



## Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Πληροφορική»

### Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	<b>Σύγχρονη Προσέγγιση Διαχείρισης Ιατρικών Δεδομένων</b>
Όνοματεπώνυμο Φοιτητή	<b>Παναγιώτης Κοκκότης</b>
Πατρώνυμο	<b>Φώτιος</b>
Αριθμός Μητρώου	<b>ΜΠΠΛ/ 09069</b>
Επιβλέπουσα	<b>Μαρία Βίβου, Καθηγήτρια</b>

---

**Τριμελής Εξεταστική Επιτροπή**

(υπογραφή)

(υπογραφή)

(υπογραφή)

Μαρία  
Καθηγήτρια

Βίρβου  
Γεώργιος  
Καθηγητής

Τσιχριντζής  
Ευάγγελος  
Καθηγητής

Φούντας

Ευχαριστώ θερμά τους καθηγητές μου για τη συμβολή τους στο εξαιρετικό Πρόγραμμα Μεταπτυχιακών Σπουδών Πληροφορικής του Πανεπιστημίου Πειραιώς, το οποίο με βοήθησε να γνωρίσω νέες τεχνολογίες και να επεκτείνω τις γνώσεις μου σε πολύ μεγαλύτερο βαθμό απ' ό,τι είχα αρχικά φανταστεί. Θεωρώ τις σπουδές μου στο Τμήμα ως το σπουδαιότερο εφόδιο που έχω αποκομίσει μέχρι σήμερα και το σημαντικότερο σκαλοπάτι για την περαιτέρω σταδιοδρομία μου. Θέλω να τους συγχαρώ για την μεγάλη προσπάθεια που γίνεται στα πλαίσια του Προγράμματος. Επίσης, θέλω να ευχαριστήσω την οικογένειά μου και κυρίως την Ακριβή για την ολόψυχη στήριξη και τη συμβολή τους στην πραγματοποίηση των στόχων μου. Επιπλέον θέλω να ευχαριστήσω την Ηρώ για την συμπαράσταση και για την πίστη της ότι σίγουρα κάποτε θα καταφέρω αυτά που επιθυμώ.

## Περιεχόμενα

1. ΕΙΣΑΓΩΓΗ.....	1
ΠΕΡΙΛΗΨΗ .....	1
ABSTRACT .....	1
ΣΚΟΠΟΣ ΤΗΣ ΕΡΓΑΣΙΑΣ.....	2
ΙΑΤΡΙΚΗ ΠΛΗΡΟΦΟΡΙΚΗ .....	3
ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ ΥΓΕΙΑΣ.....	4
ΓΕΝΙΚΑ .....	4
ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΠΣΝ .....	5
ΠΣΝ ΣΤΗΝ ΕΛΛΑΔΑ .....	5
ΓΙΑΤΙ JAVA;.....	6
2. ΕΠΙΣΚΟΠΗΣΗ ΠΕΔΙΟΥ .....	10
TECHNICON MEDICAL INFORMATION SYSTEM .....	11
COMPUTER STORED AMBULATORY RECORD (COSTAR) SYSTEM .....	12
PROBLEM-ORIENTED MEDICAL INFORMATION SYSTEM (PROMIS) .....	15
HELP (HEALTH EVALUATION THROUGH LOGICAL PROCESSING) .....	16
ΆΛΛΑ ΣΥΣΤΗΜΑΤΑ.....	17
ΠΛΕΟΝΕΚΤΗΜΑΤΑ.....	18
ΣΥΓΚΡΙΣΗ ΜΕ ΤΟ ΠΑΡΟΝ ΣΥΣΤΗΜΑ.....	18
3. ΑΝΑΛΥΣΗ – ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ .....	21
UML.....	21
A) ΔΙΑΓΡΑΜΜΑΤΑ ΠΕΡΙΠΤΩΣΕΩΝ ΧΡΗΣΗΣ.....	22
B) ΔΙΑΓΡΑΜΜΑΤΑ ΣΕΙΡΑΣ – ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ – ΣΥΝΕΡΓΑΣΙΑΣ .....	24
I) ΚΑΤΑΓΡΑΦΗ.....	25
II) ΕΠΕΞΕΡΓΑΣΙΑ.....	27
III) ΕΠΙΣΚΟΠΗΣΗ .....	28
IV) ΑΝΑΖΗΤΗΣΗ .....	30
V) ΔΙΑΓΡΑΦΗ.....	32
C) ΔΙΑΓΡΑΜΜΑ ΕΞΑΡΤΗΜΑΤΩΝ.....	34
4. ΠΑΡΟΥΣΙΑΣΗ ΣΥΣΤΗΜΑΤΟΣ.....	36
Η ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ .....	37
ΤΟ SERVLET.....	40
ΤΟ ΛΟΓΙΣΜΙΚΟ .....	44

ΟΙ 5 ΒΑΣΙΚΕΣ ΛΕΙΤΟΥΡΓΙΕΣ.....	50
I. ΚΑΤΑΓΡΑΦΗ.....	51
II. ΕΠΕΞΕΡΓΑΣΙΑ.....	53
III. ΕΠΙΣΚΟΠΗΣΗ.....	53
IV. ΑΝΑΖΗΤΗΣΗ.....	54
V. ΔΙΑΓΡΑΦΗ.....	55
MOBILE ΕΦΑΡΜΟΓΗ.....	55
SQL INJECTION.....	62
ΑΞΙΟΛΟΓΗΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ.....	64
ΣΥΝΔΕΣΗ ΜΕ ΔΙΑΚΟΜΙΣΤΗ – DDNS.....	67
ECLIPSE.....	68
5. ΣΥΜΠΕΡΑΣΜΑΤΑ – ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ.....	71
6. ΠΑΡΑΡΤΗΜΑ.....	76
Η ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ.....	76
ΣΤΑΤΙΣΤΙΚΑ.....	84
MEDINFOSYSTEM.....	85
SERVLET.....	86
MEDINFODROID.....	88
ΟΔΗΓΙΕΣ ΕΓΚΑΤΑΣΤΑΣΗΣ.....	90
ΠΕΡΙΕΧΟΜΕΝΑ CD.....	91
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	92

## Εικόνες

Εικόνα 1-1. Java .....	8
Εικόνα 2-1. Technicon Medical Information System (a) .....	11
Εικόνα 2-2. Technicon Medical Information System (b) .....	13
Εικόνα 2-3. COSTAR (a) .....	14
Εικόνα 2-4. COSTAR (b) .....	15
Εικόνα 2-5. PROMIS .....	17
Εικόνα 3-1. Use Case (Software) .....	22
Εικόνα 3-2. Use Case (Mobile App) .....	24
Εικόνα 3-3. Διάγραμμα Σειράς (Καταγραφή) .....	26
Εικόνα 3-4. Διάγραμμα Δραστηριοτήτων (Καταγραφή) .....	26
Εικόνα 3-5. Διάγραμμα Συνεργασίας (Καταγραφή) .....	27
Εικόνα 3-6. Διάγραμμα Σειράς (Επεξεργασία) .....	27
Εικόνα 3-7. Διάγραμμα Δραστηριοτήτων (Επεξεργασία) .....	28
Εικόνα 3-8. Διάγραμμα Συνεργασίας (Επεξεργασία) .....	28
Εικόνα 3-9. Διάγραμμα Σειράς (Επισκόπηση) .....	29
Εικόνα 3-10. Διάγραμμα Δραστηριοτήτων (Επισκόπηση) .....	30
Εικόνα 3-11. Διάγραμμα Συνεργασίας (Επισκόπηση) .....	30
Εικόνα 3-12. Διάγραμμα Σειράς (Αναζήτηση) .....	31
Εικόνα 3-13. Διάγραμμα Δραστηριοτήτων (Αναζήτηση) .....	32
Εικόνα 3-14. Διάγραμμα Συνεργασίας (Αναζήτηση) .....	32
Εικόνα 3-15. Διάγραμμα Σειράς (Διαγραφή) .....	33
Εικόνα 3-16. Διάγραμμα Δραστηριοτήτων (Διαγραφή) .....	33
Εικόνα 3-17. Διάγραμμα Συνεργασίας (Διαγραφή) .....	33
Εικόνα 3-18. Διάγραμμα Εξαρτημάτων .....	34
Εικόνα 4-1. Eclipse UI .....	36
Εικόνα 4-2. EER διάγραμμα .....	37
Εικόνα 4-3. Εγγραφές στο μητρώο .....	39
Εικόνα 4-4. Εγγραφές φαρμάκων .....	39
Εικόνα 4-5. Απεικόνιση κλάσης Database.java .....	41
Εικόνα 4-6. Απεικόνιση κλάσης Immunizations.java .....	42
Εικόνα 4-7. Καρτέλα Εισόδου .....	44
Εικόνα 4-8. Επιτυχής Είσοδος .....	45
Εικόνα 4-9. Καρτέλα Φαρμάκων .....	46
Εικόνα 4-10. Επεξεργασία φαρμάκου .....	47

Εικόνα 4-11. Πίνακας επιλογής φαρμάκου .....	47
Εικόνα 4-12. Καρτέλα ραντεβού .....	48
Εικόνα 4-13. Καρτέλα ιστορικού (α) .....	49
Εικόνα 4-14. Καρτέλα ιστορικού (β) .....	49
Εικόνα 4-15. Καρτέλα διαχειριστή .....	50
Εικόνα 4-16. Καρτέλα διαχείρισης προσώπων - Παράδειγμα Καταγραφής ..	51
Εικόνα 4-17. Μήνυμα Επιβεβαίωσης Καταγραφής .....	52
Εικόνα 4-18. Ανανέωση πίνακα .....	52
Εικόνα 4-19. Επεξεργασία προσώπου .....	53
Εικόνα 4-20. Παράδειγμα Αναζήτησης .....	54
Εικόνα 4-21. Επιβεβαίωση Διαγραφής .....	55
Εικόνα 4-22. Εικονίδιο Εφαρμογής .....	56
Εικόνα 4-23. Λειτουργικό Android .....	57
Εικόνα 4-24. Android Smartphone .....	58
Εικόνα 4-25. Είσοδος Εφαρμογής .....	59
Εικόνα 4-26. Κεντρική καρτέλα εφαρμογής .....	60
Εικόνα 4-27. Λίστα εξετάσεων .....	61
Εικόνα 4-28. Λίστα γιατρών .....	62
Εικόνα 4-29. Παράδειγμα απόκρουσης sql injection στο λογισμικό .....	63
Εικόνα 4-30. Παράδειγμα απόκρουσης sql injection στην εφαρμογή .....	64
Εικόνα 4-31. Eclipse Juno .....	68
Εικόνα 5-1. Windows phone εφαρμογή .....	74
Εικόνα 6-1. MedInfoSystem - Γραμμές κώδικα .....	85
Εικόνα 6-2. MedInfoSystem - Αριθμός παραμέτρων .....	85
Εικόνα 6-3. MedInfoSystem - Αριθμός statements .....	86
Εικόνα 6-4. Servlet - Γραμμές κώδικα .....	86
Εικόνα 6-5. Servlet - Αριθμός παραμέτρων .....	87
Εικόνα 6-6. Servlet - Αριθμός statements .....	87
Εικόνα 6-7. MedInfoDroid - Γραμμές κώδικα .....	88
Εικόνα 6-8. MedInfoDroid - Αριθμός παραμέτρων .....	88
Εικόνα 6-9. MedInfoDroid - Αριθμός statements .....	89
Εικόνα 6-10. Περιεχόμενα CD .....	91



# Κεφάλαιο 1

## Εισαγωγή

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ



## 1. Εισαγωγή

### Περίληψη

Στα κεφάλαια της Διπλωματικής Εργασίας που κρατάτε στα χέρια σας αναλύεται η προσπάθεια σχεδιασμού και ανάπτυξης ενός Ιατρικού Πληροφοριακού Συστήματος, προσαρμοσμένου στις σύγχρονες επιταγές της τεχνολογίας και του Μάρκετινγκ που να συμβαδίζει με τις ολοένα και βελτιούμενες υπηρεσίες που απολαμβάνει ο μέσος καταναλωτής, ακόμα και όταν πρόκειται για την κατανάλωση προϊόντων υγείας. Θα εξεταστεί η δυνατότητα σύνδεσης πολλών διαφορετικών χρηστών, είτε από τερματικά εντός της ιατρικής μονάδας είτε από φορητές συσκευές, που αποτελούν αναπόσπαστο κομμάτι της καθημερινότητας του σύγχρονου ανθρώπου.

Θα επιχειρηθεί μια ανάλυση των κινήτρων που οδήγησαν στην εκπόνηση της παρούσας Διατριβής και των στόχων που έχουν προκαθοριστεί. Στη συνέχεια θα εξερευνηθούν οι ιστορικές προεκτάσεις και οι τάσεις της αγοράς, από το παρελθόν μέχρι τις μέρες μας, στον τομέα των Πληροφοριακών Συστημάτων με θέμα τις υπηρεσίες υγείας. Στο τρίτο κεφάλαιο ξεκινάει η παρουσίαση του Συστήματος που σχεδιάζεται και αναπτύσσεται ενώ στο επόμενο κεφάλαιο μελετάται η αρχιτεκτονική του. Θα εξαχθούν κάποια συμπεράσματα τα οποία μπορεί ο αναγνώστης να διαβάσει επιγραμματικά στο πέμπτο κεφάλαιο, σε συνδυασμό με τις μελλοντικές σκέψεις του συγγραφέα αλλά και τη σύνοψη των πιθανών μελλοντικών προεκτάσεων του Συστήματος.

Όπως θα γίνει κατανοητό και στις επόμενες σελίδες η Διπλωματική Εργασία αυτή θα συνδυάσει γνώσεις από διαφορετικά πεδία της Επιστήμης της Πληροφορικής, με βασικό στόχο να αποτελέσει μια μοντέρνα ματιά στην πολύπλοκη αγορά των Πληροφοριακών Συστημάτων. Επίσης, επιθυμεί να προσφέρει μια πρωτότυπη και καινοτόμο προσέγγιση του θέματος, χρωματίζοντας μια γκρίζα και σχετικά δυσπρόσιτη περιοχή.

Λέξεις κλειδιά: Ιατρικό Πληροφοριακό Σύστημα, Ιατρική, σχεδιασμός λογισμικού

### Abstract

The Master Thesis 'Modern Approach to Medical Data Management' is a try to design and develop a Medical Information System, adapted to the modern technological needs, consistent with the improving services enjoyed by the average consumer, even if he consumes medical products. It will be tested the possibility to connect many different users, either from terminals inside the medical unit or from mobile devices, daily used by numerous individuals.

The motivations that led to the preparation of this Master Thesis will be analyzed and the targets will be set. Then, some similar Information Systems will be surveyed from a historical view and this will help to clarify the environment in which such a System could be competitive, respecting the aspects covered or not. In the third chapter will be presented the software and the architecture will be studied in the next chapter. After all, a conclusion will be

reached and the possible future updates will be stated.

However, this Master Thesis is a combination of different disciplines of Computer Science that will try to prove the general knowledge that has become an asset acquired through the study in University of Piraeus and the Master Program in Informatics.

Keywords: medical informatics, health informatics, software engineering

## Σκοπός της Εργασίας

Στα πλαίσια της παρούσας Μεταπτυχιακής Διατριβής επιδιώκεται η αξιοποίηση και ο συνδυασμός των γνώσεων που αποκομισθήκαν κατά τη διάρκεια της διετούς φοίτησης στο Πρόγραμμα Μεταπτυχιακών Σπουδών Πληροφορικής του Πανεπιστημίου Πειραιώς. Οι γνώσεις αυτές δεν περιορίζονται σε συγκεκριμένο μάθημα ή πεδίο δραστηριοτήτων αλλά γίνεται προσπάθεια σκέδασης της παρακάτω προσέγγισης σε όσο το δυνατόν μεγαλύτερο εύρος αντικειμένων, σχετικών με την Πληροφορική. Μία λίστα των μαθημάτων που έχουν συμβάλλει στην εκπόνηση της τελικής αυτής εργασίας είναι ενδεικτική του παραπάνω συλλογισμού:

- Τεχνολογία Λογισμικού
- Ιατρική Πληροφορική
- Πληροφοριακά Συστήματα
- Βάσεις Δεδομένων
- Αλληλεπίδραση Ανθρώπου - Υπολογιστή
- Αντικειμενοστραφής Προγραμματισμός
- Τεχνολογίες Διαδικτύου
- Δίκτυα Υπολογιστών

Πρόκειται για μια προσπάθεια συνεργασίας διαφορετικών πεδίων γνώσης με σκοπό την αποτελεσματικότερη ανάπτυξη λογισμικού.

Μελετάται η σχεδίαση και ανάπτυξη ενός Ιατρικού Πληροφοριακού Συστήματος και mobile εφαρμογής. Το σύστημα αυτό θα προσφέρει τη δυνατότητα καταγραφής απαραίτητων στοιχείων που αφορούν τη λειτουργία μιας ιατρικής μονάδας (νοσοκομείου ή κλινικής). Πρόκειται για στοιχεία που σχετίζονται με:

- τις παροχές της ιατρικής μονάδας, όπως οι εξετάσεις και το κόστος τους,
- τα άτομα που εργάζονται μέσα στη μονάδα (γιατροί, εργαζόμενοι κλπ)
- τη διαχείριση των ραντεβού
- το ιστορικό των επιβλεπόμενων

Οι πληροφορίες που θα καταγράφονται στη Βάση Δεδομένων του Πληροφοριακού Συστήματος θα πρέπει να είναι ασφαλείς και να προστατεύονται από εξωτερικές επιθέσεις. Ο χρήστης του συστήματος θα μπορεί να τις επεξεργάζεται και να τις διαγράφει ενώ ο επιβλεπόμενος θα έχει πρόσβαση στα προσωπικά του δεδομένα, μέσω του κινητού του τηλεφώνου, αφού λάβει το προσωπικό του όνομα χρήστη και τον κωδικό πρόσβασης και εφόσον έχει επιβεβαιωθεί ως εξουσιοδοτημένος χρήστης από τον διαχειριστή του Συστήματος.

Η προσπάθεια επικεντρώνεται κυρίως στην σχεδίαση του λογισμικού με τέτοιο τρόπο ώστε να είναι εύκολα και γρήγορα επεκτάσιμο για να μπορεί να χρησιμοποιηθεί στο μέλλον καλύπτοντας όσο το δυνατόν μεγαλύτερο φάσμα δραστηριοτήτων μιας ιατρικής μονάδας. Επίσης, θα πρέπει να είναι εύχρηστο και προσιτό στον χρήστη, ακολουθώντας συγκεκριμένες μεθοδολογίες που θα του δώσουν γρήγορα τη δυνατότητα να γίνει έμπειρος πάνω στο συγκεκριμένο σύστημα. Η εμπειρία του χρήστη είναι βασικό πλεονέκτημα στη σωστή λειτουργία οποιασδήποτε μονάδας και βοηθάει στην καλύτερη και ευκολότερη εξυπηρέτηση του 'πελάτη'. Είναι δηλαδή, εκτός των άλλων, μία προσπάθεια προσέγγισης της σκοπιάς του χρήστη από την πλευρά του προγραμματιστή.

Άλλο ένα σημαντικό χαρακτηριστικό του είναι η εκμετάλλευση της φορητότητας που προσφέρει η γλώσσα Java, ώστε να μπορεί να εκτελεστεί σε οποιοδήποτε περιβάλλον (Windows, Mac, Linux), αλλά και η εύκολη εγκατάσταση και λειτουργία του, ακόμα και σε υπολογιστές χαμηλών δυνατοτήτων.

Σε προηγούμενη εργασία έχει δημιουργηθεί εφαρμογή για κινητά τηλέφωνα με λειτουργικό Android, η οποία συνδέεται με βάση δεδομένων που δημιουργείται και εκτελείται τοπικά στη συσκευή. Δεδομένου ότι ο χρήστης πρέπει να περνάει μόνος του τα ιατρικά ιστορικά του στοιχεία, για να τα έχει συγκεντρωμένα, η εφαρμογή καταλήγει να είναι δυσλειτουργική και ίσως άχρηστη για κάποιον που δεν επιθυμεί ή ξεχνάει να την ενημερώσει. Έτσι, εμπνεύστηκε η ιδέα για επέκταση του Πληροφοριακού Συστήματος ώστε ο επιβλεπόμενος να έχει πρόσβαση στα προσωπικά του στοιχεία, τα οποία είναι καταχωρημένα σε μία Βάση Δεδομένων, που βρίσκεται στον Διακομιστή της μονάδας. Ο χρήστης βέβαια, για λόγους ασφαλείας, δεν μπορεί να τα επεξεργαστεί αλλά είναι σημαντικό να τα έχει διαθέσιμα ανά πάσα στιγμή. Είναι ακόμα πιο σημαντικό για την μονάδα, που μπορεί να έχει άμεση επικοινωνία με τον επιβλεπόμενο - πελάτη της και γνωρίζει πότε έχει προγραμματιστεί η επόμενη εξέταση ώστε να τον ενημερώσει έγκαιρα, σε περίπτωση που αυτός το αμελήσει. Για τον λόγο αυτό, το σύστημα έχει αξία και από τη σκοπιά του Μάρκετινγκ.

Από τα παραπάνω φαίνεται ότι η παρούσα εργασία θα επεκταθεί σε τομείς όπως:

- Η σχεδίαση Πληροφοριακών Συστημάτων με τη βοήθεια της γλώσσας UML
- Η σχεδίαση Βάσεων Δεδομένων
- Η ανάπτυξη σε κώδικα Java, συνεπικουρούμενη από βοηθητικές τεχνολογίες, όπως η XML και η JSON.
- Η χρήση της γλώσσας SQL και η επικοινωνία με τη Βάση Δεδομένων
- Η κατασκευή λογισμικού
- Η δημιουργία mobile εφαρμογών
- Η επικοινωνία και μεταφορά δεδομένων, με ασφαλή και γρήγορο τρόπο, μέσω διαδικτύου

## Ιατρική Πληροφορική

Στο σταυροδρόμι της Επιστήμης των Υπολογιστών και της Ιατρικής βρίσκεται η Ιατρική Πληροφορική, η οποία μπορεί να περιγραφεί και με τους όρους Πληροφορική της Υγείας, της ιατρικής φροντίδας, Κλινική

Πληροφορική, Βιοπληροφορική κ.α. Βασικό της αντικείμενο είναι η μελέτη των πόρων, των συσκευών και των μεθόδων που μπορούν να αξιοποιηθούν με απώτερο σκοπό την βέλτιστη συλλογή, αποθήκευση και διαχείριση ιατρικών δεδομένων.

Η είσοδος των ηλεκτρονικών υπολογιστών στην καθημερινότητα των ανθρώπων από τις αρχές της δεκαετίας του '50 έγινε αισθητή και στον χώρο της Ιατρικής. Κατά τη διάρκεια της δεκαετίας του '60 εμφανίστηκαν τα πρώτα εξειδικευμένα τμήματα Πληροφορικής σε Πανεπιστήμια της Γαλλίας, της Γερμανίας, του Βελγίου και της Ολλανδίας ενώ μια δεκαετία αργότερα έκαναν την εμφάνισή τους και οι πρώτες ερευνητικές μονάδες Ιατρικής Πληροφορικής στην Πολωνία και στις Η.Π.Α.

Η πρώτη επισήμανση της ανάγκης εισαγωγής της Ιατρικής Πληροφορικής στο πρόγραμμα σπουδών των ιατρικών τμημάτων έγινε το 1984 από το σώμα αμερικάνικων κολεγίων Ιατρικής (Association of American Medical Colleges). Πλέον διδάσκεται σε προπτυχιακό και μεταπτυχιακό επίπεδο και με εξειδικεύσεις σε τομείς όπως η οδοντιατρική, η νοσηλευτική, η ακτινολογία κ.α.

Όσον αφορά την Ευρώπη οι χώρες που πρωτοστάτησαν στην δημιουργία τμημάτων Πληροφορικής, όπως αναφέρθηκε παραπάνω, ήταν αυτές που συμπεριέλαβαν πρώτες τον κλάδο της Ιατρικής Πληροφορικής σε πολλά ακαδημαϊκά τμήματά τους. Στην Ελλάδα διδάσκεται σε ορισμένες σχολές Ιατρικής και τείνει να συμπεριληφθεί στο πρόγραμμα σπουδών και των τμημάτων Πληροφορικής.

Η ανάγκη μελέτης θεμάτων Ιατρικής Πληροφορικής έγινε από νωρίς αισθητή με αποτέλεσμα να οδηγήσει τη διεθνή ομοσπονδία επεξεργασίας της πληροφορίας (International Federation of Information Processing ή IFIP) στη δημιουργία μιας σχετικής επιτροπής το 1967. Η προσπάθεια αυτή κατέληξε στην σύσταση της διεθνούς ένωσης ιατρικής πληροφορικής (International Medical Informatics Association ή IMIA) το 1978. Βασικές αρμοδιότητές της είναι:

- ✓ Η συνεργασία ακαδημαϊκών επιστημόνων και ερευνητών πληροφορικής με εργαζόμενους στην υγεία και κατασκευαστές ιατρικών μηχανημάτων
- ✓ η παγκόσμια συνεργασία για θέματα Ιατρικής Πληροφορικής
- ✓ η διοργάνωση από το 1974 ενός διεθνούς συνεδρίου Ιατρικής Πληροφορικής κάθε 3 έτη
- ✓ Η μελέτη ειδικών θεμάτων μέσω επιτροπών εργασίας
- ✓ Η έκδοση ενός ετήσιου βιβλίου με ειδικά αφιερώματα και επιλεγμένα άρθρα σε θέματα Ιατρικής Πληροφορικής

## Πληροφοριακά Συστήματα Υγείας

### Γενικά

Η ανάγκη συλλογής, επεξεργασίας, αποθήκευσης και ανάκτησης ιατρικών πληροφοριών με τη βοήθεια ηλεκτρονικών υπολογιστών και δικτυακού εξοπλισμού οδήγησε στην δημιουργία ενός είδους Πληροφοριακών Συστημάτων που είναι γνωστά με τον όρο Πληροφοριακά Συστήματα Νοσοκομείου (ΠΣΝ). Τα Συστήματα αυτά διαχειρίζονται δύο είδη

1. Πληροφορίες που αφορούν τη διοικητική διαχείριση των ασθενών και του νοσοκομείου γενικότερα
2. Πληροφορίες υγείας που συλλέγονται κατά την ιατρονοσηλευτική φροντίδα των ασθενών.

Με βάση τον παραπάνω διαχωρισμό τα ΠΣΝ χωρίζονται σε δύο κατηγορίες υποσυστημάτων, τα υποσυστήματα διοικητικής και οικονομικής διαχείρισης και τα υποσυστήματα διαχείρισης των πληροφοριών υγείας. Αυτά με τη σειρά τους μπορούν να υποδιαιρεθούν περαιτέρω σε διοικητικά και οικονομικά, επικοινωνιών και δικτύωσης, πυρήνα, διαχείρισης και υποστήριξης επιμέρους τμημάτων, ιατρικής τεκμηρίωσης, ιατρικής υποστήριξης, νοσηλευτικής δραστηριότητας, ιατρικής έρευνας κ.α.

### **Αρχιτεκτονική ΠΣΝ**

Όλα τα παραπάνω συστήματα και υποσυστήματα δομούνται με συγκεκριμένο τρόπο. Η αρχιτεκτονική τους ασχολείται με τη διάρθρωσή τους και μπορεί να διακριθεί σε τρεις κατηγορίες:

1. Κεντρικά Συστήματα: Διαχειρίζονται την απαραίτητη πληροφορία μέσα από ένα κεντρικό σύστημα ηλεκτρονικών υπολογιστών που βρίσκεται εγκατεστημένο μέσα στην ιατρική μονάδα.
2. Άρθρωτά Συστήματα: Η επεξεργασία της πληροφορίας, στο μεγαλύτερο μέρος της, συμβαίνει σε τοπικό επίπεδο αλλά τα επιμέρους Συστήματα επικοινωνούν με ένα κεντρικό με άμεση σύνδεση
3. Κατανεμημένα Συστήματα: Τα Συστήματα είναι ανεξάρτητα μεταξύ τους και η επεξεργασία της πληροφορίας γίνεται τοπικά, ωστόσο μοιράζονται μεταξύ τους τα δεδομένα. Πρόκειται για την πιο δεδομένη μορφή σε αυτή την κατηγορία Πληροφοριακών Συστημάτων εξαιτίας κάποιων χαρακτηριστικών όπως ο διαμοιρασμός πόρων, η ευρύτητα, η κλιμάκωση, ο συγχρονισμός, η διαφάνεια και η ανοχή στα λάθη.

### **ΠΣΝ στην Ελλάδα**

Στη χώρα μας φαίνεται ότι δεν έχει γίνει ακόμα κατανοητή η συμβολή των ολοκληρωμένων Πληροφοριακών Συστημάτων στην εύρυθμη λειτουργία των νοσοκομείων και των οργανισμών υγείας. Ελάχιστοι από αυτούς διαθέτουν τέτοια συστήματα και σε ακόμα λιγότερες περιπτώσεις τα συστήματα αυτά αφορούν την εξυπηρέτηση νοσηλευτικών αναγκών. Φαίνεται πως υπάρχουν αρκετοί λόγοι που μπορούν να δικαιολογήσουν την έλλειψη τέτοιου είδους υποδομής στις μονάδες υγείας. Επιγραμματικά μπορεί να αναφερθεί ότι οι μονάδες αυτές:

- στερούνται τυποποιημένων διαδικασιών και ροών εργασίας
- αλλάζουν σε τακτά χρονικά διαστήματα τις στρατηγικές που αφορούν την ανάπτυξη της Πληροφορικής
- στερούνται επίσης εκπαιδευμένου και εξειδικευμένου προσωπικού
- δεν διαθέτουν τμήματα Πληροφορικής και οργάνωσης αλλά ακόμα και στις περιπτώσεις που υφίστανται τέτοια τμήματα

το προσωπικό είναι ανεπαρκές

- δεν τηρούνται τα καθήκοντα σχετικά με τη χρήση της Πληροφορικής
- Απουσιάζουν από τον τακτικό προϋπολογισμό οι επενδύσεις για την ανάπτυξη της Πληροφορικής
- Δεν υπάρχει θεσμικός φορέας με σκοπό τη μελέτη θεμάτων ιατρονοσηλευτικής Πληροφορικής.

Μέσα στους σκοπούς της παρούσας Εργασίας είναι να καταστήσει σαφές ότι η χρήση τέτοιου είδους Πληροφοριακών Συστημάτων από τις ιατρικές μονάδες είναι εξαιρετικής σημασίας. Ο εθνικός σχεδιασμός για την υγεία πρέπει να προσανατολιστεί άμεσα σε αυτόν τον τομέα προκειμένου να οργανώσει τις μονάδες αυτές και να παρέχει επιτέλους υψηλού επιπέδου υπηρεσίες στους ασθενείς, με κόστος που μακροπρόθεσμα θα είναι μειωμένο, σε σχέση με το σημερινό καθεστώς χάους που επικρατεί. Επιπλέον, αυτά τα Συστήματα παίζουν καθοριστικό ρόλο στην εύρυθμη λειτουργία και στις υπηρεσίες που προσφέρονται και από τις ιδιωτικές μονάδες, αποδεικνύοντας ότι τα μακροπρόθεσμα οφέλη επεκτείνονται και στην διευκόλυνση των καθημερινών δραστηριοτήτων αλλά και στην αποκόμιση οικονομικών ωφελειών.

## Γιατί Java;

Η επιλογή της βασικής γλώσσας προγραμματισμού που θα χρησιμοποιηθεί για την υλοποίηση του Πληροφοριακού Συστήματος που εξετάζουμε δεν έγινε τυχαία. Βασίζεται σε συγκεκριμένα κριτήρια που λήφθηκαν υπόψη κατά την εξέταση των διαθέσιμων επιλογών. Μια σύγκριση μεταξύ των πιθανών υποψήφιων γλωσσών μπορεί να δώσει ικανοποιητική απάντηση στο ερώτημα. Αρχικά όμως θα πρέπει να μελετήσουμε τις ιδιότητες της Java, που την κάνουν μια ξεχωριστή γλώσσα προγραμματισμού, ιδιαίτερως δημοφιλή στην παγκόσμια κοινότητα των Developers.

Ένα από τα πιο σημαντικά χαρακτηριστικά της Java λοιπόν είναι η ανεξαρτησία που προσφέρει, όσον αφορά τα Λειτουργικά Συστήματα που την υποστηρίζουν. Ουσιαστικά τρέχει μέσα από τη δική της εικονική μηχανή (virtual machine) κάτι που της προσδίδει φορητότητα, με αποτέλεσμα πάνω από τρία Δισεκατομμύρια συσκευές σε όλο τον κόσμο να τρέχουν το λογισμικό της. Έτσι, μπορεί να κατασκευαστεί ένα πρόγραμμα που θα μπορεί να εγκατασταθεί με ευκολία σε οποιονδήποτε υπολογιστή, χωρίς να είναι διαδικτυακό και άρα εκτεθειμένο σε οποιαδήποτε κακόβουλη επίθεση.

Πρόκειται για μία γλώσσα που εκμεταλλεύεται στο έπακρο τα θετικά στοιχεία του αντικειμενοστραφούς προγραμματισμού. Η αλήθεια είναι ότι η επιλογή του συγκεκριμένου είδους σε σχέση με τον δηλωτικό προγραμματισμό είναι υποκειμενική και υπάρχουν αντικρουόμενες απόψεις που υποστηρίζουν τη μία ή την άλλη εκδοχή. Πάντως οι υποστηρικτές του αντικειμενοστραφούς προγραμματισμού δείχνουν να είναι περισσότεροι και τα επιχειρήματά τους να κερδίζουν στα σημεία. Εξάλλου, σχετικά με τα Πληροφοριακά Συστήματα η συγκεκριμένη επιλογή μοιάζει να εξυπηρετεί σε μεγαλύτερο βαθμό.

Ένα από τα σημαντικότερα ζητήματα που προκύπτει, ειδικά όταν πρόκειται για ευαίσθητα ιατρικά δεδομένα, είναι αυτό της ασφάλειας και σε αυτό το σημείο η Java προσφέρει ξεχωριστές δυνατότητες και υψηλό επίπεδο προστασίας. Φυσικά το θέμα αυτό εξαρτάται και από τις επιλογές του προγραμματιστή, ο οποίος πρέπει να επινοήσει κάποιες καινοτόμες ιδέες και να θωρακίσει τα δεδομένα με όσο το δυνατόν καλύτερο και πιο εφευρετικό τρόπο. Όμως η Java παρέχει επιπλέον χρήσιμες δυνατότητες.

Για τον προγραμματισμό σε Java παρέχονται ελεύθερα εργαλεία που μπορούν να χρησιμοποιηθούν από οποιονδήποτε υπολογιστή σε οποιοδήποτε λειτουργικό Σύστημα. Τα ολοκληρωμένα Περιβάλλοντα Ανάπτυξης (Integrated Development Environments) είναι λογισμικά που χρησιμοποιούνται για την ανάπτυξη προγραμμάτων. Τα δημοφιλέστερα από αυτά, όσον αφορά τη Java, είναι το Eclipse και το Netbeans. Παρέχονται δωρεάν και προσφέρουν πληθώρα εργαλείων και πρόσθετων (plug-ins) που διευκολύνουν τη δουλειά του προγραμματιστή.

Μια γλώσσα παρόμοιας φιλοσοφίας με την Java και αρκετά πιο φιλική είναι η C#, η οποία θα μπορούσε να περιγραφεί με τον όρο 'η Java της Microsoft'. Οι δύο γλώσσες μοιάζουν μεταξύ τους σε πολλά σημεία και κυρίως στον τρόπο που εκμεταλλεύονται τις αντικειμενοστραφείς τους ιδιότητες. Ωστόσο, η C# έχει απλοποιήσει τα περισσότερα από τα 'δυσκολοχώνευτα' χαρακτηριστικά της Java, προσφέροντας μάλιστα εύχρηστες επιλογές σε σημεία που θα έπρεπε να κοπιάσει ο Java προγραμματιστής. Δεδομένου όμως ότι και με τις δύο γλώσσες μπορεί να επιτευχθεί ο σκοπός της εργασίας, ενώ ζητούμενο δεν είναι η φιλικότητα αλλά η εξερεύνηση σε βάθος, το χαρακτηριστικό που δίνει πλεονέκτημα στην Java είναι η ανεξαρτησία της σε σύγκριση με μια γλώσσα προσαρμοσμένη στις επιταγές της Microsoft, που δεν μπορεί να τρέξει με ευκολία οπουδήποτε και τα περιβάλλοντα ανάπτυξής της αφενός δεν είναι μεταφέρσιμα μεταξύ των λειτουργικών συστημάτων και αφετέρου υστερούν σε σχέση με τα αντίστοιχα της Java. Μπορεί το Visual Studio να είναι μια εξαιρετική επιλογή αλλά, εκτός του γεγονότος ότι εγκαθίσταται μόνο σε Windows, είναι ένα βαρύ πρόγραμμα που δεν μπορεί να αξιοποιηθεί σε οποιονδήποτε υπολογιστή. Επίσης, προσφέρεται δωρεάν μόνο η Express έκδοσή του, άρα υστερεί στα χαρακτηριστικά που το Eclipse και το Netbeans προσφέρουν χωρίς αντίτιμο.

Η C++ από τη μεριά της είναι μια γλώσσα που μοιάζει και αυτή με την Java αλλά είναι ελαφρώς πιο πολύπλοκη και έχει κληρονομήσει από τη C αρκετά στοιχεία, τα οποία της δίνουν τη δυνατότητα να χρησιμοποιηθεί, χωρίς βέβαια να αξιοποιείται απόλυτα, και ως δηλωτική γλώσσα. Για ορισμένους προγραμματιστές θεωρείται μία από τις γρηγορότερες γλώσσες προγραμματισμού αλλά η διαφορά της από την Java είναι σχεδόν ανεπαίσθητη και εξαρτάται από τις προγραμματιστικές τεχνικές που θα ακολουθηθούν. Υστερεί και αυτή σε προγραμματιστικά περιβάλλοντα που προσφέρονται για την ανάπτυξη της. Το ίδιο θέμα υπάρχει και με τη C που απορρίπτεται εξάλλου επειδή δεν προσφέρει την αντικειμενοστρέφεια που έχει αποφασιστεί να αξιοποιηθεί στην παρούσα Μεταπτυχιακή Διατριβή.

