



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
& ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΦΑΚΕΛΟΣ ΥΓΕΙΑΣ ΣΥΜΦΩΝΟΣ ΜΕ ΤΟ ΠΡΟΤΥΠΟ FHIR

Υπότιτλος

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του/της

ΒΟΥΔΟΥΡΗ ΠΑΣΧΑΛΙΝΑΣ

Επιβλέπων: ΑΓΓΕΛΙΔΗΣ ΠΑΝΤΕΛΗΣ

ΚΑΘΗΓΗΤΗΣ

ΘΕΣΣΑΛΟΝΙΚΗ/ΙΟΥΛΙΟΣ/2022



HELLENIC DEMOCRACY
UNIVERSITY OF WESTERN MACEDONIA
SCHOOL OF ENGINEERING
DEPARTMENT OF ELECTRICAL
& COMPUTER ENGINEERING

MEDICAL RECORD IN ACCORDANCE WITH HL7 FHIR STANDARD

Subtitle

THESIS

VOUDOURI PASCHALINA'S

SUPERVISOR: AGGELIDIS PANTELIS

PROFESSOR

THESSALONIKI/JULY/2022



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
& ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ ΚΑΙ ΑΝΑΛΗΨΗΣ ΠΡΟΣΩΠΙΚΗΣ ΕΥΘΥΝΗΣ

Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν. 1599/1986 και τα άρθρα 2,4,6 παρ. 3 του Ν.

1256/1982, η παρούσα Διπλωματική Εργασία με τίτλο “Φάκελος Υγείας Σύμφωνος με το Πρότυπο FHIR” καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας και αναφέρονται ρητώς μέσα στο κείμενο που συνοδεύουν, και η οποία έχει εκπονηθεί στο Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Δυτικής Μακεδονίας, υπό την επίβλεψη του μέλους του Τμήματος κ. Αγγελίδη Παντελή αποτελεί αποκλειστικά προϊόν προσωπικής εργασίας και δεν προσβάλλει κάθε μορφής πνευματικά δικαιώματα τρίτων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή / και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. 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 μόνο.

Copyright (C) Πασχαλίνα Βουδούρη, Παντελής Αγγελίδης , 2022,

Θεσσαλονίκη Copyright (C)

Υπογραφή Φοιτητή: _____

Περίληψη

Αντικείμενο αυτής της διπλωματικής είναι η δημιουργία μιας διαδικτυακής πλατφόρμας, η οποία να βελτιώνει την επαφή μεταξύ κάποιου ιατρού με έναν ασθενή, παρέχοντας προδιαγραφές για καλύτερη ανταλλαγή πληροφοριών υγείας. Συγκεκριμένα, αξιοποιείται η διαδικτυακή υπηρεσία REST και οι ρυθμίσεις του προτύπου HL7 FHIR ώστε να είναι εφικτή η σύνδεση των γιατρών ο ένας με τον άλλον, και να παρέχει τον ολοκληρωμένο ιατρικό φάκελο των πελατών. Έτσι οι γιατροί προσφέρουν στον κάθε ασθενή τις υπηρεσίες αυτές ξεχωριστά χωρίς να υπάρχουν πολλά ιατρικά λάθη.

Η εφαρμογή που σχεδιάστηκε έχει ως θεμελιώδη στόχο στην καλύτερη προσφορά υπηρεσιών υγείας στην καθημερινότητα ενός ιατρού, π.χ. τη διαχείριση της επίσκεψης ασθενή, την συνταγογράφηση φαρμακευτικής αγωγής, κ.α.. Παράλληλα, δίνει την δυνατότητα σε ανταλλαγή του ιατρικού φακέλου ενός ασθενή μεταξύ γιατρών καθώς και τη δημιουργία διαδικτυακών διαγνωστικών εκθέσεων και συνταγών φαρμάκων.

Η ανάπτυξη της εφαρμογής έγινε κατόπιν ανάλυσης απαιτήσεων και λεπτομερούς σχεδιασμού, λαμβάνοντας υπόψη διάφορα σενάρια χρήσης στο χώρο της παροχής ιατρικών υπηρεσιών. Εξετάστηκαν και υιοθετήθηκαν σύγχρονες τεχνολογίες ανάπτυξης, όπως PHP, JavaScript, Ajax, JSON, Node.js και PostgreSQL.

Στο κεφάλαιο πρώτο, παρουσιάζονται οι κύριες έννοιες στις οποίες άπτεται η διπλωματική. Στο επόμενο κεφάλαιο, παρατίθενται οι τεχνολογίες με τις οποίες αναπτύχθηκε η εφαρμογή. Στο τρίτο κεφάλαιο, αναλύονται τα κύρια ζητούμενα και οι απαιτήσεις της εφαρμογής, και περιγράφονται τα βασικότερα στοιχεία της. Στη συνέχεια, γίνεται λεπτομερής ανάλυση της σχεδίασης της εφαρμογής, περιγράφοντας παράλληλα τις βασικές οντότητες που χρησιμοποιούνται. Στο πέμπτο κεφάλαιο, δίνονται οι απαραίτητες πληροφορίες σχετικά με την υλοποίηση της παραγόμενης εφαρμογής, ενώ ταυτόχρονα περιγράφονται διάφορα σενάρια χρήσης της σε πρακτικό επίπεδο. Στο τελευταίο κεφάλαιο, κατατίθενται τα εξαγόμενα συμπεράσματα που ελήφθησαν στο πέρας της εργασίας, και συγχρόνως γίνονται και σχετικές προτάσεις για ενδεχόμενες επεκτάσεις σε επόμενες φάσεις εξέλιξης της εφαρμογής.

Λέξεις Κλειδιά

Φάκελος υγείας, HL7 FHIR, υπηρεσίες REST, διαδικτυακή εφαρμογή, ανταλλαγή πληροφοριών

Abstract

The object of this project is to create an online application, which improves the communication between a doctor and a patient, providing an important tool for the effective exchange of health information. In particular, the REST online service and the structures of the HL7FHIR standard are utilized in order to enable the connectivity between doctors, and to provide the complete medical file of patients. This way doctors are able to provide personalized services and reduce medical errors.

The main goal of the designed application is to improve the quality of provision of health services in a doctor's daily life, e.g. to handle a visit of a patient or prescribe medication. At the same time, it enables the exchange of patients medical files by doctors and the creation of online diagnostic reports and medicine prescription.

The development of the application required demand analysis and detailed planning, taking several use cases in the field of provision of health services into account. Modern development technologies were examined and adopted, like PHP, JavaScript, AJAX, JSON, Node.js and PostgreSQL.

First of all, there is a presentation to the fundamentals on which the thesis lies. In the next chapter, the technologies with which the application was developed are laid. Also, there is a description of the main requirements of the application and its basic elements are described. Afterwards, there is an analysis of the develop of the application, while describing the basic entities used. In the fifth chapter, the necessary information regarding the development of the created application are given, while simultaneously describing several use cases in a practical level. In the final chapter, the extracted realizations made throughout the course of the thesis and relevant suggestions for hypothetical extensions in future evolution stages of the application are shown.

Keywords

Medical file, HL7 FHIR, REST Web Services, Web Application, Information Exchange

Ευχαριστίες

ΤΟΠΟΣ/ΜΗΝΑΣ/ΕΤΟΣ ΕΚΔΟΣΗ

Περιεχόμενα

Περίληψη	7
Abstract	8
Ευχαριστίες	9
Περιεχόμενα	12
Κατάλογος Σχημάτων	15
Κατάλογος Εικόνων	16
Κατάλογος Πινάκων	17
Πρόλογος	18
Κεφάλαιο 1: Εισαγωγή	19
Βασικές έννοιες	19
Αντικείμενο της Διπλωματικής	20
Οργάνωση της Διπλωματικής	20
Κεφάλαιο 2: Τεχνολογίες	21
HL7 FHIR	21
PHP	21
JavaScript	21
Ajax	22
JSON	22
Node.js	23
PostgreSQL	23
Κεφάλαιο 3: Ανάλυση του Θέματος	24
Απαιτήσεις εφαρμογής	24
Αρχιτεκτονική της πλατφόρμας	26
Κεφάλαιο 4: Σχεδίαση Θέματος	28
Σχεδίαση Βάσης Δεδομένων	28
Οντότητες-Πόροι του FHIR	28
Patient	29

Encounter	29
Observation	29
Diagnostic Report	30
Medication Request	30
Σχεδίαση Εφαρμογής Πελάτη (Ιατρού)	30
Σχεδίαση Διεπαφής REST API Εφαρμογής	32
Υποστηριζόμενες Λειτουργίες Ασθενών (Patient)	32
Υποστηριζόμενες Λειτουργίες Επισκέψεων (Encounter)	32
Υποστηριζόμενες Λειτουργίες Φαρμακευτικής Αγωγής (MedicationRequest)	33
Υποστηριζόμενες Λειτουργίες Διαγνωστικών Εξετάσεων (DiagnosticReports)	34
Κεφάλαιο 5: Υλοποίηση	35
Ανάπτυξη Βάσης Δεδομένων	35
PostgreSQL	35
FHIR Base	35
Υλοποίηση Σελίδων Εφαρμογής	37
Υλοποίηση REST API Διεπαφής	46
Υποστηριζόμενες Λειτουργίες Ασθενών (Patient)	46
Υποστηριζόμενες Λειτουργίες Επισκέψεων (Encounter)	47
Υποστηριζόμενες Λειτουργίες Φαρμακευτικής Αγωγής (MedicationRequest)	48
Υποστηριζόμενες Λειτουργίες Διαγνωστικών Εξετάσεων (DiagnosticReports)	49
Κεφάλαιο 6: Συμπεράσματα	50
Αποτελέσματα	50
Συμπεράσματα	51
Μελλοντικές Επεκτάσεις	51
Παράρτημα Α – Δεδομένα FHIR	52
Ενδεικτικά Παραδείγματα Περιεχομένου Πόρων FHIR	52
Παράρτημα Β – Εγκατάσταση Εργαλείων	65
Εγκατάσταση του Wapp	65
Εγκατάσταση της FHIRBase	65
Node.js	66

Παράρτημα Γ – Κώδικας Εφαρμογής	67
Κώδικας Ιστοσελίδων PHP	67
Κώδικας REST API Διεπαφής	99
Βιβλιογραφία	111
Συνομογραφίες - Αρκτικόλεξα - Ακρωνύμια	112
Απόδοση ξενόγλωσσων όρων	113

Κατάλογος Σχημάτων

Κατάλογος Εικόνων

Εικόνα 1: Μοντέλο Διάδρασης Χρηστών	25
Εικόνα 2: Αρχιτεκτονική Πλατφόρμας	26
Εικόνα 3: Μοντέλο Οντοτήτων της Εφαρμογής.....	28
Εικόνα 4: Προβολή Οντότητας Patient με χρήση του εργαλείου pgAdmin4	36
Εικόνα 5: Φόρμα Αναζήτησης Ασθενών με κριτήριο μέρος του Ονόματος	37
Εικόνα 6: Λίστα Ασθενών με κριτήριο μέρος του Ονόματος.....	38
Εικόνα 7- Προβολή Φακέλου Επιλεγμένου Ασθενή - Εμφάνιση Καρτέλας Προσωπικών Στοιχείων.....	38
Εικόνα 9: Φόρμα αποστολής φακέλου	39
Εικόνα 10: Ειδοποίηση χρήστη για την αποστολή του φακέλου	39
Εικόνα 11: Λήψη κρυπτογραφημένου μηνύματος στο σύστημα mailtrap	40
Εικόνα 12: Προβολή Καρτέλας Φαρμακευτικών Αγωγών Επιλεγμένου Ασθενή	41
Εικόνα 13: Προβολή Καρτέλας Αποτελεσμάτων Διαγνωστικών Εξετάσεων Ασθενή.....	41
Εικόνα 14: Προβολή Καρτέλας Ιατρικών Επισκέψεων Επιλεγμένου Ασθενή	42
Εικόνα 15: Φόρμα Εισαγωγής νέας Επίσκεψης σε Ασθενή.....	43
Εικόνα 16: Σελίδα Διαχείρισης νέας Επίσκεψης σε Ασθενή.....	43
Εικόνα 17: Φόρμα Προσθήκης νέας Φαρμακευτικής Αγωγής	44
Εικόνα 18: Σελίδα Διαχείρισης Φαρμακευτικής Αγωγής.....	44
Εικόνα 19: Φόρμα Ενημέρωσης Φαρμακευτικής Αγωγής.....	45
Εικόνα 20: Διαγραφή Φαρμακευτικής Αγωγής	45
Εικόνα 21: Χρήση του εργαλείου pgAdmin4	65

Κατάλογος Πινάκων

Πίνακας 1: Πλήθος Οντοτήτων στη βάση FHIRBase	36
---	----

Πρόλογος

Η εργασία είχε σκοπό την επαφή με το πρότυπο ανταλλαγής πληροφοριών HL7 FHIR και την εφαρμογή του πρότυπου αυτού σε μία διαδικτυακή εφαρμογή. Με βάση το παραπάνω σημαντικά στοιχεία της διπλωματικής εργασίας αποτελούν οι τεχνολογίες, η ανάλυση της εφαρμογής, η σχεδίαση της, η υλοποίηση και τέλος τα παραγόμενα αποτελέσματα, καθώς και τα επακόλουθα συμπεράσματα που προέκυψαν. Για αυτόν τον λόγο στα επόμενα κεφάλαια θα αναλυθούν τα παραπάνω βασικά στοιχεία της εφαρμογής.

Κεφάλαιο 1: Εισαγωγή

Στο κεφάλαιο αυτό θα παρουσιαστούν ορισμένα εισαγωγικά στοιχεία της διπλωματικής εργασίας, όπως το κίνητρο για την συγγραφή της καθώς και οι βασικές έννοιες που χρησιμοποιούνται στην εργασία, όπως τα ιατρικά πληροφοριακά συστήματα. Επιπλέον, θα γίνει περιγραφή του αντικειμένου της εργασίας, καθώς και της οργάνωσης της συγκεκριμένης διπλωματικής εργασίας.

Βασικές έννοιες

Στον κλάδο της υγείας μεταφέρεται σε καθημερινή βάση πολλαπλός όγκος δεδομένων και πληροφοριών ανάμεσα σε χρήστες του κλάδου, όπως γιατρούς, νοσηλευτές, μέλη διοικητικού προσωπικού, κ.α. Παρατηρείται επίσης το φαινόμενο να χρησιμοποιούνται στο χώρο της υγείας μια πληθώρα εφαρμογών που έχουν αναπτυχθεί χωρίς να ληφθεί υπόψη η δυνατότητα ανταλλαγής δεδομένων.

Πολλές φορές, οι εμπλεκόμενες πλευρές καταλήγουν να ανταλλάσσουν δεδομένα, χωρίς να χρησιμοποιείται κάποια προτυποποίηση, που καθιστά δύσκολη, και μερικές φορές αδύνατη, τη διαλειτουργικότητα των υφιστάμενων ιατρικών εφαρμογών. Με βάση τα παραπάνω προκύπτει η ανάγκη υιοθέτησης διεθνών προτύπων ανταλλαγής ιατρικών πληροφοριών από τις εφαρμογές που χρησιμοποιούνται στον τομέα της Υγείας.

Στην παρούσα εργασία εξετάστηκε η ανάπτυξη μιας διαδικτυακής εφαρμογής που θα διαχειρίζεται πληροφορίες Ιατρικών Φακέλων Ασθενών συμβατές με το πρότυπο ανταλλαγής ιατρικών δεδομένων HL7 FHIR[1].

Η εφαρμογή που αναπτύχθηκε εντάσσεται στη γενικότερη κατηγορία των Ιατρικών Πληροφοριακών Συστημάτων. Ένα Πληροφοριακό Σύστημα είναι ένα λογισμικό που αποτελείται από συνιστώσες που εργάζονται από κοινού για την ολική επεξεργασία της παραγόμενης πληροφορίας για έναν οργανισμό.

Με βάση τον παραπάνω ορισμό, ως Ιατρικό Πληροφοριακό Σύστημα ορίζεται το πληροφοριακό σύστημα, που έχει ως θεμελιώδη σκοπό την συνολική διαχείριση των πληροφοριών που ανταλλάσσονται σε μία δομή Υγείας, καθώς και την εποπτεία της λειτουργικότητας των εφαρμογών που βρίσκονται στο εσωτερικό της.

Στην διπλωματική εργασία αναπτύχθηκε ένα Κλινικό Ιατρικό Πληροφοριακό Σύστημα βασισμένο σε τεχνολογίες ανάπτυξης Διαδικτυακών Εφαρμογών, που σχεδιάστηκε με βασικό χρήστη έναν γιατρό που διαχειρίζεται ιατρικούς φακέλους ασθενών.

Αντικείμενο της Διπλωματικής

Κύριο αντικείμενο της διπλωματικής εργασίας είναι να γίνει μία πρακτική εφαρμογή της χρήσης διεθνών προτύπων ανταλλαγής ιατρικών δεδομένων σε ένα Ιατρικό Πληροφοριακό Σύστημα. Για τον λόγο αυτό, αποφασίστηκε η δημιουργία διαδικτυακής εφαρμογής, που επικεντρώνεται στην επικοινωνία και την ανταλλαγή πληροφοριών μεταξύ των παρόχων υγείας, δημιουργώντας ένα συγκεκριμένο πρότυπο προκειμένου τα δεδομένα και οι πληροφορίες που μεταφέρονται να είναι όχι μόνο αποτελεσματικά και πρακτικά, αλλά και ασφαλή.

Για την ανάπτυξη της εφαρμογής χρησιμοποιήθηκαν το πρότυπο ανταλλαγής HL7 FHIR για την επιτυχή επικοινωνία μεταξύ των επαγγελματιών υγείας, η διαδικτυακή υπηρεσία REST API, και σύγχρονες τεχνολογίες ανάπτυξης διαδικτυακών εφαρμογών, προκειμένου η εφαρμογή να είναι εύκολη στην χρήση της.

Επιπροσθέτως, αντικείμενο της εφαρμογής αποτελεί η ποιοτική αναβάθμιση των υπηρεσιών υγείας για καθημερινά σενάρια χρήσης. Χαρακτηριστικά παραδείγματα περίπτωσης χρήσης τέτοιας εφαρμογής είναι η επίσκεψη σε έναν γιατρό για γνωμάτευση, παραπομπή για εξετάσεις, ή χορήγηση φαρμακευτικής αγωγής, η μετάβαση σε ένα φαρμακείο για την εκτέλεση της, κλπ.

Οργάνωση της Διπλωματικής

Η διπλωματική εργασία οργανώθηκε σύμφωνα με τα επόμενα κεφάλαια της. Στο επόμενο κεφάλαιο, παρατίθενται οι τεχνολογίες με τις οποίες αναπτύχθηκε η εφαρμογή. Στο τρίτο κεφάλαιο, αναλύονται τα κύρια ζητούμενα και οι απαιτήσεις της εφαρμογής, και περιγράφονται τα βασικότερα στοιχεία της. Στη συνέχεια, γίνεται λεπτομερής ανάλυση της σχεδίασης της εφαρμογής, περιγράφοντας παράλληλα τις βασικές οντότητες που χρησιμοποιούνται. Στο πέμπτο κεφάλαιο, δίνονται οι απαραίτητες πληροφορίες σχετικά με την υλοποίηση της παραγόμενης εφαρμογής, ενώ ταυτόχρονα περιγράφονται διάφορα σενάρια χρήσης της σε πρακτικό επίπεδο. Στο τελευταίο κεφάλαιο, κατατίθενται τα εξαγόμενα συμπεράσματα που ελήφθησαν στο πέρας της εργασίας, και συγχρόνως γίνονται και σχετικές προτάσεις για ενδεχόμενες επεκτάσεις σε επόμενες φάσεις εξέλιξης της εφαρμογής.

Κεφάλαιο 2: Τεχνολογίες

Το συγκεκριμένο κεφάλαιο δείχνει, σε εισαγωγικό επίπεδο, τις τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής. Συγκεκριμένα αναλύονται το πρότυπο ανταλλαγής ιατρικών δεδομένων HL7 FHIR, και παρουσιάζονται οι τεχνολογίες ανάπτυξης διαδικτυακών εφαρμογών PHP, JavaScript, Ajax, JSON και Node.js, που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής. Τέλος παρουσιάζεται το Σύστημα Διαχείρισης Βάσεων Δεδομένων PostgreSQL, που επιλέχθηκε για την υποστήριξη των δεδομένων που θα χρησιμοποιεί η εφαρμογή.

HL7 FHIR

Το πρότυπο FHIR(Fast Healthcare Interoperability Resources)[1] του οργανισμού Health Level 7 (HL7) περιγράφει ένα μοντέλο ανταλλαγής δεδομένων ιατρικών στοιχείων, που μπορεί να χρησιμοποιηθεί από ιατρικά πληροφοριακά συστήματα.

Η ανάγκη υιοθέτησης ενός προτύπου ανταλλαγής ιατρικών δεδομένων προέκυψε από τη διαπίστωση ότι υφιστάμενες εφαρμογές ιατρικού περιεχομένου αποθηκεύουν και μεταφέρουν δεδομένα με ποικίλες μορφές. Κατά συνέπεια, ήταν πολύ δύσκολη η διασφάλιση της διαλειτουργικότητας μεταξύ διαφόρων εφαρμογών που χειρίζονταν ιατρικά δεδομένα με διαφορετικό τρόπο.

Το πρότυπο HL7 FHIR χρησιμοποιεί οντολογίες για την περιγραφή ιατρικών εννοιών και υποστηρίζει διεπαφή διαδικτυακών υπηρεσιών REST. Η υιοθέτηση του προτύπου HL7 FHIR δίνει τη δυνατότητα στην εφαρμογή που θα αναπτυχθεί να ανταλλάσσει ιατρικά δεδομένα με άλλα ιατρικά πληροφοριακά συστήματα, που βασίζονται στο πρότυπο αυτό.

PHP

Η PHP[2] είναι γλώσσα προγραμματισμού που χρησιμοποιείται για την δημιουργία δυναμικών ιστοσελίδων, εκτελείται σε περιβάλλον εξυπηρετητή ιστοσελίδων (π.χ. Apache). Όταν ο εξυπηρετητής ιστοσελίδων δεχθεί αίτημα για μία σελίδα PHP, τη διαβιβάζει στην PHP μηχανή (engine), που εκτελεί τις εντολές και παράγει πληροφορία σε μορφή HTML, που αποστέλλεται στο πρόγραμμα πελάτη (browser) για παρουσίαση. Η PHP έχει επίσης τη δυνατότητα να επικοινωνεί με βάσεις δεδομένων, δίνοντας τη δυνατότητα ανάκτησης/ενημέρωσης/εισαγωγής δεδομένων μέσα από το περιβάλλον του Παγκόσμιου Ιστού.

JavaScript

Η JavaScript[3] (JS) είναι γλώσσα σεναρίων για εφαρμογές που εκτελούνται σε περιβάλλον προγράμματος πελάτη (web browser) και ενσωματώνεται σε σελίδες HTML. Μας δίνει τη δυνατότητα διαχείρισης στοιχείων που εμφανίζονται σε μια ιστοσελίδα, όπως το

άνοιγμα/κλείσιμο menu, τον έλεγχο συμπλήρωσης των υποχρεωτικών πεδίων μιας φόρμας κ.α.

Υπάρχουν πολλές βιβλιοθήκες Javascript που μπορούν να χρησιμοποιηθούν για την ανάπτυξη μιας διαδικτυακής εφαρμογής, όπως η jQuery, η Angular κ.α.

Με την υιοθέτηση της τεχνολογίας Node.js, είναι εφικτή η ανάπτυξη εφαρμογών που χρησιμοποιούν Javascript σε περιβάλλον διακομιστή.

