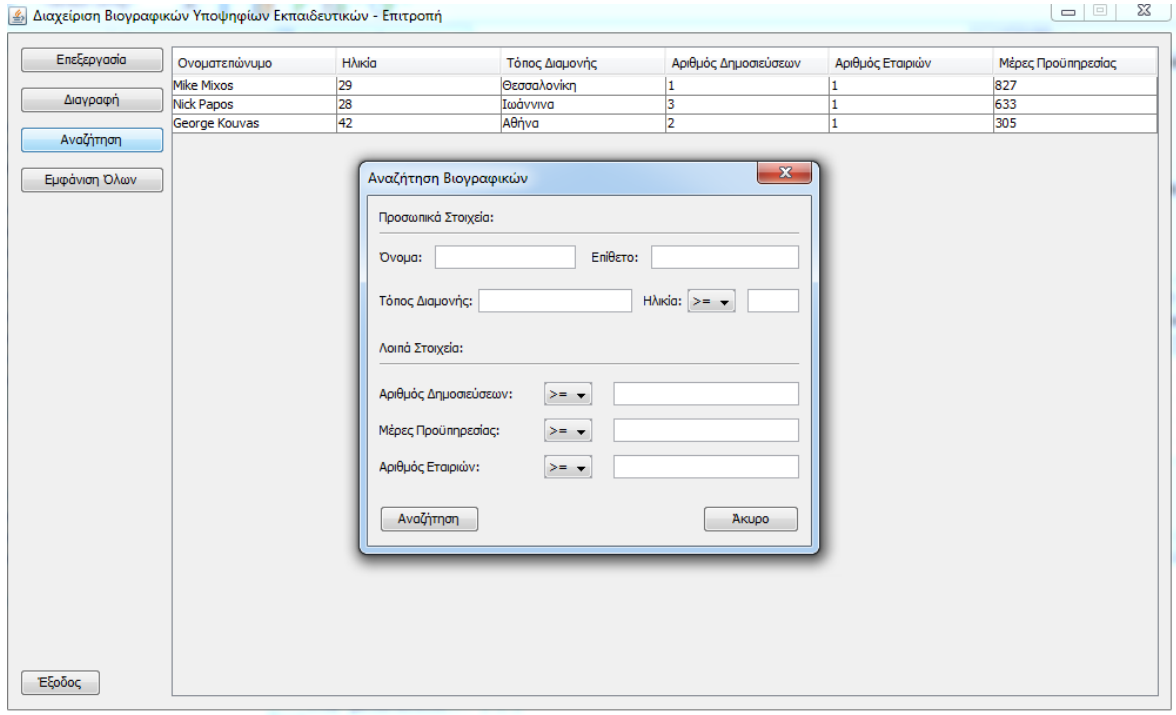


Εικόνα 37 "Φόρμα Ενημέρωσης Βιογραφικού"

