UNIVERSITY OF WESTERN
MACEDONIA
DEPT. OF INFORMATICS AND
TELECOMMUNICATIONS ENGINEERING

Δημιουργία εφαρμογής διαχείρισης Ηλεκτρονικού Φακέλου Ασθενούς(EMR) στη πλατφόρμα Android

Διπλωματική εργασία

Σίδερης Γεώργιος Πέτρος

Επιβλέποντες: Δρ. Αγγελίδης Παντελής, Επίκουρος Καθηγητής
Δρ. Δασυγένης Μηνάς, Λέκτορας

Κοζάνη, Μάρτιος 2013

UNIVERSITY OF WESTERN
MACEDONIA
DEPT. OF INFORMATICS AND
TELECOMMUNICATIONS ENGINEERING

Creating an Android application for managing an Electronic Medical Record (EMR)

Diploma Thesis

Peter George Sideris

Supervisors: Dr. Aggelidis Pantelis, Assistant Professor
Dr. Dasygenis Minas, Lecturer

Kozani, March 2013

Η Διπλωματική εργασία εμπνεύσθηκε από την διδαχή που μου προσέφερε ο Δρ. Πολυδώρου Α. : “Το 80% μιας σωστής διάγνωσης είναι το ιατρικό ιστορικό”. Οι ευχαριστίες μου απευθύνονται στον Δρ. Πολυδώρου, διότι ακόμη και όταν βρισκόταν σε ένα χαλαρό μεσημεριανό καλοκαιρινών διακοπών ήταν διατεθειμένος να προσφέρει γνώση σε μια παρέα νέων. Η Γνώση είναι η κινητήριος δύναμη πίσω από οτιδήποτε θετικό έχει επιδείξει ο Άνθρωπος. Τέλος, πρέπει να ευχαριστήσω τα άτομα που με στήριξαν αυτή την περίοδο

Πίνακας Περιεχομένων

Περίληψη

Σκοπός της διπλωματικής είναι να εξερευνήσει ένα κεντροποιημένο ψηφιακό Σύστημα Υγειονομικής Περίθαλψης, τα οφέλη που παρέχει ένα τέτοιο σύστημα και τη τρέχουσα υλοποίηση τέτοιων συστημάτων σε διάφορες περιπτώσεις. Επίσης, θα εξομοιωθεί στη πλατφόρμα Android η διάδραση μεταξύ Ιατρού - ασθενή για την παροχή δικαιωμάτων προβολής ενός Ηλεκτρονικού Ιατρικού Φακέλου Ασθενούς(ΗΙΦΑ) και θα εξετάσουμε τις σύγχρονες τεχνολογίες που αφορούν έναν ΗΙΦΑ. Έως σήμερα δεν υπάρχει ένα συγκεκριμένο πρότυπο για τον ΗΙΦΑ, παρά πολλά ανταγωνιζόμενα, ενώ οι επικρατούσες λύσεις δεν προσφέρουν τον έλεγχο των δεδομένων στον ασθενή/χρήστη, αλλά βρίσκονται στον έλεγχο όποιου τυγχάνει να φιλοξενεί τον φάκελό του.

Στο «Α' Μέρος» θα δούμε τη διαμόρφωση του τοπίου, από άποψη οικονομικοπολιτικών παραγόντων, για την υλοποίηση τέτοιων συστημάτων και την εγκόλπωση σύγχρονων μέσων στα Συστήματα Υγείας. Θα επικεντρωθούμε, επί το πλείστον, στην Ευρώπη, καθώς μας αφορά άμεσα και υπάρχει έντονη δράση στο τομέα Ηλεκτρονικής Υγείας· θα εξετάσουμε την ίδια περίπτωση για τις Η.Π.Α., αλλά σε μικρότερη έκταση. Επιπλέον, θα δούμε τις τεχνολογίες και τα πρότυπα που χρησιμοποιούνται για ΗΙΦΑ και για Ηλεκτρονικά Μητρώα Υγείας(ΗΜΥ).

Στο «Β' Μέρος» θα δείξω την υλοποίηση μιας δοκιμαστικής εφαρμογής διαχείρισης Ιατρικού Φακέλου στη πλατφόρμα Android, η οποία βασίζεται – όσον αφορά τα δεδομένα – σε μια διαδικτυακή ιστοσελίδα με την ονομασία «Aegle»(Αίγλη). Η εφαρμογή μας θα επικοινωνεί με την ιστοσελίδα και θα ανταλλάσσει δεδομένα. Για να χρησιμοποιήσει κάποιος την εφαρμογή, θα πρέπει, πρωτίστως, να έχει δημιουργήσει λογαριασμό χρήστη στο «Aegle». Επιπροσθέτως, θα εξετάσουμε την ευκολία που παρέχει η ευρέως διαδεδομένη νέα τεχνολογία των έξυπνων κινητών τηλεφώνων("smartphones") και την αξιοποίηση της τεχνολογίας "NFC(Near Field Communication)" για τον ευκολότερο διαμοιρασμό Ιατρικού Ιστορικού· με στόχο την αποτελεσματικότερη διάγνωση σε ιατρική επίσκεψη ή σε επείγουσες περιπτώσεις, δίχως την χρήση εγγράφων έντυπης μορφής.

Στο «Γ' Μέρος» θα δούμε προτάσεις για επέκταση της λειτουργικότητας της εφαρμογής με επιπλέον λειτουργίες, προτάσεις για καλύτερη απόδοση, εξοικονόμηση εύρους ζώνης, επιπλέον ασφάλεια και το πως θα μπορούσε να εφαρμοστεί ένα τέτοιο σύστημα σε ένα Σύστημα Υγείας.

Summary

The purpose of this Thesis is to explore a centralized digital Health Care system, the benefits of such a system and different case scenarios. Also, I will develop an Android Application that emulates the patient - doctor interaction in the case of providing viewing rights of an Electronic Medical Record(EMR). Until today there isn't a certain standard for EHR, but rather various competing ones, whilst the existing solutions don't offer control of the data to the user/patient, which are in the control of whoever hosts the patient's file.

In "Part 'A" we will see how the landscape is shaped, in socioeconomic terms, regarding the implementation of such systems and the engulfment of contemporary means in Health Care systems. We will, mostly, focus in Europe, as it is of direct concern to us and there is intense activity in the Electronic Health field; we will look the same case for U.S.A., but in lesser extent. Also, we will examine the technologies and standards that are used for the EMR and the Electronic Health Record(EHR).

In "Part 'B" I will demonstrate the implementation of a tentative application for managing Electronic Health Records in Android, which is based – regarding the data – in a web-page named "Aegle". Our application will communicate with the site and will exchange data. For one to use the application, one should have created a user account in Aegle. In addition, we will see the commodity that the widespread technology of smart-phones offers and the use of NFC(Near Field Communication) technology for the easy sharing of Health Records; for more efficient diagnosis in a clinical visit or in an emergency situation, without using printed form documents.

In "part 'C" we will see suggestions for extended functionality of the application with more functions, suggestions for better performance, bandwidth saving, extra security and how we could apply such a system in a Health Care system.

Α’ Μέρος - Ηλεκτρονικά Μητρώα Υγείας, Ηλεκτρονικοί Ιατρικοί Φάκελοι

1. Εισαγωγή

Στα αγγλικά υπάρχουν δύο παρεμφερείς όροι που αντιπροσωπεύουν δύο όμοιες, αλλά όχι ίδιες, έννοιες. Είναι το «Electronic Health Record» και το «Electronic Medical Record»[2]. Ο πρώτος όρος αναφέρεται σε ένα εξελισσόμενο σχέδιο το οποίο ορίζεται ως η συστηματική συλλογή ηλεκτρονικών πληροφοριών που αφορούν την υγεία ενός συγκεκριμένου ατόμου ομάδα ατόμων[1][23] και μπορεί να μοιραστεί μέσω πληροφοριακών συστημάτων και δικτύων. Ο δεύτερος όρος αναφέρεται σε ένα ψηφιακό ή μηχανογραφημένο Ιατρικό φάκελο ο οποίος έχει δημιουργηθεί από ένα οργανισμό, ο οποίος προσφέρει Ιατροφαρμακευτική φροντίδα, όπως ένα νοσοκομείο ή το γραφείο ενός Ιατρού[2]. Συνήθως είναι μέρος ενός αυτόνομου συστήματος πληροφοριών Υγείας. Συχνά αυτοί οι όροι συγχέονται, καθώς έως πρόσφατα δεν υπήρχε σαφής διαχωρισμός των δύο. Θα αναφέρομαι στο «Electronic Health Record» ως «Ηλεκτρονικό Μητρώο Υγείας» και το «Electronic Medical Record» ως «Ηλεκτρονικό Ιατρικό Φάκελο(Ασθενούς)».

Παρακάτω θα εξεταστεί η τρέχουσα προσπάθεια ενοποίησης – αν όχι ενοποίησης τότε σύγκλισης – των δύο και θα δούμε πως μπορεί να συνυπάρξει ένας Ηλεκτρονικός Ιατρικός Φάκελος σε ένα κεντροποιημένο Ηλεκτρονικό Μητρώο Υγείας. Το τελικό όνομα, ενός τέτοιου συστήματος, θα το αφήσουμε στην κρίση των ατόμων που χαράσσουν την πολιτική Υγείας.

2. Ιστορικό Ηλεκτρονικών Μητρώων Υγείας(ΗΜΥ) και Ηλεκτρονικού Ιατρικού Φακέλου Ασθενούς(ΗΙΦΑ)

Οι Ηλεκτρονικοί Ιατρικοί φάκελοι ασθενούς υπάρχουν άνω των 30 χρόνων. πάρα ταύτα δεν υπήρξε εκτενής υλοποίηση αυτών των συστημάτων έως και το 2006, όπου λιγότερο από 10% των νοσοκομείων στις Η.Π.Α.[3] λειτουργούσαν με τέτοια συστήματα. Βασικοί λόγοι ήταν το κόστος για μικρούς οργανισμούς, καθώς επίσης και η δυσκολία στη χρήση τέτοιων ψηφιακών μέσων από τους ιατρούς και τους ασθενείς. Επιπλέον, δεν υπήρχε ένα σταθερό πρότυπο, το οποίο να επιτρέπει την μεταφορά των δεδομένων από τον ένα οργανισμό στον άλλον, κάτι που δημιούργησε αρνητικό κλίμα όσον αφορά την υιοθέτηση τέτοιων πρακτικών συντήρησης Ιατρικού ιστορικού[4][5][6].

Το Ηλεκτρονικό Μητρώο Υγείας είναι μία ιδέα η οποία έχει αρχίσει να "στερεοποιείται" τα τελευταία 15 χρόνια, χωρίς να έχει υπάρξει – έως τώρα – μια ολοκληρωμένη, κεντροποιημένη υλοποίηση σε κρατικό ή διακρατικό επίπεδο.

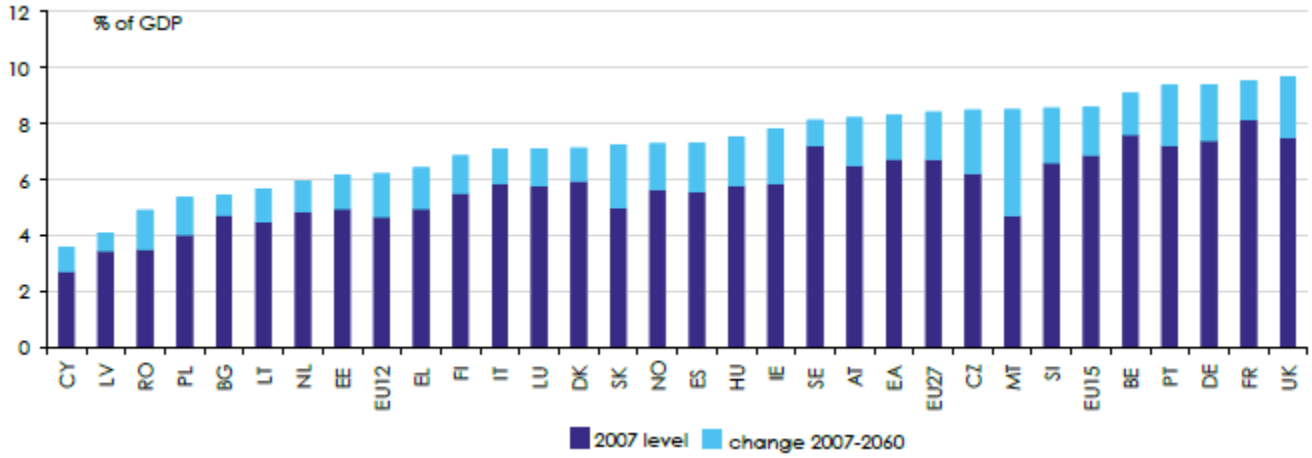
Στις Η.Π.Α. έως πρόσφατα οι προσπάθειες επικεντρώνονταν στην ενοποίηση των κέντρων που υποστηρίζουν ΗΙΦΑ., μέσω προτυποποίησης και με την εδραίωση ενός κεντρικού μη κερδοσκοπικού οργανισμού, του «Healthcare Information and Management Systems Society (HIMSS)», αλλά δίχως τα επιθυμητά αποτελέσματα. Το 2009 με το «Health Information Technology for Economic and Clinical Health Act», αλλιώς «HITECH Act», στα πλαίσια του «American Recovery and Reinvestment Act of 2009», παρέχονται κίνητρα για την υιοθέτηση ενός πανεθνικού δικτύου για την παροχή Ηλεκτρονικού Ιατρικού Φακέλου και εξασφαλίζει την κινητικότητα των αρχείων σε όλες τις πολιτείες των Η.Π.Α. Την αντίστοιχη πορεία έχει ακολουθήσει και το Ηνωμένο Βασίλειο.

Στην Ευρωπαϊκή Ένωση, έως τώρα, καμία προσπάθεια δεν είχε γίνει· εκτός της Ευρωπαϊκής κάρτας Ασφάλισης Ασθενείας, η οποία δεν πλησιάζει τον ορισμό του Ηλεκτρονικού Μητρώου Υγείας. Η πρώτη ενοποιημένη και οργανωμένη προσπάθεια επήλθε με τη «Ψηφιακή Ατζέντα για την Ευρώπη 2011 - 2020» και συγκεκριμένα την οδηγία 2011/24/EU της Commission, όπου ορίζονται τα δικαιώματα των πολιτών για πανευρωπαϊκή, διασυνοριακή παροχή υπηρεσιών υγείας, κάτι που πρέπει να έχει υλοποιηθεί έως το 2013[7]. Το άρθρο 14 ορίζει ένα δίκτυο «eHealth»(επίσημη ονομασία), το οποίο αποτελείται από τους οργανισμούς κάθε κράτους υπεύθυνα για την υλοποίηση της οδηγίας. Αυτό κατ' επέκταση οδήγησε στην ανάγκη δημιουργίας ενός συστήματος Ηλεκτρονικού Ιατρικού Μητρώου[8], κάτι που εντάχθηκε στην Ψηφιακή Ατζέντα. Μια αναφορά χρήζει και στο European Institute for Health Records, μία Μ.Κ.Ο. με σκοπό την υιοθέτηση υψηλής ποιότητας Ηλεκτρονικού Μητρώου Υγείας στην Ευρωπαϊκή Ένωση.

3. Το μέλλον της Υγείας στην Ε.Ε.

Η Ευρώπη είναι μια γηράσκων ήπειρος και κατ' επέκταση τα έξοδα για την Υγειονομική Περίθαλψη καταλαμβάνουν από 9% έως και 15% του Α.Ε.Π. των κρατών της Ε.Ε. Αυτό το ποσοστό ενδέχεται να αυξηθεί όσο αυξάνει ο μέσος όρος ηλικίας των πληθυσμών της.

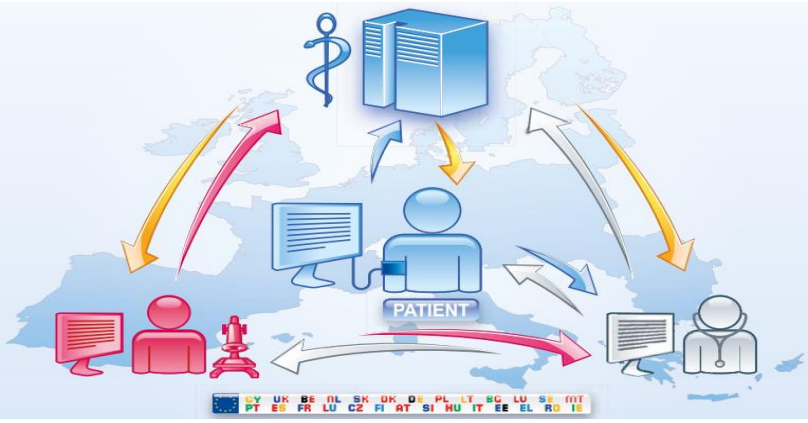
Πιο συγκεκριμένα, ένα σημαντικό ποσοστό του πληθυσμού θα χρειάζεται συνεχόμενη παρακολούθηση της κατάστασής τους, λόγω προβλημάτων υγείας. Οι χρόνιες ασθένειες αποτελούν το 80% των ασθενειών και συνεχίζουν να αυξάνονται. Το 40% των Ευρωπαίων άνω των 15(100εκ. άτομα) πάσχουν από τουλάχιστον μία χρόνια ασθένεια. Τέλος, υπολογίζεται πως το 70% του κόστους Υγειονομικής περίθαλψης αφορά χρόνιες ασθένειες, δηλαδή περί τα 700 δις €[9].



Εικόνα 1. Επιπτώσεις της δημογραφικής αλλαγής της δημόσιας δαπάνης στην Υγειονομική περίθαλψη ως ποσοστό του Α.Ε.Π.(%) [10]

Αυτές οι ανάγκες οδήγησαν την Ε.Ε. στη χάραξη πολιτικής για την Υγεία το 2020. Συγκροτήθηκε η Ομάδα Εργασίας Ηλεκτρονικής Υγείας("eHealth Task Force") και δημιούργησε ένα πλάνο για το μέλλον της Υγείας στην Ευρώπη έως το 2020[10]. Αυτό το πλάνο αποτελείται από 5 "μοχλούς", οι οποίοι αναφέρουν βασικές οδηγίες για την υλοποίησή τους και ο καθένας είναι προαπαιτούμενο για ένα πλήρως ψηφιακό σύστημα Υγείας.

3.1. Μοχλός 1: Δικά μου δεδομένα, Δικές μου αποφάσεις



Εικόνα 2. Οπτικοποίηση του Μοχλού 1

Ο "Μοχλός 1" τοποθετεί το άτομο στο κέντρο της διαδικασίας και τον καθιστά διαχειριστή των δεδομένων του. Οι υπάρχοντες νόμοι της Ε.Ε. καθορίζουν την σχέση πολίτη - δεδομένων, αλλά σπάνια υλοποιείται με αυτό κάτι τέτοιο στα υπάρχοντα συστήματα υγείας. Απαραίτητες προϋποθέσεις για την επιτυχία του "μοχλού 1" είναι να υπάρχει μια ευρέως αποδεκτή και έμπιστη διαχείριση και συλλογή δεδομένων, σαφήνεια και εξασφαλισμένη αποφυγή διακρίσεων στη χρήση των δεδομένων αυτών.

Η ηλεκτρονική επιχειρηματικότητα έχει αρχίσει να επικεντρώνεται και στον τομέα της προσωπικής Υγείας, αλλά οι υλοποιήσεις διαφέρουν πολύ μεταξύ τους, όσον αφορά την ιδιοκτησία και την ιδιωτικότητα των δεδομένων. Μερικές επιτρέπουν στους χρήστες να διαμοιραστούν εύκολα τα προσωπικά δεδομένα τους, εκθέτοντας τους σε σημείο πέραν της κατανόησης τους. Στόχος αυτών των επιχειρήσεων είναι να δημιουργούν έσοδα, μέσω της αξιοποίησης των δεδομένων χρηστών μέσω της πώλησης προϊόντων ή υπηρεσιών.

Αυτός είναι και ο λόγος που στη πρόταση αυτή συμπεριλαμβάνεται η θέσπιση πλαισίου για τα προσωπικά δεδομένα υγείας και ποιοι περιορισμοί πρέπει να υπάρχουν για την διαμοίραση τους.

Τα βασικά οφέλη αυτού του "μοχλού" είναι ότι επιτρέπει στους πολίτες να διαχειρίζονται τα δεδομένα που αφορούν την υγείας τους και – αν θέλουν – μπορούν να δέχονται εξατομικευμένη φροντίδα. Επίσης, οι ιατροί και οι επαγγελματίες υγείας μπορούν να λαμβάνουν πιο ακριβείς αποφάσεις και να κατευθύνουν τις ενέργειές τους πιο αποδοτικά με τα κατάλληλα δεδομένα.

3.2. Μοχλός 2: Απελευθέρωση Δεδομένων

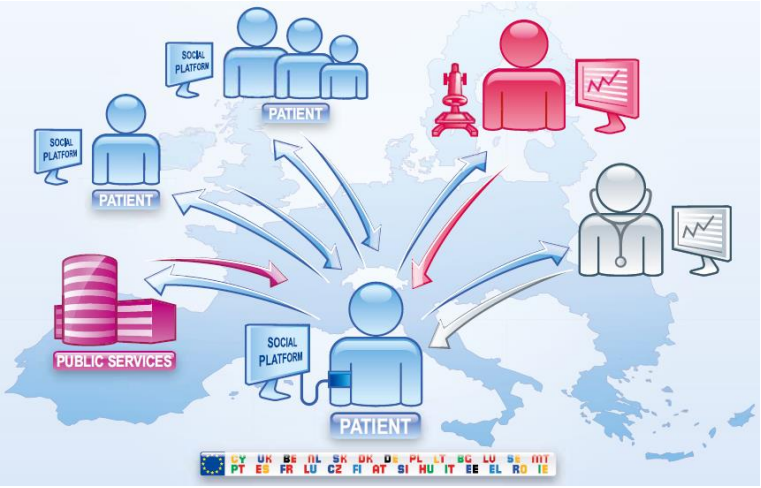
Εικόνα 3. Οπτικοποίηση του Μοχλού 2

Ερευνητές και επιχειρήσεις χρειάζονται δεδομένα για την αμέριστη λειτουργία τους, ώστε να παρέχουν εμπεριστατωμένες και ακριβείς υπηρεσίες και γνώση υψηλής ποιότητας. Αυτή τη στιγμή υπάρχουν πολλά Ιατρικά κέντρα με αφθονία αναξιοποίητων δεδομένων, τα οποία με τη κατάλληλη εξόρυξη και οργάνωσή τους μπορούν να μας προσφέρουν ένα θησαυρό πληροφοριών.

Ο "Μοχλός 2" ορίζει την ευθύνη της εκάστοτε κυβέρνησης να εξασφαλίζει ότι τα παρεχόμενα δεδομένα είναι ακριβή και αξιόπιστα, ότι μπορούν να συγκεντρωθούν με προτυποποιημένες μεθόδους και ότι είναι είναι ανώνυμα. Στη συνέχεια θα πρέπει να γίνονται διαθέσιμα σε επιχειρήσεις και ερευνητικά κέντρα, τα οποία με μία προτυποποιημένη αίτηση θα μπορούν να αξιοποιήσουν αυτά τα δεδομένα. Για όλα αυτά οι αρχικοί πάροχοι των δεδομένων – οι πολίτες/ασθενείς – θα πρέπει να έχουν δώσει τη συγκατάθεσή τους για την αξιοποίηση των δεδομένων τους.

Αυτή η κίνηση θα διευκολύνει τα κράτη να μετριάσουν τον προϋπολογισμό υγείας τους, καθώς, υπό την ομπρέλα της απελευθέρωσης δεδομένων, θα μπορούν να αναπτυχθούν εφαρμογές τηλεϊατρικής("telemedicine") και ιατρικής παρακολούθησης για τους χρόνια πάσχοντες. Επιπρόσθετα οφέλη είναι η καθιέρωση προτυποποίησης για τα ιατρικά δεδομένα, οι ενσωματωμένες υπηρεσίες και η παροχή πολλαπλών επιλογών με μειωμένο κόστος για το κράτος. Οι πολίτες - χρήστες θα ωφεληθούν από την εξατομικευμένη, αποδοτικότερη και φθηνότερη φροντίδα, αφού η Υγειονομική περίθαλψη θα έχει νέα δεδομένα να καινοτομήσει, και οι παρεχόμενες υπηρεσίες θα περιστρέφονται γύρω από τον χρήστη.

3.3. Μοχλός 3: Διασύνδεση των πάντων



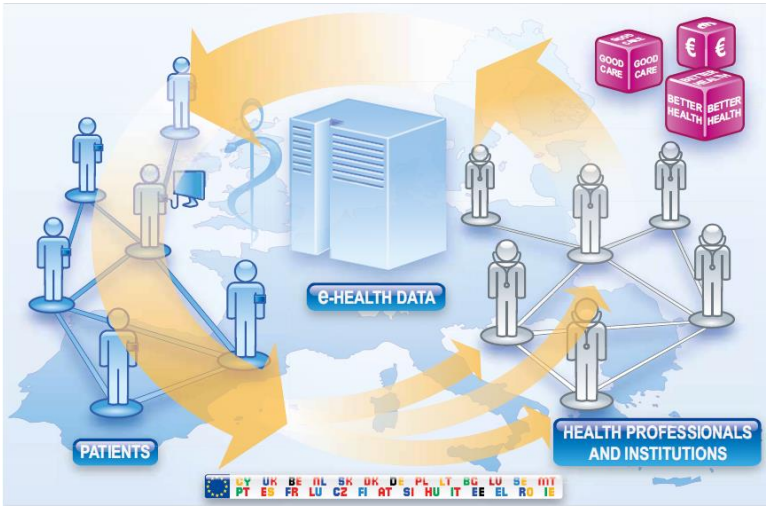
Εικόνα 4. Οπτικοποίηση του Μοχλού 3

Αυτή την εποχή πολλοί άνθρωποι, με την διάδοση των ψηφιακών κοινωνικών μέσων, έχουν αποκτήσει μια δευτερεύουσα διαδικτυακή ταυτότητα. Καθένας είναι δημιουργός και ιδιοκτήτης της πληθώρας των προσωπικών ψηφιακών δεδομένων, όπου επιλέγει που και με ποιους θα τα μοιραστεί. Έτσι και σε μία σύγχρονη κοινωνία, με ψηφιακές υποδομές υγείας, το άτομο θα μπορεί να μοιράζεται τα δεδομένα υγείας του, τα οποία έχουν τη δυνατότητα να φέρουν γεωγραφικά δεδομένα, κάτι που τα καθιστά σημαντικό παράγοντα για την Επιδημιολογία και την επιτήρηση της Δημόσιας Υγείας.

Στόχος του "Μοχλού 3" είναι να σκιαγραφήσει τη σχεδίαση ενός συστήματος, όπου οι υπηρεσίες και οργανισμοί υγείας θα μπορούν να έχουν πρόσβαση σε δεδομένα, τα οποία οι χρήστες τους έχουν εισάγει στο σύστημα, το οποίο ορίζεται στα πλαίσια του "μοχλού". Ένα γενικότερο πρόβλημα θα είναι ότι χρήστες θα απαιτούν από τους ιατρούς τους να χρησιμοποιούν τις ίδιες εφαρμογές, διαφόρων πλατφορμών, που χρησιμοποιούν οι ίδιοι. Οπότε θα πρέπει να εξασφαλιστεί η προτυποποίηση, για την εύκολη μεταφορά τέτοιων κοινωνικών - ιατρικών δεδομένων, αλλά και η τεχνολογία για την αξιοποίησή τους, ειδικά σε μια περίοδο όπου τα δεδομένα καθημερινής ζωής("social data") και οι πληροφορίες προσωπικής υγείας δύσκολα διαχωρίζονται.

Οφέλη για τους πολίτες είναι ότι θα μπορούν να λαμβάνουν συνεχόμενη ιατρική φροντίδα και συμβουλές για υγιή ζωή και όχι μόνο σε περιπτώσεις όπου χρειάζεται άμεση παρέμβαση, αλλά υπό οποιεσδήποτε συνθήκες.

