



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΣΤΕΡΕΑΣ ΕΛΛΑΔΑΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗ ΒΙΟΙΑΤΡΙΚΗ**

**Σχεδιασμός και Ανάπτυξη Λογισμικού Διαχείρισης Ηλεκτρονικών
Ιατρικών Φακέλων με τη χρήση της πλατφόρμας Android**

Τυχάλας Δημήτριος

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ
Υπεύθυνος
Κακαρούντας Αθανάσιος**

Λαμία, 2010

© ΠΑ.ΣΤ.ΕΛ., 2010

Η παρούσα διατριβή και τα λοιπά αποτελέσματα της αντίστοιχης Πτυχιακής Εργασίας (ΠΕ) αποτελούν συνιδιοκτησία του ΠΑ.ΣΤ.ΕΛ. και του φοιτητή, ο καθένας από τους οποίους έχει το δικαίωμα ανεξάρτητης χρήσης και αναπαραγωγής τους (στο σύνολο ή τμηματικά) για διδακτικούς και ερευνητικούς σκοπούς, σε κάθε περίπτωση αναφέροντας τον τίτλο και το συγγραφέα και το ΠΑ.ΣΤ.ΕΛ. όπου εκπονήθηκε η ΠΕ καθώς και τον επιβλέποντα και την επιτροπή κρίσης.

ΠΡΟΛΟΓΟΣ

Η επιστήμη των υπολογιστών εισχωρεί στην καθημερινότητά μας όλο και πιο πολύ. Χρησιμοποιούμε την τεχνολογία άλλοτε σε μικρό και άλλοτε σε μεγάλο βαθμό. Η αγάπη μου για την επιστήμη των υπολογιστών με «ανάγκασε» να θέλω να σπουδάσω σε ένα τμήμα πληροφορικής. Από την άλλη, η έλλειψη λύσεων τεχνολογίας στον τομέα της υγείας με «έκανε» να επιλέξω το συγκεκριμένο τμήμα πληροφορικής.

Η στήριξη των δικών μου ανθρώπων αλλά και των συμφοιτητών μου που με την τριβή του χρόνου έγιναν και εκείνοι δικοί μου άνθρωποι μου ενδυνάμωσαν το ενδιαφέρον για την επιστήμη των υπολογιστών.

Η πρόκληση να δημιουργήσω ένα πρωτότυπο πρόγραμμα για μια εντελώς καινούργια πλατφόρμα λογισμικού, είναι αυτή που μου ενέτεινε το ενδιαφέρον και με οδήγησε σε πολύωρη και επίπονη ενασχόληση με το πολύ ενδιαφέρον θέμα που πραγματεύεται η παρούσα εργασία. Το αποτέλεσμα βέβαια θα το κρίνετε εσείς.

Κλείνοντας θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κύριο Κακαρούντα Αθανάσιο που μου εμπιστεύτηκε αυτή την εργασία αλλά και τους καθηγητές μου του ΠΑ.ΣΤ.ΕΛ. που με βοήθησαν να οικοδομήσω την απαραίτητη γνώση. Ξεχωριστές ευχαριστίες οφείλω και στους συμφοιτητές μου Βέρδη Ιωάννη, Μαραγκό Νικόλαο, Σιφνιάδη Ιωάννη που μου συμπαραστάθηκαν τις δύσκολες ώρες.

Αυτή η εργασία είναι αφιερωμένη στους γονείς μου, Ελένη και Γιώργο.

Εύχομαι καλή ανάγνωση σε όλους,
Τυχάλας Δημήτριος

ΠΕΡΙΛΗΨΗ

Στην παρούσα εργασία θα πραγματοποιούμε την ανάπτυξη λογισμικού για χρήση ως βάση δεδομένων ιατρικού φακέλου για όποιον διαθέτει συσκευή κινητού τηλεφώνου βασισμένο στην ανοικτή πλατφόρμα Android. Δεδομένου ότι κατά τη χρονική περίοδο υλοποίησης της παρούσης εργασίας, η Android είναι μια καινούργια πλατφόρμα, θα μελετηθούν τα πλεονεκτήματα και μειονεκτήματα της νέας πλατφόρμας. Στη συνέχεια θα εξετάσουμε τα εργαλεία που θα χρησιμοποιηθούν στην παρούσα διατριβή και θα αναπτυχθούν μία σειρά αναφορών και διαγραμμάτων στηριγμένα στη μεθοδολογία της ενοποιημένης προσέγγισης που θα μας οδηγήσουν στη σωστή και τεκμηριωμένη σχεδίαση του λογισμικού.

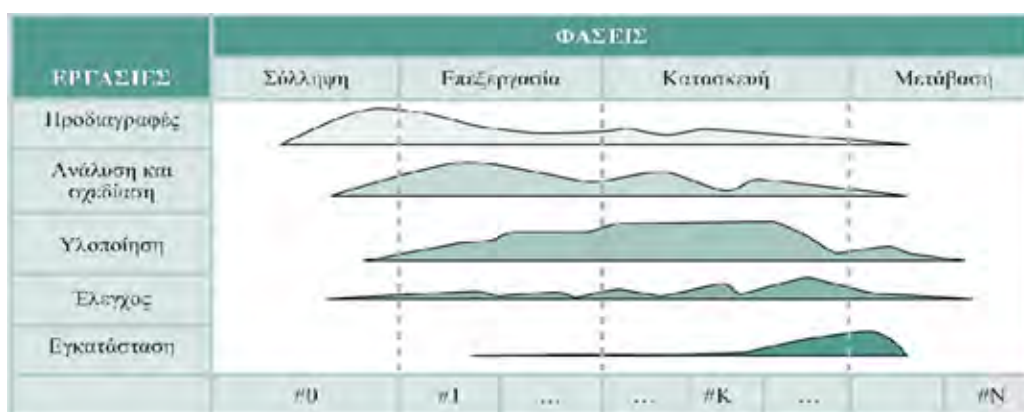
Στόχος της παρούσας διατριβής είναι να δημιουργηθεί λογισμικό που με την μορφή ψηφιακού ιατρικού φακέλου θα αποθηκεύει σε έναν server τα ιατρικά δεδομένα ενός ασθενή. Το λογισμικό θα έχει τη δυνατότητα να «κατεβάζει» τις εγγραφές του ασθενή, ύστερα από σχετικό ερώτημα που τίθεται στην βάση δεδομένων από τον ιατρό. Δευτερεύοντα στόχο αποτελεί η εκπόνηση μιας εργασίας καλής πρακτικής ανάπτυξης λογισμικού για την πλατφόρμα Android δεδομένου ότι δεν υπάρχει διαθέσιμο κάτι παρόμοιο στην Ελλάδα, ενώ υπάρχει αντίστοιχη έλλειψη και στο εξωτερικό.

Η παρούσα μελέτη απευθύνεται σε προπτυχιακούς και μεταπτυχιακούς φοιτητές, αλλά και σε επιστήμονες και τεχνολόγους της γνωστικής περιοχής της Τεχνολογίας Λογισμικού που ενδιαφέρονται να αντλήσουν πληροφοριακό υλικό, για ζητήματα της γνωστικής αυτής περιοχής, αλλά και εξειδικευμένες γνώσεις και δεξιότητες στο σχεδιασμό και ανάπτυξη λογισμικού για συσκευές κινητών τηλεφώνων, έχοντας ως στόχο μια ρεαλιστική εφαρμογή. Στις περιπτώσεις που είναι αναγκαία η παράθεση ορισμών και αποτελεσμάτων, αυτή γίνεται έτσι ώστε ο αναγνώστης να μην αποθαρρυνθεί αλλά να συνεχίσει την ανάγνωση της διατριβής χωρίς δυσκολία. Απαιτούνται ασφαλώς, κάποιες βασικές γνώσεις από την τεχνολογία λογισμικού των υπολογιστών.

ΜΕΘΟΔΟΛΟΓΙΑ ΥΛΟΠΟΙΗΣΗΣ

Στόχος μας είναι η ολοκλήρωση του έργου: «Σχεδιασμός και Ανάπτυξη Λογισμικού Διαχείρισης Ηλεκτρονικών Ιατρικών Φακέλων με τη χρήση της πλατφόρμας Android» που στο εξής θα αποκαλούμε: AMRS (Android's Medical Records Server).

Για την ολοκλήρωση του έργου θα ακολουθήσουμε την μεθοδολογία της ενοποιημένης προσέγγισης (unified process) ανάπτυξης λογισμικού. Ακριβώς για αυτό τον λόγο θα εξετάσουμε τέσσερις φάσεις ανάπτυξης όπως φαίνεται και από το παρακάτω σχήμα.



Σχήμα 1: Μια απεικόνιση της αναλογίας των εργασιών ανάπτυξης λογισμικού κατά της γενικές φάσεις της ενοποιημένης προσέγγισης.

Φάση σύλληψης: Σε αυτό το στάδιο θα εξετάσουμε την πλατφόρμα (Android) υλοποίησης του έργου και θα αναζητήσουμε στοιχεία για τον ανταγωνισμό που θα μας βοηθήσουν στην ανάλυση των προδιαγραφών του έργου τόσο σε ότι αφορά την πλατφόρμα, όσο και το λογισμικό (Ιατρικός Φάκελος) για ενσωματωμένα συστήματα. Θα εξετάσουμε τα εργαλεία που πρόκειται να χρησιμοποιηθούν για την ανάπτυξη του έργου με βάση τα οποία θα προδιαγράψουμε τις απαιτήσεις και θα οδηγηθούμε στη σχεδίαση.

Φάση Επεξεργασίας: Κατά τη φάση της επεξεργασίας θα γίνει η εκτίμηση των προδιαγραφών και απαιτήσεων του AMRS και θα δημιουργηθούν όλα τα σχεδιαγράμματα ανάλυσης και σχεδίασης του. Εκτιμάται πως θα στηριχθούμε στη μεθοδολογία UML αναπτύσσοντας όλα τα σχεδιαγράμματα που η εφαρμογή μας απαιτεί. Στο τέλος αυτής της φάσης θα έχουμε δημιουργήσει ένα σύνολο κειμένων και μοντέλων παράστασης λογισμικού.

Φάση Κατασκευής: Σε αυτό το στάδιο θα περάσουμε στην συγγραφή πηγαίου και εκτελέσιμου κώδικα με ολοένα και περισσότερα από τα απαιτούμενα λειτουργικά χαρακτηριστικά του λογισμικού. Στο τέλος αυτού του σταδίου θα έχουμε κατασκευάσει, εγκαταστήσει και δοκιμάσει στο εργαστήριο τον εκτελέσιμο κώδικα του AMRS σε αρχική έκδοση (Beta Release) και ουσιαστικά θα αποτελεί την έκδοση AMRS v0.8 η οποία θα πλησιάζει κατά πολύ την τελική έκδοση (AMRS v1.0) του έργου.

Φάση Μετάβασης: Σε αυτό το στάδιο θα γίνει η εγκατάσταση και ο έλεγχος του λογισμικού στο τελικό σύστημα ώστε να επαληθεύσουμε την ικανοποίηση των απαιτήσεων και κατά συνέπεια να οδηγηθούμε στην διόρθωση και στην τροποποίηση που θα οδηγήσει τελικά στην 1^η έκδοση του έργου (AMRS v1.0). Στο τέλος αυτού του σταδίου θα έχουμε κατασκευάσει την 1^η έκδοση (AMRS v1.0) του έργου.

Λέξεις-κλειδιά: Ανοικτή πλατφόρμα Android, Ψηφιακός Ιατρικός Φάκελος, AMRS.

Περιεχόμενο: Η παρούσα εργασία αποτελείται από το παρόν κείμενο, τα σχεδιαγράμματα υλοποίησης και τον κώδικα σε γλώσσα Java και Android.

ABSTRACT

In the present thesis we will analyze the development of software for use as a medical records database for anyone who has a mobile device based on the open platform Android. Given that by the time of composing this thesis the platform Android is a new software platform, existed corresponding platforms will be studied and will be analyzed the advantages and disadvantages of the new platform. Afterwards we will examine the tools that will be used in the present thesis and we will develop a line of reports and diagrams supported in the methodology of unified process that will lead us to the right and argued designing of software.

The aim of the present thesis is to create software that with the form of digital health records will store the patient's medical data on a server. The software will be able to download the patient's records upon such question to the database by the physician. Secondary goal is the creation of a thesis of software development for the Android platform given that there is nothing similar available in Greece, whether there is lack in the abroad too.

The present text is suitable as well for graduate and postgraduate students as for scientists of the cognitive region of Software Technology who are interested to get informative material for issues of this cognitive region and for specialized knowledge and dexterities in the drawing and software development for mobile devices, compared to a realistic application. Where is necessary to give definitions and results this happens so the reader is not discouraged to continue the reading of this thesis. Basic knowledge of Software Technology is required.

Περιεχόμενα

ΠΡΟΛΟΓΟΣ.....	3
ΠΕΡΙΛΗΨΗ	4
ΜΕΘΟΔΟΛΟΓΙΑ ΥΛΟΠΟΙΗΣΗΣ	5
ABSTRACT	6
ΠΙΝΑΚΑΣ ΣΧΗΜΑΤΩΝ	10
ΠΙΝΑΚΑΣ ΕΒΔΟΜΑΔΙΑΙΑΣ ΠΡΟΟΔΟΥ	11
ΚΕΦΑΛΑΙΟ 1	14
1.1 ΕΙΣΑΓΩΓΗ	14
1.2 ΦΟΡΗΤΕΣ ΣΥΣΚΕΥΕΣ.....	14
1.3 ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ ΚΙΝΗΤΩΝ ΤΗΛΕΦΩΝΩΝ	15
1.4 ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ ΓΙΑ ΚΙΝΗΤΑ ΤΗΛΕΦΩΝΑ	16
1.4.1 Symbian OS.....	16
1.4.2 UIQ.....	16
1.4.3 S60	16
1.4.4 Access Linux.....	16
1.4.5 Qtopia	17
1.4.6 JavaFX Mobile	17
1.4.7 Windows Mobile.....	17
1.4.8 BlackBerry OS	17
1.4.9 Άλλες πλατφόρμες	17
1.5 ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ ΣΤΗΝ ΑΓΟΡΑ.....	18
ΚΕΦΑΛΑΙΟ 2	19
2.1 ANDROID	19
2.1.1 Ιστορία.....	19
2.1.2 Τι είναι το Android.....	19
2.2 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ.....	21
2.3 ANDROID SDK	22
2.4 ΕΡΓΑΛΕΙΑ ΤΟΥ ANDROID SDK	23
2.4.1 Android Emulator	23
2.4.2 Hierarchy Viewer	23
2.4.3 Android Development Tools Plug-in for the Eclipse IDE	23
2.4.4 Android Debug Bridge	23
2.5 ΠΛΑΤΦΟΡΜΑ ΑΝΟΙΚΤΟΥ ΚΩΔΙΚΑ	24

2.6 ΕΥΡΟΣ ΕΦΑΡΜΟΓΩΝ	24
2.7 Η ΕΠΙΛΟΓΗ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ ANDROID	25
2.7.1 Πλεονεκτήματα	25
2.7.2 Μειονεκτήματα	25
ΚΕΦΑΛΑΙΟ 3	26
3.1 ΗΛΕΚΤΡΟΝΙΚΟΣ ΙΑΤΡΙΚΟΣ ΦΑΚΕΛΟΣ.....	26
3.1.1 Ιατρικός Φάκελος	26
3.1.2 Ηλεκτρονικός Ιατρικός Φάκελος	27
3.1.3 Υπάρχουσα κατάσταση στην Ελλάδα.....	27
3.2 ΠΡΟΔΙΑΓΡΑΦΕΣ ΗΛΕΚΤΡΟΝΙΚΟΥ ΙΑΤΡΙΚΟΥ ΦΑΚΕΛΟΥ	28
3.3 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΗΛΕΚΤΡΟΝΙΚΟΥ ΙΑΤΡΙΚΟΥ ΦΑΚΕΛΟΥ	28
3.3.1 Μείωση του κόστους υγειονομικής περίθαλψης	28
3.3.2 Βελτίωση της ποιότητας της περίθαλψης.....	28
3.4 ΜΕΙΟΝΕΚΤΗΜΑΤΑ ΤΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΙΑΤΡΙΚΩΝ ΦΑΚΕΛΩΝ	29
3.5 ΠΡΟΤΥΠΑ	29
3.6 ΜΑΚΡΟΠΡΟΘΕΣΜΗ ΔΙΑΤΗΡΗΣΗ ΚΑΙ ΦΥΛΑΞΗ ΑΡΧΕΙΩΝ.....	29
3.7 Ο ΣΥΓΧΡΟΝΙΣΜΟΣ ΤΩΝ ΑΡΧΕΙΩΝ	30
3.8 ΠΡΟΣΤΑΣΙΑ ΠΡΟΣΩΠΙΚΩΝ ΔΕΔΟΜΕΝΩΝ	30
3.8.1 Προστασία και ασφάλεια των Ηλεκτρονικών Εγγράφων Υγείας.....	31
ΚΕΦΑΛΑΙΟ 4	33
4.1 ANDROID: ON-LINE MEDICAL RECORDS, SERVER.....	33
4.2 ΠΡΟΔΙΑΓΡΑΦΗ ΤΩΝ ΑΠΑΙΤΗΣΕΩΝ ΑΠΟ ΤΟ ΛΟΓΙΣΜΙΚΟ ΣΥΣΤΗΜΑ	33
4.2.1 Λειτουργικές απαιτήσεις.....	33
4.2.2 Μη λειτουργικές απαιτήσεις.....	34
4.3 ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ	34
4.3.1 SQLite.....	34
4.3.2 Διαδίκτυο.....	34
4.4 ΕΡΓΑΛΕΙΑ ΥΛΟΠΟΙΗΣΗΣ	34
4.4.1 MySQL.....	35
4.4.2 PHP	35
4.4.3 Apache	35
4.4.4 Java	36
4.4.5 Eclipse	36
4.5 ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ ΕΡΓΑΣΙΩΝ	37

4.6 ΣΧΕΔΙΑΣΜΟΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ	37
ΚΕΦΑΛΑΙΟ 5	40
5.1 AMRS v1.0.....	40
5.2 ΕΓΚΑΤΑΣΤΑΣΗ ΤΩΝ ΑΠΑΡΑΙΤΗΤΩΝ ΠΡΟΓΡΑΜΜΑΤΩΝ & ΒΙΒΛΙΟΓΡΑΦΙΑ.....	40
5.2.1 Τεχνικά χαρακτηριστικά	40
5.2.2 Λειτουργικό σύστημα.....	41
5.2.4 Eclipse	43
5.2.5 MySQL Navigator	43
5.3 ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ	44
5.3.1 Πρώτα βήματα	44
5.3.2 Κύριος στόχος.....	44
5.3.3 Αντιμετώπιση προβλημάτων	46
5.3.4 Έλεγχος του προγράμματος.....	49
ΚΕΦΑΛΑΙΟ 6	51
6.1 ΑΝΑΛΥΣΗ ΠΕΡΙΠΤΩΣΕΩΝ ΧΡΗΣΗΣ	51
6.1.1 Πρώτη περίπτωση χρήσης.....	52
6.1.2 Δεύτερη περίπτωση χρήσης.....	53
6.1.3 Τρίτη περίπτωση χρήσης.....	53
ΚΕΦΑΛΑΙΟ 7	54
7.1 ΑΝΑΣΚΟΠΗΣΗ ΥΛΟΠΟΙΗΣΗΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΚΑΙ ΟΛΟΚΛΗΡΩΣΗ	54
7.2 ΠΡΟΒΛΗΜΑΤΑ ΣΤΗΝ ΣΥΓΓΡΑΦΗ ΤΟΥ ΚΩΔΙΚΑ.....	54
7.3 ΣΥΜΠΕΡΑΣΜΑΤΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ	55
7.4 ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΔΙΟΡΘΩΣΕΙΣ	56
7.5 ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ	56
7.6 ΜΕΛΛΟΝΤΙΚΗ ΧΡΗΣΗ ΤΗΣ ΠΑΡΟΥΣΑΣ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ	57
ΠΗΓΕΣ.....	58
ΒΙΒΛΙΟΓΡΑΦΙΑ	58
ΙΣΤΟΣΕΛΙΔΕΣ	58
ΚΩΔΙΚΑΣ.....	59

ΠΙΝΑΚΑΣ ΣΧΗΜΑΤΩΝ

Σχήμα 1: Μια απεικόνιση της αναλογίας των εργασιών ανάπτυξης λογισμικού κατά της γενικές φάσεις της ενοποιημένης προσέγγισης

Σχήμα 2: Υλοποίηση έργου σε εβδομαδιαία αναφορά

Σχήμα 3: Η σχέση ανάμεσα σε υλικό, λειτουργικό σύστημα, εφαρμογές και χρήστη

Σχήμα 4: Στατιστικά στοιχεία πωλήσεων έξυπνων συσκευών και των λειτουργικών του συστημάτων. Πηγή Canallys.

Σχήμα 5: Στατιστικά στοιχεία διείσδυσης κινητών τηλεφώνων στην αγορά και πρόσβασης στο διαδίκτυο. Πηγή eMarketer.

Σχήμα 6: Διάγραμμα Περιπτώσεων Χρήσης Πρώτου Επιπέδου

Σχήμα 7: Διάγραμμα Περιπτώσεων Χρήσης Δεύτερου Επιπέδου

ΠΙΝΑΚΑΣ ΕΒΔΟΜΑΔΙΑΙΑΣ ΠΡΟΟΔΟΥ

Α/Α ΕΒΔΟΜΑΔΩΝ ΜΕΛΕΤΗΣ	ΗΜΕΡΟΜΗΝΙΕΣ ΜΕΛΕΤΗΣ		ΚΕΦΑΛΑΙΑ	ΓΡΑΠΤΕΣ ΕΡΓΑΣΙΕΣ
1	29/12/09	~ 4/1/09	Σύλληψη της ΠΕ	
2	5/1/09	~ 11/1/09	Σύλληψη της ΠΕ	
3	12/1/09	~ 18/1/09	Σύλληψη της ΠΕ	
4	19/1/09	~ 25/1/09	Σύλληψη της ΠΕ	
5	26/1/09	~ 1/2/09	Σύλληψη της ΠΕ	
6	2/2/09	~ 8/2/09	Σύλληψη της ΠΕ	
7	9/2/09	~ 15/2/09	Σύλληψη της ΠΕ	
8	16/2/09	~ 22/2/09	Σύλληψη της ΠΕ	
9	23/2/09	~ 1/3/09	Σύλληψη της ΠΕ	
10	2/3/09	~ 8/3/09	Επεξεργασία και Σχεδίαση του AMRS	
11	9/3/09	~ 15/3/09	Επεξεργασία και Σχεδίαση του AMRS	
12	16/3/09	~ 29/3/09	Επεξεργασία και Σχεδίαση του AMRS	
13	30/3/09	~ 5/4/09	Επεξεργασία και Σχεδίαση του AMRS	
14	6/4/09	~ 12/4/09	Επεξεργασία και Σχεδίαση του AMRS	
15	13/4/09	~ 19/4/09	Επεξεργασία και Σχεδίαση του AMRS	
16	20/4/09	~ 26/4/09	Επεξεργασία και Σχεδίαση του AMRS	
17	27/4/09	~ 3/5/09	Επεξεργασία και Σχεδίαση του AMRS	
18	4/5/09	~ 10/5/09	Επεξεργασία και Σχεδίαση του AMRS	
19	11/5/09	~ 17/5/09	Επεξεργασία και Σχεδίαση του AMRS	
20	18/5/09	~ 24/5/09	Επεξεργασία και Σχεδίαση του AMRS	
21	25/5/09	~ 31/5/09	Κατασκευή και Ανάπτυξη του AMRS	
22	1/6/09	~ 7/6/09	Κατασκευή και Ανάπτυξη του AMRS	
23	8/6/09	~ 14/6/09	Κατασκευή και Ανάπτυξη του AMRS	
24	15/6/09	~ 21/6/09	Κατασκευή και Ανάπτυξη του AMRS	
25	22/6/09	~ 28/6/09	Κατασκευή και Ανάπτυξη του AMRS	
26	29/6/09	~ 5/7/09	Κατασκευή και Ανάπτυξη του AMRS	
27	6/7/09	~ 12/7/09	Κατασκευή και Ανάπτυξη του AMRS	
28	13/7/09	~ 19/7/09	Κατασκευή και Ανάπτυξη του AMRS	
29	20/7/09	~ 26/7/09	Κατασκευή και Ανάπτυξη του AMRS	
30	27/7/09	~ 2/8/09	Κατασκευή και Ανάπτυξη του AMRS	
31	3/8/09	~ 9/8/09	Κατασκευή και Ανάπτυξη του AMRS	

32	10/8/09	~	16/8/09	Κατασκευή και Ανάπτυξη του AMRS	
33	17/8/09	~	23/8/09	Κατασκευή και Ανάπτυξη του AMRS	
34	24/8/09	~	30/7/09	Κατασκευή και Ανάπτυξη του AMRS	
35	31/8/09	~	6/9/09	Κατασκευή και Ανάπτυξη του AMRS	
36	7/9/09	~	13/9/09	Κατασκευή και Ανάπτυξη του AMRS	
37	14/9/09	~	20/9/09	Κατασκευή και Ανάπτυξη του AMRS	
38	21/9/09	~	27/9/09	Κατασκευή και Ανάπτυξη του AMRS	
39	28/9/09	~	4/10/09	Κατασκευή και Ανάπτυξη του AMRS	
40	5/10/09	~	11/10/09	Κατασκευή και Ανάπτυξη του AMRS	
41	12/10/09	~	18/10/09	Κατασκευή και Ανάπτυξη του AMRS	
42	19/10/09	~	25/10/09	Κατασκευή και Ανάπτυξη του AMRS	
43	26/10/09	~	1/11/09	Κατασκευή και Ανάπτυξη του AMRS	
44	2/11/09	~	8/11/09	Κατασκευή και Ανάπτυξη του AMRS	
45	9/11/09	~	15/11/09	Κατασκευή και Ανάπτυξη του AMRS	
46	16/11/09	~	22/11/09	Κατασκευή και Ανάπτυξη του AMRS	Αποστολή έκδοσης AMRS 0.8
47	23/11/09	~	29/11/09	Εγκατάσταση κι έλεγχος	
48	30/11/09	~	6/12/09	Διόρθωση	Αποστολή έκδοσης AMRS 0.9
49	7/12/09	~	13/12/09	Εγκατάσταση κι έλεγχος	
50	14/12/09	~	20/12/09	Διόρθωση	Αποστολή έκδοσης AMRS 1.0
51	21/12/09	~	27/12/09	Προσθήκη σχολίων	
52	28/12/09	~	3/1/010	Εγγραφή του Word	Αποστολή προόδου εβδομάδας
53	4/1/10	~	10/1/10	Εγγραφή του Word	Αποστολή προόδου εβδομάδας
54	11/1/10	~	17/1/10	Εγγραφή του Word	Αποστολή προόδου εβδομάδας
55	18/1/10	~	24/1/10	Εγγραφή του Word	Αποστολή προόδου εβδομάδας
56	25/1/10	~	31/1/10	Εγγραφή του Word	Αποστολή προόδου εβδομάδας

57	1/2/10	~	7/2/10	Εγγραφή του Word	Αποστολή τελικού κειμένου
58	8/2/10	~	14/2/10	Σχεδίαση του PowerPoint	Αποστολή παρουσίασης
59	15/2/10	~	21/2/10	Ολοκλήρωση	
60	22/2/10	~	28/2/10	Ολοκλήρωση	Παρουσίαση πτυχιακής – 25/2/10

ΚΕΦΑΛΑΙΟ 1

1.1 ΕΙΣΑΓΩΓΗ

Βρισκόμαστε στον 21ο αιώνα και η κινητή τηλεφωνία αποτελεί αναπόσπαστο κομμάτι της ζωής μας. Η χρήση του κινητού τηλεφώνου είναι καθημερινή και πολυεπίπεδη, τηλεφωνικές κλήσεις, ανταλλαγή μηνυμάτων, λήψη φωτογραφιών και βίντεο, αναπαραγωγή τραγουδιών και φυσικά όλα αυτά μπορούμε να τα ανταλλάξουμε μεταξύ μας μέσω IrDA (υπέρυθρες) και Bluetooth. Τα πρώτα κινητά υποστήριζαν GPRS, δηλαδή μπορούσαμε να «σερφάρουμε» σε wap σελίδες και να ενημερωθούμε για κάποιες ειδήσεις, αλλά οι χρεώσεις ήταν αποτρεπτικές για τη χρήση των παραπάνω υπηρεσιών χωρίς να υπήρχε σοβαρή ανάγκη. Τα τελευταία χρόνια όμως, ειδικά με την νέα γενιά κινητών τηλεφώνων ή αλλιώς PDA μπορούμε και κάνουμε χρήση του Internet (όχι μόνο του wap) και μέσω WiFi. Αυτό σημαίνει πως οπουδήποτε υπάρχει κάποιο WiFi Hot Spot μπορούμε να συνδεθούμε από μια πολύ μικρή συσκευή και να την χρησιμοποιήσουμε όπως και τον υπολογιστή μας. Παρατηρούμε λοιπόν πως υπάρχει το τεχνικό υπόβαθρο για να αναπτυχθεί μια εφαρμογή όπου θα χρησιμοποιείται διαρκώς το WiFi του κινητού μας για την σύνδεση σε έναν Server. Τι γίνεται όμως με τον Ιατρικό Φάκελο; Λύσεις για ψηφιακούς ιατρικούς φακέλους έχουν προταθεί κατά δεκάδες τα τελευταία χρόνια από πολλά ιδρύματα ή εταιρίες αλλά καμιά τους δεν απευθυνόταν πραγματικά σε κάποιον ιατρό, γιατί όλες οι εφαρμογές ήταν δύσκολες στην χρήση, η εγκατάστασή τους έπρεπε να γίνει από κάποιον τεχνικό και τα λειτουργικά συστήματα τα οποία υποστήριζε δεν ήταν αυτά των PDAs, οπότε δεν υπήρχε φορητότητα, πράγμα πολύ σημαντικό για έναν ιατρό που πρέπει συνεχώς να κινείται μέσα σε ένα νοσοκομείο. Όπως είδαμε και παραπάνω οι φορητές συσκευές έχουν την δυνατότητα να υποστηρίξουν τις παραπάνω λειτουργίες που χρειάζεται ένας ιατρικός φάκελος, οπότε σε αυτή την πτυχιακή εργασία θα «μεταφέρουμε» τις υπάρχουσες λύσεις ψηφιακών ιατρικών φακέλων σε μια φορητή πλατφόρμα η οποία θα συνδέεται μέσω WiFi σε έναν Server για να αντλεί τις κατάλληλες εγγραφές του ιατρικού φακέλου.

