



**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**

**ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ ΚΑΙ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ**

**ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ  
ΥΠΟΛΟΓΙΣΤΩΝ**

**Σχεδίαση και ανάπτυξη πλατφόρμας διαχείρισης  
προσωπικών αρχείων υγείας με χρήση του προτύπου HL7  
FHIR**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

του

**ΑΝΑΣΤΑΣΗ ΜΠΟΥΦΗ**

**Επιβλέπων :** Ιάκωβος Βενιέρης  
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2014





ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ  
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ  
ΚΑΙ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

**Σχεδίαση και ανάπτυξη πλατφόρμας διαχείρισης  
προσωπικών αρχείων υγείας με χρήση του προτύπου HL7  
FHIR**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

**ΑΝΑΣΤΑΣΗ ΜΠΟΥΦΗ**

**Επιβλέπων :** Ιάκωβος Βενιέρης  
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την

.....  
Ιάκωβος Βενιέρης  
Καθηγητής Ε.Μ.Π.

.....  
Δήμητρα Κακλαμάνη  
Καθηγήτρια Ε.Μ.Π.

.....  
Αντώνιος Συμβώνης  
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2014

.....  
**Αναστάσης Μπούφης**

Διπλωματούχος Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών Ε.Μ.Π.

**Copyright, Αναστάσης Μπούφης, Παναγιώτης Καρμίρης, 2014**

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας εξ ολοκλήρου ή τμήματος αυτής για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευτεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

## Περίληψη

Η ανταλλαγή πληροφοριών στον τομέα της υγείας αποτελεί βασικό παράγοντα για τη γόνιμη εκτέλεση των ιατρικών ροών εργασίας. Ωστόσο, η ποικιλία των εμπλεκόμενων φορέων, καθώς και η ποικιλομορφία των κλινικών δεδομένων, μετατρέπει την ανταλλαγή αρχείων υγείας σε πρόκληση, η οποία δεν έχει ακόμη επιλυθεί πλήρως, παρά την πρόσφατη εξέλιξη στον τομέα των συστημάτων πληροφόρησης. Η έλλειψη επικοινωνίας μπορεί να οδηγήσει σε αναποτελεσματική θεραπευτική αγωγή και να αυξήσει το κόστος της υγειονομικής περίθαλψης, που αποτελεί σημαντική επιβάρυνση για τις εθνικές οικονομίες σε όλο τον κόσμο, ενώ η επίτευξη της διαλειτουργικότητας μεταξύ των πληροφοριακών συστημάτων υγειονομικής περίθαλψης καθίσταται πολύ ευεργετική.

Ο σκοπός αυτής της διπλωματικής εργασίας είναι να σχεδιάσει και να αναπτύξει μια τεχνολογική πλατφόρμα, η οποία να ενισχύει την αλληλεπίδραση μεταξύ των επαγγελματιών παρόχων υγείας και των ασθενών, παρέχοντάς τους με ένα διαδραστικό εργαλείο για την ασφαλή και αποτελεσματική ανταλλαγή πληροφοριών υγείας. Η πλατφόρμα αξιοποιεί τις διαδικτυακές υπηρεσίες REST και τις διαλειτουργικές δομές του προτύπου HL7 FHIR προκειμένου να υποστηρίξει τη συνδεσιμότητα μεταξύ των παρόχων υγειονομικής περίθαλψης, και να παρέχει το πλήρες ιατρικό ιστορικό των ασθενών, τόσο στο σημείο φροντίδας αλλά και στο σπίτι. Αυτό, επιτρέπει στους ασθενείς να παρακολουθούν καθημερινά την πορεία της υγείας τους, αλλά και ενδυναμώνει τους επαγγελματίες της υγείας να παρέχουν πιο εξατομικευμένες υπηρεσίες και να μειώνουν τα ιατρικά λάθη.

Η πλατφόρμα AidIT αποσκοπεί στην ενίσχυση της ποιότητας παροχής υπηρεσιών υγείας στην καθημερινή ζωή, π.χ., σε μια επίσκεψη σε ιατρό ή στο φαρμακείο για την αγορά συνταγογραφούμενων φαρμάκων. Επιτρέπει την ανταλλαγή του ιστορικού υγείας των ασθενών με τους παρόχους υγείας και τη δημιουργία και αποθήκευση ηλεκτρονικών διαγνωστικών εκθέσεων, συνταγών φαρμάκων και πλάνων φροντίδας.

**Λέξεις κλειδιά:** Προσωπικά αρχεία υγείας, HL7 FHIR, διαδικτυακές υπηρεσίες REST



# Abstract

Information exchange in healthcare is a key factor towards the fruitful execution of medical workflows. However, the variety of involved actors, as well as the diversity of clinical data, turn the healthcare information exchanging process into a challenging integration issue, which is still not completely resolved despite the recent evolution in the domain of information systems. This lack of information can lead to inefficient medical treatment and can raise healthcare costs which pose a considerable burden to national economies throughout the world, while achieving interoperability across the healthcare information systems can be significantly cost beneficial.

The purpose of this thesis is to design and develop an platform that enhances the interaction between health professionals and patients by providing them with an interactive tool for a secure and efficient health information exchange. The platform leverages the REST web service interface and the interoperable resources of the emerging healthcare standard HL7 FHIR in order to support the connectivity between healthcare providers as well as patients by making their full medical history available both at the point of care and at home. This enables patients to review and monitor their health, but also empowers health professionals to provide more personalized services, while reducing medical errors.

The AidIT platform aims at enhancing everyday life health service provision, e.g., a visit to a doctor or going to the pharmacy in order to purchase the prescribed drugs. It enables the sharing of patients' health history with health professionals and allows the definition and storage of electronic diagnostic reports, medication prescriptions and care plans.

**Keywords:** Personal Health Records, HL7 FHIR, REST web services





## Ευχαριστίες

Για την εκπόνηση της παρούσας διπλωματικής εργασίας, την ανάπτυξη της εφαρμογής, αλλά και γενικότερα για την ολοκλήρωση του κύκλου των σπουδών μου, νιώθω την ανάγκη να ευχαριστήσω όλους τους ανθρώπους που με βοήθησαν στο εγχείρημα αυτό.

Αρχικά, θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα της διπλωματικής εργασίας μου κ. Ιάκωβο Βενιέρη, Καθηγητή Ε.Μ.Π., ο οποίος μου έδωσε τη δυνατότητα να ασχοληθούμε με ένα αντικείμενο που πραγματικά επιθυμούσα. Τον ευχαριστώ ιδιαίτερα για τις πολύτιμες γνώσεις και συμβουλές που μου παρείχε κατά την εκπόνηση της εργασίας και καθ' όλη τη διάρκεια των σπουδών μου, αλλά κυρίως για την προσωπική ικανοποίηση που εισέπραξα με το πέρας της παρούσας διπλωματικής εργασίας και την αίσθηση πως συμπεριλαμβάνομαι πλέον κι εγώ στους εν δυνάμει προγραμματιστές.

Στη συνέχεια, θα ήθελα να ευχαριστήσω ξεχωριστά τα μέλη της επιτροπής, την κ. Δήμητρα-Θεοδώρα Κακλαμάνη, Καθηγήτρια Ε.Μ.Π., καθώς και τον κ. Αντώνιο Συμβώνη, Καθηγητή Ε.Μ.Π., για τη σημαντική συμβολή τους στην περαίωση της.

Ευχαριστίες θα ήθελα να απευθύνω στους υποψήφιους διδάκτορες Γιώργο Λαμπρινάκο & Ανδρέα Καψάλη, χωρίς τη βοήθεια των οποίων θα ήταν αδύνατη η ολοκλήρωση της διπλωματικής εργασίας, αλλά και για την τιμή την όποια μου έκαναν να αποτελέσει αυτή κομμάτι διεθνούς δημοσίευσης.

Ιδιαίτερες ευχαριστίες θα ήθελα να απευθύνω στον υποψήφιο διδάκτορα Αζίζ Μούσα καθώς και στον Σπυρίδων Μαντζουράτο, των οποίων η συμβολή και η παροχή γνώσεων στην ανάπτυξη της εφαρμογής και στη συγγραφή της εργασίας υπήρξε καθοριστική.

Ακόμη, θα ήθελα να ευχαριστήσω αρχικά τον καλό φίλο Αγάπιο που με βοήθησε να ξαναβρώ τον δρόμο μου και τη μητέρα μου Νίκη για την συμπαράσταση και την υπομονή της. Τέλος, θα ήθελα να αφιερώσω την εργασία αυτή στην μνήμη του πατέρα μου



# Πίνακας Περιεχομένων

|   |           |
|---|-----------|
| <b>Πίνακας Περιεχομένων .....</b>                       | <b>11</b> |
| <b>Εισαγωγή.....</b>                                    | <b>13</b> |
| 1.1. Κίνητρο διπλωματικής εργασίας.....                 | 13        |
| 1.2. Ιατρικά πληροφοριακά συστήματα .....               | 15        |
| 1.3. Προκλήσεις ιατρικών πληροφοριακών συστημάτων ..... | 15        |
| 1.4. Αντικείμενο διπλωματικής εργασίας.....             | 18        |
| 1.5. Διάρθρωση διπλωματικής εργασίας.....               | 19        |
| <b>Τεχνολογίες.....</b>                                 | <b>21</b> |
| 2.1. HL7 FHIR.....                                      | 21        |
| 2.2. Java.....  | 23        |
| 2.3. Eclipse .....                                      | 25        |
| 2.4. REST .....   | 26        |
| 2.5. JSON .....   | 27        |
| 2.6. Hibernate .....                                    | 27        |
| <b>Ανάλυση.....</b>                                     | <b>29</b> |
| 3.1. Απαιτήσεις εφαρμογής AidIT .....                   | 29        |
| 3.2. Αρχιτεκτονική της πλατφόρμας AidIT.....            | 33        |
| 3.3. Περιπτώσεις χρήσης.....                            | 34        |
| 3.3.1. UC-001 – Σύνδεση χρήστη.....                     | 35        |
| 3.3.2. UC-002 – Διαλογή ασθενών.....                    | 37        |
| 3.3.3. UC-003 – Κλινική εξέταση ασθενή.....             | 39        |
| 3.3.4. UC-004 – Εργαστηριακή εξέταση ασθενή .....       | 41        |
| 3.3.5. UC-005 – Διαγνωστική εξέταση ασθενή.....         | 43        |
| 3.3.6. UC-006 – Εκτέλεση ιατρικής συνταγής .....        | 46        |
| 3.3.7. UC-007 – Εκτέλεση θεραπευτικής αγωγής.....       | 47        |

|                          |   |           |
|--------------------------|---|-----------|
| 3.3.8.                   | UC-008 – Εξουσιοδότηση πρόσβασης στον ιατρικό φάκελο του ασθενή ..... | 49        |
| <b>Σχεδίαση.....</b>     |   | <b>53</b> |
| 4.1.                     | Η πλατφόρμα AidIT .....   | 53        |
| 4.1.1.                   | Λειτουργίες του εξυπηρετητή.....                                      | 53        |
| 4.1.2.                   | Περιγραφή κλάσεων.....  | 54        |
| 4.1.3.                   | Σχήμα βάσης δεδομένων .....   | 58        |
| <b>Υλοποίηση.....</b>    |   | <b>63</b> |
| 5.1.                     | Υλοποίηση της εφαρμογής.....  | 63        |
| 5.2.                     | Παράδειγμα χρήσης.....  | 69        |
| 5.2.1.                   | Αρχικά δεδομένα .....   | 69        |
| 5.2.2.                   | Εκτέλεση σεναρίου.....  | 71        |
| <b>Επίλογος .....</b>    |   | <b>81</b> |
| 6.1.                     | Συμπεράσματα.....   | 81        |
| 6.2.                     | Μελλοντικές επεκτάσεις.....   | 83        |
| <b>Βιβλιογραφία.....</b> |   | <b>87</b> |

# 1

## Εισαγωγή

Η ανταλλαγή πληροφοριών στον τομέα της υγείας αποτελεί βασικό παράγοντα για τη γόνιμη εκτέλεση των ιατρικών ροών εργασίας. Ωστόσο, η ποικιλία των εμπλεκόμενων φορέων, καθώς και η ποικιλομορφία των κλινικών δεδομένων, μετατρέπει την ανταλλαγή αρχείων υγείας σε πρόκληση, η οποία δεν έχει ακόμη επιλυθεί πλήρως, παρά την πρόσφατη εξέλιξη στον τομέα των συστημάτων πληροφόρησης. Η έλλειψη επικοινωνίας μπορεί να οδηγήσει σε αναποτελεσματική θεραπευτική αγωγή και να αυξήσει το κόστος της υγειονομικής περίθαλψης, που αποτελεί σημαντική επιβάρυνση για τις εθνικές οικονομίες σε όλο τον κόσμο, ενώ η επίτευξη της διαλειτουργικότητας μεταξύ των πληροφοριακών συστημάτων υγειονομικής περίθαλψης καθίσταται πολύ ευεργετική.

### 1.1. Κίνητρο διπλωματικής εργασίας

Ο τομέας της υγείας είναι ένας από τους τομείς με την μεγαλύτερη ανταλλαγή δεδομένων/πληροφοριών. Ανταλλαγές μεταξύ γιατρών, μεταξύ διοικητικών υπηρεσιών και μεταξύ παρόχων υγείας/ασφαλιστικών οργανισμών βρίσκονται στην ημερήσια διάταξη.

Είναι γεγονός πως ειδικά τον τελευταίο αιώνα, η Ιατρική Επιστήμη έχει εξελιχθεί δραματικά. Ασθένειες, οι οποίες απειλούσαν μεγάλο μέρος του πληθυσμού έχουν εξαφανιστεί ή βρίσκονται υπό έλεγχο, το προσδόκιμο ζωής έχει αυξηθεί και πλέον δεν μιλάμε για απλή θεραπεία των ασθενών, αλλά για βελτίωση της συνολικής ποιότητας της ζωής τους.

Το μεγάλο πρόβλημα, ή μάλλον καλύτερα η μεγάλη αντίφαση, είναι πως παρ' όλη την θεαματική ανάπτυξη της Ιατρικής Επιστήμης και Τεχνολογίας, και παρόλο τον άνευ προηγουμένου όγκο πληροφοριών που αυτές προσφέρουν, δεν υπάρχει η αντίστοιχη

ανάπτυξη συστημάτων, τα οποία να μπορούν να διαχειριστούν επιτυχώς τις πληροφορίες αυτές. Είναι αρκετά συνηθισμένο η ανταλλαγή ιατρικών δεδομένων να γίνεται εκτός πληροφοριακών συστημάτων (πχ μέσω τηλεφώνων, φαξ, εντύπου), με αποτέλεσμα μεγάλο μέρος της πληροφορίας να μην αποθηκεύεται ή να αποθηκεύεται λάθος. Έτσι η ανάκτησή της γίνεται αρκετά δύσκολη υπόθεση.

Είναι πάρα πολύ σημαντικό κάθε εργαζόμενος στον τομέα των υπηρεσιών υγείας να έχει άμεση πρόσβαση σε έγκυρη, έγκαιρη, ακριβή και σχετική με το αντικείμενό του πληροφορία. Μία νοσοκόμα χρειάζεται πληροφορίες για να μπορέσει να προσφέρει φαρμακευτική αγωγή σε έναν ασθενή σε κωματώδη κατάσταση, ένας γιατρός για να ενημερώσει τον ασθενή του σχετικά με τις νέες εξελίξεις/θεραπείες για την καταπολέμηση του καρκίνου, ένας διοικητικός υπάλληλος χρειάζεται πληροφορίες για να μπορέσει να συμπληρώσει αιτήσεις/φόρμες, ενώ τέλος ένας διευθυντής/manager χρειάζεται πληροφορίες για να μπορέσει να πείσει για την ανάγκη δημιουργίας μιας νέας Μονάδας Εντατικής Θεραπείας ή για την χάραξη μιας μακροπρόθεσμης στρατηγικής.

Από τα παραπάνω μπορούμε να καταλήξουμε σε κάποια συμπεράσματα:

- Πρώτον ότι η ραγδαία ανάπτυξη των επιστημών υγείας προσφέρει έναν μεγάλο όγκο δεδομένων/πληροφοριών.
- Δεύτερον, η ανάγκη για βελτίωση των υπηρεσιών υγείας και των παροχών σε συνδυασμό με την βούληση για μείωση του συνολικού κόστους δημιουργούν ανάγκη για υψηλής ποιότητας, ακριβή πληροφορία. Με άλλα λόγια η ιατρική πληροφορία αποτελεί έναν πολύ σημαντικό πόρο για τα σύγχρονα Συστήματα Υγείας και για αυτό τον λόγο πρέπει να μπορεί να διαχειρίζεται.
- Τρίτον, οι ιατρικές αποφάσεις στηρίζονται σε δεδομένα/πληροφορίες και γι αυτό είναι πολύ σημαντικό τα δεδομένα αυτά να είναι υψηλής ποιότητας.
- Τέταρτον, είναι αναγκαίο τα ιατρικά δεδομένα να βρίσκονται σε μία ενοποιημένη μορφή για αν μπορεί να γίνεται πιο εύκολη η επεξεργασία τους και η εξαγωγή ορθών συμπερασμάτων.

Από τα παραπάνω καταλαβαίνουμε την ανάγκη για την χρήση Πληροφοριακών Συστημάτων στον τομέα της υγείας. Τα συστήματα αυτά θα πρέπει να μπορούν να συλλέγουν τις ανεπεξέργαστες, από διαφορετικές πηγές πληροφορίες, να τις επεξεργάζονται και να τις φέρνουν σε μία ενιαία μορφή, να τις επικαιροποιήσουν και τέλος να τις προσφέρουν έγκαιρα σε όποιον επαγγελματία της υγείας τις χρειάζεται.

## 1.2. Ιατρικά πληροφοριακά συστήματα

Στο σημείο αυτό καλό είναι να δώσουμε και έναν ορισμό του τι ονομάζουμε Ιατρικό Πληροφοριακό Σύστημα [1]. Αρχικά θα πρέπει να τονίσουμε πως πρόκειται για μία ευρεία έννοια, η οποία περιλαμβάνει τόσο διοικητικά/διαχειριστικά όσο και κλινικά/ιατρικά συστήματα. Πληροφοριακό Σύστημα είναι μία διαχείριση πληροφοριών, διαδικασιών, ανθρώπων και πληροφοριών, τα οποία αλληλεπιδρούν, με στόχο την συλλογή, επεξεργασία, αποθήκευση και διανομή δεδομένων, με στόχο την υποστήριξη ενός Οργανισμού (στην προκειμένη περίπτωση ενός Οργανισμού Παροχής Υγείας).

Τα Διοικητικά/Διαχειριστικά συστήματα αφορούν κατά κύριο λόγο διαχειριστικά/οικονομικά δεδομένα. Η χρήση τους είναι για την υποστήριξη της Διοίκησης ενός Οργανισμού σε πολλά επίπεδα (οικονομικό, λογιστικό, προσωπικού, διαχείρισης υλικού, αναμονητικά συστήματα κλπ). Τα Ιατρικά/Κλινικά Συστήματα περιέχουν δεδομένα υγείας των ασθενών. Χρησιμοποιούνται από τους επαγγελματίες υγείας για την συλλογή και επικαιροποίηση πληροφοριών γύρω από την κατάσταση της υγείας του ασθενή, με σκοπό την ορθή λήψη ιατρικών αποφάσεων.

Η διπλωματική εργασία παρουσιάζει ένα Σύστημα το οποίο έχει τόσο διαχειριστική όσο και κλινική λειτουργία, όμως η ανάπτυξη του πραγματοποιήθηκε από την πλευρά του software developer. Για την δημιουργία ενός εμπορικού συστήματος, θα ήταν απαραίτητη η εμπειριστατωμένη μελέτη από έμπειρους επαγγελματίες υγείας, που θα ανέπτυσαν σε βάθος και με ακρίβεια τις απαιτήσεις της ιατρικής κοινότητας από ένα ιατρικό πληροφοριακό σύστημα.

## 1.3. Προκλήσεις ιατρικών πληροφοριακών συστημάτων

Ο τομέας της Υγείας είναι ένας από τους πιο ανεπτυγμένους τεχνολογικά στην κοινωνία μας. Παρόλα αυτά, αν κάποιος απευθύνει ερώτημα σε επαγγελματίες υγείας ή διευθυντές σε ασφαλιστικούς οργανισμούς αν έχουν πλήρεις, ακριβείς, αξιόπιστες και σχετικές πληροφορίες όταν παίρνουν σημαντικές αποφάσεις για την θεραπεία κάποιου ασθενή ή όταν χαράσσουν μία μακροπρόθεσμη στρατηγική, η απάντηση θα είναι μάλλον όχι. Παρόλη την ανάγκη για διοικητικές ή/και κλινικές πληροφορίες οι οποίες μπορούν να εξασφαλίσουν καλύτερου επιπέδου περίθαλψη με μικρότερο κόστος, οι οργανισμοί υγείας συνεχίζουν να χρησιμοποιούν πεπαλαιωμένα, μη επαρκή πληροφοριακά συστήματα. Υπάρχουν πολλοί

λόγοι γι αυτό και ένας απ' τους σημαντικότερους είναι η περιπλοκότητα των οργανισμών υγείας τόσο σαν σύνολο, όσο και στις επί μέρους λειτουργίες τους. Η περιπλοκότητα αυτή δημιουργεί αρκετά σημαντικές προκλήσεις για τους κατασκευαστές πληροφοριακών συστημάτων που απευθύνονται στον τομέα της υγείας.

Ο τομέας της υγείας συχνά κατηγορείται ότι μένει πίσω, σε σχέση με άλλους κλάδους, στην χρησιμοποίηση τεχνολογιών πληροφορίας. Οι στατιστικές δείχνουν πως ο τομέας της υγείας δαπανά το 2,7% των εσόδων του στην Πληροφορική. Σαν ποσοστό βρίσκεται στον μέσο όρο των υπόλοιπων βιομηχανιών, είναι όμως χαμηλό αν αναλογιστεί κανείς τον όγκο των πληροφοριών που διακινείται. Σαν σύγκριση, ο τραπεζικός τομέας δαπανά το 5,1 ενώ οι ασφαλιστικές το 4,1. Η πολυπλοκότητα και η δομή των οργανισμών παροχής υγείας, τόσο μεμονωμένα όσο και στο σύνολο, κάνουν αρκετά δύσκολη την ενσωμάτωση πληροφοριακών συστημάτων και πληροφοριακών τεχνολογιών εν γένει.

### **Μεγάλος αριθμός μικρών μονάδων υγείας**

Το σύστημα υγείας περιλαμβάνει έναν μεγάλο αριθμό μικρών μονάδων υγείας ή μεμονωμένων ελεύθερων επαγγελματιών. Τέτοιες μικρές επιχειρήσεις είναι δύσκολο να δαπανήσουν ένα σεβαστό ποσό χρημάτων που απαιτείται για την αγορά πληροφοριακών συστημάτων. Επιπλέον είναι δύσκολο να απασχολούν και το κατάλληλα εκπαιδευμένο προσωπικό που θα λειτουργεί τα συστήματα αυτά.

Πρόβλημα δημιουργείται και στους κατασκευαστές λογισμικού και hardware, καθώς είναι οικονομικά ασύμφορη η παροχή υλικού, υπηρεσιών και υποστήριξης σε τόσο μικρές επιχειρήσεις. Οι μεγάλοι κατασκευαστές το αποφεύγουν. Έτσι μένουν οι πιο μικροί, οι οποίοι όμως, λόγω μεγέθους, αντιμετωπίζουν κι αυτοί πρόβλημα βιωσιμότητας άρα και αξιοπιστίας.

### **Τμηματική Φροντίδα**

Οι περισσότεροι από εμάς θα αναζητήσουμε κατά την διάρκεια της ζωής μας, φροντίδα από διάφορους οργανισμούς. Τα ιατρικά μας δεδομένα όμως δεν διαμοιράζονται μεταξύ των οργανισμών αυτών. Τα αποτελέσματα αυτής της έλλειψης επικοινωνίας είναι πολλά: από το να χρειαστεί να επαναλάβουμε μία εργαστηριακή εξέταση έως το να μας δώσει ο γιατρός μας μία ακατάλληλη ιατρική συνταγή, μη έχοντας πληροφορηθεί εγκαίρως κάποια αλλεργία μας.

Στο παρελθόν, αυτή η έλλειψη επικοινωνίας μεταξύ των διάφορων ιατρικών υπηρεσιών δικαιολογείτο σε ένα βαθμό από την έλλειψη κοινών προτύπων. Σήμερα τα κοινά πρότυπα



υπάρχουν, παρόλα αυτά η έλλειψη επικοινωνίας παραμένει. Αυτό οφείλεται τόσο στην μη χρήση/έλλειψη τεχνικών μέσων (ας μην ξεχνάμε άλλωστε ότι πολλοί οργανισμοί παροχής υγείας διατηρούν ακόμη μεγάλο μέρος των αρχείων τους σε έντυπη μορφή), όσο και στην μη επιθυμία των διάφορων εμπλεκόμενων να επικοινωνήσουν, όσο οξύμωρο και αν ακούγεται αυτό.

Έχουν γίνει προσπάθειες για την δημιουργία συστημάτων ανταλλαγής δεδομένων, κυρίως μεταξύ εγκαταστάσεων και ιδρυμάτων που ανήκουν στον ίδιο οργανισμό. Οι προσπάθειες αυτές επιλύουν σε κάποιο βαθμό την έλλειψη επικοινωνίας. Το πρόβλημα εμφανίζεται αν κάποιος οργανισμός θέλει να συνθέσει μία πλήρη κλινική εικόνα ενός ασθενή. Στην περίπτωση αυτή θα πρέπει να αναζητήσει δεδομένα τα οποία βρίσκονται σε πολλαπλά σημεία, τυχαία καταναμημένα.

