



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ
ΑΤΤΙΚΗΣ**

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΥΠΟΛΟΓΙΣΤΩΝ**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**«ΚΑΤΑΓΡΑΦΗ ΚΑΙ ΟΠΤΙΚΟΠΟΙΗΣΗ ΖΩΤΙΚΩΝ ΣΗΜΕΙΩΝ
ΕΞΩΤΕΡΙΚΩΝ ΑΣΘΕΝΩΝ ΚΑΙ ΚΑΤΑΓΡΑΦΗ ΚΑΙ ΟΠΤΙΚΟΠΟΙΗΣΗ
ΔΕΔΟΜΕΝΩΝ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ ΕΞΩΤΕΡΙΚΩΝ ΑΣΘΕΝΩΝ, ΜΕ
ΕΞΥΠΝΑ ΚΙΝΗΤΑ ΤΗΛΕΦΩΝΑ»**

Εμμανουήλ Γρηγοράκης

ΑΜ: 18390071

Επιβλέπων: Καθηγητής Νικήτας Ν. Καρανικόλας



UNIVERSITY OF WEST ATTICA

FACULTY OF ENGINEERING

DEPARTMENT OF INFORMATICS AND

COMPUTER ENGINEERING

DIPLOMA THESIS

**«RECORDING AND VISUALIZATION OF OUTPATIENT VITAL
SIGNS AND RECORDING AND VISUALIZING OF OUTPATIENT
MONITORING DATA, WITH SMARTPHONES»**

Emmanouil Grigorakis

RN: 18390071

Supervisor: Professor Nikitas N.Karanikolas

Η Διπλωματική Εργασία εξετάστηκε και βαθμολογήθηκε από την τριμελή επιτροπή:

Νικήτας Καρανικόλας Καθηγητής	Ιωάννης Βογιατζής Καθηγητής	Πάρις Μαστοροκόστας Καθηγητής

Copyright © Με επιφύλαξη παντός δικαιώματος. All rights reserved.

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ και Γρηγοράκης Ι. Εμμανουήλ, Ιούλιος,
2024**

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 τους συγγραφείς.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον/την συγγραφέα του και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις θέσεις του επιβλέποντος, της επιτροπής εξέτασης ή τις επίσημες θέσεις του Τμήματος και του Ιδρύματος.

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Γρηγοράκης Εμμανουήλ του Ιωάννη, με αριθμό μητρώου 18390071 φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής ΜΗΧΑΝΙΚΩΝ του Τμήματος ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ,

δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του διπλώματός μου.»

Ο Δηλών

Γρηγοράκης Εμμανουήλ



ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα διπλωματική εργασία ολοκληρώθηκε μετά από επίμονες προσπάθειες, σε ένα ενδιαφέρον γνωστικό αντικείμενο, αυτό της πληροφορικής υγείας. Σε αυτό το σημείο θα ήθελα να ευχαριστήσω την οικογένεια μου και τους φίλους μου που ήταν δίπλα μου όλα τα χρόνια. Τέλος, θα ήθελα να ευχαριστήσω και τον επιβλέποντα καθηγητή μου, Νικήτα Καρανικόλα, όσο για τις γνώσεις που μου προσέφερε, τόσο και για την εμπιστοσύνη του σε εμένα.

ΠΕΡΙΛΗΨΗ

Στον χώρο της πληροφορικής υγείας, για την παροχή ποιοτικής υγειονομικής περίθαλψης, η ακριβής και σύντομη αποτύπωση των δεδομένων αλλά και η αποδοτική διαχείριση τους, είναι αδιαμφισβήτητα κρίσιμη. Παραδοσιακά, όλες οι πληροφορίες των ασθενών συνήθως αποθηκεύονται γραπτώς, χωρίς την διαμεσολάβηση κάποιου ψηφιακού συστήματος, κι αν αυτο διαμεσολαβεί, τότε γίνεται χειροκίνητα, οδηγώντας έτσι σε ανακρίβειες και ενδεχόμενα λάθη. Με την πάροδο του χρόνου και συνάμα της τεχνολογίας, η πολυπλοκότητα των ιατρικών δεδομένων αυξάνεται, όσο αυξάνεται και η ανάγκη για πιο ολοκληρωμένες και, κυρίως, αυτοματοποιημένες λύσεις. Αυτή η μελέτη, παρουσιάζει την ανάπτυξη ενός ολοκληρωμένου συστήματος που είναι σχεδιασμένο να βελτιώσει την παρακολούθηση των ασθενών, αφού αυτοί καταχωρούν τις μετρήσεις του χρησιμοποιώντας το κινητό του τηλέφωνο και ο εκάστοτε ιατρός αφού τις μελετήσει μπορεί και προσαρμόζει τη θεραπεία που θα ακολουθηθεί.

Το σύστημα αποτελείται από τρία κύρια συστατικά: μία εφαρμογή Android για την καταχώρηση δεδομένων από τους ασθενείς, έναν Spring Boot server για τη διαχείριση και αποθήκευση των δεδομένων και μία web εφαρμογή React.js για τους ιατρούς, για την προβολή και διαχείριση των δεδομένων ασθενών.

Λέξεις κλειδιά: πληροφορική υγείας, υγειονομική περίθαλψη, ψηφιακό σύστημα, ολοκληρωμένο σύστημα, παρακολούθηση ασθενών, Android, Spring Boot server, React.js

ABSTRACT

For the delivery of top-tier medical care, the accurate, quick way of data recording and the efficient management of patient data, is undeniably crucial. Traditionally, patient information is recorded on paper, without the intermediation of any digital system, and if the latter acts as an intermediate, it is in a manual mode, leading to inefficiencies and potential errors. Over time, as the complexity of medical data is increased, the need for a more universal, integrated and, more importantly, automated solution is also needed. This study presents the development of an integrated, universal system designed to improve patient monitoring, where patients record their measurements using their smartphones, and doctors can adjust the treatment based on the patient's measurements.

The system comprises of three main components: an Android application for patient data recording, a Spring Boot server for data management and storage, and a React.js web application for healthcare professionals to view and manage patient data.

Keywords: medical care, patient data, digital system, patient monitoring, smartphones, Android, Spring Boot server, React.js

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΥΧΑΡΙΣΤΙΕΣ.....	6
Κεφάλαιο 1: Εισαγωγή.....	11
1.1 Εισαγωγή στην Πληροφορική Υγείας.....	11
1.2 Χαρακτηριστικά Πληροφορικής Υγείας.....	12
1.2.1 Πληροφοριακά Συστήματα Υγείας.....	12
1.2.2 Κωδικοποίηση Ιατρικής Πληροφορίας.....	13
1.2.3 Επεξεργασία Ιατρικού Σήματος.....	13
1.2.4 Βιοπληροφορική.....	13
1.2.5 Στατιστική Επεξεργασία Ιατρικών Δεδομένων.....	13
1.2.6 Διαχείριση/Ανάκτηση Ιατρικής Τεκμηρίωσης.....	14
1.2.7 Τηλεϊατρική.....	14
1.2.8 Συστήματα Υποστήριξης Αποφάσεων (D.S.S.) για την Ιατρική.....	14
1.3 Περιγραφή προβλήματος: Παρακολούθηση Μεταμοσχευμένων Ασθενών.....	14
Κεφάλαιο 2: Σχεδιασμός λύσης προβλήματος.....	17
2.1 Android App για Ασθενή.....	17
2.2 Web Εφαρμογή για Ιατρό.....	18
2.3 Προσαρμόσιμο Σύστημα.....	18
2.4 Επεκτάσιμο Σύστημα.....	18
2.5 Πιθανές Τεχνολογίες.....	19
Κεφάλαιο 3: Τεχνική ανάλυση συστήματος.....	20
3.1 Βάση Δεδομένων.....	20
3.2 Μοντέλο Entity-Attribute-Value ως πίνακας αποθήκευσης δεδομένων.....	23
3.2.1 Μοντέλο Entity-Attribute-Value (EAV).....	23
3.2.2 Λόγοι Επιλογής του Μοντέλου EAV.....	24
3.2.3 Προκλήσεις και πιθανές λύσεις για το μοντέλο EAV.....	24
3.3 Δημιουργία καθολικού πίνακα εξετάσεων.....	25
3.4 Σύντομη αναφορά σε πιθανή εναλλακτική λύση.....	27
3.5 Εφαρμογή Android:.....	28
3.6 Spring Boot Server:.....	28
3.6.1 Επιγραμματικός τρόπος λειτουργίας του server.....	29
3.7 Εφαρμογή Ιατρού (Frontend):.....	30

Κεφάλαιο 4: Το αποτέλεσμα – παραχθέν σύστημα.....	31
4.1 Λογισμικό πρόγραμμα για τον επιβλεπόμενο ασθενή	31
4.2 Λογισμικό πρόγραμμα για τον εποπτεύοντα Ιατρό.....	33
Βιβλιογραφία.....	42

Κεφάλαιο 1: Εισαγωγή

1.1 Εισαγωγή στην Πληροφορική Υγείας

Χάρη στην ανάπτυξη της επιστήμης των υπολογιστών, του λογισμικού και της τεχνολογίας γενικότερα, έχει συνάμα αναπτυχθεί ιδιαίτερα και ο κλάδος της υγείας παρέχοντας προηγμένους τρόπους για διάγνωση, θεραπεία, παρακολούθηση και γενικότερη φροντίδα των ασθενών. Οι τεχνολογίες που έχουν προκύψει μετά την ψηφιακή επανάσταση, μπορούν, ουσιαστικά, να επαναπροσδιορίσουν τον τομέα της πληροφορικής υγείας και να τον προσαρμόσουν στις πιο σύγχρονες ατομικές ανάγκες.

Η πληροφορική υγείας, ήδη μεταμορφώνει την υγειονομική περίθαλψη. Με τις νεοεισαγόμενες σύγχρονες εφαρμογές πληροφορικής έχουν αναβαθμιστεί οι υπηρεσίες προς τον πολίτη και η ανάπτυξη καινοτόμων προϊόντων και υπηρεσιών είναι πλέον γεγονός. Λόγω της αυξημένης ανάγκης για καλύτερες υπηρεσίες υγείας, μεγάλες εταιρείες διαθέτουν ήδη κενοτόμα προϊόντα και υπηρεσίες που αφορούν το χώρο της υγείας.

Οι φορητές συσκευές υγείας, όπως έξυπνα ρολόγια, ζυγαριές και αισθητήρες, προσφέρουν συνεχή παρακολούθηση της υγείας των χρηστών, παρέχοντας πολύτιμα δεδομένα που μπορούν να χρησιμοποιηθούν για την πρόληψη προβλημάτων υγείας.

Η τεχνητή νοημοσύνη έχει αναδείξει νέους τρόπους πρόβλεψης αλλά και πρόληψης κάποιων ασθενειών, αναλύοντας τεράστια σύνολα δεδομένων, εντοπίζοντας μοτίβα και τάσεις που δεν θα υπήρχε περίπτωση να αναγνωριστούν με πιο παραδοσιακές μεθόδους. Η εξατομικευμένη ιατρική, βασισμένη σε γενετικά δεδομένα και άλλες προσωπικές πληροφορίες, παρέχει θεραπείες που είναι προσαρμοσμένες στις μοναδικές ανάγκες κάθε ασθενούς.

Η τηλεϊατρική έχει διευκολύνει την πρόσβαση σε ιατρικές υπηρεσίες για άτομα που ζουν σε απομακρυσμένες περιοχές, προσφέροντας ιατρική φροντίδα χωρίς την ανάγκη φυσικής παρουσίας. Αυτό είναι ιδιαίτερα σημαντικό, ειδικά για μια χώρα όπως η Ελλάδα, καθώς έχει μειώσει την ανάγκη για μετακινήσεις και έχει αυξήσει την άνεση και την αποτελεσματικότητα της φροντίδας υγείας.

Τέλος, η αύξηση των περιστατικών υγείας σε έναν γηράσκοντα πληθυσμό, όπως αυτόν της Ελλάδας, η οποία δεν συνοδεύεται από ανάλογη αύξηση κατάλληλου υγειονομικού προσωπικού καθιστά απαραίτητη τη χρήση προηγμένων πληροφοριακών συστημάτων που μπορούν να συμβάλουν στην αποτελεσματικότερη διαχείριση των υγειονομικών αναγκών και στη βελτίωση της ποιότητας της παρεχόμενης ιατρικής φροντίδας, χωρίς ανάλογη αύξηση των πόρων.

1.2 Χαρακτηριστικά Πληροφορικής Υγείας

Εν συντομία, η Πληροφορική Υγείας είναι η επιστήμη με την οποία ο άνθρωπος διαχειρίζεται πλέον, με τη βοήθεια του ηλεκτρονικού υπολογιστή, μεγάλο όγκο πληροφοριών στο χώρο της υγείας ή της βιοϊατρικής. Σκοπός της πληροφορικής υγείας είναι βασικά δύο πράγματα. Το πρώτο είναι η περίθαλψη του ασθενούς, δηλαδή η πρόληψη, διάγνωση και θεραπεία κάποιας ασθένειας. Το δεύτερο είναι το διοικητικό, που αναλαμβάνει τις υποστηρικτικές διαδικασίες, για την επίτευξη του πρώτου.

Πλέον, η ιατρική πληροφορική, επιγραμματικά μπορεί να αποτυπωθεί και ως εξής:

Ιατρική Πληροφορική = Πληροφοριακά Συστήματα Υγείας +
Κωδικοποίηση Ιατρικής Πληροφορίας +
Επεξεργασία Ιατρικού Σήματος +
Βιοπληροφορική +
Στατιστική Επεξεργασία Ιατρικών Δεδομένων +
Διαχείριση / Ανάκτηση Ιατρικής Τεκμηρίωσης +
Τηλεϊατρική +
D.S.S. για την Ιατρική

(Nikitas Karanikolas, Πληροφορική και Επαγγέλματα Υγείας, 2010)

Στις επόμενες υποενότητες θα εξηγηθούν αναλυτικά τα κύρια συστατικά που συνθέτουν την ιατρική πληροφορική.