Ajax

Ο όρος Ajax[4] (Asynchronous JavaScript and XML), αν και δεν αποτελεί ξεχωριστή τεχνολογία, περιγράφει τη χρήση ασύγχρονης κλήσης σε περιβάλλον JavaScript και ανταλλαγής δεδομένων με μορφή κειμένου. Είναι ένας όρος που επινοήθηκε το 2005 από τον Jesse James Garrett για τη χρήση μιας σειράς υπάρχουσών τεχνολογιών μαζί, δηλαδή, HTML ή XHTML, CSS, JavaScript, DOM, XML, XSLT, XMLHttpRequest. Όταν αυτές οι τεχνολογίες συνδυάζονται στο πρότυπο Ajax, οι εφαρμογές Ιστού πραγματοποιούν γρήγορες, σταδιακές ενημερώσεις στη διεπαφή χρήστη. Δηλαδή η σελίδα δεν φορτώνεται από την αρχή σε ένα πρόγραμμα περιήγησης κάνοντας πιο γρήγορη την εφαρμογή.

Παρόλο που το X στον Ajax σημαίνει XML, πλέον στις μέρες μας προτιμάται το πρότυπο ανταλλαγής δεδομένων JSON έναντι του XML, λόγω του ότι έχει πολλά οφέλη, δηλαδή να είναι μέρος της JavaScript, επομένως είναι πιο ελαφρύ σε μέγεθος. Το JSON και το XML συμμετέχουν για πληροφορίες συσκευασίας στο πρότυπο Ajax.

JSON

Το JSON[5] είναι μία τεχνολογία που χρησιμοποιείται για την ανταλλαγή δεδομένων που περιλαμβάνονται σε αντικείμενα εφαρμογών σε μορφή κειμένου, που περιλαμβάνει ζεύγη της μορφής (όνομα : τιμή) για κάθε στοιχείο του αντικειμένου. Η τεχνολογία αυτή έχει τη δυνατότητα αναπαράστασης σύνθετης δομής, καθώς υποστηρίζει πίνακες στοιχείων και ενθυλάκωση. Τα παραπάνω χαρακτηριστικά, την κατέστησαν προσφιλή τεχνολογία ανταλλαγής δεδομένων και σε άλλα προγραμματιστικά περιβάλλοντα και στην ουσία αντικατέστησε την XML, που χρησιμοποιούταν στο παρελθόν για τον ίδιο σκοπό.

Ένα βασικό πλεονέκτημα του JSON είναι η χρήση του σε διαδικτυακές εφαρμογές, όπου υπάρχει συχνή ασύγχρονη επικοινωνία πελάτη – εξυπηρετητή.

Επίσης σύγχρονα περιβάλλοντα διαχείρισης δεδομένων, όπως η PostgreSQL για ανάπτυξη της εφαρμογής, υποστηρίζουν πεδία τύπου json, καθιστώντας έτσι εφικτή τη δυνατότητα αποτύπωσης πολλαπλών, σύνθετων τιμών σε ένα πεδίο.

Node.js

Το Node.js[6] είναι ένα περιβάλλον εκτέλεσης ServerSide JavaScript εφαρμογών, δηλαδή εφαρμογών που αναπτύχθηκαν σε JavaScript που εκτελούνται στον εξυπηρετητή (server), καθώς και μία πλατφόρμα ανάπτυξης λογισμικού, (κυρίως διακομιστών) η οποία έχει αναπτυχθεί σε περιβάλλον JavaScript.

Το Node.js υποστηρίζει την ασύγχρονη επικοινωνία μεταξύ πελάτη εξυπηρετητή με τη χρήση μεθόδων, που αντιδρούν σε ενέργειες-συμβάντα (events).

PostgreSQL

Η PostgreSQL[7] είναι μία σχεσιακή βάση δεδομένων, που έχει αναπτυχθεί με ανοικτό κώδικα και υποστηρίζει όλους τους τύπους που χρησιμοποιούνται από σχεσιακές βάσεις δεδομένων και επιπλέον, πεδία JSON, που επιτρέπουν την αποθήκευση πολλαπλών τιμών στο ίδιο πεδίο.

Η δυνατότητα αυτή την καθιστά κατάλληλη για την δημιουργία της βάσης δεδομένων που βασίζεται η εφαρμογή.

Κεφάλαιο 3: Ανάλυση του Θέματος

Στο παρόν κεφάλαιο αναφέρεται η διαδικασία που ακολουθήθηκε για την ανάλυση των απαιτήσεων της εφαρμογής, που βασίζονται σε σενάριο χρήσης. Στη συνέχεια περιγράφεται η Αρχιτεκτονική της πλατφόρμας που θα αναπτυχθεί για να υποστηριχθούν οι παραπάνω λειτουργίες της εφαρμογής.

Απαιτήσεις εφαρμογής

Η εφαρμογή θα καλύπτει τις ανάγκες ενός ιατρού και συγκεκριμένα θα δίνει τη δυνατότητα σε ένα θεράποντα ιατρό να συμβουλευέται και να ενημερώνει το φάκελο ενός ασθενή που τον επισκέπτεται.

Το σενάριο χρήσης που εξετάστηκε είναι το εξής:

Ένας Ιατρός υποδέχεται έναν Ασθενή στο Ιατρείο του. Ο ιατρός αναζητά από το σύστημα τον ιατρικό φάκελο του ασθενή. Συγκεκριμένα, μέσα από το φάκελο, ο θεράπων ιατρός έχει πρόσβαση στα γενικά στοιχεία του Ασθενή και το ιστορικό του:

- Επισκέψεις σε Ιατρό με τις συνοδευόμενες Διαγνώσεις
- Αποτελέσματα Εργαστηριακών Εξετάσεων που έχουν γίνει
- Φαρμακευτική Αγωγή που έχει δοθεί στον Ασθενή
- Ιστορικό περίθαλψης/φροντίδας του Ασθενή

Ο Ιατρός στη συνέχεια, με βάση το ιστορικό και αφού εξετάσει τον Ασθενή έχει τις ακόλουθες επιλογές:

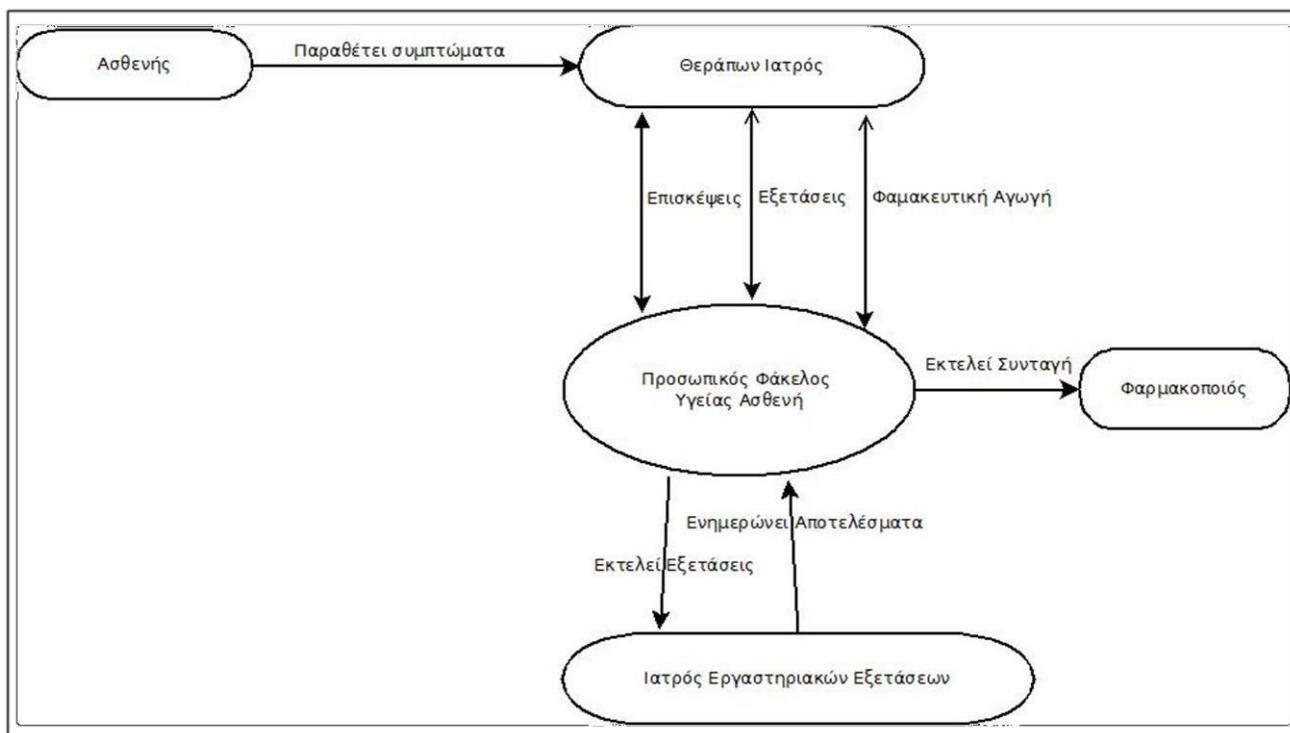
- 1) να καταλήξει σε μια διάγνωση
- 2) να χορηγήσει στον Ασθενή Φαρμακευτική Αγωγή
- 3) να ζητήσει από τον Ασθενή να κάνει Διαγνωστικές Εξετάσεις, τις οποίες και θα καταγράψει στην εφαρμογή.
- 4) Να τον οδηγήσει σε άλλον Ιατρό, δηλώνοντας αναρμόδιος να χειριστεί το ιατρικό ζήτημα του Ασθενή

Κατά συνέπεια ο Ιατρός χρησιμοποιεί την εφαρμογή και ενημερώνει τον Προσωπικό Φάκελο Υγείας ως εξής:

- Στην αρχική περίπτωση ο Ιατρός καταχωρεί την επίσκεψη και τη διάγνωση του στο ιστορικό
- στη δεύτερη περίπτωση της Φαρμακευτικής Αγωγής, καταχωρεί μία συνταγή που συμπεριλαμβάνει φαρμακευτικές ουσίες για να λαμβάνει ο Ασθενής και καθορίζει και ένα Πλάνο Φροντίδας. Τη συνταγή θα εκτελέσει κάποιος Φαρμακοποιός που θα έχει πρόσβαση στο φάκελο του Ασθενή μόνο όσον αφορά τις συνταγές υπό εκτέλεση. Ο Φαρμακοποιός, αποκτά πρόσβαση στο φαρμακευτικό ιστορικό του Ασθενή και βλέπει τη συνταγή που πρέπει να εκτελέσει.

Ο Φαρμακοποιός αφού εκτελέσει τη συνταγή, ενημερώνει το ιστορικό για την εκτέλεση της συνταγής.

- στην τρίτη περίπτωση ο ιατρός υποδεικνύει τον ασθενή του στο να πραγματοποιήσει Εργαστηριακές Εξετάσεις. Ο ιατρός καταχωρεί στην εφαρμογή ένα παραπεμπτικό με δέκτη ένα Διαγνωστικό Κέντρο. Ο υπεύθυνος ιατρός του Διαγνωστικού Κέντρου θα καταχωρήσει με τη σειρά του τα αποτελέσματα των εξετάσεων μέσω της εφαρμογής στον Προσωπικό Ιατρικό Φάκελο του Ασθενή, ώστε να είναι στη διάθεση του Ιατρού που τις παρήγγειλε.
- Στην τελευταία περίπτωση ο Ιατρός καταχωρεί στο φάκελο το λόγο της επίσκεψης του Ασθενή και αιτιολογεί την παραπομπή του Ασθενή σε Ιατρό άλλης Ειδικότητας



Εικόνα 1: Μοντέλο Διάδρασης Χρηστών

Σύμφωνα με το παραπάνω σενάριο προκύπτουν οι ακόλουθες απαιτήσεις από την πλευρά των χρηστών:

- Ο Ασθενής θα έχει είσοδο στον ιατρικό του φάκελο χωρίς να μπορεί να τον αλλάξει.
- Ο Ιατρός θα έχει καθολική είσοδο στον ιατρικό φάκελο του ασθενή του.
- Ο Ασθενής εξουσιοδοτεί τον Ιατρό που επισκέπτεται, για έλεγχο του φακέλου του.
- Ο ιατρός εργαστηριακής ειδικότητας θα έχει περιορισμένη πρόσβαση στο φάκελο του Ασθενή και συγκεκριμένα μόνο στα παραπεμπτικά για εξετάσεις από τον θεράποντα Ιατρό.

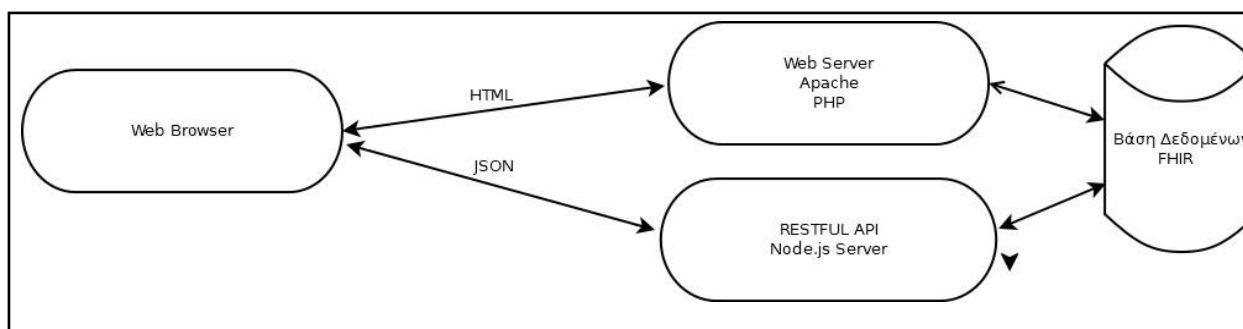
- Ο φαρμακοποιός θα έχει περιορισμένη πρόσβαση στο φάκελο του Ασθενή και συγκεκριμένα μόνο στη φαρμακευτική αγωγή που έχει ορίσει ο θεράπων Ιατρός.

Ο Ατομικός Φάκελος του Ασθενή θα φιλοξενείται σε εξυπηρετητή με Σύστημα Βάσης Δεδομένων και θα είναι προσβάσιμος μέσω Διαδικτύου.

Η ανταλλαγή των δεδομένων με τον εξυπηρετητή θα βασίζεται στο πρότυπο ανταλλαγής δεδομένων HL7 FHIR.

Αρχιτεκτονική της πλατφόρμας

Η δομή της πλατφόρμας της εφαρμογής βασίζεται στο μοντέλο του πελάτη-εξυπηρετητή (client/server) με έναν εξυπηρετητή, ο οποίος θα βρίσκεται στη διάθεση όλων των εμπλεκόμενων χρηστών (Ασθενών, Ιατρών, Φαρμακοποιών, Διαγνωστικών Κέντρων) και θα παρέχει πρόσβαση σε Ιατρικούς Φακέλους Ασθενών υπό μορφή Ηλεκτρονικού Ιατρικού Αρχείου (Electronic Medical Record – EMR).



Εικόνα 2: Αρχιτεκτονική Πλατφόρμας

Ο χρήστης θα χρησιμοποιεί το Φυλλομετρητή του (Web Browser) για να αποκτήσει πρόσβαση στον εξυπηρετητή που θα παρέχει πρόσβαση σε ιατρικά δεδομένα τα οποία θα αντλούνται από βάση δεδομένων βασισμένη σε Οντότητες του προτύπου FHIR και θα υποστηρίζει τις λειτουργίες αναζήτησης και ανάγνωσης στους ακόλουθους πόρους:

- τον πόρο για τον ασθενή, προκειμένου να παρέχονται δημογραφικά στοιχεία στον πελάτη. Όταν ένας χρήστης αναζητά ασθενείς χωρίς κριτήρια αναζήτησης, λαμβάνει μια λίστα με όλους τους ασθενείς στους οποίους έχει πρόσβαση
- αναζήτηση και πρόσβαση σε γενικά έγγραφα ασθενών
- αναζήτηση και ανάγνωση των κλινικών πόρων του ασθενούς (Εξετάσεις, Επισκέψεις, Φάρμακα, κ.α.)

Τα δεδομένα της βάσης δεδομένων θα μπορούν να ανταλλάσσονται με το πρότυπο HL7 FHIR, δίνοντας τη δυνατότητα επέκτασης της λειτουργικότητας και σε άλλες εφαρμογές που θα αναπτυχθούν στο πλαίσιο της ιατρικής φροντίδας ασθενών.

Η εφαρμογή που θα αναπτυχθεί θα είναι βασισμένη σε PHP σελίδες που θα έχουν δυνατότητα να επικοινωνούν με τη βάση δεδομένων FHIR και θα παρέχει την ευκαιρία στο χρήστη (Ιατρό) να:

- επιλέξει Ασθενή
- ανοίξει τον Προσωπικό του φάκελο
- προωθήσει το φάκελο σε άλλο Ιατρό
- προσθέσει διάγνωση
- προσθέσει Φαρμακευτική Αγωγή
- προσθέσει Παραπεμπτικό για Διαγνωστικές Εξετάσεις

Στο σύστημα θα χρησιμοποιηθεί η τεχνολογία διαδικτυακών υπηρεσιών REST, που επιτρέπουν τις αλληλεπιδράσεις με πόρους της εφαρμογής μέσω αιτημάτων που μεταβιβάζονται διαδικτυακά με το πρωτόκολλο HTTP, υπό μορφή αιτημάτων/αποκρίσεων GET/PUT/DELETE/POST και εκτελούνται απευθείας στον πόρο του διακομιστή.

Το πρότυπο HL7 FHIR, προσδιορίζει σαφώς το πλαίσιο ανάπτυξης RESTful εφαρμογών σε FHIR δεδομένα και επιβάλλει ότι κάθε τύπος πόρου θα έχει το ίδιο σύνολο αλληλεπιδράσεων ώστε να γίνει η διαχείριση των πόρων με έναν εξαιρετικά λεπτομερή τρόπο. Έτσι για παράδειγμα ορίζονται από το πλαίσιο οι παρακάτω αλληλεπιδράσεις με πόρους της εφαρμογής:

- read - Πρόσβαση στην τρέχουσα κατάσταση του πόρου
- update – Ενημέρωση ενός υπάρχοντος πόρου με το αναγνωριστικό του
- delete - Διαγραφή πόρου

Κεφάλαιο 4: Σχεδίαση Θέματος

Το κεφάλαιο αυτό περιγράφει τη Σχεδίαση της Εφαρμογής και περιλαμβάνει:

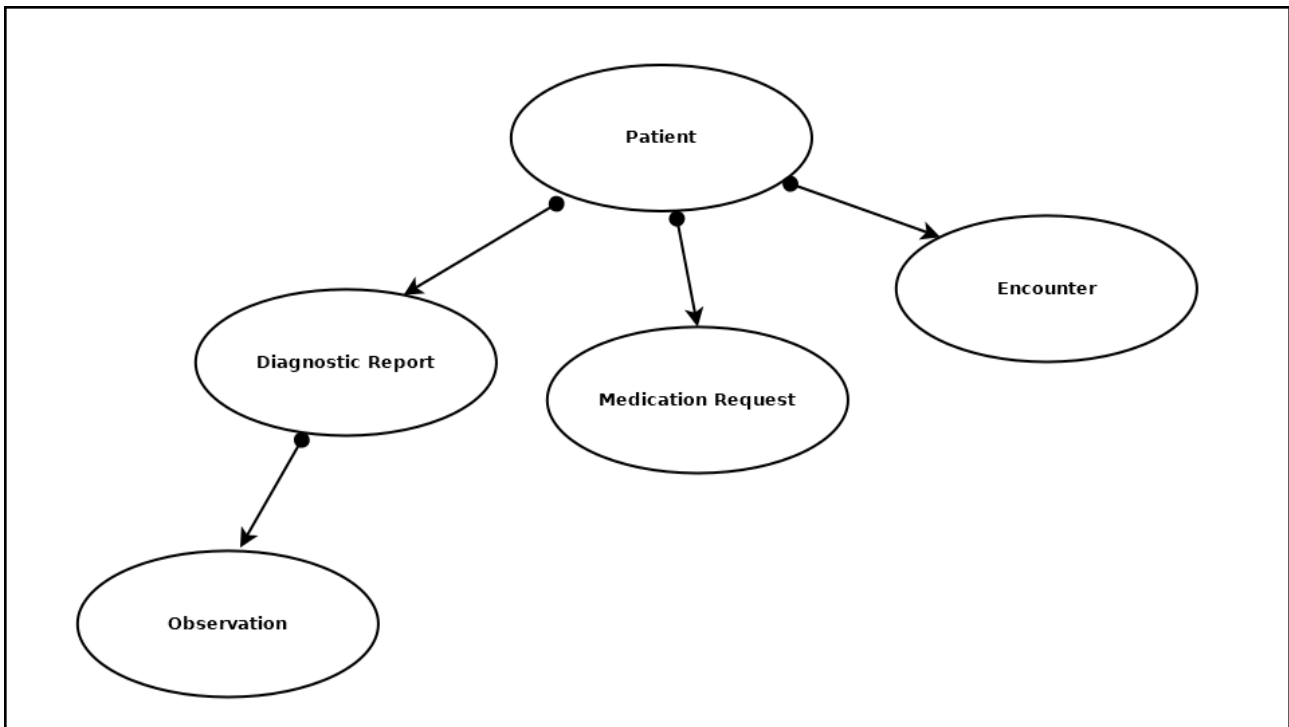
- Το Σχεδιασμό της Βάσης.
- Τη περιγραφή των λειτουργιών που θα προσφέρουν οι Σελίδες που θα αναπτυχθούν για την Εφαρμογή Πελάτη (Ιατρού).
- Την περιγραφή των λειτουργιών που θα παρέχονται μέσω της REST API διεπαφής που θα αναπτυχθεί.

Σχεδίαση Βάσης Δεδομένων

Οντότητες-Πόροι του FHIR

Η βάση δεδομένων σχεδιάστηκε ώστε να υποστηρίζει τις Οντότητες-Πόρους που ορίζει το πρότυπο HL7 FHIR, ώστε να είναι εφικτό στο μέλλον να ανταλλάσσονται δεδομένα με άλλα ιατρικά πληροφοριακά συστήματα που βασίζονται στο HL7 FHIR.

Στο παρακάτω Σχήμα φαίνονται οι Οντότητες που χρησιμοποιήθηκαν από την εφαρμογή και οι συσχετίσεις τους:



Εικόνα 3: Μοντέλο Οντοτήτων της Εφαρμογής

Patient

Τα δεδομένα σε αυτό τον πόρο καλύπτουν κυρίως τις δημογραφικές πληροφορίες ενός Ασθενή (**Patient**[9]). Οι πληροφορίες αυτές είναι χρήσιμες στην εφαρμογή για την ταυτοποίηση του Ασθενή και την παράθεση στοιχείων προσωπικού χαρακτήρα που πρέπει να είναι προσβάσιμες στην χρήστη-Ιατρό, όπως ο Αριθμός Ασφάλισης τηλέφωνο επικοινωνίας, διεύθυνση, κ.α. (βλ. Παράδειγμα πληροφορίας σε πόρο **Patient**: στο Παράρτημα Α).

Τα πεδία που αναφέρονται στο πρότυπο για τον Ασθενή, περιλαμβάνουν βασικούς τύπους δεδομένων (Λογικές, Ημερομηνίες, κλπ.) και αναφορές σε τύπους άλλων πόρων του προτύπου FHIR. Για παράδειγμα το πεδίο `generalPractitioner` (Γενικός Ιατρός) είναι αναφορά σε πόρο `Practitioner`.

Encounter

Τα δεδομένα σε αυτό τον πόρο καλύπτουν τις πληροφορίες για μια Επίσκεψη (**Encounter**[10]) ασθενή και υποστηρίζει τις πληροφορίες που χρειάζονται κατά τη διαδικασία που ακολουθεί ο ασθενής για να λάβει τις ιατρικές υπηρεσίες που επιθυμεί (βλ. Παράδειγμα πληροφορίας σε πόρο του **Encounter**: Παράδειγμα πληροφορίας σε πόρο **Patient**: στο Παράρτημα Α).