### **Πολυπλοκότητα των διαδικασιών υγείας**

Η πολυπλοκότητα στις διαδικασίες υγείας/θεραπείας/ιατρικές πράξεις οφείλεται σε τρεις διαφορετικούς παράγοντες. Πρώτον, στην δυσκολία του προσδιορισμού της καλύτερης δυνατής θεραπείας, δεύτερον στο μεγάλο πλήθος θεραπειών/ποικιλότητα στις θεραπευτικές διαδικασίες και τρίτον, στην συνεχή μεταβλητότητα/εξέλιξη των ιατρικών πρακτικών

Η ικανότητά μας να ορίσουμε την καλύτερη δυνατή θεραπεία είναι περιορισμένη και στηρίζεται στην εμπειρία, στα ειδικά χαρακτηριστικά του κάθε περιστατικού (πχ ηλικία, άλλες παθήσεις κλπ) και στις γνώσεις /εμπειρία/πεποιθήσεις του κάθε γιατρού ξεχωριστά (πχ ένας γιατρός να εμπιστεύεται πιο πολύ έναν συγκεκριμένο τύπο φαρμάκου ή ένα πρωτότυπο φάρμακο από το αντίστοιχο γενόσημο).

Ένας οργανισμός υγείας είναι απίθανο να υιοθετήσει μία τυπική/σπάντα προσέγγιση σε κάθε τύπο θεραπείας. Ακόμη και αν το κάνει, επιτρέπει κάποια περιθώρια και κάποια ελευθερία κινήσεων από την πλευρά του γιατρού. Για παράδειγμα, σε μία πανεπιστημιακή κλινική, ο γιατρός έχει στην διάθεσή του περί τα 2500 φάρμακα, 1100 εργαστηριακές εξετάσεις, 300 ραδιολογικές εξετάσεις και ένα πλήθος άλλων διαδικασιών. Η ποικιλία των θεραπειών έρχεται σε άμεση συνάρτηση και με την ποικιλία των ασθενειών. Υπάρχουν περίπου 10000 γνωστές ασθένειες, σύνδρομα και προβλήματα υγείας, καθένα από τα οποία απαιτεί την δική του προσέγγιση, κάποιες φορές δε κάποιο πρόβλημα μπορεί να έχει περισσότερες από μία προσεγγίσεις.

Τέλος η πολυπλοκότητα αυξάνεται ακόμα πιο πολύ αν συνυπολογίσουμε και την εξέλιξη/πρόοδο των ιατρικών πρακτικών. Κάθε χρόνο δημοσιεύονται περίπου 400000 άρθρα/εργασίες σε ιατρικά περιοδικά, άρθρα τα οποία κάνουν τους επιστήμονες να επαναπροσδιορίζουν συνεχώς την θέση τους γύρω από συγκεκριμένα ζητήματα. Επιπλέον η εξέλιξη της ιατρικής τεχνολογίας επιβάλλει αλλαγές πρακτικών.

### **Πολυπλοκότητα Ιατρικών Δεδομένων**

Τα ιατρικά δεδομένα είναι δύσκολο να κωδικοποιηθούν. Ακόμη και αν αυτό γίνει, είναι αμφίβολο το αν οι τελικοί χρήστες/επαγγελματίες υγείας θα επιλέγουν την εισαγωγή κωδικοποιημένων δεδομένων σε σχέση με απλό κείμενο. Τέλος δεν υπάρχει ένας μόνο τρόπος να οργανώσει κανείς αυτοματοποιημένα ιατρικά δεδομένα, το σχεσιακό μοντέλο δεν εξυπηρετεί ιδιαίτερα τον ιατρικό τομέα και για αυτό πολλοί οργανισμοί έχουν αναπτύξει πολλούς ιδιοσυγκρασιακούς τρόπους να κωδικοποιούν δεδομένα. Πολλές από αυτές τις προσπάθειες έχουν γίνει για καλό σκοπό ενώ έχουν γίνει και προσπάθειες για να χρηματοδοτηθεί δωρεών η εφαρμογή τους.

## **1.4. Αντικείμενο διπλωματικής εργασίας**

Το αντικείμενο της διπλωματικής εργασίας είναι η σχεδίαση και ανάπτυξη τεχνολογικής πλατφόρμας, η οποία να ενισχύει την αλληλεπίδραση μεταξύ των επαγγελματιών παρόχων υγείας και των ασθενών, παρέχοντάς τους με ένα διαδραστικό εργαλείο για την ασφαλή και αποτελεσματική ανταλλαγή πληροφοριών υγείας.

Η εφαρμογή AidIT αξιοποιεί τις διαδικτυακές υπηρεσίες REST και τις διαλειτουργικές δομές του προτύπου HL7 FHIR προκειμένου να υποστηρίξει τη συνδεσιμότητα μεταξύ των παρόχων υγειονομικής περίθαλψης, και να παρέχει το πλήρες ιατρικό ιστορικό των ασθενών, τόσο στο σημείο φροντίδας αλλά και στο σπίτι. Αυτό, επιτρέπει στους ασθενείς να παρακολουθούν καθημερινά την πορεία της υγείας τους, αλλά και ενδυναμώνει τους επαγγελματίες της υγείας να παρέχουν πιο εξατομικευμένες υπηρεσίες και να μειώνουν τα ιατρικά λάθη.

Η πλατφόρμα AidIT αποσκοπεί στην ενίσχυση της ποιότητας παροχής υπηρεσιών υγείας στην καθημερινή ζωή, π.χ., σε μια επίσκεψη σε ιατρό ή στο φαρμακείο για την αγορά συνταγογραφούμενων φαρμάκων. Επιτρέπει την ανταλλαγή του ιστορικού υγείας των

ασθενών με τους παρόχους υγείας και τη δημιουργία και αποθήκευση ηλεκτρονικών διαγνωστικών εκθέσεων, συνταγών φαρμάκων και πλάνων φροντίδας.

## **1.5. Διάρθρωση διπλωματικής εργασίας**

Τα βήματα τα οποία ακολουθήθηκαν στα πλαίσια της διπλωματικής εργασίας για την ανάπτυξη της πλατφόρμας AidIT, αντικατοπτρίζονται στα κεφάλαια της διπλωματικής εργασίας. Στο δεύτερο κεφάλαιο, παρουσιάζονται οι τεχνολογίες που επιλέχθηκαν για την υλοποίηση της εφαρμογής, περιγράφοντας τα στοιχεία τους και τα πλεονεκτήματά τους. Το τρίτο κεφάλαιο, αποτελεί την ανάλυση των απαιτήσεων που πρέπει να ικανοποιεί η εφαρμογή καθώς και η περιγραφή των βασικών λειτουργιών και σεναρίων χρήσης. Στη συνέχεια, το τέταρτο κεφάλαιο αναφέρεται στη σχεδίαση της εφαρμογής και του εξυπηρετητή της, παραθέτοντας περιγραφές των βασικών κλάσεων τους. Έπειτα, το πέμπτο κεφάλαιο αφορά την υλοποίηση της εφαρμογής και παρουσιάζεται ένα πλήρες σενάριο χρήσης της. Τέλος, στο έκτο κεφάλαιο, καταγράφονται τα συμπεράσματα που προέκυψαν κατά την περάτωση της διπλωματικής εργασίας και προτείνονται πιθανές μελλοντικές επεκτάσεις.



# 2

## Τεχνολογίες

Το παρόν κεφάλαιο παρουσιάζει επιγραμματικά τις τεχνολογίες οι οποίες επιλέχθηκαν για στις οποίες βασίζεται η εφαρμογή. Οι τεχνολογίες αυτές αφορούν τόσο στο περιβάλλον της κινητής συσκευής αλλά και στο περιβάλλον του εξυπηρετητή.

### 2.1. HL7 FHIR

Ο οργανισμός Health Level 7 (HL7) αποτελεί μια διεθνή κοινότητα εμπειρογνομόνων οι οποίοι δραστηριοποιούνται στην περιοχή της προτυποποίησης των συναλλαγών, της διαχείρισης και διαλειτουργικότητας μεταξύ ιατρικών πληροφοριακών συστημάτων, και έχει παράξει σειρά προτύπων με κυριότερα εξ αυτών τα HL7 v2.x, 3.x και το FHIR.

Το πρότυπο HL7 v2.x, η πρώτη έκδοση του οποίου χρονολογείται το 1989, έχει υιοθετηθεί ευρέως από εθνικούς αλλά και ιδιωτικούς οργανισμούς που δραστηριοποιούνται στο χώρο της υγείας (ασφαλιστικοί φορείς, ιδιωτικές κλινικές κτλ.). Ωστόσο η ευελιξία των μηνυμάτων τα οποία ορίζει οδηγεί σε παρερμηνείες των δομών τους, με αποτέλεσμα πολλές φορές η επικοινωνία μεταξύ δύο συστημάτων τα οποία στηρίζονται στο πρότυπο HL7 v2.x να απαιτεί σημαντική προσπάθεια.

Το πρότυπο HL7 v3.x, η πρώτη έκδοση του οποίου χρονολογείται το 2005, προσπαθεί να λύσει το παραπάνω πρόβλημα ορίζοντας πιο αυστηρές δομές βασισμένες στη γλώσσα XML και το Reference Information Model [\cite{hl7\\_rim}](#) περιγράφει τα δεδομένα που χρειάζονται

σε κλινικά αλλά και διαχειριστικά περιβάλλοντα. Ωστόσο, η πολυπλοκότητα του RIM αποτελεί σημαντική τροχοπέδη στην υιοθέτηση του προτύπου από τα ιατρικά πληροφοριακά συστήματα.

Το πρότυπο HL7 FHIR (Fast Healthcare Interoperability Resources - [5]), η πρώτη έκδοση του οποίου χρονολογείται το 2012, προσεγγίζει τα προβλήματα διαλειτουργικότητας μεταξύ ιατρικών πληροφοριακών συστημάτων με διαφορετικό τρόπο, σκοπεύοντας στην γρήγορη ενσωμάτωση του προτύπου. Αρχικά, ορίζει ευέλικτες δομές για την περιγραφή των ιατρικών δεδομένων και προωθεί τη χρήση σύγχρονων τεχνολογιών επικοινωνίας μεταξύ πληροφοριακών συστημάτων ορίζοντας διεπαφή διαδικτυακών υπηρεσιών REST. Οι προσπάθειες προτυποποίησης στο πλαίσιο του HL7 FHIR στοχεύουν στην κάλυψη των πιο κοινών διαδικασιών, ενώ οι πιο πολύπλοκες μπορούν να καλυφθούν με επέκταση των δομών του, κάτι που διευκολύνεται με τη χρήση οντολογικών μοντέλων για την αναπαράσταση λεπτομερών ιατρικών δεδομένων.

Το παρακάτω παράδειγμα παρουσιάζει μια ιατρική συνταγή φαρμάκων με το στοιχείο MedicationPrescription, ενώ το πρότυπο ορίζει δομές για όλες τις οντότητες που δραστηριοποιούνται σε υγειονομικά σενάρια, όπως ασθενής, προμηθευτής υγείας, οργανισμός αλλά και πολλές κλινικές διαδικασίες, όπως ιατρικές εξετάσεις, φαρμακευτική αγωγή, περίθαλψη κτλ. Η σχεδίαση ωστόσο των παραπάνω, οδηγείται από την απαίτηση για αυτοτελείς, συμπαγείς, ανεξάρτητα διαχειρίσιμες και συνδυάσιμες δομές, ώστε να είναι σύμφωνες με την αρχιτεκτονική διαδικτυακών υπηρεσιών REST που ορίζει, η οποία διευκολύνει την επικοινωνία μεταξύ πληροφοριακών συστημάτων με τρόπο ταιριαστό με τις σύγχρονες τάσεις ανάπτυξης εφαρμογών.

```
<MedicationPrescription xmlns="http://hl7.org/fhir">
  <text> Theophylline 200mg twice a day </text>
  <Medication id="med1">
    <name value="Theophylline 200mg"/>
    <system value="http://snomed.org"/>
    <code value="66493003"/>
  </Medication>
<patient>
  <reference value="Patient/1028"/>
  <display value="Peter Patient"/>
</patient>
```

```

<prescriber>
  <reference value="Practitioner/example"/>
  <display value="Peter Practitioner"/>
</prescriber>
<reasonResource>
  <reference value="Condition/f201"/>
  <display value="fever"/>
</reasonResource>
<medication>
  <reference value="Medication/example"/>
  <display value="Theophylline 200mg BD"/>
</medication>
<dosageInstruction>
  <text value="Take with Food"/>
  <timingSchedule>
    <repeat>
      <frequency value="2"/>
      <duration value="1"/>
      <units value="d"/>
    </repeat>
  </timingSchedule>
  <route>
    <system value="http://snomed.info/sct"/>
    <code value="394899003"/>
    <display value="oral administration of treatment"/>
  </route>
  <doseQuantity>
    <value value="1"/>
    <units value="tablet"/>
  </doseQuantity>
</dosageInstruction>
</MedicationPrescription>

```

## 2.2. Java

Η Java είναι μία γλώσσα αντικειμενοστραφούς προγραμματισμού που σχεδιάστηκε από την εταιρεία Sun Microsystems σαν μέρος ενός ερευνητικού έργου ανάπτυξης λογισμικού για ηλεκτρονικές συσκευές καταναλωτικού επιπέδου. Ο τρόπος αυτός ανάπτυξής της τη μετέτρεψε σε μία ιδανική γλώσσα για τη διανομή εκτελέσιμων προγραμμάτων μέσω του παγκόσμιου ιστού καθώς επίσης σε μία γενικού σκοπού γλώσσα προγραμματισμού για την

ανάπτυξη προγραμμάτων που θα μπορούν εύκολα να μεταφέρονται σε διαφορετικά λειτουργικά συστήματα.

Οι δημιουργοί της Java την έχουν χαρακτηρίσει ως μία γλώσσα “απλή, αντικειμενοστραφή, διαμοιραζόμενη, εύρωστη, ασφαλή, με ουδέτερη αρχιτεκτονική, εύκολη στη μεταφορά, υψηλής απόδοσης και πολυνηματική”.

Ένα από τα βασικά χαρακτηριστικά της Java έναντι των περισσότερων άλλων γλωσσών είναι η ανεξαρτησία του λειτουργικού συστήματος της πλατφόρμας. Τα προγράμματα που είναι γραμμένα σε Java τρέχουν ακριβώς το ίδιο σε Windows, Linux, Unix, Macintosh, χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση (compiling) ή να αλλάξει ο πηγαίος κώδικας για κάθε λειτουργικό σύστημα. Για να επιτευχθεί αυτό χρειαζόταν κάποιος τρόπος έτσι ώστε τα προγράμματα γραμμένα σε Java να μπορούν να είναι «κατανοητά» από κάθε υπολογιστή ανεξάρτητα από το είδος του επεξεργαστή και του λειτουργικού συστήματος. Η λύση δόθηκε με την ανάπτυξη της εικονικής μηχανής.

### **Java Virtual Machine**

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

### **Garbage Collector**

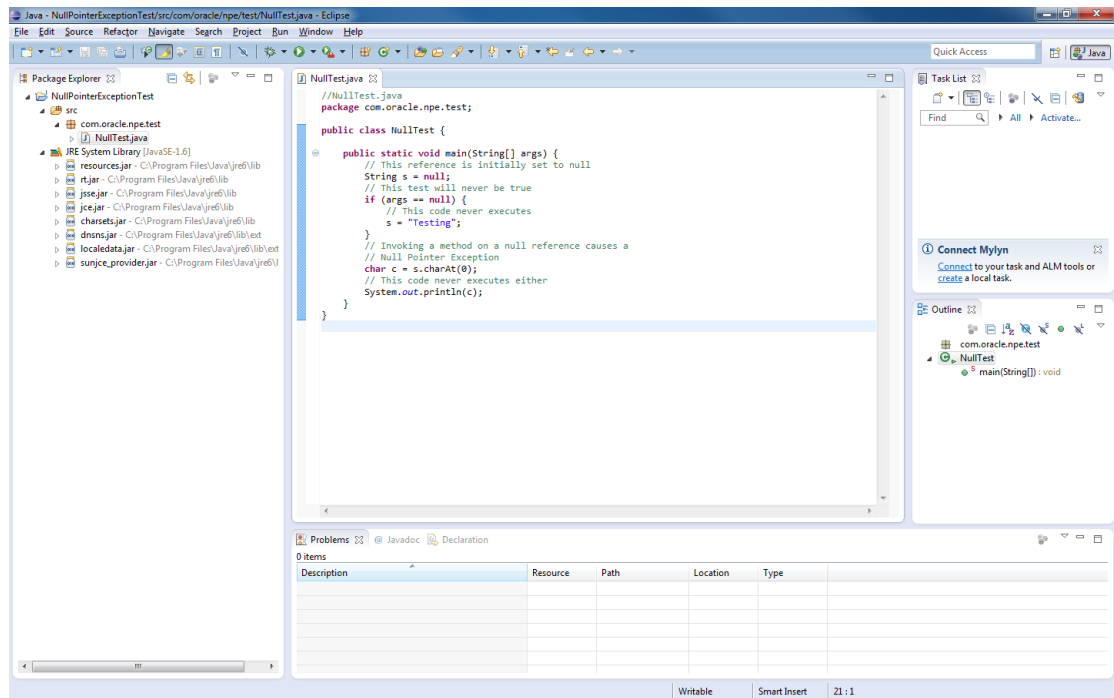
Ακόμα μία ιδέα που βρίσκεται πίσω από τη Java είναι η ύπαρξη του συλλέκτη απορριμμάτων (Garbage Collector). Συλλογή απορριμμάτων είναι μία κοινή ονομασία που χρησιμοποιείται στον τομέα της πληροφορικής για να δηλώσει την ελευθέρωση τμημάτων μνήμης από δεδομένα που δε χρειάζονται και δε χρησιμοποιούνται άλλο. Αυτή η



απελευθέρωση μνήμης στη Java είναι αυτόματη και γίνεται μέσω του συλλέκτη απορριμμάτων. Υπεύθυνη για αυτό είναι και πάλι η εικονική μηχανή η οποία μόλις «καταλάβει» ότι ο σωρός (heap) της μνήμης (στη Java η συντριπτική πλειοψηφία των αντικειμένων αποθηκεύονται στο σωρό) κοντεύει να γεμίσει ενεργοποιεί το συλλέκτη απορριμμάτων. Έτσι ο προγραμματιστής δε χρειάζεται να ανησυχεί για το πότε και αν θα ελευθερώσει ένα συγκεκριμένο τμήμα της μνήμης, ούτε και για σφάλματα δεικτών. Αυτό είναι ιδιαίτερα σημαντικό γιατί είναι πολύ συχνά τα σφάλματα προγραμμάτων που οφείλονται σε λανθασμένο χειρισμό της μνήμης.

## **2.3. Eclipse**

Το Eclipse είναι ένα περιβάλλον προγραμματισμού συμβατό με πολλές γλώσσες προγραμματισμού. Είναι γραμμένο κυρίως σε Java και μπορεί να χρησιμοποιηθεί για την ανάπτυξη εφαρμογών σε Java αλλά και σε άλλες γλώσσες μέσω διαφόρων plug-ins. Το Eclipse SDK έχει γραφεί από προγραμματιστές Java και προορίζεται για αυτή τη γλώσσα προγραμματισμού. Είναι φυσικά ένα πολύ χρήσιμο εργαλείο, γι' αυτό και είναι τόσο διαδεδομένη η χρήση του. Ακόμα και αν η κύρια γλώσσα προγραμματισμού που χρησιμοποιεί κάποιος δεν είναι η Java, δεν είναι λίγες οι περιπτώσεις που το Eclipse μπορεί να «λύσει» τα χέρια του προγραμματιστή.



## 2.4. REST

Η πληροφοριοκεντρική αρχιτεκτονική REST (REpresentational State Transfer - REST - [2]) έχει τύχει τα τελευταία χρόνια ευρείας αποδοχής λόγω της απλότητας και ευελιξίας την οποία προσφέρει στην ανάπτυξη μιας διαδικτυακής υπηρεσίας. Οι βασικές σχεδιαστικές αρχές μιας υπηρεσίας REST εντοπίζονται στην διευθυνσιοδότηση, την ομοιόμορφη διεπαφή και την έλλειψη καταγραφής κατάστασης. Κάθε ουσιαστικό τμήμα πληροφορίας λαμβάνει μοναδικό αναγνωριστικό και διεύθυνση πρόσβασης. Παράλληλα, ανεξάρτητα από το διαφορετικό μέρος πληροφορίας και των λειτουργιών που εκτίθεται μέσω μιας διεπαφής REST, η διεύθυνση πρόσβασης ακολουθεί συνήθως συγκεκριμένο πρότυπα οδηγώντας σε οικίες και ομοιόμορφες διεπαφές για την υπηρεσία, βασισμένες στις μεθόδους του πρωτοκόλλου HTTP (GET/POST/PUT/DELETE). Επιπρόσθετα, κάθε κλήση προς τη διεπαφή είναι ανεξάρτητη από τις προηγούμενες και έτσι δεν επιβάλλει την καταγραφή κατάστασης από τον εξυπηρετητή. Τα παραπάνω, καθιστούν την αρχιτεκτονική διαδικτυακών υπηρεσιών REST ιδανική για μεγάλης κλίμακας εφαρμογές οι οποίες αναπτύσσονται σε πολλαπλούς τομείς, αφού πετυχαίνει ενίσχυση της διαλειτουργικότητας, μείωση του κόστους ενσωμάτωσης σε υπάρχουσες εφαρμογές και ακόμα πιο χαλαρή σύζευξη τόσο μεταξύ άλλων υπηρεσιών αλλά και από τα συνεχώς εξελισσόμενα τεχνολογικά πρότυπα.

## 2.5. JSON

Το JSON (Javascript Object Notation) είναι μια τεχνολογία ανταλλαγής δεδομένων που στηρίζεται στην σειριοποίησή τους για εύκολη ανάγνωση από τον άνθρωπο. Προέρχεται από τη γλώσσα Javascript για την αναπαράσταση απλών δεδομένων και συσχετισμένων πινάκων, που ονομάζονται αντικείμενα. Παρά τη σχέση του με τη Javascript, είναι ανεξάρτητη γλώσσα με προγράμματα ανάλυσης που διατίθενται σε πολλές γλώσσες προγραμματισμού. Το JSON χρησιμοποιείται συχνά για σειριοποίηση και διαβίβαση δεδομένων πάνω από μία σύνδεση δικτύου. Κατά κύριο λόγο χρησιμοποιείται για τη μετάδοση δεδομένων μεταξύ ενός server και μίας web εφαρμογής την οποία μπορεί να εξυπηρετήσει και λειτουργεί σαν εναλλακτική λύση στη θέση της XML.

Το JSON είναι χτισμένο πάνω σε δύο δομές:

- Ένα σύνολο από ζεύγη ονόματος-τιμής. Στις διάφορες γλώσσες αυτό μεταφράζεται ως αντικείμενο, record, struct, dictionary, πίνακας ή λίστα.
- Μία ταξινομημένη λίστα τιμών. Στις περισσότερες γλώσσες προγραμματισμού αυτό γίνεται αντιληπτό σαν πίνακας ή λίστα.

Αυτές είναι ευρέως αναγνωρίσιμες δομές δεδομένων. Όλες οι σύγχρονες γλώσσες προγραμματισμού μπορούν να τις υποστηρίξουν με τον ένα τρόπο ή τον άλλον. Αυτό φυσικά είναι και το μεγάλο πλεονέκτημα του JSON. Το γεγονός δηλαδή ότι μέσω του JSON τα δεδομένα μπορούν να πάρουν μία μορφή που είναι αναγνώσιμη από πολλές διαφορετικές γλώσσες μπορεί να λύσει σημαντικά προβλήματα που αφορούν στη μεταφορά δεδομένων.

## 2.6. Hibernate

Το Hibernate Framework είναι λογισμικό ανοιχτού κώδικα που σκοπό έχει να συνδέσει τα αντικείμενα που δημιουργούνται στη Java με τους πίνακες μιας σχεσιακής βάσης δεδομένων. Η σύνδεση αυτή επιτυγχάνεται με την χρήση επιπρόσθετης πληροφορίας (metadata) που τοποθετείται κατάλληλα (μαζί με τον κώδικα Java ή σε ξεχωριστά xml αρχεία) και περιγράφει την αντιστοιχία μεταξύ των αντικειμένων και της βάσης δεδομένων. Γενικά το Hibernate προσφέρει την αυτόματη μετατροπή της μιας μορφής (αντικείμενα) στην άλλη (σχεσιακή βάση δεδομένων).

Το Hibernate συγκαταλέγεται στην κατηγορία του λογισμικού Object Relational Mapping. Το λογισμικό ORM στοχεύει στην δημιουργία μιας διεπαφής (interface) μεταξύ των διαδεδωμένων σχεσιακών βάσεων δεδομένων και του αντικειμενοστραφούς προγραμματισμού. Με απλά λόγια, προσφέρει την χρησιμοποίηση μιας σχεσιακής βάσης δεδομένων σαν να ήταν αντικειμενοστραφής. Για να το επιτύχει αυτό δημιουργεί αντιστοιχίες

μεταξύ των εννοιών του αντικειμενοστραφούς προγραμματισμού (συσχετίσεις, κληρονομικότητα, πολυμορφισμός), που δεν υπάρχουν σε μια σχεσιακή βάση δεδομένων, και των πινάκων και σχέσεων μεταξύ των πινάκων μιας σχεσιακής βάσης. Με αυτό τον τρόπο ο προγραμματιστής βλέπει τελικά μια αντικειμενοστραφή βάση δεδομένων, παρόλο που στην ουσία χρησιμοποιεί μια σχεσιακή. Έτσι, ο προγραμματιστής χρησιμοποιεί τα αντικείμενα της συγκεκριμένης εφαρμογής, τα τροποποιεί σχετικά με τη λογική της εφαρμογής που αναπτύσσει και τα αποθηκεύει (τροποποιεί, διαγράφει και αναζητά) στην βάση ως αντικείμενα, σκεπτόμενος δηλαδή με αντικειμενοστραφείς έννοιες και όχι με βάση το σχήμα της σχεσιακής βάσης δεδομένων. Σε αυτό το σημείο είναι το Hibernate, που γνωρίζοντας την αντιστοιχία μεταξύ βάσης και λογικής της εφαρμογής, αναλαμβάνει να κατασκευάσει την κατάλληλη εντολή της SQL η οποία και στέλνεται τελικά στην βάση δεδομένων. Έπειτα, τα αποτελέσματα που επιστρέφει η βάση το Hibernate τα επιστρέφει στον προγραμματιστή ως αντικείμενα της εφαρμογής. Είναι δηλαδή ένα ενδιάμεσο επίπεδο μεταξύ της εφαρμογής και της βάσης δεδομένων.

