



ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
ΣΧΟΛΗ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Τμήμα Πληροφορικής και Τηλεπικοινωνιών

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΜΗΧΑΝΟΓΡΑΦΗΣΗ ΙΑΤΡΕΙΟΥ

ΜΕ ΤΗ ΒΟΗΘΕΙΑ ΤΗΣ JAVA

Ιωάννης Βαλάντιος Κολόκας

Επιβλέπων: Ευριπίδης Γλαβάς

*Καθηγητής Τμήματος Πληροφορικής & Τηλεπικοινωνιών,
Σχολή Πληροφορικής & Τηλεπικοινωνιών, Πανεπιστήμιο Ιωαννίνων*

Ιωάννινα, Ιούνιος 2025

ΜΗΧΑΝΟΓΡΑΦΗΣΗ ΙΑΤΡΕΙΟΥ

ΜΕ ΤΗ ΒΟΗΘΕΙΑ ΤΗΣ JAVA

MEDICAL RECORDING
WITH THE HELP OF JAVA

Εγκρίθηκε από τριμελή εξεταστική επιτροπή

Άρτα, 7 Ιουλίου 2025

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

Επιβλέπων καθηγητής

Ευριπίδης Γλαβάς

Καθηγητής Τμήματος Πληροφορικής & Τηλεπικοινωνιών,

Σχολή Πληροφορικής & Τηλεπικοινωνιών,

Πανεπιστήμιο Ιωαννίνων

Μέλος επιτροπής

Νικόλαος Γιαννακέας

Αναπληρωτής Καθηγητής, Τμήμα Πληροφορικής & Τηλεπικοινωνιών,

Σχολή Πληροφορικής & Τηλεπικοινωνιών, Πανεπιστήμιο Ιωαννίνων

Μέλος επιτροπής

Αλέξανδρος Τζάλλας

Αναπληρωτής Καθηγητής, Τμήμα Πληροφορικής & Τηλεπικοινωνιών,

Σχολή Πληροφορικής & Τηλεπικοινωνιών, Πανεπιστήμιο Ιωαννίνων

© Κολόκας, Ιωάννης Βαλάντιος, 2025.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Δήλωση μη λογοκλοπής

Δηλώνω υπεύθυνα και γνωρίζοντας τις κυρώσεις του Ν. 2121/1993 περί Πνευματικής Ιδιοκτησίας, ότι η παρούσα μεταπτυχιακή εργασία είναι εξ ολοκλήρου αποτέλεσμα δικής μου ερευνητικής εργασίας, δεν αποτελεί προϊόν αντιγραφής ούτε προέρχεται από ανάθεση σε τρίτους. Όλες οι πηγές που χρησιμοποιήθηκαν (κάθε είδους, μορφής και προέλευσης) για τη συγγραφή της περιλαμβάνονται στη βιβλιογραφία.

Κολόκας, Ιωάννης Βαλάντιος

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να εκφράσω τις ειλικρινείς μου ευχαριστίες στον επιβλέποντα Καθηγητή μου κ. Ευριπίδη Γλαβά για την πολύτιμη καθοδήγηση, τη διαρκή υποστήριξη και τις χρήσιμες παρατηρήσεις του σ' όλη τη διάρκεια της εκπόνησης της παρούσας πτυχιακής εργασίας.

Επίσης, ευχαριστώ θερμά όλους τους Καθηγητές του Τμήματός Πληροφορικής & Τηλεπικοινωνιών του Πανεπιστημίου Ιωαννίνων για τις γνώσεις και τα εφόδια που μου μεταλαμπάδευσαν με απλόχερα κατά τη διάρκεια των σπουδών μου. Ακόμη, θα ήθελα να ευχαριστήσω όλο το προσωπικό του Τμήματός μου για την αμέριστη βοήθειά τους.

Τέλος, οφείλω ιδιαίτερες ευχαριστίες στην οικογένειά μου τόσο για την υλική στήριξη, αλλά κυρίως για την ηθική υποστήριξή, τη συμπαράσταση και την ενθάρρυνση που μου πρόσφερε όλη την αγάπη της, γεγονός που συνέβαλε καθοριστικά στην ολοκλήρωση αυτού του στόχου.

Αφιερώνω αυτή την προσπάθειά μου στην αγαπημένη μου μητέρα.

ΠΕΡΙΛΗΨΗ

Η εξέλιξη της Πληροφορικής έχει επηρεάσει σημαντικά όλους τους τομείς της κοινωνίας, τόσο της καθημερινής ζωής των πολιτών, όσο και τις επαγγελματικές τους δραστηριότητες. Η μηχανογράφηση μίας επιχείρησης αποτελεί πλέον απολύτως απαραίτητο βήμα για τον εκσυγχρονισμό των λειτουργιών της επιχείρησης και τη βελτίωση των υπηρεσιών που παρέχει. Η καταγραφή και επεξεργασία δεδομένων με τη βοήθεια των ηλεκτρονικών υπολογιστών επομένως είναι απαραίτητη για όλες τις επιχειρήσεις ανεξαρτήτως του είδους των υπηρεσιών που παρέχουν. Η παρούσα πτυχιακή εργασία παρουσιάζει μία προσπάθεια για την ανάλυση και σχεδίαση ενός πληροφοριακού συστήματος για την μηχανογράφηση ενός Ιατρείου. Στα πλαίσια της εργασίας αυτής, αναπτύχθηκε μία εφαρμογή η οποία καλύπτει τις βασικές ανάγκες για τη μηχανογράφηση ενός σύγχρονου ιατρείου. Η εν λόγω εφαρμογή αναπτύχθηκε με τη βοήθεια της γλώσσας προγραμματισμού Java και χρησιμοποιήθηκαν οι αρχές της Αντικειμενοστρεφούς σχεδίασης. Η διαχείριση του αρχείου των ασθενών (πελατών) και των δεδομένων που απαιτούνται για την καταγραφή των συναλλαγών και την εν γένει λειτουργία του Ιατρείου γίνεται με τη βοήθεια του λογισμικού Apache Derby (Java DB), του λογισμικού δηλαδή που προσφέρει η Java EE για τη διαχείριση συστημάτων Βάσεων Δεδομένων. Η εργασία παρουσιάζει αναλυτικά τη σχεδίαση και την υλοποίηση του συστήματος διαχείρισης ενός Ιατρείου περιγράφοντας όλα τα βήματα που απαιτήθηκαν.

Λέξεις-κλειδιά: Java, Μηχανογράφηση Ιατρείου, Αντικειμενοστρεφής Προγραμματισμός, Διαχείριση Ασθενών, Ανάπτυξη Εφαρμογών

ABSTRACT

The evolution of Information Technology has significantly influenced all sectors of society, affecting both the daily lives of citizens and their professional activities. The computerization of a business has become an essential step toward modernizing its operations and enhancing the quality of services it provides. The recording and processing of data through the use of computers is therefore necessary for all businesses, regardless of the type of services they offer. This undergraduate thesis presents an effort to analyze and design an information system for the computerization of a medical practice. Within the scope of this study, an application was developed to meet the fundamental needs of a modern medical office's digital operations. The application was implemented using the Java programming language, incorporating the principles of Object-Oriented programming. The management of patient (client) records and the data required for the documentation of transactions and the overall operation of the medical practice is handled through Apache Derby (Java DB), a database management system provided by Java EE. The thesis offers a detailed presentation of the design and implementation of the medical practice management system, describing all the necessary development steps.

Keywords: Java, Medical Recording, Object-Oriented Programming, Patient Management, Application Development

ΠΕΡΙΕΧΟΜΕΝΑ

1. Εισαγωγή.....	xiii
1.1. Μηχανογράφηση επιχειρήσεων και οργανισμών	xiii
1.2. Μηχανογράφηση Ιατρείου	xiv
1.3. Στόχοι της εργασίας.....	xvi
2. Ανάλυση και σχεδίαση συστήματος μηχανογράφησης.....	17
2.1. Πληροφοριακά συστήματα επιχειρήσεων.....	17
2.2. Ανάλυση των απαιτήσεων ενός πληροφοριακού συστήματος	18
2.2.1. Λειτουργικές απαιτήσεις πληροφοριακού συστήματος διαχείρισης Ιατρείου	22
2.2.2. Μη λειτουργικές απαιτήσεις πληροφοριακού συστήματος διαχείρισης Ιατρείου	23
2.3. Μεθοδολογία ανάπτυξης λογισμικού	25
2.3.1. Η γλώσσαUML (Unified Modeling Language)	29
2.3.2. Διάγραμμα περιπτώσεων χρήσης.....	31
2.3.3. Σενάρια περιπτώσεων χρήσης	34
3. Ανάπτυξη του συστήματος μηχανογράφησης Ιατρείου	37
3.1. Ανάλυση και υλοποίηση της Βάσης Δεδομένων του συστήματος.....	37
3.2. Προγραμματισμός με τη γλώσσα Java	43
3.3. Java Development Kit (JDK) και Apache NetBeans.....	46
3.4. GlassFish και Java Persistence API (JPA)	49
4. Παρουσίαση της εφαρμογής μηχανογράφησης Ιατρείου	52
4.1. Λειτουργίες της εφαρμογής	52
4.2. Λειτουργίες που αφορούν στους πελάτες-ασθενείς του Ιατρείου	54
4.3. Λειτουργίες που αφορούν στα στοιχεία των Γιατρών	57
4.4. Λειτουργίες που αφορούν στα στοιχεία των Φαρμακευτικών Αγωγών	58
4.5. Λειτουργίες που αφορούν στα Ραντεβού και στις Επισκέψεις των πελατών.....	59
5. Επίλογος και συμπεράσματα	61
BIBΛΙΟΓΡΑΦΙΑ	63

ΚΑΤΑΛΟΓΟΣ ΔΙΑΓΡΑΜΜΑΤΩΝ

Διάγραμμα 2.1 Διαγράμματα που περιλαμβάνονται στη UML.....	31
Διάγραμμα 2.2 Διάγραμμα περιπτώσεων χρήσης.....	33
Διάγραμμα 3.1 Διάγραμμα Οντοτήτων Συσχετίσεων για τη ΒΔ του συστήματος	38

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 2.1 Λειτουργικές και μη λειτουργικές απαιτήσεις	21
Εικόνα 2.2 Το μοντέλο του καταρράκτη	26
Εικόνα 2.3 Μοντέλο της Λειτουργικής Επαύξησης	27
Εικόνα 2.4 Το σπειροειδές μοντέλο.....	28
Εικόνα 2.5 Σενάριο εισαγωγής ραντεβού για νέο ασθενή	35
Εικόνα 2.6 Σενάριο ολοκλήρωσης ενός ραντεβού	35
Εικόνα 2.7 Σενάριο για την εισαγωγή του ιστορικού ενός ασθενή	36
Εικόνα 3.1 Εντολές SQL για τη δημιουργία της Βάσης Δεδομένων	42
Εικόνα 3.2 Δημιουργία περιορισμών που αφορούν τα ξένα κλειδιά της ΒΔ	43
Εικόνα 3.3 Δείκτης TIOBE για τις γλώσσες προγραμματισμού	44
Εικόνα 3.4 Δείκτης Statista για τη δημοφιλή γλώσσων προγραμματισμού	45
Εικόνα 3.5 Δείκτης PYPL ως προς τις αναζητήσεις βοηθημάτων	45
Εικόνα 3.6 Δημοφιλή IDEs για Java.....	47
Εικόνα 3.7 Το περιβάλλον του Apache NetBeans IDE.....	48
Εικόνα 3.8 Διαχείριση Apache Derby μέσα από το NetBeans	50
Εικόνα 3.9 Δημιουργία κλάσεων οντοτήτων (entity classes).....	51
Εικόνα 3.10 Η entity class Patient	51
Εικόνα 4.1 Η κεντρική οθόνη της εφαρμογής	52
Εικόνα 4.2 Καταχώρηση νέου πελάτη	54
Εικόνα 4.3 Προβολή λίστας πελατών	54
Εικόνα 4.4 Εμφάνιση περιγραφής στα εικονίδια	55
Εικόνα 4.5 Προβολή στοιχείων πελάτη	55
Εικόνα 4.6 Προβολή ιστορικού Ασθενή	56
Εικόνα 4.7 Τροποποίηση των στοιχείων ενός πελάτη	56
Εικόνα 4.8 Καταχώρηση νέου γιατρού	57
Εικόνα 4.9 Προβολή λίστας με τα στοιχεία των γιατρών	57
Εικόνα 4.10 Καταχώρηση νέου φαρμάκου	58
Εικόνα 4.11 Πίνακας με τα φάρμακα που έχουν καταχωρηθεί	58

Εικόνα 4.12 Τροποποίηση στοιχείων φαρμάκου.....	59
Εικόνα 4.13 Κλείσιμο νέου ραντεβού.....	60
Εικόνα 4.14 Λίστα με τα κλεισμένα ραντεβού.....	60
Εικόνα 4.15 Επεξεργασία ενός ραντεβού.....	60

1. Εισαγωγή

1.1. Μηχανογράφηση επιχειρήσεων και οργανισμών

Η χρήση των Ηλεκτρονικών Υπολογιστών έχει επηρεάσει όλους τους τομείς της κοινωνίας μας, τόσο τη καθημερινή ζωή όσο και το σύνολο των επαγγελματικών δραστηριοτήτων. Όλες οι σύγχρονες επιχειρήσεις και οι οργανισμοί βασίζονται στη λειτουργία τους στη χρήση των ηλεκτρονικών υπολογιστών. Η μηχανογράφηση μίας επιχείρησης αποτελεί βασική προϋπόθεση πλέον για τον εκσυγχρονισμό μίας επιχείρησης και τη βελτίωση των υπηρεσιών τις οποίες παρέχει. Ο όρος μηχανογράφηση περιλαμβάνει όλες εκείνες τις λειτουργίες που αφορούν την καταγραφή και επεξεργασία των δεδομένων μίας επιχείρησης με τη βοήθεια των ηλεκτρονικών υπολογιστών. Η σωστά σχεδιασμένη μηχανογράφηση δίνει τη δυνατότητα της άμεσης και έγκυρης πληροφόρησης και διευκολύνει την διεκπεραίωση των υπηρεσιών που προσφέρει μία επιχείρηση, ελαχιστοποιώντας ταυτόχρονα τα σφάλματα τα οποία ενδέχεται να συμβούν και που μπορεί να οφείλονται στον ανθρώπινο παράγοντα.

Κατά τη μηχανογράφηση μίας οποιασδήποτε επιχείρησης θα πρέπει να γίνει προσεκτική επιλογή και εγκατάσταση του κατάλληλου λογισμικού (software). Υπάρχουν διάφορα έτοιμα πακέτα λογισμικού τα οποία υπόσχονται την κάλυψη όλων των αναγκών και τη σωστή μηχανογράφηση μίας επιχείρησης. Παρόλα αυτά, πολλές φορές τα πακέτα αυτά είτε δεν ταιριάζουν επακριβώς με τις απαιτήσεις της επιχείρησης ή συχνά προσφέρουν πολλές επιπλέον δυνατότητες τις οποίες δεν τις χρειάζεται στην πραγματικότητα η επιχείρηση. Επομένως, είναι συχνά απαραίτητη η εξειδικευμένη ανάλυση των αναγκών της επιχείρησης, έτσι ώστε να καλύπτει οι απαιτήσεις της και μόνο αυτές.

Κάποιες από τις βασικότερες εργασίες που θα πρέπει να εκτελεστούν κατά την προσπάθεια μηχανογράφησης μίας επιχείρησης, είναι οι πιο κάτω:

- Ορισμός των στόχων του συστήματος που θα αναπτυχθεί και ανάλυση των αναγκών της επιχείρησης.
- Επιλογή του απαραίτητου υλικού που θα χρειαστεί, όπως είναι για παράδειγμα, οι υπολογιστές, οι περιφερειακές μονάδες, η σύνδεση με το Διαδίκτυο κ.τ.λ.

- Μοντελοποίηση της λειτουργίας της επιχείρησης και της ροής των δεδομένων που διακινούνται σε αυτή με τη βοήθεια εξειδικευμένων εργαλείων και τεχνικών.
- Σχεδίαση και υλοποίηση της βάσης δεδομένων όπου θα αποθηκεύονται τα δεδομένα που αφορούν τις συναλλαγές που πραγματοποιεί η επιχείρηση με τους πελάτες της ή με άλλες επιχειρήσεις και οργανισμούς.
- Επιλογή ή ανάπτυξη από την αρχή του κατάλληλου λογισμικού (software) το οποίο θα αναλάβει να ικανοποιήσει τις ανάγκες της επιχείρησης και να προσφέρει τις υπηρεσίες της στους πελάτες της, όπως και σε όλους τους εμπλεκομένους.
- Έλεγχος και πιστοποίηση της σωστής λειτουργία του συστήματος.
- Εκπαίδευση του προσωπικού της επιχείρησης, έτσι ώστε να μπορεί να χειριστεί με επιτυχία το λογισμικό που εγκαταστάθηκε.
- Τεχνική υποστήριξη έτσι ώστε να μπορούν να επιλυθούν τυχόν προβλήματα που θα προκύψουν μελλοντικά ή να υλοποιηθούν τυχόν αλλαγές, τροποποιήσεις και αναβαθμίσεις του συστήματος.

Η παρούσα εργασία παρουσιάζει μία προσπάθεια ανάλυσης και σχεδίασης ενός συστήματος για την μηχανογράφηση ενός Ιατρείου, υλοποιώντας εκ του μηδενός μία εφαρμογή σε γλώσσα προγραμματισμού Java, έτσι ώστε να καλύψει τις επιθυμητές λειτουργικές ανάγκες.

1.2. Μηχανογράφηση Ιατρείου

Η μηχανογράφηση ενός ιατρείου θεωρείται μία απολύτως απαραίτητη προϋπόθεση έτσι ώστε να είναι σε θέση το Ιατρείο να προσφέρει ποιοτικές υπηρεσίες στους πελάτες του. Ένα Ιατρείο παρουσιάζει κάποιες απολύτως εξειδικευμένες ανάγκες σε σχέση με άλλου είδους επιχειρήσεις και κάποιες από αυτές τις ανάγκες θα παρουσιαστούν στη συνέχεια. Σε ένα Ιατρείο προσφέρονται υπηρεσίες και εκτελούνται συναλλαγές για τις οποίες θα πρέπει να καταγραφούν πληροφορίες οι οποίες αφορούν ασθενείς, γιατρούς, ραντεβού, επισκέψεις, φάρμακα, θεραπείες κ.λ.π. Η διαχείριση όλου αυτού του όγκου πληροφοριών και ο έλεγχος των σχέσεων που υπάρχουν μεταξύ τους, αποτελεί μία αρκετά πολύπλοκη διαδικασία. Απαιτείται επομένως ιδιαίτερα μεγάλη προσοχή κατά την ανάλυση του

συστήματος και κατά τον προσδιορισμό των απαιτήσεων, όπως και κατά τη φάση της υλοποίησης του συστήματος.

Μερικές από τις ανάγκες που θα πρέπει να καλυφθούν κατά τη μηχανογράφηση ενός ιατρείου είναι οι πιο κάτω:

- Διαχείριση των στοιχείων των ασθενών (πελατών) οι οποίοι επισκέπτονται το ιατρείο.
- Διαχείριση του ιατρικού φακέλου και του ιστορικού του κάθε ασθενή, έτσι ώστε να υπάρχει εύκολη και ολοκληρωμένη ενημέρωση του γιατρού.
- Καταγραφή των γιατρών που συνεργάζονται με το ιατρείο και διαχείριση των δεδομένων που τους αφορούν.
- Διαχείριση των δεδομένων που αφορούν τα ραντεβού που κλείνουν οι ασθενείς με κάποιον γιατρό.
- Καταγραφή των επισκέψεων που πραγματοποιούν οι ασθενείς στο Ιατρείο μετά από κάποιο ραντεβού που έχουν κλείσει.
- Αποθήκευση της θεραπείας που προτείνεται στον κάθε ασθενή από τον γιατρό που τον έχει εξετάσει.
- Καταγραφή της φαρμακευτικής αγωγής που προτείνεται σε έναν ασθενή που πραγματοποίησε επίσκεψη στο Ιατρείο.

Υπάρχουν διαθέσιμες διάφορες εμπορικές εφαρμογές οι οποίες προσφέρουν μερική ή πλήρη μηχανογράφηση ενός ιατρείου και καλύπτουν όλες τις πιο πάνω απαιτήσεις και πολύ περισσότερες. Κάποιες από αυτές τις εφαρμογές εκτελούνται τοπικά στον υπολογιστή, ενώ άλλες έχουν μόνιμα αποθηκευμένα τα δεδομένα τους στο Διαδίκτυο (cloud computing) και οι χρήστες, όπως το προσωπικό του Ιατρείου, μπορούν να έχουν πρόσβαση σε αυτά ακόμη και μέσω κάποιας έξυπνης συσκευής. Παρόλη την ύπαρξη αρκετών τέτοιων εμπορικών ιατρικών εφαρμογών, η παρούσα εργασία κάνει μία προσπάθεια περιγραφής της διαδικασίας ανάλυσης, σχεδίασης και υλοποίησης ενός τέτοιου συστήματος μηχανογράφησης και αναπτύσσει μία εφαρμογή η οποία υλοποιεί κάποιες από τις βασικές λειτουργίες μηχανογράφησης ενός Ιατρείου.