5.4 ΠΡΟΒΟΛΗ /ΔΙΑΓΡΑΦΗ ΒΙΟΓΡΑΦΙΚΩΝ

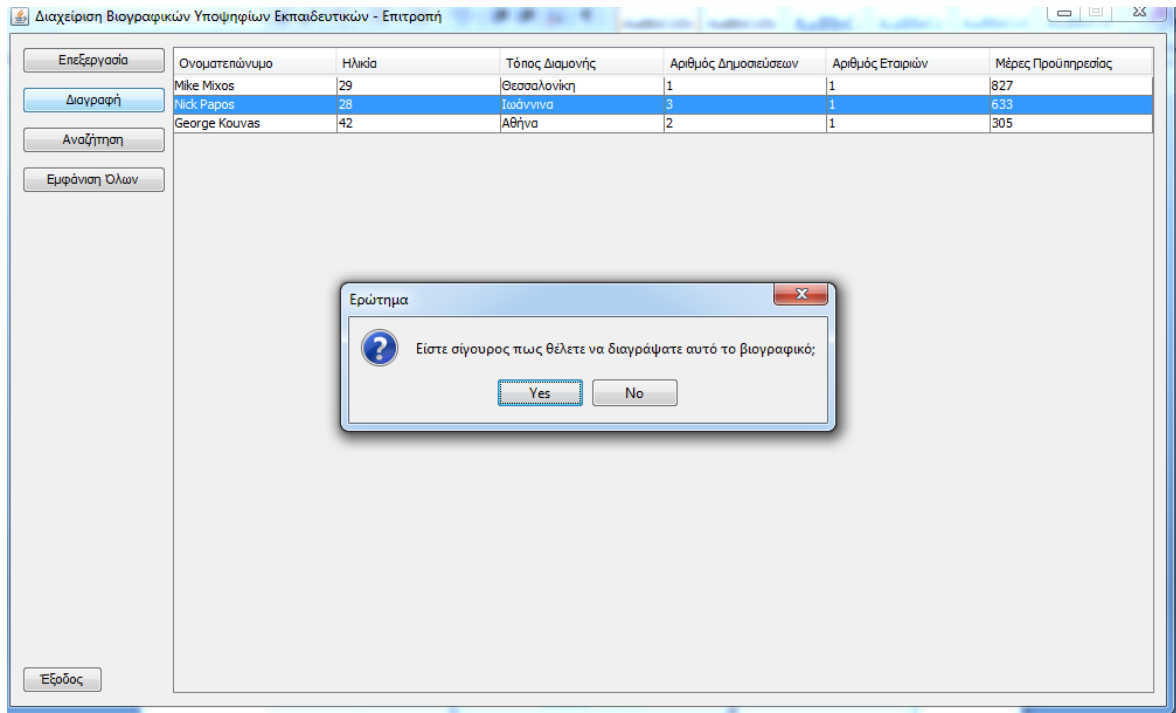
Σε αυτή την επιλογή έχουν πρόσβαση μόνο οι χρήστες που εισάγονται ως επιτροπή. Υπάρχουν 2 επιλογές αναζήτησης: με βάση τα προσωπικά στοιχεία του υποψηφίου όπως όνομα, επίθετο, τόπος διαμονής και ηλικία και με βάση κάποιων λοιπών στοιχείων όπως τον αριθμό των δημοσιεύσεων, το χρονικό διάστημα προϋπηρεσίας και τον αριθμό των εταιριών που έχει ιδρύσει. Αφού ο χρήστης επιλέξει αναζήτηση, εμφανίζονται στους πίνακες τα αποτελέσματα σύμφωνα με τα κριτήρια που έθεσε. Η φόρμα Προβολής Βιογραφικών φαίνεται παρακάτω:

