

**ΑΝΑΠΤΥΞΗ ΙΑΤΡΙΚΗΣ ΕΦΑΡΜΟΓΗΣ ΚΑΙ ΕΚΜΕΤΑΛΕΥΣΗ  
ΤΟΥ FHIR ΠΡΩΤΟΚΟΛΛΟΥ**

Πλήρες Νικόλας Κυριάκου

Η Διατριβή αυτή

Υποβλήθηκε προς Μερική Εκπλήρωση των

Απαιτήσεων για την Απόκτηση

Τίτλου Σπουδών Master

σε Προηγμένες Τεχνολογίες Πληροφορικής

στο

Πανεπιστήμιο Κύπρου

Συστήνεται προς Αποδοχή

από το Τμήμα Πληροφορικής

Δεκέμβριος, 2020

# ΣΕΛΙΔΑ ΕΓΚΡΙΣΗΣ

Διατριβή Master

## ΑΝΑΠΤΥΞΗ ΙΑΤΡΙΚΗΣ ΕΦΑΡΜΟΓΗΣ ΚΑΙ ΕΚΜΕΤΑΛΕΥΣΗ ΤΟΥ FHIR ΠΡΩΤΟΚΟΛΛΟΥ

Παρουσιάστηκε από

Νικόλας Κυριάκου

Ερευνητικός Σύμβουλος

---

Όνομα Ερευνητικού Συμβούλου

Μέλος Επιτροπής

---

Όνομα Μέλους Επιτροπής

Μέλος Επιτροπής

---

Όνομα Μέλους Επιτροπής

Πανεπιστήμιο Κύπρου

Δεκέμβριος , 2020

## **ΕΥΧΑΡΙΣΤΙΕΣ**

Αρχικά θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Κωνσταντίνο Παττίχη, ο οποίος με καθοδήγησε και μου έδωσε την ευκαιρία να ασχοληθώ με το συγκεκριμένο θέμα. Όπως, επίσης ευχαριστώ τον διδακτορικό Φοιτητή Παναγιώτη Σάββα για την συνεργασία.

Ιδιαίτερα, θα ήθελα να ευχαριστήσω την οικογένεια, τους φίλους αλλά και όλους τους συναδέλφους μου που με διάφορους και διαφορετικούς τρόπους ο κάθε ένας, ήταν δίπλα σε οποιαδήποτε δυσκολία ή πίεση αντιμετώπισα σε όλη τη διάρκεια της μελέτης και υλοποίησης της μεταπτυχιακής διατριβής μου.

## ΠΕΡΙΛΗΨΗ

Στόχος της παρούσας διατριβής ήταν η ανάπτυξη μιας ιατρικής εφαρμογής η οποία θα μπορεί να εκμεταλλευτεί τα πλεονεκτήματα που προσφέρονται από το ένα Fast Healthcare Interoperability Resources (FHIR) πρωτότυπο καθώς και των Application Programming Interface (APIs) τα οποία προσφέρονται από το πρωτότυπο αυτό. Επιπλέον, μας απασχόλησε η ασφάλεια των προσωπικών δεδομένων, τόσο των ασθενών όσο και των γιατρών, όπου μπορεί κάποιος να τα ανακτήσει μέσω του πρωτοκόλλου αυτού.

Για τους σκοπούς αυτή της διατριβής, επιλέξαμε σε συνεργασία με τον κ. Κ. Παττίχη να χρησιμοποιήσουμε τον FHIR Server που αναπτύχθηκε από το Πανεπιστήμιο Κύπρου και πιο συγκεκριμένα από τον διδακτορικό φοιτητή κ. Παναγιώτη Σάββα. Ο εν λόγω FHIR Server, παρέχει ένα επιπλέον Layer ανάμεσα στα 3<sup>rd</sup> party applications και τον FHIR Server. Το Layer αυτό και η ασφάλεια στην αλληλεπίδραση επιτυγχάνεται μέσω του Keycloak και του OAuth2 πρωτοκόλλου που θα αναλύσουμε στη συνέχεια.

Τέλος, το 3<sup>rd</sup> party application που αναπτύχθηκε με σκοπό να αναδείξει τις πιο πάνω δυνατότητες, είναι ένα Web Application, το οποίο έχει προορισμό να χρησιμοποιείται από γιατρούς με στόχο να διαχειρίζονται τους ασθενείς τους. Υπάρχει δηλαδή η δυνατότητα για δημιουργία καινούργιου ασθενή, διαχείριση κάποιου υφιστάμενου ασθενή, προβολή ιστορικού των ραντεβού, δημιουργία και διαχείριση των ραντεβού του γιατρού, καθώς και η επεξεργασία και αλλαγή των προσωπικών δεδομένων του ασθενή.

# ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

<b>Κεφάλαιο 1</b> .....	<b>3</b>
1.1 Κίνητρο .....	3
1.2 Στόχος .....	4
1.3 Δομή Εργασίας .....	5
<b>Κεφάλαιο 2</b> .....	<b>6</b>
2.1 Σχετικές Εργασίες .....	6
2.2 Ηλεκτρονική Υγεία .....	7
2.3 FHIR Πρότυπο .....	8
2.4 ehealth4u Project .....	11
2.5 Πλεονεκτήματα και δυνατότητες FHIR .....	12
<b>Κεφάλαιο 3</b> .....	<b>14</b>
3.1 Ανάλυση Απαιτήσεων .....	14
3.2 Καταγραφή και Ανάλυση Απαιτήσεων .....	15
3.2.1 Περιορισμοί Εφαρμογής .....	15
3.3 Χαρακτηριστικά Συστήματος Λογισμικού .....	16
3.3.1 Αξιοπιστία .....	16
3.3.2 Διαθεσιμότητα .....	16
3.3.3 Συντήρηση .....	16
3.4 Κανόνες Ευχρηστίας Συστήματος .....	17
3.5 Βασικές Λειτουργίες Χρήστη .....	18
3.6 Βασικές Λειτουργίες του Συστήματος .....	19
<b>Κεφάλαιο 4</b> .....	<b>20</b>
4.1 Λογισμικό Ανάπτυξης .....	20
4.1.1 XAMPP .....	20
4.1.2 Visual Studio Code .....	20
4.1.3 SOAP UI .....	20

4.2 Γλώσσες Προγραμματισμού .....	21
4.2.1 PHP .....	21
4.2.2 JavaScript .....	21
4.3 Μοντελοποίηση δεδομένων .....	21
4.3.1 XML .....	21
4.3.2 JSON(JavaScript Object Notation) .....	22
4.3.3 YAML.....	23
4.4 Τεχνολογίες.....	23
4.4.1 Docker .....	23
4.4.2 Docker Images .....	24
4.4.3 Docker Container .....	24
4.4.3 Docker-compose file .....	24
4.4.4 KeyCloak.....	25
4.4.4.1 Αλληλεπίδραση με Υφιστάμενα Applications.....	26
4.4.4.2 Client Adapters.....	27
4.4.4.3 Admin Console .....	27
4.4.4.4 Account Management Console.....	28
<b>Κεφάλαιο 5.....</b>	<b>29</b>
5.1 3 <sup>rd</sup> Party Service Provider .....	29
5.2 Ehealth4u FHIR Server.....	29
5.3 Αρχιτεκτονική-Ασφάλεια Συστήματος .....	33
<b>Κεφάλαιο 6.....</b>	<b>37</b>
6.1 Κωδικοποίηση Ανάπτυξη 3 <sup>rd</sup> Party Service Provider .....	37
6.1.1 Front-End .....	37
6.1.2 Calendar.....	37
6.1.3 Back-End.....	38
6.2 Εγκατάσταση Εφαρμογής.....	39
6.3 Περιγραφή Συστήματος.....	41

6.3.1 Log in:.....	41
6.3.2 Υπόλοιπες Λειτουργίες Εφαρμογής.....	42
6.3.2.1 Στοιχεία γιατρού .....	43
6.3.2.2 Ημερολόγιο Γιατρού.....	43
6.3.2.3 Προσωπικά στοιχεία Ασθενή .....	44
6.3.2.4 Δημιουργία Νέου Ραντεβού.....	44
6.3.2.5 Ιστορικό Ραντεβού .....	45
6.3.2.6 Δημιουργία νέου ασθενή .....	45
6.3.2.7 Λίστα Ασθενών.....	46
6.4 Δομή Κώδικά.....	48
<b>Κεφάλαιο 7.....</b>	<b>53</b>
7.1 Γενικά Συμπεράσματα.....	53
7.2 Αποδοχή από τον κόσμο .....	53
7.3 Μελλοντικές Προεκτάσεις.....	54
7.3.1 Μεγαλύτερη αξιοποίηση των δυνατοτήτων που προσφέρει το FHIR πρωτόκολλο. ....	54
7.3.2 Επέκταση του συστήματος.....	55
7.3.3 Δυναμική χρήση του FHIR πρωτόκολλου και των περιορισμών του.....	55
7.3.4 Εφαρμογή του FHIR πρωτοκόλλου από τους παρόχους Υγείας.....	55
<b>Βιβλιογραφία.....</b>	<b>56</b>

# ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

<i>Εικόνα</i>	<i>Σελίδα</i>
Εικ. 2-1 FHIR-Interactions	9
Εικ. 2-2 FHIR-Options	10
Εικ. 2-3 FHIR- _summary responses	10
Εικ. 3-1 Use cases	14
Εικ. 4-1 Docker Architecture	23
Εικ. 4-2 OAUTH2 Podocol	25
Εικ. 4-3 Keycloak	26
Εικ. 4-4 Keycloak-Admin Console	27
Εικ. 4-5 Keycloak-Account Management	28
Εικ. 5-1 FHIR Architecture	33
Εικ. 5-2 User cases	35
Εικ. 6-1 Goofle – Cloud Platform	40
Εικ. 6-2 Login Page	41
Εικ. 6-3 Main Doctor Page	42
Εικ. 6-4 Doctor Details	43
Εικ. 6-5 Doctor Calendar	44
Εικ. 6-6 Patient Details	44
Εικ. 6-7 Patient Calendar	45
Εικ. 6-8 Appointment History	45
Εικ. 6-9 Patient Creation	46
Εικ. 6-10 List of Patients	46
Εικ. 6-11 FHIR Interface	47



Εικ. 6-12 FHIR Search Interface	47
Εικ. 6-13 FHIR XML Update	48
Εικ. 6-14 FHIR Json Update	48
Εικ. 6-15 FHIR Code files	48
Εικ. 6-16 packages και plugins folders	49
Εικ. 6-17 Js folder	50
Εικ. 6-18 API folder	50
Εικ. 6-19 Backend folder	50
Εικ. 6-20 Search practitioner by id api	51
Εικ. 6-21 Keycloak configuration	51
Εικ. 6-22 docker-compose file	52
Εικ. 7-1 Google Trends on the “FHIR” key word	54

# Κεφάλαιο 1

## Εισαγωγή

### 1.1 Κίνητρο

Η Ηλεκτρονική Υγεία αποτέλεσε ένας από τους βασικούς στόχους της Ευρωπαϊκής Ένωσης και για αυτό άλλωστε τον λόγο από το 2004 και έπειτα έχει δαπανήσει πολύ μεγάλα ποσά προς επίτευξη του στόχου αυτού. Η ηλεκτρονική υγεία μπορεί να βελτιώσει σε μεγάλο βαθμό την ποιότητα ζωής του κάθε ασθενή ξεχωριστά, μπορεί να μετατρέψει το σύστημα υγείας σε ασθενοκεντικό και έτσι θα έχουμε ραγδαίες βελτιώσεις τόσο στην ποιότητα όσο και στην αποτελεσματικότητα της κάθε ιατροφαρμακευτικής περίθαλψης.

Οι τρεις βασικοί στόχοι που οριοθέτησε η Ε.Ε για την ανάπτυξη της Ηλεκτρονικής Υγείας ήταν:

1. Να δομηθεί ένα ορθό πλαίσιο υποστήριξης της, το οποίο θα μπορούσε να υποστηρίξει τις κοινές προκλήσεις όλων. Ένα τέτοιο μέτρο θα ήταν η δημιουργία του ηλεκτρονικού φακέλου ο οποίος θα περιείχε πληροφορίες για τον κάθε ασθενή ξεχωριστά και έτσι οι γιατροί θα μπορούσαν ανά πάσα χρονική στιγμή να ταυτοποιήσουν τον ασθενή που έχουν μπροστά τους, και έτσι ο ηλεκτρονικός φάκελος να τους οδηγήσει στην σωστή περίθαλψη. Όπως επίσης ακόμα ένα βήμα προς την υλοποίηση του πιο πάνω στόχου θα ήταν η ανάπτυξη ενός εθνικού χρονοδιαγράμματος για εφαρμογή του σχεδίου υγείας. Επιπλέον, η εξεύρεση από τα κράτη μέλη επενδύσεων προς προώθηση του σχεδίου αυτού είναι απαραίτητη. Τέλος, η ανάπτυξη νομοθεσιών που να προωθούν και να οριοθετούν τα πλαίσια του συστήματος αυτού θα πρέπει να είναι απαραίτητη.

2. Να αρχίσουν σε πιλοτικό πλαίσιο δράσεις διανομής υπηρεσιών για ηλεκτρονική υγεία. Αρχικώς στόχος ήταν η δημιουργία μιας διαδικτυακής πύλης της Ε.Ε η οποία θα παρείχε πληροφορίες σχετικά με τη δημόσια υγεία, όπως για παράδειγμα την ασφάλεια στην εργασία και σχετικά με τους κινδύνους που υπάρχουν για την υγεία μας. Επίσης, η ανάπτυξη τεχνολογιών που θα μπορούσαν να βοηθήσουν την υγεία. Ένα ακόμη μέτρο που θα βοηθούσε τον πιο πάνω στόχο ήταν η δημιουργία ηλεκτρονικής συντακογράφησης, ηλεκτρονικά παραπεμπτικά, τηλεπαρακολούθηση, τηλεπερίθαλψη και προώθηση της χρήση ηλεκτρονικής κάρτας ασφάλισης υγείας.

3. Ανταλλαγή πληροφοριών ανάμεσα στους ενδιαφερόμενους για βελτιστοποίηση του συστήματος και ταυτόχρονα να υπάρχουν μετρικές προόδου. Τέλος για την παρακολούθηση της βελτίωσης του συστήματος αυτού θα πρέπει σε τακτικά χρονικά διαστήματα να εκδίδεται μία μελέτη για την κατάσταση του και ως προς το βαθμό εξάπλωσης του.<sup>1</sup>[1]

Στην Κύπρο η Ηλεκτρονική Υγεία βρίσκεται ακόμα σε αρχικά στάδια, υπήρξαν σημαντικά σημάδια προόδου ειδικά με την εφαρμογή του ΓΕΣΥ, παρόλα αυτά μπορούμε να βελτιωθούμε ακόμη περισσότερο. Ένας από τους σκοπούς της μεταπτυχιακής αυτής διατριβής είναι να δείξουμε ότι μπορούμε να μετατρέψουμε τα υφιστάμενα εργαλεία, με απλούς τρόπους ούτως ώστε να μπορούν να εφαρμόσουν το πρωτόκολλο αυτό για να υπάρχει η δυνατότητα για άμεση, εύκολη και ελεγχόμενη πρόσβαση στα στοιχεία των ασθενών.

## 1.2 Στόχος

Η εργασία αυτή έχει σκοπό να εφαρμόσει και να λύσει μερικά από τα πιο πάνω θέματα και να δείξει σε πρακτικό επίπεδο πως μπορεί να υλοποιηθεί ένας κεντρικός και τυποποιημένος τρόπος για άμεση ανταλλαγή δεδομένων μεταξύ των ενδιαφερόμενων. Τα δεδομένα αυτά μπορούν να χρησιμοποιηθούν για τη δημιουργία του Ηλεκτρονικού Φάκελου Υγείας και αυτός να μπορεί να χρησιμοποιηθεί από τους

ειδικούς για να υπάρξει καλύτερη και πιο άμεση εξυπηρέτηση του κάθε ασθενή ξεχωριστά.

### 1.3 Δομή Εργασίας

Αυτή η διατριβή αποτελείται από επτά κεφάλαια.

- Στο 2<sup>ο</sup> Κεφάλαιο, θα μελετήσουμε άλλες σχετικές εργασίες για το θέμα αυτό και θα αναλύσουμε τις δυνατότητες που προσφέρει ο FHIR.
- Στο Κεφάλαιο 3 θα αναλύσουμε τις απαιτήσεις που προκύπτουν για την επίλυση του θέματος.
- Στο Κεφάλαιο 4 της διατριβής αυτής περιγράφονται τα εργαλεία που χρειάστηκα για να υλοποιήσω το λογισμικό αυτό. Πιο αναλυτικά περιγράφονται τα λογισμικά για την ανάπτυξη, οι γλώσσες προγραμματισμού, οι τρόποι μοντελοποιήσεις των δεδομένων και οι τεχνολογίες που χρησιμοποιήθηκαν για να μπορεί το λογισμικό αυτό να είναι σε λειτουργία.
- Στο Κεφάλαιο 5 θα αναλυθεί και θα παρουσιαστεί η Σχεδίαση του Συστήματος και η τελική του αρχιτεκτονική.
- Στο Κεφάλαιο 6 θα περιγράψουμε την μεθοδολογία που ακολουθήθηκε κατά την υλοποίηση του συστήματος. Θα παρουσιάσουμε το εγχειρίδιο χρήσης του Keyckoak, του FHIR Server και το interface του 3<sup>rd</sup> party Service Provider. Επίσης, θα αναλυθεί και η δομή του κώδικα και πως αυτός λειτουργεί.
- Στο τελευταίο, Κεφάλαιο 6, αναλύονται τα τελικά συμπεράσματα της διατριβής και μελλοντικές εργασίες που μπορούν να αναπτυχθούν για καλύτερη λειτουργία του FHIR πρωτοκόλλου.

## Κεφάλαιο 2

### Σχετικές Εργασίες

#### 2.1 Σχετικές Εργασίες

Το FHIR (Fast Healthcare Interoperability Resources) είναι ένα ανοιχτό, κοινοτικό πρότυπο που χρησιμοποιεί σύγχρονες τεχνολογίες ιστού όπως JSON (JavaScript Object Notation) και δι' επαφές προγραμματισμού εφαρμογών (API) για την αποθήκευση και ανταλλαγή διοικητικών, κλινικών και οικονομικών δεδομένων. Τα δεδομένα της υγειονομικής περίθαλψης σήμερα αποθηκεύονται με μη τυποποιημένους τρόπους, γεγονός που καθιστά δύσκολη την επικοινωνία μεταξύ διαφορετικών συστημάτων πληροφοριών. Τα περισσότερα συστήματα υγείας χρησιμοποιούν το HL7 v.2 για τα δεδομένα τους, ένα πρότυπο που αναπτύχθηκε τη δεκαετία του 1980 πριν το διαδίκτυο γίνει ευρέως διαδεδομένο στο κοινό. Η μορφή HL7 v.2 δεν είναι αναγνώσιμη από τον άνθρωπο, απαιτεί την κατασκευή προσαρμοσμένων δι' επαφών για την επικοινωνία κάθε συστήματος και δεν επιτρέπει την ανταλλαγή λεπτομερών δεδομένων. Το FHIR επιλύει αυτά τα προβλήματα παρέχοντας έναν απλό, τυποποιημένο τρόπο αποθήκευσης δεδομένων υγειονομικής περίθαλψης σε μορφή αναγνώσιμη από τον άνθρωπο και προσφέροντας APIs για την απρόσκοπτη ανταλλαγή δεδομένων σε πραγματικό χρόνο. Το FHIR διευκολύνει τους προγραμματιστές να έχουν πρόσβαση στα δεδομένα υγειονομικής περίθαλψης που χρειάζονται για τη γρήγορη δημιουργία εφαρμογών.