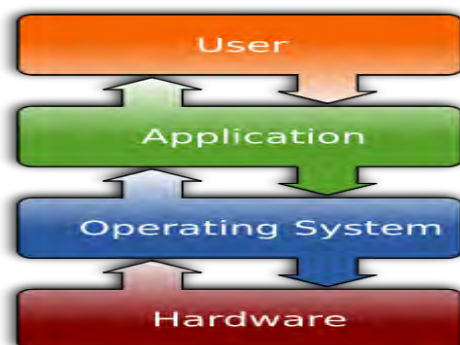
1.2 ΦΟΡΗΤΕΣ ΣΥΣΚΕΥΕΣ

Αναφέρθηκε παραπάνω ότι η εφαρμογή θα χρησιμοποιηθεί σε φορητές συσκευές, οπότε τώρα πρέπει να αποσαφηνιστεί και παράλληλα να στοχεύσει σε μια πιο συγκεκριμένη κατηγορία αυτών των συσκευών.

Με τον όρο φορητές συσκευές εννοούνται όλες οι ηλεκτρονικές συσκευές σε μέγεθος 'τσέπης', δηλαδή αρκετά μικρές ώστε να τις κρατάμε με ένα χέρι και να τις χειριζόμαστε με το άλλο με σχετική ευκολία. Τέτοιου είδους ηλεκτρονικές συσκευές είναι τα έξυπνα τηλέφωνα (smart phones), mp3 players, PDA, smart card readers, iPods και άλλα.

Όλες οι φορητές συσκευές αποτελούνται από τα εξής 3 μέρη: α) το υλικό (hardware) που περιλαμβάνει το μέγεθος της μνήμης, τον πυρήνα και άλλα τεχνικά χαρακτηριστικά, β) το λειτουργικό σύστημα (Operating System ή OS) που ορίζει τον τρόπο λειτουργίας του υλικού της συσκευής και γ) τις εφαρμογές (applications) οι οποίες είναι τα τελικά προγράμματα που έρχεται σε επαφή ο χρήστης. Το υλικό είναι ο πρώτος και κύριος παράγοντας μιας συσκευής γιατί ουσιαστικά αποτελεί τα τεχνικά της χαρακτηριστικά και τις δυνατότητες όπου αυτή θα έχει. Από αυτό καθορίζεται κατά μεγάλο μέρος και η τελική τιμή της συσκευής. Το λειτουργικό σύστημα είναι ο τρόπος με τον οποίο πετυχαίνει το υλικό να επικοινωνεί μεταξύ του και πως θα αξιοποιηθεί όσο το δυνατόν καλύτερα ώστε να δουλεύει γρήγορα και αποτελεσματικά. Παρατηρείται λοιπόν ότι το λειτουργικό σύστημα συνδέεται άμεσα με το υλικό, οπότε δεν γίνεται να δημιουργηθεί ένα λειτουργικό σύστημα χωρίς να είναι γνωστό το υλικό για το οποίο προορίζεται, αλλά και το αντίστροφο, δεν γίνεται να υλοποιηθεί μια συσκευή (το υλικό μέρος) χωρίς να έχουν αποφασιστεί πρώτα οι δυνατότητες τις οποίες θα έχει το λειτουργικό της σύστημα. Οπότε κάθε λειτουργικό σύστημα φτιάχνεται για ένα συγκεκριμένο συνδυασμό υλικού το οποίο μπορεί να χρησιμοποιηθεί με κάποιες αλλαγές σε

έναν παραπλήσιο. Τέλος οι εφαρμογές αποτελούν το τρίτο και τελευταίο μέρος και συντελούν πολύ σημαντικό ρόλο, γιατί αυτές είναι όπου ο χρήστης βλέπει. Οι εφαρμογές επικοινωνούν με το λειτουργικό σύστημα και αυτό με την σειρά του επικοινωνεί με το υλικό και εκτελεί τις αντίστοιχες εντολές. Όλα αυτά όμως ο χρήστης δεν τα βλέπει, το μόνο που είναι ορατό σε αυτόν είναι το μενού της εφαρμογής και για αυτό μια εφαρμογή πρέπει να είναι εύχρηστη και απλή, με κάθε επιλογή κατανοητή από το κοινό στο οποίο απευθύνεται. Οι παραπάνω σχέσεις μεταξύ χρήστη και των τριών μερών μια φορητής συσκευής φαίνονται στο παρακάτω σχήμα:



Σχήμα 2: Η σχέση ανάμεσα σε υλικό, λειτουργικό σύστημα, εφαρμογές και χρήστη.

Στην παρούσα διατριβή στόχος είναι η δημιουργία μιας εφαρμογής για έξυπνα τηλέφωνα (smart phones) οπότε θα αναπτυχθεί μόνο αυτή η κατηγορία φορητών συσκευών. Εναλλακτικά οι παραπάνω φορητές συσκευές ονομάζονται συσκευές ενσωματωμένων συστημάτων αφού (όπως φαίνεται και στο σχήμα 2) αποτελούνται πέρα από το υλικό της συσκευής και από λειτουργικό σύστημα και εφαρμογές όπως αναφέρθηκε νωρίτερα.

1.3 ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ ΚΙΝΗΤΩΝ ΤΗΛΕΦΩΝΩΝ

Το λειτουργικό σύστημα, όπως φαίνεται και από το σχήμα 2, αποτελεί το ενδιάμεσο στάδιο μεταξύ εφαρμογής και υλικού. Χωρίς αυτό δεν θα μπορούσε να επικοινωνήσει μια εφαρμογή με το υλικό της συσκευής και το αντίστροφο. Επιπλέον, παραπάνω αναφέρθηκε πως το λειτουργικό σύστημα αξιοποιεί τις δυνατότητες της συσκευής, οπότε ανάλογα με την εφαρμογή που πρόκειται να αναπτυχθεί, πρέπει να υπάρχει και το αντίστοιχο υλικό, άρα και το ανάλογο λειτουργικό σύστημα ώστε να συνδέσει τα 2 προηγούμενα. Πέρα από αυτό το λειτουργικό σύστημα μοιράζει τους πόρους του υλικού αλλά και τους ελέγχει ώστε να επιτρέπει την καλύτερη αξιοποίησή του από τις εφαρμογές. Παρέχει δυνατότητες στην συσκευή όπως αυτή της ταυτόχρονης χρήσης περισσότερων εφαρμογών (multitasking) και ακόμα καθορίζει τη χρήση των οπτικών και αποθηκευτικών μέσων, των εξωτερικών συσκευών και τη χρήση του διαδικτύου μέσω συγκεκριμένων πρωτοκόλλων. Τέλος κάθε λειτουργικό σύστημα έχει και το δικό της περιβάλλον διεπαφής με το χρήστη και έτσι δίνεται η δυνατότητα στον τελευταίο να διαχειρίζεται εύκολα τις εφαρμογές του και να ρυθμίζει τις διάφορες παραμέτρους της συσκευής. Καθίσταται λοιπόν φανερό ότι οι προδιαγραφές του λειτουργικού συστήματος, που θα επιλεγθούν, πρέπει να καλύπτουν πλήρως τις προϋποθέσεις για την άριστη λειτουργία της εφαρμογής.

1.4 ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ ΓΙΑ ΚΙΝΗΤΑ ΤΗΛΕΦΩΝΑ

Υπάρχουν πάρα πολλές φορητές συσκευές διαθέσιμες για τους καταναλωτές, αλλά αυτή που χρησιμοποιείται από το σύνολο σχεδόν του πληθυσμού, είναι χωρίς αμφιβολία το κινητό τηλέφωνο. Ήταν αναμενόμενο λοιπόν, τα περισσότερα λειτουργικά συστήματα να εξυπηρετούν αυτήν την κατηγορία. Παρακάτω υπάρχει μια λίστα με τα πιο γνωστά λειτουργικά συστήματα για κινητά τηλέφωνα και μια ανάλυση αυτών μέχρι το σημείο που αφορά την εργασίας μας.

1.4.1 Symbian OS

Η πλατφόρμα Symbian OS [11] αποτελεί την εξέλιξη της EPOC (προήρθε από τη λέξη epoch 'εποχή') της εταιρείας Psion και είναι σχεδιασμένη για φορητές συσκευές. Στόχος της γραμμένης σε γλώσσα C++ πλατφόρμας και των αντικειμενοστραφών εφαρμογών που δημιουργήθηκαν για αυτή την πλατφόρμα ήταν να δημιουργηθούν διαδραστικές εφαρμογές από τις μέχρι τότε ακολουθιακές. Η πρώτη φορητή συσκευή που χρησιμοποίησε περιβάλλον Symbian κυκλοφόρησε το 2000. Η ταχύτερη εξέλιξη στα φορητά τηλέφωνα και τις πλατφόρμες που τα πλαισιώνουν, βοήθησε στη μεγιστοποίηση της βελτίωσης της πλατφόρμας Symbian με την δημιουργία μιας πληθώρας νέων εκδόσεων που κάλυπταν και καλύπτουν ένα μεγάλο αριθμό διαφορετικών συσκευών. Αποτέλεσε δε την βάση για άλλες πλατφόρμες που προσπάθησαν να εισέρθουν στην αγορά διαφοροποιώντας τον τρόπο λειτουργίας της πλατφόρμας αλλά και τον τρόπο ανάπτυξης εφαρμογών για την πλατφόρμα. Η Symbian αρχικά δεν ήταν ανοικτού κώδικα (θα αναλύσουμε τον όρο στη συνέχεια) προοδευτικά όμως σχεδιάζει την μετάβαση της πλατφόρμας προς αυτόν τον δρόμο.

1.4.2 UIQ

Η πλατφόρμα UIQ [12] (προέρχεται από τις λέξεις User Interface Quartz) βασίζεται στο περιβάλλον Symbian OS. Στην ουσία αποτελεί ένα ενδιάμεσο γραφικό περιβάλλον που παρέχει κάποια πρόσθετα συστατικά στον πυρήνα OS, με στόχο να δέχεται εφαρμογές γραμμένες σε διάφορες γλώσσες προγραμματισμού και να επιτρέψει την ανάπτυξη των κινητών τηλεφώνων μέσω των εφαρμογών τρίτων. Η κεντρική ιδέα ανάπτυξης της πλατφόρμας ήταν να μπορεί ο χρήστης να χρησιμοποιεί μεγαλύτερο εύρος εφαρμογών μέσω της φορητότητας που εφαρμόζει η πλατφόρμα βοηθώντας τον χρήστη να απλοποιήσει την χρήση του κινητού τηλεφώνου μέσα από ένα απλό και πολύ φιλικό γραφικό περιβάλλον. Η πλατφόρμα UIQ παρέχει ένα περιβάλλον ανάπτυξης λογισμικού για δημιουργία εφαρμογών το 'UIQ development platform' που προσκαλεί κατασκευαστές λογισμικού να το χρησιμοποιήσουν και να αναπτύξουν εφαρμογές για μια πληθώρα διαφορετικών συσκευών που χρησιμοποιούν την πλατφόρμα UIQ.

1.4.3 S60

Η πλατφόρμα S60 [13] στηρίζεται στο περιβάλλον Symbian και υποστηρίζει εφαρμογές γραμμένες σε διάφορες γλώσσες προγραμματισμού (native C++, Java και άλλες). Η πλατφόρμα υποστηρίζει τα ψηφιακά δικαιώματα (DRM). Δημιουργήθηκε από την εταιρεία κατασκευής κινητών τηλεφώνων NOKIA και αποτελεί μια πλατφόρμα κλειστού κώδικα. Περιλαμβάνει γραφικό περιβάλλον και όλες τις εφαρμογές που απαιτούνται για την χρήση ενός κινητού τηλεφώνου, από εφαρμογή διαχείρισης κλήσεων μέχρι εφαρμογές γραφείου, διαδικτύου και παιχνιδιών. Οι εφαρμογές εγκαθίστανται στην συσκευή με την εγκατάσταση της πλατφόρμας και είναι ενσωματωμένες σε αυτή ενώ δεν παρέχει μεγάλη δυνατότητα πρόσθεσης εφαρμογών ή τροποποίησης των υπαρχουσών.

1.4.4 Access Linux

Η πλατφόρμα Access Linux [17] είναι η εξέλιξη της πλατφόρμας Palm OS, εμφανίστηκε επίσημα στο κοινό τον Φεβρουάριο του 2007 και βασίζεται σε περιβάλλον Linux. Ο κεντρικός στόχος της πλατφόρμας είναι να κάνει τις εφαρμογές όσο πιο εύκολες σε χρήση γίνεται. Η πλατφόρμα είναι ιδιαίτερα ευέλικτη ώστε να μπορεί να χρησιμοποιηθεί σε διαφορετικές φορητές συσκευές. Οι εφαρμογές μπορεί να είναι γραμμένες είτε σε C ή C++

όπως συμβαίνει με τις εφαρμογές για Palm OS είτε σε Java. Και η πλατφόρμα Access Linux είναι ανοικτού κώδικα και παρέχει τα απαραίτητα Plug Ins για προγραμματισμό στο ολοκληρωμένο περιβάλλον ανάπτυξης λογισμικού Eclipse. Η συγκεκριμένη πλατφόρμα έχει τα περισσότερα κοινά χαρακτηριστικά με την πλατφόρμα Android ενώ διαφοροποιείται στο τρόπο διαχείρισης του υλικού της συσκευής.

1.4.5 Qtopia

Η πλατφόρμα Qtopia [14] δημιούργημα της εταιρείας Trolltech βασίζεται σε περιβάλλον Linux, είναι προγραμματισμένη σε γλώσσα C++ και παρέχει πλήρες παραθυρικό σύστημα. Αυτό που προβάλλεται ως πλεονέκτημα είναι η δυνατότητα χρησιμοποίησης της πλατφόρμας σε έκδοση για ηλεκτρονικό υπολογιστή με δυνατότητα συγχρονισμού των εφαρμογών γραφείου όπως του τηλεφωνικού καταλόγου, σημειώσεων κ.α. Η πλατφόρμα εγκαθίσταται ολοκληρωμένη, περιλαμβάνοντας από προγράμματα διαχείρισης λογισμικού και τις κύριες εφαρμογές για τη διαχείριση της συσκευής μέχρι και μη κύριες εφαρμογές όπως παιχνίδια και εφαρμογές γραφείου.

1.4.6 JavaFX Mobile

Η πλατφόρμα JavaFX Mobile [15] είναι βασισμένη σε πυρήνα Linux και σε γλώσσα προγραμματισμού Java. Αποτελεί επέκταση της πλατφόρμας JavaMe στις φορητές συσκευές και έχει ως στόχο την δημιουργία πλούσιων διαδικτυακών εφαρμογών.

1.4.7 Windows Mobile

Αυτή η πλατφόρμα μεταφέρει την εμπειρία των Windows [16] στους χρήστες φορητών συσκευών και έχει στηριχθεί στην εφαρμογή [Microsoft Win32](#). Δεν είναι πλατφόρμα ανοικτού κώδικα και εγκαθίσταται από την αρχή με μια σειρά εφαρμογών που έχουν προεγκριθεί και συνοδεύουν την πλατφόρμα. Η ιδέα της χρήσης ενός γνωστού περιβάλλοντος (windows) σε φορητές συσκευές και η προσπάθεια μεταφοράς των γνωστών εφαρμογών της εταιρείας σε φορητές συσκευές είναι ο κύριος στόχος της συγκεκριμένης πλατφόρμας.

1.4.8 BlackBerry OS

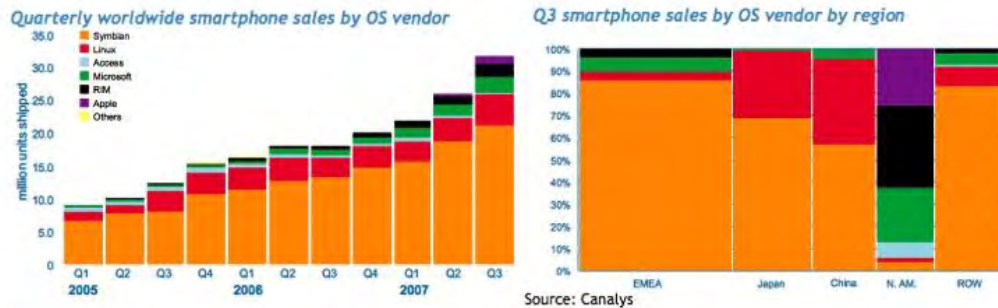
Η πλατφόρμα BlackBerry OS κατασκευασμένη από την εταιρεία RIM ([Research In Motion](#)) έδωσε πρώτη την δυνατότητα αποστολής και λήψης email σε κινητά τηλέφωνα. Υποστηρίζει multi-tasking και είναι κλειστού λογισμικού. Προσφέρει πλήρη χρήση ασύρματης επικοινωνίας μέσω WiFi δικτύων καθώς και 3G δικτύου για τον άμεσο συγχρονισμό εφαρμογών αλλά όπως είπαμε και για την λήψη ηλεκτρονικών μηνυμάτων. Κάθε εφαρμογή που δημιουργείται από εξωτερικούς, ανεξάρτητους της εταιρείας κατασκευαστές λογισμικού πρέπει να περάσει από αδειοδότηση από την εταιρεία RIM πριν την χρήση του στις συσκευές που υποστηρίζει η πλατφόρμα BlackBerry OS.

1.4.9 Άλλες πλατφόρμες

Βέβαια η λίστα με τις πλατφόρμες δεν σταματάει εδώ αλλά μεγαλώνει με την προσθήκη από πλατφόρμες όπως η iPhone OS, OpenMoko, Helix, Palm OS και άλλων.

1.5 ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ ΣΤΗΝ ΑΓΟΡΑ

Όπως προαναφέρθηκε υπάρχει μια πληθώρα λειτουργικών συστημάτων στην αγορά. Για να αποφασιστεί ποια πρέπει να χρησιμοποιηθεί είναι απαραίτητο να καλύπτει τις ανάγκες μας, αλλά και να έχει και απήχηση στο κοινό. Είναι σημαντικό το λειτουργικό σύστημα που θα επιλεγεί να χρησιμοποιείται ήδη από τους καταναλωτές γιατί αυτό σημαίνει πως δεν θα χρειαστεί οι χρήστες να αλλάξουν λειτουργικό σύστημα και προφανώς και συσκευή, απλά θα χρειαστεί να εγκαταστήσουν την εφαρμογή στο λειτουργικό το οποίο ήδη έχουν. Σύμφωνα με τα στατιστικά στοιχεία που ανακοινώνει η εταιρεία Canallys η πλατφόρμα Symbian κατακλύζει συντριπτικά την παγκόσμια αγορά με ποσοστό που ξεπερνάει το 50% ενώ ακολουθούν οι πλατφόρμες Microsoft και RIM με ποσοστά 12% και 11% αντίστοιχα.



Σχήμα 3: Στατιστικά στοιχεία πωλήσεων έξυπνων συσκευών και των λειτουργικών του συστημάτων. Πηγή Canallys.

Παράλληλα η χρήση του Διαδικτύου μέσω των φορητών συσκευών αυξάνεται συνεχώς και σύμφωνα με την εταιρεία eMarketer τα κινητά τηλέφωνα θα είναι η κυριότερη οθόνη για πρόσβαση στο Διαδίκτυο σύμφωνα με τις τάσεις ανερχόμενων αγορών όπως είναι αυτές των χωρών Βραζιλίας, Ρωσίας, Ινδίας και Κίνας που αθροιστικά αποτελούν το 40% του πληθυσμού του πλανήτη.

Mobile Usage Metrics for BRIC, 2008 (millions and % penetration)				
	Brazil	Russia	India	China
Population (1)	191.9	140.7	1,148.0	1,330.0
Mobile phone subscribers*	140.0	174.0	290.0	595.0
Mobile phone penetration	73.0%	123.7%	25.3%	44.7%
Mobile Internet users**	7.1	3.6	106.0	85.0
Mobile Internet users % of mobile subscribers	5.1%	2.1%	36.6%	14.3%

Note: *active SIM cards; **accounts that use a browser (WAP or XHTML) to access the Internet via mobile phone; SMS/MMS/IM not included
Source: eMarketer, June 2008; (1) US Census Bureau, July 2007

095343 www.eMarketer.com

Σχήμα 4: Στατιστικά στοιχεία διείσδυσης κινητών τηλεφώνων στην αγορά και πρόσβασης στο διαδίκτυο. Πηγή eMarketer.

Σύμφωνα λοιπόν με τα παραπάνω η ενασχόλησή μας με την ανάπτυξη λογισμικού για χρήση σε φορητές συσκευές (κινητά τηλέφωνα) αποτελεί ιδέα με ευοίωνες προοπτικές, αφού απευθύνεται σε έναν συνεχώς και ραγδαία αναπτυσσόμενο χώρο με έλλειψη από αντίστοιχες εφαρμογές. Γίνεται επίσης κατανοητό πως εφαρμογές με μεγάλη απήχηση θα είναι αυτές που χρησιμοποιούν το διαδίκτυο αλλά και κάθε νέα και μοντέρνα τεχνολογία.

ΚΕΦΑΛΑΙΟ 2

2.1 ANDROID

Παραπάνω αναλύθηκαν πολλά λειτουργικά συστήματα που υπάρχουν ήδη στην αγορά, όμως δεν αναφέρθηκε καθόλου το λειτουργικό σύστημα της Google, Android [20]. Ένα νέο και πολλά υποσχόμενο λειτουργικό σύστημα για κινητά τηλέφωνα. Παρακάτω θα αναλυθούν όλες οι πληροφορίες σχετικά με το Android αλλά και γιατί καθίσταται το ιδανικό περιβάλλον για την ανάπτυξη της εφαρμογής αυτής της πτυχιακής εργασίας.

2.1.1 Ιστορία

Τον Ιούλιο του 2005 η εταιρεία Google ανακοινώνει την απόκτηση της προσφάτως εμφανισμένης εταιρείας Android inc. η οποία ιδρύθηκε από τους Andy Rubin, Rich Miner, Nick Sears και Chris White. Την στιγμή εκείνη η εταιρεία είχε ως στόχο την δημιουργία λογισμικού και εφαρμογών για κινητά τηλέφωνα. Από την στιγμή της ενσωμάτωσης στην Google η ομάδα της Android υπό την καθοδήγηση του Andy Rubin δημιούργησε ένα λειτουργικό σύστημα που υποσχόταν ευελιξία για τους κατασκευαστές και τους παρόχους κινητής τηλεφωνίας. Το Android βασίζεται στον πυρήνα του λειτουργικού συστήματος Linux. 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 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού Open Handset Alliance [22], 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. Μερικές από αυτές είναι οι Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, Sprint Nextel και NVIDIA η οποία παρουσιάστηκε έχοντας ως στόχο να δημιουργήσει ανοιχτά πρότυπα για κινητές συσκευές. Η Google δημοσίευσε το μεγαλύτερο μέρος του κώδικα του Android υπό τους όρους της Apache License, μιας ελεύθερης άδειας λογισμικού.

2.1.2 Τι είναι το Android

Το Android είναι μια πλατφόρμα ανοικτού λογισμικού που περιλαμβάνει λειτουργικό σύστημα, ενδιάμεσο λογισμικό και βασικές εφαρμογές για χρήση σε κινητές συσκευές βασισμένες στον πυρήνα Linux 2.6.

Σύμφωνα με τις αρχές πλατφόρμας λογισμικού, σχεδιαστικά, στην βάση υπάρχουν οι οδηγοί (drivers) του υλικού της συσκευής. Ένα επίπεδο επάνω βρίσκονται οι βιβλιοθήκες της πλατφόρμας καθώς και η απαραίτητη εικονική μηχανή για την μετατροπή και τον χρόνο εκτέλεσης των εφαρμογών. Αυτά παισιώνονται από βασικές εφαρμογές που είναι απαραίτητες για τη βασική διαχείριση της συσκευής και για την αρχική επικοινωνία και χρήση της συσκευής από τον χρήστη. Στην κορυφή βρίσκονται οι εφαρμογές που αναπτύσσονται ελεύθερα από κατασκευαστές λογισμικού ή ακόμα και από τους ίδιους τους χρήστες και αποτελούν τα προγράμματα-εφαρμογές που κάθε χρήστης θα επιλέξει να εντάξει για χρήση στο κινητό του τηλέφωνο. Η αρχιτεκτονική της πλατφόρμας εμφανίζεται στο παρακάτω σχήμα.



Σχήμα 5: Η αρχιτεκτονική της πλατφόρμας Android.

Η δημιουργία της πλατφόρμας Android έγινε με λογική σχεδίασης από κάτω προς τα πάνω (bottom up) ώστε να προσφέρει συμβατότητα με πολλές συσκευές και να παρέχει το μέγιστο των δυνατοτήτων τους στις εφαρμογές που θα δημιουργηθούν για την πλατφόρμα.

