

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη εφαρμογής Διαχείρισης Ιατρικών Δεδομένων
και Ιστορικού Υγείας για κινητές συσκευές



Του φοιτητή
Αλέξανδρου Βουρλιώτη
Αρ. Μητρώου: 2019212

Επιβλέπων
Αδαμίδης Παναγιώτης
Καθηγητής

Θεσσαλονίκη, Ιούνιος 2026

Τίτλος Δ.Ε. Ανάπτυξη εφαρμογής Διαχείρισης Ιατρικών Δεδομένων και Ιστορικού Υγείας για κινητές συσκευές

Κωδικός Δ.Ε.: 24293

Όνοματεπώνυμο φοιτητή: Αλέξανδρος Βουρλιώτης

Όνοματεπώνυμο εισηγητή: Αδαμίδης Παναγιώτης

Ημερομηνία ανάληψης Δ.Ε.: 30 Οκτωβρίου 2025

Ημερομηνία περάτωσης Δ.Ε.: 29 Μαΐου 2026

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Αλέξανδρου Βουρλιώτη που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιοδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

Στον καθηγητή μου, για την ευκαιρία και την καθοδήγησή του.

Στην οικογένειά μου, για τη συμπαράστασή τους.

Πρόλογος

Η εργασία αυτή στοχεύει στη δημιουργία μίας εφαρμογής, για κινητές συσκευές, που θα επιτρέπει την αποθήκευση, τροποποίηση και παρακολούθηση ιατρικών δεδομένων. Κίνητρο για την επιλογή του θέματος της παρούσας διπλωματικής ήταν η επιθυμία να αξιοποιηθούν οι γνώσεις που αποκτήθηκαν μέσω του προγράμματος σπουδών – όπως η ενσωμάτωση και διαχείριση απομακρυσμένης βάσης δεδομένων – με δεδομένο ότι υπάρχει μεγάλη ποικιλία διαθέσιμων τεχνολογιών για την ανάπτυξη τέτοιων εφαρμογών.

Η επιλογή, το θέμα της εφαρμογής να αφορά στον ιατρικό τομέα, έγινε λόγω της χρησιμότητας μίας εφαρμογής, που οργανώνει πολλά από τα διαφορετικά τμήματα του τομέα σε ένα σύστημα. Για το μέσο άτομο, η καταγραφή και οργάνωση δεδομένων όπως ραντεβού, παθήσεις και μετρήσεις ζωτικών σημείων, ανάμεσα σε άλλα, είναι μία περίπλοκη διαδικασία, όπου, σε πολλές περιπτώσεις, τα σχετικά δεδομένα χάνονται ή απλά ξεχνιούνται. Τα χρήσιμα έγγραφα μπορεί να είναι αποθηκευμένα σε διαφορετικές τοποθεσίες, ή οργανωμένα με τρόπο μη λειτουργικό για το σκοπό της ανάκτησής τους, ακόμα και για τον κάτοχό τους.

Πέραν αυτών, πρέπει να λαμβάνονται υπόψιν οι περιπτώσεις που απαιτούν περισσότερη προσοχή και καλύτερη οργάνωση των πληροφοριών, όπως ραντεβού που προγραμματίστηκαν πριν από πολύ καιρό, ή η λήψη φαρμάκων που μόλις ορίστηκαν από το γιατρό. Για περιπτώσεις όπως οι μετρήσεις ζωτικών σημείων, μπορεί και να χρειαστεί να οργανωθούν με τρόπο ώστε να είναι δυνατή η παρακολούθηση της πορείας της υγείας στο χρόνο, που σημαίνει ότι πρέπει να έχει γίνει καταγραφή όχι μόνο των μετρήσεων αλλά και της ημερομηνίας, ώστε να είναι όλες αποθηκευμένες και διαθέσιμες οποιαδήποτε στιγμή.

Η εφαρμογή για αυτή την εργασία όχι μόνο καλύπτει όλες αυτές τις ανάγκες, αλλά προσφέρει και επιπλέον δυνατότητες. Δεν αποθηκεύει απλά όλα τα διάφορα δεδομένα του χρήστη σε μία βάση δεδομένων, αλλά αναλαμβάνει και το καθήκον να τον ενημερώνει για εκκρεμή ραντεβού ή φάρμακα, ενώ παράλληλα οργανώνει και παρουσιάζει τα δεδομένα που χρειάζεται με εύκολα κατανοητό τρόπο.

Περίληψη

Η εργασία αυτή αφορά την ανάπτυξη μιας εφαρμογής διαχείρισης προσωπικών ιατρικών δεδομένων. Κάθε χρήστης πρέπει να μπορεί να καταθέτει δεδομένα για διάφορα θέματα υγείας, τα οποία θα κρυπτογραφούνται και θα αποθηκεύονται σε μία απομακρυσμένη βάση δεδομένων. Τα αποθηκευμένα δεδομένα θα μπορούν μετά να προβληθούν με διαφορετικούς τρόπους στο χρήστη που τα κατέθεσε, στον οποίο θα δίνεται και η ικανότητα να τροποποιήσει ή και να διαγράψει εντελώς τόσο τις τιμές που αποθήκευσε, όσο και ολόκληρο το αντικείμενο που τις περιείχε.

Η πρόσβαση του χρήστη στα δεδομένα του πρέπει να περιλαμβάνει και την δυνατότητα να τα κατεβάσει ως αρχεία, χρησιμοποιώντας διάφορες διαμορφώσεις. Επιπλέον, θα παρέχει υπηρεσίες ειδοποίησης του χρήστη για ορισμένα αντικείμενα ενδιαφέροντος που κατέθεσε, τα οποία περιέχουν πεδία χρόνου, όπως για ένα ραντεβού με γιατρό που έχει για την τρέχουσα ημέρα.

Στην πράξη, η υλοποίηση αυτής της ιδέας εκπληρώνει με επιτυχία τους λειτουργικούς στόχους που ορίζονται για αυτή, αλλά παρουσιάζει ταυτόχρονα συγκεκριμένους περιορισμούς και ελαττώματα. Από τη μια, υπάρχει πληθώρα τεχνολογιών, για τη διευκόλυνση της ανάπτυξης εφαρμογών για κινητές συσκευές, από την άλλη όμως υπάρχουν κάποιοι πρακτικοί περιορισμοί στην απόδοση αυτής της εφαρμογής. Η εφαρμογή ικανοποιεί όλες τις βασικές της απαιτήσεις, με όλες τις λειτουργίες που αυτό περιλαμβάνει, με όλη τη σταθερότητα που χρειάζεται για τις οθόνες και τις αποκρίσεις της, αλλά υποφέρει από χαμηλότερη ταχύτητα από όσο θα ήταν επιθυμητό λόγω της ανάγκης επικοινωνίας με τη βάση δεδομένων.

Development of an Application for the Management of Medical Data and Medical History for mobile devices

Alexandros Vourliotis

Abstract

This project is about the development of an application for the handling of personal medical data. Every user needs to be able to submit various types of data, which will then be encrypted and saved in a remote database. This saved data will then need to be able to be projected through a variety of means for the use of the user who deposited them, who will have the ability to then edit the submitted values or even delete the document inside which they are saved outright.

The user's access to this data must also include giving them the means to download their data entries as files, using any among several different formats. Furthermore, the application needs to have the functionality to send notifications to the user for certain items which they submitted that contain fields for time values and information that the user should logically be kept abreast of, like doctor appointments for the current day.

In practice, the implementation of this idea succeeds at fulfilling all functional requirements, while simultaneously encountering specific challenges and limitations. On one hand, there are plenty of technologies that ease the development of mobile applications, but, on the other hand, there exist certain practical constraints when it comes to the performance of this application. It manages to satisfy the basic requirements listed, with all the pieces of functionality that includes, as well as the stability required for parts like screens and their responsiveness but suffers from lower speed due to the need for communication with the database.

Περιεχόμενα

Πρόλογος.....	iv
Περίληψη.....	v
Abstract	vi
Περιεχόμενα	vii
Κατάλογος Σχημάτων	x
Συνομογραφίες.....	xi
Κεφάλαιο 1ο: Εισαγωγή	12
1.1 Εισαγωγή.....	12
1.2 Αντικείμενο και Στόχοι της Εργασίας.....	12
1.3 Κίνητρο και Σημασία Διαχείρισης Προσωπικών Ιατρικών Δεδομένων	12
1.4 Συμβολή της εργασίας.....	1
1.5 Δομή της Διπλωματικής Εργασίας.....	1
1.6 Επίλογος.....	1
Κεφάλαιο 2ο: Υπόβαθρο και Σχετικές Τεχνολογίες.....	3
2.1 Εισαγωγή.....	3
2.2 Περιεχόμενο	3
2.2.1 Mobile εφαρμογές στον τομέα της υγείας.....	3
2.2.2 Διαχείριση προσωπικών ιατρικών δεδομένων.....	3
2.2.3 Θέματα ασφάλειας και προστασίας προσωπικών δεδομένων	4
2.2.4 Τεχνολογίες ανάπτυξης mobile εφαρμογών	5
2.2.5 Γραφήματα και οπτικοποίηση δεδομένων υγείας.....	5
2.2.6 Πρότυπα διαλειτουργικότητας ιατρικών δεδομένων.....	6
2.3 Επίλογος.....	6
Κεφάλαιο 3ο: Ανάλυση Απαιτήσεων.....	7
3.1 Εισαγωγή.....	7
3.2 Έννοιες-Κλειδιά	7
3.2.1 Σκοπός και λειτουργίες της εφαρμογής.....	7
3.2.2 Σενάρια Χρήσης	8
3.2.3 Λειτουργικές απαιτήσεις	8
3.2.4 Μη λειτουργικές απαιτήσεις.....	9
3.2.5 Περιγραφή βασικών λειτουργιών της εφαρμογής	9
3.3 Επίλογος.....	10

Κεφάλαιο 4ο:	Σχεδιασμός Συστήματος	11
4.1	Εισαγωγή.....	11
4.2	Συνολική αρχιτεκτονική εφαρμογής	12
4.2.1	Κεντρικό αρχείο και είσοδος στην εφαρμογή	12
4.2.2	Καταγραφή χρήστη στη βάση δεδομένων	15
4.2.3	Κεντρική οθόνη εφαρμογής.....	15
4.2.4	Οθόνη ιστορικού.....	18
4.2.5	Ενδιάμεση οθόνη πλοήγησης	19
4.2.6	Δομή οθονών εισαγωγής	21
4.2.7	Οθόνες εισαγωγής	22
4.2.8	Οθόνες εισαγωγής αποτελεσμάτων	25
4.2.9	Οθόνες τροποποίησης και διαγραφής.....	26
4.2.10	Οθόνη Επαφών Εκτάκτου Ανάγκης.....	27
4.2.11	Οθόνη λήψης/αποστολής αρχείων	27
4.2.12	Οθόνη δημιουργίας αρχείων	28
4.2.13	Οθόνη τελευταίας μηνιαίας αναφοράς.....	28
4.2.14	Οθόνη ημερολογίου	28
4.2.15	Γραφήματα της εφαρμογής.....	30
4.2.16	Υπηρεσίες.....	31
4.2.17	Βασικές υπηρεσίες της εφαρμογής	32
4.3	Σχεδιασμός Βάσης Δεδομένων.....	33
4.3.1	Συλλογές περιορισμένης πρόσβασης.....	33
4.3.2	Συλλογές πλήρους πρόσβασης	35
4.4	Επίλογος.....	38
Κεφάλαιο 5ο:	Υλοποίηση Εφαρμογής.....	40
5.1	Εισαγωγή.....	40
5.2	Τομείς υλοποίησης	40
5.2.1	Περιβάλλον Ανάπτυξης.....	40
5.2.2	Υλοποίηση μηχανισμού πιστοποίησης χρήστη	40
5.2.3	Υλοποίηση διαχείρισης ιατρικών δεδομένων.....	41
5.2.4	Υλοποίηση μέτρων ασφαλείας	42
5.2.5	Υλοποίηση αποθήκευσης δεδομένων στο Firebase.....	44
5.2.6	Υλοποίηση λειτουργιών ημερολογίου και ραντεβού	46
5.2.7	Υλοποίηση γραφημάτων και οπτικοποίησης δεδομένων	49
5.2.8	Υλοποίηση ειδοποιήσεων	52

5.2.9	Διαχείριση διαγραφών και τροποποιήσεων δεδομένων	54
5.3	Επίλογος.....	56
Κεφάλαιο 6ο:	Αξιολόγηση της Εφαρμογής.....	57
6.1	Εισαγωγή.....	57
6.2	Δοκιμές Εφαρμογής	57
6.2.1	Δοκιμές Λειτουργικότητας.....	57
6.2.2	Δοκιμές χρηστικότητας	58
6.3	Αξιολόγηση απόδοσης	59
6.4	Συζήτηση αποτελεσμάτων.....	60
6.5	Επίλογος.....	60
Κεφάλαιο 7ο:	Συμπεράσματα και Μελλοντικές Επεκτάσεις.....	61
	ΒΙΒΛΙΟΓΡΑΦΙΑ.....	62

Κατάλογος Σχημάτων

Σχήμα 4.1: Διάγραμμα Αρχιτεκτονικής	12
Σχήμα 4.2: Διάγραμμα ροής Εισόδου Χρήστη (Μετά την Συναίνεσή του)	13
Σχήμα 4.3: Screenshot Οθόνης Ιατρικών	16
Σχήμα 4.4: Screenshot Οθόνης Κεντρικού Πλέγματος	20
Σχήμα 4.5: Screenshot Οθόνης Ημερολογίου	29
Σχήμα 4.6: Διάγραμμα Σχέσεων Ιστορικού	35
Σχήμα 4.7: Διάγραμμα Βάσης Δεδομένων	38

Συντομογραφίες

VSCoDe	Visual Studio Code
SSL	Secure Sockets Layer
TLS	Transport Layer Security
Δ.Ε.	Διπλωματική Εργασία
IEE	Information and Electronic Engineering
IHU	International Hellenic University
AES	Advanced Encryption Standard
API	Application Programming Interface
APK	Android Package Kit
CSS	Cascading Style Sheets
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
HTTPS	Hyper Text Transfer Protocol Secure
JSON	JavaScript Object Notation
SQL	Structured Query Language
NoSQL	Not Only SQL
OS	Operating System
PDF	Portable Document Format
PHR	Personal Health Record
SDK	Software Development Kit
UI	User Interface
UX	User Experience
FHIR	Fast Healthcare Interoperability Resources
HL7	Health Level Seven

Κεφάλαιο 1ο: Εισαγωγή

1.1 Εισαγωγή

Η παρούσα Διπλωματική Εργασία αφορά την ανάπτυξη μιας εφαρμογής διαχείρισης προσωπικών ιατρικών δεδομένων για κινητές συσκευές.

1.2 Αντικείμενο και Στόχοι της Εργασίας

Η εργασία αυτή ασχολείται με τον ιατρικό τομέα και συγκεκριμένα επικεντρώνεται στον τρόπο με τον οποίο τον αντιλαμβάνεται ο μέσος άνθρωπος και τους τρόπους με τους οποίους αλληλοεπιδρά με αυτόν. Στις περισσότερες περιπτώσεις, κύρια απασχόληση ενός τέτοιου ατόμου είναι μόνο τα τμήματα του τομέα που τον επηρεάζουν. Αυτό μπορεί να είναι συνηθισμένες καταστάσεις, όπως κοινές ασθένειες, φάρμακα και επισκέψεις σε γιατρούς, ή πιο σπάνια, σοβαρότερα γεγονότα όπως εξετάσεις και εγχειρήσεις.

Ανεξάρτητα από το είδος της περίπτωσης που ενδιαφέρει κάθε άτομο, θα χρειαστεί να καταχωρηθούν κάποιες πληροφορίες σχετικά με αυτήν. Οι ημερομηνίες και οι ώρες προγραμματισμένων ραντεβού αποτελούν την κύρια περίπτωση καταχώρισης, αλλά η καταγραφή των δεδομένων μπορεί να απαιτείται και για γεγονότα όπως οι ασθένειες που παρατηρήθηκαν και οι ημερομηνίες των παρατηρήσεων αυτών, ή οι μετρήσεις ζωτικών σημείων του ατόμου.

Η αποθήκευση του περιεχομένου ιατρικών εγγράφων, όπως ιατρικές συνταγές, ή αποτελέσματα ιατρικών εξετάσεων, είναι άλλο ένα παράδειγμα χρήσης της εφαρμογής. Οι πληροφορίες που περιλαμβάνονται σε τέτοια έγγραφα πρέπει, αφενός μεν να αποθηκευτούν με ασφάλεια, αφ' ετέρου δε να είναι εύκολα διαθέσιμες, σε κάθε περίπτωση που αυτό είναι επιθυμητό.

Έτσι, στόχος της εργασίας αυτής είναι η δημιουργία μιας εφαρμογής που θα επιτρέπει σε ένα άτομο να ικανοποιεί όλες αυτές τις ανάγκες. Θα μπορεί να αποθηκεύσει τα ιατρικά του δεδομένα με ασφάλεια και να τα διαχειρίζεται ελεύθερα μέσω της εφαρμογής. Ταυτόχρονα, η εφαρμογή θα τον κρατάει ενήμερο για όλα τα σχετικά γεγονότα που έχει προγραμματίσει και για τα δεδομένα που έχει καταθέσει, χρησιμοποιώντας απλές και εύκολα κατανοητές μεθόδους παρουσίασής τους.

1.3 Κίνητρο και Σημασία Διαχείρισης Προσωπικών Ιατρικών Δεδομένων

Όσον αφορά την υγεία ενός ατόμου, εμφανίζεται μεγάλη ποικιλία στα είδη των προβλημάτων που μπορεί αυτό να αντιμετωπίσει. Το γεγονός αυτό οδηγεί στην πιθανή ανάγκη για εξίσου πολλές υπηρεσίες στον ιατρικό τομέα, η κάθε μία από τις οποίες απαιτεί την καταγραφή διαφορετικών δεδομένων. Σε κάποιες περιπτώσεις, μπορεί να είναι απαραίτητο να είναι διαθέσιμο ολόκληρο το ιστορικό υγείας του ατόμου, ενώ σε άλλες μπορεί να απαιτείται τμήμα των δεδομένων από μια συγκεκριμένη κατηγορία. Για παράδειγμα, αν στο ιστορικό του ατόμου σημειώνεται μια σοβαρή πάθηση, αυτό μπορεί να πρέπει να ληφθεί υπόψιν κατά τη διάρκεια μιας εξέτασης.

Ο όγκος δεδομένων, που χρειάζεται να καταγραφεί, μπορεί έτσι να μεγαλώσει σε σημείο όπου δεν είναι λογικό να αναμένεται κάποιος να τα διατηρεί όλα ως γραπτά έγγραφα. Η αποθήκευσή του συνόλου των εγγράφων από μόνη της θα μπορούσε να δημιουργήσει δυσκολίες, ενώ η αναζήτηση για συγκεκριμένα έγγραφα ανάμεσά τους θα ήταν μία αργή διαδικασία. Έτσι, η ψηφιοποίησή τους γίνεται το επόμενο

λογικό βήμα, που σημαίνει ότι χρειάζονται και υπηρεσίες αποθήκευσης και διαχείρισης αυτών των δεδομένων.

Με αυτό τον τρόπο, τα προσωπικά ιατρικά δεδομένα του κάθε ατόμου είναι συγκεντρωμένα σε μία βάση δεδομένων, προστατευμένα από τις διαδικασίες ταυτοποίησής της, καθώς και από τις μεθόδους κρυπτογράφησης που χρησιμοποιούνται σε αυτήν. Από τη βάση δεδομένων, οι πληροφορίες είναι εύκολα διαθέσιμες στο χρήστη, μέσω της αντίστοιχης εφαρμογής, αρκεί να υπάρχει δυνατότητα σύνδεσης με τη βάση. Επιπλέον, η χρήση βάσης δεδομένων καθιστά την αναζήτηση συγκεκριμένων εγγράφων πιο εύκολη, κάτι το οποίο μπορεί να απλουστευθεί περισσότερο με τη σωστή υλοποίηση της εφαρμογής που χρησιμοποιείται.

1.4 Συμβολή της εργασίας

Με τη χρήση της εφαρμογής που παρουσιάζεται σε αυτήν την εργασία, ένα άτομο μπορεί να έχει γρήγορη πρόσβαση στα προσωπικά ιατρικά δεδομένα του από οπουδήποτε, εφόσον έχει πρόσβαση στο Διαδίκτυο. Μπορεί να δει τα ραντεβού του στο δρόμο, ή να παρουσιάσει το ιστορικό παθήσεών του στο γιατρό του, χωρίς να έχει προετοιμαστεί για αυτό, να ελέγξει πότε έκανε μια συγκεκριμένη εξέταση για τελευταία φορά και άλλες παρόμοιες λειτουργίες.

Με τον τρόπο αυτό η εφαρμογή εξοικονομεί χρόνο και ενέργεια για το χρήστη. Δε χρειάζεται να οργανώνει ο ίδιος όλα τα σχετικά δεδομένα σε χρονολογική σειρά ή να έρχεται στο γιατρό προετοιμασμένος, είτε μεταφέροντας, είτε έχοντας απομνημονεύσει ολόκληρο το πρόσφατο ιατρικό ιστορικό του. Όλες αυτές οι πληροφορίες του παρέχονται εύκολα και γρήγορα, μέσω των υπηρεσιών της εφαρμογής.

1.5 Δομή της Διπλωματικής Εργασίας

Σε αυτή τη εργασία, θα αναφερθούν πρώτα οι τεχνολογίες που έδρασαν ως έμπνευση για την εφαρμογή, καθώς και αυτές που χρησιμοποιήθηκαν για την υλοποίησή της. Η αναφορά αυτή θα περιλαμβάνει τεχνολογίες για διάφορους σκοπούς, όπως την ανάπτυξη της ίδιας της εφαρμογής, την διαχείριση και ασφάλεια των δεδομένων που καταθέτονται, την οπτικοποίησή τους, ακόμη και τα πρότυπα διαλειτουργικότητας που χρησιμοποιούνται στην εφαρμογή **για αυτά**.

Θα ακολουθήσει μία αναλυτική εξήγηση των απαιτήσεων για την υλοποίηση της εφαρμογής και μερικά σενάρια χρήσης για αυτήν. Μαζί με αυτό θα παρουσιαστεί και μία γενική περιγραφή των λειτουργιών της εφαρμογής.

Στη συνέχεια θα γίνει μια λεπτομερής εξήγηση του συστήματος, καθώς και των μεθόδων υλοποίησης της εφαρμογής. Αυτές θα περιλαμβάνουν κάθε κομμάτι της εφαρμογής, από τη δική της δομή και αυτή της βάσης δεδομένων, μέχρι και του περιβάλλοντος ανάπτυξης, που χρησιμοποιήθηκε κατά την υλοποίησή της.

Τέλος, θα παρουσιαστεί μία αξιολόγηση της εφαρμογής και τα τελικά συμπεράσματα που προέκυψαν μετά την ολοκλήρωσή της. Θα εξεταστεί τι λειτουργήσε καλά, τι ελαττώματα και προβλήματα βρέθηκαν στην εφαρμογή, καθώς και το τι θα μπορούσε να βελτιωθεί σε μελλοντικές εκδόσεις της.

1.6 Επίλογος

Η παρούσα εργασία στοχεύει στη δημιουργία μιας εφαρμογής για την καλύτερη διαχείριση των ιατρικών δεδομένων του μέσου ατόμου. Τα δεδομένα αυτά θα συγκεντρώνονται σε μία βάση δεδομένων, στην οποία θα έχουν πρόσβαση οι χρήστες από τις κινητές συσκευές τους, καθιστώντας την

Κεφάλαιο 1

ανάκτηση πληροφοριών εύκολη και γρήγορη, από οπουδήποτε. Στη συνέχεια αυτού του κειμένου, το θέμα θα αναλυθεί με περισσότερες λεπτομέρειες, από διάφορες οπτικές γωνίες.

Κεφάλαιο 2ο: Υπόβαθρο και Σχετικές Τεχνολογίες

2.1 Εισαγωγή

Η εφαρμογή αυτή, βασισμένη σε άλλες υπάρχουσες εφαρμογές στον τομέα, χρησιμοποιεί τεχνολογίες από πολλές πηγές, για τις διάφορες λειτουργίες της. Για να είναι ικανή να στηρίξει αυτές τις τεχνολογίες, καθώς και για να τις δέσει όλες σε ένα σύστημα, χρειάζεται μία υποδομή, η οποία να διαθέτει ευελιξία στο έργο αυτό.

Ιδιαίτερη σημασία δίνεται στην ικανότητα της υποδομής να διατηρήσει την ενσωμάτωση των διάφορων τεχνολογιών ομοιόμορφη και απλή. Πρέπει δηλαδή να είναι εύκολα συμβατή με αυτές, χωρίς την απαίτηση για ογκώδεις υπηρεσίες και ενδιάμεσους, σε μία εφαρμογή προορισμένη για κινητές συσκευές, οι οποίες μπορεί να μη έχουν ιδιαίτερα μεγάλο χώρο αποθήκευσης.

2.2 Περιεχόμενο

2.2.1 Mobile εφαρμογές στον τομέα της υγείας

Έμπνευση για αυτή την εργασία αποτέλεσε η εφαρμογή MyHealth [4]. Η εφαρμογή αυτή, που εκδίδεται από την ελληνική κυβέρνηση, παρέχει πολλές σημαντικές υπηρεσίες στο μέσο πολίτη, που διευκολύνουν σημαντικά την διαχείριση των ιατρικών του δεδομένων, καθώς και άλλους τομείς της εμπειρίας τους, όσον αφορά την επαφή τους με τον ιατρικό τομέα.