Η πολυπλοκότητα του τομέα της υγειονομικής περίθαλψης, η μοντελοποίηση δεδομένων, η αποθήκευση ιατρικών δεδομένων και οι προσαρμοσμένες ενσωματώσεις με παλαιά συστήματα είναι παράγοντες που οδηγούν σε μακροχρόνιους κύκλους ανάπτυξης και υψηλό κόστος έργου για τεχνολογίες υγείας. Το νέο πρότυπο HL7 FHIR δημιουργήθηκε για να ανοίξει τις πόρτες στην καινοτομία,

αντιμετωπίζοντας πολλά από τα προβλήματα που σχετίζονται με την παραδοσιακή διαδικασία ανάπτυξης πληροφορικής υγείας. Άμεσος, σκοπός του FHIR είναι να λύσει όλα τα πιο πάνω προβλήματα είτε αυτά αφορούν συστήματα που τρέχουν σε παλαιότερα λογισμικά, σε οποιαδήποτε συσκευή όπως υπολογιστές, tablet και κινητά τηλέφωνα.<sup>23</sup>[2][3]

Σκοπός της διατριβής αυτής ήταν η υλοποίηση ενός πλήρους μοντέλου, για ανταλλαγή δεδομένων μεταξύ 3<sup>rd</sup> party Service Providers και του FHIR Server με απόλυτη ασφάλεια για τα προσωπικά δεδομένα όλων των εμπλεκόμενων μελών.

## **2.2 Ηλεκτρονική Υγεία**

Η Ευρωπαϊκή Επιτροπή ορίζει την ηλεκτρονική υγεία ως:

"... η χρήση σύγχρονων τεχνολογιών πληροφορίας και επικοινωνιών για την κάλυψη των αναγκών των πολιτών, των ασθενών, των επαγγελματιών του τομέα της υγείας, των παροχών υγειονομικής περίθαλψης, καθώς και των υπευθύνων χάραξης πολιτικής".

Ενώ ο Παγκόσμιος Οργανισμός Υγείας (Π.Ο.Υ.) ορίζει την ηλεκτρονική υγεία ως:

"... η αποδοτική και ασφαλής χρήση των τεχνολογιών πληροφορίας και επικοινωνιών για την υποστήριξη της υγείας αλλά και πεδίων που σχετίζονται με την υγεία, συμπεριλαμβανομένης της υγειονομικής περίθαλψης, της παρακολούθησης και της αγωγής υγείας, της γνώσης και της έρευνας".<sup>4</sup>[4]

Και οι δύο πιο πάνω ορισμοί τονίζουν την ανάγκη για χρήση των σύγχρονων τεχνολογιών καθώς και την ανάγκη για διαλειτουργικότητα μεταξύ των εμπλεκόμενων μελών στον τομέα της Υγείας. Τα δύο αυτά στοιχεία είναι τα κύρια εμπόδια που αντιμετωπίζαμε τα τελευταία χρόνια και σταματούσαν σε σημαντικό βαθμό την εφαρμογή της Ηλεκτρονικής Υγείας στην ζωή μας.

Με τον όρο διαλειτουργικότητα εννοούμε την ανεμπόδιστη χρήση και ολοκλήρωση ετερογενών συστημάτων. Με αυτό τον τρόπο θα είναι δυνατή η ασφαλής και άμεση πρόσβαση στις υπηρεσίες και πληροφορίες υγείας από οπουδήποτε και αν προέρχονται αυτές. Η έλλειψη διαλειτουργικότητας καθιστά τις υπάρχουσες λύσεις περιορισμένης χρήσης και αποθαρρύνει με αυτό τον τρόπο την υιοθέτησή τους τόσο από τους φορείς όσο και από τους τελικούς χρήστες. Τα θέματα που άπτονται αυτού του προβλήματος είναι προτυποποίησης, αρχιτεκτονικής, ταξινόμησης, ανταλλαγής ιατρικών μηνυμάτων, κ.ά. Η προσπάθεια για την υπερπήδηση αυτών των προβλημάτων θα πρέπει και είναι συνολική (σε ευρωπαϊκό ή ακόμα και σε παγκόσμιο επίπεδο).[5]

Την άμεση και εύκολη λύση στα δύο αυτά προβλήματα που θα μας οδηγήσουν σίγουρα στην καλύτερη έκδοση μιας Ηλεκτρονικής Υγείας για όλους, έρχεται να δώσει το FHIR πρότυπο. Το οποίο προσφέρει πρόσβαση, με την βοήθεια σύγχρονων τεχνολογιών, στον οποιοδήποτε σε ιατρικά δεδομένα των ασθενών με τυποποιημένη μορφή για να μπορούν και αυτοί με τη σειρά τους να τα εντάξουν στο σύστημα υγείας που χρησιμοποιούν.

### **2.3 FHIR Πρότυπο**

Ας προχωρήσουμε όμως και στην πιο λεπτομερή ανάλυση στο τι μπορεί να προσφέρει το πρωτόκολλο του FHIR.<sup>6</sup>[6]

Στον FHIR Server υπάρχουν προκαθορισμένα τα resources όπου μπορεί ο κάθε χρήστης να αλληλεπιδράσει μέσω των APIs. Η αλληλεπίδραση είναι ποικίλη και μπορούμε να δούμε τις δυνατότητες που προσφέρει στον πιο κάτω πίνακα:

#### Instance Level Interactions

read	Read the current state of the resource
vread	Read the state of a specific version of the resource
update	Update an existing resource by its id (or create it if it is new)
patch	Update an existing resource by posting a set of changes to it
delete	Delete a resource
history	Retrieve the change history for a particular resource

#### Type Level Interactions

create	Create a new resource with a server assigned id
search	Search the resource type based on some filter criteria
history	Retrieve the change history for a particular resource type

#### Whole System Interactions

capabilities	Get a capability statement for the system
batch/transaction	Update, create or delete a set of resources in a single interaction
history	Retrieve the change history for all resources
search	Search across all resource types based on some filter criteria

Εικ. 2-1 FHIR-Interactions

#### Μορφή του τελικού URL:

Για να μπορέσουμε να στείλουμε οποιοδήποτε request προς τον Server μας, θα πρέπει να ακολουθήσουμε μια δομή ως προς την σύνταξη του τελικού URL, δηλαδή, το request θα στέλετε στην πιο κάτω μορφή:

```
http{s}://server{/path}
https://server/path/Patient
https://fhri.ehealth4u.eu/fhir/Patient?identifier=151515
```

#### Ασφάλεια:

Για το θέμα της ασφάλεια των δεδομένων, η χρήση HTTPS είναι προαιρετική, αλλά όλες οι ανταλλαγές παραγωγής δεδομένων υγειονομικής περίθαλψης πρέπει να χρησιμοποιούν SSL. Επιπλέον, για επιπρόσθετη ασφάλεια εμείς χρησιμοποιήσαμε και το OAuth2 πρωτόκολλο το οποίο θα αναλυθεί στη συνέχεια.

#### Κωδικοποίηση:

Ο επίσημος τρόπος κωδικοποίησης του FHIR είναι το application/fhir+xml ή application/fhir+json. Η εφαρμογή τύπου περιεχομένου / x-www-form-urlencoded είναι επίσης αποδεκτή για την ανάρτηση αιτημάτων αναζήτησης.



Εάν ένας πελάτης παρέχει έναν γενικό τύπο στο Accept Header (application/xml, text/json ή application/json), ο Server θα ανταποκριθεί με τον αντίστοιχο απαιτούμενο τύπο κωδικοποίησης, χρησιμοποιώντας τις μορφές XML ή JSON.

### Γενικές Παράμετροι:

Εικ. 2-2 FHIR-Options

**\_format:** Αυτή η παράμετρος επιτρέπει σε έναν client να παρακάμψει την τιμή όταν δεν είναι σε θέση να την ορίσει σωστά λόγω εσωτερικών περιορισμών(για παράδειγμα το gender πρέπει να είναι female ή male)

**\_pretty:** Επιστρέφει μια αρκετά τυπωμένη απόκριση για response.

**\_summary:** Επιστρέφει μια προκαθορισμένη σύντομη μορφή του resource σε response

<b>true</b>	Return a limited subset of elements from the resource. This subset SHOULD consist solely of all supported elements that are marked as "summary" in the base definition of the resource(s) (see <code>ElementDefinition.isSummary</code> )
<b>text</b>	Return only the "text" element, the "id" element, the "meta" element, and only top-level mandatory elements
<b>data</b>	Remove the text element
<b>count</b>	Search only: just return a count of the matching resources, without returning the actual matches
<b>false</b>	Return all parts of the resource(s)

Εικ. 2-3 FHIR- \_summary responses

**\_elements:** Επιστρέφει ένα συγκεκριμένο σύνολο στοιχείων

```
GET [base]/Patient?_elements=identifier,active,link
```

### **Αλληλεπίδραση:**

Όλες οι αλληλεπιδράσεις από ένα 3<sup>rd</sup> party application προς τον FHIR Server γίνονται με APIs και τις γνωστές μεθόδους:

- GET για διάβασμα δεδομένων
- POST για εγγραφή
- PUT για ανανέωση
- PATCH για ανανέωση των ιδιοτήτων του resource
- DELETE για διαγραφή δεδομένων

### **2.4 ehealth4u Project**

Το συγκεκριμένο Project αναπτύσσεται και βασίζεται στην φιλοσοφία και τις οδηγίες της Ε.Ε για την υγειονομική περίθαλψη. Έχει ως επίκεντρο τον πολίτη, πλήρη διαλειτουργικότητα και ελεγχόμενη ποιότητα με απώτερο σκοπό την ανάπτυξη ενός οικοσυστήματος υγείας.

Το eHealth4U φιλοδοξεί να αξιοποιήσει τα ευρήματα από ερευνητικές πρωτοβουλίες της ΕΕ, όπως CEFCY, FiSTAR, Linked2Safety, eENERCA, ANTILOPE, CALLIOPE και να προωθήσει περαιτέρω την ανάπτυξη και εφαρμογή έρευνας προς συγκεκριμένες λύσεις για ένα πλήρες ηλεκτρονικό αρχείο υγείας (EHR), διαλειτουργικό, ασφαλές, ασθενοκεντικό και νομικά συμβατό.

Τα μέλη που απαρτίζουν το project αυτό είναι:

- University of Cyprus
- State Health Services Organisation (SHSO)
- Cyprus Telecommunications Authority CYTA
- 3AeHealth LTD
- UBITECH LTD
- International Institute for Compassionate Care
- HIPPOCRATEON Private Hospital

- Iron Mountain Cyprus LTD

Στόχος του project είναι:

- Η ανάπτυξη ενός πλαισίου τόσο σε νομικό, ηθικό επίπεδο όσο και στους περιορισμούς που υπάρχουν στο απόρρητο των δεδομένων και στις δυνατότητες που παρέχει η τεχνολογία.
- Σχεδιασμός ενός ασθενοκεντρικού Ηλεκτρονικού Συστήματος Υγείας.
- Ανάπτυξη και υλοποίηση της πλατφόρμας eHealth4U για την υποστήριξη ενός ολοκληρωμένου, διαλειτουργικού, επεκτάσιμου και έξυπνου οικοσυστήματος eHealth.
- Να σχεδιάσει και να εφαρμόσει ένα πρόγραμμα υιοθεσίας πρώιμου eHealth στην Υγειονομική περίθαλψη της Κύπρου και να υιοθετήσει μια αποτελεσματική στρατηγική διαχείρισης αλλαγών για την επιτυχή υιοθέτηση ενός καθολικού EHR.
- Αξιοποίηση της βιωσιμότητας της ολοκληρωμένης πλατφόρμας και υπηρεσιών eHealth4U.<sup>7</sup>[7]

## **2.5 Πλεονεκτήματα και δυνατότητες FHIR**

Με την επιλογή και την χρήση του FHIR πρωτοκόλλου ουσιαστικά επιλέγουμε να χρησιμοποιήσουμε την επόμενη γενιά των HL7 προτύπων και την εκμετάλλευση της πλειάδας των δυνατοτήτων που προσφέρει και την επίλυση των συγχρόνων απαιτήσεων της εποχής.

1. Το πρότυπο αυτό δίνει λύση στην ανάγκη για ανάπτυξη γρήγορα και εύκολα ιατρικών εφαρμογών, αφού το FHIR δίνει έμφαση στις ανάγκες των προγραμματιστών και βασίζεται στις σύγχρονες τεχνολογίες (REST APIs).
2. Το πιο πάνω σημείο εκτός από τους προγραμματιστές βοηθά σημαντικά και τις εταιρείες που αναπτύσσουν ιατρικές εφαρμογές, αφού μειώνουν σημαντικά το κόστος ανάπτυξης αλλά και συντήρησης των συστημάτων Υγείας.

3. Παρέχεται η δυνατότητα για αλληλεπίδραση σε πραγματικό χρόνο. Με αυτό τον τρόπο το πρότυπο αυτό δίνει λύσεις στις ανάγκες της σημερινής εποχής όπου ο κάθε ασθενής ή ακόμα και γιατρός μπορεί να έχει πρόσβαση στα στοιχεία του FHIR προτύπου (Cloud Computing) κάθε στιγμή από οποιαδήποτε συσκευή και αν χρησιμοποιεί (διαλειτουργικότητα).
4. Ένα ακόμα σημαντικό στοιχείο που μας οδηγεί στη χρήση του πρότυπου αυτού είναι τα άλματα που μπορεί να κάνει ο ιατρικός κλάδος. Αυτό συμβαίνει αφού τα δεδομένα αυτά πλέον θα είναι συγκεντρωμένα σε ένα κεντρικό σημείο, και έτσι οι ερευνητές μπορούν εύκολα να έχουν πρόσβαση για την μαζική ανάλυση ιατρικών δεδομένων.
5. Το πρότυπο αυτό είναι δωρεάν και επεκτάσιμο οπότε αν μπορεί να εφαρμοστεί σε οποιαδήποτε εφαρμογή θελήσουμε.

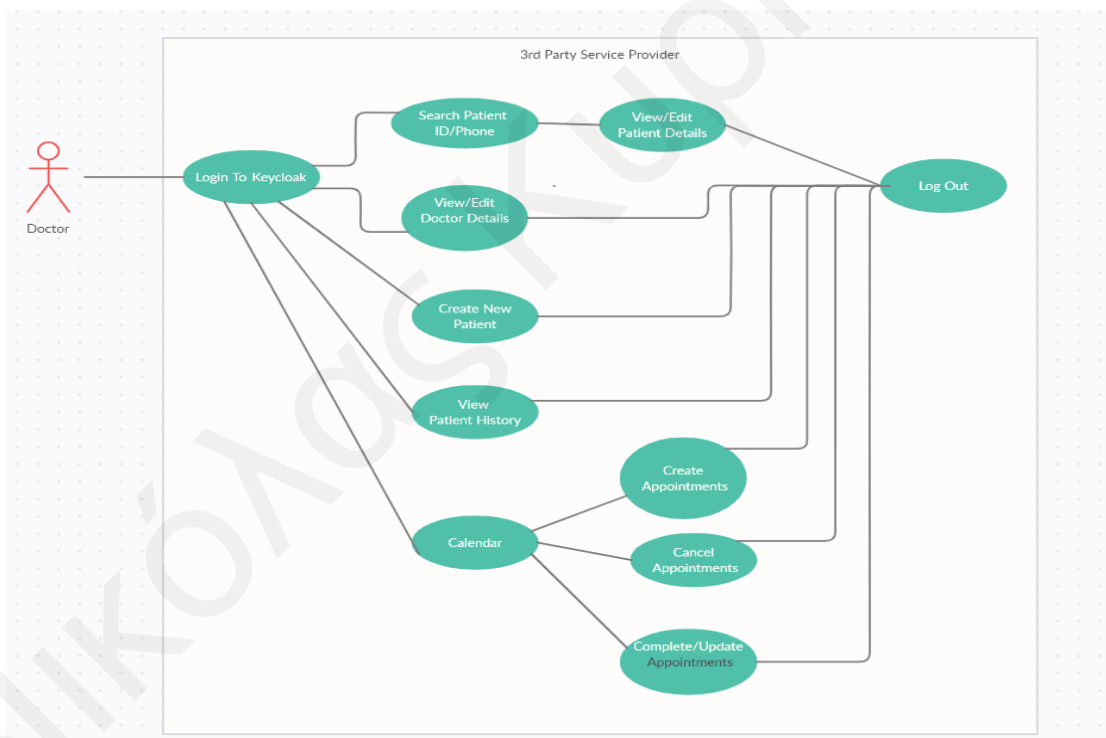
# Κεφάλαιο 3

## Ανάλυση Απαιτήσεων

### 3.1 Ανάλυση Απαιτήσεων

Το 3rd Party Application που αποφασίσαμε να προχωρήσουμε στην υλοποίηση του, αφορούσε γιατρούς και πως αυτοί θα μπορούσαν να διαχειριστούν τους ασθενείς τους όσο και τα ραντεβού για τις επισκέψεις αυτών.

Στο πιο κάτω Use Case διάγραμμα φαίνονται οι δυνατότητες που μπορεί να εκμεταλλευτεί κάποιος από το σύστημα αυτό.



Εικ. 3-1 Use cases

### **3.2 Καταγραφή και Ανάλυση Απαιτήσεων**

Για τον καθορισμό των απαιτήσεων, που θα έπρεπε να ικανοποιεί η εφαρμογή αυτή, εκτός από τις συναντήσεις που είχα με τον επιβλέποντα καθηγητή μου κ. Κωνσταντίνο Παττίχη, πραγματοποίησα και μια συνάντηση με μια γραμματέα σε Κλινική Υγείας ούτως ώστε να καταγράψω τις πραγματικές ανάγκες που θα έπρεπε να επιλύει ένα τέτοιο σύστημα.

Η 1η λειτουργία για ένα τέτοιο σύστημα θα έπρεπε να ήταν η δημιουργία και η ηλεκτρονική καταχώρηση κάποιων προσωπικών δεδομένων του ασθενή.

Σύμφωνα με τα συμπεράσματα των συναντήσεων μου με το προσωπικό του κέντρου υγείας που επισκέφτηκα ήταν το Ονοματεπώνυμο, Αρ. Ταυτότητας, Τηλέφωνο Επικοινωνίας, Φύλο, Email και Διεύθυνση του ασθενή.