1.3. Στόχοι της εργασίας

Ο σκοπός της παρούσας εργασίας είναι να παρουσιάσει τις απαιτήσεις που υπάρχουν κατά την μηχανογράφηση ενός ιατρείου και να περιγράψει τα βήματα που πρέπει να ακολουθηθούν κατά τη διαδικασία της ανάπτυξης και δημιουργίας του λογισμικού για ένα τέτοιο σύστημα. Σκοπός της εργασίας δεν είναι φυσικά ο ανταγωνισμός με αντίστοιχες εμπορικές εφαρμογές, αλλά να παρουσιαστούν ο τρόπος και οι τεχνικές που μπορούν να χρησιμοποιηθούν για την ανάπτυξη ενός συστήματος που καλύπτει τις βασικές ανάγκες λειτουργίας ενός ιατρείου, χωρίς όμως να παρέχει δυνατότητες προχωρημένων λειτουργιών, όπως π.χ. απομακρυσμένη διαχείριση, αποθήκευση δεδομένων στο Διαδίκτυο, σύνδεση με ιατρικά εργαλεία και συσκευές κ.τ.λ.

Στο κεφάλαιο 2 παρουσιάζεται η ανάλυση και η σχεδίαση ενός συστήματος μηχανογράφησης ιατρείου. Στο κεφάλαιο 3 περιγράφεται η υλοποίηση του συστήματος, η ανάπτυξη του λογισμικού και οι τεχνολογίες που χρησιμοποιήθηκαν. Τέλος, στο κεφάλαιο 4 παρουσιάζεται με εικόνες (screenshots) η εφαρμογή που αναπτύχθηκε και ο τρόπος με τον οποίο μπορεί να χρησιμοποιηθεί από τους χρήστες του ιατρείου.

2. Ανάλυση και σχεδίαση συστήματος μηχανογράφησης

2.1. Πληροφοριακά συστήματα επιχειρήσεων

Με τον όρο «πληροφοριακό σύστημα» αναφερόμαστε στο σύνολο των στοιχείων εκείνων τα οποία είναι υπεύθυνα που συλλέγουν, επεξεργάζονται, αποθηκεύουν και ανακτούν πληροφορίες που αφορούν τη λειτουργία μίας επιχείρησης ή ενός οργανισμού και των υπηρεσιών που προσφέρει. Ένα πληροφοριακό σύστημα πρέπει να στοχεύει στην υποστήριξη των λειτουργιών της επιχείρησης ή του οργανισμού ή ακόμη και να βοηθήσει στη λήψη των αποφάσεων που θα πρέπει να ληφθούν και κατά κανόνα περιλαμβάνει την απαραίτητη τεχνολογία που απαιτείται για τη λειτουργία του, όπως υλικό, λογισμικό, διαδικασίες και δεδομένα, αλλά και το αντίστοιχο εξειδικευμένο ανθρώπινο δυναμικό.

Ένα πληροφοριακό σύστημα μπορεί να θεωρηθεί ως ένα ειδικού τύπου σύστημα εργασίας. Ως σύστημα εργασίας θεωρείται ένα σύστημα μέσα στο οποίο άνθρωποι οι υπολογιστές εκτελούν κάποιες συγκεκριμένες διαδικασίες και δραστηριότητες και χρησιμοποιούν τους δεδομένους πόρους του συστήματος με σκοπό την παραγωγή συγκεκριμένων προϊόντων και υπηρεσιών που προσφέρονται στους πελάτες του συστήματος. Σύμφωνα με αυτή τη θεώρηση, ένα πληροφοριακό σύστημα είναι ένα σύστημα εργασίας, οι δραστηριότητες του οποίου είναι αφιερωμένες στη συλλογή, αποθήκευση, τροποποίηση, μετάδοση, ανάκτηση και εμφάνιση πληροφοριών. Επομένως, ένα πληροφοριακό σύστημα αλληλεπιδρά τόσο με υπάρχοντα συστήματα δεδομένων, όσο και με συστήματα πραγματοποίησης δραστηριοτήτων και αποτελεί υπό την έννοια αυτή μια μορφή συστήματος επικοινωνίας όπου τα δεδομένα αντιπροσωπεύονται και επεξεργάζονται ως μια μορφή κοινωνικής μνήμης.

Υπάρχουν διάφορα είδη πληροφοριακών συστημάτων τα οποία μπορούν να χρησιμοποιηθούν για διαφορετικούς σκοπούς σε μια επιχείρηση ή οργανισμό, ανάλογα με τις υπάρχουσες ανάγκες. Τέτοια πληροφοριακά συστήματα μπορεί να χρησιμοποιούν διάφορες σύγχρονες τεχνολογίες και να επικεντρώνουν τη λειτουργία τους σε διάφορες λειτουργικές ανάγκες που κυμαίνονται από την απλή επεξεργασία δεδομένων μέχρι τη στρατηγική σχεδίαση και τη λήψη αποφάσεων. Μερικά από τα βασικότερα είδη πληροφοριακών συστημάτων είναι τα πιο κάτω:

- **Συστήματα Επεξεργασίας Συναλλαγών (Transaction Processing Systems):** Είναι συστήματα τα οποία έχουν ως σκοπό να αυτοματοποιήσουν την επεξεργασία των δεδομένων τα οποία αφορούν τις καθημερινές συναλλαγές μίας επιχείρησης.
- **Πληροφοριακά Συστήματα Διοίκησης (Management Information Systems):** Τα συστήματα αυτά προσφέρουν πληροφορίες για την υποστήριξη της διαχείρισης των αποφάσεων που πρέπει να ληφθούν.
- **Συστήματα Υποστήριξης Αποφάσεων (Decision Support Systems):** Είναι πληροφοριακά συστήματα τα οποία συνδυάζουν δεδομένα και χρησιμοποιούν προηγμένα αναλυτικά μοντέλα και εργαλεία ανάλυσης για την υποστήριξη της λήψης αποφάσεων.
- **Συστήματα Διαχείρισης Γνώσης (Knowledge Management Systems):** Τα συστήματα αυτά ενθαρρύνουν την ανταλλαγή γνώσης και την καινοτομία μέσω της διαχείρισης των δεδομένων που αποθηκεύονται για τη λειτουργία της επιχείρησης.
- **Έμπειρα συστήματα (Expert systems).** Είναι συστήματα τα οποία στηρίζονται στην προσπάθεια άντλησης γνώσης που προέρχεται από την εμπειρία και τη δεξιοτεχνία κάποιων εμπειρογνομόνων ενός γνωστικού χώρου και έχουν σκοπό την εξαγωγή συμπερασμάτων και κανόνων, προσφέροντας μάλιστα τη δυνατότητα της αιτιολόγησης των κανόνων που εξάγονται.
- **Συστήματα Πραγματικού Χρόνου (Real-time systems).** Πρόκειται για πληροφοριακά συστήματα τα οποία λαμβάνουν συνεχώς μεταβαλλόμενα δεδομένα από κάποιες εξωτερικές πηγές και είναι σε θέση να τα επεξεργάζονται σε πολύ σύντομο χρονικό διάστημα.

Σκοπός της παρούσας εργασίας είναι η ανάπτυξη ενός απλού συστήματος Επεξεργασίας Συναλλαγών για ένα ιατρείο. Ενόσ συστήματος δηλαδή το οποίο θα αυτοματοποιεί κάποιες από τις καθημερινές συναλλαγές και την επεξεργασία των δεδομένων που αφορούν ένα ιατρείο.

2.2. Ανάλυση των απαιτήσεων ενός πληροφοριακού συστήματος

Η ανάλυση των απαιτήσεων ενός πληροφοριακού συστήματος αποτελεί μια κρίσιμη διαδικασία που θα πρέπει να γίνει στα αρχικά στάδια της σχεδίασης του συστήματος και

αφορά τη διερεύνηση και τον ορισμό των λειτουργικών και μη λειτουργικών απαιτήσεων του συστήματος. Στα πλαίσια της ανάλυσης των απαιτήσεων, θα πρέπει να πραγματοποιηθεί αναγνώριση των στόχων, των λειτουργιών και των περιορισμών του συστήματος, όπως και των αναγκών των χρηστών και γενικότερα όλων των εμπλεκομένων. Με άλλα λόγια, η ανάλυση απαιτήσεων αποτελεί μία βασική και πολύ σημαντική διαδικασία η οποία βοηθά στον προσδιορισμό της αλληλεπίδρασης του συστήματος με τους χρήστες και με τα άλλα συνεργαζόμενα συστήματα.

Η φάση της ανάλυσης απαιτήσεων πολλές φορές περιλαμβάνει τεχνικές όπως συνεντεύξεις, ερωτηματολόγια και παρατηρήσεις και μπορεί να μοντελοποιηθεί με τη βοήθεια διαγραμμάτων και σεναρίων χρήσης, τα οποία έχουν σκοπό την απεικόνιση των απαιτήσεων και των διαδικασιών του συστήματος με γραφικό τρόπο και με την απαιτούμενη σαφήνεια και παραστατικότητα.

Με τον όρο «απαίτηση» αναφερόμαστε σε μια δήλωση η οποία καταγράφεται και περιγράφει το τι πρέπει να κάνει το σύστημα κατά τη λειτουργία του. Αρχικά θα πρέπει οι απαιτήσεις να καταγραφούν από τη σκοπιά της επιχείρησης, αποφεύγοντας όσο είναι δυνατόν την συμπερίληψη τεχνικών όρων. Στη συνέχεια προστίθενται οι διάφορες τεχνικές απαιτήσεις, οι οποίες συχνά αναφέρονται και ως “απαιτήσεις συστήματος”. Η καταγραφή των απαιτήσεων αυτών χρησιμοποιείται ως μία μορφή συμβολαίου που γίνεται ανάμεσα στην επιχείρηση ή στον οργανισμό και του φορέα που αναλαμβάνει να υλοποιήσει το πληροφοριακό σύστημα. Η καταγραφή αυτή είναι ένα από τα πιο κρίσιμα στάδια της συνολικής διαδικασίας και θα πρέπει η καταγραφή των απαιτήσεων να είναι όσο το δυνατόν πιο πλήρης και σαφής. Σε περίπτωση ύπαρξης ελλείψεων και ασαφειών είναι πολύ πιθανό να προκύψουν προβλήματα στη συνέχεια, κατά τη φάση δηλαδή της ανάπτυξης και λειτουργίας του συστήματος.

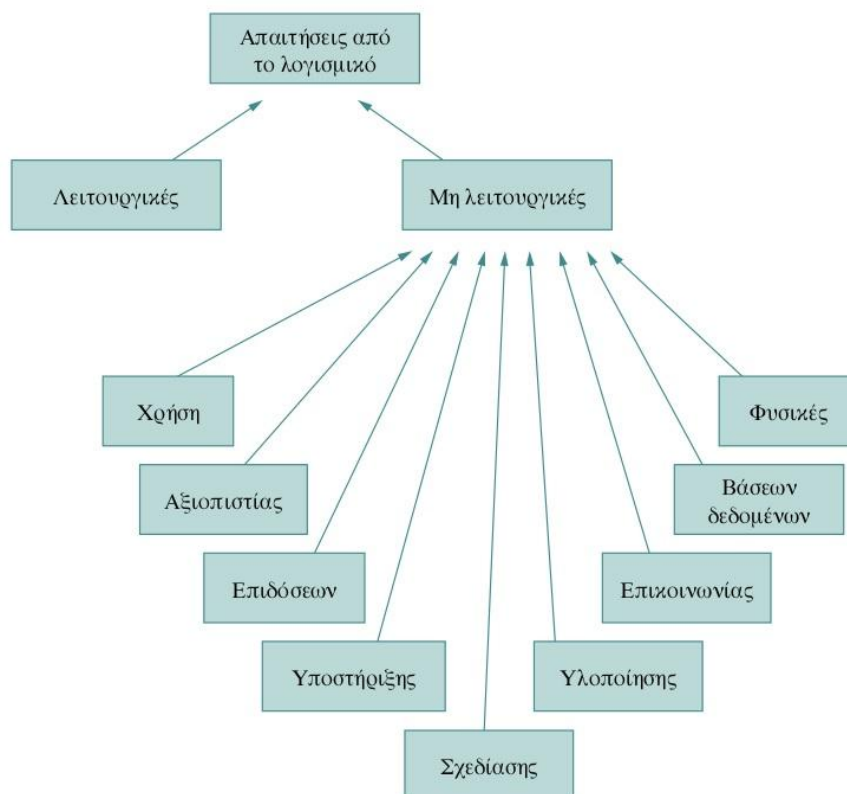
Κατά την καταγραφή των απαιτήσεων ενός πληροφοριακού συστήματος, αναγνωρίζονται απαιτήσεις οι οποίες ανήκουν σε μία από τις πιο κάτω γενικές κατηγορίες:

- **Λειτουργικές απαιτήσεις.** Οι λειτουργικές απαιτήσεις περιγράφουν με ακρίβεια και σαφήνεια τι θέλουμε να κάνει το σύστημα, δηλαδή ποιες ακριβώς λειτουργίες θα πρέπει να επιτελεί, χωρίς φυσικά να αναφέρουμε τον τρόπο με τον οποίο θα επιτευχθεί αυτό.

- **Μη λειτουργικές απαιτήσεις.** Οι μη λειτουργικές απαιτήσεις περιγράφουν κάποια από τα επιπλέον επιθυμητά χαρακτηριστικά του συστήματος τα οποία όμως δεν αναφέρονται σε αυτές καθαυτές τις λειτουργίες του. Τέτοια χαρακτηριστικά μπορεί να αφορούν την απόδοση του συστήματος, την ασφάλειά του, την φιλικότητα προς τον χρήστη κ.τ.λ. Στις μη λειτουργικές απαιτήσεις δηλαδή περιλαμβάνονται κυρίως τεχνικά θέματα σε αντίθεση με τις λειτουργικές απαιτήσεις που αφορούν τις λειτουργίες που εκτελεί το σύστημα. Οι μη λειτουργικές απαιτήσεις ενός συστήματος, ανάλογα με τα χαρακτηριστικά που περιγράφουν, μπορούν να ταξινομηθούν σε μία από τις πιο κάτω κατηγορίες:
- **Απαιτήσεις χρήσης.** Αφορούν τα χαρακτηριστικά της χρήσης του συστήματος και την διεπαφή με το χρήστη (user interface), όπως και το υλικό τεκμηρίωσης και εκπαίδευσης το οποίο περιγράφει τις υπηρεσίες του συστήματος και τον τρόπο με τον οποίο ο τελικός χρήστης μπορεί να επωφεληθεί από αυτές.
- **Απαιτήσεις αξιοπιστίας.** Καθορίζουν τον τρόπο που συμπεριφέρεται το λογισμικό του συστήματος σε καταστάσεις ενδογενών ή εξωγενών σφαλμάτων και τη διαδικασία της αποκατάστασης του συστήματος μετά από τέτοια σφάλματα.
- **Απαιτήσεις επιδόσεων.** Οι απαιτήσεις αυτές αφορούν τα χαρακτηριστικά του συστήματος σχετικά με το χρόνο εκτέλεσης και απόκρισης του συστήματος και τη χρήση υπολογιστικών πόρων, όπως η μνήμη και οι μονάδες επεξεργασίας.
- **Απαιτήσεις υποστήριξης.** Είναι οι απαιτήσεις οι οποίες περιγράφουν τα επιθυμητά χαρακτηριστικά για τον έλεγχο και τη συντήρηση του λογισμικού του συστήματος.
- **Απαιτήσεις σχεδίασης.** Καθορίζουν τον τρόπο με τον οποίο θα πρέπει να γίνει η σχεδίαση του λογισμικού, όπως και τη μεθοδολογία και τα πρότυπα που θα ακολουθηθούν.
- **Απαιτήσεις υλοποίησης.** Καθορίζουν τον τρόπο με τον οποίο θα πρέπει να γίνει η συγγραφή του πηγαίου κώδικα (source code) του λογισμικού.
- **Απαιτήσεις επικοινωνίας με άλλα συστήματα.** Οι απαιτήσεις αυτές καθορίζουν τα εξωτερικά συστήματα με τα οποία το επικοινωνεί το λογισμικό του πληροφοριακού συστήματος, όπως και τον τρόπο πραγματοποίησης της επικοινωνίας αυτής.

- **Απαιτήσεις Βάσεων Δεδομένων.** Είναι οι απαιτήσεις που αναφέρονται στα δεδομένα που αποθηκεύει και επεξεργάζεται το σύστημα, δηλαδή στις οντότητες που διαχειρίζεται το σύστημα, καθώς και τα γνωρίσματα καθεμιάς από αυτές τις οντότητες.
- **Φυσικές απαιτήσεις.** Εδώ περιλαμβάνονται απαιτήσεις που περιγράφουν τα επιθυμητά φυσικά χαρακτηριστικά του συστήματος, όπως π.χ. τις προδιαγραφές των υπολογιστικών συστημάτων και των δικτυακών υποδομών κ.τ.λ.

Στην Εικόνα 2.1 εμφανίζονται τα διάφορα είδη λειτουργικών και μη λειτουργικών απαιτήσεων. Στις επόμενες παραγράφους θα αναλυθούν οι λειτουργικές και οι μη λειτουργικές απαιτήσεις ενός πληροφοριακού συστήματος για τη διαχείριση των λειτουργιών ενός ιατρείου.



Εικόνα 2.1 Λειτουργικές και μη λειτουργικές απαιτήσεις

2.2.1. Λειτουργικές απαιτήσεις πληροφοριακού συστήματος διαχείρισης Ιατρείου

Ένα πληροφοριακό σύστημα για τη διαχείριση ενός ιατρείου, θα πρέπει να υποστηρίζει τις βασικές λειτουργίες οι οποίες θεωρούνται απαραίτητες για την μηχανογράφηση του. Οι λειτουργικές απαιτήσεις καθορίζουν τις βασικές αυτές λειτουργίες, οι οποίες περιγράφονται στην πιο κάτω λίστα:

- **Διαχείριση δεδομένων που αφορούν γιατρούς.** Η διαχείριση των δεδομένων που αφορούν γιατρούς αναφέρεται στα δεδομένα των γιατρών που συνεργάζονται με το Ιατρείο και μπορεί να είναι διαφορετικών ειδικοτήτων και να παρέχοντας τις υπηρεσίες τους προς τους ασθενείς/πελάτες. Η διαχείριση των δεδομένων αυτών περιλαμβάνει τα προσωπικά στοιχεία ενός γιατρού, αλλά και επιπλέον δεδομένα που αφορούν την ειδικότητά του, την εργασία του κ.τ.λ. Τα δεδομένα αυτά μπορούν να αποθηκευτούν, να ανακτηθούν και να προβληθούν σε μία οθόνη και αν χρειαστεί στη συνέχεια να τροποποιηθούν ή ακόμη και να διαγραφούν.
- **Διαχείριση δεδομένων που αφορούν ασθενείς.** Η διαχείριση των δεδομένων των ασθενών αφορά την διατήρηση στο σύστημα των προσωπικών στοιχείων ενός ασθενή, όπως επίσης και του ιατρικού ιστορικού του ασθενή. Τα δεδομένα αυτά μπορούν να ανακτηθούν, να προβληθούν σε κάποια οθόνη, να τροποποιηθούν ή ακόμη και να διαγραφούν. Το ιστορικό ενός ασθενή αποθηκεύεται στο σύστημα και περιλαμβάνει πληροφορίες που αφορούν ασθένειες και νοσηλείες του παρελθόντος, τους γιατρούς που είχαν παρακολουθήσει τον ασθενή, την θεραπεία που έλαβε κ.τ.λ.
- **Διαχείριση των ραντεβού.** Το λογισμικό του συστήματος θα είναι σε θέση να κλείσει κάποιο ραντεβού ανάμεσα σε έναν γιατρό και έναν πελάτη του ιατρείου. Κατά την καταγραφή ενός ραντεβού, αποθηκεύονται δεδομένα όπως η ημερομηνία και η ώρα του ραντεβού, ο ασθενής που ενδιαφέρεται για το ραντεβού, ο γιατρός που θα αναλάβει την εξέταση κ.τ.λ. Ο χρήστης του λογισμικού, πέρα από την καταχώρηση ενός νέου ραντεβού, θα μπορεί να προβάλει τα μελλοντικά ραντεβού και να τροποποιήσει ή να ακυρώσει ένα ραντεβού.
- **Καταχώρηση δεδομένων που αφορούν επισκέψεις ασθενών.** Κάθε φορά που κάποιος πελάτης του ιατρείου πραγματοποιεί μία επίσκεψη μετά από ραντεβού που έχει κλείσει, το σύστημα είναι σε θέση να αποθηκεύσει στοιχεία που αφορούν τα

συμπτώματα του ασθενή, την θεραπεία που του πρότεινε ο γιατρός, τα φάρμακα που πρέπει να πάρει, την πληρωμή της επίσκεψης κ.τ.λ.

- **Διαχείριση δεδομένων που αφορούν φάρμακα.** Τα δεδομένα αυτά αφορούν στοιχεία για διάφορα φάρμακα και τα συστατικά που περιλαμβάνει.