Η PHP είναι μια γλώσσα σχεδιασμένη για να τρέχει στην πλευρά του server, οπότε θα μπορούσε να χρησιμοποιηθεί για αυτόν τον σκοπό. Θα έπρεπε όμως να χρησιμοποιηθεί κάποια άλλη γλώσσα (ή συνδυασμός

Μεταπτυχιακή Διατριβή Παναγιώτης Κοκκότης  
γλώσσών) για τον σχεδιασμό του λογισμικού. Επιπλέον, έχει κι αυτή αντικειμενοστραφείς ιδιότητες αλλά όχι στον βαθμό και τη μορφή που τις συναντάμε στην Java, η οποία έχει και το πλεονέκτημα της ασφάλειας, με περισσότερες και πιο σταθερές επιλογές. Όσον αφορά τα σχεδιαστικά περιβάλλοντα η PHP μπορεί να αναπτυχθεί σε Eclipse και Netbeans με την εγκατάσταση των απαραίτητων πρόσθετων.

Το συμπέρασμα λοιπόν που προκύπτει είναι ότι η Java υπερέχει οριακά έναντι των υπόλοιπων γλωσσών που εξετάστηκαν. Είναι πλήρης, καθώς μπορεί να χρησιμοποιηθεί και για την δημιουργία εφαρμογών αλλά και στην πλευρά του διακομιστή ενώ είναι κατάλληλη και για την κατασκευή διεπαφών. Είναι ασφαλής σε μεγάλο βαθμό (αν και αυτό εξαρτάται από τον προγραμματιστή περισσότερο) και αντικειμενοστραφής. Επίσης, τα περιβάλλοντα ανάπτυξης είναι ανεξάρτητα λειτουργικού, με μεγάλες δυνατότητες και δωρεάν. Έτσι κι αλλιώς η Java θα χρησιμοποιούταν για την ανάπτυξη της εφαρμογής για Android. Ωστόσο δεν είναι η μοναδική γλώσσα που θα χρησιμοποιηθεί στην παρούσα Διατριβή. Άλλες τεχνολογίες, όπως η XML, JSON και φυσικά η SQL για την Βάση Δεδομένων θα συμβάλλουν στην ολοκληρωμένη ανάπτυξη του Ιατρικού Πληροφοριακού Συστήματος.



**Εικόνα 1-1. Java**





# Κεφάλαιο 2

## Επισκόπηση Πεδίου

## 2. Επισκόπηση Πεδίου

Η επιστήμη της Πληροφορικής, από τότε που γεννήθηκε και άρχισε να εξαπλώνεται, έχει συμβάλει σε πολύ μεγάλο βαθμό στην εξέλιξη του ανθρώπινου είδους και όσο οι δυνατότητές της διευρύνονται τόσο μεγαλύτερη είναι και η συμβολή της στην διευκόλυνση και εξυπηρέτηση των ανθρώπινων αναγκών. Οι επιστήμονες δεν αδιαφόρησαν μπροστά στις δυνατότητες που μπορούσε να τους προσφέρει ο υπολογιστής, ακόμα και στον ευαίσθητο τομέα της υγείας, κι έτσι επικέντρωσαν από πολύ νωρίς τις προσπάθειές τους στην ανάπτυξη βοηθητικών εργαλείων για την καλύτερη και πιο αποτελεσματική ανάπτυξη του κλάδου της Ιατρικής. Ήδη από τις αρχές της δεκαετίας του '50 οι τάσεις της εποχής στον επιστημονικό κόσμο έδιναν έμφαση στην ανάπτυξη δραστηριοτήτων υγειονομικής περίθαλψης. Παρόλο που οι δυνατότητες των υπολογιστών εκείνης της εποχής ήταν περιορισμένες και το κόστος μεγάλο είχε αρχίσει να διαφαίνεται ότι θα διαδραμάτιζαν σταδιακά καθοριστικό ρόλο στον τρόπο με τον οποίο η Ιατρική και οι σχετικές με αυτή δραστηριότητες θα επηρέαζαν τις ζωές μας μελλοντικά.

Υπάρχουν διάφοροι τίτλοι και ακρώνυμα για την κατηγοριοποίηση των Ιατρικών Πληροφοριακών Συστημάτων και των διαφορετικών προσεγγίσεων της διαχείρισης της ροής και αποθήκευσης πληροφοριών στην καθημερινότητα των ιατρικών μονάδων. Μερικοί από αυτούς είναι:

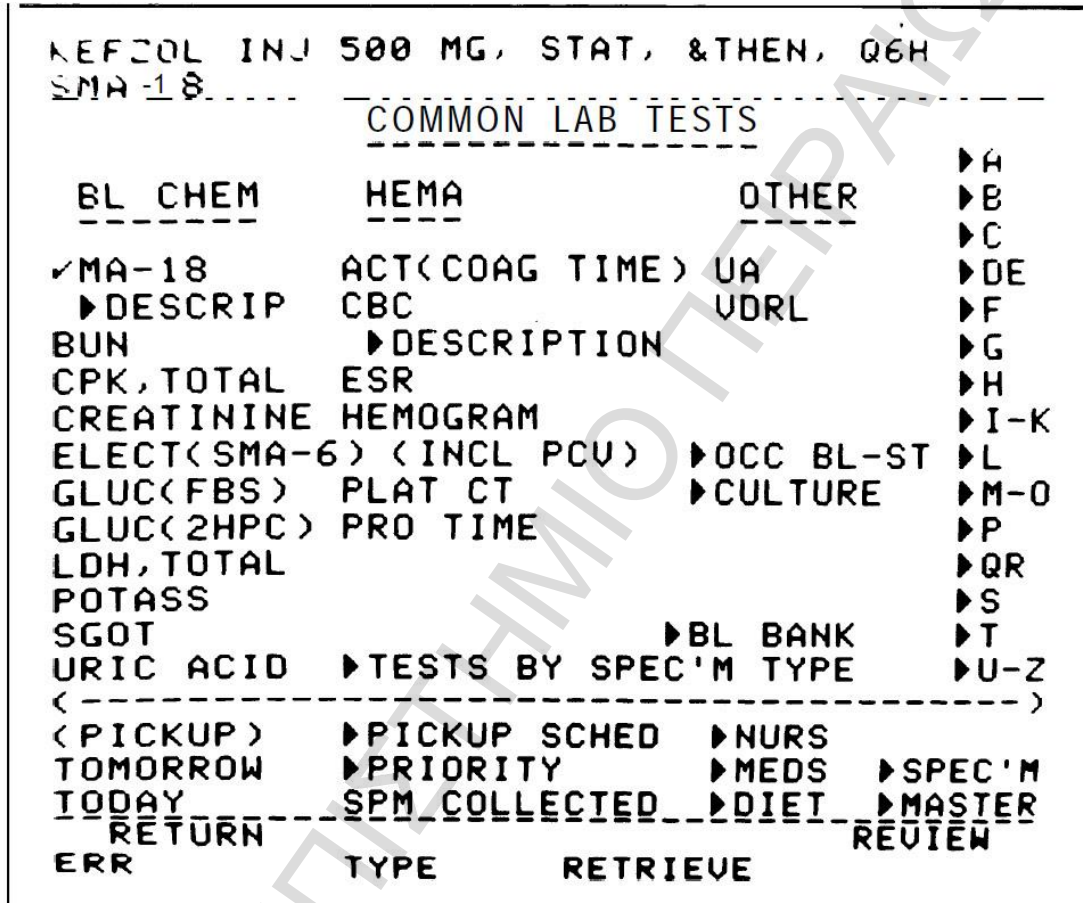
- **Hospital Information System (HIS)**
- **Healthcare Information System**
- **Clinical Information System (CIS)**
- **Patient Data Management System (PDMS)**

Πρόκειται για Πληροφοριακά Συστήματα σχεδιασμένα για να διαχειρίζονται ιατρικές, νομικές και οικονομικές υπηρεσίες. Όλοι οι τύποι αυτών των συστημάτων υποστηρίζουν αρχιτεκτονική Πελάτη - Διακομιστή (Client - Server) και στις μέρες μας, ακολουθώντας τα παραγγέλματα των καιρών, έχουν αρχίσει να χρησιμοποιούν φορητές συσκευές, όπως κινητά τηλέφωνα και tablets. Το πολύ διαδεδομένο στις μέρες μας Cloud Computing δεν συνιστάται για τέτοιου είδους Συστήματα, εξαιτίας της ιδιαίτερης σημασίας που έχει η ασφάλεια των προσωπικών ιατρικών δεδομένων, που δεν μπορεί σε αυτή την περίπτωση να εξασφαλιστεί. Καθένα από αυτά τα λογισμικά μπορεί να αποτελείται από ένα ή περισσότερα στοιχεία και εξειδικευμένες επεκτάσεις, ιδιαίτερα στην περίπτωση των HIS.

Σκοπός τους είναι να επιτύχουν τη μέγιστη δυνατή υποστήριξη της ιατρικής φροντίδας των ασθενών παρέχοντας πληροφορίες, όπου είναι αυτό απαραίτητο, και συλλέγοντας δεδομένα τα οποία στη συνέχεια επεξεργάζονται με τον αποτελεσματικότερο δυνατό τρόπο. Στη συνέχεια παρουσιάζονται ορισμένα από τα σημαντικότερα και ιστορικής σημασίας Πληροφοριακά Συστήματα που έχουν αναπτυχθεί για ιατρικούς σκοπούς.

**Technicon Medical Information System**

Στα μέσα της δεκαετίας του '60 και συγκεκριμένα το 1965 ξεκίνησε η ανάπτυξη ενός από τα πρώτα Ιατρικά Πληροφοριακά Συστήματα, του 'Technicon Medical Information System' [1] (Εικόνες 2-1, 2-2) από το Νοσοκομείο 'El Camino' της Καλιφόρνια σε συνεργασία με την εταιρεία Lockheed. Το Σύστημα αυτό εγκαταστάθηκε για πρώτη φορά το 1971 σε ιατρική μονάδα και είχε πολύπλοκες κλινικές, βοηθητικές και διαχειριστικές δυνατότητες, οι οποίες αποτελούσαν σημείο αναφοράς για τα υπολογιστικά δεδομένα της εποχής.



SOURCE Technicon Corp

**Εικόνα 2-1. Technicon Medical Information System (a)**

Μεταξύ των πολυάριθμων επιλογών του το TMIS προσέφερε στο νοσοκομείο El Camino τη δυνατότητα να διαχειρίζεται τις εγγραφές των ασθενών του. Η γενικότερη διαχείριση των προσωπικών στοιχείων του κάθε επιβλεπόμενου προερχόταν από τον συνδυασμό των υπολογιστικών ικανοτήτων του Πληροφοριακού Συστήματος αλλά και από χειροποίητες εγγραφές που συμπλήρωναν τις κάρτες στις οποίες καταγράφονταν τα δεδομένα και διατηρούνταν στο τμήμα Διαχείρισης Ιατρικών Εγγραφών. Τα στοιχεία του κάθε ασθενή αποθηκεύονταν στον ενεργό υπολογιστή για 48 ώρες μετά την εξαγωγή του από το νοσοκομείο και στη συνέχεια μεταφέρονταν σε μαγνητικές ταινίες για μόνιμη αποθήκευση. Το TMIS

Μεταπτυχιακή Διατριβή Παναγιώτης Κοκκότης  
συνεχίζει να υφίσταται μέχρι τις μέρες μας αλλά ακόμα και σήμερα δεν παρέχει τη δυνατότητα μακροπρόθεσμης διαχείρισης των εγγραφών ενός ασθενούς και κάθε φορά που αυτός εισέρχεται στο νοσοκομείο δημιουργείται μια νέα κάρτα.

Ανέκαθεν το TMIS περιελάμβανε και επιλογές επιχειρησιακής διαχείρισης, όπως η χρέωση των υπηρεσιών και των προμηθειών, οι λογαριασμοί των ασθενών, η μισθοδοσία, στατιστικά στοιχεία κ.α. Το Σύστημα αναπτύσσεται ακόμα και στο παρόν θέτοντας ως προτεραιότητες την ιατρική φροντίδα, τη διαχείριση της ιατρικής μονάδας και την παροχή πληροφοριών για την διευκόλυνση σημαντικών κλινικών αποφάσεων από τους γιατρούς.

### **COMPUTER STORED AMBULATORY RECORD (COSTAR) SYSTEM**

Ακόμα ένα ορόσημο στην ιστορία των Ιατρικών Πληροφοριακών Συστημάτων αποτελεί το 'COSTAR' [1] (Εικόνες 2-3, 2-4), το οποίο αποτελούσε μέρος του 'Κοινοτικού Σχεδίου Υγείας' του Χάρβαρντ (σήμερα γνωστό ως Harvard Pilgrim), ένα πρόγραμμα μη κερδοσκοπικού χαρακτήρα που προσφέρει πληθώρα ασφαλιστικών επιλογών και διακανονισμών αυτοχρηματοδότησης σε περισσότερα από ένα εκατομμύριο μέλη στη Μασαχουσέτη. Το πρόγραμμα αναπτύχθηκε στο Πανεπιστήμιο του Χάρβαρντ σε συνεργασία με το Εργαστήριο Υπολογιστών και το κόστος για τα πρώτα 5 χρόνια της λειτουργίας του υπολογίζεται στα 2,5 εκατομμύρια δολάρια.

03-16-72 12:11 PM

NEW MEDICAL ORDERS

MILES, NELSON APPLETON M 86 10391925

BED: 249B N/S: 2NORT SERV: SURG CHILDS. W

PRIMARY DIAGNOSIS: GALL BLADDER

ENTERED BY: CHILDS. W

TIME ENTERED: 03-16-72 12:10 PM

X-RAY: GALLBLADDER- USE STANDARD PREP #7-

IPPB. AIR DILUTION

ACTIVITY. AMBULATE

DIET. REGULAR

VITAL SIGNS. T-P-R/BP. Q30M

IV'S START D5/RINGERS. 1000 ML. INFUSE OVER 8HR. THEN DC IV:  
ADD TO IV BOTTLE. COMPAZINE-INJ. 100MG. IN IV

CBC

UA

VDRL

SMA-12 (PREP #1)

DIGOXIN-0.25MG. IM. STAT

SECONAL-INJ: SECOBARBITAL-20MG. IM. QID. (03/16 01PM-...)

SECONAL-INJ: SECOBARBITAL-20MG. IM MR X1 FOR SLEEP

-----  
DR. CHILDS. W

NOTED -----

- - -

SOURCE Technicon Corp

Εικόνα 2-2. Technicon Medical Information System (b)

4) SITE A <input type="checkbox"/> MH-PH B <input type="checkbox"/> CAMBRIDGE C <input checked="" type="checkbox"/> KENMORE D <input type="checkbox"/> KEN. TRIAGE E <input type="checkbox"/> CHMC F <input type="checkbox"/> PBBH G <input type="checkbox"/> BLI H <input type="checkbox"/> BI J <input type="checkbox"/> HOUSE CALL K <input type="checkbox"/> OTHER		5) TYPE A <input checked="" type="checkbox"/> SCHEDULED B <input type="checkbox"/> WALK-IN C <input type="checkbox"/> TEL EPHONE D <input type="checkbox"/> CANCEL LEO E <input type="checkbox"/> DNK F <input type="checkbox"/> CALL-IN G <input type="checkbox"/> IN-PATIENT H <input type="checkbox"/> EW I <input type="checkbox"/> NON-ENCOUNTER J <input type="checkbox"/> GROUP		NAME: <u>Record, Sample enr</u> UNIT: <u>07-39-07 I</u> DATE: <u>4/4/75</u> DOB: PROV #1: <u>Plotkin</u> PROV #2: <u>Imbernino</u> (if seen) ABOVE MUST BE COMPLETE FOR INPUT	
6) Hospital or EW Visit Approved: <input type="checkbox"/> Yes <input checked="" type="checkbox"/> No					
***** THIS INFORMATION FOR FIRST VISIT PRIMARY PROVIDER ONLY *****					
7) PRIMARY MO <u>Plotkin</u>		8) PRIMARY RN <u>Imbernino</u>			
9) RACE: A <input checked="" type="checkbox"/> CAUCASIAN B <input type="checkbox"/> BLACK C <input type="checkbox"/> SPANISH SPKG D <input type="checkbox"/> OTHER					
10) MAR. STATUS: A <input type="checkbox"/> SINGLE B <input checked="" type="checkbox"/> MARRIED C <input type="checkbox"/> WIDOWED D <input type="checkbox"/> SEPARATED E <input type="checkbox"/> DIVORCED					
11) # OF CHILDREN <u>0</u>		12) PT. OCCUPATION <u>Staff acct. - Boston based consulting firm</u>			
PERSONAL BACKGROUND OF PATIENT - Only for primary providers (To change, rewrite entire section; max. three lines. Do not repeat above data.)					
<p><u>Pt is a fairly active person, generally healthy and feeling well. More physically active in summer - tennis, swimming. Commutes from southern New Hampshire during summer</u></p>					
***** TO BE COMPLETED FOLLOWING EACH VISIT *****					
40) DISPOSITION Future appt w/ _____ in _____ (provider's full name) A _____ DAYS B _____ WEEKS C _____ MONTHS D _____ PRN E _____ PT to call MD G _____ PT to call RN F _____ MD to call PT H _____ RN to call PT I _____ Other _____					
REFERRALS 42) INTERNAL HCHP CONSULTATION Consultation w/ <u>Urology # 2993</u> (enter specialty and provider's last name, if specified) Consultation w/ _____ (enter specialty and provider's last name, if specified) (IF A CONSULTATION NOTE IS NECESSARY, INCLUDE REASON FOR CONSULT IN ROUTINE DICTATION)					
43) REFERRAL TO OUTSIDE AGENCY - NON-HOSPITAL _____ (enter full name & address of physician and/or agency)					
DOCUMENTS TO BE FORWARDED: _____					
44) HOSPITAL ARRANGEMENTS (make a choice in each column) A <input type="checkbox"/> BETH ISRAEL B <input type="checkbox"/> PBBH C <input type="checkbox"/> BOSTON HOSPITAL FOR WOMEN D <input type="checkbox"/> CHMC F <input type="checkbox"/> UNAFFILIATED 1. <input type="checkbox"/> EMERGENCY WARD 2. <input type="checkbox"/> URGENT ADMISSION 3. <input type="checkbox"/> SCHEDULED ADMISSION Date of Admission (if known) _____					
45) SUPPORTIVE SERVICES A <input type="checkbox"/> HOMEMAKER B <input type="checkbox"/> VNA C <input type="checkbox"/> HOMECARE D <input type="checkbox"/> TRANSPORTATION		46) DICTATION _____		47) REVIEW OF CHART _____	
48) FOLLOW-UP IMPORTANT _____					

Εικόνα 2-3. COSTAR (a)

Το 'Κοινοτικό Σχέδιο Υγείας' χρησιμοποιεί το COSTAR για να αντλεί πληροφορίες σχετικά με την ιατρική φροντίδα των ασθενών και την διαχείριση του προγράμματος. Τα δεδομένα του κάθε ασθενή διανέμονται στους γιατρούς πριν από τα ραντεβού με σκοπό την καλύτερη επίβλεψή του

Μεταπτυχιακή Διατριβή Παναγιώτης Κοκκότης  
 και την παροχή σημαντικών ιατρικών ιστορικών στοιχείων, που  
 εξυπηρετούν την αποτελεσματικότερη διάγνωση ασθενειών αλλά και την  
 φροντίδα του.

Το COSTAR εξυπηρετεί και αυτό τις διαχειριστικές απαιτήσεις του  
 Σχεδίου. Προσφέρει στατιστικές αναλύσεις, διαχειριστικές εκθέσεις,  
 διαχείριση του προϋπολογισμού, διευκόλυνση του επιχειρησιακού  
 σχεδιασμού, μισθοδοσία κλπ. Τα προσωπικά ιατρικά δεδομένα των ασθενών  
 διατηρούνται σε σκληρούς δίσκους που δεν είναι συνδεδεμένοι με την  
 κεντρική μονάδα επεξεργασίας. Η προσπάθεια στις μέρες μας  
 επικεντρώνεται στον εμπλουτισμό του Συστήματος με υπολογιστικά  
 πρωτόκολλα που θα προσφέρουν επιπλέον ιατρική φροντίδα αλλά και η  
 επέκταση του Συστήματος σε ανάλογα προγράμματα, όπως του Cambridge.

```

RECORD SAMPLE ENR-(M)
:FF 1/75          PRIMARY MD PLOTKIN      87-39-87-1
:RP FFS          PRIMARY RN IMBERNINO    56 YRS.-DOB:1/2/1919

_CS MGH, BOSTON 02109 TEL:726-3933
_RACE:CAUCASIAN
_MAR STATUS:MARRIED
# OF CHILDREN:0
PT OCCUPATION:STAFF ACCOUNTANT, BOSTON BASED CONSULTING FIRM
PT IS A FAIRLY ACTIVE PERSON. GENERALLY HEALTHY AND FEELING
WELL. MORE PHYSICALLY ACTIVE IN SUMMER - TENNIS, SWIMMING.
COMMUTES FROM SOUTHERN NEW HAMPSHIRE DURING SUMMER. 4/4/75

4688 INITIAL HEALTH ASSESSMENT GENERALLY HEALTHY. P.E. WNL
                                         4/4/75 (PLOTKIN: M.D.)

MAJOR PROBLEMS
1993 HEMATURIA NO PREVIOUS HX PRIOR 1 WK. IN PROCESS OF EVALUATION
                                         4/1/75-2-4/4/75 (PLOTKIN: M.D.) #D

MINOR PROBLEMS
A150 OBESITY WOULD LIKE TO LOSE AT LEAST 10 POUNDS
P666 TOBACCO ADDICTION 1 PACK PER DAY      4/4/75 (PLOTKIN: M.D.)
                                         4/4/75 (PLOTKIN: M.D.)

INACTIVE PROBLEMS
S100 S/P APPENDECTOMY 1957                4/4/75 (PLOTKIN: M.D.)

CURRENT THERAPY
R010 FLUIDS 8 GLASSES PER DAY            4/1/75 (GOLDSMITH: R.N.)

TEST RESULTS
4/4/75 R078 PYELOGRAM (IVP)
        NORMAL, NO EVIDENCE STONE, CALYCES NORMALLY VISUALIZED
4/1/75 N204 URINALYSIS INCL MICRO 10-20 RBC/HPF (2) #A
        B100 URINE CULTURE NO GROWTH
        A156 WHITE BLOOD COUNT 8900
        A128 HEMOGLOBIN 14.1
        A127 RBC 4.4
        A126 HEMATOCRIT 39

CONSULTATIONS AND REFERRALS
UROLOGY 4/4/75 (PLOTKIN) FOR L993
    
```

SOURCE Laboratory of Computer Sciences Massachusetts General Hospital

**Εικόνα 2-4. COSTAR (b)**

## **PROBLEM-ORIENTED MEDICAL INFORMATION SYSTEM (PROMIS)**

Το **PROMIS** [1] (Εικόνα 2-5) σχεδιάστηκε και αναπτύχθηκε από το

Μεταπτυχιακή Διατριβή Παναγιώτης Κοκκότης  
εργαστήριο του Πανεπιστημίου του Vermont στην περιοχή του Burlington, για τις ανάγκες του πανεπιστημιακού νοσοκομείου. Ο σχεδιασμός του ξεκίνησε το 1967 και εγκαταστάθηκε πειραματικά το 1971 σε μία γυναικολογική μονάδα του νοσοκομείου με 20 κλίνες. Η λειτουργία του κράτησε 4 χρόνια, μέχρι το 1975, και κατά τη διάρκειά του χρησιμοποιήθηκε στον τομέα της ραδιολογίας, στη διαχείριση του εργαστηρίου της κλινικής, των φαρμάκων αλλά και των σημειώσεων των χειρουργών μετά τις επεμβάσεις.

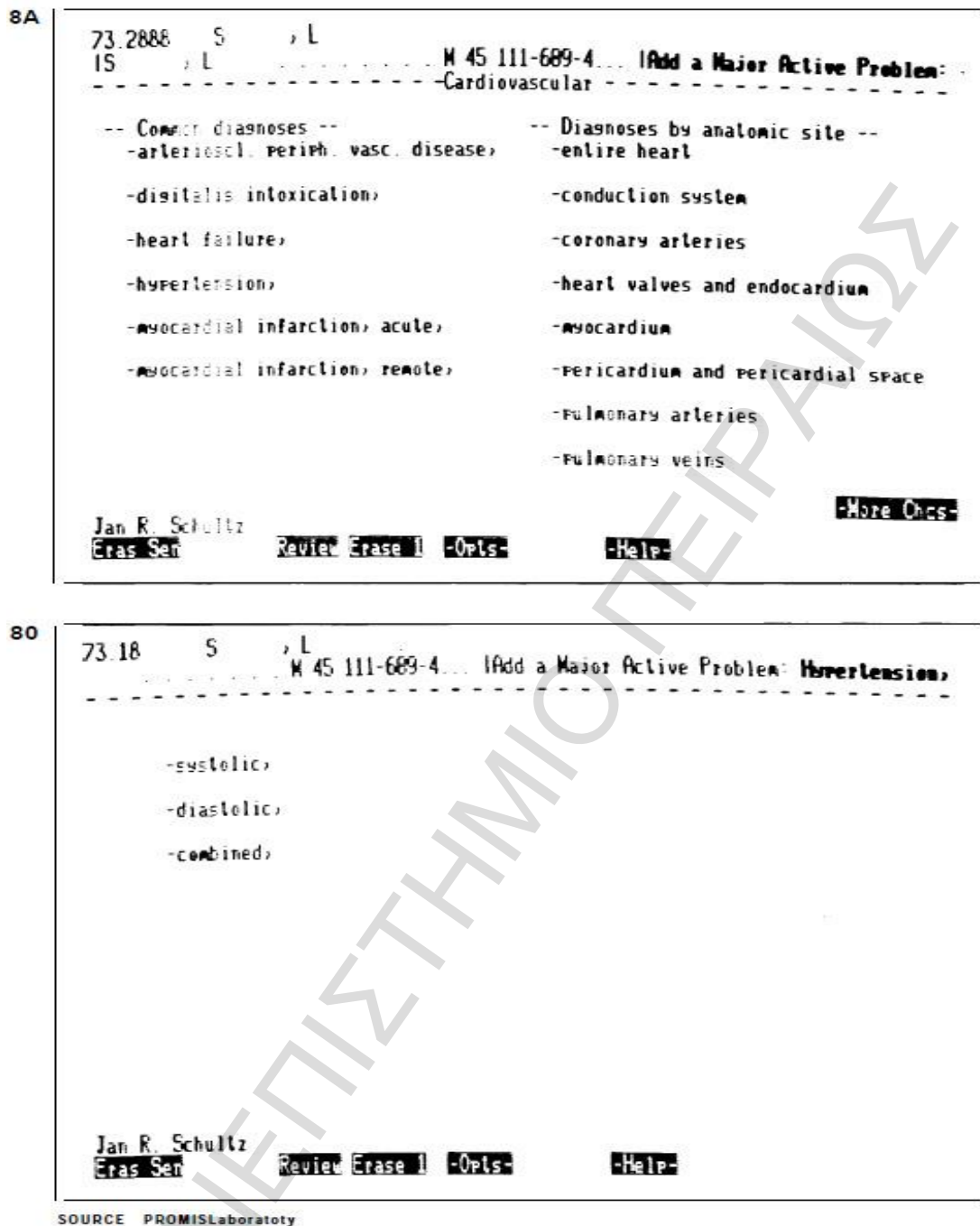
Πρόκειται για ένα πραγματικά πρωτοποριακό Σύστημα της εποχής του, το οποίο διέθετε οθόνη αφής και ήταν εντυπωσιακά γρήγορο αναλογικά με το τι συνέβαινε τη χρονική περίοδο που αναπτύχθηκε. Ο Δρ Lawrence Weed ήταν ο άνθρωπος που δούλεψε από το 1969 μέχρι το 1982 για τη δημιουργία του PROMIS, λαμβάνοντας υπόψη του ότι ένας γιατρός δεν είναι σε θέση να συγκρατήσει στο μυαλό του τον τεράστιο όγκο δεδομένων που αφορούσαν τον κάθε ασθενή του ξεχωριστά και με αυτόν τον γνώμονα αποφάσισε να παρέχει στην επιστήμη ένα Πληροφοριακό Σύστημα που θα εξυπηρετούσε την καλύτερη οργάνωση αυτών των δεδομένων και θα είχε θετικό αντίκτυπο στην φροντίδα των ασθενών. Το προφίλ του κάθε ασθενούς ήταν διαθέσιμο στην οθόνη του κάθε τερματικού σε μικρό χρόνο και αυτό έκανε την ιατρική φροντίδα πιο αποδοτική.

### **HELP (Health Evaluation through Logical Processing)**

Πρόκειται για ένα από τα πιο δημοφιλή Ιατρικά Πληροφοριακά Συστήματα, το οποίο ξεκίνησε να αναπτύσσεται στο νοσοκομείο LDS του Salt Lake City στην περιοχή της Utah στα τέλη της δεκαετίας του '60 και συνέχισε την ανάπτυξή του για σχεδόν 30 χρόνια, φτάνοντας το 1990 τις 6 εκατομμύρια γραμμές κώδικα.

Το **HELP** [2] κατάφερε αποτελεσματικά αφενός να αυξήσει την ποιότητα των παρεχόμενων στους ασθενείς υπηρεσιών και αφετέρου να μειώσει το κόστος αυτών των υπηρεσιών. Διαδόθηκε σε μεγάλο φάσμα ιατρικών κέντρων και σήμερα δέχεται πληροφορίες από περισσότερα από 5000 τερματικά, έχοντας τη δυνατότητα να διαχειρίζεται πάνω από 18000 εγγραφές την ημέρα. Παρόλα αυτά δεν παρέχει την δυνατότητα επικοινωνίας μεταξύ των ιατρικών μονάδων. Την τελευταία εικοσαετία πολλά παρόμοια Συστήματα, βασισμένα στα πρωτόκολλα που εφάρμοσε το HELP έχουν κατακλύσει τα ιατρικά κέντρα, κυρίως των ΗΠΑ. Η πρωτοπορία του συνεχίζεται μέχρι το παρόν, με την χρήση Τεχνητής Νοημοσύνης για την διευκόλυνση της αποδοτικότερης λήψης αποφάσεων.





Εικόνα 2-5. PROMIS

### Άλλα Συστήματα

Τα Συστήματα αυτά είναι διεπαφές που διευκολύνουν την επικοινωνία του χρήστη με μία κεντρική Βάση Δεδομένων. Εξυπηρετούν πολλαπλούς σκοπούς, με κυριότερο αυτό της μηχανογράφησης αλλά επεκτείνονται στη διαχείριση προϋπολογισμών, την υποστήριξη αποφάσεων και σε ποικιλία εφαρμογών που λαμβάνουν χώρα σε μία μονάδα υγείας, όπως οι εξετάσεις, οι

Μεταξύ άλλων το **Tri-Service Medical Information System (TRIMIS)** [4] έδωσε βάση στην διευκόλυνση των φαρμακευτικών σκοπών και την καλύτερη οργάνωση της συνταγογράφησης. Το TMR ήταν επίσης ένα δημοφιλέστερο Σύστημα που εκμεταλλεύτηκε στο έπακρο την εξάπλωση των μικροϋπολογιστών τη δεκαετία του '80 και επικεντρώθηκε στην εξυπηρέτηση των εργαστηριακών αναγκών. Ήταν ένα από τα πρώτα Συστήματα που χρησιμοποίησε ιεραρχικές δομές δεδομένων με ποικίλο μήκος κειμένου ενώ θεωρήθηκε επαναστατική καινοτομία η προστασία των προσωπικών δεδομένων με κωδικούς χρήστη και εκεί οφείλεται μεγάλο μέρος της επιτυχίας του που συνεχίζει να υφίσταται μέχρι σήμερα.

Το **Regenstrief Medical Record System (RMRS)** ξεκίνησε από την προσπάθεια του Clement McDonald και του Charles Clark το 1972 να συλλέξουν δεδομένα πάνω σε 35 διαβητικούς ασθενείς και στη συνέχεια να δημιουργήσουν ένα πρόγραμμα, που θα αποθήκευε τα δεδομένα αυτά και θα εκτύπωνε διαγράμματα, που απεικόνιζαν την πορεία της υγείας τους. Αυτή η προσπάθεια έπεσε στο κενό εξαιτίας διάφορων παραγόντων, μεταξύ των οποίων και η αστάθεια του προγράμματος κι έτσι δημιουργήθηκε η ανάγκη να αντικατασταθεί από ένα Σύστημα Βάσεως Δεδομένων.

Το RMRS εξυπηρετεί στις μέρες μας τις ανάγκες 1,3 εκατομμυρίων ασθενών, καταγράφοντας 200 εκατομμύρια κωδικοποιημένες παρατηρήσεις, 15 εκατομμύρια συνταγές και 212 χιλιάδες ηλεκτροκαρδιογραφήματα. Επιπλέον, 1300 νοσηλευτές, 1000 γιατροί και 220 φοιτητές Ιατρικής έχουν πρόσβαση στα δεδομένα των ασθενών περισσότερες από 628000 φορές το μήνα. Σκοποί του Συστήματος είναι να μειώσει τα προβλήματα της μηχανογράφησης, να βοηθήσει στη διαχείριση και την ενημέρωση των ασθενών για τα ιατρικά τους αποτελέσματα και τις διαγνώσεις και να προσφέρει ένα χρυσόρυχο δεδομένων ικανών να εξυπηρετήσουν τεράστιας σημασίας ερευνητικούς σκοπούς.

## Πλεονεκτήματα

Τα Συστήματα αυτής της κατηγορίας προσφέρουν πληθώρα πλεονεκτημάτων και επιλογών για τις ιατρικές μονάδες. Παρέχουν εύκολη πρόσβαση σε ιατρικά δεδομένα για τη δημιουργία εγγραφών και την κατηγοριοποίησή τους, με δημογραφικά, ηλικιακά και λοιπά κριτήρια. Η διαδικτυακή βάση τους παρέχει απομακρυσμένη πρόσβαση σε αυτά τα δεδομένα. Επιπλέον, διευκολύνουν τη λήψη αποφάσεων για τη διαμόρφωση πολιτικών ιατρικής φροντίδας από τις μονάδες. Προσφέρουν αποδοτική και ακριβή διαχείριση των οικονομικών δεδομένων, της δίαιτας των ασθενών και της διασποράς της ιατρικής βοήθειας. Ενισχύουν την επίβλεψη της χρήσης φαρμάκων και μελετούν την αποτελεσματικότητά τους ενώ μειώνουν τα σφάλματα καταγραφής και περιορίζουν την πιθανότητα διπλοεγγραφών, διασφαλίζοντας την ακεραιότητα των δεδομένων.

## Σύγκριση με το παρόν Σύστημα

Όπως και οι προκάτοχοί του, το Σύστημα που αναπτύσσεται στην παρούσα Διατριβή προσφέρει εύκολη πρόσβαση και διαχείριση των ιατρικών δεδομένων. Επιπλέον, επεκτείνει τις δυνατότητες πρόσβασης σε μεγάλο αριθμό χρηστών, συμπεριλαμβανομένων των ίδιων των επιβλεπόμενων. Προσφέρει μάλιστα μία καινοτόμα ιδέα για την επέκταση και διαχείρισή του στο μέλλον και από τους υπόλοιπους χρήστες, μέσω φορητών συσκευών.

Προτείνει μία πελατοκεντρική φόρμουλα άμεσης ενημέρωσης του ενδιαφερόμενου και ειδοποίησής του για πιθανές προγραμματισμένες δραστηριότητες. Σε γενικές γραμμές, είναι μια προσπάθεια συλλογής ιατρικών πληροφοριών και διάθεσής τους με ευκολία σε ευρύ κοινό.

Το Σύστημα δεν επεκτείνεται στη διαχείριση οικονομικών δεδομένων και την αξιοποίηση των πληροφοριών για χάραξη επιχειρηματικής στρατηγικής και δεν επικεντρώνεται στη διασπορά ιατρικής φροντίδας. Αυτοί όμως οι τομείς, με βάση τη λογική της σχεδίασής του, μπορούν αποτελεσματικά να περιληφθούν σε μελλοντικές του προεκτάσεις. Ένας τομέας που δεν προβλέπεται στο παρόν σύστημα, σε σύγκριση με άλλους προκατόχους του, είναι η αποθήκευση και ανάλυση γραφημάτων, ακτινογραφιών και λοιπών δραστηριοτήτων που απαιτούν επικοινωνία με εξειδικευμένα μηχανήματα που διαθέτουν οι σύγχρονες μονάδες υγείας.