# 3

## Ανάλυση

Στο κεφάλαιο αυτό θα αναλυθούν σενάρια που προκύπτουν σε περιβάλλοντα παροχής υπηρεσιών υγείας και πιο συγκεκριμένα σε ένα νοσοκομείο (σε αρκετά απλουστευμένη εκδοχή), καθώς και τους ρόλους που εμφανίζονται σε αυτά. Στην συνέχεια, θα περιγράψουν σε βάθος οι λειτουργίες που απαιτούνται για κάθε ρόλο ξεχωριστά, οι οποίες θα καθορίσουν τελικά τον τρόπο με τον οποίο πρέπει να σχεδιαστεί και να υλοποιηθεί η πλατφόρμα AidIT.

### 3.1. Απαιτήσεις εφαρμογής AidIT

Ακολουθεί ένα ενδεικτικό σενάριο σύμφωνα το οποίο παρουσιάζει την καθημερινή λειτουργία ενός νοσοκομείου, από την πλευρά του ασθενή.:

Ένας Ασθενής επισκέπτεται το Νοσοκομείο. Το πρώτο πράγμα που θα πρέπει να γίνει είναι ο ασθενής να δώσει τα στοιχεία του στο Ιατρείο Διαλογής του Νοσοκομείου και να ταυτοποιηθεί /ανασυρθεί ο ιατρικός του φάκελος. Το ιατρείο διαλογής θα πρέπει στη συνέχεια, αφού ταυτοποιήσει τον ασθενή, να κάνει και την διαλογή/triage, να στείλει δηλαδή τον ασθενή στο κατάλληλο ιατρικό τμήμα, ανάλογα με τα συμπτώματά του. Γι αυτόν ακριβώς τον λόγο θεωρούμε ότι θα πρέπει να έχει στο δυναμικό της και κάποιον έμπειρο γιατρό, ο οποίος θα κάνει την αρχική αυτή εκτίμηση. Για την ορθότερη αρχική εκτίμηση, θεωρούμε πως θα πρέπει να έχει πρόσβαση και στο γενικό ιατρικό ιστορικό του Ασθενούς.

Στην συνέχεια, ο Ασθενής θα πρέπει να δώσει εξουσιοδότηση/συγκατάθεση για την πρόσβαση στον ιατρικό του φάκελο. Αυτό είναι ένα αρκετά ευαίσθητο σημείο και αφορά

στην λειτουργία και δομή του Συστήματος Υγείας. Το Νοσοκομείο θα πρέπει να θεωρείται ως Οργανισμός, ο οποίος παρέχει περίθαλψη με βάση συγκεκριμένα, επιστημονικά αποδεκτά πρότυπα και μεθόδους. Κάθε γιατρός, νοσοκόμος, παραϊατρικό προσωπικό κλπ, θεωρείται υπάλληλος του Οργανισμού αυτού και θα πρέπει να ασκεί το λειτούργημά του με βάση τις οδηγίες και τις κατευθυντήριες γραμμές του Νοσοκομείου. Το Νοσοκομείο είναι υπεύθυνο για την σωστή παροχή φροντίδας και ιατρικών υπηρεσιών από τους υπαλλήλους του, όπως επίσης και για τον πειθαρχικό τους έλεγχο σε περίπτωση παραπτώματων ή αμέλειας.

Για τους παραπάνω λόγους η εξουσιοδότηση θα πρέπει να δίνεται από τον Ασθενή μόνο μία φορά (στην αρχική επίσκεψη στο Ιατρείο Διαλογής) και στην συνέχεια το Νοσοκομείο ρυθμίζει την πρόσβαση ανάλογα με τα ιατρικά δεδομένα/πληροφορίες που χρειάζεται ο κάθε ρόλος.

Στην συνέχεια ο Ασθενής επισκέπτεται τον Θεράποντα Ιατρό που του υποδείχθηκε. Ο γιατρός αυτός θα πρέπει να έχει πλήρη πρόσβαση στον ιατρικό φάκελο του ασθενούς. Θα πρέπει να μπορεί να δει τα γενικά στοιχεία/demographics, να έχει πρόσβαση στο γενικό ιατρικό ιστορικό του Ασθενούς, όπως επίσης και στα ιστορικά εξετάσεων και φαρμάκων. Θα πρέπει τέλος να μπορεί να δει το ιστορικό περίθαλψης/φροντίδας του Ασθενή.

Από κει και πέρα ο γιατρός, με βάση το ιστορικό και την κλινική εξέταση, έχει τρεις επιλογές:

- α) Να κάνει διάγνωση
- β) Να παραπέμψει τον ασθενή για εργαστηριακό έλεγχο
- γ) Να δηλώσει αναρμόδιος και να παραπέμψει τον Ασθενή σε άλλο Τμήμα

Στην πρώτη περίπτωση ο γιατρός σημειώνει την διάγνωση στο γενικό ιστορικό και καθορίζει Φαρμακευτική Αγωγή, γράφει δηλαδή μία ιατρική συνταγή με παραλήπτη τον Φαρμακοποιό του Νοσοκομείου. Επίσης καθορίζει και ένα Πλάνο Φροντίδας.

Στην δεύτερη περίπτωση ο γιατρός παραπέμπει τον ασθενή για κάποια Εργαστηριακή Εξέταση. Ο γιατρός δηλαδή γράφει ένα παραπεμπτικό με αποδέκτη κάποιο από τα Εργαστήρια του Νοσοκομείου. Μετά το πέρας των εξετάσεων, ο Ασθενής επιστρέφει στον γιατρό, ο οποίος πλέον μπορεί να κάνει διάγνωση και έτσι επιστρέφουμε πάλι στην πρώτη περίπτωση.

Στην τρίτη περίπτωση, ο γιατρός στέλνει τον ασθενή σε κάποιο άλλο, πιο κατάλληλο Τμήμα

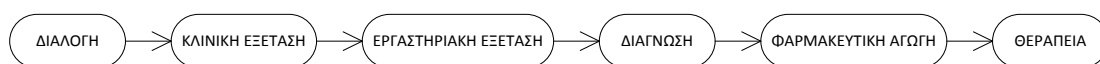
Επόμενο βήμα είναι η επίσκεψη του Ασθενούς στο Εργαστήριο. Ο υπεύθυνος του Εργαστηρίου βλέπει το παραπεμπτικό του γιατρού και κάνει την ζητούμενη Εξέταση. Μέσω του παραπεμπτικού, ο υπεύθυνος του εργαστηρίου έχει πρόσβαση στο εργαστηριακό ιστορικό του Ασθενούς, όπου και σημειώνει τα αποτελέσματα της Εξέτασης. Μπορεί επίσης να δει αποτελέσματα προηγούμενων εξετάσεων, να κάνει συγκρίσεις κλπ.

Επόμενο βήμα η επίσκεψη στον Φαρμακοποιό του Νοσοκομείου. Ο Φαρμακοποιός, λόγω της συνταγής έχει πρόσβαση στο φαρμακευτικό ιστορικό του Ασθενούς.

Ο Φαρμακοποιός μπορεί να:

α) εκτελέσει την συνταγή. Στην περίπτωση αυτή σημειώνει στο ιστορικό ότι η συνταγή εκτελέστηκε.

β) μην εκτελέσει την συνταγή για διάφορους λόγους ( να μην έχει το συγκεκριμένο φάρμακο, να διαπιστώσει από το ιστορικό αντενδείξεις με κάποιο άλλο σκεύασμα κλπ). Στην περίπτωση αυτή σημειώνει στο ιστορικό ότι η συνταγή δεν εκτελέστηκε.



Η περιγραφείσα ροή εργασιών απεικονίζεται στο παραπάνω σχήμα, ενώ συμπερασματικά στο σενάριο εμφανίζονται οι εξής ρόλοι: Ασθενής, Ιατρός διαλογής, Θεράπων ιατρός, Ιατρός εργαστηριακής ειδικότητας, Φαρμακοποιός

### **Ασθενής**

Πρέπει να μπορεί να δει τα προσωπικά του δεδομένα/ demographics, το ιατρικό, το φαρμακευτικό και το εργαστηριακό του ιστορικό χωρίς να μπορεί να επέμβει. Πρέπει να μπορεί να δει το Πλάνο Φροντίδας και να σημειώνει αν το εκτελεί κανονικά. Τέλος πρέπει να δίνει εξουσιοδότηση στον Ιατρό Διαλογής, ώστε μέσω αυτής να μπορεί το Νοσοκομείο να έχει πρόσβαση στον ιατρικό του φάκελο.

### **Ιατρός Διαλογής**

Πρέπει να μπορεί να κάνει ταυτοποίηση του Ασθενή άρα πρέπει να έχει πρόσβαση στα προσωπικά δεδομένα/demographics. Επίσης πρέπει να έχει πρόσβαση στο ιατρικό ιστορικό του ασθενή ώστε να μπορεί να κάνει διαλογή/triage, να σημειώσει μία εκτίμηση και να παραπέμψει τον Ασθενή στο κατάλληλο Τμήμα/Ιατρό. Πρέπει να δέχεται την εξουσιοδότηση πρόσβασης στα ιατρικά δεδομένα από τον Ασθενή, για λογαριασμό του Νοσοκομείου και την διαμοιράζει στα Τμήματα του Νοσοκομείου που χρειάζεται.

Αρχικά πρέπει να μπορεί να δει το εισιτήριο εισόδου. Πρέπει να έχει πρόσβαση σε όλον τον ιατρικό φάκελο του Ασθενούς. Πρέπει να μπορεί να δει:

- α) τα προσωπικά δεδομένα/demographics,
- β) το ιατρικό ιστορικό και να το τροποποιήσει γράφοντας διάγνωση
- γ) το εργαστηριακό ιστορικό και να προσθέσει παραπεμπτικό για εξετάσεις και
- δ) το φαρμακευτικό ιστορικό και να προσθέσει ιατρική συνταγή.

Πρέπει ακόμα να μπορεί να δει τα υπάρχοντα πλάνα φροντίδας και να προσθέσει κάποιο ακόμη. Τέλος μπορεί να προωθήσει την πρόσβασή του σε κάποιον άλλο Ιατρό, αν κρίνει ότι είναι αναρμόδιος.

#### **Ιατρός εργαστηριακής ειδικότητας**

Πρέπει να μπορεί να δει αν υπάρχει παραπεμπτικό για εξετάσεις από γιατρό του νοσοκομείου. Αν υπάρχει, πρέπει να μπορεί να δει το εργαστηριακό ιστορικό και να το τροποποιήσει προσθέτοντας τα αποτελέσματα των εξετάσεων. Η πρόσβαση παρέχεται από τον Ιατρό μέσω του παραπεμπτικού.

#### **Φαρμακοποιός**

Πρέπει να μπορεί να δει αν υπάρχει ενεργή συνταγή για φάρμακα. Αν υπάρχει, μπορεί να έχει πρόσβαση στο φαρμακευτικό ιστορικό του ασθενούς. Μπορεί να τροποποιήσει το φαρμακευτικό ιστορικό, είτε με το να εκτελέσει την συνταγή, είτε με το να την αφήσει ανεκτέλεστη σημειώνοντας τους λόγους. Η πρόσβασή του προέρχεται από τον Ιατρό μέσω της φαρμακευτικής συνταγής

Η παραπάνω περιγραφή αφορά ένα απλουστευμένο πρότυπο λειτουργίας ενός Νοσοκομείου, το οποίο όμως είναι επαρκές για την επίδειξη των δυνατοτήτων του πρωτοκόλλου HL7 FHIR. Σαφώς υπάρχουν αρκετά σημαντικά τμήματα και υπηρεσίες μιας σύγχρονης Μονάδας Υγείας ( θάλαμοι νοσηλείας, αναμονητικά συστήματα, κοστολόγηση/λογιστήριο κλπ), τα οποία δεν



εξετάζονται, ωστόσο μια περαιτέρω επέκταση της εφαρμογής θα απομάκρυνε από τον κύριο στόχο της διπλωματική εργασίας και θα ήταν χρήσιμη μόνο στην περίπτωση ανάπτυξης μίας πλήρους εμπορικής εφαρμογής.

## 3.2. Αρχιτεκτονική της πλατφόρμας AidIT

Η αρχιτεκτονική της πλατφόρμας της εφαρμογής AidIT ακολουθεί το μοντέλο του πελάτη-εξυπηρετητή (client/server) με έναν κεντροκοποιημένο εξυπηρετητή, ο οποίος θεωρείται πως βρίσκεται υπό τη διαχείριση του Εθνικού Οργανισμού Υγείας, και με τον οποίον αλληλεπιδρούν οι χρήστες της εφαρμογής για την παραγωγή και κατανάλωση ιατρικών δεδομένων.

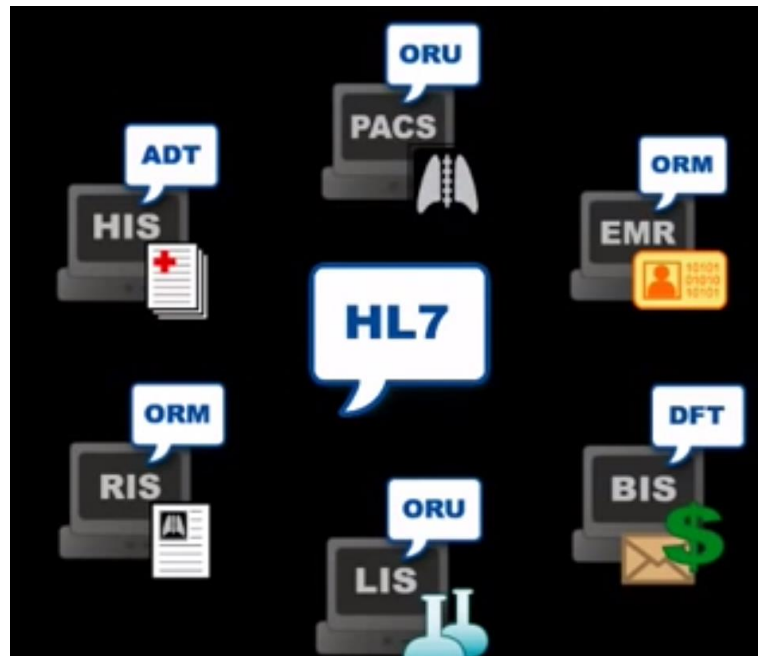
Το μοντέλο αυτό, επιτρέπει τη δημιουργία ιατρικών εφαρμογών που βασίζονται σε ιατρικά δεδομένα, ενώ παράλληλα αποφεύγει ζητήματα συγχρονισμού δεδομένων μεταξύ διαφορετικών συστημάτων. Ωστόσο, στο σημείο αυτό, αξιωματικά σημειωθεί πως η χρησιμοποίηση του προτύπου HL7 FHIR ως διαλειτουργικής δομής για την περιγραφή και ενθυλάκωση των ιατρικών δεδομένων, διευκολύνει την επικοινωνία και την ενσωμάτωση με άλλα συστήματα.

Περνώντας στην πλευρά του πελάτη, η αρχιτεκτονική προωθεί την δημιουργία ιατρικών εφαρμογών που να βασίζονται στις δομές και το πρωτόκολλο επικοινωνίας του προτύπου HL7 FHIR. Η τεχνολογία διαδικτυακών υπηρεσιών REST διευκολύνει τη δημιουργία πρακτικών εφαρμογών για τους ασθενείς, αλλά και επαγγελματικών εφαρμογών για τους παρόχους υγείας ανεξάρτητα από το περιβάλλον εκτέλεσης τους. Επιπρόσθετα, η τεχνολογία REST είναι ιδιαίτερα ταιριαστή για τη δημιουργία πρωτότυπων εφαρμογών κινητών συσκευών.

Στο παρακάτω σχήμα παρουσιάζεται ένα τυπικό νοσοκομειακό σύστημα αποτελούμενο από τα επιμέρους τμήματα του :

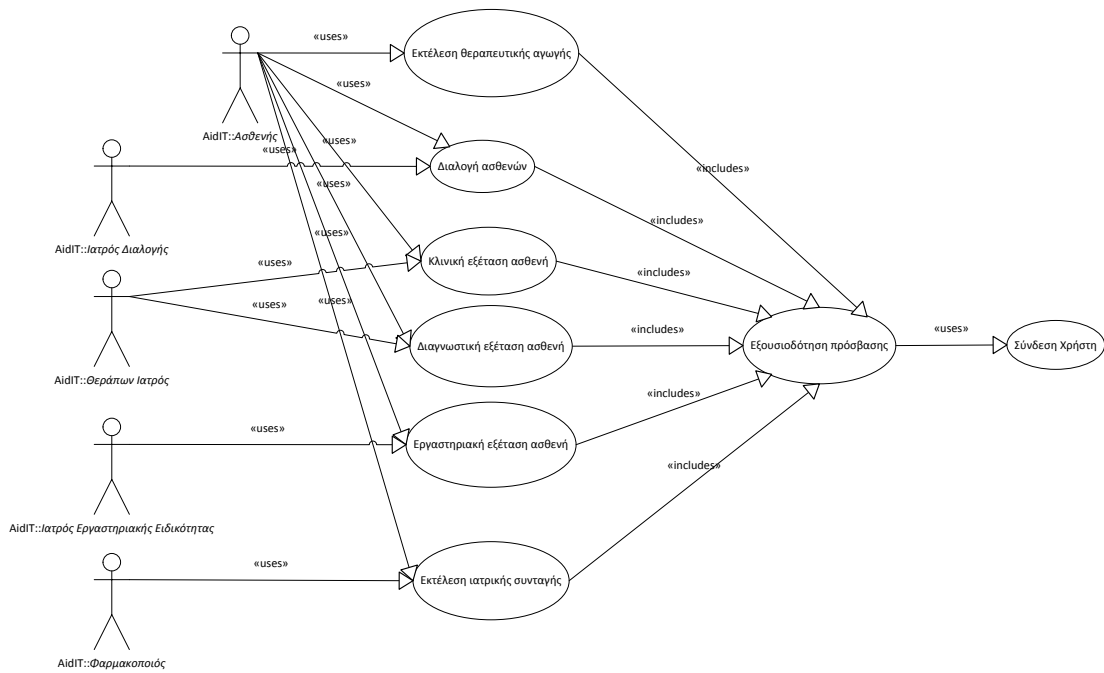
- Πληροφοριακό σύστημα νοσοκομείου (Hospital Information System – HIS)
- Πληροφοριακό σύστημα εργαστηρίων (Laboratory Information System - LIS)
- Σύστημα αποθήκευσης και επικοινωνίας ιατρικών απεικονίσεων (Picture Archiving and Communication System - PACS)
- Πληροφοριακό σύστημα ακτινολογικού (Radiology Information System - RIS)
- Σύστημα διαχείρισης ηλεκτρονικών αρχείων υγείας (Electronic Medical Records System - EMR)
- Πληροφοριακό σύστημα λογιστηρίου νοσοκομείου (Billing Information Systems - BIS)

Η αρχιτεκτονική της πλατφόρμας έχει στο επίκεντρο το πρωτόκολλο HL7 FHIR και ενεργοποιεί τόσο την αποδοτική επικοινωνία μεταξύ των επιμέρους νοσοκομειακών συστημάτων, αλλά και την ενσωμάτωση εφαρμογών κινητών συσκευών στις διαδικασίες παροχής υγείας.



### 3.3. Περιπτώσεις χρήσης

Μια επισκόπηση των βασικών περιπτώσεων χρήσης της εφαρμογής μαζί με τα ενδιαφερόμενα μέρη δίνεται στο παρακάτω σχήμα:



Στις επόμενες παραγράφους περιγράφονται οι περιπτώσεις χρήσης της εφαρμογής που συλλέχτηκαν κατά τη φάση της ανάλυσης απαιτήσεων και από τις οποίες προέκυψαν οι λειτουργικές και μη λειτουργικές απαιτήσεις.

### 3.3.1. UC-001 – Σύνδεση χρήστη

#### 3.3.1.1. Περιγραφή

Η συγκεκριμένη περίπτωση χρήσης αναφέρεται στην διαδικασία εισόδου ενός χρήστη στην εφαρμογή, ανάλογα με τον ρόλο του. Εδώ, θεωρείται πως ο χρήστης έχει ήδη εγγραφεί ως ασφαλισμένος/επαγγελματίας στον Εθνικό Οργανισμό Υγείας, και έχει αποκτήσει μοναδικό αναγνωριστικό κοινωνικής ασφάλισης καθώς και όνομα/κωδικό χρήστη. Μετά από την επιτυχή είσοδο του χρήστη στην εφαρμογή, πρέπει να εμφανίζονται τα στοιχεία του, τα οποία μπορούν να προσπελαστούν από τους συνεργαζόμενους φορείς του ενικού οργανισμού υγείας.

#### 3.3.1.2. Κύριο σενάριο

| Χρήστης   | Εφαρμογή                          | Server                 |
|---|-----------------------------------|------------------------|
| 1. Εκκίνηση εφαρμογής.<br>2. Εμφάνιση αρχικής οθόνης<br>3. Πληκτρολόγηση ονόματος και κωδικού χρήστη. |                                   |                        |
|   | 4. Αποστολή στοιχείων στο server. |                        |
|   |                                   | 5. Έλεγχος εγκυρότητας |

|  |  |  |
|--|--|--|
|  |  | στοιχείων.<br>6. Επιστροφή μηνύματος επιτυχούς σύνδεσης. |
|  | 7. Εμφάνιση κεντρικής οθόνης με στοιχεία χρήστη. |  |

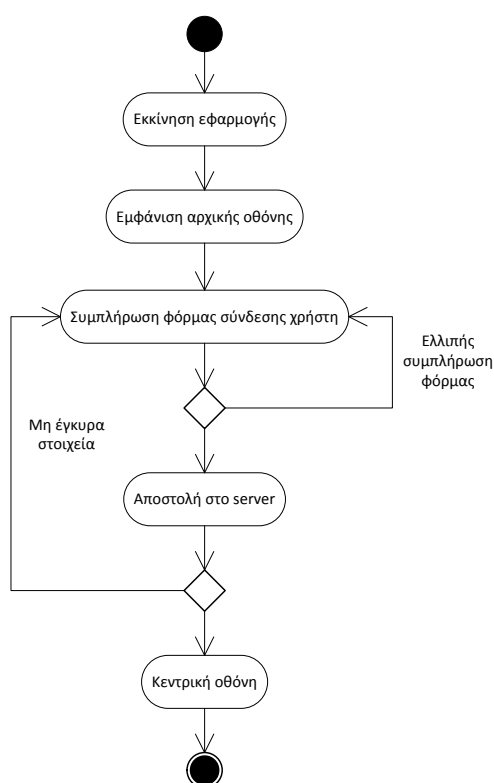
### 3.3.1.3. Ροή διαχείρισης σφάλματος – Λάθος κωδικός/όνομα χρήστη.

| Εφαρμογή  | Server                                       |
|---|--|
|   | 6. Επιστροφή μηνύματος ανεπιτυχούς σύνδεσης. |
| 7. Εμφάνιση μηνύματος σφάλματος.<br>8. Επιστροφή στη φόρμα σύνδεσης χρήστη. |  |

### 3.3.1.4. Λειτουργικές απαιτήσεις

| FREQ-001   | Στοιχεία χρήστη |
|--|-----------------|
| <p>Η εφαρμογή απαιτεί τα παρακάτω στοιχεία χρήστη:</p> <ul style="list-style-type: none"> <li>• ΟΝΟΜΑΤΕΠΩΝΥΜΟ</li> <li>• ΗΜΕΡΟΜΗΝΙΑ ΓΕΝΝΗΣΗΣ</li> <li>• ΔΙΕΥΘΥΝΣΗ ΚΑΤΟΙΚΙΑΣ (Οδός, Αριθμός, Δήμος, Πόλη)</li> <li>• ΤΗΛΕΦΩΝΟ ΕΠΙΚΟΙΝΩΝΙΑΣ</li> <li>• EMAIL</li> <li>• ΑΡΙΘΜΟ ΜΗΤΡΩΟΥ ΚΟΙΝΩΝΙΚΗΣ ΑΣΦΑΛΙΣΗΣ (ΑΜΚΑ)</li> <li>• ΦΩΤΟΓΡΑΦΙΑ ΑΣΦΑΛΙΣΜΕΝΟΥ</li> </ul> |                 |

### 3.3.1.5. Διάγραμμα ροής



## 3.3.2. UC-002 – Διαλογή ασθενών

### 3.3.2.1. Περιγραφή

Η συγκεκριμένη περίπτωση χρήσης αναφέρεται στην διαδικασία διαλογής των ασθενών η οποία ενεργοποιείται με την επίσκεψη ενός ασθενή στο νοσοκομείο, μετά από ενδεχόμενο σύμπτωμα το οποίο εντόπισε ο ίδιος και θα επιθυμούσε να ελεγχθεί από τους ιατρούς του εφημερεύοντος νοσοκομείου.

Πρώτο βήμα πριν την εξέταση του ασθενή αποτελεί η ταυτοποίηση του και ο ενδεχόμενος έλεγχος της ασφαλιστικής ενημερότητάς του. Για τον λόγο αυτό, ο ασθενής ενεργοποιεί την κινητή του εφαρμογή και παρέχει στο Ιατρείο Διαλογής του Νοσοκομείου το αναγνωριστικό του ώστε να μπορεί να ανασυρθούν τα στοιχεία του και ο ιατρικός του φάκελος.

Κατόπιν, ο Ιατρός Διαλογής διενεργεί έναν σύντομο ιατρικό έλεγχο και καταγραφή των συμπτωμάτων στην καρτέλα του ασθενή και τον προωθεί στις κατάλληλης ειδικότητας ιατρό.

Το σενάριο θεωρεί πως τόσο ο ασθενής, όσο και ο ιατρός που διενεργεί τη διαδικασία της διαλογής έχουν ήδη συνδεθεί με την εφαρμογή (UC-001) και πως ο ιατρός διαλογής έχει λάβει την απαραίτητη εξουσιοδότηση (UC-008).