Στα πεδία που αναφέρονται στην Επίσκεψη υπάρχουν αναφορές σε άλλους πόρους με τις οποίες μπορούν να γίνουν οι αντίστοιχες συσχετίσεις. Για παράδειγμα χρησιμοποιώντας το πεδίο `subject` του πόρου **Encounter**, όπου αποθηκεύεται ο κωδικός του Ασθενή (**Patient**), μπορούμε να βρούμε πληροφορίες για τον Ασθενή στον οποίο αναφέρεται μια συγκεκριμένη Επίσκεψη.

Observation

Τα δεδομένα σε αυτό τον πόρο καλύπτουν τις πληροφορίες για τις Καταγραφές (**Observations**[11]) που αφορούν ιατρικά δεδομένα μετρήσεων και εξετάσεων, όπως:

- Ζωτικά στοιχεία όπως το σωματικό βάρος, η αρτηριακή πίεση και η θερμοκρασία
- Εργαστηριακά δεδομένα όπως γλυκόζη αίματος ή εκτιμώμενο GFR
- Αποτελέσματα απεικονίσεων, όπως μετρήσεις οστικής πυκνότητας
- Μετρήσεις από συσκευές όπως δεδομένα ΗΚΓ ή Παλμική Οξυμετρία

Στα πεδία του πόρου της Καταγραφής (Observation), υπάρχουν αναφορές σε άλλους πόρους μπορούμε να αντλήσουμε επιπλέον πληροφορίες (βλ. Παράδειγμα πληροφορίας σε πόρο Observation: Παράδειγμα πληροφορίας σε πόρο **Patient**: στο Παράρτημα Α). Για παράδειγμα χρησιμοποιώντας το πεδίο `encounter` μπορούμε να βρούμε πληροφορίες για την Επίσκεψη όπου έγινε μια συγκεκριμένη Καταγραφή.

Diagnostic Report

Μια διαγνωστική αναφορά (**Diagnostic Report**[12]) περιέχει τις πληροφορίες που παρέχονται από ένα Διαγνωστικό Κέντρο όταν ολοκληρωθούν οι εξετάσεις που διεξήγαγε (βλ. Παράδειγμα πληροφορίας σε πόρο **Diagnostic Report**: στο Παράρτημα Α).

Ο πόρος Diagnostic Report στην ουσία χρησιμοποιείται για να σχηματιστεί μια πλήρης αναφορά που περιλαμβάνει ένα σύνολο Καταγραφών (Observations) που προκύπτουν από εργαστηριακά, απεικονιστικά και άλλα κλινικά και διαγνωστικά δεδομένα.

Ο πόρος Diagnostic Report είναι κατάλληλος για τα ακόλουθα είδη διαγνωστικών αναφορών:

- Εργαστήριο (Κλινική Χημεία, Αιματολογία, Μικροβιολογία κ.λπ.)
- Παθολογία / Ιστοπαθολογία / συναφείς κλάδοι
- Απεικόνιση (ακτινογραφία, αξονική τομογραφία, μαγνητική τομογραφία κ.λπ.)
- Άλλα διαγνωστικά (Καρδιολογία, Γαστρεντερολογία κ.λπ.)

Και σε αυτό τον πόρο υπάρχουν αναφορές σε άλλους πόρους που θα μπορούν να μας φανούν χρήσιμες για την εφαρμογή. Για παράδειγμα χρησιμοποιώντας το πεδίο **encounter**, όπου αποθηκεύεται ο κωδικός του της Επίσκεψης (Encounter) μπορούμε να βρούμε πληροφορίες για την Επίσκεψη όπου ζητήθηκαν οι εν λόγω εξετάσεις. Επίσης χρησιμοποιώντας το πεδίο **result** βρίσκουμε πίνακα με Καταγραφές (Observations) απ' όπου μπορούμε να αντλήσουμε τις τιμές για όλες τις Καταγραφές που έχουν γίνει στην Εξέταση.

Medication Request

Αυτός ο πόρος καλύπτει όλους τους τύπους παραγγελιών για φάρμακα (**MedicationRequest**[13]) για έναν ασθενή (βλ. Παράδειγμα πληροφορίας σε πόρο **Medication Request**: στο Παράρτημα Α).

Στα πεδία αυτού του πόρου περιλαμβάνονται, μεταξύ άλλων, αναφορές σε αντικείμενα τύπου Patient (πεδίο **subject**) και Encounter (πεδίο **encounter**), μέσω των οποίων μπορούν να γίνουν οι αντίστοιχες συσχετίσεις στην εφαρμογή.

Όπως φαίνεται και από τα παραδείγματα δεδομένων των παραπάνω πόρων που παρατίθενται στο Παράρτημα Α, η βάση δεδομένων πρέπει να υποστηρίζει πεδία τύπου JSON, ώστε να υπάρχει η δυνατότητα αποθήκευσης πολλαπλών και σύνθετων τιμών σε διάφορα πεδία και αναφορά σε άλλες οντότητες της βάσης.

Σχεδίαση Εφαρμογής Πελάτη (Ιατρού)

Η εφαρμογή θα βασίζεται στο μοντέλο ανάπτυξης διαδικτυακών εφαρμογών και θα συμπεριλαμβάνει τις παρακάτω σελίδες:

- **Σελίδα Επιλογής Ασθενή**, που θα δίνεται στο χρήστη (Ιατρό) η δυνατότητα επιλογής του Ασθενή βάσει του Ονόματός του. Ο χρήστης (Ιατρός) έχει δυνατότητα να δώσει μέρος του ονόματός του Ασθενή. Η εφαρμογή στη συνέχεια θα εμφανίσει λίστα Ασθενών που τα ονόματά τους περιέχουν το ζητούμενο κείμενο για να διαλέξει ο χρήστης το φάκελο του ασθενή που θέλει να εξετάσει.
- **Σελίδα παρουσίασης πληροφοριών Ασθενή**, όπου θα παρουσιάζονται οι πληροφορίες υπό μορφή των παρακάτω Καρτελών (Tabs):
 - **Καρτέλα γενικών πληροφοριών ασθενή**, με τα δεδομένα του Ασθενή, όπως όνομα, επώνυμο, διεύθυνση, τηλέφωνο .
 - **Καρτέλα ιατρικού ιστορικού ασθενή**, με τα συμπτώματα και τις παθήσεις με κατάταξη κατά ημερομηνία, μαζί με σημειώσεις από ιατρούς.
 - **Καρτέλα ιστορικού εργαστηριακών εξετάσεων**, με κατάταξη κατά ημερομηνία όλων των εργαστηριακών εξετάσεων .
 - **Καρτέλα ιστορικού φαρμακευτικών αγωγών**, με κατάταξη κατά ημερομηνία όλων των φαρμακευτικών αγωγών που λαμβάνει ο Ασθενής.

Στη σελίδα αυτή ο ιατρός θα μπορεί να ζητήσει να μεταφερθεί ο φάκελος του ασθενή με μήνυμα σε άλλο ιατρό.

- **Σελίδα προσθήκης ιατρικής επίσκεψης**, όπου ο χρήστης – Ιατρός θα εισάγει σε φόρμα τον τύπο και την κατηγορία του **συμπτώματος/πάθησης** του ασθενή που εξετάζει. Τα συμπτώματα θα εμφανίζονται σε λίστα για πιο εύκολη συμπλήρωση της φόρμας.
- **Σελίδα προσθήκης φαρμακευτικής αγωγής**, όπου ο χρήστης – Ιατρός θα εισάγει την Φαρμακευτική Αγωγή για τον ασθενή που εξετάζει. Θα μπορεί χρησιμοποιώντας φόρμα να προσθέσει στην φαρμακευτική αγωγή τη φαρμακευτική ουσία/φάρμακο που πρέπει να δοθεί στον ασθενή επιλέγοντας από λίστα με τις διαθέσιμες φαρμακευτικές ουσίες/φάρμακα. Ο χρήστης – Ιατρός θα μπορεί να προσθέσει και άλλες φαρμακευτικές ουσίες/φάρμακα επαναλαμβάνοντας την παραπάνω διαδικασία.
- **Σελίδα αλλαγής φαρμακευτικής αγωγής**, όπου ο χρήστης – Ιατρός μπορεί να αλλάξει ή να διαγράψει τη φαρμακευτική ουσία/φάρμακο που έχει συμπεριλάβει στην Φαρμακευτική Αγωγή πριν την οριστικοποιήσει.

Σχεδίαση Διεπαφής REST API Εφαρμογής

Η εφαρμογή θα καλύπτει τις παρακάτω λειτουργίες μέσω της REST API διεπαφής :

Υποστηριζόμενες Λειτουργίες Ασθενών (Patient)

Εύρεση στοιχείων όλων των Ασθενών

Η διεπαφή θα επιστρέφει τη λίστα όλων των εγγραφών του πόρου Patient σε μορφή json όταν σταλεί αίτημα GET στη διεύθυνση <serverBaseURL>/patients, επιλέγοντας όλες τις εγγραφές της οντότητας Patient.

Εύρεση στοιχείων ενός Ασθενή βάσει Κωδικού

Η διεπαφή θα επιστρέφει τα περιεχόμενα μιας εγγραφής του πόρου Patient που έχει συγκεκριμένο κωδικό (ID) σε μορφή json όταν σταλεί αίτημα GET στη διεύθυνση <serverBaseURL>/patient/:id, επιλέγοντας όλες τις εγγραφές της οντότητας Patient

Αποστολή στοιχείων ενός Ασθενή βάσει Κωδικού

Η διεπαφή θα αποστέλλει κρυπτογραφημένα τα περιεχόμενα μιας εγγραφής του πόρου Patient που έχει συγκεκριμένο κωδικό (ID) όταν σταλεί αίτημα POST στη διεύθυνση <serverBaseURL>/patient/send/:id

Εύρεση στοιχείων ενός Ασθενή βάσει Ονόματος

Η διεπαφή θα επιστρέφει τα περιεχόμενα μιας εγγραφής του πόρου Patient που το όνομά του περιέχει το δοθέν κείμενο σε μορφή json όταν σταλεί αίτημα GET στη διεύθυνση <serverBaseURL>/patient/name/:id

Υποστηριζόμενες Λειτουργίες Επισκέψεων (Encounter)

Εύρεση στοιχείων Επίσκεψης βάσει Κωδικού

Η διεπαφή θα επιστρέφει τα περιεχόμενα μιας εγγραφής του πόρου Encounter που έχει συγκεκριμένο κωδικό (ID) σε μορφή json όταν σταλεί αίτημα GET στη διεύθυνση <serverBaseURL>/encounter/:id

Εύρεση στοιχείων ενός Ασθενή βάσει Κωδικού

Η διεπαφή θα επιστρέφει τα περιεχόμενα μιας εγγραφής του πόρου Encounter που έχει κάνει Ασθενής (Patient) που έχει συγκεκριμένο κωδικό (ID) σε μορφή json όταν σταλεί αίτημα GET στη διεύθυνση <serverBaseURL>/patientencounters/:id

Προσθήκη Επίσκεψης

Η διεπαφή θα δημιουργεί νέα εγγραφή στον πόρο Encounter όταν σταλεί αίτημα POST στη διεύθυνση <serverBaseURL>/encounters. Τα στοιχεία της επίσκεψης θα δοθούν σε μορφή json.

Υποστηριζόμενες Λειτουργίες Φαρμακευτικής Αγωγής (MedicationRequest)

Εύρεση στοιχείων Φαρμακευτικής Αγωγής βάσει Κωδικού

Η διεπαφή θα επιστρέφει τα περιεχόμενα μιας εγγραφής του πόρου Medication Request που έχει συγκεκριμένο κωδικό (ID) σε μορφή json όταν σταλεί αίτημα GET στη διεύθυνση <serverBaseURL>/medication/:id

Εύρεση στοιχείων Φαρμακευτικής Αγωγής βάσει Επίσκεψης

Η διεπαφή θα επιστρέφει τα περιεχόμενα μιας εγγραφής του πόρου Medication Request που έχει αναφορά σε συγκεκριμένο κωδικό επίσκεψης (ID) σε μορφή json όταν σταλεί αίτημα GET στη διεύθυνση <serverBaseURL>/encountermedications/:id

Εύρεση Φαρμακευτικής Αγωγής ενός Ασθενή βάσει Κωδικού

Η διεπαφή θα επιστρέφει τα περιεχόμενα μιας εγγραφής του πόρου Medication Request που έχει κάνει Ασθενής (Patient) που έχει συγκεκριμένο κωδικό (ID) σε μορφή json όταν σταλεί αίτημα GET στη διεύθυνση <serverBaseURL>/patientmedication/:id

Προσθήκη Φαρμακευτικής Αγωγής

Η διεπαφή θα δημιουργεί νέα εγγραφή στον πόρο Medication Request με στοιχεία για τα φάρμακα που θα δοθούν σε μορφή json όταν σταλεί αίτημα POST στη διεύθυνση <serverBaseURL>/medication

Διαγραφή Φαρμακευτικής Αγωγής βάσει Κωδικού

Η διεπαφή θα διαγράψει εγγραφή του πόρου Medication Request, με βάση τον Κωδικό (ID), όταν σταλεί αίτημα DELETE στη διεύθυνση <serverBaseURL>/medication

Ενημέρωση Φαρμακευτικής Αγωγής βάσει Κωδικού

Η διεπαφή θα ενημερώσει εγγραφή του πόρου Medication Request, με βάση τον Κωδικό (ID), όταν σταλεί αίτημα PUT στη διεύθυνση <serverBaseURL>/medication. Τα στοιχεία για τα φάρμακα θα δοθούν σε μορφή json.

Υποστηριζόμενες Λειτουργίες Διαγνωστικών Εξετάσεων (DiagnosticReports)

Εύρεση στοιχείων Διαγνωστικών Εξετάσεων Αγωγής βάσει Επίσκεψης

Η διεπαφή θα επιστρέφει τα περιεχόμενα μιας εγγραφής του πόρου DiagnosticReports που έχει αναφορά σε συγκεκριμένο κωδικό επίσκεψης (ID) σε μορφή json όταν σταλεί αίτημα GET στη διεύθυνση <serverBaseUrl>/encounterreports/:id

Εύρεση Διαγνωστικών Εξετάσεων ενός Ασθενή βάσει Κωδικού

Η διεπαφή θα επιστρέφει τα περιεχόμενα μιας εγγραφής του πόρου DiagnosticReports που έχει κάνει Ασθενής (Patient) που έχει συγκεκριμένο κωδικό (ID) σε μορφή json όταν σταλεί αίτημα GET στη διεύθυνση <serverBaseUrl>/patientreports

Κεφάλαιο 5: Υλοποίηση

Στο παρόν κεφάλαιο θα παρουσιαστεί η Ανάπτυξη Βάσης Δεδομένων που χρησιμοποιήθηκε στην εφαρμογή. Επίσης, θα περιγραφούν οι Σελίδες που αναπτύχθηκαν για την Εφαρμογή Πελάτη (Ιατρού). Τέλος, θα περιγραφούν οι REST API λειτουργίες που αναπτύχθηκαν

Ανάπτυξη Βάσης Δεδομένων

PostgreSQL

Για την υποστήριξη της πλατφόρμας της εφαρμογής Ηλεκτρονικού Ιατρικού Αρχείου χρησιμοποιήθηκε η βάση δεδομένων PostgreSQL που δίνει τη δυνατότητα αποθήκευσης πολλαπλών τιμών σε πεδίο, υπό μορφή json πίνακα και επιπλέον την ανταλλαγή δεδομένων σε json μορφή, ώστε να ικανοποιούνται οι απαιτήσεις της εφαρμογής του πρότυπου FHIR, που περιγράφηκαν στο προηγούμενο Κεφάλαιο.

Για την εγκατάσταση της PostgreSQL προτιμήθηκε το πακέτο Wapp της Bitnami (βλ. Παράρτημα Β – Εγκατάσταση Εργαλείων), το οποίο περιλαμβάνει, επιπλέον από τη PostgreSQL, τον εξυπηρετητή ιστοσελίδων Apache με υποστήριξη PHP σελίδων. Η εγκατάσταση του μας δίνει τη δυνατότητα ανάπτυξης διαδικτυακής εφαρμογής με σελίδες PHP που έχουν τη δυνατότητα επικοινωνίας με Βάση Δεδομένων PostgreSQL.

Η βάση δεδομένων που αναπτύχθηκε περιλαμβάνει Οντότητες που περιλαμβάνονται στο πρότυπο FHIR και περιγράφηκαν στο προηγούμενο Κεφάλαιο.

FHIR Base

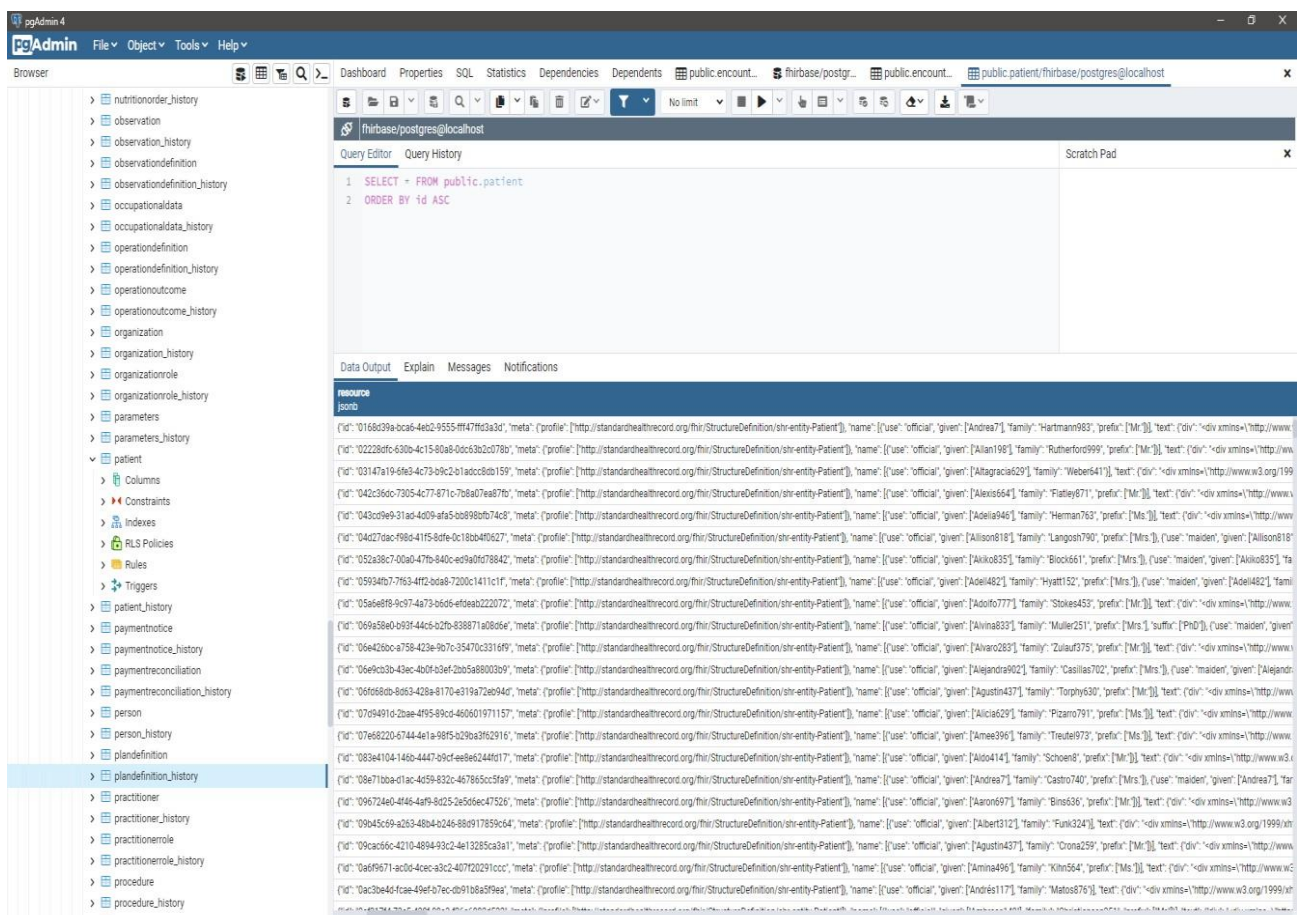
Επιπλέον χρησιμοποιήθηκε το εργαλείο ανοικτού κώδικα FHIRbase, για την αποθήκευση και εργασία με δεδομένα FHIR σε βάση δεδομένων PostgreSQL. Το εργαλείο αυτό μας δίνει την ευκαιρία να φτιάξουμε μια βάση Δεδομένων PostgreSQL που θα διαθέτει Οντότητες – Πόρους που υποστηρίζονται από το πρότυπο FHIR καθώς και να εισάγουμε ενδεικτικά δεδομένα διαθέσιμα από το διαδίκτυο¹ σε αυτήν. Με αυτόν τον τρόπο η εφαρμογή που θα αναπτύξουμε μπορεί να έχει πρόσβαση σε δεδομένα FHIR και να γίνουν με αυτά δοκιμές στην εφαρμογή.

¹ <https://github.com/fhirbase/fhirbase/blob/master/demo/bundle.ndjson.zip>

Ενδεικτικά το πλήθος των στοιχείων που φορτώνονται με την παραπάνω διαδικασία σε Οντότητες της βάσης δεδομένων είναι:

Οντότητα	Χρήση	Πλήθος
<i>Patient</i>	Ασθενείς	600
<i>Encounter</i>	Επισκέψεις	15384
<i>DiagnosticReport</i>	Διαγνωστικές Εξετάσεις	4143
<i>MedicationRequest</i>	Φαρμακευτική Αγωγή	2553
<i>Observation</i>		57619

Πίνακας 1: Πλήθος Οντοτήτων στη βάση FHIRBase



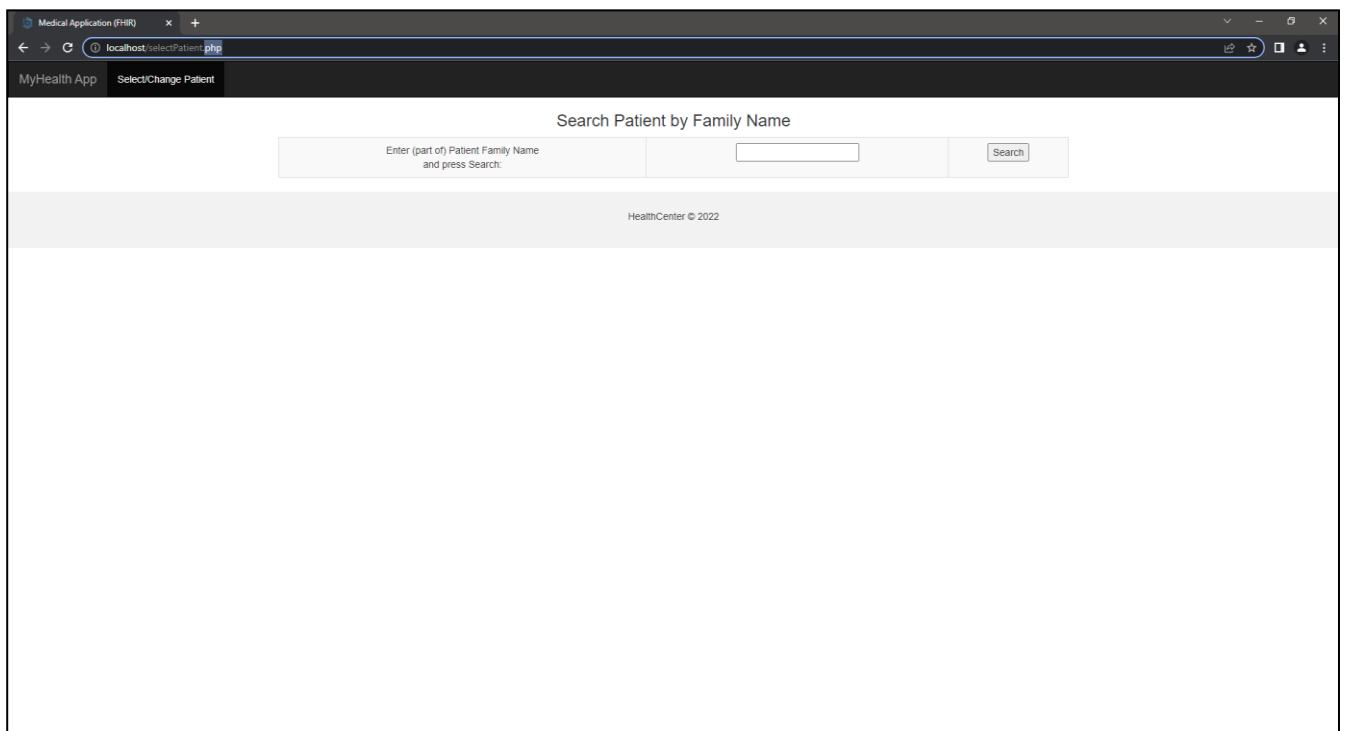
Εικόνα 4: Προβολή Οντότητας Patient με χρήση του εργαλείου pgAdmin4

Υλοποίηση Σελίδων Εφαρμογής

Οι σελίδες της εφαρμογής αναπτύχθηκαν σε HTML5 με χρήση PHP κώδικα. Επίσης έγινε χρήση JavaScript βιβλιοθηκών (όπως η βιβλιοθήκη DataTables) και του μηχανισμού AJAX, που υποστηρίζει η βιβλιοθήκη jQuery. Για την παρουσίαση των ιστοσελίδων χρησιμοποιήθηκαν στοιχεία Bootstrap (βλ. Παράρτημα Γ - Κώδικας Εφαρμογής).