# Κεφάλαιο 3

## Ανάλυση - Αρχιτεκτονική του Συστήματος

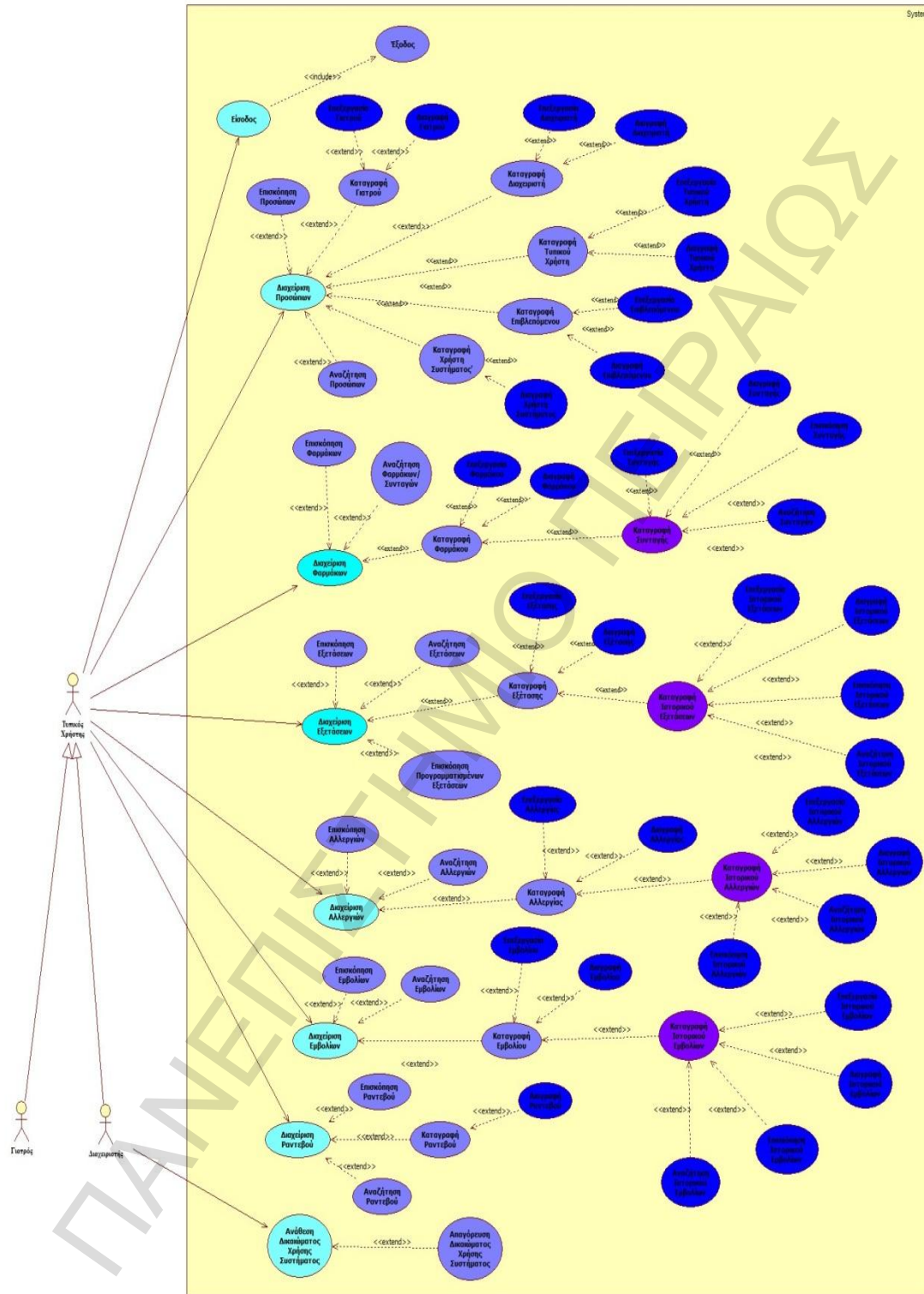
### 3. Ανάλυση – Αρχιτεκτονική του Συστήματος

#### UML

Η Unified Modeling language είναι μια γλώσσα που χρησιμοποιείται για τη μοντελοποίηση και τον σχεδιασμό λογισμικού. Απεικονίζει γραφικά, προσδιορίζει, κατασκευάζει και τεκμηριώνει όλα τα στοιχεία που συντελούν στην ανάπτυξη ενός συστήματος λογισμικού. Όπως είναι γνωστό, ένα τέτοιο σύστημα περνάει από διάφορα στάδια ανάπτυξης (ανάλυση, σχεδίαση, ανάπτυξη, έλεγχος), στα οποία χρησιμοποιείται η UML. Τα είδη των διαγραμμάτων που χρησιμοποιούνται είναι τα εξής:

- Περιπτώσεων Χρήσης: Χρησιμοποιούνται για να μοντελοποιήσουν το πλαίσιο λειτουργίας και τις προδιαγραφές του συστήματος. Βασικά συστατικά τους είναι οι δρώντες, οι περιπτώσεις χρήσης, οι σχέσεις (εξάρτησης, γενίκευσης, σύνδεσης) και τα όρια του συστήματος.
- Κλάσεων: Απεικονίζουν τη δομή του συστήματος, περιλαμβάνοντας κλάσεις και δεσμούς (εξάρτησης, γενίκευσης, σύνδεσης). Μπορούν να απεικονίσουν άριστα τη χρήση της κληρονομικότητας.
- Σειράς ή Ακολουθίας: Παρουσιάζουν τον τρόπο που διάφορα αντικείμενα συνεργάζονται μεταξύ τους σε μια χρονική ακολουθία.
- Δραστηριοτήτων: Οι μεταπτώσεις σε αυτά τα διαγράμματα είναι ανάμεσα σε διαφορετικές δραστηριότητες εκφράζοντας π.χ. τη ροή εργασιών (*workflow*). Όπως και τα διαγράμματα καταστάσεων που θα δούμε παρακάτω είναι διαγράμματα συμπεριφοράς και έχουν πολλά κοινά στοιχεία.
- Συνεργασίας: Διαγράμματα αλληλεπίδρασης (συμπεριφοράς) που παρουσιάζουν τον τρόπο που διαφορετικά αντικείμενα σχετίζονται και ανταλλάσσουν μηνύματα μεταξύ τους.
- Καταστάσεων: Εμφανίζουν μια μηχανή καταστάσεων με έμφαση στις μεταπτώσεις μεταξύ καταστάσεων από διάφορα γεγονότα.
- Αντικειμένων: Είναι διαγράμματα δομής που χρησιμοποιούνται για το σχεδιασμό της στατικής κατάστασης του συστήματος κατά μια συγκεκριμένη χρονική στιγμή.
- Εξαρτημάτων: Πρόκειται για διαγράμματα δομής υλοποίησης που χρησιμοποιούνται για τη μοντελοποίηση πηγαίου κώδικα, εκτελέσιμων εκδόσεων, βάσεων δεδομένων ή δυναμικά προσαρμοζόμενων συστημάτων.
- Ανάπτυξης: Παρουσιάζουν τον τρόπο διαμόρφωσης των υπολογιστικών κόμβων του συστήματος κατά τη λειτουργία του.
- Ροής Δεδομένων: Είναι μια γραφική αναπαράσταση της "ροής" των δεδομένων διαμέσου ενός συστήματος και μπορούν να χρησιμοποιηθούν επίσης για οπτικοποίηση της επεξεργασίας των δεδομένων (δομημένη σχεδίαση).
- Μετάβασης Καταστάσεων: Περιγράφουν τη δυναμική συμπεριφορά του λογισμικού, δηλ. τη χρονική σειρά εκτέλεσης των εργασιών ανάλογα με εξωτερικά γεγονότα

**α) Διαγράμματα Περιπτώσεων Χρήσης**

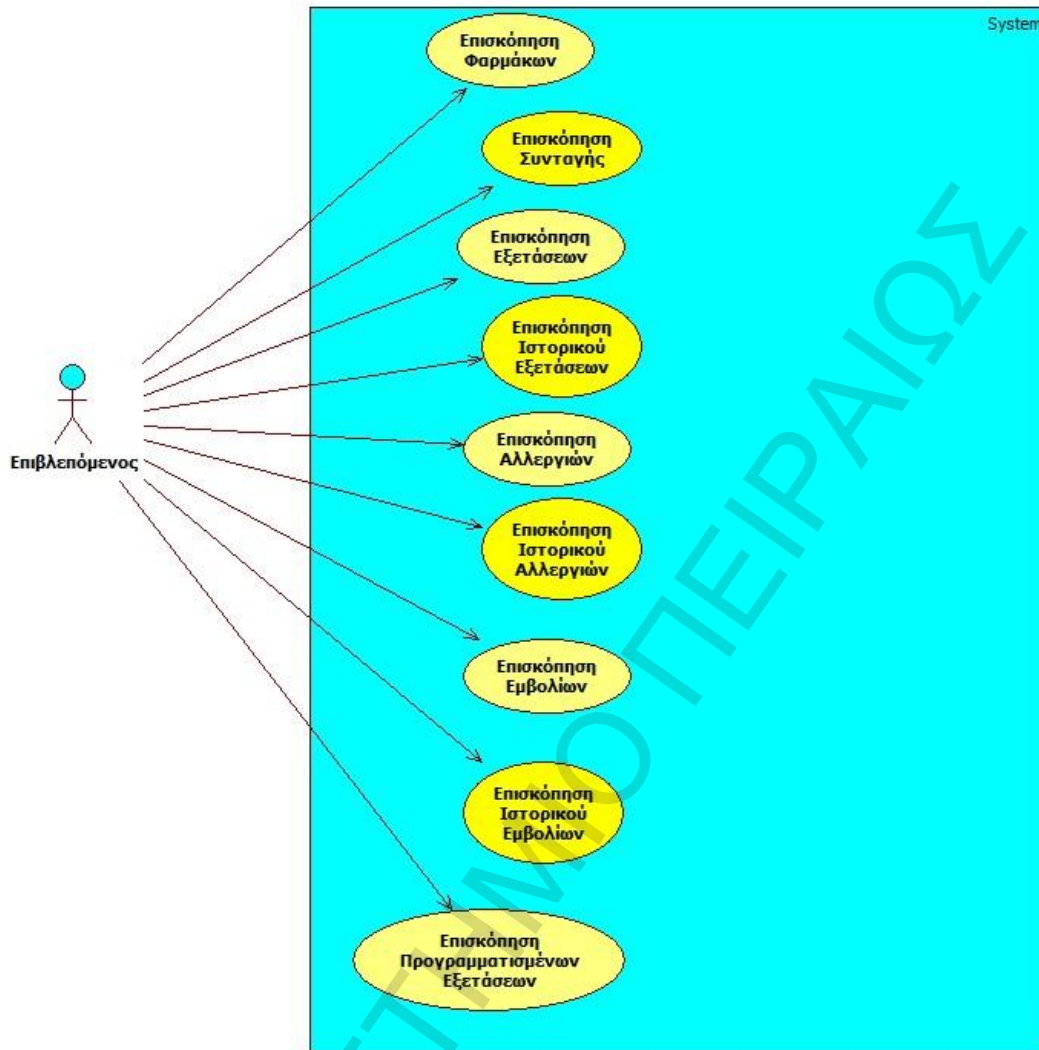


**Εικόνα 3-1. Use Case (Software)**

Το παραπάνω διάγραμμα (Εικόνα 3-1) συνοψίζει τις λειτουργίες που επιτελεί το Πληροφοριακό Σύστημα το οποίο αναπτύσσεται στην παρούσα Μεταπτυχιακή Διατριβή. Οι περιπτώσεις χρήσης που παρατηρούμε αφορούν μόνο το λογισμικό και όχι την mobile εφαρμογή. Οι χρήστες του Συστήματος αυτού χωρίζονται σε τρεις κατηγορίες. Υπάρχει ο 'Τυπικός Χρήστης' από τον οποίο κληρονομούν τις ιδιότητές τους ο 'Γιατρός' και ο 'Διαχειριστής'. Ως 'Τυπικός Χρήστης' λογίζεται οποιοσδήποτε υπάλληλος της ιατρικής μονάδας μπορεί να έχει πρόσβαση στο Σύστημα. Οι περιπτώσεις χρήσης χωρίζονται στο διάγραμμα σε επίπεδα, ξεκινώντας από ανοιχτό μπλε και συνεχίζοντας με πιο σκούρο στα επόμενα. Σε πρώτο επίπεδο εμφανίζεται η διαχείριση των διαφόρων λειτουργιών της μονάδας. Στη συνέχεια έχουμε τη δυνατότητα του χρήστη να βλέπει στην οθόνη του τις εγγραφές που υπάρχουν στη Βάση Δεδομένων και την δυνατότητα αναζήτησής. Ακολουθεί η επεξεργασία και διαγραφή των δεδομένων αυτών ενώ στη συνέχεια έχουμε πιο σύνθετες λειτουργίες που απαιτούν την ύπαρξη παραγόντων του πρώτου επιπέδου. Για παράδειγμα, η καταγραφή μιας συνταγής προαπαιτεί την ύπαρξη φαρμάκων, γιατρών και επιβλεπόμενων. Στη συνέχεια έχουμε τη δυνατότητα να αναζητήσουμε, να επεξεργαστούμε και να διαγράψουμε και αυτά τα στοιχεία.

Οι γιατροί της μονάδας έχουν πρόσβαση στο Σύστημα και κληρονομούν τις ιδιότητές τους από τον 'Τυπικό Χρήστη'. Όπως φαίνεται, στο λογισμικό μας, επιτελούν ακριβώς τις ίδιες λειτουργίες με αυτόν, χωρίς να κάνουν κάτι επιπλέον, ωστόσο αυτό δεν σημαίνει ότι δεν πρέπει να γίνει διαχωρισμός. Μπορεί προς το παρόν να μην είναι εμφανείς οι διαφορές τους με έναν απλό υπάλληλο αλλά είναι προφανές ότι σε μελλοντικές εκδόσεις του Συστήματος ο γιατρός μπορεί εύκολα να διαφοροποιηθεί και να επιτελεί ξεχωριστές λειτουργίες. Ο διαχειριστής από την άλλη πλευρά έχει επιπλέον δυνατότητες, οι οποίες προς το παρόν αφορούν την πολύ σημαντική λειτουργία της εξουσιοδότησης των χρηστών. Για να μπορεί κάποιος να έχει πρόσβαση στο Σύστημα, ακόμα κι αν διαθέτει όνομα χρήστη και κωδικό πρόσβασης, θα πρέπει οπωσδήποτε να έχει ελεγχθεί από τον διαχειριστή και να έχει επιβεβαιωθεί ως εξουσιοδοτημένος χρήστης.

Το επόμενο διάγραμμα (Εικόνα 3-2) αφορά τις περιπτώσεις χρήσης για την mobile εφαρμογή. Σε αυτή έχει πρόσβαση ο επιβλεπόμενος, ο οποίος έχει στοιχεία εισόδου και είναι εξουσιοδοτημένος. Δεν έχει πρόσβαση σε όλα τα δεδομένα, όπως είναι λογικό, αλλά μόνο σε όσα αφορούν μόνο τον ίδιο. Σκοπός της εφαρμογής είναι να ενημερώνεται εύκολα και άμεσα για τα προσωπικά του στοιχεία που βρίσκονται αποθηκευμένα στη Βάση Δεδομένων της ιατρικής μονάδας. Η αμεσότητα εξυπηρετεί και τους σκοπούς του Marketing, δεδομένου ότι η ενημέρωση π.χ. για τις προγραμματισμένες εξετάσεις παροτρύνει τον επιβλεπόμενο να μην τις αμελεί και αυτό μπορεί να αποφέρει οικονομικά οφέλη στην επιχείρηση.



**Εικόνα 3-2. Use Case (Mobile App)**

### **b) Διαγράμματα Σειράς – Δραστηριοτήτων – Συνεργασίας**

Ακολουθούν τα διαγράμματα Σειράς, Δραστηριοτήτων και Συνεργασίας που περιγράφουν τις πέντε πιο σημαντικές λειτουργίες του Ιατρικού Πληροφοριακού Συστήματος. Πρόκειται για την καταγραφή νέων εγγραφών στη Βάση Δεδομένων, τη δυνατότητα επεξεργασίας τους, τη διαγραφή τους, την αναζήτηση με βάση συγκεκριμένα κριτήρια και την άντληση συνόλου δεδομένων. Αυτές οι λειτουργίες συμβαίνουν σε όλα τα δεδομένα που διαχειρίζεται το Σύστημα και στα διαγράμματα συνοψίζεται ο τρόπος με τον οποίο λαμβάνουν χώρα. Συγκεκριμένα, στα διαγράμματα Σειράς δίνεται βάση κυρίως στη χρονολογική σειρά με την οποία συμβαίνουν οι παραπάνω λειτουργίες. Στα διαγράμματα δραστηριοτήτων απεικονίζεται η επικοινωνία μεταξύ των μερών του προγράμματος με περισσότερες λεπτομέρειες, όσον αφορά τον τρόπο και την κατάσταση σε κάθε στιγμή. Στα διαγράμματα Συνεργασίας φαίνεται



πιο καθαρά αυτή η επικοινωνία και το πώς συνεργάζονται μεταξύ τους τα διαφορετικά κομμάτια που απαρτίζουν το Πληροφοριακό Σύστημα.

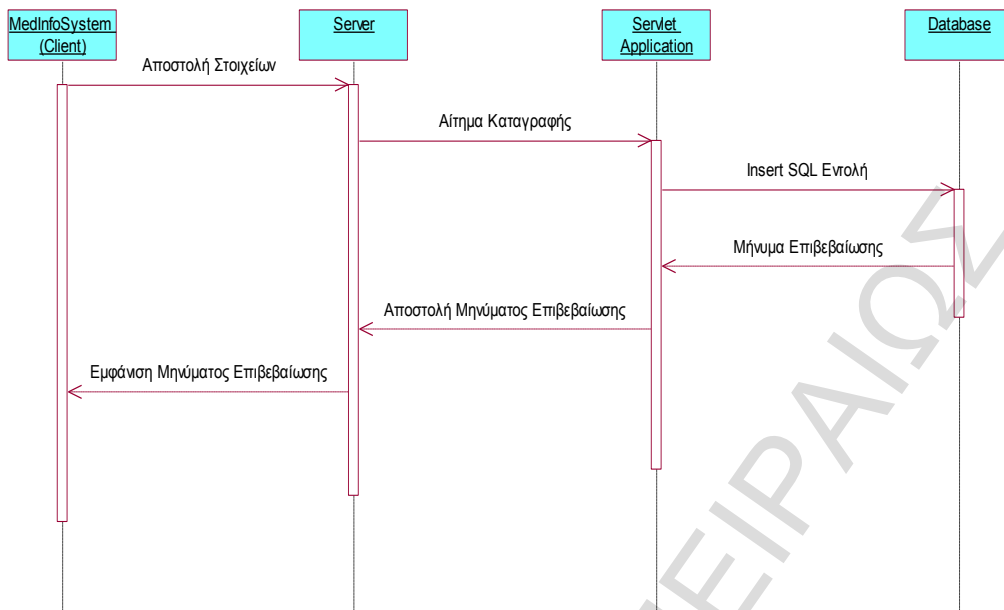
Όπως φαίνεται και στα διαγράμματα που ακολουθούν τα βασικά μέρη που συμμετέχουν σε κάθε κύκλο διαδικασιών που επιτελείται κατά τη λειτουργία του Πληροφοριακού Συστήματος είναι τα εξής:

- i. Η κεντρική εφαρμογή που έρχεται απευθείας σε επαφή με το χρήστη.
- ii. Ο Διακομιστής (server) που λαμβάνει μηνύματα και στέλνει απαντήσεις σε μια συνεχή διαδικασία ερωτήσεων – απαντήσεων
- iii. Το πρόγραμμα Java Servlet που αναλαμβάνει να μεταφράσει τα αιτήματα που δέχεται ο διακομιστής, να επικοινωνήσει με τη Βάση Δεδομένων και να μετατρέψει τα δεδομένα σε τυποποιημένη μορφή JSON, πριν αποσταλούν ξανά πίσω στον χρήστη
- iv. Η Βάση Δεδομένων που διατηρεί αποθηκευμένα τα δεδομένα και είναι υλοποιημένη σε MySQL.

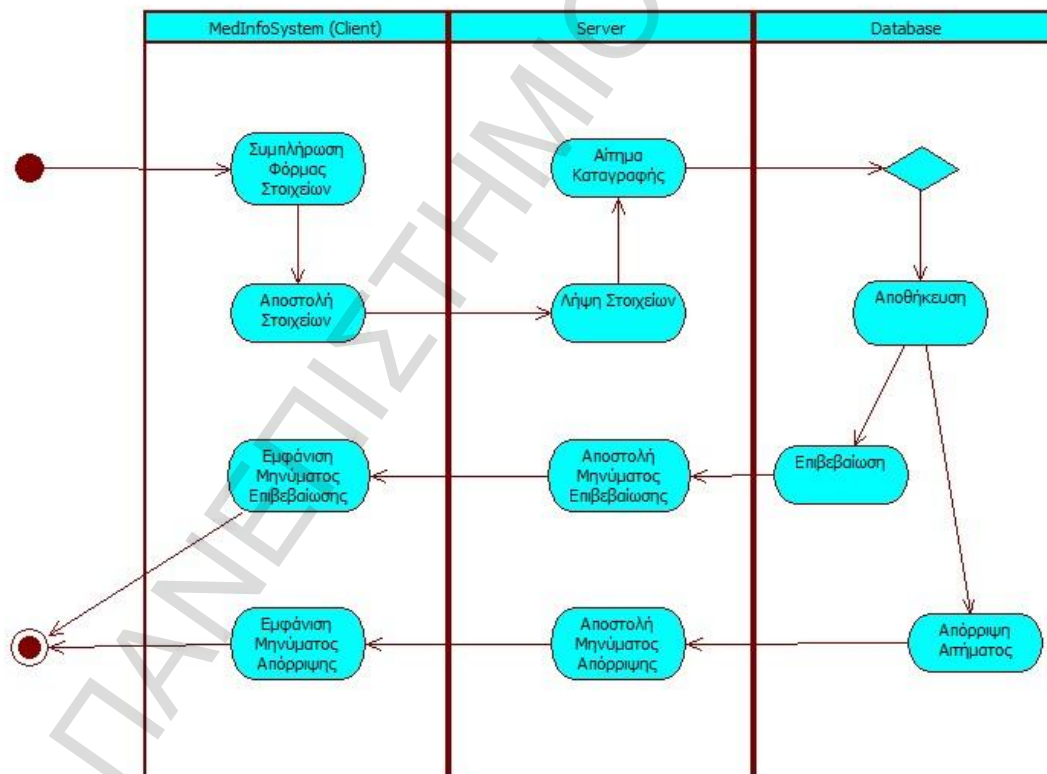
## **i) Καταγραφή**

Η διαδικασία καταγραφής των δεδομένων στη ΒΔ ξεκινάει από την πληκτρολόγηση των δεδομένων, με τη συμπλήρωση της κατάλληλης φόρμας του λογισμικού. Με το πάτημα του κουμπιού 'Αποθήκευση' τα δεδομένα αποστέλλονται στον Server, ο οποίος αναλαμβάνει να διαχειριστεί το σχετικό αίτημα. Το πρόγραμμα που τρέχει στον διακομιστή (servlet) έχοντας λάβει τα στοιχεία ενεργοποιεί την κατάλληλη συνάρτηση, η οποία θα τα εισάγει στη Βάση Δεδομένων με μία εντολή τύπου 'INSERT INTO tablename(col1, col2, col3, ...) VALUES(?, ?, ?, ...);'. Τα ερωτηματικά μετατρέπονται στον κώδικα από εντολές τύπου setString στις κατάλληλες τιμές. Πρόκειται για τα Prepared Statements που προσφέρει η Java και χρησιμοποιούνται για λόγους ασφαλείας, κυρίως για να αποφεύγονται τα SQL injections.

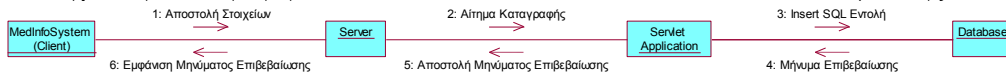
Αν όλα πάνε καλά θα επιστραφεί ένα μήνυμα επιβεβαίωσης το οποίο στη συνέχεια θα σταλεί από τον διακομιστή στον πελάτη και θα ενημερώσει τον χρήστη για την επιτυχή έκβαση της διαδικασίας. Σε διαφορετική περίπτωση θα αποσταλεί ένα μήνυμα σφάλματος που θα προτρέπει τον χρήστη να δοκιμάσει ξανά.



**Εικόνα 3-3. Διάγραμμα Σειράς (Καταγραφή)**



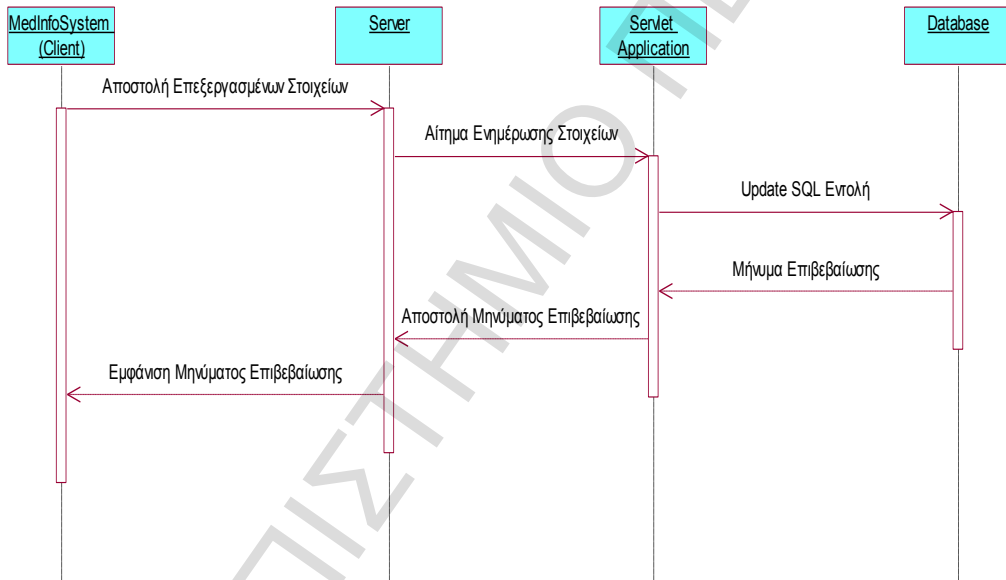
**Εικόνα 3-4. Διάγραμμα Δραστηριοτήτων (Καταγραφή)**



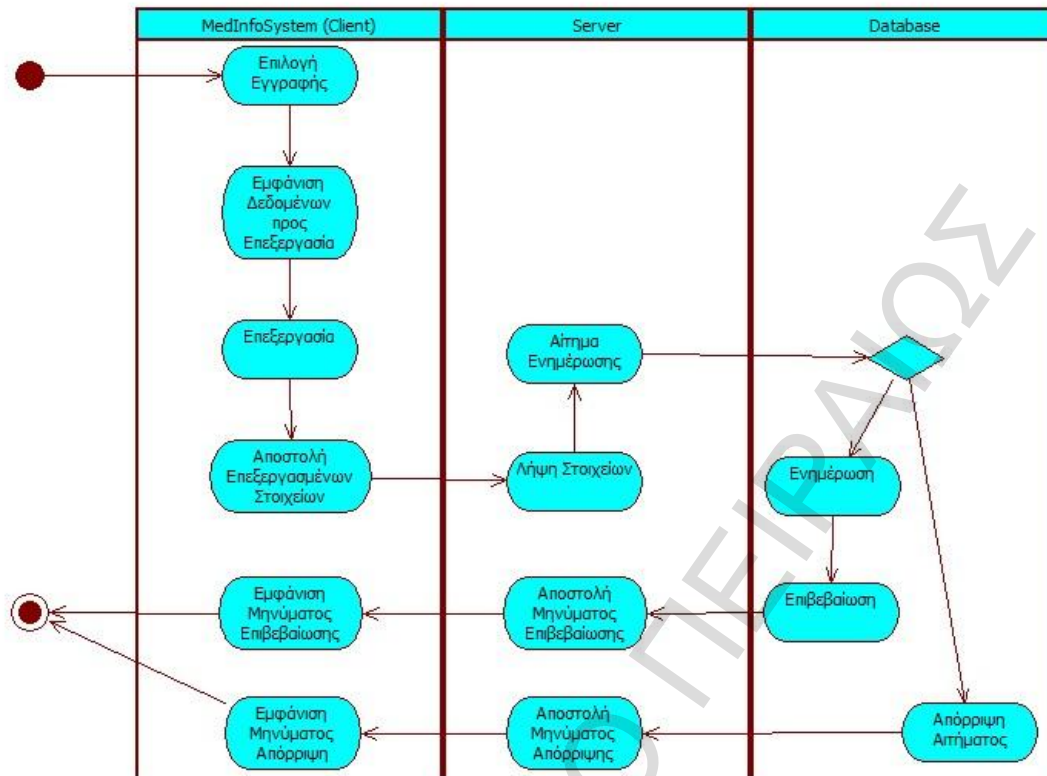
**Εικόνα 3-5. Διάγραμμα Συνεργασίας (Καταγραφή)**

**ii) Επεξεργασία**

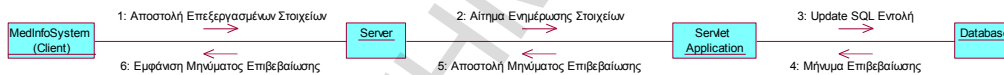
Το Σύστημα προσφέρει τη δυνατότητα επεξεργασίας των ήδη καταχωρημένων εγγράφων στη Βάση Δεδομένων. Για να γίνει αυτό θα πρέπει ο χρήστης να επιλέξει την εγγραφή που επιθυμεί να επεξεργαστεί από τους σχετικούς πίνακες και να πατήσει το κουμπί 'Επεξεργασία'. Θα εμφανιστεί μία φόρμα με τα δεδομένα σε textboxes έτσι ώστε να πραγματοποιήσει τις αλλαγές που θέλει και να τις στείλει στον διακομιστή, ο οποίος με τη σειρά τους θα ενεργοποιήσει το πρόγραμμα (servlet) και θα αποσταλεί μία εντολή τύπου 'UPDATE tablename SET col1 = ?, col2 = ?, col3 = ? ... WHERE id = ?;' στη ΒΔ. Θα ακολουθήσει η γνωστή διαδικασία επιστροφής ενός μηνύματος επιβεβαίωσης ή απόρριψης ανάλογα με την εξέλιξη του αιτήματος.



**Εικόνα 3-6. Διάγραμμα Σειράς (Επεξεργασία)**



**Εικόνα 3-7. Διάγραμμα Δραστηριοτήτων (Επεξεργασία)**



**Εικόνα 3-8. Διάγραμμα Συνεργασίας (Επεξεργασία)**

### iii) Επισκόπηση

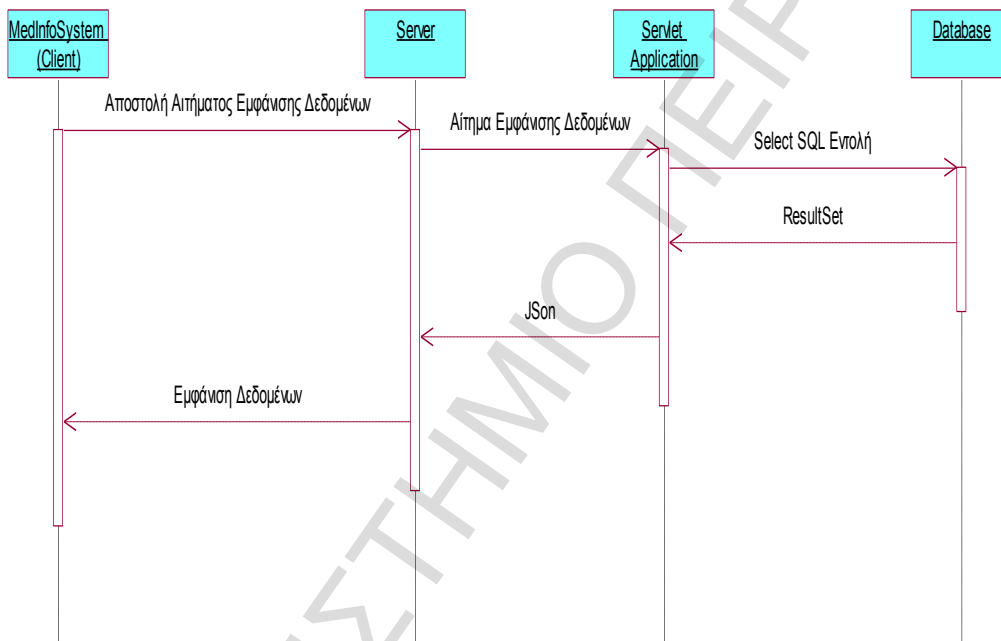
Άλλη μία δομική λειτουργία του Πληροφοριακού Συστήματος είναι η δυνατότητα που προσφέρει να εμφανίζονται σε πίνακες όλες οι καταχωρημένες εγγραφές στη Βάση Δεδομένων. Πρόκειται για μία διαδικασία πολύ λεπτή και ευαίσθητη, για την οποία πρέπει να ληφθούν υπόψη ορισμένοι σημαντικοί παράγοντες. Κατ’ αρχάς, εάν επιθυμούμε το Σύστημά μας να έχει επιτυχία θα πρέπει να διαχειρίζεται μεγάλο όγκο δεδομένων. Αυτό σημαίνει ότι πρέπει τα δεδομένα αυτά να διακινούνται γρήγορα, αξιόπιστα και με ασφάλεια. Όπως είναι λογικό δεν θα ήταν καθόλου αποδοτικό να στέλνονται οι εγγραφές μία μία από τον διακομιστή στον πελάτη. Θα πρέπει λοιπόν να σκεφτούμε έναν τρόπο που να συμπυκνώνει τα δεδομένα σε πακέτα και σε μία μορφή που να μπορούν να διαβαστούν και να μετατραπούν με ευκολία σε πίνακες δεδομένων.

Η διαδικασία ξεκινάει με την αποστολή του αιτήματος από τον χρήστη, ο οποίος έχει επιλέξει τη φύση των δεδομένων που επιθυμεί να αντλήσει, δηλαδή αν θα είναι φάρμακα, εξετάσεις, γιατροί κλπ. Ο διακομιστής επεξεργάζεται το αίτημα και

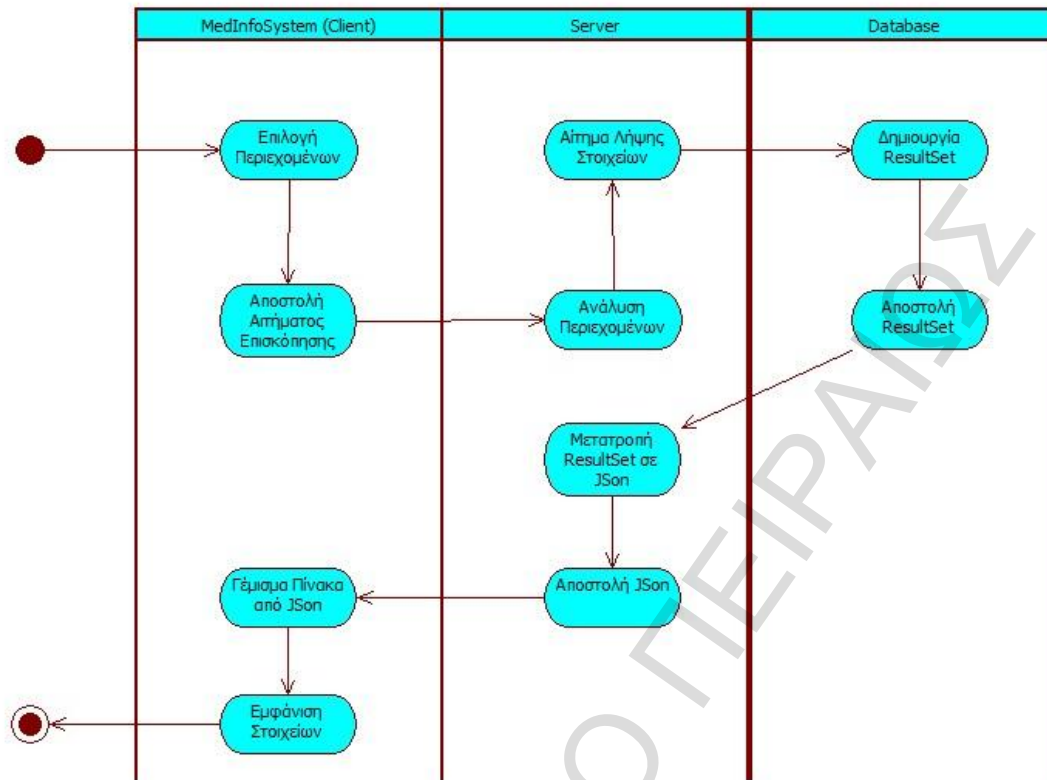
### Μεταπτυχιακή Διατριβή

Παναγιώτης Κοκκότης

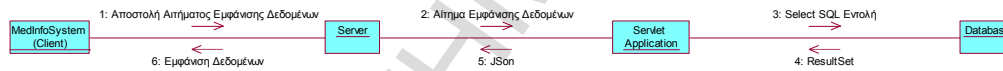
αποστέλλεται μία εντολή τύπου `'SELECT * FROM tablename;'` στη ΒΔ. Αυτό που επιστρέφεται από τη ΒΔ είναι ένα `ResultSet` το οποίο όμως δεν μπορεί να αποσταλεί αυτούσιο στον πελάτη. Γι' αυτόν τον λόγο το πρόγραμμα (`servlet`) το μετατρέπει σε μορφή `JSON`. Μετά την αποστολή του, στην άλλη μεριά αναλαμβάνεται η αποκωδικοποίηση και το γέμισμα του πίνακα με τα δεδομένα, μέχρι να εμφανιστούν στον χρήστη. Αξίζει να σημειωθεί ότι τα δεδομένα ενώ χρησιμοποιούν το πρωτόκολλο `JSON` δεν στέλνονται ως αντικείμενο (κάτι που δεν είναι εφικτό) αλλά σε μορφή `text`. Αυτό εξυπηρετεί τους σκοπούς της ταχύτητας, που είναι εξαιρετικής σημασίας, ειδικά όταν μιλάμε για μεταφορά δεδομένων μέσω διαδικτύου. Θα πρέπει επίσης να αναφερθεί ότι σε αυτή την περίπτωση δεν μπορεί να έχουμε ανεπιτυχή ολοκλήρωση του αιτήματος. Σίγουρα θα επιστραφεί κάτι στον πελάτη ακόμα κι αν αυτό είναι κενό.