2.2.2. Μη λειτουργικές απαιτήσεις πληροφοριακού συστήματος διαχείρισης Ιατρείου

Υπάρχουν διάφοροι τύποι μη λειτουργικών απαιτήσεων, όπως περιγράφηκε στις προηγούμενες παραγράφους. Στην περίπτωση της ανάπτυξης ενός συστήματος μηχανογράφησης ενός ιατρείου, θεωρήθηκε σημαντική η υιοθεσία των πιο κάτω μη λειτουργικών απαιτήσεων:

- **Ανεξαρτησία από την πλατφόρμα του υπολογιστικού συστήματος.** Το λογισμικό θα πρέπει να μπορεί να λειτουργήσει σωστά ανεξαρτήτως του λειτουργικού συστήματος που διαθέτει ο υπολογιστικό σύστημα του ιατρείου. Θα πρέπει επίσης να είναι σε θέση να λειτουργεί σωστά ακόμη και μετά από τυχόν αναβαθμίσεις ή τροποποιήσεις στο λειτουργικό σύστημα, λόγω της επιθυμητής λειτουργικής συνέχειας που θα πρέπει οπωσδήποτε να υπάρχει. Για να ικανοποιηθεί αυτή η βασική απαίτηση, επιλέχθηκε η χρήση της γλώσσα προγραμματισμού Java για την ανάπτυξη του λογισμικού. Η Java είναι μία διαδεδομένη γλώσσα προγραμματισμού και η δημοφιλία της ξεκίνησε εξ' αιτίας ακριβώς αυτού του γεγονότος. Εφαρμογές δηλαδή που έχουν προγραμματιστεί σε γλώσσα Java μπορούν να εκτελεστούν με τον ίδιο ακριβώς τρόπο σε οποιοδήποτε λειτουργικό σύστημα και σε όλες τις υπολογιστικές πλατφόρμες.
- **Δυνατότητα αποθήκευσης δεδομένων και εγγύηση της ακεραιότητά τους.** Το λογισμικό θα πρέπει να δίνει τη δυνατότητα της μόνιμης αποθήκευσης των δεδομένων, έτσι ώστε τα δεδομένα αυτά να μπορούν να ανακτηθούν με ασφάλεια και να είναι δυνατή η τροποποίησή τους και η επαναποθήκευσή τους με αξιόπιστο τρόπο. Για να επιτευχθεί αυτή η απαίτηση έχει γίνει χρήση μίας σχεσιακής βάσης δεδομένων, το σύστημα διαχείρισης της οποίας είναι υπεύθυνο για όλες αυτές τις λειτουργίες. Ως σύστημα διαχείρισης βάσης δεδομένων (DBMS) επιλέχθηκε η

ApacheDerby (Java DB). Η Java DB μπορεί να συνεργαστεί με πλήρη αρμονία με τη γλώσσα προγραμματισμού Java, όπως θα δούμε στη συνέχεια.

- **Φιλικότητα προς τον τελικό χρήστη και ευχρηστία.** Η εφαρμογή πρέπει να προσφέρει ένα εύχρηστο και φιλικό προς τον τελικό χρήστη γραφικό περιβάλλον (Graphical User Interface). Με άλλα λόγια, η επικοινωνία με τους τελικούς χρήστες θα πρέπει να είναι εύκολη απλή και ευχάριστη. Η ανάπτυξη της γραφικής διεπαφής του συστήματος έγινε με τη χρήση της ενσωματωμένης βιβλιοθήκης γραφικών Swing της Java. Με τη βοήθεια της βιβλιοθήκης αυτής σχεδιάστηκαν οι απαραίτητες φόρμες και διάλογοι κατά την επικοινωνία με τον χρήστη.
- **Επιδόσεις και απόκριση του συστήματος.** Ο χρόνος που απαιτείται, είτε για την ανάκτηση των πληροφοριών που ζητά ο χρήστης είτε για την αποθήκευση νέων ή τροποποιημένων δεδομένων πρέπει να είναι μικρός και δεν πρέπει να είναι μεγαλύτερος από 1-2 δευτερόλεπτα. Η μη λειτουργική αυτή απαίτηση μπορεί να ικανοποιηθεί με τη χρήση του συστήματος διαχείρισης βάσης δεδομένων Java DB.
- **Έλεγχος για σφάλματα.** Το λογισμικό θα πρέπει να ελέγχει τα δεδομένα που εισάγει ο χρήστης και τις αλλαγές τις οποίες κάνει. Σε περίπτωση λανθασμένων δεδομένων, το σύστημα θα πρέπει να εμφανίζει τα κατάλληλα μηνύματος λάθους και να καθοδηγεί τον χρήστη για τη διόρθωσή τους. Για να επιτευχθεί ο στόχος αυτός, χρησιμοποιήθηκαν διάφορες μέθοδοι που προσφέρουν οι βιβλιοθήκες της Java οι οποίες επιτρέπουν τον έλεγχο της μορφής των δεδομένων που καταχωρούνται, αν π.χ. είναι συμβολοσειρές, ακέραιοι αριθμοί, ημερομηνίες κ.τ.λ. Στις περιπτώσεις ύπαρξης λανθασμένης καταχώρησης, το λογισμικό εμφανίζει τα κατάλληλα μηνύματα μέσω του γραφικού του περιβάλλοντος, έτσι ώστε να γίνουν από τον χρήστη οι απαραίτητες διορθώσεις.

2.3. Μεθοδολογία ανάπτυξης λογισμικού

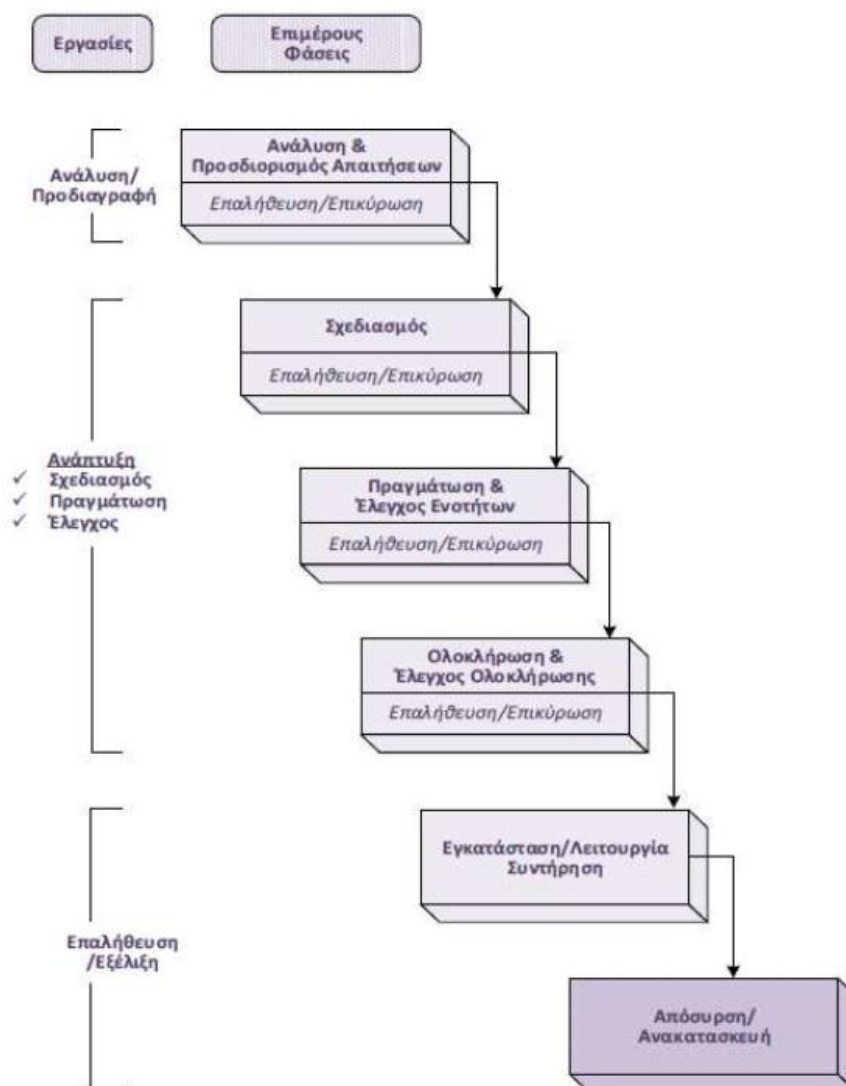
Με τον όρο «μεθοδολογία ανάπτυξης λογισμικού» αναφερόμαστε σε μία σειρά από προτεινόμενα βήματα, τεχνικές και διαδικασίες που ακολουθούνται κατά την ανάπτυξη ενός πληροφοριακού συστήματος. Συχνά, οι διάφορες τυποποιημένες διεργασίες και μεθοδολογίες που χρησιμοποιούνται κατά την ανάπτυξη ενός πληροφοριακού συστήματος αναφέρονται και ως «κύκλος ζωής» του συστήματος. Ο κύκλος ζωής ενός συστήματος αναφέρεται στην ουσία στο χρονικό εκείνο πλαίσιο που εκτείνεται από την εκκίνηση της ανάπτυξης ενός νέου συστήματος μέχρι την τελική απόσυρσή του. Υπάρχουν διάφορες μεθοδολογίες ανάπτυξης λογισμικού οι οποίες εξετάζονται στα πλαίσια της επιστήμης της Τεχνολογίας Λογισμικού και αποτελούν ένα σύνολο από προτάσεις για τις τεχνικές και τις μεθοδολογίες που θα πρέπει να χρησιμοποιηθούν κατά τον κύκλο ζωής ενός συστήματος. Οι βασικότερες μεθοδολογίες ανάπτυξης λογισμικού είναι οι πιο κάτω.

- **Μοντέλο καταρρακτοειδούς ανάπτυξης (καταρράκτης).**

Το μοντέλο αυτό περιλαμβάνει διακριτές μεταξύ τους φάσεις οι οποίες εκτελούνται σειριακά ή μία μετά την άλλη, χωρίς να υπάρχει η δυνατότητα να επιστρέψουμε από κάποια φάση σε κάποια από τις προηγούμενες φάσεις. Είναι διαδεδομένο και θεωρείται εφαρμόσιμο σε περιπτώσεις όπου οι απαιτήσεις είναι από την αρχή γνωστές και σαφείς και δεν μεταβάλλονται στη συνέχεια, αφού το μοντέλο επιβάλλει τη σειριακή εκτέλεση των φάσεων. Επιγραμματικά οι φάσεις που περιλαμβάνει το μοντέλο είναι οι εξής:

- Ανάλυση και προσδιορισμός απαιτήσεων
- Σχεδίαση του συστήματος
- Υλοποίηση του συστήματος
- Έλεγχος του συστήματος
- Λειτουργία και συντήρηση

Οι φάσεις τις οποίες περιλαμβάνει το μοντέλο του καταρράκτη εμφανίζονται επιγραμματικά στην Εικόνα 2.2.

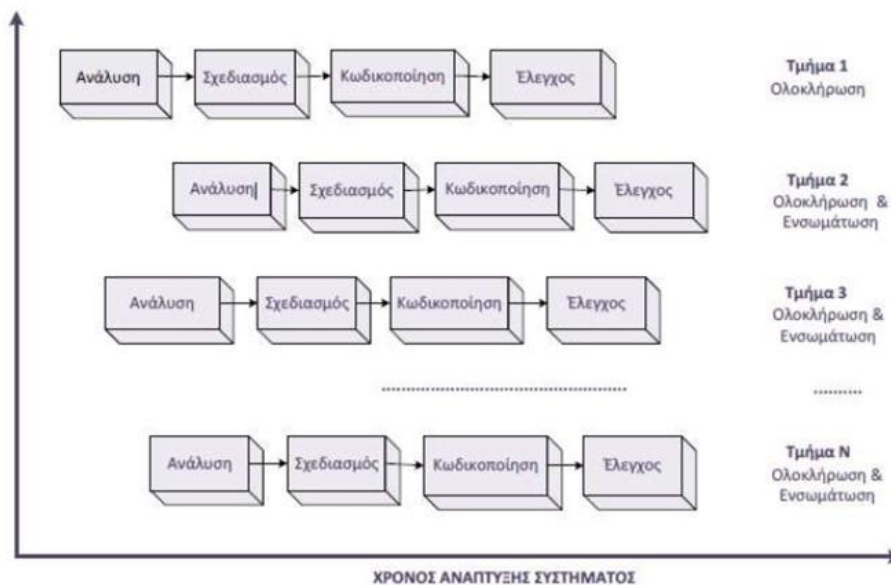


Εικόνα 2.2 Το μοντέλο του καταρράκτη

- **Μοντέλο της λειτουργικής επαύξησης.**

Το μοντέλο αυτό προτείνει την κατάτμηση του συστήματος σε επιμέρους υποσυστήματα που αντιστοιχούν σε λειτουργικές μονάδες του συνολικού συστήματος. Τα επί μέρους τμήματα αναπτύσσονται ανεξάρτητα και παράλληλα το ένα με το άλλο. Όταν ολοκληρωθεί η ανάπτυξη κάποιων τμημάτων, τότε συνενώνονται μεταξύ τους και ενσωματώνονται στο συνολικό σύστημα. Η διαδικασία αυτή συνεχίζεται μέχρι να αναπτυχθούν όλα τα τμήματα και να ενσωματωθούν στο συνολικό σύστημα (Εικόνα 2.3) Με το μοντέλο αυτό μειώνεται σε σχέση με το παραδοσιακό μοντέλο του καταρράκτη ο κίνδυνος αποτυχίας του συστήματος, διαιρώντας το συνολικό σύστημα σε μικρότερα μέρη, παρέχοντας

έτσι ευελιξία σε τυχόν αλλαγές που μπορούν να προκύψουν κατά τον κύκλο ανάπτυξης ενός συστήματος.



Εικόνα 2.3 Μοντέλο της Λειτουργικής Επαύξησης

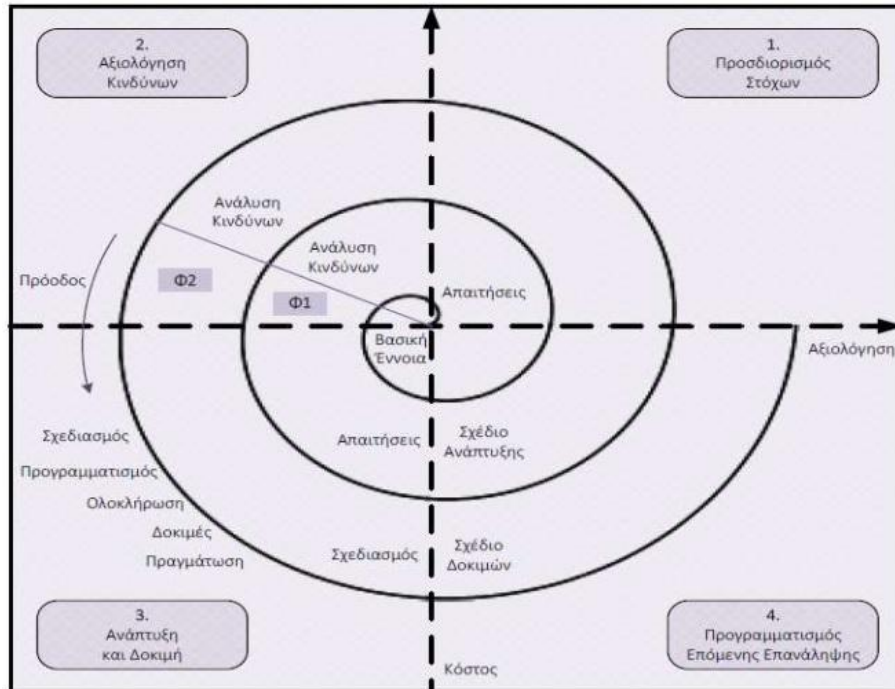
- **Μοντέλο δημιουργίας πρωτοτύπου (πρωτοτυποποίησης).**

Σύμφωνα με το μοντέλο αυτό κατασκευάζονται κάποιες μη ολοκληρωμένες εκδόσεις του συστήματος που πρόκειται να αναπτυχθεί, οι οποίες ονομάζονται πρωτότυπα. Ένα πρωτότυπο μπορεί να περιλαμβάνει κάποια μόνο από τα χαρακτηριστικά και τις λειτουργίες που επιθυμούμε να έχει το τελικό λογισμικό. Είναι μάλιστα πιθανόν ένα πρωτότυπο σε κάποιο στάδιο να είναι πολύ διαφορετικό από το τελικό προϊόν. Κάθε πρωτότυπο αποτελεί σε μικρογραφία μία λειτουργική έκδοση του συστήματος και μπορεί να αλλάζει συνεχώς και να βελτιώνεται έως ότου θεωρηθεί ότι ικανοποιεί όλες τις απαιτήσεις (λειτουργικές και μη λειτουργικές). Η μέθοδος δηλαδή αποτελεί μια επαναληπτική διαδικασία που έχει καθορισμένα βήματα και κατά την οποία κάθε νέα έκδοση του πρωτοτύπου προσεγγίζει με μεγαλύτερη ακρίβεια τις απαιτήσεις των τελικών χρηστών.

- **Σπειροειδές Μοντέλο.**

Αποτελεί μία μεθοδολογία η οποία εφαρμόζει την επαναληπτική ανάπτυξη του συστήματος με την οποία ορίζεται ο κύκλος (επανάληψη) ως η βάση για την προσθήκη αναβαθμισμένων λειτουργικών χαρακτηριστικών του συστήματος. Η

καινοτομία που προσέφερε το μοντέλο αυτό είναι ότι πριν από κάθε κύκλο γίνεται ανάλυση των κινδύνων που αφορούν την επικείμενη εκτέλεση του κύκλου αυτού, όπως φαίνεται στην Εικόνα 2.4.



Εικόνα 2.4 Το σπειροειδές μοντέλο

- **Ευέλικτες (Agile) Μεθοδολογίες Ανάπτυξης Λογισμικού.**

Οι ευέλικτες μέθοδοι προσπαθούν να ικανοποιήσουν τον στόχο της γρήγορης ανάπτυξης εφαρμογών, συνδυάζοντας την αποδοτική χρήση εργαλείων και τεχνικών για παράδοση έργων σε στενές χρονικές κλίμακες. Το 2001, για πρώτη φορά γίνεται η ιδρυτική διακήρυξη Agile Manifesto με την ονομασία «ευέλικτες μέθοδοι» (agile methods) θέτοντας στόχους, όπως: τη δυνατότητα ανταπόκρισης στις αλλαγές, τις παραγωγικότερες πρακτικές και τη λιγότερη γραφειοκρατία, σε αντίθεση με την παραδοσιακή μέθοδο του καταρράκτη. Η ονομασία «ευέλικτες μέθοδοι» αναφέρεται ουσιαστικά στην προσαρμοστικότητα των μεθόδων αυτών στις αλλαγές κατά την ανάπτυξη ενός συστήματος.

Οι μέθοδοι αυτές ενθαρρύνουν την ομαδική εργασία, τη διαρκή προσαρμογή και επιθεώρηση, την αυτό-οργάνωση, την προσαρμογή στις ανάγκες των χρηστών και υιοθετούν ένα σύνολο καλών πρακτικών για τη γρήγορη ανάπτυξη λογισμικού

υψηλής ποιότητας. Υπάρχουν διαφορετικές μεθοδολογίες οι οποίες ανήκουν στην οικογένεια των ευέλικτων μεθόδων όπως:

- Ο Ακραίος Προγραμματισμός (Extreme Programming).
- Η ανάπτυξη με Βάση τις Δοκιμές (Test Driven Development)
- Η ανάπτυξη με Βάση τα Χαρακτηριστικά (Feature Driven Development)
- Η μεθοδολογία ανάπτυξης Δυναμικών Συστημάτων (Dynamic Systems Development Method)
- Η μεθοδολογία DevOps
- Η μεθοδολογία Scrum

2.3.1. Η γλώσσα UML (Unified Modeling Language)

Η γλώσσα UML (Unified Modeling Language - Ενοποιημένη Γλώσσα Μοντελοποίησης) αποτελεί μία πρότυπη γλώσσα που χρησιμοποιείται για την μοντελοποίηση συστημάτων και γενικότερα τη Μηχανική Λογισμικού. Η γλώσσα UML περιλαμβάνει διάφορους τύπους διαγραμμάτων τα οποία μπορούν να βοηθήσουν έναν αναλυτή συστημάτων να περιγράψει με οπτικό τρόπο τη λειτουργία ενός συστήματος λογισμικού, αλλά και να κάνει την τεκμηρίωση των στοιχείων του. Η UML μπορεί να χρησιμοποιηθεί σε όλες τις φάσεις ανάπτυξης ενός συστήματος, ξεκινώντας από την ανάλυση των απαιτήσεων (requirements analysis) και φτάνοντας μέχρι τον έλεγχο (testing) της λειτουργίας του τελικού συστήματος. Στην ουσία η γλώσσα αποτελεί μία συλλογή κάποιων καλών πρακτικών οι οποίες έχουν χρησιμοποιηθεί κατά τη μοντελοποίηση μεσαίων ή μεγάλων συστημάτων.