Ακολούθως, η 2<sup>η</sup> βασική λειτουργία θα έπρεπε να ήταν η αναζήτηση του κάθε ασθενή ξεχωριστά με βασικά στοιχεία αναζήτησής την ταυτότητα ή το τηλέφωνο του πελάτη.

Η λειτουργία αυτή είναι η πιο συνηθής για τους γιατρούς και έτσι θα έπρεπε να ήταν διαθέσιμη σε οποιαδήποτε οθόνη βρισκόταν ο χρήστης.

Η 3<sup>η</sup> λειτουργία του συστήματος θα ήταν η προβολή σε μια λίστα όλων των ασθενών που υπάρχουν στο σύστημα.

Η 4<sup>η</sup> λειτουργία θα έπρεπε να μπορεί η γραμματέας ή ο γιατρός εύκολα και γρήγορα να μπορεί να ορίζει αλλάζει ή ακυρώνει κάποιο ραντεβού, όπως επίσης και η συνολική προβολή του προγράμματος του γιατρού με τα στοιχεία κάθε φορά του ασθενή να είναι διαθέσιμα.

Τέλος, θα ήταν αρκετά χρήσιμο να μπορεί ο χρήστης του συστήματος να βλέπει το ιστορικό των ραντεβού του κάθε ασθενή.

#### **3.2.1 Περιορισμοί Εφαρμογής**

Η εφαρμογή αυτή αναπτύχθηκε, όμως για να χρησιμοποιηθεί απαιτείται η τήρηση κάποιων βασικών προϋποθέσεων. Πρώτα από όλα, ο χρήστης δεν θα μπορεί να έχει πρόσβαση στην συγκεκριμένη εφαρμογή εάν δεν έχει πρόσβαση προς το διαδίκτυο.

Αυτό γιατί τα δεδομένα της λαμβάνονται σε πραγματικό χρόνο από το FHIR Base. Τέλος, ο κάθε χρήστης θα πρέπει πριν την εκμετάλλευση της εφαρμογής αυτής να λάβει τα credentials του από τον διαχειριστή του Keycloak server, όπου θα αναλύσουμε στην συνέχεια.

### **3.3 Χαρακτηριστικά Συστήματος Λογισμικού**

#### **3.3.1 Αξιοπιστία**

Ένα από τα σημαντικότερα χαρακτηριστικά που θα πρέπει να διέπει οποιαδήποτε εφαρμογή είναι η αξιοπιστία της, ιδιαίτερα μια ιατρική εφαρμογή όπως τη δική μας. Για παράδειγμα στην παρούσα εφαρμογή ο χρήστης δεν θα μπορούσε να έχει πρόσβαση στο ιστορικό κάποιου ασθενή ενώ στην πραγματικότητα να ήθελε να ψάξει το ιστορικό κάποιου αλλού.

Από την αντίθετη πλευρά, στην εφαρμογή αυτή απαιτούνται κάποια από τα στοιχεία του ασθενή τα οποία δεν μπορεί να είναι εντελώς αυθαίρετα. Έτσι, η εφαρμογή περιορίζει και καθοδηγεί τον χρήστη ως προς την μορφή των δεδομένων που στέλνονται προς το FHIR Base. Με τους περιορισμούς αυτούς η εφαρμογή γίνεται πιο αξιόπιστη.

#### **3.3.2 Διαθεσιμότητα**

Το σύστημα αυτό θα είναι οποιαδήποτε στιγμή διαθέσιμο προς τους χρήστες αφού άλλωστε αυτός είναι και ο σκοπός του, η άμεση δηλαδή ανταλλαγή δεδομένων από και προς τον FHIR Server.

#### **3.3.3 Συντήρηση**

Η συντήρηση ενός συστήματος λογισμικού είναι ένα από τα βασικά στάδια κατά την διαδικασία της ανάπτυξης του. Για να μπορεί να γίνει κατορθωτό όμως θα πρέπει να διατυπώνονται και να αναλύονται τα λάθη που τυχόν να προκύψουν κατά τη διάρκεια της λειτουργίας της εφαρμογής αυτής. Για αυτό ακριβώς το λόγο ο κώδικας έχει

αρκετά σχόλια ούτως ώστε να μπορεί στη συνέχεια να εντοπιστεί το σημείο που πρέπει να συντηρηθεί και να διορθωθεί. Για το ίδιο ακριβώς λόγο, οι μεταβλητές και τα ονόματα των Web Services που δημιουργήθηκαν είναι άκρως κατατοπιστικά για την λειτουργία τους.

### 3.4 Κανόνες Ευχρηστίας Συστήματος

Κανόνες του Nielsen: <sup>8</sup>[8]

1. Ορατότητα της κατάστασης του συστήματος: Ανά πάσα χρονική στιγμή σε όποιο σημείο της εφαρμογής και εάν βρίσκεται ο χρήστης έχει τη δυνατότητα να αναγνωρίζει σε ποιο σημείο της εφαρμογής είναι και ποιες οι πιθανές του επιλογές.
2. Συσχέτιση συστήματος και πραγματικού κόσμου: Οι λειτουργίες του συστήματος είναι αρκετά επεξηγηματικές για να μπορεί ο χρήστης να καταλαβαίνει εύκολα τη χρήση τους. Επιπλέον, χρησιμοποιούνται και μερικές μεταφορές από την καθημερινότητα μας όπως η χρήση της παραδοσιακή πόρτα εξόδου log out. Μέσω αυτών των μεταφορών πετυχαίνουμε την κατανόηση της λειτουργίας της εφαρμογής προς το χρήστη.
3. Έλεγχος του συστήματος από το χρήστη: Ο χρήστης από όποια οθόνη και να βρίσκεται μπορεί να ελέγχει το σύστημα αφού του παρέχεται η λειτουργία του menu στο αριστερό μέρος της οθόνης του όπου μπορεί να μεταβεί ανά πάσα χρονική στιγμή στη λειτουργία που θέλει.
4. Συνέπεια και τήρηση προτύπων: Όλες οι οθόνες του συστήματος είναι δομημένες με τον ίδιο ακριβώς τρόπο. Στο κέντρο τους βρίσκεται η κύρια λειτουργία της οθόνης ενώ στα αριστερά του χρήστη βρίσκεται το menu πλοήγησης.
5. Υποβοήθηση χρηστών στην αναγνώριση, διάγνωση και ανάνηψη σφάλματων: Σε περίπτωση που υπάρξουν σφάλματα από το χρήστη σε κάθε μία από αυτές εμφανίζεται ένα Notification το οποίο εμφανίζει το κατάλληλο μήνυμα



για να βοηθήσει το χρήστη να επαναφέρει το σύστημα στην επιθυμητή κατάσταση.

6. Σχεδιασμός για αποτροπή σφαλμάτων χρήστη: Σε συνέχεια του πιο πάνω, το σύστημα παρέχει διάφορα hints προς το χρήστη για να αποτρέπονται τα λάθη.
7. Ελαχιστοποίηση του μνημονικού φορτίου: Για να το πετύχω αυτό, σχεδίασα της οθόνες με τέτοιο τρόπο ούτως ώστε το σύστημα να είναι πολύ απλό στην πλοήγηση και έτσι ο χρήστης να μπορεί να το χειριστεί χωρίς να επιβάλλεται να θυμάται οτιδήποτε.
8. Ευελιξία και αποδοτικότητα χρήσης: Ο κάθε χρήστης μπορεί εύκολα να κάνει πλοήγηση στο σύστημα αφού αυτός είναι και ο κύριος του σκοπός.
9. Αποφυγή περιττών στοιχείων: Τα περιττά στοιχεία και αχρείαστες λεπτομέρειες αποφεύγονται σε όλη τη υλοποίηση της εφαρμογής.
10. Επαρκής υποστήριξη - Βοήθεια και Εγχειρίδια: Ο χρήστης μέσα στην εφαρμογή θα μπορεί να δει και να αναγνωρίσει τα πεδία που πρέπει να συμπληρώσει πολύ εύκολα αφού σε κάθε ένα από αυτά υπάρχει η κατάλληλη βοήθεια και το ανάλογο hint

### **3.5 Βασικές Λειτουργίες Χρήστη**

Λειτουργία 1: Log in Ο κάθε γιατρός μπορεί να εισέλθει στο σύστημα με το να καταχωρήσει το username του σε συνδυασμό με τον προσωπικό του κωδικό.

Λειτουργία 2: Όταν ο γιατρός επιλέξει το κατάλληλο πεδίο από την εφαρμογή, θα του παρέχεται η δυνατότητα να δει τα προσωπικά του στοιχεία που έδωσε κατά την εγγραφή του στο σύστημα και εάν κάποια από αυτά δεν ισχύει πλέον θα μπορεί να τα τροποποιήσει. Αυτό δεν είναι εφικτό μόνο για το πεδίο της ταυτότητας για λόγους ασφάλειας.

Λειτουργία 3: Δημιουργία και καταχώριση νέου ασθενή

Λειτουργία 4: Εύρεση ασθενών με βάση την Ταυτότητα ή το τηλέφωνο τους

Λειτουργία 5: Προβολή και επεξεργασία των Προσωπικών δεδομένων του κάθε ασθενή.

Λειτουργία 6: Καταχώρηση νέου ραντεβού για ασθενείς

Λειτουργία 7: Ακύρωση κάποιου ραντεβού

Λειτουργία 8: Καταχώρηση αποτελέσματος για το κάθε ραντεβού

Λειτουργία 9: Προβολή Προγράμματος του γιατρού μέσω ενός ημερολογίου

### **3.6 Βασικές Λειτουργίες του Συστήματος**

1. Αλληλεπίδραση με το Keycloak για τη διασφάλιση των δεδομένων τόσο των γιατρών όσο και των ασθενών
2. Αλληλεπίδραση με τον FHIR Server με τη χρήση REST APIs για την άμεση ανταλλαγή δεδομένων του ασθενή και του γιατρού με το Σύστημα.

# Κεφάλαιο 4

## Λογισμικό Ανάπτυξης

### 4.1 Λογισμικό Ανάπτυξης

Για να προχωρούσα στην υλοποίηση και στην ανάπτυξη τόσο της τελικής μου εφαρμογής, όσο και στην υλοποίηση του FHIR πρωτοκόλλου έπρεπε πρώτα να προχωρήσω στην εγκατάσταση κάποιων απαραίτητων εργαλείων όπου θα βοηθούσαν τον σκοπό αυτό.

#### 4.1.1 XAMPP

Πριν το application μεταφερθεί στο Cloud Server, προχώρησα στην υλοποίηση τους σε τοπικό επίπεδο στον υπολογιστή μου. Έτσι με την βοήθεια του εργαλείου αυτού κατάφερα να ενεργοποιήσω apache server όπως και το MySQL database σε επίπεδο υπολογιστή, αφού το εργαλείο αυτό είναι ένας εικονικός εξυπηρετητής όπου μπορεί να εξυπηρετήσει βάσης δεδομένων τύπου MySQL καθώς και να ιστοσελίδων τύπου Apache. Τέλος μπορεί να επεξεργαστεί κώδικα πραγματισμού PHP. <sup>9</sup>[9]

#### 4.1.2 Visual Studio Code

Το third party application είναι υλοποιημένο σε PHP γλώσσα προγραμματισμού. Για να υλοποιήσω τα php files χρειαζόμουν κάποιο PHP IDE. Έτσι χρησιμοποίησα το Visual Studio Code το οποίο είναι δωρεάν και αρκετά 'ελαφρύ' για την ανάπτυξη PHP applications σε τοπικό επίπεδο.<sup>10</sup>[10]

#### 4.1.3 SOAP UI

Το SoapUI είναι ένα εργαλείο για δοκιμές των Web Services. Αυτές μπορούν να είναι SOAP Web Services καθώς RESTful Web Services ή HTTP. Το SoapUI είναι ένα Open Source και εντελώς δωρεάν εργαλείο και πολύ εύκολο στη χρήση. Για αυτό το λόγο και εγώ επέλεξα να χρησιμοποιήσω το συγκεκριμένο εργαλείο για να μπορώ να ελέγξω τα Web Services που προσφέρει το FHIR πρωτόκολλο.<sup>11</sup>[11]

## **4.2 Γλώσσες Προγραμματισμού**

Για την ανάπτυξη της εφαρμογής μου χρησιμοποίησα διαφορετικά είδη στις γλώσσες προγραμματισμού τα οποία θα αναλύσω πιο κάτω

### **4.2.1 PHP**

Η PHP είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML.<sup>12</sup>[12]

### **4.2.2 JavaScript**

Η JavaScript είναι παράλληλα με το HTML και το CSS, μία από τις βασικές τεχνολογίες του World Wide Web. Η JavaScript επιτρέπει στις ιστοσελίδες να είναι διαδραστικές και αποτελεί ουσιαστικό μέρος των εφαρμογών ιστού. Η συντριπτική πλειονότητα των ιστότοπων την χρησιμοποιούν για να αλληλεπιδρούν από τον χρήστη, και όλα τα μεγάλα προγράμματα περιήγησης στο Web διαθέτουν έναν ειδικό μηχανισμό JavaScript για να το εκτελέσουν.<sup>13</sup>[13]

## **4.3 Μοντελοποίηση δεδομένων**

Το FHIR πρωτόκολλο παρέχει την δυνατότητα να διαχειρίζεσαι τα δεδομένα σου με XML ή JSON μορφή.

### **4.3.1 XML**

Η XML (Extensible Markup Language) είναι μία γλώσσα σήμανσης, που περιέχει ένα σύνολο κανόνων για την ηλεκτρονική κωδικοποίηση κειμένων. Ορίζεται, κυρίως, στην προδιαγραφή XML 1.0 (XML 1.0 Specification), που δημιούργησε ο διεθνής οργανισμός 19 προτύπων W3C (World Wide Web Consortium), αλλά και σε διάφορες άλλες σχετικές προδιαγραφές ανοιχτών προτύπων. Η XML σχεδιάστηκε δίνοντας

έμφαση στην απλότητα, τη γενικότητα και τη χρησιμότητα στο Διαδίκτυο. Είναι μία μορφοποίηση δεδομένων κειμένου, με ισχυρή υποστήριξη Unicode για όλες τις γλώσσες του κόσμου. Αν και η σχεδίαση της XML εστιάζει στα κείμενα, χρησιμοποιείται ευρέως για την αναπαράσταση αυθαίρετων δομών δεδομένων, που προκύπτουν για παράδειγμα στις υπηρεσίες ιστού. Υπάρχει μία ποικιλία διεπαφών προγραμματισμού εφαρμογών, που μπορούν να χρησιμοποιούν οι προγραμματιστές, για να προσπελαίνουν δεδομένα XML, αλλά και διάφορα συστήματα σχημάτων XML, τα οποία είναι σχεδιασμένα για να βοηθούν στον ορισμό γλωσσών, που προκύπτουν από την XML. Έως το 2009, έχουν αναπτυχθεί εκατοντάδες γλώσσες που βασίζονται στην XML, συμπεριλαμβανομένων του RSS, του SOAP και της XHTML. Προεπιλεγμένες κωδικοποιήσεις βασισμένες στην XML, υπάρχουν για τις περισσότερες σουίτες εφαρμογών γραφείου, συμπεριλαμβανομένων του Microsoft Office (Office Open XML), του OpenOffice.org (OpenDocument) και του iWork της εταιρίας Apple.<sup>14</sup>[14]

#### **4.3.2 JSON(JavaScript Object Notation)**

Είναι μια ελαφριά μορφή κειμένου για ανταλλαγή δεδομένων. Είναι εύκολο για τους ανθρώπους να διαβάζουν και να γράφουν, αλλά και για τα μηχανήματα να αναλύσουν και να τα δημιουργήσουν. Βασίζεται σε ένα υποσύνολο της JavaScript Programming Language Standard ECMA-262 3rd Edition - December 1999. Το JSON είναι μια μορφή κειμένου που είναι εντελώς ανεξάρτητη από τη γλώσσα προγραμματισμού, αλλά χρησιμοποιεί συμβάσεις που είναι γνωστές στους προγραμματιστές διάφορων γλωσσών, συμπεριλαμβανομένου της C, C ++, C #, Java, JavaScript, Perl, Python και πολλές άλλες. Αυτές οι ιδιότητες καθιστούν την JSON μορφή ως μια ιδανική γλώσσα ανταλλαγής δεδομένων.<sup>15</sup>[15]

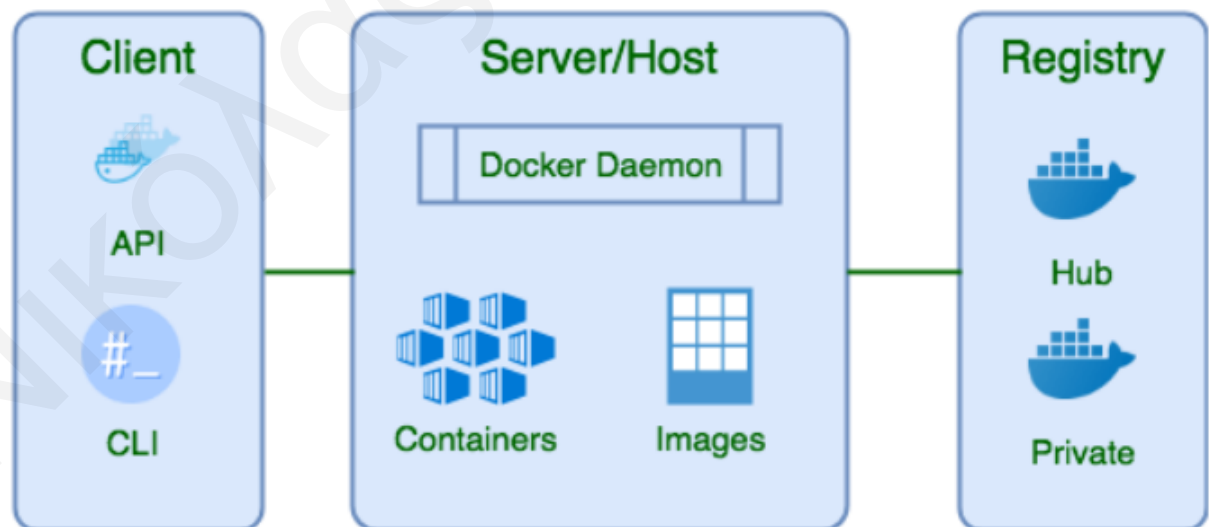
### 4.3.3 YAML

Είναι μια γλώσσα ανάγνωσης δεδομένων που είναι αναγνώσιμη από τον άνθρωπο. Χρησιμοποιείται συνήθως για configuration files και σε εφαρμογές όπου αποθηκεύονται ή μεταδίδουν δεδομένα. Τη συγκεκριμένη μορφή δεδομένων στη δική μας περίπτωση την χρησιμοποιούμε για τα Docker-compose files που θα αναλυθούν στη συνέχεια.<sup>16</sup>[16]

## 4.4 Τεχνολογίες

### 4.4.1 Docker

Το Docker είναι ένα εργαλείο που έχει σχεδιαστεί για να διευκολύνει τη δημιουργία, ανάπτυξη και εκτέλεση εφαρμογών χρησιμοποιώντας Containers. Τα Containers επιτρέπουν σε έναν προγραμματιστή να ομαδοποιεί μια εφαρμογή με όλα τα μέρη που χρειάζεται, όπως βιβλιοθήκες και άλλες εξαρτήσεις και να την αναπτύξει ως ένα πακέτο.