Εικόνα 3-9. Διάγραμμα Σειράς (Επισκόπηση)



**Εικόνα 3-10. Διάγραμμα Δραστηριοτήτων (Επισκόπηση)**



**Εικόνα 3-11. Διάγραμμα Συνεργασίας (Επισκόπηση)**

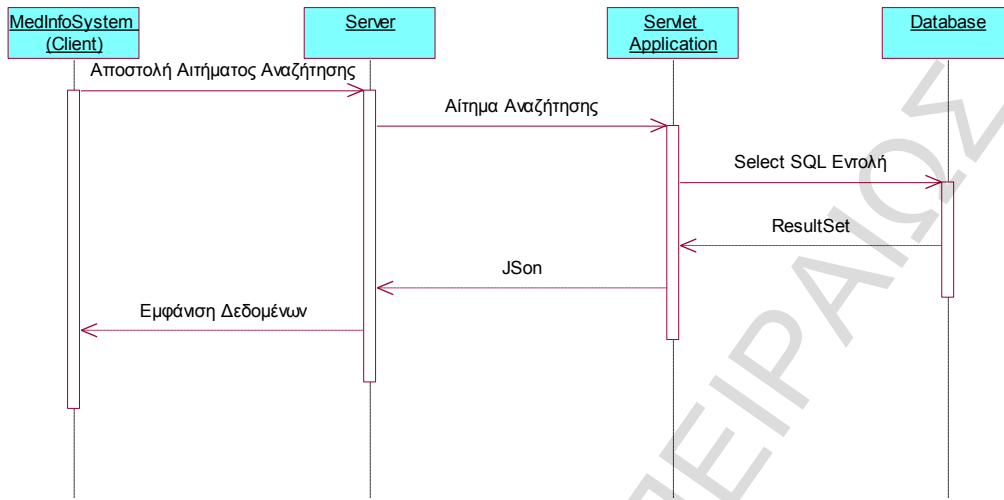
**iv) Αναζήτηση**

Η αναζήτηση είναι μια διαδικασία που μοιάζει αρκετά με αυτή της επισκόπησης. Εδώ όμως ο χρήστης έχει πληκτρολογήσει μία λέξη ή κάποια γράμματα που θεωρεί ότι ταιριάζουν με τα κριτήρια που αναζητά και αυτά στέλνονται στον διακομιστή μαζί με το σχετικό αίτημα. Η διατύπωση του ερωτήματος προς την ΒΔ είναι η εξής: `'SELECT * FROM tablename WHERE col1 LIKE %keyword% OR col2 LIKE %keyword% OR col3 LIKE %keyword% ...;'`

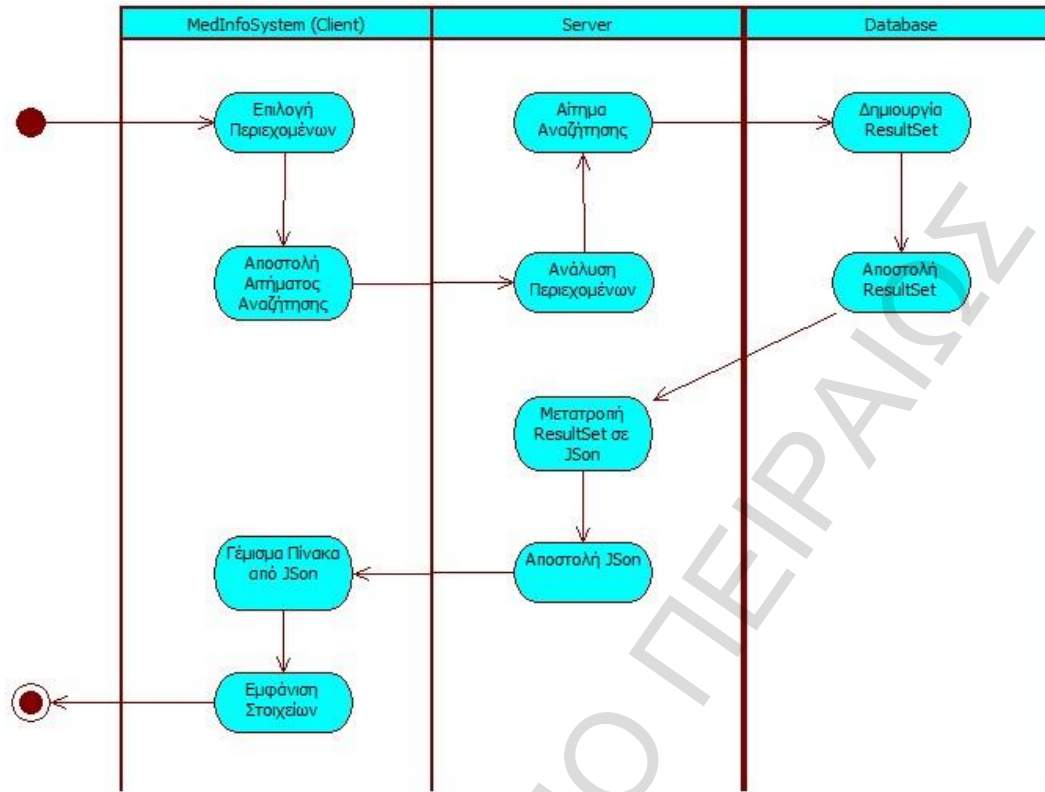
Ακολουθεί η διαδικασία μετατροπής του ResultSet σε μορφή JSON και η αποστολή του στον πελάτη. Σε αυτό το σημείο θα πρέπει να σημειωθεί ότι η διατύπωση των παραπάνω SQL ερωτημάτων δεν είναι πάντα σε αυτή την μορφή. Σε κάποιους πίνακες που χρησιμοποιούν ξένα κλειδιά υπάρχουν στήλες που αντί για ένα όνομα έχουν γραμμένο το αντίστοιχο id. Προκειμένου λοιπόν ο χρήστης να μην δει έναν αριθμό στη θέση που θα περίμενε π.χ. να δει το όνομα του γιατρού πραγματοποιούνται κάποια joins που λένε ουσιαστικά στη Βάση Δεδομένων να ψάξει στους πίνακες που αναφέρονται τα ξένα κλειδιά και να εμφανίσει τα ονόματα αντί για

τους αριθμούς. Η μορφή των ερωτημάτων αυτών είναι η παρακάτω:

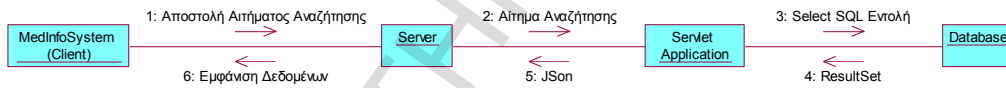
```
'SELECT * FROM tablename t1 LEFT JOIN tablename t2 ON t1.value1 = t2.value_id ...;'
```



**Εικόνα 3-12. Διάγραμμα Σειράς (Αναζήτηση)**



**Εικόνα 3-13. Διάγραμμα Δραστηριοτήτων (Αναζήτηση)**

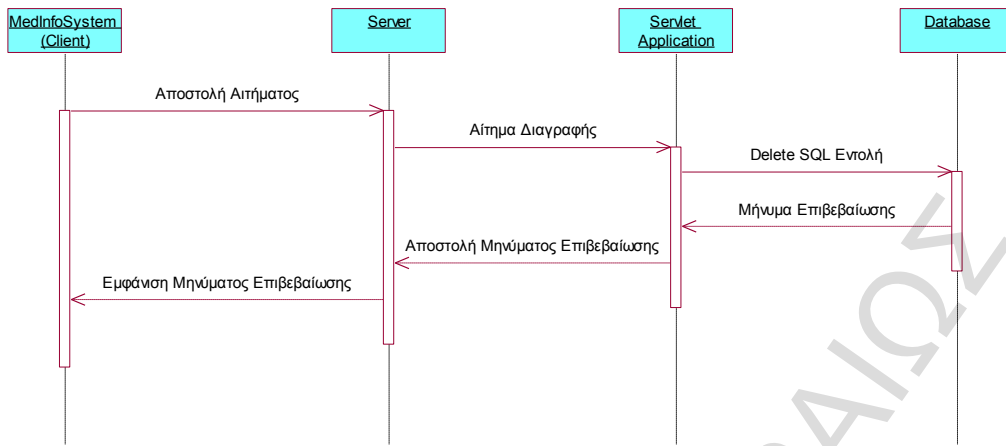


**Εικόνα 3-14. Διάγραμμα Συνεργασίας (Αναζήτηση)**

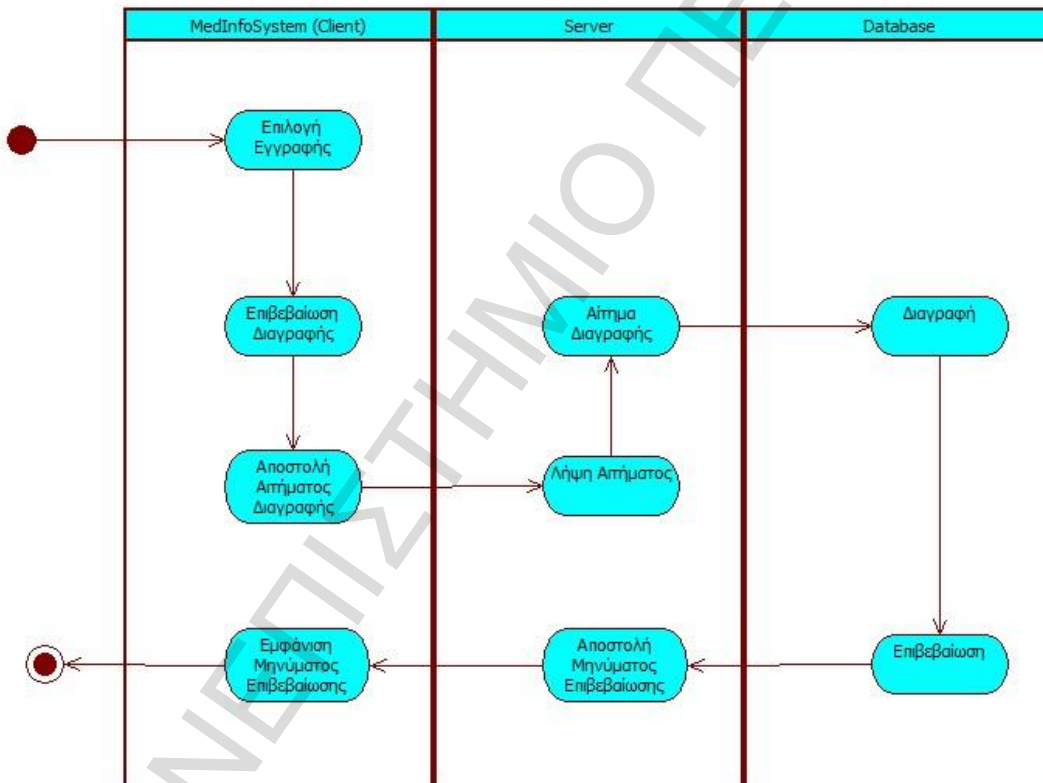
**ν) Διαγραφή**

Σε αυτή την περίπτωση το πρόγραμμα φροντίζει να στείλει στον διακομιστή το id της εγγραφής που έχει επιλέξει ο χρήστης να διαγράψει. Η εντολή SQL που αποστέλλεται στη ΒΔ έχει τη μορφή 'DELETE FROM tablename WHERE id = ?;'. Στη συνέχεια επιβεβαιώνεται η επιτυχής ολοκλήρωση του αιτήματος εάν όλα πήγαν καλά.

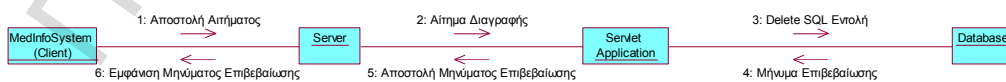




**Εικόνα 3-15. Διάγραμμα Σειράς (Διαγραφή)**

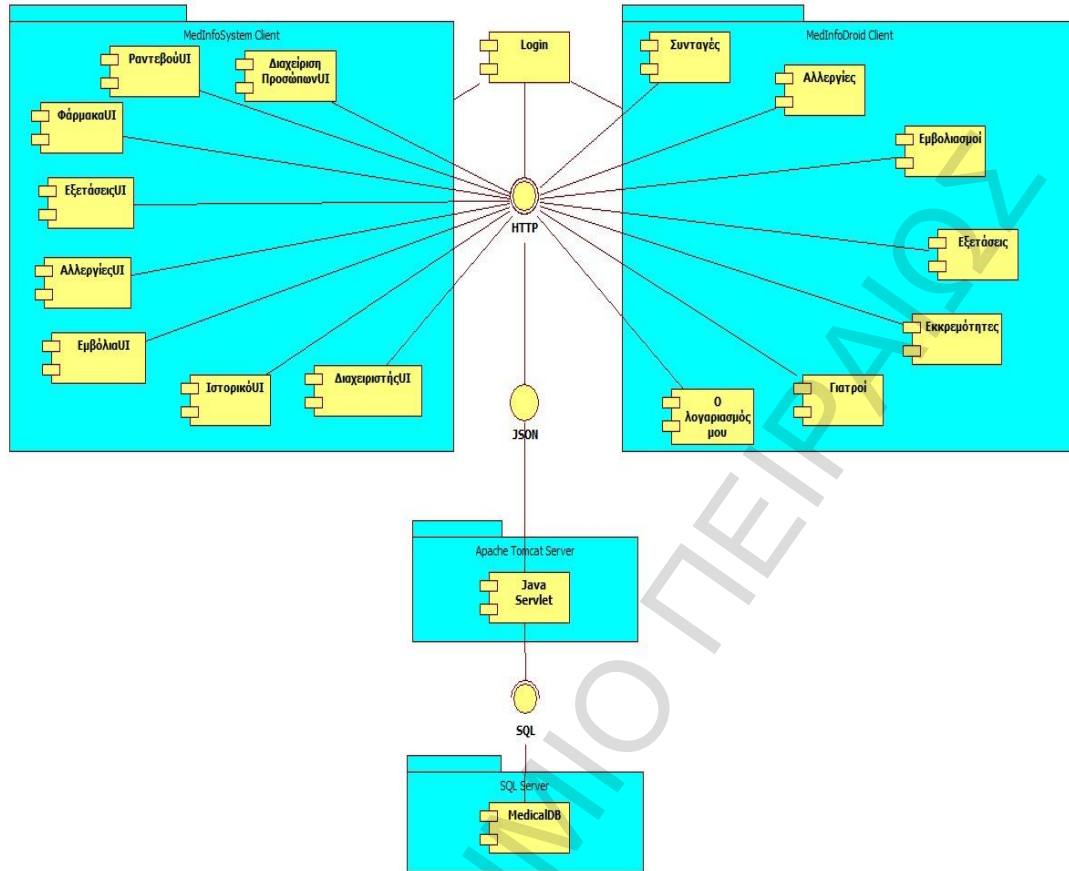


**Εικόνα 3-16. Διάγραμμα Δραστηριοτήτων (Διαγραφή)**



**Εικόνα 3-17. Διάγραμμα Συνεργασίας (Διαγραφή)**

**c) Διάγραμμα Εξαρτημάτων**



**Εικόνα 3-18. Διάγραμμα Εξαρτημάτων**

Στο παραπάνω διάγραμμα εξαρτημάτων διακρίνονται τα μέρη που αποτελούν το Πληροφοριακό Σύστημα. Πρόκειται για τέσσερα ξεχωριστά πακέτα που συνεργάζονται μεταξύ τους για την επίτευξη των στόχων του Συστήματος και είναι τα εξής:

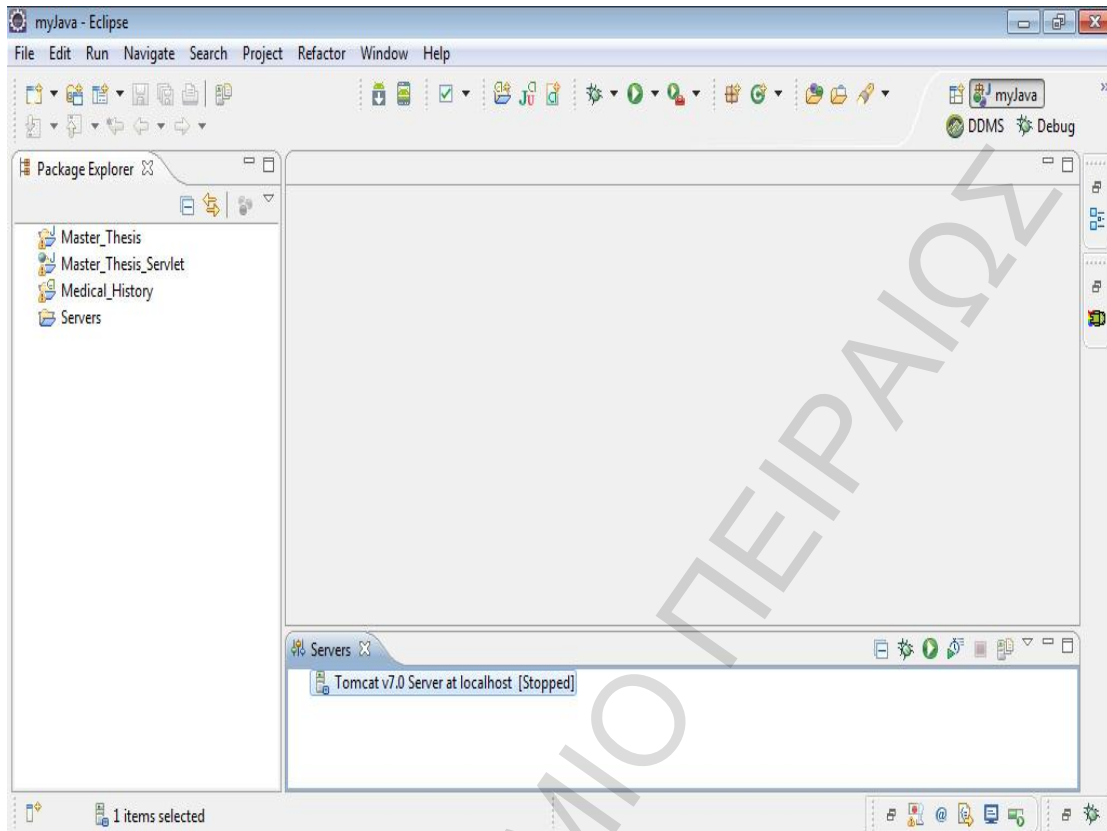
- Το Κεντρικό πρόγραμμα (MedInfoSystem): Είναι η βασική διεπαφή που περιλαμβάνει ξεχωριστές καρτέλες, κάθε μία από τις οποίες επιτελεί συγκεκριμένες λειτουργίες
- Η εφαρμογή για φορητές συσκευές (MedInfoDroid): Μπορεί να εγκατασταθεί σε όλες τις συσκευές με λειτουργικό Android και περιλαμβάνει επιλογές για την άντληση προσωπικών δεδομένων από τους επιβλεπόμενους.
- Ο Διακομιστής: Είναι ένας server Apache Tomcat, στον οποίο τρέχει ένα πρόγραμμα java servlet. Η επικοινωνία του με τους πελάτες επιτυγχάνεται μέσω του πρωτοκόλλου http αλλά και μέσω της τυποποιημένης μορφής json. Στη συνέχεια χρησιμοποιείται η γλώσσα sql για την επικοινωνία του με τη ΒΔ.
- Η Βάση Δεδομένων: Βρίσκεται εγκατεστημένη στον ίδιο υπολογιστή με τον apache server, στον οποίο υπάρχει εγκατεστημένος και ένας



# Κεφάλαιο 4

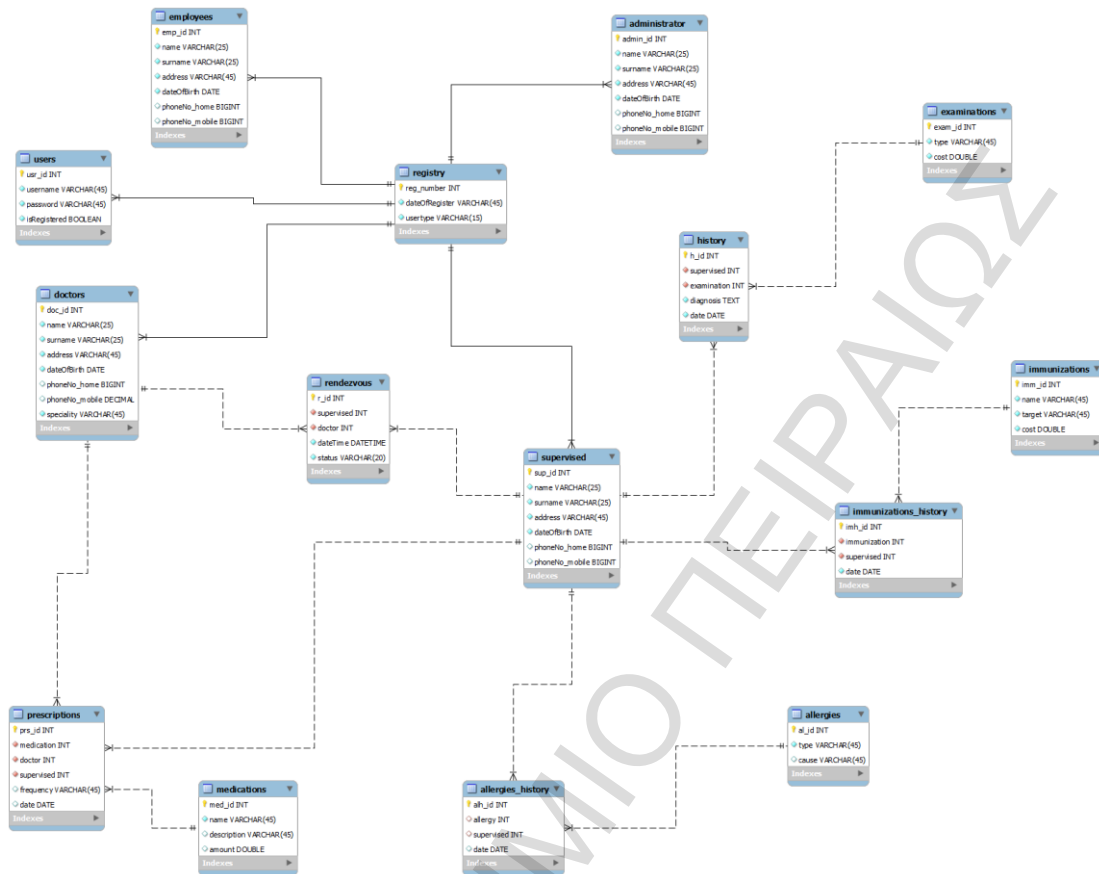
## Παρουσίαση Συστήματος

## 4. Παρουσίαση Συστήματος



**Εικόνα 4-1. Eclipse UI**

Το Ιατρικό Πληροφοριακό Σύστημα που αναπτύσσεται στην παρούσα Μεταπτυχιακή Διατριβή αποτελείται από τρία projects και την Βάση Δεδομένων. Πρόκειται για το κύριο μέρος του συστήματος, την εφαρμογή για συσκευές Android και ένα πρόγραμμα που τρέχει στον διακομιστή (server) και είναι τύπου Java Servlet. Η γλώσσα Java είναι η βασική τεχνολογία που χρησιμοποιείται στο μεγαλύτερο μέρος της εργασίας σε συνδυασμό με άλλες βοηθητικές γλώσσες και τεχνολογίες, όπως η XML, η Json και η SQL. Έχει χρησιμοποιηθεί το περιβάλλον Eclipse (Εικόνα 4-1) και άλλα εργαλεία, όπως το Mysql Workbench για τον σχεδιασμό και την διαχείριση της Βάσης Δεδομένων. Το σύστημα ακολουθεί το πρότυπο πελάτη - διακομιστή (Client - Server). Για την ανάπτυξή του έχουν γραφτεί πάνω από **10.000** γραμμές κώδικα (Java, xml, sql) που περιλαμβάνονται σε περίπου **100** κλάσεις και αρκετά ακόμα βοηθητικά αρχεία. Τα βασικά μέρη του συστήματος έχουν ως εξής:

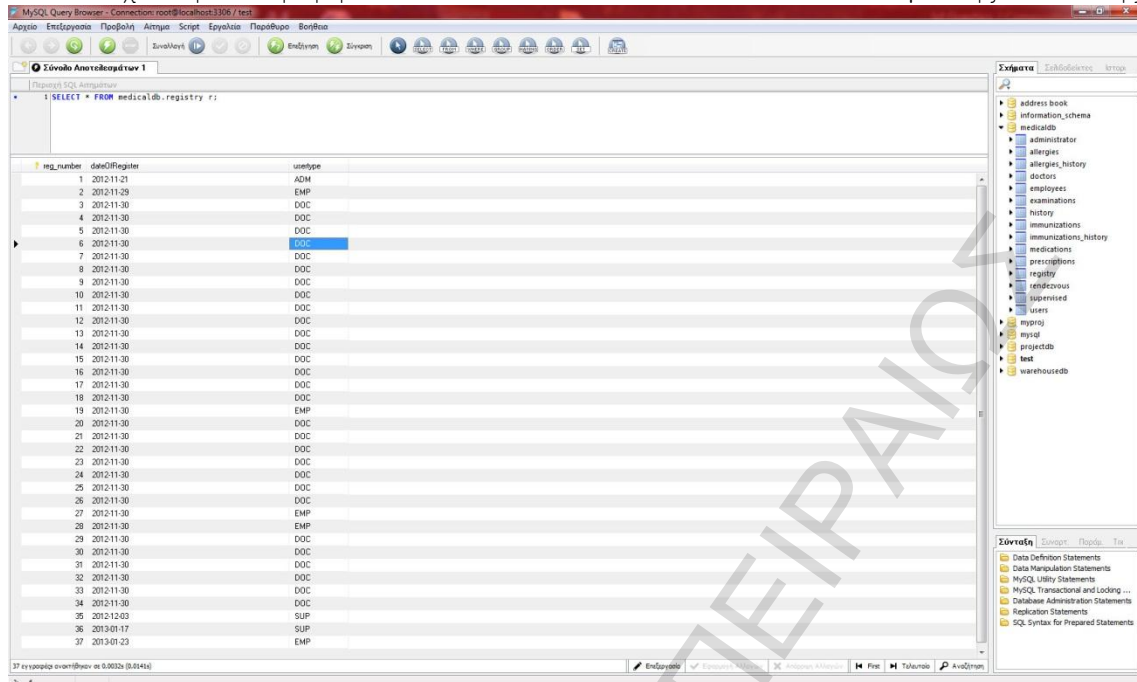


**Εικόνα 4-2. EER διάγραμμα**

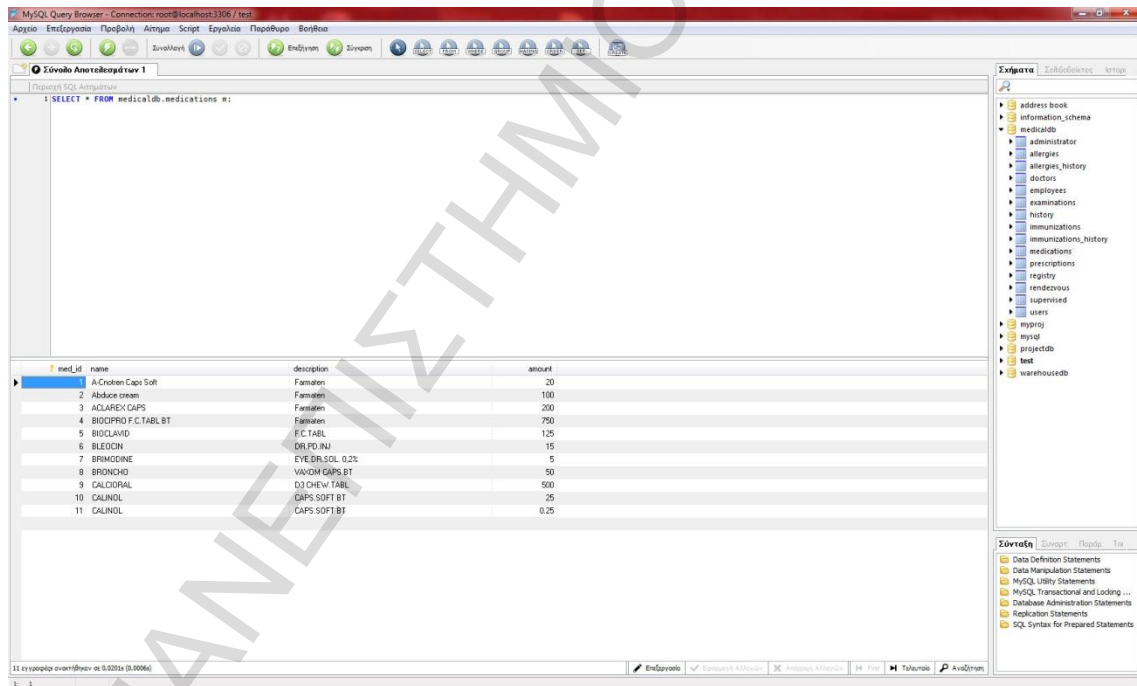
Είναι υλοποιημένη σε MySQL και αποτελείται από 15 πίνακες (Εικόνα 4-2). Βρίσκεται σε έναν κεντρικό διακομιστή, κατά προτίμηση μέσα στην ιατρική μονάδα και, για λόγους ασφαλείας, δεν είναι εκτεθειμένη στο κοινό. Κανένας χρήστης του συστήματος δεν έχει άμεση πρόσβαση σε αυτή. Περιέχει καταχωρημένες όλες τις πληροφορίες που διαχειρίζεται το Σύστημα και επικοινωνεί με τους χρήστες μόνο μέσω ενός προγράμματος servlet το οποίο τρέχει (συνήθως) στον ίδιο διακομιστή και απευθύνει ερωτήματα sql για την καταγραφή και άντληση δεδομένων, τα οποία στη συνέχεια μετατρέπονται σε μορφή Json και αποστέλλονται στα διάφορα λογισμικά - πελάτες. Τα προγράμματα που διαχειρίζονται οι χρήστες μπορούν να απευθύνουν ερωτήματα του τύπου 'Φέρε μου τους καταχωρημένους Γιατρούς' προς τον διακομιστή αλλά δεν μπορούν να επικοινωνήσουν απευθείας με τη Βάση Δεδομένων. Οι πίνακες που έχουν δημιουργηθεί για την καταχώρηση των δεδομένων αναλύονται ως εξής:

- ο Το μητρώο: Είναι ένας πίνακας που περιέχει τα βασικά χαρακτηριστικά της καταχώρησης νέου προσώπου στο σύστημα, όπως η ημερομηνία εγγραφής του, η ιδιότητά του και ένας αύξων αριθμός, που είναι μοναδικός και τον ξεχωρίζει από τους υπόλοιπους. Από τον συνδυασμό αυτού του αριθμού με την ιδιότητα προκύπτει ο αριθμός μητρώου. Για παράδειγμα ο A.M. DOC0000039 ανήκει σε έναν γιατρό που αποτελεί το 39<sup>ο</sup>

- άτομο που καταγράφηκε στο σύστημα.
- ο **Εργαζόμενοι, Διαχειριστές, Γιατροί, Επιβλεπόμενοι:** Είναι πίνακες που χρησιμοποιούνται για την καταχώρηση των δημογραφικών στοιχείων των χρηστών, όπως το ονοματεπώνυμο, η διεύθυνση, η ημερομηνία γέννησης κλπ. Ο αύξων αριθμός που αποτελεί το πρωτεύον κλειδί στον πίνακα 'Μητρώο' είναι και εδώ πρωτεύον κλειδί αλλά έχει παράλληλα και την ιδιότητα του ξένου κλειδιού.
  - ο **Φάρμακα, Εξετάσεις, Αλλεργίες, Εμβόλια:** Είναι πίνακες που συγκρατούν τα δεδομένα των ιατρικών αντικειμένων που διαχειρίζεται το Σύστημα. Π.χ. Το είδος των εξετάσεων, το κόστος τους ή ο τύπος και η ποσότητα σε mg των φαρμάκων είναι ορισμένα στοιχεία που αποθηκεύονται στους παραπάνω πίνακες.
  - ο **Συνταγές, Ιστορικό Εξετάσεων, Αλλεργιών και Εμβολίων:** Αυτοί οι πίνακες αποθηκεύουν τα ιστορικά στοιχεία των επιβλεπόμενων και τις απαραίτητες ημερομηνίες. Στα περιεχόμενά τους καταγράφονται ως ξένα κλειδιά από τους πίνακες των προηγούμενων κατηγοριών τα διάφορα ids αλλά στο Σύστημα εμφανίζονται τα κανονικά ονόματα, μέσω των Joins που πραγματοποιούνται στα ερωτήματα, τα οποία απευθύνει το servlet στη ΒΔ. Ο πίνακας του ιστορικού εξετάσεων περιέχει και μελλοντικές προγραμματισμένες εξετάσεις.
  - ο **Ραντεβού:** Ο πίνακας αυτός χρησιμοποιείται για την καταχώρηση των ραντεβού των γιατρών με τους επιβλεπόμενους. Περιέχει στοιχεία ώρας και ημερομηνίας και μία στήλη που εμφανίζει την κατάσταση του κάθε ραντεβού, ανάλογα με το αν εκκρεμεί επιβεβαίωση, εάν ακυρώθηκε, εάν ολοκληρώθηκε κλπ.
  - ο **Χρήστες:** Ο συγκεκριμένος πίνακας χρησιμοποιείται για να αποθηκεύονται τα στοιχεία όσων από τους χρήστες μπορούν να έχουν πρόσβαση στο Σύστημα. Συγκρατεί το όνομα χρήστη και τον κωδικό πρόσβασης. Όπως θα δούμε παρακάτω ένας χρήστης μπορεί να δημιουργηθεί μέσα από το Σύστημα αλλά για να μπορέσει να αποκτήσει πρόσβαση σε αυτό θα πρέπει να εξουσιοδοτηθεί από τον Διαχειριστή. Για τον λόγο αυτό υπάρχει μία στήλη 'isRegistered' με περιεχόμενο τύπου Boolean, με default τιμή false, που μετατρέπεται σε true από τον διαχειριστή, εφόσον επιβεβαιωθεί ότι ο χρήστης έχει τη δυνατότητα να εισέρχεται στο Σύστημα. Δεν έχουν όλοι οι χρήστες τα ίδια δικαιώματα, όπως θα δούμε και στη συνέχεια. Για παράδειγμα, ο επιβλεπόμενος μπορεί να έχει πρόσβαση μόνο στα προσωπικά του δεδομένα, μέσω του κινητού του τηλεφώνου.



Εικόνα 4-3. Εγγραφές στο μητρώο



Εικόνα 4-4. Εγγραφές φαρμάκων

## To servlet

Είναι ένα είδος προγράμματος που μπορεί να εκτελεστεί όταν ζητηθεί από έναν browser. Τα servlets είναι γραμμένα σε Java, φορτώνονται στον διακομιστή Web και εκτελούνται όταν μία κατάλληλη εντολή HTTP που ζητά την εκτέλεση τους λαμβάνεται από τον διακομιστή. Τα servlets έχουν ορισμένα σημαντικά χαρακτηριστικά:

- Μπορούν να εκτελεστούν χωρίς αλλαγές σε διάφορους τύπους διακομιστών.
- Από την στιγμή που τα servlets είναι γραμμένα σε Java έχουν πρόσβαση σε διάφορα εργαλεία της Java όπως η CORBA, RMI, εργαλεία ασφαλείας της Java και εργαλεία σύνδεσης βάσεων δεδομένων.
- Τα servlets είναι εγκατεστημένα στην μνήμη. Με τον προγραμματισμό CGI σε γλώσσες όπως η Perl, κάθε φορά που ένα καινούργιο αίτημα προωθείται από τον Web browser μια καινούργια επεξεργασία πρέπει να ξεκινήσει και να τερματιστεί με την λήξη της σύνδεσης και το κατάλληλο πρόγραμμα πρέπει να φορτωθεί και να ξεφορτωθεί από τη μνήμη. Αυτό σημαίνει ότι το φορτίο μπορεί να είναι μεγάλο για τον διακομιστή Web.
- Από την στιγμή που τα servlets βασίζονται στην Java περιλαμβάνουν όλα τα χαρακτηριστικά ασφαλείας της γλώσσας.
- Τα servlets μπορούν να διατηρήσουν την κατάσταση τους, να διατηρήσουν κάποιο είδος μνήμης, ανάμεσα στα αιτήματα, κάτι που, στα πρώτα βήματα του Παγκόσμιου Ιστού, ήταν δύσκολο. Αυτό σημαίνει ότι τα servlets μπορούν να θυμηθούν δεδομένα και λεπτομέρειες ενός προηγούμενου αιτήματος.
- Το αντικειμενοστραφές μοντέλο της Java βοηθά τον κώδικα της να είναι πιο κομψός από κώδικα γραμμένο σε άλλες γλώσσες όπως η C και η Perl. Αν εξετάσετε τα προγράμματα Perl CGI θα δείτε, συχνά, ότι είναι τεράστια, μονοκόμματα και δυσνόητα. Η προσανατολισμένη σι' αντικείμενα φύση της Java παρέχει επίσης και την δυνατότητα μεγαλύτερης επαναχρησιμοποίησης κώδικα.
- Τα servlets, επειδή ενσωματώνονται καλά με τις τεχνολογίες ασφαλείας που σχετίζονται με την Java, μπορούν να διαμορφωθούν ώστε να παρέχουν υψηλά επίπεδα ασφαλείας.

Ένα servlet δεν περιέχει κύρια μέθοδο και πρέπει να υλοποιεί την διεπαφή `javax.servlet.Servlet`. Για ένα servlet που προορίζεται για το Web και το οποίο θα χρησιμοποιεί HTTP, αυτό συνήθως γίνεται φτιάχνοντας μια υποκλάση της `HttpServlet`. Όταν το κατάλληλο αίτημα λαμβάνεται από τον διακομιστή Web εκτελείται ανάλογα με το αίτημα η κατάλληλη μέθοδος της κλάσης. Για να υλοποιήσει ο προγραμματιστής την απαραίτητη λειτουργικότητα πρέπει να γράψει τον κώδικα αυτών των μεθόδων.