Η ανάπτυξη των εφαρμογών για την πλατφόρμα Android γίνεται μέσω του εργαλείου Android SDK και χρησιμοποιείται η γλώσσα προγραμματισμού Java. Επιπλέον υποστηρίζεται η χρήση πολυμέσων και διαδικτύου. Οι εφαρμογές τρέχουν μέσω του Dalvik virtual machine [21]. Η εικονική μηχανή Dalvik μετατρέπει τα γραμμένα σε γλώσσα προγραμματισμού Java προγράμματα σε μορφή Dalvic Executable (DEX) τα οποία είναι βελτιστοποιημένα για χαμηλές απαιτήσεις μνήμης. Η Dalvic VM επιτρέπει πολλαπλές ταυτόχρονες εμφανίσεις της εικονικής μηχανής και υποστηρίζει καλύτερα τις εφαρμογές στην χρήση τους σε φορητές συσκευές.

Η πλατφόρμα Android αναπτύχθηκε για να φέρει την επανάσταση στις φορητές συσκευές προωθώντας την έννοια του ελεύθερου λογισμικού και της ελεύθερης διακίνησης προγραμμάτων (open source code).

Στόχος της πλατφόρμας είναι να επιτρέψει στους προγραμματιστές να δημιουργήσουν εφαρμογές που θα χρησιμοποιούν πλήρως τις δυνατότητες των φορητών συσκευών και του διαδικτύου. Δύναται να χρησιμοποιηθούν οι βασικές λειτουργίες ενός τηλεφώνου (κλήσεις, γραπτά μηνύματα κτλ) για την δημιουργία νέων πρωτοποριακών εφαρμογών. Ο χρήστης της φορητής συσκευής θα έχει πλέον την δυνατότητα να μετατρέψει τη συσκευή του σε εργαλείο προσωπικής έκφρασης κάνοντας χρήση εφαρμογών της απόλυτης επιλογής του. Το διαδίκτυο γίνεται φορέας ανάπτυξης της φορητής συσκευής που πλέον σχεδιάζεται για να έχει άμεση σχέση με αυτό καθώς πολλές εφαρμογές μπορούν να εναρμονιστούν και να χρησιμοποιήσουν άμεσα το διαδίκτυο.

Το μεγάλο πλεονέκτημα της πλατφόρμας Android είναι η στήριξη στην κοινότητα ανοικτού κώδικα γεγονός που επιτρέπει σε περισσότερες εταιρείες να χρησιμοποιήσουν την πλατφόρμα χωρίς κόστος και αυτό αναμένεται να προξενήσει έντονο ενδιαφέρον για την πλατφόρμα αφενός και αφετέρου θα προσελκύσει ένα μεγάλο μέρος κατασκευαστών λογισμικού για την

δημιουργία νέων και πρωτοποριακών εφαρμογών που θα απευθύνονται σε ευρύ φάσμα συσκευών και κατ' επέκταση σε μεγάλο καταναλωτικό κοινό.

2.2 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ

Σε αυτή τη ενότητα θα αναφερθούν τα χαρακτηριστικά που υποστηρίζει η πλατφόρμα Android και την καθιστούν την ιδανική για την επίτευξη του σκοπού μας.

Λειτουργίες Οθόνης	Η πλατφόρμα είναι προσαρμόσιμη σε μεγαλύτερη VGA ανάλυση, δισδιάστατες ψηφιακές γραφικές βιβλιοθήκες, τρισδιάστατα γραφικά βασισμένα στην OpenGL ES 1.0 έκδοση χαρακτηριστικών, καθώς και παραδοσιακές απεικονίσεις οθόνης "έξυπνων" συσκευών κινητής τηλεφωνίας.
Αποθήκευση Δεδομένων	Χρήση βάσης δεδομένων SQLite για τις ανάγκες αποθήκευσης.
Συνδεσιμότητα	Το Android υποστηρίζει τεχνολογίες συνδεσιμότητας συμπεριλαμβανομένου GSM/EDGE, CDMA, EV-DO, UMTS, Bluetooth, και WiFi.
Αποστολή μηνυμάτων	SMS και MMS είναι οι διαθέσιμοι τρόποι ανταλλαγής μηνυμάτων.
Περιήγηση στον Ιστό	Για την περιήγηση στον ιστό το Android διαθέτει ένα φυλλομετρητή βασισμένο στην ανοιχτή τεχνολογία WebKit.
Υποστήριξη Java	Λογισμικό γραμμένο στην Java είναι δυνατόν να μεταγλωττιστεί και να εκτελεστεί στην εικονική μηχανή Dalvik , η οποία είναι μια εξειδικευμένη υλοποίηση εικονική μηχανής, σχεδιασμένη για χρήση σε φορητές συσκευές, παρόλο που δεν είναι μια πρότυπη εικονική μηχανή Java.
Υποστήριξη Πολυμέσων	Το λειτουργικό Android υποστηρίζει τις ακόλουθα μορφές ήχου, στατικής και κινούμενης εικόνας: H.263, H.264 (σε 3GP ή MP4 container), MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC, MP3, MIDI, OGG Vorbis, WAV, JPEG, PNG, GIF, BMP.
Επιπλέον υποστήριξη hardware	Το λειτουργικό Android μπορεί να συνεργαστεί με κάμερες στατικής ή κινούμενης εικόνας, οθόνες αφής, GPS, αισθητήρες επιτάχυνσης, μαγνητόμετρα, δισδιάστατους καθώς και τρισδιάστατους επιταχυντές γραφικών.
Περιβάλλον Ανάπτυξης Λογισμικού	Περιλαμβάνει ένας προσομοιωτή συσκευής, εργαλεία για διόρθωση σφαλμάτων, μνήμη και εργαλεία ανάλυσης της απόδοσης του εκτελέσιμου λογισμικού καθώς και ένα επιπρόσθετο για το Eclipse IDE.
Αγορά και Εγκατάσταση Εφαρμογών	Παρόμοια με το App Store του iPhone OS, το Android Market είναι ένας κατάλογος εφαρμογών που μπορούν να μεταφορτωθούν και εγκατασταθούν στην συσκευή άμεσα μέσω ασύρματων καναλιών, χωρίς την χρήση υπολογιστή. Αρχικά μόνο δωρεάν εφαρμογές ήταν δυνατόν να εγκατασταθούν. Εφαρμογές επί πληρωμή ήταν μετέπειτα διαθέσιμες στο Android Market στις ΗΠΑ ύστερα από τις 19 Φεβρουαρίου 2009.
Οθόνη Αφής Πολλαπλών Σημείων	Το λειτουργικό Android είχε εξ ορισμού υποστήριξη για οθόνες πολλαπλών σημείων αλλά η δυνατότητα αυτή έχει κλειδωθεί σε επίπεδο πυρήνα (πιθανόν για αποφυγή παραβιάσεων των πατεντών λογισμικού της Apple στις τεχνολογίες οθονών αφής).

2.3 ANDROID SDK

Το Android SDK περιλαμβάνει ένα ολοκληρωμένο σύνολο εργαλείων ανάπτυξης. Αυτά περιλαμβάνουν ένα πρόγραμμα εντοπισμού σφαλμάτων, βιβλιοθήκες, έναν εξομοιωτή (με βάση QEMU), τεκμηρίωση, δείγματα κώδικα, και tutorials. Μπορεί να εγκατασταθεί σε υπολογιστές με x86-αρχιτεκτονική με λειτουργικό σύστημα α) Linux (οποιαδήποτε σύγχρονη διανομή Linux Desktop), β) Mac OS X 10.4.8 ή αργότερα, γ) τα Windows XP, Vista και 7. Απαιτείται η ύπαρξη της Java Development Kit, Apache Ant, και Python 2.2 ή νεότερη έκδοση. Το επίσημο υποστηριζόμενο περιβάλλον ανάπτυξης (IDE) είναι το Eclipse (έκδοση 3.2 ή μεταγενέστερη), χρησιμοποιώντας το Android Development Tools (ADT), αν και οι προγραμματιστές μπορούν να χρησιμοποιούν οποιοδήποτε πρόγραμμα επεξεργασίας κειμένου για να επεξεργαστούν Java και XML αρχεία και στη συνέχεια να χρησιμοποιήσουν τα εργαλεία της γραμμής εντολών για τη δημιουργία, την κατασκευή και τον εντοπισμό σφαλμάτων. Μια πρώτη έκδοση του Android Software Development Kit (SDK) κυκλοφόρησε στις 12 Νοεμβρίου 2007. Στις 15 Ιουλίου 2008, το Android Developer Challenge Team έστειλε κατά λάθος ένα e-Mail σε όλους τους συμμετέχοντες του Android Developer Challenge ανακοινώνοντας ότι μια νέα έκδοση του SDK ήταν διαθέσιμη σε μια "ιδιωτική" περιοχή για download. Το e-Mail προοριζόταν για τους νικητές του πρώτου γύρου του Android Developer Challenge. Η αποκάλυψη ότι η Google θα προμήθευε τις νέες εκδόσεις του SDK σε κάποιους προγραμματιστές και όχι σε όλους είχε προκαλέσει την δυσαρέσκεια στην κοινότητα των προγραμματιστών του Android.

Στις 18 Αυγούστου 2008, το Android 0.9 SDK Beta κυκλοφόρησε και παρείχε μια ενημερωμένη και εκτεταμένη API καθώς και βελτιωμένα εργαλεία ανάπτυξης. Λεπτομερείς οδηγίες για την αναβάθμιση είναι διαθέσιμες σε όσους ήδη εργάζονται με μια προηγούμενη έκδοση. Στις 23 Σεπτεμβρίου 2008, το Android 1.0 SDK ήταν διαθέσιμο στους χρήστες. Σύμφωνα με την Google, περιελάμβανε "κυρίως διορθώσεις σφαλμάτων, ενώ προστέθηκαν και κάποιες μικρότερες δυνατότητες". Περιλαμβάνονται επίσης αρκετές αλλαγές API από την έκδοση 0.9.

Στις 9 Μαρτίου 2009, η Google κυκλοφόρησε την έκδοση 1.1 για το Android Dev τηλέφωνο. Ενώ υπάρχουν μερικές αισθητικές ανανεώσεις, υπάρχουν και μερικές ζωτικής σημασίας βελτιώσεις όπως "αναζήτηση από φωνή, εφαρμογές μη δωρεάν, διορθώσεις στο ξυπνητήρι, διορθώσεις στην αποστολή e-Mail (gmail), διόρθωση ειδοποιήσεων μέσω ηλεκτρονικού ταχυδρομείου και διορθώσεις στα χρονικά διαστήματα μεταξύ των ανανεώσεων της συσκευής.". Ένα άλλο σημαντικό είναι ότι τα Android Dev τηλέφωνα μπορούν πλέον να δουν και τις μη δωρεάν εφαρμογές και οι προγραμματιστές μπορούν να ανεβάσουν τα προγράμματά τους στο Android Market.

Στα μέσα του Μάη του 2009, η Google κυκλοφόρησε την έκδοση 1.5 (Cupcake) του Android OS και SDK. Αυτή η ενημερωμένη έκδοση παρέχει πολλές νέες δυνατότητες όπως την εγγραφή βίντεο, υποστήριξη για το στερεοφωνικό προφίλ Bluetooth, ένα προσαρμοζόμενο σύστημα πληκτρολόγιο οθόνης και αναγνώρισης φωνής. Αυτή η έκδοση επίσης άνοιξε το πλαίσιο AppWidget σε τρίτους προγραμματιστές και έτσι θα επιτρέπεται σε οποιονδήποτε να δημιουργήσει το δικό του widgets για την επιφάνεια εργασίας (αρχική οθόνη).

Τον Σεπτέμβριο του 2009, η "Donut" έκδοση (1.6) που κυκλοφόρησε προσέφερε καλύτερη αναζήτηση, δείκτη χρήσης μπαταρίας και Virtual Personal Network εφαρμογή. Νέες τεχνολογίες προστέθηκαν όπως η ανάγνωση (εκφώνηση) κειμένου (δεν ήταν διαθέσιμο σε όλες τις συσκευές) και συντομεύσεις χειριού (Gestures).

Οι εφαρμογές Android αποθηκεύονται σε ένα αρχείο με κατάληξη .apk και εγκαθίστανται στον φάκελο /data/app του Android OS. Ο χρήστης μπορεί να τρέξει την εντολή `adb root` για να έχει πρόσβαση σε αυτό το φάκελο, καθώς μόνο ο διαχειριστής (root) έχει δικαιώματα πρόσβασης σε αυτόν το φάκελο.

2.4 ΕΡΓΑΛΕΙΑ ΤΟΥ ANDROID SDK

Το Android SDK μας προσφέρει μια πληθώρα «εργαλείων» που μας βοηθούν ώστε να αναπτύξουμε γρήγορα και εύκολα την εφαρμογή μας αλλά και να μπορέσουμε να την δοκιμάσουμε. Παρακάτω αναφέρονται αυτά που χρησιμοποιήσαμε για την ανάπτυξη της εφαρμογής μας:

2.4.1 Android Emulator

Είναι η προσομοίωση κινητού τηλεφώνου για τον υπολογιστή. Με βάση αυτό το «εργαλείο» θα σχεδιαστεί, θα ελεγχθεί για σφάλματα και θα δοκιμαστεί η εφαρμογή σε πραγματικό χρόνο συσκευής Android.

2.4.2 Hierarchy Viewer

Το συγκεκριμένο «εργαλείο» επιτρέπει τον εντοπισμό και την διόρθωση σφαλμάτων στην διεπαφή με τον χρήστη (User Interface). Παρέχει ένα μεγεθυμένο πλέγμα σχεδίασης ώστε να διευκολύνει το σχεδιασμό και τη διάταξη της οθόνης.

2.4.3 Android Development Tools Plug-in for the Eclipse IDE

Το συγκεκριμένο εργαλείο προσθέτει δυνατότητες στο ολοκληρωμένο περιβάλλον εργασίας Eclipse διευκολύνοντας τη δημιουργία εφαρμογών και τον εντοπισμό σφαλμάτων για την πλατφόρμα Android. Παρέχει πρόσβαση σε άλλα εργαλεία του Eclipse [24] όπως για παράδειγμα στο εργαλείο DDMS και στις πολλές του δυνατότητες (παρακολούθηση νήματος, breakpoints κ.α.). Θα μας διευκολύνει στην συγγραφή κώδικα αλλά κυρίως στην αυτόματη και χωρίς κόπο δημιουργία και διαχείριση όλων των βοηθητικών απαραίτητων στοιχείων για την σωστή λειτουργία της εφαρμογής μας.

2.4.4 Android Debug Bridge

Για να πετύχουμε σύνδεση με την βάση δεδομένων της εφαρμογής μας απαιτείται το παραπάνω «εργαλείο». Αυτό ρυθμίζει την σύνδεση με τον προσομοιωτή (emulator) που θα δοκιμαστεί η εφαρμογή μας. Ξεκινάει αυτόματα από το περιβάλλον υλοποίησης της εφαρμογής αλλά μας δίνει την δυνατότητα να το χρησιμοποιήσουμε και για σύνδεση με την βάση δεδομένων σε command prompt οθόνη.

2.5 ΠΛΑΤΦΟΡΜΑ ΑΝΟΙΚΤΟΥ ΚΩΔΙΚΑ

Το Android είναι μια πλατφόρμα Open Source, ανοικτού κώδικα ή ελεύθερου λογισμικού. Ελεύθερο ορίζεται από το Ίδρυμα Ελεύθερου Λογισμικού (Free Software Foundation), το λογισμικό που μπορεί να χρησιμοποιηθεί, αντιγραφεί, μελετηθεί, τροποποιηθεί και αναδιανεμηθεί χωρίς περιορισμό. Η απαγκίστρωση από τέτοιους περιορισμούς είναι βασικό στοιχείο στην ιδέα του "ελεύθερου λογισμικού", έτσι ώστε το αντίθετο του ελεύθερου είναι το ιδιόκτητο και όχι το λογισμικό που πωλείται για κέρδος, όπως το αντίστοιχο εμπορικό. Το ελεύθερο λογισμικό ορισμένες φορές αναφέρεται και σαν ανοιχτό λογισμικό ή λογισμικό ανοιχτού κώδικα αλλά οι δύο έννοιες δεν είναι ταυτόσημες. Σύμφωνα με τον Richard Stallman δεν είναι κάθε λογισμικό ελεύθερο μόνο και μόνο επειδή είναι ανοιχτού κώδικα. Εν γένει, σύμφωνα με την ισχύουσα νομοθεσία περί πνευματικής ιδιοκτησίας, η ελεύθερη αντιγραφή, διανομή και τροποποίηση του λογισμικού δεν επιτρέπεται. Για το λόγο αυτό στις εκδόσεις ελεύθερου λογισμικού γίνεται χρήση ειδικής άδειας (free software licence) σύμφωνα με την οποία, παραχωρείται το δικαίωμα αντιγραφής, τροποποίησης και αναδιανομής του λογισμικού, στους χρήστες. Σύμφωνα με το Ίδρυμα Ελεύθερου Λογισμικού [10], οι άδειες χρήσης ελεύθερου λογισμικού πρέπει να περιλαμβάνουν τις εξής ελευθερίες:

- Ελευθερία 0: Ελευθερία χρήσης του προγράμματος για οποιονδήποτε σκοπό.
- Ελευθερία 1: Ελευθερία μελέτης και τροποποίησης του προγράμματος.
- Ελευθερία 2: Ελευθερία αντιγραφής του προγράμματος.
- Ελευθερία 3: Ελευθερία βελτίωσης του προγράμματος και επανέκδοσης του, προς το συμφέρον της κοινότητας των χρηστών.

Οι ελευθερίες 1 και 3 προϋποθέτουν την πρόσβαση των χρηστών στον πηγαίο κώδικα του λογισμικού.

2.6 ΕΥΡΟΣ ΕΦΑΡΜΟΓΩΝ

Το Android είναι νέο και πολλά υποσχόμενο λειτουργικό σύστημα το οποίο αυτή τη στιγμή υποστηρίζει οτιδήποτε μπορεί να υποστηρίξει ένα λειτουργικό σύστημα για κινητά. Αυτά είναι, τηλεφωνικές κλήσεις, μηνύματα, MMS, Camera, Video, Java, Memory Card, GPS, Bluetooth, GPRS, 3G, HSDPA, WiFi, Πυξίδα κλπ. Από ότι γίνεται αντιληπτό το λειτουργικό σύστημα της Google διαθέτει τα πάντα για να γίνει και να είναι ανταγωνιστικό. Επιπλέον δίνει την δυνατότητα στον προγραμματιστή να συνδυάσει άρτια όλες τις παραπάνω δυνατότητες σε μια εφαρμογή. Όπως για παράδειγμα μέσω του GPS να αντιλαμβάνεται σε ποια θέση βρίσκεται (π.χ. ένας γιατρός μέσα σε ένα νοσοκομείο) και έπειτα με την χρήση του WiFi να αντλεί τα δεδομένα από έναν Server. Από τα παραπάνω προκύπτει πως οι πραγματικοί περιορισμοί για το Android είναι η φαντασία του κάθε προγραμματιστή. Το Android σε λίγα χρόνια θα αποτελεί το κύριο λειτουργικό όλων των κινητών τηλεφώνων και όχι μόνο. Και σε όλα αυτά δεν θα πρέπει να ξεχασθεί πως το στηρίζει η Google. Μια εταιρεία που αποτελεί τον κυρίαρχο στο διαδίκτυο και όπως φάνηκε στα παραπάνω, η σύγχρονη τάση είναι να γίνονται όλο και πιο πολλά πράγματα αποκλειστικά στο Internet και οι χρήστες να έχουν απλά ένα terminal όπου θα μπορούν να βλέπουν τα αποτελέσματα, το λεγόμενο Cloud System. Για αυτό το λόγο επιλέχθηκε η πλατφόρμα Android για την ανάπτυξη της εφαρμογής όπου θα επικοινωνεί με την βάση δεδομένων που υπάρχει σε έναν Server μέσω WiFi.

2.7 Η ΕΠΙΛΟΓΗ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ ANDROID

Μετά την ανάλυση της πλατφόρμας Android αλλά και την αναφορά στα γενικά χαρακτηριστικά άλλων αντίστοιχων πλατφορμών φορητών συσκευών, θα αναφερθούν οι λόγοι που καθιστούν την πλατφόρμα Android ως την καλύτερη ή τουλάχιστον μια από τις καλύτερες επιλογές για την ανάπτυξη της εφαρμογής αυτής της πτυχιακής εργασίας. Παρακάτω αναφέρονται τα πλεονεκτήματα και τα μειονεκτήματα της πλατφόρμας Android.

2.7.1 Πλεονεκτήματα:

- ❖ Χαμηλότερο κόστος στην αγορά κινητού:
 - Δωρεάν στους κατασκευαστές κινητών.
 - Φθηνότερες συσκευές.
- ❖ Open Source:
 - Μπορείς να προσαρμόσεις πλήρως το τηλέφωνο με τις εκάστοτε απαιτήσεις σου.
 - Νέες καινοτομικές εφαρμογές.
- ❖ Java API & development tools for Windows, Mac, Linux.
- ❖ Θα αποτελέσει την βάση για την μελλοντική εξέλιξη των κινητών τηλεφώνων αυτού του τύπου.
- ❖ Η Google παρέχει πολύ καλό documentation.
- ❖ Δεν υπάρχουν κλειδωμένα προγράμματα.
- ❖ Μικρές και νέες επιχειρήσεις «μπαίνουν» στην αγορά.

2.7.2 Μειονεκτήματα:

- ❖ Απόρριψη τα πλατφόρμας από μεταφορείς οι οποίοι επιβάλλουν τέλη:
 - Web-based εφαρμογές επιτρέπουν δωρεάν πρόσβαση στους κάτοχους κινητών με Google Android.
- ❖ Ακόμα δεν υπάρχουν πάρα πολλές συσκευές που να «υποστηρίζουν» το Android και αυτό σημαίνει λίγες δυνατές επιλογές για τον χρήστη.
- ❖ Το emulator που παρέχεται έχει περιορισμένες δυνατότητες.

ΚΕΦΑΛΑΙΟ 3

3.1 ΗΛΕΚΤΡΟΝΙΚΟΣ ΙΑΤΡΙΚΟΣ ΦΑΚΕΛΟΣ

3.1.1 Ιατρικός Φάκελος.

Για να γίνει κατανοητός ο Ηλεκτρονικός Ιατρικός Φάκελος, θα πρέπει πρώτα να γνωρίζουμε τι είναι ένας Ιατρικός Φάκελος. Η έννοια του ιατρικού φακέλου είναι τόσο παλιά όσο η ιατρική όπως την εννοούμε στο σύγχρονο πολιτισμό. Οι ιατροί εκπαιδεύονται από τα πρώτα χρόνια των σπουδών τους στη συγκέντρωση στοιχείων που αφορούν τον ασθενή. Άλλωστε, η ίδια η φύση της ιατρικής πρακτικής στηρίζεται στη σωστή συγκέντρωση στοιχείων και στην αξιολόγηση αυτών. Ως ιατρικό φάκελο γίνεται να οριστεί το σύνολο των πληροφοριών που αφορούν έναν ασθενή και τη κατάσταση της υγείας του. Ο ιατρικός φάκελος αποτελεί ένα πολύ σημαντικό εργαλείο στα χέρια του ιατρού και συνεπώς κατέχει μια πολύ σημαντική θέση στη διαφοροδιαγνωστική προσέγγιση και στην επιλογή της κατάλληλης θεραπείας του ασθενή. Ο φάκελος περιέχει πληροφορίες για την αιτία, την πορεία και την κατάληξη μιας ασθένειας. Ο ιατρικός φάκελος ενός συγκεκριμένου ατόμου, όταν είναι σωστά οργανωμένος δίνει στο θεράποντα ιατρό τη δυνατότητα να γνωρίζει στοιχεία από προηγούμενες ή παράλληλες θεραπευτικές αγωγές και να επεμβαίνει με τον καταλληλότερο, για την περίπτωση, τρόπο. Ακόμα και σήμερα πολλοί ιατροί τηρούν τους ιατρικούς φακέλους σε έντυπη μορφή, γεγονός το οποίο έχει σοβαρά μειονεκτήματα όπως για παράδειγμα η δυσκολία ανεύρεσης στοιχείων από το οικογενειακό και ατομικό αναμνηστικό, ο μεγάλος όγκος ενός εκτεταμένου έντυπου ιατρικού φακέλου, η διάσπαση των στοιχείων ενός ασθενή σε κάθε ιατρό χωριστά, η δυσκολία τήρησης επιδημιολογικών δεδομένων, κλπ. Παράλληλα, είναι γεγονός ότι ο κλάδος της Ιατρικής πάντοτε καθυστερεί στην υιοθέτηση των τεχνολογιών που του προσφέρει η Πληροφορική. Το γεγονός αυτό αγγίζει τα όρια της παραδοξολογίας αν αναλογιστεί κανείς την εμπιστοσύνη που δείχνουν οι ιατροί στη τεχνολογία των προηγμένων διαγνωστικών και θεραπευτικών ιατρικών συσκευών. Δικαιολογείται όμως εν μέρει αν αναλογιστούμε την πολυπλοκότητα που παρουσιάζει η κωδικοποίηση των ιατρικών πράξεων, των ασθενειών, κλπ. Η διάδοση των ηλεκτρονικών υπολογιστών στη σημερινή κοινωνία και η σταδιακή ανέλιξη σε στρατηγικές θέσεις νέων ανθρώπων με εμπιστοσύνη στις νέες τεχνολογίες αλλάζει σταδιακά τη κατάσταση. Τα τελευταία είκοσι χρόνια γίνεται εκτεταμένη έρευνα σχετικά με τη δημιουργία **ηλεκτρονικών ιατρικών φακέλων**. Σύμφωνα με το Ινστιτούτο Ιατρικής των Η.Π.Α. αναφέρει ότι ο ηλεκτρονικός ιατρικός φάκελος "είναι ένα σύστημα σχεδιασμένο με τέτοιο τρόπο ώστε να υποστηρίζει την απόλυτη διαθεσιμότητα και την ακρίβεια ιατρικών ή άλλων πληροφοριών με σκοπό τη παροχή ιατρικής περίθαλψης". Σε διεθνές επίπεδο είναι πλέον ευρέως αποδεκτό ότι ο ηλεκτρονικός ιατρικός φάκελος έρχεται να διευκολύνει ζητήματα όπως:

- Η συγκέντρωση, η διαχείριση και ο αυτόματος συγχρονισμός δεδομένων.
- Η ιατρική έρευνα.
- Η τήρηση επιδημιολογικών δεδομένων σε ευρεία κλίμακα.
- Η κοστολόγηση των ιατρικών πράξεων.
- Η μείωση ιατρικών σφαλμάτων και φαινομένων διαφθοράς.
- Η συνεχιζόμενη εκπαίδευση του ιατρικού προσωπικού.
- Η τήρηση των νόμων και των ιατρικών πρωτοκόλλων.

Η δημιουργία τέτοιων αρχείων έχει πολύ θετικά αποτελέσματα στην έγκαιρη και σωστή διάγνωση ασθενειών, στη σωστή παρακολούθηση ασθενών, στη εξάλειψη διπλών εγγράφων, στη μείωση του κόστους λειτουργίας (μείωση εξετάσεων και ακτινογραφιών), ενώ επιπλέον δημιουργείται μια συνεχώς αναπτυσσόμενη ηλεκτρονική βιβλιοθήκη η οποία είναι εξαιρετικής σημασίας στην ιατρική έρευνα.

3.1.2 Ηλεκτρονικός Ιατρικός Φάκελος.