3.4. Μοχλός 4: Επανάσταση Υγείας



Εικόνα 5. Οπτικοποίηση του Μοχλού 4

Η πλήρης διαφάνεια δύναται να δημιουργήσει δραστική καινοτομία στο τομέα της Υγείας. Το γεγονός ότι οι πολίτες μπορούν να έχουν πληροφορίες για την αποδοτικότητα του κάθε ιατρού και οργανισμού θα τους δώσει την ελευθερία να αποφασίζουν που θα δέχονται ιατρική φροντίδα. Αυτό θα οδηγήσει τους παρόχους ιατρικών υπηρεσιών στον εκσυγχρονισμό τους και στην υιοθέτηση πολιτικών διαφάνειας. Κάτι τέτοιο μπορεί να επιτευχθεί μέσω ενός δικτύου “e-Health”, που ορίζεται στον “Μοχλό 4”.

Ο τομέας της Υγείας έχει αρχίσει να γίνεται δυσδιάκριτος από τους υπόλοιπους τομείς. Οι τομείς της κοινωνικής φροντίδας και πρόνοιας θα πρέπει να ενσωματωθούν, για να παρέχονται αφανείς και αποδοτικές υπηρεσίες στο πολίτη. Ο, δε, πολίτης πλέον απαιτεί περισσότερη διαφάνεια και μεγαλύτερο έλεγχο στη σχέση του με τον ιατρό. Τέλος, η διαφάνεια είναι κίνητρο αποδοτικότητας στη καλή διαχείριση της ιατροφαρμακευτικής φροντίδας.

Για το “μοχλό 4” απαιτείται η αφοσίωση των ατόμων που χαράσσουν τις πολιτικές για ολοκληρωτική αλλαγή του Συστήματος Υγείας, αλλά και του προσωπικού υγείας για παραχώρηση του κυρίαρχου ρόλου στην σχέση Ιατρού - ασθενή.

Τα οφέλη που απορρέουν για τους πολίτες είναι ότι τους ενεργοποιεί και κινητοποιεί στο να συμμετέχουν στη καλή διαχείριση της υγείας τους. Οι επαγγελματίες υγείας θα μπορούν να προσφέρουν ενοποιημένες υπηρεσίες σε κοινωνικό επίπεδο, με αντίστοιχα κίνητρα αναλόγως της απόδοσής τους. Σε κρατικό επίπεδο, θα μπορεί να γίνεται πιο οργανωμένη και στοχευμένη χρηματοδότηση, εξασφαλίζοντας καλύτερη απόδοση και “Επιστροφή Στην Επένδυση”.

3.5. Μοχλός 5: Συμπερίληψη όλων

Εικόνα 6. Οπτικοποίηση του Μοχλού 5

Η επιτροπή του Παγκόσμιου Οργανισμού Υγείας(WHO) αναφέρει πως σε άνισες κοινωνίες υπάρχουν χειρότερες συνθήκες υγείας. Ανάμεσα στις χώρες της Ε.Ε. υπάρχει μεγάλη ανισότητα με διαφορές προσδόκιμου ζωής έως και 15 χρόνια, ανά κράτος και κοινωνική τάξη. Ο "Μοχλός 5" ορίζει το πως μπορούμε να βελτιώσουμε αυτές τις ανισότητες μέσω εργαλείων και δικτύων "e-Health"("μοχλός 4").

Τα νέα εργαλεία Πληροφορικής και Τηλεπικοινωνιών μπορούν να βοηθήσουν να μειωθεί αισθητά αυτό το χάσμα, αλλά πρέπει να σχεδιαστούν με τέτοιο τρόπο ώστε να περιλαμβάνονται όλες οι κοινωνικές ομάδες, με διαφορετικό επίπεδο παιδείας και τεχνικές ικανότητες. Παρ' όλα αυτά υπάρχουν ομάδες πληθυσμού που δεν έχουν πρόσβαση σε τέτοια εργαλεία. Πρέπει να προσεχθούν ιδιαίτερα οι ανάγκες αυτών των πληθυσμών, ώστε να αποφευχθεί η στέρηση πολιτικών δικαιωμάτων τους.

Για να αποδώσει ο "μοχλός 5" είναι απαραίτητη η ίση πρόσβαση, η προώθηση Ψηφιακής Παιδείας και η χρήση εργαλείων "e-Health". Αν επιτύχει, το επίπεδο Υγείας θα αυξηθεί αισθητά, τα πολύπλοκα προβλήματα θα αντιμετωπίζονται ευκολότερα, οι πόροι θα διαθέτονται αποδοτικότερα και οι πολίτες θα είναι πιο ενημερωμένοι για θέματα υγείας και υγιούς ζωής.

3.6. Προτάσεις για υλοποίηση και σύνοψη των Μοχλών

Όλοι οι "Μοχλοί" αλλαγής μαζί με τις ομάδες ενδιαφερόμενων, αλλά και το βαθμό επίδρασης(στις ομάδες ενδιαφερόμενων) – εάν υλοποιηθούν – παρουσιάζονται στο παρακάτω πίνακα.

Ομάδα Ενδιαφερόμενων	**Μοχλός 1**	**Μοχλός 2**	**Μοχλός 3**	**Μοχλός 4**	**Μοχλός 5**
(σύνοψη, οφέλη ή και αποτελέσματα)	Ασθενείς και Οργανισμοί μοιράζονται τα δεδομένα τους. Ευέλικτοι μηχανισμοί συναίνεσης.	Ιατρικά αποτελέσματα και δεδομένα απόδοσης μοιράζονται ελεύθερα με υψηλή διαφάνεια	Η διαχείριση της τεχνολογίας και της πληροφορίας οδηγεί τον ρυθμό καινοτομίας.	Διασύνδεση δεδομένων καθημερινότητας με Υγείας, πολλαπλά εργαλεία και εφαρμογές από επιχειρηματίες	Οφέλη και συνεισφορά όλων από την καθολική συμμετοχή σε "eHealth" δίκτυα και εργαλεία.
Πολίτες και ασθενείς	**Υψηλός**	**Υψηλός**	**Υψηλός**	**Υψηλός**	**Υψηλός**
Δημιουργοί πολιτικής	**Μέσος**	**Υψηλός**	**Μέσος**	**Μέσος**	**Μέσος**
Επαγγελματίες Υγείας	**Μέσος**	**Υψηλός**	**Υψηλός**	**Υψηλός**	**Υψηλός**
Ασφαλιζόμενοι και ασφαλιστικές	**Μέσος**	**Υψηλός**	**Υψηλός**	**Μέσος**	**Χαμηλός**
Παροχείς υπηρεσιών και Διαχειριστές	**Χαμηλός**	**Υψηλός**	**Υψηλός**	**Υψηλός**	**Υψηλός**
Ερευνητές	**Υψηλός**	**Υψηλός**	**Μέσος**	**Υψηλός**	**Υψηλός**

Πίνακας 1. Επίδραση των "μοχλών" δράσεων σε ενδιαφερόμενες ομάδες

Ο παρακάτω πίνακας συνοψίζει τις προτεινόμενες δράσεις της “eHealth Task Force” για την υιοθέτηση και εφαρμογή όλων των μοχλών.

Προτάσεις	**Προτεινόμενες δράσεις**
I. Μια νέα νομική βάση για Ιατρικά δεδομένα στην Ευρώπη	Γρήγορες κινήσεις για την νομική διαύγεια της πανευρωπαϊκής χρήσης των δεδομένων υγείας, καθιερώνοντας ισχυρής προστασίας και παροχής σταθερής αγοράς για την ενθάρρυνση καινοτομίας. Αυτό θα πρέπει να καθορίσει τις διάφορες χρήσεις των δεδομένων και ένα ανανεωμένο πλαίσιο προσέγγισης για συναίνεση.
II. Δημιουργία ομάδας “Πρωτοστατών” κρατών- μελών αφοσιωμένα σε “Open Data” και “eHealth”	Οι δημόσιες Αρχές θα δημιουργήσουν ομάδα “πρωτοστατών” για ταχεία ανάπτυξη. Αυτή η ομάδα μπορεί να παρέχει ηγεσία και έμπνευση για χώρες της Ε.Ε. και άλλες τρίτες. Η Ε.Ε. μπορεί να βοηθήσει στη διάδοση τεχνογνωσίας, προωθώντας την υιοθέτηση αποδεδειγμένων τεχνολογιών και στη κοινοποίηση των αποτελεσμάτων νέων δράσεων
III. Υποστήριξη παιδείας υγείας	Ενίσχυση της κατανόησης του κοινού για τα οφέλη του “eHealth” στη μέτρηση, διαχείριση και παρακολούθηση της ευεξίας τους. Ενίσχυση της επίγνωσης για τα δεδομένα που συλλέγονται και τους διάφορους τρόπους που μπορούν να χρησιμοποιηθούν για το καλό του ατόμου και του συστήματος υγείας. Παροχή μεσολαβητών και εξασφάλιση ικανοτήτων χρήσης για τις ευαίσθητες ομάδες.
IV. Αξιοποίηση της δύναμης των δεδομένων	Δημιουργία κουλτούρας διαφάνειας στην υγεία. Συγκριτική αξιολόγηση και παρακολούθηση του Συστήματος Υγείας. Ενθάρρυνση της ενσωμάτωσης των δεδομένων σε ευρύτερες Ευρωπαϊκές συλλογές δεδομένων και ενίσχυση πρόσβασης ερευνητών. Καλύτερη ενσωμάτωση έρευνας και πρακτικής υγείας.
V. Επαναπροσανατολισμός της χρηματοδότησης και των πολιτικών της Ε.Ε.	Απαίτηση διαφάνειας από τους οργανισμούς υγείας που χρησιμοποιούν ηδεία συστήματα υγείας μέσω κριτηρίων προμήθειας και χρηματοδότησης. Διάθεση πόρων από την Ε.Ε. για καινοτομία γύρω από τον χρήστη, υποστήριξη για γρήγορη προτυποποίηση και διατήρηση χαμηλών απαιτήσεων για πρόσβαση.

Πίνακας 2. Προτεινόμενες δράσεις της “eHealth Task Force” για την υλοποίηση των “μοχλών”

4. Πρότυπα HL7 και DICOM

Για την μεταφορά, παρουσίαση, διασύνδεση και αποθήκευση δεδομένων υγείας έχουν προταθεί και ήδη υπάρχουν πολλά πρότυπα τα οποία χρησιμοποιούνται από διάφορους οργανισμούς. Επιπλέον, υπάρχουν διάφορα πρότυπα για την κατηγοριοποίηση ασθενειών, ιών και ιατρικών όρων. Εμείς θα ασχοληθούμε με τα πρότυπα HL7 και DICOM, τα οποία είναι από τα πιο ευρέως διαδεδομένα σε υλοποιήσεις "ΗΜΥ" και "ΗΙΦΑ".

4.1. HL7(Health Level 7)

Το πρωτόκολλο HL7 – από τον ομώνυμο μη κερδοσκοπικό οργανισμό – ορίζει το περιεχόμενο και τη δομή των ιατρικών δεδομένων και δεδομένων ασφάλισης υγείας, τα οποία ανταλλάσσονται μεταξύ συστημάτων υγείας ή και οργανισμών. Σημαντικό είναι πως το πρωτόκολλο αυτό δεν είναι ανοιχτό, έτσι κάθε χρήστης του πρωτοκόλλου πρέπει να πληρώνει ετήσια συνδρομή[11] κάτι που θα αλλάξει στο κοντινό μέλλον.

Πρότυπα του HL7[12]:

- Version 2.x Messaging Standard – προδιαγραφές διαλειτουργικότητας για ιατρικές και φαρμακευτικές συναλλαγές
- Version 3 Messaging Standard – an interoperability specification for health and medical transactions, based on RIM
- Version 3 Rules/GELLO – Μια τυποποιημένη γλώσσα έκφρασης για αποφάσεις κλινικής υποστήριξης.
- Arden Syntax – Μια γραμματική για την αναπαράσταση ιατρικών καταστάσεων και προτάσεων ως "Medical Logic Module (MLM)"
- Clinical Context Object Workgroup (CCOW) – προδιαγραφές διαλειτουργικότητας για την οπτική ενσωμάτωση των εφαρμογών χρηστών.
- Claims Attachments – Ένα "Standard Healthcare Attachment" για την ενίσχυση μιας άλλης ιατροφαρμακευτικής συναλλαγής
- Clinical Document Architecture (CDA) – Ένα μοντέλο συναλλαγών για κλινικά έγγραφα, βασισμένο στο πρότυπο HL7 έκδοση 3
- Electronic Health Record (EHR) / Personal Health Record (PHR) – Για υποστήριξη αυτών των εγγράφων, ορίζεται μια προτυποποιημένη περιγραφή ιατρικών λειτουργιών και λειτουργιών υγείας· νέων ή μελλοντικών
- Structured Product Labeling (SPL) – Οι δημοσιευμένες πληροφορίες που συνοδεύουν ένα φάρμακο, βασισμένο στο πρότυπο HL7 έκδοση 3

4.1.1 Μέθοδοι του HL7

Πλαίσιο Ενήμερων Υπηρεσιών Διαλειτουργικότητας[13]

Το "Πλαίσιο Ενήμερων Υπηρεσιών Διαλειτουργικότητας"(ΠΕΥΔ) του HL7 παρέχει συνοχή μεταξύ όλων των αντικειμένων στο πρότυπο αυτό και δίνει την δυνατότητα προτυποποιημένης προσέγγισης σε ανάπτυξη και υλοποίηση με "Enterprise Architecture (EA)", καθώς και ένα τρόπο να μετρηθεί η συνοχή της.

Αποτελεί ένα τρόπο για τη δημιουργία προδιαγραφών, οι οποίες περιγράφουν ρητά τη διακυβέρνηση και τη συμμόρφωση απαραίτητων σημασιολογικών συμπεριφορών για την υλοποίηση υπολογίσιμης διαλειτουργικότητας. Η προσβλέπουσα τεχνολογία μετάδοσης πληροφορίας μπορεί να λειτουργεί με μηνύματα, εναλλαγή αρχείων ή υπηρεσιών.

Το ΠΕΥΔ είναι το απαραίτητο πλαίσιο για την εκλογίκευση της διαλειτουργικότητας με άλλα πρότυπα, αλλά όχι πλήρης λύση για διαχείριση σε επίπεδο "Enterprise Architecture".

Συντακτικό Arden[13]

Το "συντακτικό Arden" είναι μια γλώσσα για την κωδικοποίηση ιατρικής γνώσης. Υιοθετήθηκε και επιβλέπεται από τον οργανισμό HL7 με το συντακτικό Arden 2.0. Αυτά τα Ιατρικά Λογικά Εξαρτήματα(ΙΛΕ) χρησιμοποιούνται σε συνθήκες κλινικής, καθώς περιέχουν επαρκή γνώση για αυτόνομες ιατρικές αποφάσεις. Μπορούν να παράξουν ειδοποιήσεις, διαγνώσεις και ερμηνείες με διάφορες λειτουργίες διασφάλισης ποιότητας, καθώς και λειτουργίες διοικητικής υποστήριξης.

MLLP[13]

Ένα μεγάλο ποσοστό των μηνυμάτων στο HL7 μεταφέρονται με το πρωτόκολλο "Minimal Lower Layer Protocol (MLLP ή απλά LLP)"[14]. Για μετάδοση μέσω TCP/IP, προστίθενται στο μήνυμα χαρακτήρες header και trailer για να αναγνωρίζεται η αρχή και το τέλος του, διότι το TCP/IP είναι μία συνεχής ροή bytes.

Μια παραλλαγή του MLLP, είναι το "Hybrid Lower Layer Protocol (HLLP)", το οποίο περιλαμβάνει και άθροισμα ελέγχου για να ελέγχεται η ακεραιότητα των μηνυμάτων.

CCOW[13]

Το "Clinical Context Object Workgroup(CCOW)" είναι ένα πρότυπο, το οποίο σχεδιάστηκε για να επιτρέπει ανόμοιες εφαρμογές να μοιράζονται σε πραγματικό χρόνο, σε επίπεδο διεπαφής χρήστη, δεδομένα χρηστών και ασθενών. Οι υλοποιήσεις CCOW απαιτούν, τυπικά, ένα σύστημα "CCOW vault" για την διαχείριση ασφάλειας μεταξύ των εφαρμογών.

Προδιαγραφές λειτουργίας ΗΜΥ και ΗΙΦ[13]

Περιλαμβάνονται προδιαγραφές λειτουργίας για "Ηλεκτρονικό Μητρώο Υγείας".

4.1.2 Εκδόσεις του HL7 και χαρακτηριστικά τους

α. HL7, έκδοση 2.x

Η έκδοση 2 του προτύπου HL7, εισήχθη το 1989(με την έκδοση 2.7 να θεσπίζεται ως πρότυπο ANSI το 2007) και έχει ως στόχο την υποστήριξη των ροών εργασίας νοσοκομείων[15].

Μηνύματα της έκδοσης 2.x

Η έκδοση αυτή ορίζει διάφορα ηλεκτρονικά μηνύματα για την υποστήριξη διοικητικών, λογιστικών, οικονομικών, καθώς και ιατρικών διαδικασιών. Οι πολλές αναβαθμίσεις στο πρότυπο, έως την έκδοση 2.7, κατέστησαν απαραίτητη την υποστήριξη συμβατότητας και για πρότερες εκδόσεις.

Η μορφή των μηνυμάτων είναι σε αναγνώσιμη μορφή(συγκεκριμένα σε ASCII), με συντακτικό κωδικοποίησης βασισμένο σε γραμμές/τμήματα και χαρακτήρες/οριοθέτες[16]. Τα τμήματα αυτά έχουν πεδία τα οποία διαχωρίζονται με τον οριοθέτη και μπορούν να έχουν υπό-πεδία που διαχωρίζονται με έναν αντίστοιχο υπό- οριοθέτη. Οι προεπιλεγμένοι οριοθέτες είναι η κάθετος (|) για τα πεδία, το σημείο παράλειψης(^) για τα τμήματα και ο συμπλεκτικός σύνδεσμος(&) για τα υπό- τμήματα. Η περισπωμένη(~) είναι η προεπιλογή για την επανάληψη[17].

Το ακόλουθο είναι ένα παράδειγμα ενός αρχείου εισαγωγής ασθενούς. Το “MSH” είναι το αρχείο εισαγωγής, το “PID” είναι η “Ταυτότητα Ασθενούς”. Εισήγα κενές γραμμές μεταξύ τομέων/γραμμών για ευκολία ανάγνωσης, ενώ στην πραγματικότητα δεν είναι αυτή η μορφή του αρχείου.

```
MSH|^~\&|MegaReg|XYZHospC|SuperOE|XYZImgCtr|20060529090131-0500||ADT^A01^ADT_A01|01052901|P|2.5

EVN||200605290901||||200605290900

PID|||56782445^^^UAReg^PI||KLEINSAMPLE^BARRY^Q^JR||19620910|M||2028-9^^HL70005^RA99113^^XYZ|260
GOODWIN    CREST    DRIVE^^BIRMINGHAM^AL^35    209^^M~NICKELL’S    PICKLES^10000    W    100TH
AVE^BIRMINGHAM^AL^35200^^O |||||||0105I30001^^^99DEF^AN

PV1||I|W^389^1^UABH^^^^3||||12345^MORGAN^REX^J^^^MD^0010^UAMC^L||678
90^GRAINGER^LUCY^X^^^MD^0010^UAMC^L|MED|||||A0||13579^POTTER^SHER
MAN^T^^^MD^0010^UAMC^L|||||||||||||||||||||||||||200605290900

OBX|1|NM|^Body Height||1.80|m^Meter^ISO+|||||F

OBX|2|NM|^Body Weight||79|kg^Kilogram^ISO+|||||F

AL1|1||^ASPIRIN

DG1|1||786.50^CHEST PAIN, UNSPECIFIED^I9|||A
```

Το σημαντικό που εισήγαγε το HL7 2.x είναι η διαλειτουργικότητα μεταξύ ηλεκτρονικών "Συστημάτων Διαχείρισης Ασθενών(ΣΔΑ)", "Ηλεκτρονικών συστημάτων Διαχείρισης Ιατρικής Πρακτικής(ΗΔΙΠ)", "Συστημάτων Πληροφοριών Εργαστηρίων(ΣΠΕ)", Διατροφικών συστημάτων , Φαρμακευτικών συστημάτων και συστημάτων Χρέωσης, αλλά και "Ηλεκτρονικών Μητρώων Υγείας" και "Ηλεκτρονικών Φακέλων Ασθενούς".

β. HL7, έκδοση 3

Σκοπός της έκδοσης 3 είναι να να υποστηρίζει τις ροές εργασίας σε όλο το φάσμα της Υγειονομικής περίθαλψης. Η ανάπτυξη ξεκίνησε το 1995 και η πρώτη έκδοση κυκλοφόρησε το 2005. Αντίθετα με τον προκάτοχό του βασίζεται σε επίσημη μεθοδολογία(HL7 Development Framework) και σε αντικειμενοστραφείς αρχές.

RIM - ISO/HL7 21731

Το "Reference Information Model(RIM)"[18] είναι από τα βασικότερα κομμάτια της διαδικασίας ανάπτυξης της έκδοσης 3 και ουσιώδες μέρος της μεθοδολογίας ανάπτυξής του. Εκφράζει τα περιεχόμενα δεδομένων, απαραίτητα σε συγκεκριμένο κλινικό ή διοικητικό πλαίσιο και παρέχει μία ρητή αναπαράσταση των σημασιολογικών και λεκτικών συνδέσμων μεταξύ της πληροφορίας στα πεδία των μηνυμάτων του HL7. Σύμφωνα με τον οργανισμό, είναι απαραίτητο μοντέλο για καλύτερη ακρίβεια και βοηθά στην μείωση του κόστους υλοποίησης[19].

HL7 Development Framework - ISO/HL7 27931

Το "HL7 Version 3 Development Framework (HDF)" είναι μία διαδικασία υπό συνεχή εξέλιξη, η οποία στοχεύει στην ανάπτυξη απαραίτητων προδιαγραφών για τη διαλειτουργικότητα μεταξύ συστημάτων Ιατρικής Περίθαλψης. Το RIM, το λεξικό, οι προδιαγραφές και οι οδηγούμενες-από-μοντέλα διαδικασίες αναλύσεων και σχεδιασμού του HL7 συνδυάζονται για να γίνει η έκδοση 3 μία αυτόνομη μεθοδολογία για την συναινετική ανάπτυξη προτύπων με βάση την διαλειτουργικότητα μεταξύ "Πληροφοριακών Συστημάτων Ιατρικής Περίθαλψης". Αποτελεί το πιο πρόσφατο πλαίσιο εργασίας της μεθοδολογίας ανάπτυξης για την έκδοση 3.

Το "HDF" περιγράφει, πέρα από τα μηνύματα, τις διεργασίες, τα εργαλεία, τους κανόνες και τα αντικείμενα σχετικά με την ανάπτυξη όλων των προδιαγραφών για το HL7. Μελλοντικά θα περιλαμβάνει όλες τις προδιαγραφές του πρωτοκόλλου.

Μηνύματα της έκδοσης 3

Στην έκδοση 3 ορίζονται πληθώρα τύπων ηλεκτρονικών μηνυμάτων(ονομάζονται "διαδράσεις") για την υποστήριξη όλων των σχετικών ροών εργασίας και βασίζονται στο "συντακτικό XML".

γ. Clinical Document Architecture - ISO/HL7 27932

Η “Clinical Document Architecture(CDA)” είναι ένα πρότυπο βασισμένο στο XML, που αποβλέπει να καθορίσει την κωδικοποίηση, τη δομή και τη σημασιολογία των κλινικών αρχείων για μετάδοση μεταξύ φορέων και συστημάτων[20].

4.2 DICOM

Το “Digital Imaging and Communications in Medicine(DICOM)” είναι ένα πρότυπο για την διαχείριση, αποθήκευση, εκτύπωση και μετάδοση ιατρικών απεικονίσεων. Περιλαμβάνει τον ορισμό μιας συγκεκριμένης μορφής αρχείου, καθώς επίσης και ένα πρωτόκολλο δικτυακής επικοινωνίας, το οποίο χρησιμοποιεί το TCP/IP. Η ανταλλαγή των αρχείων DICOM μπορεί να γίνει μόνο μέσω οντοτήτων που υποστηρίζουν το πρότυπο. Τα πνευματικά δικαιώματα ανήκουν στον οργανισμό “NEMA”.

4.2.1 Μέρη του DICOM

Το πρότυπο χωρίζεται σε ξεχωριστά, αλλά συσχετισμένα μεταξύ τους, μέρη τα οποία ενημερώνονται συνεχώς. Ακολουθεί η έκδοση του 2011(τα μέρη 9 και 13 αποσύρθηκαν), χωρίς τις αλλαγές που εγκρίθηκαν το 2012[21].

- Μέρος 1: Εισαγωγή και Επισκόπηση
- Μέρος 2: Συμμόρφωση
- Μέρος 3: Ορισμοί Αντικειμένων Πληροφορίας
- Μέρος 4: Προδιαγραφές Κλάσεων Υπηρεσιών
- Μέρος 5: Δομές Δεδομένων και Κωδικοποίηση
- Μέρος 6: Λεξικό Δεδομένων
- Μέρος 7: Ανταλλαγή Μηνυμάτων
- Μέρος 8: Υποστήριξη Επικοινωνίας Δικτύων για Ανταλλαγή Μηνυμάτων
- Μέρος 10: Αποθήκευση Μέσων και μορφή Φακέλου για Ανταλλαγή Μέσων
- Μέρος 11: Προφίλ Εφαρμογής Αποθήκευσης Μέσων
- Μέρος 12: Μορφή Μέσων και Φυσικά Μέσα για Ανταλλαγή Μέσων
- Μέρος 14: Βασική Λειτουργία Προβολής σε κλίμακα του Γκρι
- Μέρος 15: Προφίλ Ασφάλειας και Διαχείρισης Συστήματος
- Μέρος 16: Πηγές Χαρτογράφησης Περιεχομένου
- Μέρος 17: Επεξηγητικές Πληροφορίες
- Μέρος 18: Web Access to DICOM Persistent Objects (WADO)
- Μέρος 19: Φιλοξενία Εφαρμογών
- Μέρος 20: Μεταλλαγή του DICOM σε και από πρότυπα

4.2.2 Μορφή Δεδομένων DICOM

Το DICOM διαφέρει σε μερικές “μορφές δεδομένων”, στο γεγονός ότι ομαδοποιεί την πληροφορία σε σετ από δεδομένα. Αυτό, παραδείγματος χάριν, σημαίνει ότι το αρχείο μιας ακτινογραφίας στήθους ακτίνων - X περιέχει το αναγνωριστικό(ID) ασθενή, έτσι ώστε να μην αποχωριστεί, κατά λάθος, η εικόνα από τις πληροφορίες.

Ένα “αντικείμενο δεδομένων DICOM”(δηλαδή το αρχείο) αποτελείται από αρκετά χαρακτηριστικά όπως όνομα, ID, κτλ. και πιο συγκεκριμένα αποτελείται από ένα ειδικό χαρακτηριστικό με τα δεδομένα “pixel”, το οποίο είναι μοναδικό. Για αρκετές παραλλαγές αυτό εκπροσωπεί μία και μόνο εικόνα, αλλά μπορεί να περιέχει και πολλαπλές τριών ή και τεσσάρων διαστάσεων. Τα “δεδομένα pixel” μπορούν να συμπιεστούν με διάφορα πρότυπα όπως JPEG, Lossless JPEG, JPEG 2000 και RLE. Επιπροσθέτως, η μορφή συμπίεσης LZW μπορεί να χρησιμοποιηθεί για όλο το αρχείο και όχι μόνο τα δεδομένα “pixel”, αλλά κάτι τέτοιο είναι σπάνιο.