Πτυχιακή εργασία του φοιτητή Μασούρα Μιχάλη

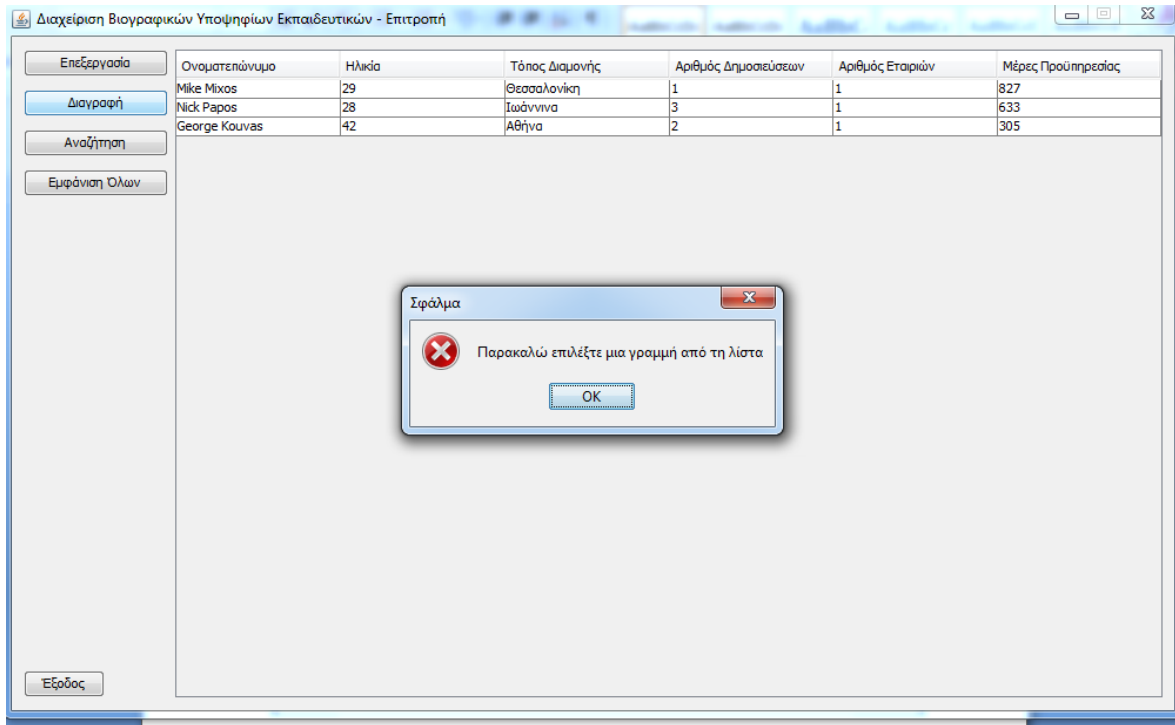


Εικόνα 38 "Φόρμα Αναζήτησης Βιογραφικών"

Ακόμα ο χρήστης έχει τη δυνατότητα να διαγράψει όποιον υποψήφιο επιθυμεί, επιλέγοντας τον από τη λίστα στο πλαίσιο που φαίνεται στην παρακάτω εικόνα, και να πιάσει Διαγραφή:



Εικόνα 21 "Επιλογή Διαγραφής Υποψηφίου"



Εικόνα 40 "Μήνυμα Σφάλματος Επιλογής Διαγραφής Υποψηφίου"

Κατά τη Διαγραφή, εκτελείται ο παρακάτω κώδικας:

```
DELETE FROM Users WHERE userID = userId

public boolean deleteUser(int userId, int type)
{
    boolean success = true;

    if (type==User.USER_CANDIDATE)
    {
```

Γίνεται σύνδεση στη βάση αν δεν υπάρχει ήδη και εκτελείται το ερώτημα. Αν εκτελεστεί σωστά επιστρέφεται ο ακέραιος 1 και εμφανίζεται το μήνυμα επιτυχούς διαγραφής. Αν δεν εκτελεστεί το ερώτημα, δηλαδή ο υποψήφιος δεν υπάρχει στη βάση, εμφανίζεται μήνυμα λάθους.

5.5 ΕΓΓΡΑΦΗ ΥΠΟΨΗΦΙΩΝ

Αυτή η λειτουργία εκτελείται από τη γραμματεία. Μόλις ο χρήστης επιλέξει Εγγραφή Χρήστη από το κεντρικό μενού εμφανίζεται το panel για την καταχώρηση των στοιχείων του υποψηφίου που έχουν κάνει αίτηση στη γραμματεία για εγγραφή. Ο χρήστης στη συνέχεια εισάγει τα στοιχεία του υποψηφίου και επιλέγει εγγραφή.

The screenshot shows a web application window titled "Διαχείριση Βιογραφικών Υποψηφίων Εκπαιδευτικών - Γραμματεία". On the left, there is a sidebar with three buttons: "Εγγραφή Χρήστη" (highlighted in blue), "Πίνακας Διδασκαλίας", and "Επιλογή Μαθημάτων". At the bottom of the sidebar is an "Εξοδος" button. The main content area on the right displays a list of three candidates: "1) Mike Mixos", "2) Nick Papos", and "3) George Kouvas". A modal dialog box titled "Εγγραφή Χρήστη" is centered on the screen. This dialog has a title bar with a close button (X) and contains four input fields labeled "Όνομα:", "Επίθετο:", "Username:", and "Email:". Below the input fields are two buttons: "Εγγραφή" and "Άκυρο".

Εικόνα 41 "Παράθυρο Εγγραφής Υποψηφίου"

Κατά την αποστολή e-mail , εκτελείται ο παρακάτω κώδικας:

```
public static boolean sendEmail(String recipient, String password)
{
    MailServerData mailServerData = new MailServerData();
    DiaxeirisiCvApp.myDB.getMailServerData(mailServerData);

    boolean success = true;

    try
    {
        Properties props = new Properties( );
        props.put(mailServerData.host.trim(), mailServerData.hostname.trim());
        Session mailConnection = Session.getInstance(props, null);
        Message msg = new MimeMessage(mailConnection);
        Address nick = new InternetAddress(mailServerData.sender.trim(), mailServerData.senderName.trim());
        Address recep = new InternetAddress(recipient.trim());

        msg.setContent("Your password is: " + password, "text/plain");

        msg.setFrom(nick);
        msg.setRecipient(Message.RecipientType.TO, recep);
        msg.setSubject("Ο κωδικός σας!!!");
        Transport.send(msg);
    }
}
```