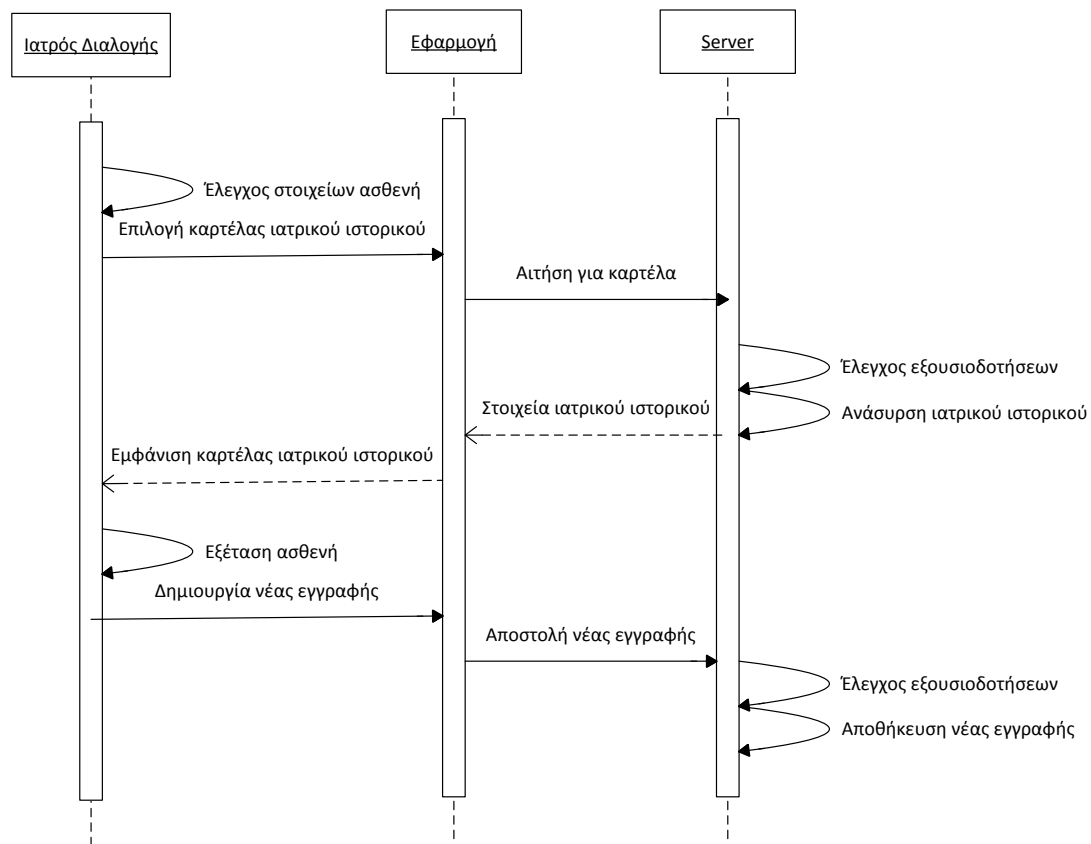
### 3.3.2.2. Κύριο σενάριο

| Ιατρός Διαλογής   | Εφαρμογή                                     | Server  |
|---|--|---|
| 1. Έλεγχος προσωπικών στοιχείων ασθενή<br>2. Επιλογή καρτέλας ιατρικού ιστορικού ασθενή       |  |   |
|   | 3. Αίτηση για λήψη ιατρικού ιστορικού ασθενή |   |
|   |  | 4. Έλεγχος εξουσιοδοτήσεων.<br>5. Ανάσυρση & επιστροφή ιατρικού ιστορικού       |
|   | 6. Εμφάνιση οθόνης ιατρικού ιστορικού.       |   |
| 7. Εξέταση ασθενή<br>8. Δημιουργία ανοικτής εγγραφής εξέτασης στο ιατρικό ιστορικό του ασθενή |  |   |
|   | 9. Αποστολή νέας εγγραφής στον server        |   |
|   |  | 10. Έλεγχος εξουσιοδοτήσεων.<br>11. Αποθήκευση νέας εγγραφής ιατρικού ιστορικού |

### 3.3.2.3. Λειτουργικές απαιτήσεις

| FREQ-002  | Ιατρικό ιστορικό ασθενή |
|---|-------------------------|
| <p>Η εφαρμογή απαιτεί τα παρακάτω στοιχεία για το ιατρικό ιστορικό του ασθενή:</p> <ul style="list-style-type: none"> <li>• ΤΥΠΟΣ ΣΥΜΠΤΩΜΑΤΟΣ/ΑΣΘΕΝΕΙΑΣ</li> <li>• ΗΜΕΡΟΜΗΝΙΑ ΕΚΔΗΛΩΣΗΣ και ΕΛΕΓΧΟΥ ΣΥΜΠΤΩΜΑΤΟΣ</li> <li>• ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΕΛΕΓΚΤΗ ΙΑΤΡΟΥ</li> <li>• ΚΑΤΑΣΤΑΣΗ ΕΛΕΓΧΟΥ ΣΥΜΠΤΩΜΑΤΟΣ</li> <li>• ΣΗΜΕΙΩΣΕΙΣ</li> </ul> |                         |

### 3.3.2.4. Διάγραμμα ακολουθίας



### 3.3.3. UC-003 – Κλινική εξέταση ασθενή

#### 3.3.3.1. Περιγραφή

Η συγκεκριμένη περίπτωση χρήσης αναφέρεται στην διαδικασία εξέταση των ασθενών η οποία ενεργοποιείται μετά από παραπεμπτικό του Ιατρού Διαλογής ή του Θεράποντος Ιατρού. Στην πρώτη περίπτωση, ο Ιατρός Διαλογής έχει πραγματοποιήσει έναν πρώτο έλεγχο στα συμπτώματα του ασθενή και τον παραπέμπει στον κατάλληλης ειδικότητας Θεράπων Ιατρό. Στη δεύτερη περίπτωση, ο Θεράπων Ιατρός παραπέμπει τον ασθενή για επιπρόσθετη κλινική εξέταση από Θεράποντα Ιατρό άλλης ειδικότητας.

Η κλινική εξέταση του ασθενούς εκκινείται με την αξιολόγηση των συμπτωμάτων του ασθενή με παράλληλο έλεγχο του ιατρικού ιστορικού του και της φαρμακευτικής αγωγής την οποία λαμβάνει. Αποτέλεσμα της κλινικής εξέτασης είναι η λεπτομερής καταγραφή της αξιολόγησης των συμπτωμάτων του ασθενή, και η ενδεχόμενη παραπομπή του για εργαστηριακές εξετάσεις ή νέα κλινική εξέταση.

Το σενάριο θεωρεί πως τόσο ο ασθενής, όσο και ο ιατρός που διενεργεί τη διαδικασία της κλινικής εξέτασης έχουν ήδη συνδεθεί με την εφαρμογή (UC-001) και πως ο θεράπων ιατρός έχει λάβει την απαραίτητη εξουσιοδότηση (UC-008).

### 3.3.3.2. Κύριο σενάριο

| Θεράπων Ιατρός   | Εφαρμογή   | Server  |
|--|--|---|
| 1. Προσπέλαση ιατρικού φακέλου ασθενή.   |  |   |
|  | 2. Αίτηση για λήψη καρτελών ασθενή                     |   |
|  |  | 3. Έλεγχος εξουσιοδοτήσεων.<br>4. Ανάσυρση & επιστροφή καρτελών |
|  | 5. Εμφάνιση σχετικής καρτέλας ιατρικού φακέλου ασθενή. |   |
| 6. Εξέταση ασθενή<br>7. Ενημέρωση εγγραφής ιατρικού ιστορικού.<br>8. Δημιουργία παραπεμπτικού για εργαστηριακές εξετάσεις. |  |   |
|  | 9. Αποστολή αλλαγών στον server.                       |   |
|  |  | 10. Έλεγχος εξουσιοδοτήσεων<br>11. Αποθήκευση αλλαγών           |

### 3.3.3.3. Λειτουργικές απαιτήσεις

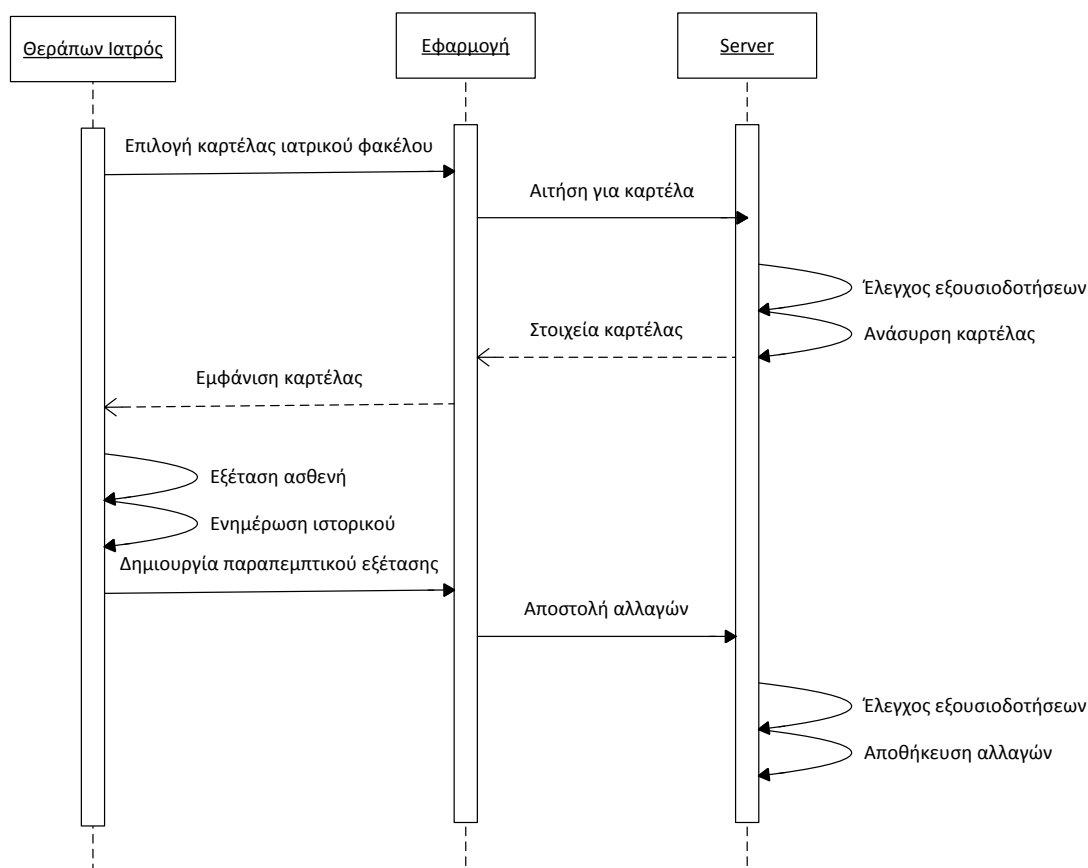
| FREQ-003   | Εργαστηριακή εξέταση ασθενή                               |
|--|---|
| <p>Η εφαρμογή απαιτεί τα παρακάτω στοιχεία για κάθε εργαστηριακή εξέταση ασθενή:</p> <ul style="list-style-type: none"> <li>• ΤΥΠΟΣ ΕΞΕΤΑΣΗΣ</li> <li>• ΗΜΕΡΟΜΗΝΙΑ ΠΑΡΑΠΕΜΠΤΙΚΟΥ και ΕΚΤΕΛΕΣΗΣ ΕΞΕΤΑΣΗΣ</li> <li>• ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΘΕΡΑΠΟΝΤΟΣ ΙΑΤΡΟΥ</li> <li>• ΤΥΠΟΣ ΕΞΕΤΑΣΗΣ</li> <li>• ΚΑΤΑΣΤΑΣΗ ΕΞΕΤΑΣΗΣ</li> <li>• ΣΗΜΕΙΩΣΕΙΣ/ ΑΠΟΤΕΛΕΣΜΑ ΕΞΕΤΑΣΗΣ</li> </ul> |   |
| FREQ-004   | Θεράπων Ιατρός: Δικαιώματα χρήσης ιατρικού φακέλου ασθενή |



Ο Θεράπων Ιατρός πρέπει να έχει δικαίωμα

- Διαβάσματος όλων των καρτελών του ασθενή.
- Ενημέρωσης/Δημιουργίας νέας εγγραφής στο ιατρικό ιστορικό του ασθενή

### 3.3.3.4. Διάγραμμα ακολουθίας



## 3.3.4. UC-004 – Εργαστηριακή εξέταση ασθενή

### 3.3.4.1. Περιγραφή

Η συγκεκριμένη περίπτωση χρήσης αναφέρεται στην εκτέλεση εργαστηριακών εξετάσεων που αφορά κάποιον ασθενή. Η ενεργοποίησή της πραγματοποιείται με την επίσκεψη του ασθενή σε έναν Ιατρό Εργαστηριακής Ειδικότητας. Ο τελευταίος προσπελάζει τα παραπεμπτικά που έχει στην κατοχή του ο ασθενής, πραγματοποιεί τις απαραίτητες εξετάσεις και ενημερώνει με τα παραπεμπτικά με τα αποτελέσματα των εξετάσεων.

Το σενάριο θεωρεί πως τόσο ο ασθενής, όσο και ο ιατρός που διενεργεί τη διαδικασία της εργαστηριακής εξέτασης έχουν ήδη συνδεθεί με την εφαρμογή (UC-001) και πως ο ιατρός

έχει λάβει την απαραίτητη εξουσιοδότηση (UC-008) για να προσπελάσει/ενημερώσει το παραπεμπτικό εξετάσης του ασθενή.

### 3.3.4.2. Κύριο σενάριο

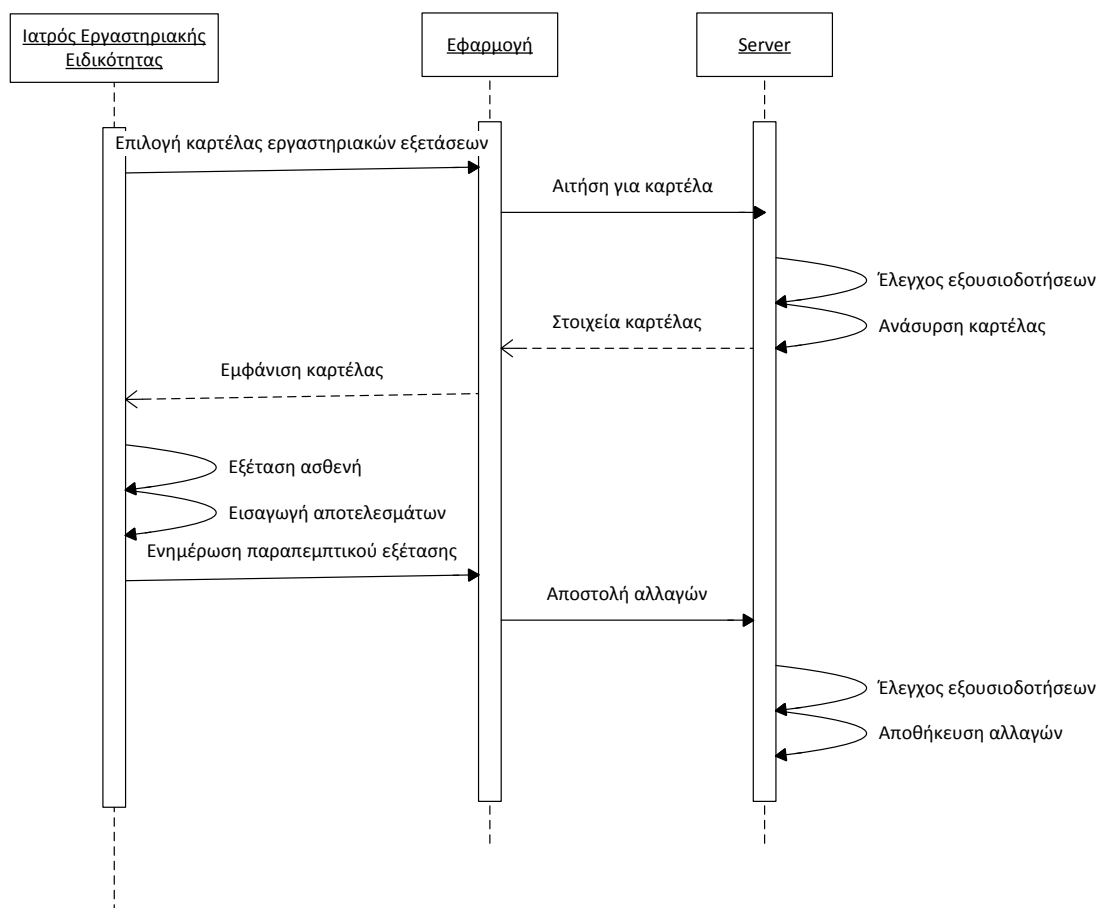
| Ιατρός Εργαστηριακής Ειδικότητας   | Εφαρμογή   | Server  |
|--|--|---|
| 1. Προσπέλαση παραπεμπτικών εργαστηριακών εξετάσεων.                             |  |   |
|  | 2. Αίτηση για λήψη εργαστηριακών παραπεμπτικών ασθενή. |   |
|  |  | 3. Έλεγχος εξουσιοδοτήσεων.<br>4. Ανάσυρση & επιστροφή εργαστηριακών εξετάσεων      |
|  | 5. Εμφάνιση καρτέλας εργαστηριακών εξετάσεων.          |   |
| 6. Διενέργεια εξετάσεων<br>7. Ενημέρωση παραπεμπτικών με αποτελέσματα εξετάσεων. |  |   |
|  | 8. Αποστολή αλλαγών στον server                        |   |
|  |  | 9. Έλεγχος εξουσιοδοτήσεων.<br>10. Αποθήκευση αποτελεσμάτων εργαστηριακών εξετάσεων |

### 3.3.4.3. Λειτουργικές απαιτήσεις

|  |  |
|--|--|
| <b>FREQ-005</b>  | <b>Ιατρός Εργαστηριακής Ειδικότητας: Δικαιώματα χρήσης ιατρικού φακέλου ασθενή</b> |
| <p>Ο Ιατρός Εργαστηριακής Ειδικότητας πρέπει να έχει δικαίωμα</p> <ul style="list-style-type: none"> <li>• Διαβάσματος της καρτέλας προσωπικών στοιχείων του ασθενή</li> </ul> |  |

- Ενημέρωσης εγγραφής εργαστηριακών εξετάσεων

### 3.3.4.4. Διάγραμμα ακολουθίας



### 3.3.5. UC-005 – Διαγνωστική εξέταση ασθενή

#### 3.3.5.1. Περιγραφή

Η συγκεκριμένη περίπτωση χρήσης έπεται της αρχικής κλινικής εξέτασης και της διενέργειας των εργαστηριακών εξετάσεων και αφορά τη συνολική ερμηνεία και αξιολόγηση της υγείας και των συμπτωμάτων του ασθενούς από τον Θεράποντα Ιατρό.

Εδώ, ο Θεράπων Ιατρός προσπελάζει τα αποτελέσματα των εργαστηριακών εξετάσεων τις οποίες είχε παραγγείλει και ενημερώνει το ιατρικό ιστορικό του ασθενούς με την τελική διάγνωση. Στη συνέχεια, ορίζει τη φαρμακευτική αγωγή που πρέπει να ακολουθήσει ο ασθενής, και ενδεχομένως κι ένα πλάνο φροντίδας με στόχο π.χ. την μείωση του βάρους του ασθενούς.

Το σενάριο θεωρεί πως τόσο ο ασθενής, όσο και ο ιατρός που διενεργεί τη διαδικασία της διαγνωστικής εξέτασης έχουν ήδη συνδεθεί με την εφαρμογή (UC-001) και πως ο θεράπων ιατρός έχει λάβει την απαραίτητη εξουσιοδότηση για την προσπέλαση των καρτελών του ασθενή (UC-008).

### 3.3.5.2. Κύριο σενάριο

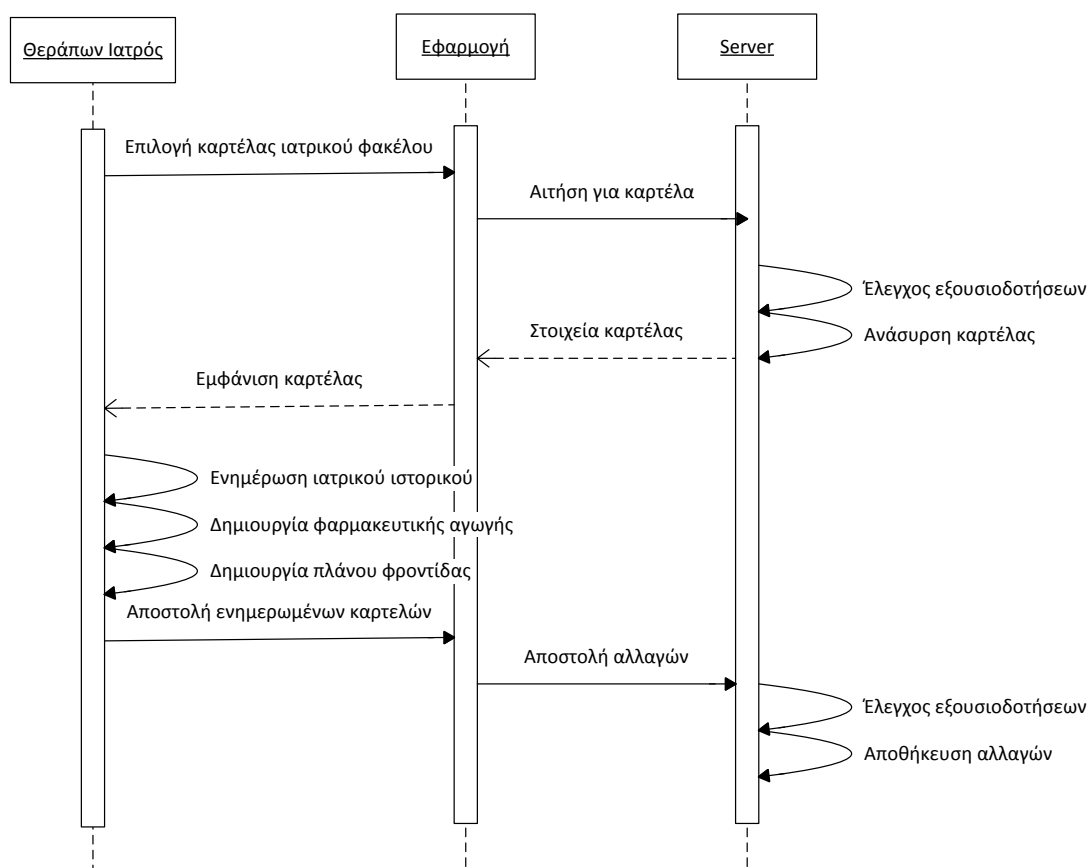
| Θεράπων Ιατρός   | Εφαρμογή   | Server  |
|--|--|---|
| 1. Προσπέλαση ιατρικού φακέλου ασθενή.   |  |   |
|  | 2. Αίτηση για λήψη καρτελών ασθενή                     |   |
|  |  | 3. Έλεγχος εξουσιοδοτήσεων.<br>4. Ανάσυρση & επιστροφή καρτελών |
|  | 5. Εμφάνιση σχετικής καρτέλας ιατρικού φακέλου ασθενή. |   |
| 6. Ενημέρωση εγγραφής ιατρικού ιστορικού.<br>7. Δημιουργία φαρμακευτικής αγωγής.<br>8. Δημιουργία πλάνου φροντίδας |  |   |
|  | 9. Αποστολή αλλαγών στον server.                       |   |
|  |  | 10. Έλεγχος εξουσιοδοτήσεων<br>11. Αποθήκευση αλλαγών           |

### 3.3.5.3. Λειτουργικές απαιτήσεις

| FREQ-006  | Φαρμακευτική αγωγή ασθενή |
|---|---------------------------|
| <p>Η εφαρμογή απαιτεί τα παρακάτω στοιχεία για τον ορισμό μιας φαρμακευτικής αγωγής για τον ασθενή:</p> <ul style="list-style-type: none"> <li>• ΔΡΑΣΤΙΚΗ ΟΥΣΙΑ</li> <li>• ΗΜΕΡΗΣΙΑ ΔΟΣΟΛΟΓΙΑ</li> <li>• ΠΕΡΙΟΔΟΣ ΘΕΡΑΠΕΙΑΣ</li> <li>• ΗΜΕΡΟΜΗΝΙΑ ΔΗΜΙΟΥΡΓΙΑΣ ΕΓΓΡΑΦΗΣ</li> <li>• ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΙΑΤΡΟΥ</li> </ul> |                           |

|  |  |
|--|--|
| <b>FREQ-007</b>  | <b>Πλάνο φροντίδας ασθενή</b>                                    |
| <p>Η εφαρμογή απαιτεί τα παρακάτω στοιχεία τον ορισμό ενός πλάνου φροντίδας για τον ασθενή:</p> <ul style="list-style-type: none"> <li>• ΤΥΠΟΣ ΜΕΤΡΗΣΗΣ</li> <li>• ΕΠΙΘΥΜΗΤΗ ΗΜΕΡΟΜΗΝΙΑ ΛΗΨΗΣ ΜΕΤΡΗΣΗΣ</li> <li>• ΑΠΟΤΕΛΕΣΜΑ ΜΕΤΡΗΣΗΣ</li> <li>• ΤΥΠΟΣ ΕΞΕΤΑΣΗΣ</li> <li>• ΚΑΤΑΣΤΑΣΗ ΜΕΤΡΗΣΗΣ</li> <li>• ΣΗΜΕΙΩΣΕΙΣ</li> </ul> |  |
| <b>FREQ-008</b>  | <b>Θεράπων Ιατρός: Δικαιώματα χρήσης ιατρικού φακέλου ασθενή</b> |
| <p>Ο Θεράπων Ιατρός πρέπει να έχει δικαίωμα</p> <ul style="list-style-type: none"> <li>• Ενημέρωσης/Διαγραφής/Δημιουργίας νέας εγγραφής στις φαρμακευτικές αγωγές του ασθενή</li> <li>• Ενημέρωσης/Διαγραφής/Δημιουργίας νέας εγγραφής στα πλάνα φροντίδας του ασθενή.</li> </ul>  |  |

### 3.3.5.4. Διάγραμμα ακολουθίας



### 3.3.6. UC-006 – Εκτέλεση ιατρικής συνταγής

#### 3.3.6.1. Περιγραφή

Η συγκεκριμένη περίπτωση χρήσης αναφέρεται στην εκτέλεση της ιατρικής συνταγής που έχει χορηγήσει ο θεράπων ιατρός στον ασθενή. Η ενεργοποίησή της πραγματοποιείται με την επίσκεψη του ασθενή στο Φαρμακείο και την επίδειξη της στον Φαρμακοποιό. Ο τελευταίος προσπελάζει την ιατρική συνταγή μέσω της καρτέλας φαρμακευτικής αγωγής και ελέγχοντας παράλληλα το ιατρικό ιστορικό του ασθενή για τυχόν αλλεργίες ή επικαλυπτόμενες αγωγές τον προμηθεύει με το κατάλληλο φαρμακευτικό σκεύασμα.