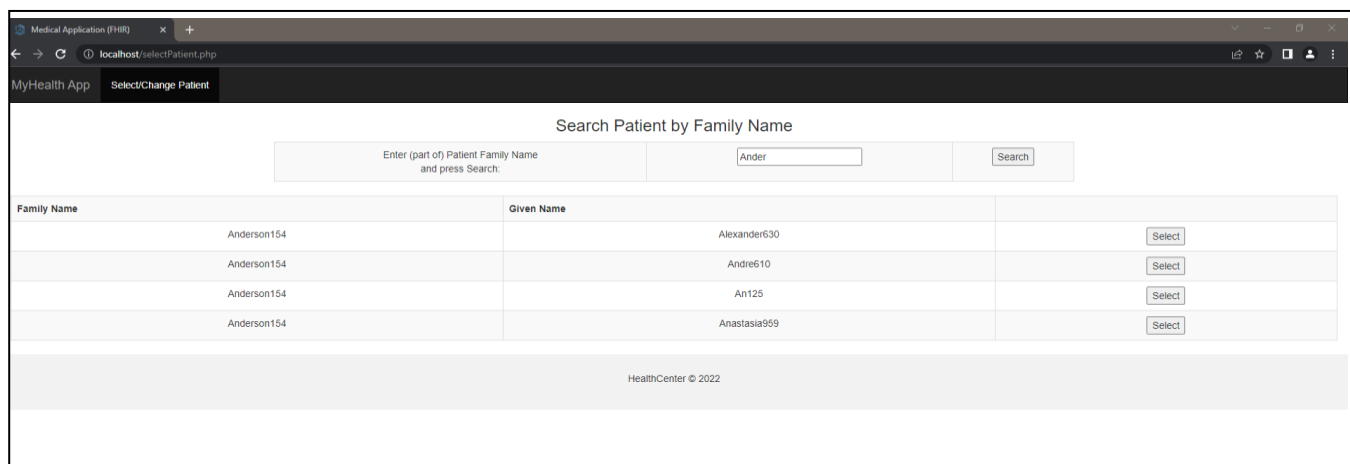
Αρχική Σελίδα (Φόρμα επιλογής Ασθενή)

Στη σελίδα αυτή ([selectPatient.php](#)) ο χρήστης - Ιατρός εισάγει στη φόρμα το μέρος του Ονόματος του Ασθενή που αναζητά στη βάση.



Εικόνα 5: Φόρμα Αναζήτησης Ασθενών με κριτήριο μέρος του Ονόματος

Με την υποβολή υποβάλλεται ερώτημα στη βάση δεδομένων που επιστρέφει τη λίστα των Ασθενών που το όνομά τους περιέχει το κείμενο που όρισε ο χρήστης – Ιατρός. Από τη λίστα αυτή μπορεί να επιλέξει ο χρήστης – Ιατρός τον φάκελο του Ασθενή τον οποίο θέλει να εξετάσει.



Search Patient by Family Name

Enter (part of) Patient Family Name and press Search.

Ander Search

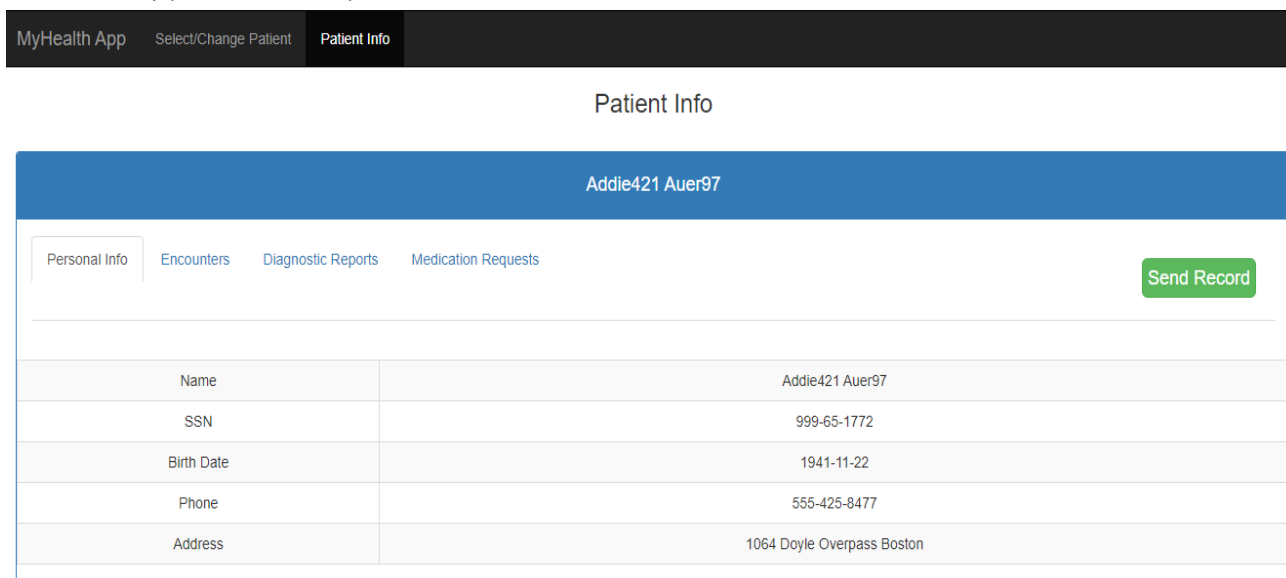
Family Name	Given Name	
Anderson154	Alexander630	Select
Anderson154	Andre610	Select
Anderson154	An125	Select
Anderson154	Anastasia959	Select

HealthCenter © 2022

Εικόνα 6: Λίστα Ασθενών με κριτήριο μέρος του Ονόματος

Σελίδα προβολής Ιατρικού Φακέλου Ασθενή

Όταν ανοίγει η σελίδα ([showPatient.php](#)) εμφανίζεται αρχικά η Καρτέλα Προσωπικών Στοιχείων του Ασθενή (Personal Info).



MyHealth App Select/Change Patient Patient Info

Patient Info

Addie421 Auer97

Personal Info Encounters Diagnostic Reports Medication Requests Send Record

Name	Addie421 Auer97
SSN	999-65-1772
Birth Date	1941-11-22
Phone	555-425-8477
Address	1064 Doyle Overpass Boston

Εικόνα 7- Προβολή Καρτέλας Προσωπικών Στοιχείων

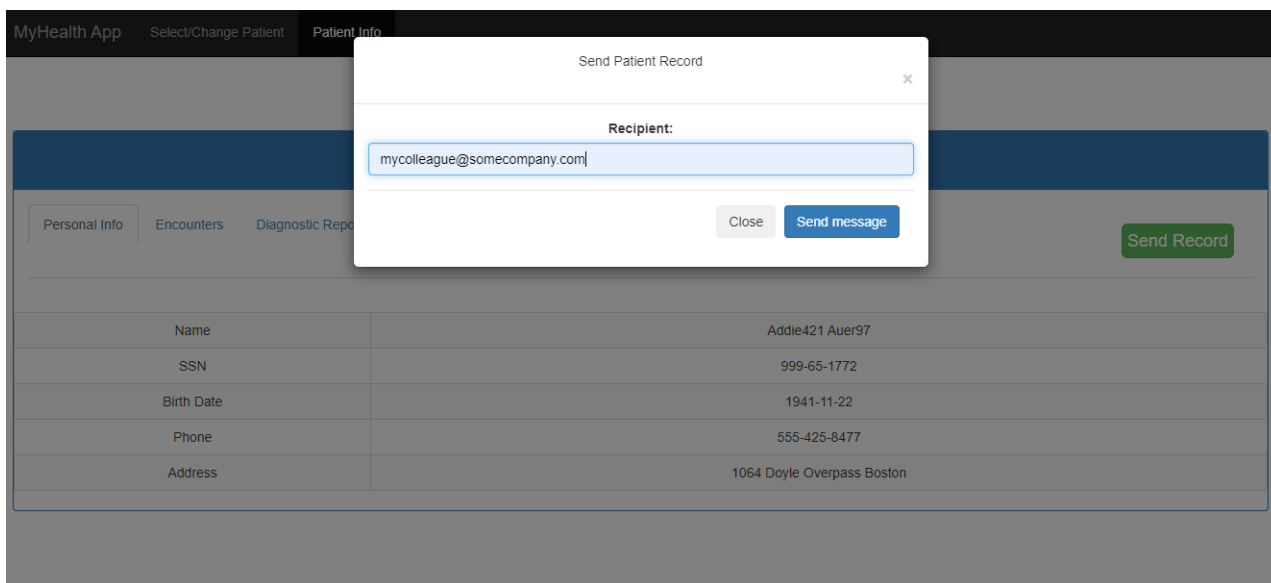
Ο χρήστης-ιατρός μπορεί να επιλέξει να :

- Αποστέλλει τον Ιατρικό Φάκελο σε άλλο Ιατρό, πατώντας το Κουμπί **SendRecord**

- Εμφανίσει Αποτελέσματα Διαγνωστικών Εξετάσεων, επιλέγοντας την καρτέλα **Diagnostic Reports**
- Εμφανίσει Φαρμακευτικές Αγωγές, επιλέγοντας την καρτέλα **Medication Requests**
- Εμφανίσει Επισκέψεις του Ασθενή, επιλέγοντας την καρτέλα **Encounters**

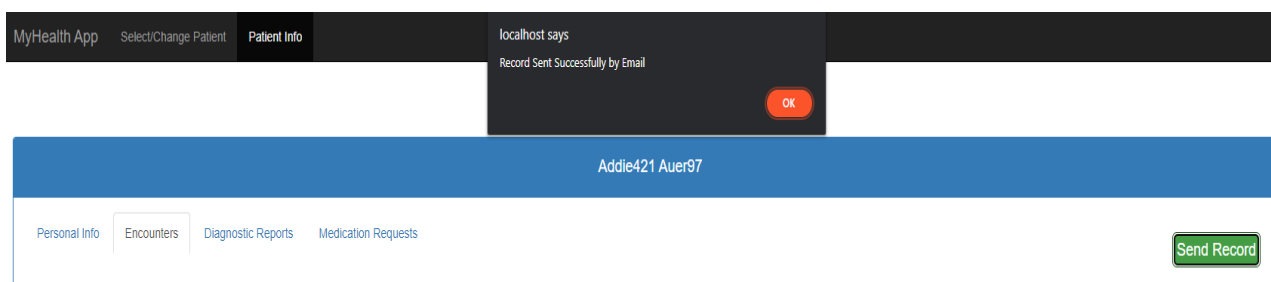
Αποστολή Ιατρικού Φακέλου

Σε περίπτωση που ο χρήστης-ιατρός πατήσει το κουμπί αποστολής του φακέλου (Send Record) εμφανίζεται φόρμα όπου συμπληρώνει το email του παραλήπτη ιατρού.



Εικόνα 8: Φόρμα αποστολής φακέλου

Όταν πατήσει το κουμπί Send Message της φόρμας στέλνεται με POST ο κωδικός του ασθενή και η διεύθυνση του παραλήπτη στην υπηρεσία `<serverBaseURL>/patient/send/:id` της REST API διεπαφής και αυτή αναλαμβάνει την αποστολή του φακέλου κρυπτογραφημένα στον παραλήπτη. Ο χρήστης-ιατρός ειδοποιείται σχετικά με το αποτέλεσμα από την απόκριση της διεπαφής με κατάλληλο alert μήνυμα.

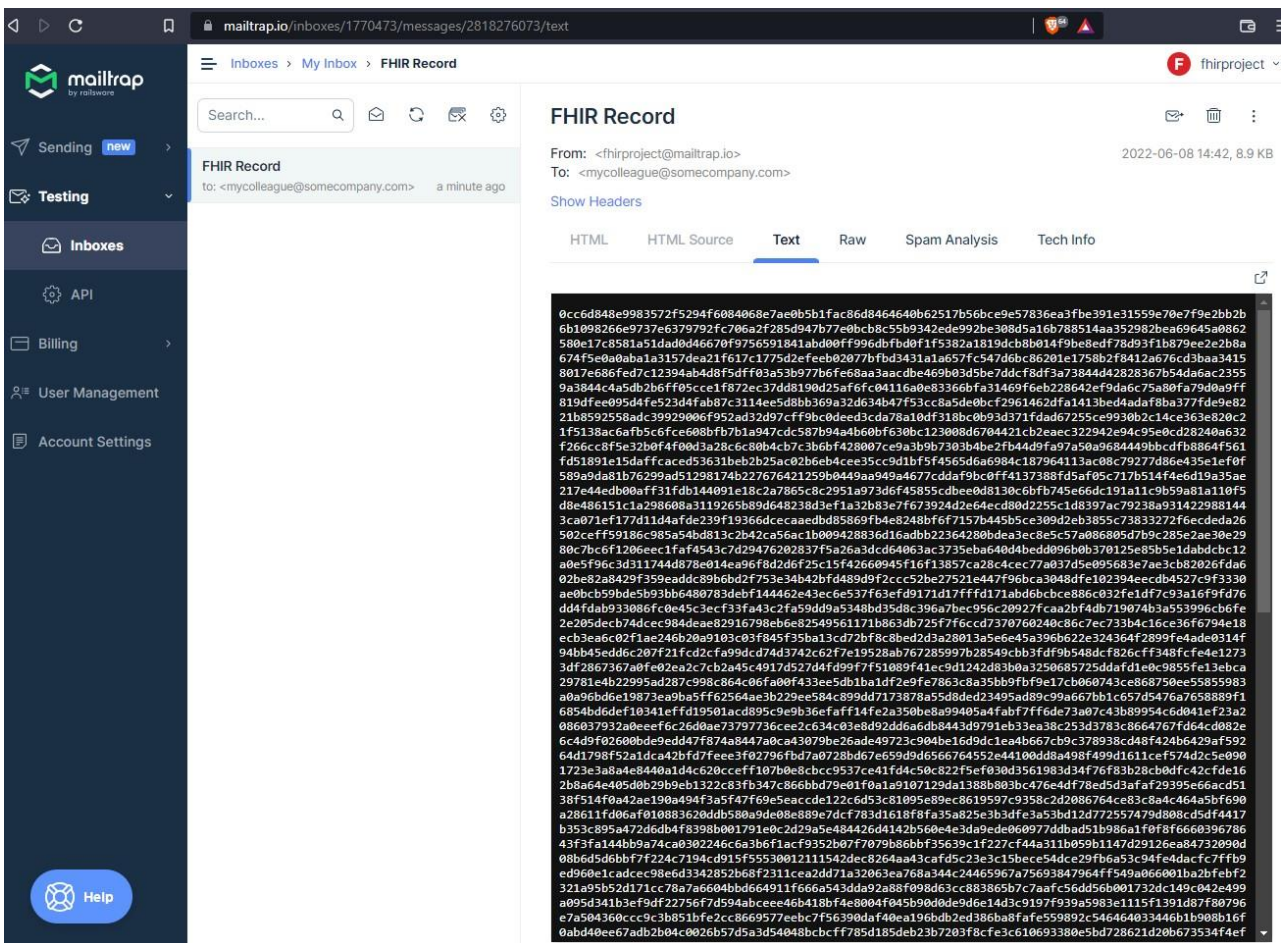


Εικόνα 9: Ειδοποίηση χρήστη για την αποστολή του φακέλου

Το σύστημα στέλνει το φάκελο του Ασθενή με κρυπτογραφημένο μήνυμα, όπως φαίνεται στην παρακάτω εικόνα, για λόγους ασφαλείας. Ο παραλήπτης έχοντας το κλειδί της κρυπτογράφησης μπορεί να αποκρυπτογραφήσει το μήνυμα και να ανοίξει το φάκελο του Ασθενή.

Στην παρακάτω εικόνα παρουσιάζεται ένα κρυπτογραφημένο μήνυμα που στάλθηκε από την εφαρμογή προς μια πλασματική ηλεκτρονική διεύθυνση από διεύθυνση του συστήματος διαχείρισης ηλεκτρονικής αλληλογραφίας mailtrap που μας δίνει τη δυνατότητα να παρακολουθήσουμε την αποστολή της ηλεκτρονικής αλληλογραφίας από λογαριασμούς που δημιουργούμε σε αυτό.

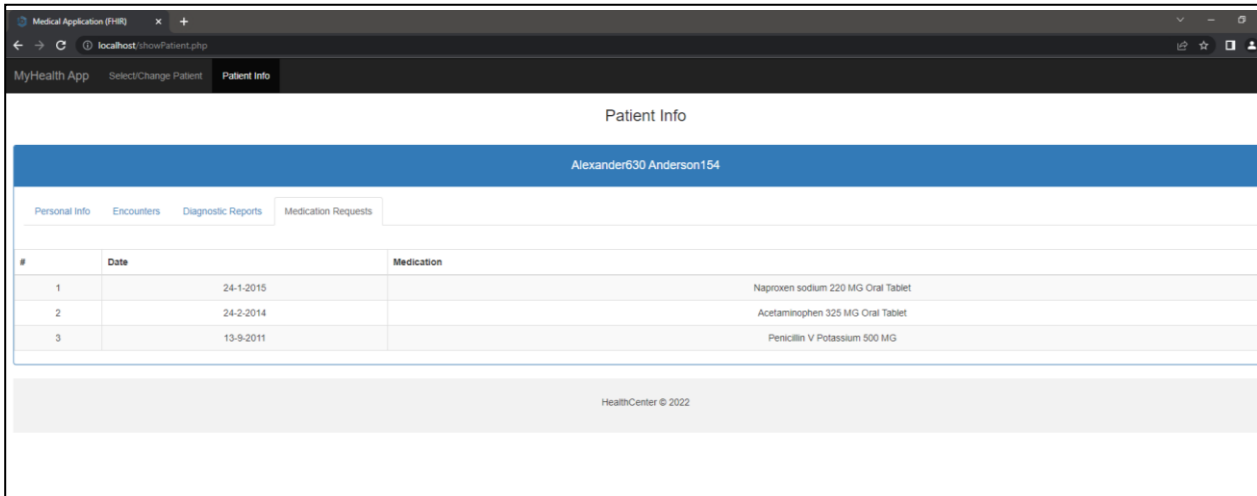
Συγκεκριμένα το μήνυμα φαίνεται να στέλνεται από το λογαριασμό fhirproject@mailtrap.io προς την εικονική διεύθυνση mycolleague@somecompany.com



Εικόνα 10: Λήψη κρυπτογραφημένου μηνύματος στο σύστημα mailtrap

Εμφάνιση Φαρμακευτικών Αγωγών του Ασθενή

Επιλέγοντας την καρτέλα Medication Requests ο χρήστης-Ιατρός βλέπει όλες τις Φαρμακευτικές Αγωγές που έχουν δοθεί στον Ασθενή που εξετάζει.



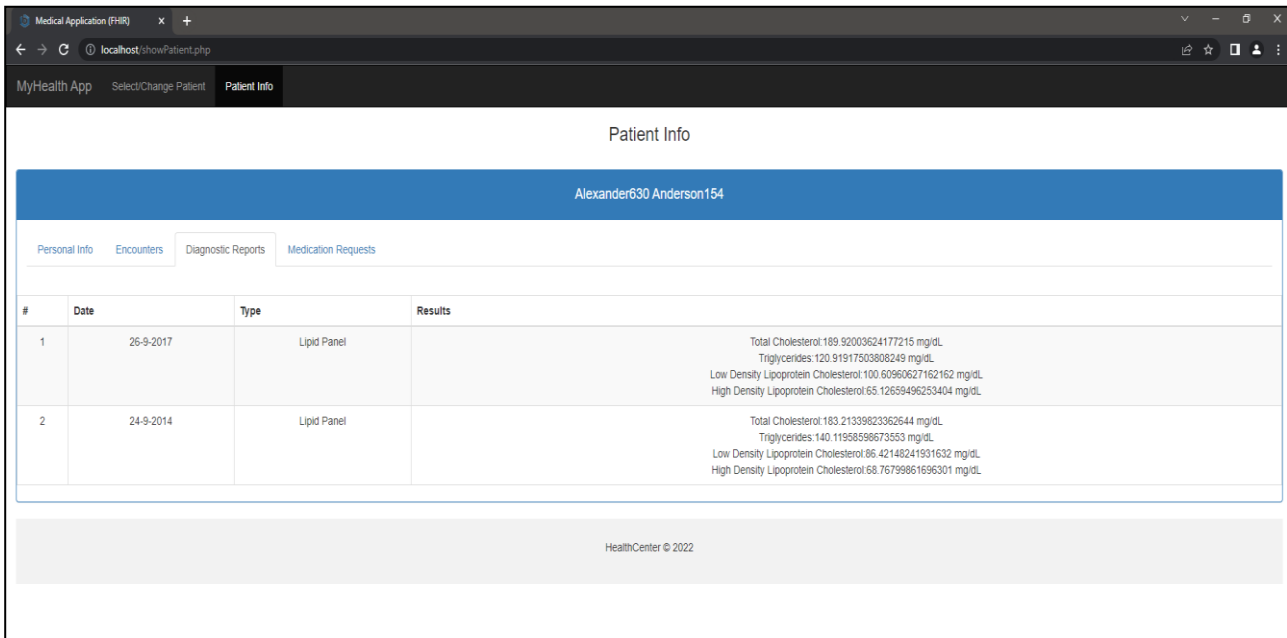
The screenshot shows a web browser window displaying the 'Patient Info' page for Alexander630 Anderson154. The 'Medication Requests' tab is selected. The table below lists three medication requests:

#	Date	Medication
1	24-1-2015	Naproxen sodium 220 MG Oral Tablet
2	24-2-2014	Acetaminophen 325 MG Oral Tablet
3	13-9-2011	Penicillin V Potassium 500 MG

Εικόνα 11: Προβολή Καρτέλας Φαρμακευτικών Αγωγών Επιλεγμένου Ασθενή

Εμφάνιση Διαγνωστικών Εξετάσεων Ασθενή

Επιλέγοντας την καρτέλα Diagnostic Reports ο χρήστης-Ιατρός βλέπει τα αποτελέσματα των Διαγνωστικών Εξετάσεων που έχει κάνει ο Ασθενής.