Επιτρέπει την καταγραφή ραντεβού του χρήστη και τον κρατάει ενήμερο για αυτά μέσω ενημερώσεων που δημιουργεί, όσο βρίσκεται σε λειτουργία. Επιπλέον, σε συνεργασία με τη βάση δεδομένων της κυβέρνησης, μπορεί και να κλείσει ραντεβού για μερικές επιπτώσεις.

Πέρα από αυτό, παρουσιάζει στο χρήστη διάφορα έγγραφα, που συντάσσονται από γιατρούς σχετικά με αυτόν. Τέτοια έγγραφα περιλαμβάνουν συνταγές για φάρμακα, παραπεμπτικά για εξετάσεις ή άλλου είδους διαδικασίες, καθώς και διαγνώσεις. Τα έγγραφα αυτά εμφανίζονται αυτόματα από την οπτική του χρήστη, καθώς εισάγονται από τους σχετικούς γιατρούς.

2.2.2 Διαχείριση προσωπικών ιατρικών δεδομένων

Η διαχείριση των δεδομένων του χρήστη γίνεται μέσω της υπηρεσίας Cloud Firestore [5] του Firebase [6]. Αυτή η υπηρεσία, στη οποία έχει πρόσβαση ο χρήστης με τη χρήση του «cloud_firestore» πακέτου [7], παρέχει μία βάση δεδομένων για την αποθήκευση αυτών, καθώς και τον χώρο, όπου γίνεται η τροποποίηση και η διαγραφή τους, όταν εκτελούνται οι σχετικές εντολές από την εφαρμογή.

Για τη γρήγορη δημιουργία του προφίλ του, το email του χρήστη αποθηκεύεται στις κοινές προτιμήσεις της εφαρμογής. Υπεύθυνο για αυτή τη λειτουργία είναι το πακέτο «shared_preferences» [8], το οποίο επιτρέπει την καταγραφή δεδομένων απλής μορφής για την εφαρμογή.

Ανάμεσα σε άλλες λειτουργίες της, η εφαρμογή επιτρέπει στο χρήστη να μεταφέρει δεδομένα του από το Google Health Connect σε αυτήν, καθώς και να μεταφέρει δεδομένα του πίσω σε αυτό στην περίπτωση διαγραφής του λογαριασμού του. Αυτό επιτυγχάνεται με τη βοήθεια του πακέτου «health» [9]. Το πακέτο αυτό αναλαμβάνει τη αίτηση των αδειών που χρειάζεται από το Google Health Connect για πρόσβαση στα δεδομένα του, είτε πρόκειται για ανάγνωσή τους ή για καταγραφή καινούργιων αντικειμένων σε αυτό.

2.2.3 Θέματα ασφάλειας και προστασίας προσωπικών δεδομένων

Τα δεδομένα του χρήστη αποθηκεύονται στη βάση δεδομένων. Οποιαδήποτε δεδομένα εισάγονται, που μπορούν να θεωρηθούν προσωπικά ή εμπιστευτικά, ειδικά αυτά που σχετίζονται με τον ιατρικό τομέα, περνούν πρώτα από μία διαδικασία κρυπτογράφησης, πριν αποσταλούν από τη συσκευή.

Για την κρυπτογράφηση αυτών των δεδομένων χρησιμοποιείται η μέθοδος AES-256, η οποία θεωρείται μια από τις πιο ασφαλείς μεθόδους κρυπτογράφησης που υπάρχουν. Η μέθοδος αυτή χωρίζει το κείμενο που εισάχθηκε σε τμήματα μεγέθους 128-bit και τα κρυπτογραφεί με βάση ενός 256-bit κλειδιού. Είναι αναγνωρισμένη παγκοσμίως για την ασφάλειά της και τη δυσκολία της παραβίασής της, με την έννοια ότι η προσπάθεια αποκρυπτογράφησης των δεδομένων χωρίς κλειδί, δηλαδή με εξαντλητική δοκιμή κλειδιών, που μπορεί να πραγματοποιήσει η μέθοδος κατά την κρυπτογράφηση, θα έπαιρνε υπερβολικά πολύ χρόνο, για τα δεδομένα να έχουν ακόμα κάποια χρησιμότητα.

Ενσωμάτωση και χρήση της AES-256 στην εφαρμογή γίνεται μέσω του πακέτου Flutter «aes256 2.2.2» [10]. Το πακέτο αυτό μετατρέπει μία αποθηκευμένη φράση-κλειδί σε ένα 256-bit κλειδί, για την κρυπτογράφηση και αποκρυπτογράφηση δεδομένων. Επιπλέον, παρέχει στην εφαρμογή έτοιμες συναρτήσεις για την κρυπτογράφηση και την αποκρυπτογράφηση δεδομένων με αυτή τη μέθοδο.

Το κλειδί που χρησιμοποιείται από την εφαρμογή βρίσκεται αποθηκευμένο σε ένα αρχείο «.env», από το οποίο ανακαλείται μέσω του πακέτου «flutter_dotenv» [11]. Η κύρια λειτουργία αυτού το πακέτου είναι ακριβώς αυτό, η φόρτωση τιμών από πόρους της εφαρμογής αυτού του τύπου, σε οποιοδήποτε μέρος του κώδικα απαιτείται.

Πέρα από αυτό, η ίδια η βάση δεδομένων παρέχει προστασία στα δεδομένα του χρήστη. Η εφαρμογή υλοποιείται με τη χρήση του Firebase [6] για τη βάση δεδομένων, η οποία δεν περιορίζεται μόνο στην αποθήκευση και διαχείριση δεδομένων που εισάγει ο χρήστης. Περιλαμβάνει και υπηρεσίες καταγραφής νέων χρηστών και πιστοποίησής τους [12], είτε μέσω του email, με το οποίο κάνουν εγγραφή στην εφαρμογή, είτε με κάποιον πάροχο που έχει οριστεί από το δημιουργό της εργασίας. Στην περίπτωση αυτή, ο μόνος πάροχος που επιτρέπεται από την εφαρμογή είναι η Google.

Η υλοποίηση αυτής της μεθόδου σύνδεσης γίνεται με τη βοήθεια δύο διαφορετικών πακέτων, το «firebase_ui_auth» [13] και το «firebase_ui_oauth_google» [14]. Το πρώτο από αυτά παρέχει μία συλλογή από υπηρεσίες και Widget για την ενσωμάτωση της λειτουργίας πιστοποίησης του Firebase στην εφαρμογή, όπως το Widget οθόνης εισαγωγής, το οποίο χρησιμοποιείται για την εγγραφή και, αργότερα, την είσοδο του χρήστη στην εφαρμογή. Αποτελεί έτσι τη βάση του συστήματος πιστοποίησης του χρήστη. Για τη χρήση της ίδιας της υπηρεσίας απαιτείται και η χρήση του «firebase_auth» [15].

Όσο για το «firebase_ui_oauth_google» [14], αποτελεί ένα πιο συγκεκριμένο πακέτο ως προς τις λειτουργίες του. Για την ακρίβεια, χρησιμοποιείται για τον ορισμό της Google ως έναν αποδεκτό πάροχο στην εφαρμογή. Αυτό επιβεβαιώνει και από την πλευρά της εφαρμογής ότι ο χρήστης μπορεί να συνδεθεί μέσω του λογαριασμού Google του, ενώ από την πλευρά του το Firebase [12] λαμβάνει αυτή την απόπειρα σύνδεσης και την αποδέχεται, καθώς η επιλογή χρήσης της είναι ενεργοποιημένη και εκεί.

Τελευταία τεχνολογία σχετική με την ασφάλεια του χρήστη είναι η δυνατότητα χρήσης βιομετρικών μετρήσεων για την αναγνώρισή του από την εφαρμογή. Αυτό γίνεται με τη βοήθεια του πακέτου «local_auth» [16], που παρέχει στην εφαρμογή την ικανότητα να ελέγξει τη συσκευή για την παρουσία βιομετρικών μετρήσεων και να τις χρησιμοποιήσει. Οι μετρήσεις αυτές μπορεί να είναι δαχτυλικό αποτύπωμα ή σάρωση προσώπου.

2.2.4 Τεχνολογίες ανάπτυξης mobile εφαρμογών

Η εφαρμογή υλοποιήθηκε με τη χρήση του Flutter [17], μία υποδομή ανάπτυξης εφαρμογών βασισμένη στη γλώσσα προγραμματισμού Dart. Κύριο χαρακτηριστικό της είναι ότι απλουστεύει τη δημιουργία εφαρμογών, ώστε αυτή να γίνεται με βάση το UI, που προσπαθεί να φτιάξει ο προγραμματιστής χρησιμοποιώντας δομές στοιχείων UI, τα Widgets.

Επιπλέον, παρέχει στον προγραμματιστή τη λειτουργία «hot reload», ένα αξιοσημείωτο πλεονέκτημα για τη διαδικασία σχεδίασης και δοκιμής μιας εφαρμογής. Με τη χρήση της κατά το τρέξιμο της εφαρμογής, γίνεται δυνατόν για τον προγραμματιστή να τροποποιεί τη λογική και το UI της εφαρμογής, την ώρα που το τρέχει και να παρατηρεί άμεσα τα αποτελέσματα αυτών των αλλαγών.

Το Flutter υποστηρίζεται από την Google, με ένα μεγάλο πλήθος έτοιμων πακέτων και βιβλιοθηκών ειδικά σχεδιασμένων για αυτό. Σε αυτά περιλαμβάνονται και όλα τα πακέτα που αναφέρονται σε αυτό το κείμενο.

Όσο για τη βάση δεδομένων, όπως αναφέρθηκε στο 2.2.2, χρησιμοποιήθηκε το Cloud Firestore [5] του Firebase [6], ενώ η πιστοποίηση χρήστη σε αυτήν υλοποιήθηκε με τη βοήθεια της υπηρεσίας Firebase Authentication [12]. Οι υπηρεσίες αυτές γίνονται διαθέσιμες μέσω έτοιμων πακέτων για το Flutter, αλλά απαιτούνται και δύο ακόμα διαδικασίες για τη χρήση του Firebase [6] από την εφαρμογή.

Συγκεκριμένα, μία εργασία Flutter απαιτεί την προσαρμογή της για τη συνεργασία με το Firebase [6] μέσω του Firebase CLI [18], καθώς και την εγκατάσταση του κεντρικού πακέτου για τη σύνδεση στο Firebase [6], το «firebase_core» [19].

Έχοντας ολοκληρώσει όλα αυτά τα βήματα, όλες οι υπηρεσίες του Firebase [6] στις οποίες έχει πρόσβαση ο προγραμματιστής είναι διαθέσιμες και στην εφαρμογή, εφόσον έχει εγκατασταθεί το κατάλληλο πακέτο για αυτό. Όπως θα φανεί και παρακάτω με περισσότερες λεπτομέρειες, αυτό αποτελεί κρίσιμο τμήμα των περισσότερων λειτουργιών της εφαρμογής.

2.2.5 Γραφήματα και οπτικοποίηση δεδομένων υγείας

Δύο λειτουργίες της εφαρμογής, όπου χρησιμοποιήθηκαν επιπλέον τεχνολογίες, ήταν η δημιουργία ημερολογίου, για την οπτικοποίηση δεδομένων του χρήστη, καθώς και η σχεδίαση γραφημάτων για συγκεκριμένες κατηγορίες δεδομένων.

Την πρώτη λειτουργία ολοκληρώνει το πακέτο Flutter «table_calendar» [20]. Ο σκοπός αυτού είναι η δημιουργία και τροποποίηση ενός κομψά μορφοποιημένου ημερολογίου. Οι πιο απλές υλοποιήσεις του αποτελούνται απλά από ένα ημερολόγιο με εύρος από μια ορισμένη ημερομηνία έως και μία άλλη, με μία εστιασμένη ημέρα σημειωμένη σε αυτό.

Οι δυνατότητές του, όμως, δεν περιορίζονται σε αυτό μόνο. Ο προγραμματιστής μπορεί να τροποποιήσει την εμφάνιση του ημερολογίου, καθώς και αυτή των ημερών, όπως προβάλλονται στις σελίδες του. Μπορεί επίσης να προσθέσει και την ικανότητα να επιλέγει ο χρήστης ημέρες στις οποίες θα εστιάζει το σύστημα, κάτι το οποίο μπορεί να χρησιμοποιηθεί για διάφορων ειδών γεγονότα και λειτουργίες. Ένα απλό παράδειγμα τέτοιων γεγονότων θα ήταν η εμφάνιση άλλων στοιχείων στην οθόνη με βάση το αν η ημέρα που επιλέχθηκε ήταν χρωματισμένη πράσινη ή κόκκινη.

Ως προς τη δεύτερη λειτουργία οπτικοποίηση, η σχεδίαση γραφημάτων γίνεται με τη βοήθεια του «fl_chart» [21], ένα πακέτο για τη δημιουργία πολλαπλών διαφορετικών ειδών γραφημάτων με μεγάλο βαθμό ελευθερίας στο σχεδιασμό της εμφάνισής τους. Παρέχει υποστήριξη για γραμμικά γραφήματα, γραφήματα μπάρας, πίτας, καθώς και άλλα. Η χρήση τους σε αυτή την εφαρμογή περιορίζεται μόνο σε

συγκεκριμένα είδη δεδομένων και μόνο για συγκεκριμένους σκοπούς, όπως για την εμφάνιση των μετρήσεων ζωτικών σημείων στο χρόνο.

2.2.6 Πρότυπα διαλειτουργικότητας ιατρικών δεδομένων

Προκειμένου να μπορούν να χρησιμοποιηθούν τα ιατρικά δεδομένα του χρήστη και σε άλλα συστήματα, στα οποία μπορεί να θέλει να τα παραδώσει, είναι σημαντικό να μπορούν να παρουσιαστούν σε μορφή που θα γίνει αποδεκτή από αυτά. Για αυτό το λόγο, υπάρχουν πρότυπα που ορίζουν μερικές προδιαγραφές, για τη μορφοποίηση αυτών των δεδομένων, για γενική χρήση από συστήματα στον τομέα της υγείας.

Από αυτά, χρησιμοποιούνται δύο στην εφαρμογή, με το πρώτο να είναι το HL7. Το πρότυπο αυτό είναι ένα παλιότερο μοντέλο που βρίσκει ακόμα ευρεία χρήση στα περισσότερα συστήματα νοσοκομείων. Στην εφαρμογή εφαρμόζεται ακολουθώντας απλά την προτεινόμενη δομή [1] κατά τη διατύπωση του κειμένου.

Το άλλο πρότυπο που χρησιμοποιείται είναι το FHIR, μία παραλλαγή του HL7 σχεδιασμένη για την μεταφορά σε ηλεκτρονική μορφή μεταξύ διαφορετικών συστημάτων. Όπως και με τη χρήση του HL7, η προτεινόμενη δομή κειμένου [2] εφαρμόζεται για τη μετατροπή των δεδομένων του χρήστη, ώστε να ακολουθεί το πρότυπο, με τη διαφορά ότι χρησιμοποιείται ένα ακόμη πακέτο Flutter, το «fhir_r4» [3], για τη δημιουργία ενός συγκεκριμένου τύπου αντικείμενου, το οποίο δρα ως η δομή για το κείμενο. Στη συνέχεια, το αντικείμενο αυτό μετατρέπεται σε κείμενο JSON, με τη χρήση μιας από τις διαδικασίες που παρέχει το πακέτο.

2.3 Επίλογος

Στην εργασία αυτή χρησιμοποιείται το Flutter ως υποδομή, λόγω της απλότητάς του στο σχεδιασμό της εφαρμογής, καθώς και της υποστήριξης που λαμβάνει από όλα τα διάφορα πακέτα που μπορούν να ενσωματωθούν σε εφαρμογές που το χρησιμοποιούν. Αυτό με τη σειρά του επιτρέπει στη χρήση διάφορων από αυτά τα πακέτα για την υλοποίηση των λειτουργιών της εφαρμογής.

Για τη βάση δεδομένων και την πιστοποίηση του χρήστη, χρησιμοποιούνται οι υπηρεσίες του Firebase [6], ένα εύκολα διαθέσιμο σύστημα που υποστηρίζει το Flutter και μπορεί εύκολα να ενσωματωθεί σε αυτό [15]. Τα δεδομένα του χρήστη μπορούν μετά να ανακληθούν από την εφαρμογή και να χρησιμοποιηθούν για διάφορους σκοπούς, όπως για τη σχεδίαση ενός γραφήματος [21] είτε ως μορφοποιημένο κείμενο για κάποιο έγγραφο.

Το Flutter, έτσι, γίνεται η βάση στην οποία χτίζονται όλα τα άλλα τμήματα της εφαρμογής και στο οποίο προστίθενται όλες οι άλλες σχετικές τεχνολογίες στην εφαρμογή.

Κεφάλαιο 3ο: Ανάλυση Απαιτήσεων

3.1 Εισαγωγή

Η εφαρμογή, στην οποία αναφέρεται αυτή η αναφορά, ασχολείται με τον ιατρικό τομέα. Συγκεκριμένα, αφορά την αποθήκευση και διαχείριση ιατρικών δεδομένων από το μέσο άτομο. Σε συνδυασμό με αυτό, προσφέρει διάφορες λειτουργίες οπτικοποίησης αυτών των δεδομένων, με σκοπό τη διευκόλυνση της εμπειρίας του χρήστη. Η υλοποίηση αυτής της ιδέας οδηγεί στην υιοθεσία διαφορετικών μεθόδων, οι οποίες με τη σειρά τους δημιουργούν τις δικές τους απαιτήσεις. Αυτό αποτελεί τη βασική πρόκληση της δημιουργίας της εφαρμογής αυτής.

3.2 Έννοιες-Κλειδιά

3.2.1 Σκοπός και λειτουργίες της εφαρμογής

Στόχος της εφαρμογής είναι να παρέχει στο χρήστη την ικανότητα να καταγράφει, να παρακολουθεί και να διαχειρίζεται τα προσωπικά ιατρικά του δεδομένα, ανεξάρτητα από το είδος τους και για όσα περισσότερα είδη δεδομένων είναι δυνατόν. Επιπλέον, προσπαθεί να προβάλει τα δεδομένα αυτά με τρόπο εύκολα κατανοητό από το χρήστη.

Προς αυτό το σκοπό επιτρέπει την κατάθεση δεδομένων χρησιμοποιώντας διάφορες φόρμες, για το κάθε είδος δεδομένων που προβλέπεται από την εφαρμογή. Τα δεδομένα μετά θα αποθηκευτούν σε μία απομακρυσμένη βάση δεδομένων, υπό την ταυτότητα του χρήστη, με την κάθε κατηγορία δεδομένων να αντιστοιχεί και σε μία διαφορετική συλλογή στη βάση, όπου θα αποθηκευτούν.

Οι κατηγορίες αυτές περιλαμβάνουν παθήσεις και ασθένειες του χρήστη, τη φαρμακευτική αγωγή, που μπορεί να λαμβάνει, καθώς και ξεχωριστές κατηγορίες για τα είδη ραντεβού, τα οποία προβλέπεται να μπορεί να προγραμματίσει, που είναι ραντεβού εξετάσεων ή εγχειρήσεων για συγκεκριμένες περιπτώσεις, καθώς και μία τρίτη συλλογή για οποιοδήποτε άλλο είδος ραντεβού.

Τα αποτελέσματα των εξετάσεων ή εγχειρήσεων μπορούν επίσης να αποθηκευτούν, αλλά σε διαφορετικές συλλογές. Παρόμοια λειτουργία υπάρχει και για την αποθήκευση οποιουδήποτε άλλου είδους αρχείου PDF, που δεν ανήκει σε αυτές τις κατηγορίες.

Σε περίπτωση που τα δεδομένα αυτά απαιτούν ενημέρωση του χρήστη, όπως για ένα ραντεβού που εκκρεμεί, η εφαρμογή αναλαμβάνει και αυτό το ρόλο, στέλνοντάς του ειδοποιήσεις στις σχετικές στιγμές. Όσο για το πότε θα λάβει αυτές τις ειδοποιήσεις ή για τον τρόπο με τον οποίο θα σταλούν, αυτό εξαρτάται από το είδος του αντικειμένου και τις τιμές που περιέχει αυτό.

Επιπλέον κατηγορίες υπάρχουν για τις μετρήσεις ζωτικών σημείων, που μπορεί να έχει πάρει ο χρήστης, τους στόχους που έχει θέσει για αυτές τις μετρήσεις, καθώς και για τις πληροφορίες εκτάκτου ανάγκης. Οι πληροφορίες αυτές είναι διαθέσιμες μέσω ειδοποιήσεων ή από κάποια σχετική οθόνη.

Τέλος, προκειμένου να παραμένει ο χρήστης ενήμερος και πληροφορημένος για την υγεία του, η εφαρμογή του παρέχει μηνιαίες αναφορές για διάφορα θέματα υγείας του. Αυτό περιλαμβάνει το πλήθος παθήσεών του τον τελευταίο μήνα, καθώς και τις τελευταίες του μετρήσεις σε σχέση με τους στόχους που έχει θέσει.

3.2.2 Σενάρια Χρήσης

Η εφαρμογή, όπως αναφέρθηκε ήδη, παρέχει διάφορες λειτουργίες στο χρήστη για διαφορετικές περιπτώσεις, όλες διαθέσιμες όποτε αυτός θέλει. Ένας χρήστης θα μπορούσε να θέλει μόνο να καταγράψει τα ραντεβού του και να ενημερωθεί όταν έρχεται η ημέρα για κάθε ένα. Η εφαρμογή τα αποθηκεύει στη βάση δεδομένων και στέλνει ειδοποίηση εάν βρει ένα ραντεβού κατά την εκκίνηση που εκκρεμεί για εκείνη την ημέρα.

Εναλλακτικά, θα μπορούσε να θέλει να δει όλα τα ραντεβού του. Η εφαρμογή μπορεί να προβάλει όλα τα καταγεγραμμένα ραντεβού από τη βάση, ανεξάρτητα από την κατηγορία τους, σε ένα ημερολόγιο, ή να προβάλει μόνο μία κατηγορία κάθε φορά, σε μία άλλη οθόνη ως λίστα. Στην περίπτωση της προβολής με λίστα, προσφέρεται επιπλέον στο χρήστη η ικανότητα να τροποποιήσει τα καταγεγραμμένα δεδομένα του, ή να τα διαγράψει εντελώς.

Στο τέλος μίας εξέτασης ή εγχείρησης, ο χρήστης μπορεί επίσης να καταγράψει και τα αποτελέσματά του στη βάση δεδομένων της εφαρμογής. Αν μετά χρειαστεί την πληροφορία, μπορεί να κατεβάσει τα δεδομένα ως κείμενο σε ένα έγγραφο στη συσκευή του.

Παρόμοια μέθοδος παρακολούθησης υπάρχει και για τις μετρήσεις ζωτικών σημείων, για κάποιο χρήστη που πρέπει να δίνει τακτική προσοχή σε αυτό, για παράδειγμα λόγω χαμηλού ή υψηλού σακχάρου. Εφόσον οι μετρήσεις καταγραφούν, ο χρήστης μπορεί μετά να τις δει σε μία λίστα, από όπου μπορεί να τροποποιήσει ή να διαγράψει δεδομένα. Αν θέλει, η εφαρμογή μπορεί να τις συγκεντρώσει σε γραφήματα [21], δείχνοντάς του την πορεία της υγείας του στο χρόνο.

3.2.3 Λειτουργικές απαιτήσεις

Καθώς η εφαρμογή είναι σχεδιασμένη να αποθηκεύει δεδομένα από πολλούς διαφορετικούς χρήστες, είναι απαραίτητο να υπάρχει τρόπος να τους ξεχωρίσει μεταξύ τους. Η πιο αποτελεσματική μέθοδος για αυτό, μιας και χρησιμοποιείται απομακρυσμένη βάση δεδομένων και όχι τοπική, είναι η δημιουργία λογαριασμού για την πιστοποίηση και αναγνώριση του κάθε χρήστη.

Η δημιουργία λογαριασμού οδηγεί επίσης στην ανάγκη για το χρήστη να μπορεί να συνδεθεί στο λογαριασμό του. Τα δεδομένα του κάθε χρήστη αναμένεται να παραμείνουν αποθηκευμένα, για όσο αυτός επιθυμεί, αντί για μόνο μία χρήση της εφαρμογής, οπότε δεν γίνεται να δημιουργούνται προσωρινές συνεδρίες, άρα ο λογαριασμός πρέπει να είναι μόνιμος. Εναλλακτικά, ο χρήστης πρέπει να μπορεί να συνδεθεί και μέσω λογαριασμού Google.

Ταυτόχρονα όμως, ο λογαριασμός του χρήστη δεν μπορεί να είναι πραγματικά μόνιμος. Ο χρήστης έχει δικαίωμα στη διαγραφή των δεδομένων του, άρα η εφαρμογή χρειάζεται να έχει και λειτουργία για τη διαγραφή του λογαριασμού του.

Σε περίπτωση που ο χρήστης επιθυμεί περισσότερη ασφάλεια, η εφαρμογή πρέπει να υποστηρίζει και τη χρήση ταυτοποίησης μέσω βιομετρικών μετρήσεων. Η λειτουργία αυτή πρέπει να είναι διαθέσιμη σε συσκευές που έχουν αυτή τη δυνατότητα ενεργοποιημένη και μία μέτρηση αποθηκευμένη, είτε πρόκειται για δαχτυλικό αποτύπωμα είτε για σάρωση του προσώπου.

Πέρα από αυτό, η εφαρμογή πρέπει να μπορεί να διακρίνει μεταξύ εγγράφων σε μια συλλογή, για τις περιπτώσεις όπου χρειάζεται να κάνει αλλαγές σε ένα και μόνο ένα από αυτά. Επομένως, τα ίδια τα έγγραφα πρέπει να περιέχουν μοναδικά αναγνωριστικά.