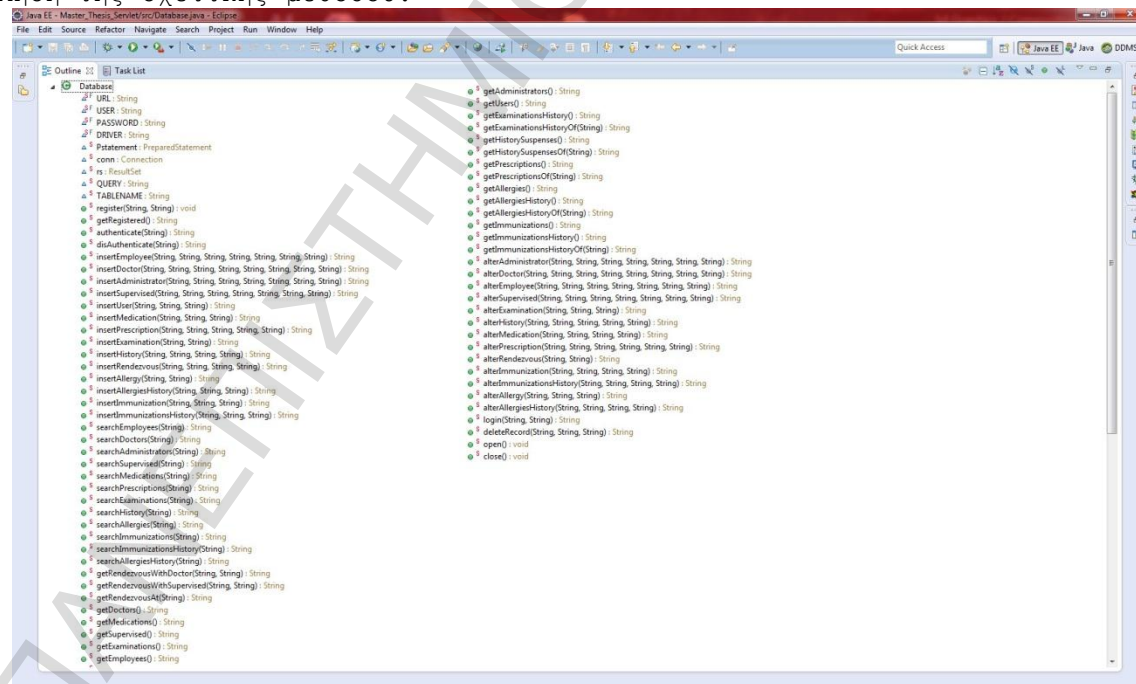
Ένα από τα πιο σημαντικά χαρακτηριστικά ενός Πληροφοριακού Συστήματος και μάλιστα ιατρικού περιεχομένου είναι η ασφάλεια. Για να επιτευχθεί η μεγαλύτερη δυνατή ασφάλεια λοιπόν για τα δεδομένα μας η Βάση Δεδομένων δεν είναι εκτεθειμένη στο κοινό. Οι διάφοροι χρήστες, μέσω των εφαρμογών που έχουν στη διάθεσή τους απευθύνουν ερωτήματα στον διακομιστή και αυτός αναλαμβάνει να επικοινωνήσει με τη ΒΔ και να επιστρέψει τα δεδομένα που έχουν ζητηθεί ή να τα καταγράψει. Για την



Μεταπτυχιακή Διατριβή

επικοινωνία μεταξύ πελάτη – διακομιστή έχει επιλεγεί η τυποποιημένη μορφή Json εξαιτίας της απλότητας, της ταχύτητας και της πολύ καλής συνεργασίας της με τη Java. Τα δεδομένα τυποποιούνται σε αυτή τη μορφή από τον διακομιστή και, αφού αποσταλούν, μετατρέπονται στην κατάλληλη μορφή, στην πλευρά του πελάτη, για την παρουσίασή τους στον χρήστη. Το πρόγραμμα τρέχει σε έναν κεντρικό υπολογιστή που, κατά προτίμηση, βρίσκεται σε ασφαλές μέρος μέσα στην ιατρική μονάδα.

Αποτελείται από 8 κλάσεις. Οι 6 από αυτές είναι αφιερωμένες στη διαχείριση των δεδομένων και χωρισμένες ανά κατηγορία, με την ίδια λογική που χωρίζονται και οι καρτέλες στην κεντρική εφαρμογή. Έτσι, έχουμε κλάσεις για τα φάρμακα, τις αλλεργίες, τα εμβόλια, τις εξετάσεις, τα ραντεβού και μία με το μητρώο, όπου χειρίζεται όλων των ειδών τα πρόσωπα. Η κάθε μία από αυτές κληρονομεί από την HttpServlet και έχει δύο βασικές μεθόδους, την doGet και την doPost, οι οποίες ενεργοποιούνται ανάλογα με την επιλεγμένη μέθοδο αποστολής δεδομένων. Στο παρόν Σύστημα έχει επιλεγεί η μέθοδος 'GET'. Στο εσωτερικό της σχετικής μεθόδου γίνεται έλεγχος για το είδος της ερώτησης που ήρθε από τον πελάτη. Στη συνέχεια καλείται η κλάση Database, που έχει κατασκευαστεί για να επικοινωνεί με τη Βάση Δεδομένων. Στο εσωτερικό της έχουν δηλωθεί όλα τα απαραίτητα στοιχεία, όπως ο κωδικός και το όνομα χρήστη του διαχειριστή της ΒΔ. Κάθε μία από τις πολλές μεθόδους της περιέχει κώδικα SQL και επιστρέφει δεδομένα σε μορφή αλφαριθμητικού. Τα δεδομένα αυτά είτε είναι ένα μήνυμα επιβεβαίωσης είτε τα ζητούμενα στοιχεία, σε μορφή JSON. Για την μετατροπή των ResultSets που επιστρέφονται από τη ΒΔ σε JSON, έχει δημιουργηθεί μία κλάση JSONParser, η οποία αναλαμβάνει αυτόματα τη λειτουργία αυτή με την κλήση της σχετικής μεθόδου.



Εικόνα 4-5. Απεικόνιση κλάσης Database.java



**Εικόνα 4-6. Απεικόνιση κλάσης Immunizations.java**

Οι δύο παραπάνω Εικόνες (4-5, 4-6) παρουσιάζουν τη μορφή των κλάσεων, έτσι όπως έχουν σχεδιαστεί και υλοποιηθεί, όσον αφορά το πρόγραμμα που τρέχει στον server. Στην Εικόνα 4-5 φαίνεται η κλάση Database, η οποία περιέχει όλα τα χαρακτηριστικά και τις μεθόδους που χρησιμεύουν για την επικοινωνία απευθείας με τη Βάση Δεδομένων. Στα χαρακτηριστικά περιλαμβάνονται το όνομα της ΒΔ, μαζί με το url, το όνομα του χρήστη (που συνήθως συναντάται ως root) ο κωδικός κ.α. Οι μέθοδοι χωρίζονται σε κατηγορίες και είναι ομαδοποιημένες για να διευκολύνουν τη δουλειά του προγραμματιστή - αναλυτή. Οι κατηγορίες αυτές έχουν σχέση με τις 5 βασικές λειτουργίες που επιτελούνται και έχουν αναφερθεί νωρίτερα, κατά τον σχεδιασμό του Συστήματος. Αυτές είναι οι εξής:

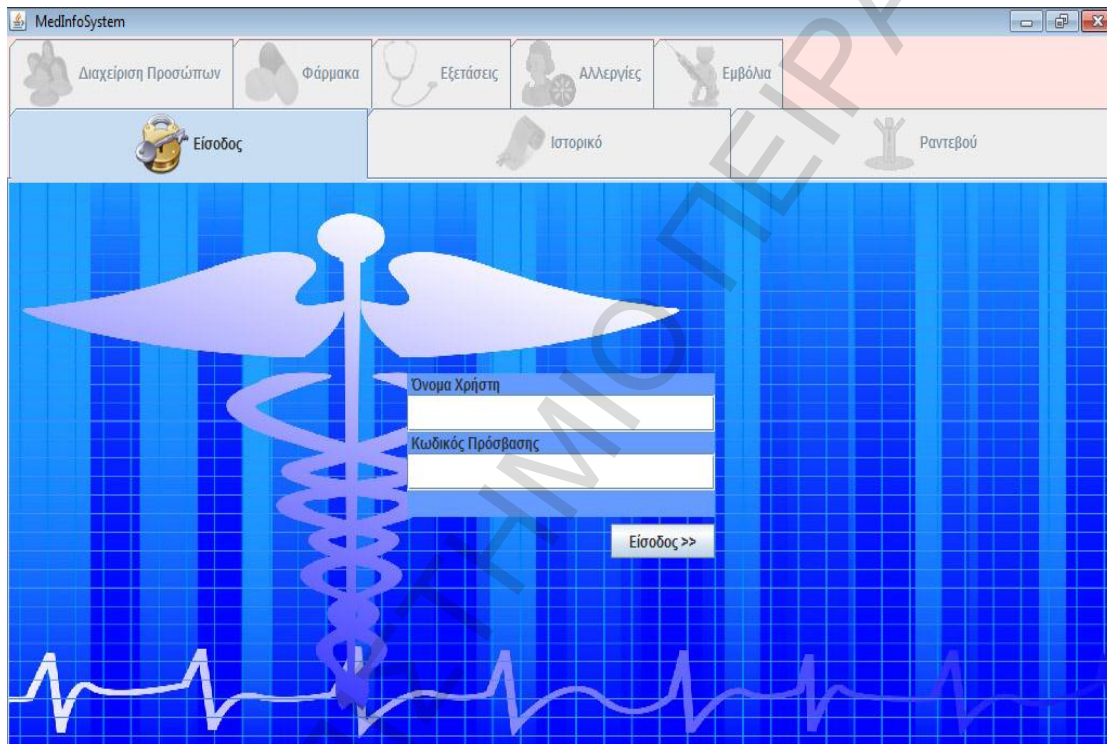
- **Insert:** Πρόκειται για μια σειρά από μεθόδους που παίρνουν ως παραμέτρους τα στοιχεία που καταχώρησε ο χρήστης και αναλαμβάνουν, μέσω του κατάλληλου κώδικα sql που θα δούμε παρακάτω, να καταχωρήσουν τα δεδομένα στη ΒΔ.
- **Search:** Οι μέθοδοι αυτές πρέπει οπωσδήποτε να περιέχουν, μεταξύ των παραμέτρων τους, μία λέξη κλειδί και χρησιμεύουν για την αναζήτηση εγγραφών στη ΒΔ. Υπάρχουν και οι σχετικές μέθοδοι που αναζητούν εγγραφές για μια συγκεκριμένη κατηγορία.
- **Get:** Εξυπηρετούν τη λειτουργία της επισκόπησης και αναλαμβάνουν να ζητήσουν από τη ΒΔ ένα σύνολο εγγραφών, που συνήθως περιέχονται σε έναν πίνακα. Υποκατηγορία αυτών των μεθόδων είναι αυτές που ζητούν τις εγγραφές που αφορούν συγκεκριμένο πρόσωπο ή αντικείμενο.
- **Alter:** Κατά την επεξεργασία των δεδομένων, αυτές οι μέθοδοι καλούνται με σκοπό να τροποποιήσουν συγκεκριμένες καταχωρήσεις της ΒΔ.
- **Delete:** Πρόκειται για μία μόνο μέθοδο, που παίρνει παραμέτρους τον πίνακα, από τον οποίο θέλουμε να γίνει η διαγραφή, τη στήλη στην οποία αναφερόμαστε και τον αύξοντα αριθμό εγγραφής (id).
- Επίσης, υπάρχουν κάποιες βοηθητικές μέθοδοι, όπως η open και η close, που καλούνται σε κάθε μέθοδο για να ανοίξουν αρχικά και μετά να κλείσουν την επικοινωνία με τη ΒΔ. Η

μέθοδος `login` αναλαμβάνει να εξερευνήσει την ορθότητα των στοιχείων κατά τη διαδικασία εισόδου στο Πληροφοριακό Σύστημα και η `register` να καταγράψει τα στοιχεία στο μητρώο.

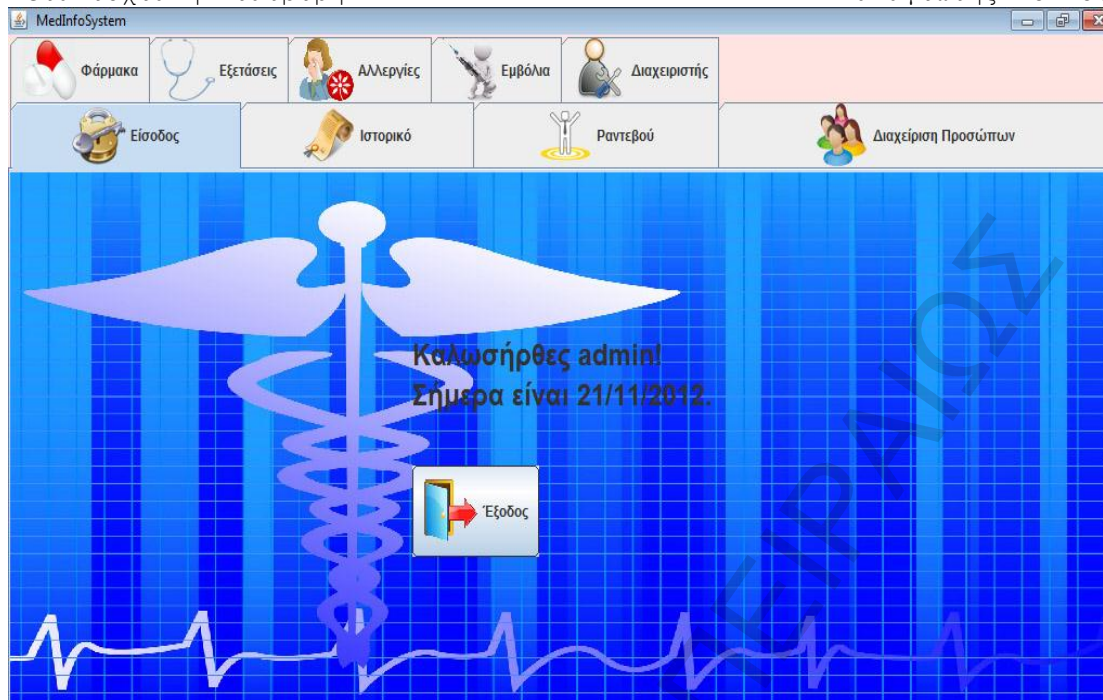
Το παράδειγμα της Εικόνας 4-6 αντικατοπτρίζει τον τρόπο με τον οποίο είναι υλοποιημένες οι υπόλοιπες κλάσεις. Υπάρχουν δύο βασικές μέθοδοι (η `doGet` και η `doPost`) που, ανάλογα με την επιλογή του προγραμματιστή αναλαμβάνουν να επικοινωνήσουν με τον `client` δεχόμενες αλλά και αποστέλλοντας πληροφορίες. Μέσα στο κυρίως σώμα τους υπάρχουν υποθέσεις (`if statements`) που ξεχωρίζουν τα αιτήματα και αναλαμβάνουν να καλέσουν την κατάλληλη μέθοδο από την κλάση `Database`. Τέλος, υπάρχει και η κλάση `JsonParser` η οποία περιλαμβάνει τη μέθοδο που μετατρέπει τις πληροφορίες από τη ΒΔ σε μορφή `JSON` για να αποσταλούν πίσω στον `client`.

## Το Λογισμικό

Είναι μία εφαρμογή που τρέχει σε οποιονδήποτε υπολογιστή είναι εφοδιασμένος με το κατάλληλο περιβάλλον της Java. Είναι ελαφρύ και ικανό να εκτελεστεί ακόμα και σε υπολογιστές χαμηλών δυνατοτήτων και σε οποιοδήποτε λειτουργικό σύστημα (Windows, Linux, Mac). Είναι μια γραφική διεπαφή, σχεδιασμένη με τρόπο που να είναι προσιτή και φιλική για οποιονδήποτε χρήστη. Θεωρείται πολύ σημαντικό, για την εύρυθμη λειτουργία μιας μονάδας και την καλύτερη εξυπηρέτηση των 'πελατών' της, οι χρήστες του Συστήματος να διαθέτουν εμπειρία, η οποία αποκτάται με την χρήση του. Για τον σκοπό αυτό το πρόγραμμα είναι σχεδιασμένο με τέτοιο τρόπο ώστε να είναι φιλικό και εύκολο στην εκμάθηση.

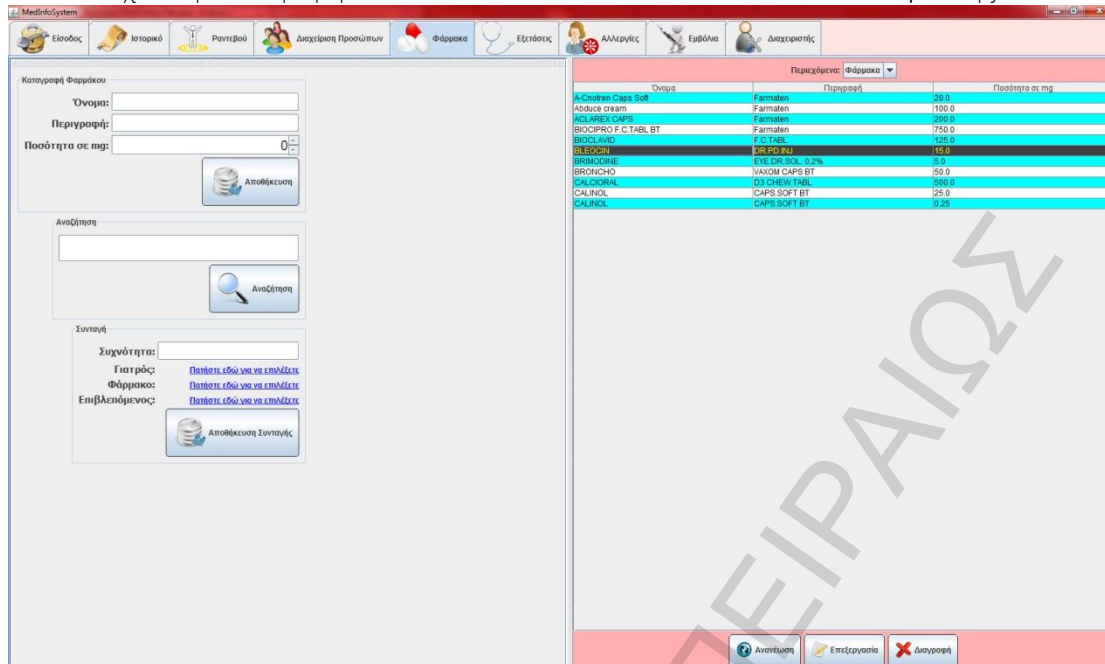


Εικόνα 4-7. Καρτέλα Εισόδου



**Εικόνα 4-8. Επιτυχής Είσοδος**

Η πρώτη επαφή του χρήστη με το πρόγραμμα είναι η καρτέλα εισόδου της Εικόνας 4-8. Απαιτείται όνομα χρήστη και κωδικός πρόσβασης και, όπως φαίνεται, οι υπόλοιπες λειτουργίες του Συστήματος είναι απενεργοποιημένες. Εφόσον δοθούν τα σωστά στοιχεία και διαπιστωθεί η εγκυρότητά τους οι άλλες καρτέλες ενεργοποιούνται και ο χρήστης έχει πλέον πρόσβαση στο Σύστημα. Εάν ο χρήστης αυτός έχει δικαιώματα Διαχειριστή ανοίγει μία επιπλέον κρυφή καρτέλα (Εικόνες 4-8, 4-9) που συγκεντρώνει τις μοναδικές λειτουργίες οι οποίες μπορούν να επιτελεστούν από έναν χρήστη με πλήρη πρόσβαση στη Βάση Δεδομένων και θα δούμε παρακάτω. Η διαπίστωση της ιδιότητας του επίδοξου χρήστη είναι εξαιρετικής σημασίας, δεδομένου ότι στοιχεία εισόδου διαθέτει και ένας επιβλεπόμενος, για να έχει πρόσβαση από το κινητό του μόνο στα προσωπικά του δεδομένα, αλλά δεν μπορεί να εισέλθει στο Πληροφοριακό Σύστημα.

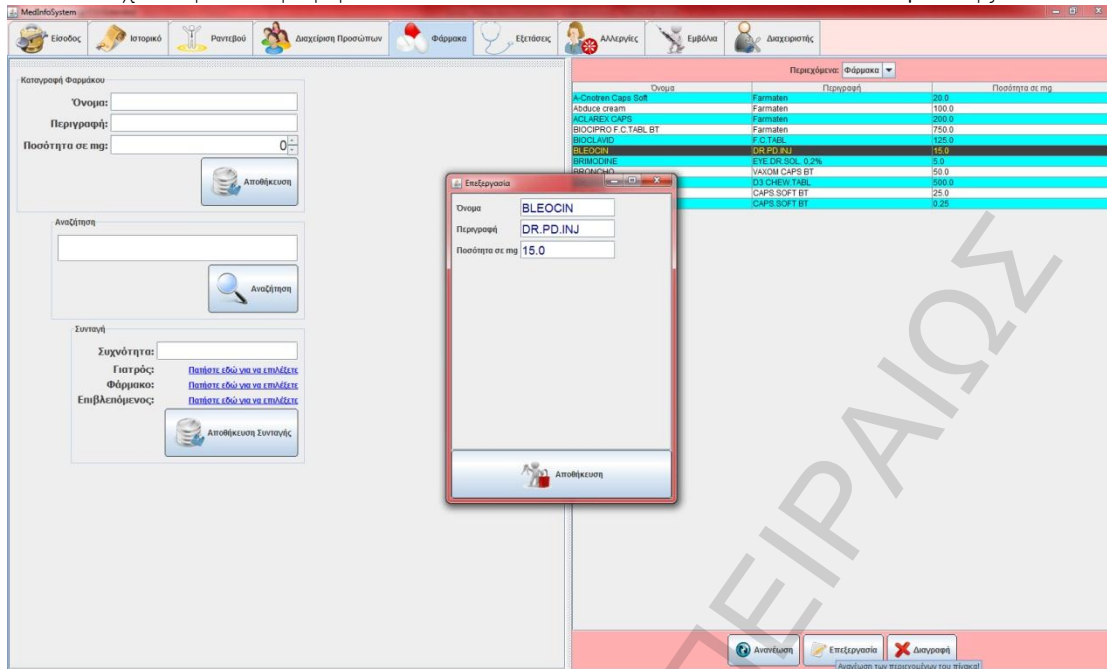


**Εικόνα 4-9. Καρτέλα Φαρμάκων**

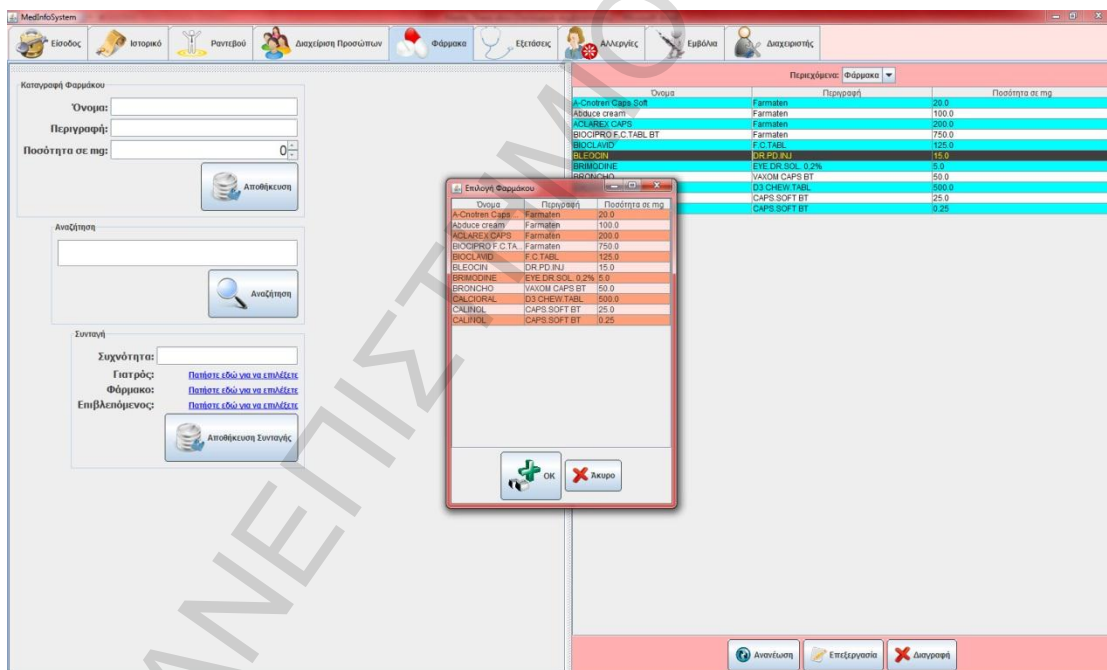
Η κάθε καρτέλα είναι αφιερωμένη σε ένα συγκεκριμένο τομέα, όπως τα φάρμακα, οι εξετάσεις, τα πρόσωπα και οι χρήστες, τα ραντεβού κλπ. Η οθόνη είναι χωρισμένη σε δύο μέρη. Στα αριστερά υπάρχει μια μπάρα εργαλείων, στην οποία ο χρήστης έχει τη δυνατότητα να πραγματοποιήσει διάφορες λειτουργίες, μεταξύ των οποίων η καταγραφή δεδομένων στο Σύστημα και η Αναζήτησή τους. Στα δεξιά υπάρχει ένας πίνακας που εμφανίζει τις καταχωρήσεις στη Βάση Δεδομένων. Στο πάνω μέρος υπάρχει η επιλογή των περιεχομένων τα οποία π.χ. μπορεί να είναι φάρμακα ή συνταγές. Πατώντας το κουμπί 'Ανανέωση' που είναι το μοναδικό κουμπί που υπάρχει αρχικά στο κάτω μέρος του πίνακα έχουμε τη δυνατότητα να τον γεμίσουμε ή να ανανεώσουμε τα δεδομένα του, τα οποία ωστόσο ανανεώνονται αυτόματα με κάθε νέα εγγραφή. Επιλέγοντας μία από τις εγγραφές του πίνακα εμφανίζονται, όπως φαίνεται και στην Εικόνα 4-9, δύο επιπλέον κουμπιά, τα οποία προσφέρουν δυνατότητα επεξεργασίας (Εικόνα 4-10) ή διαγραφής των επιλεγμένων στοιχείων κατόπιν επιβεβαίωσης. Εάν επιθυμούμε να τα επεξεργαστούμε εμφανίζεται μία φόρμα με τις υπάρχουσες τιμές, ικανές να αλλάξουν.

Στο παράδειγμά μας φαίνεται η δυνατότητα καταγραφής μιας συνταγής. Για να γίνει αυτό θα πρέπει να επιλεγεί γιατρός, φάρμακο και επιβλεπόμενος από τους ήδη καταγεγραμμένους στη Βάση Δεδομένων. Για να εξυπηρετηθεί λοιπόν αυτός ο σκοπός, πατώντας στην κατάλληλη επιλογή ανοίγει μία καρτέλα με έναν πίνακα που περιέχει τις εγγραφές από τις οποίες μπορούμε να επιλέξουμε.

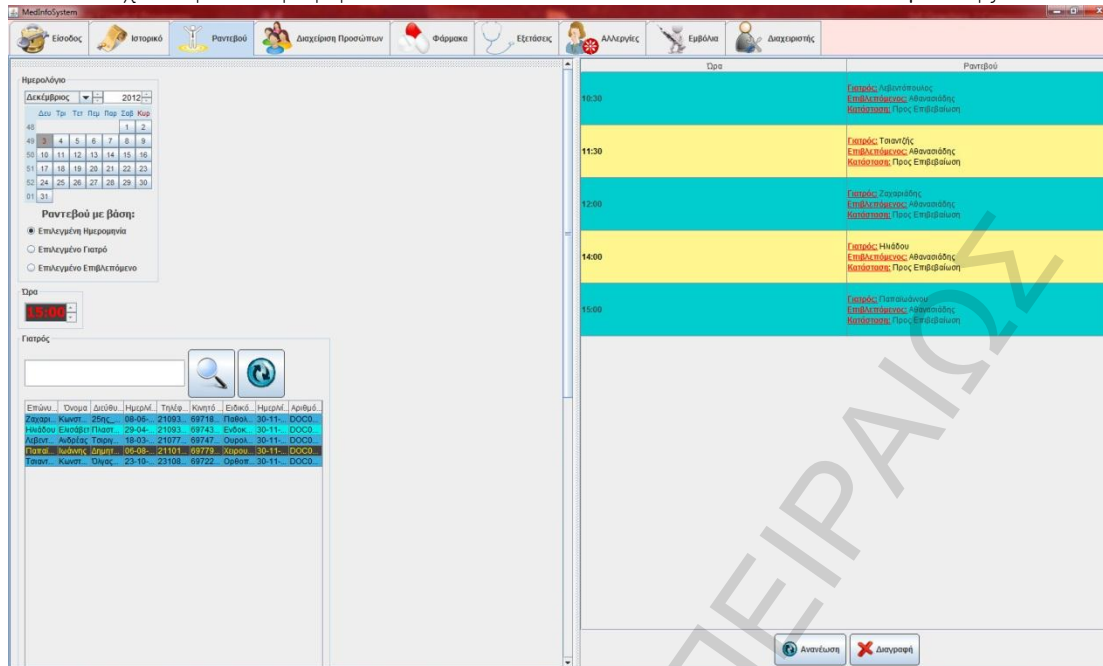
Με παρόμοιο τρόπο λειτουργούν και οι καρτέλες 'Εξετάσεις', 'Αλλεργίες', 'Εμβόλια' αλλά και η 'Διαχείριση Προσώπων'.



**Εικόνα 4-10. Επεξεργασία φαρμάκου**



**Εικόνα 4-11. Πίνακας επιλογής φαρμάκου**



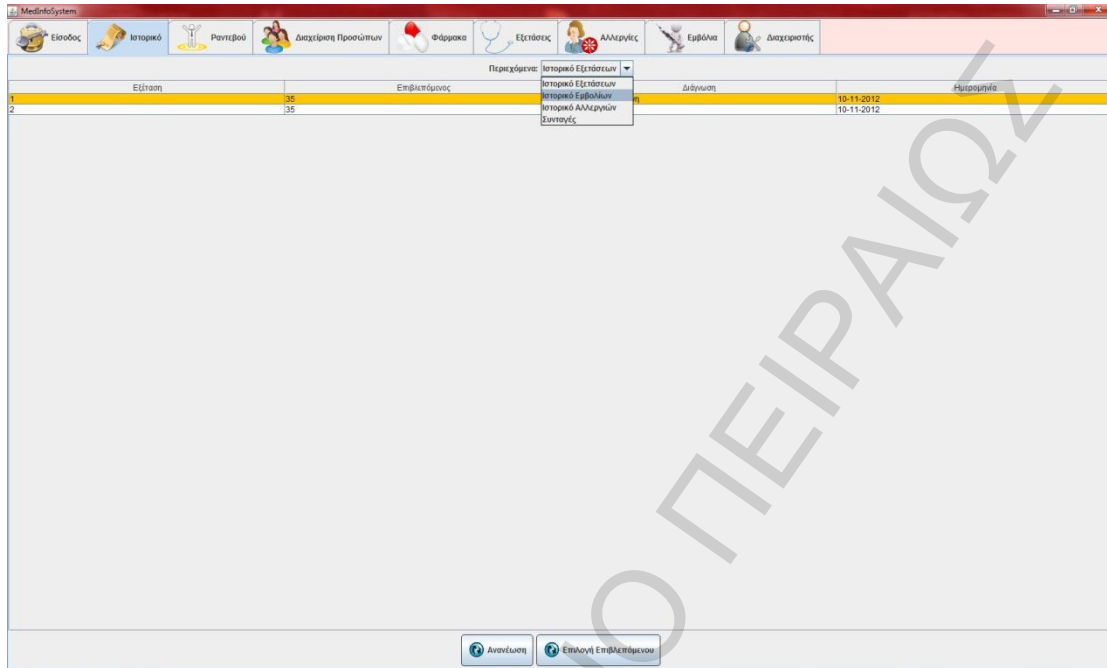
**Εικόνα 4-12. Καρτέλα ραντεβού**

Πολύ σημαντική και καινοτόμα λειτουργία είναι αυτή της διαχείρισης των ραντεβού που προσφέρεται από το Πληροφοριακό Σύστημα. Στο αριστερό μισό της οθόνης μπορούμε να διακρίνουμε (Εικόνα 4-12) ένα ημερολόγιο από το οποίο μπορούμε να επιλέξουμε την ημερομηνία που επιθυμούμε. Στη συνέχεια μπορούμε να διαλέξουμε αν θέλουμε να δούμε όλα τα ραντεβού της συγκεκριμένης ημερομηνίας, τα ραντεβού με συγκεκριμένο γιατρό ή με συγκεκριμένο επιβλεπόμενο. Η επιλογή γιατρού ή επιβλεπόμενου πρέπει να γίνει στους πίνακες που ακολουθούν από κάτω, οι οποίοι προσφέρουν και επιλογές αναζήτησης. Επίσης, υπάρχει και ένα 24-ωρο ρολόι στο οποίο θα επιλέξουμε την ώρα ενός νέου ραντεβού, που θέλουμε να καταχωρήσουμε. Για παράδειγμα, αν θέλουμε να κλείσουμε ένα ραντεβού μεταξύ του γιατρού 'Λεβεντόπουλου' και του επιβλεπόμενου 'Αθανασιάδη' θα επιλέξουμε τα ονόματά τους από τους κατάλληλους πίνακες, θα διαλέξουμε την επιθυμητή ώρα και την ημερομηνία και θα καταγράψουμε το ραντεβού. Αυτό θα εμφανιστεί στο δεξί μέρος της οθόνης, με την ένδειξη 'Προς Επιβεβαίωση'. Εάν εξακριβωθεί ότι μπορεί να πραγματοποιηθεί μπορεί να αλλαχτεί η κατάσταση σε 'Εκκρεμεί' ενώ αν αυτό δεν είναι δυνατό τότε έχουμε τη δυνατότητα να το μετατρέψουμε σε 'Ακυρώθηκε'. Επίσης, μετά την ολοκλήρωσή του, η ένδειξη αλλάζει σε 'ολοκληρώθηκε' κι έτσι έχουμε ένα πλήρες αρχείο με τα ραντεβού και την εξέλιξή τους.

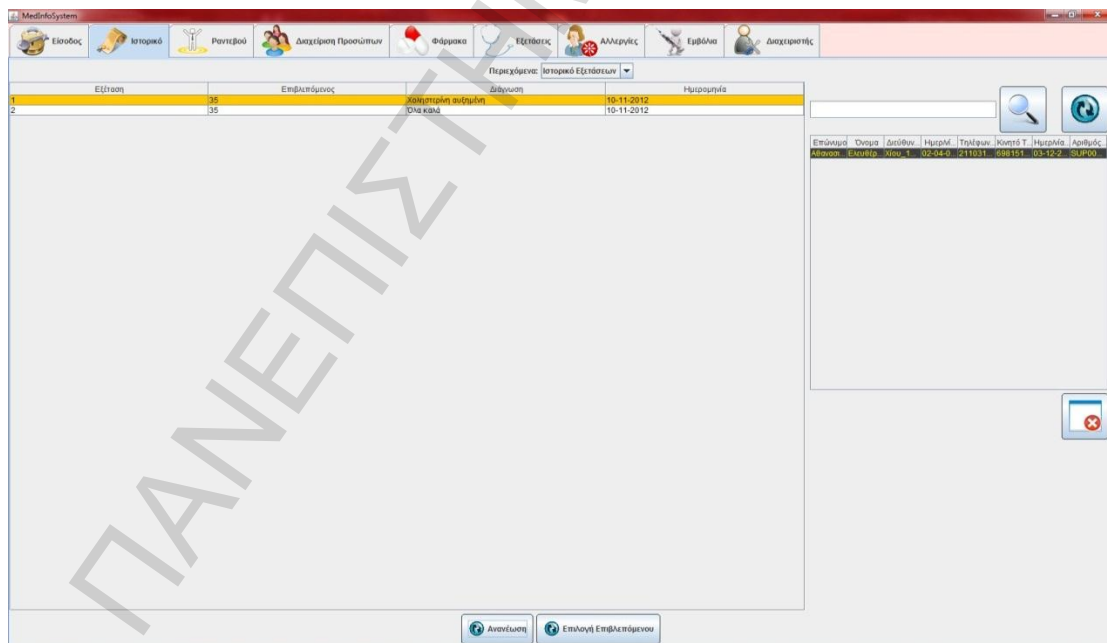
Άλλη μία καρτέλα με ξεχωριστό τρόπο λειτουργίας είναι αυτή του ιστορικού (Εικόνες 4-13, 4-14). Εδώ συγκεντρώνεται η κεντρική ιδέα της χρησιμότητας του Πληροφοριακού Συστήματος. Ο πίνακας στην αρχή πιάνει ολόκληρη την οθόνη και στο πάνω μέρος υπάρχει η κλασική επιλογή περιεχομένων. Κάτω βλέπουμε το κουμπί 'Ανανέωση' και ένα ακόμα κουμπί που μας παρέχει τη δυνατότητα να επιλέξουμε επιβλεπόμενο. Η ουσία της καρτέλας αυτής είναι να αντιλήσουμε πληροφορίες για το ιστορικό συγκεκριμένου προσώπου. Έτσι, πατώντας το κουμπί αυτό εμφανίζεται ένας πίνακας στο δεξί μέρος της οθόνης στον οποίο μπορούμε να επιλέξουμε



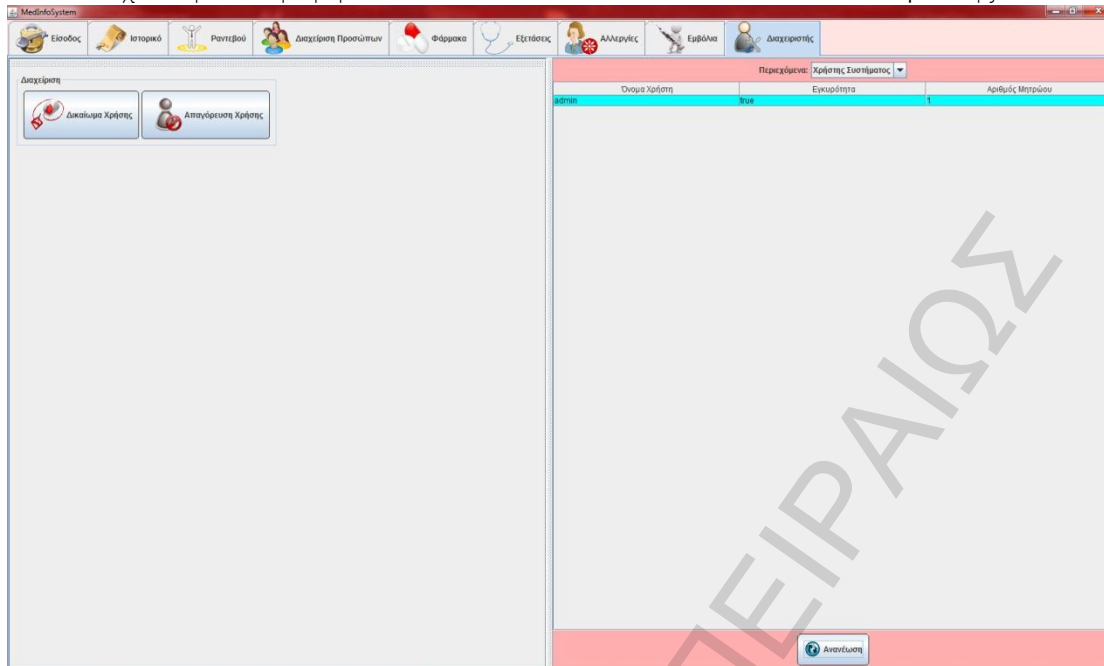
Μεταπτυχιακή Διατριβή Παναγιώτης Κοκκότης  
τον επιβλεπόμενο που μας ενδιαφέρει (Εικόνα 4-14). Η διαδικασία είναι απλή και εύχρηστη. Διαλέγουμε το πρόσωπο που θέλουμε, επιλέγουμε τα επιθυμητά περιεχόμενα και πατάμε 'ανανέωση' για να γεμίσει ο πίνακας με τα ιστορικά στοιχεία του, όπως τα εμβόλια, οι εξετάσεις, οι αλλεργίες και οι συνταγές που βρίσκονται καταχωρημένες στο Σύστημα.