1.2.1 Πληροφοριακά Συστήματα Υγείας

Τα Πληροφοριακά Συστήματα Υγείας, είναι λογισμικά συστήματα που χρησιμοποιούνται για τη διαχείριση των πληροφοριών στον τομέα της υγείας. Συμπεριλαμβάνουν ηλεκτρονικά αρχεία υγείας (EHRs), συστήματα διαχείρισης κλινικών πληροφοριών (CIS)

και άλλα εργαλεία που συμβάλλουν στη συλλογή, αποθήκευση, ανάκτηση και ανταλλαγή δεδομένων υγείας. Αυτά τα συστήματα επιτρέπουν στους επαγγελματίες υγειονομικούς να αποκτούν καίριες πληροφορίες εύκολα και γρήγορα έτσι ώστε να βελτιώνεται η αποτελεσματικότητα της ιατρικής περίθαλψης.

1.2.2 Κωδικοποίηση Ιατρικής Πληροφορίας

Η Κωδικοποίηση Ιατρικής Πληροφορίας περιλαμβάνει τη διαδικασία της μετατροπής των ιατρικών πληροφοριών και δεδομένων σε κωδικοποιημένη μορφή που είναι εύκολα κατανοητή και διαχειρίσιμη. Η διαδικασία αυτή περιλαμβάνει τη χρήση διεθνών προτύπων όπως τα ICD (International Classification of Diseases) και CPT (Current Procedural Terminology). Οι κωδικοποιήσεις αυτές τυποποιούν τα δεδομένα, διευκολύνοντας και τη συλλογή καθώς επίσης και την ανάλυση μεγάλων ποσοτήτων πληροφοριών.

1.2.3 Επεξεργασία Ιατρικού Σήματος

Η Επεξεργασία Ιατρικού Σήματος αναφέρεται στην ανάλυση και την επεξεργασία βιοϊατρικών σημάτων όπως είναι τα ηλεκτροκαρδιογραφήματα (ECG) ή τα εγκεφαλογραφήματα (EEG). Αυτή βοηθά στην ανίχνευση, ανάλυση και κατανόηση των ασθενειών.

1.2.4 Βιοπληροφορική

Η Βιοπληροφορική συνδυάζει τη βιολογία, την πληροφορική και τη στατιστική για την ανάλυση βιολογικών δεδομένων. Η επιστήμη αυτή χρησιμοποιεί λογισμικά για την κατανόηση και την ερμηνεία βιολογικών δεδομένων, βοηθώντας στην ανακάλυψη νέων φαρμάκων και στη εφεύρεση νέων θεραπευτικών προσεγγίσεων.

1.2.5 Στατιστική Επεξεργασία Ιατρικών Δεδομένων

Η Στατιστική Επεξεργασία Ιατρικών Δεδομένων είναι πολύ σημαντική για την ανάλυση και την κατανόηση των ιατρικών δεδομένων. Μέσω της στατιστικής εξάγονται χρήσιμες πληροφορίες από τα δεδομένα. Για παράδειγμα, με τη στατιστική μπορεί να γίνει ανάλυση σε μια ομάδα ηλεκτρονικών φακέλων ασθενών που πάσχουν από μια κοινή ασθένεια, να βρεθεί μια τάση-σχέση μεταξύ αυτών και έτσι να κατανοηθεί καλύτερα η εξέλιξη της ασθένειας, η αποτελεσματικότητα μιας θεραπείας ή η επίδραση άλλων παραγόντων στην υγεία των ασθενών με την εκάστοτε ασθένεια. Στο παραπάνω, μπορεί να προστεθεί και η ανάλυση επιβίωσης (ανάλυση κινδύνου) με διάφορες τεχνικές. Όλα τα παραπάνω, πλέον, με την άνοδο της τεχνητής νοημοσύνης μπορούν να υποστηριχθούν από αυτή και τη

μηχανική μάθηση, επιτρέποντας την ανάπτυξη μοντέλων πρόβλεψης ή/και βελτίωσης της ακρίβειας των θεραπευτικών αποφάσεων.

1.2.6 Διαχείριση/Ανάκτηση Ιατρικής Τεκμηρίωσης

Η Διαχείριση και Ανάκτηση Ιατρικής Τεκμηρίωσης αναφέρεται στη διαδικασία της οργάνωσης, αποθήκευσης και ανάκτησης ιατρικών εγγράφων/φακέλων και δεδομένων. Οι Ηλεκτρονικοί Φάκελοι Υγείας (EHR) αποτελούν τον πυρήνα της διαχείρισης ιατρικής τεκμηρίωσης, καθώς περιλαμβάνουν την πλήρη εικόνα του ασθενούς, δηλαδή αποτελέσματα εξετάσεων, ιατρικές διαγνώσεις, θεραπείες, συνταγογραφήσεις, κλπ.

1.2.7 Τηλεϊατρική

Η Τηλεϊατρική χρησιμοποιεί τις τηλεπικοινωνιακές τεχνολογίες για την παροχή υγειονομικών υπηρεσιών από απόσταση. Αυτό περιλαμβάνει την απομακρυσμένη διάγνωση, παρακολούθηση και, ενδεχομένως, θεραπεία των ασθενών, επιτρέποντας την παροχή φροντίδας σε απομακρυσμένες περιοχές.

1.2.8 Συστήματα Υποστήριξης Αποφάσεων (D.S.S.) για την Ιατρική

Τα Συστήματα Υποστήριξης Αποφάσεων (Decision Support Systems - D.S.S.) για την Ιατρική είναι λογισμικά που βοηθούν τους επαγγελματίες υγείας στη λήψη κλινικών αποφάσεων. Αυτά τα συστήματα λαμβάνουν δεδομένα από διάφορες πηγές (ηλεκτρονικούς φακέλους υγείας, εργαστηριακά αποτελέσματα, απεικονιστικές εξετάσεις) και τα ενοποιούν, παρέχοντας μια συνολική εικόνα της κατάστασης του ασθενούς. Έπειτα, χρησιμοποιούν μηχανική μάθηση ή/και στατιστική ανάλυση για εξαγωγή πληροφοριών και τελικά προσφέρουν ουσιαστική βοήθεια στους επαγγελματίες υγειονομικούς.

1.3 Περιγραφή προβλήματος: Παρακολούθηση Μεταμοσχευμένων Ασθενών

Αφού έχει γίνει η εισαγωγή στην Πληροφορική Υγείας, μπορούμε να αναπτύξουμε το πρόβλημα στο οποίο αναφέρεται η παρούσα μελέτη.

Η διαρκής παρακολούθηση μεταμοσχευμένων ασθενών είναι ένα κρίσιμο ζήτημα στον τομέα της υγειονομικής φροντίδας. Μετά από σοβαρές ιατρικές επεμβάσεις, όπως μεταμοσχεύσεις, οι ασθενείς χρειάζονται συνεχή παρακολούθηση για να διασφαλιστεί η αποτελεσματικότητα της θεραπείας και η πρόληψη των επιπλοκών. Η τρέχουσα διαδικασία παρακολούθησης περιλαμβάνει διάφορα βήματα, τα οποία συνήθως με τη σειρά τους περιλαμβάνουν περιοδικές επισκέψεις σε ιατρικά κέντρα και την καταγραφή των μετρήσεων μέσω διαφορετικών εργαλείων και συστημάτων.

Η τρέχουσα προσέγγιση για την παρακολούθηση των μεταμοσχευμένων ασθενών περιλαμβάνει:

1. **Τακτικές επισκέψεις στον Ιατρό:** Οι ασθενείς συνήθως επισκέπτονται τακτικά ή έκτακτα εξωτερικά ιατρεία (συνήθως) το νοσοκομείο στο οποίο έγινε η μεταμόσχευση. Αυτές οι επισκέψεις πραγματοποιούνται συνήθως σε εβδομαδιαία, δεκαπενθήμερη ή μηνιαία βάση, ανάλογα με την κατάσταση του ασθενούς και τις ανάγκες της θεραπείας.
2. **Συλλογή μετρήσεων:** Οι μετρήσεις που καταγράφονται κατά τη διάρκεια αυτών των επισκέψεων προέρχονται από διάφορες πηγές:
 - **Εργαστηριακές εξετάσεις:** Υποβάλλονται σε εργαστηριακή ανάλυση για την μέτρηση τιμών (συνήθως) στο αίμα ή στα ούρα.
 - **Απλούστερες συσκευές:** Χρησιμοποιούνται οξύμετρα, πιεσόμετρα, θερμόμετρα, ζυγαριές.
 - **Αυτοκαταγραφές από τον ασθενή:** Ο ασθενής καταγράφει ο ίδιος δεδομένα όπως ο όγκος λήψης υγρών και ο όγκος ούρων.
3. **Καταγραφή και επεξεργασία δεδομένων:** Οι μετρήσεις καταγράφονται είτε σε έντυπα είτε σε ηλεκτρονικά συστήματα, όπως πληροφοριακά συστήματα ή υπολογιστικά φύλλα. Οι ιατροί χρησιμοποιούν αυτές τις καταγραφές για να προσαρμόσουν τις θεραπείες (φαρμακευτική αγωγή, ιατρικές πράξεις κλπ.) που θα ακολουθήσει ο ασθενής.

Η υπάρχουσα μέθοδος παρακολούθησης έχει ορισμένα σημαντικά περιοριστικά σημεία:

- **Σπανιότητα Επισκέψεων:** Οι επισκέψεις στους ιατρούς είναι αραιές, γεγονός που μπορεί να περιορίσει την δυνατότητα των ιατρών να παρακολουθούν καθημερινές αλλαγές στην υγεία του ασθενούς.
- **Καθυστέρηση Πληροφόρησης:** Οι αναλύσεις και οι αποφάσεις βασίζονται σε δεδομένα που μπορεί να είναι παρωχημένα, καθώς καταγράφονται μόνο κατά τη διάρκεια των επισκέψεων.
- **Μη Επαρκής Καταγραφή Αυτο-Μετρήσεων:** Οι αυτο-καταγραφές του ασθενούς δεν ενοποιούνται πάντα με τις υπόλοιπες κλινικές πληροφορίες, γεγονός που μπορεί να οδηγήσει σε κενά στην παρακολούθηση.

Για την αντιμετώπιση των προαναφερθέντων περιορισμών προτείνεται η ανάπτυξη ενός πιλοτικού συστήματος καταγραφής που θα περιλαμβάνει τα εξής:

1. **Καταγραφή Μετρήσεων από τον Ασθενή:** Το σύστημα θα επιτρέπει στους ασθενείς να καταγράφουν καθημερινά μετρήσεις και δεδομένα χρησιμοποιώντας

έξυπνα κινητά τηλέφωνα. Αυτές οι μετρήσεις περιλαμβάνουν δεδομένα από απλές συσκευές όπως οξύμετρα και πιεσόμετρα, καθώς και πληροφορίες που καταγράφονται αυτομάτως από τον ίδιο τον ασθενή, όπως η κατανάλωση υγρών και ο όγκος ούρων.

2. **Πινακοποιημένη Προβολή Δεδομένων για τους Ιατρούς:** Τα δεδομένα θα προβάλλονται σε πινακοποιημένη μορφή μέσω υπολογιστών (desktop, laptop, tablet) και έξυπνων κινητών τηλεφώνων των ιατρών -μια γραμμή ανα ημερομηνία, μια στήλη ανά εξέταση/μέτρηση-. Η χρήση αυτής της μορφής παρουσίασης θα επιτρέπει τη γρήγορη και αποτελεσματική αξιολόγηση των ιατρικών δεδομένων.
3. **Οπτικοποίηση Δεδομένων:** Το σύστημα θα περιλαμβάνει εργαλεία για την οπτικοποίηση των μετρήσεων μέσω διαγραμμάτων και γραφημάτων, παρέχοντας στους ιατρούς μια πιο κατανοητή και ευσύνοπτη εικόνα των δεδομένων.

Οφέλη της Προτεινόμενης Προσέγγισης

Η υιοθέτηση του πιλοτικού συστήματος καταγραφής αναμένεται να προσφέρει τα εξής οφέλη:

- **Βελτιωμένη Παρακολούθηση:** Οι καθημερινές καταγραφές θα επιτρέψουν τη συνεχή παρακολούθηση της υγείας των ασθενών και την άμεση ανίχνευση αλλαγών.
- **Ακριβέστερη Αξιολόγηση:** Η άμεση και πλήρης καταγραφή δεδομένων θα βοηθήσει τους ιατρούς να κάνουν πιο ακριβείς και ενημερωμένες αποφάσεις για τη θεραπεία.
- **Αυξημένη Συμμετοχή Ασθενών:** Η δυνατότητα αυτοκαταγραφής και η εύκολη πρόσβαση στις πληροφορίες μέσω έξυπνων συσκευών θα ενθαρρύνει τη συμμετοχή των ασθενών στη διαχείριση της υγείας τους.

Η εισαγωγή ενός τέτοιου συστήματος θα μπορούσε να σημάνει μια σημαντική πρόοδο στην παρακολούθηση και την ενίσχυση της θεραπευτικής διαδικασίας για τους μεταμοσχευμένους ασθενείς, προσφέροντας μια πιο αποτελεσματική και ενδεδειγμένη λύση για την παρακολούθηση της υγείας τους.

Κεφάλαιο 2: Σχεδιασμός λύσης προβλήματος

Η ανάπτυξη ενός ολοκληρωμένου συστήματος για την παρακολούθηση των μεταμοσχευμένων ασθενών απαιτεί προσεκτικό σχεδιασμό και συνδυασμό τεχνολογιών για την επίτευξη αποτελεσματικής καταγραφής και παρουσίασης των ιατρικών δεδομένων. Το σύστημα περιλαμβάνει ουσιαστικά τρεις οντότητες. Μια εφαρμογή Android για τους ασθενείς, μια web εφαρμογή για τους ιατρούς, και μια ενδιάμεση οντότητα (server) για επικοινωνία των δυο προηγούμενων οντοτήτων με τη βάση δεδομένων.