Τα αναγνωριστικά αυτά οδηγούν στην επόμενη ανάγκη για τη λειτουργία της εφαρμογής, την ικανότητα να τροποποιεί τα δεδομένα, που ανήκουν στο χρήστη, στη βάση δεδομένων. Συγκεκριμένα, η εφαρμογή

πρέπει να παρέχει στο χρήστη τη δυνατότητα να δει τα δεδομένα του και να τα τροποποιήσει όπως αυτός θέλει, ή ακόμη και να τα διαγράψει.

Επιπλέον, μερικές λειτουργίες της εφαρμογής δημιουργούν δύο ακόμα απαιτήσεις. Η εφαρμογή πρέπει να μπορεί να δημιουργεί αρχεία PDF, καθώς και να έχει πρόσβαση στο χώρο αποθήκευσης της συσκευής, για να μπορεί να επιλέγει και να καταθέτει αρχεία αυτού του τύπου. Χωρίς αυτό, θα ήταν πολύ πιο δύσκολο για το χρήστη να ανεβάσει αποτελέσματα εξετάσεων και εγχειρήσεών του.

Τέλος, η εφαρμογή δημιουργεί ειδοποιήσεις για το χρήστη [22]. Αυτό απαιτεί να μπορεί να αναλύει τα δεδομένα του από τη βάση, για να γνωρίζει πότε πρέπει να εκτελέσει αυτή τη λειτουργία, καθώς και το περιεχόμενο που θα πρέπει να έχει η ειδοποίηση. Απαραίτητο για αυτό είναι, ασφαλώς, να έχει την ικανότητα να στέλνει ειδοποιήσεις.

3.2.4 Μη λειτουργικές απαιτήσεις

Για τις μη λειτουργικές απαιτήσεις της εφαρμογής, η πιο βασική από όλες θα ήταν η ευκρίνεια των δεδομένων που προβάλλονται στις οθόνες της. Τα αντικείμενα στις λίστες δεδομένων πρέπει να είναι δομημένα ώστε να φαίνονται ξεκάθαρα οι σημαντικότερες πληροφορίες, για το αναφερόμενο έγγραφο, η οθόνη γραφημάτων πρέπει να διευκρινίζει το είδος μετρήσεων που προβάλλει και το έτος στο οποίο αναφέρεται, ενώ το ημερολόγιο δεν πρέπει να δίνει στα σημειωμένα ραντεβού την ίδια μορφοποίηση, χωρίς κάποιο στοιχείο διάκρισης μεταξύ κατηγοριών για το χρήστη. Οι λειτουργίες αυτές υπάρχουν για να διευκολύνουν την παρακολούθηση των ιατρικών δεδομένων του χρήστη, άρα πρέπει να είναι ευανάγνωστα από αυτόν.

Στο θέμα της διευκόλυνσης του χρήστη, οι φόρμες εισαγωγής δεδομένων πρέπει επίσης να είναι εύκολες στη χρήση τους. Ο χρήστης πρέπει να μπορεί να καταλάβει γρήγορα και χωρίς οδηγίες τον τρόπο με τον οποίο αναμένεται να τις συμπληρώσει και να καταθέσει τα δεδομένα του στη βάση. Επιπλέον, πρέπει να ενημερώνεται για την κατάσταση της απόπειράς του, δηλαδή αν τα δεδομένα καταγράφηκαν με επιτυχία ή όχι.

Πέρα από την ευκολία χρήσης όμως, υπάρχει και το πρόβλημα της ασφάλειας των δεδομένων του. Η εφαρμογή χρειάζεται να κρυπτογραφεί όλα τα δεδομένα που κατατέθηκαν από τον χρήστη, πριν τα στείλει στη βάση. Η μέθοδος κρυπτογράφησης θα πρέπει επίσης να είναι ιδιαίτερα ασφαλής, καθώς η εφαρμογή ασχολείται με την αποθήκευση και διαχείριση προσωπικών ιατρικών δεδομένων.

Τέλος, οι ενέργειες που κάνει ο χρήστης πρέπει να καταγράφονται, για ασφάλεια, σε έναν κατάλογο στη βάση δεδομένων. Ο κατάλογος αυτός θα περιέχει μόνο την ενέργεια, την ημερομηνία και την πηγή της και θα δρα ως ιστορικό για τη βάση δεδομένων.

3.2.5 Περιγραφή βασικών λειτουργιών της εφαρμογής

Η εφαρμογή αυτή είναι σχεδιασμένη για την αποθήκευση και διαχείριση ιατρικών δεδομένων από πολλαπλούς χρήστες, καθώς και για την ενημέρωσή τους για εκκρεμή γεγονότα, όταν έρχεται η κατάλληλη ημέρα και ώρα. Διακρίνει μεταξύ χρηστών χρησιμοποιώντας διαφορετικά συστήματα ταυτοποίησης στη βάση και στη συσκευή και τους επιτρέπει πρόσβαση μόνο στα δεδομένα που οι ίδιοι εισήγαγαν στη βάση.

Τα δεδομένα που εισάγονται στη βάση καλύπτουν διάφορους τομείς. Αυτοί περιλαμβάνουν παρατηρήσεις παθήσεων, προγραμματισμένα ραντεβού, περιεχόμενο αρχείων, πληροφορίες εκτάκτου ανάγκης, μετρήσεις ζωτικών σημείων και φάρμακα που λαμβάνονται. Ο κάθε χρήστης μπορεί να

εισάγει τα δεδομένα μέσω έτοιμων φορμών, οι οποίες περιλαμβάνουν υποχρεωτικά και προαιρετικά πεδία, τα οποία θα περάσουν από μία διαδικασία κρυπτογράφησης προτού καταχωρηθούν στη βάση.

Με την πρόσβασή τους στη βάση, οι χρήστες μπορούν επίσης να τροποποιήσουν ή να διαγράψουν τα δεδομένα τους σε αυτή. Σε περιπτώσεις που χρειάζεται να γίνει αλλαγή σε ένα συγκεκριμένο έγγραφο στη βάση, χρησιμοποιούνται μοναδικά αναγνωριστικά για την αναγνώρισή του, αποτρέποντας την αλλαγή άλλων άσχετων αρχείων.

Από την πλευρά της, η εφαρμογή αναλύει τα καταγεγραμμένα δεδομένα και παρέχει διαφορετικές υπηρεσίες με βάση αυτά. Μερικά από αυτά παρουσιάζονται σε γραφήματα [21], λίστες και ημερολόγια [20], ενώ για άλλα στέλνονται ειδοποιήσεις στο χρήστη [22]. Για δεδομένα που περιέχουν ένα στοιχείο χρόνου και απαιτούν ενημέρωση του χρήστη, όπως η έγκαιρη λήψη φαρμακευτικής αγωγής ή ένα ραντεβού που έχει προγραμματιστεί για εκείνη την ημέρα, η εφαρμογή δημιουργεί επιπλέον ειδοποιήσεις, τις οποίες στέλνει στο χρήστη, όταν έρθει ο κατάλληλος χρόνος [22].

Ως προς τα αρχεία των οποίων το περιεχόμενο ανεβαίνει στη βάση, ο χρήστης μπορεί στη συνέχεια να το χρησιμοποιήσει από οποιαδήποτε κινητή συσκευή έχει την εφαρμογή και να το κατεβάσει πάλι πίσω σε μορφή αρχείου [23] χρησιμοποιώντας διαφορετικές μορφολογίες [1] [2] [3]. Η διαδικασία αυτή, είναι σημαντικό να αναφερθεί, διατηρεί μόνο το κείμενο του αρχικού αρχείου, όχι το ίδιο το αρχείο.

Πέρα από αυτό, δίνεται στον κάθε χρήστη ο απαιτούμενος έλεγχος στα δεδομένα του. Αυτό περιλαμβάνει την ικανότητα να τα διαγράψει όλα, μαζί και το λογαριασμό του, κατά την οποία διαδικασία μπορεί και να μεταφέρει μερικά από αυτά στο Google Health Connect [9].

Στο θέμα αυτό, ο χρήστης μπορεί σε οποιοδήποτε σημείο να μεταφέρει δεδομένα μετρήσεων ζωτικών του σημείο από το Google Health Connect [9] στην εφαρμογή. Έτσι, οι μετρήσεις που είχε καταγεγραμμένες σε αυτό, μπορούν να προστεθούν στα δεδομένα που ήδη περιέχει η εφαρμογή και να δώσουν μία πιο πλήρη εικόνα της υγείας του στο χρόνο.

3.3 Επίλογος

Όπως φαίνεται και παραπάνω, ο βασικός σκοπός της εφαρμογής δημιουργεί διάφορες άλλες απαιτήσεις από την πλευρά της βάσης αλλά και από την πλευρά της συσκευής. Στο τοπικό τμήμα του σχεδίου ανάμεσα σε άλλα που χρειάζονται, η εφαρμογή πρέπει να μπορεί να διαβάσει τη διαθεσιμότητα μετρήσεων για βιομετρική ταυτοποίηση, να έχει πρόσβαση στο χώρο αποθήκευσης της συσκευής, καθώς και να εφαρμόζει διαδικασίες κρυπτογράφησης και αποκρυπτογράφησης για τα δεδομένα που στέλνει στη βάση και λαμβάνονται από αυτήν.

Από την πλευρά της βάσης, απαιτείται να υπάρχουν συστήματα ταυτοποίησης του χρήστη, δημιουργίας και διαγραφής λογαριασμού, μαζί και την ικανότητα να διακρίνει μεταξύ αρχείων με σιγουριά, για όποτε χρειάζεται να γίνει αλλαγή σε ένα συγκεκριμένο έγγραφο. Το πλήθος λειτουργιών, που σχετίζονται ή βασίζονται στην πρόσβαση στα δεδομένα της βάσης, σημαίνει ότι αυτή έχει και τις περισσότερες απαιτήσεις.

Πέρα από αυτό, ο ίδιος ο σκοπός της εφαρμογής, δηλαδή η διευκόλυνσή του χρήστη, ως προς τη διαχείριση των ιατρικών του δεδομένων, σημαίνει ότι προστίθενται επιπλέον απαιτήσεις ως προς την παρουσίασή της. Οι λειτουργίες της πρέπει να είναι εύκολα κατανοητές. Οι προβολές της πρέπει να είναι ευανάγνωστες και να περιέχουν ταυτόχρονα όλες τις πληροφορίες που χρειάζεται ο χρήστης. Κάθε στοιχείο της εφαρμογής πρέπει να σχεδιάζεται με στόχο να μπορεί να χρησιμοποιηθεί με ευκολία, ανεξάρτητα από την ειδικότητα ή την εμπειρία του χρήστη. Η διαδικασία σχεδιασμού της υλοποίησής της γίνεται έτσι πιο περίπλοκη, αλλά καθίσταται και πιο αποτελεσματική στο τελικό προϊόν.

Κεφάλαιο 4ο: Σχεδιασμός Συστήματος

4.1 Εισαγωγή

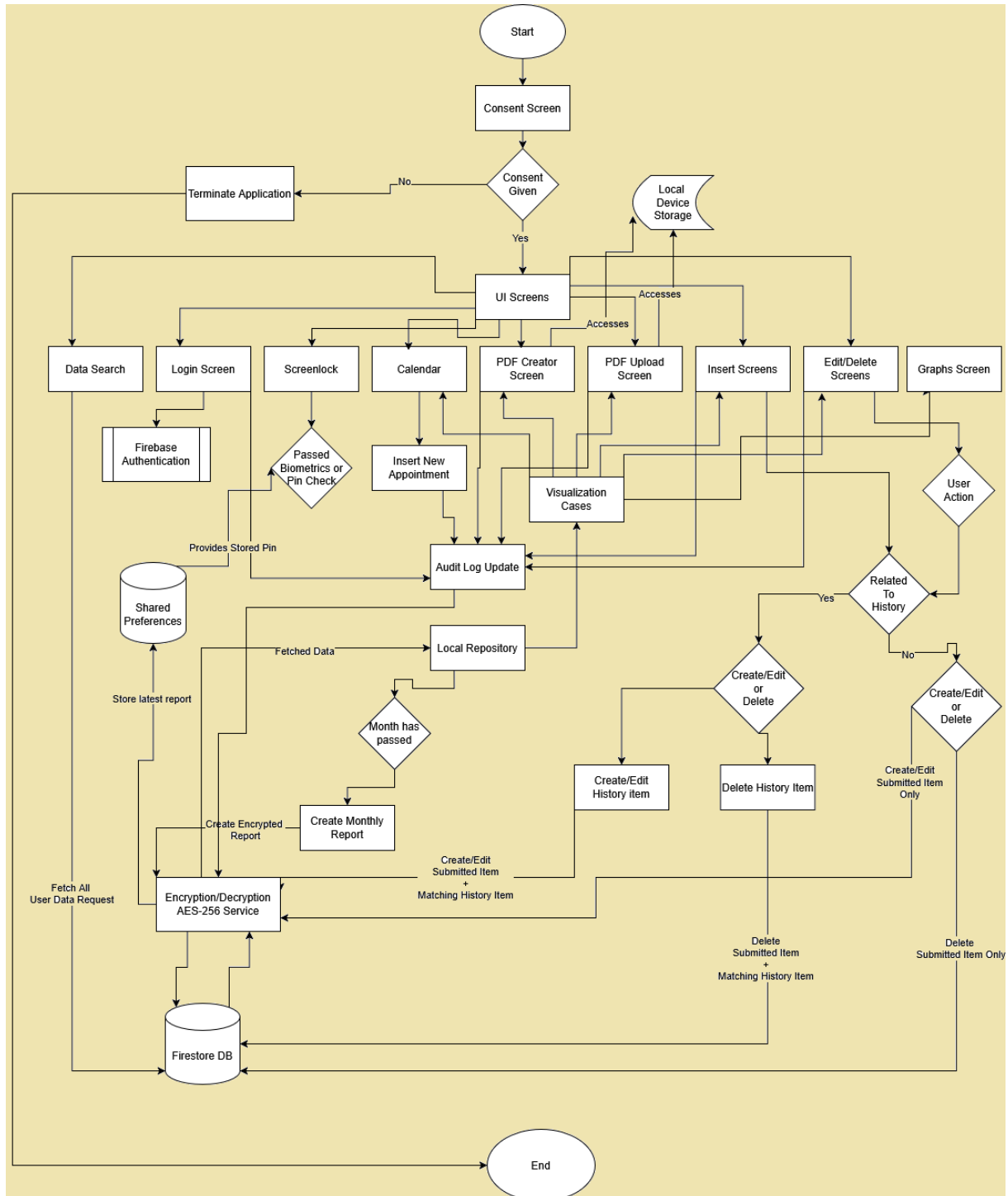
Σημαντικότερο κομμάτι της εφαρμογής είναι η κεντρική της οθόνη. Αυτή δρα ως βάση, στην οποία προβάλλονται όλες οι υπόλοιπες οθόνες, ενώ ταυτόχρονα παρακολουθεί τη σύνδεση του χρήστη, ανάμεσα σε άλλες τιμές, για να κρίνει σε ποιον κλάδο του δέντρου αποφάσεων της πρέπει να καθοδηγήσει το σύστημα και ποια αρχική οθόνη θα πρέπει να προβληθεί.

Από την κεντρική οθόνη, ο χρήστης μπορεί να μεταβεί σε διάφορες οθόνες για κάθε είδους ενέργεια, από την εισαγωγή δεδομένων ως την προβολή τους σε ένα ημερολόγιο ή γράφημα. Ανεξάρτητα όμως από την οθόνη στη οποία βρίσκεται ανά πάσα στιγμή ο χρήστης, η οθόνη που υπάρχει στη στοίβα, από την πλευρά του κώδικα, είναι η κεντρική οθόνη.

Η εφαρμογή δουλεύει σε συνεργασία με τη βάση δεδομένων, παρέχοντας τη λογική και τους κανόνες περιορισμού, ενώ η βάση παρέχει την ικανότητα αποθήκευσης των δεδομένων του χρήστη και τη δυνατότητα πρόσβασης σε αυτά τα δεδομένα από οπουδήποτε. Ταυτόχρονα, η βάση παρέχει και τις υπηρεσίες πιστοποίησης του χρήστη, βεβαιώνοντας ότι αυτός έχει δικαίωμα στα δεδομένα, στα οποία θα έχει πρόσβαση, ενώ η εφαρμογή βεβαιώνεται τοπικά ότι ο χρήστης θα έχει πρόσβαση μόνο στα έγγραφα τα οποία δημιούργησε ο ίδιος. Έτσι, ο χρήστης μπορεί να εισάγει, να τροποποιεί και να διαγράφει δεδομένα, αλλά μόνο τα δεδομένα τα οποία δημιούργησε ο ίδιος.

4.2 Συνολική αρχιτεκτονική εφαρμογής

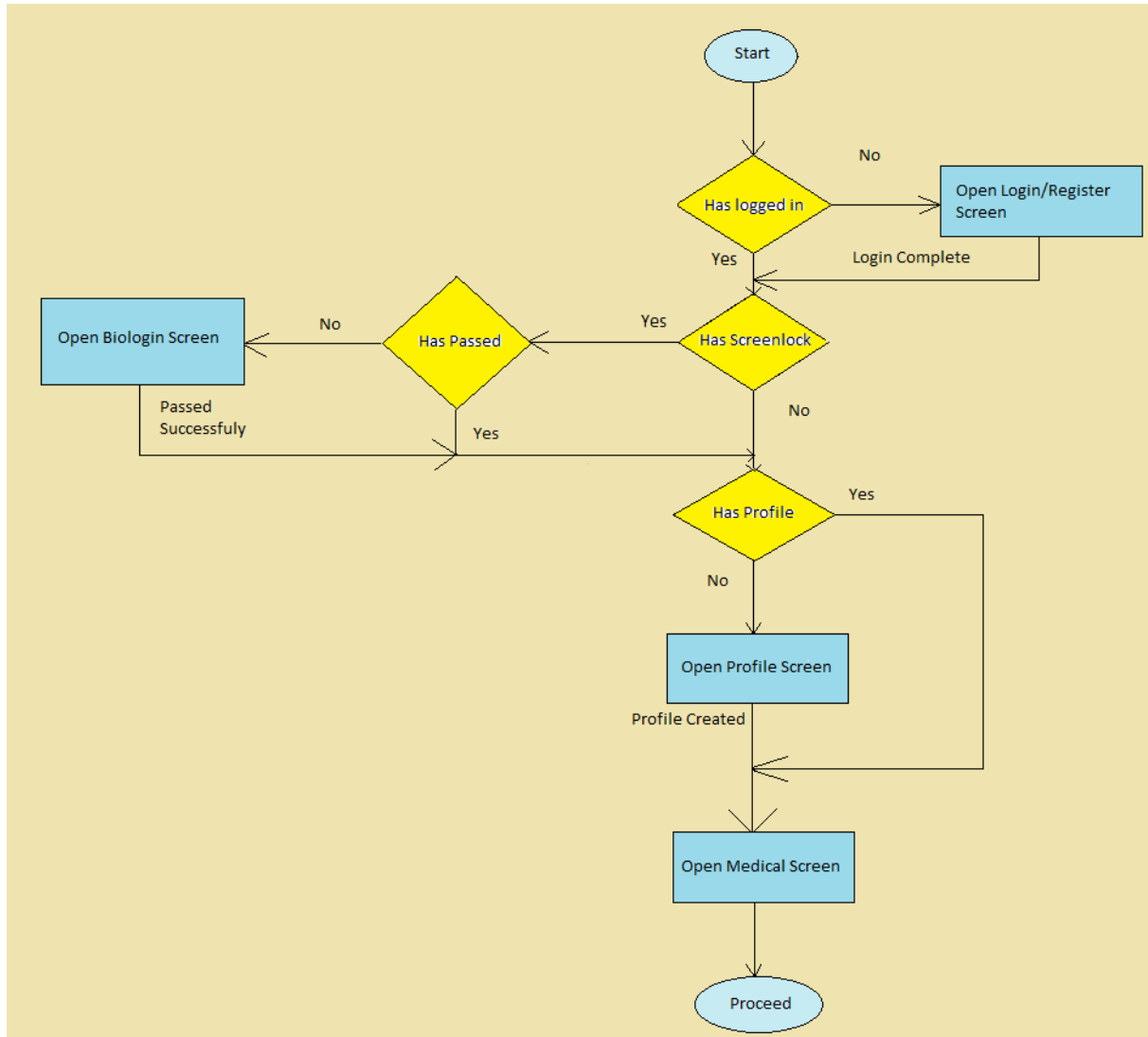
4.2.1 Κεντρικό αρχείο και είσοδος στην εφαρμογή



Σχήμα 4.1: Διάγραμμα Αρχιτεκτονικής

Το κεντρικό αρχείο της εφαρμογής παρακολουθεί διάφορες τιμές bool αποθηκευμένες στις κοινές προτιμήσεις, από τη στιγμή που ξεκινάει η εφαρμογή. Οι τιμές αυτές περιλαμβάνουν το αν ο χρήστης έχει δώσει συναίνεση για τη χρήση των δεδομένων του από την εφαρμογή, το αν ο χρήστης έχει

δημιουργήσει προφίλ και το αν ο χρήστης έχει ενεργοποιήσει το κλείδωμα της εφαρμογής με βιομετρικές ενδείξεις ή με PIN. Με βάση αυτές, κρίνεται ποια οθόνη θα πρέπει να παρουσιάζεται στον χρήστη σε οποιοδήποτε σημείο της λειτουργίας της εφαρμογής. Ταυτόχρονα, παρακολουθεί και μερικές τιμές που δεν αποθηκεύονται στις κοινές προτιμήσεις για επόμενες εκκινήσεις της εφαρμογής, όπως το αν ο χρήστης έχει περάσει από την σελίδα σύνδεσης ή αν έχει περάσει έλεγχο επιπλέον μέτρων πιστοποίησης, που μπορεί να έχει ενεργοποιήσει.



Σχήμα 4.2: Διάγραμμα ροής Εισόδου Χρήστη (Μετά την Συναίνεσή του)

Η πρώτη από τις οθόνες, που θα συναντήσει ο χρήστης, είναι η οθόνη με την φόρμα συναίνεσης. Αυτή περιέχει κείμενο, που προειδοποιεί τον χρήστη, ότι η εφαρμογή θα αποθηκεύσει και θα επεξεργάζεται δεδομένα που θα εισάγει, όπως το όνομά, το email, την ηλικία και διάφορα ιατρικά δεδομένα του. Ο χρήστης μπορεί να πατήσει ένα από δύο κουμπιά για να το αποδεχθεί και να συνεχίσει ή να αρνηθεί, οπότε η εφαρμογή θα κλείσει. Αν το κουμπί αποδοχής πατηθεί, αλλάζει μια τιμή στις κοινές προτιμήσεις [8] που δείχνει στην εφαρμογή ότι ο χρήστης έχει δώσει συναίνεση, στο οποίο σημείο η εφαρμογή θα αλλάξει σελίδα αυτόματα στη σελίδα σύνδεσης/εγγραφής. Η οθόνη της φόρμας συναίνεσης δε θα ξαναεμφανιστεί σε περίπτωση που η εφαρμογή ανοίξει, εφόσον ο χρήστης έχει δώσει συναίνεση.

Η σελίδα σύνδεσης/εγγραφής επιτρέπει στο χρήστη να συνδεθεί και να πιστοποιηθεί με το email και τον κωδικό του, ή με τον λογαριασμό Google του, χρησιμοποιώντας ένα SignInScreen widget [13]. Το widget αυτό μπορεί να αλλάζει μεταξύ εισαγωγής χρήστη και εγγραφής καινούργιου χρήστη. Μετά την εγγραφή ή τη σύνδεση, το session token και το email του χρήστη περνούν από AES-256 κρυπτογράφηση [10] και αποθηκεύονται στις κοινές προτιμήσεις της εφαρμογής [8].

Επιπλέον, κατά την επιτυχημένη εγγραφή ή είσοδο του χρήστη, εκτελείται αναζήτηση στη βάση δεδομένων του Firebase [5] για προηγούμενες εισαγωγές ιατρικών δεδομένων από αυτόν. Υπάρχοντα δεδομένα ζωτικών σημείων, παθήσεων, επείγουσες πληροφορίες, προηγούμενο ιστορικό, ραντεβού, δεδομένα φαρμάκων και αποτελεσμάτων εξετάσεων και εγχειρήσεων, ανάμεσα σε άλλα, συγκεντρώνονται και αποθηκεύονται σε τοπικά αρχεία για τη διάρκεια όπου η εφαρμογή θα δουλεύει.

Έχοντας όλα τα ραντεβού του χρήστη, η εφαρμογή επίσης θα στείλει ειδοποιήσεις για κάθε ραντεβού για εκείνη την ημέρα [22]. Η επιτυχής εκτέλεση της αναζήτησης στο Firebase [5] σημειώνεται σε μία μεταβλητή bool στην κεντρική σελίδα, ώστε να γνωρίζει η εφαρμογή, ότι ο χρήστης έχει περάσει από τη σελίδα αυτή και οι προσωρινές τοπικές λίστες δεδομένων έχουν συγχρονιστεί με τα δεδομένα του χρήστη στο Firebase [5].

Στην εκκίνησή της, η σελίδα θα ελέγξει αν η συσκευή υποστηρίζει την αναγνώριση χρήστη μέσω βιομετρικών μετρήσεων [16], καθώς και το αν υπάρχουν βιομετρικές μετρήσεις για την πιστοποίηση του χρήστη στη συσκευή [16]. Αν επιβεβαιωθεί η ύπαρξή τους, θα εμφανιστεί ένα κουτί Checkbox που, όταν πατηθεί, θα σημαίνει στην εφαρμογή ότι ο χρήστης επιθυμεί οι διάφορες λειτουργίες της να παραμένουν κλειδωμένες, αρχικά, πίσω από μια οθόνη πιστοποίησης μέσω βιομετρικών. Η ενεργοποίηση αυτής της επιλογής αποθηκεύεται στις κοινές προτιμήσεις της εφαρμογής [8] και θα αλλάζει την τιμή του κουτιού ώστε να είναι ήδη επιλεγμένο, σε περίπτωση που ο χρήστης έχει αποσυνδεθεί και καταλήγει πάλι σε αυτή την οθόνη.