Docker Architecture Overview

Εικ. 4-1 Docker Architecture

Για να πετύχουμε αυτή την ομαδοποίηση των εφαρμογών μας με όλες τις εξαρτήσεις πρέπει μέσω των Dockerfiles να δημιουργήσουμε τα images.

#### 4.4.2 Docker Images

Τα Docker images αποτελούν τη βάση των Containers. Image είναι μια ταξινομημένη συλλογή των αλλαγών του συστήματος και οι παράμετροι εκτέλεσης που χρησιμοποιεί το Container όταν είναι σε χρόνο εκτέλεσης. Ένα image περιέχει συνήθως μια ένωση πολυστρωματικών συστημάτων αρχείων που στοιβάζονται το ένα πάνω στο άλλο. Επίσης, ένα image δεν έχει κατάσταση και δεν αλλάζει ποτέ.

<sup>17</sup>[17]

#### 4.4.3 Docker Container

Το container είναι ένα runtime instance από μία docker image.

Ένα Docker Container αποτελείται από

- Ένα Docker image
- Ένα περιβάλλον εκτέλεσης
- Ένα τυπικό σύνολο οδηγιών

Η ιδέα δανείζεται από Shipping Containers, τα οποία ορίζουν ένα πρότυπο για την αποστολή εμπορευμάτων παγκοσμίως. Το Docker καθορίζει ένα πρότυπο για το λογισμικό αποστολής.<sup>18</sup>[18]

#### 4.4.3 Docker-compose file

Το Compose είναι ένα εργαλείο για τον καθορισμό και την εκτέλεση πολλαπλών Docker container εφαρμογών. Με το Compose, χρησιμοποιούμε ένα αρχείο YAML για να διαμορφώσουμε τις υπηρεσίες της εφαρμογής μας. Στη συνέχεια, με μία μόνο εντολή, δημιουργούμε και ξεκινούμε όλες τις υπηρεσίες από το configuration μας.<sup>19</sup>[19]

#### 4.4.4 KeyCloak

Για την εφαρμογή του πρωτοκόλλου OAuth2, όπου θα αναλύσουμε στο επόμενο Κεφάλαιο, εφαρμόσαμε στην υλοποίηση του 3rd party application το Keycloak.



Εικ. 4-2 OAUTH2 Podocol

Το Keycloak είναι ένα open source Identity and Access Management solution, που στοχεύει σε σύγχρονα κυρίως applications και services. Διευκολύνει την ασφάλεια των applications όσο και των services με ελάχιστο προγραμματιστικό κόστος.

Αρχικά το Keycloak παρέχει στον χρήστη του τη δυνατότητα για μια Single-Sign On Solution. Οι χρήστες δηλαδή κάνουν έλεγχο ταυτότητας με το Keycloak και όχι με μεμονωμένες εφαρμογές. Με αυτό τον τρόπο οι εφαρμογές των Service Providers μας δεν πρέπει να ασχολούνται με σχεδίαση και υλοποίηση φορμών σύνδεσης, έλεγχο ταυτότητας, και δικαιωμάτων των χρηστών της. Από την διαδικασία αυτή φυσικά εξαιρείται και η ανάγκη για αποθήκευση σε συγκεκριμένη βάση δεδομένων των χρηστών για το κάθε application ξεχωριστά. Ένα ακόμη πλεονέκτημα του Keycloak είναι και η διευκόλυνση του χρήστη να μην χρειάζεται να συνδεθεί ξανά για να αποκτήσει πρόσβαση σε άλλη εφαρμογή, η οποία είναι επίσης συνδεδεμένη με το εργαλείο αυτό. Τα πιο πάνω ισχύουν και για την αποσύνδεση. Το Keycloak παρέχει απλή έξοδο, πράγμα που σημαίνει ότι οι χρήστες πρέπει να αποσυνδεθούν μόνο μία φορά για να αποσυνδεθούν από όλες τις εφαρμογές που χρησιμοποιούν το Keycloak.<sup>20</sup>[20]



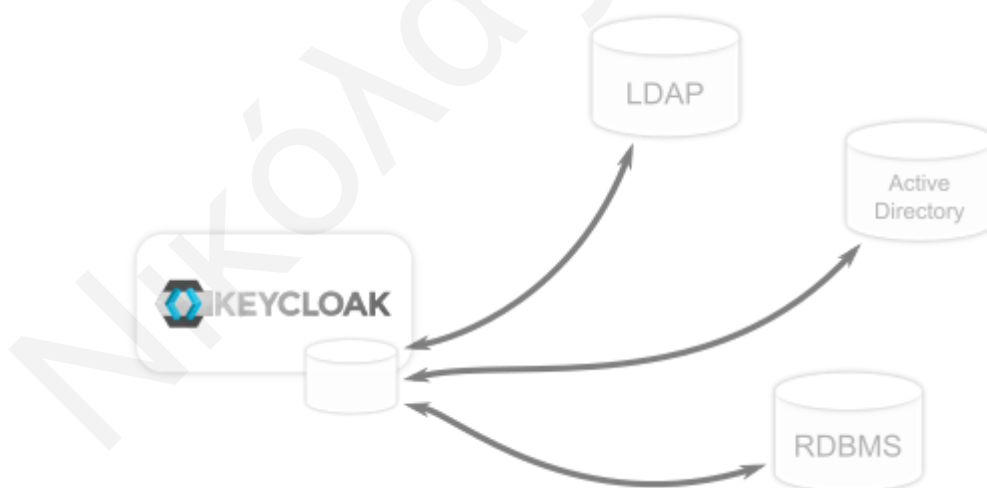
#### 4.4.4.1 Αλληλεπίδραση με Υφιστάμενα Applications

Το Keycloak, παρέχει επίσης, την δυνατότητα, αλληλεπίδρασης με συστήματα LDAP ή active directory. Τα συστήματα αυτά χρησιμοποιούνται κυρίως από μεγαλύτερους οργανισμούς, που απασχολούν περισσότερα άτομα, και οι έλεγχοι ταυτότητας και δικαιωμάτων των χρηστών γίνεται μέσω αυτών.

Επιπλέον, η ενεργοποίηση της σύνδεσης με κοινωνικά δίκτυα είναι εύκολο να προστεθεί μέσω της κονσόλας του διαχειριστή. Είναι απλώς θέμα επιλογής του κοινωνικού δικτύου που θέλετε να προσθέσετε. Δεν απαιτείται κωδικός ή αλλαγή στην εφαρμογή σας.

Το Keycloak μπορεί επίσης να πιστοποιήσει τους χρήστες με υπάρχοντες παρόχους OpenID Connect ή SAML 2.0. Και πάλι, πρόκειται απλώς για τη διαμόρφωση του Identity Provider μέσω της κονσόλας του διαχειριστή.

Με την πιο πάνω δυνατότητα ευκολά μπορεί να γίνει και η αλληλεπίδραση των υφιστάμενων applications με το Keycloak, αγνοώντας πως ο χρήστης προχωρούσε παλαιότερα στην σύνδεση με την εφαρμογή αυτή.



Εικ. 4-3 Keycloak

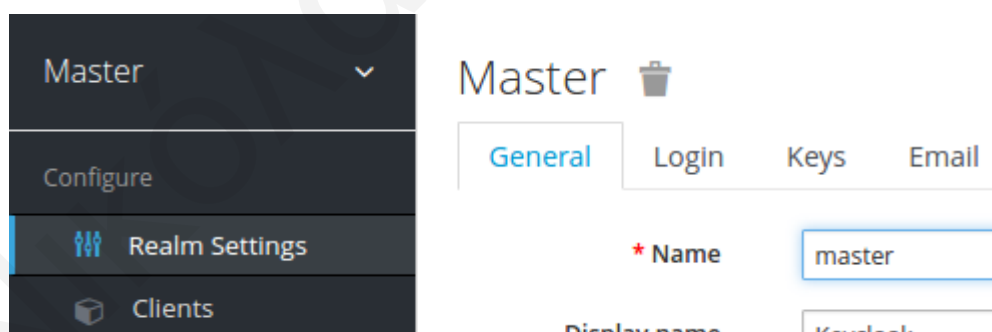
#### 4.4.4.2 Client Adapters

Μέσω των Keycloak Client Adapters καθίσταται πολύ εύκολο να ασφαλίσουμε τα applications και τα services. Το Keycloak διαθέτει adapters για αρκετές πλατφόρμες και γλώσσες προγραμματισμού. Εκτός από αυτούς όμως, το Keycloak είναι βασισμένο σε τυπικά πρωτόκολλα, ώστε να μπορούμε να χρησιμοποιήσουμε οποιαδήποτε βιβλιοθήκη OpenID Connect Resource Library or SAML 2.0 Service Provider library.

#### 4.4.4.3 Admin Console

Μέσω της σελίδας διαχείρισης, οι διαχειριστές μπορούν να επεξεργαστούν όλες τις πτυχές του Keycloak server.

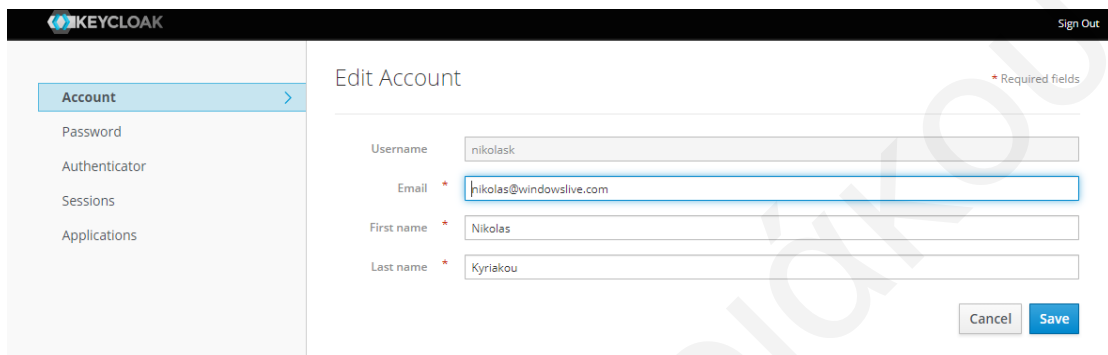
Μπορούν να ενεργοποιήσουν και να απενεργοποιήσουν διάφορες λειτουργίες. Μπορούν να διαμορφώσουν τη αλληλεπίδραση με του συστήματος με το LDAP ή το Active Directory. Επίσης, έχουν τη δυνατότητα να δημιουργήσουν και να διαχειριστούν εφαρμογές και υπηρεσίες, να ορίσουν λεπτομερείς πολιτικές για τα δικαιώματα του κάθε χρήστη ξεχωριστά ακόμα και για τα sessions.



Εικ. 4-4 Keycloak-Admin Console

#### 4.4.4.4 Account Management Console

Μέσω της κονσόλας διαχείρισης λογαριασμών οι χρήστες μπορούν να διαχειρίζονται τους δικούς τους λογαριασμούς. Μπορούν να ενημερώσουν το προφίλ, να αλλάξουν κωδικούς πρόσβασης και να ρυθμίσουν τον έλεγχο ταυτότητας δύο παραγόντων. Οι χρήστες μπορούν επίσης να διαχειρίζονται sessions καθώς και να βλέπουν ιστορικό για τον λογαριασμό τους.



The screenshot displays the Keycloak Account Management Console. The interface features a dark header with the 'KEYCLOAK' logo on the left and a 'Sign Out' link on the right. A left-hand navigation menu is visible, with 'Account' selected and highlighted in blue. Below the menu, the main content area is titled 'Edit Account' and includes a red asterisk indicating required fields. The form contains four input fields: 'Username' (value: nikolask), 'Email' (value: nikolas@windowslive.com), 'First name' (value: Nikolas), and 'Last name' (value: Kyriakou). At the bottom right of the form are 'Cancel' and 'Save' buttons.

Εικ. 4-5 Keycloak-Account Management

# Κεφάλαιο 5

## Σχεδίαση Συστήματος

Ο σχεδιασμός του συστήματος είναι η διαδικασία που μας επιτρέπει να φτάσουμε στη λύση που ταιριάζει καλύτερα στις απαιτήσεις του προβλήματος και στους περιορισμούς που έχουν ήδη καθοριστεί. Είναι η διαδικασία κατά την οποία γίνονται διαδοχικά καλύτερες προσεγγίσεις του προβλήματος και γίνεται η εύρεση της τελικής μορφής που θα έχει το σύστημα.

Παράγοντες που λαμβάνονται υπόψιν κατά τη σχεδίαση του συστήματος είναι οι απαιτούμενες λειτουργίες που χρειάζεται το σύστημα, το επίπεδο απόδοσης, η συχνότητα επεξεργασίας και ο βαθμός εξυπηρέτησης των χρηστών καθώς και η ασφάλεια των δεδομένων.

### 5.1 3<sup>rd</sup> Party Service Provider

Το σύστημα αυτό, σύμφωνα με τις προδιαγραφές που ορίσαμε θα πρέπει να είναι ένα Web Application, για να μπορεί να μας εξασφαλίσει την διαλειτουργικότητα που είναι μία από τις βασικές μας προτεραιότητες. Με αυτό τον τρόπο ο γιατρός – χρήστης του συστήματος, θα μπορεί να έχει πρόσβαση στα δεδομένα που παρέχονται μέσω αυτού ανά πάσα χρονική στιγμή από οποιαδήποτε συσκευή θελήσει να αποκτήσει πρόσβαση.

### 5.2 Ehealth4u FHIR Server

Σύμφωνα με τις προδιαγραφές και τις λειτουργίες που ορίσαμε πιο πάνω, και θα πρέπει το σύστημα να εκτελεί, τα resources που θα πρέπει να χρησιμοποιήσουμε από το FHIR πρωτόκολλο είναι τα ακόλουθα:

- Practitioner
- Patient

- Appointment

### Practitioner:

Ο βασικός στόχος του Resource αυτού είναι να αντιπροσωπεύσει/αποθηκεύσει όλα τα στοιχεία που αφορούν τον κάθε γιατρό ξεχωριστά και να του δώσει μια ψηφιακή ταυτότητα ούτως ώστε η μονάδα αυτή να είναι προς βάσιμη από οποιοδήποτε application.

Τα στοιχεία που μπορούμε να αποθηκεύσουμε για τον κάθε γιατρό φαίνονται πιο κάτω σε Json μορφή.<sup>21</sup>[21]

```
{
  "resourceType" : "Practitioner",
  // from Resource: id, meta, implicitRules, and language
  // from DomainResource: text, contained, extension, and modifierExtension
  "identifier" : [{ Identifier }], // An identifier for the person as this agent
  "active" : <boolean>, // Whether this practitioner's record is in active use
  "name" : [{ HumanName }], // The name(s) associated with the practitioner
  "telecom" : [{ ContactPoint }], // A contact detail for the practitioner (that apply to all roles)
  "address" : [{ Address }], // Address(es) of the practitioner that are not role specific (typically home address)
  "gender" : "<code>", // male | female | other | unknown
  "birthDate" : "<date>", // The date on which the practitioner was born
  "photo" : [{ Attachment }], // Image of the person
  "qualification" : [{ // Certification, licenses, or training pertaining to the provision of care
    "identifier" : [{ Identifier }], // An identifier for this qualification for the practitioner
    "code" : { CodeableConcept }, // R! Coded representation of the qualification
    "period" : { Period }, // Period during which the qualification is valid
    "issuer" : { Reference(Organization) } // Organization that regulates and issues the qualification
  }],
  "communication" : [{ CodeableConcept }] // A language the practitioner can use in patient communication
}
```

### Patient:

Αντίστοιχα, όπως και στον Practitioner, ο βασικός στόχος του Resource αυτού είναι να αντιπροσωπεύσει/αποθηκεύσει όλα τα στοιχεία που αφορούν τον κάθε ασθενή ξεχωριστά για να μπορεί να αποτυπωθεί ψηφιακά η εικόνα των ασθενών.

Τα στοιχεία που μπορούμε να αποθηκεύσουμε σε αυτό το Resource είναι:<sup>22</sup>[22]

```
{
  "resourceType" : "Patient",
  // from Resource: id, meta, implicitRules, and language
  // from DomainResource: text, contained, extension, and modifierExtension
  "identifier" : [{ Identifier }], // An identifier for this patient
  "active" : <boolean>, // Whether this patient's record is in active use
  "name" : [{ HumanName }], // A name associated with the patient
  "telecom" : [{ ContactPoint }], // A contact detail for the individual
  "gender" : "<code>", // male | female | other | unknown
  "birthDate" : "<date>", // The date of birth for the individual
  // deceased[x]: Indicates if the individual is deceased or not. One of these 2:
  "deceasedBoolean" : <boolean>,
  "deceasedDateTime" : "<dateTime>",
  "address" : [{ Address }], // An address for the individual
  "maritalStatus" : { CodeableConcept }, // Marital (civil) status of a patient
  // multipleBirth[x]: Whether patient is part of a multiple birth. One of these 2:
  "multipleBirthBoolean" : <boolean>,
  "multipleBirthInteger" : <integer>,
  "photo" : [{ Attachment }], // Image of the patient
  "contact" : [{ // A contact party (e.g. guardian, partner, friend) for the patient
    "relationship" : [{ CodeableConcept }], // The kind of relationship
    "name" : { HumanName }, // A name associated with the contact person
    "telecom" : [{ ContactPoint }], // A contact detail for the person
    "address" : { Address }, // Address for the contact person
    "gender" : "<code>", // male | female | other | unknown
    "organization" : { Reference(Organization) }, // C? Organization that is associated
    with the contact
    "period" : { Period } // The period during which this contact person or organization
    is valid to be contacted relating to this patient
  }],
  "communication" : [{ // A language which may be used to communicate with the patient
    about his or her health
    "language" : { CodeableConcept }, // R! The language which can be used to
    communicate with the patient about his or her health
    "preferred" : <boolean> // Language preference indicator
  }],
}
```

```

"generalPractitioner" : [{ Reference(Organization|Practitioner|
  PractitionerRole) }], // Patient's nominated primary care provider
"managingOrganization" : { Reference(Organization) }, // Organization that is the
custodian of the patient record
"link" : [{ // Link to another patient resource that concerns the same actual person
  "other" : { Reference(Patient|RelatedPerson) }, // R! The other patient or related
person resource that the link refers to
  "type" : "<code>" // R! replaced-by | replaces | refer | seealso
}]
}