Για τα αντικείμενα DICOM χρησιμοποιούνται, κυρίως, τρία διαφορετικά συστήματα κωδικοποίησης “Στοιχείων Δεδομένων”. Πιο συνηθισμένη είναι η “Ρητή Απεικόνιση Τιμών(VR)”, όπου η μορφή(οι σημαντικές απεικονίσεις) για κάθε αντικείμενο είναι: GROUP (2 bytes), ELEMENT (2 bytes), VR (2 bytes), LengthInByte (2 bytes) Data (κυμαινόμενου μεγέθους). Τα υπόλοιπα “Ρητά” ή “Έμμεσα Στοιχεία Δεδομένων” περιγράφονται εκτενέστατα στο “Μέρος 5” του προτύπου.

Η ίδια βασική δομή χρησιμοποιείται για όλες τις εφαρμογές, συμπεριλαμβανομένων της δικτυακής χρήσης και της χρήσης αρχείων, μόνο που στη τελευταία κατά την εγγραφή προστίθεται και μία “κεφαλίδα”. Ο παρακάτω πίνακας περιλαμβάνει τις απεικονίσεις δεδομένων, όπως ορίζονται στο “Μέρος 3”.

Value Representation	Description
AE	Application Entity(Οντότητα Εφαρμογής)
AS	Age String(Συμβολοσειρά Ηλικίας)
AT	Attribute Tag(Ετικέτα Χαρακτηριστικών)
CS	Code String(Συμβολοσειρά Κώδικα)
DA	Date(Ημερομηνία)
DS	Decimal String(Δεκαδική Συμβολοσειρά)
DT	Date/Time(Ημερομηνία/Ώρα)
FL	Floating Point Single (4 bytes)()
FD	Floating Point Double (8 bytes)
IS	Integer String(Συμβολοσειρά Ακέραιου)
LO	Long String
LT	Long Text
OB	Other Byte
OF	Other Float
OW	Other Word
PN	Person Name(Όνομα Ατόμου)
SH	Short String
SL	Signed Long
SQ	Sequence of Items(Ακολουθία Αντικειμένων)
SS	Signed Short
ST	Short Text
TM	Time(Ώρα)
UI	Unique Identifier(Μοναδικό Αναγνωριστικό)
UL	Unsigned Long
UN	Unknown(Άγνωστο)
US	Unsigned Short
UT	Unlimited Text(Απεριόριστο Κείμενο)

Πίνακας 3. Απεικονίσεις δεδομένων ενός αρχείου DICOM

4.2.3 Υπηρεσίες DICOM

Το DICOM αποτελείται από πολλές διαφορετικές υπηρεσίες και οι περισσότερες περιλαμβάνουν μετάδοση δεδομένων σε δίκτυο. Οι υπηρεσίες που περιλαμβάνονται στο πρότυπο:

Υπηρεσία αποθήκευσης

Η υπηρεσία αποθήκευσης χρησιμοποιείται για την αποστολή εικόνων ή άλλων αντικειμένων σε ένα τερματικό εργασίας ή σε ένα "σύστημα αποθήκευσης και επικοινωνίας εικόνων(PACS)".

Υπηρεσία Δέσμευσης Αποθήκευσης

Η υπηρεσία δέσμευσης αποθήκευσης χρησιμοποιείται για να επιβεβαιωθεί ότι μια εικόνα έχει μόνιμα αποθηκευθεί από μία συσκευή. Ο "Χρήστης Υπηρεσιών Κλάσεων(SCU, δηλαδή ο πελάτης)" ή ένα τερματικό εργασίας λαμβάνουν και χρησιμοποιούν την επιβεβαίωση αυτή από τον "Πάροχο Υπηρεσιών Κλάσεων(SCP, δηλαδή ο διακομιστής)".

Υπηρεσία Ερωτημάτων/Ανάκτησης

Αυτή επιτρέπει ένα σταθμό εργασία να βρει λίστες εικόνων ή αντικειμένων και να τους ανακτήσει από ένα PACS.

Υπηρεσία Λίστας Εργασιών "Modality"

Αυτή η υπηρεσία δίνει την δυνατότητα σε ένα κομμάτι εξοπλισμού("modality") να αποκτήσει πληροφορίες ασθενών και προγραμματισμένων εξετάσεων με ηλεκτρονικό τρόπο, έτσι αποφεύγοντας την πληκτρολόγηση των πληροφοριών κάθε φορά και κατ' επέκταση την ελαχιστοποίηση λαθών.

Υπηρεσία Βηματικής Διεργασίας από "Modality"

Συμπληρωματική υπηρεσία της προηγούμενης, η οποία επιτρέπει στο κομμάτι εξοπλισμού να στέλνει αναφορά για μια εκτελεσμένη εξέταση, συμπεριλαμβανομένων των δεδομένων για τις ληφθείσες εικόνες, διάρκεια εξέτασης, αρχή και τέλος της, δόσεις που χορηγήθηκαν κτλ. Βοηθάει στην καλύτερη χρήση των πόρων και των δεδομένων από τα τμήματα ραδιολογίας.

Υπηρεσία Εκτύπωσης

Η υπηρεσία εκτύπωσης χρησιμοποιείται για να στείλει εικόνες σε έναν εκτυπωτή DICOM, συνήθως για να εκτυπωθεί ένα φιλμ ακτίνων - Χ. Υπάρχει μια σταθερή βαθμονόμηση(ορίζεται στο "Μέρος 14") για την εξασφάλιση συνοχής μεταξύ διάφορων συσκευών προβολής, συμπεριλαμβανομένης της εκτύπωσης φυσικών εγγράφων.

Μέσα Εκτός - Σύνδεσης (DICOM αρχεία)

Τα αρχεία μέσων εκτός - σύνδεσης είναι μια πρόσφατη προσθήκη και συναντώνται στο "Μέρος 10". Περιγράφεται το τρόπο αποθήκευσης πληροφοριών ιατρικών απεικονίσεων σε αφαιρέσιμα μέσα. Είναι απαραίτητο να συμπεριλαμβάνονται μεταπληροφορίες("metadata") Φακέλου. Το πρότυπο DICOM περιορίζει το όνομα αρχείου σε 8 χαρακτήρες και δεν πρέπει να υπάρχει η δυνατότητα εξόρυξης πληροφορίας από αυτά. Συνήθως, τα αρχεία έχουν επέκταση ".dcm" και ο "τύπος MIME" είναι application/dicom.

5. Πάροχοι Ηλεκτρονικού Μητρώου Υγείας.

Υπάρχουν διάφοροι πάροχοι, οι οποίοι παρέχουν ηλεκτρονικές υπηρεσίες για ασθενείς, γιατρούς και ιατρικά ιδρύματα. Στις ΗΠΑ η πιο μεγάλη βιομηχανία είναι αυτή της Υγείας, αλλά η πολυπλοκότητά της και τα νομικά πλαίσια έκαναν τους επιχειρηματίες διστακτικούς στο να επενδύσουν. Θα αναφερθούν μερικά παραδείγματα παρόχων που κινούνται προς την κατεύθυνση του Ηλεκτρονικού Μητρώου Υγείας, και κυρίως εκείνοι, οι οποίοι έχουν στόχο την αμεσότητα στη σχέση ιατρού ασθενή. Ο λόγος αναφοράς τους είναι ότι η εφαρμογή που αναπτύσσεται και η ιστοσελίδα που βασίζεται αυτή η εφαρμογή έχουν ακριβώς αυτό τον σκοπό.

Αξίζει να σημειωθεί ότι υπάρχει μεγάλο ενδιαφέρον πλέον σε αυτό το τομέα και – παράδειγμα – στις ΗΠΑ τον Ιανουάριο του 2012 δημιουργήθηκε μια ένωση από επενδυτές στο τομέα της υγείας, με πρωτοστάτη τον κολοσσό “General Motors”, με ονομασία “StartupHealth”[22].

5.1 Microsoft HealthVault

Το “HealthVault”[57] από την “Microsoft” είναι μια διαδικτυακή πλατφόρμα για την διαχείριση και αποθήκευση δεδομένων ασθενών, καθώς και για την επικοινωνία τους με ιδρύματα και ιατρούς. Είναι διαθέσιμη στους πολίτες του Ηνωμένου Βασιλείου και στις ΗΠΑ. Επιπλέον, υποστηρίζονται συσκευές οι οποίες μπορούν να επικοινωνούν με τη πλατφόρμα και να προσθέτουν δεδομένα για τους χρήστες. Μπορεί να γίνει διαχείριση πολλαπλών λογαριασμών(π.χ. ένας γονέας για τα παιδιά του) και διαμοίραση δικαιωμάτων σε άλλους χρήστες της πλατφόρμας, όπως ένας γονέας, συγγενής, ιατρός κτλ.

Πέραν από μια ευρεία γκάμας συσκευών υποστηρίζει και γνωστά πρότυπα ιατρικής απεικόνισης, όπως το DICOM. Οι χρήστες μπορούν να ανεβάζουν/κατεβάζουν ολόκληρα αρχεία μέσω της πλατφόρμας. Τέλος, γίνεται μια επέκταση των υπηρεσιών του σε ολόκληρο το φάσμα της Ιατρικής Περίθαλψης με Φαρμακεία, Ιατρικά ινστιτούτα, Ασφαλιστικές Εταιρείες κτλ.

5.2 Dossia

Το “Dossia”[58] αποτελεί έναν βασικό ανταγωνιστή του “HealthVault”, αλλά αντίθετα με την προσέγγιση της Microsoft λειτουργεί με ανοιχτό λογισμικό και δίνουν πλήρη έλεγχο στο χρήστη πάνω στα δεδομένα του, τα οποία μπορεί να τα εξάγει. Η δημιουργία της πλατφόρμας αυτής είναι αποτέλεσμα της συνεργασίας διάφορων επιχειρηματικών κολοσσών των ΗΠΑ για την παροχή Ηλεκτρονικού Ιατρικού Φακέλου στους εργαζόμενους τους.

Το σύστημα δίνει τη δυνατότητα στους χρήστες να συγκεντρώσουν ψηφιακά αντίγραφα των ιατρικών δεδομένων τους και να δημιουργήσουν τα δικά τους φορητά Ιατρικά έγγραφα. Αρχικά τα δεδομένα θα προέρχονται από ιατρικές βάσεις δεδομένων από ινστιτούτα που κατέχουν τα στοιχεία των χρηστών, αλλά και από τα σχόλια και προσθήκες από τους ίδιους τους χρήστες. Το 2008 η “Walmart” ήταν η πρώτη από τους συνεταίρους η οποία προσφέρει την υπηρεσία σε όλους τους ασθενείς και στους εξαρτώμενους από αυτούς[24].

5.2 World Medical Card

Η “World Medical Card”[59] είναι ένα προϊόν της Νορβηγικής εταιρείας “World Medical Center”. Το “World Medical Card” παρέχει τρεις βασικές υπηρεσίες:

- Δικτυακό λογαριασμό το οποίο επιτρέπει στο χρήστη να προσθέσει και να διαχειριστεί προσωπικά δεδομένα υγείας(“onWeb”)[25]
- Μια εφαρμογή σε κινητά τηλέφωνα, η οποία βασίζεται στο πρωτόκολλο “WAP” και δίνει πρόσβαση στα ίδια δεδομένα(“onMobile”)[26]
- Μια σφραγισμένη κάρτα η οποία περιλαμβάνει μια συνοπτική αναφορά των πληροφοριών του κατόχου(“onCard”)[27]

Ο κύριος σκοπός του είναι να παρέχει την δυνατότητα στο χρήστη να έχει πρόσβαση και να διαχειρίζεται ουσιώδη προσωπικά στοιχεία, όπως:

- Διαγνώσεις
- Ιατρικό Ιστορικό
- Φαρμακευτικές Αγωγές
- Αλλεργίες and αντιδράσεις σε φάρμακα
- Εμβολιασμούς
- Επαφές για επείγουσες περιστάσεις
- Σημαντικά προσωπικά έγγραφα (με “ανέβασμα” αρχείου)

5.3 Avado

Το “Avado”[60] είναι μια διαδικτυακή ιστοσελίδα η οποία προσπαθεί να διευκολύνει και να εκσυγχρονίσει την σχέση ιατρού - ασθενή. Οι υπηρεσίες που παρέχει στους ασθενείς είναι η δυνατότητα Ηλεκτρονικού Ιατρικού Φακέλου με ιστορικό, αγωγές, εμβολιασμούς, αλλεργίες, προβλήματα υγείας αλλά και την επικοινωνία και τον διαμοιρασμό των αρχείων τους με παρόχους ιατρικών υπηρεσιών.

Για τους παρόχους προσφέρει δυνατότητα για την προώθησή τους, προγραμματισμό χρονοδρομολόγησης, καθώς επίσης και τη δυνατότητα να παρακολουθούν τους ασθενείς. Ο λόγος που αναφέρεται αυτή η ιστοσελίδα, παρά, το μικρό μέγεθός της είναι ότι κινείται στις ίδιες γραμμές με την λειτουργία της “Aegle”, για το οποίο αναπτύσσουμε την εφαρμογή μας.

Β' Μέρος - Ανάπτυξη εφαρμογής διαχείρισης και διαμοιρασμού δικαιωμάτων στη πλατφόρμα Android

1. Εισαγωγή

Η ύπαρξη ενός Ηλεκτρονικού Μητρώου Υγείας το οποίο θα δίνει στον χρήστη τον έλεγχο των δεδομένων του και ταυτόχρονα του παρέχει εύκολη πρόσβαση στις υπηρεσίες υγειονομικής περίθαλψης ήταν ένα από τα κύρια σημεία του Α' Μέρους. Σε αυτό το Μέρος θα εξεταστεί ένα κομμάτι μιας τέτοιας υποδομής, όπου ο χρήστης/ασθενής θα είναι υπεύθυνος για την διαχείριση των προσωπικών δεδομένων υγείας του. Συγκεκριμένα, θα έχει την δυνατότητα να διαχειρίζεται το Ιστορικό Υγείας του και να παρέχει άδειες προβολής αυτού, μέσω μίας εφαρμογής που εκτελείται στο λειτουργικό σύστημα Android· αυτό είναι ένα διαδεδομένο Λειτουργικό Σύστημα το οποίο εκτελείται σε κινητές συσκευές και υπολογιστές - ταμπλέτες. Η εφαρμογή ονομάζεται "Aegle"(από την Αίγλη, θεότητα υγείας και ευεξίας) και υποστηρίζει ως μοναδική οντότητα χρήστη τον "ασθενή". Στον ασθενή/χρήστη δίνεται η δυνατότητα να αλληλεπιδράσει με ιατρούς μέσω της τεχνολογίας NFC ή μέσω μιας ειδικής δραστηριότητας, με στόχο να δώσει δικαιώματα προβολής του ιστορικού του. Η "Aegle" δίνει στον χρήστη τον πλήρη έλεγχο στα δεδομένα του, σύμφωνα με τις οδηγίες της Ε.Ε. και με την λογική του ΗΜΥ που εξετάστηκε στο πρώτο μέρος. Ο σκοπός δημιουργίας μιας τέτοιας εφαρμογής είναι η εξομοίωση Ηλεκτρονικού Φακέλου Ασθενούς σε ένα Ηλεκτρονικό Μητρώο Υγείας, αλλά όχι η πλήρης υλοποίηση ενός τέτοιου προτυποποιημένου συστήματος, καθώς κάτι τέτοιο απαιτεί αρκετούς πόρους που τώρα έχουν αρχίσει να διαθέτονται για ανάπτυξη τέτοιων συστημάτων από τις ανεπτυγμένες χώρες.

Μερικές από τις ενέργειες που επιτρέπονται μέσω τις εφαρμογής είναι:

- Αναζήτηση, προσθήκη και διαγραφή Ιατρών από προσωπική λίστα χρήστη
- Προβολή Ιστορικού(εμβόλια, επεμβάσεις, αλλεργίες κτλ.)
- Διαγραφή αρχείων από το Ιστορικό
- Προβολή, διαγραφή παραχωρημένων δικαιωμάτων σε Ιατρούς
- Προσθήκη δικαιωμάτων σε Ιατρούς είτε σε επιλεγμένα αρχεία, είτε σε όλα, μόνιμα ή και προσωρινά μέσω NFC
- Εγγραφή NFC ετικέτας("tag") με τα βασικά μας στοιχεία
- Αλλαγή βασικών στοιχείων(όνομα, κωδικό, e-mail κτλ)

2. Η ιστοσελίδα “Aegle” και η σχέση με την εφαρμογή “Aegle”

Η ανάπτυξη της εφαρμογής “Aegle” βασίστηκε σε μια ομώνυμη διαδικτυακή ιστοσελίδα δικής μου ανάπτυξης, η οποία έχει λειτουργία ίδιας φύσης με την εφαρμογή. Η ιστοσελίδα υποστηρίζει δύο οντότητες χρηστών, του ιατρού και του ασθενή, με ξεχωριστές λειτουργίες διαθέσιμες στην κάθε μία. Είναι υλοποιημένη σε γλώσσες προγραμματισμού “PHP”, “javascript”, “jquery” και “HTML” και λειτουργεί σε έναν “Apache server”, μαζί με Σύστημα Διαχείρισης Βάσης Δεδομένων “MySQL”.

Ο χρήστης/ασθενής έχει στη διάθεσή του όλες τις λειτουργίες που υποστηρίζονται μέσω της εφαρμογής, αλλά μπορεί, επιπλέον, να προσθέσει εγγραφές στο Ιατρικό Ιστορικό του, να προσθέσει επαφές επικοινωνίας για επείγουσες περιστάσεις, να κάνει επιπλέον ρυθμίσεις όπως καθορισμός βάρους και ύψους και να προσθέσει εικόνες στο Ιστορικό του ή στο προφίλ του. Βέβαια, δεν υπάρχει η δυνατότητα λειτουργιών που προσφέρονται στην εφαρμογή λόγω της φύσης υλικού της κινητής συσκευής, όπως η χρήση NFC για παροχή δικαιωμάτων. Ο χρήστης/ιατρός μπορεί να δει μια λίστα με τους χρήστες/ασθενείς που του έχουν παράσχει δικαιώματα προβολής των ιστορικών τους, να δει τις εγγραφές που του έχουν δώσει εξουσιοδότηση και να αλλάξει βασικά στοιχεία του όπως όνομα, επίθετο, τομέα Ιατρικής που ασχολείται κτλ.

Η εφαρμογή δέχεται δεδομένα από την βάση δεδομένων “MySQL” της ιστοσελίδας και τα αποθηκεύει στην τοπική βάση δεδομένων, η οποία είναι μία ελαχιστοποιημένη εκδοχή της βάσης “MySQL” με στοιχεία μόνο για τις ανάγκες της εφαρμογής. Ο λόγος που υπάρχει βάση δεδομένων στην εφαρμογή είναι η εξοικονόμηση εύρους ζώνης, κρατώντας απλά ένα στιγμιότυπο των δεδομένων τοπικά στην κινητή συσκευή. Σε περίπτωση που προκύψει αλλαγή σε αυτά μέσω της εφαρμογής θα χρειαστεί η διεκπεραίωση επικοινωνίας μεταξύ τους για ενημέρωση της βάσης δεδομένων “MySQL”.

Για επικοινωνία της εφαρμογής με την βάση δεδομένων της ιστοσελίδας, με στόχο την ανταλλαγή δεδομένων, έχουν υλοποιηθεί αρχεία σε “PHP” τα οποία υλοποιούν σύνδεση στην βάση “MySQL” και κάνουν ερωτήματα σε αυτή. Στην συνέχεια, ανάλογα με τα ερωτήματα και τα αποτελέσματα, αποστέλλουν δεδομένα στην εφαρμογή.

Από εδώ και στο εξής θα αναφερόμαστε στην ιστοσελίδα με τον όρο “απομακρυσμένος διακομιστής”, κυρίως όπου προκύπτει επικοινωνία με την εφαρμογή. Επίσης, θα λαμβάνουμε την πρόσβαση στις λειτουργίες της ιστοσελίδας ως “κλειστή” πέραν των αρχείων “PHP” που μας παρέχονται. Τέλος, πρέπει να σημειωθεί ότι οτιδήποτε είναι σε διαμόρφωση κειμένου “*italics*”, σημαίνει ότι είναι κλάση “Java”, που είτε δημιουργήθηκε για την εφαρμογή ή ανήκει στην βιβλιοθήκη του Android.

3. Η πλατφόρμα Android

Από το 2007 και έπειτα έγινε μία έκρηξη ενδιαφέροντος στην κινητή υπολογιστική και σήμερα, πολλοί έχουμε στις τσέπες μας έναν μικρό υπολογιστή με δυνατότητες κινητής επικοινωνίας. Υπάρχουν τρεις κυρίαρχες πλατφόρμες στην αγορά έξυπνων τηλεφώνων: Android, iOS, Windows Phone. Η πλατφόρμα Android είναι η πιο εμπορική και συνεπώς πιο διαδεδομένη με 500 εκατομμύρια συσκευές και 1.3 εκατομμύρια προσθήκες καθημερινά[28][31] και εκεί θα αναπτύξουμε την εφαρμογή μας, όχι μόνο για το ευρύ καταναλωτικό κοινό, αλλά και διότι έχει μεγάλη κοινότητα ανοιχτού κώδικα και οι συσκευές με αυτό το λειτουργικό σύστημα φέρουν την τελευταία λέξη της τεχνολογίας.

3.1. Το Λειτουργικό Σύστημα Android

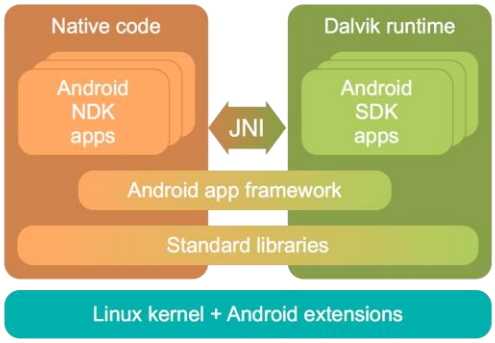
Εικόνα 7. Το λογότυπο και μασκότ του Android

Το Android είναι ένα Λειτουργικό Σύστημα βασισμένο στο “Linux” και έχει σχεδιαστεί ειδικά για κινητές συσκευές ή υπολογιστές ταμπλέτες με οθόνη αφής. Αρχικά αναπτυσσόταν από την Android Inc., την οποία υποστήριζε οικονομικά η Google και το 2005 την αγόρασε[29]. Το 2007 κυκλοφόρησε το λειτουργικό Android, ενώ ταυτόχρονα ιδρύθηκε η Open Handset Alliance· μια κοινοπραξία εταιρειών στους τομείς των υλικών, λογισμικού και τηλεπικοινωνιών, οι οποίες είναι αφοσιωμένες στην ανάπτυξη ανοιχτών προτύπων για κινητές συσκευές[30].

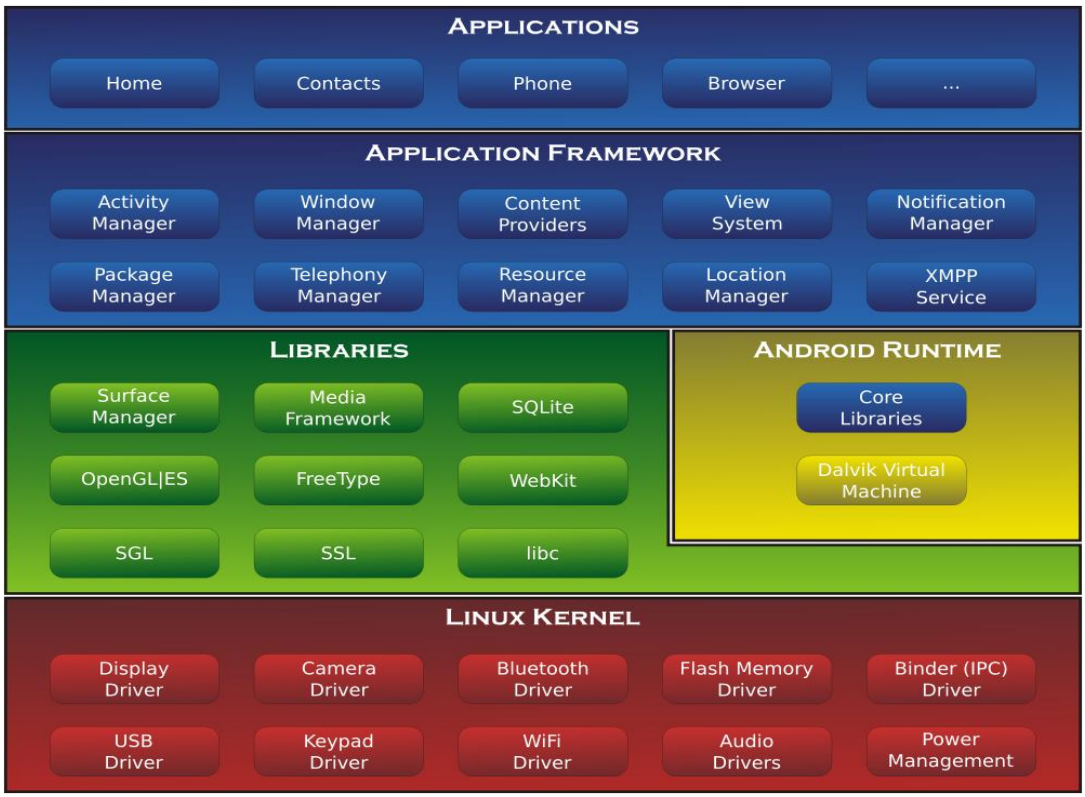
3.1.1. Ο πυρήνας του λειτουργικού και ανάπτυξη του

Το Android βασίζεται στον πυρήνα Linux 2.6 και από την έκδοση 4.0(“Ice Cream Sandwich”) και μετά στο πυρήνα 3.x, όπου περιλαμβάνονται βιβλιοθήκες, “middleware” και “API” που τρέχουν σε γλώσσα “C” και λογισμικό εφαρμογών συμβατό με βιβλιοθήκες Java. Το Android χρησιμοποιεί την εικονική μηχανή “Dalvik” με μεταγλώττιση “just-in-time” για να τρέξει το Dalvik 'dex-code' (Dalvik εκτελέσιμο), το οποίο μεταφράζεται από “bytecode Java”[32]. Η κύρια πλατφόρμα υλικού είναι η αρχιτεκτονική επεξεργαστών “ARM”, αλλά υπάρχει και υποστήριξη για x86 αρχιτεκτονική μέσω του προγράμματος “Android x86”[33].

Ο πυρήνας Linux του Android έχει υποστεί επιπλέον αλλαγές από τη Google, οι οποίες παρεκκλίνουν από το τυπικό κύκλο ανάπτυξης του πυρήνα Linux[34]. Επιπροσθέτως δεν περιλαμβάνεται εντόπιο "X Window System"(βασική διεπαφή χρήστη) και ούτε υποστηρίζει τις βιβλιοθήκες "GNU", κάτι που καθιστά δύσκολη την εισαγωγή εφαρμογών ή βιβλιοθηκών Linux στο Android[35]. Υποστήριξη για εφαρμογές με C είναι εφικτή με την εισαγωγή μιας μικρής βιβλιοθήκης και χρήση του "Java Native Interface(JNI)"[36]. Τέλος, παρέχεται η δυνατότητα ανάπτυξης εφαρμογών μέσω "Native Development Kit(NDK)" με "C" ή "C++" πέρα από το "Standard Development Kit(SDK)", όπου η ανάπτυξη γίνεται με "Java".



Εικόνα 8. Σχέση μεταξύ Dalvik Machine, Native Code και JNI.[32]



Εικόνα 9. Η αρχιτεκτονική του Android

Τον Αύγουστο του 2011 ο Λίνους Τόρβαλντς, δημιουργός του Linux και συντονιστής της ανάπτυξης του πυρήνα Linux, είπε: “τελικά το Android και το Linux θα συγκλίνουν σε ένα κοινό πυρήνα, αλλά δεν θα γίνει ακόμα παρά σε τέσσερα με πέντε χρόνια”[37]. Από το Δεκέμβρη του ίδιου έτους και οι δύο μεριές άλλαξαν την πορεία ανάπτυξης του λογισμικού τους, έτσι ώστε οι δύο πλατφόρμες να γίνουν συμβατές[38][39].