Ως εναλλακτική επιλογή στην χρήση πιστοποίησης μέσω βιομετρικών μετρήσεων, ή σε συνδυασμό με αυτήν ως μια δεύτερη μέθοδο πιστοποίησης, ο χρήστης μπορεί να προσθέσει ένα PIN, μέσω μιας φόρμας εισαγωγής κειμένου, που επίσης θα σημαίνει τη μελλοντική χρήση της οθόνης πιστοποίησης [8], ακόμα και αν δεν υπάρχουν βιομετρικές μετρήσεις για την πιστοποίηση. Στην περίπτωση αυτή θα παρουσιάζεται μόνο η επιλογή για εισαγωγή μέσω PIN.

Η φόρμα εισαγωγής PIN έχει δύο κουμπιά δίπλα της, ένα για τη χρήση του PIN που εισήγαγε ο χρήστης και ένα για τη διαγραφή του PIN, το οποίο επίσης ακυρώνει τη χρήση του στην οθόνη ελέγχου, καθώς και την χρήση της ίδιας της σελίδας κλειδώματος. Το αν χρησιμοποιούνται βιομετρικές μετρήσεις ή/και PIN για τη σελίδα του κλειδώματος, καθώς και η τιμή του PIN, αποθηκεύονται στις κοινές προτιμήσεις της εφαρμογής [8]. Η τιμή του PIN αποθηκεύεται μόνο μετά την κρυπτογράφηση του με AES-256 [10] και διαγράφεται κάθε φορά που ο χρήστης καταργεί την επιλογή του στο κουτί.

Εφόσον το Checkbox κουτί επιπλέον πιστοποίησης έχει ενεργοποιηθεί από τον χρήστη σε οποιοδήποτε σημείο, η οθόνη πιστοποίησης μπορεί να εμφανιστεί σε μία από δύο περιπτώσεις. Η πρώτη περίπτωση είναι αν ο χρήστης έχει μόλις περάσει από τη σελίδα εισόδου/εγγραφής. Η δεύτερη περίπτωση είναι όταν ο χρήστης ξεκινάει την εφαρμογή, αφού έχει κάνει είσοδο σε προηγούμενη περίπτωση. Και στις δύο περιπτώσεις, στον χρήστη παρουσιάζεται ένα κουμπί, για την βιομετρική πιστοποίησή του [16], μία φόρμα κειμένου με ένα κουμπί για την κατάθεση του PIN, ή και τα δύο, ανάλογα με τις επιλογές που έκανε ο χρήστης στην προηγούμενη οθόνη. Όταν η πιστοποίηση γίνεται με επιτυχία, εκτελείται επανέκδοση του token συνεδρίας για το χρήστη.

Με την επιτυχία της εγγραφής ή της σύνδεσης του χρήστη, η εφαρμογή θα ξεκινήσει τρεις επιπλέον συναρτήσεις. Η πρώτη από αυτές θα κρυπτογραφήσει, με AES-256 [10], και θα αποθηκεύσει το token της τρέχουσας συνεδρίας στις κοινές προτιμήσεις της εφαρμογής [8]. Η δεύτερη θα ξεκινήσει την αναζήτηση για προηγούμενα καταγεγραμμένα ιατρικά δεδομένα, όπως αναφέρθηκε παραπάνω.

Τέλος, η επιτυχημένη είσοδος ή εγγραφή του χρήστη θα καταγραφεί σε ένα κατάλογο καταγραφής ελέγχου (audit log) στο Firestore [5]. Σε περίπτωση που έχει αλλάξει ο μήνας τουλάχιστον μία φορά από την τελευταία φορά που ο χρήστης μπήκε στην εφαρμογή, θα εκδοθεί και μία μηνιαία αναφορά του τελευταίου μήνα, κατά τον οποίο η εφαρμογή ήταν ενεργή. Η αναφορά θα περιλαμβάνει το πλήθος ασθενειών και ραντεβού του χρήστη για εκείνο το μήνα, τους τελευταίους στόχους υγείας που είχε θέσει, καθώς και τις τελευταίες μετρήσεις ζωτικών σημείων του μήνα.

4.2.2 Καταγραφή χρήστη στη βάση δεδομένων

Μετά την επιτυχημένη πιστοποίηση του χρήστη, είτε από την οθόνη πιστοποίησης ή απευθείας μέσω της οθόνης εισόδου/εγγραφής, η επόμενη οθόνη που θα παρουσιαστεί είναι η οθόνη δημιουργίας προφίλ. Προκειμένου να επιβεβαιωθεί ότι ο χρήστης έχει προφίλ, είτε ο χρήστης έχει μόλις φτιάξει λογαριασμό, είτε έχει είδη λογαριασμό και απλά κάνει είσοδο για πρώτη φορά στη συγκεκριμένη συσκευή, ξεκινάει μια αναζήτηση στο Firestore [5] για το προφίλ του χρήστη, αν υπάρχει, αμέσως αφού δημιουργηθεί η οθόνη.

Μέχρι η διαδικασία αυτή να ολοκληρωθεί, η οθόνη προβάλλει μία κυκλική ένδειξη φόρτωσης, μαζί με ένα κείμενο που ενημερώνει το χρήστη ότι εκτελείται μία διαδικασία για την επιβεβαίωση της ύπαρξης του προφίλ του. Αν υπάρχει προφίλ, η επιβεβαίωση αποθηκεύεται στις κοινές προτιμήσεις της εφαρμογής [8]. Σ' αυτό το σημείο η εφαρμογή δε θα έχει λόγο να παραμείνει σε αυτή την οθόνη και η ενεργή οθόνη θα αλλάξει.

Σε περίπτωση που ο χρήστης δεν έχει δημιουργήσει προφίλ, η ένδειξη φόρτωσης στην οθόνη θα αλλάξει με μια νέα μορφολογία. Το κείμενο που υπήρχε πριν θα αντικατασταθεί με ένα νέο κείμενο που θα ζητάει από το χρήστη να συμπληρώσει τη φόρμα με τα πεδία του προφίλ. Η φόρμα αυτή περιλαμβάνει ένα πεδίο κειμένου για το όνομα του χρήστη και ένα πεδίο για την ηλικία του σε χρόνια, καθώς και ένα κουμπί για την κατάθεση του περιεχομένου των πεδίων.

Τα πεδία έχουν περιορισμούς, για να βεβαιώνουν ότι η φόρμα είναι σωστά συμπληρωμένη. Για τα πεδία ονόματος, επωνύμου και φύλου, κανένα από αυτά δεν πρέπει να αφήνεται άδειο ή να έχει μόνο κενά διαστήματα. Το πεδίο ηλικίας, από την άλλη, έχει και έναν επιπλέον περιορισμό, όπου η ηλικία του χρήστη δεν πρέπει να είναι μικρότερη του μηδέν.

Με την επιτυχημένη καταχώρηση του προφίλ του χρήστη, τα στοιχεία του θα καταγραφούν στον πίνακα προφίλ στο Firebase [5]. Αυτά τα στοιχεία περιλαμβάνουν τα προαναφερόμενα πεδία, μαζί με το email και το ID του χρήστη. Επιπλέον, αποθηκεύεται στις κοινές προτιμήσεις το ότι ο χρήστης έχει πλέον προφίλ [8]. Με τον τρόπο αυτό, μελλοντικές χρήσεις της εφαρμογής δεν θα απαιτούν από τον χρήστη να ξαναφτιάχνει προφίλ.

Εφόσον το προφίλ του χρήστη είναι πλέον επιβεβαιωμένο ότι υπάρχει, η εφαρμογή θα μεταβεί στην οθόνη ιατρικών.

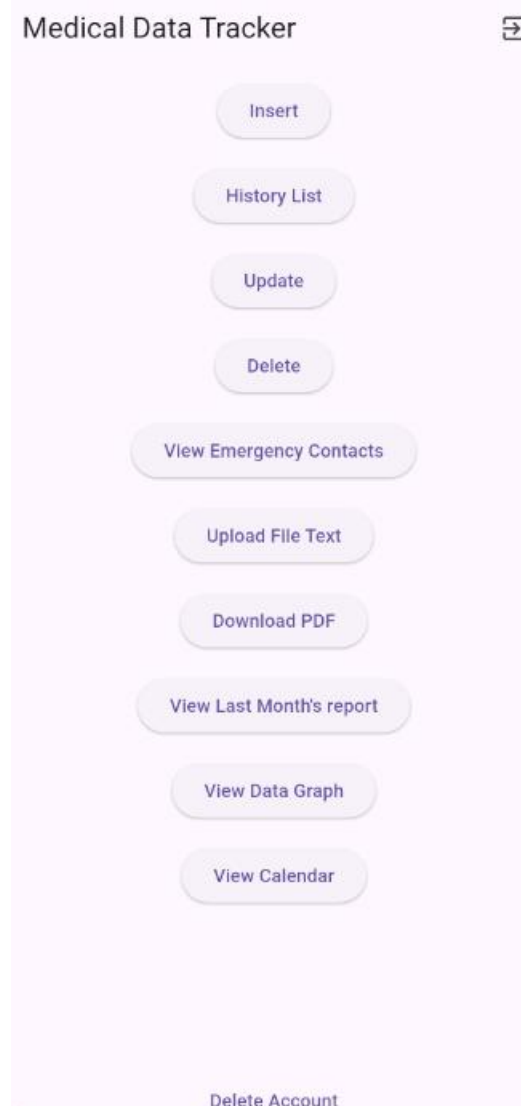
4.2.3 Κεντρική οθόνη εφαρμογής

Η οθόνη ιατρικών αποτελεί την κύρια σελίδα της εφαρμογής από την οποία ο χρήστης μπορεί να μεταβεί σε οποιαδήποτε από τις υπόλοιπες οθόνες της εφαρμογής. Προκειμένου να επιτρέπει στην κεντρική

Κεφάλαιο 4

σελίδα να παρακολουθεί αν ο χρήστης παραμένει συνδεδεμένος και να αλλάζει την οθόνη άμεσα, σε περίπτωση που η σύνδεση κοπεί για οποιοδήποτε λόγο, η οθόνη ιατρικών, όποτε είναι δυνατόν, δεν μεταβαίνει σε άλλες οθόνες απευθείας, αλλά τις προβάλλει αντικαθιστώντας τη διαμόρφωσή της με αυτήν της διαλεγμένης οθόνης.

Με σκοπό τη σωστή εκτέλεση αυτής της αλλαγής, η οθόνη ιατρικών διατηρεί μία μεταβλητή τύπου “Widget?” που ξεκινά χωρίς τιμή. Για όσο η μεταβλητή συνεχίζει να μην έχει τιμή, η οθόνη που προβάλλεται είναι η αρχική διαμόρφωση της οθόνης ιατρικών. Αυτή θα παρουσιάζει στο χρήστη ένα πλήθος κουμπιών για πλοήγηση σε άλλες οθόνες, με βάση την λειτουργία που θα απαιτείται.



Σχήμα 4.3: Screenshot Οθόνης Ιατρικών

Οι λειτουργίες αυτές περιλαμβάνουν την εισαγωγή, τροποποίηση ή διαγραφή δεδομένων, την απεικόνιση συγκεκριμένων ιατρικών δεδομένων σε μορφή γραφημάτων, την προβολή του ιστορικού του χρήστη, το ανέβασμα, κατέβασμα ή την διαγραφή περιεχομένου αρχείου PDF και την προβολή ημερολογίου ραντεβού, επαφών έκτακτης ανάγκης ή της τελευταίας μηνιαίας αναφοράς. Κατά τη μετάβαση σε μία άλλη οθόνη, η τρέχουσα οθόνη αποθηκεύεται σε μια μεταβλητή, ώστε η εφαρμογή να «θυμάται» από ποια οθόνη έφτασε στην καινούργια οθόνη.

Κάθε άλλη οθόνη, στην οποία μπορεί να μεταβεί ο χρήστης από την οθόνη ιατρικών, κληρονομεί από μία συνάρτηση επιστροφής στην αρχική οθόνη. Η συνάρτηση αυτή αλλάζει την τιμή της μεταβλητής ενεργής οθόνης πίσω σε κενή τιμή, επιστρέφοντας την οθόνη που προβάλλεται πίσω στην αρχική διαμόρφωση της οθόνης ιατρικών. Σε περίπτωση που μια οθόνη μπορεί να μεταβεί μετά σε άλλες οθόνες, θα κληρονομηθούν δύο επιπλέον συναρτήσεις από την οθόνη ιατρικών και θα περαστούν στις επόμενες οθόνες, αν ο χρήστης εκτελέσει μια ενέργεια που θα τον μεταφέρει σε μια από αυτές.

Η πρώτη συνάρτηση είναι μια συνάρτηση μετάβασης, υπεύθυνη για την αλλαγή της μεταβλητής ενεργής οθόνης και την αποθήκευση της τρέχουσας οθόνης στη μεταβλητή προηγούμενης οθόνης. Η δεύτερη επιστρέφει το χρήστη στην προηγούμενη οθόνη, αλλάζοντας την ενεργή οθόνη πίσω στην τιμή της μεταβλητής προηγούμενης οθόνης. Καθώς η εφαρμογή δεν έχει σειρές μεταβάσεων μεγαλύτερου βάθους, η μεταβλητή δεν χρειάζεται να είναι λίστα.

Κατά την εκκίνηση της οθόνης ιατρικών, ελέγχεται αν ο χρήστης έχει περάσει από την οθόνη εισόδου/εγγραφής, με σκοπό την επιβεβαίωση ότι οι προσωρινές τοπικές λίστες δεδομένων έχουν συγχρονιστεί με την απομακρυσμένη βάση στο Firebase [5]. Αν βρεθεί ότι δεν έχει ολοκληρωθεί μετάβαση από εκείνη την οθόνη, εκτελείται η διαδικασία αναζήτησης και προσωρινής αποθήκευσης των απομακρυσμένων δεδομένων του χρήστη σε τοπικά αρχεία, καθώς μπορεί να χρειαστούν τα δεδομένα σε κάποια από τις διάφορες διαθέσιμες οθόνες.

Ανεξάρτητα από το αποτέλεσμα του παραπάνω ελέγχου, εκτελείται επίσης μια ακόμα συνάρτηση, για την εξακρίβωση του αν ο χρήστης εισήρθε με χρήση πιστοποίησης email και κωδικού ή απλά με την χρήση του λογαριασμού του Google του [12]. Με βάση το αποτέλεσμα, μπορεί να εμφανιστεί και ένα επιπλέον κουμπί στην οθόνη, για την εισαγωγή αποθηκευμένων δεδομένων υγείας του από το Google Health στην εφαρμογή και τη βάση δεδομένων στο Firebase [5] [9]. Τα δεδομένα που αποθηκεύει η εφαρμογή αφορούν μετρήσεις για τον παλμό, την διαστολική και συστολική πίεση, τη θερμοκρασία σώματος και το σάκχαρο του χρήστη.

Η συλλογή αυτών των δεδομένων γίνεται με χρήση του πακέτου “health” του Flutter [9]. Επειδή οι τιμές, που λαμβάνονται με αυτή τη μέθοδο, είναι της μορφής τιμών τύπου HealthValue [9], χρειάζεται πρώτα να μεταφερθούν σε ένα String, για να μπορούν να αποθηκευτούν στο Firebase [5]. Από εκεί, θα περάσουν από μία συνάρτηση επαλήθευσης, που θα επιβεβαιώσει ότι οι τιμές αυτές δεν υπάρχουν ήδη στη βάση. Αφού τα δεδομένα συγκριθούν με τις υπάρχοντες τιμές του Firebase [5], οι τιμές που δεν υπάρχουν ήδη θα αποθηκευτούν σε αυτό και θα προστεθούν ταυτόχρονα στο τοπικό αρχείο της εφαρμογής, ανάμεσα στα υπόλοιπα δεδομένα που είχαν καταγραφεί, κατά την είσοδο του χρήστη στην εφαρμογή.

Πέρα από τα κουμπιά για τη μετάβαση σε άλλες οθόνες και το κουμπί συγχρονισμού με το Google Health, που έχουν ήδη αναφερθεί, η οθόνη ιατρικών περιέχει ένα τελευταίο κουμπί, το κουμπί διαγραφής λογαριασμού. Τοποθετημένο στο κάτω μέρος της οθόνης, το κουμπί αυτό, όταν πατηθεί, παρουσιάζει το χρήστη ένα παράθυρο επιβεβαίωσης πρόθεσης. Το παράθυρο αυτό περιέχει ένα κείμενο, το οποίο ζητά από το χρήστη να επιβεβαιώσει το ότι όντως επιθυμεί να διαγράψει το λογαριασμό του, με δύο κουμπιά από κάτω του, για να επιβεβαιώσει τη διαδικασία, ή την ακυρώσει και να κλείσει το παράθυρο.

Το παράθυρο διαγραφής λογαριασμού περιέχει, επίσης, ένα κουτί Checkbox που, όταν πατηθεί, αλλάζει μια μεταβλητή bool, με βάση την τρέχουσα κατάσταση του κουτιού, από αληθή σε ψευδή όταν απενεργοποιείται και το αντίστροφο όταν ενεργοποιείται. Η τιμή αυτής της μεταβλητής αποτελεί τη μέθοδο με την οποία αναγνωρίζεται αν ο χρήστης επιθυμεί να μεταφέρει τα δεδομένα του από το

Firebase [5] πίσω στο Google Health [9]. Το κουτί αυτό εμφανίζεται μόνο εάν ο χρήστης είναι συνδεδεμένος με λογαριασμό Google και η τιμή του αντιστοιχεί στην τιμή της μεταβλητής.

Εάν επιβεβαιωθεί η διαγραφή του λογαριασμού, η συνάρτηση διαγραφής λογαριασμού περιμένει όσο εκτελούνται πολλαπλές αναζητήσεις στο Firebase [5] σε κάθε συλλογή που ο χρήστης θα μπορούσε να έχει δεδομένα. Πέρα από τους πίνακες, στους οποίους μπορεί ο χρήστης να ανεβάσει δεδομένα από τις διάφορες οθόνες εισαγωγής, αναζητά επίσης και έγγραφα, με την ταυτότητα του χρήστη, στους πίνακες με περιεχόμενα αρχείων PDF, στους πίνακες με τις ημερομηνίες τελευταίων αναφορών, καθώς και στον πίνακα με τη λίστα προφίλ χρηστών. Εκτελείται επίσης και αναζήτηση για έγγραφα στον πίνακα με τον κατάλογο ενεργειών χρήστη, που αναφέρουν το email του χρήστη ως την πηγή από την οποία προήλθαν, ανεξάρτητα από την ενέργεια που έχει καταγραφεί.

Με την ολοκλήρωση της κάθε αναζήτησης, το αποτέλεσμα χρησιμοποιείται για να διαγραφούν όλα τα σχετικά έγγραφα στην αντίστοιχη συλλογή. Η μόνη εξαίρεση σε αυτό είναι η αναζήτηση στον πίνακα ζωτικών σημείων. Προτού προχωρήσει στη διαγραφή των κατάλληλων εγγράφων σε αυτό τον πίνακα, η συνάρτηση ελέγχει πρώτα την τιμή της μεταβλητής, που ελέγχει αν ο χρήστης θέλει να μεταφερθούν τα σχετικά δεδομένα ζωτικών σημείων στο Google Health [9] και, αν είναι αληθής, περιμένει μέχρι την ολοκλήρωση μιας άλλης συνάρτησης, που εκτελείται για αυτό το σκοπό.

Αφού έχει λάβει επιβεβαίωση ότι έχει εκτελεστεί η διαγραφή όλων των σχετικών εγγράφων σε κάθε πιθανό πίνακα, η συνάρτηση διαγραφής λογαριασμού προχωρά στο να διαγράψει τον χρήστη από τον πίνακα πιστοποίησης του Firebase [12].

Η εκτέλεση της διαδικασίας μεταφοράς δεδομένων ζωτικών σημείων του χρήστη στο Google Health, κατά τη διαγραφή του χρήστη από την εφαρμογή, γίνεται πολλαπλές φορές, για κάθε έγγραφο που προκύπτει από την αναζήτηση στον πίνακα του Firestore [5]. Αφού εξακριβωθεί το είδος της μέτρησης, από τα διάφορα είδη δεδομένων ζωτικών σημείων, που συλλέγει η εφαρμογή, χρησιμοποιείται το πακέτο «health» [9] και οι συναρτήσεις εγγραφής του, για να περαστούν οι τιμές των μετρήσεων στο Google Health, αφού πρώτα ζητηθεί άδεια, για την πρόσβαση στους τύπους δεδομένων του πακέτου [9].

Επαναλαμβάνοντας τη διαδικασία αυτή, οι καταγεγραμμένες μετρήσεις μεταφέρονται στο Google Health με τις σωστές τιμές και ημερομηνίες μέτρησης. Αν δεν υπάρχει κατάλληλη ημερομηνία για κάποιο λόγο, χρησιμοποιείται η τρέχουσα ημερομηνία.

4.2.4 Οθόνη ιστορικού

Σχετικά με τις διάφορες άλλες οθόνες της εφαρμογής, η πρώτη από αυτές που θα αναφερθεί εδώ είναι η οθόνη ιστορικού του χρήστη. Ως μία από τις πολλές οθόνες στις οποίες μπορεί να καταλήξει ο χρήστης κατευθείαν από την οθόνη ιατρικών, περιέχει ένα στοιχείο κοινό σε όλες τις οθόνες αυτής της κατηγορίας, ένα κουμπί επιστροφής, που αλλάζει την τιμή της ενεργής οθόνης σε κενή τιμή. Το περιεχόμενο της οθόνης ιστορικού αλλάζει με βάση το περιεχόμενο της λίστας αντικειμένων τύπου Ιστορίας, στο τοπικό αρχείο ιστορικού χρήστη στην εφαρμογή. Όταν η λίστα είναι άδεια, η οθόνη περιέχει απλά ένα κείμενο που ενημερώνει το χρήστη για το γεγονός αυτό.

Από την άλλη, σε περίπτωση που η λίστα δεν είναι άδεια, το περιεχόμενο αλλάζει, ώστε να παρουσιάσει μια λίστα καρτών, με κάθε αντικείμενο της λίστας ιστορικού να έχει τη δική του κάρτα. Ανεξάρτητα από τον τύπο του αντικειμένου που παρατηρείται, μεταξύ πάθησης, ραντεβού για εγχείρηση, ραντεβού για εξέταση, για τα αποτελέσματα αυτών, για ραντεβού κάποιου άλλου τύπου, ή για φάρμακα που παίρνει ο χρήστης, η κάρτα παρουσιάζει τον τύπο του αντικειμένου και την ημερομηνία που

καταγράφηκε στο ιστορικό. Τα αντικείμενα τύπου πάθησης παρουσιάζουν επίσης το όνομα της πάθησης. Τα ραντεβού για εγχειρήσεις παρουσιάζονται και με το όνομα της τοποθεσίας, όπου έχει κλειστεί το ραντεβού, καθώς και με την ώρα του ραντεβού.

Αυτά τα παραπάνω στοιχεία, μαζί με το είδος της εξέτασης, παρουσιάζονται για τα ραντεβού εξετάσεων. Αν το αντικείμενο είναι αποτέλεσμα εξέτασης ή εγχείρησης, μιας και το περιεχόμενο αυτών των αντικειμένων είναι ολόκληρο το κείμενο αρχείων pdf, στην κάρτα χρησιμοποιείται μόνο ο τύπος και η ημερομηνία καταγραφής του αντικειμένου, μαζί με τον τίτλο του εγγράφου αποτελεσμάτων, που ανέβασε ο χρήστης. Τα ραντεβού εξέτασης περιέχουν επίσης τα ονόματα της τοποθεσίας του ραντεβού, την ημερομηνία και την ώρα, καθώς και τον τύπο του ραντεβού, ακριβώς όπως οι άλλες κατηγορίες ραντεβού, αν και αφήνουν το όνομα του τύπου ραντεβού να εξαρτάται πλήρως από το χρήστη, αντί να παρέχουν περιορισμένες επιλογές από τις οποίες θα πρέπει να επιλέξει.

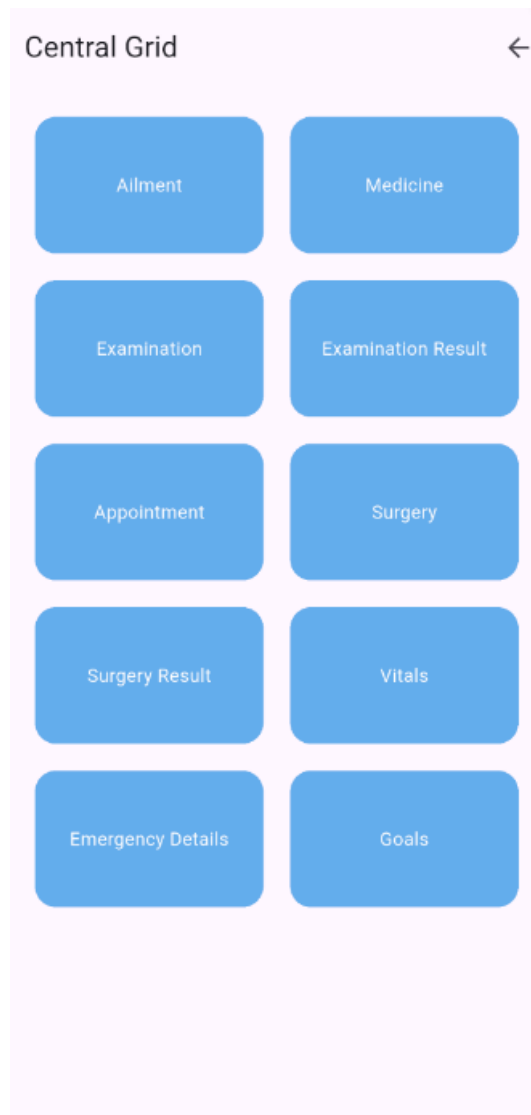
Κάθε κάρτα στη λίστα που προβάλλεται μπορεί να συρθεί στο πλάι και να διαγραφεί. Αφού η διαδικασία αυτή ολοκληρωθεί, η κάρτα δε θα εμφανίζεται στην προβαλλόμενη λίστα και το αντίστοιχο αντικείμενο θα διαγραφεί από το τοπικό αρχείο ιστορικού. Από εκεί, εμφανίζεται ένα πλαίσιο ενημέρωσης για διάρκεια τριών δευτερολέπτων, που επιβεβαιώνει την επιτυχία της διαγραφής του αντικειμένου. Τμήμα του πλαισίου αυτού περιέχει ένα κουμπί ακύρωσης, ώστε το αντικείμενο να ξαναμπει στη θέση του στη λίστα όπου ήταν πριν, επαναφέροντάς το στην προβαλλόμενη λίστα και στο τοπικό αρχείο ιστορικού χρήστη.