```

## Appointment:

Σε αυτό το Resource έχουμε τη δυνατότητα να αποθηκεύσουμε όλα τα απαραίτητα στοιχεία που χρειαζόμαστε για να ορίσουμε ένα ραντεβού. Τα στοιχεία αυτά φαίνονται στο πιο κάτω Json.<sup>23</sup>[23]

```

{
  "resourceType" : "Appointment",
  // from Resource: id, meta, implicitRules, and language
  // from DomainResource: text, contained, extension, and modifierExtension
  "identifier" : [{ Identifier }], // External Ids for this item
  "status" : "<code>", // R! proposed | pending | booked | arrived | fulfilled |
cancelled | noshow | entered-in-error | checked-in | waitlist
  "cancellationReason" : { CodeableConcept }, // The coded reason for the appointment
being cancelled
  "serviceCategory" : [{ CodeableConcept }], // A broad categorization of the service
that is to be performed during this appointment
  "serviceType" : [{ CodeableConcept }], // The specific service that is to be performed
during this appointment
  "specialty" : [{ CodeableConcept }], // The specialty of a practitioner that would be
required to perform the service requested in this appointment
  "appointmentType" : { CodeableConcept }, // The style of appointment or patient that
has been booked in the slot (not service type)
  "reasonCode" : [{ CodeableConcept }], // Coded reason this appointment is scheduled
  "reasonReference" : [{ Reference(Condition|Procedure|Observation|
  ImmunizationRecommendation) }], // Reason the appointment is to take place (resource)
  "priority" : "<unsignedInt>", // Used to make informed decisions if needing to re-
prioritize
  "description" : "<string>", // Shown on a subject line in a meeting request, or
appointment list
  "supportingInformation" : [{ Reference(Any) }], // Additional information to support
the appointment
  "start" : "<instant>", // When appointment is to take place
  "end" : "<instant>", // When appointment is to conclude

```

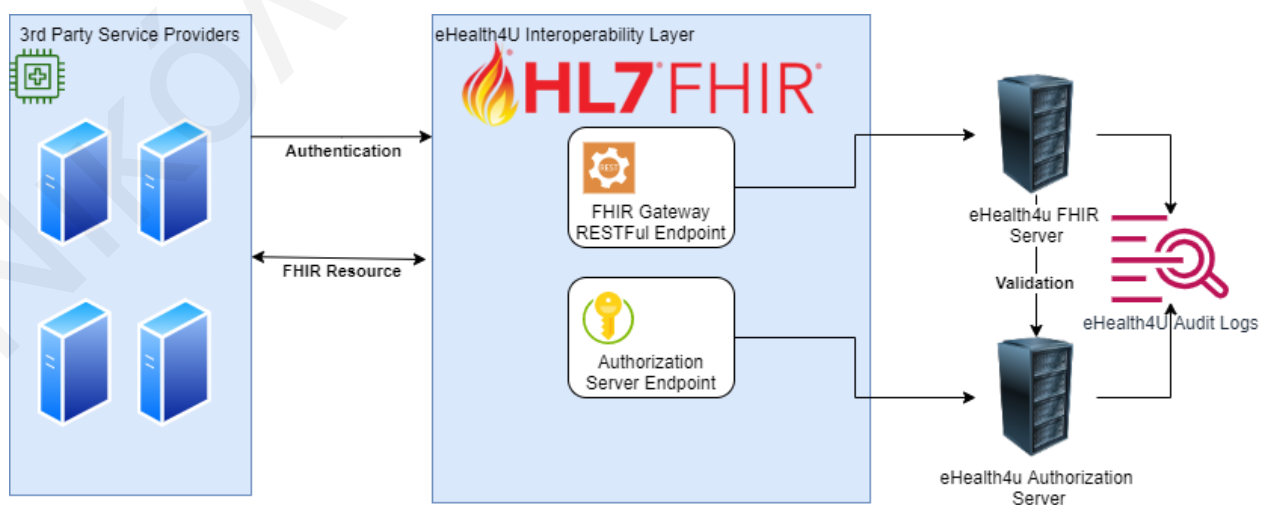
```

"minutesDuration" : "<positiveInt>", // Can be less than start/end (e.g. estimate)
"slot" : [{ Reference(Slot) }], // The slots that this appointment is filling
"created" : "<dateTime>", // The date that this appointment was initially created
"comment" : "<string>", // Additional comments
"patientInstruction" : "<string>", // Detailed information and instructions for the patient
"basedOn" : [{ Reference(ServiceRequest) }], // The service request this appointment is allocated to assess
"participant" : [{ // R! Participants involved in appointment
  "type" : [{ CodeableConcept }], // Role of participant in the appointment
  "actor" : { Reference(Patient|Practitioner|PractitionerRole|RelatedPerson|Device|HealthcareService|Location) }, // Person, Location/HealthcareService or Device
  "required" : "<code>", // required | optional | information-only
  "status" : "<code>", // R! accepted | declined | tentative | needs-action
  "period" : { Period } // Participation period of the actor
}],
"requestedPeriod" : [{ Period }] // Potential date/time interval(s) requested to allocate the appointment within
}

```

### 5.3 Αρχιτεκτονική-Ασφάλεια Συστήματος

Η συνολική αρχιτεκτονική του τελικού συστήματος, και η επικοινωνία του 3<sup>rd</sup> Party Service Provider με το eHealth4u FHIR πρωτόκολλο καθώς και η εξασφάλιση της ασφάλειας των δεδομένων, που απαιτείται επίσης από τις προδιαγραφές που ορίσαμε πιο πάνω, φαίνεται στο πιο κάτω διάγραμμα.



Εικ. 5-1 FHIR Architecture



Τα εμπλεκόμενα μέλη είτε αυτοί είναι ασθενείς, είτε γιατροί μέσω ενός 3<sup>rd</sup> party Service Provider (mobile application / Web Application /Desktop Application) θα έχουν τη δυνατότητα για αλληλοεπιδρούν (Read, Update, Delete Post) με τα δεδομένα που βρίσκονται αποθηκευμένα στον FHIR Server μας και πιο συγκεκριμένα στην eHealth4u FHIR server.

Το πιο πάνω σενάριο από μόνο του δημιουργεί κάποια θέματα ασφαλείας στην όλη λειτουργία του πρωτοκόλλου αφού οι 3<sup>rd</sup> party Service Providers μπορεί να είναι οποιοσδήποτε οπουδήποτε. Έτσι για να περιορίσουμε την δυνατότητα αλληλεπίδρασης των 3<sup>rd</sup> party Service Providers με τα FHIR Data χρειάστηκε η δημιουργία ενός επιπλέον Layer ανάμεσα στα 2 εμπλεκόμενα μέλη. Το Layer αυτό είναι το FHIR Gateway, όπου ο κύριος του στόχος είναι να εξουσιοδοτεί το 3<sup>rd</sup> party Service Provider ούτως ώστε αυτός να μπορεί να στείλει requests στον FHIR Server ανάλογα φυσικά με τα permissions τα οποία έλαβε προηγουμένως.

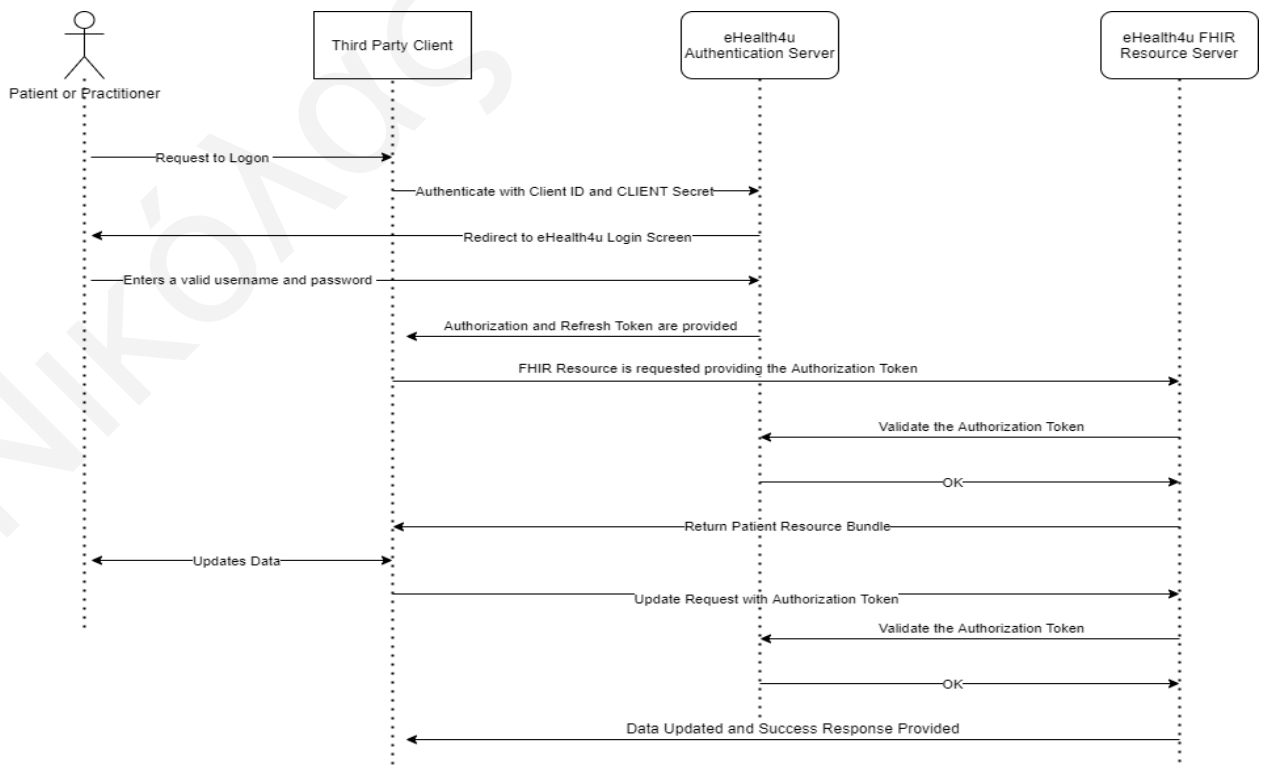
Το πιο πάνω πρόβλημα επιλύθηκε από την υλοποίηση που εφαρμόστηκε από την eHealth4u platform όπου βασίζεται στο ανοιχτό πρότυπο RFC6749 OAUTH2 για Authorization που βασίζονται στο ανοιχτό πρότυπο RFC7519 JSON Web Token (JWT). Το JWT είναι ένας αυτόνομος τρόπος για την ασφαλή μετάδοση πληροφοριών μεταξύ των εμπλεκόμενων μερών, όπου τα μέλη αυτά είναι εξουσιοδοτημένα και επικυρωμένα.

Μετά την επιτυχή εγγραφή, του 3<sup>rd</sup> party Service Provider, στην πλατφόρμα eHealth4u, ο διαχειριστής θα παρέχει στον εκπρόσωπο ασφαλείας του τρίτου μέρους ένα μοναδικό αναγνωριστικό πελάτη και ένα μυστικό κωδικό που μπορεί να εναλλάσσεται περιοδικά.

Αυτό το αναγνωριστικό πελάτη και ο μυστικός κωδικός θα επιτρέψει την εξουσιοδοτημένη πρόσβαση του 3<sup>rd</sup> party System στην υπηρεσία eHealth4u FHIR σε συνδυασμό με τα έγκυρα πιστοποιητικά του χρήστη του συστήματος είτε πρόκειται για ιατρό είτε για ασθενή.

Η ροή του πρωτοκόλλου OAuth2:

- Ένας ασκούμενος / ασθενής ζητά να συνδεθεί στο third-party system.
- Το third-party system στη συνέχεια προωθεί τον χρήστη στο login page του authorization server με το μοναδικό client id and client secret στην υπηρεσία ασφαλείας eHealth4u με μια διεύθυνση URL που δημιουργήθηκε μοναδικά για το client id μέσω ενός κρυπτογραφημένου καναλιού SSL TLS 1.2
- Ο eHealth4u Authorization server θα παρέχει στο 3<sup>rd</sup> party system έναν έγκυρο authentication και ένα token.
- Το Authentication token έχει συγκεκριμένη διάρκεια ζωής και όταν λήξει, ο χρήστης πρέπει να αιτηθεί τη δημιουργία ενός νέου token. Επιπλέον, μετά τη λήξη του token, θα ζητηθεί εκ νέου από τον χρήστη να πραγματοποιήσει ξανά έλεγχο ταυτότητας, διαφορετικά η πρόσβαση δεν θα επιτρέπεται.
- Μόλις το τρίτο μέρος έχει το έγκυρο token, τότε μπορεί να στείλει ένα αίτημα στο τελικό σημείο eHealth4u FHIR REST, το οποίο θα παρέχει απάντηση σύμφωνα με την προδιαγραφή HL7 FHIR.<sup>24</sup>[24]



Εικ. 5-2 User cases

Η μορφή του JSON Web Token είναι η ακόλουθη:

```
{ "access_token": "eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkiaA6ICJvS0YzYmFtNndYUGZ  
DZ2Q2NHY0S2F3NDNqdjd2ODFzeGQ3eksxeGVsNlZrIn0 ",  
"expires_in":300,  
"refresh_expires_in":1800,  
"refresh_token":"eyJhbGciOiJIUzI1NiIsInR5cCIgOiAiSldUIGwiwia2lkiaA6ICI1NDk1ZThhYS0yZGQy  
LTQ1MDgtYTJhYy0zNDliNGM0ZDYzMmQifQ",  
"token_type":"bearer",  
"not-before-policy":0,  
"session_state":"b1c99c99-36dd-4367-b6f2-4badf929c4a0",  
"scope":"profile email" }
```

# Κεφάλαιο 6

## Υλοποίηση Συστήματος

Στη αυτή τη φάση ανάπτυξης του συστήματος, αφού έχουμε οριστεί από πριν τους χρήστες, τις ανάγκες, τις λειτουργίες, τις προδιαγραφές και τους περιορισμούς του συστήματος, είμαστε σε θέση να το υλοποιήσουμε.

Σε αυτό λοιπόν το σημείο της ανάπτυξης του συστήματος, γίνεται η κωδικοποίηση των διαφόρων μερών του λογισμικού καθώς και η σύνδεση των διάφορων εργαλείων που θα χρησιμοποιήσουμε πιο συγκεκριμένα γίνεται η σύνδεση της εφαρμογής μας με το eHealth4u FHIR Server και το Keycloak.

Στο κεφάλαιο αυτό θα επεξηγηθούν τα βήματα που ακολουθήθηκαν για την υλοποίηση του συστήματος, οι έλεγχοι που έγιναν και πραγματοποιήθηκαν καθώς επίσης και θα παρουσιαστούν οι φόρμες του τελικού συστήματος.

### 6.1 Κωδικοποίηση Ανάπτυξη 3<sup>rd</sup> Party Service Provider

#### 6.1.1 Front-End

Για να αναπτυχθεί το front-end μέρος του συστήματος και να μπορέσουμε με αυτό τον τρόπο να πετύχουμε την εύκολη αλληλεπίδραση με τον χρήστη, χρησιμοποιήσα Bootstrap framework. Το Bootstrap είναι ένα free open-source framework, το οποίο απευθύνεται για την ανάπτυξη ιστοσελίδων με δυναμικά χαρακτηριστικά. Βασίζεται σε σύγχρονες τεχνολογίες HTML5, CSS και Javascript για πλοήγηση αλλά και σχεδιασμό φορμών, κουμπιών, λιστών και καλύπτει ένα πλήρες φάσμα λειτουργιών που χρειάζεται μια ιστοσελίδα.<sup>25</sup>

#### 6.1.2 Calendar

Για την ανάπτυξη του ημερολογίου και να μπορέσω να πετύχω την άμεση και δυναμική αλληλεπίδραση χρήστη - συστήματος χρησιμοποιήσα την βιβλιοθήκη του

fullcalendar. Πρόκειται για μια free open-source βιβλιοθήκη, η οποία βοηθά τους προγραμματιστές να αναπτύξουν ημερολόγια στις εφαρμογές τους ανάλογα με τις προτιμήσεις και τους περιορισμούς τους.<sup>26</sup>[26]

Οι βασικές της λειτουργίες είναι η προβολή των διάφορων events που μπορεί να φιλοξενήσει οργανωμένα ανά μήνα, βδομάδα, μέρα ή ταξινομημένα σε μια λίστα. Επίσης, μπορεί να δημιουργήσει, διαγράψει ή αλλάξει προγραμματισμένα ραντεβού. Οι πιο πάνω λειτουργίες γίνονται μέσω Rest APIs.

### 6.1.3 Back-End

Για να μπορέσει η εφαρμογή μας το, 3<sup>rd</sup> party service provider, να επικοινωνήσει και να αλληλοεπιδράσει με τον eHealth4u FHIR Server αλλά και με Keycloak δημιουργήσα κάποια Rest APIs τα οποία καλούνται από την Front-End κωδικοποίηση για να πετύχουν τον εκάστοτε σκοπό τους.

Το Back-End κομμάτι του κώδικα καλεί τα Rest APIs που προσφέρονται από το eHealth4u FHIR πρωτόκολλο. Αυτό γίνεται μέσω Curl command το οποίο είναι ένα open-source software το οποίο χρησιμοποιείται για μεταφορά δεδομένων.

Στη δική μας περίπτωση καλώ τα Rest APIs με το αντίστοιχο Request Action:

- GET – διάβασμα δεδομένων
- POST – Εγγραφή νέων δεδομένων
- PUT – Ανανέωση υφιστάμενων δεδομένων
- DELETE – Διαγραφή υφιστάμενων δεδομένων

Επιπλέον, για την επικοινωνία του δικού μας application μαζί με τον FHIR Server θα πρέπει στο header για κάθε request να περιέχεται:

- Content-Type: application/json -> ενημερώνει τον FHIR Server ότι η μορφή που στέλνεται το Request δηλαδή json.
- Authorization: Bearer + token -> Αυτό χρησιμοποιείται αφού, όπως αναλύσαμε πιο πάνω, ο FHIR Server μόλις λάβει κάποιο Request από οποιοδήποτε Provider πριν εκτελέσει το Request πρέπει να ελέγξει ότι το

token που έλαβε είναι ορθό. Σε αντίθετη περίπτωση απορρίπτει το Request. Επιπλέον, σε περίπτωση που ο Service Provider δεν στείλει στο header του Request το συγκεκριμένο πεδίο, ο FHIR Server ενημερώνει τον Provider ότι το Request του δεν μπορεί να εκτελεστεί γιατί ο χρήστης δεν είναι εγκεκριμένος.

- Cache-Control: no-store -> Η συγκεκριμένη παράμετρος στέλνεται στα Requests τα οποία προκαλούν αλλαγές στα δεδομένα του FHIR Server (POST,PUT,DELETE). Αυτό συμβαίνει αφού λόγο Caching παρατηρήσαμε μια καθυστέρηση από τον FHIR Server να ενημερώσει τα δεδομένα του, παρόλο που ο Service Provider λαμβάνει response ότι στα δεδομένα έγινε επιτυχώς η ανανέωση τους.