Με τον όρο «Ηλεκτρονικός Ιατρικός Φάκελος» εννοείται η ανάπτυξη ενός ψηφιοποιημένου λογισμικού όπου δύναται να καταγράφει όλα τα δεδομένα που προκύπτουν από την ποικιλότητα επικοινωνία του ιατρού με τον ασθενή. Έτσι η εφαρμογή αυτή αποκτά την δυναμική της αναπαράστασης της εικόνας της υγείας του ασθενούς, με όλες τις σχετικές πληροφορίες, καθώς και την δυναμική της παρουσίασης της ιατρικής μέριμνας και φροντίδας βήμα προς βήμα. Ο Ηλεκτρονικός Ιατρικός Φάκελος περιλαμβάνει στοιχεία του ασθενούς τα οποία προκύπτουν από τις επισκέψεις σε γιατρούς με ότι αυτές εμπεριέχουν (ιστορικό, διάγνωση, θεραπείες). Περιλαμβάνει επίσης και όλα τα δημογραφικά και ιατρικά στοιχεία ατομικού και οικογενειακού αναμνηστικού (π.χ. λήψη φαρμάκων, πιθανή ύπαρξη αλλεργιών). Παράλληλα εμπεριέχει αναλυτικά στοιχεία και εκτιμήσεις της παρούσας κατάστασης του ασθενούς, τις κλινικές σημειώσεις και τη διάγνωση που έχει τεθεί από τον εκάστοτε επαγγελματία υγείας. Η σημαντικότητα του Ηλεκτρονικού Ιατρικού Φακέλου έγκειται στο γεγονός ότι τα στοιχεία του αποθηκεύονται με ασφάλεια σε μία βάση δεδομένων και είναι ανά πάσα στιγμή στη διάθεση του ασθενούς και του επαγγελματία υγείας. Επίσης, η διαφορετικότητα των επαγγελματιών υγείας έχει επιβάλει την παραμετροποίηση αυτού στις εκάστοτε ανάγκες.

3.1.3 Υπάρχουσα κατάσταση στην Ελλάδα.

Στην Ελλάδα σήμερα δεν τηρούνται ηλεκτρονικοί ιατρικοί φάκελοι, ούτε καν στην πιο απλή τους μορφή, ενώ όταν συναντώνται βρίσκονται ακόμα σε πιλοτικό στάδιο. Γενικά η κατάσταση που επικρατεί στα ελληνικά νοσοκομεία και κατά γενικό κανόνα στα νοσοκομεία του εξωτερικού συνοψίζεται στα παρακάτω:

1. Ανυπαρξία της «ολιστικής» θεώρησης του ασθενούς
2. Ανυπαρξία ανταγωνιστικών λογισμικών προγραμμάτων που να υποστηρίζουν την επικοινωνία ιατρού – ασθενούς, η οποία αποτελεί και τον ακρογωνιαίο λίθο της επιτυχούς καθημερινής ιατρικής πράξης.
3. Τα όποια υπάρχοντα συστήματα οργάνωσης ηλεκτρονικού ιατρικού φακέλου:
 - 3.1. Έχουν μονοαξονικό προσανατολισμό κινούμενα είτε σε χρονοκεντρικό, είτε σε προβληματοκεντρικό άξονα προσέγγισης του τομέα παροχής υπηρεσιών υγείας.
 - 3.2. Δυσχεραίνουν και κάνουν πολυδαίδαλη τη διαδικασία για τους χρήστες μέσα από την χρήση πολυάριθμων παραθύρων.
 - 3.3. Εφαρμόζουν ως μεθόδους εισαγωγής δεδομένων είτε μόνο την ελεύθερη περιγραφή, είτε μόνο την δομημένη είσοδο πληροφοριών.
 - 3.4. Υφίσταται μία δυσκολία στη διακίνηση της πληροφορίας, δηλαδή όλων των στοιχείων που απαρτίζουν το υγεινολογικό προφίλ του ασθενή.
 - 3.5. Συνήθως δεν είναι δυνατή η επικοινωνία επιμέρους ιατρικών ειδικοτήτων ή όταν υφίσταται είναι αναποτελεσματική.

3.2 ΠΡΟΔΙΑΓΡΑΦΕΣ ΗΛΕΚΤΡΟΝΙΚΟΥ ΙΑΤΡΙΚΟΥ ΦΑΚΕΛΟΥ

Οι προδιαγραφές που οφείλει να έχει ένας ηλεκτρονικός ιατρικός φάκελος πρέπει να είναι υψηλότερες, ανταποκρινόμενες στις υψηλές απαιτήσεις του επιστημονικού ιατρικού χώρου, όσο και στις απαιτήσεις για εύκολη χρήση και αξιοπιστία. Οπότε ένα ηλεκτρονικός ιατρικός φάκελος πρέπει να διαθέτει:

- 1) Φιλικό περιβάλλον εργασίας.
- 2) Εύχρηστο και ταχύτατο περιβάλλον εργασίας.
- 3) Ευληπτότητα δεδομένων.
- 4) Ταυτόχρονη χρήση από πολλούς χρήστες.
- 5) Εξάλειψη προβλημάτων αναγνωσιμότητας.
- 6) Ελευθερία έκφρασης μέσω της διαδραστικής κίνησης σε μία τριπλέτα:
 - i) Ελεύθερου κειμένου.
 - ii) Δομημένης εισόδου δεδομένων.
 - iii) Επισημείωσης επί εικόνων.
- 7) Δυνατότητα περίπλοκης ανάλυσης δεδομένων και υποστήριξη έρευνας (επιδημιολογικές μελέτες, κλινική έρευνα, αξιολογήσεις κλπ.).
- 8) Η ολική ή μερική αναφορά όλων των ιατρικών πράξεων σύμφωνα με όσα ορίζουν οι ισχύουσες συναφείς νομικές διατάξεις.
- 9) Στήριξη της ιατρικής απόφασης.
- 10) Εύκολη και ασφαλή διακίνηση δεδομένων ποικιλοτρόπως μέσω:
 - i) Ψηφιακών αποθηκευτικών μέσων.
 - ii) Διαδικτύου.
 - iii) Τοπικών διαδικτυακών συστημάτων.
- 11) «Ολιστική» προσέγγιση του ασθενή μέσα από την απρόσκοπτη ανταλλαγή πληροφοριών μεταξύ διαφορετικών ιατρικών ειδικοτήτων.
- 12) Ενίσχυση της επικοινωνίας μεταξύ ιατρού – ασθενή.

3.3 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΗΛΕΚΤΡΟΝΙΚΟΥ ΙΑΤΡΙΚΟΥ ΦΑΚΕΛΟΥ

Τα συστήματα Ηλεκτρονικού Ιατρικού Φακέλου (ΗΙΦ) πετυχαίνουν αύξηση της αποτελεσματικότητας του ιατρού και μείωση του κόστους, καθώς και την προώθηση τυποποίησης της περίθαλψης. Παρά το γεγονός ότι τα συστήματα ΗΙΦ υπάρχουν εδώ και περισσότερο από 30 χρόνια, λιγότεροι από το 10% των νοσοκομείων από το 2006 διαθέτει ένα πλήρως ολοκληρωμένο σύστημα.

3.3.1 Μείωση του κόστους υγειονομικής περίθαλψης.

Μία από τις κύριες πηγές της ταχείας αύξησης του κόστους υγειονομικής περίθαλψης προέρχεται από τις ιατρικές εικόνες. Έρευνες έδειξαν πως οι δαπάνες για την απεικόνιση αυξήθηκαν από 6,80 δισ. δολάρια το 2000 σε 14,11 δισεκατομμυρίων δολαρίων το 2006. Η πρόσβαση σε εικόνες ενός ασθενούς σε ένα ΗΙΦ είναι ένας αποτελεσματικός τρόπος για την αποφυγή της επανάληψης των δαπανηρών διαδικασιών απεικόνισης. Επίσης μείωση του κόστους μπορεί να επιτευχθεί από τη μείωση των ιατρικών σφαλμάτων που μπορεί διαφορετικά να οδηγούσαν σε περαιτέρω δαπάνες.

3.3.2 Βελτίωση της ποιότητας της περίθαλψης.

Ένα σύστημα ΗΙΦ συμβάλλει στη μείωση των ιατρικών σφαλμάτων με την παροχή στο νοσηλευτικό προσωπικό την δυνατότητα υποστήριξης της λήψης αποφάσεων. Η γρήγορη πρόσβαση σε ιατρική βιβλιογραφία και τις τρέχουσες βέλτιστες πρακτικές στον τομέα της ιατρικής είναι μπορεί να επιφέρει βελτιώσεις στην αποτελεσματικότητα της υγειονομικής περίθαλψης.

3.4 ΜΕΙΟΝΕΚΤΗΜΑΤΑ ΤΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΙΑΤΡΙΚΩΝ ΦΑΚΕΛΩΝ

Οι επικριτές επισημαίνουν ότι ενώ τα συστήματα ΗΙΦ μπορεί να σώσουν τα χρήματα του «συστήματος υγείας», οι γιατροί, οι οποίοι αγοράζουν τα συστήματα, δεν μπορούν να επωφεληθούν οικονομικά. Οι τιμές των συστημάτων ΗΙΦ καλύπτουν ένα ευρύ φάσμα, ανάλογα με το τι περιλαμβάνει, πώς εύρωστο το σύστημα είναι, και πόσοι πάροχοι το χρησιμοποιούν. Σε μια online έρευνα ρωτήθηκαν οι γιατροί πόσα λεφτά τους δίνονται για να αγοράσουν ένα σύστημα ΗΙΦ. Τα αποτελέσματα χωρίζονται σε τρεις κατηγορίες i) \$500 με \$3.000, ii) \$3.001 με \$6.000 και iii) πάνω από \$6.000 ανά ιατρό. Οι γιατροί έχουν την τάση να βλέπει τουλάχιστον για ένα μικρό χρονικό διάστημα μια μείωση της παραγωγικότητας μετά την χρήση ενός συστήματος ΗΙΦ. Ξοδεύουν περισσότερο χρόνο για την εισαγωγή δεδομένων σε έναν άδειο ΗΙΦ σε σχέση με το να περάσουν την ενημέρωση σε ένα γράφημα με μια απλή υπαγόρευση. Τα εν λόγω εμπόδια μπορούν να ξεπεραστούν όταν το λογισμικό διαθέτει ορισμένα δεδομένα, καθώς οι γιατροί μαθαίνουν να χρησιμοποιούν εύκολα πρότυπα για την εισαγωγή δεδομένων. Μελέτες επίσης θέτουν υπό αμφισβήτηση το κατά πόσον, στην πραγματική ζωή, τα συστήματα ΗΙΦ βελτιώνουν την ποιότητα παροχής υπηρεσιών υγείας.

3.5 ΠΡΟΤΥΠΑ

Μέχρι στιγμής είδαμε τι είναι και από τι πρέπει να αποτελείται ένας Ηλεκτρονικός Ιατρικός Φάκελος. Αξίζει όμως να αναφέρουμε και μερικές υλοποιήσεις – πρότυπα που ήδη υπάρχουν στην αγορά:

- 1) **ANSI X12 (EDI)**, Πρωτόκολλα συναλλαγής χρησιμοποιούνται για τη ανταλλαγή δεδομένων των ασθενών. Δημοφιλές στις Ηνωμένες Πολιτείες για τη διαβίβαση των στοιχείων χρέωσης.
- 2) **CEN 's TC/251**, Παρέχει πρότυπα ΗΙΦ στην Ευρώπη, μεταξύ των οποίων:
 - a) **EN 13606**, Πρότυπα επικοινωνίας για πληροφορίες ΗΙΦ.
 - b) **CONTSYS (EN 13940)**, Υποστηρίζει τη συνέχεια της τυποποίησης των εγγραφών φροντίδας.
 - c) **HISA (EN 12967)**, Ένα πρότυπο για την ενδοεπικοινωνία ενός συστήματος ΗΙΦ σε ένα νοσοκομείο.
- 3) **Continuity of Care Record – ASTM**, Διεθνής συνέχεια του προτύπου Care Record.
- 4) **DICOM**, Διεθνές πρωτόκολλο επικοινωνιών για την εκπροσώπηση και τη διαβίβαση ακτινογραφιών και άλλων δεδομένων που υπάρχουν σε εικόνες, υποστηρίζεται από την NEMA (National Electrical Manufacturers Association).
- 5) **HL7**, Ένα τυποποιημένο σύστημα επικοινωνίας μέσω μηνυμάτων και κειμένων μεταξύ των συστημάτων ΗΙΦ που έχει ένα νοσοκομείο και ένας ιατρός.
- 6) **ISO - ISO TC 215**, Παρέχει διεθνείς τεχνικές προδιαγραφές για τα συστήματα ΗΙΦ. Το ISO 18308 περιγράφει τις αρχιτεκτονικές των ΗΙΦ.

3.6 ΜΑΚΡΟΠΡΟΘΕΣΜΗ ΔΙΑΤΗΡΗΣΗ ΚΑΙ ΦΥΛΑΞΗ ΑΡΧΕΙΩΝ

Μία σημαντική παράμετρος κατά τη διαδικασία της ανάπτυξης ΗΙΦ είναι το σχέδιο για την μακροπρόθεσμη διατήρηση και την αποθήκευση αυτών των αρχείων. Αυτό είναι πολύ σημαντικό και για τους ίδιους τους ασθενείς αλλά και για τρίτα άτομα. Για τον ασθενή είναι σημαντικό ώστε όσα χρόνια και αν έχουν περάσει να μπορεί να βρει κάποιος γιατρός το ιατρικό τους ιστορικό και να είναι σε θέση να συνδυάσει όλα τα απαραίτητα στοιχεία για την καλύτερη περίθαλψη του. Επιπλέον αν υπάρχουν τα δεδομένα για μεγάλο χρονικό διάστημα θα μπορούμε με μεγαλύτερη ευκολία να εξάγουμε κάποια στατιστικά αποτελέσματα, π.χ. πως εξελίχθηκε μια ασθένεια και με ποιους τρόπους αντιμετωπίστηκε. Αυτή η δυνατότητα θα βοηθήσει τρίτα άτομα που πάσχουν από την ίδια ασθένεια. Η απαιτούμενη διάρκεια της αποθήκευσης ενός ΗΙΦ ενός ατόμου θα εξαρτηθεί από τους εθνικούς και τοπικούς

κανονισμούς, που μπορεί να αλλάξουν με την πάροδο του χρόνου. Οι Ruotsalainen και Manning έχουν διαπιστώσει ότι ο τυπικός χρόνος διατήρησης των δεδομένων για τους ασθενείς κυμαίνεται μεταξύ 20 και 100 χρόνια. Αν και είναι προς το παρόν άγνωστο ακριβώς πόσο καιρό ένας ΗΙΦ θα πρέπει να διατηρηθεί, είναι βέβαιο ότι η διάρκεια του χρόνου θα υπερβαίνει τη μέση διάρκεια ζωής των εγγραφών χαρτιού. Η εξέλιξη της τεχνολογίας είναι τέτοια ώστε τα προγράμματα και τα συστήματα που χρησιμοποιούνται για την εισαγωγή πληροφοριών θα μπορούν να είναι στη διάθεση ενός χρήστη που επιθυμεί να εξετάσει τα αρχειοθετημένα δεδομένα. Μία προτεινόμενη λύση για την πρόκληση της μακράς διάρκειας της προσβασιμότητας και χρηστικότητας των δεδομένων από μελλοντικά συστήματα είναι η τυποποίηση των πεδίων πληροφοριών σε ένα time-invariant τρόπο, όπως με τη γλώσσα XML. Αυτή την στιγμή, δύο είναι οι Η.Ι.Φ. που επικρατούν, το Microsoft Medical Records και το Google Medical Records, αλλά παρόλα αυτά δεν έχουν γίνει ακόμα καθημερινή συνήθεια των ιατρών, πόσο μάλλον των ασθενών όπου στην περίπτωση της Google πρέπει ο ασθενής να συμπληρώνει το ιστορικό, διότι αυτό δεν δημιουργείτε αυτόματα μετά από τις εκάστοτε εξετάσεις που κάνει σε κάποιο νοσοκομείο.

3.7 Ο ΣΥΓΧΡΟΝΙΣΜΟΣ ΤΩΝ ΑΡΧΕΙΩΝ

Στην προηγούμενη ενότητα μελετήθηκε η δυνατότητα και τα οφέλη για μια μακροπρόθεσμη αποθήκευση των δεδομένων ενός ΗΙΦ. Μια άλλη όμως παράμετρος που πρέπει να μελετηθεί είναι το κατά πόσο είναι δυνατόν και με ποιο τρόπο η συγχώνευση – συγχρονισμός ενός ΗΙΦ ενός ασθενούς όταν αυτός χρησιμοποιείται σε παραπάνω από ένα νοσοκομείο ή γιατρό. Όταν η φροντίδα παρέχεται σε δύο διαφορετικές εγκαταστάσεις, μπορεί να είναι δύσκολο να ενημερωθούν οι εγγραφές σε δύο τοποθεσίες με έναν συντονισμένο τρόπο. Προτάθηκαν δύο μοντέλα για να λυθεί αυτό το πρόβλημα i) μια ενιαία λύση ενός διακομιστή δεδομένων και ii) ένα Peer-to-Peer αρχείο προγράμματος συγχρονισμού (όπως έχει αναπτυχθεί για άλλα peer-to-peer δίκτυα). Στις Ηνωμένες Πολιτείες Αμερικής, την Μεγάλη Βρετανία, και την Γερμανία, η έννοια ενός εθνικού συγκεντρωτικού μοντέλου διακομιστή των δεδομένων ΗΙΦ έχει βρει ελάχιστη εφαρμογή. Ζητήματα της ιδιωτικής ζωής και της ασφάλειας σε ένα τέτοιο μοντέλο έχουν προκαλέσει ανησυχίες. Η συγχώνευση των ήδη υφιστάμενων δημόσιων βάσεων δεδομένων υγειονομικής περίθαλψης είναι μια κοινή πρόκληση για τους προγραμματιστές. Η ικανότητα των συστημάτων ΗΙΦ να παρέχουν αυτή τη λειτουργία είναι ένα βασικό πλεονέκτημα και μπορούν να βελτιώσουν την παροχή υγειονομικής περίθαλψης.

3.8 ΠΡΟΣΤΑΣΙΑ ΠΡΟΣΩΠΙΚΩΝ ΔΕΔΟΜΕΝΩΝ

Η προστασία των προσωπικών δεδομένων είναι ένα ζήτημα που απασχολεί την κοινωνία μας τις τελευταίες δεκαετίες πάρα πολύ και έχουν θεσπιστεί νόμοι όπου προστατεύουν τα προσωπικά μας δεδομένα όπως ηλικία, έσοδα, κατοικία κλπ. Όπως είναι εύλογο δεν θα μπορούσε ο ιατρικός μας φάκελος να αποτελεί εξαίρεση. Είναι πολύ σημαντικό να προστατεύεται ο ιατρικό φάκελος και να μην έχει κανείς πρόσβαση σε αυτόν χωρίς την άδειά του κατόχου. Στην υγειονομική περίθαλψη η προστασία προσωπικών δεδομένων εφαρμόζεται τόσο στα έντυπα αλλά και στα ηλεκτρονικά αρχεία. Σύμφωνα με την εφημερίδα Los Angeles Times, περίπου 150 άτομα, όπως γιατροί, νοσοκόμες, τεχνικοί και υπάλληλοι χρέωσης πρέπει να έχουν πρόσβαση σε τουλάχιστον ένα μέρος των αρχείων του ασθενούς κατά τη διάρκεια της νοσηλείας του. Επιπλέον, πρόσβαση πρέπει να έχουν και 600.000 φορολογούμενοι, πάροχοι και άλλοι φορείς που χειρίζονται τα δεδομένα χρέωσης. Πρόσφατες έρευνες έδειξαν πως παραβιάσεις της «ασφάλειας» δεδομένων σε κεντρικά σημεία αποθήκευσης δεδομένων, σε τράπεζες και άλλα χρηματοπιστωτικά ιδρύματα, στον κλάδο λιανικής, και από τις βάσεις δεδομένων της κυβέρνησης, έχουν προκαλέσει την ανησυχία για το αν θα πρέπει η αποθήκευση των ηλεκτρονικών ιατρικών φακέλων να γίνεται σε μια Online κεντρική θέση (Server). Τα αρχεία που ανταλλάσσονται μέσω του Internet

υπόκεινται στις ίδιες ανησυχίες για την ασφάλεια όπως και κάθε άλλο είδος δεδομένων για τις συναλλαγές μέσω του διαδικτύου. Το Health Insurance Portability και Accountability Act (HIPAA) ψηφίστηκε στις ΗΠΑ το 1996, ώστε να θεσπίσει κανόνες για την πρόσβαση, την παροχή άδειας εισόδου, την αποθήκευση και τον έλεγχο, καθώς και αποστολή/λήψη των ηλεκτρονικών ιατρικών φακέλων. Αυτό το πρότυπο έχει αυστηρότερους περιορισμούς για τις ηλεκτρονικές εγγραφές έναντι των εγγραφών που γίνονται στο χαρτί. Ωστόσο, υπάρχουν ανησυχίες ως προς την καταλληλότητα της εφαρμογής αυτών των προτύπων. Στην Ευρωπαϊκή Ένωση (ΕΕ), αρκετοί νόμοι του Ευρωπαϊκού Κοινοβουλίου και του Συμβουλίου που προστατεύει την επεξεργασία και την ελεύθερη κυκλοφορία των δεδομένων προσωπικού χαρακτήρα, αναφέρονται στα συστήματα υγείας. Στον Καναδά, στις 13 Απριλίου 2000, δόθηκε στην Personal Information Protection and Electronic Documents Act (PIPEDA) να θεσπίσει κανόνες για τη χρήση, την αποκάλυψη και τη συλλογή των προσωπικών πληροφοριών. Τα προσωπικά στοιχεία τόσο τα μη-ψηφιακή αλλά και αυτά που έχουν ηλεκτρονική μορφή. Το 2002, η PIPEDA επεκτάθηκε στον τομέα της υγείας στο 2ο στάδιο της εφαρμογής του νόμου. Υπάρχουν τέσσερις επαρχίες, όπου ο νόμος αυτός δεν ισχύει, επειδή το δικαίωμα προστασίας της ιδιωτικής ζωής τους θεωρείται παρόμοιο με αυτό της PIPEDA i) Alberta, ii) British Columbia, iii) Ontario και iv) Quebec.

3.8.1 Προστασία και ασφάλεια των Ηλεκτρονικών Εγγράφων Υγείας.

Δεδομένου ότι η συνεχώς μεταβαλλόμενη βιομηχανία υγείας εξελίσσεται, ένα βασικό θέμα στο πλαίσιο των ηλεκτρονικών μητρώων υγείας (ΗΜΥ), είναι η προστασία. Η ομοσπονδιακή κυβέρνηση έχει θέσει τις κατευθυντήριες γραμμές ότι όλες οι οργανώσεις υγειονομικής περίθαλψης θα πρέπει να συμμορφώνονται με ότι αφορά τις ηλεκτρονικές συναλλαγές υγείας. Οι περισσότεροι, που υποστηρίζουν τα συστήματα ΗΙΦ πιστεύουν ότι αυτός θα βελτιώσει την φροντίδα και θα μειώσει το κόστος, θα μετατρέψει τον τρόπο λειτουργίας του συστήματος υγειονομικής περίθαλψης, αλλά κατά πόσο μπορεί να επιτευχθεί και να γίνει δεκτή η προστασία των αρχείων του ΗΙΦ δεν μπορεί ακόμη να προσδιοριστεί. Μια επιτυχημένη υλοποίηση για την διαχείριση των δεδομένων του ΗΙΦ μπορεί να προωθήσει την ανάπτυξη των προτύπων των κλινικών δεδομένων και την εφαρμογή τους στα συστήματα ηλεκτρονικής καταγραφής των ασθενών. Ένα σημαντικό θέμα που έχει αυξηθεί για την προστασία της ασφάλειας του δικτύου των ΗΠΑ για τα ηλεκτρονικά μητρώα υγείας είναι η στρατηγική για να εξασφαλίσει την προστασία της ιδιωτικής ζωής των ασθενών. Ο πρώην πρόεδρος των ΗΠΑ Bush ζήτησε τη δημιουργία των δικτύων, αλλά ομοσπονδιακοί ερευνητές αναφέρουν ότι δεν υπάρχει σαφής τρόπος για την προστασία των προσωπικών δεδομένων των ασθενών και των ηλεκτρονικών ιατρικών φακέλων. Η απειλή της ασφάλειας που θέτει η διαλειτουργικότητα του εθνικού δικτύου αποτελεί βασικό μέλημα. Ένας από τους επικριτές του ΗΙΦ, ο καθηγητής Jacob M. Appel (Πανεπιστήμιο Νέας Υόρκης), ισχυρίστηκε ότι ο αριθμός των ατόμων που θα πρέπει να έχουν πρόσβαση σε ένα τέτοιο πραγματικά διαλειτουργικό εθνικό σύστημα, που εκτιμάται περίπου σε 12 εκατομμύρια, θα οδηγήσει αναπόφευκτα για παραβιάσεις της ασφάλειας σε μαζική κλίμακα. Ο Appel έχει γράψει ότι ενώ "τα νοσοκομεία τηρούν προσεκτικά καρτέλες σχετικά με το ποιος έχει πρόσβαση στα διαγράμματα των VIP ασθενών," δεν έχουν την δυνατότητα να ενεργήσουν εναντίον "ενός ενοχλητικού φαρμακοποιού στην Αλάσκα" που "αναζητά την τοξικολογία ούρων για τον αρραβωνιαστικός της κόρης του στη Φλόριντα, για να ελέγξει αν ο μελλοντικός γαμπρός του κάνει χρήση κοκαΐνης." Αυτό είναι ένα σημαντικό εμπόδιο για την υιοθέτηση ενός ΗΙΦ. Η λογοδοσία για εμπιστευτικότητα μεταξύ όλων των φορέων που εμπλέκονται στην επεξεργασία των ηλεκτρονικών συναλλαγών, συμπεριλαμβανομένου του ασθενή, του γιατρού του προσωπικού γραφείου, και των ασφαλιστικών εταιρειών, είναι το κλειδί για την επιτυχή έκβαση των ΗΙΦ. Σύμφωνα με την Wall Street Journal, το DHHS δεν λαμβάνει μέτρα προστασίας σύμφωνα με την HIPAA. Σύμφωνα με το νόμο, οι εταιρείες υποχρεούνται να ακολουθήσουν όλα τα πρότυπα HIPAA και να υιοθετήσουν τις ίδιες πληροφορίες-χειρισμού πρακτικές που έχουν τεθεί σε ισχύ για την ομοσπονδιακή κυβέρνηση για πολλά χρόνια. Αυτό περιλαμβάνει δύο μέρη i) τυποποιημένη μορφοποίηση των δεδομένων που ανταλλάσσονται με ηλεκτρονικά μέσα και ii) εφαρμογή της ασφάλειας και της προστασίας της ασφάλειας μεταξύ του ιδιωτικού τομέα. Οι ιδιωτικές εταιρείες έχουν υποσχεθεί ότι θα

έχουν "αυστηρές πολιτικές προστασίας της ασφάλειας και των διαδικασιών." Εάν η προστασία και η ασφάλεια δεν αποτελούν μέρος των συστημάτων που έχουν αναπτυχθεί, οι άνθρωποι δεν θα εμπιστεύονται την τεχνολογία. Έτσι, ο ιδιωτικός τομέας γνωρίζει τη σημασία της προστασίας και της ασφάλειας των συστημάτων.