Η διαδικασία που γίνεται εμφανής στο χρήστη όμως είναι μόνο η αρχή. Από τη στιγμή που λήξει το χρονικό περιθώριο για την ακύρωση της διαγραφής και το πλαίσιο ενημέρωσης κλείσει, η έλλειψη ενέργειας για ακύρωση λαμβάνεται ως επιβεβαίωση της οριστικοποίησης της διαγραφής και οδηγεί στην μετάβαση στο επόμενο στάδιο. Συγκεκριμένα, ξεκινούν δύο ακόμα συναρτήσεις. Η πρώτη από αυτές ψάχνει στα τοπικά αρχεία για τη λίστα, που αποθηκεύει το σχετικό τύπο αντικειμένων και ξεκινά μια αναζήτηση για το πρώτο αντικείμενο του οποίου τα πεδία αντιστοιχούν σε αυτά του αντικειμένου ιστορικού που μόλις διαγράφηκε. Η δεύτερη ξεκινάει μια παρόμοια αναζήτηση και διαγραφή στην απομακρυσμένη βάση δεδομένων του Firebase [5].

4.2.5 Ενδιάμεση οθόνη πλοήγησης

Για την πλοήγηση στις σωστές οθόνες εισαγωγής, τροποποίησης ή διαγραφής αντικειμένου οποιουδήποτε πεδίου, η οθόνη ιατρικών χρησιμοποιεί ένα σύνολο χαρτών που αντιστοιχούν τιμές String με τα αντίστοιχα Widget για τις διάφορες συγκεκριμένες οθόνες. Ανάλογα με τη λειτουργία που επιθυμεί να εκτελέσει ο χρήστης, μεταξύ των τριών που αναφέρθηκαν παραπάνω, ο αντίστοιχος χάρτης θα περαστεί, ενώ καλείται η κεντρική οθόνη πλέγματος. Πέρα από το χάρτη, περνιούνται επίσης και η συνάρτηση επιστροφής στην οθόνη ιατρικών, η συνάρτηση της επιστροφής στην προηγούμενη οθόνη, μία λίστα String με ονόματα-κλειδιά για κάθε τύπο οθόνης που μοιράζονται οι διαφορετικοί διαθέσιμοι χάρτες, καθώς και μία μεταβλητή-δείκτης, για να μπορεί η νέα οθόνη να αναγνωρίσει το είδος λειτουργιών, που περιμένει ο χρήστης.

Η οθόνη κεντρικού πλέγματος λαμβάνει το χάρτη με τις τιμές Widget και δημιουργεί ένα πλέγμα από αντικείμενα τα οποία, όταν πατηθούν από το χρήστη, τον μεταφέρουν στην αντίστοιχη οθόνη εισαγωγής, τροποποίησης, ή διαγραφής. Το πλήθος αντικειμένων του πλέγματος είναι ίσο με το πλήθος αντικειμένων στο χάρτη, που παρέχεται στην οθόνη, όταν αυτή καλείται και το κάθε ένα από αυτά είναι ένα ειδικά σχεδιασμένο Widget, που μπορεί να πατηθεί και να εκτελέσει μία προκαθορισμένη διαδικασία. Η διαδικασία αυτή πλοηγεί το χρήστη, με βάση το αντικείμενο του χάρτη που κληρονομεί από την οθόνη κεντρικού πλέγματος.



Σχήμα 4.4: Screenshot Οθόνης Κεντρικού Πλέγματος Για Εισαγωγή Αντικειμένου

Το Widget για τα αντικείμενα του πλέγματος αποτελεί ένα ευέλικτο κομμάτι κώδικα, που επιτρέπει στα αντικείμενα που δημιουργεί να πλοηγούν το χρήστη σε οθόνες εισαγωγής, τροποποίησης ή διαγραφής ανάλογα με το τι επιθυμεί. Για να μπορεί να προσθέσει το σύστημα τα αντικείμενα για την πλοήγηση στα Widget του τύπου που επιθυμεί ο χρήστης και μόνο σε αυτά, χρειάζεται να μπορεί να ξεχωρίσει μεταξύ του αν ο χρήστης ζητάει τα Widget εισαγωγής, τροποποίησης ή διαγραφής. Με αυτό το σκεπτικό, κληρονομεί την μεταβλητή-δείκτη από την οθόνη κεντρικού πλέγματος. Επιπλέον, κληρονομεί ένα String από την λίστα ονομάτων-κλειδιών για την οθόνη, στην οποία πρέπει το αντικείμενο να πλοηγεί το χρήστη, το χάρτη Widget που πήρε η οθόνη κεντρικού πλέγματος, καθώς και τις συναρτήσεις για επιστροφή στην προηγούμενη οθόνη και επιστροφή στην οθόνη ιατρικών.

Λόγω ενός περιορισμού στον ορισμό Widget ως τιμή σε ένα χάρτη, οι τιμές του χάρτη δεν μπορούν να δηλωθούν με συναρτήσεις, επομένως οι δηλώσεις τους στον χάρτη δεν έχουν δηλωμένες τις συναρτήσεις. Για να ξεπεραστεί αυτό το εμπόδιο, το αντικείμενο του πλέγματος, όταν ο χρήστης πατήσει πάνω του, δεν χρησιμοποιεί την τιμή του χάρτη απευθείας ως στόχο πλοήγησης για την ειδική συνάρτηση μετάβασης.

Η πρώτη ενέργεια που γίνεται είναι η ανάλυση του ονόματος-κλειδιού και της μεταβλητής-δείκτη που παρέχεται από το χάρτη, για να καθοριστεί το σωστό Widget, στο οποίο θα πρέπει να μεταβεί η εφαρμογή. Σε περίπτωση που δεν είναι δυνατόν να αντιστοιχηθεί ο συνδυασμός ονόματος-κλειδιού και μεταβλητής-δείκτη με μία οθόνη, η αντιστοιχη τιμή Widget του χάρτη θα χρησιμοποιηθεί αναγκαστικά ως η τελευταία αποδεκτή επιλογή που απομένει, αν και δεν θα έχει τις συναρτήσεις που χρειάζεται και η οθόνη στην οποία θα μπορεί να μεταβεί ο χρήστης θα πρέπει να βασιστεί στην δική της έκδοση μιας συνάρτησης επιστροφής. Εάν συμβεί αυτό, δε θα χρησιμοποιηθεί η προετοιμασμένη συνάρτηση αλλαγής της μεταβλητής ενεργής οθόνης που έχει κληρονομήσει, αλλά μία εντολή για απευθείας μετάβαση στην καινούργια οθόνη.

Η χρήση αυτής της εντολής αποφεύγεται όσο γίνεται και χρησιμοποιείται μόνο αν δεν υπάρχει άλλη επιλογή, καθώς σημαίνει ότι η οθόνη που είναι ορατή στον χρήστη δε θα αλλάξει, σε περίπτωση που η σύνδεσή του διακοπεί για κάποιο λόγο. Αυτό μπορεί να οδηγήσει σε περιπτώσεις όπου ο χρήστης προσπαθεί να επικοινωνήσει με τη βάση δεδομένων στο Firebase [5] και δεν καταλαβαίνει γιατί αποτυγχάνει χωρίς λόγο, καθώς η αποσύνδεσή του δε θα γίνει προφανής μέχρι να κάνει επανεκκίνηση της εφαρμογής ή προσπαθήσει να επιστρέψει στην προηγούμενη οθόνη ή πίσω στην οθόνη ιατρικών και βρεθεί στη σελίδα εγγραφής/εισόδου.

4.2.6 Δομή οθονών εισαγωγής

Πρώτες από τις κατηγορίες οθονών, που μπορεί να παρουσιάσει ως επιλογές η οθόνη κεντρικού πλέγματος, είναι η κατηγορία εισαγωγής, δηλαδή όλες οι οθόνες που είναι σχεδιασμένες για τη εισαγωγή δεδομένων. Μοιράζονται ένα κοινό σχεδιασμό, όπου η κάθε οθόνη περιέχει πεδία εισαγωγής κειμένου, για κάθε πεδίο που μπορεί να περιέχει το είδος αντικειμένου στο οποίο αναφέρεται η οθόνη αυτή. Κάθε πεδίο συνοδεύεται από ένα κουμπί από δίπλα του το οποίο αποθηκεύει το περιεχόμενό του σε μία αντίστοιχη μεταβλητή.

Σε περίπτωση που ο χρήστης αναμένεται να προσθέσει παραπάνω από μία εισαγωγή για κάποιο πεδίο, όπως στην περίπτωση που θέλει να ανεβάσει τις αλλεργίες του, η κάθε εισαγωγή πρέπει να προστεθεί μία-μία σε μία μεταβλητή λίστας, με την ίδια μέθοδο όπως αν ήταν μόνο για μία εισαγωγή. Το περιεχόμενο του πεδίου εισαγωγής κειμένου αδειάζει κάθε φορά που το κουμπί προσθήκης εισαγωγής ενεργοποιείται.

Αν το αντικείμενο απαιτεί την επιλογή κάποιας ημερομηνίας ή ώρας της ημέρας, όπως για το κλείσιμο ραντεβού, η οθόνη θα περιέχει επίσης ένα πεδίο κειμένου και ένα κουμπί εικόνας με την εμφάνιση ενός ημερολογίου ή ενός ρολογιού, ανάλογα με το είδος της τιμής. Για όσο ο χρήστης δεν έχει επιλέξει κάποια ημερομηνία ή ώρα, το πεδίο κειμένου έχει μία κανονική τιμή, που τον ενημερώνει για αυτό. Με την επιλογή μιας τιμής μέσω του κουμπιού, το πεδίο θα αλλάξει στο να παρουσιάζει την επιλεγμένη τιμή. Τα κουμπιά δουλεύουν εμφανίζοντας ένα προσωρινό παράθυρο, για την επιλογή ημερομηνίας ή ώρας, μια λειτουργία που περιέχει το βασικό πακέτο του Flutter.

Οι περισσότερες από τις οθόνες εισαγωγής περιέχουν ένα κουμπί καταχώρησης αντικειμένου. Με το πάτημα αυτού του κουμπιού ξεκινάει η συνάρτηση για την επιβεβαίωση της ορθής συμπλήρωσης των πεδίων και, με την επιβεβαίωση αυτής, την κατοχύρωση του καινούργιου αντικειμένου στην απομακρυσμένη βάση δεδομένων του Firebase [5], μετά από κρυπτογράφηση των δεδομένων που εισήγαγε ο χρήστης, καθώς και την ημερομηνία δημιουργίας αυτού του αντικειμένου.

Εφόσον η κατοχύρωση ολοκληρωθεί με επιτυχία, το αντικείμενο προστίθεται και στην τοπική προσωρινή μνήμη της εφαρμογής στην αρμόδια λίστα, ενώ η προσθήκη του αντικειμένου καταγράφεται στο ιστορικό του χρήστη, που περιλαμβάνει τις εκδόσεις στο Firebase [5] και στην τοπική μνήμη, καθώς

και στον κατάλογο ενεργειών χρήστη. Στη συνέχεια, θα εμφανιστεί μία μπάρα ενημέρωσης, για να πληροφορήσει το χρήστη για την επιτυχία της εισαγωγής του αντικειμένου.

Αν όμως η κατοχύρωση αποτύχει για κάποιο λόγο και προκύψει σφάλμα, η συνάρτηση θα προχωρήσει σε μία άλλη σειρά ενεργειών. Συγκεκριμένα, αντί για τη ενημέρωση της τοπικής λίστας του αντικειμένου και του ιστορικού, ή την προσθήκη καινούργιας εισαγωγής στον κατάλογο ενεργειών του χρήστη στο Firebase [5], η συνάρτηση απλά εμφανίζει μια μπάρα ενημέρωσης, που πληροφορεί ότι η κατοχύρωση απέτυχε.

Ανεξάρτητα από την επιτυχία ή αποτυχία της κατοχύρωσης, το περιεχόμενο των πεδίων εισαγωγής κειμένου αδειάζει. Επιπλέον, η μεταβλητής ημερομηνίας ή/και ώρας, αν υπάρχουν τέτοια πεδία στην οθόνη, έχουν τις τιμές τους να επιστρέφουν στο να είναι απλές κενές τιμές και τα αντίστοιχα πεδία κειμένου, που παρουσιάζουν τις τιμές που επέλεξε ο χρήστης, επιστρέφουν στο να προβάλλουν τα κανονικά τους μηνύματα, αντί για την επιλεγμένη τιμή.

Πέρα από αυτά, εφόσον η συνάρτηση επιστροφής στην οθόνη ιατρικών έχει κληρονομηθεί με επιτυχία, οι οθόνες εισαγωγής που την κληρονομούν εμφανίζουν και ένα ακόμα κουμπί, το κουμπί επιστροφής στην αρχική οθόνη. Αυτό το κουμπί αποτελεί μία πιο γρήγορη μέθοδο επιστροφής, προκειμένου να μην αναγκάζεται ο χρήστης να περάσει πρώτα από την οθόνη κεντρικού πλέγματος. Η συνάρτηση αυτή και το κουμπί που την τρέχει υπάρχουν στις περισσότερες οθόνες εισαγωγής, με μερικές εξαιρέσεις όπου δεν εφαρμόστηκε για να μην γεμίζει υπερβολικά η οθόνη.

4.2.7 Οθόνες εισαγωγής

Πρώτη από τις οθόνες εισαγωγής είναι η οθόνη εισαγωγής πάθησης. Τα μόνα πεδία που χρειάζεται και μπορεί να συμπληρώσει ο χρήστης σε αυτή την οθόνη, για να εισάγει ένα νέο αντικείμενο τύπου «Πάθησης» είναι το όνομα της ασθένειας και η ημερομηνία, στην οποία αυτή παρατηρήθηκε. Σε περίπτωση που γίνει απόπειρα να εισαχθεί νέο αντικείμενο χωρίς ένα από αυτά τα πεδία να είναι συμπληρωμένα, ή αν ο χρήστης προσπαθήσει απλά να αφήσει κενό διάστημα για το όνομα της ασθένειας, η διαδικασία καταχώρησης θα επιστρέψει αμέσως, χωρίς να προχωρήσει στην απόπειρα να περάσει το αντικείμενο στο Firestore [5].

Επόμενη είναι η οθόνη εισαγωγής πληροφοριών εκτάκτου ανάγκης. Σε αυτή την οθόνη ο χρήστης μπορεί να προσθέσει ιατρικά δεδομένα όπως την ομάδα αίματός του, επαφές σε περίπτωση έκτακτης ανάγκης, πιθανές αλλεργίες που έχει, καθώς και φάρμακα, που μπορεί να χρειάζεται. Περιέχει ένα πεδίο εισαγωγής κειμένου για κάθε μία από αυτές τις κατηγορίες, με εξαίρεση τις επαφές, όπου έχει συνολικά δύο πεδία, ένα για το όνομα της επαφής και ένα για τον αριθμό τηλεφώνου της. Μοναδικός περιορισμός στην εισαγωγή δεδομένων σε αυτή την οθόνη είναι ότι τουλάχιστον μία από τις μεταβλητές, που αποθηκεύουν τις τιμές των πεδίων δεν πρέπει να είναι κενή ή, αν η μεταβλητή είναι μεταβλητή λίστας, άδεια.

Για να μπορεί να βλέπει ο χρήστης τα δεδομένα που εισάγει, σε κατηγορίες που προβλέπεται να θέλει να προσθέσει πάνω από ένα αντικείμενο, όπως τις επαφές, τα φάρμακα και τις αλλεργίες, η οθόνη έχει επίσης και τρία διαφορετικά πλαίσια που παρουσιάζουν λίστες από τα δεδομένα, με το κάθε πλαίσιο να αντιστοιχεί σε διαφορετική κατηγορία και τα οποία, όταν είναι άδεια, παρουσιάζουν κείμενο που πληροφορεί το χρήστη ποια είναι η κατηγορία που θα παρακολουθούν. Κάθε πλαίσιο έχει και ένα κουμπί μέσα του, που διαγράφει την πιο πρόσφατη από τις εισαγωγές του χρήστη από την αντίστοιχη μεταβλητή λίστας και από το κείμενο που παρουσιάζει το πλαίσιο. Η μέθοδος αυτή είναι μόνο για την διαγραφή δεδομένων στη μεταβλητή, προτού ο χρήστης κατοχυρώσει τα δεδομένα στο Firebase [5], όχι για τη διαγραφή δεδομένων από αυτό.

Κάθε χρήστης μπορεί να έχει μόνο ένα αντικείμενο τύπου «Πληροφορίες εκτάκτου ανάγκης» στον πίνακα του Firebase [5]. Για να βεβαιωθεί αυτό, κατά τη διαδικασία εισαγωγής δεδομένων, πραγματοποιείται πρώτα έλεγχος, για να εξακριβωθεί αν υπάρχει ήδη αντικείμενο από το χρήστη στον πίνακα. Εφόσον η ύπαρξη αντικειμένου επιβεβαιωθεί, συγκρίνονται οι τιμές των πεδίων στο αντικείμενο με τις τιμές των αντίστοιχων πεδίων από τα δεδομένα, που προσπαθεί να εισάγει ο χρήστης. Οι τιμές που διαφέρουν από τις προηγούμενες εισάγονται στην θέση των παλιών.

Ο λόγος για αυτόν τον περιορισμό είναι ότι οι πληροφορίες, που εισάγει ο χρήστης σε αυτόν τον πίνακα, χρησιμοποιούνται κατά την εκκίνηση της εφαρμογής, για να σταλθεί μια ειδοποίηση στη συσκευή του με τις πληροφορίες εκτάκτου ανάγκης του. Καθώς ο πίνακας αυτός είναι σχεδιασμένος για την αποθήκευση και παρουσίαση σημαντικών ιατρικών πληροφοριών, που θα συνεχίσουν να ισχύουν για αρκετό καιρό ή και για πάντα, όπως η ομάδα αίματος του χρήστη, δεν προβλέπεται να χρειάζονται πολλές διαφορετικές εισαγωγές στον πίνακα, οπότε ο περιορισμός θεωρείται ότι δεν παρουσιάζει ιδιαίτερο ή αισθητό πρόβλημα.

Με την ολοκλήρωση της εισαγωγής καινούργιου αντικειμένου καταγράφεται η ενέργεια στον κατάλογο ενεργειών χρήστη, ενημερώνεται ο τοπικός πίνακας ώστε να αντικατοπτρίζει τις τιμές στην απομακρυσμένη βάση και στέλνεται καινούργια ειδοποίηση με την καρτέλα πληροφοριών έκτακτης ανάγκης [22]. Το περιεχόμενο αυτής θα είναι ένας τίτλος για το είδος της πληροφορίας, καθώς και ένα σώμα ειδοποίησης με τις συμπληρωμένες κατηγορίες του αντικειμένου και τις τιμές που έχουν καταχωρηθεί.

Προχωρώντας στην επόμενη οθόνη εισαγωγής, η οθόνη εισαγωγής εξέτασης είναι προορισμένη για την εισαγωγή ραντεβού για εξέταση. Όπως και οι προηγούμενες οθόνες που αναφέρθηκαν, περιέχει πεδία εισαγωγής κειμένου, για τα δεδομένα που αναμένεται να εισαχθούν ως κείμενο, συγκεκριμένα το όνομα της τοποθεσίας που έχει κλείσει ραντεβού ο χρήστης, καθώς και κουμπιά που, όταν πατηθούν, ανοίγουν παράθυρα για την επιλογή ημερομηνίας και ώρας. Τα δεδομένα αυτά πρέπει να έχουν όλα συμπληρωθεί, προκειμένου το ραντεβού να εισαχθεί με επιτυχία. Το σημείο όπου αυτή η οθόνη διαφέρει από τις προηγούμενες που αναφέρθηκαν, είναι ότι περιέχει ένα κουμπί `DropDownButton`, με το οποίο ο χρήστης μπορεί να αποφασίσει αν η εξέταση είναι εργαστηριακή ή απεικονιστική, με την προεπιλεγμένη τιμή να είναι ότι η εξέταση θα είναι εργαστηριακή.

Η λογική για την συνάρτηση κατάθεσης του αντικειμένου στην απομακρυσμένη βάση απλά προσθέτει το αντικείμενο στον αντίστοιχο πίνακα της βάσης [5] και, αν αυτό γίνει με επιτυχία, ενημερώνει τον τοπικό πίνακα, τον τοπικό και τον απομακρυσμένο πίνακα ιστορικού, καθώς και τον κατάλογο ενεργειών χρήστη στη βάση. Ανάλογα με το αν η διαδικασία είναι επιτυχής ή όχι, εμφανίζεται μια μπάρα ενημέρωσης, με το κατάλληλο μήνυμα, για την πληροφόρηση του χρήστη.

Παρόμοια δομή έχει η οθόνη εισαγωγής εγχειρήσεων. Αποτελείται από ένα πεδίο εισαγωγής κειμένου για το όνομα της τοποθεσίας, καθώς και από ένα κουμπί για την επιλογή ημερομηνίας και ώρας. Η λογική της συνάρτησης καταχώρησης δεδομένων είναι ίδια με αυτή της οθόνης εισαγωγής εξέτασης.

Στην ίδια κατηγορία ανήκει και η οθόνη εισαγωγής επισκέψεων, που συμπληρώνει την ομάδα οθονών ραντεβού καλύπτοντας όλα τα ραντεβού που δεν είναι ούτε για εξετάσεις ούτε για εγχειρήσεις. Διαφέρει από την οθόνη εισαγωγής εγχειρήσεων μόνο στο ότι, μαζί με το πεδίο εισαγωγής κειμένου και τα κουμπιά που αναφέρθηκαν παραπάνω, περιέχει και ένα επιπλέον πεδίο για το είδος του ραντεβού. Σε αντίθεση με την οθόνη εισαγωγής εξετάσεων, η οθόνη αυτή δεν περιορίζει το εισαγόμενο είδος ραντεβού μόνο σε συγκεκριμένες τιμές και αφήνει την ονομασία στο χρήστη. Η λογική που ακολουθεί κατά τη διαδικασία καταχώρησης δεν παρουσιάζει αξιοσημείωτες αλλαγές από τις παραπάνω οθόνες.

Προχωρώντας πέρα από τις οθόνες εισαγωγής ραντεβού διάφορων τύπων, η επόμενη οθόνη είναι η οθόνη εισαγωγής ζωτικών σημείων. Στην οθόνη αυτή, ο χρήστης μπορεί να καταχωρήσει δεδομένα, συμπληρώνοντας πεδία κειμένου για τον παλμό, τη θερμοκρασία, το σάκχαρο και την πίεσή του. Η πίεση συγκεκριμένα καταχωρείται σε δύο διαφορετικά πεδία, ένα για την τιμή διαστολικής πίεσης και ένα για τη συστολική. Επιπλέον, η οθόνη έχει και ένα κουμπί για την επιλογή ημερομηνίας.

Καθώς προβλέπεται ο χρήστης να επιθυμεί να εισάγει δεδομένα μόνο για μερικές, ή και μόνο μία από τις μετρήσεις, η οθόνη αυτή έχει ένα επιπλέον στοιχείο, το οποίο δεν εμφανίζεται στις οθόνες, που έχουν ήδη αναφερθεί. Αυτό είναι ένα σύνολο κουμπιών εναλλαγής, ένα για κάθε κατηγορία. Όταν ένα από αυτά είναι πατημένο, το πεδίο εισαγωγής κειμένου για την αντίστοιχη μεταβλητή εμφανίζεται στην οθόνη, ενώ όλα τα πεδία των οποίων τα κουμπιά δεν είναι ενεργοποιημένα δεν εμφανίζονται στην οθόνη. Τα κουμπιά εναλλαγής έχουν ένα περιορισμό όπου, αν ο χρήστης προσπαθήσει να απενεργοποιήσει ένα κουμπί και αυτό είναι το τελευταίο ενεργό κουμπί της ομάδας, το κουμπί δε θα απενεργοποιηθεί. Έτσι, υπάρχει πάντα τουλάχιστον ένα κουμπί που παραμένει ενεργό και ένα τουλάχιστον πεδίο που είναι ορατό.

Συνεχίζοντας με τις οθόνες εισαγωγής, η οθόνη εισαγωγής φαρμάκων παρουσιάζει μια πιο περίπλοκη δομή, από αυτήν που αναφέρθηκε παραπάνω. Πέρα από τα πεδία εισαγωγής κειμένου, που χρησιμοποιούνται σε όλες τις οθόνες, ένα απαραίτητο για το όνομα του φαρμάκου και ένα προαιρετικό για οδηγίες χρήσης, παρέχει και μία σειρά από κουμπιά, για τη ρύθμιση της ώρας και της συχνότητας, με την οποία ο χρήστης θα λαμβάνει ειδοποιήσεις, για να παίρνει το φάρμακο που εισάγει [22].