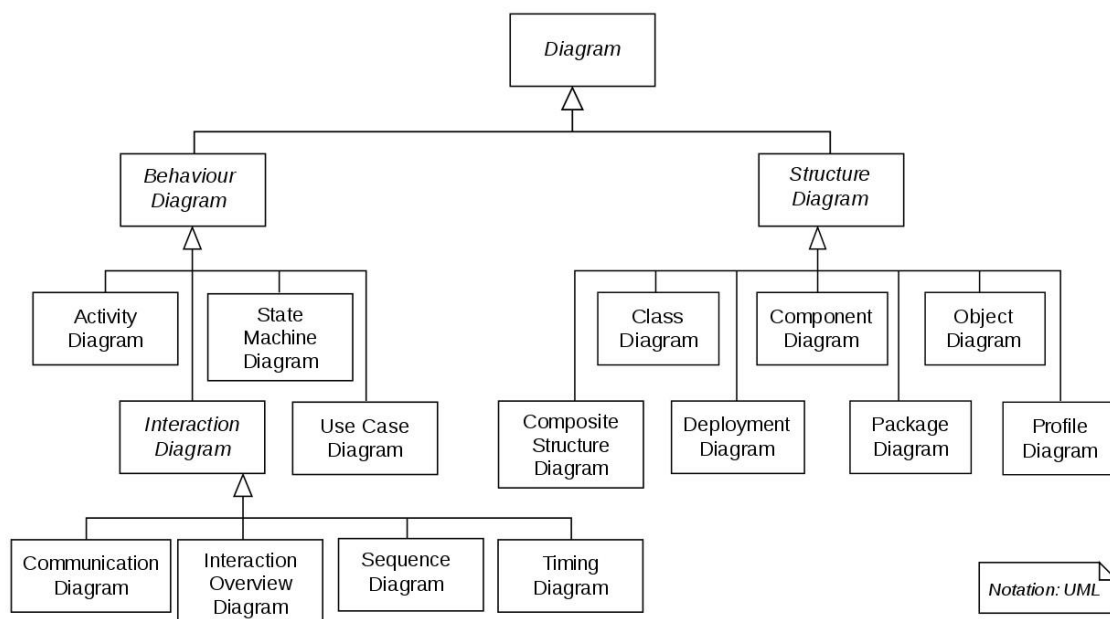
Η γλώσσα UML υποστηρίζει ένα σύνολο διαγραμμάτων, τα οποία δεν είναι υποχρεωτικό να σχεδιαστούν στο σύνολό τους, αλλά η επιλογή των διαγραμμάτων που θα υλοποιηθούν μπορεί να επηρεάσει τον τρόπο προσέγγισης του προβλήματος. Ο βασικός στόχος είναι η εισαγωγή του κατάλληλου βαθμού αφαιρετικότητας (abstraction) έτσι να επιτευχθεί η επιθυμητή εστίαση στα πιο σημαντικά στην κάθε περίπτωση στοιχεία του προβλήματος, αγνοώντας τις λεπτομέρειες οι οποίες δεν μας ενδιαφέρουν σε κάποιο συγκεκριμένο στάδιο. Οι τύποι των διαγραμμάτων που υποστηρίζονται από τη UML επιτρέπουν:

- την αποτύπωση των ορίων ενός συστήματος, όπως και των βασικών λειτουργιών του, με τη βοήθεια των κατάλληλων παραδειγμάτων χρήσης (usecases) και ορίζοντας με σαφήνεια του χειριστές του συστήματος (actors).
- την περιγραφή του τρόπου με τον οποίο αλληλεπιδρούν τα μέρη ενός συστήματος με τη δημιουργία διαγραμμάτων αλληλεπίδρασης.
- την αναπαράσταση της στατικής δομής ενός συστήματος με τη δημιουργία των κατάλληλων διαγραμμάτων κλάσεων.
- τη μοντελοποίηση της συμπεριφοράς των αντικειμένων με τα κατάλληλα διαγράμματα καταστάσεων.
- την περιγραφή της εργασιακής ροής ενός συστήματος με τη βοήθεια διαγραμμάτων δραστηριοτήτων.
- την οπτικοποίηση της αρχιτεκτονικής ενός συστήματος με διαγράμματα συστατικών και ανάπτυξης.
- την επέκταση της λειτουργικότητας με τα κατάλληλα στερεότυπα.

Η έκδοση UML 2.0 της γλώσσας αντικατέστησε το 2005 την έκδοση 1.5, έτσι ώστε να προσθέσει κάποια νέα συστατικά και να βελτιώσει τη γλώσσα. Η επίσημη έκδοση UML 2.5.1 κυκλοφόρησε το Δεκέμβριο του 2017 και υποστηρίζει διάφορους τύπους διαγραμμάτων, όπως:

- Διαγράμματα περιπτώσεων χρήσης (use case diagrams)
- Διαγράμματα κλάσεων (class diagrams)
- Διαγράμματα καταστάσεων (state diagrams)
- Διαγράμματα δραστηριοτήτων (activity diagrams)
- Διαγράμματα ακολουθίας (sequence diagrams)
- Διαγράμματα συνεργασίας (collaboration diagrams)

Οι τύποι που περιλαμβάνονται στη γλώσσα UML και οι κατηγορίες τους, εμφανίζονται στο Διάγραμμα 2.1. **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.**



Διάγραμμα 2.1 Διαγράμματα που περιλαμβάνονται στη UML

Το πλήθος και το είδος των διαγραμμάτων UML που χρησιμοποιούνται, εξαρτάται από τους στόχους που έχουν τεθεί στην κάθε περίπτωση. Στην περίπτωση της ανάπτυξης του λογισμικού μηχανογράφησης ενός Ιατρείου, έγινε χρήση διαγραμμάτων περιπτώσεων χρήσης, διαγραμμάτων κλάσεων, όπως και κάποιων διαγραμμάτων ακολουθίας. Τα διαγράμματα αυτά θα παρουσιαστούν στις επόμενες παραγράφους και έχουν δημιουργηθεί με το πρόγραμμα Visual Paradigm Community Edition version 17.1.

2.3.2. Διάγραμμα περιπτώσεων χρήσης

Η δημιουργία διαγραμμάτων περιπτώσεων χρήσης αποτελεί, σύμφωνα με τη γλώσσα UML, ένα από τα βασικότερα εργαλεία που χρησιμοποιούνται στην ανάλυση ενός συστήματος για την περιγραφή των απαιτήσεών του. Τα διαγράμματα περιπτώσεων χρήσης χρησιμοποιούνται για να μοντελοποιήσουν τον τρόπο λειτουργίας ενός πληροφοριακού συστήματος και τις απαιτήσεις που υπάρχουν σε σχέση με τις υπηρεσίες που προσφέρει. Οι βασικότεροι στόχοι κατά τη δημιουργία ενός διαγράμματος περιπτώσεων χρήσης είναι οι ακόλουθοι:

- Η περιγραφή και ο καθορισμός των λειτουργικών απαιτήσεων του συστήματος

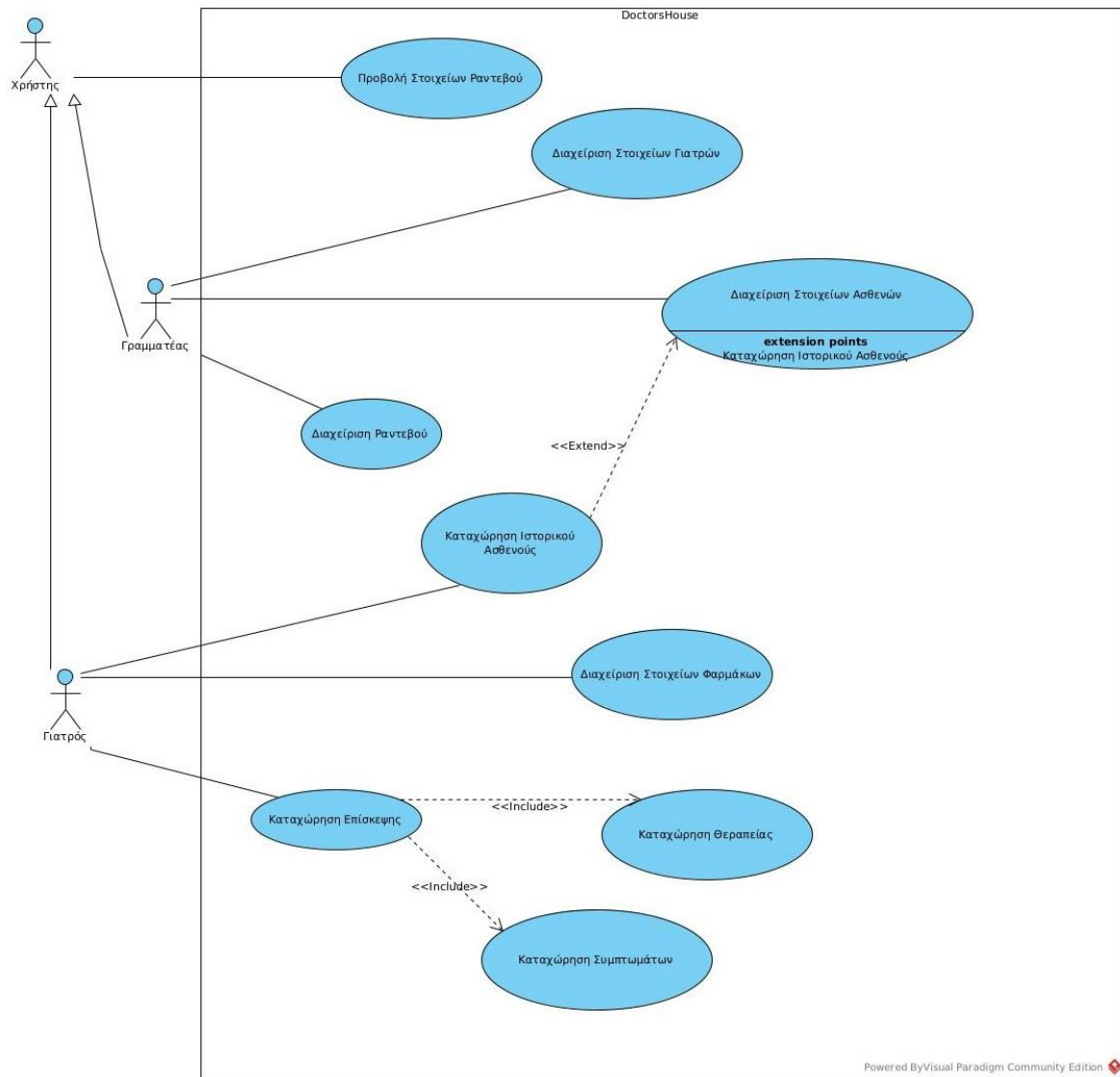
- Η σαφής περιγραφή σχετικά με το τι πρέπει να κάνει το σύστημα, με βάση τις ανάγκες του χρήστη και τις απαιτήσεις που εντοπίστηκαν κατά τη φάση ανάλυσης των απαιτήσεων.
- Η δημιουργία ενός αξιόπιστου σημείου αναφοράς που μπορεί να χρησιμοποιηθεί για την αξιολόγηση του συστήματος μετά την ολοκλήρωση της υλοποίησής του

Ένα διάγραμμα περιπτώσεων χρήσης περιλαμβάνει τις λειτουργίες του συστήματος, όπως και τους χειριστές (actors) που το χρησιμοποιούν για να χρησιμοποιήσουν τις λειτουργίες και τις υπηρεσίες του. Με τον όρο χειριστές του συστήματος αναφερόμαστε στα άτομα, αλλά και τα εξωτερικά συστήματα τα οποία με κάποιον τρόπο επικοινωνούν με το πληροφοριακό σύστημα. Στην περίπτωση της μηχανογράφησης ενός ιατρείου, οι χειριστές του συστήματος μπορεί να είναι οι πιο κάτω:

- ο/η γραμματέας του ιατρείου
- οι γιατροί του ιατρείου
- οποιοδήποτε εξωτερικό σύστημα, όπως π.χ. το σύστημα συνταγογράφησης του Υπουργείου Υγείας, με το οποίο επικοινωνεί το σύστημα,

Ο ασθενής δεν αποτελεί δυνητικό χειριστή του συστήματος, αφού η επικοινωνία του με το σύστημα γίνεται είτε μέσω του γραμματέα είτε μέσω κάποιου γιατρού.

Στο Διάγραμμα 2.2 εμφανίζεται ένα διάγραμμα περιπτώσεων χρήσης για την περίπτωση μηχανογράφησης ενός ιατρείου. Σ' αυτό εμφανίζονται οι βασικότερες από τις λειτουργίες του συστήματος, όπως και οι χειριστές του.



Διάγραμμα 2.2 Διάγραμμα περιπτώσεων χρήσης

Στο διάγραμμα περιπτώσεων χρήσης εμφανίζονται κάποιες σχέσεις τύπου “include” και τύπου “extend”. Οι σχέσεις τύπου “include” χρησιμοποιούνται σε περιπτώσεις όπου μία λειτουργία αποτελεί υποχρεωτικά τμήμα μίας άλλης λειτουργίας. Για παράδειγμα, η περίπτωση χρήσης «Καταχώρηση Συμπτωμάτων» αποτελεί υποχρεωτική λειτουργία της περίπτωσης χρήσης «Καταχώρηση Επίσκεψης». Από την άλλη μεριά, οι σχέσεις τύπου “extend” χρησιμοποιούνται σε κάποιες περιπτώσεις όπου έχουμε μία περίπτωση χρήσης η οποία παρέχει κάποιες επιπλέον λειτουργίες σε μία περίπτωση χρήσης και εκτελείται προαιρετικά. Για παράδειγμα, η περίπτωση χρήσης «Καταχώρηση Ιστορικού Ασθενούς» αποτελεί επέκταση της περίπτωσης χρήσης «Διαχείριση στοιχείων Ασθενών», αφού δεν

καταχωρείται πάντοτε το ιστορικό ενός ασθενούς, αφού μπορεί να μην υπάρχουν διαθέσιμα κάποια στοιχεία για το ιστορικό του.

2.3.3. Σενάρια περιπτώσεων χρήσης

Στην παράγραφο αυτή παρουσιάζονται κάποιες από τις βασικές περιπτώσεις χρήσης που θα πρέπει να υποστηρίζει το λογισμικό μηχανογράφησης ιατρείου. Ένα σενάριο περίπτωση χρήσης παρουσιάζει τα βήματα που θα πρέπει να εκτελέσει ο χειριστής του συστήματος έτσι ώστε να επιτελέσει μία λειτουργία. Τα σενάρια περιπτώσεων χρήσης βοηθούν στην κατανόηση μίας περίπτωσης χρήσης του συστήματος, έτσι ώστε να διευκολυνθεί η υλοποίησή της αντίστοιχης λειτουργίας στα επόμενα στάδια. Κατά την υλοποίηση του λογισμικού στη συνέχεια, θα πρέπει να ενσωματωθούν σε αυτό λειτουργίες οι οποίες θα ικανοποιούν τις προδιαγραφές που έχουν τεθεί κατά το στάδιο αυτό της ανάλυσης.

Μία από τις βασικότερες λειτουργίες του λογισμικού είναι το κλείσιμο ραντεβού ενός ασθενή με κάποιον γιατρό. Αν ο ασθενής δεν υπάρχει, τότε θα πρέπει ο χρήστης να δημιουργήσει έναν νέο ασθενή καταχωρώντας τα στοιχεία του. Τα βήματα τα οποία θα πρέπει να εκτελέσει ο χειριστής του συστήματος εμφανίζονται αναλυτικά στην Εικόνα 2.5. **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε..** Όμοια, στην Εικόνα 2.6 έχουμε τα βήματα που απαιτούνται όταν πραγματοποιηθεί ένα ραντεβού και θα πρέπει να δηλωθεί από τον χειριστή του συστήματος ως ολοκληρωμένο. Τέλος, στην Εικόνα 2.7 έχουμε τα βήματα για το σενάριο της εισαγωγής του ιστορικού ενός ασθενούς. Αυτά αποτελούν τρία ενδεικτικά σενάρια χρήσης του συστήματος περιγράφοντας κάποιες από τις βασικές λειτουργίες του.

Τίτλος Σεναρίου: Κλείσιμο ραντεβού για νέο Ασθενή

1. Ο χειριστής του συστήματος επιλέγει “Δημιουργία νέου Ασθενούς” από το μενού “Ασθενής” στο κεντρικό παράθυρο της εφαρμογής.
2. Το σύστημα εμφανίζει μία φόρμα στην οποία μπορεί ο χειριστής να κάνει την καταχώρηση των στοιχείων του νέου ασθενή.
3. Ο χειριστής καταχωρεί τα προσωπικά στοιχεία του νέου ασθενή και πατάει το κουμπί “Αποθήκευση”.
4. Στη συνέχεια, ο χρήστης επιλέγει “Νέο Ραντεβού” από το μενού “Επισκέψεις”.
5. Το σύστημα εμφανίζει μία νέα φόρμα στην οποία μπορεί ο χρήστης να κάνει την καταχώρηση των στοιχείων για το νέο ραντεβού.
6. Ο χρήστης καταχωρεί τα στοιχεία που αφορούν το νέο ραντεβού, όπως την ημέρα και την ώρα του ραντεβού, επιλέγει τον ασθενή που αφορά το ραντεβού και τον γιατρό που θα τον εξετάσει και στη συνέχεια πατάει το κουμπί “Αποθήκευση”.
7. Το σύστημα επιστρέφει αυτόματα στο κεντρικό μενού της εφαρμογής.

Εικόνα 2.5 Σενάριο εισαγωγής ραντεβού για νέο ασθενή

Τίτλος Σεναρίου: Πραγματοποίηση ενός ραντεβού

1. Ο χειριστής επιλέγει “Λίστα Ραντεβού” από το μενού “Επισκέψεις” του κεντρικού παραθύρου της εφαρμογής.
2. Το σύστημα εμφανίζει μία λίστα με τα ραντεβού τα οποία δεν έχουν πραγματοποιηθεί ακόμη.
3. Ο χειριστής επιλέγει το ραντεβού για τον συγκεκριμένο ασθενή το οποίο θέλει να τροποποιήσει και πατάει το κουμπί “Τροποποίηση”.
4. Το σύστημα προβάλλει στην οθόνη τα στοιχεία του ραντεβού.
5. Ο χρήστης αλλάζει την κατάσταση του ραντεβού σε “Ολοκληρωμένη Επίσκεψη” και πατάει το κουμπί “Αποθήκευση”.
6. Το σύστημα επιστρέφει αυτόματα στο κεντρικό μενού της εφαρμογής.

Εικόνα 2.6 Σενάριο ολοκλήρωσης ενός ραντεβού

Τίτλος Σεναρίου: Εισαγωγή του ιστορικού ενός ασθενή

1. Ο χειριστής επιλέγει από το μενού “Ασθενείς” της κεντρικής οθόνης την επιλογή “Λίστα Ασθενών”.
2. Το σύστημα εμφανίζει μία λίστα με όλους τους καταχωρημένους ασθενείς με αλφαβητική σειρά.
3. Ο χρήστης επιλέγει τον ασθενή για τον οποίο θέλει να εισάγει το ιστορικό του και πατάει το κουμπάκι “Τροποποίηση”.
4. Το σύστημα εμφανίζει τα προσωπικά στοιχεία του ασθενή.
5. Ο χρήστης πατάει το κουμπάκι “Εμφάνιση Ιστορικού”.
6. Το σύστημα εμφανίζει μία φόρμα όπου μπορεί να γίνει καταχώρηση των στοιχείων που αφορούν το ιστορικό του ασθενή.
7. Ο χρήστης εισάγει τα στοιχεία που θέλει και πατάει το κουμπάκι “Αποθήκευση”.
8. Το σύστημα επιστρέφει αυτόματα στο κεντρικό μενού της εφαρμογής.