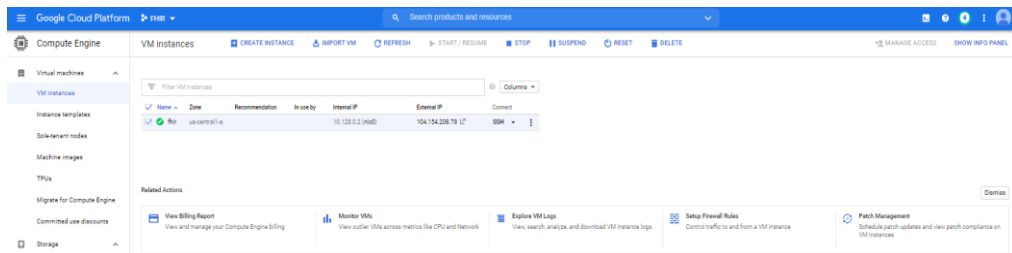
Τέλος, η επικοινωνία του application και του FHIR Server για να μπορέσει να πραγματοποιηθεί, θα πρέπει για τα Requests που προκαλούν αλλαγές στα δεδομένα (POST,PUT) να αποστέλλεται στο body του Request το αντίστοιχο JSON.

## 6.2 Εγκατάσταση Εφαρμογής

Για την εγκατάσταση της εφαρμογής και την ομαλή λειτουργία του όλου συστήματος θα χρειαστούμε ένα apache Server καθώς και ένα Version PHP γλώσσας Προγραμματισμού.

Στην δική μου περίπτωση, χρησιμοποίησα Apache/2.4.18 (Ubuntu) καθώς και την PHP 7.0.33 Version. Παρόλα αυτά το application μπορεί να χρησιμοποιηθεί σε οποιαδήποτε νεότερη PHP version.

Για τα δύο αυτά προ-απαιτούμενα χρησιμοποίησα το Google-Cloud για να μπορέσω να κάνω host την μηχανή μου.



Εικ. 6-1 Google – Cloud Platform

Η δημιουργία του Google-Cloud Account και της μηχανής είναι αρκετά εύκολη για τον οποιοδήποτε. Χρειάζεται αρχικά να δώσεις τα προσωπικά σου στοιχεία για την δημιουργία του λογαριασμού και στην συνέχεια, με την ενεργοποίηση του Billing Account παρέχεται από την πλατφόρμα μια δοκιμαστική δωρεάν περίοδος. Ακολούθως, στο menu μπορείς να επιλέξεις το παράθυρο του Virtual Machine και έτσι μπορείς να δημιουργήσεις τον Server ανάλογα με τις απαιτήσεις των εφαρμογών μας. Με αυτό ακριβώς τον τρόπο δημιούργησα ένα Ubuntu Server με IP: 104.154.208.78. Τέλος, εγκατέστησα την PHP με τα πιο κάτω commands:<sup>27</sup>[27]

- `sudo add-apt-repository ppa:ondrej/php`
- `sudo apt-get update`
- `sudo apt-get install php7.0`
- `sudo apt-get install php7.0-curl`

Με όλα τα πιο πάνω βήματα μπόρεσα να ετοιμάσω το production-environment μου για να μπορέσει να φιλοξενήσει την εφαρμογή μου. Τελευταίο βήμα ήταν να μεταφέρω το project από το τοπικό μου περιβάλλον στον /var/www/html/ path του Server όπου είναι το root path για ένα apache-server.

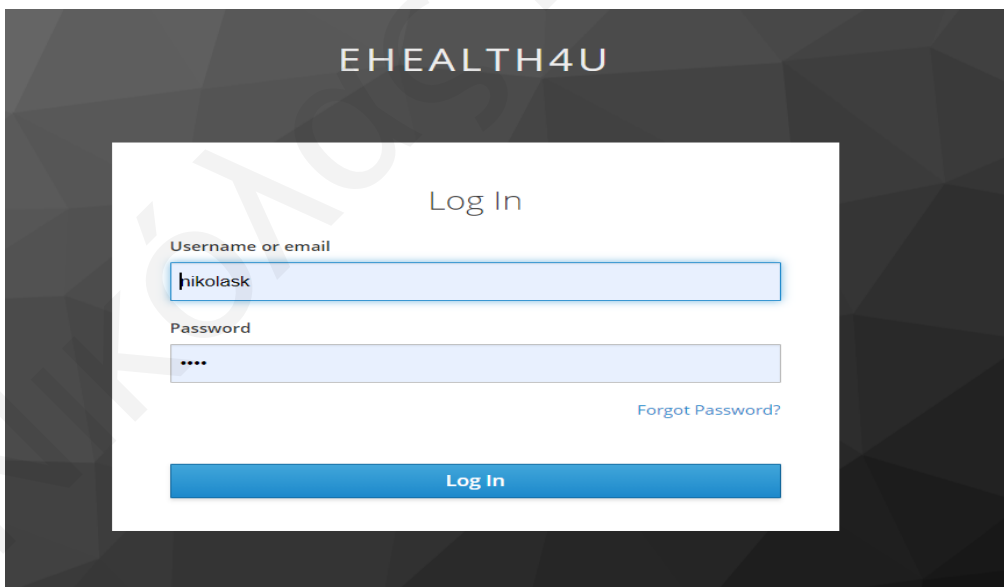
Όπως προανάφερα πιο πάνω για τον test-environment μου χρησιμοποίησα τον δικό μου test-FHIR και όχι τον ehalth4u. Για να το πετύχω αυτό, χρησιμοποίησα docker-compose file, με το οποίο κατάφερα τη δημιουργία ενός docker-container που έκανε host τον FHIR-Server. Για να μπορέσω όμως να το πετύχω αυτό εγκατέστησα τα Dockers στον Server μου με τα ακόλουθα commands:<sup>28</sup>[28]

- `sudo apt-get install docker-engine -y`
- `sudo service docker start`
- `sudo curl -L`  
`https://github.com/docker/compose/releases/download/1.21.2/docker-`  
`compose-`uname -s`-`uname -m` -o /usr/local/bin/docker-compose`
- `sudo chmod +x /usr/local/bin/docker-compose`

## 6.3 Περιγραφή Συστήματος

### 6.3.1 Log in:

Για να μπορεί οποιοσδήποτε να χρησιμοποιήσει το σύστημα αυτό απαιτείται η δημιουργία λογαριασμού στο Keycloak. Με την επιτυχή σύνδεση, το Keycloak επιστρέφει στοιχεία του user στην εφαρμογή μας, ένα από αυτά είναι το email του γιατρού. Ακολούθως, το σύστημα απευθύνεται στον FHIR server, όπου ανακτά τα στοιχεία του user μας.



Εικ. 6-2 Login Page



### 6.3.2 Υπόλοιπες Λειτουργίες Εφαρμογής

Στην πιο κάτω εικόνα μπορούμε να διακρίνουμε όλες τις κύριες λειτουργίες της εφαρμογής μας. Στην αριστερή μεριά της οθόνης υπάρχει το menu για την εύκολη πλοήγηση στις λειτουργίες μας.

Αυτές μπορούμε να τις χωρίσουμε σε δύο υποκατηγορίες.

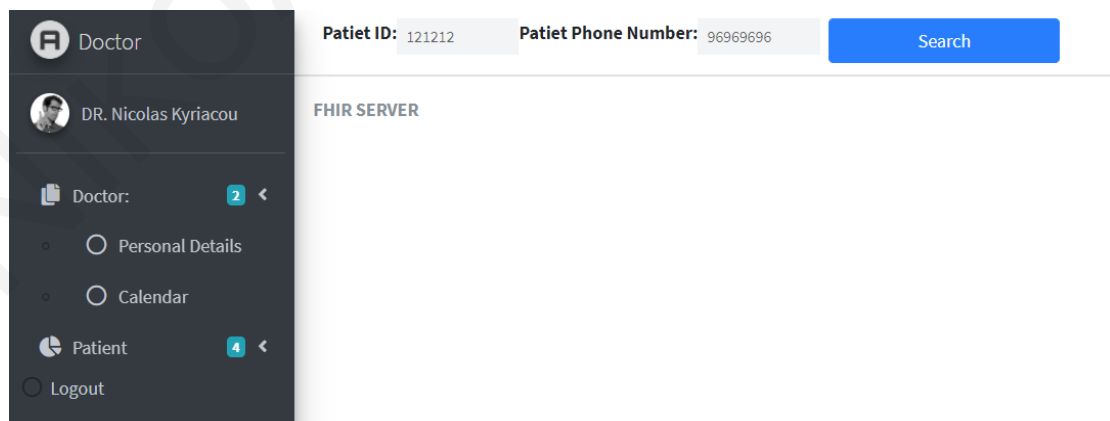
Πρώτα τις λειτουργίες που αφορούν άμεσα τον γιατρό:

- Προσωπικές Πληροφορίες για τον ίδιο
- Το ημερολόγιο με τα ραντεβού του.

Η δεύτερη υποκατηγορία αφορά της λειτουργίες που έχουν να κάνουν με τον ασθενή

- Προσωπικές Πληροφορίες ασθενή
- Ορισμός ραντεβού Ασθενή
- Ιστορικό των Ραντεβού για κάθε Ασθενή
- Δημιουργία νέου ασθενή
- Λίστα με όλους τους ασθενής

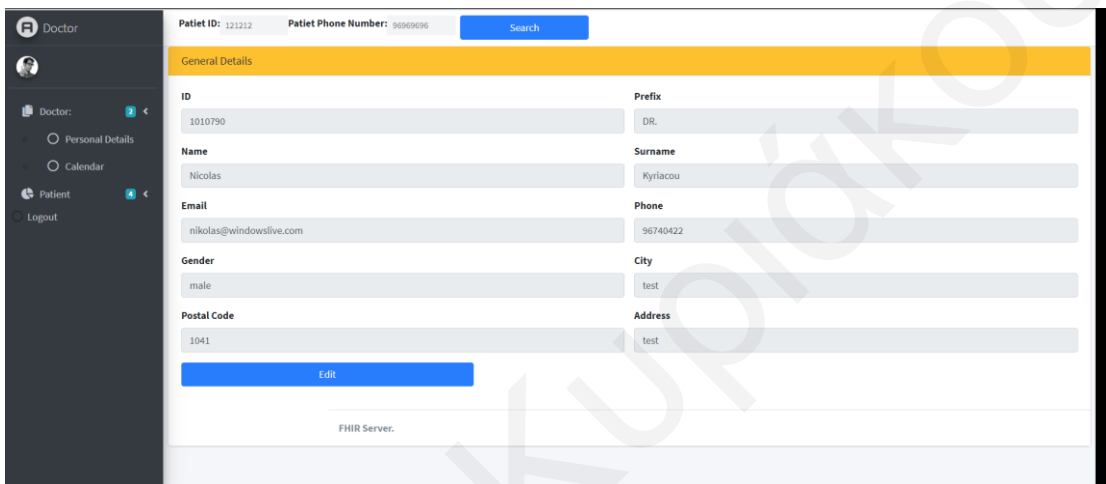
Επίσης, στο πάνω μέρος της σελίδας υπάρχει η επιλογή για αναζήτηση οποιουδήποτε ασθενή με βάση την ταυτότητα ή το τηλέφωνο του. Επέλεξα, τα δύο αυτά στοιχεία για αναζήτηση αφού αυτά μπορούν να καθορίσουν εύκολα την μοναδικότητα του κάθε ασθενή, καθώς επίσης γιατί τα στοιχεία αυτά μπορούν να εντοπιστούν εύκολα χωρίς την φυσική παρουσία των ασθενών στο χώρο.



Εικ. 6-3 Main Doctor Page

### 6.3.2.1 Στοιχεία γιατρού

Στην πιο κάτω σελίδα μπορεί ο user / γιατρός να δει αλλά και να τροποποιήσει τα προσωπικά του στοιχεία. Με το πάτημα του κουμπιού Edit όλα τα Textbox, εκτός από την ταυτότητα, μετατρέπονται σε editable και ταυτόχρονα εμφανίζεται ένα 2ο κουμπί (Save) για να μπορεί ο user να αποθηκεύσει στον FHIR server μας τα νέα ανανεωμένα στοιχεία του γιατρού.



The screenshot shows a web interface for a doctor's profile. On the left is a dark sidebar with navigation options: Doctor (selected), Personal Details, Calendar, Patient, and Logout. The main content area has a header with 'Patient ID: 121212' and 'Patient Phone Number: 96969696' next to a 'Search' button. Below this is a yellow header for 'General Details'. The form fields are arranged in two columns:

Field	Value
ID	1010790
Prefix	DR.
Name	Nicolas
Surname	Kyriacou
Email	nikolas@windowslive.com
Phone	96740422
Gender	male
City	test
Postal Code	1041
Address	test

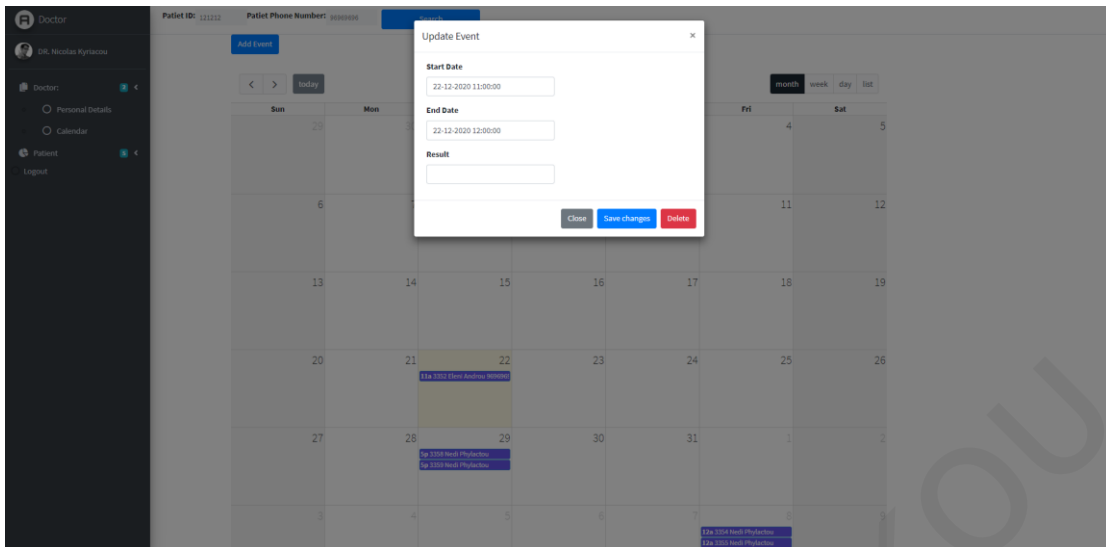
At the bottom of the form is a blue 'Edit' button. Below the form, there is a section for 'FHIR Server'.

Εικ. 6-4 Doctor Details

### 6.3.2.2 Ημερολόγιο Γιατρού

Στην σελίδα αυτή ο γιατρός μπορεί να ελέγχει το πρόγραμμα με τα ραντεβού του για τους επόμενους μήνες και ταυτόχρονα μπορεί να ολοκληρώσει το κάθε ραντεβού και να περάσει στον FHIR Server τα τελικά αποτελέσματα/διαγνώσεις για τα ραντεβού.

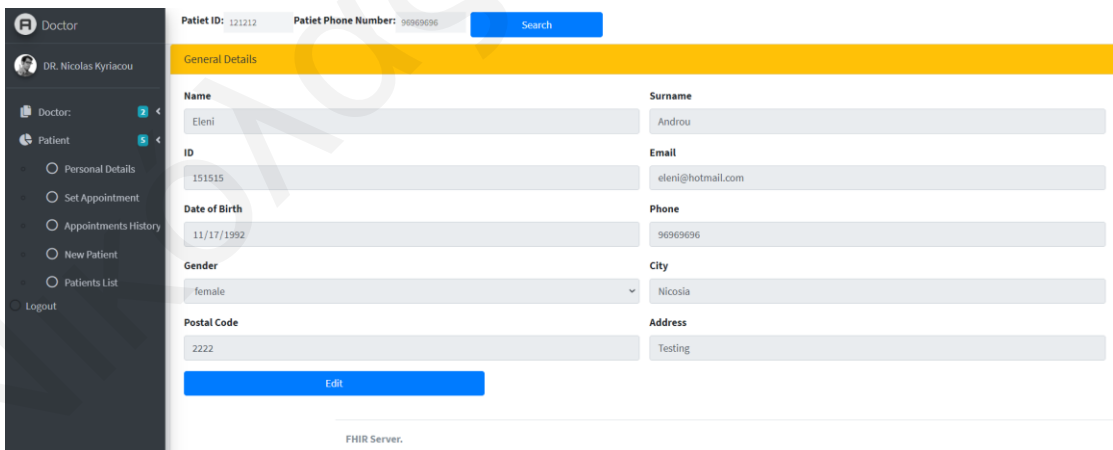
Επιπλέον, υπάρχει η δυνατότητα προβολής των ραντεβού ανά βδομάδα ή μέρα στο ημερολόγιο ή ακόμα και σε λίστα προβολής.



Εικ. 6-5 Doctor Calendar

### 6.3.2.3 Προσωπικά στοιχεία Ασθενή

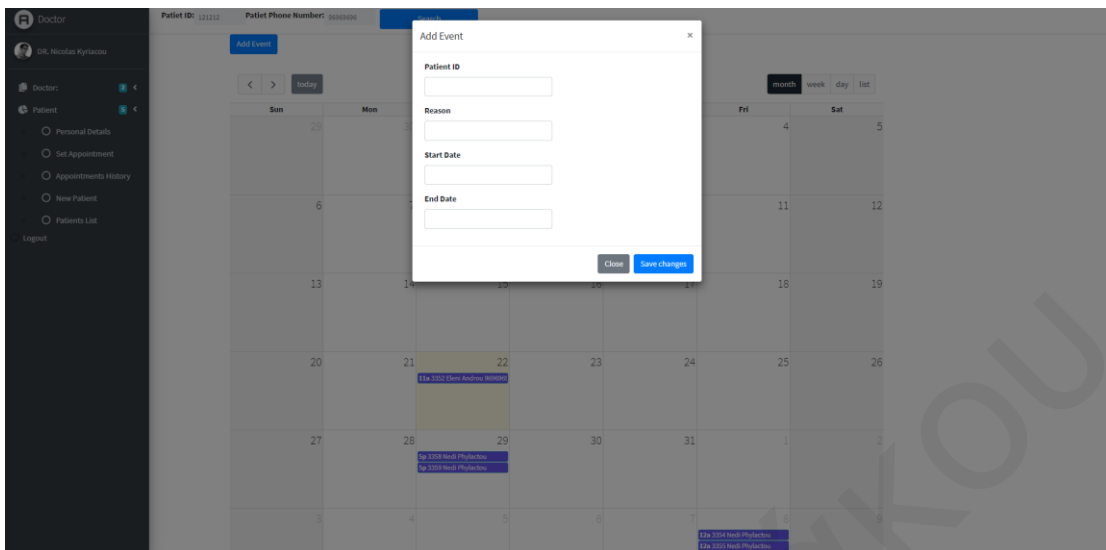
Μέσω της σελίδας αυτής ο γιατρός μπορεί να δει και να τροποποιήσει τα στοιχεία για τον κάθε ασθενή ξεχωριστά. Η σελίδα αυτή λειτουργεί παρόμοια όπως την Σελίδα με τα προσωπικά στοιχεία του γιατρού.