Το πρώτο Widget, που πρέπει να αναφερθεί σε αυτή την περίπτωση, είναι ένα DropdownButton, προορισμένο για την επιλογή της συχνότητας, με την οποία ο χρήστης θα ειδοποιείται να πάρει το φάρμακό του. Ο χρήστης μπορεί να διαλέξει μεταξύ των πιο συνηθισμένων επιλογών, από μία μέχρι και τρεις φορές την ημέρα, μια φορά την εβδομάδα, πριν ή μετά τα γεύματα, ανά τακτικές ώρες, από κάθε τέσσερις μέχρι και κάθε οκτώ, ή απλά να επιλέξει ότι θα παίρνει το φάρμακο όποτε αυτός θέλει ή χρειάζεται. Κάθε επιλογή ορίζει προκαθορισμένες ώρες, που αρχίζουν και τελειώνουν οι ειδοποιήσεις, καθώς και το διάστημα μεταξύ ειδοποιήσεων. Οι ώρες εκκίνησης και αποκοπής εμφανίζονται ως κείμενο στην οθόνη, με διευκρίνιση του ρόλου που παίζουν.

Αν ο χρήστης επιθυμεί να θέσει διαφορετικές ώρες εκκίνησης και αποκοπής, δηλαδή την ώρα από την οποία και μετά θα λαμβάνει ειδοποιήσεις και την ώρα από όπου δε θα λαμβάνει ειδοποιήσεις, πλέον, για εκείνη την ημέρα, υπάρχει μια μέθοδος γι' αυτό. Με την αλλαγή του ενεργού αντικειμένου σε ένα RadioGroup Widget, από τη ρύθμιση χρήσης τυπικών τιμών στη ρύθμιση χειροκίνητης εισαγωγής, μπορεί να εμφανίσει δύο κουμπιά εικονιδίου για τον καθορισμό της κάθε ώρας. Αυτά τα κουμπιά θα εμφανίζονται δίπλα στο κείμενο για την αντίστοιχη ώρα και, όταν πατηθεί ένα από αυτά, ανοίγουν ένα παράθυρο για την επιλογή ώρας.

Ένας περιορισμός αυτής της εφαρμογής είναι ότι ο χρήστης δεν μπορεί να θέσει το διάστημα μεταξύ ειδοποιήσεων ελεύθερα, πέρα από τις επιλογές που του δίνονται. Συγκριμένα, κάθε επιλογή στο DropdownButton θέτει το διάστημα μεταξύ ειδοποιήσεων και το είδος αυτής της αναμονής, μεταξύ ωρών και ημερών. Από την ώρα εκκίνησης μέχρι την ώρα αποκοπής, εφόσον έχουν οριστεί, θα στέλνονται ειδοποιήσεις [22] για το κάθε φάρμακο, στο οποίο έχει θέσει ότι θέλει να ενημερώνεται, όποτε πρέπει να το πάρει για όσο διάστημα χρειάζεται, από την πρώτη μέρα που ξεκινά η φαρμακευτική αγωγή μέχρι την τελευταία.

Όσο για τις ημερομηνίες αρχής και τέλους φαρμακευτικής αγωγής, αυτές οι τιμές μπορούν να οριστούν μέσω κουμπιών επιλογής ημερομηνίας, όπως και σε προαναφερόμενες οθόνες. Τα κουμπιά αυτά είναι διαθέσιμα ακόμα και στη ρύθμιση χρήσης τυπικών τιμών.

Σε περίπτωση που ο χρήστης, όμως, επιλέξει ότι θα παίρνει το φάρμακο που εισάγει όποτε θέλει, χωρίς πρόγραμμα, το αντικείμενο που εισάγεται θα σημαδευτεί ότι δεν χρειάζεται να στέλνεται ειδοποίηση από την εφαρμογή. Επιπλέον, η επιλογή αυτή κρίνεται ότι δεν χρειάζεται τα πεδία προγραμματισμού της ώρας ή ημερομηνίας, οπότε όλα τα κουμπιά και τα πεδία με τις τιμές δεν εμφανίζονται και δεν εισάγονται με το αντικείμενο.

4.2.8 Οθόνες εισαγωγής αποτελεσμάτων

Τελευταίες είναι οι οθόνες εισαγωγής αποτελεσμάτων εξετάσεων και εγχειρήσεων. Οι δύο αυτές οθόνες έχουν την ίδια δομή, με μοναδική διαφορά τη συλλογή του Firebase [5], στην οποία στέλνονται τα εισαγόμενα δεδομένα από την κάθε μία.

Οι οθόνες αυτές επιτρέπουν στο χρήστη να καταγράψει τα αποτελέσματά του με δύο διαφορετικούς τρόπους. Ένας από αυτούς είναι να ανεβάσει το περιεχόμενο ενός αρχείου PDF στη βάση δεδομένων, όπου το κείμενο του αρχείου θα διαβαστεί [25] και θα αποθηκευτεί. Αν ο χρήστης, για κάποιο λόγο, δεν επιθυμεί να ανεβάσει το περιεχόμενο του αρχείου, ή αντιμετωπίζει δυσκολίες σε αυτό, υπάρχει μια εναλλακτική μέθοδος, που του επιτρέπει να γράψει ο ίδιος τα αποτελέσματα, όπως αυτός θέλει να καταγραφούν και να εισάγει αυτό.

Ξεκινώντας με τα στοιχεία της δομής, που υποστηρίζουν την εναλλακτική μέθοδο χειροκίνητης εισαγωγής κειμένου, οι οθόνες έχουν δύο πεδία εισαγωγής κειμένου, για τον τίτλο και το περιεχόμενο του κειμένου, καθώς και ένα κουμπί επιλογής κειμένου. Για το πεδίο του περιεχομένου, κάθε εισαγωγή στο πεδίο προσθέτει μία γραμμή στο κείμενο, που θα ανεβάσει η εφαρμογή στη βάση δεδομένων, όταν κληθεί η διαδικασία εισαγωγής δεδομένων. Το κείμενο που θα εισαχθεί παρουσιάζεται στο χρήστη σε ένα κουτί στο κάτω μέρος της οθόνης, το οποίο έχει και ένα κουμπί για τη διαγραφή της τελευταίας γραμμής που προστέθηκε.

Η εισαγωγή κειμένου από αρχείο, από την άλλη, γίνεται με το πάτημα ενός κουμπιού. Αυτό θα πλοηγήσει το χρήστη σε μία προσωρινή οθόνη επιλογής αρχείου, που παρέχεται από το πακέτο «file_picker» [24]. Εφόσον ο χρήστης επιλέξει ένα αρχείο PDF, το όνομα του αρχείου χρησιμοποιείται ως ο τίτλος για το αντικείμενο που θα εισαχθεί, ενώ το περιεχόμενο του αρχείου διαβάζεται και εισάγεται ως η κατηγορία περιεχομένου του αντικειμένου. Η ανάγνωση αυτή απαιτεί τη συνεργασία διάφορων πακέτων, με το διάβασμα του περιεχομένου του κειμένου να γίνεται μέσω του «flutter_pdf_text» [25]. Άδεια για αυτή την ανάγνωση ζητείται από τον «permission_handler» [26] ή, εφόσον η συσκευή χρησιμοποιεί Android με SDK 33 ή παραπάνω, ορίζεται από την εφαρμογή, με τη χρήση του ίδιου πακέτου. Ο αριθμός έκδοσης του SDK της συσκευής παρέχεται με τη βοήθεια του «device_info_plus» [27]. Το όνομα του αρχείου που επιλέχθηκε παρουσιάζεται στο κείμενο σε ένα πεδίο κειμένου. Αν δεν έχει επιλεγεί αρχείο, το κείμενο αλλάζει, ώστε να ενημερώνει το χρήστη για αυτό.

Η κατάθεση δεδομένων για χειροκίνητη εισαγωγή, ή εισαγωγή κειμένου από αρχείο, γίνεται μέσω δύο διαφορετικών κουμπιών, για τη διευκόλυνση του χρήστη. Σε περίπτωση που ο χρήστης θέλει να ακυρώσει την επιλογή αρχείου του, υπάρχει επίσης και ένα κουμπί ακύρωσης, που θα σβήσει τις τιμές των μεταβλητών αυτών.

Πέρα από αυτές τις λειτουργίες, οι οθόνες αυτές παρέχουν και τη δυνατότητα να διαγράψει ο χρήστης ένα αρχείο που ανέβασε, ή να κατεβάσει το περιεχόμενο ενός αντικειμένου και να το μετατρέψει σε αρχείο PDF [25].

Πρώτο βήμα για την πρόσβαση σε αυτές τις λειτουργίες είναι το πάτημα ενός ακόμα κουμπιού που βρίσκεται δίπλα στο κουμπί εισαγωγής κειμένου από αρχείο. Αυτό θα προκαλέσει διάφορες αλλαγές στην οθόνη, με πολλά υπάρχοντα κουμπιά να αλλάζουν κείμενο και λειτουργία, ή να εξαφανιστούν εντελώς από την οθόνη, ενώ άλλα Widget εμφανίζονται για πρώτη φορά. Οι αλλαγές αυτές μπορούν να αντιστραφούν και η οθόνη θα γυρίσει πίσω στη λειτουργία ανεβάσματος δεδομένων, με τη χρήση του κουμπιού κατεβάσματος αρχείου, του οποίου το κείμενο θα έχει αλλάξει για να δείξει ότι είναι πλέον το κουμπί ακύρωσης.

Σχετικά με τις αλλαγές που θα γίνουν στην οθόνη, το πρώτο που θα αλλάξει θα είναι το κουμπί χειροκίνητης εισαγωγής, που θα εξαφανιστεί από την οθόνη. Επιπλέον, το πεδίο με το κείμενο ενημέρωσης της κατάστασης επιλογής αρχείου θα αντικατασταθεί με ένα DropdownButton. Οι επιλογές σε αυτό το κουμπί θα είναι οι τίτλοι όλων των αρχείων που έχει ανεβάσει ο χρήστης, από τις οποίες θα μπορεί να διαλέξει οποιαδήποτε.

Τέλος, τα κουμπιά εισαγωγής αρχείου και ακύρωσης επιλογής θα αλλάξουν, ώστε να υποστηρίζουν το κατέβασμα αρχείου και τη διαγραφή αρχείου αντίστοιχα. Το πρώτο από αυτά παίρνει το επιλεγμένο αντικείμενο και μετατρέπει το κείμενο του περιεχομένου του σε σελίδες PDF, οι οποίες θα καταγραφούν ως καινούργιο αρχείο, με όνομα που θα είναι ίδιο με τον τίτλο του αντικειμένου. Το κουμπί διαγραφής, από την άλλη, διαγράφει το επιλεγμένο αντικείμενο από τη βάση δεδομένων, την προσωρινή μνήμη της εφαρμογής, καθώς και από το ιστορικό του χρήστη.

4.2.9 Οθόνες τροποποίησης και διαγραφής

Προχωρώντας στις οθόνες τροποποίησης και διαγραφής, οι διάφορες οθόνες, που ανήκουν σε αυτές τις κατηγορίες, έχουν κοινά σημεία μεταξύ τους και μπορούν να χωριστούν σε δύο κατηγορίες. Η πρώτη κατηγορία είναι οι οθόνες που υπολογίζουν μόνο για ένα αντικείμενο, αποθηκευμένο από το χρήστη στην αντίστοιχη συλλογή της βάσης δεδομένων και περιλαμβάνουν μόνο τις οθόνες τροποποίησης και διαγραφής στόχων και πληροφοριών εκτάκτου ανάγκης. Περιέχουν κυρίως τα ίδια σύνολα κουμπιών και πεδίων εισαγωγής δεδομένων όπως η αντίστοιχη οθόνη εισαγωγής, αλλά διαφέρουν μεταξύ οθονών διαγραφής και τροποποίησης ως προς τη λειτουργία τους κατά την καταχώρηση δεδομένων.

Για τις οθόνες τροποποίησης, εκτελείται έλεγχος για να εξακριβωθεί αν υπάρχει ήδη αντικείμενο αποθηκευμένο στην κατάλληλη συλλογή και, αν δεν υπάρχει τίποτα, απλά αποθηκεύεται το καινούργιο αντικείμενο, με τα δεδομένα που εισήγαγε ο χρήστης [5]. Αν όμως βρεθεί αντικείμενο στη βάση, το παλιό αντικείμενο θα συγκριθεί με το καινούργιο και τα πεδία, στα οποία έχουν εισαχθεί τιμές, θα αντικαταστήσουν τις τιμές των αντίστοιχων πεδίων του παλιού αντικειμένου. Παλιά δεδομένα, σε πεδία που παρέμειναν κενά στην οθόνη, δε θα αλλάξουν.

Στις οθόνες διαγραφής αυτής της κατηγορίας, τα μόνα πεδία εισαγωγής κειμένου που παραμένουν είναι τα πεδία, για τα οποία προβλέπονται πολλαπλές εισαγωγές, όπως οι αλλεργίες του χρήστη. Τα υπόλοιπα πεδία θα αντικατασταθούν από κουμπιά, για τη διαγραφή ή μη διαγραφή του πεδίου στο αντικείμενο, αν το πεδίο είναι προαιρετικό, ή απλά δεν θα υπάρχουν, αν το πεδίο είναι απαραίτητο. Το κουμπί καταχώρησης εκτελεί μία συνάρτηση που διαγράφει όλα τα πεδία, ή/και τις τιμές πεδίων, που έχουν δηλωθεί. Οι οθόνες περιέχουν και ένα ακόμα κουμπί, για την ολική διαγραφή του αντικειμένου, ώστε να είναι πιο εύκολη η διαδικασία για το χρήστη.

Η δεύτερη κατηγορία οθονών τροποποίησης και διαγραφής είναι αυτές όπου προβλέπεται ο χρήστης να έχει πολλαπλά αντικείμενα ανά συλλογή στη βάση δεδομένων. Οι οθόνες αυτές χρησιμοποιούν παρόμοιες ή και ίδιες λίστες, για την παρουσίαση όλων των αντικειμένων, που βρίσκονται στο αντίστοιχο τοπικό αρχείο δεδομένων. Με το σύρσιμο ενός αντικειμένου της λίστας στο πλάι, το αντικείμενο αυτό μπορεί να διαγραφεί από το τοπικό αρχείο και, μετά από μία μικρή περίοδο, που δίνεται στο χρήστη για να ακυρώσει την απόφασή του, από την απομακρυσμένη βάση. Σε περίπτωση που ο χρήστης ακυρώσει αυτή την απόφαση, το αντικείμενο θα προστεθεί πίσω στο τοπικό αρχείο και τη λίστα και η διαγραφή του αντικειμένου στην απομακρυσμένη βάση δε θα εκτελεστεί.

Επιπλέον, κρατώντας πατημένο το αντικείμενο της λίστας που επιθυμούν να τροποποιήσουν, η οθόνη προβάλλει ένα παράθυρο με μία φόρμα τροποποίησης δεδομένων. Η δομή της φόρμας εξαρτάται από το είδος του αντικειμένου, που παρουσιάζει η οθόνη, αλλά συνήθως ακολουθεί τη δομή των αντίστοιχων οθονών εισαγωγής. Ο χρήστης εισάγει καινούργια δεδομένα, τα οποία θα πάρουν τη θέση των προηγούμενων τιμών του αντικειμένου. Πεδία στα οποία δεν έχουν συμπληρωθεί τιμές θα συνεχίσουν να χρησιμοποιούν τις προηγούμενες τιμές τους. Η λειτουργία αυτή υπάρχει ανεξάρτητα αν πρόκειται για οθόνη τροποποίησης ή διαγραφής.

Οι οθόνες της δεύτερης κατηγορίας χρησιμοποιούν την ίδια ακριβώς λίστα, με την ίδια λογική μεταξύ οθόνης διαγραφής και οθόνης τροποποίησης ανά αντικείμενο, με μόνη διαφορά τον τίτλο στην κορυφή της οθόνης. Η διάκριση μεταξύ τους υπάρχει κυρίως για την ευκολία της χρήσης της εφαρμογής, καθώς γίνεται ξεκάθαρο στο χρήστη προς ποια επιλογή να μεταβεί και τι είδους λειτουργίες μπορεί να αναμένει στην κάθε οθόνη.

4.2.10 Οθόνη Επαφών Εκτάκτου Ανάγκης

Επιστρέφοντας στην οθόνη ιατρικών, η επόμενη οθόνη στην οποία μπορεί να μεταβεί ο χρήστης είναι η οθόνη λίστας επαφών εκτάκτου ανάγκης. Η οθόνη αυτή παρέχει γρήγορη πρόσβαση στις λεπτομερείς πληροφορίες επαφών εκτάκτου ανάγκης, που έχει καταγράψει ο χρήστης, σε περίπτωση που χρειαστεί. Αποτελείται από μία λίστα, όπου κάθε αντικείμενο δείχνει το όνομα και τον αριθμό μιας επαφής. Για γρήγορη τροποποίηση αυτών των δεδομένων από το χρήστη, το κάθε αντικείμενο μπορεί να συρθεί προς το πλάι και να διαγραφεί από τη λίστα, το τοπικό αρχείο, καθώς και τη βάση δεδομένων.

4.2.11 Οθόνη λήψης/αποστολής αρχείων

Επόμενη από τις οθόνες-προορισμούς είναι η οθόνη ανεβάσματος και κατεβάσματος αρχείων. Η οθόνη αυτή, ανάλογα με τη λειτουργία την οποία έχει επιλέξει ο χρήστης εκείνη τη στιγμή μέσω δύο κουμπιών, μπορεί να παρέχει υπηρεσίες για το ανέβασμα ή τη λήψη αρχείων.

Με το πάτημα του κουμπιού επιλογής αρχείου, ο χρήστης μπορεί να επιλέξει αρχείο από τη συσκευή του [24] για να την ανεβάσει. Το όνομα του αρχείου παρουσιάζεται σε ένα πλαίσιο κειμένου ή, αν δεν έχει επιλεγεί ακόμα αρχείο και η οθόνη είναι σε λειτουργία ανεβάσματος, το πλαίσιο ενημερώνει το χρήστη ότι δεν έχουν επιλεγεί ακόμα αρχεία. Το αρχείο μπορεί μετά να σταλθεί στη βάση δεδομένων μέσω του κουμπιού καταχώρησης, ή να ακυρωθεί μέσω του κουμπιού ακύρωσης. Πρέπει να σημειωθεί ότι αυτό που στέλνεται στη βάση δεδομένων είναι το κείμενο του αρχείου και μόνο αυτό [25].

Πατώντας όμως το κουμπί λήψης, η οθόνη αλλάζει σε λειτουργία λήψης αρχείου. Ψάχνει στη βάση δεδομένων [5] για ανεβασμένα αρχεία και τα παρουσιάζει ως επιλογές σε ένα DropDownButton, με τις ημερομηνίες στις οποίες ανέβηκε το κάθε αρχείο και τον τίτλο τους. Σε αυτή τη λειτουργία, τα κουμπιά ανεβάσματος και ακύρωσης μετατρέπονται σε κουμπιά λήψης και διαγραφής. Το κουμπί λήψης δημιουργεί ένα αρχείο PDF, με τη βοήθεια του πακέτου «pdf» [23], και, έχοντας πρόσβαση στο χώρο

αποθήκευσης της συσκευής χάρη στο πακέτο «path_provider» [28], το αποθηκεύει στη συσκευή και επιστρέφει την οθόνη στη λειτουργία επιλογής αρχείου. Το κουμπί διαγραφής, από την άλλη, διαγράφει το συγκεκριμένο αρχείο από τη βάση δεδομένων.

4.2.12 Οθόνη δημιουργίας αρχείων

Πέρα από την οθόνη ανεβάσματος και κατεβάσματος αρχείων, υπάρχει και η οθόνη δημιουργίας PDF. Η οθόνη αυτή είναι σχεδιασμένη ώστε να παρουσιάζει μία λίστα για το είδος αντικειμένου, που ενδιαφέρει το χρήστη, μεταξύ Πάθησης, Εγχείρησης, Ραντεβού με γιατρό, Φαρμάκου και Ζωτικών Σημείων. Ο χρήστης επιλέγει την κατηγορία που τον ενδιαφέρει, μέσω ενός DropDownButton και η αντίστοιχη λίστα εμφανίζεται. Τα αποτελέσματα εξετάσεων και εγχειρήσεων δεν είναι επιλογές, επειδή θεωρείται δεδομένο ότι τα σχετικά αρχεία, που μπορεί να έχει καταθέσει ο χρήστης, είναι ήδη στην κατοχή του και μορφοποιημένα με βάση τους κώδικες που απαιτούνται σε κάθε περίπτωση.

Δίπλα στο πρώτο DropDownButton υπάρχει και ένα δεύτερο, που αφορά τη δομή του περιεχομένου αρχείου, που θέλει να δημιουργήσει ο χρήστης. Η πρώτη επιλογή κρατάει το κείμενο που θα δημιουργηθεί σε μια απλή μορφή, που μπορεί να διαβαστεί από τον οποιοδήποτε. Οι άλλες δύο επιλογές, όμως, το μετατρέπουν σε κείμενο που ακολουθεί τον κώδικα FHIR ή HL7 αντίστοιχα [1] [2] [3]. Ο χρήστης μπορεί να ξεκινήσει τη διαδικασία κρατώντας πατημένο οποιοδήποτε αντικείμενο της λίστας που προβάλλεται, οπότε θα ανοίξει ένας διάλογος επιβεβαίωσης, όπου απαιτείται έγκριση, για την δημιουργία του αρχείου. Σε περίπτωση ακύρωσης, ο διάλογος θα κλείσει.

4.2.13 Οθόνη τελευταίας μηνιαίας αναφοράς

Μια από τις πιο απλές οθόνες-προορισμούς, που είναι διαθέσιμες στο χρήστη, είναι η οθόνη τελευταίας αναφοράς. Είναι σχεδιασμένη ώστε, κατά την είσοδο του χρήστη σε αυτήν, να εκτελείται μία αναζήτηση, για την πιο πρόσφατη μηνιαία αναφορά αποθηκευμένη στη συσκευή [8]. Με την ολοκλήρωση αυτής της αναζήτησης, η αναφορά προβάλλεται σε ένα πλαίσιο κειμένου ή, αν δεν βρεθεί αναφορά, το κείμενο που εμφανίζεται απλά ενημερώνει το χρήστη για αυτό.

4.2.14 Οθόνη ημερολογίου

Απαραίτητο μέρος μιας εφαρμογής παρατήρησης ιατρικών δεδομένων και ραντεβού είναι η δυνατότητα, για το χρήστη, να μπορεί να βλέπει τις ημερομηνίες, στις οποίες έχει προγραμματίσει κάποιο ραντεβού, για κάθε είδος ραντεβού, που προβλέπεται να έχει κανονίσει. Για αυτή την εργασία, οι τρεις προβλεπόμενοι τύποι είναι ραντεβού για εγχείρηση, ραντεβού για εξέταση, ή ραντεβού κάποιου άλλου είδους.



Σχήμα 4.5: Screenshot Οθόνης Ημερολογίου

Την ανάγκη αυτή ικανοποιεί η επόμενη οθόνη προς πλοήγηση που θα αναφερθεί, η οθόνη ιατρικού ημερολογίου. Το κύριο κομμάτι αυτής της οθόνης είναι ένα απλό ημερολόγιο [20]. Σε αυτό το ημερολόγιο προβάλλονται με διαφορετικά χρώματα οι μέρες όπου έχει προγραμματιστεί ραντεβού, ανάλογα με το είδος ραντεβού που έχει προγραμματιστεί. Τα ραντεβού για εγχείρηση χρωματίζουν τη συγκεκριμένη ημέρα πορτοκαλί, δείχνοντας ότι είναι ένα πιο επείγον ραντεβού. Ακολουθώντας αυτή τη λογική, οι ημέρες με προγραμματισμένο ραντεβού για εξέταση χρωματίζονται με ένα πιο κίτρινο χρώμα, ενώ αυτές με προγραμματισμένο ραντεβού κάποιου άλλου είδους χρωματίζονται πράσινες. Σε περίπτωση που έχουν προγραμματιστεί πολλά διαφορετικά γεγονότα διαφορετικών κατηγοριών για την ίδια ημέρα, προτεραιότητα δίνεται στις εγχειρήσεις για το χρώμα που θα πάρει η ημέρα στο ημερολόγιο, ακολουθούμενη από τα ραντεβού για εξετάσεις και τελευταία αυτά για οποιοδήποτε άλλο σκοπό. Οι ημέρες χωρίς κάποιο ραντεβού διατηρούν το προεπιλεγμένο τους χρώμα.

Το ημερολόγιο [20] επιτρέπει στο χρήστη να αγγίξει οποιαδήποτε μέρα σε αυτό και, ανάλογα με το αν η ημέρα που επιλέχθηκε περιέχει προγραμματισμένα ραντεβού ή όχι, εμφανίζει διαφορετικά επιπλέον στοιχεία. Η επιλεγμένη ημέρα, αν δεν έχει ραντεβού, εμφανίζεται ως μπλε, ή, αν έχει ραντεβού, με ένα από τα χρώματα και τους κανόνες που αναφέρθηκαν παραπάνω.

Η πρώτη περίπτωση που θα αναφερθεί είναι το στοιχείο που εμφανίζεται αν η ημέρα δεν περιέχει τίποτα. Σε αυτή την περίπτωση, θεωρείται ότι ο χρήστης προσπαθεί να εισάγει ένα καινούργιο αντικείμενο κάποιου τύπου απευθείας από το ημερολόγιο, αντί να περάσει από τις οθόνες εισαγωγής. Επομένως, εμφανίζεται ένα παράθυρο, με τη φόρμα για την επιλογή είδους αντικειμένου και την εισαγωγή δεδομένων για αυτό το αντικείμενο.