The screenshot shows the 'Patient Info' page for Alexander630 Anderson154 with the 'Diagnostic Reports' tab selected. The table below displays two diagnostic reports:

#	Date	Type	Results
1	26-9-2017	Lipid Panel	Total Cholesterol:189.92003624177215 mg/dL Triglycerides:120.91917503808249 mg/dL Low Density Lipoprotein Cholesterol:100.60960627162162 mg/dL High Density Lipoprotein Cholesterol:65.12659496253404 mg/dL
2	24-9-2014	Lipid Panel	Total Cholesterol:183.21339623362644 mg/dL Triglycerides:140.11958598673553 mg/dL Low Density Lipoprotein Cholesterol:86.42148241931632 mg/dL High Density Lipoprotein Cholesterol:68.76799861696301 mg/dL

Εικόνα 12: Προβολή Καρτέλας Αποτελεσμάτων Διαγνωστικών Εξετάσεων Ασθενή

Εμφάνιση Ιατρικών Επισκέψεων του Ασθενή

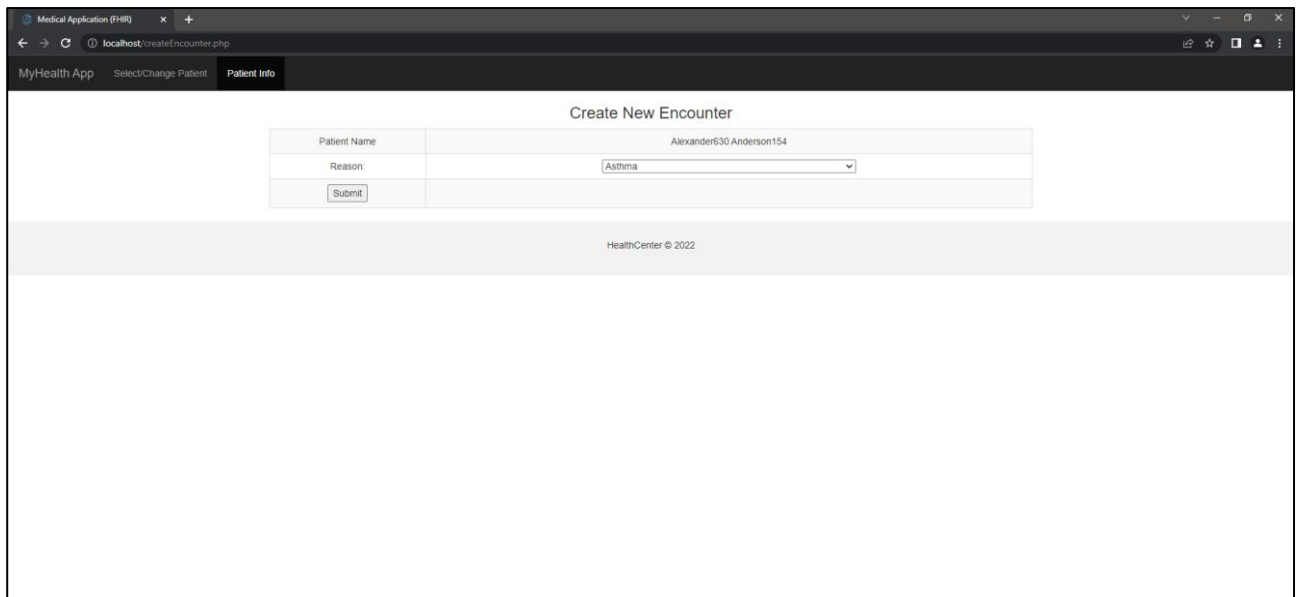
The screenshot shows a web browser window with the URL `localhost/showPatient.php`. The application is titled "MyHealth App" and has tabs for "Select/Change Patient" and "Patient Info". The main content area is titled "Patient Info" and displays the name "Alexander630 Anderson154". Below the name are tabs for "Personal Info", "Encounters", "Diagnostic Reports", and "Medication Requests". An "ADD ENCOUNTER" button is located in the top right corner of the encounters section. The main table lists 11 encounters with the following data:

#	Date	Symptom	Medication	
1	20-2-2018	Acute viral pharyngitis (disorder)		Show Details
2	26-9-2017			Show Details
3	23-12-2015	Acute viral pharyngitis (disorder)		Show Details
4	18-5-2015	Acute viral pharyngitis (disorder)		Show Details
5	24-1-2015		Naproxen sodium 220 MG Oral Tablet	Show Details
6	24-9-2014			Show Details
7	24-2-2014		Acetaminophen 325 MG Oral Tablet	Show Details
8	12-2-2014	Acute viral pharyngitis (disorder)		Show Details
9	21-9-2011			Show Details
10	13-9-2011	Streptococcal sore throat (disorder)	Penicillin V Potassium 500 MG	Show Details
11	25-8-2011	Viral sinusitis (disorder)		Show Details

Εικόνα 13: Προβολή Καρτέλας Ιατρικών Επισκέψεων Επιλεγμένου Ασθενή

Στην σελίδα αυτή ο χρήστης – Ιατρός μπορεί να ζητήσει να δει λεπτομέρειες από προηγούμενες επισκέψεις του Ασθενή και επιπλέον να προσθέσει μια νέα Επίσκεψη, οπότε θα οδηγηθεί στην επόμενη σελίδα ([createEncounter.php](#)), όπου θα επιλέξει την αιτία της επίσκεψης από πλαίσιο επιλογής.

Σελίδα Εισαγωγής Νέας Επίσκεψης

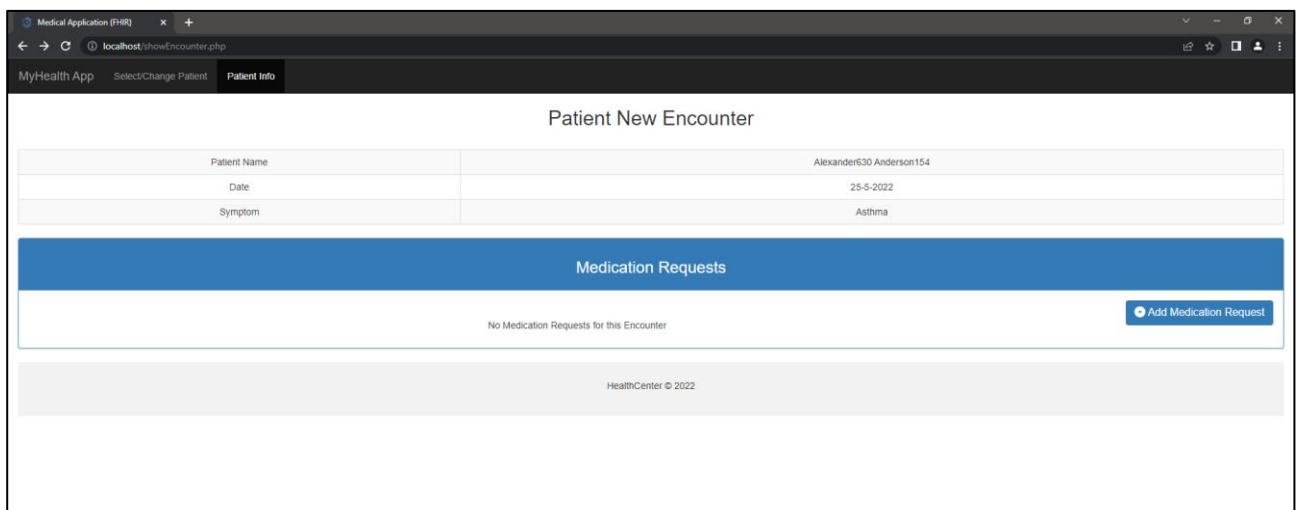


Patient Name	Alexander630 Anderson154
Reason	Asthma
<input type="button" value="Submit"/>	

HealthCenter © 2022

Εικόνα 14: Φόρμα Εισαγωγής νέας Επίσκεψης σε Ασθενή

Με την υποβολή της Φόρμας δημιουργείται νέα Επίσκεψη για τον Ασθενή στη Βάση Δεδομένων. Στη συνέχεια ο χρήστης – Ιατρός οδηγείται στη σελίδα Διαχείρισης Νέας Επίσκεψης ([showEncounter.php](#))



Patient Name	Alexander630 Anderson154
Date	25-5-2022
Symptom	Asthma

Medication Requests

No Medication Requests for this Encounter

HealthCenter © 2022

Εικόνα 15: Σελίδα Διαχείρισης νέας Επίσκεψης σε Ασθενή

Ο χρήστης – Ιατρός μπορεί να προσθέσει σε αυτή τη σελίδα νέα Φαρμακευτική Αγωγή οπότε οδηγείται στη Φόρμα προσθήκης Φαρμακευτικής Αγωγής ([createMedication.php](#)).

Σελίδα προσθήκης Φαρμακευτικής Αγωγής

Medical Application (FHR) x +
localhost/createMedication.php
MyHealth App Select/Change Patient Patient Info

Create New Medicine Request

Choose a medicine from the list or enter a new one:

Alteplase 1 MG/ML Injec

Submit

HealthCenter © 2022

Εικόνα 16: Φόρμα Προσθήκης νέας Φαρμακευτικής Αγωγής

Με την υποβολή της Φόρμας δημιουργείται νέα Φαρμακευτική Αγωγή για τον Ασθενή και ο χρήστης – Ιατρός οδηγείται και πάλι στη σελίδα Διαχείρισης της Φαρμακευτικής Αγωγής, όπου έχει τη δυνατότητα να προσθέσει επιπλέον Φαρμακευτική Αγωγή, όπως επίσης να αλλάξει ή να διαγράψει μια υπάρχουσα.

Σελίδα Διαχείρισης Φαρμακευτικής Αγωγής

Medical Application (FHR) x +
localhost/showEncounter.php
MyHealth App Select/Change Patient Patient Info

Patient New Encounter

Patient Name	Alexander630 Anderson154
Date	25-5-2022
Symptom	Asthma

Medication Requests

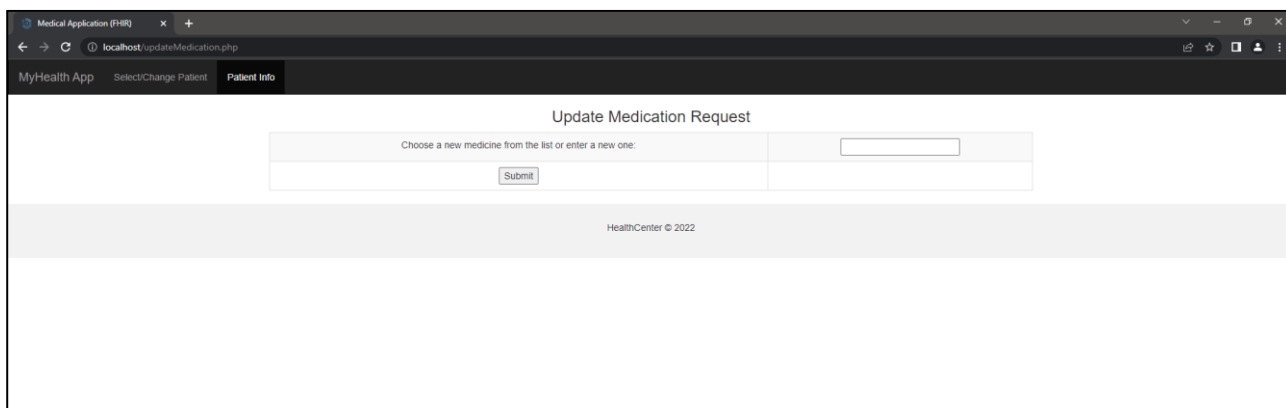
Add Medication Request

#	Name		
1	Alteplase 1 MG/ML Injectable Solution	Update	Delete

HealthCenter © 2022

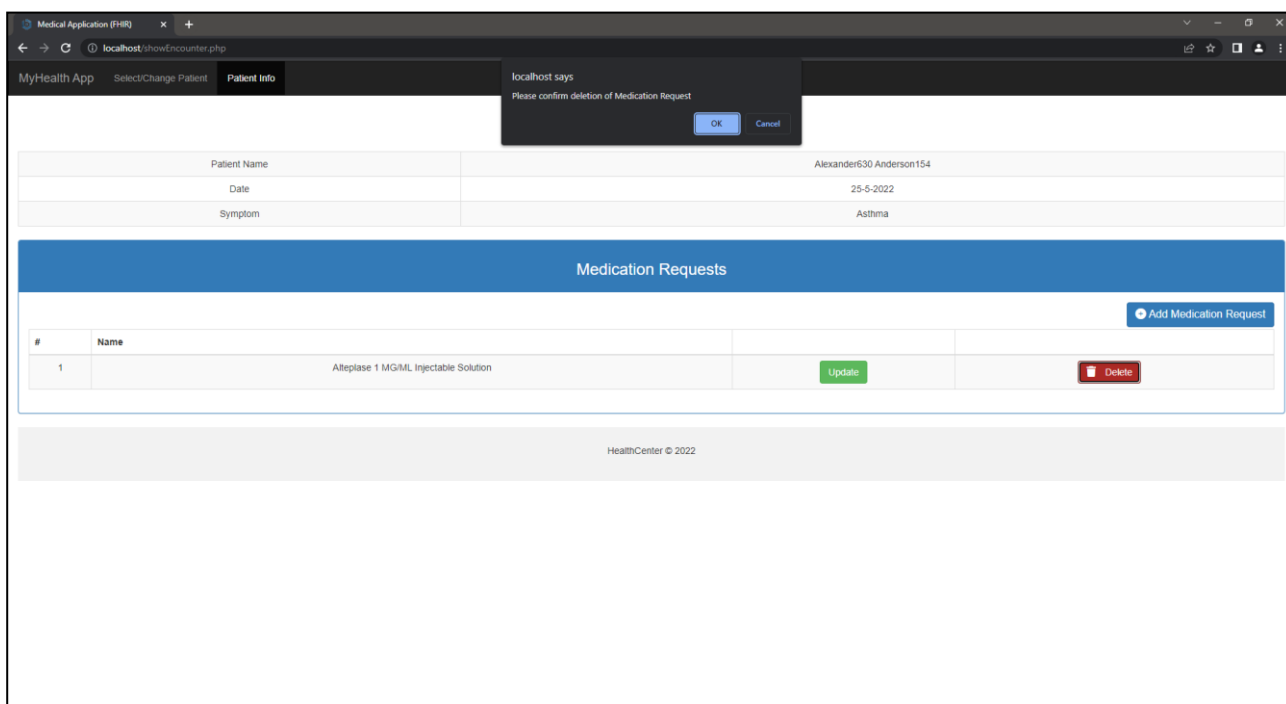
Εικόνα 17: Σελίδα Διαχείρισης Φαρμακευτικής Αγωγής

Αν ο χρήστης – Ιατρός επιλέξει να αλλάξει μια υπάρχουσα Φαρμακευτική αγωγή οδηγείται στη σελίδα **updateMedication.php**. Με την υποβολή της φόρμας ενημερώνεται η εγγραφή της Φαρμακευτικής Αγωγής στη Βάση Δεδομένων.



Εικόνα 18: Φόρμα Ενημέρωσης Φαρμακευτικής Αγωγής

Αν ο χρήστης – Ιατρός επιλέξει να διαγράψει μια υπάρχουσα Φαρμακευτική αγωγή οδηγείται στη σελίδα **deleteMedication.php**. Με την υποβολή της φόρμας, και αφού ο χρήστης – Ιατρός επιβεβαιώσει την πρόθεσή του για διαγραφή, διαγράφεται η εγγραφή της Φαρμακευτικής Αγωγής από τη Βάση Δεδομένων.



Εικόνα 19: Διαγραφή Φαρμακευτικής Αγωγής

Υλοποίηση REST API Διεπαφής

Η ανάπτυξη των διεπαφών του REST API πρωτοκόλλου που αναπτύχθηκε έγινε σε Node.js που δίνει τη δυνατότητα γρήγορης ανάπτυξης server-side εφαρμογών σε JavaScript.

Ο κώδικας του REST API (βλέπε Παράρτημα Γ – Κώδικας Εφαρμογής) αποτελείται από τη σελίδα index.js που φορτώνεται με την έναρξη του node.js server με την εντολή node ./index.js και υλοποιεί όλες τις λειτουργίες της διεπαφής που αναλύθηκαν στο Κεφάλαιο του Σχεδιασμού της Εφαρμογής και τη σελίδα queries.js, όπου υλοποιούνται όλα οι μέθοδοι που αναφέρονται στη σελίδα index.js και εκτελούν στην ουσία τα ερωτήματα στη βάση δεδομένων firebase.

Στη σελίδα index.js, ορίζονται τα αιτήματα που μπορεί να δεχθεί ο node.js server και ποιες μέθοδοι θα κληθούν σε κάθε αίτημα. Για παράδειγμα αν δεχθεί ο node.js server αίτημα GET <URL>/patient/<id>, όπου <id> είναι ο κωδικός (πεδίο id) ενός Ασθενή (Patient), θα κληθεί η μέθοδος getPatientById που είναι ορισμένη στη σελίδα queries.js.

Υποστηριζόμενες Λειτουργίες Ασθενών (Patient)

Εύρεση στοιχείων όλων των Ασθενών

Όταν η διεπαφή δεχθεί αίτημα GET στη διεύθυνση <serverBaseURL>/patients εκτελείται η μέθοδος getPatients που αναζητά από τη βάση όλες τις εγγραφές του πίνακα Patient, επιλέγοντας τα πεδία Όνομα (given), Επώνυμο (family), Πόλη (city), Οδό (street), Ημερομηνία Γέννησης (birthdate) και Αριθμό Ασφάλισης (ssn), και επιστρέφει τα αποτελέσματα σε μορφή json.

Εύρεση στοιχείων ενός Ασθενή βάσει Κωδικού

Όταν η διεπαφή δεχθεί αίτημα GET στη διεύθυνση <serverBaseURL>/patient/:id εκτελείται η μέθοδος getPatientById που αναζητά από τη βάση την εγγραφή του πίνακα Patient που έχει κωδικό ίσο με την παράμετρο id επιλέγοντας τα πεδία Όνομα (given), Επώνυμο (family), Πόλη (city), Οδό (street), Ημερομηνία Γέννησης (birthdate) και Αριθμό Ασφάλισης (ssn), και επιστρέφει τα αποτελέσματα σε μορφή json.

Αποστολή στοιχείων ενός Ασθενή βάσει Κωδικού

Όταν η διεπαφή δεχθεί αίτημα POST στη διεύθυνση <serverBaseURL>/patient/send/:id εκτελείται η μέθοδος sendPatientById που αναζητά από τη βάση την εγγραφή του πίνακα Patient που έχει κωδικό ίσο με την παράμετρο id, την μετατρέπει σε json μορφή, την κρυπτογραφεί και τη στέλνει στη διεύθυνση του παραλήπτη που έχει λάβει στο σώμα του POST αιτήματος.

Εύρεση στοιχείων ενός Ασθενή βάσει Ονόματος

Όταν η διεπαφή δεχθεί αίτημα **GET** στη διεύθυνση `<serverBaseURL>/patient/name/:text` εκτελείται η μέθοδος **getPatientByName** που αναζητά από τη βάση τις εγγραφές του πίνακα **Patient** που το Επίθετό τους (πεδίο **family**) περιέχει την παράμετρο **text**, επιλέγοντας τα πεδία Όνομα (**given**), Επώνυμο (**family**), Πόλη (**city**), Οδό (**street**), Ημερομηνία Γέννησης (**birthdate**) και Αριθμό Ασφάλισης (**ssn**), και επιστρέφει τα αποτελέσματα σε μορφή json.

Υποστηριζόμενες Λειτουργίες Επισκέψεων (Encounter)

Εύρεση στοιχείων Επίσκεψης βάσει Κωδικού

Όταν η διεπαφή δεχθεί αίτημα **GET** στη διεύθυνση `<serverBaseURL>/encounter/:id` εκτελείται η μέθοδος **getEncounterById** που αναζητά από τη βάση την εγγραφή του πίνακα **Encounter** που έχει κωδικό (πεδίο **id**) ίσο με την παράμετρο **id**, επιλέγοντας τα πεδία που αφορούν τη Διάγνωση (**symptom**) και την Ημερομηνία (**date**) της Επίσκεψης, και επιστρέφει τα αποτελέσματα σε μορφή json.

Εύρεση Επισκέψεων ενός Ασθενή βάσει Κωδικού

Όταν η διεπαφή δεχθεί αίτημα **GET** στη διεύθυνση `<serverBaseURL>/patientencounters/:id` εκτελείται η μέθοδος **getPatientEncounters** που αναζητά από τη βάση τις εγγραφές του πίνακα **Encounter** που έχουν στην αναφορά στο πεδίο του κωδικού Ασθενή (πεδίο **subject**) τιμή ίση με την παράμετρο **id**. Στην απάντηση επιλέγονται τα πεδία που αφορούν τη Διάγνωση (**symptom**) και την Ημερομηνία (**date**) της Επίσκεψης, και επιστρέφει τα αποτελέσματα σε μορφή json.

Προσθήκη Επίσκεψης

Όταν η διεπαφή δεχθεί αίτημα **POST** στη διεύθυνση `<serverBaseURL>/encounters` εκτελείται η μέθοδος **createEncounter** που δημιουργεί νέα εγγραφή στον πόρο **Encounter**. Τα στοιχεία για την επίσκεψη (Ασθενής, Σύμπτωμα) θα συνοδεύουν το αίτημα **POST** και προέρχονται από την υποβολή αντίστοιχης φόρμας.

Υποστηριζόμενες Λειτουργίες Φαρμακευτικής Αγωγής (MedicationRequest)

Εύρεση στοιχείων Φαρμακευτικής Αγωγής βάσει Κωδικού

Όταν η διεπαφή δεχθεί αίτημα **GET** στη διεύθυνση `<serverBaseURL>/medication/:id` εκτελείται η μέθοδος `getMedicationById` που αναζητά από τη βάση την εγγραφή του πίνακα `MedicationRequest` που έχει κωδικό φαρμακευτικής αγωγής (πεδίο `id`) ίσο με την παράμετρο `id`, και επιστρέφει το πεδίο που αφορά την ονομασία του Φαρμάκου (`medication`) σε μορφή json.

Εύρεση στοιχείων Φαρμακευτικής Αγωγής βάσει Επίσκεψης

Το αίτημα **GET** στη διεύθυνση `<serverBaseURL>/encountermedications/:id` χειρίζεται η μέθοδος `getMedicationByEncounter`, που αναζητά από τη βάση την εγγραφή του πίνακα `MedicationRequest` που έχει αναφορά σε Επίσκεψη (πεδίο `context`) ίσο με την παράμετρο `id`, και επιστρέφει την ονομασία του Φαρμάκου (`medication`) και την ημερομηνία χορήγησης, σε φθίνουσα ταξινόμηση όσον αφορά τη ημερομηνία χορήγησης,

Εύρεση Φαρμακευτικής Αγωγής ενός Ασθενή βάσει Κωδικού

Το αίτημα **GET** στη διεύθυνση `<serverBaseURL>/patientmedication/:id` χειρίζεται η μέθοδος `getPatientMedicationRequests`, που αναζητά από τη βάση την εγγραφή του πίνακα `MedicationRequest` που έχει αναφορά σε Patient (πεδίο `subject`) ίσο με την παράμετρο `id`, και επιστρέφει την ονομασία του Φαρμάκου (`medication`) και την ημερομηνία χορήγησης, σε φθίνουσα ταξινόμηση όσον αφορά τη ημερομηνία χορήγησης, σε μορφή json.

Προσθήκη Φαρμακευτικής Αγωγής

Όταν η διεπαφή δεχθεί αίτημα **POST** στη διεύθυνση `<serverBaseURL>/medication` καλείται η μέθοδος `createMedicationRequest`, που δημιουργεί νέα εγγραφή στον πόρο `Medication Request`. Τα στοιχεία για τα φάρμακα συνοδεύουν το αίτημα **POST** και προέρχονται από την υποβολή αντίστοιχης φόρμας.

Διαγραφή Φαρμακευτικής Αγωγής βάσει Κωδικού

Όταν η διεπαφή δεχθεί αίτημα **DELETE** στη διεύθυνση `<serverBaseURL>/medication` καλείται η μέθοδος `deleteMedicationRequest` που διαγράφει εγγραφή του πόρου `Medication Request`, με βάση τον Κωδικό της Φαρμακευτικής Αγωγής (ID), που συνοδεύει την αποστολή του αιτήματος.