Το σενάριο θεωρεί πως τόσο ο ασθενής, όσο και ο φαρμακοποιός έχουν ήδη συνδεθεί με την εφαρμογή (UC-001) και πως ο φαρμακοποιός έχει λάβει την απαραίτητη εξουσιοδότηση (UC-008) για να προσπελάσει την καρτέλα ιατρικού ιστορικού του ασθενή και της φαρμακευτικής αγωγής του, ώστε να εκτελέσει την ιατρική συνταγή.

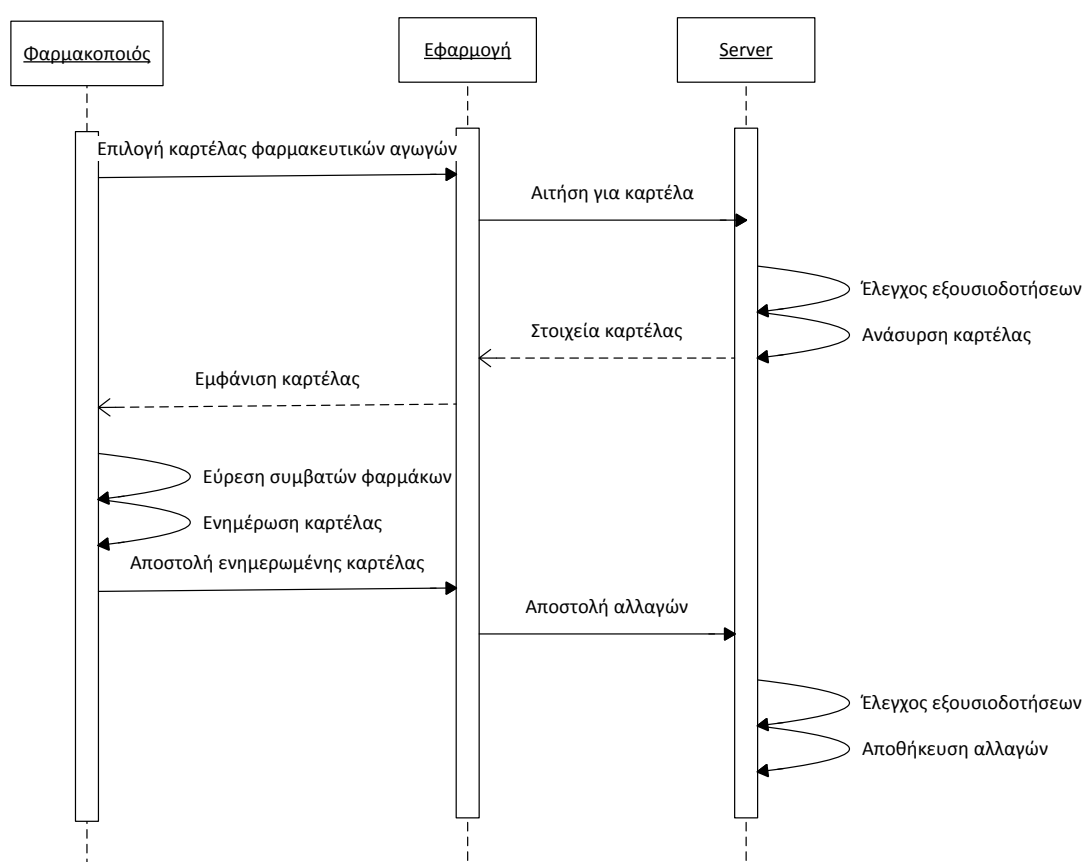
#### 3.3.6.2. Κύριο σενάριο

| Φαρμακοποιός   | Εφαρμογή                                    | Server   |
|--|---|--|
| 1. Προσπέλαση καρτέλας φαρμακευτικής αγωγής και ιστορικού ασθενή.                              |   |  |
|  | 2. Αίτηση για λήψη των καρτελών του ασθενή. |  |
|  |   | 3. Έλεγχος εξουσιοδοτήσεων.<br>4. Ανάσυρση & επιστροφή στοιχείων                 |
|  | 5. Εμφάνιση καρτέλας φαρμακευτικών αγωγών.  |  |
| 6. Ανεύρεση συμβατού φαρμακευτικού σκεύασματος<br>7. Ενημέρωση εκτέλεσης φαρμακευτικής αγωγής. |   |  |
|  | 8. Αποστολή επιβεβαίωσης στον server        |  |
|  |   | 9. Έλεγχος εξουσιοδοτήσεων.<br>10. Αποθήκευση ενημερωμένης φαρμακευτικής αγωγής. |

### 3.3.6.3. Λειτουργικές απαιτήσεις

| FREQ-009   | Δικαιώματα χρήσης ιατρικού φακέλου ασθενή: Φαρμακοποιός |
|--|---|
| <p>Ο Φαρμακοποιός πρέπει να έχει δικαίωμα</p> <ul style="list-style-type: none"><li>• Διαβάσματος της καρτέλας προσωπικών στοιχείων του ασθενή</li><li>• Διαβάσματος της καρτέλας ιατρικού ιστορικού ασθενή</li><li>• Διαβάσματος και ενημέρωσης της καρτέλας φαρμακευτικών αγωγών του ασθενή.</li></ul> |   |

### 3.3.6.4. Διάγραμμα ακολουθίας



### 3.3.7. UC-007 – Εκτέλεση θεραπευτικής αγωγής

#### 3.3.7.1. Περιγραφή

Η συγκεκριμένη περίπτωση χρήσης αναφέρεται στην εκτέλεση της θεραπευτικής αγωγής που έχει προταθεί στον χρήστη από τον θεράποντα ιατρό. Το σενάριο επικεντρώνεται στην υποβοήθηση του ασθενή για την σωστή της φαρμακευτικής αγωγής και των μετρήσεων υγείας που έχει ζητήσει ο θεράπων ιατρός. Όλα τα παραπάνω λαμβάνουν χώρα μέσω της

καρτέλας πλάνων φροντίδας, στην οποία εμφανίζονται ημερολογιακά ταξινομημένες οι ενέργειες που πρέπει να εκτελέσει ο ασθενής κάθε μέρα. Μέσα από την καρτέλα είναι δυνατόν να εισάγει τις μετρήσεις υγείας του ο ασθενής, αλλά και να ορίσει πως της ημερήσιας δΟΣολογίας φαρμάκου λήφθηκε έγκαιρα.

### 3.3.7.2. Κύριο σενάριο

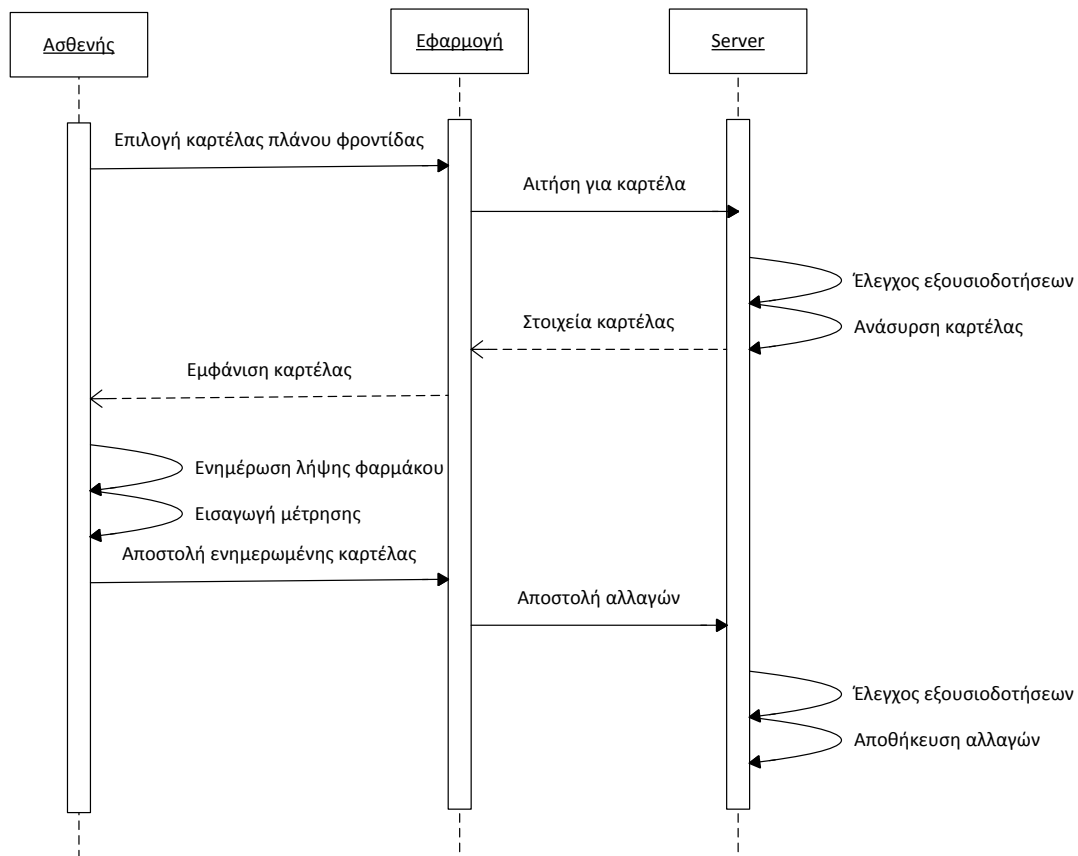
| Ασθενής  | Εφαρμογή                                    | Server   |
|--|---|--|
| 1. Προσπέλαση καρτέλας πλάνου φροντίδας.   |   |  |
|  | 2. Αίτηση για λήψη των καρτελών του ασθενή. |  |
|  |   | 3. Έλεγχος εξουσιοδοτήσεων.<br>4. Ανάσυρση & επιστροφή στοιχείων             |
|  | 5. Εμφάνιση καρτέλας πλάνου φροντίδας.      |  |
| 6. Ενημέρωση εγγραφής λήψης φαρμάκου<br>7. Εισαγωγή αποτελεσμάτων ιατρικής μέτρησης. |   |  |
|  | 8. Αποστολή στοιχείων στον server           |  |
|  |   | 9. Έλεγχος εξουσιοδοτήσεων.<br>10. Αποθήκευση ενημερωμένου πλάνου φροντίδας. |

### 3.3.7.3. Λειτουργικές απαιτήσεις

|  |   |
|--|---|
| <b>FREQ-010</b>  | <b>Δικαιώματα χρήσης ιατρικού φακέλου ασθενή: Ασθενής</b> |
| <p>Ο Ασθενής πρέπει να έχει δικαίωμα</p> <ul style="list-style-type: none"> <li>• Διαβάσματος της καρτέλας φαρμακευτικής αγωγής</li> <li>• Διαβάσματος και ενημέρωσης της πλάνου φροντίδας.</li> </ul> |   |



### 3.3.7.4. Διάγραμμα ακολουθίας



### 3.3.8. UC-008 – Εξουσιοδότηση πρόσβασης στον ιατρικό φάκελο του ασθενή

#### 3.3.8.1. Περιγραφή

Η συγκεκριμένη περίπτωση χρήσης είναι ξεχωριστής σημασίας καθώς περιγράφει τον τρόπο με τον οποίο παρέχονται εξουσιοδοτήσεις πρόσβασης στους ιατρικούς φακέλους, από τους ασθενείς προς τους επαγγελματίες παρόχους υπηρεσιών υγείας. Το μοντέλο ελέγχου που ακολουθείται λαμβάνει υπόψη τον ρόλο των χρηστών καθώς και τον τύπο των δεδομένων τα οποία επιθυμούν να προσπελάσουν. Οι εξουσιοδοτήσεις ελέγχου πρόσβασης αποτελούν ρητή καταγραφή της επιθυμίας του χρήστη να εκχωρήσουν δικαιώματα που αφορούν την αλληλεπίδραση με τον ιατρικό του φάκελο. Αρχικά, ο ασθενής λαμβάνει τα στοιχεία του χρήστη που επιθυμεί να αποκτήσει πρόσβαση στα ιατρικά του δεδομένα, και στη συνέχεια παρέχει την εξουσιοδότηση. Με αυτόν τον τρόπο πραγματοποιείται μια αμοιβαία ταυτοποίηση των δύο χρηστών.

Τέλος, για την εύκολη και απρόσκοπτη εκχώρηση δικαιωμάτων μεταξύ ασθενών, θα χρησιμοποιούνται οι κάμερες των συσκευών και την τεχνολογία QR-Code μέσω της οποίας θα απεικονίζονται και θα σαρώνονται τα προσωρινά αναγνωριστικά των χρηστών.

### 3.3.8.2. Κύριο σενάριο

| Ασθενής   | Εφαρμογή  | Χρήστης                                     |
|---|---|---|
|   |   | 1. Έναρξη διαδικασίας λήψης εξουσιοδότησης. |
|   | 2. Εμφάνιση στην οθόνη αναγνωριστικού αιτούμενου χρήστη σε μορφή QR-Code. |   |
| 3. Σάρωση και λήψη αναγνωριστικού.                                      |   |   |
|   | 4. Λήψη και εμφάνιση στοιχείων χρήστη με βάση το αναγνωριστικό.           |   |
| 5. Ταυτοποίηση χρήστη.<br>6. Έναρξη διαδικασίας παροχής εξουσιοδότησης. |   |   |
|   | 7. Εμφάνιση στην οθόνη αναγνωριστικού ασθενή σε μορφή QR-Code.            |   |
|   |   | 8. Σάρωση και λήψη αναγνωριστικού.          |
|   | 9. Λήψη και εμφάνιση στοιχείων χρήστη με βάση το αναγνωριστικό.           |   |
|   |   | 10. Ταυτοποίηση χρήστη.                     |

### 3.3.8.3. Λειτουργικές απαιτήσεις

|                 |   |
|-----------------|---|
| <b>FREQ-011</b> | <b>Έλεγχος πρόσβασης ιατρικού φακέλου ασθενή</b>  |
|                 | <ul style="list-style-type: none"> <li>Αμοιβαία ταυτοποίηση ασθενή και χρήστη</li> <li>Προσωρινό αναγνωριστικό ασθενή για αλληλεπίδραση με τον ιατρικό του φάκελο.</li> <li>Αλληλεπίδραση με τον ιατρικό φάκελο του ασθενή ανάλογα με τον ρόλο του αιτούμενου χρήστη</li> </ul> |
| <b>FREQ-012</b> | <b>Χρήση QR-Code</b>  |
|                 | <ul style="list-style-type: none"> <li>Απεικόνιση αναγνωριστικού σε μορφή QR-Code</li> <li>Σάρωση αναγνωριστικού QR-Code με κάμερα κινητής συσκευής</li> </ul>  |

### 3.3.8.4. Διάγραμμα ροής





# 4

## Σχεδίαση

Στο παρόν κεφάλαιο παρουσιάζονται οι λειτουργικότητες της πλατφόρμας AidIT, αλλά και το πώς μεταφράζονται αυτές σε κλάσεις, μεθόδους, μεταβλητές και τις αλληλεπιδράσεις τους.

### 4.1. Η πλατφόρμα AidIT

#### 4.1.1. Λειτουργίες του εξυπηρετητή

Ο server έχει κεντρικό ρόλο στην εφαρμογή, αφού μέσω εκείνου πραγματοποιείται η επικοινωνία των δεδομένων μεταξύ των χρηστών του συστήματος. Συγκεκριμένα ο ρόλος του server στην εφαρμογή φαίνεται στις παρακάτω λειτουργίες:

- Διαχειρίζεται την **σύνδεση** και **αποσύνδεση** των χρηστών της εφαρμογής
- Παράγει **προσωρινά αναγνωριστικά** για την ασφάλεια των συναλλαγών των χρηστών του συστήματος.
- Παρέχει τα ιατρικά δεδομένα των ασθενών μέσω της **διαδικτυακής υπηρεσίας REST** η οποία προδιαγράφεται από το πρότυπο **HL7 FHIR**
- Διατηρεί **σχεσιακή βάση δεδομένων** με τα ιατρικά δεδομένα των ασθενών.
- **Επεξεργάζεται τις αποθηκεύει** τις διαλειτουργικές δομές του **HL7 FHIR** σε σχεσιακή βάση δεδομένων.

- Ανασύρει και δημιουργεί μηνύματα **HL7 FHIR** από τις εγγραφές της βάσης δεδομένων.
- Ελέγχει την πρόσβαση των χρηστών στα ιατρικά δεδομένα των ασθενών με βάση τον ρόλο του αιτούμενου.

## 4.1.2. Περιγραφή κλάσεων

### 4.1.2.1. Servlets

|              |  |             |
|--------------|--|-------------|
| Όνομα κλάσης | <b>LoginServlet</b>  | HttpServlet |
| GET          | <ul style="list-style-type: none"> <li>• <i>/login?username=&lt;username&gt;&amp;password=&lt;password&gt;</i><br/>Σύνδεση χρήστη την εφαρμογή, λήψη του ρόλου του και του προσωρινού του αναγνωριστικού.</li> </ul> |             |

|              |   |             |
|--------------|---|-------------|
| Όνομα κλάσης | <b>LogoutServlet</b>  | HttpServlet |
| GET          | <ul style="list-style-type: none"> <li>• <i>/logout</i><br/>Αποσύνδεση χρήστη από την εφαρμογή. Ακύρωση του προσωρινού αναγνωριστικού του.</li> </ul> |             |

### 4.1.2.2. Restlets

|              |   |  |
|--------------|---|--|
| Όνομα κλάσης | <b>PatientRestlet</b>   |  |
| GET          | <ul style="list-style-type: none"> <li>• <i>/patients/{patientTempId}</i><br/>Επιστρέφει στη μορφή HL7 FHIR Patient τα προσωπικά στοιχεία του ασθενή με το αντίστοιχο προσωρινό αναγνωριστικό, εάν και μόνο εάν ο αιτούμενος χρήστης έχει τον κατάλληλο ρόλο. Διαφορετικά επιστρέφει HTTP status 401 Unauthorized.</li> </ul> |  |

|              |  |  |
|--------------|--|--|
| Όνομα κλάσης | <b>PractitionerRestlet</b>   |  |
| GET          | <ul style="list-style-type: none"> <li>• <i>/practitioners/{practitionerTempId}</i><br/>Επιστρέφει στη μορφή HL7 FHIR Practitioner το ιατρικό ιστορικό του ασθενή με το αντίστοιχο προσωρινό αναγνωριστικό.</li> </ul> |  |

|              |  |  |
|--------------|--|--|
| Όνομα κλάσης | <b>HealthInfoRestlet</b>   |  |
| GET          | <ul style="list-style-type: none"> <li>• <i>/patients/{patientTempId}/conditions</i><br/>Επιστρέφει στη μορφή HL7 FHIR Condition τα προσωπικά στοιχεία του παρόχου υπηρεσιών υγείας με το αντίστοιχο προσωρινό αναγνωριστικό, εάν και μόνο εάν ο αιτούμενος χρήστης έχει τον κατάλληλο ρόλο. Διαφορετικά επιστρέφει HTTP status 401 Unauthorized.</li> </ul> |  |
| POST         | <ul style="list-style-type: none"> <li>• <i>/patients/{patientTempId}/conditions</i><br/>Επεξεργάζεται το εισερχόμενο μήνυμα HL7 FHIR Condition και δημιουργεί εγγραφή ιατρικού ιστορικού για τον ασθενή με το αντίστοιχο προσωρινό</li> </ul>   |  |

|        |   |
|--------|---|
|        | αναγνωριστικό, εάν και μόνο εάν ο αιτούμενος χρήστης έχει τον κατάλληλο ρόλο. Διαφορετικά επιστρέφει HTTP status 401 Unauthorized.  |
| UPDATE | <ul style="list-style-type: none"> <li>• <b><i>/patients/{patientTempId}/conditions/{conditionId}</i></b><br/>Επεξεργάζεται το εισερχόμενο μήνυμα HL7 FHIR Condition και ενημερώνει την εγγραφή ιατρικού ιστορικού με αναγνωριστικό {conditionId}, εάν και μόνο εάν ο αιτούμενος χρήστης έχει τον κατάλληλο ρόλο. Διαφορετικά επιστρέφει HTTP status 401 Unauthorized.</li> </ul> |
| DELETE | <ul style="list-style-type: none"> <li>• <b><i>/patients/{patientTempId}/conditions/{conditionId}</i></b><br/>Διαγράφει την εγγραφή ιατρικού ιστορικού με αναγνωριστικό {conditionId}, εάν και μόνο εάν ο αιτούμενος χρήστης έχει τον κατάλληλο ρόλο. Διαφορετικά επιστρέφει HTTP status 401 Unauthorized.</li> </ul>   |

|              |   |
|--------------|---|
| Όνομα κλάσης | <b>LabOrderRestlet</b>  |
| GET          | <ul style="list-style-type: none"> <li>• <b><i>/patients/{patientTempId}/laborders</i></b><br/>Επιστρέφει στη μορφή HL7 FHIR DiagnosticOrder τις εργαστηριακές εξετάσεις του ασθενή με το αντίστοιχο προσωρινό αναγνωριστικό, εάν και μόνο εάν ο αιτούμενος χρήστης έχει τον κατάλληλο ρόλο. Διαφορετικά επιστρέφει HTTP status 401 Unauthorized.</li> </ul>                                      |
| POST         | <ul style="list-style-type: none"> <li>• <b><i>/patients/{patientTempId}/laborders</i></b><br/>Επεξεργάζεται το εισερχόμενο μήνυμα HL7 FHIR DiagnosticOrder και δημιουργεί εγγραφή εργαστηριακής εξέτασης για τον ασθενή με το αντίστοιχο προσωρινό αναγνωριστικό, εάν και μόνο εάν ο αιτούμενος χρήστης έχει τον κατάλληλο ρόλο. Διαφορετικά επιστρέφει HTTP status 401 Unauthorized.</li> </ul> |
| UPDATE       | <ul style="list-style-type: none"> <li>• <b><i>/patients/{patientTempId}/laborders/{labOrderId}</i></b><br/>Επεξεργάζεται το εισερχόμενο μήνυμα HL7 FHIR DiagnosticOrder και ενημερώνει την εγγραφή ιατρικού ιστορικού με αναγνωριστικό {labOrderId}, εάν και μόνο εάν ο αιτούμενος χρήστης έχει τον κατάλληλο ρόλο. Διαφορετικά επιστρέφει HTTP status 401 Unauthorized.</li> </ul>              |
| DELETE       | <ul style="list-style-type: none"> <li>• <b><i>/patients/{patientTempId}/laborders/{labOrderId}</i></b><br/>Διαγράφει την εγγραφή εργαστηριακής εξέτασης (και τα αποτελέσματα της) με αναγνωριστικό {labOrderId}, εάν και μόνο εάν ο αιτούμενος χρήστης έχει τον κατάλληλο ρόλο. Διαφορετικά επιστρέφει HTTP status 401 Unauthorized.</li> </ul>  |

|              |  |
|--------------|--|
| Όνομα κλάσης | <b>LabResultsRestlet</b>   |
| GET          | <ul style="list-style-type: none"> <li>• <b><i>/patients/{patientTempId}/laborders/{labOrderId}/results</i></b><br/>Επιστρέφει στη μορφή HL7 FHIR DiagnosticReport τα αποτελέσματα της εργαστηριακής εξέτασης με αναγνωριστικό {labOrderId}, εάν και μόνο εάν ο αιτούμενος χρήστης έχει τον κατάλληλο ρόλο. Διαφορετικά επιστρέφει HTTP status 401 Unauthorized.</li> </ul>  |
| POST         | <ul style="list-style-type: none"> <li>• <b><i>/patients/{patientTempId}/laborders/{labOrderId}/results</i></b><br/>Επεξεργάζεται το εισερχόμενο μήνυμα HL7 FHIR DiagnosticReport και δημιουργεί εγγραφή αποτελεσμάτων εργαστηριακών εξετάσεων για τον ασθενή με το αντίστοιχο προσωρινό αναγνωριστικό, εάν και μόνο εάν ο αιτούμενος χρήστης έχει τον κατάλληλο ρόλο. Διαφορετικά επιστρέφει HTTP status 401 Unauthorized.</li> </ul> |
| UPDATE       | <ul style="list-style-type: none"> <li>• <b><i>/patients/{patientTempId}/laborders/{labOrderId}/results/{labResultId}</i></b><br/>Επεξεργάζεται το εισερχόμενο μήνυμα HL7 FHIR DiagnosticReport και ενημερώνει την εγγραφή εργαστηριακών εξετάσεων με αναγνωριστικό</li> </ul>   |

|        |   |
|--------|---|
|        | {labResultId}, εάν και μόνο εάν ο αιτούμενος χρήστης έχει τον κατάλληλο ρόλο. Διαφορετικά επιστρέφει HTTP status 401 Unauthorized.  |
| DELETE | <ul style="list-style-type: none"> <li>• <b><i>/patients/{patientTempId}/laborders/{labOrderId}/results/{labResultId}</i></b><br/>Διαγράφει την εγγραφή αποτελεσμάτων εργαστηριακής εξέτασης με αναγνωριστικό {labResultId}, εάν και μόνο εάν ο αιτούμενος χρήστης έχει τον κατάλληλο ρόλο. Διαφορετικά επιστρέφει HTTP status 401 Unauthorized.</li> </ul> |