3.1.2 Εφαρμογές

Το Android είναι ένα σύστημα όπου τα προνόμια χρηστών είναι διαχωρισμένα και κάθε εφαρμογή τρέχει σε ένα σύστημα με σαφή διαχωρισμό μεταξύ αναγνωριστικών(ID χρήστη Linux και ID ομάδας). Έτσι κάθε εφαρμογή που εγκαθίσταται έχει το ρόλο ενός “χρήστη Linux” με ξεχωριστά επίπεδα πρόσβασης. Μέρη του συστήματος διαχωρίζονται σε ξεχωριστές οντότητες, οπότε οι εφαρμογές απομονώνονται μεταξύ τους, εκτός και αν το αναφέρουν ρητά ότι επιτρέπουν μέρη τους να χρησιμοποιηθούν. Στην εγκατάσταση μιας εφαρμογής το Android παρέχει ένα μοναδικό ID χρήστη, το οποίο παραμένει σταθερό για όλη τη διάρκεια ζωής στη συσκευή[40].

Βασικά στοιχεία ανάπτυξης εφαρμογής

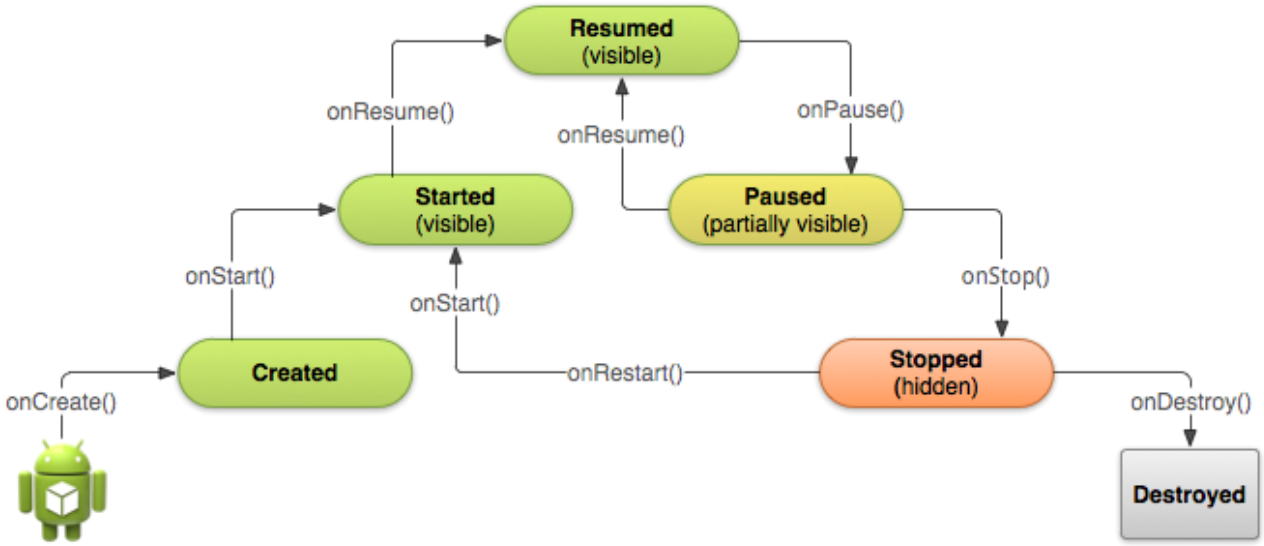
Κατά την ανάπτυξη μιας εφαρμογής ο προγραμματιστής ορίζει σε ένα ειδικό αρχείο(το μανιφέστο με μορφή XML) τις εκδόσεις του λειτουργικού που υποστηρίζονται, την έκδοση και το όνομα πακέτου της εφαρμογής, τις δραστηριότητες και τα “intents” της κάθε δραστηριότητας και τις άδειες χρήσης πόρων που απαιτεί η εφαρμογή από το λειτουργικό. Η εφαρμογή αποτελείται από διάφορες “δραστηριότητες”, οι οποίες ουσιαστικά είναι ενεργά παράθυρα(ή νήματα δράσεων) με ξεχωριστές λειτουργίες η κάθε μία. Κάθε δραστηριότητα αποτελείται από ένα “intent”(ή περισσότερα αν οριστούν από τον προγραμματιστή), το οποίο στην ουσία είναι ένα μέσο επικοινωνίας με άλλες δραστηριότητες της ιδίας εφαρμογής ή και άλλων εφαρμογών. Άλλα βασικά στοιχεία μιας εφαρμογής είναι οι εικόνες που θα περιλαμβάνονται στην εφαρμογή, τα αρχεία XML που καθορίζουν την εξωτερική εμφάνιση των δραστηριοτήτων ή και άλλων διάφορων στοιχείων(π.χ. ενός κουμπιού) και, φυσικά, οι κλάσεις “Java” που ορίζονται οι δραστηριότητες και άλλες πιο απλές ενέργειες.

Δραστηριότητες

Μία από τις βασικές προτεραιότητες του προγραμματιστή κατά την διαμόρφωση μιας δραστηριότητας είναι η διαχείριση του κύκλου ζωής της δραστηριότητας. Υπάρχει μια δεδομένη σειρά κλήσης μεθόδων, δίχως τις οποίες η δραστηριότητα δεν λειτουργεί και άλλες μέθοδοι που ενεργοποιούνται ανάλογα των ενεργειών του χρήστη που επηρεάζουν τον κύκλο ζωής. Αυτές οι μέθοδοι ονομάζονται “επιστροφής(μέθοδοι “callback”)”.

Οι κυριότερες μέθοδοι για την διαχείριση του κύκλου ζωής είναι:

- **onCreate**(): καλείται κατά την δημιουργία της δραστηριότητας
- **onStart**(): καλείται μετά την onCreate() και την onRestart()
- **onResume**(): καλείται μετά την onStart() και την onPause()
- **onPause**(): καλείται αν μπει άλλη δραστηριότητα στην αρχή της στοίβας δραστηριοτήτων(δηλαδή ορατή)
- **onStop**(): καλείται όταν ο χρήστης πατήσει το "Back" ή η διαχείριση μνήμης σταματήσει την δραστηριότητα
- **onRestart**(): καλείται μετά από την onStop(). Χρησιμοποιείται πιο σπάνια και αφήνεται στις αρχικές ρυθμίσεις
- **onDestroy**(): Τελική μέθοδος μιας δραστηριότητας και ενημερώνει το σύστημα ότι η δραστηριότητα δεν υπάρχει πλέον, οπότε απελευθερώνει όλους τους πόρους

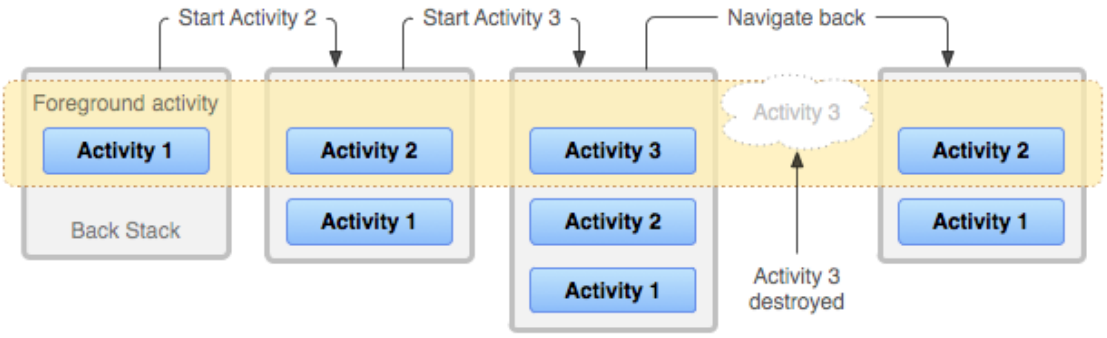


Εικόνα 10. Απλοποιημένη βηματική πυραμίδα του κύκλου ζωής μιας δραστηριότητας, μαζί με τα callbacks[41]

Στοίβα δραστηριοτήτων[42]

Μία τρέχουσα διεργασία είναι μια συλλογή δραστηριοτήτων με την οποία οι χρήστες αλληλεπιδρούν(μέσω των εφαρμογών) για συγκεκριμένες εργασίες. Οι δραστηριότητες αυτές οργανώνονται από το λειτουργικό σε μια στοίβα(“back stack“) με τη σειρά που δημιουργήθηκαν. Όταν ο χρήστης αγγίξει μια συντόμευση η διεργασία της επιλεγμένης εφαρμογής έρχεται στο προσκήνιο. Αν δεν υπάρχει ήδη διεργασία για αυτή την εφαρμογή, τότε δημιουργείται μία νέα και η “κύρια” δραστηριότητα της εφαρμογής γίνεται η ανώτερη δραστηριότητα στο “back stack”.

Όταν η τρέχουσα δραστηριότητα δημιουργεί ή καλεί μίαν άλλη δραστηριότητα, τότε η νέα δραστηριότητα προστίθεται(“push”) στη κορυφή της στοίβας και λαμβάνει τους πόρους της οθόνης. Η προηγούμενη δραστηριότητα παραμένει στη στοίβα, αλλά σταματάει(“onPause()”) και τοποθετείται στη δεύτερη θέση της στοίβας. Σε αυτή τη περίπτωση το λειτουργικό αποθηκεύει τη τρέχουσα κατάσταση της διεπαφής χρήστη. Όταν ο χρήστης πατήσει το κουμπί “Back” η τρέχουσα δραστηριότητα αφαιρείται(“pop”) από την κορυφή της στοίβας, καταστρέφεται(“onDestroy()”) και η προηγούμενη δραστηριότητα επανέρχεται στη κορυφή της στοίβας και επανεκκινείται(“onResume()”). Η στοίβα δρα ως LIFO(τελευταία μέσα, πρώτη έξω) στοίβα.



Εικόνα 11. Αναπαράσταση μιας στοίβας δραστηριοτήτων και αποτελεσμάτων ενεργειών χρήστη σε αυτή

Κατά την δημιουργία μιας δραστηριότητας ορίζεται η συσχέτισή της με τη τρέχουσα δραστηριότητα. Αυτό μπορεί να γίνει είτε από το μανιφέστο XML μέσα στο intent της δραστηριότητας ή απευθείας από το κώδικα που δημιουργεί το “intent” της δραστηριότητας. Οι πιο βασικοί τρόποι ορισμού των συσχετίσεων περιγράφονται στη συνέχεια.

- FLAG_ACTIVITY_NEW_TASK : Η δραστηριότητα δημιουργείται σε μια νέα διεργασία. Αν μια διεργασία υπάρχει ήδη τρέχει για αυτή την δραστηριότητα, τότε αυτή επανέρχεται στο προσκήνιο και η κατάσταση της επανέρχεται στο αποθηκευμένο σημείο. Λαμβάνει ένα νέο Intent.
- FLAG_ACTIVITY_SINGLE_TOP : Αν η δραστηριότητα προς εκκίνηση είναι η τρέχουσα δραστηριότητα, τότε το τρέχων στιγμιότυπο λαμβάνει ένα νέο Intent, αντί να δημιουργείται ένα νέο στιγμιότυπο της δραστηριότητας.
- FLAG_ACTIVITY_CLEAR_TOP : Αν η δραστηριότητα προς εκκίνηση ήδη τρέχει στην διεργασία(αν υπάρχει στη στοίβα), τότε αντί να δημιουργείται νέο στιγμιότυπο της δραστηριότητας, τότε όλες οι δραστηριότητες, που βρίσκονται πιο πάνω στη στοίβα, καταστρέφονται και το “intent” δίνεται στο ανανεωμένο στιγμιότυπο της δραστηριότητας.

Το FLAG_ACTIVITY_CLEAR_TOP χρησιμοποιείται πιο συχνά μαζί με το FLAG_ACTIVITY_NEW_TASK. Έτσι μπορεί να εντοπιστεί μια υπάρχουσα δραστηριότητα και να την τεθεί σε θέση που να μπορεί να αντιδράσει στο intent.

3.1.3. Dalvik Virtual Machine

Το “Dalvik” είναι το “Virtual Machine(VM)” λειτουργεί στο Android και είναι λογισμικό ανοιχτού κώδικα. Ο ρόλος του είναι να “εκτελεί” τις εφαρμογές στις συσκευές, κάτι που το καθιστά αναπόσπαστο κομμάτι του λειτουργικού. Τα προγράμματα γράφονται σε Java και μετατρέπονται σε “bytecode”, στην συνέχεια σε κλάσεις “JVM (Java Virtual Machine)” και τέλος σε εκτελέσιμα αρχεία “.dex(Dalvik Executable)”.

Αντίθετα με τα “Java VM”, τα οποία είναι “μηχανές στοίβας”, το “Dalvik VM” χρησιμοποιεί αρχιτεκτονική βασισμένη σε καταχωρητές. Ένα ειδικό εργαλείο, το “dx”, χρησιμοποιείται για να μετατρέψει μερικές(αλλά όχι όλες) κλάσεις (“.class”) σε “.dex”. Πολλαπλές κλάσεις περιλαμβάνονται σε ένα αρχείο και τα διπλότυπα συμβολοσειρών και μεταβλητών μπορούν να υπάρχουν μία φορά στο αρχείο. Επίσης, το “bytecode” μετατρέπεται σε εναλλακτικό σετ εντολών. Συνήθως ένα αρχείο “.dex” είναι ελάχιστα μικρότερο σε μέγεθος από ένα “.jar (Java Archive)”[43]. Τα εκτελέσιμα μπορεί να υποστούν ξανά αλλαγές κατά την εγκατάστασή τους για την βελτιστοποίηση της απόδοσή τους. Από την έκδοση 2.2 του Android περιλαμβάνεται μεταγλωττιστής “just-in-time”[44].

3.2. Τεχνολογία NFC

Το “NFC (Near Field Communication)” είναι ένα σύνολο προτύπων για συσκευές με τη δυνατότητα ραδιοεπικοινωνίας(π.χ. ένα έξυπνο τηλέφωνο) για την εγκαθίδρυση συνεδρίας επικοινωνίας μέσω αφής ή προσέγγισης δύο τέτοιων συσκευών. Πρόσφατες εφαρμογές αυτής της τεχνολογίας είναι ανταλλαγή δεδομένων, πληρωμές και απλοποίηση περίπλοκων διαδικασιών, όπως αλλαγή προφίλ ήχου συσκευής, “bluetooth pairing” και σύνδεση σε δίκτυα “Wi-Fi”. Τα πρότυπα καλύπτουν πρωτόκολλα επικοινωνίας και μορφές ανταλλαγής δεδομένων τα οποία βασίζονται σε προϋπάρχοντα πρότυπα “RFID”, όπως το “ISO/IEC 14443” και το “FeliCa”[45]. Επίσης, περιλαμβάνεται το “ISO/IEC 18092”[46] και άλλα που ορίστηκαν από το “NFC Forum”, το οποίο ιδρύθηκε από τις “Nokia, Sony και Philips” και σήμερα περιλαμβάνει πάνω από 160 μέλη.

Υπάρχουν πέντε βασικοί τύποι ετικετών(“tag”) για τη συγκεκριμένη τεχνολογία, με τις οποίες μπορεί να επικοινωνήσει μια συσκευή με δυνατότητες NFC.

Τύπος	Προϊόντα	Χωρητικότητα
Type1	Innovision Topaz	96 bytes
Type2	NXP MIFARE Ultralight (C)	48 – 144 bytes
Type3	Sony Felica	1, 4, 9 Kbytes
Type4	NXP DESFire	2, 4, 8 Kbytes
Mifare Classic	NXP MIFARE Classic	1, 4 Kbytes

Πίνακας 4. Οι διάφοροι τύποι και τα χαρακτηριστικά ετικετών τεχνολογίας NFC

Τέλος, υποστηρίζονται δύο μέθοδοι επικοινωνίας(παθητικό, ενεργητικό) και διάφορες κωδικοποιήσεις[62]

Ταχύτητα	Ενεργή συσκευή	Παθητική συσκευή
424 kbit/s	Manchester, 10% ASK	Manchester, 10% ASK
212 kbit/s	Manchester, 10% ASK	Manchester, 10% ASK
106 kbit/s	Modified Miller, 100% ASK	Manchester, 10% ASK

Πίνακας 5. Ρυθμοί μετάδοσης και κωδικοποιήσεις σε συνεδρία μεταξύ δύο NFC συσκευών

4. Η εφαρμογή “Aegle”

Στόχος της διπλωματικής είναι η ανάπτυξη εφαρμογής στη πλατφόρμα Android για τον διαμοιρασμό και την διαχείριση δικαιωμάτων προβολής του Ιατρικού Ιστορικού του χρήστη. Σε αυτό το κεφάλαιο θα δούμε τις προκλήσεις και την υλοποίηση μιας τέτοιας εφαρμογής. Πέραν της δυνατότητας να δίνει δικαιώματα προβολής του Ιστορικού του, ο χρήστης θα έχει την δυνατότητα να διαχειρίζεται τα προσωπικά στοιχεία του και τον Ηλεκτρονικό Ιατρικό Φάκελο του με εγγραφές που περιλαμβάνουν αλλεργίες, προβλήματα υγείας, ιατρικές εξετάσεις, φαρμακευτικές αγωγές και εμβολιασμούς.

Σε σχέση με την ιστοσελίδα “Aegle”, η πρόσβαση και διαχείριση των δεδομένων χρήστη είναι πιο περιορισμένη(π.χ. δεν υπάρχει η δυνατότητα προσθήκης εγγραφών), αλλά παρέχονται επιπλέον δυνατότητες χάριν στην φύση της πλατφόρμας υλικού. Συγκεκριμένα, δίνεται στο χρήστη η ικανότητα εγγραφής ετικέτας NFC με τα βασικά του στοιχεία(ονοματεπώνυμο, αναγνωριστικό χρήστη και ομάδα αίματος), για πρόσβαση από άλλη ειδική εφαρμογή για ειδικές περιπτώσεις(π.χ. σε ένα ατύχημα). Επιπλέον, ο χρήστης μπορεί με το άγγιγμα της συσκευής του σε μια προπρογραμματισμένη ετικέτα NFC να δώσει προσωρινά πλήρη δικαιώματα σε έναν ιατρό που επισκέπτεται, καθιστώντας την συναλλαγή με τον ιατρό εύκολη και παρέχοντας σε αυτόν άμεση πρόσβαση στο Ιστορικό του. Έτσι πέραν της εύκολης πρόσβασης στα ιατρικά αρχεία, βελτιστοποιείται και η αλληλεπίδραση μεταξύ ασθενή και ιατρού, αλλά και αυξάνονται οι πιθανότητες σωστής διάγνωσης αφού υπάρχει εύκολη πρόσβαση στο ιατρικό ιστορικό ασθενούς.

4.1. Βάση δεδομένων

Για την αποθήκευση των δεδομένων που λαμβάνουμε από τον απομακρυσμένο διακομιστή χρησιμοποιείται το ελαφρύ σχεσιακό Σύστημα Διαχείρισης Βάσης Δεδομένων "SQLite". Το σύστημα αυτό εμπεριέχεται σε μια μικρή(περί των 350 KB)[47] βιβλιοθήκη γραμμένη σε γλώσσα "C". Σε αντίθεση με άλλες σχεσιακές βάσεις δεδομένων(ΒΔ) η "SQLite" δεν είναι μια ξεχωριστή διεργασία προσβάσιμη από μία εφαρμογή "πελάτη", αλλά ένα αναπόσπαστο κομμάτι του συστήματος διαχείρισης Βάσης Δεδομένων. Είναι η πιο διαδεδομένη μηχανή ΒΔ, αφού χρησιμοποιείται σε φυλλομετρητές, λειτουργικά συστήματα και ενσωματωμένα συστήματα[48] τα οποία απαιτούν ή έχουν ελάχιστους πόρους για την λειτουργία τους.

4.1.1 Κλάσεις διαχείρισης Βάσης Δεδομένων

Για την παραμετροποιημένη διαχείριση της Βάσης Δεδομένων της εφαρμογής δημιουργήθηκε μια κλάση με την ονομασία "*Database*". Έχει οριστεί ως υποκλάση της "*SQLiteOpenHelper*", η οποία αυτοματοποιεί διαδικασίες όπως δημιουργία, διαχείριση και διαγραφή (CRUD) μιας βάσης δεδομένων. Έτσι οι βοηθητικές μέθοδοι της "*SQLiteOpenHelper"* είναι διαθέσιμες για χρήση(εφόσον είναι υπερκλάση) και ταυτόχρονα δημιουργήθηκαν επιπλέον μέθοδοι στην κλάση "*Database"* για παραμετροποιημένη διαχείριση της βάσης δεδομένων, όπως διαγραφή και ρητή προσθήκη εγγραφών.

Για την διενέργεια ερωτημάτων στη βάση έχουμε μια βοηθητική κλάση, η οποία επιστρέφει αντικείμενα της μορφής "*Cursor*". Τα αντικείμενα "*Cursor*" στην ουσία είναι δείκτες("pointers") στα δεδομένα της βάσης που επιστρέφονται με το ερώτημα και έτσι επιτυγχάνεται βέλτιστη απόδοση και εξοικονόμηση πόρων, αφού δεν περιλαμβάνονται δεδομένα αλλά αναφορές σε αυτά. Η χρήση της "*Cursor*" είναι ο πιο διαδεδομένος τρόπος διενέργειας ερωτημάτων σε βάσεις δεδομένων στο Android, κυρίως λόγο απόδοσης και αυτός είναι ο λόγος που χρησιμοποιείται και στην "Aegle". Πιο συγκεκριμένα η παρακάτω μέθοδος δημιουργήθηκε για γρήγορα ερωτήματα στην βάση δεδομένων, όπου απλά καλείται η "fetchData" με το ερώτημα που χρειάζεται και επιστρέφεται ένας *"Cursor"*.

```
public Cursor fetchData(String query) throws SQLException {
        Cursor aCursor = DB.rawQuery(query, null);
        if (aCursor != null) {
            aCursor.moveToFirst();
        }
        return aCursor;
}
```

Στην προσαρμοσμένη κλάση *"Database"* περιλαμβάνονται διάφορες μέθοδοι για την προσθήκη και διαγραφή εγγραφών από τους πίνακες. Μετά από το τερματισμό κάθε συνεδρίας τα δεδομένα δεν παραμένουν στην συσκευή πέραν ενός αναγνωριστικού χρήστη και ένα αναγνωριστικό συνεδρίας. Αυτή η διαγραφή των εγγραφών γίνεται λόγω ασφαλείας και κυρίως λόγω του ευαίσθητου χαρακτήρα των δεδομένων. Η λογική είναι ότι τα δεδομένα θα υπάρχουν στην συσκευή μόνο όσο χρησιμοποιεί ο χρήστης την εφαρμογή και έτσι δεν παραμένουν τα ευαίσθητα δεδομένα τα οποία μπορούν να υποκλαπούν από μία κακόβουλη εφαρμογή. Το αναγνωριστικό χρήστη και συνεδρίας χρησιμοποιούνται έτσι ώστε να ταυτοποιείται ο χρήστης χωρίς να εισάγει το αναγνωριστικό του κάθε φορά που ανοίγει την εφαρμογή.

4.1.2 Κριτήρια σχεδίασης της Βάσης Δεδομένων και δομή της

Για την σχεδίαση της βάσης λήφθηκε υπόψιν κυρίως η εμπειρία του χρήστη και το γεγονός ότι η εφαρμογή αποτελεί εξομοίωση Ηλεκτρονικού Ιατρικού Φακέλου Ασθενούς και όχι πλήρης υλοποίηση ενός. Κύριο κριτήριο ήταν ότι τα δεδομένα πρέπει να είναι κατανοητά από τον χρήστη και να έχουν ελάχιστο αντίκτυπο στην εφαρμογή, καθώς η βάση "SQLite" προσφέρεται για ελαφριές διεργασίες και δεν αποτελεί ολοκληρωμένο Σύστημα Διαχείρισης Βάσεων Δεδομένων. Επίσης, η εφαρμογή έχει μόνο μία οντότητα χρήστη –αυτή του ασθενή. Με γνώμονα τα προηγούμενα, η βάση δεδομένων πρέπει να αποθηκεύει βασικά στοιχεία που θα είναι κατανοητά και χρήσιμα στον απλό χρήστη και θα καταλαμβάνουν ελάχιστο χώρο και εύρος ζώνης. Για αυτό το σκοπό επιλέχθηκε η αποθήκευση δεδομένων σε μορφή αναγνώσιμου κειμένου χωρίς περιττές(για τον χρήστη/ασθενή) πληροφορίες, όπως αυτές που σχετίζονται με πρότυπα για ΗΙΦΑ.

Το περιβάλλον προγραμματισμού Android περιέχει μεθόδους για κάθε λειτουργία που μπορεί να χρειάζεται μια βάση δεδομένων μέσω της κλάσης "*SQLiteDatabase"*. Για παράδειγμα για διαγραφή όλων των γραμμών ενός πίνακα απλά καλείται η μέθοδος "delete", για εκτέλεση οποιουδήποτε ερωτήματος υπάρχει η "rawQuery" και για περιπτώσεις όπου δεν επιλέγονται δεδομένα η "execSQL", η οποία αυξάνει την απόδοση στις περιπτώσεις που χρησιμοποιείται .

Για κάθε ερώτημα που χρειάζεται εγγραφή δεδομένων πρέπει να γίνεται σύνδεση στην βάση με λειτουργία εγγραφής.

```
SQLiteDatabase db = this.getWritableDatabase();
```

Ένα παράδειγμα δημιουργίας πίνακα:

```
String sav="CREATE TABLE save (uid INTEGER, username TEXT, sessid VARCHAR PRIMARY KEY,
                                                    time_set LONGINT )";
db.execSQL(sav);
```

Παρακάτω είναι οι πίνακες και τα πεδία τους που αποτελούν την Βάση Δεδομένων της εφαρμογής “Aegle”(με μορφοποίηση “bold” τα κύρια κλειδιά).