Η κατηγορία αντικειμένου επιλέγεται με τη χρήση ενός DropDownButton, που παρέχει επιλογές μεταξύ εγχειρήσεων, ραντεβού τύπου που δεν προβλέπεται, καθώς και των δύο διαφορετικών ειδών εξετάσεων. Όλες οι κατηγορίες παρέχουν πλαίσια εισαγωγής κειμένου για το όνομα της τοποθεσίας, είτε αυτό είναι το όνομα ενός κτηρίου, ενός γιατρού ή κάτι άλλο, καθώς και ένα κουμπί για τον ορισμό της ώρας του ραντεβού.

Πέρα από αυτό, η κατηγορία ραντεβού μη προβλεπόμενου τύπου περιλαμβάνει και ένα επιπλέον πλαίσιο για την εισαγωγή του τύπου ραντεβού, με το περιεχόμενο να εξαρτάται εντελώς από το χρήστη, χωρίς περιορισμούς. Για την ημερομηνία του ραντεβού, χρησιμοποιείται η επιλεγμένη ημερομηνία.

Τα εισαγόμενα δεδομένα μπορούν να καταχωρηθούν μέσω ενός κουμπιού. Εναλλακτικά, ο χρήστης μπορεί να κλείσει το παράθυρο και να ακυρώσει την όλη διαδικασία με το πάτημα του κουμπιού ακύρωσης. Αν τα δεδομένα καταχωρηθούν με επιτυχία, το αντικείμενο θα προστεθεί και στις αντίστοιχες τοπικές λίστες.

Στη δεύτερη περίπτωση, δηλαδή αν η ημέρα που επιλέχθηκε περιέχει κάποιο ραντεβού, η οθόνη θα εμφανίσει μία λίστα ραντεβού. Κάθε αντικείμενο περιέχει την τοποθεσία, την ημερομηνία και την ώρα του ραντεβού, καθώς και ένα μικρό χρωματισμένο κουτί με το αντίστοιχο χρώμα για την κατηγορία του αντικειμένου. Πάνω από την λίστα εμφανίζεται και ένα κουμπί για την εισαγωγή καινούργιου αντικειμένου, το οποίο εμφανίζει το ίδιο παράθυρο που αναφέρθηκε παραπάνω, όταν πατηθεί.

Με αυτόν τον τρόπο, ο χρήστης διατηρεί την ικανότητα εισαγωγής δεδομένων μέσω του ημερολογίου, ακόμα και σε μέρες όπου έχει ήδη εισάγει αντικείμενα. Αξιοσημείωτο είναι ότι ο χρήστης δεν μπορεί να διαγράψει αντικείμενα από αυτή τη λίστα, καθώς δε θεωρείται απαραίτητο κομμάτι της λειτουργίας αυτής της οθόνης.

4.2.15 Γραφήματα της εφαρμογής

Τελευταία από τις οθόνες είναι η οθόνη γραφημάτων. Η οθόνη αυτή παρουσιάζει στο χρήστη γραφήματα [21] για διάφορα ιατρικά στατιστικά του, στην πάροδο του χρόνου. Τα περισσότερα από αυτά τα γραφήματα παρουσιάζουν τις μετρήσεις διάφορων ζωτικών σημείων, είτε είναι η θερμοκρασία, ο παλμός, το σάκχαρο ή η πίεση του χρήστη.

Τα γραφήματα για αυτά τα δεδομένα είναι γραμμικά γραφήματα [21], που χρησιμοποιούν τις μετρήσεις ως τιμές στον κάθετο άξονα, ενώ ο οριζόντιος άξονάς τους απλά σημειώνει τη θέση της ένδειξης στη σχετική λίστα. Οι μετρήσεις στη λίστα είναι ταξινομημένες με χρονολογική σειρά, με τις πιο πρόσφατες να είναι τοποθετημένες προς το τέλος της λίστας, οπότε αρκεί απλά να σημειώνονται οι κατακόρυφες θέσεις τους ανά τακτά διαστήματα για να γίνεται φανερή η πρόοδος στο χρόνο, αν και η μέθοδος αυτή δεν παρέχει πληροφορίες για το χρονικό διάστημα μεταξύ μετρήσεων.

Πέρα από τα γραμμικά γραφήματα που χρησιμοποιούνται για τις μετρήσεις ζωτικών σημείων, υπάρχει και μία τελευταία κατηγορία δεδομένων, που χρησιμοποιεί ένα διαφορετικό είδος γραφήματος. Η κατηγορία αυτή είναι οι παθήσεις/ασθένειες του χρήστη. Συγκεκριμένα, το γράφημα που δημιουργείται παρουσιάζει το πλήθος παθήσεων του χρήστη ανά μήνα για μία επιλεγμένη χρονιά. Αυτό επιτυγχάνεται με τη χρήση κατακόρυφων μπαρών [21], με τον οριζόντιο άξονα να περιέχει σημειωμένους τους μήνες.

Η εμφάνιση του πλήθους παθήσεων του με αυτό τρόπο μπορεί να πληροφορήσει το χρήστη για περιόδους χρόνου όπου είναι πιο επιρρεπής σε ασθένειες, ή απλά να του επιτρέπει να επιβεβαιώνει την κατάσταση της υγείας του για εκείνη τη χρονιά με απλό τρόπο, χωρίς να χρειάζεται να ελέγχει ο ίδιος το ιστορικό του και να συνδέσει τις διάφορες εισαγωγές σε μία αναφορά.

Σχετικά με τη δομή της ίδιας της οθόνης γραφημάτων, περιέχει δύο στοιχεία που παραμένουν διαθέσιμα στην οθόνη, ανεξάρτητα από οτιδήποτε άλλο. Αυτά τα στοιχεία είναι δύο κουμπιά τύπου DropDownButton. Το πρώτο από αυτά απλά διαλέγει μεταξύ των διάφορων κατηγοριών δεδομένων, που έχουν αναφερθεί παραπάνω, αλλάζοντας και ξαναχτίζοντας το αντίστοιχο γράφημα, ανάλογα με την επιλογή του χρήστη. Προεπιλεγμένη είναι η κατηγορία θερμοκρασίας.

Το δεύτερο μόνιμο στοιχείο είναι ένα ακόμα κουμπί DropDownButton, το οποίο ασχολείται μόνο με το έτος που ο χρήστης επιθυμεί να αναλύσει. Σε αντίθεση με το προηγούμενο κουμπί, οι επιλογές αυτού δεν είναι σταθερές, αλλά ορίζονται από μια λίστα που αλλάζει με βάση τις εισαγωγές του χρήστη. Αν ο χρήστης έχει εισαγωγές μετρήσεων με ημερομηνία παρατήρησης ένα προηγούμενο έτος, το έτος αυτό καταχωρείται ως επιλογή για το κουμπί και η διαδικασία επαναλαμβάνεται για κάθε διαφορετικό έτος που έχει καταχωρηθεί. Αν δεν υπάρχει ημερομηνία σε προηγούμενο έτος καταχωρημένη, ή αν απλά δεν υπάρχουν αντικείμενα στη λίστα μετρήσεων ζωτικών σημείων, η μόνη επιλογή διαθέσιμη θα είναι το τρέχον έτος. Ανεξάρτητα από το πλήθος επιλογών, προεπιλεγμένη τιμή είναι το τρέχον έτος.

Πέρα από αυτά τα δύο στοιχεία, ανάλογα με την επιλογή στο πρώτο κουμπί με τις κατηγορίες δεδομένων, η οθόνη μπορεί να εμφανίσει ένα από δύο διαφορετικά στοιχεία. Αν επιλεγμένη είναι η κατηγορία παθήσεων, εμφανίζεται το γράφημα μπαρών, που αναφέρθηκε παραπάνω [21].

Με τους μήνες ως τίτλους στον οριζόντιο άξονα και τον κατακόρυφο άξονα να έχει μέγιστη τιμή τη μεγαλύτερη τιμή για πλήθος παθήσεων, που βρίσκεται στη λίστα παθήσεων ανά μήνα, το γράφημα αυτό κατηγοριοποιεί όλες τις τιμές ημερομηνιών ανά χρόνο και μήνα και μετράει το πλήθος παθήσεων στον κάθε ένα. Αν δεν υπάρχουν δεδομένα στη λίστα, δεν θα εμφανιστούν μπάρες στο γράφημα. Επειδή θα θεωρηθεί ότι όλα τα αντικείμενα της λίστας έχουν μηδέν συμβάντα μετρημένα.

Για όλες τις υπόλοιπες επιλογές, θα εμφανιστεί ένα γραμμικό γράφημα [21]. Τα γραφήματα αυτά, πέρα από την γραμμή που σχηματίζεται από τις διαφορετικές μετρήσεις που προβάλλονται [21], παρουσιάζουν επίσης και μία χωριστή γραμμή [21], με την τυπική τιμή, που θα θεωρούνταν κανονική για την κατηγορία. Για παράδειγμα, για τη θερμοκρασία, η τυπική κανονική τιμή που θα χρησιμοποιηθεί είναι το 37, καθώς αυτό θεωρείται μια κανονική τιμή θερμοκρασίας του σώματος.

4.2.16 Υπηρεσίες

Έχοντας πλέον τελειώσει με τις οθόνες της εφαρμογής, θα πρέπει να αναφερθούν και μερικές από τις υπηρεσίες που τρέχουν στα παρασκήνια. Η πρώτη από αυτές είναι οι υπηρεσίες για την αποστολή ειδοποιήσεων για την έγκυρη λήψη φαρμακευτικής αγωγής από το χρήστη [22] [29]. Από τη στιγμή που η εφαρμογή θέτεται σε εκκίνηση, ελέγχεται κάθε λίγα λεπτά αν ο χρήστης έχει αντικείμενα στην τοπική λίστα φαρμάκων και, αν ναι, το αν επιθυμεί να λάβει ειδοποιήσεις για αυτά σε κάποιο σημείο.

Για κάθε αντικείμενο της λίστας, πραγματοποιείται επιπλέον έλεγχος αν ο χρήστης έχει θέσει ότι επιθυμεί να ειδοποιείται για αυτό, αν έχει θέσει πρώτη ώρα λήψης ειδοποιήσεων, καθώς και ποια είναι η ώρα, πέρα από την οποία δε θέλει να λαμβάνει άλλες ειδοποιήσεις. Εφόσον υπάρχει ώρα πρώτης ειδοποίησης, η εφαρμογή υπολογίζει τις ώρες στις οποίες πρέπει να στείλει ειδοποίηση, με βάση το δηλωμένο διάστημα μεταξύ ειδοποιήσεων, που έχει αποθηκευτεί στο αντικείμενο και σημειώνει αν έχει ήδη σταλθεί ειδοποίηση για εκείνη την ημέρα σε κάποιο σημείο. Αν δεν έχει σταλθεί και είναι η

κατάλληλη ώρα, στέλνεται καινούργια ειδοποίηση για το συγκεκριμένο φάρμακο και ενημερώνεται η εφαρμογή για αυτό, ώστε να μη σταλθούν άλλες ειδοποιήσεις μέχρι την επόμενη ώρα που έχει υπολογιστεί.

Σε περίπτωση που ο χρήστης έχει δηλώσει ότι επιθυμεί να ενημερώνεται μόνο μια φορά την ημέρα, ελέγχεται αν έχει παρέλθει η ώρα απενεργοποίησης ειδοποιήσεων. Αν όχι, στέλνεται ειδοποίηση, αν και η διαδικασία σταματά. Η ειδοποίηση θα σταλθεί μία φορά στην πρώτη εκτέλεση της υπηρεσίας και όχι παραπάνω [22].

Μια άλλη υπηρεσία, που δημιουργεί ειδοποιήσεις για το χρήστη, είναι η υπηρεσία μηνιαίας αναφοράς. Η υπηρεσία αυτή καλείται μία φορά, είτε στην επιτυχή σύνδεση του χρήστη στο λογαριασμό του, είτε στο άνοιγμα της εφαρμογής, ανάλογα με το αν ο χρήστης έχει παραμείνει συνδεδεμένος από προηγούμενη χρήση της εφαρμογής.

Στο πρώτο βήμα, η υπηρεσία αυτή αναζητάει στη βάση δεδομένων [5] την πιο πρόσφατη αναφορά και, αν υπάρχει, καταγράφει την αναφορά και την ημερομηνία που έγινε στις κοινές προτιμήσεις της εφαρμογής [8]. Αν δε βρεθεί αναφορά, αποθηκεύεται ένα μικρό μήνυμα για αυτό, μαζί με την τρέχουσα ημερομηνία, ενώ ταυτόχρονα δημιουργείται μία πρώτη αναφορά για τη βάση δεδομένων, με το ίδιο μήνυμα αντί για την κανονική δομή της αναφοράς [8], προκειμένου να αποφευχθούν περιπτώσεις όπου ο χρήστης έχει πάνω από ένα μήνα χωρίς να λάβει αναφορά, αλλά έχει είτε αδειάσει τις κοινές προτιμήσεις της εφαρμογής ή έχει διαγράψει και εγκαταστήσει ξανά την εφαρμογή.

Η κανονική δομή της αναφοράς αποτελείται από ένα σύντομο κείμενο, που ενημερώνει το χρήστη για το πλήθος παθήσεων και ραντεβού που εισήγαγε τον τελευταίο μήνα. Επιπλέον, παρουσιάζονται οι πιο πρόσφατοι στόχοι ή, αν δεν υπάρχουν, ένα κείμενο που σημειώνει ότι τα πεδία στόχων δεν έχουν δεδομένα. Με παρόμοιο τρόπο παρουσιάζεται και η αναφορά για τις μετρήσεις ζωτικών σημείων του τελευταίου μήνα. Το μήνυμα κειμένου της αναφοράς συγγράφεται, καταγράφεται και παρουσιάζεται [22] μόνο μετά από έλεγχο από την εφαρμογή, ότι έχει περάσει τουλάχιστον ένας μήνας από την τελευταία αναφορά.

4.2.17 Βασικές υπηρεσίες της εφαρμογής

Πέρα από τις υπηρεσίες που χρησιμοποιούνται στην εκκίνηση της εφαρμογής, ή στην επιτυχημένη σύνδεση του χρήστη στο λογαριασμό του, και οι οποίες έχουν ήδη αναλυθεί, πρέπει να αναφερθεί και μία ακόμα υπηρεσία αυτού του είδους. Η υπηρεσία αναζήτησης είναι μία υπηρεσία που καλείται μία φορά στην εκκίνηση της εφαρμογής ενώ ο χρήστης έχει παραμείνει συνδεδεμένος στο λογαριασμό του από προηγούμενη χρήση της. Είναι αυτή η υπηρεσία που αναζητάει στην εκκίνηση όλα τα σχετικά δεδομένα από κάθε πιθανή συλλογή της βάσης δεδομένων και τα αποθηκεύει σε τοπικές λίστες για εύκολη πρόσβαση.

Πέρα από αυτό, η υπηρεσία αυτή καλύπτει και διάφορες λειτουργίες που είναι κοινές μεταξύ πολλαπλών οθονών, όπως η καταγραφή στον κατάλογο ενεργειών ή οι μετατροπές, προσθήκες και διαγραφές από το ιστορικό του χρήστη.

Τελευταία και άκρως απαραίτητη υπηρεσία που χρησιμοποιείται από την εφαρμογή είναι η υπηρεσία κρυπτογράφησης/αποκρυπτογράφησης. Χρησιμοποιώντας κώδικα AES-256 με ένα αποθηκευμένο κλειδί, η υπηρεσία αυτή είναι υπεύθυνη για την κρυπτογράφηση όλων των δεδομένων που το απαιτούν, κυρίως στην εισαγωγή και τροποποίηση δεδομένων στη βάση δεδομένων, αλλά και στην αποθήκευση τιμών στις κοινές προτιμήσεις της εφαρμογής [8]. Ταυτόχρονα, καλείται όποτε χρειάζεται, για να αποκρυπτογραφήσει δεδομένα από τις δύο αυτές πηγές, ειδικά σε περιπτώσεις όπως στην προσωρινή

τοπική αποθήκευση των δεδομένων από τη βάση, καθώς είναι απαραίτητα για την ορθή και γρήγορη παρουσίασή τους με τρόπο που γίνεται εύκολα κατανοητός από το χρήστη.

4.3 Σχεδιασμός Βάσης Δεδομένων

4.3.1 Συλλογές περιορισμένης πρόσβασης

Η βάση δεδομένων στο Firebase [5] αποτελείται από ένα σύνολο συλλογών, για τα διάφορα μοντέλα δεδομένων, που χρησιμοποιούνται στην εφαρμογή, καθώς και μερικά που δεν είναι ορατά στο χρήστη ή δεν υλοποιούνται ποτέ στην εφαρμογή και υπάρχουν μόνο για τη βάση δεδομένων.

Ξεκινώντας με την πρώτη συλλογή στην οποία θα εισάγει δεδομένα ο χρήστης, η συλλογή χρηστών περιέχει όλα τα δεδομένα από τα προφίλ των χρηστών. Τα πεδία ενός εγγράφου αυτής της συλλογής είναι το όνομα του χρήστη, η ηλικία του, το email του, η ταυτότητα εγγράφου, καθώς και η ταυτότητα χρήστη του. Οι δύο αυτές ταυτότητες χρησιμοποιούνται ως μοναδικά αναγνωριστικά για τις υπόλοιπες λειτουργίες της εφαρμογής. Η χρήση αυτή της ταυτότητας του χρήστη είναι κοινό κομμάτι κάθε συλλογής, πέρα από τη συλλογή καταλόγου ενεργειών χρήστη.

Στο θέμα αυτής της συλλογής, η συλλογή καταλόγου είναι αυτή στην οποία προβλέπεται να συμβαίνουν οι περισσότερες εισαγωγές από όλες τις συλλογές της βάσης δεδομένων, καθώς καταγράφει κάθε ενέργεια του χρήστη που την επηρεάζει. Η συλλογή αυτή δεν είναι σχεδιασμένη να είναι άμεσα διαθέσιμη στο χρήστη και δεν λαμβάνει αιτήσεις πρόσβασης, παρά μόνο αυτόματα συγγεγραμμένες αιτήσεις για την εισαγωγή δεδομένων ή, στην περίπτωση που ο χρήστης αποφασίσει να διαγράψει το λογαριασμό του, μία αίτηση από αυτόν για τη διαγραφή των δεδομένων που τον αναφέρουν και προέρχονται από αυτόν. Τα πεδία ενός εγγράφου της είναι απλά η ημερομηνία και η ώρα της καταγεγραμμένης ενέργειας, καθώς και το είδος της ενέργειας και το email του χρήστη από το οποίο προήλθε.

Μια άλλη συλλογή στην οποία ο χρήστης έχει περιορισμένη πρόσβαση είναι η συλλογή μηνιαίων αναφορών. Συγκεκριμένα, στον χρήστη δεν παρουσιάζονται σε κανένα σημείο αναφορές πέρα από την πιο πρόσφατη και η αναζήτηση για αυτή, όπως και η εκπόνηση καινούργιας αναφοράς, όταν είναι απαραίτητο, γίνονται αυτόματα και χωρίς την παρέμβαση του χρήστη. Η κάθε αναφορά αποτελείται από την ταυτότητα χρήστη του χρήστη, την ημερομηνία της αναφοράς σε κείμενο, την ημερομηνία δημιουργίας της αναφοράς σε μορφή Timestamp [7] και το περιεχόμενο της αναφοράς.

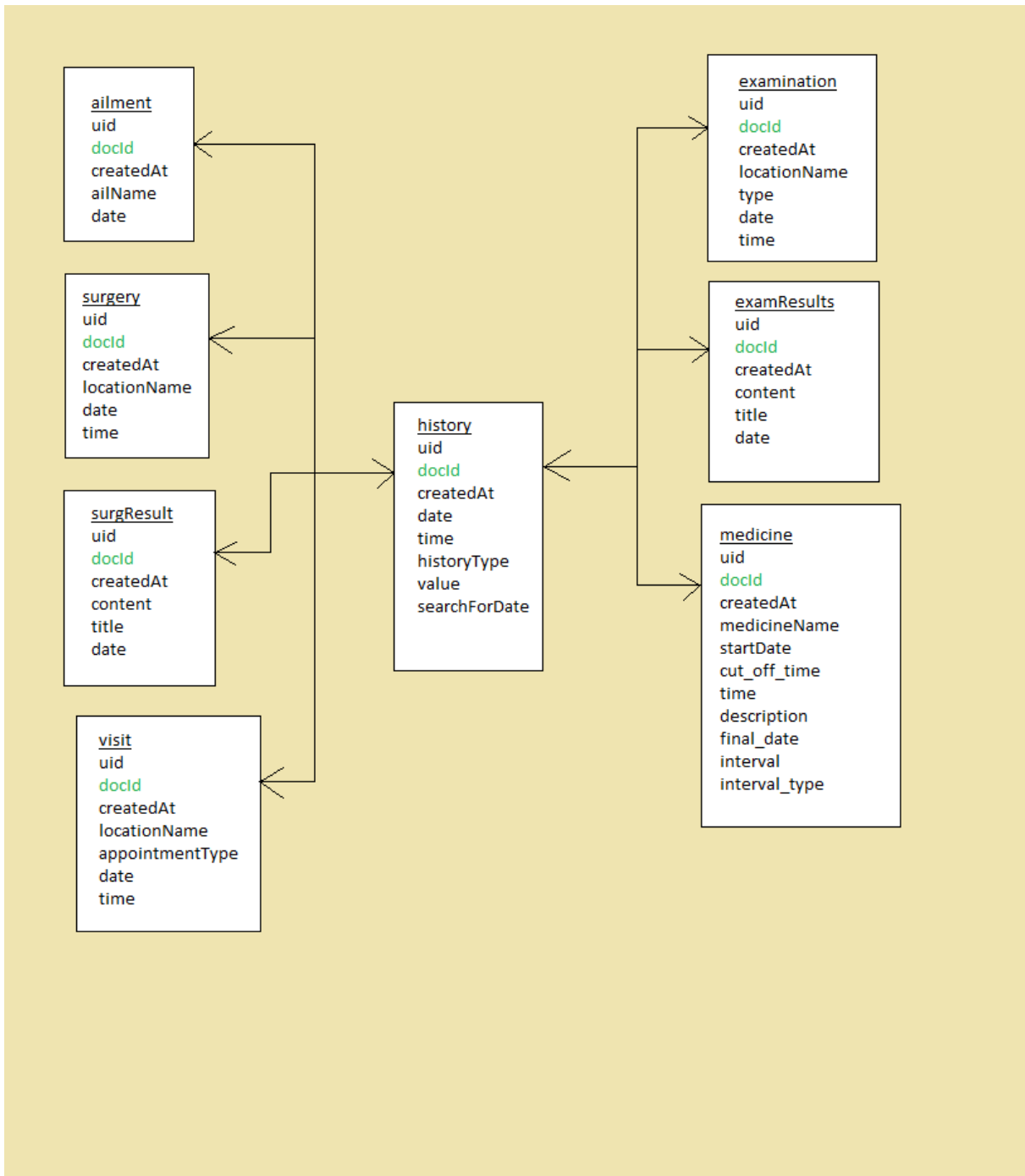
Τελευταία από τις συλλογές σε αυτή την κατηγορία είναι η συλλογή ιστορικού. Η συλλογή αυτή δέχεται καινούργιες εισαγωγές αυτόματα, όταν ο χρήστης δημιουργεί νέο έγγραφο στις συλλογές παθήσεων, εξετάσεων, εγχειρήσεων, επισκέψεων, φαρμάκων, αποτελεσμάτων εξετάσεων ή αποτελεσμάτων εγχειρήσεων. Περιέχει πεδία για την ταυτότητα χρήστη, την ταυτότητα εγγράφου, την ημερομηνία δημιουργίας, το είδος αντικειμένου που δημιουργήθηκε, την τιμή του, καθώς και την ημερομηνία παρατήρησης ή καταγραφής του αντικειμένου. Η ταυτότητα εγγράφου για κάθε αντικείμενο σε αυτήν την συλλογή είναι ίδια με αυτήν στο έγγραφο στο οποίο αντιστοιχεί.

Καθώς δεν είναι σίγουρο ότι το αντικείμενο, που προκάλεσε τη δημιουργία του νέου εγγράφου, θα έχει πεδίο ημερομηνίας, το αντίστοιχο πεδίο ημερομηνίας για τη συλλογή ιστορικού μπορεί να λάβει ως τιμή την τρέχουσα ημερομηνία, μιας και είναι απαραίτητο πεδίο του μοντέλου δεδομένων, που σημαίνει ότι μπορεί να καταλήξει άχρηστο για τη σωστή αναζήτηση αντικειμένων στη συλλογή και στην προβολή ή αλλαγή αυτών. Επομένως, ένα αυτόματα συμπληρωμένο πεδίο της συλλογής, που πάντα υπάρχει, δηλώνει εάν το έγγραφο που διαβάζεται έχει ορθή ημερομηνία ή αν πρέπει να θεωρηθεί

αμελητέο. Η τιμή αυτή εξαρτάται από τον τύπο του αντικειμένου και από τα πεδία που συμπληρώθηκαν σε εκείνο και περάστηκαν στην συνάρτηση δημιουργίας εγγράφου ιστορικού.

Τα έγγραφα της συλλογής μπορεί να έχουν και μερικά προαιρετικά πεδία, ανάλογα με το είδος του αντικειμένου στο οποίο αναφέρονται. Τα πεδία αυτά είναι το πεδίο ώρας και το επιπλέον πεδίο. Το πεδίο ώρας αφορά την καταγεγραμμένη ώρα του αναφερόμενου αντικειμένου και καταγράφεται μόνο αν το αντικείμενο είναι γνωστό ότι περιέχει ένα αντίστοιχο πεδίο και είναι σίγουρο ότι αυτό είναι συμπληρωμένο.