ΚΕΦΑΛΑΙΟ 4

4.1 ANDROID: ON-LINE MEDICAL RECORDS, SERVER

Σε αυτό το κεφάλαιο θα εξετασθεί τι θα πρέπει να έχει η εφαρμογή, μόνο στο κομμάτι της σύνδεσης με τον server και όχι στο γραφικό περιβάλλον χρήσης (GUI). Όπως αναφέρθηκε στα προηγούμενα κεφάλαια, αυτή την στιγμή υπάρχουν πρότυπα και συσκευές όπου μπορούν να υποστηρίξουν τις δυνατότητες που πρέπει να έχει μια εφαρμογή που θα είναι συνεχώς εξαρτημένη από μια βάση που βρίσκεται σε έναν server. Συνοψίζοντας λοιπόν τα προηγούμενα κεφάλαια, η εφαρμογή θα πρέπει να πληροί τις παρακάτω προδιαγραφές:

1. Σύνδεση στο Internet (ή τοπικό δίκτυο),
2. Ταχύτητα,
3. Ασφάλεια,
4. Φιλικότητα προς τον χρήστη.

Για να εξασφαλισθούν οι παραπάνω προδιαγραφές, θα πρέπει να εξετασθούν διάφοροι τρόποι υλοποίησης και να επιλεγθεί ο βέλτιστος. Με αυτό τον τρόπο θα υλοποιηθεί η εφαρμογή με τον καλύτερο δυνατό τρόπο. Τέλος, πρέπει να επαληθευθεί η σωστή λειτουργία της εφαρμογής με παραδείγματα από τους πιθανούς τρόπους χρήσης. Έτσι θα είναι σίγουρο πως για κάθε περίπτωση χρήσης, η εφαρμογή θα λειτουργεί με τον αναμενόμενο τρόπο.

4.2 ΠΡΟΔΙΑΓΡΑΦΗ ΤΩΝ ΑΠΑΙΤΗΣΕΩΝ ΑΠΟ ΤΟ ΛΟΓΙΣΜΙΚΟ ΣΥΣΤΗΜΑ

Στο σημείο αυτό θα αναλυθούν και θα καθορισθούν οι λειτουργικές και μη λειτουργικές απαιτήσεις του προς ανάπτυξη λογισμικού. Οι καθοριστικοί παράγοντες επιλογής και καθορισμού των απαιτήσεων είναι η πλατφόρμα λογισμικού και η συσκευή χρήσης της εφαρμογής καθώς και οι λόγοι ανάπτυξης της εφαρμογής. Για την δημιουργία του ηλεκτρονικού ιατρικού φακέλου πρέπει να αποφασιστεί ποια πεδία είναι αυτά που θα πρέπει να αποθηκεύονται στην βάση δεδομένων, ώστε να θεωρείται πλήρης ο ιατρικός φάκελος ενός ασθενή. Τέλος πρέπει να υλοποιηθεί η εφαρμογή με τέτοιο τρόπο ώστε να παρέχει την κατάλληλη κωδικοποίηση, ώστε να μην είναι δυνατή η υποκλοπή απόρρητων, προσωπικών πληροφοριών.

4.2.1 Λειτουργικές απαιτήσεις:

Οι λειτουργικές απαιτήσεις περιγράφουν τις εργασίες (λειτουργίες) που θα πρέπει να εκτελεί το λογισμικό:

1. Εμφάνιση πληροφοριών ανά κωδικό: Η εφαρμογή θα πρέπει να εμφανίζει πληροφορίες που σχετίζονται με έναν μοναδικό κωδικό που εισάγει ο θεράπων ιατρός.
2. Δυνατότητα αναμονής: Πρέπει να μπαίνει σε αναμονή σε κάθε εισερχόμενη κλήση ή λήψη μηνύματος της φορητής συσκευής.
3. Δυνατότητα αναζήτησης πηγής πληροφοριών: Η εφαρμογή θα δίνει την δυνατότητα στο ιατρό να αναζητήσει ποιος ιατρός έχει κάνει κάθε εισαγωγή ή τροποποίηση των δεδομένων του ιατρικού φακέλου ενός ασθενή.
4. Διαγραφή βάσης δεδομένων: Η εφαρμογή θα δίνει την δυνατότητα στο θεράπων ιατρό να διαγράφει τη βάση δεδομένων.

4.2.2 Μη λειτουργικές απαιτήσεις

Οι μη λειτουργικές απαιτήσεις περιγράφουν χαρακτηριστικά που πρέπει να έχει το λογισμικό τα οποία δεν αφορούν την εκτέλεση κάποιας λειτουργίας από αυτό.

Απαιτήσεις χρήσης: Η εφαρμογή προορίζεται να χρησιμοποιείται με οθόνη αφής. Το λογισμικό πρέπει να υποστηρίζει σύνδεση μέσω WiFi. Πρέπει η εφαρμογή να δίνει δυνατότητα διακοπής για εισερχόμενες κλήσεις ή μηνύματα και να επανέρχεται μετά την ολοκλήρωση της διακοπής.

Φυσική απαίτηση: Η εφαρμογή προορίζεται για χρήση σε φορητή συσκευή.

Απαίτηση επικοινωνίας: Το λογισμικό θα επικοινωνεί με άλλα συστήματα με την χρήση του πρωτοκόλλου TCP/IP καθώς επίσης θα χρησιμοποιεί σύνδεση WLAN.

Απαίτηση αξιοπιστίας: Η εφαρμογή πρέπει να τίθεται και να επανέρχεται από την αναμονή σε χρονικό διάστημα μικρότερο των πέντε δευτερολέπτων.

Απαίτηση σχεδίασης: Η σχεδίαση θα γίνει αποκλειστικά για την πλατφόρμα λογισμικού Android. Η εφαρμογή μας θα χρησιμοποιεί άλλες εφαρμογές για την επικοινωνία και αποθήκευση στην βάση δεδομένων.

Απαίτηση βάσης δεδομένων: Το λογισμικό πρέπει να διατηρεί βάση δεδομένων με όλα τα απαραίτητα στοιχεία του ιατρικού φακέλου.

4.3 ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

Η πλατφόρμα Android υποστηρίζει την βάση δεδομένων SQLite3 η οποία επιτρέπει την δημιουργία σύνθετης συλλογής δεδομένων. Παράλληλα μπορεί να χρησιμοποιηθεί το Διαδίκτυο για την ανάκτηση επιπλέον δεδομένων που έχουμε δημιουργήσει και τηρούμε.

4.3.1 SQLite

Σε αντίθεση με άλλες μορφές βάσεων δεδομένων η συγκεκριμένη προσφέρει αμεσότητα αναζήτησης με ελάχιστη χρονική καθυστέρηση. Η δημιουργία της βάσης δεδομένων συνδέεται άμεσα με την εφαρμογή που την χρησιμοποιεί και αποθηκεύεται σε μαγνητικό δίσκο της φορητής συσκευής.

4.3.2 Διαδίκτυο

Το Διαδίκτυο προσφέρει εναλλακτικό σημείο αποθήκευσης της βάσης δεδομένων που χρειαζόμαστε για την εφαρμογή και μας παρέχει την δυνατότητα δημιουργίας και ανάκτησης πληροφορίας από την μεγάλη σε όγκο βάση δεδομένων που θα δημιουργήσουμε για την εφαρμογή. Η ανάκτηση των πληροφοριών θα επιτυγχάνεται όταν η συσκευή θα είναι σε θέση να συνδεθεί με το Διαδίκτυο.

Για την δημιουργία της βάσης δεδομένων θα χρησιμοποιηθεί η Structured Query Language (SQL) ώστε να υπάρχει άμεση αντιστοιχία με την βάση δεδομένων της φορητής συσκευής και θα χρησιμοποιήσουμε την εφαρμογή MySQL για την δημιουργία και διατήρηση της βάσης δεδομένων.

4.4 ΕΡΓΑΛΕΙΑ ΥΛΟΠΟΙΗΣΗΣ

Για την κατασκευή της εφαρμογής πέρα από την χρήση των εργαλείων που προσφέρει η ίδια η πλατφόρμα και σε συνεργασία με αυτά, θα χρησιμοποιηθούν εργαλεία που άπτονται της πλατφόρμας λογισμικού Android και συνάδουν με την αντικειμενοστραφή προσέγγιση που ακολουθείται για την δημιουργία του ηλεκτρονικού ιατρικού φακέλου. Για την ορθή υλοποίηση της εφαρμογής θα χρησιμοποιηθεί κάποιο εργαλείο σχεδίασης της βάσης δεδομένων, MySQL, PHP, Apache, Java, το πακέτο ανάπτυξης λογισμικού της πλατφόρμας Android και για την καλύτερη και ευκολότερη συγγραφή της εφαρμογής θα συνδυαστεί και

θα χρησιμοποιηθεί το κατάλληλο περιβάλλον σχεδίασης λογισμικού, ώστε να διευκολυνθεί η συγγραφή του κώδικα παράλληλα με την χρήση των εργαλείων που παρέχει η πλατφόρμα.

4.4.1 MySQL:

Η MySQL είναι μια γρήγορη, εύκολη στη χρήση σχεσιακή βάση δεδομένων που χρησιμοποιείται για τις βάσεις δεδομένων σε πολλές τοποθεσίες στο Web. Από την αρχή οι προγραμματιστές είχαν ως κύριο στόχο την ταχύτητα. Για να καταφέρουν να επιτύχουν την επιθυμητή ταχύτητα, οι προγραμματιστές αποφάσισαν να ενσωματώσουν λιγότερες δυνατότητες σε σχέση με τους κυριότερους ανταγωνιστές (Oracle, Sybase). Ωστόσο, ακόμη και αν η MySQL είναι λιγότερο πλήρης συγκρινόμενη με τις εμπορικές λύσεις των ανταγωνιστών της, έχει όλα τα χαρακτηριστικά μιας βάσης δεδομένων που απαιτούνται κατά κύριο λόγο από τους προγραμματιστές. Είναι πιο εύκολη στην εγκατάστασή και τη χρήση της σε σχέση με τις ανταγωνιστικές εμπορικές λύσεις. Η MySQL έχει δυο άδειες χρήσης:

1. **Λογισμικό ανοικτού κώδικα (Open Source):** Η MySQL είναι διαθέσιμη μέσω του GNU GPL (General Public License) χωρίς χρέωση. Έτσι όλοι μπορούν να χρησιμοποιήσουν τη MySQL εντελώς δωρεάν.
2. **Εμπορική άδεια:** Η MySQL είναι διαθέσιμη με μια εμπορική άδεια για όσους δεν προτιμούν την άδεια χρήσης GPL. Εάν ένας developer θέλει να χρησιμοποιήσει τη MySQL ως μέρος μιας εφαρμογής και δεν θέλει να την παρέχει δωρεάν, όπως ορίζεται από την άδεια χρήσης GPL, πρέπει να αγοράσει μια εμπορική άδεια. Οι τιμές είναι πραγματικά πολύ χαμηλές σε σχέση με τις δυνατότητες που προσφέρονται.

Τέλος η MySQL κωδικοποιεί τα δεδομένα με βάση το πρότυπο MDI5, που είναι ένα από τα πιο ασφαλή. Από τα παραπάνω γίνεται κατανοητό πως η MySQL αποτελεί την πλέον κατάλληλη βάση δεδομένων για την εφαρμογή που πρέπει να υλοποιηθεί για αυτή την πτυχιακή εργασία.

4.4.2 PHP:

Η PHP είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML. Ένα από τα πλεονεκτήματα της PHP είναι πως συνεργάζεται πλήρως με την MySQL που αναφέρθηκε παραπάνω και θα αποτελέσει το μέσο για την σύνδεση της εφαρμογής με την MySQL, αφού μέχρι στιγμής δεν υπάρχει ο απαραίτητος driver στο Android για την απευθείας σύνδεση με την MySQL. Σε επόμενο κεφάλαιο θα παρουσιαστεί αναλυτικά ο τρόπος της σύνδεσης αλλά και τα scripts που χρειάστηκε να υλοποιηθούν.

4.4.3 Apache:

Για να μπορέσει να υπάρξει μια βάση δεδομένων σε έναν server και να την είναι προσπελάσιμη μέσω PHP Scripts θα χρειαστεί και η ύπαρξη ενός Web Server. Αυτός που επιλέχθηκε είναι ο Apache HTTP γνωστός και απλά σαν Apache. Είναι ένας από τους δημοφιλέστερους, εν μέρει γιατί λειτουργεί σε διάφορες πλατφόρμες όπως τα Windows, το Linux, το Unix, και το Mac OS X. Διατηρείται τώρα από μια κοινότητα ανοικτού κώδικα με επιτήρηση από το Ίδρυμα Λογισμικού Apache (Apache Software Foundation). Η πρώτη του έκδοση, γνωστή ως NCSA HTTPd, δημιουργήθηκε από τον Robert McCool και κυκλοφόρησε το 1993. Θεωρείται ότι έπαιξε σημαντικό ρόλο στην αρχική επέκταση του ιστού. Ήταν η πρώτη βιώσιμη εναλλακτική επιλογή που παρουσιάστηκε απέναντι στον εξυπηρετητή http της εταιρείας Netscape και από τότε έχει εξελιχθεί στο σημείο να ανταγωνίζεται άλλους εξυπηρετητές βασισμένους στο Unix σε λειτουργικότητα και απόδοση. Από το 1996 ήταν από τους πιο δημοφιλείς. Τον Οκτώβριο του 2007 το μερίδιο του ήταν 47.73% από όλους τους ιστοτόπους.

4.4.4 Java:

Η πλατφόρμα λογισμικού Android [20] υποστηρίζει την γλώσσα προγραμματισμού Java. Παρέχει όλα εκείνα τα εργαλεία για την υποστήριξη εφαρμογών γραμμένων σε αυτή τη γλώσσα και επιπλέον η εικονική μηχανή (Dalvik Virtual Machine [21]) που χρησιμοποιείται από τις φορητές συσκευές για τις οποίες προορίζεται η εφαρμογή υποστηρίζει πλήρως την γλώσσα Java. Επιπροσθέτως για την συγγραφή του κώδικα επιλέχθηκε η αντικειμενοστραφής προσέγγιση. Η Java είναι μια γλώσσα προγραμματισμού που προσφέρει όλες εκείνες τις απαραίτητες μεθόδους και λειτουργίες που είναι απαραίτητες για την συγγραφή κώδικα σύμφωνα με τις αρχές της αντικειμενοστραφούς προσέγγισης. Οι συνθήκες και οι δυνατότητες που προσφέρει στον κατασκευαστή λογισμικού είναι ένα από τα μεγάλα πλεονεκτήματα για την υλοποίηση της παρούσας εφαρμογής. Στην περίπτωση του ηλεκτρονικού ιατρικού φακέλου η επαναχρησιμοποίηση του κώδικα και η συμπαγής γραφή θα βοηθήσει να απαιτεί το δυνατόν μικρότερο χώρο αποθήκευσης η εφαρμογή και αυτό είναι ιδιαίτερα σημαντικό αφού προορίζεται για φορητές συσκευές που δεν χαρακτηρίζονται από ιδιαίτερα μεγάλη αποθηκευτική ικανότητα.

4.4.5 Eclipse:

Όπως αναφέρθηκε και στο δεύτερο κεφάλαιο της παρούσας εργασίας η πλατφόρμα Android διαθέτει για την ανάπτυξη εφαρμογών το Android SDK, δηλαδή το πακέτο ανάπτυξης λογισμικού που περιλαμβάνει όλα εκείνα τα απαραίτητα εργαλεία για την ανάπτυξη εφαρμογών για την συγκεκριμένη πλατφόρμα. Για την συγγραφή της εφαρμογής του ηλεκτρονικού ιατρικού φακέλου επιλέχθηκε να χρησιμοποιηθεί ένα περισσότερο φιλικό και εύχρηστο περιβάλλον ανάπτυξης λογισμικού, η πλατφόρμα σχεδίασης Eclipse [24] και συγκεκριμένα η έκδοση 3.4 Ganymede. Το συγκεκριμένο ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) υποστηρίζεται πλήρως από το Android SDK, το οποίο και παρέχει κατάλληλους συνδέσμους (plug ins) για την άμεση συσχέτιση των δύο αυτών εργαλείων. Με την χρήση του συγκεκριμένου περιβάλλοντος ανάπτυξης λογισμικού αναμένεται να επιτευχθεί η ταχύτερη συγγραφή του κώδικα και η ευκολότερη ανάπτυξη και έλεγχος για την εφαρμογή που πρόκειται να υλοποιηθεί. Η άμεση συσχέτιση με την πλατφόρμα Android και όλα τα εργαλεία της, σε συνδυασμό με την απολύτως κατανοητή και εύχρηστη διεπαφή με τον χρήστη, κάνει το Eclipse χρήσιμο εργαλείο για την επίτευξη του στόχου της πτυχιακής εργασίας και την συγγραφή του κώδικα.

4.5 ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ ΕΡΓΑΣΙΩΝ

Για να επιτευχθεί η σωστή ανάπτυξη της εφαρμογής αυτής της πτυχιακής εργασίας, θα πρέπει ακολουθηθεί ένα διάγραμμα ροής εργασιών, που μέσω αυτού θα γίνει εφικτή η ανάπτυξη της εύκολα, γρήγορα και οργανωμένα.

1. Έρευνα της αγοράς των φορητών συσκευών και των λειτουργικών τους συστημάτων, ώστε να επιλεγθεί το πλέον κατάλληλο.
2. Αναζήτηση πληροφοριών για τον ιατρικό φάκελο και τον ηλεκτρονικό ιατρικό φάκελο.
3. Μελέτη των ηλεκτρονικών φακέλων που υπάρχουν στην αγορά, ώστε να αποφασιστεί τι πρέπει να ενσωματωθεί στην εφαρμογή.
4. Εύρεση των απαραίτητων προγραμμάτων για την υλοποίηση της εφαρμογής.
5. Εύρεση βιβλιογραφίας για την σωστή χρήση των παραπάνω προγραμμάτων.
6. Υλοποίηση κάποιων απλών προγραμμάτων για την εξακρίβωση της σωστής λειτουργίας και συνεργασίας των εφαρμογών που επιλέχθηκαν.
7. Σχεδιασμός της βάσης δεδομένων που θα υπάρχει στον server, εισαγωγή των κατάλληλων πεδίων σύμφωνα με τα αποτελέσματα που εξάχθηκαν από τα βήματα 2 και 3.
8. Υλοποίηση των PHP Scripts που θα χρειαστούν για να γίνει η σύνδεση της παραπάνω βάσης με την συσκευή.
9. Υλοποίηση των κλάσεων της εφαρμογής που θα βρίσκεται στο κινητό για την αξιοποίηση των παραπάνω PHP Scripts.
10. Ενδελεχής έλεγχος της σωστής λειτουργίας όλων των παραπάνω μέσω σχετικού κώδικα.
11. Διόρθωση τυχών προβλημάτων που θα προκύψουν από το βήμα 10.

4.6 ΣΧΕΔΙΑΣΜΟΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Σε αυτή την ενότητα θα παρουσιαστούν αναλυτικά τα πεδία της βάσης δεδομένων και ποια είναι η χρήση τους.

ΙΑΤΡΙΚΟΣ ΦΑΚΕΛΟΣ	
Πεδία	Ανάλυση
Id	Κωδικός ιατρικού φακέλου
Id Ασθενή	Κωδικός ασθενή
Βιογραφικό	Γενικές πληροφορίες
Ατομικό αναμνηστικό	Καταγράφονται όλες οι νόσοι, νοσηλείες, εγχειρήσεις, αλλεργίες. Επίσης, περιλαμβάνει τα δεδομένα των προηγούμενων επισκοπήσεων (κοιλιάς, κ.λ.π.) καθώς και για την πρόσφατη χρήση φαρμάκων ή χορήγηση φαρμακευτικής αγωγής.
Οικογενειακό ιστορικό	Καταγράφονται οι κληρονομικές ασθένειες της οικογένειας του ασθενούς.
Ατομικά προηγούμενα	Πληροφορίες σχετικά με προηγούμενες παθήσεις.

ΕΞΕΤΑΣΗ	
Πεδία	Ανάλυση
Id	-
Id Ασθενή	-
Id Ιατρού	-
Ημερομηνία	-
Κλινική Εξέταση, Δέρματος	-
Κλινική Εξέταση, Κεφαλής Τραχήλου	-
Κλινική Εξέταση, Αναπνευστικού	-
Κλινική Εξέταση, Κυκλοφορικού	-
Κλινική Εξέταση, Πεπτικού	-
Κλινική Εξέταση, Νευρολογικού Συστήματος	-
Συμπτώματα	-
Πιθανή Διάγνωση	-
Παρακλινικός Έλεγχος	Υπόλοιπες εξετάσεις που κρίθηκαν απαραίτητες, π.χ. ακτινογραφίες.
Συνταγογράφηση	-
Τελική Διάγνωση	-

ΙΑΤΡΟΣ	
Πεδία	Ανάλυση
Id	-
Επώνυμο	-
Όνομα	-
Ημερομηνία γέννησης	-
Ηλικία	-
Ειδικότητα	-
Οικογενειακή κατάσταση	-
Φύλο	-
Εθνικότητα	-
Καταγωγή	-
Όνομα πατρός	-
Όνομα μητρός	-
Διαμονή, Οδός	-
Διαμονή, Αριθμός	-
Διαμονή, Περιοχή	-
Διαμονή, Ταχυδρομικός Κώδικας	-
Διαμονή, Τηλέφωνο	-
Διαμονή, Fax	-
Κινητό	-
e-Mail	-
Ασφάλεια, Κωδικός	-
Ασφάλεια, Περιγραφή	-
Ασφάλεια, Αριθμός Συμβολαίου	-
Ασφάλεια, Ημερομηνία Από	-
Ασφάλεια, Ημερομηνία Έως	-
ΑΦΜ	-
ΔΟΥ	-
Παρατηρήσεις	-

ΜΗΤΡΩΟ ΑΣΘΕΝΩΝ

Πεδία	Ανάλυση
Id	-
Επώνυμο	-
Όνομα	-
Ημερομηνία γέννησης	-
Ηλικία	-
Ομάδα Αίματος	-
Μορφωτικό Επίπεδο	-
Οικογενειακή κατάσταση	-
Φύλο	-
Εθνικότητα	-
Καταγωγή	-
Τόπος Γέννησης	-
Όνομα πατρός	-
Όνομα μητρός	-
Διαμονή, Οδός	-
Διαμονή, Αριθμός	-
Διαμονή, Περιοχή	-
Διαμονή, Ταχυδρομικός Κώδικας	-
Διαμονή, Τηλέφωνο	-
Διαμονή, Fax	-
Διαμονή, e-Mail	-
Κινητό	-
Ασφάλεια, Κωδικός	-
Ασφάλεια, Περιγραφή	-
Ασφάλεια, Αριθμός Συμβολαίου	-
Ασφάλεια, Ημερομηνία Από	-
Ασφάλεια, Ημερομηνία Έως	-
Επάγγελμα, Περιγραφή	-
Επάγγελμα, Περιοχή	-
Επάγγελμα, Τηλέφωνο	-
Επάγγελμα, Fax	-
Επάγγελμα, e-Mail	-
Περιβάλλον Εργασίας	-
ΑΦΜ	-
ΔΟΥ	-
Παρατηρήσεις	-

ΚΕΦΑΛΑΙΟ 5

5.1 AMRS v1.0

Σε αυτή την ενότητα, αφού αναλύθηκαν αυτά που χρειάζονται για να είναι πετυχημένη μια τέτοια εφαρμογή σήμερα, θα παρουσιαστεί ο τρόπος με τον οποίο χρησιμοποιήθηκαν όλα τα παραπάνω εργαλεία ώστε να υλοποιηθεί. Συγκεκριμένα, σε αυτή την πτυχιακή εργασία, θα εξετασθεί πως ακριβώς θα «χτίσουμε» την βάση δεδομένων και πως θα επιτευχθεί η επικοινωνία της πλατφόρμας Android με αυτήν. Από τις προηγούμενες ενότητες γίνεται αντιληπτό ότι μια τέτοια εφαρμογή θα πρέπει να είναι συνεχώς εξαρτημένη από έναν Server με ασφάλεια, ταχύτητα και αξιοπιστία. Επιπλέον πρέπει να διασφαλιστεί ότι το λειτουργικό σύστημα στο οποίο θα λειτουργεί ο Server θα είναι σε θέση να δουλεύει 24 ώρες την ημέρα και 7 μέρες την εβδομάδα. Τέλος θα πρέπει το πρόγραμμα που θα χρησιμοποιεί ο χρήστης να είναι όσο το δυνατόν πιο απλό στην χρήση και να περιοριστεί ο αριθμός των φορών που θα πρέπει να πληκτρολογεί κάτι, δηλαδή τα περισσότερα να γίνονται με απλά clicks μέσα από σχετικές προκαθορισμένες επιλογές. Παρακάτω θα παρουσιαστεί ο τρόπος με τον οποίο υλοποιήθηκε η εφαρμογή αυτής της πτυχιακής εργασίας, σύμφωνα με το διάγραμμα εργασιών (ενότητα 4.5).

5.2 ΕΓΚΑΤΑΣΤΑΣΗ ΤΩΝ ΑΠΑΡΑΙΤΗΤΩΝ ΠΡΟΓΡΑΜΜΑΤΩΝ & ΒΙΒΛΙΟΓΡΑΦΙΑ

Σε αυτή την ενότητα, θα αναπτυχθούν τα βήματα 4 & 5 από το διάγραμμα ροής με αναλυτικό τρόπο:

4. Εύρεση των απαραίτητων προγραμμάτων για την υλοποίηση της εφαρμογής.
5. Εύρεση βιβλιογραφίας για την σωστή χρήση των παραπάνω προγραμμάτων.

Τα βήματα 1-3:

1. Έρευνα της αγοράς των φορητών συσκευών και των λειτουργικών τους συστημάτων, ώστε να επιλεγεί το πλέον κατάλληλο.
2. Αναζήτηση πληροφοριών για τον ιατρικό φάκελο και τον ηλεκτρονικό ιατρικό φάκελο.
3. Μελέτη των ηλεκτρονικών φακέλων που υπάρχουν στην αγορά, ώστε να αποφασιστεί τι πρέπει να ενσωματωθεί στην εφαρμογή.

παραλείπονται, διότι έγινε λεπτομερής αναφορά σε αυτά στα προηγούμενα κεφάλαια.

5.2.1 Τεχνικά χαρακτηριστικά

Τύπος επεξεργαστή:	Intel Core 2 Duo Processor (CPU)
Πυρήνες:	Dual Core (2)
Ταχύτητα επεξεργαστή:	2.26GHz
Μνήμη Ram:	4 GB DDR2
Σκληρός δίσκος:	10 GB [Τόσο εκχωρήθηκε στο λειτουργικό σύστημα Linux]

Κινητό: Τι στιγμή που αποφασίστηκε το θέμα της πτυχιακής εργασίας, στην Ελλάδα δεν υπήρχε κάποιο κινητό που να έχει εγκατεστημένο το Android, έτσι επιλέχτηκε ένα παραπλήσιον χαρακτηριστικών και έγινε μεταφορά του σε σύστημα Windows Mobile 6.1 μέσω emulator. Το κινητό αυτό είναι το HTC Touch Pro το οποίο έχει ίδια χαρακτηριστικά με το HTC Magic.