|              |  |
|--------------|--|
| Όνομα κλάσης | <b>MedicationPrescriptionRestlet</b>   |
| GET          | <ul style="list-style-type: none"> <li>• <b><i>/patients/{patientTempId}/prescriptions</i></b><br/>Επιστρέφει στη μορφή HL7 FHIR MedicationPrescription τις εργαστηριακές εξετάσεις του ασθενή με το αντίστοιχο προσωρινό αναγνωριστικό, εάν και μόνο εάν ο αιτούμενος χρήστης έχει τον κατάλληλο ρόλο. Διαφορετικά επιστρέφει HTTP status 401 Unauthorized.</li> </ul>                                    |
| POST         | <ul style="list-style-type: none"> <li>• <b><i>/patients/{patientTempId}/prescriptions</i></b><br/>Επεξεργάζεται το εισερχόμενο μήνυμα HL7 FHIR MedicationPrescription και δημιουργεί εγγραφή φαρμακευτικής αγωγής για τον ασθενή με το αντίστοιχο προσωρινό αναγνωριστικό, εάν και μόνο εάν ο αιτούμενος χρήστης έχει τον κατάλληλο ρόλο. Διαφορετικά επιστρέφει HTTP status 401 Unauthorized.</li> </ul> |
| UPDATE       | <ul style="list-style-type: none"> <li>• <b><i>/patients/{patientTempId}/prescriptions/{prescriptionId}</i></b><br/>Επεξεργάζεται το εισερχόμενο μήνυμα HL7 FHIR MedicationPrescription και ενημερώνει την εγγραφή φαρμακευτικής αγωγής με αναγνωριστικό {prescriptionId}, εάν και μόνο εάν ο αιτούμενος χρήστης έχει τον κατάλληλο ρόλο. Διαφορετικά επιστρέφει HTTP status 401 Unauthorized.</li> </ul>  |
| DELETE       | <ul style="list-style-type: none"> <li>• <b><i>/patients/{patientTempId}/prescriptions/{prescriptionId}</i></b><br/>Διαγράφει την εγγραφή φαρμακευτικής αγωγής με αναγνωριστικό {prescriptionId}, εάν και μόνο εάν ο αιτούμενος χρήστης έχει τον κατάλληλο ρόλο. Διαφορετικά επιστρέφει HTTP status 401 Unauthorized.</li> </ul>   |

|              |  |
|--------------|--|
| Όνομα κλάσης | <b>CarePlanRestlet</b>   |
| GET          | <ul style="list-style-type: none"> <li>• <b><i>/patients/{patientTempId}/careplans</i></b><br/>Επιστρέφει στη μορφή HL7 FHIR CarePlan τα πλάνα φροντίδας του ασθενή με το αντίστοιχο προσωρινό αναγνωριστικό, εάν και μόνο εάν ο αιτούμενος χρήστης έχει τον κατάλληλο ρόλο. Διαφορετικά επιστρέφει HTTP status 401 Unauthorized.</li> </ul>   |
| POST         | <ul style="list-style-type: none"> <li>• <b><i>/patients/{patientTempId}/careplans</i></b><br/>Επεξεργάζεται το εισερχόμενο μήνυμα HL7 FHIR CarePlan και δημιουργεί εγγραφή πλάνου φροντίδας για τον ασθενή με το αντίστοιχο προσωρινό αναγνωριστικό, εάν και μόνο εάν ο αιτούμενος χρήστης έχει τον κατάλληλο ρόλο. Διαφορετικά επιστρέφει HTTP status 401 Unauthorized.</li> </ul> |
| UPDATE       | <ul style="list-style-type: none"> <li>• <b><i>/patients/{patientTempId}/careplans/{carePlanId}</i></b><br/>Επεξεργάζεται το εισερχόμενο μήνυμα HL7 FHIR CarePlan και ενημερώνει την εγγραφή πλάνου φροντίδας με αναγνωριστικό {carePlanId}, εάν και μόνο εάν ο αιτούμενος χρήστης έχει τον κατάλληλο ρόλο. Διαφορετικά επιστρέφει HTTP status 401 Unauthorized.</li> </ul>          |
| DELETE       | <ul style="list-style-type: none"> <li>• <b><i>/patients/{patientTempId}/careplans/{carePlanId}</i></b><br/>Διαγράφει την εγγραφή πλάνου φροντίδας με αναγνωριστικό {carePlanId}, εάν και μόνο εάν ο αιτούμενος χρήστης έχει τον κατάλληλο ρόλο. Διαφορετικά επιστρέφει HTTP status 401 Unauthorized.</li> </ul>   |



#### 4.1.2.3. Βοηθητικές κλάσεις HL7 FHIR

| Όνομα κλάσης | <b>CreateFHIRResource</b>  |
|--------------|--|
| Μέθοδοι      | <ul style="list-style-type: none"> <li>• <b><i>createCondition (ConditionDTO condition)</i></b><br/>Επεξεργασία αντικειμένου βάσης δεδομένων και μετατροπή στη διαλειτουργική δομή του προτύπου HL7 FHIR.</li> <li>• <b><i>createCarePlan(CarePlanDTO carePlan)</i></b><br/>Επεξεργασία αντικειμένου βάσης δεδομένων και μετατροπή στη διαλειτουργική δομή του προτύπου HL7 FHIR.</li> <li>• <b><i>createDiagnosticOrder (DiagnosticOrderDTO diagnosticOrder)</i></b><br/>Επεξεργασία αντικειμένου βάσης δεδομένων και μετατροπή στη διαλειτουργική δομή του προτύπου HL7 FHIR.</li> <li>• <b><i>createDiagnosticReport(DiagnosticReportDTO diagnosticReport)</i></b><br/>Επεξεργασία αντικειμένου βάσης δεδομένων και μετατροπή στη διαλειτουργική δομή του προτύπου HL7 FHIR.</li> <li>• <b><i>createMedicationPrescription (MedicationPrescriptionDTO medicationPrescription)</i></b><br/>Επεξεργασία αντικειμένου βάσης δεδομένων και μετατροπή στη διαλειτουργική δομή του προτύπου HL7 FHIR.</li> <li>• <b><i>createPractitioner (UserDTO practitioner)</i></b><br/>Επεξεργασία αντικειμένου βάσης δεδομένων και μετατροπή στη διαλειτουργική δομή του προτύπου HL7 FHIR.</li> <li>• <b><i>createPatient(UserDTO patient)</i></b><br/>Επεξεργασία αντικειμένου βάσης δεδομένων και μετατροπή στη διαλειτουργική δομή του προτύπου HL7 FHIR.</li> </ul> |

| Όνομα κλάσης | <b>ParseFHIRResource</b>   |
|--------------|--|
| Μέθοδοι      | <ul style="list-style-type: none"> <li>• <b><i>parseCondition (Condition condition)</i></b><br/>Επεξεργασία δομής του προτύπου HL7 FHIR και μετατροπή σε αντικείμενο της βάσης δεδομένων.</li> <li>• <b><i>parseCarePlan(CarePlan carePlan)</i></b><br/>Επεξεργασία δομής του προτύπου HL7 FHIR και μετατροπή σε αντικείμενο της βάσης δεδομένων..</li> <li>• <b><i>parseDiagnosticOrder (DiagnosticOrder diagnosticOrder)</i></b><br/>Επεξεργασία δομής του προτύπου HL7 FHIR και μετατροπή σε αντικείμενο της βάσης δεδομένων.</li> <li>• <b><i>parseDiagnosticReport(DiagnosticReport diagnosticReport)</i></b><br/>Επεξεργασία δομής του προτύπου HL7 FHIR και μετατροπή σε αντικείμενο της βάσης δεδομένων.</li> <li>• <b><i>parseMedicationPrescription (MedicationPrescription medicationPrescription)</i></b><br/>Επεξεργασία δομής του προτύπου HL7 FHIR και μετατροπή σε αντικείμενο της βάσης δεδομένων.</li> </ul> |

#### 4.1.2.4. Βοηθητικές κλάσεις Hibernate

| Όνομα κλάσης | <b>HibernateUtil</b> |
|--------------|----------------------|
|              |                      |

| Μεταβλητές | <i>sessionFactory</i>  | SessionFactory |
|------------|--|----------------|
| Μέθοδοι    | <ul style="list-style-type: none"> <li>• <b><i>getCurrentSession()</i></b><br/>Επιστρέφει την ενεργή σύνδεση με τη βάση δεδομένων.</li> <li>• <b><i>beginTransaction()</i></b><br/>Εκκινεί συναλλαγή με τη βάση δεδομένων.</li> <li>• <b><i>commitTransaction ()</i></b><br/>Αποθηκεύει στη βάση δεδομένων τις αλλαγές που πραγματοποιήθηκαν κατά τη διάρκεια της συναλλαγής.</li> <li>• <b><i>rollbackTransaction ()</i></b><br/>Απορρίπτει τις αλλαγές που πραγματοποιήθηκαν κατά τη διάρκεια της συναλλαγής.</li> <li>• <b><i>closeTransaction ()</i></b><br/>Κλείνει τη συναλλαγή με τη βάση δεδομένων.</li> </ul> |                |

### 4.1.3. Σχήμα βάσης δεδομένων

Παρακάτω παρουσιάζονται τα αντικείμενα τα οποία περιγράφουν τα δεδομένα της βάσης δεδομένων. Αυτά έχουν τη μορφή Data Transfer Object καθότι χρησιμοποιείται το Hibernate για την επικοινωνία με τη βάση δεδομένων. Κάθε εικόνα περιέχει τις μεταβλητές των αντικειμένων/ στήλες των πινάκων. Ενώ, πάνω από τις μεταβλητές που αναφέρονται σε άλλο πίνακα προστίθενται τα κατάλληλα annotations που δηλώνουν τον τύπο της σχέσης.

#### ConditionDTO – Εγγραφή ιατρικού ιστορικού

```

@Entity
@Table(name="CONDITIONS")
public class ConditionDTO
{
    @Id
    @GeneratedValue
    private Long id;
    private Date dateOfCreation;
    private String conditionCode;
    private String conditionCategory;
    private String status;
    private String notes;

    @ManyToOne(cascade={CascadeType.PERSIST, CascadeType.MERGE})
    private UserDTO patient;

    @ManyToOne(cascade={CascadeType.PERSIST, CascadeType.MERGE})
    private UserDTO doctor;
}

```

Τα χαρακτηριστικά μίας εγγραφής ιατρικού ιστορικού είναι το μοναδικό αναγνωριστικό id της, η ημερομηνία δημιουργίας της εγγραφής dateOfCreation σε μορφή Date, η κατηγορία ιατρικού συμβάντος conditionCategory, όπως προδιαγράφεται από το πρότυπο HL7 FHIR π.χ. σύμπτωμα, ο κωδικός του ιατρικού συμβάντος conditionCode ο οποίος αντιστοιχεί στο

πρότυπο HL7 FHIR και δηλώνει π.χ. ορθοπεδικό ζήτημα, η κατάσταση status του συμβάντος π.χ. επιβεβαιωμένο, σημειώσεις notes για το συμβάν και τέλος ποιον ασθενή patient αφορά και ποιος ιατρός doctor το κατέγραψε.

### UserDTO – Χρήστης εφαρμογής

```
@Entity
@Table(name="USER")
public class UserDTO
{
    @Id
    @GeneratedValue
    private Long id;
    private String username;
    private String password;
    private String name;
    private String surname;
    private String emailAddress;
    private String currentId;
    private String amkaNumber;
    private String profilePic;
    private String gender;
    private Date birthDate;
    private String telephoneNumber;
    private String address;
```

Τα χαρακτηριστικά ενός χρήστη είναι το μοναδικό αναγνωριστικό id του, το όνομα χρήστη username και κωδικός χρήστη password, το όνομα name και επίθετο του surname, η διεύθυνση ηλεκτρονικού ταχυδρομείου του emailAddress, το προσωρινό του αναγνωριστικό currentId, ο αριθμός κοινωνικής ασφάλισης amkaNumber, η φωτογραφία του profilePic, το γένος gender και η ημερομηνία γέννησης του birthdate, το τηλέφωνο επικοινωνίας του telephoneNumber και τέλος η διεύθυνση του address.

### MedicationPrescriptionDTO – Εγγραφή φαρμακευτικής αγωγής

```
@Entity
@Table(name="MEDICATION_PRESCRIPTION")
public class MedicationPrescriptionDTO
{
    @Id
    @GeneratedValue
    private Long id;
    private Date dateOfCreation;
    private String status;
    private String reason;
    private String medication;
    private Integer quantity;
    private Integer ratePerPeriod;
    private Date startOfPeriod;
    private Date endOfPeriod;

    @ManyToOne(cascade={CascadeType.PERSIST, CascadeType.MERGE})
    private UserDTO patient;

    @ManyToOne(cascade={CascadeType.PERSIST, CascadeType.MERGE})
    private UserDTO doctor;
```

Τα χαρακτηριστικά μιας εγγραφής φαρμακευτικής αγωγής είναι το μοναδικό αναγνωριστικό id της, η ημερομηνία δημιουργίας της dateOfCreation, η κατάσταση της status π.χ. pending όπως την ορίζει το πρότυπο HL7 FHIR, η αιτιολογία reason για την συνταγογράφηση, η δραστική ουσία του φαρμάκου medication, η ποσότητα quantity σε κάθε χορήγηση του φαρμάκου, η συχνότητα χορήγησης του φαρμάκου για μια συγκεκριμένη χρονική περίοδο ratePerPeriod, και τέλος η αρχή και το τέλος της περιόδου startOfPeriod και endOfPeriod σε μορφή Date.

### CarePlanDTO – Εγγραφή πλάνου φροντίδας

```
@Entity
@Table(name="CARE_PLAN")
public class CarePlanDTO
{
    @Id
    @GeneratedValue
    private Long id;
    private String type;
    private String creationDate;
    private String notes;
    private String status;
    private String result;

    @ManyToOne(cascade={CascadeType.PERSIST, CascadeType.MERGE})
    private UserDTO patient;

    @ManyToOne(cascade={CascadeType.PERSIST, CascadeType.MERGE})
    private UserDTO doctor;
}
```

Τα χαρακτηριστικά μίας εγγραφής πλάνου φροντίδας είναι το μοναδικό αναγνωριστικό id της, ο τύπος type της ενέργειας, η ημερομηνία δημιουργίας της εγγραφής creationDate σε μορφή Date, η κατάσταση status της ενέργειας π.χ. εκτελεσμένη, σημειώσεις notes για την ενέργεια, το αποτέλεσμα result της ενέργειας και τέλος ποιον ασθενή patient αφορά και ποιος ιατρός doctor το ζήτησε.

### DiagnosticOrderDTO – Εγγραφή εργαστηριακής εξέτασης

```
@Entity
@Table(name="DIAGNOSTIC_ORDER")
public class DiagnosticOrderDTO
{
    @Id
    @GeneratedValue
    private Long id;
    private String type;
    private String creationDate;
    private String notes;
    private String status;

    @ManyToOne(cascade={CascadeType.PERSIST, CascadeType.MERGE})
    private UserDTO patient;

    @ManyToOne(cascade={CascadeType.PERSIST, CascadeType.MERGE})
    private UserDTO doctor;
}
```

Τα χαρακτηριστικά μίας εγγραφής εργαστηριακής εξέτασης είναι το μοναδικό αναγνωριστικό id της, ο τύπος type της εξέτασης, η ημερομηνία δημιουργίας της εγγραφής creationDate σε μορφή Date, η κατάσταση status της εξέτασης π.χ. εκτελεσμένη, σημειώσεις notes για την εξέταση και τέλος ποιον ασθενή patient αφορά και ποιος ιατρός doctor τη ζήτησε.

### DiagnosticReportDTO – Εγγραφή αποτελεσμάτων εργαστηριακής εξέτασης

```
@Entity
@Table(name="DIAGNOSTIC_REPORT")
public class DiagnosticReportDTO
{
    @Id
    @GeneratedValue
    private Long id;
    private String notes;
    private String creationDate;
    private String result;

    @ManyToOne(cascade={CascadeType.PERSIST, CascadeType.MERGE})
    private DiagnosticOrderDTO diagnosticOrder;

    @ManyToOne(cascade={CascadeType.PERSIST, CascadeType.MERGE})
    private UserDTO patient;

    @ManyToOne(cascade={CascadeType.PERSIST, CascadeType.MERGE})
    private UserDTO doctor;
```

Τα χαρακτηριστικά μίας εγγραφής αποτελεσμάτων εργαστηριακής εξέτασης είναι το μοναδικό αναγνωριστικό id της,, η ημερομηνία δημιουργίας της εγγραφής creationDate σε μορφή Date, σημειώσεις notes για την εκτέλεση της εξέτασης, το αποτέλεσμα result της ενέργειας, ποιον ασθενή patient αφορά και ποιος ιατρός doctor εκτέλεσε την εξέταση και τέλος ποια εργαστηριακή εξέταση αφορά.



# 5

## Υλοποίηση

Σε αυτό το κεφάλαιο παρουσιάζονται δείγματα από τον κώδικα της εφαρμογής AidIT, που χρήζουν ιδιαίτερου ενδιαφέροντος ως προς τη χρήση των τεχνολογιών που έχουν επιλεγεί στο κεφάλαιο 2, και μπορούν αν εκληφθούν ως κώδικας αναφοράς τόσο για το περιβάλλον του εξυπηρετητή, αλλά και για το περιβάλλον της κινητής συσκευής.

### 5.1. Υλοποίηση της εφαρμογής

#### *Δημιουργία εγγραφής ιατρικού ιστορικού (HL7 FHIR Condition) στην εφαρμογή Android*

Η παρακάτω μέθοδος δημιουργεί το HL7 μήνυμα που αναπαριστά μια εγγραφή ιατρικού ιστορικού σε XML μήνυμα, έτοιμο για μεταφορά στον εξυπηρετητή. Οι κλάσεις Condition αλλά και XMLComposer που χρησιμοποιούνται στον κώδικα, παρέχονται από το λογισμικό αναφοράς του HL7 FHIR και διευκολύνουν πολύ την δημιουργία του μηνύματος.

```
public static String createConditionResource(HealthInformation hi,
String patientId, String practitionerId)
{
    String toReturnStr = null;

    try
    {
        // create an empty FHIR condition
        Condition condition = new Condition();

        // set the subject of the condition
        condition.setSubject(
            new ResourceReference().setReferenceSimple(patientId));
    }
}
```

```

// set the practitioner as the asserter of the condition
condition.setAsserter(
    new ResourceReference().setReferenceSimple(practitionerId));
// set the type of the condition
condition.setAbatement(
    new CodeableConcept().setTextSimple(hi.getType()));

CodeableConcept category = new CodeableConcept();

Coding coding = category.addCoding();
coding.setSystemSimple("http://hl7.org/fhir/condition-category");
coding.setCodeSimple(hi.getCategory());

condition.setCategory(category);

// set the status of the condition
condition.setStatusSimple(
    ConditionStatus.valueOf(hi.getStatus()));

// set notes on the condition
condition.setNotesSimple(hi.getNotes());

// write response to xml
ByteArrayOutputStream baos = new ByteArrayOutputStream();

XmlComposer x = new XmlComposer();
x.compose(baos, condition, true);

toReturnStr = baos.toString();
}
catch (Exception ex)
{
    ex.printStackTrace();
}
return toReturnStr;
}

```

### Αποθήκευση εγγραφής ιατρικού ιστορικού (HL7 FHIR Condition) στη βάση δεδομένων

Η παρακάτω μέθοδος αποτελεί μέρος της διεπαφής REST που προδιαγράφει το πρότυπο HL7 FHIR. Πιο συγκεκριμένα, αναφέρεται στη δημιουργία μιας νέας εγγραφής στο ιατρικό ιστορικό του ασθενή. Η HTTP μέθοδος είναι POST και είναι προσβάσιμη στην τοποθεσία '/patients/conditions'. Η μέθοδος ξεκινάει με την εκκίνηση μιας νέας συναλλαγής με τη βάση δεδομένων, που πραγματοποιείται μέσω της βιβλιοθήκης hibernate. Ενώ, με χρήση του ConditionDTO και ConditionDAO δημιουργούνται τα αντικείμενα προς αποθήκευση στη σχεσιακή βάση. Αυτά, αποθηκεύονται με τη μέθοδο makePersistent και ακολούθως κλείνει η συναλλαγή με τη βάση δεδομένων.

```

@POST
@Path("/patients/conditions")
@Produces(MediaType.TEXT_PLAIN)
@Consumes(MediaType.TEXT_XML)

```



```

public Response createConditionResource(String postRequest)
{
    Response response = Response.status(500).build();
    // begin a new transaction with the relational db
    HibernateUtil.beginTransaction();

    try
    {
        // create an XMLParser to parse the HL7 FHIR Condition message
        XmlParser parser = new XmlParser();

        ByteArrayInputStream bais =
            new ByteArrayInputStream(postRequest.getBytes());

        Resource r = parser.parse(bais);

        Condition condition = (Condition) r;

        // create a new ConditionDTO
        ConditionDTO conditionDTO = new ConditionDTO();

        // set the creation date of the condition -> now
        conditionDTO.setDateOfCreation(new Date());

        // set the code of the condition
        conditionDTO.setConditionCode(condition.getCode().getTextSimple());

        // set the category of the condition
        conditionDTO.setConditionCategory(
            condition.getCategory().getCoding().get(0).getCodeSimple());

        // set the status of the condition
        conditionDTO.setStatus(condition.getStatus().getValue().name());

        // set the notes on the condition
        conditionDTO.setNotes(condition.getNotesSimple());

        String doctorCurrentId = condition.getAsserter().getReferenceSimple();
        String patientCurrentId = condition.getSubject().getReferenceSimple();

        UserDAO userDAO = new UserDAO();

        // set the doctor that examined the patient and the patient
        conditionDTO.setPatient(userDAO.findByCurrentId(patientCurrentId));
        conditionDTO.setDoctor(userDAO.findByCurrentId(doctorCurrentId));

        // save the new condition
        conditionDTO = new ConditionDAO().makePersistent(conditionDTO);

        // save the transaction with the relational db
        HibernateUtil.commitTransaction();

        response = Response.status(200).entity(
            conditionDTO.getId().toString()).build();
    }
    catch (Exception ex)
    {
        HibernateUtil.rollbackTransaction();
    }
}

```

```

    ex.printStackTrace();
}

// close the transaction with the relational db
HibernateUtil.closeTransaction();

return response;
}

```

### *Επεξεργασία στην εφαρμογή Android μηνυμάτων HL7 FHIR MedicationPrescription με τη φαρμακευτική αγωγή του ασθενή*

Η μέθοδος επεξεργάζεται τη λίστα των φαρμακευτικών αγωγών του ασθενή, ενεργών ή μη, και την μετατρέπει στο μοντέλο δεδομένων της εφαρμογής, το οποίο είναι δεμένο με την εφαρμογή. Το αντικείμενο XmlParser αναλαμβάνει στην επεξεργασία των XML μορφής των εισερχόμενων δεδομένων, τα οποία έρχονται στην AtomFeed.

```

public static List<PharmacyOrderHistory> parseGetMedicationsResponse(String
serverResponse)
{
    List<PharmacyOrderHistory> medicationPrescriptions =
        new ArrayList<PharmacyOrderHistory>();
    try
    {
        // initialize the xml parser
        XmlParser parser = new XmlParser();

        ByteArrayInputStream bais =
            new ByteArrayInputStream(serverResponse.getBytes());

        // parse xml fragment
        ResourceOrFeed bundle = parser.parseGeneral(bais);

        // create a new object for each medication prescription
        for (AtomEntry bundleEntry : bundle.getFeed().getEntryList())
        {
            MedicationPrescription medPresc = (MedicationPrescription)
                bundleEntry.getResource();

            // date of the prescription creation
            String dateWritten =
                medPresc.getDateWrittenSimple().toHumanDisplay();

            // the prescribing doctor
            String prescriber = medPresc.getPrescriber().getDisplaySimple();

            // the medication label
            String medication = medPresc.getMedication().getDisplaySimple();

            // status of the prescription - e.g. pending
            String status = medPresc.getStatusSimple().name();

            // unique id of the prescription
            String id = medPresc.getIdentifier().get(0).getValueSimple();

            // reason of the prescription

```

```

String reason = ((String_) medPresc.getReason()).getValue();

MedicationPrescriptionDosageInstructionComponent dosInstr =
    medPresc.getDosageInstruction().get(0);

// per administration quantity
int quantity = Integer.valueOf(
    dosInstr.getDoseQuantity().getValue().getStringValue());

// daily medication administrations
int rate = Integer.valueOf(
    dosInstr.getRate().getNumerator().getValue().getStringValue());

// period of the therapy
Period period = (Period) dosInstr.getTiming();
String startDateStr = period.getStartSimple().toHumanDisplay();

String endDateStr = period.getEndSimple().toHumanDisplay();

// creation of new entry
PharmacyOrderHistory pharmacyOrderHistory =
    new PharmacyOrderHistory();

pharmacyOrderHistory.setNotes(reason);

pharmacyOrderHistory.setStartOfPeriod(startDateStr);

pharmacyOrderHistory.setEndOfPeriod(endDateStr);

pharmacyOrderHistory.setRatePerPeriod(rate);

pharmacyOrderHistory.setPrescriberName(prescriber);

pharmacyOrderHistory.setQuantity(quantity);

pharmacyOrderHistory.setStatus(status);

pharmacyOrderHistory.setPharmacyDate(dateWritten);

pharmacyOrderHistory.setPharmacyHistoryID(id);

pharmacyOrderHistory.setType(medication);

medicationPrescriptions.add(pharmacyOrderHistory);
}
}
catch (Exception ex)
{
    ex.printStackTrace();
}
return medicationPrescriptions;
}

```

## Σάρωση QR-Code στην εφαρμογή Android

Ο κώδικας που παρατίθεται παρακάτω αφορά στη σάρωση των αναγνωριστικών των χρηστών και την επιστροφή του στην εφαρμογή. Η μέθοδος `onOptionsItemSelected` καλείται κατά το πάτημα των κουμπιών στην κορυφή της οθόνης της εφαρμογής. Σε περίπτωση που πατηθεί το κουμπί που αφορά τη σάρωση του κουμπιού, τότε δημιουργείται μια αίτηση με σκοπό (Intent) τη σάρωση. Η δημιουργία του αιτήματος επεξεργάζεται από το λειτουργικό Android και επιστρέφει πιθανές εφαρμογές που να ικανοποιούν το συγκεκριμένο αίτημα. Στην περίπτωση μας η βιβλιοθήκη ZXing είναι ενσωματωμένη στην εφαρμογή και αναλαμβάνει την σάρωση του αναγνωριστικού, επιστρέφοντας το στη μέθοδο `onActivityResult`.