Ενημέρωση Φαρμακευτικής Αγωγής βάσει Κωδικού

Όταν η διεπαφή δεχθεί αίτημα **PUT** στη διεύθυνση `<serverBaseURL>/medication` καλείται η μέθοδος `updateMedicationRequest` που ενημερώνει εγγραφή του πόρου `Medication Request`,

που επιλέγεται με βάση τον Κωδικό της Φαρμακευτικής Αγωγής (ID), που συνοδεύει την αποστολή του αιτήματος. Το αίτημα περιέχει επίσης και τις νέες τιμές για τη Φαρμακευτική Αγωγή.

Υποστηριζόμενες Λειτουργίες Διαγνωστικών Εξετάσεων (DiagnosticReports)

Εύρεση στοιχείων Διαγνωστικών Εξετάσεων βάσει Επίσκεψης

Η διεπαφή επιστρέφει τα περιεχόμενα μιας εγγραφής του πόρου DiagnosticReports που έχει αναφορά σε συγκεκριμένο κωδικό επίσκεψης (ID) σε μορφή json όταν σταλεί αίτημα GET στη διεύθυνση <serverBaseURL>/encounterreports/:id

Εύρεση Διαγνωστικών Εξετάσεων ενός Ασθενή βάσει Κωδικού

Η διεπαφή επιστρέφει τα περιεχόμενα μιας εγγραφής του πόρου DiagnosticReports που έχει κάνει Ασθενής (Patient) που έχει συγκεκριμένο κωδικό (ID) σε μορφή json όταν σταλεί αίτημα GET στη διεύθυνση <serverBaseURL>/patientreports/:id

Κεφάλαιο 6: Συμπεράσματα

Το παρόν κεφάλαιο παρουσιάζει τα αποτελέσματα που αναδύονται με το τέλος της εργασίας και προοικονομούνται οι μελλοντικές διαδόσεις της .

Αποτελέσματα

Η ανάπτυξη της εφαρμογής είχε ως κυρίαρχο στοιχείο την υιοθέτηση αυτού του μοντέλου (HL7 FHIR) για την ανταλλαγή δεδομένων ιατρικού χαρακτήρα, ώστε να είναι δυνατή η μετάδοση και λήψη πληροφοριών από άλλα ιατρικά πληροφοριακά συστήματα που ακολουθούν το πρότυπο HL7 FHIR.

Η εφαρμογή δοκιμάστηκε με δημόσια διαθέσιμα δοκιμαστικά δεδομένα που αντλήθηκαν από το διαδίκτυο <https://github.com/fhirbase/fhirbase/blob/master/demo/bundle.ndjson.gz>.

Υποστηρίζει τη λειτουργικότητα που περιγράφηκε στη φάση της ανάλυσης και δόθηκε έμφαση στη γρήγορη άντληση των δεδομένων από τη βάση δεδομένων και την εμφάνιση της πληροφορίας κάνοντας χρήση σύγχρονων τεχνικών υλοποίησης και εμφάνισης υλοποίησης διαδραστικών ιστοσελίδων (jQuery, Bootstrap).

Τα πιο σημαντικά στοιχεία της εργασίας είναι τα εξής:

Αρχιτεκτονική

Η αρχιτεκτονική βασίζεται στο μοντέλο πελάτη-εξυπηρετητή σε περιβάλλον διαδικτύου, καθιστώντας την εφαρμογή εύκολα προσβάσιμη από το φυλλομετρητή του χρήστη-ιατρού.

Χρησιμότητα

Σημαντικό κριτήριο για την αποδοχή της εφαρμογής είναι η φιλικότητά της προς το χρήστη και η δυνατότητα που δίνει να τον καθοδηγεί στις διάφορες διαδικασίες.

Ασφάλεια

Η εφαρμογή φροντίζει για τη διασφάλιση των δεδομένων που χειρίζεται, μια και μόνο ο ιατρός-χρήστης έχει πρόσβαση στα δεδομένα των ιατρικών φακέλων των ασθενών του. Είναι σημαντικό να τονιστεί ότι σε περίπτωση επέκτασης της εφαρμογής και υποστήριξης πρόσβασης από πολλαπλούς χρήστες πρέπει η εφαρμογή να μεριμνά για την αυθεντικοποίηση του χρήστη με χρήση τεχνικών, όπως κωδικούς πρόσβασης, αναγνώριση σε δύο φάσεις, κλπ, και επίσης για την ακεραιότητα και των δεδομένων που μεταδίδονται μέσω δικτύου με χρήση κρυπτογράφησης.

Διαλειτουργικότητα

Η εφαρμογή χρησιμοποιεί διεθνή πρότυπα ανταλλαγής ιατρικών δεδομένων (HL7 FHIR), που την καθιστά συμβατή με άλλα ιατρικά πληροφοριακά συστήματα που βασίζονται σε αυτά

Συμπεράσματα

Τα κύρια συμπεράσματα που γίνονται γνωστά με το τέλος της εργασίας είναι τα εξής:

- Η βάση δεδομένων ενός ιατρικού πληροφοριακού συστήματος που υποστηρίζει το πρότυπο HL7 FHIR, είναι προτιμότερο να καλύπτει όλες τις Οντότητες που ορίζονται στο πρότυπο, ώστε να είναι εφικτό στο μέλλον να ανταλλάσσονται δεδομένα με άλλα ιατρικά πληροφοριακά συστήματα που βασίζονται σε αυτό.
- Η χρήση σύγχρονων τεχνολογιών ανάπτυξης διαδικτυακών εφαρμογών καθιστά την εφαρμογή προσβάσιμη από το φυλλομετρητή του χρήστη-ιατρού, οπότε αναμένεται να υπάρχει γρήγορη εξοικείωση με τη χρήση της
- Η διασφάλιση της ασφάλειας της πληροφορίας που ανταλλάσσει η εφαρμογή μέσω διαδικτύου είναι πολύ σημαντική για την αποδοχή της εφαρμογής στον ευαίσθητο χώρο της υγείας.

Μελλοντικές Επεκτάσεις

Οι ενδεχόμενες μελλοντικές επεκτάσεις της εργασίας έχουν να κάνουν με τις λειτουργίες που θα καταστήσουν την πλατφόρμα ένα πλήρως λειτουργικό ιατρικό πληροφοριακό σύστημα.

Μια επέκταση θα ήταν η ανάπτυξη λειτουργιών επιπλέον χρηστών, εκτός του Ιατρού, μια και η πλατφόρμα σχεδιάστηκε ώστε να μπορεί να υποστηρίζει διαφορετικούς χρήστες (Ασθενή, Ιατρό, Εξεταστικό Κέντρο, Φαρμακείο).

Σε αυτή την περίπτωση θα μπορούσαν να υποστηριχτούν επιπλέον σενάρια χρήσης, όπως:

- Παρακολούθηση από τον ασθενή του ιατρικού του φακέλου
- Ενημερώσεις από τον Υπεύθυνο του Εξεταστικού Κέντρου για τα αποτελέσματα των εξετάσεων που έκανε ο ασθενής
- Ενημερώσεις από τον Φαρμακοποιό για τις συνταγές που έχουν εκτελεστεί

Μια άλλη επέκταση θα μπορούσε να περιλαμβάνει την υποστήριξη Νοσοκομείου, όπου θα χρειαστεί να ληφθεί μέριμνα για:

- υποδοχή του ασθενή
- διαλογή του από τον εφημερεύοντα ιατρό,
- την παραπομπή του ασθενή σε ιατρό ειδικότητας
- παραπομπή σε εξετάσεις
- εισαγωγή σε κλινική στο νοσοκομείο

Παράρτημα Α – Δεδομένα FHIR

Ενδεικτικά Παραδείγματα Περιεχομένου Πόρων FHIR

Παρακάτω παρατίθενται ενδεικτικά παραδείγματα πληροφοριών που περιέχονται σε πόρους FHIR που χρησιμοποιούνται στην Εφαρμογή, σε μορφή JSON.

Παράδειγμα πληροφορίας σε πόρο *Patient*:

```
{
  "id": "d3af67c9-0c02-45f2-bc91-fea45af3ee83",
  "meta": {
    "profile": [
      "http://standardhealthrecord.org/fhir/StructureDefinition/shr-entity-Patient"
    ]
  },
  "name": [
    {
      "use": "official",
      "given": [
        "Abram53"
      ],
      "family": "Hickle134",
      "prefix": [
        "Mr."
      ]
    }
  ],
  "text": {
    "div": "<div xmlns=\\"http://www.w3.org/1999/xhtml\\">Generated by <a href=\\"https://github.com/synthetichealth/synthea\\">Synthea</a>.Version identifier: v2.0.0-33-g3ab9839e\\n . Person seed: -7948439064728439692 Population seed: 1534165146158</div>",
    "status": "generated"
  },
  "gender": "male",
  "address": [
    {
      "city": "Brockton",
      "line": [
        "730 Schoen Center Apt 8"
      ]
    }
  ]
}
```

```

],
"state": "Massachusetts",
"country": "US",
"extension": [
  {
    "url": "http://hl7.org/fhir/StructureDefinition/geolocation",
    "extension": [
      {
        "url": "latitude",
        "valueDecimal": -71.024638
      },
      {
        "url": "longitude",
        "valueDecimal": 42.082543
      }
    ]
  }
]
},
"telecom": [
  {
    "use": "home",
    "value": "555-696-1511",
    "system": "phone"
  }
],
"birthDate": "1955-06-27",
"extension": [
  {
    "url": "http://hl7.org/fhir/us/core/StructureDefinition/us-core-race",
    "extension": [
      {
        "url": "ombCategory",
        "valueCoding": {
          "code": "2106-3",
          "system": "urn:oid:2.16.840.1.113883.6.238",
          "display": "White"
        }
      }
    ]
  }
]

```

```

    "url": "text",
    "valueString": "White"
  }
]
},
{
  "url": "http://hl7.org/fhir/us/core/StructureDefinition/us-core-ethnicity",
  "extension": [
    {
      "url": "ombCategory",
      "valueCoding": {
        "code": "2186-5",
        "system": "urn:oid:2.16.840.1.113883.6.238",
        "display": "Not Hispanic or Latino"
      }
    },
    {
      "url": "text",
      "valueString": "Not Hispanic or Latino"
    }
  ]
},
{
  "url": "http://hl7.org/fhir/StructureDefinition/patient-mothersMaidenName",
  "valueString": "Sharolyn456 Harris789"
},
{
  "url": "http://hl7.org/fhir/us/core/StructureDefinition/us-core-birthsex",
  "valueCode": "M"
},
{
  "url": "http://hl7.org/fhir/StructureDefinition/birthPlace",
  "valueAddress": {
    "city": "Boston",
    "state": "Massachusetts",
    "country": "US"
  }
},
{
  "url": "http://standardhealthrecord.org/fhir/StructureDefinition/shr-actor-FictionalPerson-extension",
  "valueBoolean": true
}

```

```

},
{
  "url": "http://standardhealthrecord.org/fhir/StructureDefinition/shr-entity-FathersName-extension",
  "valueHumanName": {
    "text": "Samual455 Hickle134"
  }
},
{
  "url": "http://standardhealthrecord.org/fhir/StructureDefinition/shr-demographics-SocialSecurityNumber-extension",
  "valueString": "999-81-4006"
},
{
  "url": "http://standardhealthrecord.org/fhir/StructureDefinition/shr-entity-Person-extension",
  "valueReference": {
    "reference": "Basic/bad528f4-fa8a-4f5b-a478-1b630be88847"
  }
},
{
  "url": "http://synthetichealth.github.io/synthea/disability-adjusted-life-years",
  "valueDecimal": 0
},
{
  "url": "http://synthetichealth.github.io/synthea/quality-adjusted-life-years",
  "valueDecimal": 62
}
],
"identifier": [
  {
    "value": "803f5907-5427-4930-a093-1a95190de7fd",
    "system": "https://github.com/synthetichealth/synthea"
  },
  {
    "type": {
      "text": "Medical Record Number",
      "coding": [
        {
          "code": "MR",
          "system": "http://hl7.org/fhir/v2/0203",
          "display": "Medical Record Number"
        }
      ]
    }
  }
]

```



```
},
"value": "803f5907-5427-4930-a093-1a95190de7fd",
"system": "http://hospital.smarthealthit.org"
},
{
"type": {
"text": "Social Security Number",
"coding": [
{
"code": "SB",
"system": "http://hl7.org/fhir/identifier-type",
"display": "Social Security Number"
}
]
},
"value": "999-81-4006",
"system": "http://hl7.org/fhir/sid/us-ssn"
},
{
"type": {
"text": "Driver's License",
"coding": [
{
"code": "DL",
"system": "http://hl7.org/fhir/v2/0203",
"display": "Driver's License"
}
]
},
"value": "S99988684",
"system": "urn:oid:2.16.840.1.113883.4.3.25"
},
{
"type": {
"text": "Passport Number",
"coding": [
{
"code": "PPN",
"system": "http://hl7.org/fhir/v2/0203",
"display": "Passport Number"
}
]
}
```

```

    ]
  },
  "value": "X17148555X",
  "system": "http://standardhealthrecord.org/fhir/StructureDefinition/passportNumber"
}
],
"resourceType": "Patient",
"communication": [
{
  "language": {
    "text": "English",
    "coding": [
      {
        "code": "en-US",
        "system": "urn:ietf:bcp:47",
        "display": "English"
      }
    ]
  }
}
],
"maritalStatus": {
  "text": "S",
  "coding": [
    {
      "code": "S",
      "system": "http://hl7.org/fhir/v3/MaritalStatus",
      "display": "S"
    }
  ]
},
"multipleBirth": {
  "boolean": false
}
}

```

Παράδειγμα πληροφορίας σε πόρο του *Encounter*:

```

{
  "id": "0002aaeb-31ac-42fe-93a9-3a3db68d1636",
  "meta": {
    "profile": [

```

```

    "http://standardhealthrecord.org/fhir/StructureDefinition/shr-encounter-EncounterPerformed"
  ]
},
"type": [
  {
    "text": "Encounter for symptom",
    "coding": [
      {
        "code": "185345009",
        "system": "http://snomed.info/sct",
        "display": "Encounter for symptom"
      }
    ]
  }
],
"class": {
  "code": "ambulatory"
},
"period": {
  "end": "2017-03-28T02:23:07+03:00",
  "start": "2017-03-28T02:08:07+03:00"
},
"reason": [
  {
    "coding": [
      {
        "code": "444814009",
        "system": "http://snomed.info/sct",
        "display": "Viral sinusitis (disorder)"
      }
    ]
  }
],
"status": "finished",
"subject": {
  "id": "695e31c4-37cc-4180-b1f2-794d8d713956",
  "resourceType": "Patient"
},
"extension": [
  {
    "url": "http://standardhealthrecord.org/fhir/StructureDefinition/shr-action-PerformedContext-extension",

```

```

    "extension": [
      {
        "url": "http://standardhealthrecord.org/fhir/StructureDefinition/shr-action-Status-extension",
        "valueCode": "finished"
      }
    ]
  },
  "resourceType": "Encounter",
  "serviceProvider": {
    "id": "2689eaae-7b23-4fb7-8bcc-0419d839e9e0",
    "resourceType": "Organization"
  }
}

```

Παράδειγμα πληροφορίας σε πόρο *Diagnostic Report*:

```

{
  "id": "00020372-1092-4c6a-9bdf-46e36d6ce33c",
  "code": {
    "text": "Basic Metabolic Panel",
    "coding": [
      {
        "code": "51990-0",
        "system": "http://loinc.org",
        "display": "Basic Metabolic Panel"
      }
    ]
  },
  "issued": "2014-04-06T13:05:58.375+04:00",
  "result": [
    {
      "id": "de137dd5-7d4f-4b82-b773-0dd321c121ec",
      "display": "Glucose",
      "resourceType": "Observation"
    },
    {
      "id": "a56ba8d2-77b9-4a0a-8f88-764e4f72faac",
      "display": "Urea Nitrogen",
      "resourceType": "Observation"
    }
  ]
}

```

```
"id": "c7654644-6974-40db-b417-64832441d71a",
"display": "Creatinine",
"resourceType": "Observation"
},
{
  "id": "813da97b-c09c-498e-8289-1ac10c96004a",
  "display": "Calcium",
  "resourceType": "Observation"
},
{
  "id": "213f8ead-5d64-43d0-9a18-ac15aaeddb45",
  "display": "Sodium",
  "resourceType": "Observation"
},
{
  "id": "21eea7b0-7ceb-47c9-b0c7-f695086117a5",
  "display": "Potassium",
  "resourceType": "Observation"
},
{
  "id": "201b89f8-22e4-4406-94f0-85b086f00969",
  "display": "Chloride",
  "resourceType": "Observation"
},
{
  "id": "b6ea679e-ae3-4f67-a393-a478ae2239a0",
  "display": "Carbon Dioxide",
  "resourceType": "Observation"
}
],
"status": "final",
"context": {
  "id": "4124dbfb-254d-4f44-8c4b-67d07863f2a5",
  "resourceType": "Encounter"
},
"subject": {
  "id": "e6f18934-8ed7-47fb-ace2-e89b7de2a3a4",
  "resourceType": "Patient"
},
"effective": {
  "dateTime": "2014-04-06T13:05:58+04:00"
```

```
},  
"resourceType": "DiagnosticReport"  
}
```

Παράδειγμα πληροφορίας σε πόρο **Observation**:

```
{  
  "id": "de137dd5-7d4f-4b82-b773-0dd321c121ec",  
  "code": {  
    "text": "Glucose",  
    "coding": [  
      {  
        "code": "2339-0",  
        "system": "http://loinc.org",  
        "display": "Glucose"  
      }  
    ]  
  },  
  "meta": {  
    "profile": [  
      "http://standardhealthrecord.org/fhir/StructureDefinition/shr-finding-Observation"  
    ]  
  },  
  "value": {  
    "Quantity": {  
      "code": "mg/dL",  
      "unit": "mg/dL",  
      "value": 82.94847083070687,  
      "system": "http://unitsofmeasure.org"  
    }  
  },  
  "issued": "2014-04-06T13:05:58.375+04:00",  
  "status": "final",  
  "context": {  
    "id": "4124dbfb-254d-4f44-8c4b-67d07863f2a5",  
    "resourceType": "Encounter"  
  },  
  "subject": {  
    "id": "e6f18934-8ed7-47fb-ace2-e89b7de2a3a4",  
    "resourceType": "Patient"  
  },  
}
```

```

"category": [
  {
    "coding": [
      {
        "code": "laboratory",
        "system": "http://hl7.org/fhir/observation-category",
        "display": "laboratory"
      }
    ]
  }
],
"effective": {
  "dateTime": "2014-04-06T13:05:58+04:00"
},
"resourceType": "Observation"
}

```

Παράδειγμα πληροφορίας σε πόρο *Medication Request*:

```

{
  "id": "b5c9878c-6d92-4d22-a28c-3277b42f6a51",
  "meta": {
    "profile": [
      "http://standardhealthrecord.org/fhir/StructureDefinition/shr-medication-MedicationRequested"
    ]
  },
  "intent": "order",
  "status": "active",
  "context": {
    "id": "73e0ace8-7f56-46b7-820d-79868f35c970",
    "resourceType": "Encounter"
  },
  "subject": {
    "id": "096724e0-4f46-4af9-8d25-2e5d6ec47526",
    "resourceType": "Patient"
  },
  "extension": [
    {
      "url": "http://standardhealthrecord.org/fhir/StructureDefinition/shr-base-ActionCode-extension",
      "valueCodeableConcept": {
        "text": "Prescription of drug (procedure)",

```

```

"coding": [
  {
    "code": "33633005",
    "system": "http://snomed.info/sct",
    "display": "Prescription of drug (procedure)"
  }
]
},
{
  "url": "http://standardhealthrecord.org/fhir/StructureDefinition/shr-action-RequestedContext-extension",
  "extension": [
    {
      "url": "http://standardhealthrecord.org/fhir/StructureDefinition/shr-action-Status-extension",
      "valueCode": "completed"
    },
    {
      "url": "http://standardhealthrecord.org/fhir/StructureDefinition/shr-action-RequestIntent-extension",
      "valueCode": "original-order"
    }
  ]
},
"authoredOn": "1995-10-25T17:36:01+03:00",
"medication": {
  "CodeableConcept": {
    "text": "Tacrine 10 MG Oral Capsule",
    "coding": [
      {
        "code": "313185",
        "system": "http://www.nlm.nih.gov/research/umls/rxnorm",
        "display": "Tacrine 10 MG Oral Capsule"
      }
    ]
  }
},
"resourceType": "MedicationRequest",
"reasonReference": [
  {
    "id": "adb5617f-3c94-4c62-8f77-21db8a6b68ee",
    "resourceType": "Condition"
  }
]

```



```
}  
]  
}
```

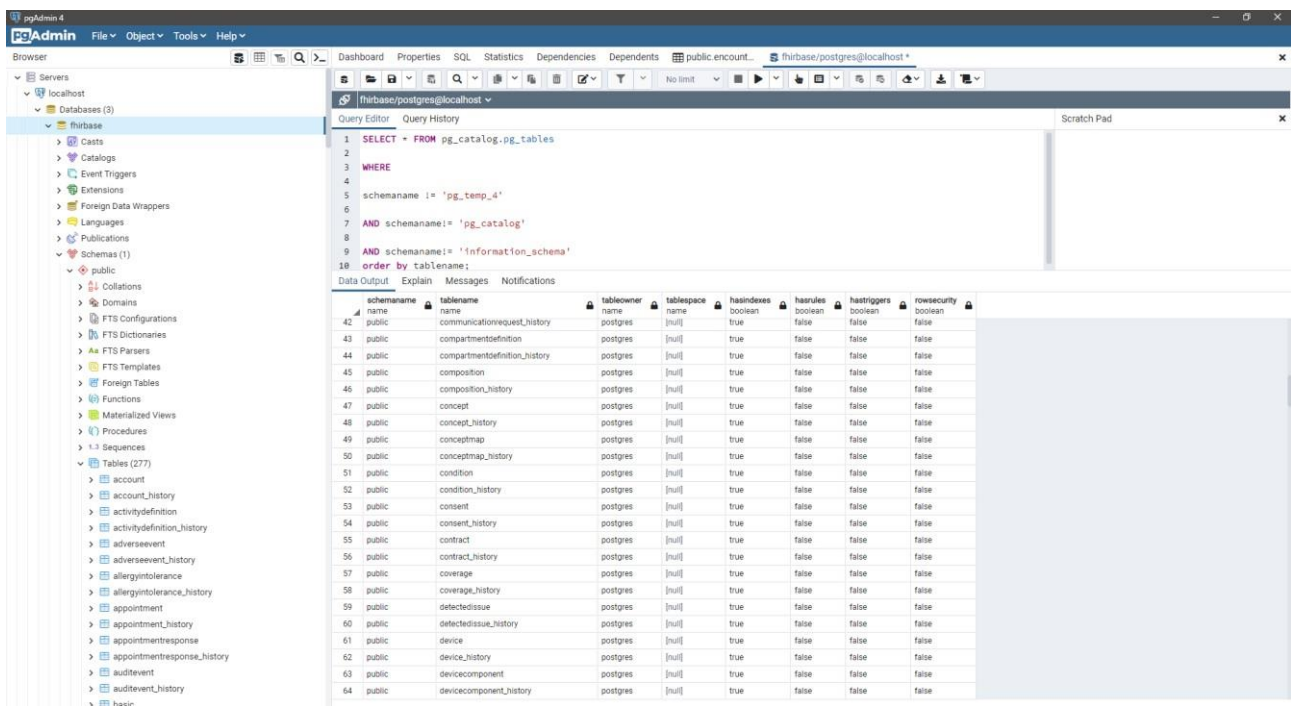
Παράρτημα Β – Εγκατάσταση Εργαλείων

Παρακάτω ακολουθούν τα βήματα εγκατάστασης των εργαλείων που χρησιμοποιήθηκαν στην ανάπτυξη της Εφαρμογής

Εγκατάσταση του Warrp

Επιλέχθηκε η έκδοση 7.3.33 από το site <https://bitnami.com/stack/warrp/installer> και έγινε εγκατάσταση. Το πακέτο Warrp είναι ελεύθερο λογισμικό και είναι διαθέσιμο σε περιβάλλον Windows, Linux και Mac.