5.2.2 Λειτουργικό σύστημα

Το λειτουργικό σύστημα που επιλέχτηκε είναι η έκδοση Linux, Ubuntu 9.04 x64. Επιλέχτηκε μια έκδοση Linux επειδή είναι δωρεάν, έτσι μειώνεται το κόστος για την εγκατάστασή του σε μια επιχείρηση ή νοσοκομείο και συγκεκριμένα η διανομή Ubuntu γιατί είναι αυτή με την καλύτερη υποστήριξη. Επιπλέον η Ubuntu είναι ένα λειτουργικό το οποίο έχει πολύ μικρές απαιτήσεις συστήματος κάτι το οποίο σημαίνει πως γίνεται να εγκατασταθεί και σε συστήματα με χαμηλό κόστος. Πέρα από τα παραπάνω το Linux παρέχει την μέγιστη δυνατή ασφάλεια που μπορεί να έχει σήμερα ένα λειτουργικό σύστημα κάτι το οποίο αν συνδυαστεί με τις σωστές εφαρμογές και υλοποιήσεις (MySQL Server, Apache, WiFi με ασφάλεια

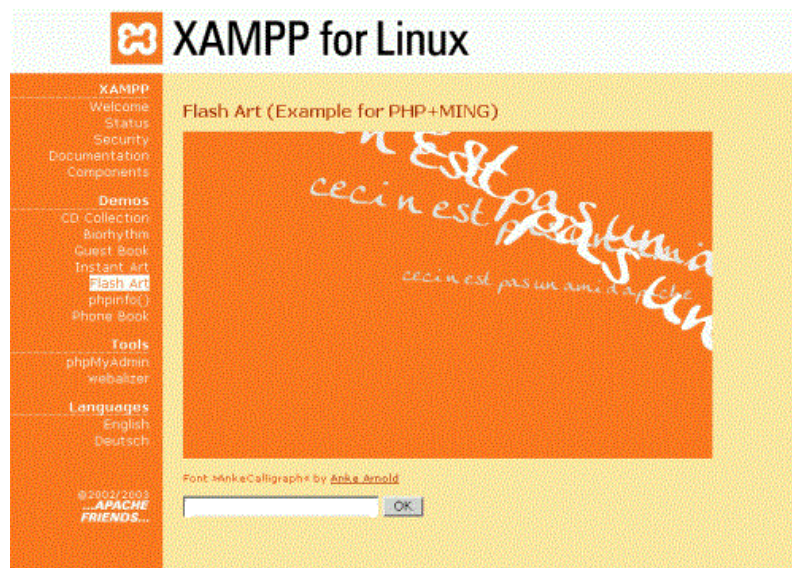
WPA2) μας δίνει την πλήρη ασφάλεια, η οποία είναι απαραίτητη για μια εφαρμογή που βασίζεται σε μια On-Line βάση δεδομένων, πόσο μάλλον αν αυτή περιέχει τους ιατρικούς φακέλους κάποιων ατόμων. Τέλος το Linux αποτελεί την καλύτερη επιλογή, διότι όπως αναφέρθηκε, είναι Open Source, κάτι το οποίο σημαίνει πως εξελίσσεται πολύ γρήγορα και είναι εύκολο τα όποια προβλήματα αντιμετωπίζει να λυθούν με ακρίβεια και άμεσα.



Download Ubuntu: <http://www.ubuntu.com/getubuntu/download>

5.2.3 Server

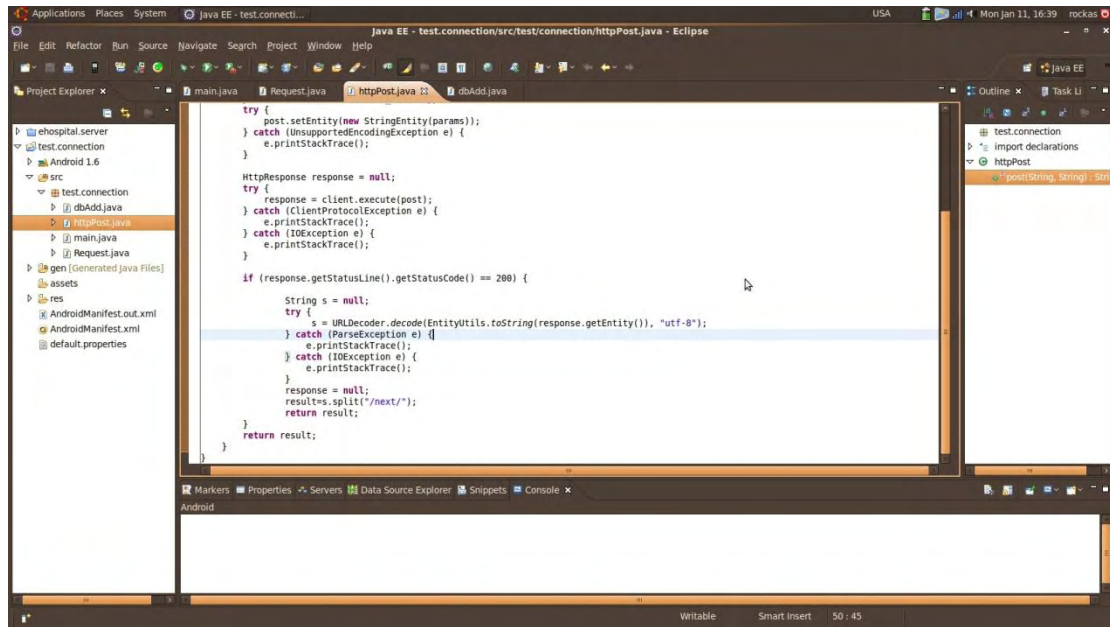
Για Server επιλέχτηκε το πακέτο Xampp το οποίο περιλαμβάνει Apache, PHP και MySQL. Συγκεκριμένα η έκδοση xampp 1.7.2 για Linux.



Download xampp: <http://www.apachefriends.org/en/xampp-linux.html#374>

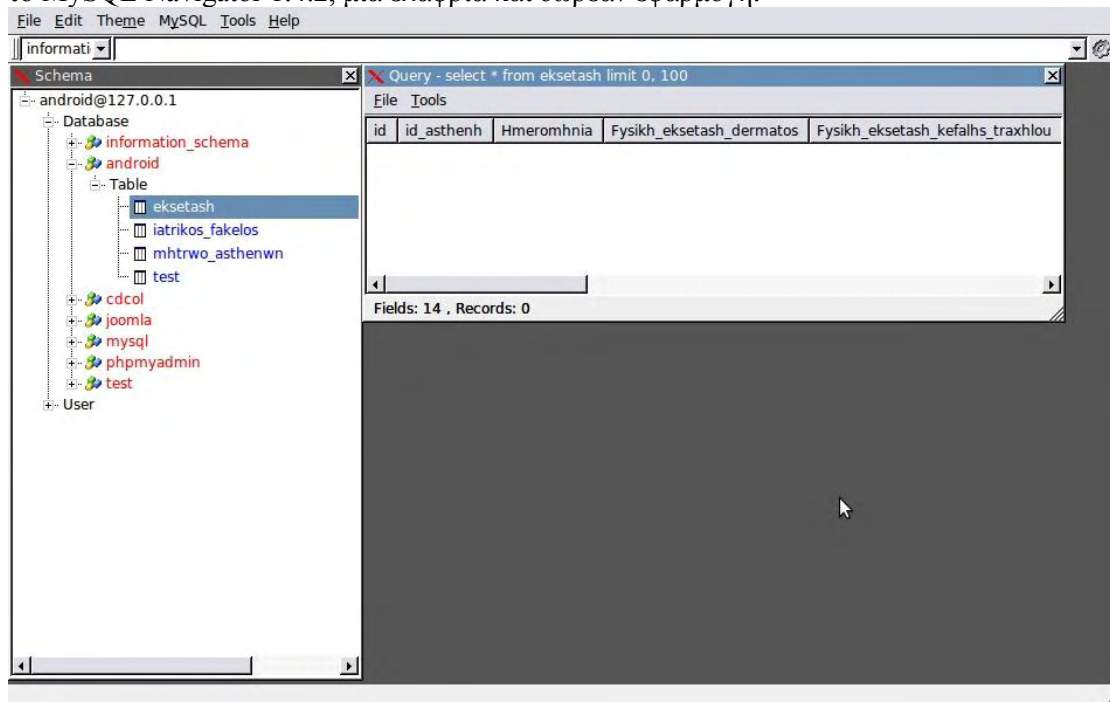
5.2.4 Eclipse

Όπως αναφέρθηκε στις προηγούμενες ενότητες για τον προγραμματισμό της εφαρμογής επιλέχτηκε το Eclipse σε συνδυασμό με το Android SDK. Συγκεκριμένα εγκαταστάθηκε το Eclipse μέσω του Synaptic Package Manager. Το Android SDK που χρησιμοποιήθηκε είναι αυτό της έκδοσης 1.6.



5.2.5 MySQL Navigator

Το πρόγραμμα που χρησιμοποιήθηκε για την διαχείριση της βάσης δεδομένων MySQL είναι το MySQL Navigator 1.4.2, μια ελαφριά και δωρεάν εφαρμογή.



5.2.6 Βιβλιογραφία

Όπως ήδη αναφέρθηκε σε προηγούμενα κεφάλαια η πλατφόρμα Android είναι καινούργια και για αυτό δεν υπήρχαν πολλά έτοιμα παραδείγματα διαθέσιμα. Έτσι ήταν απαραίτητη η εύρεση εκείνων των βιβλίων [1-7] που θα περιείχαν όλες τις πληροφορίες ώστε να μπορεί κάποιος να είναι σε θέση να μάθει να προγραμματίζει εφαρμογές για την πλατφόρμα Android.

5.3 ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Σε αυτή την ενότητα θα αναπτυχθούν αναλυτικά τα βήματα 6 – 9 από το διάγραμμα ροής:

6. Υλοποίηση κάποιων απλών προγραμμάτων για την εξακρίβωση της σωστής λειτουργίας και συνεργασίας των εφαρμογών που επιλέχθηκαν.
7. Σχεδιασμός της βάσης δεδομένων που θα υπάρχει στον server, εισαγωγή των κατάλληλων πεδίων σύμφωνα με τα αποτελέσματα που εξάχθηκαν από τα βήματα 2 και 3.
8. Υλοποίηση των PHP Scripts που θα χρειαστούν για να γίνει η σύνδεση της παραπάνω βάσης με την συσκευή.
9. Υλοποίηση των κλάσεων της εφαρμογής που θα βρίσκεται στο κινητό για την αξιοποίηση των παραπάνω PHP Scripts.

5.3.1 Πρώτα βήματα

Αρχικός στόχος ήταν να υλοποιηθεί μια απλή εφαρμογή στο Android όπου θα τύπωνε ένα απλό μήνυμα, όπως το κλασικό *Hello World!!!* που διδάσκεται στην αρχή εκμάθησης κάποιας γλώσσας προγραμματισμού.

```
package com.example.helloandroid;

import android.app.Activity;
import android.os.Bundle;

public class HelloAndroid extends Activity {
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        TextView text = new TextView(this);
        String result = " Hello World!!!";
        text.setText(result);
        text.setTextColor(Color.RED);
        setContentView(text);
    }
}
```

5.3.2 Κύριος στόχος

Κύριος στόχος ήταν να επιτευχθεί η σύνδεση μεταξύ του Android και του Server. Αυτό ήταν το πιο δύσκολο κομμάτι διότι η πλατφόρμα Android δεν παρέχει ακόμη την δυνατότητα να συνδέεται μέσω ενός driver [π.χ. *odbc driver*] με μια απομακρυσμένη βάση δεδομένων. Έτσι έπρεπε να υλοποιηθεί αυτή η σύνδεση από την αρχή. Ο μόνος τρόπος που υπήρχε ήταν η υλοποίηση ενός php script όπου θα δεχόταν ορίσματα μέσω HTTPpost και θα «τύπωνε» τα αποτελέσματα στην οθόνη της εφαρμογής. Για να επιτευχθεί αυτό υλοποιήθηκε μια συνάρτηση Java στο Android η οποία στέλνει τα ερωτήματά μέσω HTTPpost και με τον ίδιο

τρόπο τα λαμβάνει ώστε να μπορεί μετά η εφαρμογή να τα αξιοποιήσει αναλόγως. Ο κώδικας είναι ο παρακάτω:

```
public class httpPost extends Activity {
    public static String[] post(String server, String params){
        String[] result = null;
        HttpParams connectionParams = new BasicHttpParams();
        HttpConnectionParams.setSoTimeout(connectionParams, 22000);
        HttpConnectionParams.setConnectionTimeout(connectionParams, 22000);
        HttpClient client = new DefaultHttpClient(connectionParams);
        HttpPost post = new HttpPost(server);
        post.setHeader(HTTP.CONTENT_TYPE, "application/x-www-form-
urlencoded");
        try {
            post.setEntity(new StringEntity(params));
        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        }

        HttpResponse response = null;
        try {
            response = client.execute(post);
        } catch (ClientProtocolException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
        if (response.getStatusLine().getStatusCode() == 200) {

            String s = null;
            try {
                s =
                URLDecoder.decode(EntityUtils.toString(response.getEntity()), "utf-8");
            } catch (ParseException e) {
                e.printStackTrace();
            } catch (IOException e) {
                e.printStackTrace();
            }
            response = null;
            return s;
        }
        return result;
    }
}
```

Τα προβλήματα που προέκυψαν ήταν αρκετά:

1. Έπρεπε να υλοποιηθεί ένα php script το οποίο θα μπορούσε να εκτελεί τα MySQL Queries που θα λάμβανε από το HTTPpost,
2. Το php script που υλοποιήθηκε δεν υποστήριζε ελληνική κωδικοποίηση,
3. Έπειτα έπρεπε ο προγραμματιστής κάθε φορά να γράφει τα δικά του Queries κάτι το οποίο δεν θα ήταν καθόλου εύκολο στην χρήση. Έτσι έπρεπε να υλοποιηθεί για κάθε πίνακα ειδική συνάρτηση για την προσθήκη δεδομένων σε αυτόν &
4. Το php script επέστρεφε ένα αποτέλεσμα μόνο, δηλαδή για να προβληθούν 2 διαφορετικές εγγραφές (π.χ. ηλικία και φύλο) για έναν ασθενή έπρεπε να καλείται 2 φορές η συνάρτηση, για 3 εγγραφές 3 φορές και ούτω καθ' εξής, έτσι έπρεπε να βρεθεί ένας τρόπος για μια πιο εύχρηστη προσέγγιση.

5.3.3 Αντιμετώπιση προβλημάτων

Σε αυτήν την ενότητα θα παρουσιαστούν οι τρόποι με τους οποίους αντιμετωπίστηκαν τα παραπάνω προβλήματα. Μπροστά θα βρίσκεται ο αριθμός του προβλήματος και δίπλα ο τρόπος που αντιμετωπίστηκε.

1. PHP Scripts

Το php που χρειαζόταν να υλοποιηθεί έπρεπε να διαβάζει τα Queries από το HTTPpost τα οποία μπορεί να περιείχαν κενά. Το πρόβλημα βρισκόταν στην δυνατότητα του HTTPpost να επεξεργαστεί τα κενά. Επειδή το HTTPpost στέλνει μόνο ένα string δεν γινόταν να υπάρχουν κενά κι έτσι μετέτρεπε τα κενά σε %20. Δηλαδή την φράση:

```
Δοκιμαστικό μήνυμα.
```

Την έκανε:

```
Δοκιμαστικό%20μήνυμα.
```

Το παραπάνω πρόβλημα λύθηκε χρησιμοποιώντας την php εντολή `str_replace("αρχικό", "τελικό", μεταβλητή);` και συγκεκριμένα για την εφαρμογή:

```
$query = str_replace("%20", " ", $input);
```

Όπου \$query η μεταβλητή που θα αποθηκευτεί το ερώτημά, %20 το κείμενο που θέλουμε να αντικαταστήσουμε με το κενό και \$input είναι η μεταβλητή στην οποία αποθηκεύτηκε το string από το HTTPpost. Πέρα από αυτά το php script έπρεπε να εξυπηρετεί 2 σκοπούς:

- i. Εκτέλεση των ερωτημάτων και επιστροφή των αποτελεσμάτων στο πρόγραμμα &
- ii. Προσθήκη δεδομένων στους πίνακες.

Έτσι υλοποιήθηκαν 2 scripts, ένα για κάθε σκοπό.

2. Ελληνική κωδικοποίηση

Το πρόβλημα με την ελληνική κωδικοποίηση λύθηκε μετά από τις παρακάτω τροποποιήσεις:

- i. Από το πρόγραμμα στο php script με την χρήση της java εντολής `URLLEncoder.encode(string, "UTF-8");`.
- ii. Από το php script στην MySQL κάνοντας χρήση της MySQL εντολής `mysql_query("SET CHARACTER SET 'utf8'");` και ορίζοντας τον πίνακά σε `charset utf-8`.
- iii. Την ανάποδη διαδικασία από το βήμα 2 ρυθμίζοντας στο `php.ini` το `default_charset = utf-8`.
- iv. Από το php script στην εφαρμογή (άρα και στην οθόνη) με χρήση της java εντολής `URLDecoder.decode(string, "utf-8");`.

3. Συναρτήσεις

Υλοποιήθηκαν συναρτήσεις για κάθε πίνακα της βάσης δεδομένων ώστε ο προγραμματιστής όταν χρειάζεται να αποθηκεύσει κάτι σε έναν πίνακα, το μόνο που έχει να κάνει είναι να δώσει τιμές στα ορίσματα. Για παράδειγμα:

Για το:

```
public static void dbAddFolder(Integer Id_asthenh, String Biografiko,
String atomiko_anamnhstiko, String oikogeneiako_istoriko, String
atomika_prohgoumena){
    try {
        Biografiko = URLLEncoder.encode(Biografiko, "utf-
8");
        atomiko_anamnhstiko
URLLEncoder.encode(atomiko_anamnhstiko, "utf-8");
        oikogeneiako_istoriko
URLLEncoder.encode(oikogeneiako_istoriko, "utf-8");
```

```

        atomika_prohgoumena =
        URLEncoder.encode(atomika_prohgoumena, "utf-8");
    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    }
    httpPost.post("http://192.168.1.5/AddData.php",
    "input=INSERT INTO android.iatrikos_fakelos(id, Id_asthenh,
    Biografiko, atomiko_anamnhstiko, oikogeneiako_istoriko,
    atomika_prohgoumena) VALUES(NULL, "+Id_asthenh+", "\"+Biografiko+\"",
    "\"+atomiko_anamnhstiko+\"", "\"+oikogeneiako_istoriko+\"",
    "\"+atomika_prohgoumena+\"");
    }
}

```

Δίνει:

```

dbAdd.dbAddFolder(1, "Υγιέστατος", "Συκώτι", "Καρκίνος μαστού",
"Πνευμονία");

```

4. Επιστροφή παραπάνω από ένα αποτέλεσμα ταυτόχρονα

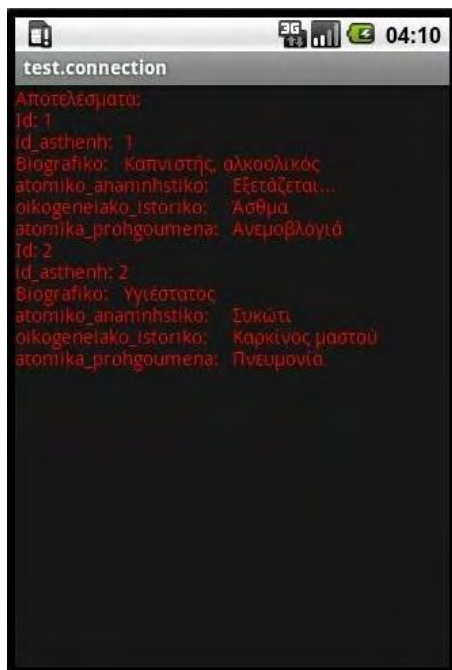
Όπως αναφέρθηκε παραπάνω το php script επιστρέφει ως αποτέλεσμα μόνο ένα string και για αυτό δεν γίνεται να ληφθούν ταυτόχρονα 2 αποτελέσματα. Για να αντιμετωπιστεί αυτό τα αποτελέσματα επιστρέφονται μαζί σε ένα string και διαχωρίζονται με ένα σύμβολο όπως ακριβώς και στην περίπτωση με το %20. Συγκεκριμένα τα αποτελέσματα διαχωρίζονται με την ακολουθία /next/ κι έπειτα ο κώδικας μέσα στο HTTPpost που λαμβάνει τα δεδομένα τα χωρίζει σε έναν πίνακα όπως πρέπει. Έστω ότι έχουμε τον πίνακα request[]. Στο request[1] μπαίνει το id, στο request[2] μπαίνει το Όνομα κλπ. Ο προγραμματιστής γνωρίζει από πριν ότι το request[1] είναι για το id, το request[2] για το Όνομα, οπότε τα χρησιμοποιεί και αναλόγως. Επιπλέον, έστω ότι έχουμε έναν πίνακα με 4 ασθενείς που αποθηκεύουμε 4 πληροφορίες για αυτούς τότε ο πίνακας θα έχει διάσταση 1x16 και από 1 - 4 θα έχουμε τα πεδία για τον πρώτο ασθενή, από το 5 - 8 για τον δεύτερο, από το 9 - 12 για τον τρίτο και από το 13 - 16 για τον τέταρτο. Οπότε, αφού και πάλι ο προγραμματιστής έχει το προνόμιο να γνωρίζει τι πεδία έχει κάθε πίνακας, κατασκευάζει μια επανάληψη για τα αποτελέσματα από 1 - 4, μετά 5 - 8 κλπ. Απόρροια αυτών είναι πως το αποτέλεσμα της request δεν είναι ανάγκη να χρησιμοποιείται με έναν γενικό τρόπο πάντα, μπορεί να προσαρμοστεί στις «ανάγκες» του κάθε πίνακα και έτσι να πραγματοποιούνται αναζητήσεις όσων δεδομένων απαιτεί η περίπτωση. Η υλοποίηση είναι η εξής:

```

//Προσαρμογή αναζήτησης για τον πίνακα android.iatrikos_fakelos.
String[] Fakelos =
{"Id","id_asthenh","Biografiko","atomiko_anamnhstiko","oikogeneiako_istoriko", "atomika_prohgoumena"};
//Εκτύπωση δεδομένων.
result = "Αποτελέσματα: ";
for(i=0; i<request.length; i++){
    result = result + "\n" + Fakelos[i%6] + ":\t" + request[i];
//Εδώ χρησιμοποιώ το i modulus 6 (i%6) για να κάνω κάθε 6 φορές κύκλο
στον πίνακα Fakelos.
}
text.setText(result);

```

Αποτέλεσμα:



Σύμφωνα με τα παραπάνω, τα php scripts τελικά παίρνουν την παρακάτω μορφή:

AddData.php:

```
<html>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<?php
$user = "android";
$user_password = "13524";
$db_name = "android";
mysql_connect("localhost", $user, $user_password) or
die(mysql_error());
mysql_select_db($db_name) or die(mysql_error());
mysql_query("SET CHARACTER SET 'utf8'");
$input = $_REQUEST["input"];
$query = str_replace("%20", " ", $input);
$rs = mysql_query($query);
?>
</html>
```

- 2: Ελληνική κωδικοποίηση.
- 4: Ορίζεται το όνομα χρήστη για την σύνδεση στην βάση.
- 5: Τον κωδικό του χρήστη.
- 6: Το όνομα της βάσης.
- 7: Προσπάθεια σύνδεσης στον Server.
- 8: Προσπάθεια σύνδεσης στην βάση.
- 9: Ορίζεται ελληνική κωδικοποίηση για την MySQL.
- 10: Αποθηκεύεται στην μεταβλητή \$input το string από το HTTPpost.
- 11: Μετατροπή του %20 σε κενό και αποθήκευσή του στην μεταβλητή \$query.
- 12: Εκτέλεση του ερωτήματος.

dbRequest.php:

```
<?php
$user = "android";
$user_password = "13524";
$db_name = "android";
mysql_connect("localhost", $user, $user_password) or
die(mysql_error());
mysql_select_db($db_name) or die(mysql_error());
mysql_query("SET CHARACTER SET 'utf8'");
$input = $_REQUEST["input"];
$query = str_replace("%20", " ", $input);
$rs = mysql_query($query);
while ($get_info = mysql_fetch_row($rs)){
    foreach ($get_info as $field)
        print $field."/next/";
}
?>
```

1-10: Ότι και στο προηγούμενο.

11: Όσο δεν έχουν τελειώσει τα αποτελέσματα από το ερώτημα,

12: «Τύπωνε» το πεδίο και δίπλα του το string */next/*.

5.3.4 Έλεγχος του προγράμματος

Στις προηγούμενες ενότητες αναφέρθηκε ο τρόπος με τον οποίο υλοποιήθηκαν τα PHP Scripts, τις κλάσεις για τους πίνακες αλλά και την συνάρτηση η οποία πραγματοποιεί το HTTPpost. Σε αυτή την ενότητα θα μελετηθεί ο κώδικας που υλοποιήθηκε ώστε να γίνει ενδελεχής έλεγχος της σωστής λειτουργίας της εφαρμογής [διάγραμμα ροής – βήμα 10]. Ο κώδικας πρέπει να ελέγχει την σωστή αποθήκευση των δεδομένων στην βάση αλλά και την ανάκτησή τους από αυτήν.

```
1. package test.connection;
2. import android.app.Activity;
3. import android.graphics.Color;
4. import android.os.Bundle;
5. import android.widget.TextView;
6. public class main extends Activity {
7. public void onCreate(Bundle savedInstanceState){
8. super.onCreate(savedInstanceState);
9. TextView text = new TextView(this);
10. String result = "Welcome!!! :-)";
11. String request[] = null;
12. Integer i=1;
13. text.setText(result);
14. text.setTextColor(Color.RED);
15. setContentView(text);
16. //Προσθήκη δεδομένων στον πίνακα android.test.
17. //dbAdd.dbAddTest("I am τυχερός!!!");
18. //Προσθήκη δεδομένων στον πίνακα android.mhtrwo_asthenwn.
19. //dbAdd.dbAddPatient("Τυχάλας", "Δημήτρης", "1988-02-24", 22, "Ο
θετικό", "Πανεπιστήμιο", "Αγαμος", "Ανδρας", "Ελληνας", "Σέρρες",
"Θεσσαλονίκη", "Γεώργιος", "Ελένη", "Πόντιου", 17, "Σέρρες", 62125,
223104555, 13584, 942620616, "tichalas@gmail.com", 13572468,
"Ατομική", 248, "2009-03-24", "20-03-24", "Android Developer",
"Internet", 13524, 2486135, "dtyxalalas@ucg.gr", "Internet", 2411235,
"Νιγρίτας", "Τίποτα απολύτως");
20. //Προσθήκη δεδομένων στον πίνακα android.iatrikos_fakelos.
```

```

21. //dbAdd.dbAddFolder(1, "Υγιέστατος", "Συκώτι", "Καρκίνος μαστού",
    "Πνευμονία");

22. //Αναζήτηση δεδομένων στους πίνακες της βάσης android.
23. request = Request.dbRequest("*", "iatrikos_fakelos", "");

24. //Προσαρμογή αναζήτησης για τον πίνακα android.iatrikos_fakelos.
25. String[] Fakelos =
    {"Id", "id_asthenh", "Biografiko", "atomiko_anamnhstiko", "oikogeneiako_is
    toriko", "atomika_prohgoumena"};

26. //Εκτύπωση δεδομένων.
27. result = "Αποτελέσματα: ";
28. for(i=0; i<request.length; i++){
29. result = result + "\n" + Fakelos[i%6] + ":\t" + request[i]; //Εδώ
    χρησιμοποιώ το i modulus 6 (i%6) για να κάνω κάθε 6 φορές κύκλο στον
    πίνακα Fakelos.
30. }
31. text.setText(result);
32. // for(i=1; i<request.length; i++){
33. // result = result + "\n" + i.toString() + ":\t" +
    request[i];
34. // }
35. // text.setText(result);
36. }
37. }

```

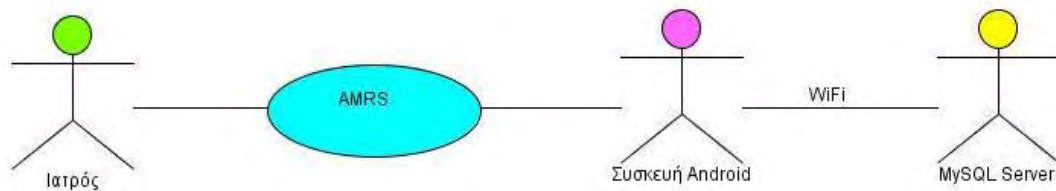
Στο παραπάνω τμήμα κώδικα παρατηρείται πως γίνεται η χρήση των συναρτήσεων για την προσθήκη δεδομένων στον πίνακα [γραμμές 16 – 21], πως πραγματοποιείται μια αναζήτηση σε έναν συγκεκριμένο πίνακα [γραμμή 23], τον τρόπο με τον οποίο προσαρμόζεται η αναζήτηση ώστε να αποθηκεύσει κάποια συγκεκριμένα πεδία [γραμμή 25] και πως εμφανίζονται τα αποτελέσματα αυτά στην οθόνη το κινητού [γραμμές 27 – 36].