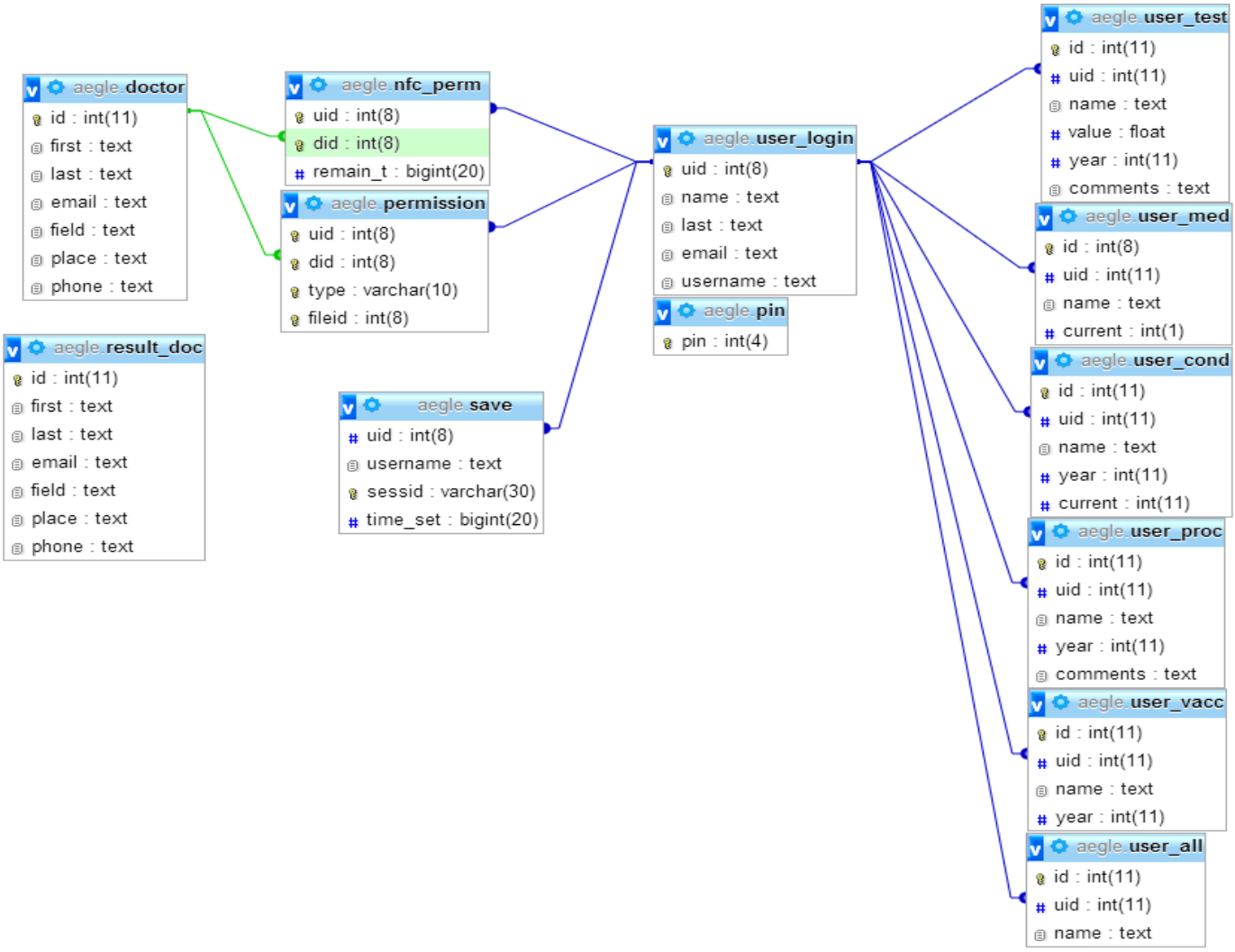
Όνομα Πίνακα	**Πεδίο 1**	**Πεδίο 2**	**Πεδίο 3**	**Πεδίο 4**	**Πεδίο 5**	**Πεδίο 6**	**Πεδίο 7**
user_login	**uid**	name	last	email	username	-	-
user_med	**id**	uid	name	current	-	-	-
user_cond	**id**	uid	name	year	current	-	-
user_test	**id**	uid	name	value	year	comments	
user_all	**id**	uid	name	-	-	-	-
user_proc	**id**	uid	name	year	comments	-	-
user_vacc	**id**	uid	name	year	-	-	-
doctor	**id**	first	last	email	field	place	phone
result_doc	**id**	first	last	email	field	place	phone
permission	**uid**	**did**	**type**	**fileid**	-	-	-
save	uid	username	**sessid**	time_set	-	-	-
nfc_perm	**uid**	**did**	remain_t	-	-	-	-
pin	**pin**	-	-	-	-	-	-

Πίνακας 6. Οι πίνακες της Βάσης Δεδομένων της εφαρμογής “Aegle” και τα πεδία τους

Με τις αρχικές ρυθμίσεις της “SQLite” τα ξένα κλειδιά και οι ρητές σχέσεις RDBMS μεταξύ ξένων και κυρίων κλειδιών δεν υποστηρίζονται [63]. Για την ενεργοποίηση της υποστήριξης ξένων κλειδιών πρέπει, κάθε φορά που γίνεται σύνδεση στη βάση, να ενεργοποιείται η εντολή: `PRAGMA foreign_keys = ON`

Στην “Aegle” επιλέχθηκε να μην οριστούν ρητά οι σχέσεις μεταξύ κλειδιών, αλλά να βασίζονται στις ενέργειες που τις συνδέουν μέσω μεθόδων της “*Database*”. Ο λόγος για αυτό είναι ότι υπάρχει ένας χρήστης κάθε φορά στη βάση και η μόνη περίπτωση που διαγράφεται είναι όταν αδειάζει η βάση. Επίσης, δεν υπάρχει η δυνατότητα αλλαγής του κυρίου κλειδιού στο πίνακα χρήση(user_login). Με βάση τα παραπάνω, αν οριζόνταν οι σχέσεις “RDBMS” θα ήταν όλες σε “CASCADE”.

Τέλος, πρέπει να σημειωθεί ότι στην “SQLite” τα αριθμητικά μεγέθη που ορίζουν το μήκος ενός κελιού παραβλέπονται(π.χ. “INT(8)”), διότι δεν επιβάλλονται περιορισμοί μήκους[61]. Ακολουθεί σχηματική αναπαράσταση της βάσης με τις συσχετίσεις μεταξύ πινάκων και το τύπο των πεδίων. Για τους λόγους που αναφέρθηκαν, οι σχέσεις και οι τύποι που φαίνονται παρακάτω δεν είναι ρητά ορισμένες, αλλά έτσι θα ήταν υλοποιημένες σε “MySQL”.



Ο πρώτος πίνακας(user_login) περιέχει τα βασικά στοιχεία του χρήστη, όπως αναγνωριστικό χρήστη(uid), όνομα(name), επίθετο(last), ηλεκτρονικό ταχυδρομείο(email) και όνομα χρήστη(username), απαραίτητα για την προσωποποίηση του χρήστη.

Οι επόμενοι έξι πίνακες ορίζουν το ιατρικό ιστορικό του χρήστη και φέρουν, αντίστοιχα, τα στοιχεία για τις φαρμακευτικές αγωγές, ιατρικές καταστάσεις, εξετάσεις, αλλεργίες, επεμβάσεις και εμβολιασμούς του χρήστη. Μοιράζονται αρκετά πεδία μεταξύ τους και περιέχουν πληροφορίες όπως αναγνωριστικό (id), αναγνωριστικό χρήστη(uid), όνομα(name), έτος(year), τρέχων κατάσταση(current), τιμή(value) και σχόλια(comments).

Οι επόμενοι δύο είναι πανομοιότυποι για λόγους ευκολίας στην αντιγραφή των αποτελεσμάτων αναζήτησης στο πίνακα των ιατρών, εάν ο χρήστης αποφασίσει να προσθέσει έναν ιατρό που αναζήτησε. Ο πίνακας του ιατρού(doctor) περιέχει τους ιατρούς που έχει προσθέσει ο χρήστης στη λίστα του. Ο πίνακας των αποτελεσμάτων αναζήτησης ιατρών(result_doc) αποθηκεύει τους ιατρούς που έχει αναζητήσει ο χρήστης. Αν ο χρήστης προσθέσει επαφή μέσω αναζήτησης τότε αυτή περνάει στον κανονικό πίνακα άμεσα. Τα πεδία των πινάκων είναι αναγνωριστικό ιατρού(id), όνομα(first), επίθετο(last), διεύθυνση ηλεκτρονικού ταχυδρομείου(email), ειδικότητα(field), μέρος εργασίας(place) και τηλέφωνο(phone).

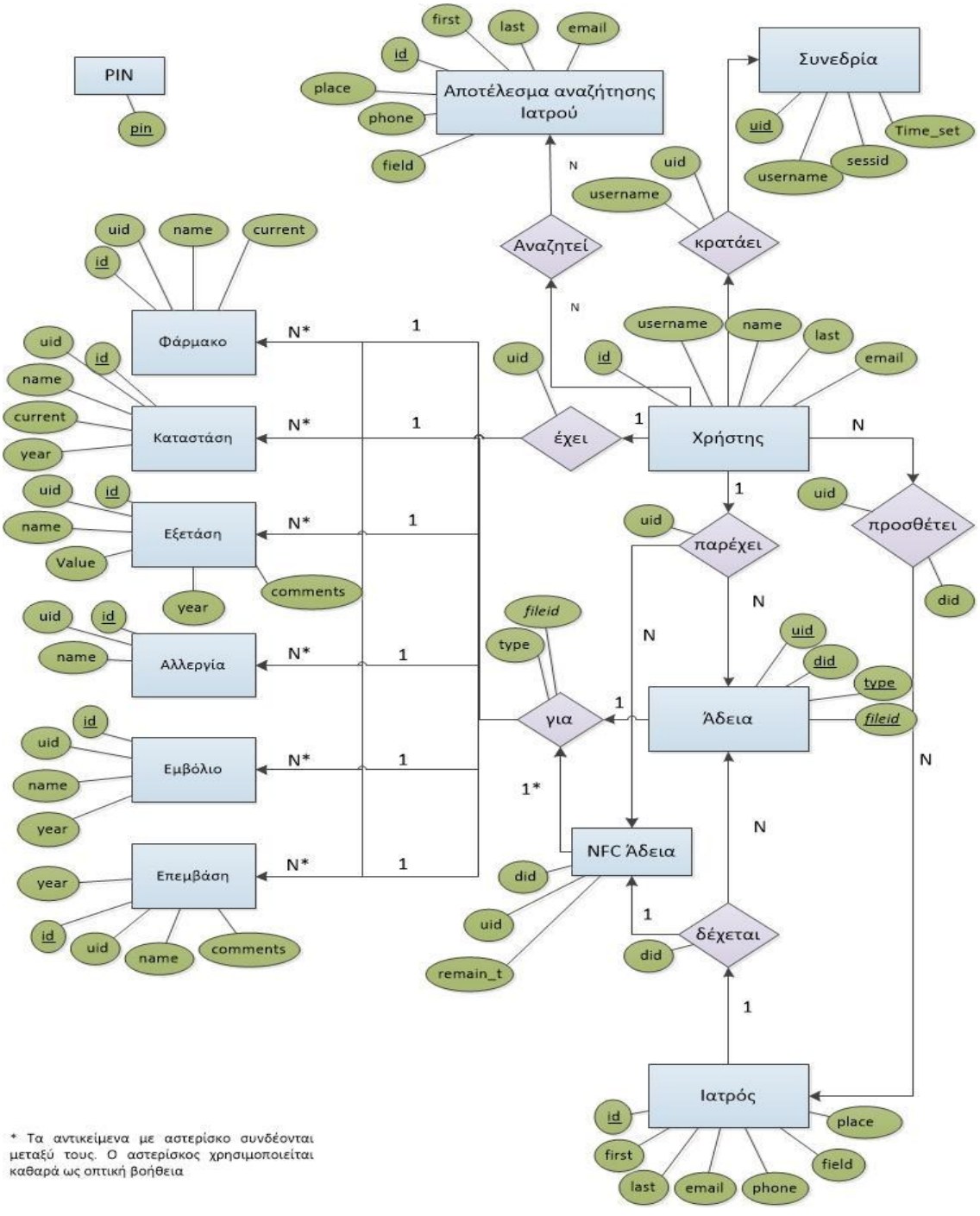
Ο πίνακας για τις άδειες(permission) που παρέχει ο χρήστης στους ιατρούς του έχει τέσσερα πεδία και όλα μαζί αποτελούν το κύριο κλειδί, καθώς ο συνδυασμός τους δημιουργεί τη μοναδική εγγραφή που χρειάζεται. Τα πεδία αυτά είναι αναγνωριστικό χρήστη(uid), αναγνωριστικό ιατρού(did), τύπος αρχείου(type) και το αναγνωριστικό αρχείου(fileid). Ο τύπος αρχείου είναι το πεδίο που καθορίζει σε ποια κατηγορία(από αυτές του Ιατρικού Φακέλου) ανήκει το αναγνωριστικό αρχείου. Έτσι έχουμε έναν πίνακα για όλες τις άδειες.

Ο πίνακας που παραμένει μετά το πέρας κάθε συνεδρίας είναι ο πίνακας αποθήκευσης(save). Τα στοιχεία που παραμένουν ώστε να ταυτοποιηθεί ο χρήστης στην επόμενη είσοδό του στην εφαρμογή είναι το αναγνωριστικό του(uid), το όνομα χρήστη(username), το αναγνωριστικό συνεδρίας(sessid) και ο χρόνος που δημιουργήθηκε η συνεδρία(time_set). Ο χρόνος δημιουργίας καθορίζει τον ορισμό του χρόνου ζωής της συνεδρίας, ο οποίος αν περάσει τότε η συνεδρία “λήγει” και ζητούνται ξανά τα στοιχεία πρόσβασης από τον χρήστη. Αυτό επιλέχθηκε ως μία επιπλέον δικλείδα ασφαλείας και με την λογική ότι ο χρήστης δεν πρόκειται να μπαίνει στην εφαρμογή συχνά, καθώς αυτή δεν έχει το χαρακτηριστικό της έντονης συμμετοχής χρήστη(“user engagement”). Συνήθως, ο χρήστης θα χρησιμοποιεί την εφαρμογή λίγες ώρες –όποτε χρειάζεται τις υπηρεσίες– και μετά την επόμενη φορά που θα χρειαστεί.

Ο προτελευταίος πίνακας(nfc_perm) αποθηκεύει τις προσωρινές άδειες που παρέχονται στους ιατρούς μέσω NFC. Τα πεδία του πίνακα είναι το αναγνωριστικό χρήστη(uid), το αναγνωριστικό του ιατρού(did) και ο χρόνος που παραμένει μέχρι να διαγραφεί η εγγραφή(remain_t). Αυτό γίνεται με την λογική ότι ο χρήστης σε μια επίσκεψη σε ένα ιατρικό κέντρο θα θέλει να δείξει το ιατρικό ιστορικό του στον γιατρό που τον εξετάζει, αλλά όχι απαραίτητα να παρέχει μόνιμη πρόσβαση. Έτσι με μία κίνηση του χεριού του έχει την δυνατότητα να παρέχει δικαιώματα για την προβολή του φακέλου του και κατόπιν δύο ωρών το δικαίωμα πρόσβασης αφαιρείται. Ο χρόνος των δύο ωρών θεωρείται αρκετός, καθώς μια τυπική επίσκεψη είναι 13 με 16 λεπτά.[49]

Ο τελευταίος πίνακας έχει μόνο ένα ομώνυμο πεδίο· τον κωδικό PIN του χρήστη, ο οποίος χρησιμοποιείται κατά την παραχώρηση των προσωρινών αδειών με NFC. Όταν ο χρήστης πατήσει το κουμπί να δώσει πρόσβαση του ζητάει το κωδικό PIN, που έχει ορίσει ο ίδιος. Αυτό γίνεται επειδή η τεχνολογία NFC είναι χωρίς αφή και μπορεί κάποιος κακόβουλος να πλησιάσει στην τσέπη μας ή στη συσκευή μας μια “πειραγμένη” καρτέλα NFC έτσι ώστε να αποκτήσει δικαιώματα προβολής στο φάκελό μας. Διαφορετικά μπορεί κάποιος, πάλι κακόβουλος, να μας πάρει τη συσκευή για τον ίδιο σκοπό, αλλά λόγω αυτού του επιπλέον μέτρου δεν μπορεί να δώσει άδεια γρήγορα και με αφανή τρόπο.

Παρακάτω είναι το διάγραμμα του μοντέλου Οντοτήτων - Συσχετίσεων της βάσης δεδομένων της εφαρμογής “Aegle”.



4.2. Επικοινωνία εφαρμογής με απομακρυσμένο διακομιστή

Υλοποίηση τοπικά (στην εφαρμογή)

Για την λήψη και αποστολή δεδομένων από και στον απομακρυσμένο διακομιστή απαιτείται μια μορφή επικοινωνίας μαζί του. Επίσης, χρειάζεται μια υπηρεσία "πελάτη" η οποία θα δέχεται και θα κάνει ερωτήματα σε αυτόν τον διακομιστή. Τέλος, πρέπει όλα αυτά τα ερωτήματα να γίνονται ασύγχρονα, διότι η εκτέλεση κώδικα είναι γραμμική και η επικοινωνία μέσω διαδικτύου υπόκειται σε καθυστερήσεις, κάτι που μπορεί να προκαλέσει είτε ανεπιθύμητη μείωση απόδοσης ή να κάνει την εφαρμογή να τερματίσει.

Για την επικοινωνία με διαδικτυακές υπηρεσίες το Android περιέχει την κλάση "*HttpClient*", η οποία είναι ένα σύνολο μεθόδων για την υλοποίηση "HTTP πελατών" και περιέχει ευρεία γκάμα αντικειμένων και μεθόδων για την διαχείριση "cookies", τη διαχείριση σύνδεσης, την διαχείριση συνεδρία καθώς και άλλα χρήσιμα εργαλεία. Η ασφάλεια αυτών των μεθόδων βασίζεται στο τρόπο υλοποίησης της υπηρεσίας του "πελάτη" από τον προγραμματιστή, αν δηλαδή επιθυμεί μα υλοποιηθεί με αυξημένη ασφάλεια ή με τις αρχικές ρυθμίσεις. Στην εφαρμογή ορίζεται μία κλάση "*CustomHttpClient*", όπου δημιουργείται το στιγμιότυπο του "HTTP πελάτη" και περιλαμβάνεται μια μέθοδος η οποία διαχειρίζεται την εκτέλεση "POST" ερωτημάτων στον απομακρυσμένο διακομιστή καθώς και τις απαντήσεις από αυτόν και άλλη μία μέθοδος που παραμετροποιεί τις ρυθμίσεις της σύνδεσης.

Για την εξασφάλιση της απρόσκοπτης λειτουργίας της εφαρμογής κατά την επικοινωνία με το διακομιστή ορίστηκε μία επιπλέον βοηθητική κλάση, η οποία υλοποιεί την "*CustomHttpClient*" με ασύγχρονο τρόπο. Πιο συγκεκριμένα δημιουργήθηκε η κλάση "*CustomHttpTask*" που ορίζεται ως υποκλάση της "*AsyncTask*" και έτσι κληρονομεί τις μεθόδους και τις ιδιότητές της για χρήση. Η κλάση "*AsyncTask*" επιτρέπει την υλοποίηση εργασιών παρασκηνίου χωρίς να διακόπτει το νήμα διεπαφής χρήστη, ο οποίος είναι και ο λόγος που χρησιμοποιείται στην εφαρμογή. Όταν ολοκληρώνεται η επιθυμητή εργασία τότε παραδίδει το αποτέλεσμα στο τρέχων νήμα. Χρησιμοποιείται για λειτουργίες που απαιτούν μερικά δευτερόλεπτα και όχι για μεγαλύτερες εργασίες(για αυτές υπάρχουν οι "FutureTask" και "ThreadPoolExecutor'), λόγω περιορισμούς που αφορούν την νηματοποίηση.

Η κλάση αυτή ορίζεται με τρεις τύπους δεδομένων:

- **Params**: Οι παράμετροι που χρειαζόμαστε για την εκτέλεση της ασύγχρονης εργασίας
- **Progress**: Η πρόοδος ολοκλήρωσης της εργασίας
- **Result**: Το αποτέλεσμα της εργασίας προς παράδοση στο τρέχων νήμα

Επίσης, έχει τέσσερα βήματα(μεθόδους) για την ολοκλήρωσή της:

- **onPreExecute:** Προετοιμασία δεδομένων για την εκτέλεση της εργασίας
- **doInBackground:** Επεξεργασία στο παρασκήνιο
- **onProgressUpdate:** Λήψη ενεργειών κατά την πρόοδο της εκτέλεσης
- **onPostExecute:** Διαχείριση αποτελεσμάτων

Για την υλοποίηση ενός ερωτήματος απλά πρέπει να δημιουργηθεί ένα αντικείμενο "*CustomHttpTask*". Με αυτό το τρόπο έχουμε δημιουργήσει έναν "HTTP πελάτη", ο οποίος επικοινωνεί ασύγχρονα με τον διακομιστή μας. Παράδειγμα:

```
ArrayList<NameValuePair> postParameters = new ArrayList<NameValuePair>();
postParameters.add(new BasicNameValuePair("user_id",user_id));
CustomHttpTask task = new CustomHttpTask();
response = task.execute(server.server+"<insert url here>", postParameters).get();
```

Αρχικά εισάγονται οι παράμετροι προς αποστολή(στην μεταβλητή "postParameters") και στην συνέχεια δημιουργείται ένα αντικείμενο της κλάσης "CustomHttpTask" που υλοποιεί τον "HTTP πελάτη". Τέλος, καλείται η μέθοδος "execute" η οποία ανήκει την κλάση "AsyncTask". Αυτή εκκινεί το νήμα στο παρασκήνιο και παραδίδει το αποτέλεσμα(αν ζητηθεί) με την μέθοδο "get". Η μέθοδος αυτή επιστρέφει ένα αντικείμενο "Object", το οποίο θα μετατραπεί στο τύπο που ορίστηκε (ο τύπος του "Result") και – στη περίπτωση αυτή– θα το παραδώσει στην μεταβλητή "response".

Η μορφή των δεδομένων που επιστρέφονται από τον διακομιστή είναι "JSON (Javascript Object Notation)". Αυτή είναι αναγνώσιμη μορφή και προέρχεται από την γλώσσα "Javascript" για την παρουσίαση δομών δεδομένων και συσχετισμένων συστοιχιών ως αντικείμενα. Χρησιμοποιείται πολύ συχνά για την γραμμικοποίηση(μετατροπή σχεσιακών δεδομένων σε αντικείμενα) και τη μετάδοση δεδομένων μέσω δικτυακής σύνδεσης και κυρίως για την επικοινωνία μεταξύ διακομιστή και διαδικτυακής εφαρμογής, ως εναλλακτικής του "XML". Ένα αντικείμενο "JSON" μπορεί να περιέχει και άλλα αντικείμενα της ίδιας μορφοποίησης, αποκτώντας έτσι την δομή πίνακα "JSON". Το Android, ως παραλλαγή της "Java", μπορεί να χειριστεί εύκολα αυτή τη μορφή αντικειμένων με τις κλάσεις "*JSONObject*" και "*JSONArray*", για αυτό και χρησιμοποιείται ως τρόπος επικοινωνίας με τον διακομιστή. Η μεταβλητή "response" που είδαμε πριν είναι αντικείμενο της κλάσης "*JSONObject*" και με τη μέθοδο "get" αποθηκεύει την απάντηση του διακομιστή.

Υλοποίηση απομακρυσμένα(στον διακομιστή)

Όπως αναφέρθηκε, ο διακομιστής είναι ένας “Apache Server” με μια βάση “MySQL”, ο οποίος δέχεται δεδομένα από τα στιγμιότυπα “πελάτη” – δηλαδή την εφαρμογή – και αναλόγως ποιο αρχείο ζητείται και τα δεδομένα που δέχεται δίνει την αντίστοιχη απάντηση. Η διαχείριση ερωτημάτων και απαντήσεων γίνεται με την γλώσσα “PHP: **P**HP **H**ypertext **P**reprocessor”.

Η επιλογή του προγράμματος διακομιστή έγινε με κριτήριο την εύκολη υλοποίηση και παραμετροποίηση που παρέχει ο “Apache”. Ομοίως, το πρόγραμμα Διαχείρισης Βάσεων Δεδομένων “MySQL” συνεργάζεται πολύ καλά με τον “Apache Server”, είναι εύκολο στην υλοποίησή του και λειτουργεί με την γλώσσα προγραμματισμού βάσεων δεδομένων “SQL”, η οποία είναι πολύ διαδεδομένη. Επιπλέον, και τα δύο προγράμματα είναι δωρεάν και ανοιχτού κώδικα. Τέλος, η επιλογή της γλώσσας “PHP” βασίστηκε στο γεγονός ότι είναι μία εύχρηστη και διαδεδομένη γλώσσα διαδικτυακού προγραμματισμού η οποία μπορεί να χειριστεί εύκολα εξωτερικές επικοινωνίες, συνεδρίες και ερωτήματα σε οποιοδήποτε σύστημα Διαχείρισης Βάσεων Δεδομένων.

Κάθε αρχείο “.php” δέχεται τα δεδομένα “POST” από την εφαρμογή και με βάση τους αλγορίθμους που περιλαμβάνει θέτει ερωτήματα στη Βάση Δεδομένων και απαντά ανάλογα. Τα δεδομένα που επιστρέφονται από τα ερωτήματα στην βάση δεδομένων κωδικοποιούνται σε μορφή “JSON” με την μέθοδο “json_encode”(κωδικοποιεί πίνακες σε μορφή “JSON”) και αποστέλλονται – με την εντολή “echo” – στο “πελάτη HTTP” που περιμένει απάντηση”. Σε όλα τα αρχεία περιλαμβάνεται στο “JSON” αντικείμενο μία μεταβλητή(“success”) με τιμή 0 ή 1, ανάλογα με το αν η επιθυμητή ενέργεια εκτελέστηκε σωστά ή όχι. Με αυτό τον τρόπο εξασφαλίζεται η εγκυρότητα της επικοινωνίας. Την μεταβλητή αυτήν λαμβάνει η εφαρμογή και αν και μόνο αν είναι ‘1’ τότε συνεχίζει τις κατάλληλες ενέργειες, άρα εξασφαλίζεται ότι δεν θα εισάγει λανθασμένα δεδομένα ή άδειες εγγραφές.

Ένα τυπικό παράδειγμα κώδικα:

```
if(isset($_POST['something'])){
     $userid=intval($_POST['user_id']);
     $fileid=intval($_POST['file_id']);
     $something=clear($_POST['something']);
     $db=open();
     $q=$db->prepare('DELETE FROM <table1> WHERE uid=? AND id=?');
     $q1=$db->prepare('DELETE  FROM  <table>  WHERE  uid=?  AND  fileid=?  AND
     smt=?');
     $q->execute(array($userid,$fileid));
     $q1->execute(array($userid,$fileid,$something));
     $response['success']=1;
     $db=null;
     echo json_encode($response);}
```

Για ασφάλεια κατά την διεξαγωγή ερωτημάτων χρησιμοποιούνται διάφορες γνωστές τεχνικές και τεχνολογίες, μερικές από τις οποίες μπορούν να βρεθούν στο παραπάνω κομμάτι κώδικα. Συγκεκριμένα, χρησιμοποιούμε την τεχνική εκκαθάρισης δεδομένων για αποφυγή ανεπιθύμητων δεδομένων. Δηλαδή, όπου χρειαζόμαστε ακέραιο αριθμό χρησιμοποιούμε την συνάρτηση “int()” της PHP, η οποία μετατρέπει τα δεδομένα που εισάγονται σε ακέραιο. Επίσης, χρησιμοποιείται η συνάρτηση “trim()”, η οποία αποκόπτει χαρακτήρες κενού, νέας γραμμής κτλ. και με αυτόν τον τρόπο αλλοιώνονται οι κακόβουλες συμβολοσειρές και εξασφαλίζεται η εισαγωγή μόνο των επιθυμητών συμβολοσειρών. Επιπροσθέτως, για εξασφάλιση ότι το ερώτημα έρχεται από έγκυρη πηγή στην αρχή κάθε αρχείου ελέγχεται αν το αναγνωριστικό συνεδρίας είναι έγκυρο και αν προέρχεται από πραγματικό χρήστη. Τέλος, δημιουργήθηκε η συνάρτηση “clear”, η οποία υλοποιεί το “trim” που αναφέρθηκε και ταυτόχρονα απαγορεύει την χρήση κάποιων χαρακτήρων όπως “#”, “\”, “_”, κτλ. αντικαθιστώντας τους με τον κενό χαρακτήρα.

Η συνάρτηση “clear” φαίνεται παρακάτω. Σε αυτήν ορίζεται ένας πίνακας(“$escape”) με τους χαρακτήρες που θέλουμε να αφαιρέσουμε από την συμβολοσειρά που μας δίνεται(“$str”) και στην συνέχεια, με παράμετρο αυτό το πίνακα, αντικαθιστώνται οι ανεπιθύμητες συμβολοσειρές με τον κενό χαρακτήρα. Επίσης, χρησιμοποιείται η “trim” για να αποφευχθούν χαρακτήρες κενού, νέας γραμμής κτλ. Τέλος, επιστρέφεται η επιθυμητά μορφοποιημένη συμβολοσειρά.

```
function clear($str){
      $escape=array("#","\\","_","SELECT ","/");
      $ret=str_replace($escape,"",trim($str));
return $ret;
}
```

Η τεχνολογία που χρησιμοποιείται για την διεξαγωγή ερωτημάτων στην βάση δεδομένων “MySQL” του απομακρυσμένου διακομιστή ονομάζεται “PDO(PHP Data Objects) MySQL”. Η “PDO” είναι μια δομημένη και ελαφριά επέκταση της PHP για πρόσβαση σε βάσεις δεδομένων, το οποίο παρέχει ένα αφηρημένο στρώμα πρόσβασης σε δεδομένα(“data-access abstraction layer”). Αυτό σημαίνει ότι ανεξάρτητα με το τύπο της βάσης που χρησιμοποιείται(MySQL, SQL, PostgreSQL, FoxPro κτλ) με την “PDO” καλούνται οι ίδιες μέθοδοι για συγκεκριμένες ενέργειες[55]. Ένα από τα πολλαπλά πλεονεκτήματα της χρήσης “PDO” είναι ότι όταν καλούνται οι μέθοδοι “prepare()” και στην συνέχεια “execute()”(η μεθοδολογία αυτή ονομάζεται προετοιμασμένες δηλώσεις), υπάρχει η δυνατότητα να εκτελεστεί το ίδιο ερώτημα πολλές φορές χωρίς να χρειαστεί να ξανά οριστεί η μορφή του ερωτήματος, παρά να περαστούν ως παράμετροι διαφορετικά δεδομένα. Έτσι αυξάνεται η απόδοση, επιτρέποντας την συνεννόηση μεταξύ πελάτη(οντότητα που κάνει ερωτήματα) και διακομιστή(το ΣΔΒΔ) για την χρήση πόρων για διενέργεια πολλαπλών ερωτημάτων στην βάση. Τέλος, η χρήση “PDO” έχει θετικό αποτέλεσμα στην ασφάλεια αφού είναι εύκολο να αποφευχθούν επιθέσεις έγχυσης δεδομένων(“SQL Injection”) πρώτου επιπέδου. Αυτό είναι δυνατόν διότι δεν υπάρχει η ανάγκη χρήσης εισαγωγικών, αλλά και γιατί η μέθοδος “prepare” εκτελεί πολλαπλές φορές δοκιμαστικά ερωτήματα, δηλαδή κάνει εξομοίωση του ερωτήματος χωρίς δεδομένα, για να ελεγχθεί αν ένα ερώτημα κάνει μη επιθυμητές ενέργειες (π.χ. σε εντολή “SELECT” να γίνεται διαγραφή εγγραφών)[56].