Η μέθοδος `onActivityResult` ελέγχει εάν ακυρώθηκε η διαδικασία της σάρωσης, κι εάν επιστραφεί το αναγνωριστικό ζητάει τον ιατρικό πληροφορίες του ασθενή από τον εξυπηρετητή και εμφανίζει τη σχετική οθόνη.

```
public boolean onOptionsItemSelected(MenuItem item)
{
    switch (item.getItemId())
    {
        case R.id.action_scan:
            // creation of an intent for the capture activity
            Intent intent = new Intent(this, CaptureActivity.class);
            intent.setAction(Intent.ACTION_SCAN);
            intent.putExtra("SCAN_MODE", "QR_CODE_MODE");
            startActivityForResult(intent, 0);

            return true;

        default:
            return super.onOptionsItemSelected(item);
    }
}

public void onActivityResult(int requestCode, int resultCode, Intent intent)
{
    // if scan was successful
    if (resultCode == RESULT_OK)
    {
        // id of patient
        String patientId = intent.getStringExtra("SCAN_RESULT");

        try
        {
            // retrieve patient info
            PatientInfoAsyncTask patientInfoAsyncTask =
                new PatientInfoAsyncTask();
            patientInfoAsyncTask.execute(patientId);

            patient = new User();
            patient = patientInfoAsyncTask.get();
        }
    }
}
```

```

catch(Exception ex)
{
    ex.printStackTrace();
}

displayView(1);
}
else if (resultCode == RESULT_CANCELED)
{
    // if scan ended with errors or cancelled, show alert dialog
    AlertDialog alertDialog = new AlertDialog.Builder(this).create();
    alertDialog.setTitle("Error:");
    alertDialog.setMessage("Patient not Found");
    alertDialog.show();
}
}
}

```

## 5.2. Παράδειγμα χρήσης

Στην παράγραφο αυτή θα περιγράψει ένα σενάριο χρήσης της πλατφόρμας AidIT, για να γίνει πιο εύκολα η κατανόηση των λειτουργιών και της χρησιμότητάς της. Θα αναλύσουμε το πως μπορούν να αλληλεπιδράσουν οι διάφοροι χρήστες και πως μπορούν να χρησιμοποιήσουν την πλατφόρμα για να καλύψουν τις ανάγκες τους. Αρχικά να πούμε πως η πλατφόρμα AidIT αποτελείται από δύο Android εφαρμογές μία εφαρμογή για το κινητό τηλέφωνο (smartphone) του ασθενή και μία για το νοσοκομειακό προσωπικό.

Το σενάριο που μελετάμε αφορά στην ξαφνική, απρογραμματίστη επίσκεψη ενός ασθενή στα εξωτερικά ιατρεία ενός νοσοκομείου. Υπάρχουν πέντε χαρακτήρες-ρόλοι: ο ασθενής, ο ιατρός διαλογής, ο θεράπων ιατρός, το βιοχημικό εργαστήριο του νοσοκομείου και ένας φαρμακοποιός. Επιγραμματικά, τα βήματα που ακολουθούνται είναι τα εξής:

- 1) υποδοχή του ασθενή και διενέργεια διαλογής ασθενών/ανάθεση σε θεράποντα ιατρό
- 2) εξέταση από τον γιατρό και παραπομπή για περαιτέρω εργαστηριακή εκτίμηση
- 3) εργαστηριακές εξετάσεις και εκτίμηση των αποτελεσμάτων τους
- 4) τελική διάγνωση και θεραπεία/ιατρική συνταγή
- 5) εκτέλεση της συνταγής από φαρμακοποιό
- 6) παρακολούθηση της κατάστασης του ασθενή για κάποιο, ορισμένο χρονικό διάστημα.

### 5.2.1. Αρχικά δεδομένα

Αρχικά θα πρέπει όλοι οι χαρακτήρες να είναι εγγεγραμμένοι και συνδεδεμένοι στις αντίστοιχες εφαρμογές τους, ανάλογα με τον ρόλο που έχουν. Επίσης θεωρούμε ότι ο

φάκελος του ασθενή προϋπάρχει στο σύστημα του νοσοκομείου και δεν χρειάζεται να δημιουργηθεί εκείνη την στιγμή.

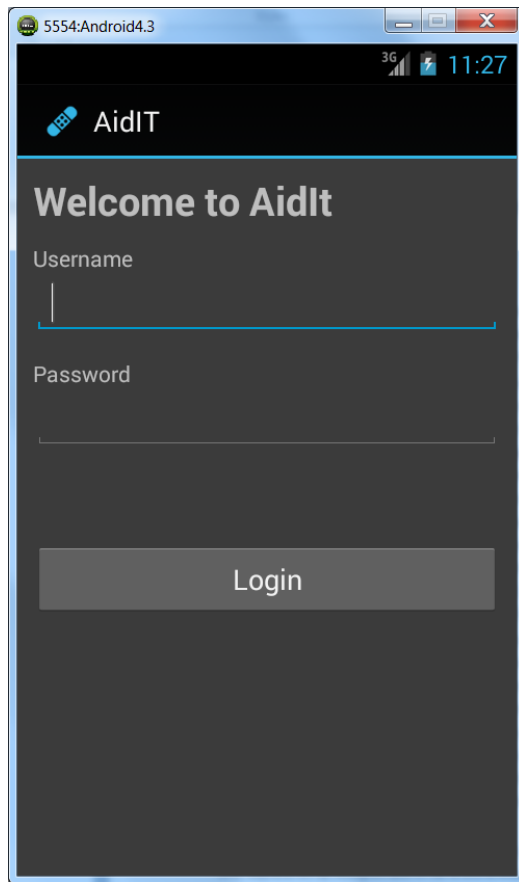
Πιο συγκεκριμένα, έχουμε τους παρακάτω χρήστες:

| Ρόλος           | Στοιχεία  |
|-----------------|---|
| Ασθενής         | <b>Spyros Pappas</b><br>Date of Birth: 14/02/1985<br>Address: Seferi 12, Zografou, Athens<br>Social No: 1145236545<br>Email: s.pappas@gmail.com<br>Phone: 6932333333        |
| Ιατρός Διαλογής | <b>Maria Kogioglou</b><br>Date of Birth: 14/02/1970<br>Address: Panepistimiou 123, Athens<br>Social No: 1145236545<br>Email: secretary@gmail.com<br>Phone: 6932333333       |
| Θεράπων Ιατρός  | <b>Ioanna Chatzimichali</b><br>Date of Birth: 29/01/1960<br>Address: Emmanouil Mpenaki 12, Kifisia<br>Social No: 1145236545<br>Email: doctor@gmail.com<br>Phone: 6932333333 |

|   |   |
|---|---|
| <p>Ιατρός Εργαστηριακών<br/>Εξετάσεων</p> | <p><b>George Papanikolaou</b><br/> Date of Birth: 11/09/1968<br/> Address: Patision 123, Patisia, Athens<br/> Social No: 1234543253<br/> Email: g.papanikolaou@gmail.com<br/> Phone: 6932000000</p> |
| <p>Φαρμακοποιός</p>                       | <p><b>George Karakostas</b><br/> Date of Birth: 17/07/1954<br/> Address: Hroon Politexneiou 11, Zograf...<br/> Social No: 1234566456<br/> Email: pharmacist@gmail.com<br/> Phone: 6932543540</p>    |

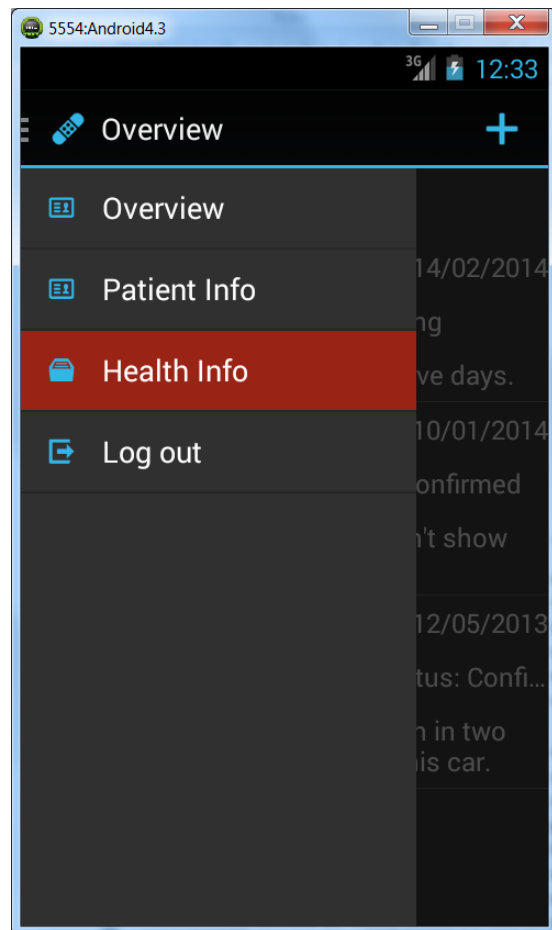
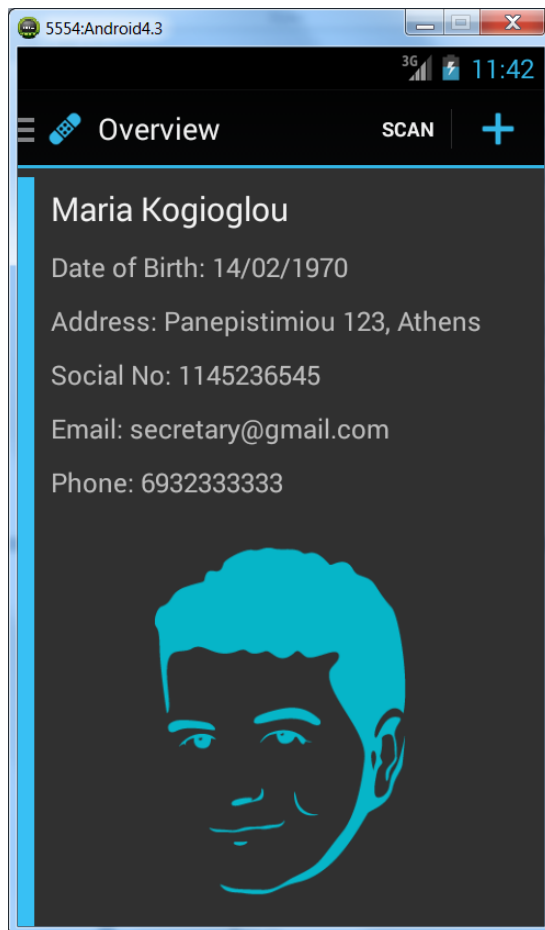
### 5.2.2. Εκτέλεση σεναρίου

Το αρχικό βήμα είναι η επίσκεψη του ασθενή κ. Σπύρου Παππά στο νοσοκομείο με κάποια συμπτώματα π.χ. πόνος στο χέρι. Ο ασθενής συνδέεται στην εφαρμογή και δείχνει το προσωρινό αναγνωριστικό του στον ιατρό που διενεργεί τη διαδικασία της διαλογής πατώντας το κουμπί ‘MYID’.

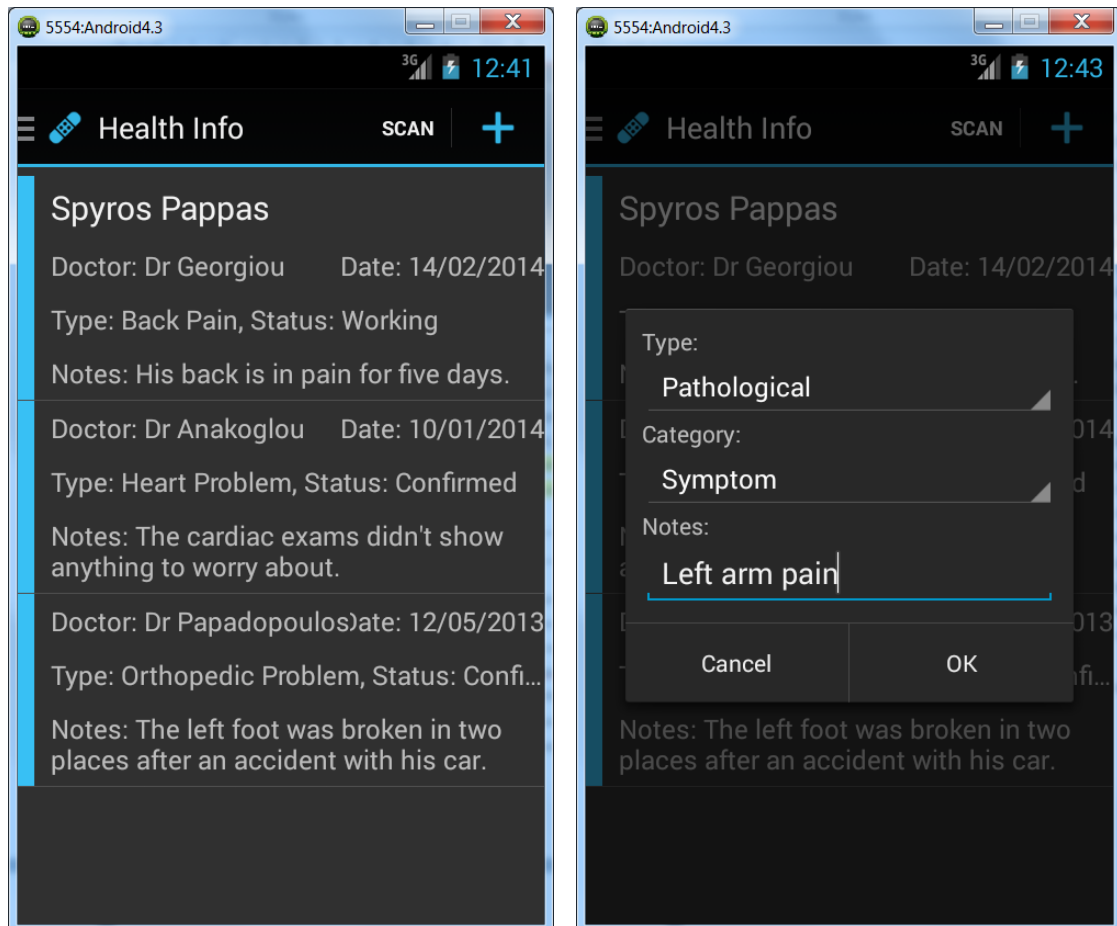


Η ιατρός κ. Μαρία Καγιόγλου που διενεργεί το triage των ασθενών είναι επίσης συνδεδεμένη στην εφαρμογή και σαρώνοντας το αναγνωριστικό του ασθενή φέρνει στην δικιά της οθόνη τον φάκελο του ασθενή. Με βάση τον ρόλο της ωστόσο μπορεί να επεξεργαστεί μονάχα τις καρτέλες που περιέχουν τις γενικές πληροφορίες του ασθενή και το γενικό ιστορικό του. Η παραπάνω διαδικασία αποτελεί ουσιαστικά την συγκατάθεση του ασθενή για την χρήση των προσωπικών, ιατρικών του δεδομένων από το νοσοκομείο.

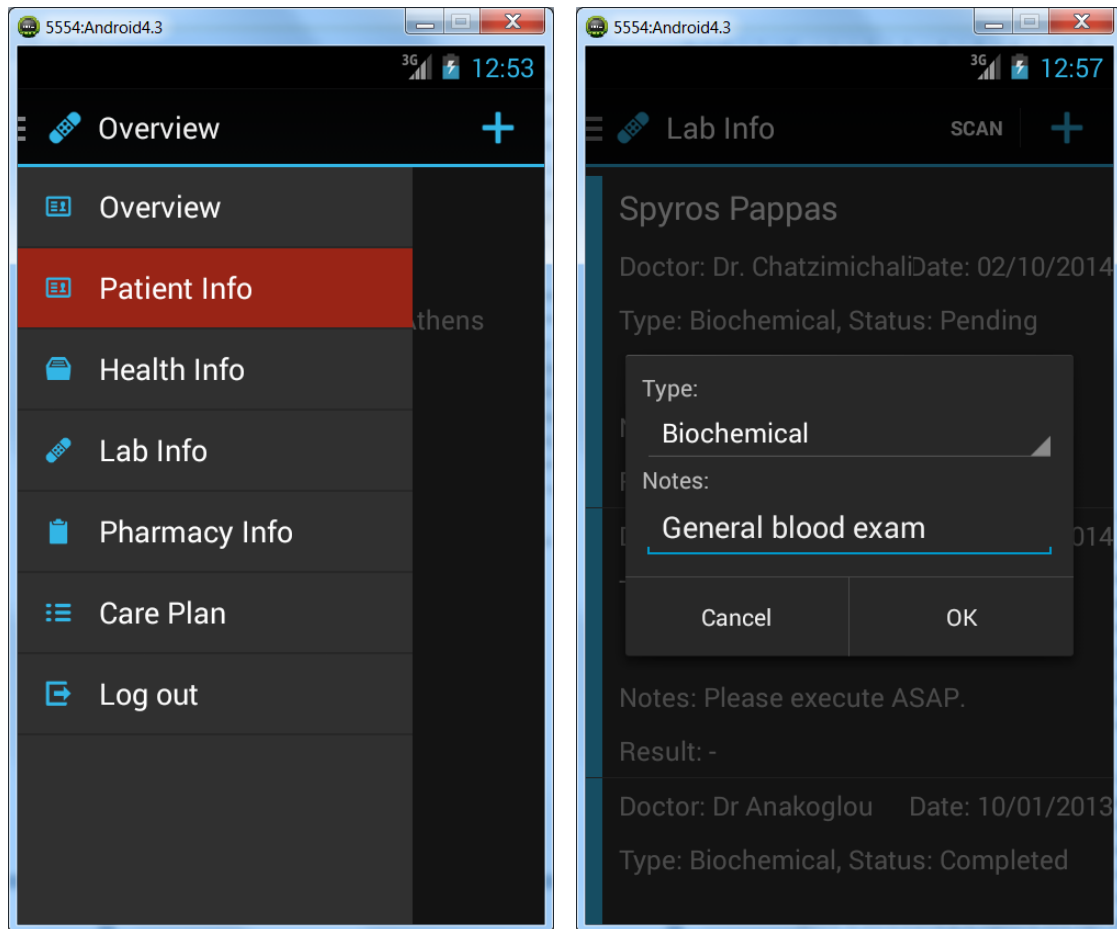




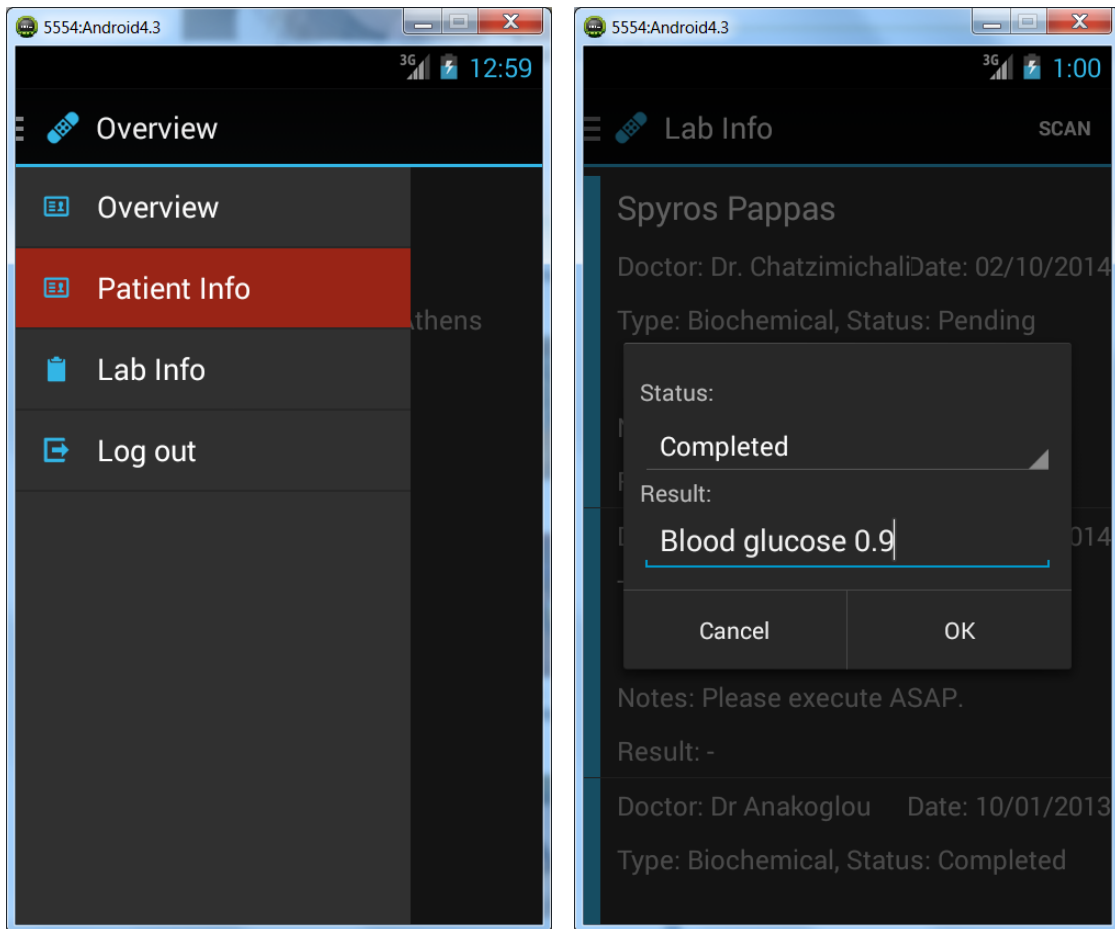
Στην συνέχεια η ιατρός διαλογής σημειώνει στην καρτέλα με το γενικό ιστορικό του ασθενή, τον λόγο της επίσκεψης και τα συμπτώματα και κάνει την λεγόμενη διαλογή/triage· εκτιμά δηλαδή τα συμπτώματα και με βάση το ιατρικό ιστορικό του ασθενή, τον παραπέμπει/αναθέτει στον θεράποντα ιατρό ανάλογης ειδικότητας.



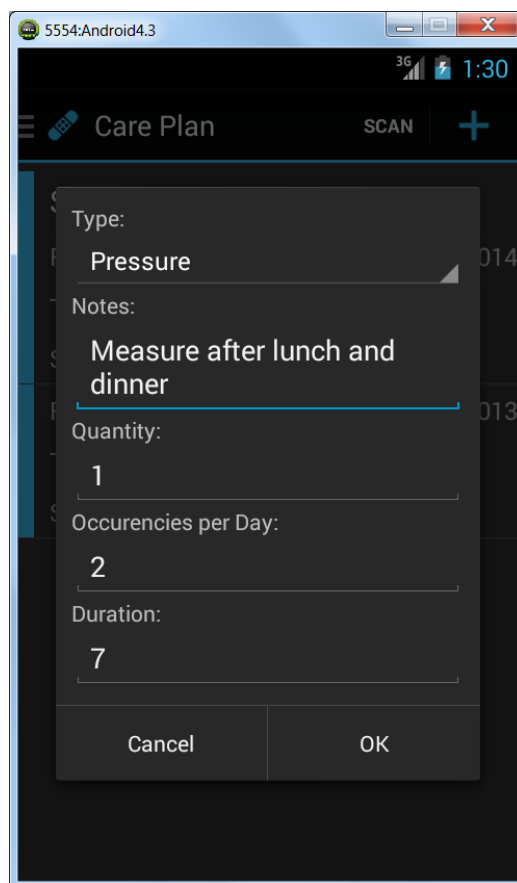
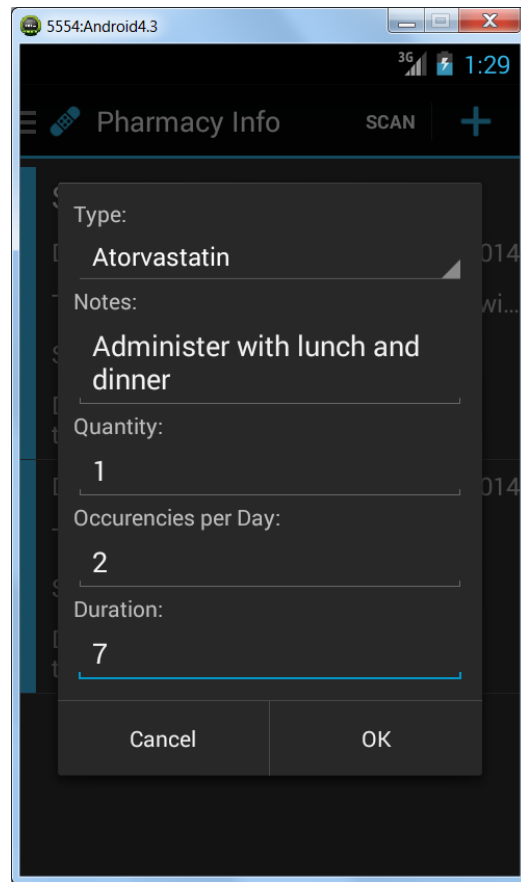
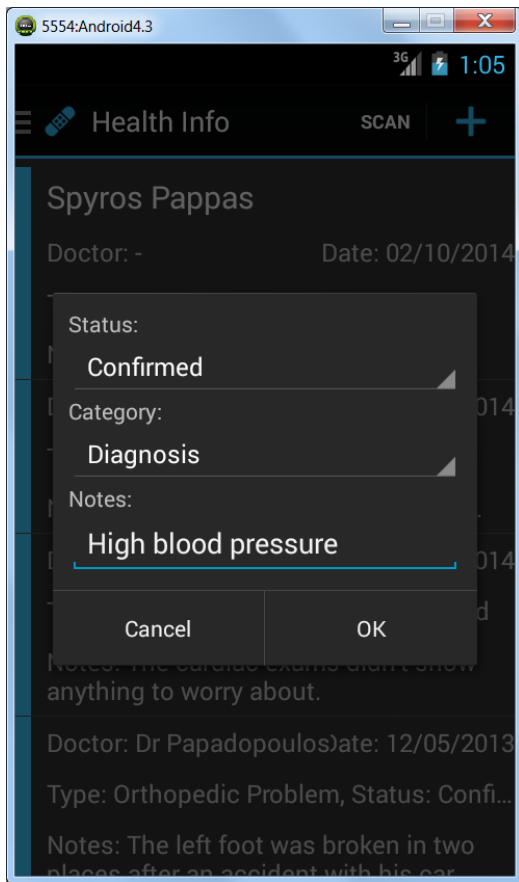
Σο δεύτερο στάδιο πραγματοποιείται η κλινική εξέταση του ασθενή από τη θεράποντα ιατρό κα. Ιωάννα Χατζημιχάλη. Ο ασθενής δείχνει το αναγνωριστικό του σε μορφή QR code στην ιατρό, η οποία σαρώνοντάς το λαμβάνει στην οθόνη τον φάκελο του ασθενή. Η ιατρός έχει πρόσβαση σε κάθε καρτέλα του ιατρικού φακέλου. Μπορεί να δει τα γενικά στοιχεία του ασθενή/demographics, χωρίς όμως να μπορεί να τα τροποποιήσει. Μπορεί να δει το γενικό ιστορικό του ασθενή και να προσθέσει μία διάγνωση, μπορεί να δει το ιστορικό ιατρικών εξετάσεων και να παραπέμψει τον ασθενή στο εργαστήριο για εξετάσεις, μπορεί να δει το ιστορικό φαρμακευτικών αγωγών του ασθενή και να προσθέσει μία ιατρική συνταγή ενώ τέλος μπορεί να ορίσει παραμέτρους για την παρακολούθηση του ασθενή από απόσταση. Στο σενάριο μας, η ιατρός δεν μπορεί να κάνει άμεσα διάγνωση και χρειάζεται κάποιες βιοχημικές εξετάσεις πχ γενική αίματος. Στην καρτέλα του εργαστηρίου ορίζει ποιες εξετάσεις χρειάζεται και παραπέμπει τον ασθενή.