2.1 Android App για Ασθενή

Η εφαρμογή Android θα αποτελεί το βασικό εργαλείο των ασθενών για την καταγραφή των ιατρικών τους δεδομένων. Οι κύριες λειτουργίες που επιθυμείται να περιλαμβάνει η εφαρμογή είναι:

- **Καταγραφή ημερήσιων μετρήσεων:** Η εφαρμογή θα επιτρέπει στους ασθενείς να εισάγουν ημερήσιες μετρήσεις, όπως θερμοκρασία σώματος, πίεση αίματος, δόση φαρμάκων, όγκος λήψης υγρών/ούρων κλπ. Αυτές οι μετρήσεις δεν καταχωρούνται υποχρεωτικά κάθε μέρα, αλλά η μικρότερη συχνότητα καταγραφής τους είναι η ημέρα. Η καταγραφή θα γίνεται όσο πιο απλά γίνεται, αφού η πολυπλοκότητα στη συγκεκριμένη περίπτωση δεν προσφέρει ουσιαστικό αποτέλεσμα.
- **Καταγραφή σχολίων:** Ο ασθενής μπορεί να εισάγει σχόλια ημέρας ή σχόλια ημερών. Αυτά τα σχόλια δίνουν τη δυνατότητα στο χρήστη να επικοινωνήσει, εμμέσως, με τον ιατρό ή την ιατρική ομάδα που τον παρακολουθεί.
- **Καταγραφή ετήσιων μετρήσεων:** Για τις ετήσιες μετρήσεις, ο χρήστης μπορεί με το πάτημα ενός κουμπιού να ενημερώσει ότι έκανε κάποια ετήσια εξέταση.

2.2 Web Εφαρμογή για Ιατρό

Η web εφαρμογή για τους ιατρούς θα παρέχει ένα σύνολο εργαλείων για την διαχείριση των ασθενών, την παρουσίαση και την οπτικοποίηση των δεδομένων που καταγράφονται από αυτούς. Τα εργαλεία αυτά είναι:

- **Πίνακας Ελέγχου/Διαχείρισης Ασθενών:** Η εφαρμογή θα παρέχει αναλυτικούς πίνακες ελέγχου με συνοπτικές πληροφορίες για κάθε ασθενή και οι ιατροί θα μπορούν να επεξεργαστούν τις πληροφορίες των ασθενών, να ανανεώσουν τα προσωπικά τους δεδομένα.
- **Ορισμός profile ασθενούς:** Ο ιατρός μπορεί να ορίζει το ποιες μετρήσεις θέλει ο κάθε ασθενής να συμπληρώνει, να θέτει οποιαδήποτε οδηγία εκείνος θέλει για την κάθε μέτρηση, καθώς και να υποδηλώνει την υποχρεωτικότητα (ή μη) συμπλήρωσης της κάθε μέτρησης.
- **Πινακοποιημένη προβολή των μετρήσεων/δεδομένων:** Θα περιλαμβάνει εργαλεία με τα οποία ο ιατρός θα μπορεί να δει λεπτομερώς όλες τις μετρήσεις σε μορφή πίνακα, με συγκεκριμένες ή μη, ημερομηνίες.
- **Οπτικοποίηση των μετρήσεων/δεδομένων:** Θα περιλαμβάνει εργαλεία για την οπτικοποίηση των μετρήσεων μέσω γραφημάτων και διαγραμμάτων, ώστε οι ιατροί να μπορούν να αναλύσουν τις τάσεις και τις αλλαγές στις μετρήσεις των ασθενών και να προσαρμόσουν τη θεραπεία τους αντίστοιχα.

2.3 Προσαρμόσιμο Σύστημα

Το σύστημα θα επιτρέπει την προσαρμογή των προφίλ των ασθενών, έτσι ώστε να ανταποκρίνεται στις ατομικές ανάγκες κάθε ασθενούς. Γι' αυτό κρίνεται απαραίτητη η εξής δυνατότητα:

- **Ορισμός profile ασθενούς:** Ο ιατρός, έπειτα από επικοινωνία με τους ασθενείς του, θα μπορεί να επιλέξει ποιες απ' όλες τις πιθανές μετρήσεις (τα πιθανά πεδία) τον ενδιαφέρουν να συμπληρώνει ο ασθενής. Έπειτα, μπορεί και προσαρμόζει τις οδηγίες του προς τον ασθενή, για κάθε πεδίο που έχει επιλέξει, καθώς και μπορεί να θέσει για το κάθε πεδίο προς συμπλήρωση ως υποχρεωτικό ή μη υποχρεωτικό.

2.4 Επεκτάσιμο Σύστημα

Για να διασφαλιστεί η επεκτασιμότητα και η εξέλιξη του συστήματος, θα υλοποιηθεί μια αρχιτεκτονική που υποστηρίζει την επέκταση των πεδίων καταγραφής και την εξελικτική ανάπτυξη του σχήματος της βάσης δεδομένων:

- **Σχήμα Βάσης Δεδομένων:** Το σύστημα θα σχεδιαστεί με μια ευέλικτη βάση δεδομένων που θα επιτρέπει την προσθήκη νέων πεδίων και τύπων δεδομένων χωρίς να επηρεάζεται η υφιστάμενη λειτουργικότητα.
- **Ενημερώσεις Συστήματος:** Το σύστημα θα παρέχει μηχανισμούς για την ενημέρωση και την εγκατάσταση νέων χαρακτηριστικών και πεδίων, διασφαλίζοντας την προσαρμογή στις εξελισσόμενες απαιτήσεις της υγειονομικής περίθαλψης.

2.5 Πιθανές Τεχνολογίες

Η υλοποίηση του συστήματος θα απαιτήσει τη χρήση διάφορων τεχνολογιών για να υποστηριχθούν οι λειτουργίες που υποδείχθηκαν παραπάνω. Οι τεχνολογίες που εξετάζονται περιλαμβάνουν:

- **Frontend:**
 - **Android App:** Java/Kotlin για την ανάπτυξη της εφαρμογής κινητού.
 - **Web Εφαρμογή:** React.js ή Angular για τη δημιουργία δυναμικών και διαδραστικών διεπαφών.
- **Backend:**
 - **Server Side:** Spring Boot για την ανάπτυξη και τη διαχείριση των υπηρεσιών και των APIs.
 - **Database:** MySQL ή PostgreSQL για την αποθήκευση και τη διαχείριση των δεδομένων. Σχεδιασμός βάσης δεδομένων με δυνατότητες εξέλιξης (schema evolution).
- **Data Visualization:**
 - **Data Visualization:** Εργαλεία για τη δημιουργία διαγραμμάτων και γραφημάτων (π.χ., D3.js, Chart.js).
- **Communication:**
 - **API:** RESTful APIs για την επικοινωνία μεταξύ του Android App, του Backend Server και της Web Εφαρμογής.

Η ολοκλήρωση αυτού του συστήματος θα επιτρέψει την αποτελεσματική και συνεχή παρακολούθηση των μεταμοσχευμένων ασθενών, βελτιώνοντας την ποιότητα της υγειονομικής περίθαλψης και την αποδοτικότητα των διαδικασιών παρακολούθησης.

Κεφάλαιο 3: Τεχνική ανάλυση συστήματος

3.1 Βάση Δεδομένων

Η τεχνική ανάλυση του συστήματος ξεκινά με την εξέταση του σχήματος της βάσης δεδομένων, το οποίο αποτελεί τη θεμελιώδη δομή για την αποθήκευση και διαχείριση των δεδομένων του συστήματος παρακολούθησης ασθενών. Παρακάτω παρουσιάζεται μια λεπτομερής ανάλυση των πινάκων και των σχέσεων μεταξύ τους.

Περιγραφή των Πινάκων

1. patient

- patient_id(PK): Μοναδικό αναγνωριστικό για κάθε ασθενή.
- first_name, second_name: Ονοματεπώνυμο του ασθενούς.
- date_of_birth: Ημερομηνία γέννησης.
- tel: Τηλέφωνο επικοινωνίας.
- email: Διεύθυνση email.
- last_transplant_date: Ημερομηνία τελευταίας μεταμόσχευσης.
- transplant_type: Τύπος μεταμόσχευσης.
- transplants_num: Αριθμός μεταμοσχεύσεων.
- kidney_failure_cause: Αιτία νεφρικής ανεπάρκειας.
- username, userpassword: username ασθενή (UNIQUE) και κωδικός

2. patient_profile (Ασθενής Οντότητα – Weak Entity)

- patient_id: Αναφορά στον ασθενή(FK, weak key).
- field_id: Αναφορά στο πεδίο καταγραφής(FK, weak key).
- required: Υποδεικνύει αν το πεδίο είναι υποχρεωτικό ή μη.

Το συγκεκριμένο πεδίο μπορεί να έχει τιμές “0,1” αντίστοιχα. Υπήρχε περίπτωση οι τιμές να είναι “0,1,2” (0 για disabled, 1 για προαιρετικό, 2 για υποχρεωτικό). Τελικά αποφασίστηκε ότι αυτό είναι περιττό, αφού δεν μας ενδιαφέρει να κρατάμε ιστορικό ενεργοποιημένων πεδίων, και τα απενεργοποιημένα πεδία δεν θα υπάρχουν σε αυτόν τον πίνακα.

- guideline: Οδηγία ιατρού για το πεδίο καταγραφής.

Εδώ ο ιατρός γράφει ό,τι θα ήθελε να ξέρει ο ασθενής για κάθε πεδίο, από δοσολογία (π.χ. 500mg την ημέρα) έως και τη συχνότητα εκχώρησης που θα ήταν εκείνος ικανοποιημένος (π.χ. 3 φορές την εβδομάδα)

3. **all_fields**

- **field_id**: Μοναδικό αναγνωριστικό για κάθε πεδίο(PK).
- **description_gr, description_en**: Περιγραφή του πεδίου στα ελληνικά και αγγλικά.
- **frequency**: Συχνότητα καταγραφής πεδίου/μέτρησης
Μπορεί να είναι (“daily,”“monthly”,”yearly”). Τα ημερήσια πεδία, δεν είναι απαραίτητο να εκχωρούνται ημερησίως αλλά η μικρότερη συχνότητα καταγραφής τους είναι η ημέρα.
- **measurement_unit**: Μονάδα μέτρησης.
- **acceptable_range**: Αποδεκτό εύρος τιμών.
- **type**: Τύπος δεδομένων πεδίου. Μπορεί να είναι “stored” ή “computed”. Τα πεδία που είναι stored εννοείται ότι αποθηκεύονται πάνω στη βάση, ενώ τα πεδία που είναι computed δεν αποθηκεύονται, αλλά υπολογίζονται με βάση το expression.
- **expression**: Κανονική έκφραση/τύπος ή NULL για τον υπολογισμό. Με βάση την κανονική έκφραση υπολογίζονται οι τιμές των “computed” πεδίων
- **order_id**: Σειρά εμφάνισης στην πινακοποιημένη προβολή, έπειτα από υποδείξεις των υγειονομικών.

4. **eav (Ασθενής οντότητα – Weak Entity)**

- **patient_id**: Αναφορά στον ασθενή (FK, weak key).
- **exam_date**: Ημερομηνία εξέτασης (weak key).
- **field_id**: Αναφορά στο πεδίο καταγραφής(FK, weak key).
- **val**: Τιμή καταγραφής. Έχει επιλεγεί η τιμή να είναι τύπου “varchar” και όχι float όπως ήταν στο αρχικό στάδιο στάδιο της υλοποίησης, για πιο σωστή καθολική χρήση του μοντέλου EAV που εξηγείται στο επόμενο υποκεφάλαιο

5. **yearly_fields(Ασθενής οντότητα – Weak Entity)**

- **patient_id**: Αναφορά στον ασθενή(FK, weak key).
- **yeardate**: Έτος καταγραφής(weak key).
- **heart_triplex, abdominal_us, dxa**: Ετήσιες εξετάσεις (καρδιογράφημα, υπερηχογράφημα κοιλίας, DXA). (τιμές 0 και 1)

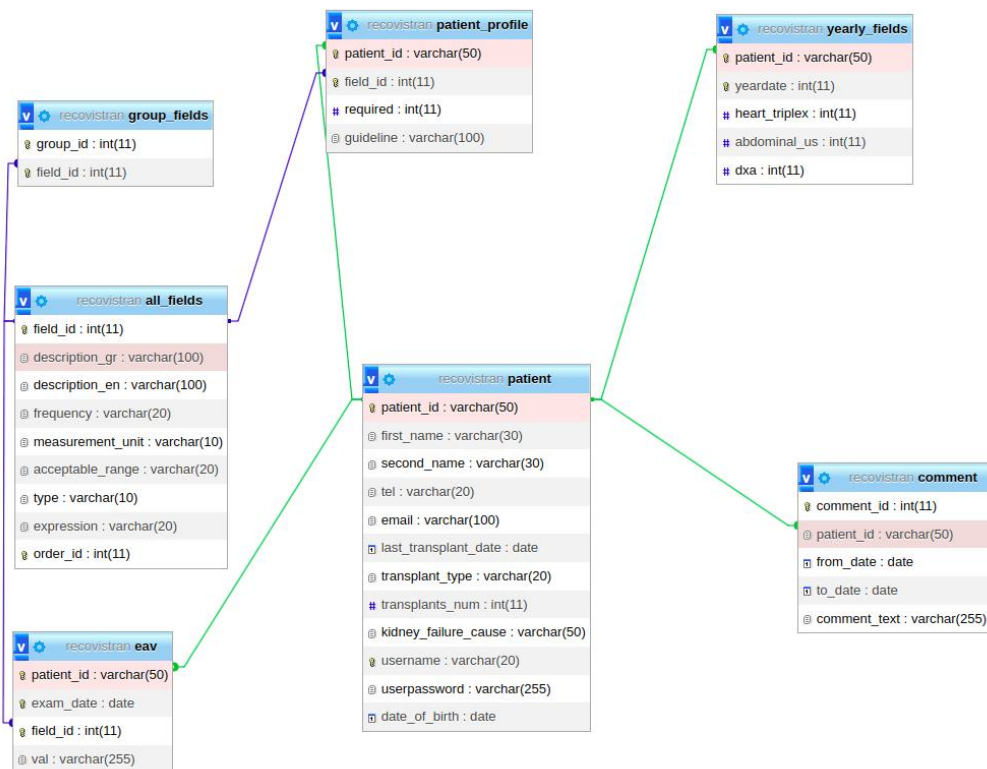
6. comment

- comment_id: Μοναδικό αναγνωριστικό για κάθε σχόλιο(PK).
- patient_id: Αναφορά στον ασθενή(FK).
- from_date, to_date: Περίοδος σχολιασμού.
- comment_text: Κείμενο σχολίου. Στον πίνακα αυτόν αποθηκεύονται τα σχόλια που κάνει ο χρήστης μέσω της εφαρμογής κινητού. Αν το from_date, to_date έχουν τιμή NULL, τότε πρόκειται για σχόλιο ημέρας, το οποίο υπάρχει σαν πεδίο στον πίνακα all_fields με field_id=13 και η ημερομηνία του σχολίου ημέρας υπάρχει στον πίνακα eav.