**Εικόνα 4-13. Καρτέλα ιστορικού (α)**



**Εικόνα 4-14. Καρτέλα ιστορικού (β)**



**Εικόνα 4-15. Καρτέλα διαχειριστή**

Η καρτέλα 'Διαχειριστής' εμφανίζεται μόνο όταν ο χρήστης έχει τα κατάλληλα δικαιώματα και σε αυτήν μπορεί κάποιος να εποπτεύει το Σύστημα και να επιτελεί λειτουργίες που δεν μπορούν να πραγματοποιηθούν από έναν τυπικό χρήστη. Μέχρι στιγμής έχει τη δυνατότητα να παραχωρεί το δικαίωμα χρήσης του Συστήματος ή να το αφαιρεί κατά περίπτωση. Αυτό είναι σημαντικό γιατί οποιοσδήποτε μπορεί να αποκτήσει όνομα χρήστη και κωδικό πρόσβασης, μέσω του Συστήματος, αλλά για να καταφέρει να το χρησιμοποιήσει θα πρέπει να έχει ελεγχθεί και πάρει τη σχετική άδεια από τον διαχειριστή. Μην ξεχνάμε ότι το Σύστημα είναι σχεδιασμένο ώστε να προβλέπονται μελλοντικές προεκτάσεις του και η καρτέλα αυτή προσφέρει τη δυνατότητα προσθήκης επιπλέον δραστηριοτήτων.

### **Οι 5 βασικές λειτουργίες**

Όπως είδαμε στο προηγούμενο κεφάλαιο (Ανάλυση - Αρχιτεκτονική του Συστήματος) το Σύστημα, πέρα από τη γενικότερη πολυπλοκότητά του, επιτελεί 5 βασικές λειτουργίες σε όλες τις καρτέλες, με σκοπό να επιτύχει τους σκοπούς της δημιουργίας του. Αυτές οι λειτουργίες είναι η Καταγραφή, η Επισκόπηση, η Αναζήτηση, η Επεξεργασία και η Διαγραφή των Δεδομένων. Όλες στοχεύουν στην χρησιμοποίηση ή τροποποίηση στοιχείων της Βάσης Δεδομένων και αποτελούν μέρος των κύριων στόχων των Ιατρικών Πληροφοριακών Συστημάτων. Παρακάτω συνοψίζονται αυτές οι λειτουργίες και ο τρόπος υλοποίησης, σε σχέση με τον αρχικό σχεδιασμό, που έχει περιγραφεί και με τα σχετικά διαγράμματα UML.

**Ι. Καταγραφή**

Καταγραφή Προσώπου

Όνομα  Τύπος χρήστη

Επίσημο  Ημερομηνία γέννησης

Οδός  Αρ

Πόλη  ΤΚ

Τηλ. Οικίας

Κινητό τηλ.

Ειδικότητα

Αποθήκευση

Πατήστε για αποθήκευση των Δεδομένων

Αναζήτηση

Καταγραφή χρήστη του συστήματος

Όνομα χρήστη

Κωδικός πρόσβασης

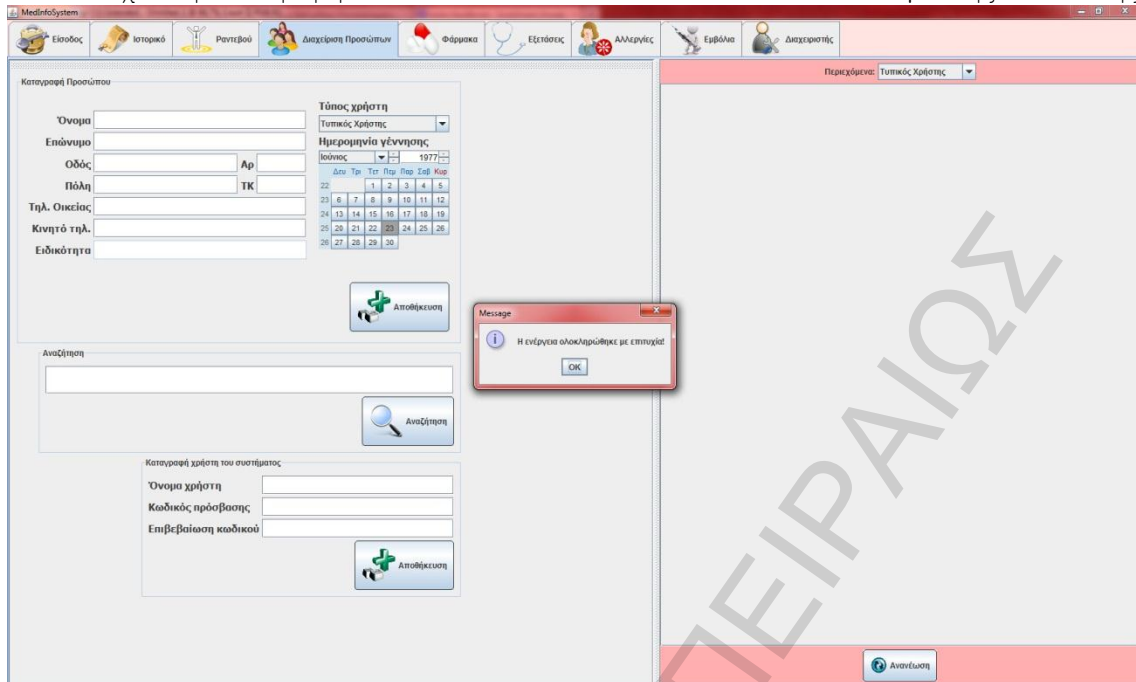
Επιβεβαίωση κωδικού

Αποθήκευση

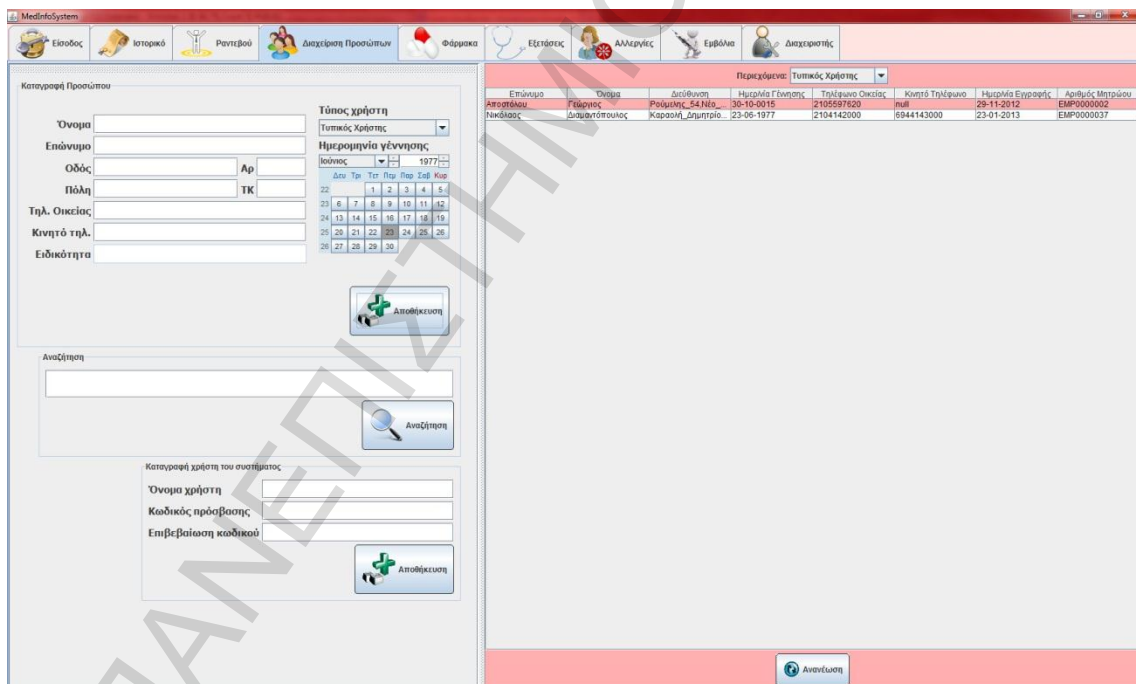
Ανανέωση

**Εικόνα 4-16. Καρτέλα διαχείρισης προσώπων – Παράδειγμα Καταγραφής**

Θα χρησιμοποιήσουμε το παράδειγμα της καταγραφής ενός τυπικού χρήστη στο Σύστημα. Όπως φαίνεται και στην Εικόνα 4-16 πρέπει να συμπληρωθούν τα πεδία με τα στοιχεία του νέου προσώπου, που πρόκειται να καταγραφεί. Στη δεύτερη στήλη υπάρχει ένα combobox, στο οποίο επιλέγεται ο τύπος του χρήστη. Εάν είναι γιατρός τότε θα ενεργοποιηθεί και το πεδίο της 'ειδικότητας', το οποίο πρέπει να συμπληρωθεί. Όταν ολοκληρωθεί η συμπλήρωση πρέπει να πατηθεί το κουμπί 'Αποθήκευση' και στη συνέχεια το Σύστημα αναλαμβάνει να ολοκληρώσει την διαδικασία, μέσα από μια σειρά λειτουργιών που δεν είναι ορατές στον χρήστη. Πρώτα από όλα γίνεται η καταγραφή στο μητρώο και αυτόματα δίνεται ένας αύξων αριθμός εγγραφής. Στη συνέχεια επιλέγεται ο πίνακας στον οποίο ανήκει, ανάλογα με τον τύπο και εκεί αποθηκεύονται τα δεδομένα. Εάν όλα πάνε κατ' ευχή τότε εμφανίζεται το μήνυμα επιβεβαίωσης της Εικόνας 4-17 και γίνεται ανανέωση του πίνακα (Εικόνα 4-18) με τον χρήστη να εμφανίζεται πλέον στα περιεχόμενά του. Σε διαφορετική περίπτωση εμφανίζεται ένα μήνυμα που ενημερώνει ότι κάτι δεν πήγε καλά στη διαδικασία. Πρέπει να επισημανθεί ότι υπάρχει ένα κελί με τον αριθμό μητρώου, ο οποίος έχει προκύψει από αυτόματες διαδικασίες και είναι συνδυασμός του τύπου χρήστη και της σειράς με την οποία καταγράφηκε στο μητρώο.

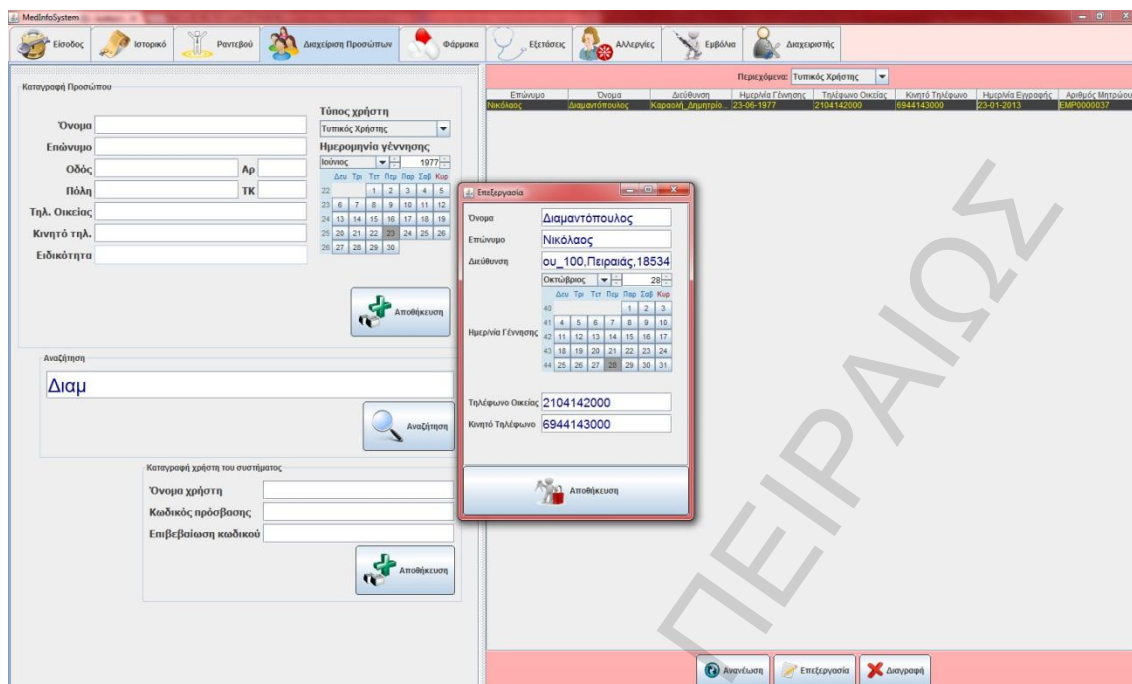


Εικόνα 4-17. Μήνυμα Επιβεβαίωσης Καταγραφής



Εικόνα 4-18. Ανανέωση πίνακα

## ii. Επεξεργασία



**Εικόνα 4-19. Επεξεργασία προσώπου**

Όπως είναι λογικό, σε πολλές περιπτώσεις μπορεί να έχει γίνει κάποιος λάθος κατά την καταγραφή των δεδομένων ή ακόμα και να έχουν αλλάξει κάποια στοιχεία, όπως η διεύθυνση ή το τηλέφωνο. Για τον λόγο αυτό το Σύστημα προσφέρει τη δυνατότητα επεξεργασίας, όπως φαίνεται και στην Εικόνα 4-19. Επιλέγοντας μία γραμμή του πίνακα εμφανίζονται δύο επιπλέον κουμπιά στο κάτω μέρος της οθόνης. Πατώντας το κουμπί 'Επεξεργασία' (με την διαγραφή θα ασχοληθούμε αργότερα) εμφανίζεται η φόρμα της παραπάνω εικόνας στην οποία είναι συμπληρωμένα όλα τα πεδία με τα στοιχεία της εγγραφής που θέλουμε να τροποποιήσουμε. Πραγματοποιώντας τις επιθυμητές αλλαγές τις κατοχυρώνουμε αποστέλλοντας το σχετικό αίτημα, μετά το πάτημα το κουμπιού 'Αποθήκευση'.

## iii. Επισκόπηση

Η διαδικασία της Επισκόπησης είναι ορατή στην Εικόνα 4-9 αλλά και στην Εικόνα 4-18. Ο χρήστης πρέπει να επιλέξει, από το πλαίσιο στο πάνω μέρος του πίνακα, τα περιεχόμενα που θέλει να δει και στη συνέχεια να πατήσει το κουμπί 'Ανανέωση' στο κάτω μέρος της οθόνης. Το Σύστημα αναλαμβάνει να στείλει το αίτημα στον διακομιστή και αυτός, με τη διαδικασία που έχει ήδη περιγραφεί, επιστρέφει τα σχετικά δεδομένα που εμφανίζονται στον πίνακα.

**iv. Αναζήτηση**

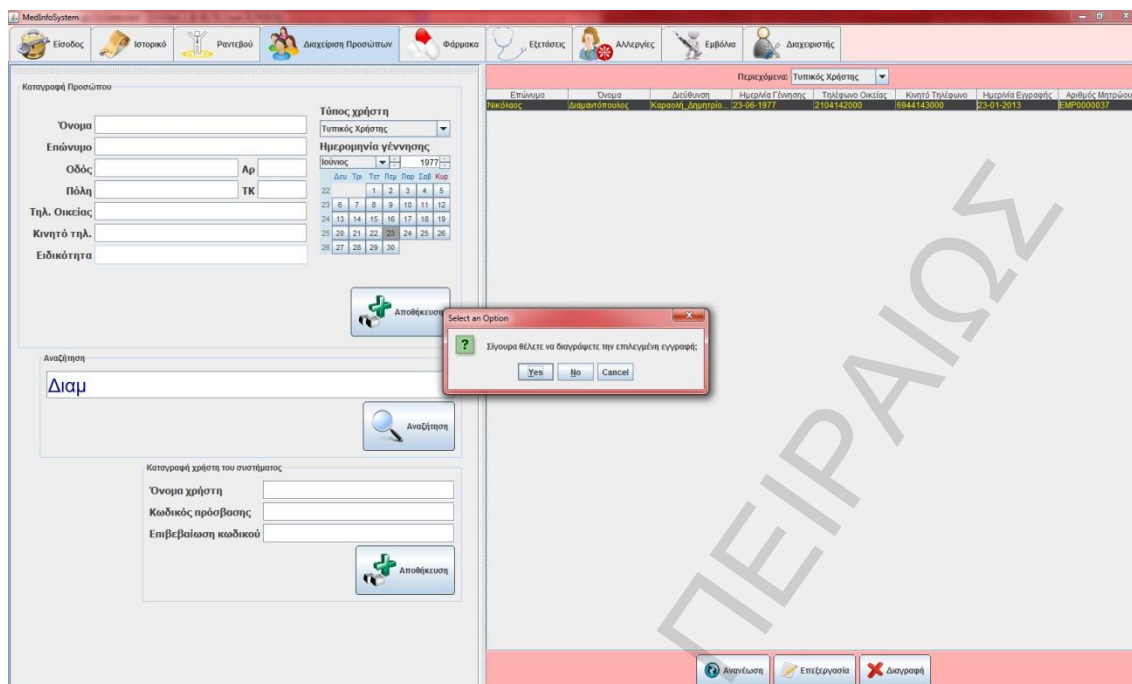
The screenshot shows the MedInfoSystem interface. On the left, there are input fields for patient details (Name, Surname, Address, City, Phone, Mobile, Speciality) and a search bar containing 'Διαμ'. Below the search bar are fields for 'Κατηγορία χρήστη του συστήματος' (User category, Code, Confirmation code). On the right, a table displays search results. The table has columns: Επώνυμο, Όνομα, Διεύθυνση, Ημερομηνία Γέννησης, Τηλέφωνο Οικίας, Κινητό Τηλέφωνο, Ημερομηνία Εγγραφής, and Αριθμός Μητρώου. The first row contains the following data: Κακός, Διαμοιστούλας, Καραούλη Δημήτριά, 23-06-1977, 12104142000, 8944143000, 23-01-2013, and EMP-0000037.

Επώνυμο	Όνομα	Διεύθυνση	Ημερομηνία Γέννησης	Τηλέφωνο Οικίας	Κινητό Τηλέφωνο	Ημερομηνία Εγγραφής	Αριθμός Μητρώου
Κακός	Διαμοιστούλας	Καραούλη Δημήτριά	23-06-1977	12104142000	8944143000	23-01-2013	EMP-0000037

**Εικόνα 4-20. Παράδειγμα Αναζήτησης**

Σε όλες τις καρτέλες του Συστήματος προσφέρεται και η πολύ σημαντική λειτουργία της αναζήτησης. Ο χρήστης μπορεί να πληκτρολογήσει οτιδήποτε θέλει να ψάξει (ακόμα και ένα γράμμα) και το Σύστημα θα του επιστρέψει στον πίνακα τις εγγραφές που, σε οποιοδήποτε κελί τους, περιέχουν τη λέξη κλειδί. Στο παράδειγμα που χρησιμοποιήθηκε πριν ο χρήστης έχει πληκτρολογήσει τα πρώτα γράμματα του επιθέτου, του πρόσφατα καταγεγραμμένου και το αποτέλεσμα φαίνεται στην Εικόνα 4-20. Αυτή η λειτουργία είναι εξαιρετικής σημασίας και για την καταγραφή, κυρίως συνταγών ή ιστορικού, όπου χρειάζεται να επιλεγούν καταχωρημένες στη ΒΔ εγγραφές και θα ήταν εντελώς αντιλειτουργικό αν έπρεπε να τις ψάξει μία μία, ειδικά όταν αυτές είναι χιλιάδες.

## v. Διαγραφή



**Εικόνα 4-21. Επιβεβαίωση Διαγραφής**

Πατώντας το άλλο κουμπί που εμφανίστηκε στο κάτω μέρος της οθόνης (αυτό της 'Διαγραφής') εμφανίζεται ένα μήνυμα, στο οποίο πρέπει να επιβεβαιώσουμε ή να απορρίψουμε την διαγραφή των δεδομένων. Οι επιλεγμένες εγγραφές διαγράφονται για πάντα από τη Βάση Δεδομένων, γι' αυτό πρέπει να είμαστε πολύ προσεκτικοί. Μια πολύ σημαντική ενέργεια για την αποφυγή δυσάρεστων καταστάσεων θα ήταν η δημιουργία backup της ΒΔ σε τακτά χρονικά διαστήματα.

## Mobile Εφαρμογή

Το **Android** είναι λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας το οποίο τρέχει τον πυρήνα του λειτουργικού Linux. Αρχικά αναπτύχθηκε από την Google και αργότερα από την Open Handset Alliance. Επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με την χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας την συσκευή μέσω βιβλιοθηκών λογισμικού ανεπτυγμένων από την Google.

Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού Open Handset Alliance, μιας κοινοπραξίας 48 τηλεπικοινωνιακών εταιριών, εταιριών λογισμικού καθώς και κατασκευής hardware, οι οποίες είναι αφιερωμένες στην ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις συσκευές κινητής τηλεφωνίας. Η Google δημοσίευσε το μεγαλύτερο μέρος του κώδικα του Android υπό τους όρους της Apache License, μιας ελεύθερης άδειας



#### **Εικόνα 4-22. Εικονίδιο Εφαρμογής**

Από τότε που η αγορά κατακλύστηκε από smartphones και ο χρήστης έχει τη δυνατότητα να τρέχει πολύπλοκες εφαρμογές στο κινητό του τηλέφωνο είναι φυσικό ο συγκεκριμένος τομέας να γνωρίζει μεγάλη ανάπτυξη. Βασικότερος λόγος είναι η ύπαρξη τέτοιων συσκευών στα χέρια της πλειοψηφίας των καταναλωτών, για τους οποίους ένα κινητό τηλέφωνο είναι παραπάνω από απαραίτητο στην καθημερινότητά τους. Επίσης, υπάρχουν και τα tablets που αρχίζουν να εισβάλουν με τη σειρά τους στα σύγχρονα σπίτια.

Ένα άλλο σημαντικό εμπόδιο που δείχνει να ξεπερνιέται και με τη σειρά του συμβάλει στην εξάπλωση αυτών των μικροπρογραμμάτων είναι η διαφοροποίηση των λειτουργικών συστημάτων. Στο πρόσφατο παρελθόν σχεδόν κάθε κατασκευαστής κινητών τηλεφώνων ανέπτυξε και χρησιμοποίησε το δικό του λειτουργικό, που σε αρκετές περιπτώσεις διέφερε και μεταξύ συσκευών της ίδιας εταιρείας. Αυτή η μεγάλη διασπορά καθιστούσε ιδιαίτερα περίπλοκη τη δημιουργία εφαρμογών, που αναπόφευκτα θα απευθύνονταν σε λίγες συσκευές και ακόμα μικρότερη μερίδα καταναλωτών.





**Εικόνα 4-23. Λειτουργικό Android**

Τα τελευταία χρόνια και ενώ τα λειτουργικά συστήματα σχεδιασμένα για κινητά τηλέφωνα συνεχίζουν να είναι πολλά οι εταιρείες προσανατολίζονται όλο και περισσότερο στην υιοθέτηση συγκεκριμένου συστήματος. Αυτό έχει σαν αποτέλεσμα να έχουν επικρατήσει τρία από αυτά, το Android, το ios και τα Windows. Έτσι, οι developers ανά τον κόσμο εκμεταλλεύτηκαν την ευκαιρία και ξεκίνησαν μία τεράστια παραγωγή εφαρμογών όλων των ειδών και για την κάλυψη όλων των πιθανών και απίθανων καθημερινών μας αναγκών. Επίσης, γίνεται προσπάθεια να ξεπεραστούν και τα τελευταία εμπόδια ώστε οι εφαρμογές στο μέλλον να γίνουν εύκολα μεταφέρσιμες σε όλων των ειδών τα λειτουργικά. Ένα καλό παράδειγμα αυτής της προσπάθειας είναι το phonedgar.

Τα Πληροφοριακά Συστήματα και οι εταιρείες παραγωγής τους, πιάνοντας τον σφυγμό της εποχής, έχουν αρχίσει να επεκτείνονται μέσω αυτών των εφαρμογών, οι οποίες μπορούν να φανούν χρήσιμες για την εποπτεία και τη διαχείριση των Συστημάτων αυτών αλλά κυρίως για την επικοινωνία και την εξυπηρέτηση των πελατών. Αυτές οι εφαρμογές

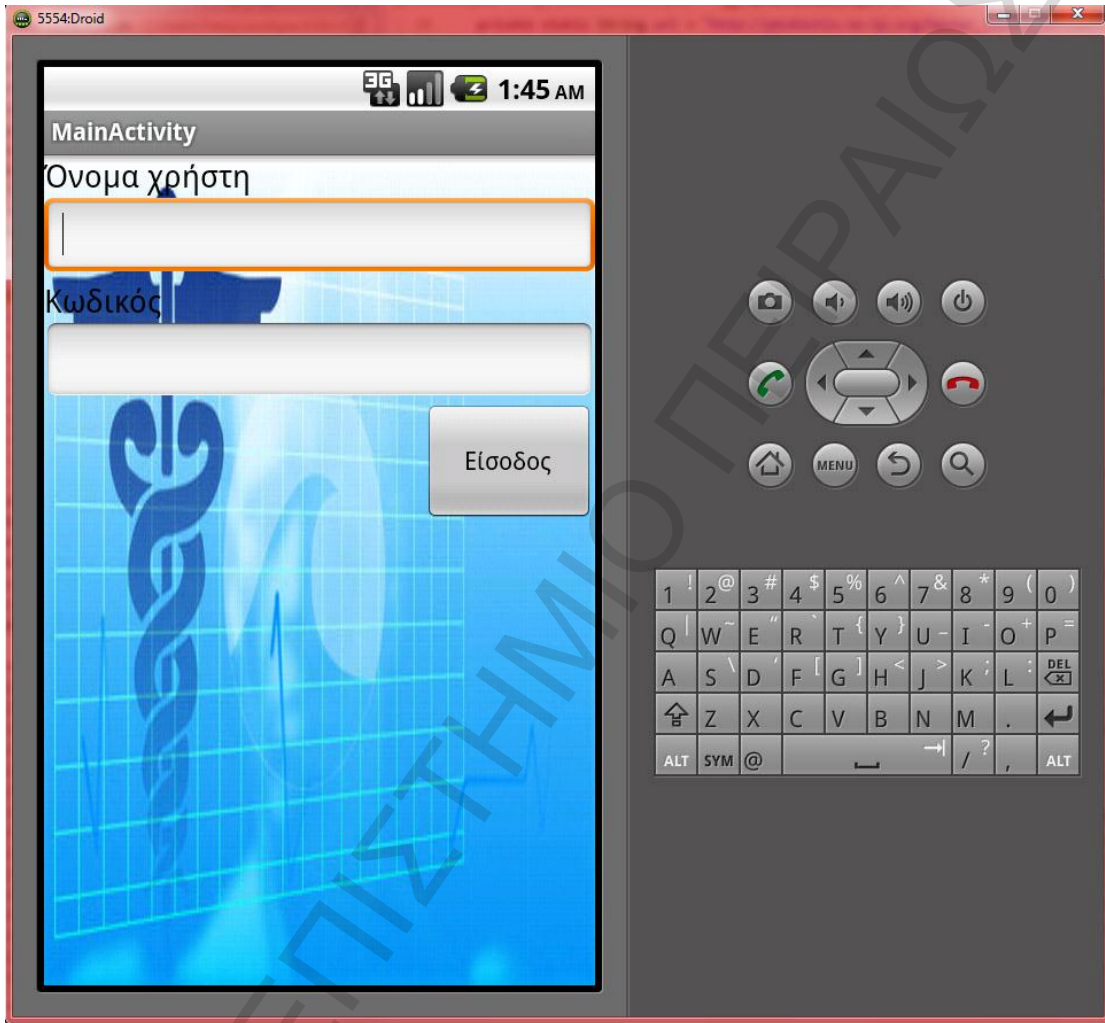
Μεταπτυχιακή Διατριβή Παναγιώτης Κοκκότης  
προσφέρουν αμεσότητα, επιτελούν σημαντικές λειτουργίες που στο παρελθόν έμοιαζαν πολύπλοκες και διευκολύνουν την επικοινωνία μεταξύ επιχείρησης και πελάτη.



**Εικόνα 4-24. Android Smartphone**

Το παρόν Πληροφοριακό Σύστημα φιλοδοξεί να προσαρμοστεί επιτυχημένα στις σύγχρονες τεχνολογικές εξελίξεις. Επομένως, επεκτείνεται μέσω της χρήσης μιας τέτοιου είδους εφαρμογής που μπορεί να παίξει καταλυτικό ρόλο, ακόμα και στη γενικότερη αντίληψη του επιβλεπόμενου για την ιατρική μονάδα με την οποία συναλλάσσεται. Πλέον, μπορεί να έχει άμεση πρόσβαση στα προσωπικά του ιατρικά δεδομένα, να ζητήσει ραντεβού, να μάθει τα στοιχεία επικοινωνίας των γιατρών, να μάθει τα αποτελέσματα των εξετάσεών του και να διατηρεί συγκεντρωμένο το ιστορικό του πιο εύκολα, ανέξοδα και ακούραστα, περισσότερο από κάθε άλλη φορά. Από την πλευρά της ιατρικής μονάδας τα οφέλη είναι εξίσου μεγάλα και σημαντικά. Η διατήρηση της επαφής με

Μεταπτυχιακή Διατριβή Παναγιώτης Κοκκότης  
τους επιβλεπόμενους καθίσταται ευκολότερη. Η εφαρμογή είναι εγκατεστημένη σε μια συσκευή που αποτελεί μέρος της καθημερινότητας του σύγχρονου ανθρώπου. Με αυτόν τον τρόπο η επαφή του με την ιατρική μονάδα γίνεται πιο συχνή. Η διευκόλυνσή του σε τομείς που αλλού χρειάζεται να κοπιάσει για να εξυπηρετηθεί ανεβάζει την εκτίμησή του για τη μονάδα. Επιπλέον, η ενημέρωσή του για προγραμματισμένες εξετάσεις που θα έπρεπε να πραγματοποιηθούν μειώνει τις πιθανότητες να τις αμελήσει και αποφέρει οικονομικά οφέλη στην επιχείρηση.



**Εικόνα 4-25. Είσοδος Εφαρμογής**

Έχοντας εγκαταστήσει στη συσκευή του την παραπάνω εφαρμογή ο χρήστης μπορεί να εισέλθει πατώντας το εικονίδιο της Εικόνας 4-22. Στην πρώτη οθόνη (Εικόνα 4-25) θα του ζητηθεί να πληκτρολογήσει το όνομα χρήστη και τον κωδικό πρόσβασης. Για να αποκτήσει στοιχεία εισόδου κάποιος θα πρέπει να είναι ήδη εγγεγραμμένος στο μητρώο της ιατρικής μονάδας και αφού τα αποκτήσει να έχει εξουσιοδοτηθεί από τον διαχειριστή. Στη συνέχεια εξακριβώνεται η εγκυρότητα των στοιχείων και εισέρχεται στην κεντρική οθόνη (Εικόνα 4-26).



**Εικόνα 4-26. Κεντρική καρτέλα εφαρμογής**

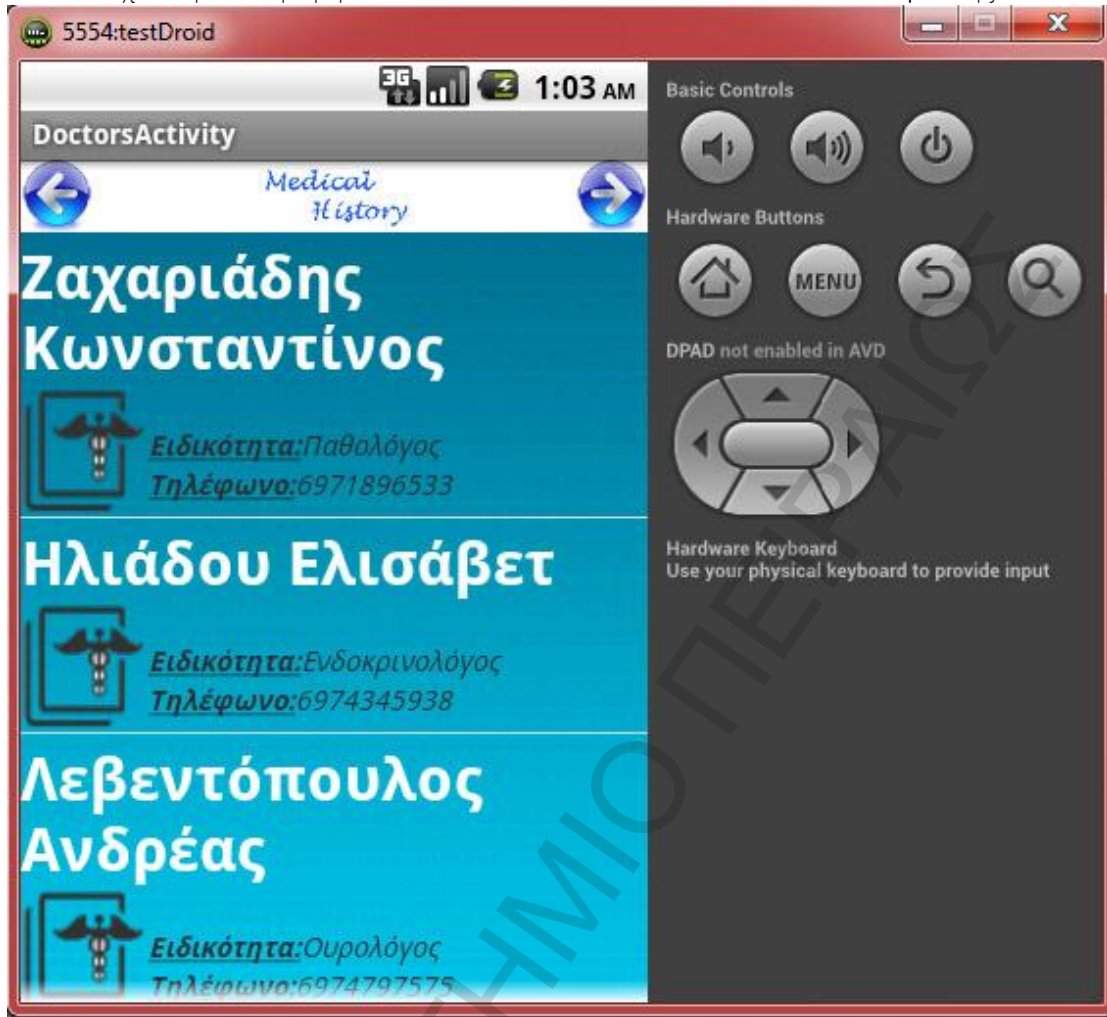
Οι επιλογές που παρέχονται στο κεντρικό μενού αφορούν τα προσωπικά στοιχεία του χρήστη και είναι οι εξής:

- **Συνταγές:** Εμφάνιση της λίστας με τις συνταγές που τον αφορούν, τα φάρμακα, τις δοσολογίες, τις ποσότητες.
- **Αλλεργίες:** Όσες αλλεργίες τον έχουν ταλαιπωρήσει.
- **Εμβολιασμοί:** Τα εμβόλια που έχει ήδη κάνει αλλά και τα προγραμματισμένα για το μέλλον.
- **Εξετάσεις:** Το ιστορικό των εξετάσεων και οι διαγνώσεις τους. Πολύ σημαντικό, σε περίπτωση που βρίσκεται σε αναμονή αποτελεσμάτων και προτιμά να γλιτώσει χρόνο και κόπο από το να επισκεφτεί τη μονάδα.
- **Εκκρεμότητες:** Η επιλογή αυτή αφορά τις εξετάσεις που είναι προγραμματισμένες για το μέλλον.
- **Γιατροί:** Από εδώ μπορεί να έχει πρόσβαση στα στοιχεία των γιατρών, όπως π.χ. τα τηλέφωνα και η ειδικότητά τους.
- **Ο λογαριασμός μου:** Μια επιλογή για να διαχειρίζεται τα στοιχεία



**Εικόνα 4-27. Λίστα εξετάσεων**

Πατώντας σε οποιαδήποτε επιλογή ο χρήστης εισέρχεται σε μία λίστα με τα προσωπικά του ιστορικά στοιχεία, παρόμοια με αυτή της Εικόνας 4-27. Δεν έχει δυνατότητα επεξεργασίας γιατί αυτό θα αναιρούσε τις αρχές του παρόντος Συστήματος, κυρίως όσον αφορά την ασφάλεια. Στην επόμενη Εικόνα (4-28) φαίνεται η λίστα με τους γιατρούς. Στο πάνω μέρος της οθόνης υπάρχουν τα κουμπιά πλοήγησης και το Logo που τον επαναφέρει στην κεντρική οθόνη. Επίσης, πρέπει να επισημανθεί ότι οι δοκιμές της εφαρμογής, που φαίνονται στις Εικόνες, έχουν γίνει σε λειτουργικό σύστημα Android 2.2 (Froyo) και 4.2 (JellyBean). Επομένως είναι ικανή να εγκατασταθεί στη μεγάλη πλειοψηφία (σίγουρα πάνω από το 95%) αν όχι σε όλα τα smartphones με αυτό το λειτουργικό.