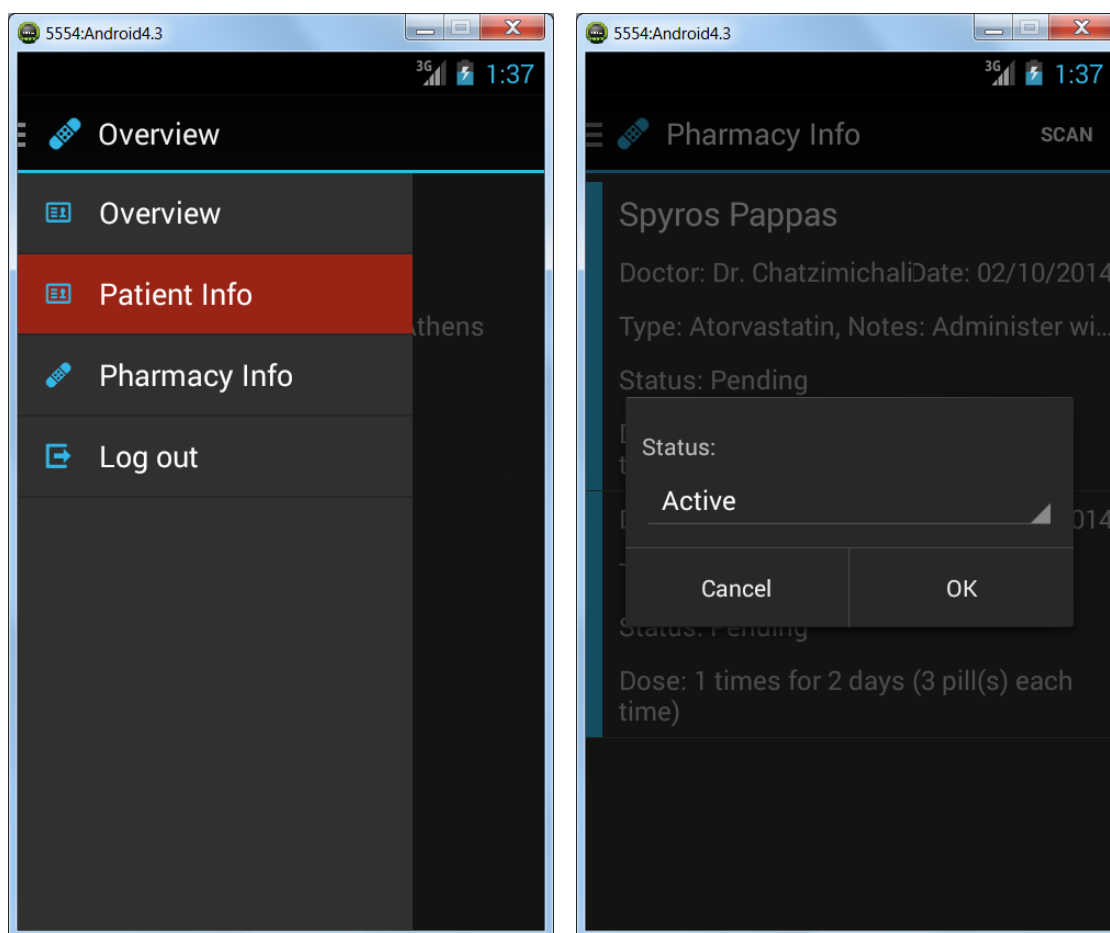
Επόμενο βήμα είναι η επίσκεψη του ασθενή στο εργαστήριο του Δρ. Γιώργου Παπανικολάου. Δείχνει το QR code του και εμφανίζεται στην οθόνη του συστήματος του εργαστηρίου ο φάκελος του ασθενή. Το εργαστήριο μπορεί να δει την καρτέλα με τα γενικά στοιχεία του ασθενή (χωρίς να μπορεί να τα τροποποιήσει) και την καρτέλα του εργαστηρίου. Στην καρτέλα του εργαστηρίου φαίνονται όλες οι ιατρικές εντολές για εξετάσεις και η κατάστασή τους (ολοκληρωμένες, σε αναμονή, ακυρωμένες). Ο τεχνικός του εργαστηρίου βλέπει την εντολή του γιατρού για γενική αίματος, κάνει τις ανάλογες εξετάσεις και περνά τα αποτελέσματά τους στο σύστημα. Πλέον η εξέταση θεωρείται ολοκληρωμένη και τα αποτελέσματά της φαίνονται στο εργαστηριακό ιστορικό του ασθενή.



Στην συνέχεια ο ασθενής πηγαίνει ξανά στη θεραπεία του, δείχνει το QR code του και πλέον στην καρτέλα του εργαστηρίου φαίνονται τα αποτελέσματα των εξετάσεων. Η ιατρός μπορεί τώρα να κάνει ασφαλή διάγνωση, την οποία καταχωρεί στο γενικό ιστορικό και να προτείνει θεραπεία. Στην συνέχεια στην καρτέλα του φαρμακείου καταχωρεί μία ιατρική συνταγή για φάρμακα. Τέλος στην καρτέλα παρακολούθησης ορίζει κάποιες παραμέτρους-μετρήσεις, τις οποίες ο ασθενής οφείλει να τηρεί πχ να μετράει την πίεσή του 3 φορές την ημέρα και να εισάγει τα αποτελέσματα ή να σημειώνει πότε πήρε τα φάρμακά του. Σε περίπτωση που οι τιμές ξεπεράσουν κάποια καθορισμένα όρια πχ κάποιες τιμές για την πίεση, το συμβάν αυτό προστίθεται στο γενικό ιστορικό.

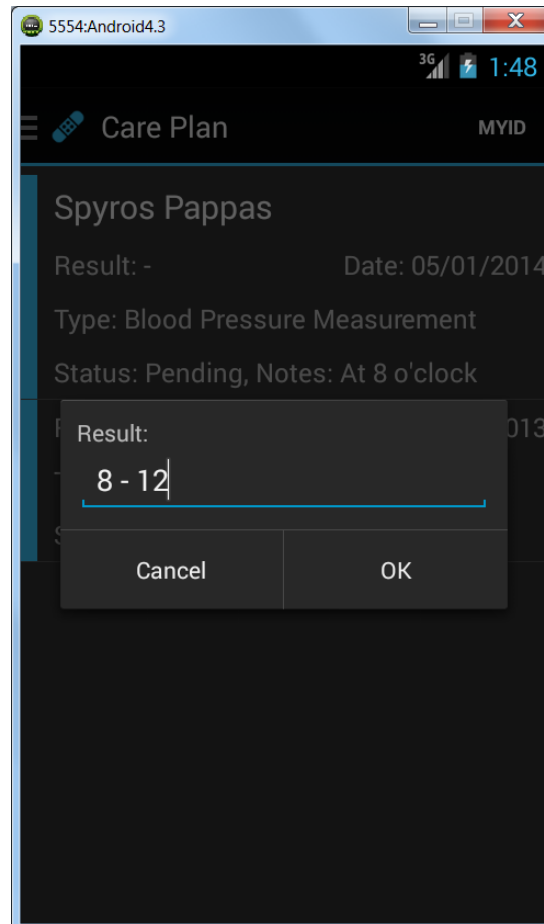


Επόμενο βήμα η επίσκεψη στο φαρμακείο του νοσοκομείου. Ο φαρμακοποιός κ. Γιώργος Καρακώστας σαρώνοντας το αναγνωριστικό του ασθενή, λαμβάνει στην οθόνη του συστήματος τον ιατρικό φάκελό του. Τα δικαιώματα του φαρμακοποιού μοιάζουν με τα δικαιώματα του εργαστηρίου. Ο φαρμακοποιός μπορεί να δει την καρτέλα με τα γενικά στοιχεία του ασθενή και την καρτέλα φαρμακείου. Στην καρτέλα φαρμακείου φαίνεται όλο το φαρμακευτικό ιστορικό του ασθενή καθώς και όλες οι συνταγές που έχουν εκδοθεί στο όνομα του ασθενή και η κατάστασή τους (εκτελεσμένες, ανεκτέλεστες/σε αναμονή, ακυρωμένες). Ο φαρμακοποιός βρίσκει τη συνταγή του γιατρού, δίνει το φάρμακο στον ασθενή και σημειώνει την συνταγή σαν εκτελεσμένη. Σε περίπτωση που δεν είχε το ζητούμενο φάρμακο, θα μπορούσε να βάλει την συνταγή σε αναμονή.



Στο τελευταίο βήμα πραγματοποιείται η εκτέλεση της θεραπείας που έχει προδιαγράψει θεράπων ιατρός και αφορά τόσο την έγκαιρη λήψη των φαρμάκων που συνιστούν τη φαρμακευτική αγωγή, αλλά και την εκτέλεση των οδηγιών του ιατρού σχετικά με το πλάνο φροντίδας του ασθενή. Το τελευταίο, ενδέχεται να περιλαμβάνει πέρα από μετρήσεις υγείας τις οποίες μπορεί να πραγματοποιήσει μόνος του ο ασθενής από το σπίτι, αλλά και να θέτει ημερήσιους ή εβδομαδιαίους στόχους, όπως η απώλεια βάρους ή 1 ώρα περπάτημα την

ημέρα. Όλα τα παραπάνω, ομαδοποιούνται στην καρτέλα πλάνου φροντίδας ως ατομικές ενέργειες ταξινομημένες με την πιο κοντινή ενέργεια που πρέπει να εκτελέσει. Για παράδειγμα, εάν η φαρμακευτική αγωγή απαιτεί την λήψη ενός φαρμάκου μαζί με κάθε γεύμα για μια εβδομάδα, τότε αυτά μεταφράζονται ως δύο εγγραφές κάθε μέρα στο πλάνο φροντίδας, τις οποίες και πρέπει να ορίσει ως εκτελεσμένες. Παρακάτω ο ασθενής ακολουθεί την οδηγία του ιατρού του και μετράει την πίεση του.







# 6

## Επίλογος

Το κεφάλαιο αυτό περιλαμβάνει τα συμπεράσματα που προκύπτουν από τη διπλωματική εργασία και σκιαγραφεί τις πιθανές μελλοντικές επεκτάσεις που θα αποτελούσαν τους άξονες της συνέχισης της εργασίας έχοντας ως βάση την πλατφόρμα AidIT.

### 6.1. Συμπεράσματα

Κατά τη διάρκεια εκπόνησης της διπλωματικής εργασίας εξήχθησαν χρήσιμα συμπεράσματα σχετικά με τις παραμέτρους που πρέπει να λάβει κάποιος υπόψη κατά τη δημιουργία ιατρικών πληροφοριακών συστημάτων. Παρακάτω παρουσιάζονται οι βασικοί άξονες μαζί με τις προσεγγίσεις που ακολούθησε η διπλωματική εργασία, οι οποίες είχαν ως στόχο την ικανοποίηση των απαιτήσεων και τη δημιουργία μιας πλήρους και ευέλικτης πλατφόρμας.

#### **Αρχιτεκτονική**

Η αρχιτεκτονική στηρίζεται σε έναν κεντρικό εξυπηρετητή. Ο εξυπηρετητής συγκεντρώνει τα δεδομένα από την εφαρμογή των ασθενών και τα διανέμει στην εφαρμογή των γιατρών. Η διαδικασία αυτή διαχωρίζει την συλλογή από την επεξεργασία των δεδομένων. Η ιατρική εφαρμογή μπορεί να ζητά πληροφορίες από τον Server είτε να ενημερώνεται για τυχόν αλλαγές. Η εφαρμογή των ασθενών αποθηκεύει και επεξεργάζεται τα δεδομένα για τον συγκεκριμένο ασθενή (τεχνικά εφικτό αφού μιλάμε για εφαρμογή smartphone).

### **Διαθεσιμότητα**

Η πλατφόρμα AidIT προϋποθέτει ότι

- τα τηλεπικοινωνιακά δίκτυα είναι σταθερά
- δεν θα γίνονται εκτεταμένες αλλαγές σε αυτά κατά την διάρκεια χρήσης της πλατφόρμας
- τα δίκτυα αυτά είναι σύμφωνα με τις ευρωπαϊκές προδιαγραφές.

### **Επεκτασιμότητα**

Η πλατφόρμα δεν σχεδιάστηκε με γνώμονα ένα σενάριο χρήσης αλλά πολλά διαφορετικά. Έτσι δύνονται πολλές δυνατότητες για επεκτασιμότητα:

- Για ασθενείς με καρδιακό ιστορικό πχ παρακολουθείται η πίεση, το βάρος και οι καρδιακοί παλμοί.
- Ερωτηματολόγια για την γενικότερη κατάσταση της υγείας του ασθενή
- Ενημερώσεις για συμπτώματα γύρω απ' την ασθένεια, νέα φάρμακα κλπ.
- Χρήση βιομετρικών αλγορίθμων για διάφορες μετρήσεις πχ για πίεση και καρδιακούς παλμούς

### **Εξατομίκευση**

Ο κάθε ασθενής έχει διαφορετικούς περιορισμούς που στηρίζονται σε διάφορες παραμέτρους: σοβαρότητα της πάθησης, ηλικία, ιατρικό ιστορικό, συνυπάρχουσες ασθένειες κλπ. Η πλατφόρμα λοιπόν πρέπει να μπορεί να παραμετροποιηθεί ανάλογα με την κατάσταση του ασθενούς. Η παραμετροποίηση αφορά στη δυνατότητα επιλογής των μετρήσεων ανάλογα με την κατάσταση της υγείας του ασθενή.

### **Χρηστικότητα**

Πρέπει να ληφθεί υπόψη ότι η εφαρμογή απευθύνεται σε ανθρώπους οι οποίοι δεν έχουν ιδιαίτερη σχέση με υπολογιστές. Θα πρέπει λοιπόν η εφαρμογή να είναι ιδιαίτερα φιλική προς τον χρήστη και να τον καθοδηγεί στις διάφορες διαδικασίες. Η ευκολία στην χρήση είναι το κλειδί για τη χρήση ή όχι της εφαρμογής από τους ασθενείς.

### **Ειδοποιήσεις**

Η πλατφόρμα θα πρέπει να έχει μηχανισμούς οι οποίοι να μπορούν να εντοπίζουν προβλήματα που μπορούν να θέσουν σε κίνδυνο τον ασθενή. Πχ η πλατφόρμα μπορεί να

εντοπίσει αν κάποιες μετρήσεις ξεφεύγουν απ τα όρια, αν κάποιος ασθενής αναφέρει χειροτέρευση της κατάστασής του κλπ.

### **Ασφάλεια**

Κανένα προσωπικό δεδομένο δεν πρέπει να αποθηκεύεται στην ιατρική εφαρμογή. Τα δεδομένα πριν σταλούν πρέπει να κρυπτογραφούνται.

### **Διαλειτουργικότητα**

Συμβατότητα με τις ήδη υπάρχουσες κλινικές διαδικασίες και συστήματα μέσω της χρήσης διεθνών προτύπων.

## **6.2. Μελλοντικές επεκτάσεις**

Οι πιθανές μελλοντικές επεκτάσεις της διπλωματικής εργασίας δεν αφορούν την ορθή λειτουργία του συστήματος, αλλά αναγκαίες επεκτάσεις ώστε να η πλατφόρμα να αποτελεί ένα πλήρως λειτουργικό σύστημα στον πραγματικό κόσμο

### **Σύστημα Αναμονής**

Ένα επαρκές σύστημα αναμονής είναι απαραίτητο σε κάθε σύγχρονη υγειονομική μονάδα και γενικώς σε κάθε οργανισμό, ο οποίος διαχειρίζεται ένα πλήθος περιστατικών/ανθρώπων/ υποθέσεων σε καθημερινή βάση. Ένα αναμονητικό σύστημα θα πρέπει να μπορεί να κάνει τις παρακάτω λειτουργίες:

- Θα πρέπει να διασυνδέεται με την γραμματεία ώστε να ρυθμίζει το πρόγραμμα των ραντεβού/συναντήσεων.
- Θα πρέπει να διασυνδέεται με κάθε νοσοκομειακή υπηρεσία/ιατρείο και να ρυθμίζει τις ουρές αναμονής.
- Θα πρέπει επίσης να καθορίζει προτεραιότητα.

Σε ένα νοσοκομειακό περιβάλλον, όπου υπάρχουν περιστατικά διαφορετικής σοβαρότητας (από επείγοντα περιστατικά με κίνδυνο ζωής έως τυπικές επισκέψεις στον γιατρό), θα πρέπει να δίνεται η δυνατότητα παράκαμψης της αναμονής όταν αυτό κρίνεται απαραίτητο. Ένας παρόμοιος μηχανισμός συμπεριλαμβάνεται και στον δικό μας σχεδιασμό και δεν είναι άλλος από την διαλογή/triage που γίνεται κατά την είσοδο του ασθενή στο νοσοκομείο. Σε ένα σύστημα αναμονής ο μηχανισμός αυτός θα αναβαθμιστεί και θα μπορεί ουσιαστικά να καθορίζει την σειρά με την οποία θα συναντάται κάθε ασθενής με τον γιατρό του.

## **Κοστολόγηση**

Ένα απαιτούμενο χαρακτηριστικό που θα πρέπει να έχει μία μελλοντική επέκταση της εφαρμογής, είναι η δυνατότητα κοστολόγησης των ιατρικών υπηρεσιών και παροχών. Ένα σύστημα κοστολόγησης:

- Θα πρέπει να μπορεί να “βλέπει” όλες τις ιατρικές πράξεις που πραγματοποιούνται σε κάποιον ασθενή κατά την διάρκεια της παραμονής του στην μονάδα υγείας και να τις κοστολογεί. Με άλλα λόγια για κάθε συνάντηση με γιατρό, εξέταση, φαρμακευτική συνταγή και γενικώς για οποιαδήποτε ιατρική πράξη εισάγεται στον φάκελο του ασθενή, θα υπάρχει αυτόματα και η αντίστοιχη χρέωση (με βάση κάποιο τιμοκατάλογο που παρέχεται από τον οργανισμό).
- Κατά την έξοδο του ασθενή από το νοσοκομείο, θα γίνεται συμψηφισμός όλων των χρεώσεων και θα εκδίδεται τελικός λογαριασμός. (ο οποίος θα επιβαρύνει είτε το ασφαλιστικό ταμείο του ασθενή, είτε τον ίδιο αν το επιλέξει)
- Το σύστημα κοστολόγησης θα μπορούσε να διασυνδεθεί και με το λογιστήριο του νοσοκομείου, ώστε να υπάρχει ανά πάσα στιγμή άμεση ενημέρωση στις καταστάσεις εσόδων/εξόδων, ενώ τέλος
- Θα μπορεί να διασυνδεθεί και με ασφαλιστικούς οργανισμούς, ώστε να μπορούν να δίδονται άμεσα εγκρίσεις για ιατρικές πράξεις και για να υπάρχει μία συνολική εικόνα για το κόστος φροντίδας του κάθε ασφαλισμένου.

## **Θάλαμοι Νοσηλείας**

Στο σενάριο εργασίας πάνω στο οποίο υλοποιήθηκε η εφαρμογή, δεν συμπεριλήφθηκε η πιθανότητα να χρειαστεί ο ασθενής να παραμείνει για θεραπεία εντός του νοσοκομείου. Με άλλα λόγια, δεν διαχειρίζεται θαλάμους νοσηλείας.

Η δυνατότητα για ηλεκτρονική παρακολούθηση την θαλάμων νοσηλείας είναι ενσωματωμένη σε όλες τις σύγχρονες νοσοκομειακές εφαρμογές και είναι απαιτούμενη από κάθε σύγχρονη μονάδα υγείας. Μία τέτοια επέκταση:

- Θα πρέπει να μπορεί να ελέγχει τα διαθέσιμα κρεβάτια (σε αριθμό και που βρίσκεται το καθένα),
- Να ελέγχει ποιός ασθενής βρίσκεται που και να διασυνδέεται με ένα σύστημα αναμονής ώστε να καθορίζεται προτεραιότητα σε περίπτωση μεγάλου αριθμού ασθενών.
- Θα πρέπει ακόμα να διασυνδέεται με κάποια εφαρμογή παρακολούθησης του ασθενή (πχ με την οθόνη Care Plan της εφαρμογής), ώστε να δίνεται η δυνατότητα στους επαγγελματίες υγείας να παρακολουθούν την κατάσταση του νοσηλευόμενου καθώς

και το αν τηρεί τις υποδείξεις των θεραπόντων ιατρών (πχ να παίρνει τα φάρμακά του στην σωστή ώρα).

- Θα μπορούσε να διασυνδεθεί τέλος με ένα σύστημα κοστολόγησης, ώστε να ελέγχεται αν το τελικό κόστος νοσηλείας ανταποκρίνεται στις υπηρεσίες που προσφέρθηκαν κατά την παραμονή εντός του ιδρύματος.

### **Αποθήκη**

Μία πιθανή μελλοντική προσθήκη είναι ένα σύστημα παρακολούθησης των αποθηκών και ελέγχου των ποσοτήτων αναλώσιμων. Ένα τέτοιο σύστημα

- θα πρέπει να μπορεί να παρακολουθεί κάθε ιατρική πράξη/εξέταση/ιατρική συνταγή που γίνεται/εκτελείται εντός της μονάδας υγείας καθώς και τα αναλώσιμα που απαιτούνται. Έτσι, στο τέλος της μέρας, θα μπορεί να υπάρχει ακριβής αριθμός από το κάθε είδος που χρησιμοποιήθηκε.
- Θα μπορούσε επίσης να διασυνδεθεί και με το λογιστήριο του νοσοκομείου, ώστε να υπάρχει ενημέρωση για το λειτουργικό κόστος των ιατρικών πράξεων καθώς και ενημέρωση για το σύνολο των εξόδων.
- Ένα τέτοιο σύστημα θα ήταν ιδιαίτερα χρήσιμο στην διοίκηση του νοσοκομείου, καθώς θα μπορεί να παρακολουθεί σε πραγματικό χρόνο την κατάσταση των αποθηκών και των αποθεμάτων,
- Θα ελέγχει το κόστος κάθε ιατρικής πράξης και να εξασφαλίζει ότι δεν θα γίνεται περιττή χρήση υλικών και πόρων.

Μία ενδιαφέρουσα μελλοντική επέκταση θα μπορούσε να αφορά ένα σύστημα διαχείρισης του προσωπικού της μονάδας υγείας (ωράρια/βάρδιες, μισθοδοσία κλπ) ή τη διαχείριση ειδικών υπηρεσιών εντός του ιδρύματος (πχ Μονάδες Εντατικής Θεραπείας).

Τέλος ειδική μνεία πρέπει να γίνει σε επεκτάσεις που να επιτρέπουν την συμβατότητα με συσκευές που ελέγχουν την κατάσταση της υγείας σε πραγματικό χρόνο (πχ Sensium) ή/και με διάφορα wearables, η χρήση των οποίων αρχίζει να αυξάνεται.



# 7

## Βιβλιογραφία

- [1] Karen A. Wager, Frances W. Lee, John P. Glaser, *Health Care Information Systems: A Practical Approach for Health Care Management*, 3rd Edition, Wiley 2013
- [2] Roy Thomas Fielding, *Architectural Styles and the Design of Network-based Software Architectures*, University of California, Irvine, Irvine, California, 2000
- [3] Bryan Basham, Kathy Sierra and Bert Bates, *Head First Servlets and JSP*, 2012
- [4] Kathy Sierra and Bert Bates, *Head First Java*, 2012
- [5] Google Android <http://www.android.com/>
- [6] Android Developer <http://developer.android.com/>
- [7] HL7 Fast Healthcare Interoperable Resources Standard  
<http://hl7.org/implement/standards/fhir/http.html>
- [8] Eclipse IDE, <http://www.eclipse.org>
- [9] Java, <http://www.oracle.com/technetwork/java>
- [10] Hibernate framework, <http://www.hibernate.org>