7. group_fields

- group_id: Μοναδικό αναγνωριστικό για κάθε ομάδα πεδίων(PK).
- field_id: Αναφορά στο πεδίο καταγραφής(PK,FK).

Ο πίνακας αυτός χρησιμοποιείται για να ομαδοποιήσει κάποια πεδία. Πιο συγκεκριμένα, η ομαδοποίηση γίνεται με τη μορφή “Ουσία αίματος – φάρμακο”. Για παράδειγμα “Επίπεδα Tacrolimus στο αίμα(Tacrolimus levels – Prograf”, “Επίπεδα Tacrolimus στο αίμα – Avagraf” ή “Επίπεδα Μυκοφαιλονάτης(mycophenolate levels) - Cellcept”



Εικόνα 1: Διάγραμμα Entity-Relationship βάσης δεδομένων

Ανάλυση Λειτουργικότητας

Η βάση δεδομένων έχει σχεδιαστεί για να υποστηρίζει τις ακόλουθες λειτουργικότητες:

- **Καταγραφή Δεδομένων:** Τα δεδομένα υγείας των ασθενών καταγράφονται στον πίνακα `ean`, ο οποίος επιτρέπει την αποθήκευση οποιουδήποτε τύπου δεδομένων με βάση τα πεδία που έχουν οριστεί στον πίνακα `all_fields`.
- **Προφίλ Ασθενών:** Ο πίνακας `patient_profile` επιτρέπει την προσαρμογή των πεδίων που είναι υποχρεωτικά ή προαιρετικά για κάθε ασθενή, με ταυτόχρονη την εγγραφή και κάποιας οδηγίας για το κάθε πεδίο, ανάλογα με την απόφαση του ιατρού.
- **Ετήσιες Εξετάσεις:** Ο πίνακας `yearly_fields` διατηρεί τις ετήσιες εξετάσεις για κάθε ασθενή, διευκολύνοντας την παρακολούθηση μακροπρόθεσμων τάσεων.
- **Σχόλια και Παρατηρήσεις:** Ο πίνακας `comment` επιτρέπει στους ασθενείς να αφήνουν τα σχόλια και τις παρατηρήσεις για συγκεκριμένες περιόδους ή για συγκεκριμένη ημερομηνία, παρέχοντας επιπλέον πληροφορίες για την κατάσταση του ασθενούς.
- **Ομαδοποίηση πεδίων:** Με την ομαδοποίηση κάποιων εκ των πεδίων, δηλαδή με τον πίνακα `group_fields`, απλουστεύουμε και συγκεκριμενοποιούμε τις επιλογές που δίνουμε στον ιατρό, αφού γίνεται ταυτόχρονη επιλογή των πεδίων που βρίσκονται στην ίδια ομάδα.

3.2 Μοντέλο Entity-Attribute-Value ως πίνακας αποθήκευσης δεδομένων

Η επιλογή της μεθόδου Entity-Attribute-Value (EAV) για την αποθήκευση δεδομένων στη μελέτη μας, προσφέρει μια σειρά από πλεονεκτήματα και χαρακτηριστικά που την καθιστούν κατάλληλη για την καταγραφή και διαχείριση ποικίλων και δυναμικών δεδομένων υγείας. Σε αυτή την ενότητα, θα εξηγήσουμε το μοντέλο EAV, τους λόγους επιλογής του, τα πλεονεκτήματα και τις προκλήσεις που μπορεί να προκύψουν κατά την υλοποίησή του.

3.2.1 Μοντέλο Entity-Attribute-Value (EAV)

Το μοντέλο EAV είναι μια αρχιτεκτονική αποθήκευσης δεδομένων που χρησιμοποιείται συχνά για την αποθήκευση πολύπλοκων και ετερογενών δεδομένων. Σε αυτό το μοντέλο, τα δεδομένα αποθηκεύονται σε τρεις βασικές κατηγορίες:

- **Οντότητες (Entities):** Οι κύριες μονάδες δεδομένων που θέλουμε να καταγράψουμε. Στο πλαίσιο της εφαρμογής μας, οι οντότητες είναι οι ασθενείς ανά ημερομηνία. (patient_id,exam_date)
- **Χαρακτηριστικά (Attributes):** Οι ιδιότητες ή τα πεδία που περιγράφουν τις οντότητες. Αυτά μπορεί να περιλαμβάνουν μετρήσεις όπως τη θερμοκρασία σώματος, αρτηριακή πίεση κλπ. (field_id)
- **Τιμές (Values):** Οι πραγματικές τιμές των χαρακτηριστικών για κάθε οντότητα. (val)

Αυτός ο σχεδιασμός επιτρέπει την αποθήκευση ποικίλων δεδομένων με διαφορετικές δομές και τη δυναμική προσαρμογή των χαρακτηριστικών χωρίς την ανάγκη αλλαγής της δομής της βάσης δεδομένων.

3.2.2 Λόγοι Επιλογής του Μοντέλου EAV

Η επιλογή του μοντέλου EAV βασίζεται σε αρκετούς λόγους, οι οποίοι καθιστούν αυτή τη μέθοδο ιδιαίτερα κατάλληλη για την εφαρμογή μας:

- **Ευελξία και επεκτασιμότητα:** Το EAV είναι ιδιαίτερα ευέλικτο, καθώς οι τιμές υποστηρίζουν διάφορους τύπους αντικειμένων-δεδομένων, ακόμα και αν η μορφή τους είναι διαφορετική.
- **Σταθερότητα με τις ενημερώσεις του μοντέλου:** Υπάρχει ευελιξία σε σχέση με την εξέλιξη των πηγών δεδομένων. Νέες εκδόσεις σχημάτων που βασίζονται στην XML ή σε κάποια σχεσιακή βάση δεδομένων δεν απαιτούν καμία τροποποίηση στη δομή του EAV.
- **Δεν χάνονται δεδομένα,** αφού δεν απαιτείται κανένας μετασχηματισμός τους για να αποθηκευτούν με αυτό το μοντέλο. Αυτό είναι ιδιαίτερα σημαντικό σε ιατρικές εφαρμογές,σαν αυτή που πραγματεύεται αυτή η μελέτη, που τα δεδομένα είναι ευαίσθητα.

3.2.3 Προκλήσεις και πιθανές λύσεις για το μοντέλο EAV

Παρά τα πλεονεκτήματα που αναφέρθηκαν, το μοντέλο αυτό παρουσιάζει και ορισμένες προκλήσεις:

- **Αποδοτικότητα:** Η αποθήκευση και ανάκτηση δεδομένων μπορεί να είναι πιο αργή σε σύγκριση με άλλες μεθόδους λόγω των πολλαπλών ενώσεων (joins) μεταξύ πινάκων. Αυτό ισχύει ιδιαίτερα για μεγάλο όγκο δεδομένων.

Λύση: Χρήση ευρετηρίων (indexes) και βελτιστοποίηση των ερωτημάτων SQL για τη βελτίωση της απόδοσης.

- **Σύνθετα Ερωτήματα:** Η σύνταξη ερωτημάτων μπορεί να γίνει περίπλοκη, ιδιαίτερα όταν πρέπει να ανακτηθούν πολλά χαρακτηριστικά ταυτόχρονα.

Λύση: Ανάπτυξη αποθηκευμένων διαδικασιών (stored procedures) και χρήση εργαλείων ανάλυσης δεδομένων για την απλοποίηση των ερωτημάτων.

- **Ακεραιότητα Δεδομένων:** Διασφάλιση της ακρίβειας και της ακεραιότητας των δεδομένων, ιδιαίτερα όταν προστίθενται νέα χαρακτηριστικά.

Λύση: Χρήση περιορισμών (constraints) και κανόνων επικύρωσης (validation rules) για την αποφυγή ασυνεπειών στα δεδομένα.

3.3 Δημιουργία καθολικού πίνακα εξετάσεων

Ο καθολικός πίνακας εξετάσεων προκύπτει από πολλαπλά LEFT JOIN και θα μπορούσε να γίνει στη βάση δεδομένων με stored procedure, αλλά αφού έχει επιλεγεί η χρήση του Spring, τότε αυτή η υλοποίηση έχει γίνει σε Java. Ακολουθούν εικόνες – κώδικα σε MySQL και σε Java αντίστοιχα. Η μόνη διαφορά είναι ότι στον κώδικα Java υπάρχει και η διαχείριση των “computed” πεδίων καθώς και των ημερήσιων σχολίων (field_id = 13)

```
SELECT
  p.patient_id,
  p.first_name,
  p.second_name,
  p.last_transplant_date,
  af.field_id,
  af.description_gr,
  af.frequency,
  af.measurement_unit,
  af.acceptable_range,
  e.exam_date,
  e.val,
  c.comment_text
FROM
  patient p
LEFT JOIN
  patient_profile pp ON p.patient_id = pp.patient_id
LEFT JOIN
  eav e ON p.patient_id = e.patient_id
LEFT JOIN
  all_fields af ON e.field_id = af.field_id
LEFT JOIN
  group_fields gf ON af.field_id = gf.field_id
LEFT JOIN
  comment c ON p.patient_id = c.patient_id
ORDER BY
  p.patient_id, e.exam_date;
```

Εικόνα 2: Κώδικας SQL δημιουργίας καθολικού πίνακα εξετάσεων

```
public List<Eva> getAllCommentsBy patient_id, String start_date, String end_date) {
    //get overall values
    List<Eva> allEvas = getAllEvas();
    List<Eva> patientEvas = new ArrayList<>();
    Date startDate = Date.valueOf(start_date);
    Date endDate = Date.valueOf(end_date);
    // get overall values
    for (int i = 0; i < allEvas.size(); i++) {
        Eva entry = allEvas.get(i);
        Data entryData = entry.get(1).getExam_data();
        if (patient_id.equals(entry.get(1).getPatient_id()) &&
            (entryData.isEntryComment() && entryData.after(startDate) || entryData.equals(startDate) || entryData.equals(endDate))) {
            // it's the entry comment
            if (entry.get(1).getField_id() == 33) {
                entry.set(1,commentRepository.findById(Integer.parseInt(entry.get(1).get(1).getComment_text()));
            }
            patientEvas.add(entry);
        }
    }

    List<AllFields> allFields = new ArrayList<>();
    Streamable.of(allFieldsRepository.findAll()).forEach(allFields::add);

    // create a map from field_id to order_id
    Map<Integer, Integer> fieldIdToOrderId = allFields.stream()
        .collect(Collectors.toMap(AllFields::getField_id, AllFields::getOrder_id));

    //match patterns
    List<Pattern> patterns = new ArrayList<>();
    patterns.add(Pattern.compile("^\\d+\\.\\d+/(\\d+)/(\\d+)$"));
    patterns.add(Pattern.compile("^\\d+\\.\\d+/(\\d+)$"));
    List<Eva> newEntries = new ArrayList<>();

    //loop through values
    for (int i = 0; i < allFields.size(); i++) {
        // for regular daily status
        if (allFields.get(i).getType().equals("computer")) {
            //get the expression
            String expression = allFields.get(i).getExpression();
            //break the expression
            for (int j = 0; j < patterns.size(); j++) {
                Matcher matcher = patterns.get(j).matcher(expression);
                if (matcher.matches()) {
                    System.out.println("Match found for pattern " + j + " in expression: " + expression);

                    //Percentage - get pattern
                    if (matcher.groupCount() == 3) {
                        System.out.println("Percentage");
                        int field_id_1 = Integer.parseInt(matcher.group(1));
                        int field_id_2 = Integer.parseInt(matcher.group(2));
                        int multiplier = Integer.parseInt(matcher.group(3));
                        System.out.println(field_id_1);

                        if (allFields.get(i).getFrequency().equals("daily")) {
                            for (int k = 0; k < patientEvas.size(); k++) {
                                EvaID evaID_1 = new EvaID(patient_id, patientEvas.get(k).get(1).getExam_data().getField_id_1);
                                EvaID evaID_2 = new EvaID(patient_id, patientEvas.get(k).get(1).getExam_data().getField_id_2);
                                // if they both exist then we can calculate the daily percentage
                                if (evaRepository.findById(evaID_1).isPresent()
                                    && evaRepository.findById(evaID_2).isPresent()) {
                                    //calculate the expression value
                                    float expressionValue = Float.parseFloat(evaRepository.findById(evaID_1).get(1).getVal(1))
                                        / Float.parseFloat(evaRepository.findById(evaID_2).get(1).getVal(1))
                                        * multiplier;
                                    //add to the list
                                    EvaID createdID = new EvaID(patient_id, patientEvas.get(k).get(1).getExam_data().allFields.get(i).getField_id());
                                    Eva createdEntry = new Eva(createdID, String.valueOf(Math.round(expressionValue)),
                                        patientRepository.findById(patient_id).get(),
                                        allFields.get(i));
                                    newEntries.add(createdEntry);
                                }
                            }
                        } else if (allFields.get(i).getFrequency().equals("monthly")) {
                            System.out.println("Monthly Percentage");

                            Map<YearMonth, List<Eva>> monthlyEntries = new HashMap<>();
                            Map<YearMonth, LocalDate> lastDayOfMonthValue = new HashMap<>();

                            for (Eva entry : allEvas) {
                                if (patient_id.equals(entry.get(1).getPatient_id()) &&
                                    entry.get(1).getField_id() == field_id_1) {
                                    LocalDate entryDate = entry.get(1).getExam_data().toLocalDate();
                                    YearMonth yearMonth = YearMonth.from(entryDate);
                                    monthlyEntries.computeIfAbsent(yearMonth, k -> new ArrayList<>()).add(entry);

                                    if (!lastDayOfMonthValue.containsKey(yearMonth)) {
                                        entryDate.isAfter(lastDayOfMonthValue.get(yearMonth)) {
                                            lastDayOfMonthValue.put(yearMonth, entryDate);
                                        }
                                    }
                                }
                            }

                            for (Map.Entry<YearMonth, List<Eva>> monthEntry : monthlyEntries.entrySet()) {
                                YearMonth yearMonth = monthEntry.getKey();
                                List<Eva> entries = monthEntry.getValue();

                                float sum1 = 0.0f;
                                float sum2 = 0.0f;

                                for (Eva entry : entries) {
                                    EvaID evaID_1 = new EvaID(patient_id, entry.get(1).getExam_data().getField_id_1);
                                    EvaID evaID_2 = new EvaID(patient_id, entry.get(1).getExam_data().getField_id_2);

                                    if (evaRepository.findById(evaID_1).isPresent()
                                        && evaRepository.findById(evaID_2).isPresent()) {
                                        sum1 += Float.parseFloat(evaRepository.findById(evaID_1).get(1).getVal(1));
                                        sum2 += Float.parseFloat(evaRepository.findById(evaID_2).get(1).getVal(1));
                                    }
                                }

                                if (sum2 != 0) {
                                    float monthlyPercentage = sum1 / sum2 * multiplier;
                                    LocalDate lastDay = lastDayOfMonthValue.get(yearMonth);
                                    EvaID createdID = new EvaID(patient_id, Date.valueOf(lastDay), allFields.get(i).getField_id());
                                    Eva createdEntry = new Eva(createdID, String.valueOf(Math.round(monthlyPercentage)),
                                        patientRepository.findById(patient_id).get(),
                                        allFields.get(i));

                                    if (createdEntry.get(1).equals("P")) {
                                        newEntries.add(createdEntry);
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }

    //count number of the pattern
    etc if (matcher.groupCount() is 2) {
        System.out.println("Count check");
    }
}

// because we added entries to the end of patientEvas
// sort the patientEvas by order_id
patientEvas.addAll(newEntries);
patientEvas.sort((Eva1, Eva2) -> {
    int orderId1 = fieldIdToOrderId.getOrDefault(eva1.get(1).getField_id(), Integer.MAX_VALUE);
    int orderId2 = fieldIdToOrderId.getOrDefault(eva2.get(1).getField_id(), Integer.MAX_VALUE);
    return Integer.compare(orderId1, orderId2);
});

return patientEvas;
}
```