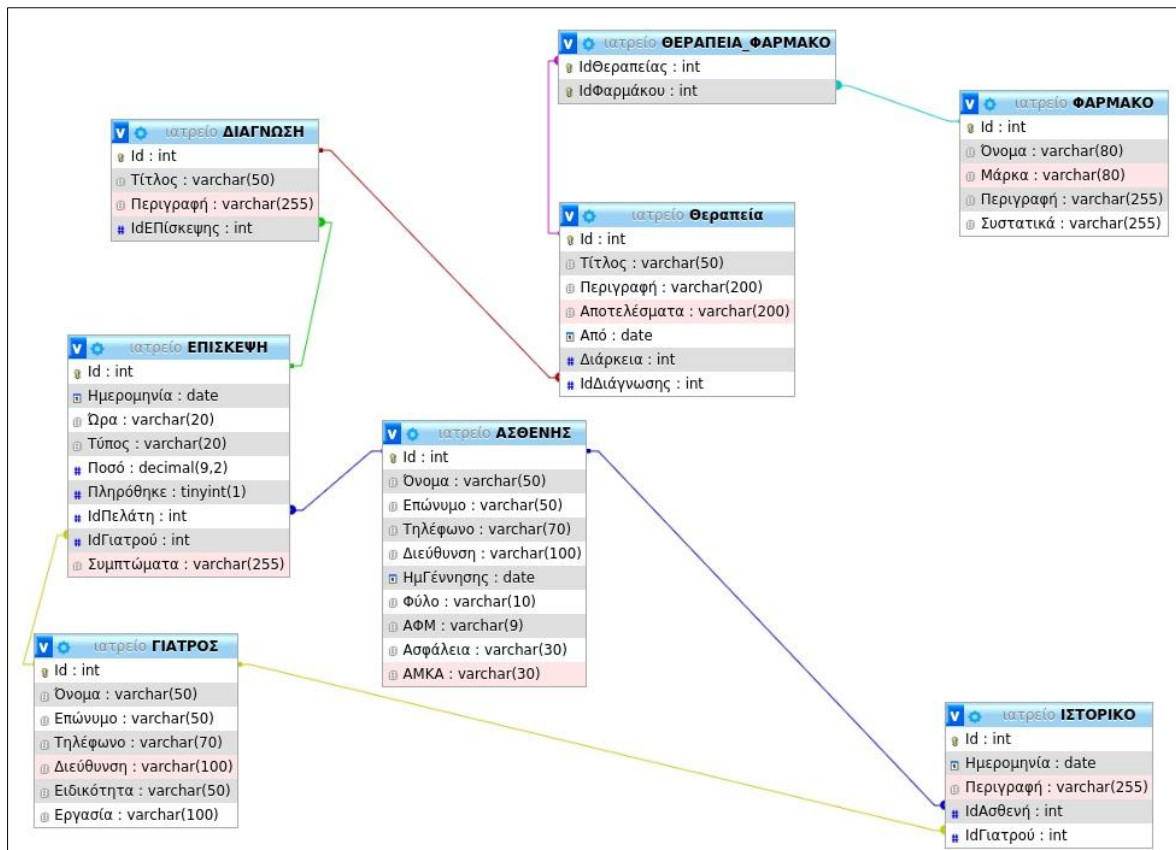
Εικόνα 2.7 Σενάριο για την εισαγωγή του ιστορικού ενός ασθενή

3. Ανάπτυξη του συστήματος μηχανογράφησης Ιατρείου

3.1. Ανάλυση και υλοποίηση της Βάσης Δεδομένων του συστήματος

Κάθε σύστημα μηχανογράφησης, για οποιαδήποτε μικρή ή μεγάλη εταιρεία ή οργανισμό, κάνει χρήση κάποιου είδους τεχνολογίας για την αποθήκευση των δεδομένων, τα οποία χειρίζεται το αντίστοιχο πληροφοριακό σύστημα. Για τον λόγο αυτό χρησιμοποιείται συνήθως κάποια Βάση Δεδομένων την οποία διαχειρίζεται ένα Σύστημα Διαχείρισης Βάσης Δεδομένων – ΣΔΒΜ (Data Base Management System – DBMS). Η σχεδίαση της Βάσης Δεδομένων (ΒΔ) αποτελεί ένα πολύ σημαντικό στάδιο κατά την υλοποίηση ενός συστήματος και θα πρέπει πάντοτε να γίνεται με ιδιαίτερη προσοχή. Οι πληροφορίες που αποθηκεύονται σε μία ΒΔ αναπαρίστανται συνήθως με τη βοήθεια των κατάλληλων Οντοτήτων (Entities) και Συσχετίσεων (Relationships) σε ένα Διάγραμμα Οντοτήτων Συσχετίσεων – ΔΟΣ (Entity Relationship Diagram–ERD).

Το Διάγραμμα Οντοτήτων Συσχετίσεων για την περίπτωση της μηχανογράφησης ενός Ιατρείου εμφανίζεται στο Διάγραμμα 3.1. Για κάθε μία από τις οντότητες, ορίζονται τα γνωρίσματά (πεδία) που περιλαμβάνει και ο τύπος του κάθε γνωρίσματος. Ορίζεται επίσης το πρωτεύον (κύριο) κλειδί (primary key) για κάθε μία από τις οντότητες. Το διάγραμμα ERD περιγράφει τη βάση δεδομένων σε λογικό επίπεδο. Κατά τη μετάβαση από το λογικό στο φυσικό επίπεδο, θα πρέπει να οριστούν οι σχέσεις (relations) της βάσης.



Διάγραμμα 3.1 Διάγραμμα Οντοτήτων Συσχετίσεων για τη ΒΔ του συστήματος

Για το σύστημα μηχανογράφησης του Ιατρείου που αναπτύσσεται, δημιουργήθηκαν οι πιο κάτω σχέσεις:

Σχέσεις	Περιγραφή σχέσης
ΓΙΑΤΡΟΣ	Η σχέση αυτή περιλαμβάνει πληροφορίες που αφορούν τους γιατρούς που συνεργάζονται με το Ιατρείο. Για κάθε γιατρό καταγράφονται τα πιο κάτω γνωρίσματα:
Id	Ένα μοναδικό αναγνωριστικό για τον κάθε γιατρό, το οποίο είναι ακέραιου τύπου και παράγεται με αυτόματο τρόπο. Αποτελεί το πρωτεύον κλειδί της σχέσης.
Όνομα	Το όνομα του γιατρού, το οποίο είναι τύπου χαρακτήρων (συμβολοσειρά).
Επώνυμο	Το επώνυμο του γιατρού, το οποίο είναι τύπου χαρακτήρων (συμβολοσειρά).
Τηλέφωνο	Τα τηλέφωνα (ένα ή περισσότερα) του γιατρού, τύπου χαρακτήρων.
Διεύθυνση	Η διεύθυνση κατοικίας του γιατρού, τύπου χαρακτήρων.

Σχέσεις	Περιγραφή σχέσης
Ειδικότητα	Η ειδικότητα του γιατρού, τύπου χαρακτήρων.
Εργασία	Ένας γιατρός μπορεί να είναι απλά συνεργάτης του ιατρείου και να εργάζεται ταυτόχρονα σε κάποιον άλλον φορέα, όπως π.χ ένα νοσοκομείο. Στο πεδίο αυτό καταγράφεται ο φορέας όπου εργάζεται πιθανόν ο γιατρός και είναι τύπου χαρακτήρων.
ΑΣΘΕΝΗΣ (ΠΕΛΑΤΗΣ)	Η σχέση αυτή περιλαμβάνει πληροφορίες που αφορούν τους ασθενείς (πελάτες) του Ιατρείου. Για κάθε ασθενή καταγράφονται τα πιο κάτω γνωρίσματα:
Id	Ένα μοναδικό αναγνωριστικό για τον κάθε ασθενή (πελάτη), το οποίο είναι ακεραίου τύπου και παράγεται με αυτόματο τρόπο. Αποτελεί το πρωτεύον κλειδί της σχέσης.
Όνομα	Το όνομα του ασθενή, τύπου χαρακτήρων.
Επώνυμο	Το επώνυμο του ασθενή, τύπου χαρακτήρων.
Τηλέφωνο	Περιλαμβάνει τα τηλέφωνα (ένα ή περισσότερα) του ασθενή. Είναι τύπου χαρακτήρων.
Διεύθυνση	Η διεύθυνση όπου μένει ο ασθενής, τύπου χαρακτήρων.
Ημερομηνία Γέννησης	Η ημερομηνία γέννησης του ασθενή, τύπου ημερομηνίας.
Φύλο	Το φύλο του πελάτη, τύπου χαρακτήρων (προκαθορισμένες τιμές).
ΑΦΜ	Ο Αριθμός Φορολογικού Μητρώου (ΑΦΜ) του ασθενή, τύπου χαρακτήρων (9 ψηφία).
Ασφάλεια	Η ασφάλεια υγείας που διαθέτει ο ασθενής, τύπου χαρακτήρων.
ΑΜΚΑ	Ο ΑΜΚΑ του ασθενή, τύπου χαρακτήρων (11 ψηφία).
ΕΠΙΣΚΕΨΗ	Περιλαμβάνει πληροφορίες που αφορούν μία επίσκεψη που κάνει ένας ασθενής (πελάτης) στο Ιατρείο. Περιλαμβάνει τα γνωρίσματα:
Id	Ένα μοναδικό αναγνωριστικό επίσκεψης, το οποίο είναι ακεραίου τύπου και παράγεται με αυτόματο τρόπο. Αποτελεί το πρωτεύον κλειδί της σχέσης.
Ημερομηνία	Η ημερομηνία της επίσκεψης, τύπου ημερομηνίας.
Ώρα	Η ώρα της επίσκεψης, τύπου χαρακτήρων.
Τύπος	Αποθηκεύει την κατάσταση στην οποία μπορεί να βρίσκεται μία επίσκεψη. Μπορεί για παράδειγμα να αποτελεί απλά ένα προγραμματισμένο ραντεβού το οποίο δεν έχει πραγματοποιηθεί ακόμη ή μπορεί να έχει ήδη ολοκληρωθεί. Είναι τύπου χαρακτήρων.

Σχέσεις	Περιγραφή σχέσης
Ποσό	Το ποσό που αντιστοιχεί στην αμοιβή που πληρώνει ο πελάτης. Είναι δεκαδικού τύπου με 2 ψηφία ακρίβεια.
Πληρώθηκε	Είναι τύπου boolean (true/false) και δηλώνει αν έχει πραγματοποιηθεί η πληρωμή ή όχι.
IdΠελάτη	Το id του πελάτη που αφορά η επίσκεψη. Αποτελεί ξένο κλειδί και αναφέρεται στη σχέση ΑΣΘΕΝΗΣ.
IdΓιατρού	Το id του γιατρού που κάνει την εξέταση. Αποτελεί ξένο κλειδί και αναφέρεται στη σχέση ΓΙΑΤΡΟΣ.
Συμπτώματα	Περιλαμβάνει τα συμπτώματα τα οποία έχει ο ασθενής κατά τη στιγμή της επίσκεψης. Είναι τύπου χαρακτήρων.
ΔΙΑΓΝΩΣΗ	Περιλαμβάνει πληροφορίες που αφορούν τη διάγνωση που έκανε ο γιατρός. Έχει τα γνωρίσματα:
Id	Ένα μοναδικό αναγνωριστικό διάγνωσης, το οποίο είναι ακέραιου τύπου και παράγεται με αυτόματο τρόπο. Αποτελεί το πρωτεύον κλειδί της σχέσης.
Τίτλος	Περιλαμβάνει κάποιου είδους ονομασία που αφορά τη διάγνωση που έγινε και είναι τύπου χαρακτήρων.
Περιγραφή	Περιλαμβάνει αναλυτική περιγραφή της διάγνωσης και είναι τύπου χαρακτήρων.
IdΕπίσκεψης	Το id της επίσκεψης που αφορά η διάγνωση. Αποτελεί ξένο κλειδί και αναφέρεται στη σχέση ΕΠΙΣΚΕΨΗ.
ΙΣΤΟΡΙΚΟ	Αποτελεί σχέση για την αποθήκευση πληροφοριών που αφορούν το ιστορικό ενός ασθενούς. Έχει τα γνωρίσματα:
Id	Ένα μοναδικό αναγνωριστικό της εγγραφής του ιστορικού, το οποίο είναι ακέραιου τύπου και παράγεται με αυτόματο τρόπο. Αποτελεί το πρωτεύον κλειδί της σχέσης.
Ημερομηνία	Η ημερομηνία που αφορά το ιστορικό, τύπου ημερομηνίας.
Περιγραφή	Περιλαμβάνει αναλυτική περιγραφή της εγγραφής του ιστορικού και είναι τύπου χαρακτήρων.
IdΑσθενή	Το id του ασθενή που αφορά η επίσκεψη. Αποτελεί ξένο κλειδί και αναφέρεται στη σχέση ΑΣΘΕΝΗΣ.
IdΓιατρού	Το id του γιατρού που σχετίζεται με την εγγραφή του ιστορικού. Αποτελεί ξένο κλειδί και αναφέρεται στη σχέση ΓΙΑΤΡΟΣ.
ΘΕΡΑΠΕΙΑ	Η σχέση αυτή περιγράφει τη θεραπεία η οποία γίνεται από έναν ασθενή. Περιλαμβάνει τα γνωρίσματα:

Σχέσεις	Περιγραφή σχέσης
Id	Ένα μοναδικό αναγνωριστικό της θεραπείας, το οποίο είναι ακέραιου τύπου και παράγεται με αυτόματο τρόπο. Αποτελεί το πρωτεύον κλειδί της σχέσης.
Τίτλος	Περιλαμβάνει κάποιου είδους ονομασία που αφορά τη θεραπεία και είναι τύπου χαρακτήρων.
Περιγραφή	Περιλαμβάνει αναλυτική περιγραφή της θεραπείας και είναι τύπου χαρακτήρων.
Αποτελέσματα	Περιλαμβάνει τα αποτελέσματα της θεραπείας και είναι τύπου χαρακτήρων.
Από:	Αφορά την ημερομηνία στην οποία ξεκινά μία θεραπεία. Είναι τύπου ημερομηνίας.
Διάρκεια	Καταγράφει τη διάρκεια μίας θεραπείας σε ημέρες. Είναι ακεραίου τύπου.
IdΔιάγνωσης	Το id της διάγνωσης στην οποία αναφέρεται η θεραπεία. Αποτελεί ξένο κλειδί και αναφέρεται στη σχέση ΔΙΑΓΝΩΣΗ.
ΦΑΡΜΑΚΟ	Περιλαμβάνει πληροφορίες για τα φάρμακα τα οποία θα πρέπει να λάβει ο ασθενής κατά τη θεραπεία του. Έχουμε τα γνωρίσματα:
Id	Ένα μοναδικό αναγνωριστικό του φαρμάκου, το οποίο είναι ακέραιου τύπου και παράγεται με αυτόματο τρόπο. Αποτελεί το πρωτεύον κλειδί της σχέσης.
Όνομα	Το όνομα του φαρμάκου, τύπου χαρακτήρων.
Μάρκα	Το όνομα της εταιρείας που παράγει το φάρμακο. Είναι τύπου χαρακτήρων.
Περιγραφή	Περιλαμβάνει μία σύντομη περιγραφή του φαρμάκου και είναι τύπου χαρακτήρων.
Συστατικά	Περιλαμβάνει τα συστατικά από τα οποία αποτελείται ένα φάρμακο και είναι τύπου χαρακτήρων.

Οι σχέσεις ΘΕΡΑΠΕΙΑ και ΦΑΡΜΑΚΟ συνδέονται μεταξύ τους με μία συσχέτιση τύπου «πολλά προς πολλά» και για τον λόγο αυτό έχει δημιουργηθεί η σχέση ΘΕΡΑΠΕΙΑ_ΦΑΡΜΑΚΟ για να καταγράψει τις πληροφορίες που αφορούν αυτή τη συσχέτιση.

Στην Εικόνα 3.1 εμφανίζονται οι εντολές SQL που χρησιμοποιήθηκαν για τη δημιουργία της Βάσης Δεδομένων, ενώ στην Εικόνα 3.2 εμφανίζονται οι αντίστοιχοι περιορισμοί.

```

CREATE TABLE `ΓΙΑΤΡΟΣ` (
  Id INT AUTO_INCREMENT,
  Όνομα varchar(50) NOT NULL,
  Επώνυμο varchar(50) NOT NULL,
  Τηλέφωνο varchar(70),
  Διεύθυνση varchar(100),
  Ειδικότητα varchar(50),
  Εργασία varchar(100),
  PRIMARY KEY (Id)
);

CREATE TABLE `ΑΣΘΕΝΗΣ` (
  Id INT AUTO_INCREMENT,
  Όνομα varchar(50) NOT NULL,
  Επώνυμο varchar(50) NOT NULL,
  Τηλέφωνο varchar(70),
  Διεύθυνση varchar(100),
  ΗμΓέννησης date,
  Φύλο varchar(10),
  ΑΦΜ varchar(9),
  Ασφάλεια varchar(30),
  ΑΜΚΑ varchar(30),
  PRIMARY KEY (Id)
);

CREATE TABLE `ΦΑΡΜΑΚΟ` (
  Id INT AUTO_INCREMENT,
  Όνομα varchar(80) NOT NULL,
  Μάρκα varchar(80),
  Περιγραφή varchar(255),
  Συστατικά varchar(255),
  PRIMARY KEY (Id)
);

CREATE TABLE `ΕΠΙΣΚΕΨΗ` (
  Id INT AUTO_INCREMENT,
  Ημερομηνία date NOT NULL,
  Ωρα varchar(20),
  Τύπος varchar(20),
  Ποσό decimal(9, 2),
  Πληρώθηκε BOOLEAN,
  IdΠελάτη INT NOT NULL,
  IdΓιατρού INT NOT NULL,
  Συμπτώματα varchar(255),
  PRIMARY KEY (Id)
);

CREATE TABLE `ΔΙΑΓΝΩΣΗ` (
  Id INT AUTO_INCREMENT,
  Τίτλος varchar(50) NOT NULL,
  Περιγραφή varchar(255),
  IdΕπίσκεψης INT NOT NULL,
  PRIMARY KEY (Id)
);

CREATE TABLE `Θεραπεία` (
  Id INT AUTO_INCREMENT,
  Τίτλος varchar(50) NOT NULL,
  Περιγραφή varchar(200),
  Αποτελέσματα varchar(200),
  Από date,
  Διάρκεια INT,
  IdΔιάγνωσης INT NOT NULL,
  PRIMARY KEY (Id)
);

CREATE TABLE `ΙΣΤΟΡΙΚΟ` (
  Id INT AUTO_INCREMENT,
  Ημερομηνία date NOT NULL,
  Περιγραφή varchar(255) NOT NULL,
  IdΑσθενή INT NOT NULL,
  IdΓιατρού INT NOT NULL,
  PRIMARY KEY (Id)
);

CREATE TABLE `ΘΕΡΑΠΕΙΑ ΦΑΡΜΑΚΟ` (
  IdΘεραπείας INT NOT NULL,
  IdΦαρμάκου INT NOT NULL,
  PRIMARY KEY (IdΘεραπείας, IdΦαρμάκου)
);

```

Εικόνα 3.1 Εντολές SQL για τη δημιουργία της Βάσης Δεδομένων

```

ALTER TABLE `ΕΠΙΣΚΕΨΗ`
  ADD CONSTRAINT C1
  FOREIGN KEY (IdΠελάτη) REFERENCES `ΑΣΘΕΝΗΣ` (Id);

ALTER TABLE `ΙΣΤΟΡΙΚΟ`
  ADD CONSTRAINT C2
  FOREIGN KEY (IdΑσθενή) REFERENCES `ΑΣΘΕΝΗΣ` (Id);

ALTER TABLE `ΔΙΑΓΝΩΣΗ`
  ADD CONSTRAINT C3
  FOREIGN KEY (IdΕπίσκεψης) REFERENCES `ΕΠΙΣΚΕΨΗ` (Id);

ALTER TABLE `Θεραπεία`
  ADD CONSTRAINT C4
  FOREIGN KEY (IdΔιάγνωσης) REFERENCES `ΔΙΑΓΝΩΣΗ` (Id);

ALTER TABLE `ΙΣΤΟΡΙΚΟ`
  ADD CONSTRAINT C5
  FOREIGN KEY (IdΓιατρού) REFERENCES `ΓΙΑΤΡΟΣ` (Id);

ALTER TABLE `ΘΕΡΑΠΕΙΑ_ΦΑΡΜΑΚΟ`
  ADD CONSTRAINT C6
  FOREIGN KEY (IdΘεραπείας) REFERENCES `Θεραπεία` (Id);

ALTER TABLE `ΘΕΡΑΠΕΙΑ_ΦΑΡΜΑΚΟ`
  ADD CONSTRAINT C7
  FOREIGN KEY (IdΦαρμάκου) REFERENCES `ΦΑΡΜΑΚΟ` (Id);

ALTER TABLE `ΕΠΙΣΚΕΨΗ`
  ADD CONSTRAINT C8
  FOREIGN KEY (IdΓιατρού) REFERENCES `ΓΙΑΤΡΟΣ` (Id);

```

Εικόνα 3.2 Δημιουργία περιορισμών που αφορούν τα ξένα κλειδιά της ΒΔ

3.2. Προγραμματισμός με τη γλώσσα Java

Για την ανάπτυξη της εφαρμογής έγινε χρήση της γλώσσας προγραμματισμού Java. Η Java αποτελεί μία δημοφιλή, πλήρως αντικειμενοστρεφή και υψηλού επιπέδου γλώσσα προγραμματισμού. Η γλώσσα αναπτύχθηκε αρχικά το 1991 από την Sun Microsystems και τον Καναδό επιστήμονα της Πληροφορικής James Gosling. Από το 2010 και μετά, τα δικαιώματά της γλώσσας τα κατέχει η εταιρεία Oracle.