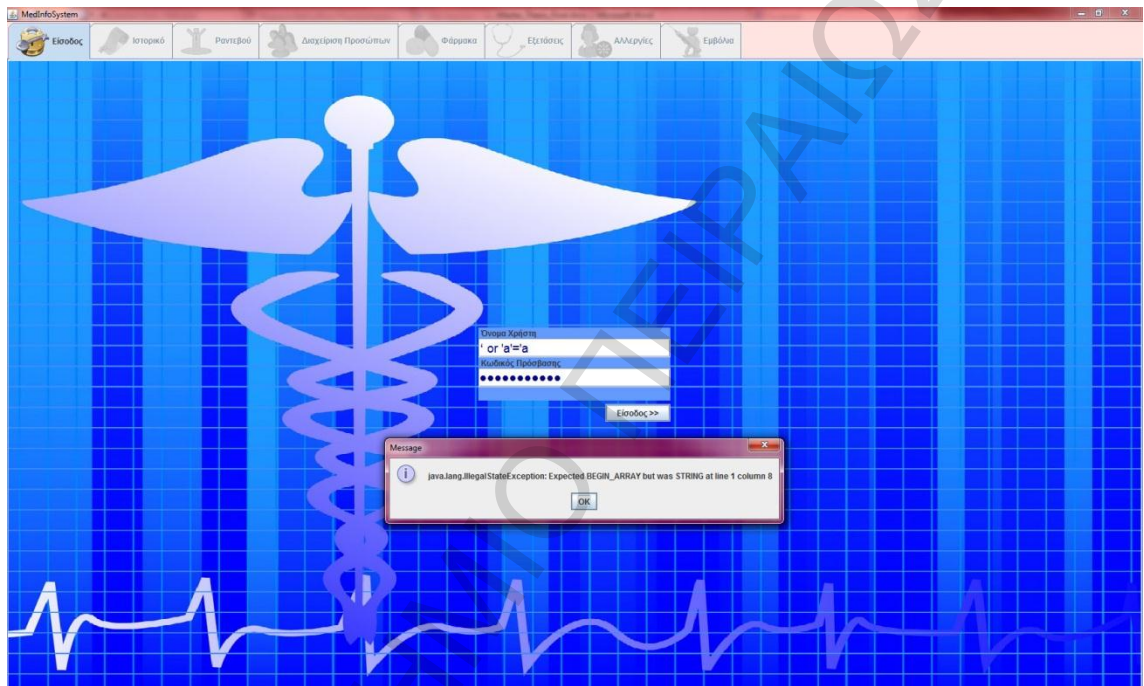
Εικόνα 4-28. Λίστα γιατρών

## SQL Injection

Ένας πολύ συνηθισμένος τρόπος για να εκτεθεί η ασφάλεια ενός Πληροφοριακού Συστήματος ονομάζεται sql injection και είναι η προσπάθεια απόκτησης πρόσβασης στη Βάση Δεδομένων, παρακάμπτοντας τα φίλτρα που χρησιμοποιούνται από την εφαρμογή. Αυτό το είδος επίθεσης είναι πολύ διαδεδομένο μέσω του διαδικτύου και αυτός είναι ένας από τους κυριότερους λόγους που επιλέχθηκε η υλοποίηση μιας standalone εφαρμογής, σε αντίθεση με τη δημιουργία μιας αντίστοιχης web.

Παρόλα αυτά, το λογισμικό μπορεί να πέσει στα χέρια οποιουδήποτε και κυρίως η mobile εφαρμογή με αποτέλεσμα να είναι ευάλωτη σε προσπάθεια επιθέσεων. Έτσι, οι δικλείδες ασφαλείας που χρησιμοποιούνται είναι αρκετές και το Σύστημα είναι χωρισμένο σε επίπεδα. Τα δεδομένα που αποστέλλονται φιλτράρονται πολλές φορές μέχρι να φτάσουν στην επικοινωνία με τη ΒΔ και κυρίως στην πλευρά του

Η Java προσφέρει τα Prepared Statements, τα οποία φιλτράρουν τα δεδομένα και ουσιαστικά απαγορεύουν την εισαγωγή ανεπιθύμητων χαρακτήρων σε μία εντολή sql. Τα δεδομένα βέβαια έχουν ήδη φιλτραριστεί κατά την εισαγωγή τους, την αποστολή, τη λήψη και τον έλεγχό τους. Η δοκιμή της Εικόνας 4-29 αποδεικνύει του λόγου το αληθές.



**Εικόνα 4-29. Παράδειγμα απόκρουσης sql injection στο λογισμικό**

Ουσιαστικά ο hacker του παραπάνω παραδείγματος προσδοκά να γίνει μέρος του query ο παραπάνω κώδικας (' or 'a'='a), ο οποίος ουσιαστικά λέει στο πρόγραμμα ότι τα στοιχεία του είναι σωστά έτσι κι αλλιώς, αφού το a=a ισχύει. Έτσι, επιθυμεί να αποκτήσει πρόσβαση χωρίς να έχει όνομα και κωδικό χρήστη αλλά αποτυγχάνει. Το μήνυμα που εμφανίζεται είναι πολύ γενικό (άρα δεν δίνει πληροφορίες για το Σύστημα) και ενημερώνει ότι η απάντηση του server ήταν μη αναμενόμενη.

Όσον αφορά την εφαρμογή, έχει επιλεγεί να μην υπάρχει δυνατότητα τροποποίησης των δεδομένων, εξαιτίας της μεγάλης έκθεσής της στο κοινό. Όμως από το παράδειγμα της Εικόνας 4-30 φαίνεται ότι ούτε αυτή είναι ευάλωτη σε επιθέσεις της μορφής που αναλύσαμε παραπάνω.



**Εικόνα 4-30. Παράδειγμα απόκρουσης sql injection στην εφαρμογή**

### **Αξιολόγηση του Συστήματος**

Στο σημείο αυτό θα προσπαθήσουμε να αξιολογήσουμε το Ιατρικό Πληροφοριακό Σύστημα με βάση κάποιες συγκεκριμένες ερωτήσεις που εστιάζουν στη δομή, τη σχεδίαση και την αλληλεπίδραση του Συστήματος με τον χρήστη. Θα χρησιμοποιηθεί η κλίμακα Likert (*πάρα πολύ*=5, 4, 3, 2, 1=*καθόλου*).

Τα κείμενα είναι ευανάγνωστα και γραμμένα σε γλώσσα απλή και κατανοητή;

Βαθμός: 5

Σχόλιο: Ό,τι λέξη και κείμενο υπάρχει είναι ευανάγνωστο γραμμένο σε



**Περιορίζεται στο ελάχιστο η απαίτηση για απομνημόνευση πληροφοριών;**

Βαθμός: 5

Σχόλιο: Δεν απαιτείται η απομνημόνευση πληροφοριών για την χρήση του παρά μόνο η εκμάθηση της χρήσης του.

**Η δομή της διεπαφής χρήστη του Συστήματος είναι τέτοια ώστε κάθε τμήμα του να υλοποιεί μια συγκεκριμένη ομάδα ενεργειών;**

Βαθμός: 5

Σχόλιο: Σχεδόν σε όλες τις περιπτώσεις κάθε τμήμα υλοποιεί μια συγκεκριμένη ομάδα ενεργειών και είναι αρκετά καλά οργανωμένο όσον αφορά την πληροφορία που προσφέρεται σχετικά με την κατανόηση των ενεργειών αυτών. Το πρόγραμμα είναι χωρισμένο σε καρτέλες που, από την αρχή της σχεδίασης, αποσκοπούσαν στην καλύτερη οργάνωση των διαφορετικών λειτουργιών που θα επιτελούσε.

**Η ποσότητα και η πυκνότητα της πληροφορίας στην οθόνη είναι λειτουργικές;**

Βαθμός: 4

Σχόλιο: Η ποσότητα και η πυκνότητα της πληροφορίας στην οθόνη είναι αρκετά λειτουργικές αποφεύγοντας την άσκοπη παροχή πληροφορίας στον χρήστη. Επίσης περνώντας τον δείκτη του ποντικιού πάνω από τα κουμπιά παρέχονται επιπλέον πληροφορίες για τη λειτουργία τους.

**Η χρήση των εικονιδίων, των κουμπιών ενεργειών (buttons), και των συνόλων επιλογών (menu), είναι προφανής;**

Βαθμός: 3

Σχόλιο: Οι εικόνες των κουμπιών ενεργειών είναι προφανείς αλλά δεν παρέχεται menu επιλογών. Θα μπορούσε να συμπεριληφθεί σε μια μελλο 4.

**Είναι εύκολη η μετάβαση μπρος πίσω;**

Βαθμός: 4

Σχόλιο: Η σχεδίαση του λογισμικού δεν περιλαμβάνει επιλογές πλοήγησης καθώς η πλοήγηση γίνεται μεταξύ των καρτελών, με μεγάλη ευκολία και αποτελεσματικότητα.

**Υπάρχει παντού η επιλογή της επιστροφής στο κεντρικό σύνολο επιλογών;**

Βαθμός: 5

Σχόλιο: Πολύ λειτουργικό μπορεί να χαρακτηριστεί το Σύστημα, ως προς αυτό το σημείο, με δεδομένο ότι μπορείς να πας οπουδήποτε από οποιοδήποτε σημείο.

**Υπάρχει δυνατότητα εξόδου από το πρόγραμμα από οποιοδήποτε σημείο;**

Βαθμός: 5

Σχόλιο: Από οποιοδήποτε σημείο στο πρόγραμμα υπάρχει η δυνατότητα εξόδου από αυτό. Επίσης υπάρχει και η δυνατότητα του logout, αρκεί να επισκεφτούμε την καρτέλα εισόδου.

**Το λογισμικό ενημερώνει το χρήστη για τις συνέπειες διαφόρων ενεργειών και επιλογών που μπορεί να οδηγήσουν σε δυσλειτουργία της εφαρμογής;**

Βαθμός: 4

Σχόλιο: Το λογισμικό ενημερώνει το χρήστη για τις συνέπειες διαφόρων ενεργειών του, με κατάλληλα μηνύματα. Έχει γίνει μεγάλη προσπάθεια ώστε τα μηνύματα αυτά να μην παρέχουν πληροφορίες για την εσωτερική λειτουργία του Συστήματος, αλλά επειδή πρόκειται για επικοινωνία με διακομιστή και πληθώρα διαφορετικών ενεργειών, τα μηνύματα σφάλματος που μπορεί να προκύψουν είναι πάρα πολλά, σε βαθμό να μην μπορεί να ελεγχθεί με απόλυτη βεβαιότητα ο συγκεκριμένος τομέας.

**Το λογισμικό επιτρέπει αναιρέσεις ενεργειών ή επιλογών του χρήστη;**

Βαθμός: 4

Σχόλιο: Κανένα Σύστημα που επικοινωνεί με ΒΔ δεν παρέχει δυνατότητα αναίρεσης. Για τον λόγο αυτό, το παρόν Σύστημα προσφέρει δυνατότητα επεξεργασίας των δεδομένων και μηνύματα επιβεβαίωσης πριν την διαγραφή.

**Τα μηνύματα που εμφανίζονται στο χρήστη είναι απλά και κατανοητά;**

Βαθμός: 3

Σχόλιο: Τα μηνύματα που εμφανίζονται στο χρήστη είναι αρκετά κατανοητά και απλά ώστε να δώσουν πληροφορίες για τον λόγο που προκάλεσε το σφάλμα ή για την επιβεβαίωση ενεργειών ωστόσο δεν είναι πάντα ξεκάθαρα, με απώτερο σκοπό την απόκρυψη πληροφοριών για την

### Υπάρχει επιλογή βοήθειας στο σύνολο επιλογών;

Βαθμός: 1

Σχόλιο: Δεν υπάρχει επιλογή βοήθειας στο σύνολο επιλογών του λογισμικού. Προβλέπεται να συμπεριληφθεί σε μελλοντική επέκταση.

### **Σύνδεση με Διακομιστή – DDNS**

Μία **διεύθυνση IP (Ip address - Internet Protocol address)**, είναι ένας μοναδικός αριθμός που χρησιμοποιείται από συσκευές για τη μεταξύ τους αναγνώριση και συνεννόηση σε ένα δίκτυο υπολογιστών που χρησιμοποιεί το Internet Protocol standard. Κάθε συσκευή που ανήκει στο δίκτυο πρέπει να έχει τη δική της μοναδική διεύθυνση. Μία διεύθυνση IP μπορεί να θεωρηθεί το αντίστοιχο μιας διεύθυνσης κατοικίας ή ενός αριθμού τηλεφώνου (σύγκριση με VoIP) για έναν υπολογιστή ή άλλη συσκευή δικτύου στο Διαδίκτυο. Όπως κάθε διεύθυνση κατοικίας και αριθμός τηλεφώνου αντιστοιχούν σε ένα και μοναδικό κτίριο ή τηλέφωνο, μια IP address χρησιμοποιείται για τη μοναδική αναγνώριση ενός υπολογιστή ή άλλης συσκευής που συνδέεται στο δίκτυο.

Μια διεύθυνση IP μπορεί να "μοιράζεται" σε πολλές συσκευές-πελάτες είτε επειδή αυτές είναι μέρος ενός shared hosting web server environment, είτε λόγω ενός proxy server (π.χ. ενός Παροχέα Υπηρεσιών Διαδικτύου (ISP) ή μιας υπηρεσίας για εξασφάλιση ανωνυμίας - anonymizer service) που λειτουργούν ως μεσολαβητές. Στην τελευταία περίπτωση (χρήση διακομιστή μεσολάβησης) η πραγματική διεύθυνση IP μπορεί να αποκρύπτεται από το διακομιστή που δέχεται αίτηση. Η αναλογία στα τηλεφωνικά συστήματα θα ήταν η χρήση διεθνών ή τοπικών αριθμών κλήσης (proxy) και επεκτάσεων (shared).

Μια υπηρεσία εύρεσης δικτύου (network lookup service), το Domain Name Service (DNS), δίνει τη δυνατότητα να αντιστοιχηθούν ονόματα υπολογιστών (hostnames) σε μια διεύθυνση IP. Με αυτό τον τρόπο οι άνθρωποι μπορούν εύκολα να θυμούνται ένα όνομα και όχι μια σειρά αριθμών. Το DNS επιτρέπει σε πολλαπλές διευθύνσεις και ονόματα να δείχνουν σε ένα πόρο του Διαδικτύου.

Ένας ακόμη λόγος ύπαρξης του DNS είναι να επιτρέπει, για παράδειγμα, σε έναν ιστότοπο που φιλοξενείται σε πολλούς διακομιστές, καθένας από τους οποίους έχει τη δική του διεύθυνση IP, να παρέχει στοιχειώδη εξισορρόπηση φόρτου (load balancing), δηλαδή μία ορθή διαμοίραση της κίνησης που δέχεται ο ιστότοπος ανάμεσα στους υπολογιστές που τον φιλοξενούν.

Οι διευθύνσεις IP ορίζονται είτε μόνιμα, σε ένα διακομιστή ο οποίος βρίσκεται πάντα στην ίδια διεύθυνση, είτε προσωρινά από ένα πλήθος διαθέσιμων διευθύνσεων. Έτσι, χωρίζονται σε δυναμικές, που αντιστοιχούν συνήθως σε προσωπικούς υπολογιστές ή προγράμματα πελάτες και σε στατικές, κυρίως για servers. Οι δυναμικές διευθύνσεις αλλάζουν

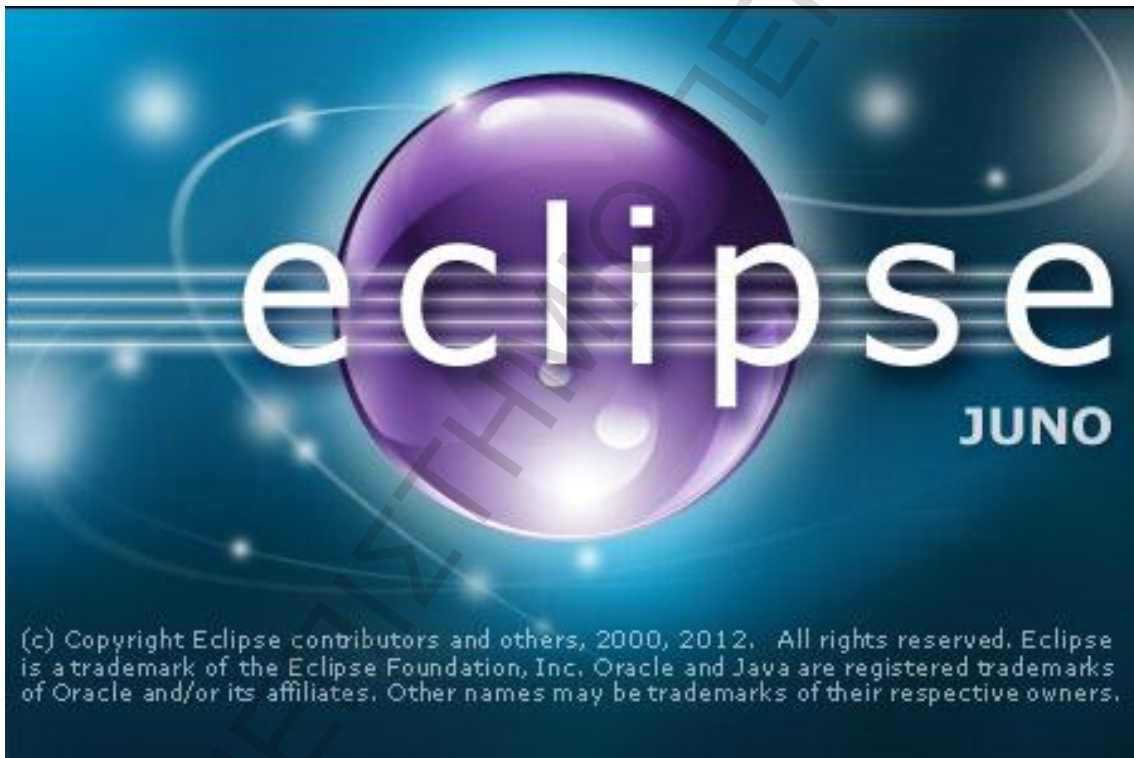
Μεταπτυχιακή Διατριβή

Παναγιώτης Κοκκότης

κάθε φορά που η συσκευή συνδέεται στο διαδίκτυο με αποτέλεσμα να καθίσταται αδύνατο να παίξει το ρόλο του διακομιστή. Η λύση που προτείνεται σε αυτό το πρόβλημα είναι η χρήση δυναμικής διεύθυνσης dns η οποία δείχνει πάντα στον υπολογιστή - server ξεπερνώντας το εμπόδιο της αλλαγής διεύθυνσης. Η διεύθυνσης που χρησιμοποιείται είναι η <http://pkokkotis.no-ip.org/>.

## Eclipse

Πρόκειται για ένα από τα πιο δημοφιλή Ολοκληρωμένα Περιβάλλοντα Ανάπτυξης (Integrated Development Environment) και πολύ διαδεδομένο στην αγορά, δεδομένου ότι προσφέρει επαγγελματικές επιλογές και πληθώρα εργαλείων ικανών να εξυπηρετήσουν οποιαδήποτε ανάγκη σε μία μεγάλη ποικιλία υποστηριζόμενων γλωσσών προγραμματισμού. Επιπλέον, προσφέρει και δυνατότητες σχεδιασμού (UML) με πολλές επιλογές και σε αυτόν τον τομέα.



### Εικόνα 4-31. Eclipse Juno

Το συγκεκριμένο λογισμικό είναι ανοιχτού κώδικα και προσφέρεται δωρεάν από την ιστοσελίδα <http://www.eclipse.org> ενώ, το μεγαλύτερο μέρος του, έχει υλοποιηθεί σε γλώσσα Java, που είναι και η βασική γλώσσα, για την οποία κάποιος επιλέγει να χρησιμοποιήσει το Eclipse, καθώς υποστηρίζεται πλήρως.

Το Eclipse ξεκίνησε ως project της IBM του Καναδά με στόχο την δημιουργία ενός ολοκληρωμένου εργαλείου ανάπτυξης σε Java. Το Νοέμβριο

Μεταπτυχιακή Διατριβή Παναγιώτης Κοκκότης  
του 2001 ιδρύθηκε μια κοινοπραξία με στόχο την περαιτέρω ανάπτυξη του εργαλείου ως λογισμικό ανοιχτού κώδικα, με ιδρυτικά μέλη τις εταιρείες Borland, IBM, Merant, QNX Software Systems, Rational Software, Red Hat, SuSE, TogetherSoft και WebGain. Τα μέλη ξεπέρασαν τα 80, στα τέλη του 2003, αποδεικνύοντας τη σπουδαιότητα του εγχειρήματος. Τον Ιανουάριο του 2004 δημιουργήθηκε η Eclipse Foundation και στις 21 Ιουνίου του ίδιου έτους κυκλοφόρησε η πρώτη έκδοση του λογισμικού. Από τότε και κάθε χρόνο, την τέταρτη Τετάρτη του Ιουνίου κυκλοφορεί μια νέα έκδοση. Στις μέρες μας έχουμε φτάσει αισίως στην έκδοση 4.2, με την χαρακτηριστική ονομασία Eclipse Juno (Εικόνα 4-31) ενώ για τις 26 Ιουνίου 2013 έχει προγραμματιστεί η κυκλοφορία του Eclipse Kepler.

Μεταξύ των αναρίθμητων πρόσθετων (plug ins) που προσφέρονται για εγκατάσταση στο Eclipse βρίσκεται και το πακέτο ανάπτυξης Android Development Tools, το οποίο παρέχει τα κατάλληλα εργαλεία για δημιουργία εφαρμογών για φορητές συσκευές που τρέχουν το λογισμικό της Google. Αυτός είναι ένας ακόμα πολύ σημαντικός λόγος που κάνει το Eclipse εξαιρετικά δημοφιλές.

Άλλα περιβάλλοντα ανάπτυξης που χρησιμοποιούνται ευρέως από τους προγραμματιστές είναι το Netbeans, το οποίο επίσης παρέχεται δωρεάν και στρέφεται και αυτό γύρω από τη Java, αλλά και το Visual Studio της Microsoft, που χρησιμοποιείται κυρίως για την κατασκευή εφαρμογών για Windows και Windows phones. Παρέχεται δωρεάν στην περιορισμένη έκδοση Express και επί πληρωμή σε μια σειρά από πλήρεις εκδόσεις.



# Κεφάλαιο 5

## Συμπεράσματα - Μελλοντικές Επεκτάσεις

## 5. Συμπεράσματα – Μελλοντικές Επεκτάσεις

Στην Παρούσα Μεταπτυχιακή Διατριβή αναπτύχθηκε ένα Ιατρικό Πληροφοριακό Σύστημα, με γνώμονα την προσαρμογή στις μελλοντικές τεχνολογικές εξελίξεις. Το Σύστημα αυτό παρέχει πρόσβαση σε διαφορετικά είδη χρηστών, διευκολύνει κάποιες συνηθισμένες χρονοβόρες διαδικασίες και προτείνει τρόπους ταχύτερης και ασφαλέστερης επικοινωνίας μεταξύ του πελάτη και του διακομιστή, που μπορεί να αξιοποιηθεί σε ανάλογες περιπτώσεις Συστημάτων που χρησιμοποιούν την συγκεκριμένη αρχιτεκτονική.

Ο διακομιστής (server) θα φιλοξενείται συνήθως μέσα σε μία ιατρική μονάδα και θα επικοινωνεί με πολλούς διαφορετικούς πελάτες, οι οποίοι θα χρησιμοποιούν είτε την desktop εφαρμογή είτε την αντίστοιχη για λειτουργικό Android, από το κινητό ή το tablet τους. Στο Σύστημα αποθηκεύονται κατά κανόνα ιστορικές πληροφορίες και δημογραφικά δεδομένα, τα οποία μόνο οι εργαζόμενοι στην ιατρική μονάδα, μπορούν να επεξεργαστούν ή να αποθηκεύσουν. Η πρωτοπορία του Συστήματος όμως είναι ότι δίνει τη δυνατότητα σε κάθε επιβλεπόμενο να έχει πρόσβαση στα προσωπικά του δεδομένα οποιαδήποτε στιγμή και από οπουδήποτε. Φυσικά δεν μπορεί να τα επεξεργαστεί αλλά είναι εξαιρετικής σημασίας να παραμένει ενημερωμένος και για τη δική του διευκόλυνση και για λόγους επιχειρηματικής στρατηγικής της ιατρικής μονάδας.

Ένας επιπλέον παράγοντας που μπορεί να θεωρηθεί πρωτοποριακός είναι το φιλικό περιβάλλον διεπαφής με τον χρήστη. Το μεγαλύτερο σφάλμα του προγραμματιστή κατά την ανάπτυξη ενός προγράμματος είναι η αντιμετώπιση των πραγμάτων από τη δική του σκοπιά. Επειδή κάτι το καταλαβαίνει αυτός δεν σημαίνει ότι μπορεί να το κατανοήσει και ο χρήστης. Για τον λόγο αυτό, έχει καταβληθεί μεγάλη προσπάθεια ώστε το πρόγραμμα να τηρεί τους βασικούς κανόνες που να το κάνουν εύχρηστο και ευέλικτο. Η κεντρική διεπαφή είναι χωρισμένη σε καρτέλες, κάθε μία από τις οποίες σχετίζεται με συγκεκριμένο αντικείμενο. Για παράδειγμα, η καρτέλα 'Φάρμακα' διαχειρίζεται τα φάρμακα και τις συνταγές. Με τον ίδιο τρόπο λειτουργούν και οι υπόλοιπες καρτέλες, υποστηρίζοντας μία συγκεκριμένη ροή λειτουργιών, που εξυπηρετεί στην γρηγορότερη εκμάθηση από τον χρήστη.

Το Σύστημα έχει επίσης σχεδιαστεί έτσι ώστε να ξεχωρίζει τα διαφορετικά πρόσωπα που μπορούν να εισέλθουν σε αυτό. Ο καθένας από τους χρήστες μπορεί να έχει διαφορετικές δυνατότητες και η πρόσβασή του στα δεδομένα να χωρίζεται σε επίπεδα. Ο επιβλεπόμενος π.χ. δεν μπορεί να παρακολουθήσει τίποτα άλλο, εκτός από τα στοιχεία που τον αφορούν. Έτσι, ακόμα κι αν πέσει στα χέρια του το εκτελέσιμο πρόγραμμα της κεντρικής εφαρμογής, δεν μπορεί να εισέλθει με το δικό του όνομα και κωδικό χρήστη, τα οποία χρησιμοποιεί στο κινητό του τηλέφωνο. Ένα άλλο παράδειγμα είναι η εξάρτηση του συστήματος από τον διαχειριστή. Πρόκειται για έναν άνθρωπο εξουσιοδοτημένο να κάνει τα πάντα, ο οποίος είναι αρμόδιος και για το ποιος και σε ποιο βαθμό μπορεί να έχει πρόσβαση στο Σύστημα. Έτσι, μπορεί να ελέγχει την εγκυρότητα καθενός από τους πολλούς (ίσως και χιλιάδες) ανθρώπους που μπορούν να αποκτήσουν πρόσβαση.

Το πιο ευαίσθητο σημείο ενός Πληροφοριακού Συστήματος και ιδιαίτερα ιατρικού περιεχομένου είναι η ασφάλεια των δεδομένων. Για

τον λόγο αυτό κανένας χρήστης δεν έχει απευθείας πρόσβαση στη Βάση Δεδομένων, η οποία βρίσκεται προστατευμένη στον κεντρικό διακομιστή και μόνο αυτός μπορεί να επικοινωνήσει μαζί της. Από τον σχεδιασμό του Συστήματος φαίνεται και ο τρόπος που λειτουργεί η εν λόγω διαδικασία. Όλοι οι πελάτες στέλνουν ερωτήσεις και ο διακομιστής αναλαμβάνει να τις αξιολογήσει και να τις μετατρέψει σε sql ερωτήματα. Στην αντίστροφη διαδικασία μετατρέπει τα δεδομένα που αντλεί από τη ΒΔ σε μορφή JSON και τα διανέμει στους πελάτες. Επιπλέον, αυτό αποκλείει την περίπτωση κακόβουλης επίθεσης (SQL Injection), η οποία σε δεύτερο επίπεδο αποκλείεται και από τα Prepared Statements που προσφέρει η γλώσσα Java.

Η παρούσα εργασία είναι μια προσπάθεια να προταθεί ένας τρόπος προσέγγισης του σχεδιασμού ανάλογων Ιατρικών Πληροφοριακών Συστημάτων. Έχει δοθεί ιδιαίτερη έμφαση στην δυνατότητα να υποστηρίζονται εύκολα μελλοντικές επεκτάσεις. Υπάρχει μία κλάση 'Tab' από την οποία κληρονομούν όλες οι καρτέλες. Με τον τρόπο αυτό κάθε νέα καρτέλα, που επιθυμούμε να προσθέσουμε, επεκτείνοντας την παραπάνω κλάση μπορεί εύκολα να ενσωματωθεί στο Σύστημα. Κάτι ανάλογο συμβαίνει και στην πλευρά του server όπου οι κλάσεις είναι χωρισμένες ανά λειτουργία. Επομένως κάθε νέα καρτέλα απαιτεί και μία νέα κλάση στο servlet, που με τη σειρά της απαιτεί έναν ή δύο νέους πίνακες στη Βάση Δεδομένων. Για να γίνει πιο κατανοητό αυτό θα δούμε το παράδειγμα των εμβολίων. Πρώτα δημιουργούμε εύκολα τη νέα καρτέλα, ως επέκταση της κλάσης 'Tab'. Ύστερα δημιουργούμε μία κλάση 'Immunizations' στην πλευρά του διακομιστή και στη συνέχεια δύο πίνακες στη ΒΔ, ένας για τα εμβόλια και ένας για το ιστορικό των εμβολιασμών. Οι υπόλοιπες ενέργειες είναι τυποποιημένες κι έτσι, σε μικρό χρονικό διάστημα και με λίγο κόπο, έχουμε μια επέκταση του συστήματος.

Επίσης, στην παρούσα Διατριβή προτείνεται η τυποποίηση των δεδομένων σε μορφή JSON. Πρόκειται για ένα ευαίσθητο κομμάτι που εξαρτάται και από τις περιορισμένες δυνατότητες που έχουμε στη διάθεσή μας. Η ταχύτητα είναι ένα σημαντικό θέμα, όταν μιλάμε για μεταφορά δεδομένων μέσω διαδικτύου. Ένα άλλο θέμα είναι η αδυναμία να ανταλλάσσονται αντικείμενα μεταξύ πελάτη και διακομιστή. Ένα τρίτο θέμα είναι η κατανόηση των δεδομένων από τις εφαρμογές και ο τρόπος γεμίσματος των πινάκων. Η λύση που προτείνεται είναι η αποστολή τυποποιημένων δεδομένων, ενός αντικειμένου JSON, σε μορφή txt. Έτσι, επιτυγχάνουμε υψηλές ταχύτητες μεταφοράς μεγάλου όγκου δεδομένων, καθώς τα txt αρχεία είναι μικρά σε μέγεθος αλλά και τα μόνα που μπορούν να χρησιμοποιηθούν για την επικοινωνία μεταξύ των μερών, έχουμε τη δυνατότητα να στείλουμε αντικείμενο, το οποίο απλά έχει μετατραπεί σε κείμενο και φυσικά γίνεται εύκολα κατανοητό από το πρόγραμμα, το οποίο αναλαμβάνει να ξεχωρίσει τα δεδομένα και να γεμίσει τους πίνακες ή τις λίστες.

Το Σύστημα μπορεί μελλοντικά να επεκταθεί με πολλούς διαφορετικούς τρόπους. Η εφαρμογή για Android θα πρέπει να επεκταθεί και σε άλλα λειτουργικά συστήματα, όπως το ios και τα windows. Επίσης, μπορεί να δοθεί δυνατότητα χρήσης του συστήματος από τέτοιες συσκευές και σε υπαλλήλους και διαχειριστές. Δηλαδή, να επιτελούνται λειτουργίες εισαγωγής και επεξεργασίας δεδομένων από φορητές συσκευές από τους εξουσιοδοτημένους χρήστες. Οι καρτέλες που περιλαμβάνει το κεντρικό πρόγραμμα είναι απλά ενδεικτικές και μπορούν εύκολα να προσαρμοστούν και να πολλαπλασιαστούν, ανάλογα με



Μέσα στις προσωπικές επιθυμίες του συντάκτη της παρούσας Διατριβής είναι να δοκιμαστεί και η περίπτωση χρήσης της γλώσσας php στην πλευρά του διακομιστή. Επίσης, θα γίνει προσπάθεια χρησιμοποίησης τεχνολογιών web για δημιουργία ιστοσελίδας αντί για την διανομή των εκτελέσιμων java προγραμμάτων. Άλλη μία επιλογή είναι η δοκιμή και διαφορετικών συστημάτων διαχείρισης Βάσεων Δεδομένων (DBMS), όπως Oracle ή PostgresQL.

Η mobile εφαρμογή θα μοιάζει με αυτή της Εικόνας 5-1 για λειτουργικό σύστημα Windows phone. Για την ανάπτυξη της έχει χρησιμοποιηθεί το phonegap, ένα template που επιτρέπει την χρήση δημοφιλών διαδικτυακών τεχνολογιών, όπως html, css και javascript για την ανάπτυξη τέτοιου είδους εφαρμογών. Προς το παρόν βρίσκεται σε αρχικό στάδιο ανάπτυξης αλλά με την ολοκλήρωσή της θα μπορεί να μεταφερθεί αυτούσια και σε πληθώρα άλλων λειτουργικών όπως τα παρακάτω:

- BlackBerry
- iOS
- Symbian
- WebOS
- Windows 8
- Bada
- Tizen

Μπορεί ακόμα και να μεταφερθεί σε Android ώστε να έχει την ίδια μορφή σε όλες τις περιπτώσεις. Με αυτόν τον τρόπο επιτυγχάνεται η κάλυψη της αγοράς των smartphones, tablets αλλά και των σύγχρονων υπολογιστικών συστημάτων με λειτουργικό window 8.