Εικόνα 3: Κώδικας Java δημιουργίας καθολικού πίνακα εξετάσεων

3.4 Σύντομη αναφορά σε πιθανή εναλλακτική λύση

Παρόλο που το τρέχον σύστημα είναι πλήρες, χρησιμοποιώντας εφαρμογή Android για τους ασθενείς και Spring Boot για το backend και React.js για το frontend, είναι κρίσιμο να αναφερθεί και να εξεταστεί μια πιο παραδοσιακή, εναλλακτική προσέγγιση χρησιμοποιώντας PHP, Apache, μια σχεσιακή βάση δεδομένων, JavaScript, και Google Charts για την απεικόνιση δεδομένων.

1. Στην πλευρά του server/backend, θα χρησιμοποιηθεί PHP και Apache. Η PHP είναι μια δημοφιλής γλώσσα για την ανάπτυξη web εφαρμογών. Ο Apache είναι αξιόπιστος και ασφαλής, ανοιχτού κώδικα, server που υποστηρίζει PHP. Θα πρέπει να γίνει ανάπτυξη των PHP scripts για τη διαχείριση εισόδου, επεξεργασίας και αποθήκευσης των δεδομένων των ασθενών. Αυτά, θα αλληλεπιδρούν με τη βάση για την εκτέλεση των CRUD λειτουργιών.
 - Το θετικό της προσέγγισης αυτής είναι η απλότητα και ευκολία της PHP , ενώ το αρνητικό είναι στην απόδοση(ταχύτητα)
2. Όσον αφορά τη Βάση Δεδομένων, θα χρησιμοποιηθεί η MySQL, γνωστή για την αξιοπιστία και την απόδοση της και θα γίνει σχεδίαση παρόμοια με την υπάρχουσα. (υλοποίηση πινάκων αποθήκευσης πληροφοριών των ασθενών, εξετάσεων κλπ)
3. Στην client πλευρά θα χρησιμοποιηθεί Javascript που είναι η πλέον πιο διαδεδομένη, εύελικτη γλώσσα που επιτρέπει δυναμικό περιεχόμενο και αλληλεπίδραση σε δυναμικές σελίδες. Η επικοινωνία με το backend(PHP) θα γίνει μέσω των AJAX requests)
4. Google charts για την απεικόνιση των δεδομένων. Είναι ένα ισχυρό εργαλείο για τη δημιουργία γραφημάτων. Περιέχει πάρα πολλές ρυθμίσεις και επιλογές, καθιστώντας το ιδανικό για την απεικόνιση ιατρικών δεδομένων

Η εναλλακτική λύση με PHP, Apache, σχεσιακή ΒΔ, JavaScript και Google Charts αποτελεί μια βιώσιμη και αξιόπιστη προσέγγιση. Παρέχει απλότητα και ευελιξία, ενώ οι εκτενείς δυνατότητες των Google Charts για την απεικόνιση δεδομένων ενισχύουν την εμπειρία του χρήστη. Ωστόσο, η υφιστάμενη λύση με Spring Boot και React.js προσφέρει υψηλότερη απόδοση, καλύτερη κλιμάκωση και μια πιο σύγχρονη διεπαφή χρήστη, καθιστώντας την προτιμότερη επιλογή για απαιτητικές εφαρμογές. Επίσης, προσφέρει μεγαλύτερη ελευθερία του λογισμικού και των δημιουργών του από παρόχους εργαλείων ανάπτυξης και από διαρκείς αναβαθμίσεις

3.5 Εφαρμογή Android:

- **Περιβάλλον Ανάπτυξης:** Χρησιμοποιείται το Android Studio Hedgehog (Έκδοση 2023.1.1) ως το ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) για την ανάπτυξη της Android εφαρμογής.
- **Γλώσσα Προγραμματισμού:** Χρησιμοποιείται η Java 17.0.7 για τον προγραμματισμό.
- **Απαιτούμενη έκδοση SDK- στοχευμένη έκδοση SDK:** Η ελάχιστη υποστηριζόμενη έκδοση SDK είναι η 28 αλλά η εφαρμογή είναι προορισμένη για την έκδοση 34.
- **Βιβλιοθήκες Δικτύου:** Χρησιμοποιούνται οι Retrofit2 (Έκδοση 2.9.0) και OkHttp3 (Έκδοση 4.9.1) για την αποστολή HTTP αιτημάτων στον Spring Boot server και τη διαχείριση της client επικοινωνίας, αντίστοιχα. Επιπλέον, χρησιμοποιείται η FasterXML Jackson (Έκδοση 2.16.0) για τη JSON σειριοποίηση(serialization) και αποσειριοποίηση(deserialization).
- **Βάση Δεδομένων:** Η εφαρμογή αλληλεπιδρά με μια βάση δεδομένων MySQL που φιλοξενείται σε έναν Spring Boot server. Αποστέλλει και λαμβάνει δεδομένα χρησιμοποιώντας το Retrofit2 για την επικοινωνία με το RESTful API(Spring Boot server).

3.6 Spring Boot Server:

- **Περιβάλλον Ανάπτυξης:** Χρησιμοποιείται το IntelliJ IDEA (Έκδοση 2024.1.4) ως το ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) για την ανάπτυξη του Spring Boot server.
- **Έκδοση Java:** Ο server αναπτύσσεται χρησιμοποιώντας Spring Boot και αυτή τη στιγμή λειτουργεί με την Java-openjdk 17.0.11
- **Έκδοση Spring:** Το project τρέχει με την έκδοση Spring 3.2.0
- **Έκδοση Gradle:** Χρησιμοποιείται η έκδοση 8.5
- **Συνδεσιμότητα Βάσης Δεδομένων:** Χρησιμοποιείται το MySQL Connector for Java (Έκδοση 8.0.33) για τη σύνδεση με τη βάση δεδομένων MySQL.
- **Spring Boot Starter Data JPA (Hibernate):** Το Spring Boot Starter Data JPA (Hibernate) είναι υπεύθυνο για τη διαχείριση της σύνδεσης με τη βάση δεδομένων και χρησιμοποιείται για την απλοποίηση της ανάπτυξης λειτουργιών CRUD.

- **RESTful API:** Το Spring Boot παρέχει τη δυνατότητα δημιουργίας RESTful APIs. Ο server παρέχει endpoints με τα οποία αλληλεπιδρά η Android εφαρμογή του ασθενούς και η web εφαρμογή του ιατρού, για την εκτέλεση διάφορων ενεργειών, όπως η ανάκτηση του profile του ασθενούς(στην πρώτη περίπτωση) και η πινακοποιημένη αποτύπωση των μετρήσεων του(στη δεύτερη περίπτωση).

3.6.1 Προδιαγραφές λειτουργίας του server

Ο Spring boot server, ουσιαστικά είναι το Spring MVC framework, με αυτόματο όμως configuration και με embedded server(στη συγκεκριμένη περίπτωση Tomcat) .

Ο τρόπος που λειτουργεί το Spring MVC είναι ο εξής:

1. Όλες οι αιτήσεις στέλνονται σε έναν κεντρικό Servlet
2. Το κεντρικό Servlet στέλνει τις αιτήσεις στον Handler
3. Ο Handler κάνει την αντιστοίχιση και στέλνει το αίτημα στον σωστό Controller (RestController στη δικιά μας περίπτωση, αφού ο server τρέχει σαν RESTful API).
4. Ο RestController διαχειρίζεται την αίτηση, εκτελεί τον κώδικα και αλληλεπιδρά με τα μοντέλα(data models) για να επεξεργαστεί τα δεδομένα
5. Ο RestController επιστρέφει την απάντηση, η οποία μπορεί να είναι σε μορφή JSON ή XML(στη δική μας περίπτωση είναι JSON)
6. Η απάντηση πηγαίνει στον κεντρικό Servlet ο οποίος στέλνει την απάντηση πίσω στον πελάτη

Ουσιαστικά, ο Spring Boot Server δουλεύει (εσωτερικά) με Servlet και τελικά παράγει Web Service(RESTful web service).

Οι λόγοι που δεν χρησιμοποιήθηκε απλώς Spring είναι οι εξής:

- Ευκολότερο configuration
- Γρήγορο setup, χρησιμοποιώντας το Spring Initializr
- Embedded Tomcat server

3.7 Εφαρμογή Ιατρού (Frontend):

- **Περιβάλλον Ανάπτυξης:** Χρησιμοποιείται το IntelliJ IDEA (Έκδοση 2023.3.4) ως το ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) για την ανάπτυξη του frontend της εφαρμογής web του ιατρού.
- **Βιβλιοθήκη JavaScript:** Χρησιμοποιείται η ReactJS (Έκδοση 18.2.0) για την κατασκευή του frontend.
- **Στοιχεία Διεπαφής Χρήστη:** Χρησιμοποιείται η Reactstrap (Έκδοση 9.2.1), διευκολύνοντας τη δημιουργία ευέλικτων και οπτικά ελκυστικών στοιχείων διεπαφής χρήστη.
- **Βιβλιοθήκη Γραφημάτων:** Χρησιμοποιείται η Chart.js (Έκδοση 4.4.1) για τη δημιουργία διαφόρων τύπων γραφημάτων για την οπτικοποίηση των ιατρικών δεδομένων.
- **Συνδεσιμότητα Backend:** Η εφαρμογή frontend επικοινωνεί με τον διακομιστή Spring Boot (RESTful API) για την ανάκτηση και εμφάνιση δεδομένων.

Κεφάλαιο 4: Το αποτέλεσμα – παραθθέν σύστημα

4.1 Λογισμικό πρόγραμμα για τον επιβλεπόμενο ασθενή

Ο ασθενής βλέπει στο κινητό του και καταχωρεί (εικόνες 4 και 5) τα στοιχεία που έχει ορίσει ο Επιβλέπων Ιατρός στο profile του ασθενούς. Υπάρχουν τρία «Tab» με ονομασίες «Ημερήσιες», «Σχόλιο Ημερών» και «Ετήσιες». Ο ασθενής, στο Tab «Ημερήσιες», πρώτα βλέπει (κάτω από την ένδειξη «Required Fields») όσα ο Ιατρός τα έχει ορίσει ως υποχρεωτικά (Mandatory στο profile του ασθενούς). Μετά βλέπει (κάτω από την ένδειξη «Optional Fields») όσα ο Ιατρός τα έχει ορίσει ως προαιρετικά (Optional στο profile του ασθενούς). Για κάθε πεδίο που συμπληρώνει, ο ασθενής βλέπει (κάτω από το πεδίο) την προσωποποιημένη για τον ασθενή οδηγία που έχει ορίσει ο Επιβλέπων Ιατρός στο Profile του ασθενούς. Επίσης βλέπει (πάνω από το πεδίο) την οδηγία που υπάρχει για το πεδίο (εξέταση, μέτρηση, δείκτη) ανεξαρτήτως του συγκεκριμένου ασθενούς.