Το πακέτο μας δίνει τη δυνατότητα διαχείρισης των servers Apache και Postgres, που περιλαμβάνει, με το εργαλείο **manager**. Επίσης διαθέτει δύο εργαλεία με τα οποία μπορούμε να εκφράσουμε ερωτήματα στη βάση δεδομένων PostgreSQL: το εργαλείο **pgAdmin4.exe** που λειτουργεί σε γραφικό περιβάλλον και το εργαλείο **psql.exe** που λειτουργεί σε περιβάλλον εντολών.



Εικόνα 20: Χρήση του εργαλείου pgAdmin4

Εγκατάσταση της FHIRBase

Το πακέτο fhirbase είναι διαθέσιμο από τη σελίδα <https://www.health-samurai.io/fhirbase-downloads>. Επίσης

Για την εγκατάσταση βάσης δεδομένων με τα ενδεικτικά δεδομένα έγιναν τα ακόλουθα βήματα:

- Σύνδεση στην postgres με την εντολή `psql -h localhost -p 5432 -U postgres -W postgres`
- Δημιουργία τη βάσης **fhirbase** με την εντολή `CREATE DATABASE fhirbase;`
- Εκτέλεση της εντολή `fhirbase.exe --host localhost -p 5432 -d fhirbase -U postgres init`, που δημιουργεί όλους τους πίνακες FHIR στη βάση δεδομένων fhirbase
- Κατέβασμα των ελεύθερα διαθέσιμων ενδεικτικών δεδομένων της βάσης από τη σελίδα <https://github.com/fhirbase/fhirbase/blob/master/demo/bundle.ndjson.gzip>
- Φόρτωση των δεδομένων στη βάση fhirbase, με την εντολή `fhirbase.exe --host localhost -p 5432 -d fhirbase -U postgres --fhir=3.3.0 load -m insert bundle.ndjson.gzip`

Node.js

Μετά την εγκατάσταση του node.js, χρειάζεται να εγκατασταθούν επιπλέον τα πακέτα: Express.js, για την ανάπτυξη της node-postgres (pg) για τη σύνδεση με την PostgreSQL. με την εντολή `npm i express pg`

Στη συνέχεια η σύνδεση στη βάση δεδομένων fhirbase που φιλοξενείται στον ίδιο server (localhost) γίνεται ως εξής:

```
const Pool = require('pg').Pool
const pool = new Pool({
  user: 'postgres',
  host: 'localhost',
  database: 'fhirbase',
  password: '123456',
  port: 5432,
})
```

Παράρτημα Γ – Κώδικας Εφαρμογής

Παρακάτω παρατίθεται ο κώδικας της Εφαρμογής

Κώδικας Ιστοσελίδων PHP

Σελίδα Επιλογής Ασθενή (selectPatient.php)

```
<?php
include('header.php');
unset($_SESSION['patient']);
?>
<div class="container-fluid bg-3 text-center">
  <h3>Search Patient by Family Name</h3>
  <div class="container">
    <table class="table table-bordered table-striped">
      <tr>
        <td>
          Enter (part of) Patient Family Name <br>and press Search:
        </td>
        <td>
          <input type="text" name="given" id="given">
        </td>
        <td>
          <button id="Search" name="search">Search</button>
        </td>
      </tr>
    </table>
  </div>
  <div id="patient">
  </div>
</div>
<!-- Script to use ajax() method to
  add text content -->
<script>
  $(document).ready(function() {

    $('#patient-info').css("visibility", "hidden");
```

```

$('#select-patient').addClass("active");
$("#Search").click(function() {
    var given = $("#given").val();
    $.ajax({
        url: "http://localhost:3000/patients/name/" + given,
        async: false,
        success: function(result) {
            var text = "<table class='table table-bordered table-striped'><tr><th>Family
Name</th><th>Given Name</th><th></th></tr>";
            for (i in result) {
                var name = result[i].given + " " + result[i].family
                var patient_id = result[i].id
                text += "<tr><td>" + result[i].family + "</td><td>" + result[i].given + " </td>";
                text += "<td><form method='post' action='showPatient.php'>";
                text += "<input type='hidden' name='id' value='" + patient_id + "'>";
                text += "<input type='submit' value='Select'> </form></td></tr>";
            }
            text += "</table>";
            $('#patient').html(text);
        }
    });
});
</script>
<?php include('footer.php'); ?>

```

Σελίδα παρουσίασης Ιατρικού Φακέλου Ασθενή (showPatient.php).

Περιλαμβάνει την παρουσίαση σε καρτέλες τις Επισκέψεις, τα Αποτελέσματα Εργαστηριακών εξετάσεων και τις Φαρμακευτικές αγωγές του Ασθενή.

```
<?php
include('header.php');

// Check if session stores patient
if (!isset($_SESSION["patient"])) {
    // Check if the patient is not selected
    if (!isset($_POST["id"])) {
        header("location: selectPatient.php");
        exit;
    } else {
        $_SESSION["patient"] = $_POST["id"];
    }
}

if (isset($_SESSION["encounter"])) {
    unset($_SESSION["encounter"]);
}

?>

<div class="container-fluid bg-3 text-center">
    <h3>Patient Info </h3>

    <br>

    <div class="panel panel-primary">
        <div class="panel-heading">
            <h4 id="patient-name"></h4>
        </div>

        <div class="panel-body">
```

```

    <div><button type="button" style="float:right; margin:20px;padding:5px;" class="btn
btn-success btn-lg" class="btn btn-primary btn-lg" data-toggle="modal" data-
target="#mailModal" data-whatever="@getbootstrap">Send Record</button></div>
    <div id="message"></div>

    <ul class="nav nav-tabs">
        <li><a data-toggle="tab" href="#info">Personal Info</a></li>
        <li class="active"><a data-toggle="tab" href="#en">Encounters</a></li>
        <li><a data-toggle="tab" href="#diag">Diagnostic Reports</a></li>
        <li><a data-toggle="tab" href="#med">Medication Requests</a></li>

    </ul>
</div>
<div class="tab-content">

    <div id="info" class="tab-pane fade">

</div>
    <div id="en" class="tab-pane fade in active">

        <div> <button id='add' style="float:right; margin:20px;padding:5px;" class="btn btn-
primary btn-lg">ADD ENCOUNTER</button></div>
        <div id="encounters">

</div>
</div>
    <div id="diag" class="tab-pane fade">

</div>
    <div id="med" class="tab-pane fade">

</div>
</div>
</div>
</div>

<script>
    $(document).ready(function() {

```

```

var value = '<?php echo $_SESSION["patient"] ?>';
console.log(value);

$.ajax({
  url: "http://localhost:3000/patient/" + value,
  async: false,

  success: function(result) {

    var text = "";

    text += "<br><table id='dataTable' class='table table-bordered table-striped'>";
    text += "<tbody>";

    for (c in result) {

      var patient = result[c];
      var name = patient["given"] + " " + patient["family"];

      $("#patient-name").text(name);
      text += "<tr>";
      text += "<td>Name</td>";
      text += "<td>" + patient["given"] + " " + patient["family"] + "</td>";
      text += "</tr>";
      text += "<tr>";
      text += "<td>SSN</td>";
      text += "<td>" + patient["ssn"] + "</td>";
      text += "</tr>";
      text += "<tr>";
      text += "<td>Birth Date</td>";
      text += "<td>" + patient["birthdate"] + " </td>";
      text += "</tr>";
      text += "<tr>";
      text += "<td>Phone</td>";
      text += "<td>" + patient["phone"] + "</td>";
      text += "</tr>";
    }
  }
});

```



```

        text += "<tr>";
        text += "<td>Address</td>";
        text += "<td>" + patient["street"] + " " + patient["city"] + "</td>";
        text += "</tr>";

    }
    text += "</table>";
    $('#info').html(text);
}
});

$.ajax({
    url: "http://localhost:3000/patientencounters/" + value,
    async: false,

    success: function(result) {
        var text = "";
        text += "<br><table id='dataTable' class='table table-bordered table-striped'>";
        text += " <thead>";
        text +=
"<tr><th>#</th><th>Date</th><th>Symptom</th><th>Medication</th><th></th></tr>";
        text += " </thead><tbody>";

        for (i in result) {
            var en = result[i];
            var id = en["id"];
            text += "<tr>";
            var jss = parseInt(i) + 1;
            text += "<td>" + jss + "</td>";

            text += "<td>" + en["day"] + "-" + en["month"] + "-" + en["year"] + "</td>";
            if (en["symptom"] != null) text += "<td>" + en["symptom"] + "</td>";
            else text += "<td></td>";
            text += "<td>";

            $.ajax({
                url: "http://localhost:3000/encountermedications/" + id,
                async: false,

                success: function(result2) {
                    var text2 = "";

```

```

        for (c in result2) {

            var name = result2[c].medication;
            text2 += name + "<br>";

        }
        text += text2;
    }
});
text += "</td>";

text += "<td>";

text += "<form method='post' action='viewEncounter.php'>";
text += "<input type='submit' class='btn btn-info' value='Show Details'>";
text += "<input type='hidden' name='id' value='" + id + "'>";
text += "</form>";

text += "</td>";
text += "</tr>";

}
text += "</table>";
$('#encounters').html(text);

}
});

$.ajax({
    url: "http://localhost:3000/patientreports/" + value,
    async: false,

    success: function(result) {

        var text = "";
        text += "<br><table id='dataTable' class='table table-bordered table-striped'>";

```

```

text += " <thead>";
text += "<tr><th>#</th><th>Date</th><th>Type</th><th>Results</th></tr>";
text += "</thead><tbody>";
for (c in result) {

    var diag = result[c];

    text += "<tr>";
    var jss = parseInt(c) + 1;
    text += "<td>" + jss + "</td>";
    text += "<td>" + diag["day"] + "-" + diag["month"] + "-" + diag["year"] + "</td>";
    text += "<td>" + diag["text"] + "</td>";
    text += "<td>";
    var reports = diag["results"];
    for (i in reports) {
        var observation = reports[i];
        $.ajax({
            url: "http://localhost:3000/observation/" + observation["id"],
            async: false,

            success: function(result2) {
                var text2 = "";

                for (i in result2) {

                    var name = result2[i].name;
                    var value2 = result2[i].value;
                    var unit = result2[i].unit;
                    var patient_id = result2[i].id
                    text2 += name + ":" + value2 + " " + unit + "<br>";

                }
                text += text2;
            }
        });
    }
    text += "</td>";
}

```

```

        text += "</tr>";

    }
    text += "</table>";
    $('#diag').html(text);
}

});

$.ajax({
    url: "http://localhost:3000/patientmedication/" + value,
    async: false,

    success: function(result) {
        var text = "";
        text += "<br><table id='dataTable' class='table table-bordered table-striped'>";
        text += " <thead>";
        text += "<tr><th>#</th><th>Date</th><th>Medication</th></tr>";
        text += " </thead><tbody>";

        for (i in result) {
            var med = result[i];
            text += "<tr>";
            var jss = parseInt(i) + 1;
            text += "<td>" + jss + "</td>";
            text += "<td>" + med["day"] + "-" + med["month"] + "-" + med["year"] +
"</td>";
            text += "<td>" + med["medication"] + "</td>";

            text += "</tr>";

        }
        text += "</table>";
        $('#med').html(text);
    }
}

```

```

});

$("#add").click(function() {
    window.location.href = 'createEncounter.php';
});

$("#sendMailButton").click(function() {
    var recipient = $("#recipient-name").val();
    $.ajax('http://localhost:3000/patient/send/' + '<?=$_SESSION["patient"] ?>', {
        type: 'POST', // http method
        dataType: 'json',
        data: {
            'recipient': recipient
        }, // data to submit
        success: function(data, status, xhr) {

            alert(data);
        },
        error: function(jqXHR, textStatus, errorMessage) {

            alert('Error: ' + errorMessage)
        }
    });
    $('#mailModal').modal('toggle');
});

});
</script>
<div class="modal fade" id="mailModal" tabindex="-1" role="dialog" aria-
labelledby="exampleModalLabel" aria-hidden="true">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="exampleModalLabel">Send Patient Record</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&times;</span></button>
            </div>
            <div class="modal-body">

```

```

        <form id="mailForm" action="http://localhost:3000/patient/send/<?php
echo $_SESSION["patient"] ?>">
        <div class="form-group">
            <label for="recipient-name" class="col-form-label">Recipient:</label>
            <input type="text" class="form-control" id="recipient-name">
            <input type="hidden" name="patient_id" value="<?php
echo $_SESSION["patient"] ?>" </div>

        </form>
    </div>
    <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>
        <button type="button" id="sendMailButton" class="btn btn-primary">Send
message</button>
    </div>
</div>
</div>
</div>
</div>
</body>

</html>

```

Σελίδα προσθήκης εγγραφής ιατρικής επίσκεψης (createEncounter.php)

Προσθέτει ιατρική επίσκεψη δίνοντας την επιλογή **συμπτώματος/πάθησης**

```
<?php
include('header.php');

// Check if the patient is not selected
if (!isset($_SESSION["patient"])) {
    header("location: selectPatient.php");
    exit;
}

if (isset($_POST["reason_code"])) {
    // include("connection.php");
    date_default_timezone_set('Europe/Athens');
    $date = date('Y-m-d h:i:s', time());

    do {
        $rand1 = str_pad(dechex(rand(0x00000000, 0xFFFFFFFF)), 8, 0, STR_PAD_LEFT);
        $rand2 = str_pad(dechex(rand(0x0000, 0xFFFF)), 4, 0, STR_PAD_LEFT);
        $rand3 = str_pad(dechex(rand(0x0000, 0xFFFF)), 4, 0, STR_PAD_LEFT);
        $rand4 = str_pad(dechex(rand(0x0000, 0xFFFF)), 4, 0, STR_PAD_LEFT);
        $rand5 = str_pad(dechex(rand(0x000000000000, 0xFFFFFFFFFFFF)), 12, 0, STR_PAD_LEFT);
        $sid = $rand1 . "-" . $rand2 . "-" . $rand3 . "-" . $rand4 . "-" . $rand5;
    }

?>
<script>
$(document).ready(function() {
    $.ajax({
        url: "http://localhost:3000/encounter/" + "<?=$id ?>",
        async: false,

        success: function(result) {
            if (result.length) {
                <?php $found = 1; ?>
            } else {
                <?php $found = 0; ?>
            }
        }
    })
}
```

```

    });
    });
</script>
<?php
$json = '{ "id": "'. $id. '", "meta": { "profile": [ "http://standardhealthrecord.org/fhir/StructureDefinition/shr-medication-MedicationRequested" ] }, "intent": "order", "status": "active", "context": { "id": "._$SESSION[\"encounter\"].", "resourceType": "Encounter" }, "subject": { "id": "._$SESSION[\"patient\"].", "resourceType": "Patient" }, "extension": [ { "url": "http://standardhealthrecord.org/fhir/StructureDefinition/shr-base-ActionCode-extension", "valueCodeableConcept": { "text": "Prescription of drug (procedure)", "coding": [ { "code": "33633005", "system": "http://snomed.info/sct", "display": "Prescription of drug (procedure)" } ] } }, { "url": "http://standardhealthrecord.org/fhir/StructureDefinition/shr-action-RequestedContext-extension", "extension": [ { "url": "http://standardhealthrecord.org/fhir/StructureDefinition/shr-action-Status-extension", "valueCode": "completed" }, { "url": "http://standardhealthrecord.org/fhir/StructureDefinition/shr-action-RequestIntent-extension", "valueCode": "original-order" } ] } ], "authoredOn": "._$date.", "medication": { "CodeableConcept": { "text": "._$POST[\"med\"].", "coding": [ { "code": "", "system": "http://www.nlm.nih.gov/research/umls/rxnorm", "display": "._$POST[\"med\"]." } ] } }, "resourceType": "MedicationRequest" }';

} while ($found > 0);
?>
<script>
$(document).ready(function() {
$.ajax('http://localhost:3000/encounters', {
type: 'POST', // http method
dataType: 'json',
data: {
'encounter_id': '<?= $id ?>',
'json': '<?= $json ?>'
}, // data to submit
success: function(data, status, xhr) {
$('p').append('status: ' + status + ', data: ' + data);
<?php
$_SESSION['encounter'] = $id;
?>
window.location.href = 'showEncounter.php';

},
error: function(jqXHR, textStatus, errorMessage) {

```



```

    $('p').append('Error: ' + errorMessage);
  }
});

});
</script>
<p></p>
<?php
  exit;
} else {

?>

<div class="container-fluid bg-3 text-center">
  <h3>Create New Encounter</h3>

  <div class="container">
    <form method='POST'>
      <table class='table table-bordered table-striped'>
        <tr>
          <td>
            Patient Name
          </td>
          <td id="patient-name">

          </td>
        </tr>
        <tr>
          <td>
            Reason:
          </td>
          <td>
            <div id="reason"></div>

          </td>
        </tr>
        <tr>
          <td>
            <input type='submit'>

```

```

        </td>
        <td>

        <input type="hidden" name="reason_value" id="reason_value">
    </td>

    </tr>
</table>
</form>
</div>
<div id="patient">
</div>
</div>
<!-- Script to use ajax() method to
    add text content -->
<script>
$(document).ready(function() {
    var patient = '<?php echo $_SESSION["patient"] ?>';

    $.ajax({
        url: "http://localhost:3000/patient/" + patient,
        async: false,
        success: function(result) {
            for (c in result) {
                var patient = result[c];
                var name = patient["given"] + " " + patient["family"];

                $("#patient-name").text(name);
            }
        }
    });
    $.ajax({

        url: "http://localhost:3000/reasons",
        async: false,
        success: function(result) {
            var text = "<select id='reason_code' name='reason_code'><option
value='00000'>NONE</option>";
            for (i in result) {

                if (result[i].reason != null)

```

```
        text += "<option value=\"" + result[i].code + "\"" + result[i].reason + "</option>";
    }
    text += "</select>";
    $('#reason').html(text);
}
});
$('#reason_code').on('change', function() {
    var text = $('#reason_code option:selected').text();
    $('#reason_value').val(text);
});
});
</script>
<?php
}
include('footer.php'); ?>
```

Σελίδα προβολής νέας ιατρικής επίσκεψης (showEncounter.php)

Δίνει τη δυνατότητα προσθήκης/ενημέρωσης/διαγραφής φαρμακευτικής αγωγής σε ιατρική επίσκεψη.

```
<?php
include("header.php");

// Check if the patient is not selected
if (!isset($_SESSION["encounter"])) {
    header("location: selectPatient.php");
    exit;
}

?>

<div class="container-fluid bg-3 text-center">
    <h2>Patient New Encounter </h2>

    <br>
    <table id='dataTable' class='table table-bordered table-striped'>
        <tbody>
            <tr>
                <td>
                    Patient Name
                </td>
                <td id="patient-name">

            </td>
        </tr>
            <tr>
                <td>
                    Date
                </td>
                <td id="date">

            </td>
        </tr>
            <tr>
                <td>
                    Symptom
                </td>
```

```

        <td id="symptom">

        </td>
    </tr>
</tbody>
</table>

<div class="panel panel-primary">
    <div class="panel-heading">
        <h3>Medication Requests</h3>

    </div>

    <div class="panel-body">

        <div id="medication">
            <div><button id='add' style="float:right" class="btn btn-primary"><i class="fa fa-plus-circle"
style="font-size:16px"></i> <span style="font-size:16px"> Add Medication Request</span></but-
ton></div>
            <br>
            <div id="message"></div>
            <div id="med" class="tab-pane fade in active">

                </div>
            </div>

        </div>
    </div>

</div>
</div>

<script>
    function confirm_delete() {

        var retVal = confirm("Please confirm deletion of Medication Request");

        if (retVal == true) {
            console.log("CONFIRMED = TRUE");
            return true;
        } else {
            console.log("CONFIRMED = FALSE");
            return false;
        }
    }

```

```

}

$(document).ready(function() {

    var patient = '<?php echo $_SESSION["patient"] ?>';

    $.ajax({
        url: "http://localhost:3000/patient/" + patient,
        async: false,

        success: function(result) {

            for (c in result) {

                var patient = result[c];
                var name = patient["given"] + " " + patient["family"];

                $("#patient-name").text(name);
            }
        }
    });

    var value = '<?php echo $_SESSION["encounter"] ?>';

    $.ajax({
        url: "http://localhost:3000/encounter/" + value,
        async: false,

        success: function(result) {

            for (c in result) {

                var encounter = result[c];
                var name = encounter["symptom"];

                var date = encounter["day"] + "-" + encounter["month"] + "-" + encounter["year"];

                $("#symptom").text(name);
                $("#date").text(date);
            }
        }
    });
}

```

```

    }
  });

$.ajax({
  url: "http://localhost:3000/encountermedications/" + value,
  async: false,

  success: function(result) {

    var text = "";

    if (result.length) {

      text += "<br><table id='dataTable' class='table table-bordered table-striped'>";
      text += " <thead>";
      text += "<tr><th>#</th><th>Name</th><th></th><th></th></tr>";
      text += " </thead><tbody>";

      for (c in result) {

        var medication = result[c];
        text += "<tr>";
        var jss = parseInt(c) + 1;
        text += "<td>" + jss + "</td>";
        // text += "<td>Name</td>";
        var med_id = medication["id"];
        text += "<td>" + medication["medication"] + "</td>";

        text += "<td>";
        text += "<form method='post' action='updateMedication.php'>";
        text += "<input type='submit' class='btn btn-success' name='update' value='Up-";
date'>";
        text += "<input type='hidden' value=" + med_id + " name='id'>";
        text += "</form>";
        text += "</td>";

        text += "<td>";

```

```

        text += "<form method='post' action='deleteMedication.php' onsubmit='return confirm_delete()'>";
        text += "<button type='submit' name='delete_button' value='delete' class='btn btn-danger' style='margin-right: 15px;' >";
        text += "<i class='fa fa-trash' style='font-size:16px;font-weight:bold;'></i>";
        text += "<span style='padding-left:10px'>Delete</span>";
        text += "</button>";
        // text += "<input type='submit' class='btn btn-danger' name='delete' value='Delete'>";
        text += "<input type='hidden' value=" + med_id + " name='id'>";
        text += "</form>";
        text += "</td>";

        text += "</tr>";

    }
    text += "</table>";
} else {
    text = "<h5>No Medication Requests for this Encounter</h5>";
}
$('#med').html(text);
}
});

$.ajax({
    url: "http://localhost:3000/encounterreports/" + value,
    async: false,

    success: function(result) {

        var text = "";
        if (result.length) {
            text += "<br><table id='dataTable' class='table table-bordered table-striped'>";
            text += " <thead>";
            text += "<tr><th>#</th><th>Date</th><th>Type</th><th>Results</th></tr>";
            text += " </thead><tbody>";
            for (c in result) {

                var diag = result[c];

```



```

text += "<tr>";
var jss = parseInt(c) + 1;
text += "<td>" + jss + "</td>";
text += "<td>" + diag["day"] + "-" + diag["month"] + "-" + diag["year"] + "</td>";
text += "<td>" + diag["text"] + "</td>";
text += "<td>";
var reports = diag["results"];
for (i in reports) {
    var observation = reports[i];
    $.ajax({
        url: "http://localhost:3000/observation/" + observation["id"],
        async: false,

        success: function(result2) {
            var text2 = "";

            for (i in result2) {

                var name = result2[i].name;
                var value2 = result2[i].value;
                var unit = result2[i].unit;
                var patient_id = result2[i].id
                text2 += name + ":" + value2 + " " + unit + "<br>";

            }
            text += text2;
        }
    });

}
text += "</td>";
text += "</tr>";

}
text += "</table>";
} else {
    text = "<h5>No Diagnostic Reports for this Encounter</h5>";
}

```

```
        $('#diag').html(text);
    }

});

$("#add").click(function() {
    window.location.href = 'createMedication.php';
});

});
</script>
<?php include('footer.php'); ?>
```