4.3 Συνεδρία χρήστη

Κάθε φορά που η διαχείριση μνήμης καταστρέφει την κύρια δραστηριότητα της εφαρμογής ή ο χρήστης τερματίσει ηθελημένα την εφαρμογή, τότε τα δεδομένα μέσα στην βάση διαγράφονται για λόγους ασφαλείας. Για να μην εισάγει ο χρήστης συνεχώς το αναγνωριστικό του, υπάρχει μια συνεδρία χρήστη, η οποία παραμένει κατά την διαγραφή των δεδομένων από την βάση δεδομένων της εφαρμογής. Επίσης, χρησιμοποιείται σε κάθε ερώτημα στο διακομιστή για προστιθέμενη ασφάλεια, διότι έχει την μορφή ενός επιπλέον –μη τυπικού– αναγνωριστικού χρήστη, το οποίο είναι μοναδικό μια δεδομένη στιγμή και αλλάζει συχνά(σε αντίθεση με το αναγνωριστικό χρήστη)· οπότε γίνεται δύσκολο να υποκλαπεί.

Ο αλγόριθμος που ελέγχει ο διακομιστής αν η συνεδρία είναι έγκυρη φαίνεται παρακάτω. Ουσιαστικά ελέγχεται αν υπάρχει μία και μοναδική συνεδρία που αντιστοιχεί στον χρήστη που μας έστειλε τα δεδομένα.

```
if(isset($_POST['session_id'])){
        $sessid=clear($_POST['session_id']);
        $uid=intval($_POST['uid']);
        $db=open();
        $check = $db->prepare('SELECT * FROM us_session WHERE uid=? and hex=?');
        $check->setFetchMode(PDO::FETCH_ASSOC);        $check-
        >execute(array($uid,$sessid));
        $count=count($check->fetchAll(PDO::FETCH_ASSOC));
        if($count==1){.....//do stuff
```

Ένας διαφορετικός τρόπος που μπορούσε να υλοποιηθεί μια συνεδρία χρήστη είναι μέσω του “*DefaultHttpClient*”. Αυτή η υλοποίηση θα απαιτούσε την ύπαρξη συνεδρίας μόνο τοπικά της συσκευής και την διατήρηση του αντικειμένου σε όλες τις δραστηριότητες(μπορεί να μεταφερθεί μέσω των “intent”). Όμως, κάτι τέτοιο θα έκανε δύσκολη την συντήρηση της εφαρμογής και η συνεδρία θα παρέμενε τοπικά στην εφαρμογή και όχι και στον διακομιστή περιπλέκοντας την επικοινωνία μεταξύ τους. Αυτοί είναι οι βασικοί λόγοι που οδήγησαν στην δημιουργία συνεδρίας εκ των χειρών.

Ο τρόπος υλοποίησης στην εφαρμογή λίγο διαφορετικός και πιο απλοποιημένος σε σχέση με την αυτοματοποιημένη δημιουργία συνεδρίας σε διαδικτυακό “πελάτη”. Η διαδικασία εγκαθίδρυσης μίας συνεδρίας αρχίζει την στιγμή που ο χρήστης εισάγει σωστά τα στοιχεία ταυτοποίησης χρήστη(όνομα χρήστη και κωδικό) στην εφαρμογή και αυτό επιβεβαιωθεί από τον διακομιστή. Όταν ταυτοποιηθεί ο χρήστης από τον διακομιστή τότε αυτός δημιουργεί ένα τυχαίο και μοναδικό “αναγνωριστικό συνεδρίας” και το αποθηκεύει στην “MySQL” βάση δεδομένων, το οποίο αποστέλλεται στην εφαρμογή. Η εφαρμογή σημειώνει το χρόνο που λήφθηκε η απάντηση και αποθηκεύει το αναγνωριστικό χρήστη, το όνομα χρήστη, το αναγνωριστικό συνεδρίας και το χρόνο που λήφθηκε.

Ο τρόπος δημιουργίας του τυχαίου αναγνωριστικού συνεδρίας με “PHP” είναι:

```
$rand=sha1(microtime(true).mt_rand(10000,90000))
```

Αυτή η μέθοδος δημιουργεί μια μοναδική συνεδρία, καθώς παράγει έναν ψευδο-τυχαίο αριθμό από 10000 έως 90000 και τον ενώνει με το αριθμό του μικροδεπτερόλεπτου της μεταβλητής χρόνου του “Unix”. Οι πιθανότητες να δημιουργηθεί ίδια συνεδρία είναι “απείρως” ελάχιστες. Πιο συγκεκριμένα, για 80,000 διαφορετικούς συνδυασμούς η πιθανότητα να δημιουργηθεί ίδιος τυχαίος είναι 50.17% αν το ίδιο μικροδεπτερόλεπτο ζητηθούν 335 συνεδρίες. Αυτό σημαίνει πως αν 20,100,000,000 άτομα χρησιμοποιούσαν την υπηρεσία κάθε λεπτό τότε θα υπήρχε πιθανότητα μία στις δύο να έχουν το ίδιο αναγνωριστικό. Ακόμη και αν ληφθεί υπόψιν ότι η μηχανή παραγωγής αριθμών δεν είναι ιδανική και η δοσοληψία της μεταβλητής χρόνου είναι ατελής η πιθανότητα ύπαρξης δύο ίδιων αναγνωριστικών είναι απόμακρη.

$$p(\tilde{n}) = \frac{n!\binom{80000}{n}}{80000^n}$$

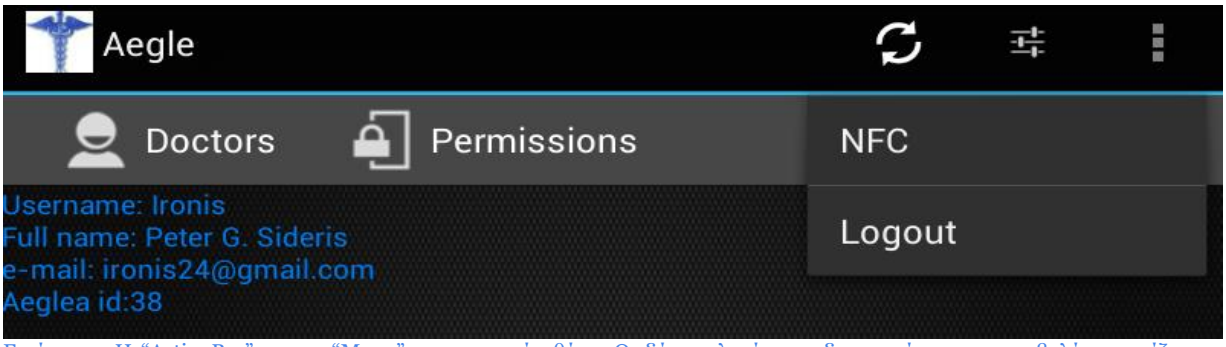
Εξίσωση 1. Η εξίσωση υπολογισμού της πιθανότητας για παραγωγή ίδιου αριθμού σε σύνολο 80,000 διαφορετικών συνδυασμών. Για τιμή n=335 μας δίνει τιμή 0.501759806222713

Κάθε φορά που εκκινείται η εφαρμογή η δραστηριότητα ταυτοποίησης (“*Login”)* ελέγχει αν η συνεδρία είναι κάτω των δύο ημερών. Αν όχι τότε ζητάει όνομα χρήστη και κωδικό, διαφορετικά στέλνει το αναγνωριστικό συνεδρίας στον απομακρυσμένο διακομιστή, όπου και ελέγχεται αν συμφωνεί η συνεδρία της συσκευής με την αποθηκευμένη στην “MySQL” βάση δεδομένων. Στη περίπτωση που συμφωνούν γίνεται αποστολή όλων των δεδομένων χρήστη και η εφαρμογή συνεχίζει τις ενέργειές της.

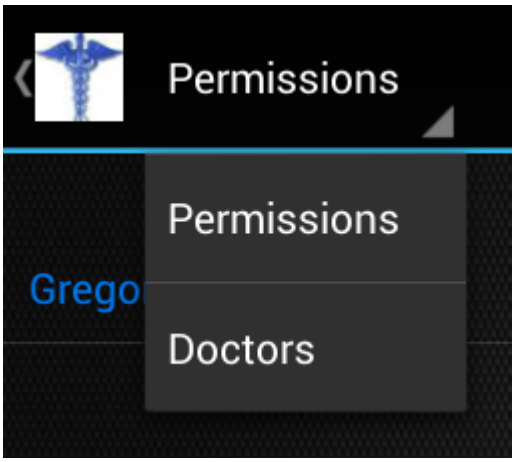
Για να καθοριστεί μια συνεδρία “μη ενεργή” υπάρχουν τρεις τρόποι. Ο πρώτος είναι ο χρήστης να επιλέξει να τερματίσει την συνεδρία του χειροκίνητα, μέσω της επιλογής “Logout” η οποία διαγράφει την συνεδρία, τα δεδομένα και τερματίζει την εφαρμογή. Ο δεύτερος είναι αυτή να λήξει μετά το πέρας δύο ημερών οπότε και θα ζητηθεί ανανέωσή της. Τέλος, ο τρίτος και πιο έμμεσος τρόπος είναι η περίπτωση ο χρήστης να εισάγει τα αναγνωριστικά ταυτοποίησης του από άλλη συσκευή έτσι δημιουργώντας νέα συνεδρία.

4.4. Πλοήγηση και Στοίβα Δραστηριοτήτων

Η πλοήγηση στην εφαρμογή περιέχει τυπικά στοιχεία μιας εφαρμογής έκδοσης Android 4.0 και άνω και σύμφωνα με της συστάσεις της Google για το σχεδιασμό πλοήγησης εφαρμογών. Μερικές από τις συστηνόμενες εκγλείσεις είναι η χρήση του “Up button”, του “Spinner” σε μενού επιλογών και της “ActionBar”. Τέλος, πολλές ενέργειες (όπως διαγραφή, προσθήκη κτλ.) γίνονται με κλικ παρατεταμένης διάρκειας(“longclick”) και αυτή είναι η πιο μη ρητή λειτουργία της εφαρμογής, κάτι που μπορεί να το κάνει δύσκολο για τον νέο χρήστη έξυπνης συσκευής, αλλά βοηθάει στην περάτωση γρήγορων ενεργειών.

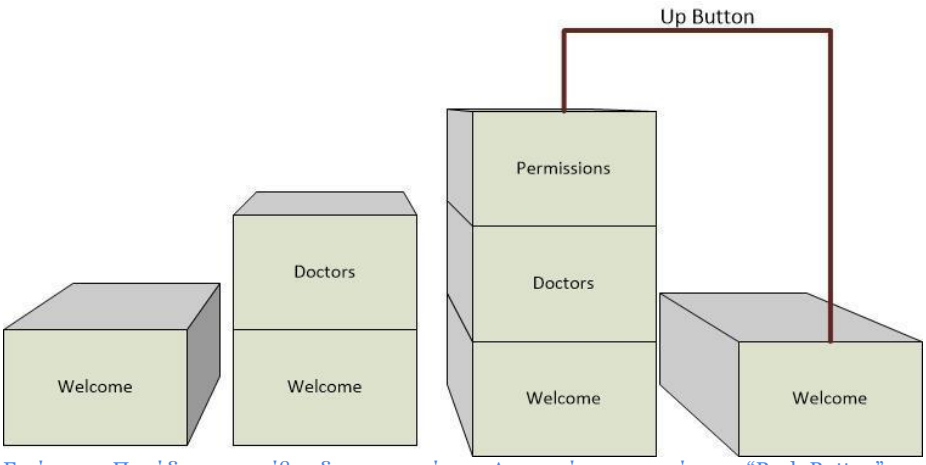


Εικόνα 12. Η “ActionBar” και το “Menu” στην αρχική οθόνη. Οι δύο επιλογές που δεν χωράνε στην προβολή εμφανίζονται ως πτυσσόμενο μενού με τρεις τελείες(αναφορά στο “varargs”).



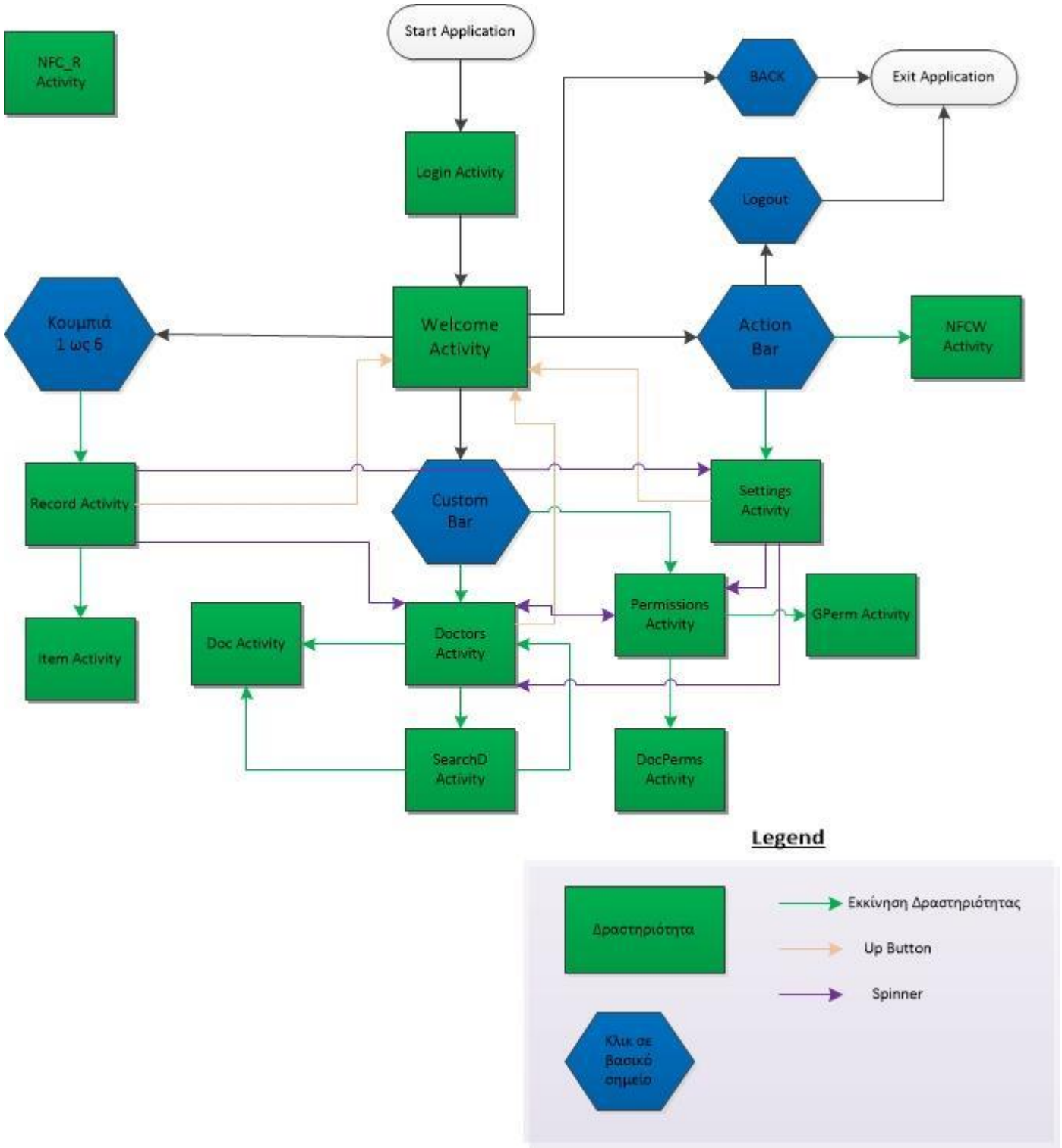
Εικόνα 13. Το “Up Button”(βελάκι αριστερά) και το μενού “Spinner”. Το “Up Button” προγραμματίστηκε έτσι ώστε να εμφανίζεται η αρχική οθόνη και να καθαρίζεται η Στοίβα Δραστηριοτήτων.

Η στοίβα δραστηριοτήτων έχει οριστεί έτσι ώστε να η αρχική δραστηριότητα(“*Welcome*”) να βρίσκεται πάντα στο κατώτερο σημείο της στοίβας. Ο λόγος για αυτό είναι ότι δεν θέλουμε το FLAG_ACTIVITY_CLEAR_TOP να την καταστρέψει, καθώς θα αδειάσει η βάση δεδομένων και η εφαρμογή θα πάψει να λειτουργεί αφού η κάθε δραστηριότητα καλεί δεδομένα από την βάση. Ταυτόχρονα για τον ίδιο λόγο δεν πρέπει να καταστραφεί και από την διαχείριση μνήμης οπότε η στοίβα δραστηριοτήτων της εφαρμογής διατηρείται συνεχώς αρκετά μικρή για αυτό χρησιμοποιείται συχνά το FLAG_ACTIVITY_CLEAR_TOP . Μπορούμε να θεωρήσουμε την αρχική δραστηριότητα ως την ρίζα ενός δέντρου με όλες τις πιθανές ενέργειες στην εφαρμογή.



Εικόνα 14. Παράδειγμα στοίβας δραστηριοτήτων. Αν ο χρήστης πατούσε το “Back Button” αντί του “Up”, τότε η δραστηριότητα “*Doctors”* θα βρισκόταν στη κορυφή, άρα θα ήταν στο προσκήνιο της οθόνης.

Το παρακάτω σχήμα είναι ένα διάγραμμα ροής για τις δραστηριότητες της εφαρμογής, μέσω του οποίου φαίνεται πως μπορεί να εκκινήσει μια δραστηριότητα ξεκινώντας από την αρχική δραστηριότητα. Συμπεριλαμβάνονται τα αποτελέσματα ενεργειών με την χρήση του “Up Button” και του “Spinner” με άλλο χρώμα για εύκολο διαχωρισμό.

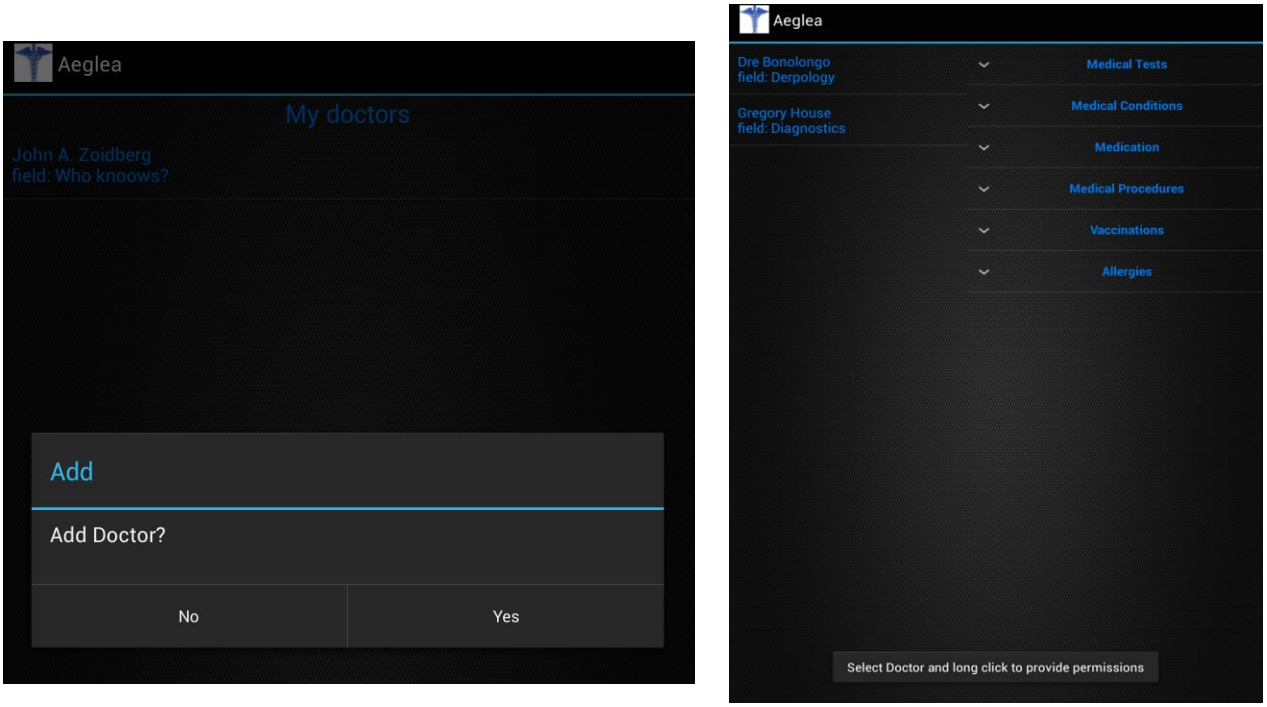


Εικόνα 15.Διάγραμμα Ροής Δραστηριοτήτων. Περιλαμβάνονται τα αποτελέσματα των δράσεων μέσω Spinner και Up Button

4.5. Εκχώρηση αδειών προβολής

Από τις πιο βασικές λειτουργίες της εφαρμογής είναι η παροχή δικαιωμάτων προβολής καταχωρίσεων του Ιατρικού Φακέλου του ασθενή σε ιατρούς. Αυτό γίνεται με δύο τρόπους και οι άδειες μπορούν να είναι μόνιμες ή προσωρινές, ανάλογα με την μέθοδο που θα ακολουθηθεί.

Ο πρώτος τρόπος είναι να διαλέξει ο ασθενής του ιατρούς από μια προσωπική λίστα και να τους παρέχει πρόσβαση στα αρχεία που θα επιλέξει. Την λίστα αυτή την δημιουργεί ο ίδιος ο ασθενής αναζητώντας και προσθέτοντας τους ιατρούς που επιθυμεί. Εάν θέλει να παρέχει δικαιώματα προβολής πρέπει να επιλέξει τον ιατρό και τις εγγραφές που θέλει να φαίνονται σε αυτούς. Για να δουν τις εγγραφές οι ιατροί πρέπει να συνδεθούν μέσω της ιστοσελίδας.



Εικόνα 16.Η οθόνη αναζήτησης και προσθήκης Ιατρού στην προσωπική λίστα του χρήστη(αριστερα) και η οθόνη παροχής δικαιωμάτων(δεξιά). Μας παροτρύνει να επιλέξουμε ιατρό και να κάνουμε LongClick όπου θέλουμε να δώσουμε δικαιώματα

Ο δεύτερος τρόπος είναι να διαβαστεί, από την συσκευή του ασθενή, μια NFC καρτέλα που είναι γραμμένη με δεδομένα ιατρού και “ειδικό mimetype”. Έτσι, μπορεί να δώσει προσωρινά άδειες προβολής σε όλο το ιστορικό του στον ιατρό που ορίζουν τα δεδομένα της καρτέλας(βλ. εικόνες 19 και 20). Μετά από δύο ώρες τα δικαιώματα προβολής διαγράφονται, καθιστόντας τα δικαιώματα προσωρινά.

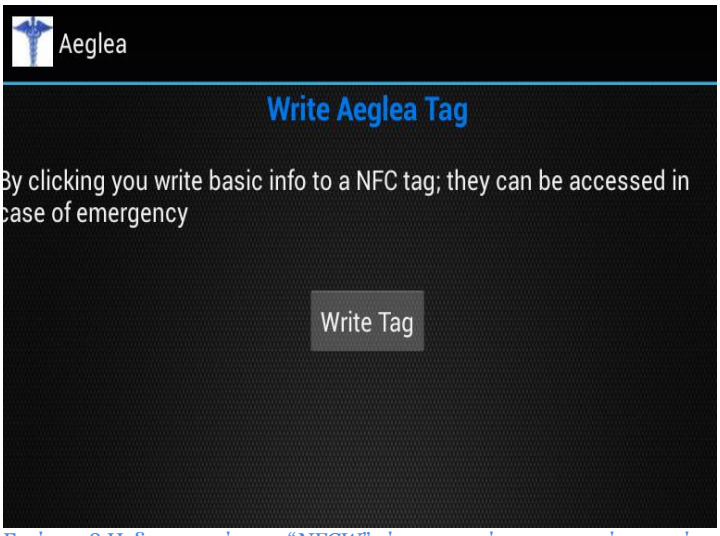
4.6. NFC

Υπάρχουν δύο δραστηριότητες που χειρίζονται τις ενέργειες με NFC για τις δύο περιπτώσεις που αξιοποιείται αυτή η τεχνολογία. Η μέθοδος αλληλεπίδρασης μιας συσκευής με μια καρτέλα NFC είναι πολύ απλή και γίνεται με την προσέγγιση της καρτέλας τεχνολογίας NFC στον αισθητήρα NFC της συσκευής.

Εικόνα 17.Οι δύο τρόποι αλληλεπίδρασης με ετικέτα NFC της εφαρμογής. Ο πρώτος τρόπος είναι η δραστηριότητα "NFCW" και ο δεύτερος η δραστηριότητα "NFC_R"

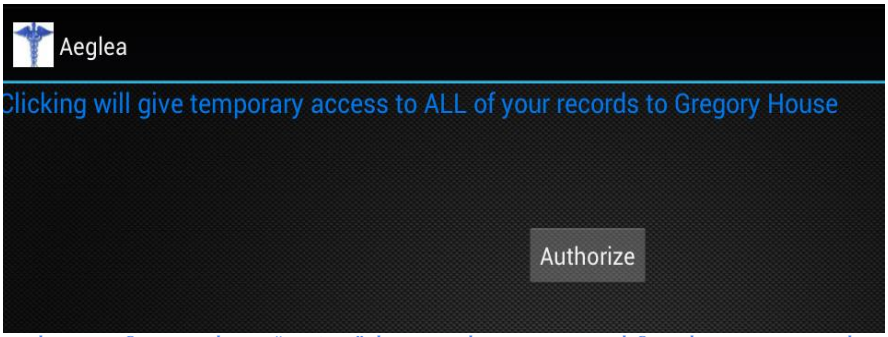
Η πρώτη δραστηριότητα που αξιοποιεί το NFC – η "*NFCW*" – είναι προσβάσιμη μέσω συντόμευσης που βρίσκεται στην "ActionBar" και ο χρήστης έχει την δυνατότητα να:

- Διαβάσει και να εμφανίσει κάποια από τα δεδομένα χρήστη γραμμένα σε καρτέλα NFC με το "mimetype": "application/com.pgsideris.aeglea"
- Γράψει μια καρτέλα NFC με κάποια δεδομένα του, με το προηγούμενο"mimetype"
- Δώσει πρόσβαση προβολής βασικών πληροφοριών και του Ιστορικού του σε "ειδική" εφαρμογή, σε περίπτωση που η πρόσβαση είναι απαραίτητη και ο χρήστης δεν είναι ικανός να την παρέχει. Αυτό είναι και ο τελικός σκοπός αυτής της δραστηριότητας

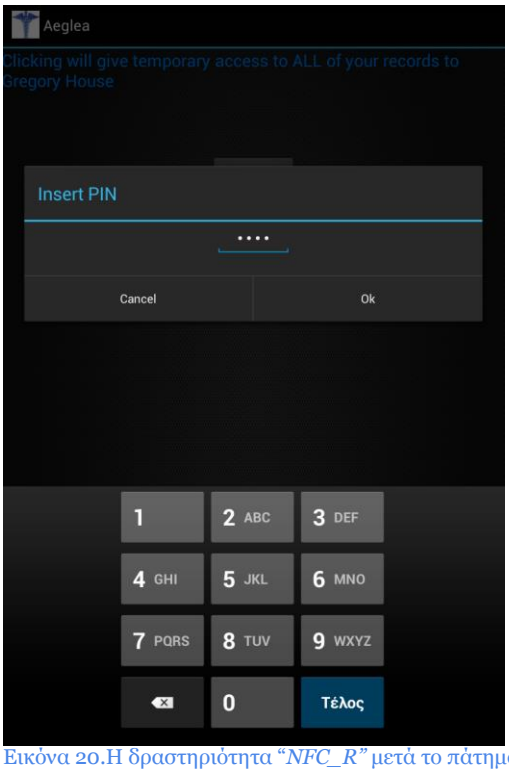


Εικόνα 18.Η δραστηριότητα *"NFCW"*, όπου ο χρήστης μπορεί να γράψει βασικά στοιχεία για την πρόσβαση τους σε επείγων περιπτώσεις από ειδική εφαρμογή(βλ. 4.7.)