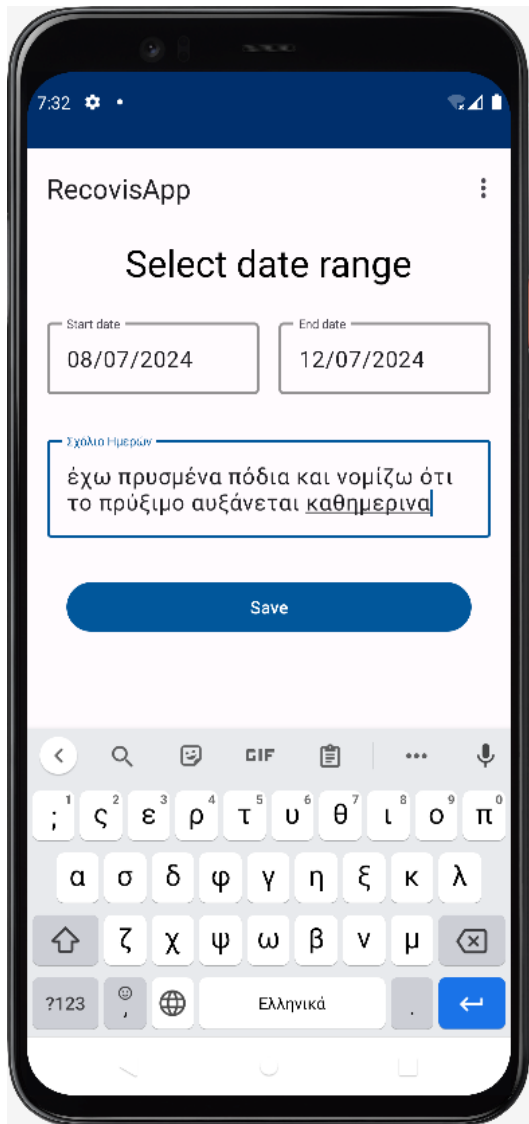
The screenshot shows the 'Required Fields' section of the mobile app. It contains four input fields: 'Θερμοκρασία Σώματος' (°C), 'Πίεση Σ...' (mmHg), 'Τακρόλι...' (ng/ml), and 'Prograf' (mg/day). Below these is the 'Optional Fields' section with 'Σφυγμός' (bpm) and 'Κρεατιν...' (mg/dL). A 'Σχόλιο Ημέρας' field is also present. A blue 'Save' button is at the bottom. The bottom navigation bar shows 'Ημερήσιες', 'Σχόλιο Ημερών', and 'Ετήσιες'.

Εικόνα 4: Tab «Ημερήσιες» πριν την καταχώρηση

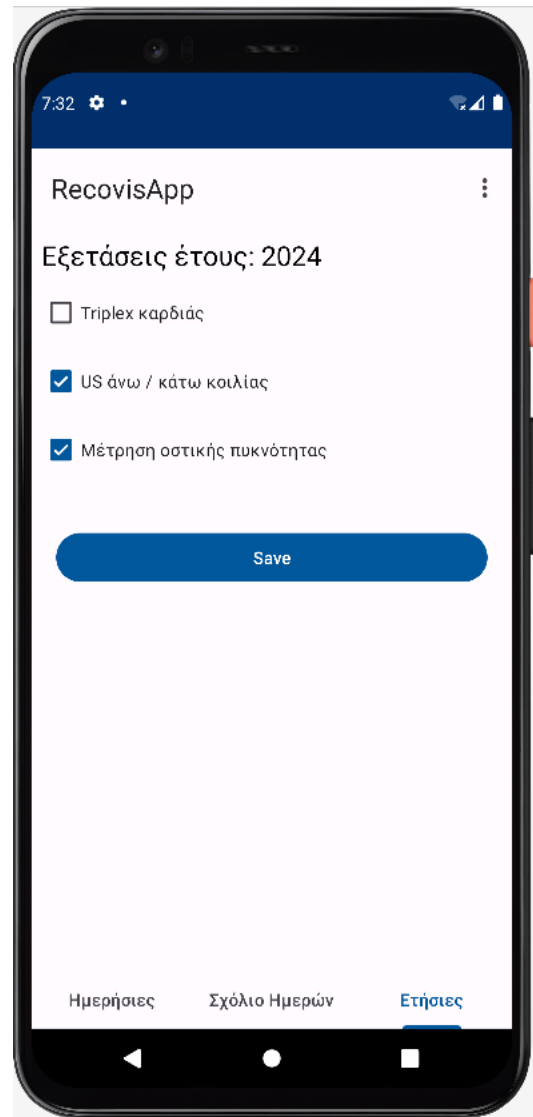
The screenshot shows the 'Required Fields' section of the mobile app with data entered. The 'Θερμοκρασία Σώματος' field contains '37,1' and the 'Πίεση Συστολική' field contains '120'. The 'Normal range: 100 - 130' is displayed above the blood pressure field, and '2 φορές την εβδομάδα' is displayed below it. The 'Τακρόλι...' field contains '4' and the 'Prograf' field contains '4'. The 'Optional Fields' section shows 'Σφυγμός' as '75' and 'Κρεατινίνη' as '1,6'. The 'Σχόλιο Ημέρας' field contains 'έχω 3 ημέρες δέκατα'. A blue 'Save' button is at the bottom. The bottom navigation bar shows 'Ημερήσιες', 'Σχόλιο Ημερών', and 'Ετήσιες'.

Εικόνα 5: Tab «Ημερήσιες» κατά την καταχώρηση

Ο ασθενής, στο Tab «Σχόλιο Ημερών», συνήθως καταχωρεί κάποια πληροφορία από αυτές που έχει εκπαιδευθεί (συνήθως μετά την μεταμόσχευση) πως πρέπει να γνωστοποιεί επειγόντως (βλέπε εικόνα 6). Στο Tab «Ετήσιες» ο ασθενής επιβεβαιώνει την πραγματοποίηση των ετησίων εξετάσεων, όταν αυτές γίνουν (βλέπε εικόνα 7).



Εικόνα 6: Tab «Σχόλιο Ημερών»




Εικόνα 7: Tab «Ετήσιες»

4.2 Λογισμικό πρόγραμμα για τον εποπτεύοντα ιατρό

Όλες οι λειτουργίες που μπορεί να πραγματοποιήσει κάθε συμμετέχων εποπτεύων ιατρός στην κεντρική σελίδα του λογισμικού (βλέπε Εικόνα 8).

Από την κεντρική σελίδα του λογισμικού ο εποπτεύων ιατρός μπορεί:

- να δει λίστα ασθενών (και για καθέναν από αυτούς να ορίσει το προφίλ του ή να ενημερώσει τα βασικά του στοιχεία),
- να δει πινακοποιημένα τα καταγεγραμμένα στοιχεία του ασθενούς,
- να παράγει πεπλεγμένα ή μη, διαγράμματα (“Charts”) στη διάρκεια μιας χρονικής περιόδου για μία ή περισσότερες εξετάσεις/μετρήσεις/δείκτες ενός ασθενούς



Home
[Patients List](#)
[Patient Exams](#)
[Charts](#)

Εικόνα 8: Κεντρική σελίδα του λογισμικού

Παράδειγμα λίστας ασθενών βλέπουμε στην Εικόνα 9. Από τη λίστα ασθενών ο εποπτεύων Ιατρός μπορεί για κάθε ασθενή να ορίσει/ενημερώσει το προφίλ του (Edit Profile) ή να ενημερώσει τα βασικά του στοιχεία (Edit Patient).

First Name	Second Name	Actions
Anonymized	Patient01	Edit Profile Edit Patient
Anonymized	Patient02	Edit Profile Edit Patient
Anonymized	Patient03	Edit Profile Edit Patient
Anonymized	Patient04	Edit Profile Edit Patient
Anonymized	Patient05	Edit Profile Edit Patient
Manos	Grigorakis	Edit Profile Edit Patient
Nikos	Nikoglou	Edit Profile Edit Patient
Kostas	Kostoglou	Edit Profile Edit Patient

Εικόνα 9: Λίστα Ασθενών

Παράδειγμα ενημέρωσης βασικών στοιχείων ασθενούς (Edit Patient) βλέπουμε στην εικόνα 10.

Home

Edit Patient

Patient ID
AnonymizedPatient01

First Name
Anonymized

Second Name
Patient01

Date of Birth
mm/dd/yyyy

Email

Last Transplant Date
mm/dd/yyyy

Transplant Type Alive

Number of Transplants

Kidney Failure Cause

Username
anonymized01

Password

[Save](#) [Cancel](#)

Εικόνα 10: Ενημέρωση βασικών στοιχείων ασθενούς

Παράδειγμα ορισμού/ενημέρωσης του προφίλ ασθενούς (Edit Profile) βλέπουμε στην εικόνα 11. Τα πεδία στο προφίλ ασθενούς μπορεί να είναι απλά (ανεξάρτητα) όπως το πεδίο «Βάρος» ή να είναι συνδυασμένα (ζεύγη). Ένα τέτοιο παράδειγμα βλέπουμε σε αυτή την εικόνα (ζεύγος επιμετρούμενης δραστικής ουσίας και φάρμακου/σκευάσματος που λαμβάνει ο ασθενής). Εδώ ο ιατρός ορίζει και την οδηγία που βλέπει ο ασθενής στο κινητό

του. Επίσης, μπορεί να ορίσει αν ένα πεδίο είναι υποχρεωτικό (για συμπλήρωση) ή προαιρετικό.

The screenshot shows a user interface for updating a patient profile. It features two main panels: 'Available Fields' and 'Selected Fields'. The 'Available Fields' panel contains a list of medical parameters such as 'Τακρόλιμους', 'Αναγραφ', 'Everolimus', 'Certican', 'Αfinitor', 'Καρτικοστεροειδή αίματος', 'Καρτικοστεροειδή λήψη', 'Πίεση Διαστολική', 'Κάλιο', 'Σχόλιο Ημέρας', 'Κένωση', 'CRP(C-Αντιδρώσα Πρωτεΐνη)', 'Ερυθρά σφύρων', 'PH ούρων', 'Ειδικό βάρος ούρων', 'Λεύκωμα ούρων(24h)', 'Πυοφαίρα ούρων', 'Αιματοκρίτης(HCT)', and 'Αιμοσφαιρίνη(HGB)'. Below this list is an 'Add >>' button. The 'Selected Fields' panel contains a list of selected parameters: 'Θερμοκρασία Σώματος', 'Πίεση Συστολική', 'Σφυγμός', 'Βάρος', 'Κρεατινίνη', 'Ούρα', 'Εισερχόμενα υγρά', 'Τακρόλιμους', 'Prograf', and 'Μυοφαινολάτη Cellcept'. Below this list are two configuration sections: 'Μυοφαινολάτη' with a 'Guideline' field and a 'Required' dropdown set to 'Optional', and 'Cellcept' with a 'Guideline' field set to '750 πριλί, 1000 βράδυ' and a 'Required' dropdown set to 'Mandatory'. There is a '<< Remove' button below these sections. At the bottom of the interface are 'Save' and 'Cancel' buttons.

Εικόνα 11: Ενημέρωση προφίλ ασθενούς

Στην εικόνα 12 βλέπουμε την αρχική σελίδα “Patient Exams” κατά την οποία ο εποπτεύων ιατρός βλέπει πινακοποιημένα τα δεδομένα/μετρήσεις ενός ασθενούς για τη χρονική περίοδο που ορίζει. Υπάρχουν, επίσης, οι παράμετροι “Rows per Page” , “Table Width(px)” για να δοθεί στον ιατρό η δυνατότητα αλληλεπίδρασης με το σύστημα, ώστε να μπορέσει να προσαρμώσει τον καθολικό πίνακα στην οθόνη του.

The screenshot shows the 'Exam Page' configuration interface. It includes a 'Home' breadcrumb at the top left. The main area contains configuration options for the exam page: 'Start Date' (2023-10-01), 'End Date' (2024-02-01), and 'Patient ID' (AnonymizedPatient01). Below these are 'Rows per Page' (20), 'Table Width (px)' (1500), and 'Days' (7). At the bottom, there are two buttons: 'Daily Exams' and 'Summarized Exams (per days)'.

Εικόνα 12: Αρχική σελίδα Patient Exams

Πατώντας το “Daily exams” εμφανίζονται όλες οι εξετάσεις που έχουν γίνει στη χρονική περίοδο που τέθηκε. Η εικόνα 13 έχει το παράδειγμα με Rows per Page = 10.

Home

Exam Page

Start Date: 2023-10-01 End Date: 2024-02-01 Patient ID: AnonymizedPatient01

Rows per Page: 10 Table Width (px): 1500 Days: 7

Daily Exams Summarized Exams (per days)

Exam Date	Θερμοκρασία Σώματος	Πίεση Συστολική	Πίεση Διαστολική	Σφυγμός	Βάρος	Όσρα	Εισερχόμενα υγρά	CRP(C-Αντιδρώσα Πρωτεΐνη)	PH ούρων	Ειδικό βάρος ούρων	Λεπίδαμα
Nov 22, 2023	36.6										
Nov 23, 2023	36.6										
Nov 24, 2023	36.6	125	80	85		210	1550				
Nov 25, 2023	36.6	148	80	94	72.6	150	1500	34			
Nov 26, 2023	36.6	140	90	85	71.6	300	1550				
Nov 27, 2023	36.6	150	87	78	72.9	350	750				
Nov 28, 2023	36.6	150	86	76	72.3	350	700	45			
Nov 29, 2023	36.6	125	75		72.2	240	700				
Nov 30, 2023	36.6	142	84	79	68	300	850	40			
Dec 01, 2023	36.6	150	100	79	68.6	300	800				

< 1 2 3 >

Εικόνα 13: Παράδειγμα καθολικού πίνακα με παράμετρο “Rows per Page” = 10

Ο ιατρός μπορεί να πλοηγηθεί και στις υπόλοιπες “σελίδες” του καθολικού πίνακα, χρησιμοποιώντας τον οδηγό κάτω από τον τελευταίο.

Στις επόμενες εικόνες βλέπουμε πινακοποιημένα τα δεδομένα/μετρήσεις (με “Rows per Page” = 30 για να είναι ευκολότερη η αποτύπωση του καθολικού πίνακα) ενός ασθενούς για τη χρονική περίοδο που έχει ορίσει ο εποπτεύων ιατρός. Ο πίνακας περιέχει πολλές στήλες και απαιτείται οριζόντια κύλιση (horizontal scroll) του πίνακα. Εδώ για διευκόλυνση έχουμε τμηματοποιήσει σε πέντε τμήματα τα πινακοποιημένα δεδομένα ενός ασθενούς (βλέπε εικόνες 14α, 14β, 14γ, 14δ, 14ε).