Η Java είναι μία γλώσσα προγραμματισμού γενικού σκοπού και χρησιμοποιείται πολύ συχνά για την ανάπτυξη εφαρμογών διαφόρων τύπων, όπως εμπορικές εφαρμογές,

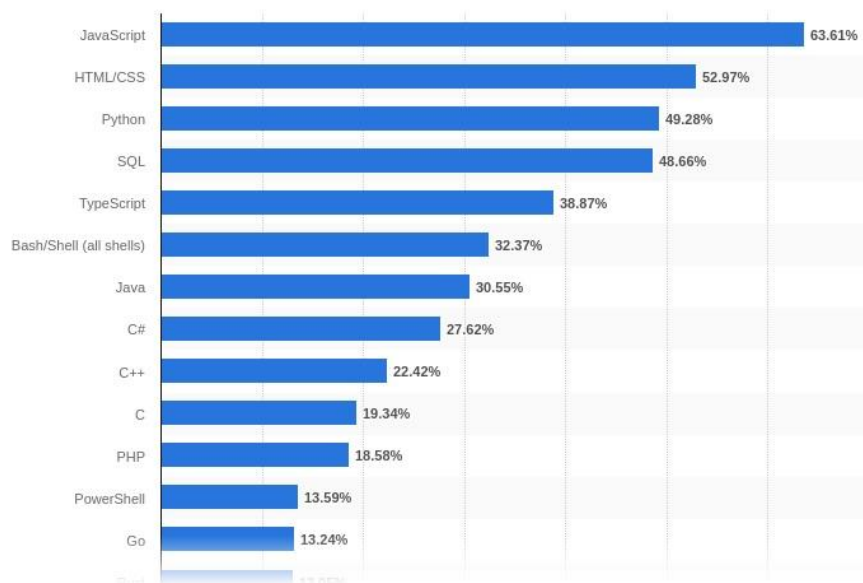
παιχνίδια, εφαρμογές Παγκόσμιου Ιστού κ.λ.π. Προσφέρει πληθώρα έτοιμων βιβλιοθηκών και επιτρέπει την επαναχρησιμοποίηση του κώδικα, μειώνοντας έτσι αισθητά τη διάρκεια του κύκλου ζωής του λογισμικού, όπως και το κόστος ανάπτυξης μίας εφαρμογής. Η χρήση της γλώσσας είναι δωρεάν και υπάρχει μία πολύ μεγάλη κοινότητα προγραμματιστών από τους οποίους μπορεί να αντλήσει βοήθεια οποιοσδήποτε ασχολείται με τον προγραμματισμό Java.

Σύμφωνα με τον δείκτη TIOBE, όπως φαίνεται στην Εικόνα 3.3, η Java είναι η τέταρτη πιο δημοφιλής γλώσσα προγραμματισμού για τον Μάιο του 2024 (TIOBE, 2024). Ο δείκτης TIOBE μετρά τη δημοφιλία των γλωσσών προγραμματισμού σε μηνιαία βάση και βασίζεται στις προτιμήσεις των προγραμματιστών και μηχανικών πληροφορικής σε όλον τον κόσμο, συλλέγοντας στοιχεία από δημοφιλείς ιστοτόπους, όπως Google, Amazon, Wikipedia κ.λ.π.

Η γλώσσα Java βρίσκεται πολύ ψηλά επίσης στην κατάταξη της Statista (Statista, 2023), όπως φαίνεται στην Εικόνα 3.4, ο οποίος βασίζεται επίσης στις προτιμήσεις των προγραμματιστών. Όσον αφορά μάλιστα στις αναζητήσεις των απλών χρηστών του Διαδικτύου για βοηθήματα που αφορούν γλώσσες προγραμματισμού, η Java βρίσκεται στη δεύτερη θέση σύμφωνα με τις εκτιμήσεις του δείκτη PYPL (PYPL, 2024), όπως εμφανίζεται στον στην Εικόνα 3.5.

May 2024	May 2023	Change	Programming Language	Ratings	Change
1	1		 Python	16.33%	+2.88%
2	2		 C	9.98%	-3.37%
3	4	▲	 C++	9.53%	-2.43%
4	3	▼	 Java	8.69%	-3.53%
5	5		 C#	6.49%	-0.94%
6	7	▲	 JavaScript	3.01%	+0.57%
7	6	▼	 Visual Basic	2.01%	-1.83%
8	12	▲	 Go	1.60%	+0.61%
9	9		 SQL	1.44%	-0.03%
10	19	▲	 Fortran	1.24%	+0.46%

Εικόνα 3.3 Δείκτης TIOBE για τις γλώσσες προγραμματισμού



Εικόνα 3.4 Δείκτης Statista για τη δημοφιλή γλωσσών προγραμματισμού

Rank	Change	Language	Share	1-year trend
1		Python	28.98 %	+1.2 %
2		Java	15.97 %	+0.1 %
3		JavaScript	8.79 %	-0.6 %
4		C#	6.78 %	-0.0 %
5		C/C++	6.46 %	-0.0 %
6	↑	R	4.76 %	+0.3 %
7	↓	PHP	4.55 %	-0.6 %
8		TypeScript	3.03 %	-0.0 %
9		Swift	2.76 %	+0.4 %
10		Rust	2.6 %	+0.4 %

Εικόνα 3.5 Δείκτης PYPL ως προς τις αναζητήσεις βοηθημάτων

Από τα πιο πάνω, γίνεται σαφές ότι η γλώσσα προγραμματισμού Java είναι μία από τις πλέον δημοφιλείς σύγχρονες γλώσσες προγραμματισμού και αυτό εξηγεί τον λόγο που επιλέχθηκε για την υλοποίηση της εφαρμογής μηχανογράφησης ενός Ιατρείου. Ένα από τα βασικότερα πλεονεκτήματα της Java, είναι ότι μία εφαρμογή που έχει υλοποιηθεί σε Java, μπορεί να χρησιμοποιηθεί σε οποιαδήποτε υπολογιστική πλατφόρμα. Ο λόγος είναι ότι

κατά τη μεταγλώττιση ενός προγράμματος Java, δεν παράγεται κώδικας σε γλώσσα μηχανής (άμεσα εκτελέσιμη μορφή), αλλά παράγεται μία ενδιάμεση μορφή από “bytecodes” (αρχεία της μορφής .class), η οποία μπορεί να εκτελεστεί από την εικονική μηχανή (JVM – Java Virtual Machine) οποιουδήποτε υπολογιστή. Το γεγονός αυτό βοηθάει πάρα πολύ στην περίπτωση ανάπτυξης εφαρμογής για τη μηχανογράφηση συστημάτων, όπως στην περίπτωση του πληροφοριακού συστήματος ενός Ιατρείου, αφού η εφαρμογή αυτή θα μπορεί να εκτελεστεί με τον επιθυμητό τρόπο, ανεξαρτήτως του λειτουργικού συστήματος που είναι διαθέσιμος.

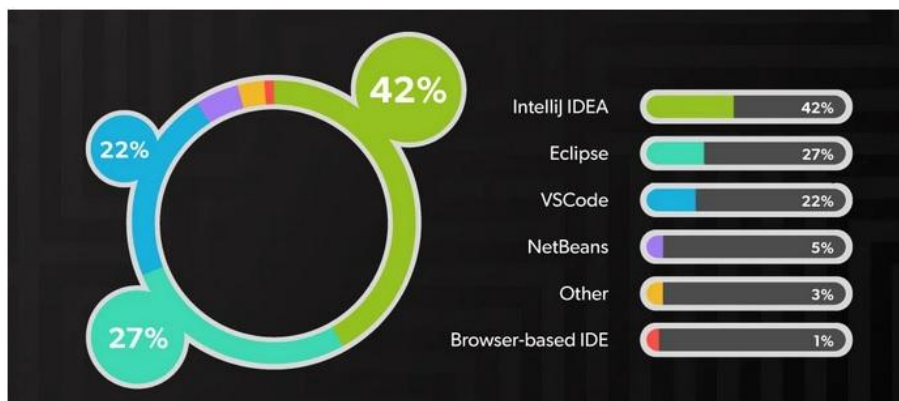
3.3. Java Development Kit (JDK) και Apache NetBeans

Για την ανάπτυξη ενός προγράμματος Java, θα πρέπει να έχουν εγκατασταθεί οι κατάλληλες βιβλιοθήκες JDK (Java Development Kit) στο σύστημα όπου αναπτύσσεται η εφαρμογή από τον προγραμματιστή. Οι νέες εκδόσεις του JDK διανέμονται από την Oracle και πέρα από την εικονική μηχανή JVM, περιλαμβάνει τον μεταγλωττιστή (compiler) της γλώσσας, έναν εκσφαλματωτή (debugger) για τον εντοπισμό και διόρθωση σφαλμάτων (bugs), εργαλεία για τη βελτίωση της απόδοσης του προγράμματος που αναπτύσσεται, όπως και άλλες χρήσιμες εφαρμογές. Η Oracle διανέμει το JDK κάτω από την άδεια NFTC (No-Fee Terms and Conditions), ενώ υπάρχει και η ανοικτού κώδικα υλοποίηση της γλώσσας, που ονομάζεται Open JDK με άδεια GPL.

Εκτός από την τυπική έκδοση της γλώσσας, η οποία αναφέρεται με τον όρο Standard Edition (SE), η Oracle κυκλοφορεί και τις εκδόσεις Enterprise Edition (EE), Micro Edition (ME) και JavaFX. Η έκδοση SE περιλαμβάνει τις βασικές βιβλιοθήκες για την ανάπτυξη απλών ή σύνθετων εφαρμογών που μπορεί να περιλαμβάνουν γραφικό περιβάλλον (GUI), διασύνδεση με δίκτυο, επικοινωνία με ΒΔ, ασφαλή επικοινωνία κ.τ.λ. Η έκδοση EE, η οποία είναι γνωστή και σαν Jakarta EE, προσφέρει επιπλέον δυνατότητες για την ανάπτυξη δικτυακών εφαρμογών που λειτουργούν πάνω σε μεγάλο πλήθος υπολογιστικών μονάδων και προσφέρει μεγάλη επεκτασιμότητα (scalability) και πρόσθετες υπηρεσίες ασφαλείας. Η έκδοση ME αφορά την ανάπτυξη εφαρμογών σε μικρές συσκευές, όπως κινητά τηλέφωνα. Η έκδοση JavaFX προσφέρει επιπρόσθετες δυνατότητες για ανάπτυξη Διαδικτυακών εφαρμογών και ανανεωμένο γραφικό περιβάλλον.

Για τις ανάγκες της εφαρμογής που αναπτύχθηκε στα πλαίσια της παρούσας εργασίας, χρησιμοποιήθηκαν οι βιβλιοθήκες της έκδοσης EE, δεδομένου ότι επιλέχθηκε η χρήση του ειδικού server της Java, όπως περιγράφεται στις επόμενες παραγράφους, για την αποθήκευση των δεδομένων στη βάση. Η τρέχουσα έκδοση του JDK SE είναι η έκδοση 22, η οποία κυκλοφόρησε τον Μάρτιο του 2024 και είναι αυτή η οποία χρησιμοποιήθηκε για την ανάπτυξη της εφαρμογής.

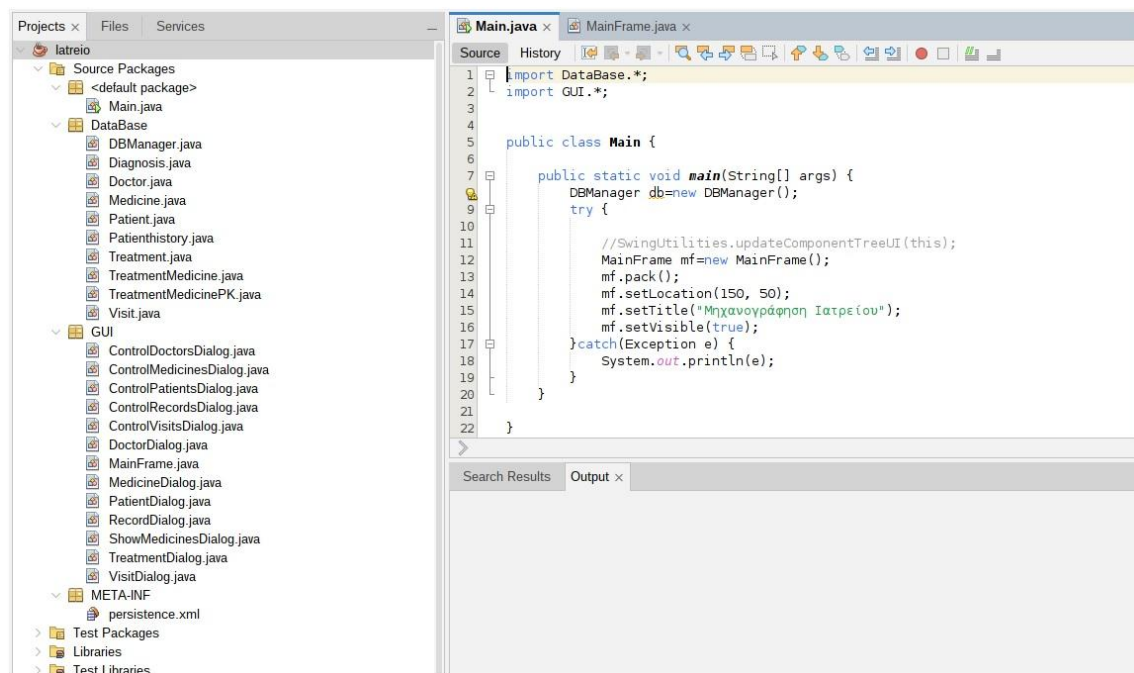
Από τη στιγμή που έχει γίνει εγκατάσταση του κατάλληλου JDK από τον προγραμματιστή, η ανάπτυξη μίας εφαρμογής Java μπορεί να γίνει σε οποιοδήποτε από τα διαθέσιμα περιβάλλοντα, ακόμη και με έναν απλό επεξεργαστή κειμένου (editor). Υπάρχουν όμως διάφορα προγραμματιστικά περιβάλλοντα που μπορούν να βοηθήσουν τους προγραμματιστές Java να είναι πιο αποδοτικοί και να αναπτύξουν εύκολα τις εφαρμογές τους, όπως είναι το Apache NetBeans, το Eclipse, το IntelliJ IDEA, το VSCode κ.λ.π. Τα περιβάλλοντα αυτά αναφέρονται συχνά με τον όρο Ολοκληρωμένα Περιβάλλοντα Προγραμματισμού (Integrated Development Environments) ή απλά IDE. Στην Εικόνα 3.6, εμφανίζονται με γραφικό τρόπο τα πιο δημοφιλή IDEs για την γλώσσα προγραμματισμού Java, σύμφωνα με την ετήσια αναφορά της JRebel.



Εικόνα 3.6 Δημοφιλή IDEs για Java

Το Net Beans έχει υλοποιηθεί σε Java και μπορεί να χρησιμοποιηθεί σε περιβάλλον Windows, macOS, Linux κ.τ.λ. για την ανάπτυξη εφαρμογών Java, αλλά με την εγκατάσταση των κατάλληλων πρόσθετων και για άλλες γλώσσες προγραμματισμού, όπως HTML, PHP, C++ κ.λ.π. Αν, και υπάρχουν πλέον πιο δημοφιλή περιβάλλοντα, το NetBeans εξακολουθεί να βρίσκεται ανάμεσα στα τέσσερα πιο δημοφιλή περιβάλλοντα και δίνει τη δυνατότητα της ανάπτυξης γραφικών διεπαφών με έναν ιδιαίτερα εύχρηστο

τρόπο (λειτουργία DragandDrop). Αυτός είναι ο λόγος που προτιμήθηκε η χρήση του περιβάλλοντος του NetBeans IDE για την υλοποίηση της εφαρμογής (Εικόνα 3.7).



Εικόνα 3.7 Το περιβάλλον του Apache NetBeans IDE

Αρχικά το NetBeans κυκλοφόρησε το 1995 με το όνομα Xelphi στο Charles University στην Πράγα, αλλά στην πορεία διατέθηκε με εμπορική άδεια χρήσης. Το 1999 εξαγοράστηκε από την Sun Microsystems και στη συνέχεια από την Oracle, μαζί με την γλώσσα Java. Το 2016 όμως η Oracleδώρισε την εφαρμογή στο μη κερδοσκοπικό ίδρυμα Apache Software Foundation, το οποίο είναι υπεύθυνο από τότε για τη διαχείριση του NetBeans ως λογισμικό ανοικτού κώδικα.

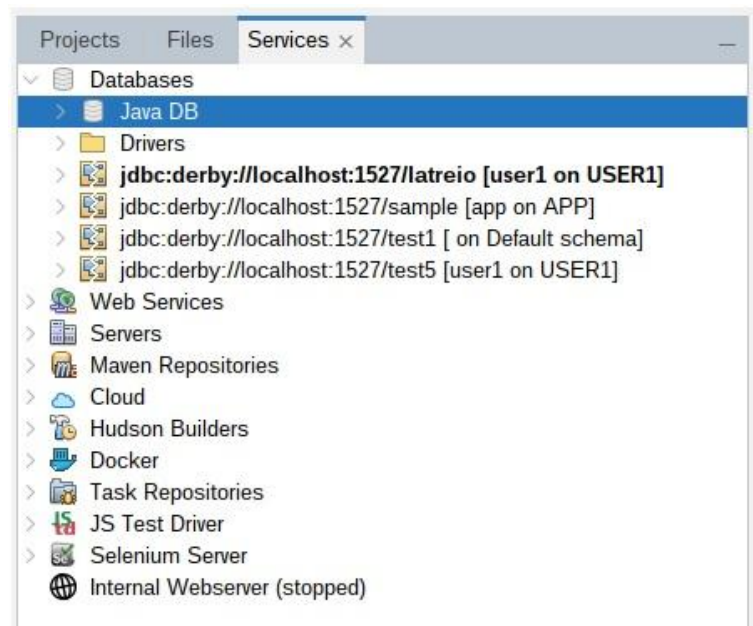
Το Apache NetBeans υποστηρίζει πλήρως τη βιβλιοθήκη Swing,για την ανάπτυξη εφαρμογών Javaπου προσφέρουν φιλικό προς τον χρήστη γραφικό περιβάλλον (GUI), όπως επίσης υποστηρίζει και τη βιβλιοθήκη γραφικών JavaFX. Η βιβλιοθήκη Swing ήταν αυτή που χρησιμοποιήθηκε, με τη βοήθεια του Apache NetBeans, για την ανάπτυξη του γραφικού περιβάλλοντος της εφαρμογής, λόγω της σταθερότητας που προσφέρει και της ωριμότητας που παρουσιάζει ως προϊόν λογισμικού.

3.4. GlassFish και Java Persistence API (JPA)

Το λογισμικό GlassFish αποτελεί έναν ανοικτού κώδικα εξυπηρετητή εφαρμογών (application server) που είναι συμβατός με την Java έκδοση EE, ο οποίος παρέχει υποστήριξη για λειτουργίες όπως Java Servlets, Enterprise JavaBeans (EJB), JPA κ.λ.π. Αποτελεί δηλαδή, λογισμικό το οποίο παρέχει την κατάλληλη πλατφόρμα για την ανάπτυξη εφαρμογών που απαιτούν εξειδικευμένες δυνατότητες, όπως είναι η λειτουργία εξυπηρετητή Βάσης Δεδομένων και η χρήση JPAs, όπως περιγράφεται στη συνέχεια. Το GlassFish υποστηρίζεται σήμερα από τον ανεξάρτητο μη κερδοσκοπικό φορέα Eclipse Foundation, ο οποίος υποστηρίζει την ανάπτυξη λογισμικού ανοικτού κώδικα, όπως το Eclipse IDE. Αρχικά ο GlassFish αναπτύχθηκε από την Sun Microsystems και στη συνέχεια πέρασε για ένα διάστημα στην Oracle. Αποτελεί ανοικτό λογισμικό με διπλή άδεια, Eclipse Public License (EPL) και GNU General Public License (GPL) και η τρέχουσα έκδοσή του είναι η έκδοση 7 από τον Δεκέμβριο του 2022.

Ένα από τα βασικά συστατικά του GlassFish server αποτελεί το Σύστημα Διαχείρισης Βάσης Δεδομένων (DBMS) Apache Derby. Το DBMS Apache Derby αρχικά αναπτύχθηκε ως IBM Cloud scape και αποτελεί μία σχεσιακή βάση δεδομένων (Relational DBMS) που υποστηρίζεται από το Apache Software Foundation. Για κάποιο διάστημα η Oracle κυκλοφόρησε και τη δική της έκδοση του Apache Derby με το όνομα Java DB.

Με τη βοήθεια του Apache Derby, μπορεί μία εφαρμογή Java να υποστηρίζει συναλλαγές, αποθηκεύοντας τα δεδομένα που παράγονται σε μία βάση. Η Javaβέβαια υποστηρίζει ούτως ή άλλως τη διασύνδεση και με άλλα Συστήματα Διαχείρισης Βάσης Δεδομένων, όπως είναι για παράδειγμα η MySQL, η Oracle DB, η PostgreSQL, η SQL server κ.λ.π. Για τη διασύνδεση αυτή απαιτείται η χρήση των κατάλληλων οδηγών JDBC (Java Data Base Connectivity drivers). Η χρήση όμως του Apache Derby προσφέρει επιπλέον δυνατότητες και αυτονομία. Το περιβάλλον του NetBeans μάλιστα, προσφέρει τη δυνατότητα διαχείρισης των βάσεων δεδομένων του Apache Derby με γραφικό τρόπο, όπως φαίνεται στην Εικόνα 3.8. Αυτός ήταν ο λόγος που επιλέχθηκε το σύστημα Apache Derby για τον χειρισμό των δεδομένων.