Σελίδα προσθήκης φαρμακευτικής αγωγής (createMedication.php)

```
<?php
include('header.php');
// Check if the patient is not selected
if (!isset($_SESSION["encounter"])) {
    header("location: selectPatient.php");
    exit;
}
if(isset($_POST['submit'])) {
    $med = $_POST['med'];
    echo $med;
}
if (isset($_POST["med"])) {
    // include("connection.php");
    date_default_timezone_set('Europe/Athens');
    $date = date('Y-m-d h:i:s', time());
    do {
        $rand1 = str_pad(dechex(rand(0x00000000, 0xFFFFFFFF)), 8, 0, STR_PAD_LEFT);
        $rand2 = str_pad(dechex(rand(0x0000, 0xFFFF)), 4, 0, STR_PAD_LEFT);
        $rand3 = str_pad(dechex(rand(0x0000, 0xFFFF)), 4, 0, STR_PAD_LEFT);
        $rand4 = str_pad(dechex(rand(0x0000, 0xFFFF)), 4, 0, STR_PAD_LEFT);
        $rand5 = str_pad(dechex(rand(0x000000000000, 0xFFFFFFFFFFFFFFF)), 12, 0, STR_PAD_LEFT);
        $id = $rand1 . "-" . $rand2 . "-" . $rand3 . "-" . $rand4 . "-" . $rand5;
    }?>
<script>
$(document).ready(function() {
    $.ajax({
        url: "http://localhost:3000/medication/" + "<?=$id ?>",
        async: false,

        success: function(result) {
            if (result.length) {
                <?php $found = 1; ?>
            } else {
                <?php $found = 0; ?>
            }
        }

    });
});
});
```

```

</script>
<?php

} while ($found > 0);
?>
<script>
$(document).ready(function() {
$.ajax('http://localhost:3000/medication', {
type: 'POST', // http method
dataType: 'json',
data: {
'encounter_id': '<?= $id ?>',
'json': '<?= $json ?>'
}, // data to submit
success: function(data, status, xhr) {
window.location.href = 'showEncounter.php';
},
error: function(jqXHR, textStatus, errorMessage) {
$('p').append('Error: ' + errorMessage);
}
});

});
</script>
<p></p>
<?php
exit;

} else {

?>
<div class="container-fluid bg-3 text-center">
<h3>Create New Medicine Request</h3>
<div class="container">
<form method='POST'>
<table class='table table-bordered table-striped'>
<tr>
<td>
Choose a medicine from the list or enter a new one:
</td>
<td>

```

```

        <div id="datalist">
            </div>
        </td>
    </tr>
</tr>
<tr>
    <td>
        <input type='submit'>
    </td>
</tr>
</table>
</form>
</div>
<div id="patient">

</div>
</div>
<script>
$(document).ready(function() {
    $.ajax({
        url: "http://localhost:3000/distinctmedications",
        async: false,
        success: function(result) {
            var text = "<input list='medicines' name='med' id='med'>"
            text += "<datalist id='medicines' name='medicines' style='width:400px;'>";

            for (i in result) {

                text += "<option value='" + result[i].medicine + "'>";
            }
            text += "</datalist>";
            $('#datalist').html(text);
        }
    });
});
</script>
<?php
}
include('footer.php'); ?>

```

Σελίδα ενημέρωσης φαρμακευτικής αγωγής (updateMedication.php)

```
<?php
include('header.php');

// Check if the patient is not selected
if (!isset($_SESSION["encounter"])) {
    header("location: selectPatient.php");
    exit;
}

if (isset($_POST["med"])) {

    date_default_timezone_set('Europe/Athens');
    $date = date('Y-m-d h:i:s', time());
    $id = $_POST["id"];
    $med = $_POST['med'];

    $json = '{ "id": "" . $id . "", "meta": { "profile": [ "http://standardhealthrecord.org/fhir/StructureDefinition/shr-medication-MedicationRequested" ] }, "intent": "order", "status": "active", "context": { "id": "" . $_SESSION["encounter"] . "", "resourceType": "Encounter" }, "subject": { "id": "" . $_SESSION["patient"] . "", "resourceType": "Patient" }, "extension": [ { "url": "http://standardhealthrecord.org/fhir/StructureDefinition/shr-base-ActionCode-extension", "valueCodeableConcept": { "text": "Prescription of drug (procedure)", "coding": [ { "code": "33633005", "system": "http://snomed.info/sct", "display": "Prescription of drug (procedure)" } ] } }, { "url": "http://standardhealthrecord.org/fhir/StructureDefinition/shr-action-RequestedContext-extension", "extension": [ { "url": "http://standardhealthrecord.org/fhir/StructureDefinition/shr-action-Status-extension", "valueCode": "completed" }, { "url": "http://standardhealthrecord.org/fhir/StructureDefinition/shr-action-RequestIntent-extension", "valueCode": "original-order" } ] } ], "authoredOn": "" . $date . "", "medication": { "CodeableConcept": { "text": "" . $_POST["med"] . "", "coding": [ { "code": "", "system": "http://www.nlm.nih.gov/research/umls/rxnorm", "display": "" . $_POST["med"] . "" } ] } }, "resourceType": "MedicationRequest" }';

    ?>
    <script>
        $(document).ready(function() {

            $.ajax('http://localhost:3000/medication/'+ '<?=$id?>', {
                type: 'PUT', // http method
                dataType: 'json',
                data: {
                    'json': '<?=$json ?>'
                }
            });
        });
    </script>
}
```

```

    }, // data to submit
    success: function(data, status, xhr) {
        window.location.href = 'showEncounter.php';

    },
    error: function(jqXHR, textStatus, errorMessage) {
        $('p').append('Error: ' + errorMessage);
    }
});

});
</script>
<p></p>
<?php
exit;

} else {

if (isset($_POST["id"])) {

?>

<script>
var medicine_desc;
$(document).ready(function() {

var med_id = '<?php echo $_POST["id"] ?>';

$.ajax({
    url: "http://localhost:3000/medication/" + med_id,
    async: false,

    success: function(result) {

        if (result.length){
            medicine_desc = result[0]["medication"];
            console.log("MEDICINE =" + medicine_desc);
        }
    }
}
}
}

```

```

    });
    });
</script>

<div class="container-fluid bg-3 text-center">
  <h3>Update Medication Request</h3>

  <div class="container">
    <form method='POST'>
      <table class='table table-bordered table-striped'>

        <tr>
          <td>
            Choose a new medicine from the list or enter a new one:
          </td>
          <td>
            <div id="datalist">

              </div>
            </td>

        </tr>
        <tr>
          <td>
            <input type='submit'>
          </td>
          <td>
            <input type='hidden' value="<?=$_POST["id"] ?>" name='id'>
          </td>

        </tr>
      </table>
    </form>
  </div>

  <div id="patient">

```



```

    </div>
</div>
<!-- Script to use ajax() method to
add text content -->
<script>
    $(document).ready(function() {

        $.ajax({

            url: "http://localhost:3000/distinctmedications",
            async: false,

            success: function(result) {

                // var text = "<label for='medicine'>Choose a medicine from the list or enter a new
one:</label>";
                var text = "<input list='medicines' name='med' id='med'>"
                text += "<datalist id='medicines' name='medicines' style='width:400px;'>";

                for (i in result) {

                    text += "<option value='" + result[i].medicine + "'";

                    text += ">";

                }
                text += "</datalist>";

                $('#datalist').html(text);

            }

        });
    });

```

```
});  
</script>  
<?php  
}  
}  
include('footer.php'); ?>
```

Σελίδα διαγραφής φαρμακευτικής αγωγής (deleteMedication.php)

```
<?php  
include('header.php');  
  
// Check if the patient is not selected  
if (!isset($_SESSION["encounter"])) {  
    header("location: selectPatient.php");  
    exit;  
}  
  
if (isset($_POST["id"])) {  
    echo "<h1>DELETE</h1>";  
  
?>  
  
<script>  
$(document).ready(function() {  
    $.ajax('http://localhost:3000/medication/' + '<?=$_POST["id"] ?>', {  
        type: 'DELETE', // http method  
        async: false,  
        success: function(result, status, xhr) {  
            $('#message').text('status: ' + status + ', result: ' + result);  
            window.location.href = 'showEncounter.php';  
        },  
        error: function(jqXHR, textStatus, errorMessage) {  
            $('#message').text('Error: ' + errorMessage);  
        }  
    });  
});
```

```
    }  
    });  
  
});  
</script>  
  
<?php  
    exit;  
}  
  
?>  
  
<div id="message"></div>  
  
<?php include('footer.php'); ?>
```

Κώδικας REST API Διεπαφής

Παρακάτω παρατίθεται ο κώδικας του REST API που αναπτύχθηκε σε node.js. Πρόκειται για τη σελίδα index.js που φορτώνεται με την έναρξη του node.js server με την εντολή node ./index.js και υλοποιεί όλες τις λειτουργίες της διεπαφής που αναλύθηκαν στο Κεφάλαιο του Σχεδιασμού της Εφαρμογής και τη σελίδα queries.js, όπου υλοποιούνται όλα οι μέθοδοι που αναφέρονται στη σελίδα index.js και εκτελούν στην ουσία τα ερωτήματα στη βάση δεδομένων firebase. Στη σελίδα index.js, ορίζονται τα αιτήματα που μπορεί να δεχθεί ο node.js server και ποιες μέθοδοι θα κληθούν σε κάθε αίτημα. Για παράδειγμα αν δεχθεί ο node.js server αίτημα GET <URL>/patient/<id>, όπου <id> είναι το id ενός Ασθενή (Patient) θα κληθεί η μέθοδος getPatientById που είναι ορισμένη στη σελίδα queries.js.

Κώδικας της index.js

```
const express = require('express')
const bodyParser = require('body-parser')
const app = express()
const db = require('./queries')
const port = 3000

//define cors
const cors = require('cors');
app.use(cors({
  origin: '*'
}));

app.use(bodyParser.json())
app.use(
  bodyParser.urlencoded({
    extended: true,
  })
)

app.get('/', (request, response) => {
  response.json({ info: 'Node.js, Express, and Postgres API' })
})

app.get('/organizations', db.getOrganizations)

app.get('/patients', db.getPatients)
app.get('/patient/:id', db.getPatientById)
app.post('/patient/send/:id', db.sendPatientById)
app.get('/patients/name/:id', db.getPatientsByName)
```

```
app.get('/patientreports/:id', db.getPatientDiagnosticReports)
app.get('/patientmedication/:id', db.getPatientMedicationRequests)
app.get('/patientencounters/:id', db.getPatientEncounters)

app.get('/encounter/:id', db.getEncounterById)
app.post('/encounters', db.createEncounter)
app.get('/encountermedications/:id', db.getMedicationByEncounter)
app.get('/encounterreports/:id', db.getEncounterDiagnosticReports)

app.get('/reasons', db.getEncounterReasons)

app.get('/observation/:id', db.getObservationById)

app.get('/medication/:id', db.getMedicationById)
app.post('/medication', db.createMedicationRequest)
app.put('/medication/:id', db.updateMedicationRequest)
app.delete('/medication/:id', db.deleteMedicationRequest)

app.get('/distinctmedications', db.getMedications)

app.listen(port, () => {
  console.log(`App running on port ${port}.`)
})
```

Κώδικας της queries.js

```
const Pool = require('pg').Pool
const pool = new Pool({
  user: 'postgres',
  host: 'localhost',
  database: 'fhirbase',
  password: '123456',
  port: 5432,
})

const nodemailer = require('nodemailer');

const Cryptr = require('cryptr');

const sendPatientById = (request, response) => {

  const patient_id = request.params.id
  const { recipient } = request.body

  const sql = "select * from public.patient WHERE id = $1";
  pool.query(sql, [patient_id], (error, results) => {
    if (error) {
      throw error
    }
  })

  var transporter = nodemailer.createTransport({
    host: "smtp.mailtrap.io",
    port: 2525,
    auth: {
      user: "b54f3204fe38f5",
      pass: "e9c690e4e28be4"
    }
  });

  var message = JSON.stringify(results.rows);

  const cryptr = new Cryptr('myTotallySecretKey');

  const encryptedMessage = cryptr.encrypt(message);
```

```

var mailOptions = {
  from: 'fhirproject@mailtrap.io',
  to: recipient,
  subject: 'FHIR Record',
  text: encryptedMessage
};

transporter.sendMail(mailOptions, function (error, info) {
  if (error) {
    console.log(error);
  } else {
    console.log('Email sent: ' + info.response);
    response.status(200).json("Record Sent Successfully by Email")
  }
});

})
}

const getOrganizations = (request, response) => {
  const sql = "select id, resource#>>'{name}' as name from public.organization order by resource#>>'{name}'";
  pool.query(sql, (error, results) => {
    if (error) {
      throw error
    }
    response.status(200).json(results.rows)
  })
}

const getPatients = (request, response) => {
  const sql = "select id, resource#>>'{name,0,given,0}' as given, resource#>>'{name,0,family}' as family, resource#>>'{address,0,city}' as city, resource#>>'{address,0,line,0}' as street, resource#>>'{telecom,0,value}' as phone, resource#>>'{birthDate}' as birthdate, resource#>>'{identifier,2,value}' as ssn from public.patient";
  pool.query(sql, (error, results) => {
    if (error) {
      throw error
    }
    response.status(200).json(results.rows)
  })
}

```

```

    })
  }

  const getPatientById = (request, response) => {
    const id = request.params.id

    const sql = "select id, resource#>>'{name,0,given,0}' as given, resource#>>'{name,0,family}'
as family, resource#>>'{address,0,city}' as city, resource#>>'{address,0,line,0}' as street,
resource#>>'{telecom,0,value}' as phone, resource#>>'{birthDate}' as birthdate,
resource#>>'{identifier,2,value}' as ssn from public.patient WHERE id = $1";
    pool.query(sql, [id], (error, results) => {
      if (error) {
        throw error
      }

      response.status(200).json(results.rows)
    })
  }

  const getPatientsByName = (request, response) => {
    const id = request.params.id

    var sql = "select id, resource#>>'{name,0,given,0}' as given, resource#>>'{name,0,family}' as
family from public.patient WHERE resource#>>'{name,0,family}' like '% | | $1 | | %'";

    pool.query(sql, [id], (error, results) => {
      if (error) {
        throw error
      }

      response.status(200).json(results.rows)
    })
  }

  const getPatientDiagnosticReports = (request, response) => {
    const id = request.params.id

    const sql = "SELECT resource#>>'{code, text}' as text, extract(year from (resource-
>'issued')::text::date) as year, extract(month from (resource->'issued')::text::date) as month,

```



```

extract(day from (resource->'issued')::text::date)as day, resource#>'{result}' as results FROM
public.diagnosticreport where resource#>>'{subject,id}'=$1 order by year DESC, month DESC,
day DESC";
pool.query(sql, [id], (error, results) => {
  if (error) {
    throw error
  }

  response.status(200).json(results.rows)
})
}

const getEncounterById = (request, response) => {
  const id = request.params.id

  const sql = "SELECT id,resource#>>'{reason,0,coding,0,display}' as symptom, extract(year
from (resource#>>'{period,start}')::text::date) as year,extract(month from
(resource#>>'{period,start}')::text::date) as month, extract(day from
(resource#>>'{period,start}')::text::date) as day from public.encounter where id=$1";
  pool.query(sql, [id], (error, results) => {
    if (error) {
      throw error
    }

    response.status(200).json(results.rows)
  })
}

const getObservationById = (request, response) => {
  const id = request.params.id

  const sql = "SELECT extract(year from (resource->'issued')::text::date) as year,
extract(month from (resource->'issued')::text::date) as month, extract(day from (resource-
>'issued')::text::date)as day, resource#>>'{code, text}' as name, resource#>>'{value, Quantity,
value}' as value, resource#>>'{value, Quantity, unit}' as unit FROM public.observation where
id=$1";
  pool.query(sql, [id], (error, results) => {
    if (error) {
      throw error
    }
  }
}

```

```

    response.status(200).json(results.rows)
  })
}

const getMedicationById = (request, response) => {
  const id = request.params.id

  const sql = "SELECT id, extract(year from (resource->'authoredOn')::text::date) as year,
extract(month from (resource->'authoredOn')::text::date) as month, extract(day from
(resource->'authoredOn')::text::date)as day, resource#>>'{medication, CodeableConcept,
text}' as medication FROM public.medicationrequest where id=$1";
  pool.query(sql, [id], (error, results) => {
    if (error) {
      throw error
    }

    response.status(200).json(results.rows)
  })
}

const getPatientMedicationRequests = (request, response) => {
  const id = request.params.id

  const sql = "SELECT id, extract(year from (resource->'authoredOn')::text::date) as year,
extract(month from (resource->'authoredOn')::text::date) as month, extract(day from
(resource->'authoredOn')::text::date)as day, resource#>>'{medication, CodeableConcept,
text}' as medication FROM public.medicationrequest where resource#>>'{subject,id}'=$1
order by year DESC, month DESC, day DESC";
  pool.query(sql, [id], (error, results) => {
    if (error) {
      throw error
    }

    response.status(200).json(results.rows)
  })
}

const getPatientEncounters = (request, response) => {
  const id = request.params.id

  const sql = "select id,resource#>>'{reason,0,coding,0,display}' as symptom, extract(year
from (resource#>>'{period,start}')::text::date) as year,extract(month from

```

```

(resource#>>'{period,start}')::text::date) as month, extract(day from
(resource#>>'{period,start}')::text::date) as day from public.encounter where
resource#>>'{subject,id}'=$1 order by year DESC, month DESC, day DESC";
pool.query(sql, [id], (error, results) => {
  if (error) {
    throw error
  }

  response.status(200).json(results.rows)
})
}

const getMedicationByEncounter = (request, response) => {
  const id = request.params.id

  const sql = "select id, resource#>>'{medication,CodeableConcept,text}' as medication from
public.medicationrequest where resource#>>'{context, id}' = $1";
  pool.query(sql, [id], (error, results) => {
    if (error) {
      throw error
    }

    response.status(200).json(results.rows)
  })
}

const getEncounterReasons = (request, response) => {
  const sql = "select distinct resource#>>'{reason,0,coding,0,code}' as code,
resource#>>'{reason,0,coding,0,display}' as reason from encounter order by reason";
  pool.query(sql, (error, results) => {
    if (error) {
      throw error
    }

    response.status(200).json(results.rows)
  })
}

const getEncounterDiagnosticReports = (request, response) => {
  const id = request.params.id

```

```

const sql = "SELECT resource#>>'{code, text}' as text, extract(year from (resource-
>'issued')::text::date) as year, extract(month from (resource->'issued')::text::date) as month,
extract(day from (resource->'issued')::text::date) as day, resource#>'{result}' as results FROM
public.diagnosticreport where resource#>>'{context,id}'=$1 order by year DESC, month
DESC, day DESC";
pool.query(sql, [id], (error, results) => {
  if (error) {
    throw error
  }

  response.status(200).json(results.rows)
})
}

const getMedications = (request, response) => {
  const sql = "select distinct resource#>>'{medication,CodeableConcept,text}' as medicine
from medicationrequest order by medicine";
  pool.query(sql, (error, results) => {
    if (error) {
      throw error
    }

    response.status(200).json(results.rows)
  })
}

const createEncounter = (request, response) => {

  const { encounter_id, json } = request.body

  const sql = "INSERT INTO encounter (id, txid, resource) VALUES ('" + encounter_id + "', 0, '" +
json + "')";

  pool.query(sql, (error, result) => {
    if (error) {
      throw error
    }

    response.status(201).json('Encounter added ');
  }
}

```

```

    })

  }

  const createMedicationRequest = (request, response) => {

    const { encounter_id, json } = request.body

    const sql = "INSERT INTO medicationrequest (id, txid, resource) VALUES ('" + encounter_id +
    "; 0, '" + json + "'";

    pool.query(sql, (error, result) => {
      if (error) {
        throw error
      }

      response.status(201).json('Medication Request added ');
    })

  }

  const deleteMedicationRequest = (request, response) => {

    const id = request.params.id

    const sql = "DELETE FROM medicationrequest where id = $1";

    pool.query(sql, [id], (error, result) => {
      if (error) {
        throw error
      }
    })
  }
}

```

```

    response.status(201).json('Medication Request Deleted ');

  })

}

const updateMedicationRequest = (request, response) => {
  const id = request.params.id
  const { json } = request.body

  pool.query(
    'UPDATE medicationrequest SET resource = $1 WHERE id = $2',
    [json, id],
    (error, results) => {
      if (error) {
        throw error
      }
      response.status(200).json('Medication Request Updated')
    }
  )
}

module.exports = {
  getOrganizations,
  getPatients,
  getPatientById,
  sendPatientById,
  getPatientDiagnosticReports,
  getEncounterById,
  getObservationById,
  getMedicationById,
  getPatientMedicationRequests,
  getPatientEncounters,
  getMedicationByEncounter,
  getPatientsByName,
  getEncounterReasons,
  getEncounterDiagnosticReports,
  getMedications,

```

```
createEncounter,  
createMedicationRequest,  
updateMedicationRequest,  
deleteMedicationRequest,  
}
```

Βιβλιογραφία

Αναφορά σε ιστοσελίδα:

- [1] Πρότυπο HL7 FHIR, Έκδοση 4.3.0, Ιούλιος 2022, διαθέσιμο από: <http://hl7.org/fhir>
- [2] Γλώσσα Προγραμματισμού PHP, Έκδοση 8.1, Ιούλιος 2022, διαθέσιμο από: <http://php.net>
- [3] Γλώσσα Προγραμματισμού JavaScript, Ιούλιος 2022, διαθέσιμο από: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- [4] Ασύγχρονη κλήση σε περιβάλλον JavaScript και ανταλλαγή δεδομένων - Ajax, Ιούλιος 2022, διαθέσιμο από: <https://developer.mozilla.org/en-US/docs/Web/Guide/AJAX>
- [5] Πρότυπο Ανταλλαγής Δεδομένων JSON, Ιούλιος 2022, διαθέσιμο από: <https://www.json.org/>
- [6] Περιβάλλον Εκτέλεσης ServerSide JavaScript Εφαρμογών Node.js, Έκδοση 16.15, Ιούλιος 2022, διαθέσιμο από: <https://nodejs.org>
- [7] Περιβάλλον Διαχείρισης Βάσης Δεδομένων PostgreSQL , Έκδοση 14, Ιούλιος 2022, διαθέσιμο από: <https://www.postgresql.org/>
- [8] Λίστα Οντοτήτων του προτύπου HL7 – FHIR, Έκδοση 4.3.0, Ιούλιος 2022, διαθέσιμο από: <http://hl7.org/fhir/patient.html>.
- [9] Περιγραφή Οντότητας Patient του προτύπου HL7 – FHIR, Έκδοση 4.3.0, Ιούλιος 2022, διαθέσιμο από: <http://hl7.org/fhir/patient.html>.
- [10] Περιγραφή Οντότητας Encounter του προτύπου HL7 – FHIR, Έκδοση 4.3.0, Ιούλιος 2022, διαθέσιμο από: <http://hl7.org/fhir/encounter.html>.
- [11] Περιγραφή Οντότητας Observation του προτύπου HL7 – FHIR, Έκδοση 4.3.0, Ιούλιος 2022, διαθέσιμο από: <http://hl7.org/fhir/observation.html>.
- [12] Περιγραφή Οντότητας Diagnostic Report του προτύπου HL7 – FHIR, Έκδοση 4.3.0, Ιούλιος 2022, διαθέσιμο από: <http://hl7.org/fhir/diagnosticreport.html>.
- [13] Περιγραφή Οντότητας Medication Request του προτύπου HL7 – FHIR, Έκδοση 4.3.0, Ιούλιος 2022, διαθέσιμο από: <http://hl7.org/fhir/medicationrequest.html>.

Συντομογραφίες - Αρκτικόλεξα - Ακρωνύμια

βλπ

βλέπε

κ.λπ.

και λοιπά

κ.ο.κ

και ούτω καθεξής

Απόδοση ξενόγλωσσων όρων

Απόδοση

Ξενόγλωσσος όρος

Ασθενής

Patient

Επίσκεψη

Encounter

Καταγραφές

Observation

Διαγνωστική Αναφορά

Diagnostic Report

Φαρμακευτική Αγωγή

Medication Request