Κατά τον υπολογισμό της προϋπηρεσίας, εκτελείται ο παρακάτω κώδικας:

```
public static int getDaysFromMilliseconds(long mls)
{
    return (int)((((mls / 1000) / 60) / 60) / 24) + 1;
}

public static int getWorkingDays(Date dateStart, Date dateEnd)
{
    long ageInMls = dateEnd.getTime() - dateStart.getTime();

    int allDays = getDaysFromMilliseconds(ageInMls);

    int dayStart = dateStart.getDay();
    int dayEnd = dateEnd.getDay();

    int workingDaysOnLeftPiece = 0;
    int workingDaysOnRightPiece = 0;

    for (int i=dayStart; i<7; i++)
    {
        if (i>0 && i<6)
        {
            workingDaysOnLeftPiece++;
        }
    }

    for (int i=dayEnd; i>=0; i--)
```

ΕΠΙΛΟΓΟΣ

Στο κεφάλαιο αυτό περιγράφηκαν αναλυτικά οι λειτουργίες του συστήματος με την παράθεση στιγμιότυπων και τμημάτων κώδικα, καθώς και με την περιγραφή της σειράς ενεργειών ώστε να πραγματοποιηθεί μια λειτουργία. Το επόμενο κεφάλαιο που αποτελεί το τελευταίο κεφάλαιο της διπλωματικής αναφέρει τα τελικά συμπεράσματα της υλοποίησης του συστήματος και προτάσεις για πιθανές μελλοντικές βελτιώσεις.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Στο κεφάλαιο αυτό περιγράφονται οι αρχικοί στόχοι της διπλωματικής, τι υλοποιήθηκε τελικά, καθώς και κάποιες ιδέες για μελλοντικές επεκτάσεις και βελτιώσεις.

ΑΡΧΙΚΟΣ ΣΤΟΧΟΣ

Αρχικός Στόχος της Διπλωματικής ήταν η δημιουργία ενός συστήματος διαχείρισης βιογραφικών υποψηφίων μελών ΕΠ που θα περιλάμβανε τις λειτουργίες καταχώρησης και ενημέρωσης του βιογραφικού από τους υποψηφίους, προβολής και ταξινόμησης βιογραφικών από τη επιτροπή και δημιουργίας και αποστολής κωδικών από τη γραμματεία.

ΤΕΛΙΚΗ ΥΛΟΠΟΙΗΣΗ

Στην εφαρμογή υλοποιήθηκαν όλες οι λειτουργίες που τέθηκαν στον αρχικό στόχο της εφαρμογής, αφού έγινε αρχικά η ανάλυση και ο σχεδιασμός τους με αντικειμενοστρεφείς μεθόδους (περιπτώσεις χρήσεις, διαγράμματα κλάσεων, διαγράμματα αλληλεπίδρασης κ.τ.λ.).

ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ ΚΑΙ ΒΕΛΤΙΩΣΕΙΣ

Υπάρχουν αρκετές ιδέες για μελλοντικές επεκτάσεις και βελτιώσεις του συστήματος που θα μπορούσαν να γίνουν στο μέλλον. Μια λειτουργία που θα μπορούσε να υλοποιηθεί είναι η ανάρτηση του συστήματος σε ιστοσελίδα ώστε να λειτουργεί διαδικτυακά. Επίσης θα ήταν χρήσιμο να προστεθούν επιπλέον επιλογές αναζήτησης και ταξινόμησης κατά την προβολή των βιογραφικών, όπως η αναζήτηση με βάση τις ώρες μη διαθεσιμότητας, κ.τ.λ. Μια επίσης σημαντική επέκταση θα ήταν η αυτόματη αποστολή κωδικού και όνομα χρήστη στους υποψηφίους που κάνουν αίτηση για εγγραφή, χωρίς να υπάρχει η μεσολάβηση της γραμματείας.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Γερογιάννης, Β., Κακαρόντζας Γ., Καμέας, Α., Σταμέλος, Γ., Φιτσιλής, Π. (2006). Αντικειμενοστρεφής Ανάπτυξη Λογισμικού με τη UML. Αθήνα: Κλειδάριθμος.