ΚΕΦΑΛΑΙΟ 6

6.1 ΑΝΑΛΥΣΗ ΠΕΡΙΠΤΩΣΕΩΝ ΧΡΗΣΗΣ

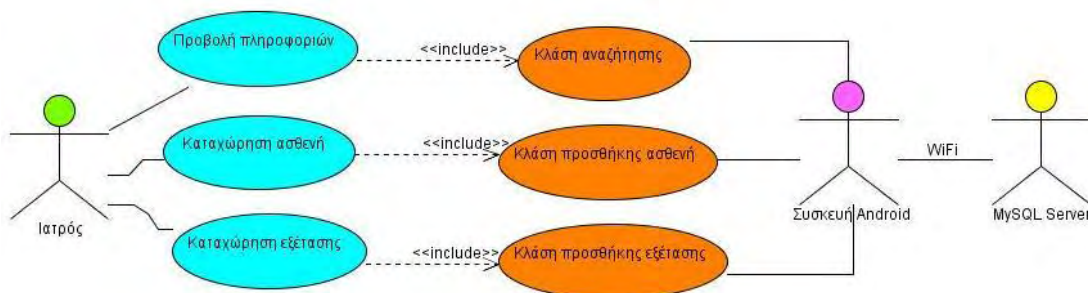
Σε αυτό το κεφάλαιο θα αναγνωρισθούν οι περιπτώσεις χρήσης και οι χειριστές του συστήματος. Η ανάλυσή θα φτάσει στο δεύτερο επίπεδο ανάπτυξης αφού η προς ανάπτυξη εφαρμογή δεν απαιτεί μεγαλύτερη ανάπτυξη λόγω του μικρού μεγέθους της. Στη συνέχεια σύμφωνα με το διάγραμμα περιπτώσεων χρήσης θα αναλυθούν λεκτικά και σύμφωνα με τα μοντέλα ανάλυσης οι περιπτώσεις χρήσης. Για την δημιουργία των διαγραμμάτων θα χρησιμοποιηθεί η γλώσσα αντικειμενοστραφούς μοντελοποίησης Unified Modeling Language (UML) [26] η οποία αποτελεί σήμερα μια από τις πιο γνωστές γλώσσες σχεδίασης λογισμικού. Στηρίζεται στην οπτική απεικόνιση μέσω συγκεκριμένων διαγραμμάτων, παρέχοντας παράλληλα ένα πλαίσιο προδιαγραφών, τεκμηρίωσης και κατασκευής των προϊόντων της διαδικασίας ανάπτυξης λογισμικού.

Η UML υποστηρίζει διαγράμματα δομής, συμπεριφοράς και αλληλεπίδρασης με κύριο χαρακτηριστικό την ομαλή μετάβαση από διάγραμμα σε διάγραμμα για τον πλήρη σχεδιασμό του λογισμικού.



Σχήμα 6: Διάγραμμα Περιπτώσεων Χρήσης Πρώτου Επιπέδου

Το παραπάνω διάγραμμα προδιαγράφει κυρίως τους χρήστες του συστήματος και είναι πολύ απλό αφού έχει σχεδιαστεί μόνο η γενική περίπτωση χρήσης του AMRS. Στο επόμενο διάγραμμα που ακολουθεί θα αναπτυχθεί η περίπτωση χρήσης σε πιο συγκεκριμένες υποπεριπτώσεις.



Σχήμα 7: Διάγραμμα Περιπτώσεων Χρήσης Δεύτερου Επιπέδου

Στο δεύτερο επίπεδο όπως φαίνεται και στο παραπάνω σχήμα υπάρχουν τρεις βασικές περιπτώσεις χρήσεις οι οποίες αφορούν την αλληλεπίδραση με τον χρήστη.

Δεχόμαστε πως με την εκκίνηση του προγράμματος εμφανίζονται κατηγορίες πληροφοριών από τις οποίες έχει την δυνατότητα να επιλέξει ο χρήστης και έτσι δημιουργείται η πρώτη περίπτωση χρήσης 'προβολή πληροφοριών'.

Η επόμενη κύρια περίπτωση χρήσης είναι η 'Καταχώρηση ασθενή' η οποία είναι μια από τις πιο βασικές, διότι για κάθε ασθενή πρέπει οπωσδήποτε να δημιουργηθεί ένας φάκελος με κάποιες βασικές πληροφορίες για αυτόν.

Η τρίτη βασική περίπτωση χρήσης αφορά αυτήν την οποία θα αποτελεί βάση καθημερινής χρήσης από τους ιατρούς, η 'προσθήκη εξέτασης', διότι σε συνδυασμό με την προηγούμενη θα δημιουργείται ένας ολοκληρωμένος ιατρικός φάκελος.

Θα ακολουθήσει ανάλυση των περιπτώσεων χρήσης σύμφωνα με τα αντίστοιχα μοντέλα περιγραφής.

6.1.1 Πρώτη περίπτωση χρήσης

Τίτλος περίπτωσης χρήσης

Προβολή πληροφοριών

Σύντομη περιγραφή

Με την εκκίνηση της εφαρμογής εμφανίζεται το βασικό μενού που αποτελείται από πεδία πληροφοριών. Ο χρήστης επιλέγει κάποιο από αυτά για να προβληθούν αντίστοιχες πληροφορίες.

Χειριστές

Ο ιατρός.

Ροή γεγονότων

Βασική Ροή

Η εφαρμογή εμφανίζει τα πεδία εισαγωγής.

Ο ιατρός προσθέτει τις κατάλληλες πληροφορίες προς αναζήτηση.

Η εφαρμογή αποστέλλει τα δεδομένα στο σχετικό php script.

Το php script αποθηκεύει τα δεδομένα στην βάση δεδομένων.

Ο χρήστης επιλέγει επιστροφή στο βασικό μενού.

Η εφαρμογή επιστρέφει στο βήμα 1.

Αυτή η περίπτωση χρήσης ολοκληρώνεται με την επιλογή εξόδου ή αναμονής από τον χρήστη.

Εναλλακτικές ροές

Εναλλακτική ροή 1: η πληροφορία δεν βρίσκεται στην βάση δεδομένων της φορητής συσκευής

Η πληροφορία δεν βρίσκεται στην βάση δεδομένων και η εφαρμογή εμφανίζει σχετικό μήνυμα.

Μη λειτουργικές απαιτήσεις

Η φορητή συσκευή πρέπει να βρίσκεται στην εμβέλεια του ασύρματου (WiFi) δικτύου.

Κατάσταση Εισόδου

Η εφαρμογή να βρίσκεται τουλάχιστον σε κατάσταση αναμονής.

Κατάσταση Εξόδου

6.1.2 Δεύτερη περίπτωση χρήσης

Τίτλος περίπτωσης χρήσης

Προσθήκη Εξέτασης.

Σύντομη περιγραφή

Ο ιατρός, με την χρήση της σχετικής επιλογής, προσθέτει το μητρώο των ασθενών στην βάση δεδομένων.

Χειριστές

Ο ιατρός.

Ροή γεγονότων

Βασική Ροή

Η εφαρμογή εμφανίζει τα πεδία εισαγωγής.

Ο ιατρός προσθέτει τις κατάλληλες πληροφορίες.

Η εφαρμογή αποστέλλει τα δεδομένα στο σχετικό php script.

Το php script αποθηκεύει τα δεδομένα στην βάση δεδομένων.

Ο χρήστης επιλέγει επιστροφή στο βασικό μενού.

Η εφαρμογή επιστρέφει στο βήμα 1.

Αυτή η περίπτωση χρήσης ολοκληρώνεται με την επιλογή εξόδου ή αναμονής από τον χρήστη.

Μη λειτουργικές απαιτήσεις

Η φορητή συσκευή πρέπει να βρίσκεται στην εμβέλεια του ασύρματου (WiFi) δικτύου.

Κατάσταση Εισόδου

Η εφαρμογή να βρίσκεται τουλάχιστον σε κατάσταση αναμονής.

Κατάσταση Εξόδου

6.1.3 Τρίτη περίπτωση χρήσης

Τίτλος περίπτωσης χρήσης

Προσθήκη Εξέτασης.

Σύντομη περιγραφή

Ο ιατρός, με την χρήση της σχετικής επιλογής, προσθέτει μια εξέταση στην βάση δεδομένων.

Χειριστές

Ο ιατρός.

Ροή γεγονότων

Βασική Ροή

Η εφαρμογή εμφανίζει τα πεδία εισαγωγής.

Ο ιατρός προσθέτει τις κατάλληλες πληροφορίες.

Η εφαρμογή αποστέλλει τα δεδομένα στο σχετικό php script.

Το php script αποθηκεύει τα δεδομένα στην βάση δεδομένων.

Ο χρήστης επιλέγει επιστροφή στο βασικό μενού.

Η εφαρμογή επιστρέφει στο βήμα 1.

Αυτή η περίπτωση χρήσης ολοκληρώνεται με την επιλογή εξόδου ή αναμονής από τον χρήστη.

Μη λειτουργικές απαιτήσεις

Η φορητή συσκευή πρέπει να βρίσκεται στην εμβέλεια του ασύρματου (WiFi) δικτύου.

Κατάσταση Εισόδου

Η εφαρμογή να βρίσκεται τουλάχιστον σε κατάσταση αναμονής.

Κατάσταση Εξόδου

ΚΕΦΑΛΑΙΟ 7

7.1 ΑΝΑΣΚΟΠΗΣΗ ΥΛΟΠΟΙΗΣΗΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΚΑΙ ΟΛΟΚΛΗΡΩΣΗ

Έως τώρα υλοποιήθηκαν τα τρία από τα τέσσερα στάδια της μεθοδολογίας της ενοποιημένης προσέγγισης, που επιλέχθηκαν για την υλοποίηση της εφαρμογής. Εξετάστηκε η πλατφόρμα υλοποίησης της εφαρμογής, αναλύθηκαν οι προδιαγραφές της και προδιαγράφηκαν οι απαιτήσεις του λογισμικού. Στην συνέχεια σύμφωνα με τα πεπραγμένα των δύο πρώτων φάσεων και τις κριτικές αποφάσεις και παραδοχές που έγινε για την δημιουργία της εφαρμογής, υλοποιήθηκε η εφαρμογή και εφαρμόστηκε ο αρχικός έλεγχος, που υλοποιήθηκε για την επαλήθευση των περιπτώσεων χρήσης, όπως αυτές είχαν αναπτυχθεί στην φάση της επεξεργασίας. Στο παρών κεφάλαιο ο στόχος είναι να γίνει αναφορά σε επιμέρους ζητήματα της εφαρμογής στην πορεία προς την ολοκλήρωση της πτυχιακής εργασίας. Τέτοια ζητήματα είναι διάφορα προβλήματα που παρουσιάστηκαν στην πορεία, ενέργειες που απαιτήθηκαν για την συγγραφή του κώδικα της εφαρμογής αλλά και γενικότερα της πτυχιακής εργασίας. Το κεφάλαιο θα ολοκληρωθεί παρουσιάζοντας τα συμπεράσματα της εργασίας.

7.2 ΠΡΟΒΛΗΜΑΤΑ ΣΤΗΝ ΣΥΓΓΡΑΦΗ ΤΟΥ ΚΩΔΙΚΑ

Ενώ έχει ολοκληρωθεί η συγγραφή του κώδικα και παρουσιάστηκε ήδη σε προηγούμενα κεφάλαια το πως καλύπτονται οι λειτουργικές και μη λειτουργικές απαιτήσεις της εφαρμογής μέσα από τον κώδικα της εφαρμογής είναι δύσκολο να παρουσιαστεί ο τρόπος με τον οποίο έγινε η συγγραφή του κώδικα και τα προβλήματα που δημιουργήθηκαν στην πορεία αλλά και πως αυτά αντιμετωπίστηκαν. Κάθε σημείο της πτυχιακής εργασίας είχε τις δυσκολίες του, με την συγγραφή του κώδικα να κρατάει τα σκήπτρα ως το πιο χρονοβόρο, επίπονο και ενεργοβόρο σημείο αυτής της προσπάθειας. Η δυσκολία στην συγγραφή του κώδικα πηγάζει αρχικά από την μεθοδολογία και ονοματολογία που χρησιμοποιεί η πλατφόρμα Android για την δημιουργία εφαρμογών. Κάθε πράγμα όταν το αντιμετωπίζεις για πρώτη φορά σε θέτει αντιμέτωπο με αρκετά προβλήματα, τα οποία αφού καταφέρεις να ξεπεράσεις, διαπιστώνεις πόσο εύκολο είναι να υλοποιήσεις εκ νέου ένα παρόμοιο εγχείρημα, αυτό βέβαια συμβαίνει αφού έχεις ήδη κατακτήσει με κόπο όλη την απαιτούμενη γνώση. Έτσι έγινε και σε αυτή την εργασία.

Ξεκινώντας την υλοποίηση της εφαρμογής το αρχικό πρόβλημα ήταν πως η πλατφόρμα Android είναι νέα στον χώρο (μόλις 2 ετών) και κατά συνέπεια δεν υπήρχαν πάρα πολλά παραδείγματα ώστε να γίνει κατανόηση κάποιων βασικών αρχών σκέψης και υλοποίησης. Ένα δεύτερο πρόβλημα ήταν πως την στιγμή που δόθηκε η συγκεκριμένη εργασία, δεν υπήρχε ούτε ένα βιβλίο ώστε να αποτελεί το σημείο αναφοράς για την αντιμετώπιση κάποιων προβλημάτων. Τέλος το μεγαλύτερο πρόβλημα που έπρεπε να αντιμετωπιστεί ήταν η μη ύπαρξη κάποιου τρόπου για την απευθείας σύνδεση της εφαρμογής με μια βάση δεδομένων που βρίσκεται σε έναν server. Στα παραπάνω τρία προβλήματα ήρθε να δώσει λύσει το internet, κάνοντας χρήση των αναζητήσεων (<http://www.google.com/>), την συμμετοχή σε κάποια forums (<http://www.anddev.org/>) όπου γινόταν συζήτηση για διάφορα θέματα προγραμματισμού στην πλατφόρμα Android και κατά κύριο λόγο η χρήση του IRC Channel που έχουν οι δημιουργοί του Android (channel: #android, server: irc.freenode.net) ώστε να είναι εφικτή η άμεση επικοινωνία. Τέλος, περίπου στην μέση αυτής της πτυχιακής, άρχισαν να εκδίδονται τα πρώτα βιβλία [1-7] για τον τρόπο χρήσης του Android SDK και την κατασκευή κάποιων προγραμμάτων.

Συμπερασματικά και κλείνοντας αυτό το κεφάλαιο πρέπει να τονιστεί η ομορφιά της δημιουργίας που προσφέρεται μέσα από την κατασκευή μιας καινούργιας εφαρμογής και σε καμία περίπτωση κάθε επίδοξος κατασκευαστής λογισμικού δεν πρέπει να αποθαρρύνεται από τα προβλήματα που αναφέρθηκαν σε αυτό το κεφάλαιο. Η κατάκτηση της γνώσης που ακολουθεί την επίλυση όσων προβλημάτων προκύπτουν είναι η ανταμοιβή.

7.3 ΣΥΜΠΕΡΑΣΜΑΤΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Με την παρούσα εργασία έγινε η πιο ήπια μεταφορά από την θεωρία των βιβλίων στην πράξη και την πραγματική υλοποίηση μιας εφαρμογής. Ακολουθήθηκαν τυπικά όλες οι διαδικασίες που περιγράφονται στα βιβλία τεχνολογίας λογισμικού και διαπιστώθηκε στην πράξη το πλήθος των παρατηρήσεων και δυσκολιών που αναφέρουν και τα οποία παρατηρούνται στην τεχνολογία λογισμικού.

Η πλατφόρμα λογισμικού Android προσφέρει πολλές δυνατότητες για την ανάπτυξη εφαρμογών όλων των κατηγοριών και η ιδέα της ανοικτής διακίνησης λογισμικού που υποστηρίζει προσφέρει την ελευθερία και το απαραίτητο κίνητρο προς τους προγραμματιστές να ασχοληθούν με την πλατφόρμα και ελεύθερα να υλοποιήσουν νέες ιδέες που επιθυμούν να εφαρμόσουν στις φορητές συσκευές. Παρέχει όλα τα απαραίτητα εργαλεία ανάπτυξης και ελέγχου λογισμικού και ενώ είναι σχετικά μια νέα πλατφόρμα πραγματοποιείται σοβαρή δουλειά και πλαισιώνεται με σωστή υποστήριξη (κυρίως μέσω διαδικτύου) προς τον επίδοξο προγραμματιστή δημιουργώντας του μια αίσθηση ομαδικότητας και αλληλεγγύης.

Διαπιστώθηκε η σημαντικότητα της χρήσης μιας σαφούς μεθοδολογίας ανάπτυξης λογισμικού, όπως είναι αυτή της ενοποιημένης προσέγγισης, η οποία πρέπει απαραίτητα να ακολουθείται για να οδηγούμαστε με σταθερά βήματα προς την επίτευξη του στόχου της ολοκλήρωσης της εφαρμογής μας. Κάθε φάση της δημιουργίας του λογισμικού πρέπει να στηρίζεται στην προηγούμενη και να δίνει συνέχεια στην επόμενη ώστε να υπάρχει σωστή και ομαλή μετάβαση και στο λογισμικό. Είναι πολύ χρήσιμο ο κατασκευαστής του λογισμικού όταν συναντάει δυσκολίες και προβληματίζεται για το πώς να συνεχίσει την ανάπτυξη του λογισμικού, να ανατρέχει στις προηγούμενες φάσεις της διαδικασίας για να επιβεβαιώνει ή ακόμα και να διορθώνει τυχόν λάθη στο σημείο που βρίσκεται.

Απαιτείται αναλυτική παρουσίαση όλων των βημάτων της ανάπτυξης του λογισμικού ώστε να μπορεί ο οποιοσδήποτε να παρακολουθήσει την εξέλιξη του έργου αλλά και για να μπορεί να εντοπίσει ευκολότερα λάθη και παραλείψεις της εφαρμογής. Ο ίδιος ο κατασκευαστής του λογισμικού πρέπει να επιδιώκει την δημιουργία των απαραίτητων διαγραμμάτων και γενικά όλων των συνοδευτικών εγγράφων που θα κάνουν την εφαρμογή πιο κατανοητή ώστε να μπορεί να ανατρέξει σε αυτά για την αντιμετώπιση οποιουδήποτε προβλήματος.

Η συγγραφή του κώδικα πρέπει να γίνεται με γνώμονα την απλότητα και πληρότητα της εφαρμογής και πρέπει να καλύπτει όλες τις απαιτήσεις όπως αυτές σαφώς έχουν οριστεί σε προηγούμενες φάσεις ανάπτυξης. Υπάρχει μια τεράστια πληθώρα συνδυασμών για να γραφεί κώδικας για κάποια εφαρμογή και αυτό κάνει την συγγραφή κώδικα μια ιδιαίτερα ευχάριστη και ενδιαφέρουσα διαδικασία. Το αποτέλεσμα της συγγραφής κώδικα σίγουρα θα περιλαμβάνει σφάλματα για αυτό η διαδικασία διόρθωσης πρέπει να είναι προσεκτική.

Ο προγραμματισμός για μια συσκευή όπως αυτή του Android δείχνει το πόσο έχει εξελιχθεί η Τεχνολογία και πως με την απαραίτητη προσοχή και θέληση είναι εφικτό κάποιος να υλοποιήσει μια εφαρμογή η οποία θα ανταποκρίνεται στις ανάγκες της εποχής μας χωρίς να χρειάζεται να έχει πολύ εξειδικευμένες γνώσεις, όπως γινόταν στο παρελθόν.

Τέλος στην επιστήμη των υπολογιστών υπάρχει περιθώριο για πολύ φαντασία και ακόμα περισσότερη δημιουργική σκέψη, που παρέχει σε κάθε έναν, από όσους ασχολούνται με αυτή την επιστήμη, την δυνατότητα να υλοποιήσουν κάθε ιδέα τους και αυτή η μετουσίωση από ιδέα σε απτή πραγματικότητα, είναι κάτι ιδιαίτερα γοητευτικό και ευχάριστο.

7.4 ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΔΙΟΡΘΩΣΕΙΣ

Κατά την διαδικασία της διόρθωσης της εφαρμογής του AMRS που έγινε ακριβώς μετά την ολοκλήρωση της συγγραφής του κώδικα, βγήκαν πολύ χρήσιμα συμπεράσματα για τα σημεία που πρέπει να τροποποιηθούν ώστε το πρόγραμμα, πέρα από απλά λειτουργικό να γίνει λειτουργικό, φιλικό και να ανταποκρίνεται περισσότερο στις νέες τάσεις και συνήθειες των χρηστών. Το AMRS όπως ήδη διαπιστώθηκε σε προηγούμενο κεφάλαιο είναι λειτουργικό και υλοποιεί όλους τους στόχους που τέθηκαν εξ' αρχής, με δυνατότητα βελτίωσης στη μέθοδο εισαγωγής των πληροφοριών. Από την χρήση της εφαρμογής διαπιστώθηκαν κάποιες βελτιώσεις που μπορούν να γίνουν ώστε να είναι πιο φιλικό στην χρήση και να συμβαδίζει με τα νέα πρότυπα των σύγχρονων εφαρμογών φορητών συσκευών. Έτσι ακολούθως θα αναφέρουμε κάποιες απ' τις βελτιώσεις που δεν αφορούν το λειτουργικό κομμάτι αλλά το εικαστικό κυρίως.

Πρέπει να εφαρμοστούν περισσότερα πεδία εισαγωγής πληροφοριών τα οποία θα προσδώσουν στην εφαρμογή ελκυστικότητα στην πλοήγηση ανάμεσα σε διαφορετικές πληροφορίες. Με αυτό τον τρόπο θα πετύχουμε μεγαλύτερη αμεσότητα στην παρεχόμενη πληροφορία και θα δώσουμε την αίσθηση στο χρήστη πως μπορεί ακαριαία να πληροφορηθεί για το σημείο ενδιαφέροντος που αναζητάει.

Η λειτουργική απαίτηση για την εμφάνιση των πληροφοριών πρέπει να εξεταστεί απ' την αρχή και να μελετηθεί και αποφασιστεί εκ νέου αν θα πρέπει να χρησιμοποιηθούν κάποιες προεπιλεγμένες συναρτήσεις που θα εμφανίζουν συγκεκριμένες πληροφορίες, όπως τελευταία καταχώρηση, ή αν θα πρέπει να δίνουμε την δυνατότητα στον ιατρό να αποφασίσει τότε και ποιες πληροφορίες θέλει να δει. Σε κάθε περίπτωση αυτό είναι ένα σοβαρό ζήτημα γιατί απ' τη μια μπορεί να ξεφύγουμε απ' τον αρχικό σχεδιασμό του AMRS με τους στόχους που θέσαμε εξ' αρχής και από την άλλη μπορεί να αποτελέσει το κριτικό σημείο βελτίωσης που θα εκτινάξει την ελκυστικότητα της εφαρμογής και θα δημιουργήσει μεγαλύτερη αποδοχή από τους υποψήφιους χρήστες του AMRS. Όποια και αν είναι βέβαια η απόφαση πρέπει οπωσδήποτε να γίνει χρήση κάποιου εικονικού σώματος έτσι ώστε το σύνολο της εφαρμογής μας να απαιτεί χειρισμό μέσω της οθόνης αφής για λόγους ομοιομορφίας και ευκολίας του χρήστη.

Βλέποντας το AMRS στη χρήση διαπιστώθηκε πως για κάθε λειτουργία απαιτείται να εισάγει κάποιο κείμενο ο ιατρός ώστε αυτή να διεκπεραιωθεί. Έτσι πρέπει να αποφασιστεί εάν θα πρέπει το πρόγραμμα να λειτουργεί αποκλειστικά με πεδία εισαγωγής ή με κάποια drop-down μενού τα οποία θα βελτιώσουν κατά πολύ την εμπειρία χρήσης της εφαρμογής.

Οι παραπάνω διορθώσεις, δεν έγιναν, διότι σκοπός της εργασίας ήταν να υλοποιηθεί κάποια λειτουργική έκδοση της εφαρμογής και όχι κάποιο τελικό προϊόν το οποίο απαιτεί την ύπαρξη περισσότερων προγραμματιστών και σχεδιαστών.

7.5 ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

Η εφαρμογή του Ηλεκτρονικού Ιατρικού Φακέλου στηρίζεται στην πρωτότυπη ιδέα της χρήσης ενός server όπου εκεί θα αποθηκεύονται οι απαραίτητες πληροφορίες ενός ασθενή και θα είναι διαθέσιμες στους ιατρούς που κάνουν χρήση της υπηρεσίας. Πρέπει να γίνει ολοκληρωμένη μελέτη εγκατάστασης πομπών (συσκευών WiFi) σε νοσοκομεία είτε δημόσια, είτε ιδιωτικά, που θα δίνουν την δυνατότητα αποστολής πληροφοριών σε συσκευές με εγκατεστημένο το AMRS. Το μικρό κόστος τέτοιων πομπών μικρής χωρητικότητας και ενεργειακής κατανάλωσης και η ευκολία χρήσης του AMRS σε συνδυασμό με την ταχεία εξάπλωση των φορητών συσκευών, θα εξασφαλίσουν την μεγάλη αποδοχή από τους χρήστες και μπορούν να αντικαταστήσουν δαπανηρότερες και πολυπλοκότερες σε χρήση συσκευές. Επιπροσθέτως το AMRS μπορεί να εμπλουτιστεί με πληροφορίες ώστε να καλύπτει ολόκληρες περιοχές ενδιαφέροντος και να χρησιμοποιείται από τους ιατρούς που κατέχουν μια αντίστοιχη φορητή συσκευή στα ιδιωτικά τους ιατρεία, παρέχοντας άμεσα και εύκολα πληροφορίες για τους ασθενείς τους.

7.6 ΜΕΛΛΟΝΤΙΚΗ ΧΡΗΣΗ ΤΗΣ ΠΑΡΟΥΣΑΣ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Η παρούσα πτυχιακή εργασία αποτελεί την ανάπτυξη μιας εφαρμογής από μηδενική βάση και μπορεί να χρησιμοποιηθεί ως παράδειγμα σχεδιασμού και ανάπτυξης για οποιαδήποτε μικρής κλίμακας εφαρμογή λογισμικού αλλά και σαν οδηγός για μεγαλύτερης κλίμακας εφαρμογές. Η ολοκλήρωση της εφαρμογής μέσω όλων των τυπικών σταδίων ανάπτυξης εφαρμογών την κάνει καλό παράδειγμα για όσους θέλουν να ασχοληθούν με την εκ βαράθρων σχεδίαση και ανάπτυξη λογισμικού. Ο κώδικας της εφαρμογής μπορεί να χρησιμοποιηθεί ως καλό παράδειγμα για τον τρόπο και την μεθοδολογία συγγραφής εφαρμογών με την γλώσσα Java καθώς και σε επίπεδο σύνταξης και εντολών κώδικα για την πλατφόρμα Android. Η πρότυπη αυτή εργασία με στοχοποιημένη γραφή για την πλατφόρμα Android μπορεί σαφέστατα να χρησιμοποιηθεί για την συγγραφή παρόμοιων εφαρμογών για χρήση στην ίδια πλατφόρμα λογισμικού ή μπορεί να χρησιμοποιηθεί ως παράδειγμα για ανάπτυξη εφαρμογών σε αντίστοιχα στοχευμένες πλατφόρμες λογισμικού.