Daily Exams		Summarized Exams (per days)										
δ βάρος ούρων	Λεύκωμα ούρων(24h)	Πυοσφαίρια ούρων	Ερυθρά ούρων	Καλλιέργεια ούρων	Αιματοκρίτης(HCT)	Αιμοσφαιρίνη(HGB)	Λευκά αιμοσφαίρια (WBC)	Π	Λ	Ε	Μ	Αιμοπετί
Nov 22, 2023						12.7	8020	81	10	1.5	6.7	218000
Nov 23, 2023						10.7	11530	96	2.1	0.1	1.8	166000
Nov 24, 2023												
Nov 25, 2023						9.5	12610	95	1.5	0.6	3.2	175000
Nov 26, 2023					27.8	9.4	8320	91	4.1	0.6	3.7	134000
Nov 27, 2023						8.7	7820	89	5	0.5	4.6	141000
Nov 28, 2023						9.5	7200	85	7	2	5	166000
Nov 29, 2023						8.8	6490	82	9	2	6	168000
Nov 30, 2023						9.1	6860	80	9	3	8	171000
Dec 01, 2023				-		10	6720	81	8	3	8	198000
Dec 02, 2023						9.5	5290	80	9	3	6	218000
Dec 03, 2023	ιχνη	20		faecalis		10	5780	80	11	2	7	249000
Dec 04, 2023						9.5	7090	84	9	0.8	5.6	285000
Dec 05, 2023						9.7	8050	82	11	0.9	5	262000
Dec 06, 2023	-	8-10		-		10	7250	83	12	1.2	4.4	265000
Dec 07, 2023	RBC			-		9.6	7320	82	12	1.5	4	216000
Dec 08, 2023						8.9	6580	84	10	1	3	219000
Dec 09, 2023					27.2	9.2	6440	84	11	0	3.9	236000
Dec 10, 2023	-	2-4	>100	-		8.7	6780	84	11	1.2	4	238000
Dec 11, 2023	-	3-5	30-35	-		8.3	6380	82	11	1	4	237000
Dec 12, 2023						10	5640	83	11	1	4	264000
Dec 15, 2023					28.6	9.7	4740	78	12	1.7	7.2	271000
Jan 02, 2024	-	1-3		klebsiella		12.6	9280	80	14	0.5	5.5	206000
Dec 19, 2023	-	1-3		-	30	11	4980	79	12	1.6	6.6	265000
Dec 22, 2023	-	2-4		klebsiella		10.7	5120	72	18	2	8	228000
Dec 27, 2023	-	2-4				13.7	7100	75	16	1	6	217000
Jan 05, 2024	-	1-3		klebsiella		12.9	7450	78	15	0.5	5	211000
Jan 12, 2024						14	7800	77	16	1	5.4	214000

Εικόνα 14α: πινακοποιημένα δεδομένα/μετρήσεις ασθενούς

Daily Exams		Summarized Exams (per days)									
Exam Date	Θερμοκρασία Σώματος	Πίση Συστολική	Πίση Διαστολική	Σφυγμός	Βάρος	Ούρα	Εισερχόμενα υγρά	CRP(C-Αντιδρώσα Πρωτεΐνη)	PH ούρων	Ειδικό βάρος ούρων	Λεύκωμα
Nov 22, 2023	36.6										
Nov 23, 2023	36.6										
Nov 24, 2023	36.6	125	80	85		210	1550				
Nov 25, 2023	36.6	148	80	94	72.6	150	1500	34			
Nov 26, 2023	36.6	140	90	85	71.6	300	1550				
Nov 27, 2023	36.6	150	87	78	72.9	350	750				
Nov 28, 2023	36.6	150	86	76	72.3	350	700	45			
Nov 29, 2023	36.6	125	75		72.2	240	700				
Nov 30, 2023	36.6	142	84	79	68	300	850	40			
Dec 01, 2023	36.6	150	100	79	68.6	300	800				
Dec 02, 2023	36.6	112	71	79	68.3	700	700	19			
Dec 03, 2023	36.6	114	70	84	67.9	1300	600		7	1010	ιχνη
Dec 04, 2023	36.6	130	81	84	66.9	850	750				
Dec 05, 2023	36.6	110	60	86	66.5	1100	1000				
Dec 06, 2023	36.6	104	60	86	66.5	1400	2000		5.5	1013	-
Dec 07, 2023	36.6	120	60	88	67.5	2500	2800	4.48			RBC
Dec 08, 2023	36.6	125	77	84	68	3500	3600				
Dec 09, 2023	36.6	125	85	77	68.6	3900	4200				
Dec 10, 2023	36.6	154	87	80	69.2	4200	4600	2.7	6.5	1008	-
Dec 11, 2023	36.6	175	86	80	68.9	3700	4700	2.69	5.3	1005	-
Dec 12, 2023	36.6	150	90	79	69		4500				
Dec 15, 2023	36.6	169	90	84	66.8	3800	4000				
Jan 02, 2024	36.6	146	99	92	67.1	3800	4000		5.5	1010	-
Dec 19, 2023		160	90	80	66.3	4700	4500		5.5	1010	-
Dec 22, 2023		156	93	97	66.5	3900	4500		5	1010	-
Dec 27, 2023		158	100	105	66.4	3800	4000		5	1014	-
Jan 05, 2024		164	94	72	66	3500	3500		5	1016	-
Jan 12, 2024		180	114	82	66	3600	3500				

Εικόνα 14β: πινακοποιημένα δεδομένα/μετρήσεις ασθενούς

Daily Exams		Summarized Exams (per days)													
E	M	Αιμοπετάλια (PLT)	Σάκχαρο	Ουρία	Κρεατινίνη	Clear Κρεατινίνη	Νάτριο - Na	Κάλιο	Ασβέστιο - Ca	Φώσφορος - P	Μαγνήσιο - Mg	Αλβουμίνη - Alb	SGOT	SGPT	γ
Nov 22, 2023		218000	82	111	11.2		138	5.6	9.6	4.4	2.19	5	9	11	2:
Nov 23, 2023		166000	159	68	7.79		138	5.7	8.3	4.4	2.03	4			
Nov 24, 2023															
Nov 25, 2023		175000	111	97	7.78		137	5.1	8.6	7.2	2.13	3.3	17	12	1:
Nov 26, 2023		134000	147	98	7.8		137	4.1	8.1	5.6		3.6			
Nov 27, 2023		141000	82	156	10.52		139	4	8	6.9		3.4			
Nov 28, 2023		166000	72	115	8.38		139	4.1	8.6	7	2.36	3.67	14	7	2:
Nov 29, 2023		168000	89	162	10.95		137	3.7	8.4	6.7	2.26	3.4			
Nov 30, 2023		171000	77	133	9.61		139	3.8	9.2	6.2	2.4	4	16	25	5:
Dec 01, 2023		198000	75	87	11.5		139	4	8.8	6.1	2.3	3.8			
Dec 02, 2023		218000	78	114	8.15		138	4.2	9.6	5.7	2.18	4.1	15	29	5:
Dec 03, 2023		249000	70	167	10.1		138	4.7	8.9	9.5	2.2	4.1			
Dec 04, 2023		285000	92	214	11.77		139	5.1	9.1	4.8	2.14	4.3	17	37	4:
Dec 05, 2023		262000	71	129	7.59		141	5	9.3	5.3	2	4.3			
Dec 06, 2023		265000	92	172	8.13		138	4.4	9	4.4	1.78	4.3			
Dec 07, 2023		216000	75	112	5.61		137	4.5	9	4.2	1.77	4			
Dec 08, 2023		219000	106	114	5.43	12	133	3.6	8.4	3.4		4			
Dec 09, 2023		236000	79	114	4.92	13	135	4	8.7			4			
Dec 10, 2023		238000	82	105	4.34	16	136	3.8	8.7	2.8	1.24	3.9	12	27	2:
Dec 11, 2023		237000	79	79	3.47	20	138	3.9	8.2	2.6	1.74	3.7	11	27	2:
Dec 12, 2023		264000	84	62	2.94	24	137	3.9	8.8	2.2	1.44	4	11	29	2:
Dec 15, 2023		271000	87	56	2.32	34	135	4	9.6	1.7	1.39	4.5			
Jan 02, 2024		206000	96	87	1.45	59	139	4	10.5	2.5	1.31	4.3	15	33	2:
Dec 19, 2023		265000	90	50	2.04	40	133	3.8	9.6	1.6		4.5	12	19	2:
Dec 22, 2023		228000	102	57	1.31	40	133	3.6	9.6	2.2	1.05	4.8	12	16	2:
Dec 27, 2023		217000	102	63	1.56		139	4.2	10.4	2	1.3	5.21			
Jan 05, 2024		211000	86	57	1.34		140	3.9	9.9	2.3	1.38	4.1	10	16	3:
Jan 12, 2024		214000	93		1.33	67	140	4.1	10.5	2.1	1.35	4.6			

Εικόνα 14γ: πινακοποιημένα δεδομένα/μετρήσεις ασθενούς

Daily Exams		Summarized Exams (per days)												
Inf	Τακρόλιμιος	Κορτικοστεροειδή λήψη	Induction therapy	Υπόλοιπα φάρμακα (Φαρμακευτική αγωγή εκτός ανοσοκατασταλής)	Σχόλιο Ημέρας	% ούρων ημέρας	% ούρων μήνα							
Nov 22, 2023				Cymevene Bactrimel Losec Tazocin Fungustatin	ΜΕΤΑΜΟΣΧΕΥΣΗ									
Nov 23, 2023		500	Simulect	Cymevene Bactrimel Losec Tazocin Fungustatin										
Nov 24, 2023		20		Cymevene Bactrimel Losec Tazocin Fungustatin	DTPA. D0 σαρ / IVC = 1.6	14								
Nov 25, 2023		20		Cymevene Bactrimel Losec Tazocin Fungustatin		10								
Nov 26, 2023		20		Cymevene Bactrimel Losec Tazocin Fungustatin		19								
Nov 27, 2023		20	Simulect	Cymevene Bactrimel Losec Tazocin Fungustatin	Ro θώρακος	47								
Nov 28, 2023		20		Cymevene Bactrimel Losec Tazocin Fungustatin D3 1200		50								
Nov 29, 2023		20		Cymevene Bactrimel Losec Tazocin Fungustatin Ferinject 1g Mimpara 30mg D3 1200	Ro θώρακος	34								
Nov 30, 2023		20		Cymevene Bactrimel Losec Tazocin Fungustatin Mimpara 30mg D3 1200		35	25							
Dec 01, 2023		20		Cymevene Bactrimel Losec Tazocin Fungustatin Mimpara 30mg D3 1200		38								
Dec 02, 2023		20		Cymevene Bactrimel Losec Tazocin Fungustatin Mimpara 30mg D3 1200	5 Lasix off Folley	100								
Dec 03, 2023		20		Cymevene Bactrimel Losec Tazocin Fungustatin Mimpara 30mg D3 1200	5 Lasix	217								
Dec 04, 2023		20		Cymevene Bactrimel Losec Tazocin Fungustatin Mimpara 30mg D3 1200	cxr	113								
Dec 05, 2023		20		Cymevene Bactrimel Losec Tazocin Fungustatin Mimpara 30mg D3 1200		110								
Dec 06, 2023		18		Cymevene Bactrimel Losec Tazocin Fungustatin Mimpara 30mg D3 1200		70								
Dec 07, 2023		18		Cymevene Bactrimel Losec Tazocin Fungustatin Mimpara 30mg D3 1200		89								
Dec 08, 2023		18		Cymevene Bactrimel Losec Tazocin Fungustatin Mimpara 30mg D3 1200		97								
Dec 09, 2023		18		Cymevene Bactrimel Losec Tazocin Fungustatin Mimpara 30mg		93								
Dec 10, 2023		18		Cymevene Bactrimel Losec Tazocin Fungustatin Mimpara 30mg		91								
Dec 11, 2023		18		Cymevene Bactrimel Losec Tazocin Fungustatin Mimpara 30mg		79								
Dec 12, 2023		18												
Dec 15, 2023		18				95								
Jan 02, 2024		14				95								
Dec 19, 2023		18				104								
Dec 22, 2023		16				87								
Dec 27, 2023		14				95	93							
Jan 05, 2024		12				100								
Jan 12, 2024		10				103	99							

Εικόνα 14δ: πινακοποιημένα δεδομένα/μετρήσεις ασθενούς

Daily Exams		Summarized Exams (per days)										
GOT	SGPT	γ - GT	Αλκαλική Φωσφατάση (ALP)	Ουρικό οξύ	Χοληρυθρίνη ολική	HCO3 (δισανθρακικά)	Ειδικό Προστατικό Αντιγόνο(PSA)	PCR-BK	Cellcept	Prograf	Τοκρόλιμους	Κορτικοί
Nov 22, 2023	21	96				19.1	1.42					
Nov 23, 2023									2	2.5-2.5		500
Nov 24, 2023									2	2.5-2.5		20
Nov 25, 2023	17	55		3.7	0.26	21.7			2	2.5-2.5		20
Nov 26, 2023						22.5			2	2.5-2.5		20
Nov 27, 2023						22.4			2	2.5-2.5		20
Nov 28, 2023	29	53		6.1	0.27	24.2			2	2.5-1	12	20
Nov 29, 2023						21.1			2	2-2		20
Nov 30, 2023	50	65		7.1	0.24	27.1			2	2-2		20
Dec 01, 2023				9.6		21.3			2	2-2	8.9	20
Dec 02, 2023	50	66		5.2	0.29	26.8			2	2-2		20
Dec 03, 2023				3.8					2	2-2		20
Dec 04, 2023	44	69		9.8	0.24	22.8			2	2-2		20
Dec 05, 2023				5.4		26.6			2	2-2	8.1	20
Dec 06, 2023									2	2-2		18
Dec 07, 2023						26.5			2	2-2		18
Dec 08, 2023						22.1			2	2-3	5	18
Dec 09, 2023									2	2.5-2.5		18
Dec 10, 2023	28	60		6.6		21.5			2	2.5-2.5		18
Dec 11, 2023	26	54		6.3	0.32	23.8			2	2.5-2.5		18
Dec 12, 2023	29	61		5.8	0.38	23.9			2	2.5-3.5	5.2	18
Dec 15, 2023									-	2	3-3	4.8
Jan 02, 2024	29	67		6.3	0.31				2	3.5-4	6.7	14
Dec 19, 2023	27	64		5.5	0.42	22.1			-	2	4-4	7.5
Dec 22, 2023	26	71		6.4	0.33				2	4-4	9.6	16
Dec 27, 2023				5.9					2	3.5-3.5	6.1	14
Jan 05, 2024	30	71		6.3	0.3				-	2	4-4	8.8
Jan 12, 2024									2	4-4	11.7	10

Εικόνα 14ε: πινακοποιημένα δεδομένα/μετρήσεις ασθενούς

Υπάρχει η δυνατότητα να γίνει σύνοψη γραμμών ανά X ημέρες(τιμή του πεδίου Days). Για παράδειγμα από περίοδο 3 μηνών (90 γραμμές) παράγεται πίνακας με 13 γραμμές (1 μέση τιμή / γραμμή ανά εβδομάδα) για κάθε δείκτη/στήλη. Το αποτέλεσμα φαίνεται στην εικόνα 15.