Πρέντζα, Α. (n.d.). Ανάλυση και Σχεδιασμός Συστημάτων [PowerPoint slides]. Ανακτήθηκε από: <http://www.ds.unipi.gr/systems-analysis-and-design/>

Kaklamanos, D. (2005). Distributed Applications & E-commerce. Available from: <http://users.uom.gr/~kaklaman/book/index.htm>

Larman, C. (2002). Applying UML and Patterns (2nd ed.). Upper Saddle River: Prentice Hall.

Lawrence, S. (2001). Software Engineering: Theory and Practice (2nd ed.). Pearson Education Inc.

Java. (n.d.). In Wikipedia, the Free Encyclopedia. Retrieved from: <http://el.wikipedia.org/wiki/Java>

Java DB. (n.d.). Retrieved from the Oracle Java SE Technologies wiki: <http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136101.html>

Java Database Connectivity. (n.d.). In Wikipedia, the Free Encyclopedia. Retrieved from: http://en.wikipedia.org/wiki/Java_Database_Connectivity

JDBC. (n.d.). In Wikipedia, the Free Encyclopedia. Retrieved from: <http://el.wikipedia.org/wiki/JDBC>

NetBeans. (n.d.). In Wikipedia, the Free Encyclopedia. Retrieved from: <http://en.wikipedia.org/wiki/NetBeans>

Object Oriented Analysis and Design. (n.d.). In Wikipedia, the Free Encyclopedia. Retrieved from: http://en.wikipedia.org/wiki/Object-oriented_analysis_and_design

ΓΛΩΣΣΑΡΙΟ

Όρος	Επεξήγηση
CV	Βιογραφικό Σημείωμα
Java DataBase Connectivity	Συνδετικότητα Βάσης Δεδομένων JAVA
Object oriented Analysis and Design	Αντικειμενοστρεφής Ανάλυση και Σχεδίαση
Unified Modeling Language	Ενοποιημένη Γλώσσα Μοντελοποίησης
Unified Process	Ενοποιημένη Προσέγγιση

ΟΔΗΓΟΣ ΧΡΗΣΗΣ ΛΟΓΙΣΜΙΚΟΥ

ΥΠΟΨΗΦΙΟΣ

Ο υποψήφιος για να εγγραφεί στη βάση πρέπει να κάνει αίτηση στη γραμματεία και στη συνέχεια θα πρέπει να περιμένει ένα email από το λογαριασμό xxx@teithe.gr το οποίο θα περιλαμβάνει το user name και το password εισαγωγής στην εφαρμογή.

Ο υποψήφιος με αυτά τα στοιχεία μπορεί να εισέλθει στο σύστημα και μπορεί να καταχωρήσει το βιογραφικό του ή να το ενημερώσει. Στην καταχώρηση βιογραφικού πρέπει να συμπληρώσει τα πεδία της ενότητας Προσωπικά στοιχεία, Εκπαίδευση, Δημοσιεύσεις, Προϋπηρεσία και Ίδρυση Εταιρίας.

ΕΠΙΤΡΟΠΗ

Η επιτροπή έχει δυο επιλογές αναζήτησης. Με βάση των προσωπικών και λοιπών στοιχείων. Στη λίστα εμφανίζονται τα στοιχεία όλων των υποψηφίων που αντιστοιχούν στα κριτήρια. Επίσης για να διαγραφεί κάποιος υποψήφιος η επιτροπή πρέπει να τον διαλέξει από τη λίστα.

ΓΡΑΜΜΑΤΕΑΣ

Η γραμματεία για να στείλει τους κωδικούς στους υποψήφιους θα πρέπει να επιλέξει Εγγραφή Υποψηφίων. Στη συνέχεια εμφανίζεται το panel της εγγραφής χρήστης για την καταχώρηση των στοιχείων του υποψηφίου. Η γραμματεία καταχωρεί τα ανάλογα στοιχεία και τέλος επιλέγει εγγραφή για να αποσταλούν οι κωδικοί στα e-mails των υποψηφίων.