Επίσης, μελλοντική χρήση της παρούσας εργασίας θα μπορούσε και πρέπει να είναι η δημιουργία κάποιου γραφικού περιβάλλοντος χρήσης όπου θα αξιοποιεί τις συναρτήσεις αλλά και την βάση δεδομένων που υλοποιήθηκαν σε αυτή την πτυχιακή εργασία, ώστε το AMRS να έρθει πιο κοντά σε αυτό που «λέμε» τελικό προϊόν. Με αυτό τον τρόπο θα μπορεί άνετα κάποιος ιατρός στο ιδιωτικό του ιατρείο ή σε κάποιο νοσοκομείο να χρησιμοποιήσει το AMRS σε καθημερινή βάση για την οργάνωση των ιατρικών φακέλων των ασθενών του.

ΠΗΓΕΣ

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Reto Meier: *'Professional Android Application Development'* (Wiley India, 12/2008)
2. Chris Haseman: *'Android Essentials'* (Apress, 7/2008)
3. Jerome DiMarzio: *'Android a Programmers Guide'* (McGraw-Hill, 7/2008)
4. Frank Ableson, Charlie Collins, Robi Sen: *'Unlocking Android - A Developer's Guide'* (Manning Publications, 4/2009)
5. Rick Rogers, John Lombardo, Zigurd Mednieks, Blake Meike: *'Android Application Development'* (O'Reilly Media, 5/2009)
6. Mark L. Murphy: *'Beginning Android'* (Apress, 6/2009)
7. Sayed Y. Hashimi, Satya Komatineni: *'Pro Android'* (Apress, 6/2009)

ΙΣΤΟΣΕΛΙΔΕΣ

8. Η.Ι.Φ. – Τεχνολογίες για την υλοποίησή του. [2/10/2009]
<http://mpl.med.uoa.gr/Downloads/PDF/ilektronikos-iatrikos-fakelos.pdf>
9. Ολοκληρωμένο πληροφορικό σύστημα περιφερειακού ιατρείου [2/10/2009]
<http://www.ehealthgr.net/ePi/ePi.htm>
10. Ίδρυμα Ελεύθερου Λογισμικού [28/2/2009]
<http://www.gnu.org/philosophy/free-sw.el.html>
11. The Symbian Platform by NOKIA [27/2/2009]
<http://www.securitytechnet.com/resource/rsc-center/vendor-wp/nokia/dec002.pdf>
12. UIQ Technology [27/2/2009]
<http://www.uiq.com/product/>
13. S60 Platform [26/2/2009]
<http://www.forum.nokia.com/main/platforms/s60/>
14. QTopia Platfrom [26/2/2009]
<http://dist.trolltech.com/pdf/qtopia-wp-platform42.pdf>
15. Java FX Mobile [20/1/2009]
<http://www.sun.com/software/javafx/mobile/index.jsp>
16. Windows Mobile [20/1/2009]
<http://msdn.microsoft.com/en-us/windowsmobile/bb250543.aspx>
17. Access Linux Platform [20/1/2009]
<http://alp.access-company.com/>
18. Android [14/1/2009]
<http://code.google.com/android/documentation.html>
19. Android [14/1/2009]
<http://code.google.com/android/index.html>

20. **Android** [14/1/2009]
<http://code.google.com/android/what-is-android.html>
21. **Dalvik Virtual Machine** [14/10/2009]
<http://dalvikvm.com/#Introduction>
22. **Open Handset Alliance** [12/1/2009]
http://www.openhandsetalliance.com/android_overview.html
23. **Open Source** [12/1/2009]
<http://www.opensource.org/>
24. **Eclipse** [10/1/2009]
<http://www.eclipse.org/>
25. **Android** [10/1/2009]
<http://code.google.com/intl/el/android/intro/installing.html>
26. **UML** [8/1/2009]
<http://www.uml.org/>

ΚΩΔΙΚΑΣ

ΚΙΝΗΤΟ - ΕΦΑΡΜΟΓΗ

main.java:

```
package ehospital.server;
import android.app.Activity;
import android.graphics.Color;
import android.os.Bundle;
import android.widget.TextView;

public class main extends Activity {
    public void onCreate(Bundle savedInstanceState){
        super.onCreate(savedInstanceState);
        TextView text = new TextView(this);
        String result = "Welcome!!! :-)";
        String request[] = null;
        Integer i=1;
        text.setText(result);
        text.setTextColor(Color.RED);
        text.setTextSize(18);
        //text.setBackgroundResource(R.drawable.background);
        setContentView(text);

        /*
        //Προσθήκη δεδομένων στον πίνακα android.mhtrwo_asthenwn.
        dbAdd.dbAddPatient("Τυχάλας", "Δημήτρης", "1988-02-24", 22, "Ο
Θετικό", "Πανεπιστήμιο", "Άγαμος", "Άνδρας", "Ελληνας", "Σέρρες",
"Θεσσαλονίκη", "Γεώργιος", "Ελένη", "Πόντου", 17, "Σέρρες", 62125,
223104555, 13584, 942620616, "tichalas@gmail.com", 13572468, "Ατομική", 248,
"2009-03-24", "20-03-24", "Android Developer", "Internet", 13524, 2486135,
"dtchalas@ucg.gr", "Internet", 2411235, "Νιγρίτας", "Τίποτα απολύτως");

        //Αναζήτηση δεδομένων στον πίνακα android.mhtrwo_asthenwn.
        request = Request.dbRequest("Onoma", "mhtrwo_asthenwn",
"kwdikos=1");

        //Εκτύπωση δεδομένων.
        if (request[0] == ""){
            result = "Δεν βρέθηκαν σχετικές εγγραφές.";
        }
        else{
            result = request[0];
        }
        */

        /*
        //Προσθήκη δεδομένων στον πίνακα android.eksetash.
        dbAdd.dbAddEksetash(1 , 2 , "2010-02-24" , "Σπυράκια" ,
"Τίποτα" , "Καπνιστής" , "Ταχυκαρδία" , "Τίποτα" , "Πολύ καλά
αντινακλαστικά" , "Βήχας" , "Κρυσολόγημα" , "Τίποτα" , "Mucosolvan" ,
"Κρυσολόγημα");

        //Αναζήτηση δεδομένων στον πίνακα android.eksetash.
        request = Request.dbRequest("*", "eksetash", "");
        if (request[0] == ""){
            result = "Δεν βρέθηκαν σχετικές εγγραφές.";
        }
        else{
            //Εκτύπωση δεδομένων.
            result = "Αποτελέσματα: ";
            for(i=0; i<request.length; i++){
                result = result + "\n" + i.toString() + "\t: \t" +
request[i];
```

```

    }
    */

    /*
    //Προσθήκη δεδομένων στον πίνακα android.iatrikos_fakelos.
    dbAdd.dbAddFolder(1, "Υγιέστατος", "Συκώτι", "Καρκίνος μαστού",
"Πνευμονία");

    //Προσαρμογή αναζήτησης για τον πίνακα
    android.iatrikos_fakelos.
    String[] Fakelos =
{"Id","id_asthenh","Biografiko","atomiko_anamnhstiko","oikogeneiako_istoriko
", "atomika_prohgoumena"};

    //Αναζήτηση δεδομένων στους πίνακες της βάσης android.
    request = Request.dbRequest("*", "iatrikos_fakelos", "");
    if (request[0] == ""){
        result = "Δεν βρέθηκαν σχετικές εγγραφές.";
    }
    else{
        //Εκτύπωση δεδομένων.
        result = "Αποτελέσματα: ";
        for(i=0; i<request.length; i++){
            result = result + "\n" + Fakelos[i%6] + ":\t" +
request[i]; //Εδώ χρησιμοποιώ το i modulus 6 (i%6) για να κάνω κάθε 6 φορές
κύκλο στον πίνακα Fakelos.
        }
    }
    */

    //Αναζήτηση δεδομένων στους πίνακες της βάσης android.
    request = Request.dbRequest("Ερwnymo, Onoma",
"iatrikos_fakelos, mhtrwo_asthenwn", "mhtrwo_asthenwn.kwdikos =
iatrikos_fakelos.Id_asthenh");
    if (request[0] == ""){
        result = "Δεν βρέθηκαν σχετικές εγγραφές.";
    }
    else{
        //Εκτύπωση δεδομένων.
        result = "Αποτελέσματα: ";
        for(i=0; i<request.length; i++){
            result = result + " " + request[i];
        }
    }

    text.setText(result);
}
}

```

dbAdd.java:

```
package ehospital.server;
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import android.app.Activity;

public class dbAdd extends Activity {
    public static void dbAddTest(String name){
        try {
            name = URLEncoder.encode(name, "utf-8");
        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        }
        httpPost.post("http://192.168.1.6/AddData.php", "input=INSERT
INTO android.test(id , name) VALUES(NULL, \""+name+"\")");
    }

    public static void dbAddPatient(String Epwnymo, String Onoma, String
hmeromhnia_gennshs, Integer Hlikia, String Omada_aimatos, String
Morfwtko_epipedo, String Oikogeneiakh_katastash, String Fylo, String
Ethnikothta, String Katagwgh, String Topos_gennshs, String Onoma_Patera,
String Onoma_mhteras, String Diamonh_odos, Integer Diamonh_arithmos, String
Diamonh_perioxh, Integer Diamonh_tk, Integer Diamonh_thlefwno, Integer
Diamonh_fax, Integer Diamonh_kinhto, String Diamonh_eMail, Integer
Asfaleia_kwdikos, String Asfaleia_perigrafh, Integer
Asfaleia_arithmos_symvolaiou, String Asfaleia_hmeromhnia_apo, String
Asfaleia_hmeromhnia_ews, String Epaggelma_perigrafh, String
Epaggelma_perioxh, Integer Epaggelma_thlefwno, Integer Epaggelma_fax, String
Epaggelma_eMail, String Perivallon_ergasias, Integer AFM, String DOY, String
Parathrhseis){
        try {
            Epwnymo = URLEncoder.encode(Epwnymo, "utf-8");
            Onoma = URLEncoder.encode(Onoma, "utf-8");
            Omada_aimatos = URLEncoder.encode(Omada_aimatos, "utf-
8");
            Morfwtko_epipedo = URLEncoder.encode(Morfwtko_epipedo,
"utf-8");
            Oikogeneiakh_katastash =
URLEncoder.encode(Oikogeneiakh_katastash, "utf-8");
            Fylo = URLEncoder.encode(Fylo, "utf-8");
            Ethnikothta = URLEncoder.encode(Ethnikothta, "utf-8");
            Katagwgh = URLEncoder.encode(Katagwgh, "utf-8");
            Topos_gennshs = URLEncoder.encode(Topos_gennshs, "utf-
8");
            Onoma_Patera = URLEncoder.encode(Onoma_Patera, "utf-8");
            Onoma_mhteras = URLEncoder.encode(Onoma_mhteras, "utf-
8");
            Diamonh_odos = URLEncoder.encode(Diamonh_odos, "utf-8");
            Diamonh_perioxh = URLEncoder.encode(Diamonh_perioxh,
"utf-8");
            Diamonh_eMail = URLEncoder.encode(Diamonh_eMail, "utf-
8");
            Asfaleia_perigrafh =
URLEncoder.encode(Asfaleia_perigrafh, "utf-8");
            Epaggelma_perigrafh =
URLEncoder.encode(Epaggelma_perigrafh, "utf-8");
            Epaggelma_eMail = URLEncoder.encode(Epaggelma_eMail,
"utf-8");
            Perivallon_ergasias =
URLEncoder.encode(Perivallon_ergasias, "utf-8");
            DOY = URLEncoder.encode(DOY, "utf-8");
            Parathrhseis = URLEncoder.encode(Parathrhseis, "utf-8");
        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        }
        httpPost.post("http://192.168.1.6/AddData.php", "input=INSERT
INTO android.mhtrwo_asthenwn(kwdikos, Epwnymo, Onoma, hmeromhnia_gennshs,
Hlikia, Omada_aimatos, Morfwtko_epipedo, Oikogeneiakh_katastash, Fylo,
```

```

Ethnikothta, Katagwgh, Topos_gennshs, Onoma_Patera, Onoma_mhteras,
Diamonh_odos, Diamonh_arithmos, Diamonh_perioxh, Diamonh_tk,
Diamonh_thlefwno, Diamonh_fax, Diamonh_kinhtho, Diamonh_eMail,
Asfaleia_kwdikos, Asfaleia_perigrafh, Asfaleia_arithmos_symvolaiou,
Asfaleia_hmeromhnia_apo, Asfaleia_hmeromhnia_ews, Epaggelma_perigrafh,
Epaggelma_perioxh, Epaggelma_thlefwno, Epaggelma_fax, Epaggelma_eMail,
Perivallon_ergasias, AFM, DOY, Parathrhseis) VALUES(NULL, \""+Epwnymo+"\",
\""+Onoma+"\", \""+hmeromhnia_gennshs+"\", \""+Hlikia+"\",
\""+Omada_aimatos+"\", \""+Morfwtko_epipedo+"\",
\""+Oikogeneiakh_katastash+"\", \""+Fylo+"\", \""+Ethnikothta+"\",
\""+Katagwgh+"\", \""+Topos_gennshs+"\", \""+Onoma_Patera+"\",
\""+Onoma_mhteras+"\", \""+Diamonh_odos+"\", \""+Diamonh_arithmos+"\",
\""+Diamonh_perioxh+"\", \""+Diamonh_tk+"\", \""+Diamonh_thlefwno+"\",
\""+Diamonh_fax+"\", \""+Diamonh_kinhtho+"\", \""+Diamonh_eMail+"\",
\""+Asfaleia_kwdikos+"\", \""+Asfaleia_perigrafh+"\",
\""+Asfaleia_arithmos_symvolaiou+"\", \""+Asfaleia_hmeromhnia_apo+"\",
\""+Asfaleia_hmeromhnia_ews+"\", \""+Epaggelma_perigrafh+"\",
\""+Epaggelma_perioxh+"\", \""+Epaggelma_thlefwno+"\", \""+Epaggelma_fax+"\",
\""+Epaggelma_eMail+"\", \""+Perivallon_ergasias+"\", \""+AFM+"\", \""+DOY+"\",
\""+Parathrhseis+"\"");
}

```

```

public static void dbAddFolder(Integer Id_asthenh, String Biografiko,
String atomiko_anamnhstiko, String oikogeneiako_istoriko, String
atomika_prohgoumena){
    try {
        Biografiko = URLEncoder.encode(Biografiko, "utf-8");
        atomiko_anamnhstiko =
URLEncoder.encode(atomiko_anamnhstiko, "utf-8");
        oikogeneiako_istoriko =
URLEncoder.encode(oikogeneiako_istoriko, "utf-8");
        atomika_prohgoumena =
URLEncoder.encode(atomika_prohgoumena, "utf-8");
    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    }
    httpPost.post("http://192.168.1.6/AddData.php", "input=INSERT
INTO android.iatrikos_fakelos(id, Id_asthenh, Biografiko,
atomiko_anamnhstiko, oikogeneiako_istoriko, atomika_prohgoumena)
VALUES(NULL, "+Id_asthenh+", \""+Biografiko+"\",
\""+atomiko_anamnhstiko+"\", \""+oikogeneiako_istoriko+"\",
\""+atomika_prohgoumena+"\"");
}

```

```

public static void dbAddEksetash(Integer id_asthenh , Integer
id_iatrou , String Hmeromhnia , String Fysikh_eksetash_dermatos , String
Fysikh_eksetash_kefalhs_traxhlou , String Fysikh_eksetash_thwraka , String
Fysikh_eksetash_kardias , String Fysikh_eksetash_koillias , String
Fysikh_eksetash_neurologikou_sythmatos , String Symptwmata , String
pithanh_diagnwsh , String Iatrikes_prakseis , String Syntagografhsh , String
Telikh_diagnwsh){
    try {
        Fysikh_eksetash_dermatos =
URLEncoder.encode(Fysikh_eksetash_dermatos, "utf-8");
        Fysikh_eksetash_kefalhs_traxhlou =
URLEncoder.encode(Fysikh_eksetash_kefalhs_traxhlou, "utf-8");
        Fysikh_eksetash_thwraka =
URLEncoder.encode(Fysikh_eksetash_thwraka, "utf-8");
        Fysikh_eksetash_kardias =
URLEncoder.encode(Fysikh_eksetash_kardias, "utf-8");
        Fysikh_eksetash_koillias =
URLEncoder.encode(Fysikh_eksetash_koillias, "utf-8");
        Fysikh_eksetash_neurologikou_sythmatos =
URLEncoder.encode(Fysikh_eksetash_neurologikou_sythmatos, "utf-8");
        Symptwmata = URLEncoder.encode(Symptwmata, "utf-8");
        pithanh_diagnwsh = URLEncoder.encode(pithanh_diagnwsh,
"utf-8");

```

```

        Iatrikes_prakseis = URLEncoder.encode(Iatrikes_prakseis,
"utf-8");
        Syntagografhsh = URLEncoder.encode(Syntagografhsh, "utf-
8");
        Telikh_diagnwsh = URLEncoder.encode(Telikh_diagnwsh,
"utf-8");
    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    }
    httpPost.post("http://192.168.1.6/AddData.php", "input=INSERT
INTO android.eksetash(id, id_asthenh , id_iatrou , Hmeromhnia ,
Fysikh_eksetash_dermatos , Fysikh_eksetash_kefalhs_traxhlou ,
Fysikh_eksetash_thwraka , Fysikh_eksetash_kardias , Fysikh_eksetash_koiliias
, Fysikh_eksetash_neurologikou_systhmatos , Symptwmata , pithanh_diagnwsh ,
Iatrikes_prakseis , Syntagografhsh , Telikh_diagnwsh) VALUES(NULL,
"+id_asthenh+" , "+id_iatrou+" , \"+Hmeromhnia+"\\" ,
\"+Fysikh_eksetash_dermatos+"\\" , \"+Fysikh_eksetash_kefalhs_traxhlou+"\\"
, \"+Fysikh_eksetash_thwraka+"\\" , \"+Fysikh_eksetash_kardias+"\\" ,
\"+Fysikh_eksetash_koiliias+"\\"
\"+Fysikh_eksetash_neurologikou_systhmatos+"\\" , \"+Symptwmata+"\\"
\"+pithanh_diagnwsh+"\\" , \"+Iatrikes_prakseis+"\\" ,
\"+Syntagografhsh+"\\" , \"+Telikh_diagnwsh+"\\"");
    }
}

```


Request.java:

```
package ehospital.server;

import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;

import android.app.Activity;

public class Request extends Activity {
    public static String[] dbRequest(String select, String from, String
where){
        String output[] = null;
        if(where==""){
            output
            httpPost.post("http://192.168.1.6/dbRequest.php", "input=SELECT " + select +
" FROM " + from);
        }
        else{
            try {
                where = URLEncoder.encode(where, "UTF-8");
            } catch (UnsupportedEncodingException e) {
                e.printStackTrace();
            }
            output
            httpPost.post("http://192.168.1.6/dbRequest.php", "input=SELECT " + select +
" FROM " + from + " WHERE " + where);
        }
        return output;
    }
}
```

HttpPost.java:

```
package test.connection;

import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.net.URLDecoder;

import org.apache.http.HttpResponse;
import org.apache.http.ParseException;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.entity.StringEntity;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.params.BasicHttpParams;
import org.apache.http.params.HttpConnectionParams;
import org.apache.http.params.HttpParams;
import org.apache.http.protocol.HTTP;
import org.apache.http.util.EntityUtils;
import android.app.Activity;

public class httpPost extends Activity {
    public static String[] post(String server, String params){
        String[] result = null;
        HttpParams connectionParams = new BasicHttpParams();
        HttpConnectionParams.setSoTimeout(connectionParams, 22000);
        HttpConnectionParams.setConnectionTimeout(connectionParams, 22000);
        HttpClient client = new DefaultHttpClient(connectionParams);
        HttpPost post = new HttpPost(server);
        post.setHeader(HTTP.CONTENT_TYPE, "application/x-www-form-
urlencoded");
        try {
            post.setEntity(new StringEntity(params));
        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        }
    }
}
```

```

    HttpResponse response = null;
    try {
        response = client.execute(post);
    } catch (ClientProtocolException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }

    if (response.getStatusLine().getStatusCode() == 200) {

        String s = null;
        try {
            s =
            URLDecoder.decode(EntityUtils.toString(response.getEntity()), "utf-8");
        } catch (ParseException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }

        response = null;
        result=s.split("/next/");
        return result;
    }
    return result;
}
}

```

SERVER

AddData.php:

```
<html>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<?php
    $user = "android";
    $user_password = "13524";
    $db_name = "android";
    mysql_connect("localhost", $user, $user_password) or
die(mysql_error());
    mysql_select_db($db_name) or die(mysql_error());
    mysql_query("SET CHARACTER SET 'utf8'");
    $input = $_REQUEST["input"];
    $query = str_replace("%20", " ", $input);
    $rs = mysql_query($query);
?>
</html>
```

dbRequest.php:

```
<?php
    $user = "android";
    $user_password = "13524";
    $db_name = "android";
    mysql_connect("localhost", $user, $user_password) or
die(mysql_error());
    mysql_select_db($db_name) or die(mysql_error());
    mysql_query("SET CHARACTER SET 'utf8'");
    $input = $_REQUEST["input"];
    $query = str_replace("%20", " ", $input);
    $rs = mysql_query($query);
    while ($get_info = mysql_fetch_row($rs)){
        foreach ($get_info as $field)
            print $field."/next/";
    }
?>
```

MySQL

```
-- phpMyAdmin SQL Dump
-- version 3.2.4
-- http://www.phpmyadmin.net
--
-- Host: localhost
-- Generation Time: Feb 21, 2010 at 01:22 AM
-- Server version: 5.1.41
-- PHP Version: 5.3.1

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

--
-- Database: `android`
--

-----

--
-- Table structure for table `eksetash`
--

CREATE TABLE IF NOT EXISTS `eksetash` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `id_asthenh` int(11) NOT NULL,
  `id_iatrou` int(11) NOT NULL,
  `Hmeromhnia` date NOT NULL,
  `Fysikh_eksetash_dermatos` text,
  `Fysikh_eksetash_kefalhs_traxhlou` text,
  `Fysikh_eksetash_thwraka` text,
  `Fysikh_eksetash_kardias` text,
  `Fysikh_eksetash_koiliias` text,
  `Fysikh_eksetash_neurologikou_systhmatos` text,
  `Symptwmata` text,
  `pithanh_diagnwsh` text,
  `Iatrikes_prakseis` text,
  `Syntagografhsh` text,
  `Telikh_diagnwsh` text,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=greek AUTO_INCREMENT=1 ;

-----

--
-- Table structure for table `iatrikos_fakelos`
--

CREATE TABLE IF NOT EXISTS `iatrikos_fakelos` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `id_asthenh` tinyint(4) NOT NULL,
  `Biografiko` text,
  `atomiko_anamnhstiko` text,
  `oikogeneiako_istoriko` text,
  `atomika_prohgoumena` text,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=greek AUTO_INCREMENT=3 ;

-----

--
```

```

-- Table structure for table `iatros`
--

CREATE TABLE IF NOT EXISTS `iatros` (
  `kwdikos` int(11) NOT NULL AUTO_INCREMENT,
  `Epwnymo` text CHARACTER SET greek,
  `Onoma` text CHARACTER SET greek,
  `hmeromhnia_gennhshs` date DEFAULT NULL,
  `Hlikia` int(11) DEFAULT NULL,
  `Eidikothta` text CHARACTER SET greek,
  `Fylo` text CHARACTER SET greek,
  `Ethnikothta` text CHARACTER SET greek,
  `Katagwgh` text CHARACTER SET greek,
  `Topos_gennhshs` text CHARACTER SET greek,
  `Onoma_Patera` text CHARACTER SET greek,
  `Onoma_mhteras` text CHARACTER SET greek,
  `Diamonh_odos` text CHARACTER SET greek,
  `Diamonh_arithmos` int(11) DEFAULT NULL,
  `Diamonh_perioxh` text CHARACTER SET greek,
  `Diamonh_tk` int(11) DEFAULT NULL,
  `Diamonh_thlefwno` int(11) DEFAULT NULL,
  `Diamonh_fax` int(11) DEFAULT NULL,
  `Diamonh_kinhtho` int(11) DEFAULT NULL,
  `Diamonh_eMail` text CHARACTER SET greek,
  `Asfaleia_kwdikos` int(11) DEFAULT NULL,
  `Asfaleia_perigrafh` text CHARACTER SET greek,
  `Asfaleia_arithmos_symvolaiou` int(11) DEFAULT NULL,
  `Asfaleia_hmeromhnia_apo` date DEFAULT NULL,
  `Asfaleia_hmeromhnia_ews` date DEFAULT NULL,
  `AFM` int(11) DEFAULT NULL,
  `DOY` text CHARACTER SET greek,
  PRIMARY KEY (`kwdikos`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=2 ;

```

```

-----
--
-- Table structure for table `mhtrwo_asthenwn`
--

CREATE TABLE IF NOT EXISTS `mhtrwo_asthenwn` (
  `kwdikos` int(11) NOT NULL AUTO_INCREMENT,
  `Epwnymo` text CHARACTER SET greek,
  `Onoma` text CHARACTER SET greek,
  `hmeromhnia_gennhshs` date DEFAULT NULL,
  `Hlikia` int(11) DEFAULT NULL,
  `Omada_aimatos` text CHARACTER SET greek,
  `Morfwtikio_epipedo` text CHARACTER SET greek,
  `Oikogeneiakh_katastash` text CHARACTER SET greek,
  `Fylo` text CHARACTER SET greek,
  `Ethnikothta` text CHARACTER SET greek,
  `Katagwgh` text CHARACTER SET greek,
  `Topos_gennhshs` text CHARACTER SET greek,
  `Onoma_Patera` text CHARACTER SET greek,
  `Onoma_mhteras` text CHARACTER SET greek,
  `Diamonh_odos` text CHARACTER SET greek,
  `Diamonh_arithmos` int(11) DEFAULT NULL,
  `Diamonh_perioxh` text CHARACTER SET greek,
  `Diamonh_tk` int(11) DEFAULT NULL,
  `Diamonh_thlefwno` int(11) DEFAULT NULL,
  `Diamonh_fax` int(11) DEFAULT NULL,
  `Diamonh_kinhtho` int(11) DEFAULT NULL,
  `Diamonh_eMail` text CHARACTER SET greek,
  `Asfaleia_kwdikos` int(11) DEFAULT NULL,
  `Asfaleia_perigrafh` text CHARACTER SET greek,
  `Asfaleia_arithmos_symvolaiou` int(11) DEFAULT NULL,
  `Asfaleia_hmeromhnia_apo` date DEFAULT NULL,
  `Asfaleia_hmeromhnia_ews` date DEFAULT NULL,

```

```
`Epaggelma_perigrafh` text CHARACTER SET greek,  
`Epaggelma_perioxh` text CHARACTER SET greek,  
`Epaggelma_thlefwno` int(11) DEFAULT NULL,  
`Epaggelma_fax` int(11) DEFAULT NULL,  
`Epaggelma_eMail` text CHARACTER SET greek,  
`Perivallon_ergasias` text CHARACTER SET greek,  
`AFM` int(11) DEFAULT NULL,  
`DOY` text CHARACTER SET greek,  
`Parathrhseis` longtext CHARACTER SET greek,  
PRIMARY KEY (`kwdikos`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=3 ;
```