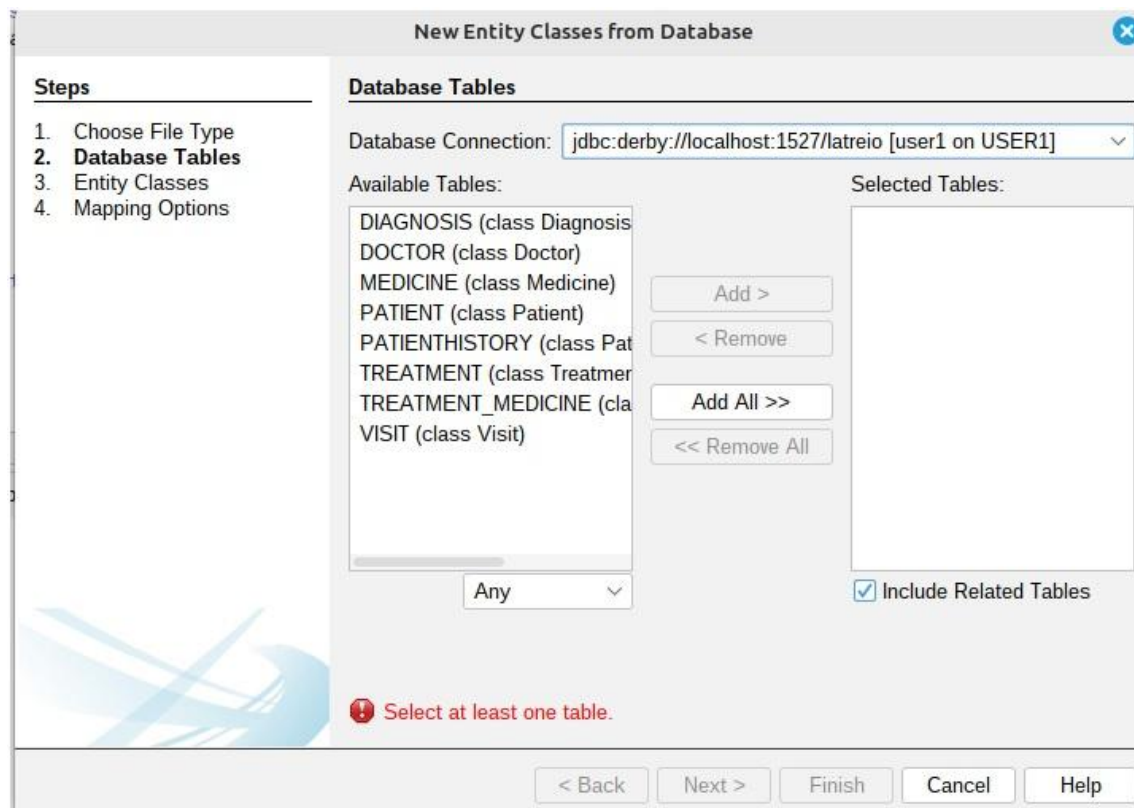
Εικόνα 3.8 Διαχείριση Apache Derby μέσα από το NetBeans

Για την αποθήκευση δεδομένων σε μία βάση και τη διαχείρισή τους, μπορεί να χρησιμοποιηθεί η διασύνδεση (API) Jakarta Persistence, γνωστή και ως JPA (Java Persistence API). Η διεπαφή αυτή η οποία προσφέρει επίσης και τη δυνατότητα χρήσης της γλώσσας Jakarta Persistence Query Language (JPQL) για τη διαχείριση των δεδομένων. Η τεχνολογία JPA κυκλοφόρησε αρχικά τον Μάιο του 2006 και η τρέχουσα έκδοσή της είναι η έκδοση 3.1 από το 2022. Με τη χρήση της τεχνολογίας JPA δημιουργούνται κάποιες Java κλάσεις οντοτήτων (entity classes) οι οποίες, με τη βοήθεια κάποιων μεταδεδομένων, αντιστοιχούν τις ιδιότητες μίας κλάσης της Java στα αντίστοιχα πεδία (γνωρίσματα), όπως αυτά είναι αποθηκευμένα σε μία σχεσιακή βάση δεδομένων, όπως είναι η Apache Derby. Το περιβάλλον του Apache NetBeans δίνει τη δυνατότητα για τη δημιουργία των κλάσεων οντοτήτων (entity classes) με αυτόματο τρόπο, όπως φαίνεται στην Εικόνα 3.9.

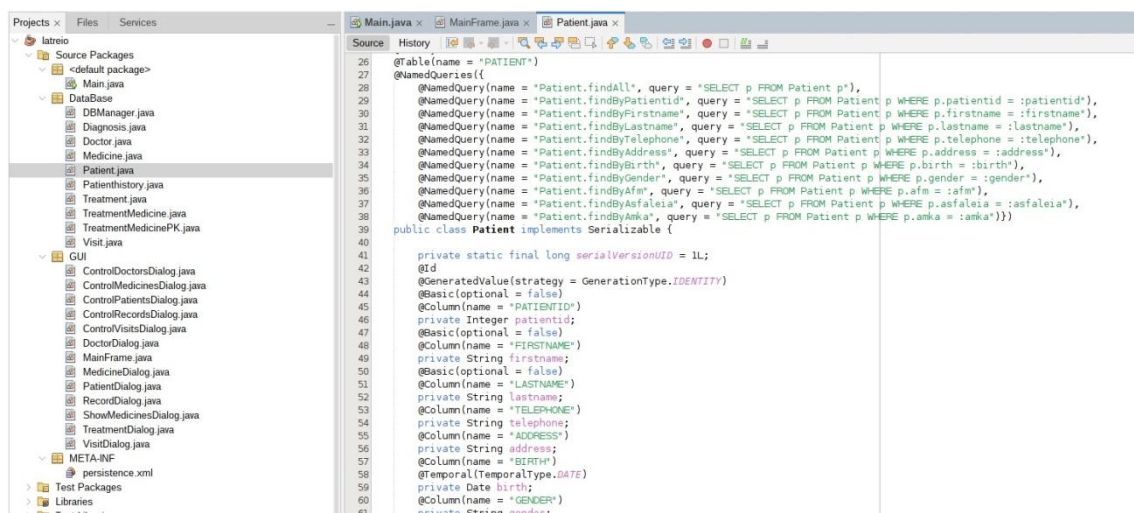
Κατά τη δημιουργία των entity κλάσεων, δημιουργείται μία Javakλάση για κάθε μία από τις σχέσεις που αποθηκεύονται στη βάση δεδομένων Derby.

Για παράδειγμα, στην Εικόνα 3.10 βλέπουμε ένα τμήμα της κλάσης “Patient”, όπου φαίνονται κάποιες από τις ιδιότητες της κλάσης, όπως αυτές αντιστοιχούν στα πεδία της αντίστοιχης σχέσης στη βάση δεδομένων (σχέση ΑΣΘΕΝΗΣ). Στο επάνω μέρος της εικόνας φαίνονται μάλιστα κάποια έτοιμα ερωτήματα (Named Queries), τα οποία

δημιουργούνται επίσης με αυτόματο τρόπο και προσφέρουν τη δυνατότητα της εύκολης δημιουργίας ερωτημάτων SQL για τη διασύνδεση με τη βάση δεδομένων.



Εικόνα 3.9 Δημιουργία κλάσεων οντοτήτων (entity classes)



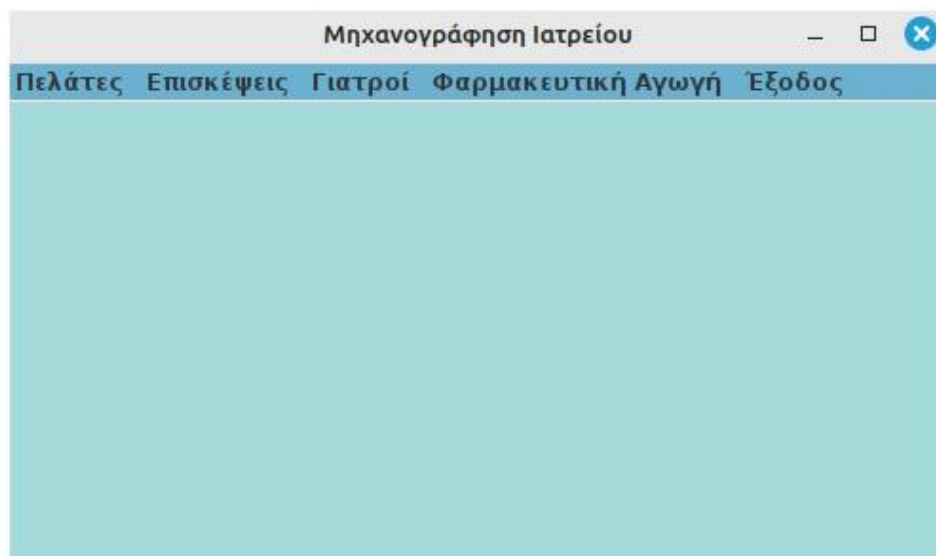
Εικόνα 3.10 Η entity class Patient

4. Παρουσίαση της εφαρμογής μηχανογράφησης Ιατρείου

Στο κεφάλαιο αυτό παρουσιάζονται οι λειτουργίες που προσφέρει η εφαρμογή μηχανογράφησης Ιατρείου η οποία αναπτύχθηκε στα πλαίσια της παρούσας πτυχιακής εργασίας. Η εφαρμογή προσφέρει τις βασικές λειτουργίες, όπως αναλύθηκαν στις προηγούμενες παραγράφους. Διαθέτει ένα φιλικό και λειτουργικό προς την χρήση περιβάλλον και χρησιμοποιούνται ευδιάκριτες οπτικές απεικονίσεις που βοηθούν τον χρήστη της εφαρμογής.

4.1. Λειτουργίες της εφαρμογής

Η κεντρική οθόνη της εφαρμογής (Εικόνα 4.1) προσφέρει ένα μενού στο επάνω μέρος της. Από αυτό ο χρήστης μπορεί να καλεί τις κατάλληλες λειτουργίες της εφαρμογής.



Εικόνα 4.1 Η κεντρική οθόνη της εφαρμογής

Το κεντρικό μενού έχει πέντε κατηγορίες (υπο-μενού): **Πελάτες**, **Επισκέψεις**, **Γιατροί**, **Φαρμακευτική Αγωγή** και **Έξοδος**. Μέσα από καθεμία κατηγορία ο χρήστης μπορεί να εκτελέσει τα παρακάτω:

Μενού «Πελάτες». Σ' αυτήν την κατηγορία ο χρήστης μπορεί: να προσθέσει νέους πελάτες, να προβάλει τα στοιχεία των υπαρχόντων πελατών, να τροποποιήσει τα στοιχεία των πελατών που επιθυμεί, να διαγράψει πελάτες.

Μενού «Επισκέψεις». Σ' αυτή την κατηγορία έχουμε λειτουργίες: για το κλείσιμο νέων ραντεβού με τους πελάτες, την καταχώρηση και τροποποίηση των στοιχείων των επισκέψεων που πραγματοποιούν οι πελάτες, την ακύρωση ραντεβού, την εισαγωγή πληροφοριών που αφορούν την επίσκεψη ενός ασθενούς-πελάτη.

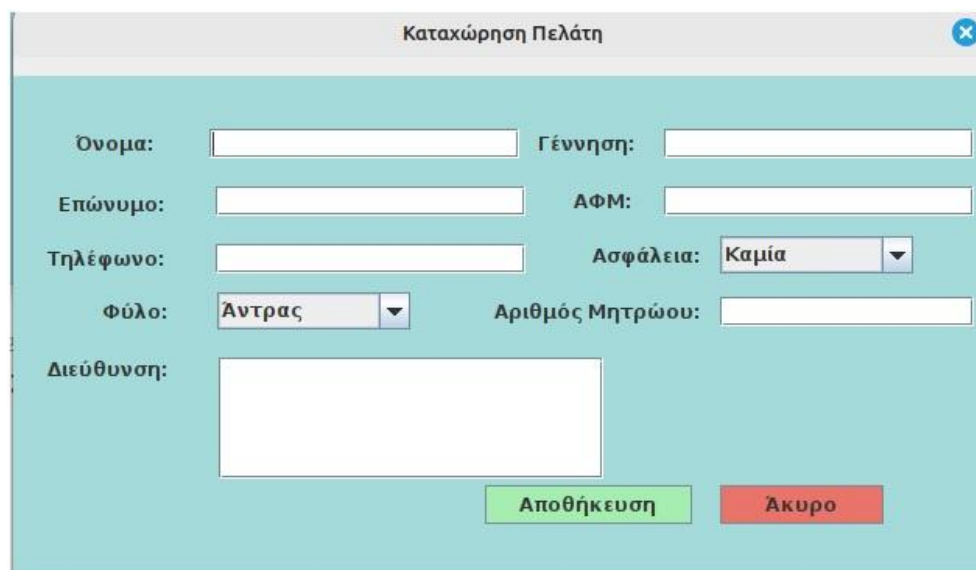
Μενού Γιατροί». Σ' αυτήν την κατηγορία ο χρήστης μπορεί: να προσθέσει νέους γιατρούς οι οποίοι συνεργάζονται με το Ιατρείο, να προβάλει τα στοιχεία των υπαρχόντων ιατρών, να τροποποιήσει τα στοιχεία τους, να διαγράψει γιατρούς.

Μενού «Φαρμακευτική Αγωγή». Στην κατηγορία αυτή έχουμε λειτουργίες για την εισαγωγή στοιχείων που αφορούν στις φαρμακευτικές αγωγές των ασθενών, δηλαδή την εισαγωγή τους, την τροποποίησή τους και τη διαγραφή τους.

Μενού «Έξοδος». Η επιλογή αυτή δίνει τη δυνατότητα στον χρήστη να κλείσει την εφαρμογή. Τα δεδομένα της εφαρμογής αποθηκεύονται αυτόματα στη βάση δεδομένων και είναι διαθέσιμα κατά το επόμενο άνοιγμα της εφαρμογής.

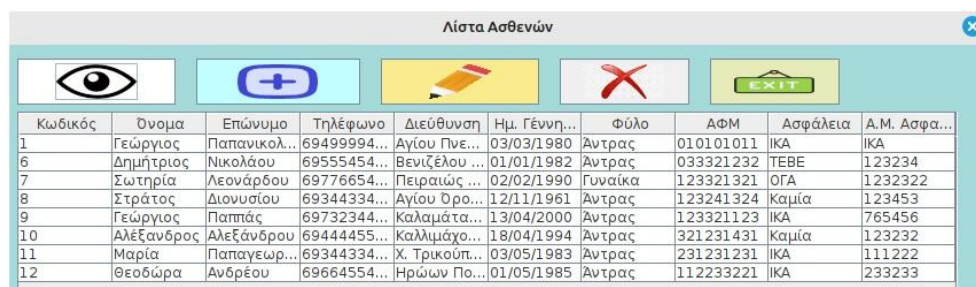
4.2. Λειτουργίες που αφορούν στους πελάτες-ασθενείς του Ιατρείου

Στην κατηγορία «Πελάτες» της εφαρμογής δίνεται η δυνατότητα στον χρήστη να εισάγει νέους πελάτες επιλέγοντας «Καταχώρηση Πελάτη». Τότε, εμφανίζεται η οθόνη **Καταχώρησης νέου πελάτη** (Εικόνα 4.2), όπου ο χρήστης μπορεί να εισάγει τα προσωπικά στοιχεία του νέου πελάτη - ασθενή. Μετά την εισαγωγή των στοιχείων ο χρήστης πρέπει να πατήσει το κουμπί «Αποθήκευση» για να αποθηκευτούν τα δεδομένα. Επίσης, του δίνεται η επιλογή ακύρωσης και επιστροφής στην κεντρική οθόνη της εφαρμογής πατώντας το κουμπάκι «Άκυρο»



Εικόνα 4.2 Καταχώρηση νέου πελάτη

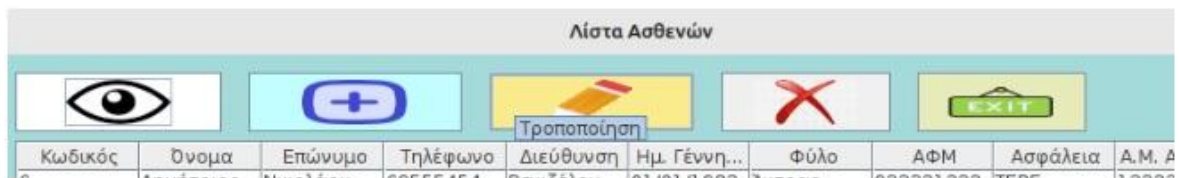
Μέσω της επιλογής «Προβολή Πελατών» ο χρήστης μπορεί να προβάλει τα στοιχεία όλων των πελατών που έχουν αποθηκευτεί στο σύστημα (Εικόνα 4.3). Στο παράθυρο που εμφανίζεται προβάλλονται τα στοιχεία όλων των πελατών που έχουν καταχωρηθεί.



Κωδικός	Όνομα	Επώνυμο	Τηλέφωνο	Διεύθυνση	Ημ. Γέννη...	Φύλο	ΑΦΜ	Ασφάλεια	Α.Μ. Ασφα...
1	Γεώργιος	Παπανικολ...	69499994...	Αγίου Πνε...	03/03/1980	Αντρας	010101011	ΙΚΑ	ΙΚΑ
6	Δημήτριος	Νικολάου	69555454...	Βενιζέλου ...	01/01/1982	Αντρας	033321232	ΤΕΒΕ	123234
7	Σωτηρία	Λεονάρδου	69776654...	Πειραιώς ...	02/02/1990	Γυναίκα	123321321	ΟΓΑ	1232322
8	Στράτος	Διονυσίου	69344334...	Αγίου Όρο...	12/11/1961	Αντρας	123241324	Καμία	123453
9	Γεώργιος	Παππάς	69732344...	Καλλιμάχο...	13/04/2000	Αντρας	123321123	ΙΚΑ	765456
10	Αλέξανδρος	Αλεξάνδρου	69444455...	Καλλιμάχο...	18/04/1994	Αντρας	321231431	Καμία	123232
11	Μαρία	Παπαγεωρ...	69344334...	Χ. Τρικούπ...	03/05/1983	Αντρας	231231231	ΙΚΑ	111222
12	Θεοδώρα	Ανδρέου	69664554...	Ηρώων Πο...	01/05/1985	Αντρας	112233221	ΙΚΑ	233233

Εικόνα 4.3 Προβολή λίστας πελατών

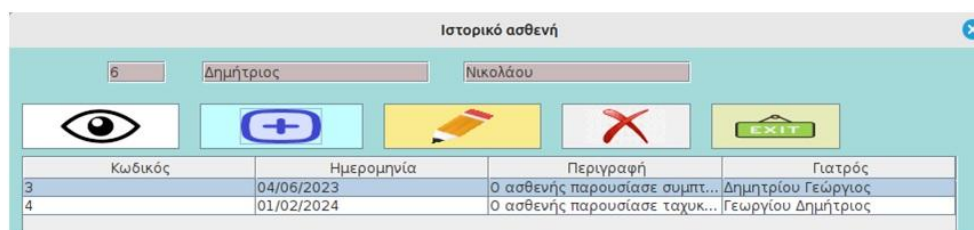
Στο επάνω μέρος υπάρχει μία σειρά από εικονίδια με τη βοήθεια των οποίων μπορούν να εκτελεστούν περισσότερες λειτουργίες. Όταν μείνει για λίγα δευτερόλεπτα το ποντίκι πάνω από ένα εικονίδιο, εμφανίζεται μία περιγραφή για το εικονίδιο αυτό, όπως για παράδειγμα φαίνεται στην Εικόνα 4.4 για το εικονίδιο «Τροποποίηση».



Εικόνα 4.4 Εμφάνιση περιγραφής στα εικονίδια

Με το πρώτο εικονίδιο «Προβολή» προβάλλονται αναλυτικά τα στοιχεία του πελάτη που έχει επιλέγει, (Εικόνα 4.5) **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε..** Πατώντας το κουμπάκι «Εμφάνιση Ιστορικού» μπορεί να προβάλει το ιστορικό ενός ασθενούς (Εικόνα 4.6). Το παράθυρο που εμφανίζεται επιτρέπει την εισαγωγή νέας εγγραφής στο ιστορικού του πελάτη και την τροποποίηση ή διαγραφή παλαιότερων εγγραφών.

Εικόνα 4.5 Προβολή στοιχείων πελάτη



Εικόνα 4.6 Προβολή ιστορικού Ασθενή

Με το δεύτερο εικονίδιο «**Νέα Εγγραφή**» του παραθύρου προβολής στοιχείων πελατών εμφανίζεται το παράθυρο για την καταχώρηση ενός νέου πελάτη (Εικόνα 4.2).

Με το τρίτο εικονίδιο «**Τροποποίηση**» εμφανίζεται νέο παράθυρο με τα στοιχεία του επιλεγμένου πελάτη, τα οποία όμως ο χρήστης έχει τη δυνατότητα να τα τροποποιήσει αν το επιθυμεί και να αποθηκεύσει τις αλλαγές (Εικόνα 4.7).