Exam Page

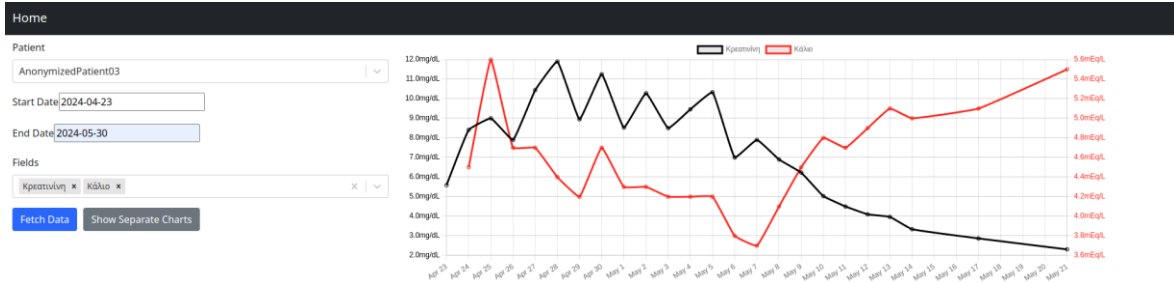
Start Date: End Date: Patient ID:

Rows per Page: Table Width (px): Days:

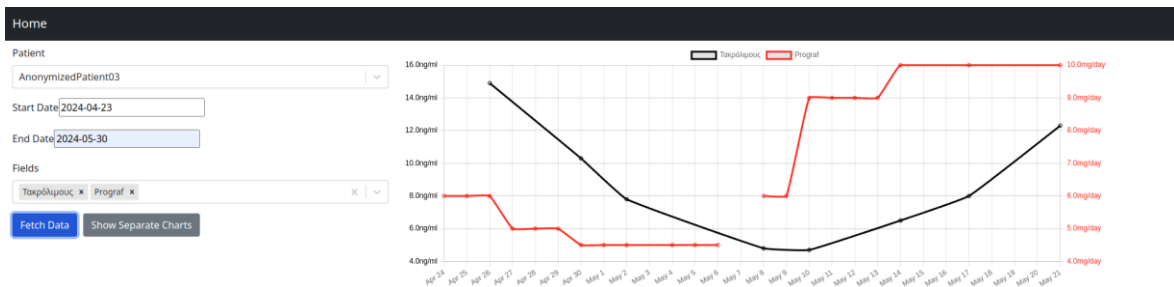
Daily Exams		Summarized Exams (per days)									
Exam Date	Θερμοκρασία Σώματος	Πίεση Συστολική	Πίεση Διαστολική	Σφυγμός	Βάρος	Ούρα	Εισερχόμενα υγρά	CRP(C-Αντιδρώσα Πρωτεΐνη)	PH ούριου	Ειδικό βάρος ούριου	Πυοσφ
Nov 28, 2023	36.6	142.6	84.6	83.6	72.4	272.0	1210.0	39.5			
Dec 05, 2023	36.6	126.1	77.3	81.8	68.3	684.3	771.4	29.5	7.0	1010.0	20.0
Dec 12, 2023	36.6	136.1	77.9	82.0	68.2	3200.0	3771.4	3.3	5.8	1008.7	
Dec 19, 2023	36.6	164.5	90.0	82.0	66.6	4250.0	4250.0		5.5	1010.0	
Jan 02, 2024	36.6	152.0	99.5	98.5	66.8	3800.0	4000.0		5.3	1012.0	
Dec 26, 2023		156.0	93.0	97.0	66.5	3900.0	4500.0		5.0	1010.0	
Jan 09, 2024		164.0	94.0	72.0	66.0	3500.0	3500.0		5.0	1016.0	
Jan 16, 2024		180.0	114.0	82.0	66.0	3600.0	3500.0				

Εικόνα 15: Συνοψισμένος πίνακας δεδομένων/μετρήσεων ασθενούς

Τα πεπλεγμένα διαγράμματα (Charts) είναι διαγράμματα που απεικονίζουν μία ή περισσότερες σειρές μετρήσεων σε ένα διάγραμμα. Κάθε σειρά αφορά μια διαφορετική εξέταση, μέτρηση ή δείκτη ενός ασθενούς για μια (από τον επιβλέποντα ιατρό) οριζόμενη περίοδο. Τέτοια διαγράμματα βλέπουμε στις εικόνες 16, 17 και 18.



Εικόνα 16: Πεπλεγμένο διάγραμμα για Κρεατινίνη και Κάλιο

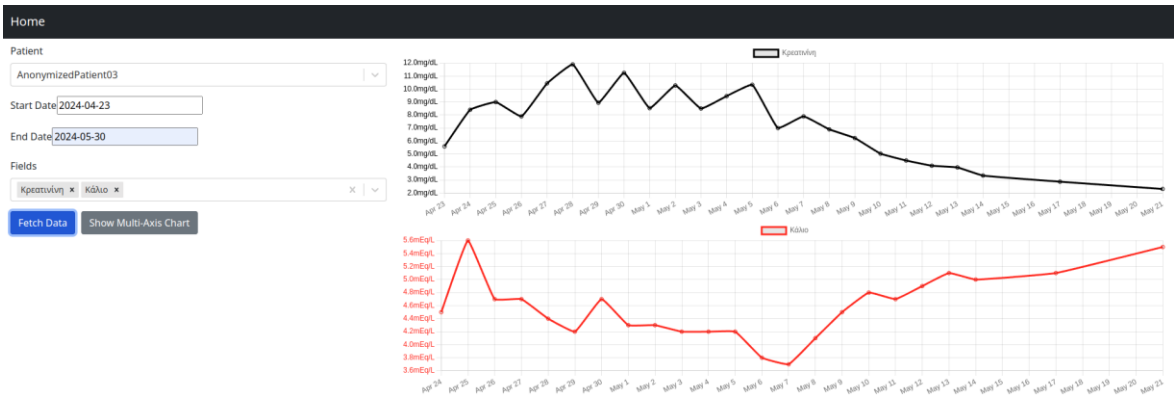


Εικόνα 17: Πεπλεγμένο διάγραμμα για Τακρόλιμους και Prograf

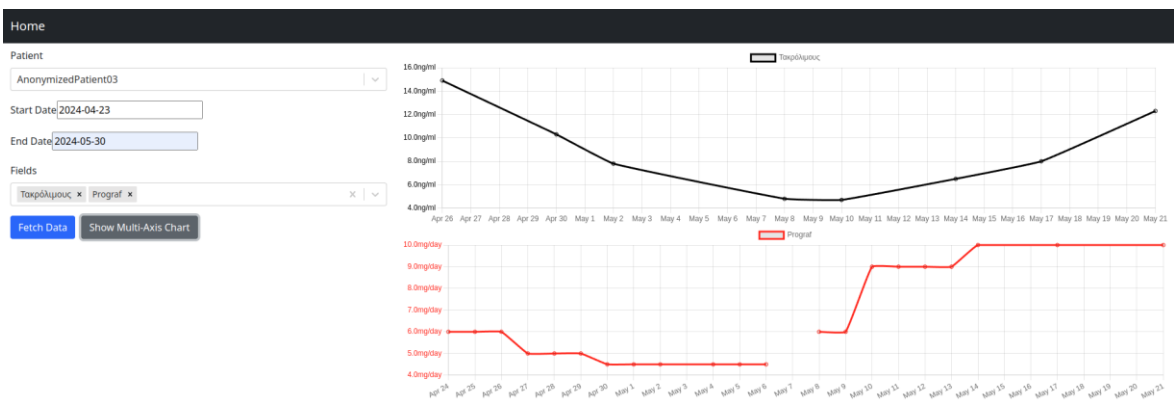


Εικόνα 18: Πεπλεγμένο διάγραμμα για Κρεατινίνη, Κάλιο, Τακρόλιμους και Prograf

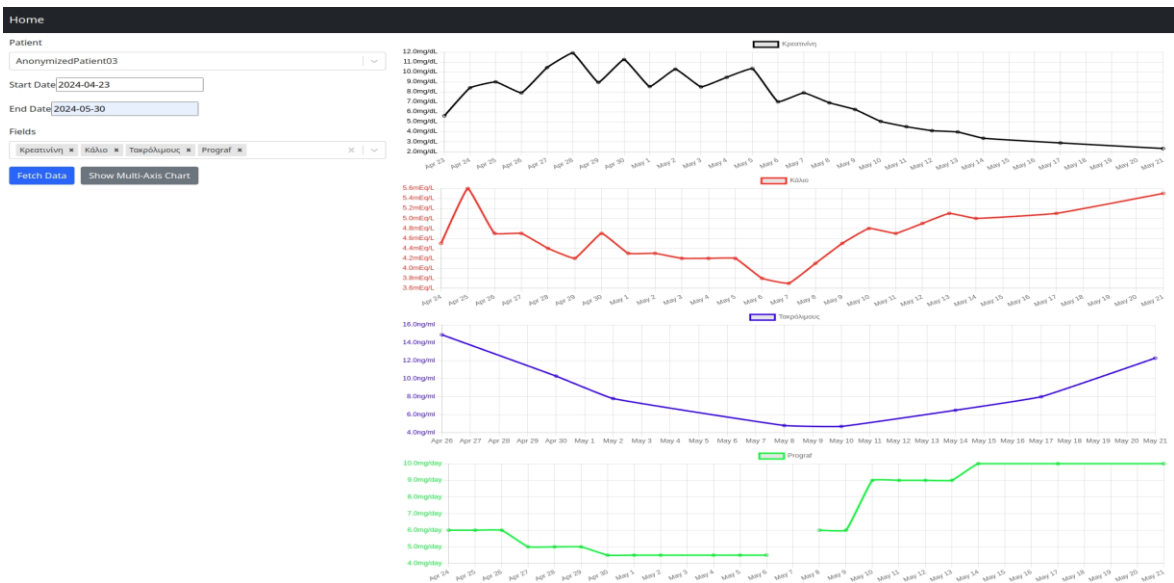
Ανα πάσα στιγμή ο χρήστης μπορεί να πατήσει το “Show seperated charts” για να δει τα διαγράμματα να απεικονίζουν πάντα μόνο μια σειρά μετρήσεων. Σειρά μετρήσεων είναι (όπως αναφέρθηκε και πριν) μια ακολουθία τιμών που αφορούν μια εξέταση, μέτρηση ή δείκτη ενός ασθενούς για την οριζόμενη (από τον επιβλέποντα ιατρό) περίοδο. Βέβαια, ο ιατρός μπορεί να δει πολλές τέτοιες σειρές σε παράλληλη παράθεση. Τέτοια μη πεπλεγμένα διαγράμματα βλέπουμε στις εικόνες 19, 20 και 21.



Εικόνα 19: Μη Πεπλεγμένο διάγραμμα για Κρεατινίνη και Κάλιο



Εικόνα 20: Μη Πεπλεγμένο διάγραμμα για Τακρόλιμους και Prograf



Εικόνα 21: Μη Πεπλεγμένο διάγραμμα για Κρεατινίνη, Κάλιο, Τακρόλιμους και Prograf

Βιβλιογραφία

1. Πληροφορική Υγείας Ενότητα 3 : Εννοιολογικές Αποσαφηνίσεις Ευγενία Τόκη
Τεχνολογικό Εκπαιδευτικό Ίδρυμα Ηπείρου
2. Διπλωματική εργασία «Modern information technology application in the health sector» Πανεπιστήμιο Πειραιώς, Τμήμα Οργάνωσης και Διοίκησης Επιχειρήσεων
Πρόγραμμα Μεταπτυχιακών Σπουδών στη Διοίκηση Επιχειρήσεων για στελέχη (E-MBA)
3. Dinu, V., & Nadkarni, P. (2007). Guidelines for the effective use of entity–attribute–value modeling for biomedical databases. **International Journal of Medical Informatics**, 76(11-12), 769–779
4. Nadkarni, P., Marengo, L., Chen, R., Skoufos, E., Shepherd, G., Miller, P., & Astakhov, V. (1999). Organization of Heterogeneous Scientific Data Using the EAV/CR Representation. **Journal of the American Medical Informatics Association**, 6(6), 478-493
5. Chute, C. G., & Beck, S. A. (2001). The Enterprise Data Trust. **Health Services Research**, 37(4), 871-884
6. Johnson, S. B. (1996). Generic data modeling for clinical repositories. **Journal of the American Medical Informatics Association**, 3(5), 328-339
7. Καρανικόλας, Νικήτας Ν. «Πληροφορική και επαγγέλματα υγείας» - 1η έκδ. -
Αθήνα : Εκδόσεις Νέων Τεχνολογιών, 2010
8. Nikitas N. Karanikolas, Maria Nitsiou, Emmanuel J. Yannakoudakis and Christos Skourlas. CUDL Language Semantics, Liven Up the FDB Data Model. AD-BIS'2007. 11th East-European Conference on Advances in Databases and Information Systems, September 29 - October 03, 2007, Varna, Bulgaria.
9. Nikitas N. Karanikolas, Maria Nitsiou, Emmanuel J. Yannakoudakis and Christos Skourlas. CUDL language semantics: updating FDB data. *Journal of Systems and Software* 82 (6) 947-962