**Εικόνα 5-1. Windows phone εφαρμογή**



# Κεφάλαιο 6

## Παράρτημα

## 6. Παράρτημα

### Η Βάση Δεδομένων

Ο παρακάτω κώδικας SQL δημιουργεί τη Βάση Δεδομένων, με τους 15 πίνακες που την απαρτίζουν. Περιέχεται σε ένα αρχείο script και χρησιμεύει κατά τη διαδικασία εγκατάστασης. Οι πίνακες είναι κανονικοποιημένοι έτσι ώστε να αποφεύγονται οι επαναλήψεις πληροφορίας. Στο κεφάλαιο 3 αναλύσαμε τα επί μέρους στοιχεία της ΒΔ και εδώ μπορούμε να δούμε τον τρόπο με τον οποίο δημιουργούνται. Είναι χωρισμένα, έτσι ώστε να διακρίνεται ο κάθε πίνακας και χρωματισμένα για να φαίνονται τα ονόματα, οι δηλώσεις τύπων και όσες ακόμα εντολές συμμετέχουν στον σχηματισμό του πιο ευαίσθητου και νευραλγικού σκέλους τους παρόντος Πληροφοριακού Συστήματος.

```

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;

SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;

SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';

CREATE SCHEMA IF NOT EXISTS `medicalDB` DEFAULT CHARACTER SET utf8
COLLATE utf8_general_ci ;
USE `medicalDB` ;

```

```

-----
-- Table `medicalDB`.`registry`
-----

```

```

CREATE TABLE IF NOT EXISTS `medicalDB`.`registry` (
  `reg_number` INT NOT NULL AUTO_INCREMENT ,
  `dateOfRegister` VARCHAR(45) NOT NULL ,
  `usertype` VARCHAR(3) NOT NULL ,
  PRIMARY KEY (`reg_number` )
ENGINE = InnoDB;

```

```

-----
-- Table `medicalDB`.`users`
-----

```

```

CREATE TABLE IF NOT EXISTS `medicalDB`.`users` (
  `usr_id` INT NOT NULL ,
  `username` VARCHAR(45) NOT NULL ,

```

```
`password` VARCHAR(45) NOT NULL ,
`isRegistered` TINYINT(1) NOT NULL DEFAULT FALSE ,
PRIMARY KEY (`usr_id`) ,
INDEX `fk_users_1_idx` (`usr_id` ASC) ,
UNIQUE INDEX `username_UNIQUE` (`username` ASC) ,
CONSTRAINT `fk_users_1`
  FOREIGN KEY (`usr_id`)
  REFERENCES `medicalDB`.`registry` (`reg_number`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-----
-- Table `medicalDB`.`employees`
-----
```

```
CREATE TABLE IF NOT EXISTS `medicalDB`.`employees` (
  `emp_id` INT NOT NULL ,
  `name` VARCHAR(25) NOT NULL ,
  `surname` VARCHAR(25) NOT NULL ,
  `address` VARCHAR(45) NOT NULL ,
  `dateOfBirth` DATE NOT NULL ,
  `phoneNo_home` VARCHAR(14) NULL ,
  `phoneNo_mobile` VARCHAR(14) NULL ,
  PRIMARY KEY (`emp_id`) ,
  INDEX `fk_employees_1_idx` (`emp_id` ASC) ,
  CONSTRAINT `fk_employees_1`
    FOREIGN KEY (`emp_id`)
    REFERENCES `medicalDB`.`registry` (`reg_number`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-----
-- Table `medicalDB`.`doctors`
-----
```

```
CREATE TABLE IF NOT EXISTS `medicalDB`.`doctors` (  
  `doc_id` INT NOT NULL ,  
  `name` VARCHAR(25) NOT NULL ,  
  `surname` VARCHAR(25) NOT NULL ,  
  `address` VARCHAR(45) NOT NULL ,  
  `dateOfBirth` DATE NOT NULL ,  
  `phoneNo_home` VARCHAR(14) NULL ,  
  `phoneNo_mobile` VARCHAR(14) NULL ,  
  `speciality` VARCHAR(45) NOT NULL ,  
  PRIMARY KEY (`doc_id`) ,  
  INDEX `fk_doctors_1_idx` (`doc_id` ASC) ,  
  CONSTRAINT `fk_doctors_1`  
    FOREIGN KEY (`doc_id` )  
    REFERENCES `medicalDB`.`registry` (`reg_number` )  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-----  
-- Table `medicalDB`.`supervised`  
-----
```

```
CREATE TABLE IF NOT EXISTS `medicalDB`.`supervised` (  
  `sup_id` INT NOT NULL ,  
  `name` VARCHAR(25) NOT NULL ,  
  `surname` VARCHAR(25) NOT NULL ,  
  `address` VARCHAR(45) NOT NULL ,  
  `dateOfBirth` DATE NOT NULL ,  
  `phoneNo_home` VARCHAR(14) NULL ,  
  `phoneNo_mobile` VARCHAR(14) NULL ,  
  PRIMARY KEY (`sup_id`) ,  
  INDEX `fk_supervised_1_idx` (`sup_id` ASC) ,  
  CONSTRAINT `fk_supervised_1`  
    FOREIGN KEY (`sup_id` )  
    REFERENCES `medicalDB`.`registry` (`reg_number` )  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)
```

```
-----  
-- Table `medicalDB`.`administrator`  
-----  
CREATE TABLE IF NOT EXISTS `medicalDB`.`administrator` (  
  `admin_id` INT NOT NULL ,  
  `name` VARCHAR(25) NOT NULL ,  
  `surname` VARCHAR(25) NOT NULL ,  
  `address` VARCHAR(45) NOT NULL ,  
  `dateOfBirth` DATE NOT NULL ,  
  `phoneNo_home` VARCHAR(14) NULL ,  
  `phoneNo_mobile` VARCHAR(14) NULL ,  
  PRIMARY KEY (`admin_id`) ,  
  INDEX `fk_administrator_1_idx` (`admin_id` ASC) ,  
  CONSTRAINT `fk_administrator_1`  
    FOREIGN KEY (`admin_id` )  
    REFERENCES `medicalDB`.`registry` (`reg_number` )  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-----  
-- Table `medicalDB`.`examinations`  
-----  
CREATE TABLE IF NOT EXISTS `medicalDB`.`examinations` (  
  `exam_id` INT NOT NULL AUTO_INCREMENT ,  
  `type` VARCHAR(45) NOT NULL ,  
  `cost` DOUBLE NOT NULL DEFAULT 0 ,  
  PRIMARY KEY (`exam_id` )  
ENGINE = InnoDB;
```

```
-----  
-- Table `medicalDB`.`history`  
-----
```

```

CREATE TABLE IF NOT EXISTS `medicalDB`.`history` (
  `h_id` INT NOT NULL AUTO_INCREMENT ,
  `supervised` INT NOT NULL ,
  `examination` INT NOT NULL ,
  `diagnosis` TEXT NOT NULL ,
  `date` DATE NOT NULL ,
  PRIMARY KEY (`h_id`) ,
  INDEX `username_FK_idx` (`supervised` ASC) ,
  INDEX `examination_FK_idx` (`examination` ASC) ,
  CONSTRAINT `username_FK`
    FOREIGN KEY (`supervised`)
    REFERENCES `medicalDB`.`supervised` (`sup_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `examination_FK`
    FOREIGN KEY (`examination`)
    REFERENCES `medicalDB`.`examinations` (`exam_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----

-- Table `medicalDB`.`rendezvous`
-----

CREATE TABLE IF NOT EXISTS `medicalDB`.`rendezvous` (
  `r_id` INT NOT NULL AUTO_INCREMENT ,
  `supervised` INT NOT NULL ,
  `doctor` INT NOT NULL ,
  `dateTime` DATETIME NOT NULL ,
  `status` VARCHAR(20) CHARACTER SET 'utf8' COLLATE 'utf8_general_ci'
  NOT NULL DEFAULT 'εκκρεμεί' ,
  PRIMARY KEY (`r_id`) ,
  INDEX `supervised_FK_idx` (`supervised` ASC) ,
  INDEX `doctor_FK_idx` (`doctor` ASC) ,
  CONSTRAINT `supervised_FK`

```



```
FOREIGN KEY (`supervised` )
REFERENCES `medicalDB`.`supervised` (`sup_id` )
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `doctor_FK`
FOREIGN KEY (`doctor` )
REFERENCES `medicalDB`.`doctors` (`doc_id` )
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-----
-- Table `medicalDB`.`medications`
-----
```

```
CREATE TABLE IF NOT EXISTS `medicalDB`.`medications` (
  `med_id` INT NOT NULL AUTO_INCREMENT ,
  `name` VARCHAR(45) NOT NULL ,
  `description` VARCHAR(45) NULL ,
  `amount` DOUBLE NULL ,
  PRIMARY KEY (`med_id` )
ENGINE = InnoDB;
```

```
-----
-- Table `medicalDB`.`prescriptions`
-----
```

```
CREATE TABLE IF NOT EXISTS `medicalDB`.`prescriptions` (
  `prs_id` INT NOT NULL AUTO_INCREMENT ,
  `medication` INT NOT NULL ,
  `doctor` INT NOT NULL ,
  `supervised` INT NOT NULL ,
  `frequency` VARCHAR(45) NULL ,
  `date` DATE NULL ,
  PRIMARY KEY (`prs_id` ) ,
  INDEX `prescriptions_fk1_idx` (`medication` ASC) ,
  INDEX `prescriptions_fk2_idx` (`doctor` ASC) ,
```

```
INDEX `prescriptions_fk3_idx` (`supervised` ASC) ,
CONSTRAINT `prescriptions_fk1`
  FOREIGN KEY (`medication` )
  REFERENCES `medicalDB`.`medications` (`med_id` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `prescriptions_fk2`
  FOREIGN KEY (`doctor` )
  REFERENCES `medicalDB`.`doctors` (`doc_id` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `prescriptions_fk3`
  FOREIGN KEY (`supervised` )
  REFERENCES `medicalDB`.`supervised` (`sup_id` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-----
-- Table `medicalDB`.`allergies`
-----
```

```
CREATE TABLE IF NOT EXISTS `medicalDB`.`allergies` (
  `al_id` INT NOT NULL AUTO_INCREMENT ,
  `type` VARCHAR(45) NOT NULL ,
  `cause` VARCHAR(45) NULL ,
  PRIMARY KEY (`al_id` )
ENGINE = InnoDB;
```

```
-----
-- Table `medicalDB`.`allergies_history`
-----
```

```
CREATE TABLE IF NOT EXISTS `medicalDB`.`allergies_history` (
  `alh_id` INT NOT NULL AUTO_INCREMENT ,
  `allergy` INT NULL ,
  `supervised` INT NULL ,
```

```

`date` DATE NULL ,
PRIMARY KEY (`alh_id`) ,
INDEX `allergies_history_fk1_idx` (`allergy` ASC) ,
INDEX `allergies_history_fk1_idx1` (`supervised` ASC) ,
CONSTRAINT `allergies_history_fk1`
  FOREIGN KEY (`allergy` )
  REFERENCES `medicalDB`.`allergies` (`al_id` )
  ON DELETE NO ACTION,
  ON UPDATE NO ACTION,
CONSTRAINT `allergies_history_fk2`
  FOREIGN KEY (`supervised` )
  REFERENCES `medicalDB`.`supervised` (`sup_id` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `medicalDB`.`immunizations`
-----

```

```

CREATE TABLE IF NOT EXISTS `medicalDB`.`immunizations` (
  `imm_id` INT NOT NULL AUTO_INCREMENT ,
  `name` VARCHAR(45) NOT NULL ,
  `target` VARCHAR(45) NOT NULL ,
  `cost` DOUBLE NOT NULL ,
  PRIMARY KEY (`imm_id` )
ENGINE = InnoDB;

```

```

-----
-- Table `medicalDB`.`immunizations_history`
-----

```

```

CREATE TABLE IF NOT EXISTS `medicalDB`.`immunizations_history` (
  `imh_id` INT NOT NULL AUTO_INCREMENT ,
  `immunization` INT NOT NULL ,
  `supervised` INT NOT NULL ,
  `date` DATE NOT NULL ,

```

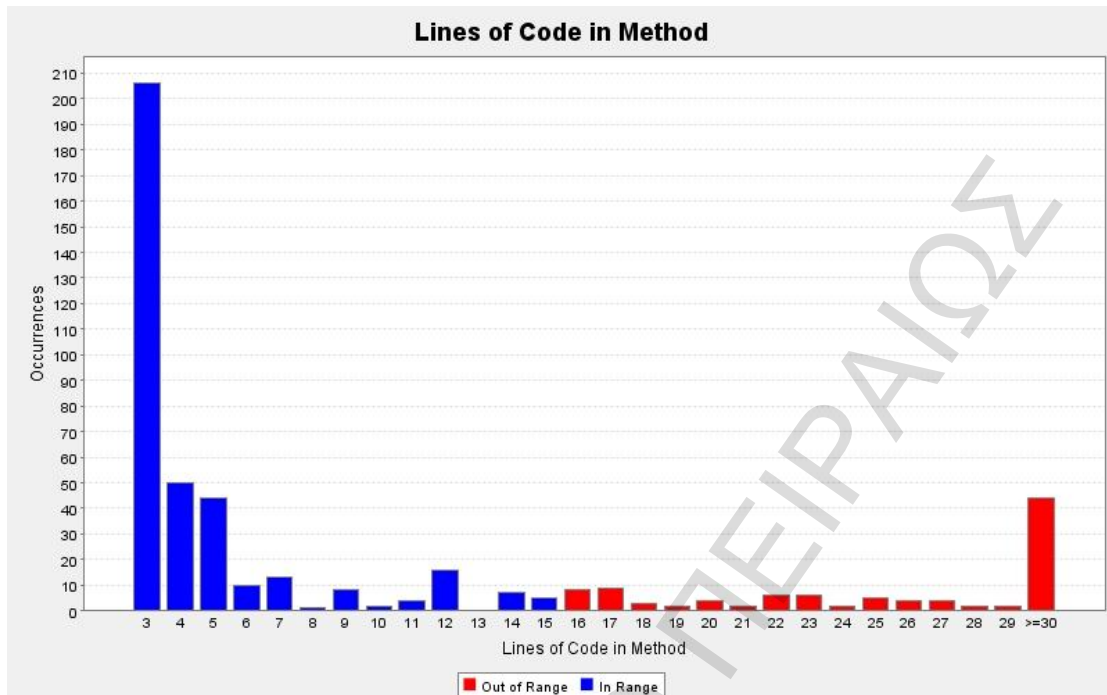
```
PRIMARY KEY (`imh_id`) ,
INDEX `immunizations_history_fk1_idx` (`immunization` ASC) ,
INDEX `immunizations_history_fk2_idx` (`supervised` ASC) ,
CONSTRAINT `immunizations_history_fk1`
  FOREIGN KEY (`immunization`)
  REFERENCES `medicalDB`.`immunizations` (`imm_id`)
  ON DELETE NO ACTION,
  ON UPDATE NO ACTION,
CONSTRAINT `immunizations_history_fk2`
  FOREIGN KEY (`supervised`)
  REFERENCES `medicalDB`.`supervised` (`sup_id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

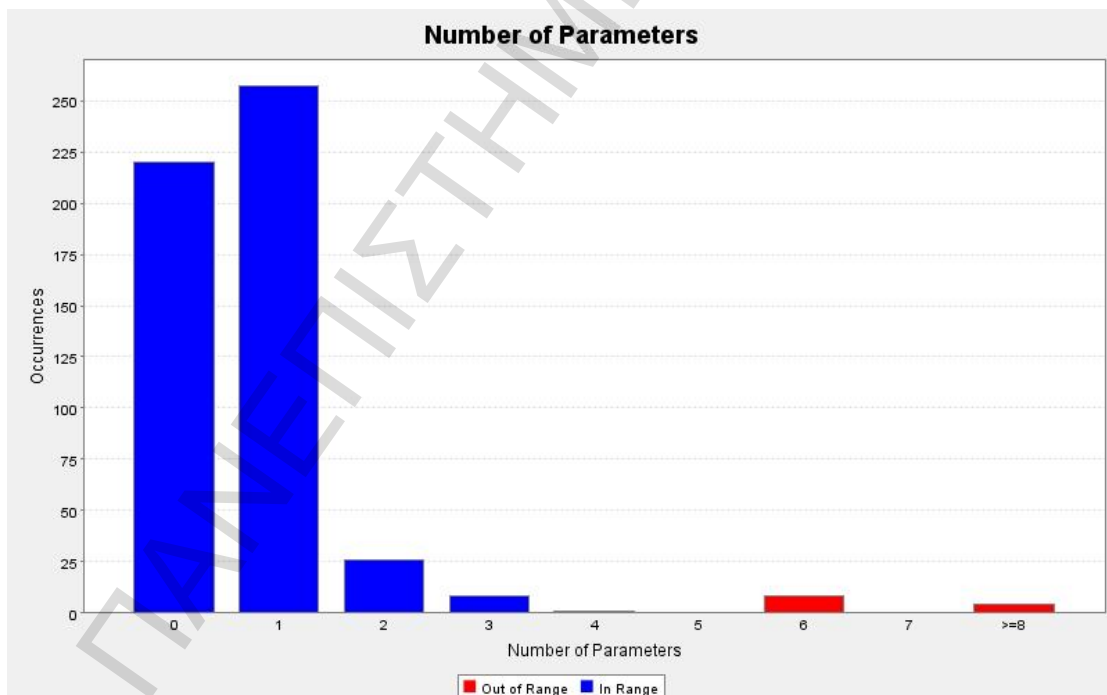
### Στατιστικά

Παρακάτω παρουσιάζονται ορισμένα στατιστικά στοιχεία των τριών βασικών projects της παρούσας Διατριβής. Για κάθε ένα παρουσιάζονται τρία διαγράμματα, που αφορούν τις γραμμές κώδικα που περιέχονται σε κάθε μέθοδο, τον αριθμό των παραμέτρων ανά μέθοδο και τον αριθμό των statements (δηλαδή των for, if, while κλπ).

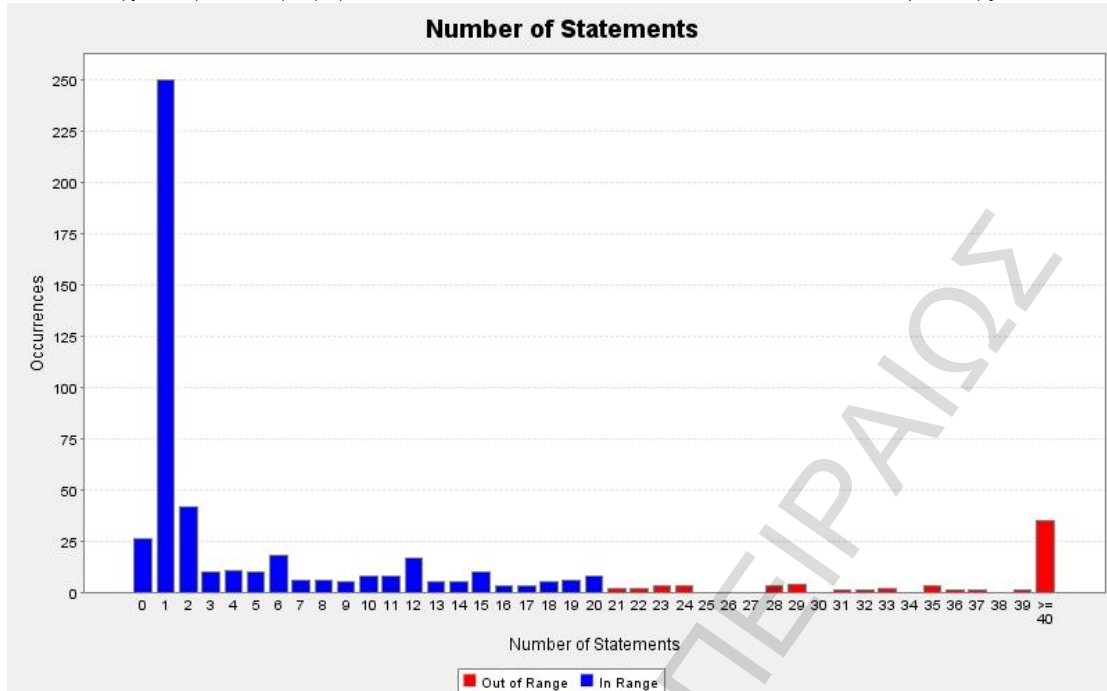
**MedInfoSystem**



**Εικόνα 6-1. MedInfoSystem – Γραμμές κώδικα**

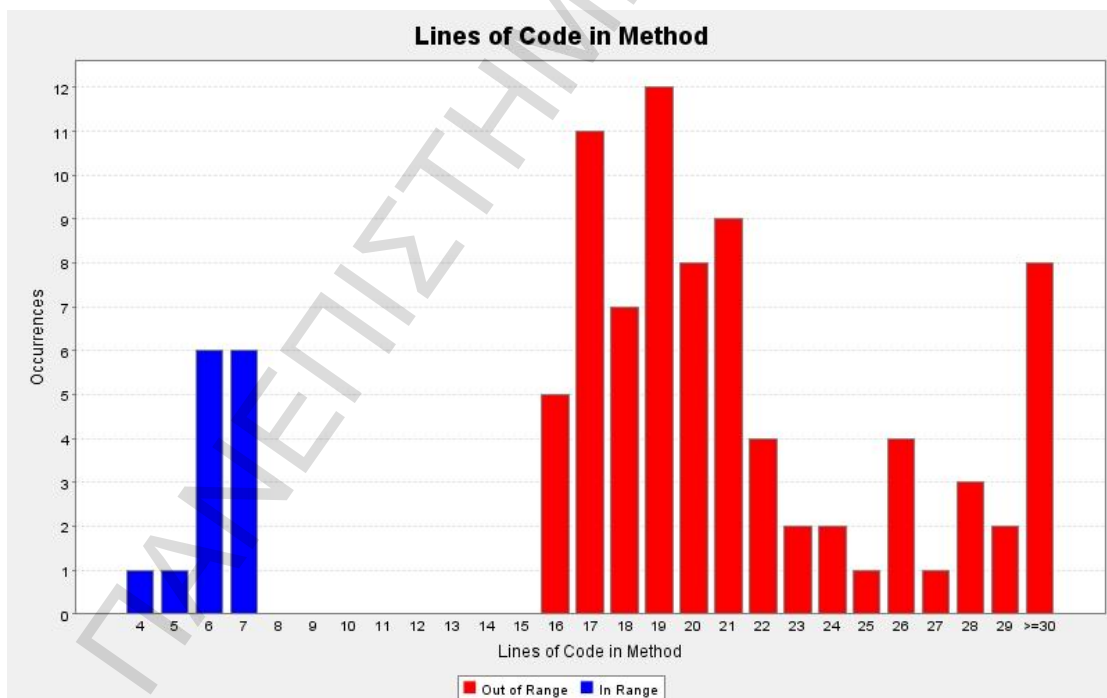


**Εικόνα 6-2. MedInfoSystem – Αριθμός παραμέτρων**

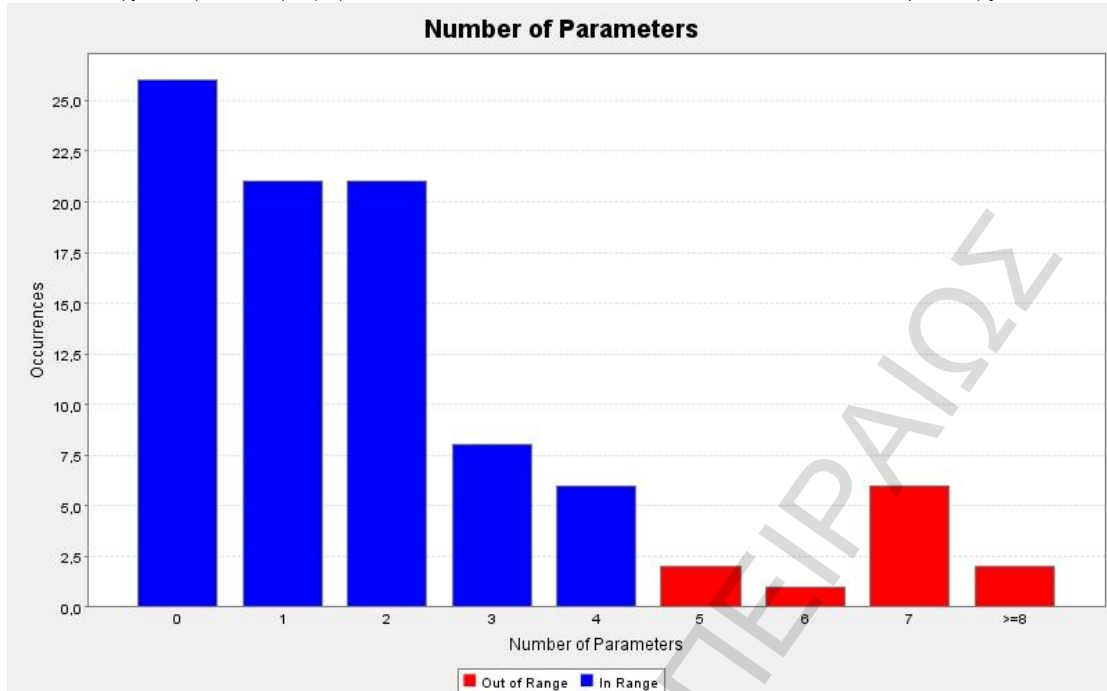


Εικόνα 6-3. MedInfoSystem – Αριθμός statements

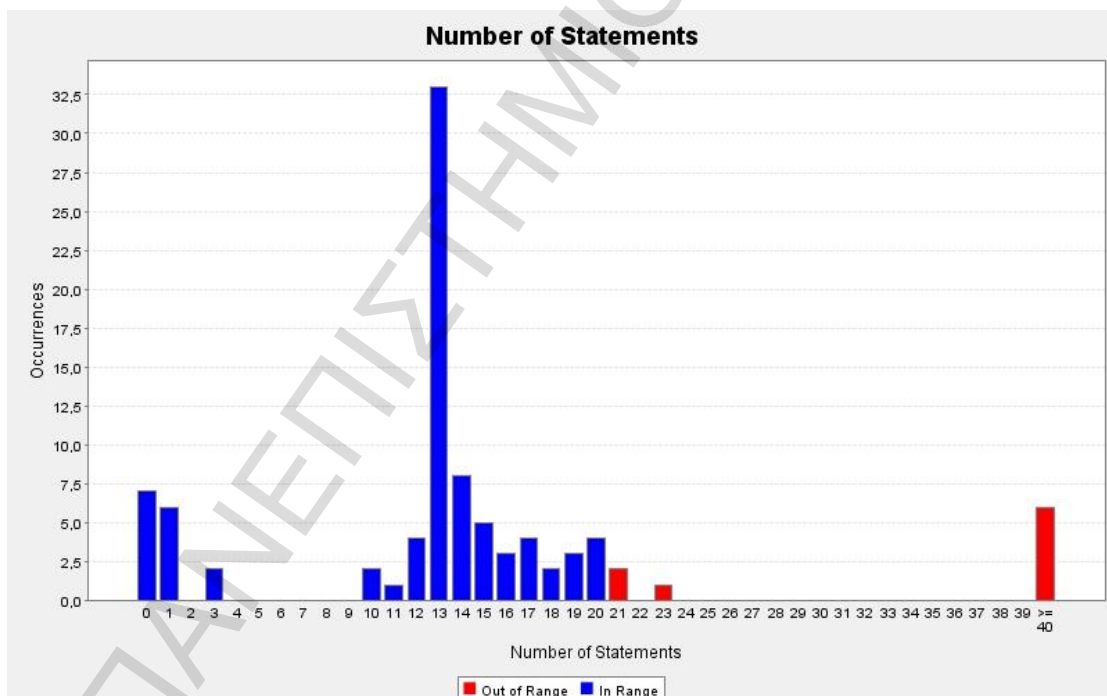
**Servlet**



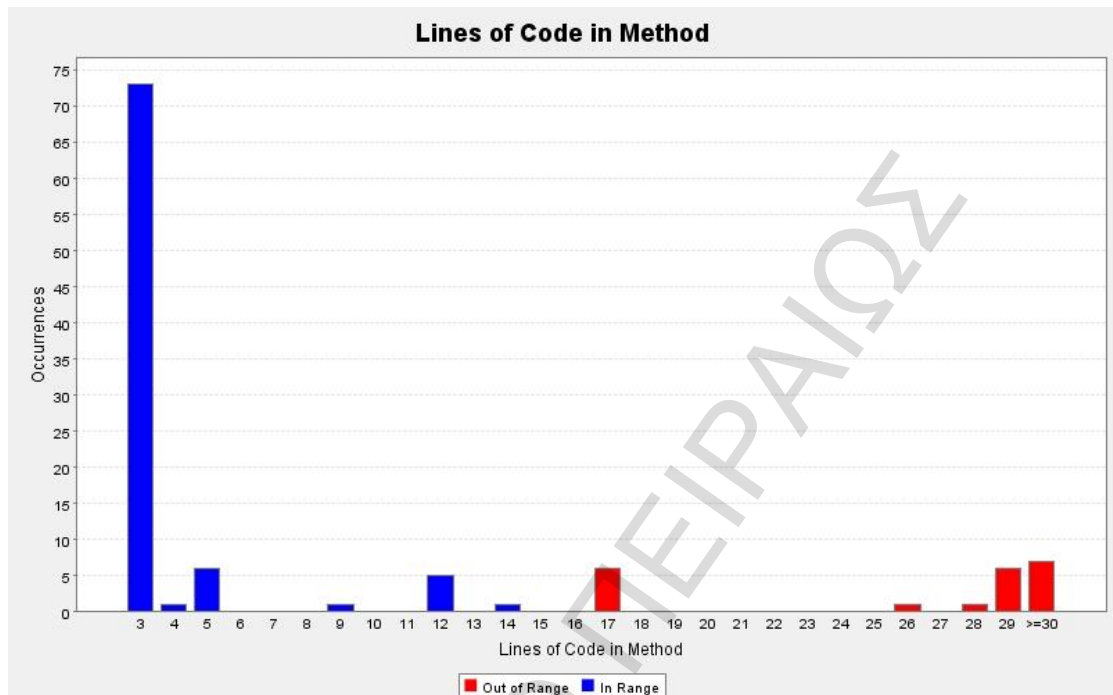
Εικόνα 6-4. Servlet – Γραμμές κώδικα



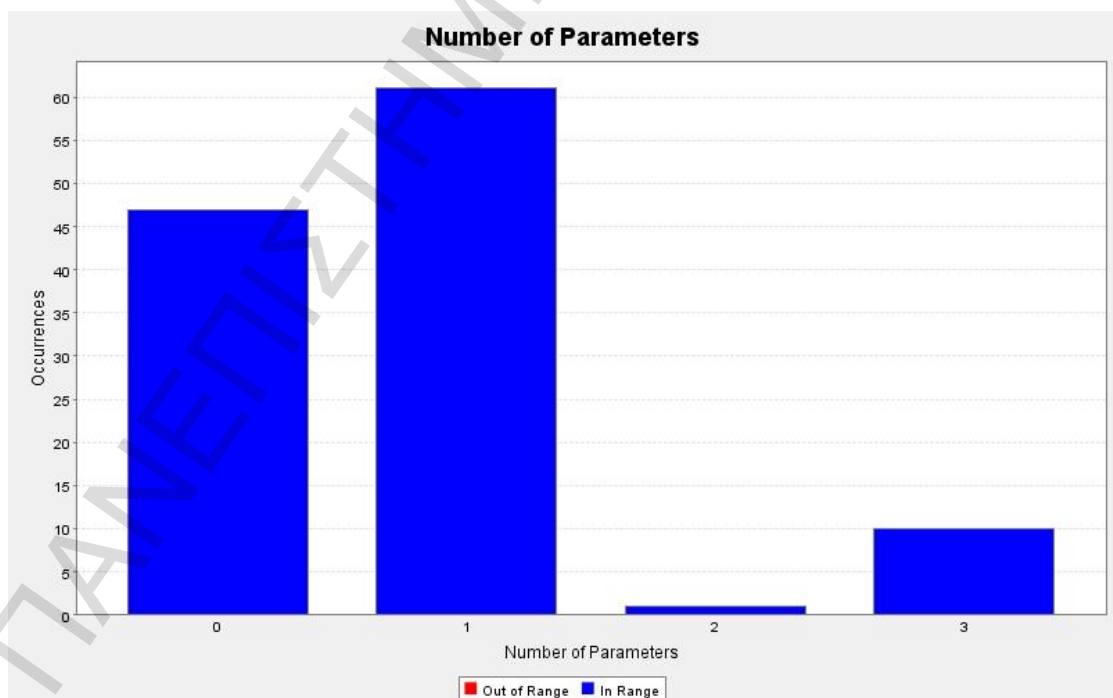
**Εικόνα 6-5. Servlet – Αριθμός παραμέτρων**



**Εικόνα 6-6. Servlet – Αριθμός statements**

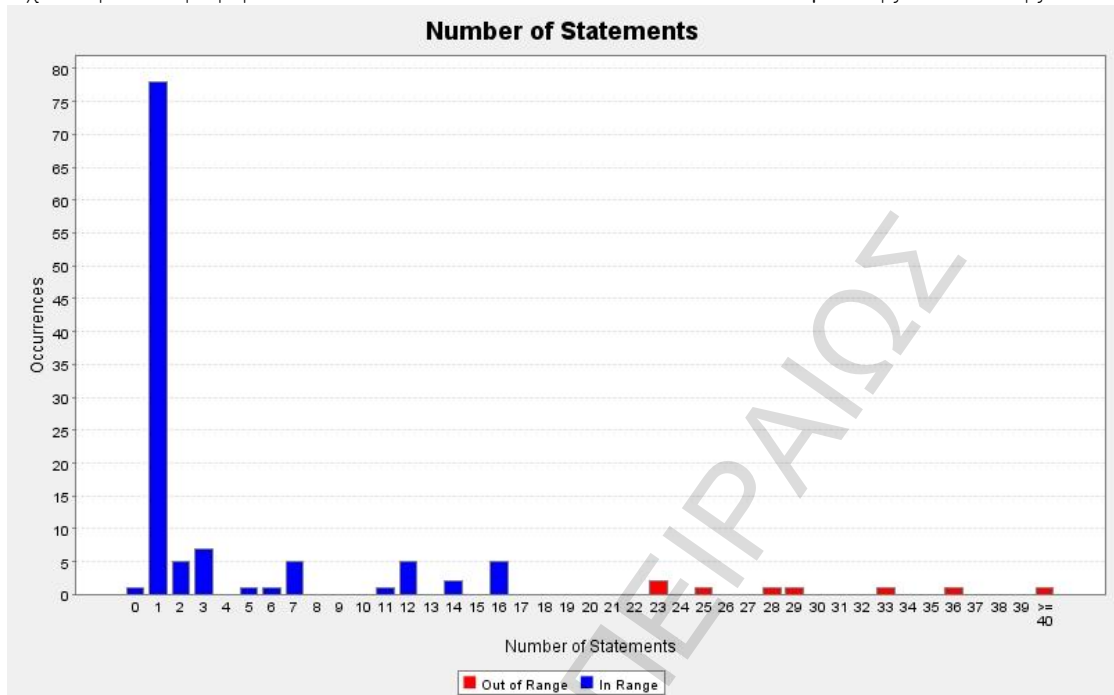


**Εικόνα 6-7. MedInfoDroid – Γραμμές κώδικα**



**Εικόνα 6-8. MedInfoDroid – Αριθμός παραμέτρων**





**Εικόνα 6-9. MedInfoDroid – Αριθμός statements**

## Οδηγίες Εγκατάστασης

Η παρούσα Μεταπτυχιακή Διατριβή συνοδεύεται από ένα CD με όλα τα απαραίτητα μέρη του Πληροφοριακού Συστήματος που αναπτύχθηκε, όπως περιγράφηκε στις προηγούμενες σελίδες. Το κεντρικό λογισμικό βρίσκεται σε ένα εκτελέσιμο αρχείο MedInfoSystem.jar και μπορεί να τρέξει χωρίς εγκατάσταση σε οποιονδήποτε υπολογιστή διαθέτει το απαραίτητο java runtime environment, το οποίο μπορεί να κατεβάσει από τη διεύθυνση <http://www.oracle.com/technetwork/java/javase/downloads/index.html>. Είναι ανεξάρτητο λειτουργικού συστήματος, όπως έχει ήδη αναφερθεί.

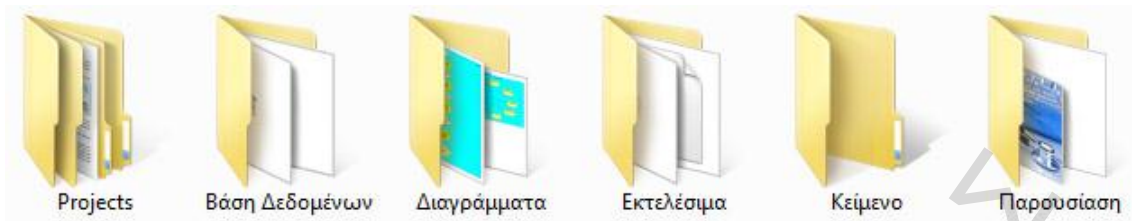
Απαραίτητο συστατικό για τη λειτουργία του Συστήματος αποτελεί η Βάση Δεδομένων. Για την εγκατάστασή της θα πρέπει να χρησιμοποιηθεί λειτουργικό από τη διεύθυνση <http://dev.mysql.com/downloads/> και συγκεκριμένα ένας server που αναλαμβάνει να εξυπηρετήσει τις ανάγκες της ΒΔ. Επίσης, στην ίδια διεύθυνση μπορεί να βρει κανείς και γραφικά περιβάλλοντα (GUI) όπως το Mysql Workbench που διευκολύνουν την διαχείριση της ΒΔ. Για την δημιουργία της ΒΔ θα πρέπει να χρησιμοποιηθεί το script MedicalDB.sql, που βρίσκεται και αυτό στα περιεχόμενα του cd, ή να αντιγραφεί ο κώδικας που έχει περιληφθεί στις προηγούμενες σελίδες. Η διαδικασία είναι απλή και η δημιουργία της ΒΔ γίνεται αυτόματα. Αμέσως μετά θα πρέπει να δημιουργηθεί ένας χρήστης που θα διαχειρίζεται το Σύστημα και η επιλογή να μην γίνει αυτό μέσω του λογισμικού αλλά σε αρχικό στάδιο έχει να κάνει με λόγους ασφαλείας. Τα βήματα για τη δημιουργία αυτού του χρήστη είναι:

1. Καταγραφή των στοιχείων στον πίνακα 'μητρώο'
2. Καταγραφή στον πίνακα 'διαχειριστής'
3. Καταγραφή στον πίνακα 'χρήστες'

Για την λειτουργία του servlet είναι απαραίτητος ο διακομιστής apache tomcat που παρέχεται στη διεύθυνση <http://tomcat.apache.org/download-70.cgi>. Η εγκατάσταση είναι απλή όπως και το ανέβασμα των αρχείων.

Όσον αφορά τη mobile εφαρμογή θα πρέπει να ακολουθηθούν τα παρακάτω βήματα:

1. Αντιγραφή του αρχείου MedInfoDroid.apk στην κάρτα μνήμης του τηλεφώνου και εισαγωγή της κάρτας στη συσκευή
2. Λήψη και εγκατάσταση της εφαρμογής Apps Installer από το Android Market
3. Το Apps Installer θα εμφανίσει όλα τα αρχεία .apk που βρίσκονται αποθηκευμένα στη μνήμη
4. Κλικ και εγκατάσταση της εφαρμογής

**Περιεχόμενα CD****Εικόνα 6-10. Περιεχόμενα CD**

- **Projects:** Περιέχονται τα τρία project της εργασίας έτοιμα για εισαγωγή στο Eclipse και ένας φάκελος με τις βιβλιοθήκες που χρησιμοποιήθηκαν. Εδώ μπορεί να βρει κάποιος και τον κώδικα.
- **Βάση Δεδομένων:** Το έργο που δημιουργήθηκε σε Mysql Workbench και το αρχείο medicalDB.sql για δημιουργία της ΒΔ
- **Διαγράμματα:** Εικόνες των διαγραμμάτων UML
- **Εκτελέσιμα:** Το λογισμικό που τρέχει κατευθείαν σε οποιοδήποτε λειτουργικό Σύστημα και η εφαρμογή για Android
- **Κείμενο:** Το παρόν κείμενο της Εργασίας
- **Παρουσίαση:** Η Παρουσίαση της Εργασίας

## Βιβλιογραφία

[1] Policy Implications Of Medical Information Systems (1977). Congress of the United States.

[2] Nancy Stagers, Cheryl Bagley Thompson, Rita Snyder-Halpern (2001). History and Trends in Clinical Information Systems in the United States.

[3] Reinhold Haux (2006). Health information systems - past, present, future. International Journal of Medical Informatics.

[4] Warren A. Barnett. Tri-service medical information system (trimis) pharmacy pilot test. Bureau of medicine and surgery, navy trimis office, washington dc.

[5] Evaluating computerised health information systems: hard lessons still to be learnt, <http://www.bmj.com/content/326/7394/860>

[6] <http://el.wikipedia.org>