Η δεύτερη δραστηριότητα με το όνομα "*NFC_R*" δεν είναι άμεσα προσβάσιμη από κάποια συντόμευση, αλλά εμφανίζεται μόνο αν προσεγγίσουμε στον αισθητήρα μία NFC καρτέλα με το "mimetype": "application/com.pgsideris.doctors". Αυτή η καρτέλα περιλαμβάνει ουσιώδη στοιχεία για την αναγνώριση του λογαριασμού του ιατρού στον οποίο θέλουμε να δώσουμε προσωρινά δικαιώματα, η οποία είναι και η κύρια λειτουργία αυτής της δραστηριότητας. Όταν ο χρήστης τοποθετήσει κάτω από τον αισθητήρα NFC της συσκευής του μια καρτέλα με το συγκεκριμένο "mimetype", τότε το λειτουργικό αναγνωρίζει ότι πρέπει να εκκινήσει την συγκεκριμένη δραστηριότητα. Στην συνέχεια η δραστηριότητα διαβάζει τα δεδομένα της καρτέλας και ενημερώνει τον χρήστη ότι πρόκειται να δώσει προσωρινά άδεια για την προβολή όλων αρχείων του ιστορικού του χρήστη στον συγκεκριμένο ιατρό. Αν επιλεχθεί να γίνει η εξουσιοδότηση, τότε του ζητείται ένας αριθμός "PIN" που χωρίς την έγκυρη εισαγωγή του δεν μπορεί να δώσει δικαιώματα προβολής. Όταν γίνει η εξουσιοδότηση ο ιατρός έχει δικαίωμα προβολής του ιστορικού του χρήστη μέσω της διαδικτυακής ιστοσελίδας. Τα δικαιώματα λήγουν μετά το πέρας δύο ωρών και αφαιρούνται από την βάση. Όπως αναφέρθηκε αυτό γίνεται, κυρίως, για λόγους ασφαλείας και διότι δεν είναι απαραίτητη η διατήρηση των δικαιωμάτων προβολής πέραν της ιατρικής επίσκεψης.



Εικόνα 19.Η δραστηριότητα *"NFC_R"*, όπου ο χρήστης προσωρινά δικαιώματα στον Ιατρό που έχει την ειδική καρτέλα NFC.



Εικόνα 20.Η δραστηριότητα *"NFC_R"* μετά το πάτημα του κουμπιού "Authorize" ζητείται το PIN

Για την υλοποίηση μιας ανταλλαγής δεδομένων με NFC πρέπει να ορίσουμε αν θα είμαστε σε κατάσταση λειτουργίας ενεργητική("write") ή παθητική("read"). Η δεύτερη δραστηριότητα δίνει την δυνατότητα μόνο παθητικής λειτουργίας, διότι η μόνη λειτουργία της που αφορά χρήση NFC είναι η ανάγνωση μίας καρτέλας NFC. Η πρώτη δραστηριότητα εξ' αρχής είναι σε παθητική κατάσταση λειτουργίας έτσι ώστε να υπάρχει η δυνατότητα ανάγνωσης μιας NFC καρτέλας με δεδομένα χρήστη, αλλά με το πάτημα του κουμπιού για την εγγραφή των δεδομένων χρήστη ενεργοποιείται η ενεργητική κατάσταση λειτουργίας, ώστε η εφαρμογή να έχει την δυνατότητα να γράψει δεδομένα σε μια καρτέλα NFC.

Για την αξιοποίηση του αισθητήρα NFC για ανάγνωση ή εγγραφή δεδομένων χρησιμοποιούνται οι μέθοδοι που μας παρέχει η κλάση του Android *"NfcAdapter"*, αλλά ταυτόχρονα και φίλτρα "intent" για να παραμετροποιηθούν οι ενέργειες των μεθόδων ανάλογα την κατάσταση λειτουργίας. Πιο συγκεκριμένα με την μέθοδο "enableForegroundDispatch" ενεργοποιούνται τα αποθηκευμένα φίλτρα intent, ανάλογα με την δράση που ανιχνεύθηκε κατά την ανάγνωση μιας καρτέλας NFC. Η μέθοδος αυτή δίνει τους πόρους προσκηνίου στα "intent" που είχαν καθοριστεί στα φίλτρα. Αυτά τα φίλτρα χρησιμοποιούνται για την εκτέλεση του "intent" όταν μια συγκεκριμένη δράση εντοπιστεί.

Για παράδειγμα όταν βρεθεί μία καρτέλα και περιλαμβάνει δεδομένα “NDEF” με το “mimetype” που έχουμε ορίσει τότε η δραστηριότητα βρίσκεται σε παθητική κατάσταση και έτσι πυροδοτεί το φίλτρο με την αντίστοιχη μεταβλητή. Σε αυτή την περίπτωση(παθητική) το φίλτρο περιλαμβάνει την περίπτωση που ο αισθητήρας ανιχνεύει μια καρτέλα NFC(ACTION_TAG_DISCOVERED) και ταυτόχρονα αυτή περιέχει δεδομένα NDEF που μπορεί να διαβάσει(ACTION_NDEF_DISCOVERED).

```
IntentFilter ndefDetected = new
IntentFilter(NfcAdapter.ACTION_NDEF_DISCOVERED);
IntentFilter tagDetected = new IntentFilter(NfcAdapter.ACTION_TAG_DISCOVERED);
ReadTagFilters = new IntentFilter[] { ndefDetected, tagDetected };
```

Ενώ όταν απλά ανιχνεύεται μια καρτέλα NFC(ACTION_TAG_DISCOVERED), αλλά δεν περιλαμβάνει αναγνώσιμα – από την συγκεκριμένη δραστηριότητα – NDEF δεδομένα, τότε η δραστηριότητα βρίσκεται σε κατάσταση εγγραφής και πυροδοτεί το κατάλληλα ορισμένο φίλτρο.

```
WriteTagFilters = new IntentFilter[] { tagDetected };
```

4.7. Ασφάλεια

Η ασφάλεια είναι από τις πιο σημαντικές προτεραιότητες για κάθε μέσο που αποθηκεύει και μεταδίδει πληροφορίες και ειδικά για ένα μέσο που διαχειρίζεται ευαίσθητα δεδομένα. Σε αυτή την ενότητα θα μελετηθούν οι δικλείδες που υπάρχουν στην εφαρμογή για την επίτευξη ασφάλειας κατά την αποθήκευση αλλά και την μετάδοση των δεδομένων. Μερικές από αυτές αναφέρθηκαν προηγουμένως, αλλά θα συνοψιστούν εδώ.

4.7.1. Στην συσκευή

Η εφαρμογή περιλαμβάνει μια βάση δεδομένων “SQLite” για την αποθήκευση των δεδομένων των χρηστών που δέχεται από τον απομακρυσμένο διακομιστή. Είναι απαραίτητο να υπάρχουν κάποια επίπεδα ασφαλείας για την ακεραιότητα των δεδομένων και την εξασφάλιση ότι δεν θα βρεθούν σε λάθος χέρια.

Για την μείωση του κινδύνου επίθεσης στην βάση δεδομένων από κακόβουλη εφαρμογή ο χρόνος ζωής των ευαίσθητων δεδομένων στην συσκευή παραμένει αρκετά μικρός. Αυτό επιτυγχάνεται αδειάζοντας την βάση κάθε φορά που τερματίζει η εφαρμογή. Σε περίπτωση που η εφαρμογή τερματίσει απρόσμενα και δεν αδειάσει η βάση κατά τον τερματισμό της εφαρμογής, έχει τεθεί να αδειάζει και κατά την εκκίνηση της εφαρμογής.

Μία επιπλέον πιθανή ευπάθεια είναι ότι ο χρήστης έχει την δυνατότητα να δώσει δικαιώματα σε έναν ιατρό μέσω της τεχνολογίας NFC, κάτι που μπορεί να γίνει γρήγορα και εύκολα μέσω ενός αγγίγματος. Η εύκολη πρόσβαση σε αυτή την δραστηριότητα έχει το ενδεχόμενο να αξιοποιηθεί από κάποιον κακόβουλο χρήστη, ο οποίος μπορεί γρήγορα και εύκολα να πάρει δικαιώματα πρόσβασης στο Ιστορικό του θύματος αν π.χ. πάρει την συσκευή του θύματος. Για την αντιμετώπιση αυτού του σεναρίου έχει τεθεί να ζητείται από τον χρήστη να οριστεί ένας αριθμός PIN, συγκεκριμένα για αυτή την λειτουργία. Κάθε φορά που θέλει να δώσει προσωρινά δικαιώματα προβολής μέσων NFC πρέπει να εισάγει σωστά τον αριθμό PIN.

4.7.2. Στον απομακρυσμένο διακομιστή

Η ασφάλεια δεδομένων πρέπει να εξασφαλίζεται όχι μόνο στην συσκευή, αλλά και από την πλευρά του απομακρυσμένου διακομιστή. Αυτό μπορεί να εξασφαλίσει από τις περιπτώσεις που κάποιος κακόβουλος χρήστης είτε έχει δημιουργήσει μια κακόβουλη εκδοχή της εφαρμογής ή εισάγει “ακάθαρτα” δεδομένα στην εφαρμογή με στόχο να αποσπάσει στοιχεία.

Όπως αναφέρθηκε χρησιμοποιούνται τεχνικές “καθαρισμού” δεδομένων για να γίνονται ερωτήματα στην βάση δεδομένων “MySQL” μετατρέποντας τα δεδομένα που δέχεται μόνο στην μορφή που απαιτείται. Σε περίπτωση που κάτι τέτοιο δεν αρκεί(π.χ. τα κακόβουλα δεδομένα είναι συμβολοσειρά και απαιτείται συμβολοσειρά), τότε χρησιμοποιούνται τα προετοιμασμένα ερωτήματα της τεχνολογίας “PDO” της γλώσσας “PHP”, τα οποία αποφεύγουν ερωτήματα που μπορούν να κάνουν ανεπιθύμητες ενέργειες, διαφορετικές από αυτές που ορίζει κανονικά το ερώτημα. Αυτό το επιτυγχάνει εξομοιώνοντας το ερώτημα δίχως δεδομένα και με αυτόν τον τρόπο αποφεύγεται η έγχυση δεδομένων πρώτου βαθμού. Επίσης, απαγορεύονται κάποια χαρακτήρες όπως “#”,”\”, “/” κτλ. και αντικαθιστώνται με τον κενό χαρακτήρα μέσω της συνάρτησης “str_replace()”.

Άλλη μια δικλείδα ασφαλείας είναι ο έλεγχος της ταυτότητας του χρήστη για κάθε αρχείο “.php” που ζητείται από την εφαρμογή. Για την διαπίστωση της ταυτότητας του χρήστη υπάρχει ένα τυχαίο μοναδικό αναγνωριστικό συνεδρίας, το οποίο δημιουργείται όταν ο χρήστης συνδέεται με τα σωστά αναγνωριστικά στην εφαρμογή. Κάθε φορά που ένα αρχείο “.php” ζητείται, ελέγχεται αν το αναγνωριστικό συνεδρίας είναι έγκυρο και αντιστοιχεί σε πραγματικό χρήστη και μόνο τότε μπορούν να γίνουν οι ενέργειες που ζητήθηκαν από την εφαρμογή.

4.7.3. Στην επικοινωνία

Η εφαρμογή "Aegle", ανάλογα των δράσεων του χρήστη, δέχεται και στέλνει δεδομένα στον απομακρυσμένο διακομιστή. Το γεγονός ότι αυτή λειτουργεί σε πλατφόρμα όπου η μόνη δυνατή επικοινωνία είναι μέσω του ασύρματου μέσου, κάνει την αποστολή δεδομένων εγγενώς μη αξιόπιστη. διότι οποιοσδήποτε είναι συνδεδεμένος στο ίδιο ασύρματο δίκτυο μπορεί να "ακούσει" τα πακέτα που κυκλοφορούν σε αυτό. Για αυτόν τον λόγο προκύπτει η ανάγκη κρυπτογράφησης της επικοινωνίας μεταξύ του απομακρυσμένου διακομιστή και της εφαρμογής.

Για να κρυπτογραφηθεί μία επικοινωνία χρειάζεται ένα πιστοποιητικό κρυπτογράφησης "SSL/TLS". Υπάρχουν δύο τρόποι για την απόκτηση τέτοιου πιστοποιητικού· ο πρώτος είναι να αγοραστεί από μια αρχή πιστοποίησης και ο δεύτερος είναι η χρήση αυτο-υπογεγραμμένου πιστοποιητικού. Σε αυτή την περίπτωση, χρησιμοποιείται πιστοποιητικό που παράχθηκε με την δεύτερη μέθοδο κάτι που καθιστά λιγότερη αξιόπιστη την επικοινωνία, αλλά είναι πρακτικό για τους σκοπούς της εφαρμογής, καθώς είναι δωρεάν και δεν χρειάζεται η διαδικασία ταυτοποίησης που μπορεί να ορίζει η κάθε πιστοποιητική αρχή.

Η ενσωμάτωση του πιστοποιητικού στην εφαρμογή γίνεται κατά τον ορισμό του "HTTP πελάτη". Το πρόβλημα είναι ότι το Android δέχεται μόνο πιστοποιητικά από μια λίστα με έγκυρες πιστοποιητικές αρχές και όχι αυτο-υπογεγραμμένα, όπως αυτό που χρησιμοποιηείται. Για να δεχθεί αυτό το είδος πιστοποιητικού πρέπει να δημιουργηθούν νέες παραμετροποιημένες εκδοχές από κάποιες κλάσεις που χρησιμοποιούνται στον "HTTP πελάτη" κατά την εγκαθίδρυση της σύνδεσης προς έναν διακομιστή.

Η πρώτη κλάση που δημιουργήθηκε είναι μια εκδοχή της *"X509TrustManager"* και ονομάστηκε *"MyTrustManager"*. Αυτή η κλάση χρησιμοποιείται για να διαχειρίζεται τα πιστοποιητικά που χρησιμοποιούνται για την επικοινωνία μέσω ασφαλών θυρών και ελέγχει αν συνάδουν με έμπιστο διακομιστή. Στην έκδοση που υλοποιήθηκε η διαφορά είναι ότι δέχεται όλους τους διακομιστές ως αξιόπιστους και οι λειτουργίες της κλάσης συνεχίζουν, ακόμη και αν η λίστα με τα πιστοποιητικά, που δέχεται ως παράμετρο, είναι κενή.

Ουσιαστική αλλαγή από την αρχική μορφή της κλάσης γίνεται με την παρακάτω γραμμή του "Constructor", όπου επιτρέπεται να δέχεται όλα τα πιστοποιητικά επαναφέροντάς τον διαχειριστή σε αρχική κατάσταση, έτσι ώστε ακόμη και αν είναι άδειος οι επόμενες μέθοδοι να συνεχίζουν τις λειτουργίες τους.

```
this.standardTrustManager = (X509TrustManager) trustmanagers[0];
```

Στην επόμενη κλάση που ορίζεται παρακάτω καλείται αυτή η εντολή, όπου το στιγμιότυπο της *"MyTrustManager"* δημιουργείται με κενές παραμέτρους .

```
context.init(null, new TrustManager[] { new MyTrustManager(null) }, null);
```

Η δεύτερη κλάση που δημιουργήθηκε ονομάζεται "*MySSLFactory"* και είναι μια εκδοχή της κλάσης "*SSLSocketFactory"*. Αυτή χρησιμοποιείται για την επικύρωση του διακομιστή "HTTPS" και ενεργοποιεί ή απορρίπτει την σύνδεση σύμφωνα με μία λίστα αξιόπιστων διακομιστών[64]. Η κλάση δημιουργεί ένα γενικό πλαίσιο "SSL" με την δυνατότητα να δέχεται όλα τα πιστοποιητικά(χρησιμοποιώντας κενό στιγμιότυπο του "*MyTrusmanager*") και στην συνέχεια ορίζεται να δέχεται όλους τους διακομιστές ως έμπιστους. Αυτό γίνεται αντικαθιστώντας την μέθοδο που ελέγχει αν ένας διακομιστής είναι έμπιστος και κάνοντάς την να επιστρέφει πάντα "αληθές".

```
public boolean isSecure(Socket socket) throws IllegalArgumentException {
        return true;
}
```

Όλα τα παραπάνω υλοποιούνται στον προσαρμοσμένο "HTTP πελάτη", "*CustomHttpclient"*. Για την περάτωση ασφαλούς σύνδεσης ορίζονται κατάλληλα οι παράμετροι σύνδεσης και αντιστοιχίζονται οι κατάλληλοι διαχειριστές στα μοτίβα που επιτρέπονται. Παράδειγμα, για το μοτίβο που περιέχει "https" αντιστοιχίζεται ένα στιγμιότυπο της "*MySSLFactory"* στην θύρα 443. Όλα αυτά πρέπει να υλοποιηθούν προτού καθοριστεί ο "HTTP πελάτης", διότι κάποιες ρυθμίσεις πρέπει να έχουν ήδη οριστεί, ώστε να γίνουν παράμετροι κατά την δημιουργία του "πελάτη".

Στο κομμάτι κώδικα που ακολουθεί ορίζονται τα επιτρεπόμενα σχεδιαγράμματα και αντιστοιχίζονται στις κλάσεις που διαχειρίζονται τις θύρες σύνδεσης.

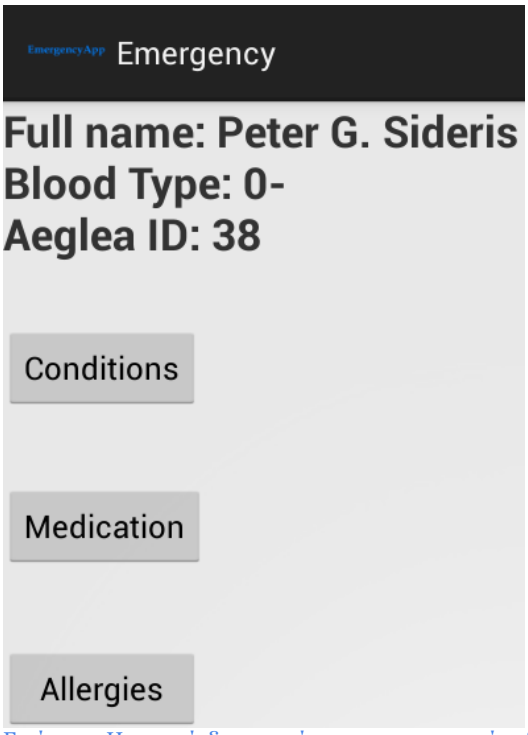
```
schemeRegistry.register(new Scheme("http", PlainSocketFactory.getSocketFactory(),
80));
schemeRegistry.register(new Scheme("https", new MySSLFactory(), 443));
```

Οι παράμετροι που θα δοθούν κατά την δημιουργία του "πελάτη HTTP" φαίνονται στο παρακάτω κομμάτι κώδικα. Οι δύο πρώτες ορίζουν ότι θα είναι μοναδική η σύνδεση και η τελευταία ενεργοποιεί τηνχειραψία("handshake"), η οποία είναι απαραίτητη για την περάτωση ασφαλούς κρυπτογραφημένης επικοινωνίας.

```
params.setParameter(ConnManagerPNames.MAX_TOTAL_CONNECTIONS, 1);
params.setParameter(ConnManagerPNames.MAX_CONNECTIONS_PER_ROUTE,
                                                    new ConnPerRouteBean(1));
params.setParameter(HttpProtocolParams.USE_EXPECT_CONTINUE, false);
```

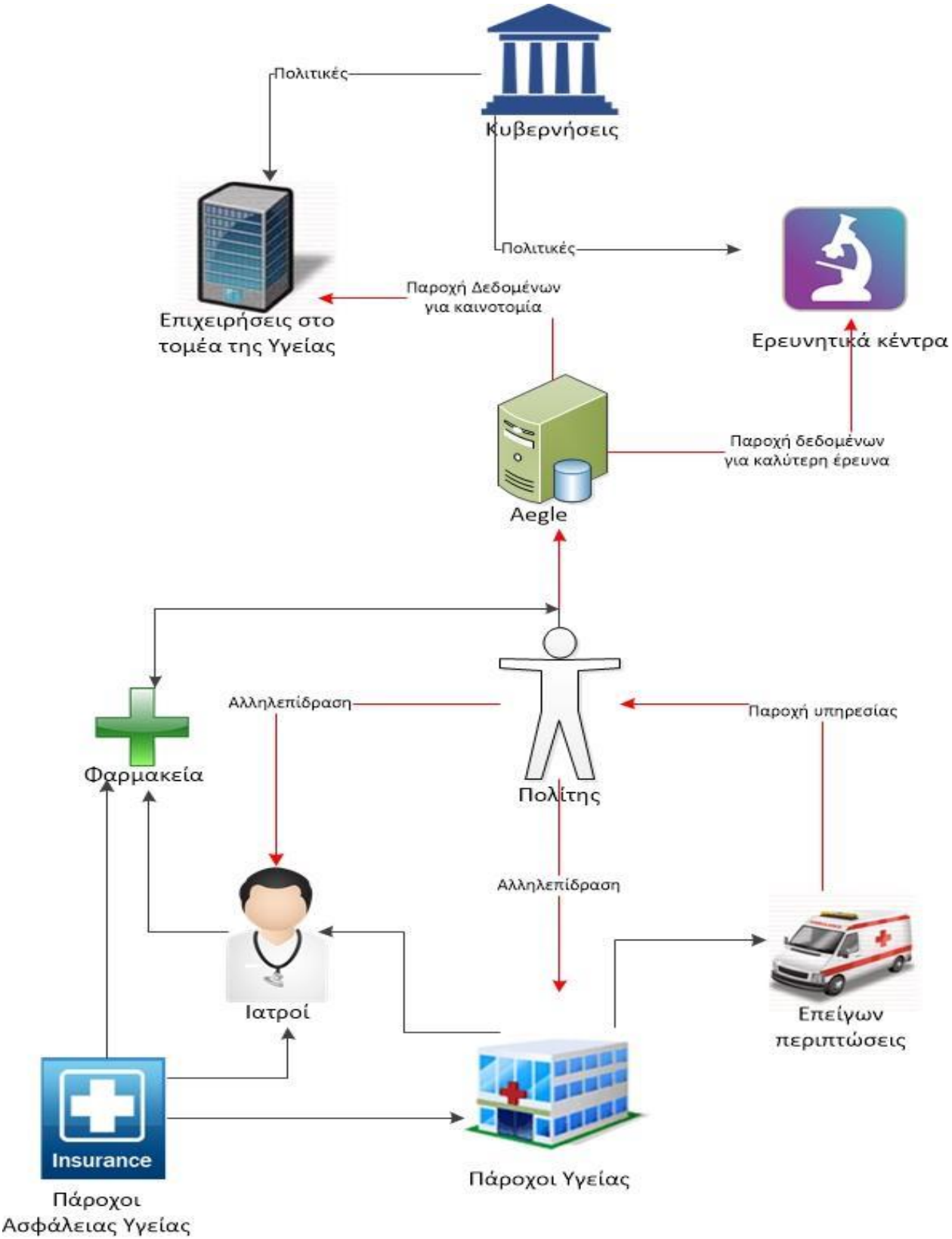
4.8. Βοηθητική εφαρμογή “Emergency”

Ο σκοπός για την εγγραφή των δεδομένων χρήστη σε καρτέλα NFC είναι να υπάρχουν διαθέσιμα αυτά τα δεδομένα όταν ο χρήστης δεν μπορεί να παρέχει ο ίδιος άδειες προβολής και είναι απαραίτητη η πρόσβαση σε αυτά(π.χ. σε ένα ατύχημα). Για την επίτευξη αυτού του σκοπού είναι απαραίτητη η ύπαρξη “ειδικής” εφαρμογής, η οποία θα αναγνωρίζει το κατάλληλο “mimetype”, θα διαβάζει την καρτέλα και θα εμφανίζει τα βασικά στοιχεία του χρήστη(όνομα, επίθετο, ομάδα αίματος). Επίσης, θα πρέπει να υπάρχει την δυνατότητα προβολής αλλεργιών, χρόνιων παθήσεων και φαρμακευτικών αγωγών σε περίπτωση που χρειαστούν. Από άποψη ασφάλειας δεδομένων αυτή η εφαρμογή είναι πολύ ευαίσθητη και θα πρέπει να υπάρχει μόνο σε εξουσιοδοτημένες συσκευές.



Εικόνα 21.Η αρχική δραστηριότητα της εφαρμογής “Emergency”, η οποία διαβαζει την καρτέλα που έγραψε ο χρήστης στην δραστηριότητα “*NFCW*”.

Με αυτή την εφαρμογή και με την “Aegle” καλύπτεται ένα σημαντικό φάσμα της αλληλεπίδρασης των πολιτών με τους φορείς της Υγειονομικής περίθαλψης, ενώ ταυτόχρονα εξασφαλίζεται ότι ο χρήστης είναι η μοναδική οντότητα που έχει τον απόλυτο έλεγχο των δεδομένων του. Μια απεικόνιση του εφικτού εύρους δράσης όλου του συστήματος μαζί με τον ιστότοπο και τις δύο εφαρμογές φαίνεται με κόκκινες γραμμές στο σχήμα παρακάτω.



Εικόνα 22. Επίδραση του "Aegle" στην Υγειονομική περίθαλψη

Γ' Μέρος - Προτάσεις για βελτιστοποίηση της εμπειρίας χρήσης της εφαρμογής "Aegle"

1. Εισαγωγή

Όπως κάθε πρόγραμμα, έτσι και η εφαρμογή "Aegle" επιδέχεται βελτιώσεις, αλλαγές και επεκτάσεις σχεδιασμένες να βελτιστοποιήσουν την λειτουργικότητα και την ασφάλεια χρήσης της εφαρμογής, αλλά και να κάνουν πιο ευχάριστη την εμπειρία χρήστη. Σε αυτό το μέρος θα εξεταστούν μερικές προτάσεις για την επίτευξη αυτών των στόχων. Συγκεκριμένα θα εξεταστεί η υλοποίηση ασφαλούς επικοινωνίας με τον διακομιστή, η επέκταση λειτουργιών, διάφορες εναλλαγές στις ήδη υπάρχουσες λειτουργίες οι οποίες μπορεί να κάνουν πιο εύκολη την συντήρηση της ανάπτυξης και ο καλύτερος σχεδιασμός της διεπαφής χρήστη.