Όνομα: Γέννηση:

Επώνυμο: ΑΦΜ:

Τηλέφωνο: Ασφάλεια:

Φύλο: Αριθμός Μητρώου:

Διεύθυνση:

Εικόνα 4.7 Τροποποίηση των στοιχείων ενός πελάτη

Με το τέταρτο εικονίδιο «**Διαγραφή**» μπορεί ο χρήστης να διαγράψει έναν πελάτη από το σύστημα, αφού πρώτα προβληθούν τα στοιχεία του έτσι ώστε να σιγουρευτεί ότι θέλει να προχωρήσει στη διαγραφή. Τέλος, με το πέμπτο εικονίδιο «**Έξοδος**» η εφαρμογή επιστρέφει στην κεντρική οθόνη της.

4.3. Λειτουργίες που αφορούν στα στοιχεία των Γιατρών

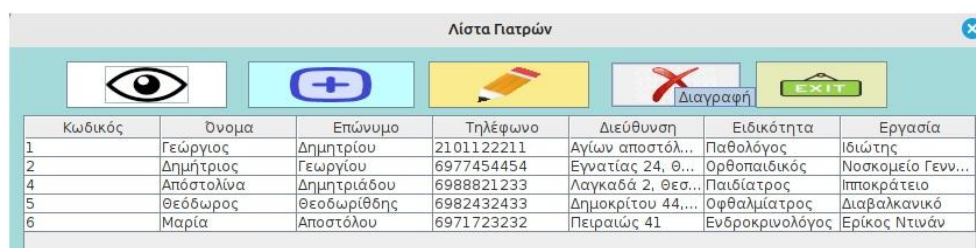
Η εφαρμογή επιτρέπει την καταχώρηση των στοιχείων τα οποία αφορούν στους γιατρούς οι οποίοι συνεργάζονται με το Ιατρείο και οι οποίοι αναλαμβάνουν να εξετάσουν τους πελάτες - ασθενείς κατά τις επισκέψεις που πραγματοποιούν. Στην κεντρική οθόνη της εφαρμογής υπάρχει η κατηγορία «**Γιατροί**», στο κεντρικό μενού, με τη βοήθεια της οποίας εκτελούνται οι λειτουργίες που αφορούν τα στοιχεία των γιατρών.

Με την επιλογή «**Καταχώρηση Γιατρού**» μπορεί να γίνει εισαγωγή των στοιχείων ενός νέου γιατρού που καταχωρείται στο σύστημα (Εικόνα 4.8). Πέρα από το ονοματεπώνυμο, το τηλέφωνο και τη διεύθυνση του γιατρού, ο χρήστης μπορεί να εισάγει την ειδικότητά του, καθώς και κάποια θέση εργασίας που ήδη κατέχει ο γιατρός σε κάποιον άλλο φορέα. Με την επιλογή «**Προβολή Γιατρών**» εμφανίζεται σε νέο παράθυρο ένας πίνακας ο οποίος περιλαμβάνει τα στοιχεία των καταχωρημένων γιατρών (Εικόνα 4.9).

Πάνω από τον πίνακα με τα στοιχεία των γιατρών υπάρχει μία σειρά από εικονίδια, τα οποία επιτρέπουν επιπλέον λειτουργίες όπως την προβολή των στοιχείων ενός γιατρού που έχει επιλεγεί, την προσθήκη νέου γιατρού, την τροποποίηση των στοιχείων ενός γιατρού ή την διαγραφή ενός γιατρού από το σύστημα.



Εικόνα 4.8 Καταχώρηση νέου γιατρού

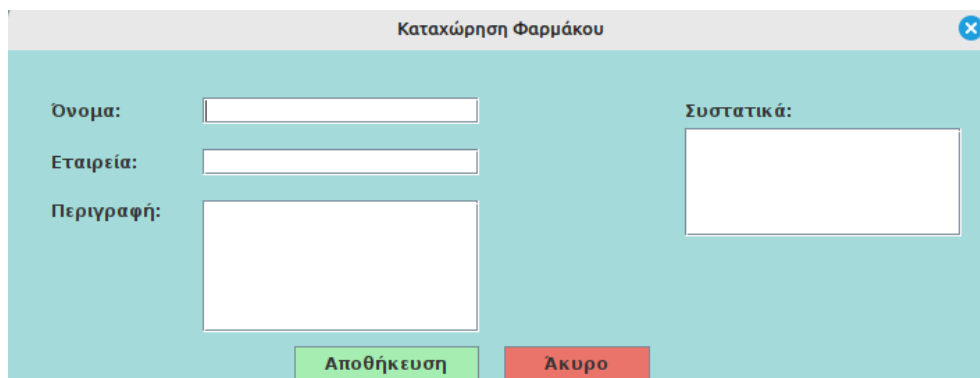


Κωδικός	Όνομα	Επώνυμο	Τηλέφωνο	Διεύθυνση	Ειδικότητα	Εργασία
1	Γεώργιος	Δημητρίου	2101122211	Αγίων αποστόλ...	Παθολόγος	Ιδιώτης
2	Δημήτριος	Γεωργίου	6977454454	Εγνατίας 24, Θ...	Ορθοπαιδικός	Νοσοκομείο Γενν...
4	Απόστολίνα	Δημητριάδου	6988821233	Λαγκαδά 2, Θεσ...	Παιδίατρος	Ιπποκράτειο
5	Θεόδωρος	Θεοδωρίδης	6982432433	Δημοκρίτου 44...	Οφθαλμίατρος	Διαβαλκανικό
6	Μαρία	Αποστόλου	6971723232	Πειραιώς 41	Ενδοκρινολόγος	Ερίκος Ντινάν

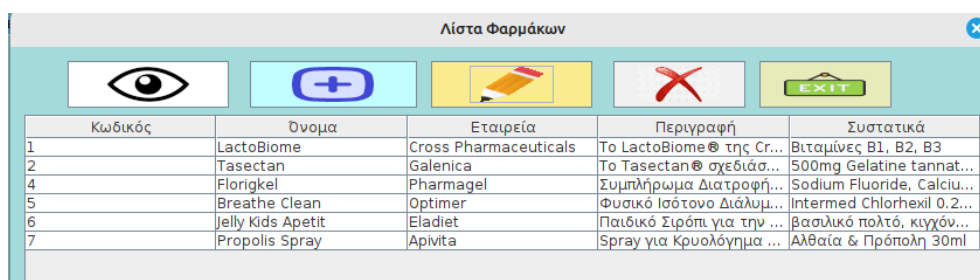
Εικόνα 4.9 Προβολή λίστας με τα στοιχεία των γιατρών

4.4. Λειτουργίες που αφορούν στα στοιχεία των Φαρμακευτικών Αγωγών

Η εφαρμογή δίνει τη δυνατότητα της καταχώρησης στοιχείων που αφορούν φάρμακα τα οποία συνταγογραφούν οι γιατροί στους ασθενείς. Στην κεντρική οθόνη της εφαρμογής υπάρχει η κατηγορία «Φαρμακευτική Αγωγή» στο κεντρικό μενού και με τη βοήθεια του μενού αυτού μπορεί να γίνει η εισαγωγή, τροποποίηση ή διαγραφή των στοιχείων των φαρμάκων. Με την επιλογή «Καταχώρηση Φαρμάκου» μπορεί να γίνει εισαγωγή των δεδομένων που αφορούν ένα νέο φάρμακο (Εικόνα 4.10). Στη φόρμα που εμφανίζεται, μπορεί ο χρήστης να εισάγει το όνομα του φαρμάκου, όπως και την εταιρεία που το παρασκευάζει. Μπορεί επίσης αν το επιθυμεί να εισάγει μία περιγραφή για το φάρμακο και τα συστατικά από τα οποία αποτελείται. Με την επιλογή «Προβολή Φαρμάκων», εμφανίζεται ένα παράθυρο με μία λίστα με τα φάρμακα που έχουν καταχωρηθεί (Εικόνα 4.11). Πάνω από τον πίνακα αυτόν εμφανίζεται μία σειρά με εικονίδια με τη βοήθεια των οποίων μπορεί να γίνει η προβολή, η τροποποίηση ή διαγραφή ενός φαρμάκου. Στην Εικόνα 4.12 εμφανίζεται η δυνατότητα για την τροποποίηση των στοιχείων ενός φαρμάκου που έχει επιλεγεί από τον πίνακα.



Εικόνα 4.10 Καταχώρηση νέου φαρμάκου



Κωδικός	Όνομα	Εταιρεία	Περιγραφή	Συστατικά
1	LactoBiome	Cross Pharmaceuticals	To LactoBiome® της Cr...	Βιταμίνες B1, B2, B3
2	Tasectan	Galenica	To Tasectan® σχεδιάσ...	500mg Gelatine tannat...
4	Florigkel	Pharmagel	Συμπλήρωμα Διατροφή...	Sodium Fluoride, Calciu...
5	Breathe Clean	Optimer	Φυσικό Ισότονο Διάλυμ...	Intermed Chlorhexil 0.2...
6	Jelly Kids Appetit	Eladiet	Παιδικό Σιρόπι για την ...	Βασιλικό πολτό, κυγχόν...
7	Propolis Spray	Apivita	Spray για Κρυολόγημα ...	Αλθαία & Πρόπολη 30ml

Εικόνα 4.11 Πίνακας με τα φάρμακα που έχουν καταχωρηθεί

Εικόνα 4.12 Τροποποίηση στοιχείων φαρμάκου

4.5. Λειτουργίες που αφορούν στα Ραντεβού και στις Επισκέψεις των πελατών

Από το μενού «**Επισκέψεις**» στην κεντρική οθόνη της εφαρμογής δίνεται η δυνατότητα στον χρήστη, μέσω της επιλογής «**Καταχώρηση νέου Ραντεβού**», να κλείσει νέα ραντεβού σε κάποιον πελάτη με κάποιον γιατρό (Εικόνα 4.13). Ο χρήστης μπορεί να επιλέξει τον πελάτη από την πτυσσόμενη λίστα που περιλαμβάνει τους πελάτες που έχουν καταχωρηθεί στο σύστημα, όπως και τον γιατρό με τον οποίο κλείνεται το ραντεβού, από την αντίστοιχη πτυσσόμενη λίστα για τους γιατρούς. Ο χρήστης θα πρέπει επίσης να εισάγει την ημερομηνία και την ώρα που έχει κλείσει το ραντεβού σε αποδεκτή μορφή ημερομηνίας και ώρας, διαφορετικά εμφανίζεται μήνυμα λάθους. Όταν πραγματοποιηθεί η επίσκεψη για κάποιο ραντεβού, τότε αλλάζει η κατάσταση στην αντίστοιχη λίστα από «**Ραντεβού**» σε «**Επίσκεψη**». Επίσης, υπάρχει η δυνατότητα ακύρωσης ένα ραντεβού, αλλάζοντας την κατάστασή του σε «**Άκυρο**».

Με την επιλογή «**Προβολή Ραντεβού**» εμφανίζονται τα ραντεβού τα οποία έχουν κλειστεί και δεν έχουν πραγματοποιηθεί ακόμη, (Εικόνα 4.14). Στο παράθυρο αυτό υπάρχουν εικονίδια με τα οποία μπορεί να γίνει η προβολή, επεξεργασία ή διαγραφή ενός ραντεβού. Κατά την επεξεργασία ενός ραντεβού, μπορεί να γίνει αλλαγή της κατάστασής του, να γίνει εισαγωγή της διάγνωσης του γιατρού, του ποσού πληρωμής της επίσκεψης κ.ά. (Εικόνα 4.15). Στο παράθυρο αυτό υπάρχουν και κουμπάκια με τη βοήθεια των οποίων μπορεί να καταγραφεί μία περιγραφή για κάποια προτεινόμενη φαρμακευτική αγωγή που θα πρέπει να ακολουθήσει ο πελάτης-ασθενής.

Καταχώρηση Επίσκεψης

Πελάτης: 6-Νικολάου Δημήτριος ΓΙΑΤΡΟΣ: 1-Δημητρίου Γεώργιος

Ημερομηνία: Ώρα: Διάγνωση:

Ραντεβού: Ποσό: ☐ Πληρωμή Περιγραφή:

Συμπτώματα:

Προβολή Δημιουργία Τροποποίηση Διαγραφή

Αγωγή

Αποθήκευση Ακύρο

Εικόνα 4.13 Κλείσιμο νέου ραντεβού

Λίστα Ραντεβού

Ημερομηνία	Ώρα	Τύπος	Ασθενής	Γιατρός	Ποσό	Πληρωμή
04/06/2024	18:00	Ραντεβού	Διονυσίου Στρά...	Δημητριάδου Α...	0.00	0
05/06/2024	9:00	Ραντεβού	Παπαγεωργίου ...	Αποστόλου Μαρία	0.00	0
06/06/2024	11:00	Ραντεβού	Παπαγεωργίου ...	Δημητρίου Γεώ...	0.00	0
07/06/2024	13:15	Ραντεβού	Ανδρέου Θεοδώ...	Αποστόλου Μαρία	0.00	0

Εικόνα 4.14 Λίστα με τα κλεισμένα ραντεβού

Τροποποίηση επίσκεψης

Πελάτης: 12-Ανδρέου Θεοδώρα ΓΙΑΤΡΟΣ: 1-Δημητρίου Γεώργιος

Ημερομηνία: 03/03/2023 Ώρα: 17:15 Διάγνωση: Έλλειψη Σιδήρου

Επίσκεψη: Περιγραφή: Αδυναμία που οφείλεται σε έλλειψη σιδήρου στ

Ποσό: 50.0 ☒ Πληρωμή Συμπτώματα: Αδυναμία, κόπωση, όχι όρεξη για εργασία.

Προβολή Δημιουργία Τροποποίηση Διαγραφή

Αγωγή

Αποθήκευση Ακύρο

Εικόνα 4.15 Επεξεργασία ενός ραντεβού

5. Επίλογος και συμπεράσματα

Οι ηλεκτρονικοί υπολογιστές (H/Y) έχουν μπει πλέον για τα καλά σε όλους τους τομείς της ζωής μας και έχουν επηρεάσει σε πολύ μεγάλο βαθμό τόσο την καθημερινή μας ζωή όσο και το σύνολο των επαγγελματικών μας δραστηριοτήτων. Οι σύγχρονες επιχειρήσεις βασίζονται στη λειτουργία τους στη χρήση των ηλεκτρονικών υπολογιστών και η μηχανογράφηση μίας επιχείρησης αποτελεί βασική προϋπόθεση για τον εκσυγχρονισμό της και τη βελτίωση των υπηρεσιών που παρέχει στους πελάτες. Οι λειτουργίες που αφορούν την καταγραφή και επεξεργασία των δεδομένων που διαχειρίζεται μία επιχείρηση θα πρέπει να σχεδιαστούν με πολύ προσεκτικό τρόπο, έτσι ώστε να ικανοποιούνται τόσο οι λειτουργικές όσο και οι μη λειτουργικές απαιτήσεις που έχουν καταγραφεί κατά την ανάλυση του συστήματος μηχανογράφησης. Με τον τρόπο αυτό ελαχιστοποιούνται τα σφάλματα που μπορούν να συμβούν κατά τη λειτουργία της επιχείρησης και βελτιώνεται η ποιότητα των παρεχόμενων υπηρεσιών.

Η παρούσα πτυχιακή εργασία παρουσιάζει την ανάλυση και τη σχεδίαση ενός πληροφοριακού συστήματος που αφορά την μηχανογράφηση ενός Ιατρείου. Στα πλαίσια της εργασίας, αναπτύχθηκε επίσης μία εφαρμογή η οποία καλύπτει κάποιες από τις βασικές ανάγκες και λειτουργίες ενός σύγχρονου Ιατρείου. Η εφαρμογή αυτή αναπτύχθηκε με τη βοήθεια της γλώσσας προγραμματισμού Java και τις αρχές του Αντικειμενοστρεφούς σχεδίασης. Η διαχείριση του αρχείου των ασθενών γίνεται με τη βοήθεια του λογισμικού Apache Derby (Java DB), του λογισμικού δηλαδή που προσφέρει η Java EE για τη διαχείριση συστημάτων Βάσεων Δεδομένων. Η σχεδίαση και η υλοποίηση του συστήματος διαχείρισης ενός Ιατρείου περιγράφονται αναλυτικά και στη συνέχεια παρουσιάζεται η εφαρμογή με τη βοήθεια κάποιων καταγραφών της οθόνης (screenshots).

Από την εκπόνηση της εργασίας μπορεί να βγει το συμπέρασμα ότι η Αντικειμενοστρεφής μεθοδολογία προσφέρει όλα εκείνα εργαλεία με τα οποία μπορεί να σχεδιαστεί η λειτουργία μίας σύγχρονης επιχείρησης, όπως για παράδειγμα η λειτουργία ενός Ιατρείου. Η χρήση της γλώσσας προγραμματισμού Java μπορεί να οδηγήσει σε ένα σωστά σχεδιασμένο σύστημα, όπου έχουμε τις αντίστοιχες κλάσεις για τις βασικές οντότητες του συστήματος, οι οποίες αλληλεπιδρούν μέσω των μεθόδων που προσφέρουν. Ταυτόχρονα, μπορούν να χρησιμοποιηθούν και οι κατάλληλες τεχνολογίες για την αποθήκευση των

δεδομένων και τη διαχείρισή τους με αποτελεσματικό τρόπο. Μία εφαρμογή Java που αναπτύσσεται με τον τρόπο αυτό είναι απολύτως μεταφέρσιμη και μπορεί να εκτελεστεί με πανομοιότυπο τρόπο σε οποιαδήποτε πλατφόρμα χρησιμοποιείται από ένα πληροφοριακό σύστημα.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- Βασιλακόπουλος Γ. (2018). *Πληροφοριακά Συστήματα*. Εκδόσεις Τσότρα.
- Βεσδούκης, Β. (2000). *Τεχνολογία λογισμικού Ι*. Ελληνικό Ανοικτό Πανεπιστήμιο. ISBN 960-538-097-8
- Κιουντούζης, Ε. (2002). *Μεθοδολογίες ανάλυσης και σχεδιασμού πληροφοριακών συστημάτων*. Εκδόσεις Ε. Μπένου.
- Πιερρακέας, Χ. (2019–2020). *Τεχνολογία πληροφοριακών συστημάτων – Μεθοδολογίες, μοντέλα, μοντελοποίηση συστημάτων* (Μέρος 2ο) (σελ. 9). [Διδακτικό υλικό, ΠΙΜΣ Διοίκηση Εκπαίδευσης, ΤΕΙ Δυτικής Ελλάδας]
- Alter, S. (2003). 18 reasons why IT-reliant work systems should replace “the IT artifact” as the core subject matter of the IS field. *Communications of the Association for Information Systems*, 12(23), 365–394.
- Avison, D., & Fitzgerald, G. (2006). *Ανάπτυξη προηγμένων πληροφοριακών συστημάτων: Μεθοδολογίες & εργαλεία* (1η έκδ.). Εκδόσεις Νέων Τεχνολογιών.
- Bacon, C. J., & Fitzgerald, B. (2001). A systemic framework for the field of information systems. *ACM SIGMIS Database: The DATABASE for Advances in Information Systems*, 32(2), 46–67.
- Cockburn, A. (2001). *Writing effective use cases*. Addison-Wesley. ISBN 978-0201702255
- Dennis, A. (2012). *Systems analysis and design* (5th ed.). Wiley. ISBN 978-1118057629
- JRebel. (2023). *2023 Java developer productivity report*. Ανακτήθηκε από <https://www.jrebel.com/resources/java-developer-productivity-report-2023>
- Kendall, K. E., & Kendall, J. E. (2013). *Systems analysis and design* (9th ed.). Prentice Hall. ISBN 978-0133023442
- Liskov, B., & Guttag, J. (2000). *Program development in Java: Abstraction, specification, and object-oriented design*. Addison-Wesley. ISBN 978-0201657685
- Miles, R., & Hamilton, K. (2006). *Learning UML 2.0*. O'Reilly. ISBN 978-0596009823
- Oracle. (2023, April). *Sun Java System Application Server Platform Edition 8 Update 1 Release Notes*. Ανακτήθηκε από <https://docs.oracle.com>
- Pfleeger, S. L. (2003). *Τεχνολογία λογισμικού: Θεωρία και πράξη* (2nd ed.). Εκδόσεις Κλειδάριθμος.
- Piccoli, G., & Pigni, F. (2018). *Information systems for managers: With cases* (4th ed.). Prospect Press. ISBN 978-1-943153-50-3
- PYPL. (2024). *PopularitY of Programming Language*. Ανακτήθηκε από <https://pypl.github.io/PYPL.html>
- Statista. (2023). *Most used programming languages among developers worldwide as of 2023*. Ανακτήθηκε από <https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/>
- TIOBE. (2024, May). *TIOBE index for May 2024*. Ανακτήθηκε από <https://www.tiobe.com/tiobe-index/>
- Wieggers, K., & Beatty, J. (2013). *Software requirements* (3rd ed.). Microsoft Press. ISBN 978-0-7356-7966-5

Wikipedia. (2024, April 10). *Apache Derby*. Ανακτήθηκε από https://en.wikipedia.org/wiki/Apache_Derby