Εικ. 6-6 Patient Details

### 6.3.2.4 Δημιουργία Νέου Ραντεβού

Ο γιατρός μέσω της Σελίδας αυτής μπορεί να δει την διαθεσιμότητα των ραντεβού του για τους επόμενους μήνες. Ταυτόχρονα, μπορεί να ορίσει νέα ραντεβού, να αλλάξει δυναμικά κάποιο υφιστάμενο ή να ακυρώσει κάποιο ραντεβού.



Εικ.6-7 Patient Calendar

### 6.3.2.5 Ιστορικό Ραντεβού

Στη Σελίδα αυτή, ο χρήστης καλείται να αναζητήσει κάποιο ασθενή με βάση το τηλέφωνο ή την ταυτότητα του. Αφού το κάνει αυτό, τότε εμφανίζονται όλα τα ραντεβού του συγκεκριμένου ασθενή, οι ημερομηνίες τους, το status, ο λόγος τα αποτελέσματα καθώς και με πιο γιατρό ήταν ορισμένο το ραντεβού αυτό.

Appointment ID	Date/Time	Status	Reason	Result	Doctor
63	2020-11-28T07:00:00-03:00	proposed	test	new test	DR. Nicolas Kyriacou
112	2020-12-10T07:00:00-03:00	proposed	test	fever	DR. Panayiotis Savva
113	2020-12-10T07:00:00-03:00	pending	test		DR. Panayiotis Savva

Εικ. 6-8 Appointment History

### 6.3.2.6 Δημιουργία νέου ασθενή

Εάν τα στοιχεία του ασθενή δεν είναι καταχωρημένα στον FHIR Server τότε μέσω τις πιο κάτω σελίδας ο γιατρός μπορεί να καταχωρήσει τον ασθενή στο σύστημα.

Doctor: DR. Nicolas Kyriacou

Patient ID: 121212 Patient Phone Number: 96969696 Search

**General Details**

Name: Enter ... Surname: Enter ...

ID: Enter ... Email: Enter ...

Date of Birth: mm/dd/yyyy Phone: Enter ...

Gender: female City: Enter ...

Postal Code: Enter ... Address: Enter ...

Create

FHIR Server.

Εικ. 6-8 Patient Creation

Όλη η πιο πάνω αλληλεπίδραση, η αλλαγή και η καταχώρηση των δεδομένων μέσω της εφαρμογής αυτής έχουν ως αποτέλεσμα, την δημιουργία ενός κεντρικού συστήματος του FHIR Server, όπου σε αυτόν θα υπάρχουν καταχωρημένα τα στοιχεία για τους ασθενείς μας και η αλληλεπίδραση με τα στοιχεία αυτά επιτυγχάνεται πολύ εύκολα.

### 6.3.2.7 Λίστα Ασθενών

Με την επιλογή αυτή υπάρχει η δυνατότητα προβολής όλων των ασθενών, ανά σελίδα, και η δυναμική εύρεση των στοιχείων τους.

Doctor: DR. Nicolas Kyriacou

Patient ID: 121212 Patient Phone Number: 96969696 Search

**Patient List:**

Show 10 entries

Search:

ID	Name	Surname	Date of Birth	Phone	Email
101082	Nedi	Phylactou	1994-04-10		nediphy_94@hotmail.com
151515	Eleni	Androu	1992-11-17	96969696	eleni@hotmail.com
7000135	Homer	Simpson			
7000135	Homer	Simpson			

Showing 1 to 4 of 4 entries  
PreviousNext

FHIR SERVER

Εικ. 6-9 List of Patients

Τέλος, όπως βλέπουμε και στην πιο πάνω εικόνα, όπου απεικονίζεται το Web Interface του FHIR , υπάρχουν 9 γιατροί, 2 ασθενείς και 2 ραντεβού.

Options

Encoding (default) XML JSON

Pretty (default) On Off

Summary (none) true text data count

Server

Server Home/Actions

Resources

Practitioner 9

Appointment 2

Patient 2

Account

ActivityDefinition

AdverseEvent

Welcome to the Cyprus National Health Data Bank

Server	HAPI FHIR R4 Server
Software	HAPI FHIR Server - 5.1.0
FHIR Base	https://fhir.ehealth4u.eu/fhir

Server Actions

Retrieve the server's **conformance** statement.

Retrieve the update **history** across all resource types on the server.

Since

Εικ. 6-10 FHIR Interface

Επιπλέον, μέσω του interface αυτού οι χρήστες μπορούν να αλληλοεπιδράσουν και να αναζητήσουν, δημιουργήσουν, ανανεώσουν ή να διαγράψουν εγγραφές σε όλα τα resources που εμφανίζονται στο αριστερό μέρος της σελίδας μας. Αυτή η αλληλεπίδραση, γίνεται μέσω προκαθορισμένων πεδίων ανάλογα με το resource που θέλουμε να επηρεάσουμε.

eHealth4u

Server: eHealth4U FHIR Server R4

This is a RESTful server tester, which can be used to send requests to, and receive responses from the server at the following URL:

Resource: Prad

This page contains

Search

Search Parameters

Includes

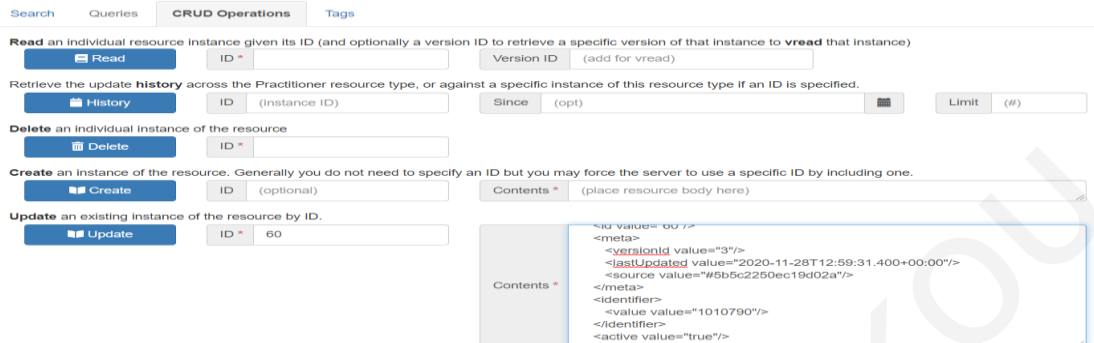
Sort Results

Sort By: Default Direction: Default

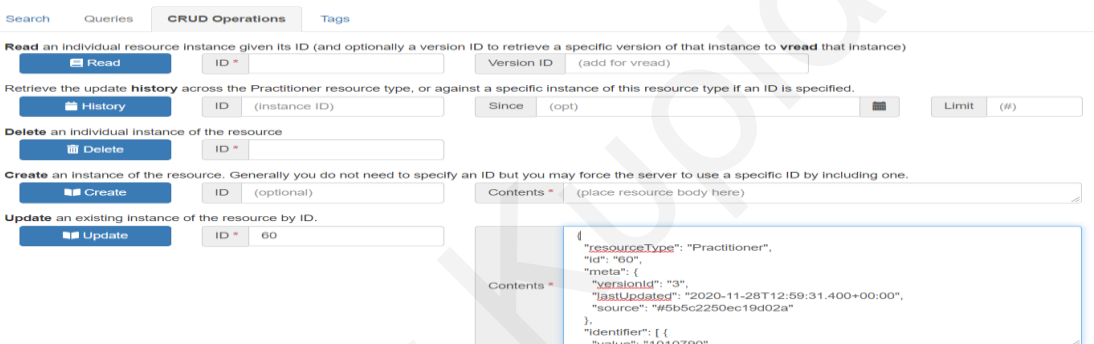
Other Options

Εικ. 6-11 FHIR Search Interface

Ένα ακόμα στοιχείο, που πρέπει να σημειώσουμε είναι ότι η αλληλεπίδραση με τον FHIR Server, μπορεί να γίνει είτε μέσω Json ή XML μορφή



Εικ. 6-12 FHIR XML Update



Εικ. 6-13 FHIR Json Update

## 6.4 Δομή Κώδικα

Την όλη υλοποίηση του συστήματος που περιγράφεται πιο πάνω μπορείτε να την βρείτε στο πιο κάτω repository:

- <https://github.com/nkyria03/doctor>

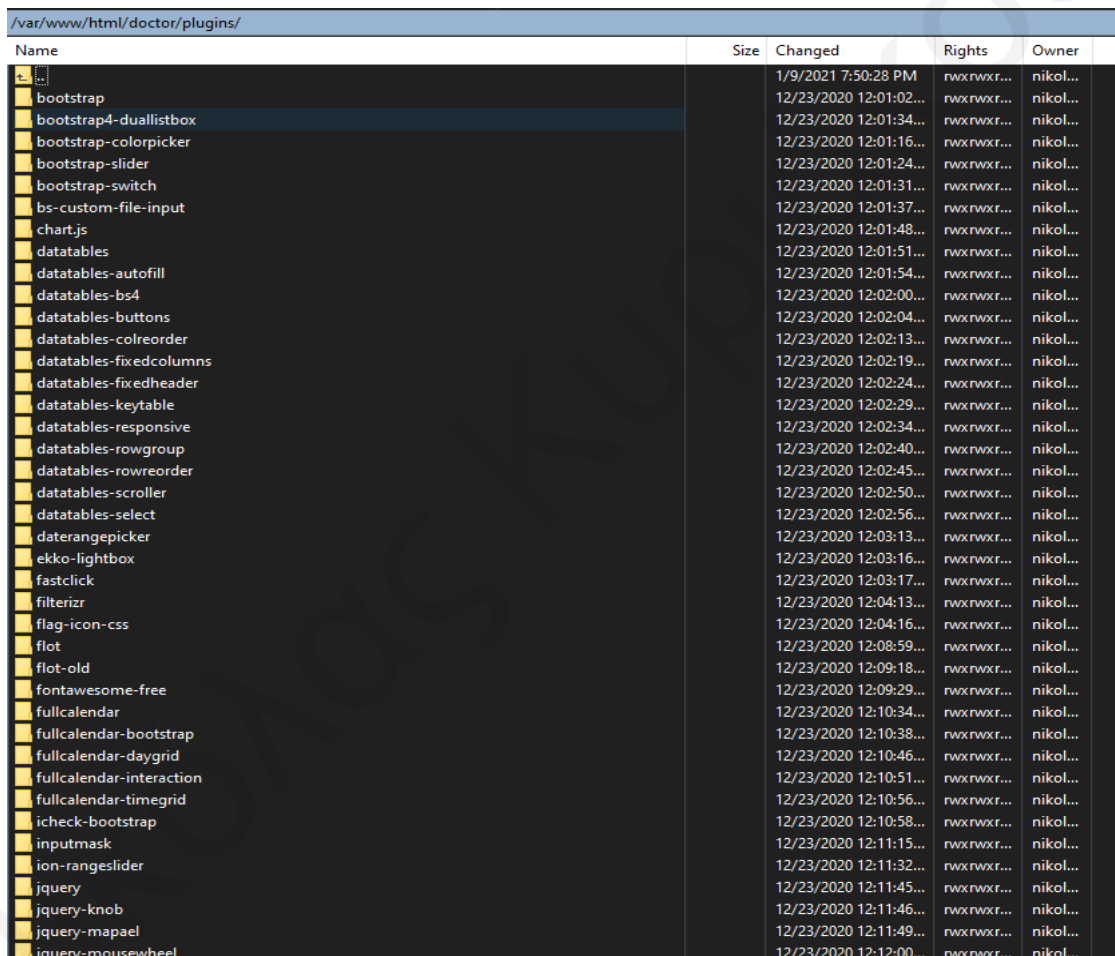
Name	Size	Changed	Rights	Owner
..		1/9/2021 7:50:28 PM	rw-rw-r--	root
api		12/23/2020 11:56:08...	rw-rw-r--	nikol...
BackEnd		12/23/2020 11:56:22...	rw-rw-r--	nikol...
js		12/23/2020 11:58:53...	rw-rw-r--	nikol...
packages		12/23/2020 12:00:50...	rw-rw-r--	nikol...
[plugins]		12/23/2020 12:21:01...	rw-rw-r--	nikol...
Calendar.php	13 KB	12/23/2020 12:56:19...	rw-rw-r--	nikol...
CreateNewPatient.php	13 KB	12/23/2020 11:54:33...	rw-rw-r--	nikol...
Doctor_MainPage.php	7 KB	12/23/2020 11:54:33...	rw-rw-r--	nikol...
Doctor_Calendar.php	13 KB	12/23/2020 11:54:33...	rw-rw-r--	nikol...
DoctorDetails.php	16 KB	12/23/2020 11:54:33...	rw-rw-r--	nikol...
ErrorPage.php	1 KB	11/28/2020 3:04:49 ...	rw-rw-r--	nikol...
index.php	2 KB	11/29/2020 1:54:44 ...	rw-rw-r--	nikol...
login_page.php	2 KB	12/23/2020 11:54:33...	rw-rw-r--	nikol...
PatientDetails.php	17 KB	12/23/2020 11:54:33...	rw-rw-r--	nikol...
PatientHistory.php	14 KB	12/23/2020 11:54:33...	rw-rw-r--	nikol...
PatientList.php	10 KB	12/23/2020 11:54:33...	rw-rw-r--	nikol...
style.css	1 KB	10/6/2020 8:19:57 PM	rw-rw-r--	nikol...

Εικ. 6-14 FHIR Code files

Όπως μπορούμε να διακρίνουμε από την πιο πάνω εικόνα ο κώδικας είναι ομαδοποιημένος σε φακέλους.

Στο αρχικό μονοπάτι του φακέλου μας υπάρχουν όλες οι σελίδες που είναι διαθέσιμες προς τον χρήστη. Ταυτόχρονα, ομαδοποίησα τα υπόλοιπα αρχεία ανάλογα με τις λειτουργίες τους.

- Το packages και plugins είναι οι διάφορες βιβλιοθήκες που χρησιμοποιώ για την υλοποίηση του συστήματος (bootstrap, jquery etc).



Name	Size	Changed	Rights	Owner
bootstrap		1/9/2021 7:50:28 PM	rw-rw-r...	nikol...
bootstrap4-duallistbox		12/23/2020 12:01:02...	rw-rw-r...	nikol...
bootstrap-colorpicker		12/23/2020 12:01:34...	rw-rw-r...	nikol...
bootstrap-slider		12/23/2020 12:01:16...	rw-rw-r...	nikol...
bootstrap-switch		12/23/2020 12:01:24...	rw-rw-r...	nikol...
bs-custom-file-input		12/23/2020 12:01:31...	rw-rw-r...	nikol...
chart.js		12/23/2020 12:01:37...	rw-rw-r...	nikol...
datatables		12/23/2020 12:01:48...	rw-rw-r...	nikol...
datatables-autofill		12/23/2020 12:01:51...	rw-rw-r...	nikol...
datatables-bs4		12/23/2020 12:01:54...	rw-rw-r...	nikol...
datatables-buttons		12/23/2020 12:02:00...	rw-rw-r...	nikol...
datatables-colreorder		12/23/2020 12:02:04...	rw-rw-r...	nikol...
datatables-colreorder		12/23/2020 12:02:13...	rw-rw-r...	nikol...
datatables-fixedcolumns		12/23/2020 12:02:19...	rw-rw-r...	nikol...
datatables-fixedheader		12/23/2020 12:02:24...	rw-rw-r...	nikol...
datatables-keytable		12/23/2020 12:02:29...	rw-rw-r...	nikol...
datatables-responsive		12/23/2020 12:02:34...	rw-rw-r...	nikol...
datatables-rowgroup		12/23/2020 12:02:40...	rw-rw-r...	nikol...
datatables-rowreorder		12/23/2020 12:02:45...	rw-rw-r...	nikol...
datatables-scroller		12/23/2020 12:02:50...	rw-rw-r...	nikol...
datatables-select		12/23/2020 12:02:56...	rw-rw-r...	nikol...
daterangepicker		12/23/2020 12:03:13...	rw-rw-r...	nikol...
ekko-lightbox		12/23/2020 12:03:16...	rw-rw-r...	nikol...
fastclick		12/23/2020 12:03:17...	rw-rw-r...	nikol...
filterizr		12/23/2020 12:04:13...	rw-rw-r...	nikol...
flag-icon-css		12/23/2020 12:04:16...	rw-rw-r...	nikol...
flot		12/23/2020 12:08:59...	rw-rw-r...	nikol...
flot-old		12/23/2020 12:09:18...	rw-rw-r...	nikol...
fontawesome-free		12/23/2020 12:09:29...	rw-rw-r...	nikol...
fullcalendar		12/23/2020 12:10:34...	rw-rw-r...	nikol...
fullcalendar-bootstrap		12/23/2020 12:10:38...	rw-rw-r...	nikol...
fullcalendar-daygrid		12/23/2020 12:10:46...	rw-rw-r...	nikol...
fullcalendar-interaction		12/23/2020 12:10:51...	rw-rw-r...	nikol...
fullcalendar-timegrid		12/23/2020 12:10:56...	rw-rw-r...	nikol...
icheck-bootstrap		12/23/2020 12:10:58...	rw-rw-r...	nikol...
inputmask		12/23/2020 12:11:15...	rw-rw-r...	nikol...
ion-rangeslider		12/23/2020 12:11:32...	rw-rw-r...	nikol...
jquery		12/23/2020 12:11:45...	rw-rw-r...	nikol...
jquery-knob		12/23/2020 12:11:46...	rw-rw-r...	nikol...
jquery-mapael		12/23/2020 12:11:49...	rw-rw-r...	nikol...
jquery-mousewheel		12/23/2020 12:12:00...	rw-rw-r...	nikol...

Εικ. 6-15 packages και plugins folders

- φάκελος js είναι τα διάφορα javascripts αρχεία που χρειαζόμαστε για την εφαρμογή του OAuth2 πρωτοκόλλου, όπως επίσης και τα javascripts για τις δυναμικές λειτουργίες του ημερολογίου.



Name	Size	Changed	Rights	Owner
..		1/9/2021 7:50:28 PM	rw-rwxr...	nikol...
daypilot		12/23/2020 11:58:51...	rw-rwxr...	nikol...
calendar.js	7 KB	12/23/2020 12:49:19...	rw-rw-r--	nikol...
keycloak.js	77 KB	11/7/2020 12:35:08 ...	rw-rw-r--	nikol...
keycloak_auth.js	1 KB	12/23/2020 11:54:33...	rw-rw-r--	nikol...

Εικ. 6-16 Js folder

- Ο φάκελος αρί είναι τα APIs που χρησιμοποιούμε για τις λειτουργίες του ημερολογίου

Name	Size	Changed	Rights	Owner
delete.php	1 KB	12/23/2020 11:54:32...	rw-rw-r--	nikol...
getevent.php	2 KB	12/23/2020 11:54:32...	rw-rw-r--	nikol...
insert.php	2 KB	12/23/2020 11:54:32...	rw-rw-r--	nikol...
load.php	3 KB	12/23/2020 11:54:32...	rw-rw-r--	nikol...
update.php	4 KB	12/23/2020 11:54:32...	rw-rw-r--	nikol...