2. Προσθήκες στην Ασφάλεια

Σε κάθε μέσο που επεξεργάζεται ηλεκτρονικά πληροφορίες μία από τις πιο σημαντικές ευθύνες είναι η ασφάλεια των δεδομένων των χρηστών της, αλλά και η ασφάλεια για την απρόσκοπτη λειτουργία του ίδιου του μέσου. Η ασφάλεια της πληροφορίας πρέπει να περιλαμβάνεται στην αποθήκευση και στην μετάδοση αυτής, έτσι ώστε να υπάρχει πλήρης κάλυψη. Στην εφαρμογή "Aegle" αυξημένα μέτρα ασφάλειας μπορούν να είναι η κρυπτογράφηση της βάσης δεδομένων, αλλά και πιο σημαντικά η κρυπτογράφηση της επικοινωνίας με τον απομακρυσμένο διακομιστή.

2.1. Κρυπτογράφηση Βάσεως Δεδομένων SQLite

Η βάση δεδομένων μπορεί να κρυπτογραφηθεί με αλλαγή του πυρήνα της βιβλιοθήκης "SQLite" μεθόδων και κλάσεων για αυτό το σκοπό. Αν επιθυμείται έτοιμη λύση υπάρχει η βιβλιοθήκη "SQLCipher" της εταιρείας "Zenetic" και στοιχίζει περί τα 120€[50], αλλά δίνεται και η δυνατότητα απόκτησης του πηγαίου κώδικα[51] αφού είναι βιβλιοθήκη ανοιχτού κώδικα.

Παρόλα αυτά δεν θα συνιστούσα την υλοποίηση κρυπτογράφησης της βάσης δεδομένων, καθώς απαιτεί την απομνημόνευση ενός επιπλέον κωδικού από τον χρήστη και κάτι τέτοιο δεν βοηθάει στην βελτίωση εμπειρίας και μειώνει την απόδοση της συσκευής εφαρμογής καθώς σε κάθε ανανέωση δεδομένων θα πρέπει να γίνεται αποκρυπτογράφηση. Επιπλέον, τα δεδομένα έχουν μικρό χρόνο ζωής στην συσκευή μειώνοντας έτσι τον κίνδυνο μη εξουσιοδοτημένης πρόσβασης σε αυτά.

2.2. Κρυπτογράφηση επικοινωνίας με τον απομακρυσμένο διακομιστή

Οι κινητές συσκευές και οι υπολογιστές - ταμπλέτες που τρέχουν το λειτουργικό Android συνδέονται στο διαδίκτυο είτε μέσω ασύρματου δικτύου "Wi-Fi"(IEEE 802.11 b/g/n) ή μέσω δικτύου κινητής επικοινωνίας, άρα κάθε επικοινωνία με τον απομακρυσμένο διακομιστή διεξάγεται μέσω ασύρματου φυσικού μέσου. Αυτό το γεγονός καθιστά την μη κρυπτογραφημένη επικοινωνία ως επισφαλή για υποκλοπή από έναν απλό "sniffer" πακέτων συνδεδεμένο στο ίδιο δίκτυο.

Η χρήση ενός αυτο-υπογεγραμμένου πιστοποιητικού είναι ανοιχτό σε επιθέσεις τύπου "man in the middle". Επίσης, το γεγονός ότι έπρεπε να γίνει όλη αυτή η παράκαμψη μεθόδων του Android, λόγω περιορισμών αυτού, δεν μας εξασφαλίζει από έναν έμπειρο κακόβουλο χρήστη ο οποίος μπορεί να εξάγει τον κώδικα από το εκτελέσιμο αρχείο. Για αυτό θα συνιστούσα την χρήση πιστοποιητικού που έχει εκδοθεί από πιστοποιητική αρχή.

Σε περίπτωση που ακόμη δεν θέλουμε να χρησιμοποιηθεί πιστοποιητικό από κάποια αρχή υπάρχει η δυνατότητα να εισάγουμε τα πιστοποιητικά μας στην λίστα με τους έμπιστους διακομιστές και να δεχόμαστε μόνο αυτό. Αυτό δεν θα απαιτούσε όλη την παράκαμψη των μεθόδων που έγινε, αλλά θα έπρεπε να αλλαχθεί η λίστα με τους έμπιστους διακομιστές.

Αν θέλουμε το πιστοποιητικό να προέρχεται από πιστοποιητική αρχή τότε απλά πρέπει να ελέγχεται αν ταιριάζει το πιστοποιητικό με το όνομα του διακομιστή και να δεχόμαστε μόνο αυτό. Αυτό γίνεται με την διαχείριση των θυρών "SSL" και πρέπει στο κώδικά μας να παρέχουμε τον κωδικό και τα πιστοποιητικά που δεχόμαστε. Για παράδειγμα στην παρακάτω υλοποίηση διαχείρισης θυρών "SSL" ορίζεται η κωδικοποίηση αποθήκευσης κλειδιών κρυπτογράφησης "Bouncy Castle(BKS)", στη συνέχεια φορτώνονται τα πιστοποιητικά(R.raw.mystore) και εισάγεται ο κωδικός("PASSWORD").

```
private SSLSocketFactory newSslSocketFactory() {
    try {
      KeyStore trusted = KeyStore.getInstance("BKS");
      InputStream in = context.getResources().openRawResource(R.raw.mystore);
      try {
        trusted.load(in, "PASSWORD".toCharArray());
      } finally {
        in.close();
      }
      return new SSLSocketFactory(trusted);
    } catch (Exception e) {
      throw new AssertionError(e);
    }}}
```

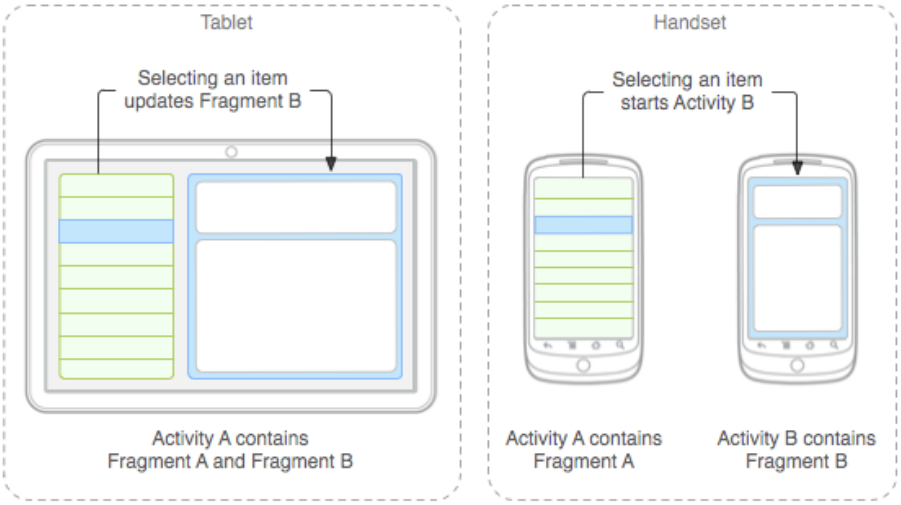
Αυτή η μέθοδος επιστρέφει τις παραμετροποιημένες θύρες και εισάγεται ως παράμετρος σε έναν νέο καταχωρητή προτύπων επικοινωνίας ("*SchemeRegistry*"), ο οποίος στη συνέχεια γίνεται παράμετρος της μεθόδου δημιουργίας του διαχειριστή σύνδεσης ("*ClientConnectionManager*"), ο οποίος που βοηθάει στην παραμετροποίηση της σύνδεσης. Όλα τα παραπάνω πρέπει να περιλαμβάνονται στην υλοποίηση του "HTTP πελάτη" και πιο συγκεκριμένα στη κλάση "*CustomHttpClient*".

3. Σχεδιασμός διεπαφής χρήστη

Ένα από τα πιο σημαντικά καθήκοντα κατά την ανάπτυξη μίας εφαρμογής είναι ο σχεδιασμός της εμφάνισης της(“design”), έτσι ώστε να είναι φιλική στον χρήστη και ταυτόχρονα να είναι πρακτική. Στην εφαρμογή “Aegle” παρά το γεγονός ότι έχουν υλοποιηθεί πολλές από τις προτάσεις της Google για το “design” μιας εφαρμογής, αυτή επιδέχεται πολλές βελτιώσεις για την επέκταση της λειτουργικότητας και ευκολίας χρήσης. Παρακάτω θα εξεταστούν μερικές από τις βελτιώσεις που μπορούν να γίνουν.

3.1. Fragments

Τα “θραύσματα”(“fragments”) αντιπροσωπεύουν ένα μέρος της διεπαφής χρήστη σε μία δραστηριότητα. Μια δραστηριότητα μπορεί να περιέχει πολλά “fragments”, τα οποία όλα μαζί αποτελούν την διεπαφή χρήστη. Από τις τελευταίες εκδόσεις του λειτουργικού Android παρέχεται η δυνατότητα εμφώλευσης ενός “fragment” μέσα σε ένα άλλο. Η λογική στην χρήση των “fragments” είναι ότι λόγω του φάσματος μεγέθους των οθονών, των συσκευών που τρέχουν Android, οι προγραμματιστές πρέπει να προσαρμόσουν κατάλληλα την διεπαφή χρήστη ανάλογα με το μέγεθος της οθόνης. Αυτό μπορεί να γίνει εύκολα στον καθορισμό του “XML” σχεδιασμού της δραστηριότητας με ένα σχετικό “RelativeLayout”, αλλά δεν διευκολύνει σε πιο πολύπλοκες σχεδιαστικές υλοποιήσεις. Τα “fragments”, ανάλογα με το μέγεθος οθόνης και το πως θέλει ο σχεδιαστής να εμφανίζονται, μπορούν να ανήκουν στην μία δραστηριότητα ή στην επόμενη, μπορούν να εμφανίζονται στο προσκήνιο ή να εξαφανίζονται στο παρασκήνιο και να βρίσκονται σε διάφορες θέσεις αναλόγως του μεγέθους της οθόνης και τον προσανατολισμό της συσκευής.



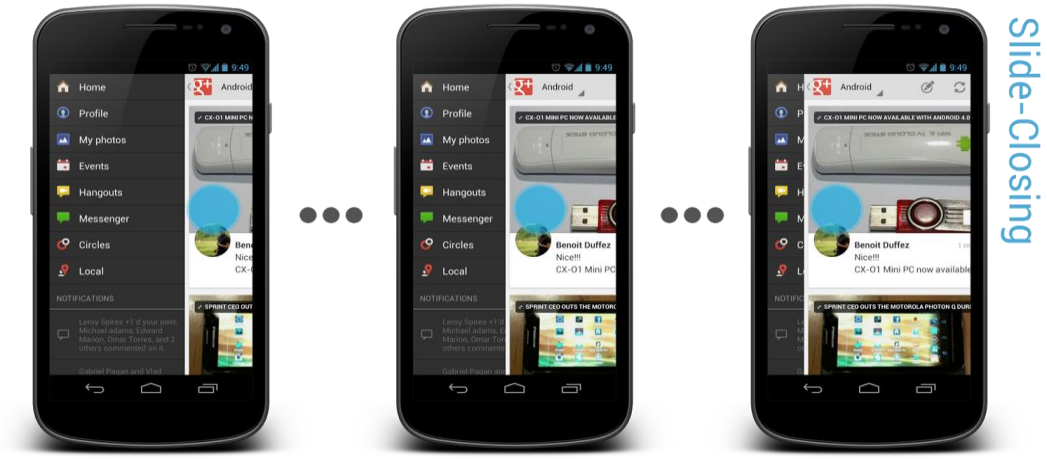
Εικόνα 23. Η διαφορετική υλοποίηση δύο fragment σε ταμπλέτα και κινητή συσκευή. Στην ταμπλέτα και τα δύο fragment είναι σε μια δραστηριότητα, ενώ στο κινητό “σπάνε” σε δύο δραστηριότητες[52]

Η υλοποίηση αυτής της σχεδιαστικής λογικής είναι αρκετά περίπλοκη και καταλαμβάνει μεγάλο κομμάτι της υλοποίησης, αλλά είναι πολύ καλή μέθοδος ώστε να είναι η εφαρμογή συμβατή με την πλειοψηφία των συσκευών που κυκλοφορούν. Πολλές εταιρείες αποφασίζουν να έχουν δύο εφαρμογές – για ταμπλέτες και για κινητές συσκευές – όμως κάτι τέτοιο ανεβάζει το κόστος ανάπτυξης και συντήρησης και ταυτόχρονα μπερδεύει τον χρήστη διατηρώντας την εμπειρία χρήσης ξεχωριστή για κάθε συσκευή.

3.2. Επιπλέον παρεμβάσεις στο “design” της εφαρμογής

Κάποιες επιπλέον προσθήκες θα μπορούσαν να διευκολύνουν την εμπειρία χρήστη. Για παράδειγμα στην δραστηριότητα προσθήκης αδειών η διαδικασία επιλογής μπορεί να μπερδέψει κάποιον χρήστη αρχικά, διότι του παρουσιάζει δύο λίστες· μία με τους γιατρούς που μπορεί να επιλέξει για να παρέχει δικαιώματα και μία άλλη με το Ιατρικό του Ιστορικό. Μια καλύτερη υλοποίηση θα ήταν με “fragments”, ανά βήμα όπου ο χρήστης θα επιλέγει πρώτα τον ιατρό που θέλει να δώσει άδεια και στη συνέχεια να εμφανίζεται η λίστα με το Ιστορικό του χρήστη. Η εμφάνιση θα εξαρτάται από το μέγεθος οθόνης.

Επίσης, άλλη μια πρόταση θα ήταν η χρήση “fragments” στην αρχική οθόνη ώστε να υπάρχει πρόσβαση στις έξι κατηγορίες του Ιατρικού Ιστορικού από μενού τύπου “συρτάρι”(“drawer”) και όχι με κουμπιά όπως είναι τώρα. Κάτι τέτοιο θα καθιστούσε την εμφάνιση πολύ πιο ελκυστική και καλύτερα οργανωμένη.



Εικόνα 24. Υλοποίηση “drawer” στην εφαρμογή Google+[53]. Αντίστοιχη υλοποίηση μπορεί να γίνει στην αρχική δραστηριότητα για πρόσβαση στο Ιατρικό Ιστορικό χρήστη.

Για να υλοποιηθούν οι παραπάνω προτάσεις πρέπει να προστεθεί η δυνατότητα υποστήριξης “fragments” και να αλλάξουν αρκετά υπάρχοντα σχέδια διεπαφής.

4. Επέκταση λειτουργιών

Ο ευρύτερος τομέας της Ιατρικής που ασχολείται η εφαρμογή παρέχει πολλές δυνατότητες για επέκταση των λειτουργιών της εφαρμογής. Έτσι η εφαρμογή αλλά και η ιστοσελίδα που βασίζεται μπορούν να εξελιχθούν σε μια ενοποιημένη πλατφόρμα που παρέχει ένα σύνολο υπηρεσιών με ευρύ φάσμα λειτουργιών και εφαρμογών στο τομέα της Ιατρικής, της Φαρμακευτικής και όλων των παρακείμενων τομέων. Για την επίτευξη αυτού χρειάζεται μια οργανωμένη προσπάθεια, οικονομικοί πόροι, επίτευξη συνεργασιών και βελτίωση του νομικού πλαισίου για τα προσωπικά ιατρικά δεδομένα.

4.1. Άμεσα υλοποιήσιμες λειτουργίες

Η πιο βασική προσθήκη λειτουργίας στην εφαρμογή είναι να δοθεί στον χρήστη η δυνατότητα να προσθέτει εγγραφές στο Ιστορικό του από την εφαρμογή. Παρόλα αυτά δεν συνιστώ την ύπαρξη ιατρικών απεικονίσεων προς “ανέβασμα” σε κινητή συσκευή, διότι τα προγράμματα “gallery” που ανοίγουν τις εικόνες παρέχουν ανοιχτή πρόσβαση σε αυτές μέσω των intent τους και έτσι μπορεί να υποκλαπεί από μία κακόβουλη εφαρμογή.

Άλλη μια βασική αναβάθμιση που μπορεί να γίνει στην εφαρμογή, είναι η ανανέωση της βάσης δεδομένων σύμφωνα με υπάρχοντα πρότυπα Ηλεκτρονικού Ιατρικού Φακέλου Ασθενούς έτσι ώστε να υπάρχει διαλειτουργικότητα και με άλλες αντίστοιχες υπηρεσίες, αλλά και η εύκολη ενσωμάτωση ή απορρόφηση άλλων προ-υπάρχοντων συστημάτων. Αυτό θα απαιτούσε και την αναβάθμιση της βάσης δεδομένων του απομακρυσμένου διακομιστή. Αυτό θα έφερνε την εφαρμογή πιο κοντά στην υλοποίηση ενός πλήρους ΗΙΦΑ.

Μια διαφοροποίηση που θα διευκόλυνε την συντήρηση της εφαρμογής σε περίπτωση που επεκταθεί και η συντήρηση της γίνει δαπανηρή, θα ήταν η αλλαγή του συστήματος επικοινωνίας με τον απομακρυσμένο διακομιστή από “JSON” σε ένα σύστημα που να βασίζεται στο “XML”. Για μικρά μηνύματα υπάρχει το “Google Cloud Messaging for Android”, το οποίο είναι επί πληρωμή και επιτρέπει την αποστολή(“push”) δεδομένων σε εφαρμογές, αλλά δεν επιτρέπει μηνύματα πάνω από 4KB. Γενικότερα για διαδικτυακή επικοινωνία με “XML” υπάρχει το πρωτόκολλο “SOAP(Simple Object Access Protocol)”, το οποίο είναι πολύ γνωστό για μετάδοση δομημένης πληροφορίας σε υπηρεσίες διαδικτύου. Όμως, περιέχει αρκετές επιπλέον λειτουργίες από αυτές που μπορεί να χρειαστεί μια εφαρμογή και έτσι επιβαρύνεται η επικοινωνία. Για μια ελαχιστοποιημένη υλοποίηση των υπηρεσιών που παρέχει το “SOAP” μπορούμε να στραφούμε σε υπηρεσίες “ReST(Representational State Transfer)” για υλοποίηση αποδοτικότερης προτυποποιημένης επικοινωνίας με τον διακομιστή. Οι διαδικτυακές υπηρεσίες “REST” είναι γενικότερα ελαφρύτερες, πιο εύκολες στην κατανόηση και πιο ευέλικτες στην ανάπτυξή τους. Γενικότερα, η χρήση “REST” ή “SOAP” για υλοποίηση διαδικτυακών υπηρεσιών έχει διχάσει την συγκεκριμένη κοινότητα, αλλά φαίνεται πως κινούμαστε σε τεχνολογίες “RESTful”[65].

Μερικά από τα πλεονεκτήματά των υπηρεσιών που παρέχει η τεχνολογία “REST” είναι[54]:

- Συγκεκριμένη υλοποίηση HTTP μεθόδων(POST, GET κτλ)
- Είναι “stateless”, δηλαδή αντιμετωπίζει κάθε ερώτημα ως ξεχωριστό
- Έχει δομή καταλόγου με την μορφή “URI”, κάτι που είναι εγγενές στο Android
- Μπορεί να μεταφέρει πληροφορία κωδικοποιημένη με XML, JSON ή και τα δύο

4.2. Μελλοντικές λειτουργίες

Η περαιτέρω ανάπτυξη της ιστοσελίδας και της εφαρμογής σε μία πλήρη, ενοποιημένη και αποκεντρωμένης πλατφόρμας μπορεί να γίνει μόνο υπό την αιγίδα μιας επιχείρισης με πόρους για έρευνα, ανάπτυξη και για σύναψη στρατηγικών συνεργασιών. Η φύση της πλατφόρμας παρέχει τις δυνατότητες επέκτασης σε αρκετούς τομείς, καθώς η Υγειονομική περίθαλψη είναι πάρα πολύ ευρύς τομέας. Για παράδειγμα υπάρχει η δυνατότητα δημιουργίας υπηρεσιών για νέες οντότητες χρηστών όπως ιατρικά κέντρα, νοσοκομεία και ασφαλιστικές υπηρεσίες. Όλες αυτές οι οντότητες χρηστών μπορούν να συνεργάζονται μεταξύ τους για το όφελος του χρήστη/ασθενή/πολίτη.

Ένα παράδειγμα είναι ότι σε μία επίσκεψη σε ένα κέντρο Περίθαλψης ο ασθενής μπορεί να δώσει προσωρινά δικαιώματα στους ιατρούς, οι οποίοι μπορούν μέσω των συσκευών διάγνωσης να “ανεβάζουν” απευθείας στο Ιστορικό του χρήστη τις εξετάσεις του ασθενή, όπου θα τις βλέπει ο συνεργάτης του ιατρού μερικά χιλιόμετρα μακριά την στιγμή που βγαίνουν τα αποτελέσματα. Τέλος, το κέντρο θα ειδοποιούσε την ασφαλιστική που καλύπτει τον χρήστη για τα κόστη περίθαλψης.

Ακούγεται ιδανικό και ουτοπικό, αλλά αυτό στόχευε να δείξει η αυτή η διπλωματική εργασία· ένα μικρό κομμάτι της τεχνολογίας που θα χρησιμοποιούμε – ελπίζω όχι πολλά χρόνια από σήμερα – για την παρακολούθηση της υγείας του ατόμου. Η υλοποίηση που περιγράφηκε προηγουμένως θα κάλυπτε το μεγαλύτερο μέρος του τομέα Υγειονομικής Περίθαλψης, όπως συζητήθηκε στο Α’ Μέρος, και σύμφωνα με το όραμα και τους στόχους που έθεσε η Ευρωπαϊκή Ένωση. Τέλος, βαθύτερος στόχος αυτής της διπλωματικής είναι να αποτελέσει τον κινητήριο μοχλό έμπνευσης για την υλοποίηση ενός τέτοιου συστήματος που θα φέρει την Ιατρική Φροντίδα του Ατόμου στον 21ο αιώνα.

Πηγές

1 Gunter T.D., Terry N.P. (2005). *"The Emergence of National Electronic Health Record Architectures in the United States and Australia: Models, Costs, and Questions". J Med Internet Res* 7: 1.

2 Dave Garets, Mike Davis,Electronic Patient Records, (Oct. 2005). *"EMRs and EHRs ,Concepts as different as apples and oranges at least deserve separate names."*. Healthcare Informatics Online

3 Smaltz, Detlev, Eta Berner. *The Executive's Guide to Electronic Health Records.' (2007, Health Administration Press) p.03*

4 "Health Information Exchanges and Your EMR Selection Process," New England Journal of Medicine, January 25, 2011

5 Healthcare Information and Management Systems Society. 2003. Retrieved 2006-07-28.

6 Are More Doctors Adopting EHRs? (http://goo.gl/IBOcw)

7 DIRECTIVE 2011/24/EU OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL (http://goo.gl/tKrWl)

8 Patrick Kierkegaard (2011) Electronic health record: Wiring Europe's healthcare, Computer Law & Security Review, Volume 27, Issue 5, September 2011, Pages 503-515,

9 (http://goo.gl/WU8PC) και "The future of healthcare in Europe", The Economist Intelligence Unit Limited 2011 (http://goo.gl/Ma1RB)

10 Redesigning Health in Europe for 2020 (http://goo.gl/WbYlG)

11 HL7 IP Policy, (http://goo.gl/mrfdr)

12 HL7 Standards, (http://goo.gl/KBFMh)

13 Wikipedia, Methods applied by HL7 (http://goo.gl/i3TR5)

14 LLP - Lower Layer Protocol(http://goo.gl/wNT4e)

15 HL7 FAQs (http://goo.gl/y4WzH)

16 Understanding HL7 Messages (http://goo.gl/MdsL5)

17 HL7 Messages and descriptions (http://goo.gl/wB7c1)

18 Reference Information Model, (http://goo.gl/4jMeF)

19 Tools and Resources(http://goo.gl/B4QkI)

20 The CDAtm Book, Keith W. Boone(2011)

21 DICOM Final Supplements in the 2011 version (http://goo.gl/8PjlF)

22 www.startuphealth.com/

23 Gunter T.D., Terry N.P. (2005). "The Emergence of National Electronic Health Record Architectures in the United States and Australia: Models, Costs, and Questions". *J Med Internet Res* 7: 1.

24 McGee, Marianne Kolbasuk (January 19, 2010). "Vanguard Rolls out Dossia PHRs To Employees".

25 OnWeb (http://goo.gl/v9qJq)

26 OnMobile(http://goo.gl/Bo2Mh)

27 OnCard(http://goo.gl/aervz)

28 Andy Rubin tweet(http://goo.gl/oMpQJ)

29 Elgin, Ben (August 17, 2005). "Google Buys Android for Its Mobile Arsenal". *Bloomberg Businessweek*. Bloomberg.

30 "Industry Leaders Announce Open Platform for Mobile Devices" (Press release). Open Handset Alliance. November 5, 2007

31 "500 million devices activated globally, and over 1.3 million added every single day.". official Android Engineering teams. 2012-09-12. (http://goo.gl/HHC82)
32 Tim Bray (November 24, 2010). "What Android Is".
33 Shah, Agam (December 1, 2011). "Google's Android 4.0 ported to x86 processors". *Computerworld*. International Data Group.
34 *Androidology – Part 1 of 3 – Architecture Overview*(Video). YouTube. (http://goo.gl/pLE7R)
35 Paul, Ryan (February 23, 2009). "Dream(sheep++): A developer's introduction to Google Android". *Ars Technica*.
36 "Simple DirectMedia Layer for Android". sdl.org. 2012-08-12.
37 Steven J. Vaughan-Nichols (August 18, 2011). "Linus Torvalds on Android, the Linux fork". (http://goo.gl/Xihrx)
38 Chris von Eitzen (December 23, 2011). "Android drivers to be included in Linux 3.3 kernel". *h-online.com*
39 Jonathan, Corbet. "Autosleep and wakelocks". LWN.(http://goo.gl/LfId9)
40 Android Permissions (http://goo.gl/agdnR)
41 Android Developers, Starting an activity (http://goo.gl/auDlQ)
42 Android Developers, Tasks and Back Stack(http://goo.gl/02i2o)
43 Bornstein, Dan (2008-05-29). "Presentation of Dalvik VM Internals" Google. p. 22(http://goo.gl/drnQc)
44 Nexus One Is Running Android 2.2 Froyo. How Fast Is It Compared To 2.1? Oh, Only About 450% Faster". 2010-05-13. (http://goo.gl/psrCt)
45 "Technical Specifications". NFC Forum.(http://goo.gl/VEzvi)
46 "ISO/IEC 18092:2004 Information technology -- Telecommunications and information exchange between systems -- Near Field Communication -- Interface and Protocol (NFCIP-1)"(http://goo.gl/5P84L)
47 "Distinctive Features Of SQLite". SQLite. June 14, 2012. (http://goo.gl/pqyCG)
48 "Most Widely Deployed SQL Database Estimates". Sqlite.org. (http://goo.gl/a20TM)
49 Medscape Physician Compensation Report: 2011 (http://goo.gl/n2Mfp)
50 Zenetic LLC Store(http://goo.gl/lNwv1)
51 SQLCipher Github source code (http://goo.gl/ATJYu)
52 Android Developers: Fragments (http://goo.gl/Ot82b)
53 Android UIUX: Nav - Drawer (http://goo.gl/8ynlu)
54 IBM: RESTful Web Services: (http://goo.gl/jDYco)
55 PHP.net: PDO introduction(http://goo.gl/Z5PpF)
56 PHP.net: PDO::prepare() (http://goo.gl/WE4Z2)
57 HealthVault (http://goo.gl/cr9Jv)
58 Dossia (http://goo.gl/X5uUk)
59 World Medical Card (http://www.wmc-card.com/)
60 Avado (http://www.avado.com/)
61 Datatypes in SQLite 3 (http://goo.gl/kMgJs)
62 NFC Forum: Specifications (http://goo.gl/ZDTYx)
63 SQLite: Foreign key support (http://goo.gl/FHMn9)
64 Android Developers:SSLSocketFactory(http://goo.gl/Mk8OS)
65 Benslimane, Djamal; Schahram Dustdar, and Amit Sheth (2008). "Services Mashups: The New Generation of Web Applications". *IEEE Internet Computing, vol. 12, no. 5*. Institute of Electrical and Electronics Engineers. pp. 13–15.