Εικ. 6-17 API folder

- Τέλος, ο φάκελος Backend περιέχει όλα τα APIs και την αλληλεπίδραση που έχουμε με τον FHIR Server.

Name	Size	Changed	Rights	Owner
Calendar		12/23/2020 11:56:12...	rw-rwxr...	nikol...
Doctor		12/23/2020 11:56:13...	rw-rwxr...	nikol...
Keycloak		12/23/2020 11:56:16...	rw-rwxr...	nikol...
Login		12/23/2020 11:56:18...	rw-rwxr...	nikol...
Patient		12/23/2020 11:56:21...	rw-rwxr...	nikol...
Practitioner		12/23/2020 11:56:23...	rw-rwxr...	nikol...

Εικ. 6-18 Backend folder

Ένα παράδειγμα από την αλληλεπίδραση του συστήματος μας με τον FHIR Server είναι το πιο κάτω αρχείο για την ανάκτηση των δεδομένων του γιατρού με δεδομένο την ταυτότητα του:

```

1 <?php
2 //Get token
3 $token_url="http://104.154.208.78/doctor/BackEnd/Keycloak/GetToken.php";
4 $token=file_get_contents($token_url);
5 $doctor_id=$_GET['doctor_id'];
6 $url='https://fhir.ehealth4u.eu/fhir/Practitioner?identifier='.$doctor_id;
7 //add headers
8 $options = array(
9     'http'=>array(
10         'method'=>"GET",
11         'header'=>"Authorization: Bearer ".$token
12     )
13 );
14 $context=stream_context_create($options);
15 //response
16 echo $data=file_get_contents($url,false,$context);
17
18 ?>

```

Εικ. 6-19 Search practitioner by id api

Στη γραμμή 3-4 καλούμε ένα άλλο API το οποίο υλοποιήσαμε για να διαχειριζόμαστε τα tokens που απαιτεί το πρωτόκολλο OAuth2 για να επικοινωνήσουμε με τον FHIR.

Στη γραμμή 5 διαβάζουμε την ταυτότητα του γιατρού που στάλθηκε από την αρχική μας σελίδα.

Γραμμή 6-16 καλούμε το Get Request με το συγκεκριμένο header: Authorization: Bearer ".\$token και επιστρέφουμε το αποτέλεσμα.

Με παρόμοια λογική εκτελούμε και τα υπόλοιπα requests (GET,POST,PUT,DELETE) προς τον FHIR Server.

### Ροή δεδομένων:

Frontend -> BackendAPI->FHIR API-> BackendAPI-> Frontend

### Εφαρμογή OAuth2 προτύπου:

```

1 //Keycloak config
2 var keycloak = Keycloak({
3     url: 'https://auth.ehealth4u.eu/auth/',
4     realm: 'ehealth4u',
5     clientId: 'ucy',
6     credentials: {
7         secret: '50b0b62d-6305-40ab-81a3-41e092738ec4'
8     }
9 });
10
11 var loadData = function ()
12 {
13     var form_data = {
14         email: keycloak.tokenParsed.email
15     };
16
17     $.ajax({
18         type: "POST",
19         url: "login_page.php",
20         data: form_data,
21         async:false,
22         success: function(response)
23         {
24             if(response == 'OK'){
25                 console.log("sessions ajax established");
26                 window.location.href = 'http://104.154.208.78/doctor/Doctor_MainPage.php';
27             }
28             else
29             {
30                 window.location.href = 'http://104.154.208.78/doctor/ErrorMessage.php';

```

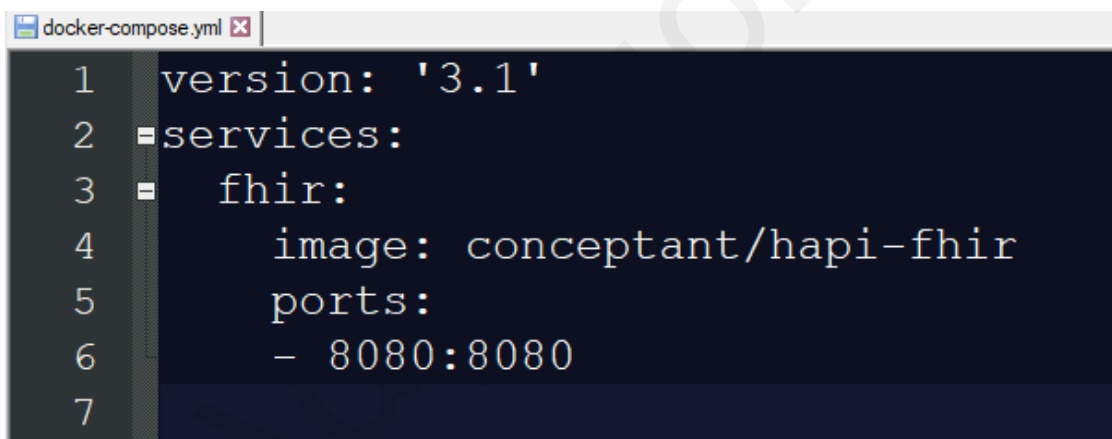
Εικ. 6-20 Keycloak configuration

Για να μπορέσουμε να πετύχουμε την αλληλεπίδραση της εφαρμογής μας με το OAuth2 πρότυπο χρησιμοποιήσαμε την JavaScript βιβλιοθήκη του Keycloak και όπως φαίνεται και στην πιο πάνω εικόνα το μόνο που χρειάστηκε να δώσουμε ήταν τα τελικό url για authentication, realm name, το client id και το secret password τα οποία δημιουργεί και διανέμει ο System admin του keycloak.

### Test FHIR:

Όπως προανάφερα για την ανάπτυξη της εφαρμογής χρησιμοποιήσα αρχικά ένα test FHIR πρωτόκολλο. Για αυτό είχα χρησιμοποιήσει, ένα docker image για να δημιουργήσω το docker container για να μπορέσω να αλληλοεπιδράσω μαζί του.

Το docker-compose file φαίνεται στην πιο κάτω εικόνα:



```
1 version: '3.1'
2 services:
3   fhir:
4     image: conceptant/hapi-fhir
5     ports:
6     - 8080:8080
7
```

Εικ. 6-21 docker-compose file

Το file αυτό με την εντολή docker compose up, δημιουργεί το interface του FHIR πρωτοκόλλου σύμφωνα με το image conceptant/hapi-fhir και το κάνει expose στο port 8080 του Server μας.

# Κεφάλαιο 7

## Συμπεράσματα

### 7.1 Γενικά Συμπεράσματα

Στις μέρες μας όπου ο τομέας της πληροφορικής έχει πάρει ραγδαίους ρυθμούς ανάπτυξης, ένας από τους κύριους της στόχους είναι η βελτίωση και διευκόλυνση των συνθηκών ζωής του κάθε ανθρώπου ξεχωριστά. Ένας από τους τομείς της ζωής μας που χρειάζεται στήριξη, σε αυτή ειδικότερα την εποχή, την εποχή της πανδημίας λόγω Covid-19, είναι ο τομέας της υγείας. Τα εμπλεκόμενα μέλη στον τομέα αυτό επιβάλλεται να αλληλεπιδρούν με ασφάλεια, άμεσα, με απλό και κατανοητό κυρίως τρόπο ούτως ώστε οι ειδικοί να έχουν όσο το δυνατό περισσότερα στοιχεία στην κατοχή τους προς όφελος του κάθε ασθενή ξεχωριστά. Όλα τα πιο πάνω είναι πλέον εφικτά μέσω του FHIR πρωτοκόλλου εάν αυτό καταφέρει σιγά σιγά να ενταχθεί και να εφαρμοστεί στα υφιστάμενα αλλά και τα νέα συστήματα υγείας που θα αναπτυχθούν τα επόμενα χρόνια.

Με την διατριβή αυτή δείξαμε ότι με απλούς τρόπους και τη χρήση τεχνολογιών διαδικτύου ο κάθε 3<sup>rd</sup> Party Service Provider μπορεί να έχει άμεση πρόσβαση σε δεδομένα ασθενών και να αλληλοεπιδρά ανάλογα με τα permissions που του έδωσε ο εκάστοτε διαχειριστής του πρωτοκόλλου αυτού.

### 7.2 Αποδοχή από τον κόσμο

Όπως φαίνεται και από τα γεγονότα το FHIR πρωτόκολλο σιγά σιγά άρχισε να υιοθετείται σε διεθνές επίπεδο, λόγω των δυνατοτήτων που προσφέρει αλλά και τις προοπτικές που φανερώνει στον τομέα της Ηλεκτρονικής Υγείας.

Πιο συγκεκριμένα μεγάλες εταιρείες στον χώρο της τεχνολογίας και της πληροφορικής άρχισαν να αναπτύσσουν το πρότυπο αυτό και να το παρέχουν στο κοινό.

Αξιωματούχοι από την Alphabet (μητρική εταιρεία της Google), Amazon, IBM, Microsoft, Oracle και Salesforce στο δεύτερο CMS Blue Button 2.0 Developer Conference στην Ουάσιγκτον επανέλαβαν την υπόσχεσή τους να συνεχίσουν να χρησιμοποιούν και να καινοτομούν το πρωτόκολλο FHIR της HL7.<sup>29</sup>[29] Επίσης, ένα ακόμη στοιχείο που μας οδηγεί στην νέα τάση της Ηλεκτρονικής Υγείας και πως αυτή θα εφαρμοστή είναι και τα Google Trends για το FHIR πρότυπο, που ολοένα και περισσότερες χώρες όπως φαίνεται και πιο κάτω άρχισε να τους απασχολεί.<sup>30</sup>[30]

---



*Εικ. 7-1 Google Trends on the “FHIR” key word*

### **7.3 Μελλοντικές Προεκτάσεις**

Μέσα από τη διαδικασία της αξιολόγησης της εφαρμογής, προέκυψαν κάποιες πιθανές ιδέες οι οποίες θα μπορούσαν να προχωρήσουν, με σκοπό την βελτιστοποίηση της εφαρμογής αυτής.

#### **7.3.1 Μεγαλύτερη αξιοποίηση των δυνατοτήτων που προσφέρει το FHIR πρωτόκολλο.**

Αυτό θα ήταν εφικτό να γίνει με την περεταίρω ανάπτυξη της εφαρμογής μας, και την υλοποίηση ενός πλήρους Customer Relationship Management (CRM). Όπως για παράδειγμα την δυνατότητα για πληρωμή μετά την ολοκλήρωση της επίσκεψης από

τους ασθενείς ή και την εισαγωγή της ιατροφαρμακευτικής περιθαλψης για τον κάθε ασθενή.

### **7.3.2 Επέκταση του συστήματος**

Με αυτό τον τρόπο το 3<sup>rd</sup> party application αυτό μπορεί να είναι διαθέσιμο και προς τους ασθενείς και έτσι θα μπορούν και οι ίδιοι να έχουν άμεση πρόσβαση στο ιστορικό τους η ακόμη και στην εισαγωγή κάποιων ιατρικών δεδομένων. Αυτό μπορεί να πραγματοποιηθεί με την ανάπτυξη μίας κινητής εφαρμογής η οποία να είναι διαθέσιμη προς το κοινό και μέσω αυτής ο κάθε ασθενής να μπορεί να ενημερώσει το FHIR Server, συνεπώς και τον ιατρό του για οποιαδήποτε συμπτώματα μπορεί να αντιμετωπίζουν.

### **7.3.3 Δυναμική χρήση του FHIR πρωτόκολλου και των περιορισμών του**

Η εφαρμογή αυτή δημιουργήθηκε με βάση το επίσημο FHIR πρωτόκολλο και ως εκ τούτου δεν θα μπορούσε να λάβει υπόψη της της οποιοσδήποτε διαφοροποιήσεις θελήσει ο system admin του πρωτοκόλλου αυτού να εφαρμόσει, όπως για παράδειγμα την εισαγωγή του μοναδικού αριθμού του ΓΕΣΥ για τα Κυπριακά δεδομένα.

### **7.3.4 Εφαρμογή του FHIR πρωτοκόλλου από τους παρόχους Υγείας**

Τα τρία πιο πάνω σημεία αφορούν μόνο τη συγκεκριμένη εφαρμογή ως 3<sup>rd</sup> party Service Provider. Οι μελλοντικές όμως προεκτάσεις που θα βοηθούσαν το κοινό καλό είναι η πιο ευρείας και με σωστό προγραμματισμένο τρόπο εφαρμογή του πρωτοκόλλου αυτού στους υφιστάμενους παρόχους υγείας, για να μπορέσουμε να αναπτύξουμε και να εκμεταλλευτούμε τα πλεονεκτήματα που προσφέρει ένα τέτοιο πρωτόκολλο. Αυτό, θα μπορεί να αναπτυχθεί σταδιακά με την εφαρμογή μιας νομοθεσίας η οποία θα αναγκάζει τους παρόχους Υγείας να προχωρήσουν σιγά σιγά στην αναβάθμιση των συστημάτων τους.

# Βιβλιογραφία

---

## Internet Sites:

- [1] HealthITAnalytics. 2021. *4 Basics To Know About The Role Of FHIR In Interoperability*. [online] Available at: <<https://healthitanalytics.com/news/4-basics-to-know-about-the-role-of-fhir-in-interoperability>> [Accessed 4 December 2020].
- [2] HealthITAnalytics. 2021. *4 Basics To Know About The Role Of FHIR In Interoperability*. [online] Available at: <<https://healthitanalytics.com/news/4-basics-to-know-about-the-role-of-fhir-in-interoperability>> [Accessed 4 December 2020].
- [3] HL7.org. 2021. *Overview - FHIR V4.0.1*. [online] Available at: <<https://www.hl7.org/fhir/overview.html>> [Accessed 4 December 2020].
- [4] Moh.gov.gr. 2021. *Ehealth - Ηλεκτρονική Υγεία - Υπουργείο Υγείας*. [online] Available at: <<https://www.moh.gov.gr/articles/ehealth/>> [Accessed 4 December 2020].
- [5] Repository.kallipos.gr. 2021. [online] Available at: <[https://repository.kallipos.gr/bitstream/11419/288/1/02\\_chapter\\_03.pdf](https://repository.kallipos.gr/bitstream/11419/288/1/02_chapter_03.pdf)> [Accessed 4 December 2020].
- [6] HL7.org. 2021. *Overview-Dev - FHIR V4.0.1*. [online] Available at: <<https://www.hl7.org/fhir/overview-dev.html>> [Accessed 4 December 2020].
- [7] Ehealth4u.cs.ucy.ac.cy. 2021. *Integrated National Ehealth Ecosystem – University Of Cyprus*. [online] Available at: <<http://ehealth4u.cs.ucy.ac.cy/>> [Accessed 4 December 2020].
- [8] Www2.cs.ucy.ac.cy. 2021. [online] Available at: <<http://www2.cs.ucy.ac.cy/~nicolast/courses/cs435/lectures/hci19.pdf>> [Accessed 4 December 2020].
- [9] 2021. [online] Available at: <<https://www.apachefriends.org/index.html>> [Accessed 4 December 2020].
- [10] Code, V., 2021. *Visual Studio Code - Code Editing. Redefined*. [online] Code.visualstudio.com. Available at: <<https://code.visualstudio.com/>> [Accessed 4 December 2020].
- [11] Soapui.org. 2021. *The World's Most Popular API Testing Tool | Soapui*. [online] Available at: <<https://www.soapui.org/>> [Accessed 4 December 2020].
- [12] Php.net. 2021. *PHP: Hypertext Preprocessor*. [online] Available at: <<https://www.php.net/>> [Accessed 4 December 2020].
- [13] Developer.mozilla.org. 2021. *Javascript | MDN*. [online] Available at: <<https://developer.mozilla.org/en-US/docs/Web/JavaScript>> [Accessed 4 December 2020].

- 
- [14] El.wikipedia.org. 2021. *XML*. [online] Available at: <<https://el.wikipedia.org/wiki/XML>> [Accessed 4 December 2020].
- [15] El.wikipedia.org. 2021. *JSON*. [online] Available at: <<https://el.wikipedia.org/wiki/JSON>> [Accessed 4 December 2020].
- [16] En.wikipedia.org. 2021. *YAML*. [online] Available at: <<https://en.wikipedia.org/wiki/YAML>> [Accessed 4 December 2020].
- [17] Docker Documentation. 2021. *Docker Images*. [online] Available at: <<https://docs.docker.com/engine/reference/commandline/images/>> [Accessed 4 December 2020].
- [18] Docker. 2021. *What Is A Container? | Docker*. [online] Available at: <<https://www.docker.com/resources/what-container>> [Accessed 4 December 2020].
- [19] Docker Documentation. 2021. *Compose File*. [online] Available at: <<https://docs.docker.com/compose/compose-file/>> [Accessed 4 December 2020].
- [20] Team, K., 2021. *Keycloak - Documentation*. [online] Keycloak.org. Available at: <<https://www.keycloak.org/documentation.html>> [Accessed 4 December 2020].
- [21] HI7.org. 2021. *Practitioner - FHIR V4.0.1*. [online] Available at: <<https://www.hl7.org/fhir/practitioner.html>> [Accessed 4 December 2020].
- [22] HI7.org. 2021. *Patient - FHIR V4.0.1*. [online] Available at: <<https://www.hl7.org/fhir/patient.html>> [Accessed 4 December 2020].
- [23] HI7.org. 2021. *Appointment - FHIR V4.0.1*. [online] Available at: <<https://www.hl7.org/fhir/appointment.html>> [Accessed 4 December 2020].
- [24] Ehealth4u.cs.ucy.ac.cy. 2021. *Project – Integrated National Ehealth Ecosystem*. [online] Available at: <<http://ehealth4u.cs.ucy.ac.cy/index.php/project/>> [Accessed 8 January 2021].
- [25] Zhixin Wen, a., 2021. *Bootstrap Table · An Extended Table To The Integration With Some Of The Most Widely Used CSS Frameworks. (Supports Bootstrap, Semantic UI, Bulma, Material Design, Foundation)*. [online] Bootstrap-table.com. Available at: <<https://bootstrap-table.com/>> [Accessed 15 January 2021].
- [26] Shaw, A., 2021. *Fullcalendar - Javascript Event Calendar*. [online] Fullcalendar.io. Available at: <<https://fullcalendar.io/>> [Accessed 4 December 2020].
- [27] 2021. [online] Available at: <<https://www.linuxfork.com/how-to-install-php-7-0-33-on-ubuntu-18-04-20-04-lts/>> [Accessed 4 December 2020].
- [28] Docker Documentation. 2021. *Install Docker Compose*. [online] Available at: <<https://docs.docker.com/compose/install/>> [Accessed 4 December 2020].
- [29] Healthcare IT News. 2021. *Amazon, Google, IBM, Microsoft, Oracle, Salesforce Re-Up On Interoperability Promise*. [online] Available at: <<https://www.healthcareitnews.com/news/amazon-google-ibm-microsoft-oracle-salesforce-reup-interoperability-promise>> [Accessed 8 January 2021].



---

[30] 2021. [online] Available at: <<https://trends.google.com/trends/explore?q=FHIR>>  
[Accessed 15 January 2021].

Νικόλας Κυριάκου