Παρομοίως, το επιπλέον πεδίο δεν υπάρχει, όταν το έγγραφο αναφέρεται σε αντικείμενα χωρίς ένα επιπλέον πεδίο, που να το απαιτεί. Το επιπλέον πεδίο είναι ευέλικτο, καθώς περιέχει την «επιπλέον» τιμή που μπορεί να περιέχει ένα αντικείμενο και χρειάζεται να καλύπτει διάφορες πιθανές τιμές, ανάλογα με το αν το αντικείμενο είναι τύπου φαρμάκου, εξέτασης, επίσκεψης ή αποτελεσμάτων εγχείρησης ή εξέτασης. Για λόγους ευκολίας και ευελιξίας, το μοντέλο δεδομένων, που χρησιμοποιεί η εφαρμογή, έχει τις περισσότερες τιμές του ως προαιρετικές, πέρα από την ταυτότητα χρήστη, την ημερομηνία δημιουργίας, το είδος αντικειμένου και την ημερομηνία καταγραφής.



Σχήμα 4.6: Διάγραμμα Σχέσεων Ιστορικού

4.3.2 Συλλογές πλήρους πρόσβασης

Εκτός από τις συλλογές περιορισμένης πρόσβασης, υπάρχουν και άλλες, που αντιστοιχούν σε τοπικά μοντέλα δεδομένων και το περιεχόμενό τους είναι διαθέσιμο και στο χρήστη. Κυρίως σχετίζονται με τα διάφορα είδη αντικειμένων, που μπορεί να δημιουργήσει ο χρήστης, αλλά υπάρχουν μερικές σημαντικές λεπτομέρειες στις οποίες οι συλλογές διαφέρουν από τα τοπικά μοντέλα δεδομένων και η εφαρμογή δρα ως μεταφραστής μεταξύ αυτών των δύο.

Μια ιδιαίτερα σημαντική διαφορά είναι ότι, για λόγους ευκολίας στην επιλογή ημερομηνίας, τα μοντέλα δεδομένων, που απαιτούν επιλογή ημερομηνίας από το χρήστη, χρησιμοποιούν τιμές `DateTime` για τα πεδία επιλεγμένης ημερομηνίας, ενώ τα αντίστοιχα πεδία στις συλλογές είναι τύπου κειμένου για ευκολία προβολής τους. Η μετατροπή των τιμών των πεδίων από κείμενο σε `DateTime` και αντίστροφα γίνεται με τις διαμορφώσεις που παρέχονται από το πακέτο «intl» [30].

Η συλλογή με την πιο απλή δομή είναι η συλλογή παθήσεων, η οποία αποτελείται απλά από την ταυτότητα χρήστη του χρήστη, την ταυτότητα εγγράφου, το όνομα της παθήσεως, την ημερομηνία παρατήρησης αυτής και την ημερομηνία δημιουργίας του εγγράφου. Η ημερομηνία δημιουργίας συμπληρώνεται αυτόματα από την εφαρμογή πάντα και αποθηκεύεται στη συλλογή με τη μορφή `Timestamp` [7].

Σε αυτή την περίπτωση, το μοντέλο δεδομένων για τις παθήσεις στην εφαρμογή είναι σχεδόν ίδιο, με εξαίρεση των τιμών ημερομηνίας, που είναι και οι δύο αποθηκευμένες ως τύπου `DateTime`. Η διαφορά αυτή βρίσκεται και σε κάθε άλλο μοντέλο που περιέχει αυτά τα πεδία και δε θα αναφερθεί ξανά.

Επόμενη είναι η συλλογή πληροφοριών εκτάκτου ανάγκης, που αποτελείται από την ταυτότητα χρήστη, την ομάδα αίματός του, μια λίστα με σημειωμένες αλλεργίες, μία λίστα φαρμάκων, που παίρνει ο χρήστης και ένα χάρτη από ονόματα και σειρά αριθμών, που μαζί δρουν ως επαφές εκτάκτου ανάγκης. Οι τιμές του χάρτη είναι όλες τύπου κειμένου, μιας και υπάρχουν μόνο για να προβάλλονται και δεν υπάρχει λόγος να είναι καθαροί αριθμοί.

Τα πεδία αυτά είναι όλα προαιρετικά και ο χρήστης μπορεί να συμπληρώσει οποιαδήποτε επιθυμεί και να αφήσει τα υπόλοιπα κενά. Η συλλογή είναι προορισμένη να περιέχει μόνο ένα έγγραφο από κάθε χρήστη, οπότε ο χρήστης μόνο τροποποιεί ή διαγράφει το έγγραφό του. Ως συνέπεια, δεν υπάρχει πεδίο ημερομηνίας δημιουργίας, καθώς δε θεωρείται απαραίτητο.

Προχωρώντας σε μια συλλογή με μια πιο περίπλοκη δομή, η συλλογή φαρμάκων περιέχει πιθανότατα τα περισσότερα πεδία και σίγουρα τα πιο διασυνδεδεμένα από όλες τις συλλογές. Πέρα από την ταυτότητα χρήστη, την ταυτότητα εγγράφου, και την ημερομηνία δημιουργίας εγγράφου, περιέχει το όνομα του φαρμάκου, μια περιγραφή των οδηγιών χρήσης του, την ημερομηνία αρχής φαρμακευτικής αγωγής, την ημερομηνία λήξης φαρμακευτικής αγωγής, την ώρα αρχής ειδοποίησης για τη λήψη του φαρμάκου, την ώρα πέρα από την οποία δεν είναι επιθυμητές περισσότερες ειδοποιήσεις, το διάστημα μεταξύ ειδοποιήσεων, το είδος του διαστήματος (μεταξύ ημερών ή ωρών), καθώς και το αν ο χρήστης επιθυμεί ειδοποιήσεις ή όχι.

Με εξαίρεση την ώρα λήξης ειδοποιήσεων, την περιγραφή, το όνομα του φαρμάκου, την ημερομηνία δημιουργίας και το αν είναι επιθυμητές οι ειδοποιήσεις, τα υπόλοιπα πεδία είναι προαιρετικά. Επιπλέον, η ώρα λήξης μπορεί να οριστεί και αυτόματα αν ο χρήστης δεν ορίσει ο ίδιος τιμή, ενώ η περιγραφή μπορεί να παραμείνει χωρίς κείμενο, οπότε πρακτικά τα μόνα πεδία που απαιτούν ορισμό από το χρήστη είναι το όνομα του φαρμάκου και το αν επιθυμεί ειδοποιήσεις για την έγκαιρη λήψη του.

Οι συγκεκριμένες ώρες στις οποίες θα σταλθούν οι ειδοποιήσεις υπολογίζονται ανά αντικείμενο από την εφαρμογή, με βάση τα διάφορα πεδία. Η λογική που ακολουθεί η εφαρμογή για αυτό περιγράφεται στην υπό-ενότητα 4.2.16.

Μερικές συλλογές είναι παρόμοιες, με αποτέλεσμα τα μοντέλα τους να είναι σχεδόν ίδια. Σε αυτή την κατηγορία ανήκουν οι συλλογές εξετάσεων, εγχειρήσεων και επισκέψεων. Και οι τρεις αυτές συλλογές προορίζονται για την καταγραφή ραντεβού, με τις συλλογές εξετάσεων και εγχειρήσεων να έχουν συγκεκριμένο σκοπό, ενώ η συλλογή επισκέψεων είναι για κάθε άλλο είδος ραντεβού, που μπορεί να θέλει να καταγράψει ο χρήστης, αλλά δεν προβλέπεται από την εφαρμογή.

Κοινά πεδία και των τριών συλλογών είναι η τοποθεσία του ραντεβού, η ημερομηνία, η ώρα και, ως συνήθως, η ταυτότητα χρήστη, η ταυτότητα εγγράφου, και η ημερομηνία δημιουργίας εγγράφου. Εκτός από αυτά, οι συλλογές επισκέψεων και εξετάσεων έχουν η κάθε μία ένα επιπλέον πεδίο, το πεδίο τύπου, αν και οι τιμές που παίρνει το πεδίο διαφέρουν μεταξύ τους. Για τη συλλογή εξετάσεων, το πεδίο παίρνει μόνο δύο τιμές, ανάλογα με το αν η εξέταση είναι εργαστηριακή ή απεικονιστική. Στη συλλογή επισκέψεων, από την άλλη, μπορεί να πάρει ό,τι τιμή επιθυμεί ο χρήστης, καθώς η συλλογή είναι σχεδιασμένη για να καλύπτει κάθε άλλο είδος επίσκεψης.

Ως προς τα μοντέλα δεδομένων που αντιστοιχούν σε αυτές τις συλλογές, οι διαφορές τους σε σχέση με τις συλλογές είναι μικρές. Πέρα από τις ημερομηνίες, οι ώρες που καταγράφονται ορίζονται ως τύπου TimeOfDay στα μοντέλα δεδομένων της εφαρμογής, ενώ μετατρέπεται σε κείμενο για τις συλλογές. Επιπλέον, για το μοντέλο ραντεβού εξετάσεων, το πεδίο τύπου εξέτασης είναι μια ειδικά ορισμένη κλάση αντικειμένου, αντί να είναι κείμενο, ενώ στη συλλογή αποθηκεύεται απλά ένα κείμενο, με βάση την τιμή τύπου του αντικειμένου.

Μια παρόμοια κατάσταση παρουσιάζεται και με τις συλλογές στόχων και μετρήσεων ζωτικών σημείων. Στην περίπτωση αυτή, όμως, οι δύο έχουν αρκετές ομοιότητες ώστε, αν και αποθηκεύονται στη βάση σε διαφορετικές συλλογές, να μοιράζονται το ίδιο μοντέλο δεδομένων. Συγκεκριμένα, η συλλογή μετρήσεων ζωτικών σημείων καταγράφει την ταυτότητα χρήστη, την ταυτότητα εγγράφου, την ημερομηνία δημιουργίας εγγράφου, την ημερομηνία μέτρησης των συμπληρωμένων στοιχείων, τον ρυθμό παλμού του χρήστη, τη θερμοκρασία του, την πίεση και το σάκχαρό του, ενώ η συλλογή στόχων περιέχει τα ίδια πεδία εκτός από τη θερμοκρασία, την ημερομηνία μέτρησης και, μια και είναι σχεδιασμένη ώστε να περιέχει μόνο ένα έγγραφο ανά χρήστη, όπως και η συλλογή πληροφοριών εκτάκτου ανάγκης, την ημερομηνία δημιουργίας.

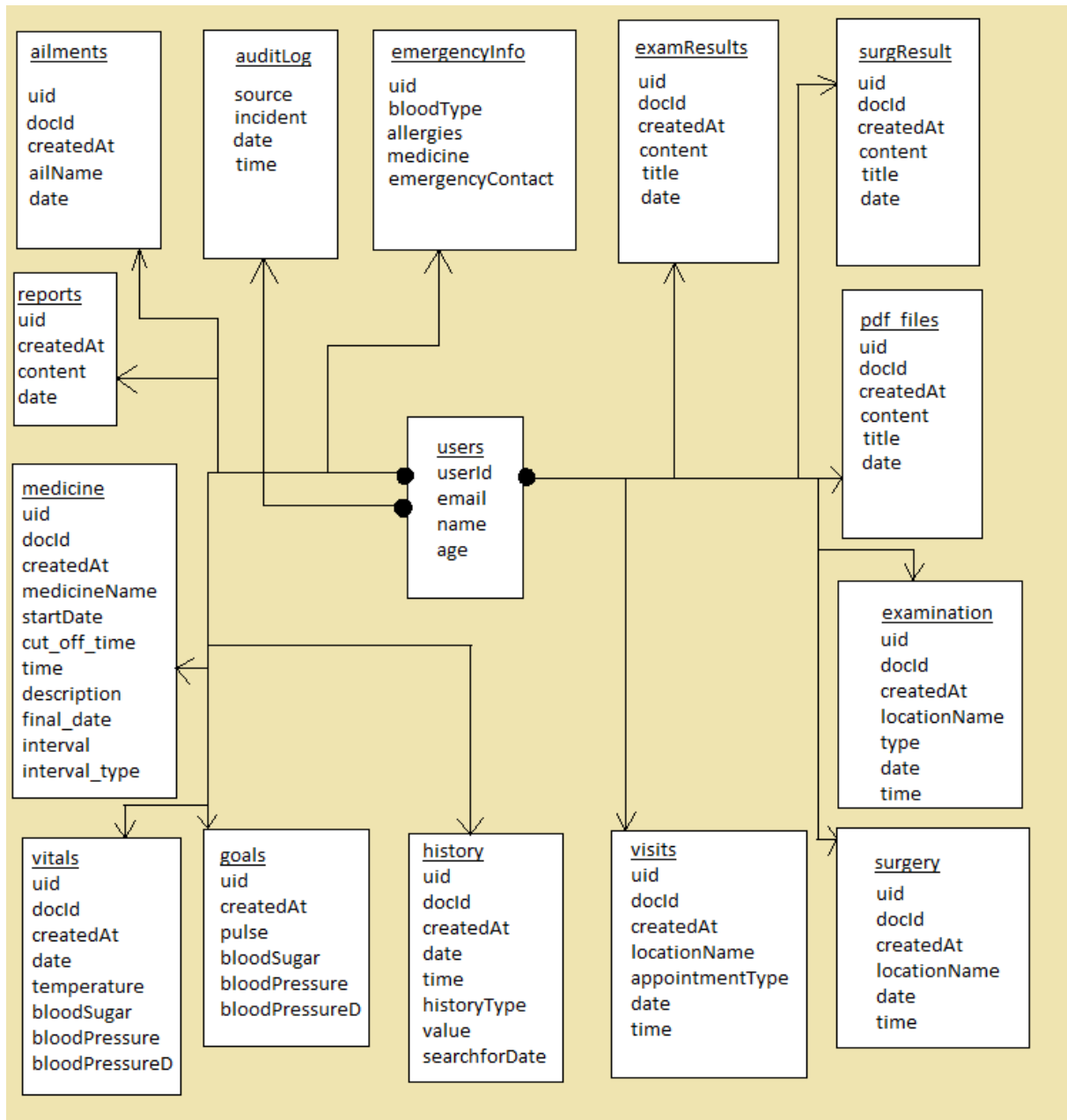
Καθώς όλα τα πεδία, τα οποία ο χρήστης συμπληρώνει ο ίδιος, είναι προαιρετικά, το μοντέλο δεδομένων της εφαρμογής πρέπει να είναι ευέλικτο, ώστε να μπορεί να δημιουργεί αντικείμενα του κατάλληλου τύπου με επιτυχία. Έτσι, μια και οι δύο συλλογές μοιράζονται τόσα πολλά προαιρετικά κοινά πεδία, αρκεί ένα μοντέλο δεδομένων με δύο διαφορετικές μεθόδους μετάφρασης από το Firestore [5] σε αντικείμενο του κατάλληλου τύπου στην εφαρμογή.

Το μοντέλο αυτό καλύπτει όλα τα πεδία της συλλογής μετρήσεων ζωτικών σημείων που αναφέρθηκαν και αρκεί η εφαρμογή να μην προσπαθεί να συμπληρώσει το πεδίο θερμοκρασίας, όταν πρόκειται για τους στόχους του χρήστη, που δεν προκαλεί πρόβλημα μιας και όλα τα πεδία μετρήσεων είναι προαιρετικά. Για την ημερομηνία μέτρησης, η οποία είναι ένα από τα λίγα απαραίτητα πεδία του μοντέλου, απλά συμπληρώνεται η τρέχουσα ημερομηνία, όταν πρόκειται για τη δημιουργία αντικειμένου στη λίστα στόχων.

Τελευταίες είναι οι συλλογές αποτελεσμάτων εξετάσεων και εγχειρήσεων. Οι δύο αυτές συλλογές έχουν την ίδια ακριβώς δομή, με τα πεδία τους να είναι μόνο η ταυτότητα χρήστη, η ταυτότητα εγγράφου, η ημερομηνία δημιουργίας, ο τίτλος και το περιεχόμενο του εγγράφου. Οι δύο αυτές συλλογές αντιστοιχούν σε ένα κοινό μοντέλο δεδομένων, το μοντέλο αποτελεσμάτων.

Πέρα από την αναφερόμενη διαφορά μεταξύ συλλογών και μοντέλων δεδομένων, ως προς το θέμα των ημερομηνιών, το μοντέλο αποτελεσμάτων έχει μία επιπλέον διαφορά με την αντίστοιχη συλλογή. Περιέχει ένα ακόμα πεδίο, το πεδίο τύπου αποτελεσμάτων. Το πεδίο αυτό δέχεται τιμές ειδικού τύπου, που καθορίζονται με βάση τη συλλογή, από την οποία συλλέχθηκε το έγγραφο, από το οποίο δημιουργείται το αντικείμενο. Αν το έγγραφο προέρχεται από τη συλλογή αποτελεσμάτων εγχειρήσεων, η τιμή θα είναι «Τύπος Εγχειρήσεων», ενώ αν προέρχεται από τη συλλογή αποτελεσμάτων εξετάσεων,

το πεδίο θα πάρει τιμή «Τύπος Εξετάσεων». Με βάση την τιμή αυτού του πεδίου, το αντικείμενο θα αποθηκευτεί στην τοπική λίστα αποτελεσμάτων εξετάσεων, ή στη λίστα αποτελεσμάτων εγχειρήσεων.



Σχήμα 4.7: Διάγραμμα Βάσης Δεδομένων

4.4 Επίλογος

Ο σχεδιασμός της δομής της εφαρμογής χρησιμοποιεί την κεντρική οθόνη ως το πλαίσιο, στο οποίο προβάλλονται όλες οι υπόλοιπες οθόνες. Ανάλογα με την κατάσταση των διάφορων τιμών που επιβλέπει, καθώς και τις πράξεις του χρήστη, η ενεργή οθόνη μπορεί να αλλάξει πολλαπλές φορές χωρίς πρόβλημα.

Η κύρια λειτουργία της εφαρμογής είναι να παρέχει στο χρήστη την ικανότητα να εισάγει, να βλέπει, να τροποποιεί και να διαγράφει ιατρικά δεδομένα με εύκολο τρόπο. Η προβολή των δεδομένων,

ανάλογα με την κατηγορία τους και τις επιθυμίες του χρήστη, μπορεί να είναι σε μορφή λίστας, ως ημερομηνίες σε ένα ημερολόγιο [20], ή και ως γραφήματα [21].

Επιπλέον, ο χρήστης μπορεί να λαμβάνει ειδοποιήσεις για τα ραντεβού που έχει κανονίσει, για την έγκαιρη λήψη φαρμάκων που έχει σημειώσει, καθώς και να δει μια μηνιαία αναφορά που συνοψίζει το ιστορικό που κατέγραψε για τον τελευταίο μήνα [22].

Κατά το σχεδιασμό της εφαρμογής γίνεται προσπάθεια να παρέχεται κάλυψη για τις περισσότερες ανάγκες του χρήστη, σχετικά με το θέμα, όσο είναι εφικτό, παρέχοντας ακόμα και λειτουργίες συγχρονισμού με λογαριασμό Google Health, εφόσον ο χρήστης διαθέτει τέτοιο λογαριασμό. Ταυτόχρονα, γίνεται προσπάθεια πρόβλεψης πιθανών αναγκών, που μπορεί να προκύψουν και ανάλογης διαμόρφωσης των λειτουργιών της εφαρμογής, με στόχο την ευκολία του χρήστη και την αποφυγή της αναγκαστικής εισαγωγής δεδομένων, που δεν επιθυμεί ο χρήστης, όσο περισσότερο είναι δυνατό.

Κεφάλαιο 5ο: Υλοποίηση Εφαρμογής

5.1 Εισαγωγή

Στόχος της εφαρμογής είναι να παρέχει πολλές διαφορετικές υπηρεσίες, που αφορούν τα ιατρικά δεδομένα του χρήστη. Πέρα από την απλή διαχείριση και αποθήκευση των δεδομένων, περιλαμβάνει, ανάμεσα σε άλλα, την οπτικοποίηση τους, την παρουσίαση των ραντεβού του χρήστη σε μορφή σημειωμένων ημερών σε ημερολόγιο, καθώς και τη δημιουργία μηνιαίας αναφοράς για το χρήστη. Η υλοποίηση των υπηρεσιών αυτών απαιτεί όχι μόνο λεπτομερή σχεδιασμό, αλλά και χρήση τεχνολογιών, που συνεργάζονται εύκολα μεταξύ τους και ένα ξεκάθαρο σύνολο περιορισμών και κανόνων, για να δουλεύει σωστά η λογική του συστήματος.

5.2 Τομείς υλοποίησης

5.2.1 Περιβάλλον Ανάπτυξης

Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκε κυρίως το Visual Studio Code (VSCode) [31] και γράφτηκε με τη γλώσσα προγραμματισμού Dart. Για μεγαλύτερη ευκολία στη σωστή διατύπωση εντολών και τον ταχύτερο εντοπισμό σφαλμάτων, χρησιμοποιήθηκε η επέκταση Dart για το VSC [32], η οποία του παρέχει δυνατότητες υποστήριξης για τη γλώσσα. Επιπλέον, έγινε χρήση και της επέκτασης Flutter [33], η οποία παρέχει εργαλεία για την οργάνωση του κώδικα και τη γρήγορη αλλαγή του.

Μεγάλο ρόλο στην δοκιμή των διάφορων λειτουργιών της εφαρμογής έπαιξε το Android Studio [34]. Με τη χρήση αυτού, δημιουργήθηκε μία ψηφιακή συσκευή [34], στην οποία μπορούσε να δοκιμαστεί γρήγορα η εφαρμογή.

Καθώς η εφαρμογή χρησιμοποιεί το Firebase [12] για την λειτουργία πιστοποίησης χρήστη και για την αποθήκευση δεδομένων στην απομακρυσμένη βάση δεδομένων του [5], ήταν απαραίτητη επίσης η εγκατάσταση του πακέτου του Firebase [19].

5.2.2 Υλοποίηση μηχανισμού πιστοποίησης χρήστη

Η πιστοποίηση χρήστη βασίζεται κυρίως στους μηχανισμούς πιστοποίησης του Firebase [12]. Η διαδικασία υλοποίησης του μηχανισμού πιστοποίησης είναι απλή, απαιτώντας μόνο την ενεργοποίηση της αντίστοιχης λειτουργίας, για την εργασία στην ιστοσελίδα του και τη δήλωση της χρήσης λογαριασμών Google και συνδυασμών email και κωδικών πρόσβασης, ως τους αποδεκτούς παρόχους στοιχείων, για την εγγραφή ενός χρήστη, επιτρέποντάς του μετά να συνδεθεί στην εφαρμογή.

Από τη στιγμή που ο χρήστης δημιουργήσει λογαριασμό στην εφαρμογή, μέχρι να δηλώσει ότι επιθυμεί να διαγράψει το λογαριασμό του, παραμένει αποθηκευμένος στον πίνακα καταγεγραμμένων χρηστών και μπορεί να πιστοποιηθεί παραδίδοντας τα στοιχεία σύνδεσής του στην εφαρμογή.

Από την πλευρά της εφαρμογής, υπάρχουν δύο διαφορετικοί τρόποι, με τους οποίους ο χρήστης μπορεί να πιστοποιηθεί. Ο πρώτος, που αφορά καθαρά τη σύνδεσή του, είναι η ορθή καταχώρηση των πληροφοριών σύνδεσης, με τις οποίες γράφτηκε στην εφαρμογή. Αυτή είναι η πιο ξεκάθαρη μέθοδος και πιστοποιεί το χρήστη στο Firebase [6][12].

Για την υλοποίηση αυτής της μεθόδου με ασφάλεια, η εφαρμογή χρησιμοποιεί ένα Widget SignInScreen [13] με δηλωμένους παρόχους το Google και τη χρήση πιστοποιημένου email [14]. Η φόρμα που δημιουργείται επιτρέπει στο χρήστη να εισάγει τα στοιχεία του, χωρίς φόβο για πιθανή κλοπή τους. Στο θέμα της ασφάλειας, η φόρμα αυτή είναι ιδιαίτερα αποτελεσματική σε αυτό, καθώς ούτε η ίδια η εφαρμογή δεν έχει πρόσβαση στα πιο κρίσιμα στοιχεία που εισάγονται σε αυτήν, όπως τον κωδικό σύνδεσης του χρήστη, οπότε ο κίνδυνος ότι τα στοιχεία μπορεί να κλαπούν από αυτή από την πλευρά της συσκευής είναι ελάχιστος.

Η δεύτερη μέθοδος πιστοποίησης της εφαρμογής, από την άλλη, είναι καθαρά τοπική. Αυτό σημαίνει ότι αντί να πιστοποιεί ότι ο χρήστης είναι εγγεγραμμένος στην εφαρμογή και ότι είναι αυτός που ισχυρίζεται μέσω των στοιχείων σύνδεσής του, η δεύτερη μέθοδος βεβαιώνεται ότι το άτομο που προσπαθεί να μπει στην εφαρμογή είναι το ίδιο άτομο που είναι ο κάτοχος της συσκευής, όπου είναι εγκατεστημένη η εφαρμογή, μέσω βιομετρικών μετρήσεων. Η μέθοδος αυτή είναι προαιρετική και μπορεί να ενεργοποιηθεί ή να απενεργοποιηθεί από το χρήστη όποτε θέλει, με την τυπική της τιμή να την έχει απενεργοποιημένη. Επιπλέον, η επιλογή για τη χρήση αυτού του ελέγχου φανερώνεται μόνο σε συσκευές που έχουν τη δυνατότητα να υποστηρίξουν αυτή τη λειτουργία και μόνο αν διαθέτουν ήδη βιομετρικές μετρήσεις στη συσκευή [16].

Σε περίπτωση που ο χρήστης επιθυμεί, ή χρειάζεται μια εναλλακτική στη δεύτερη μέθοδο πιστοποίησης, η εφαρμογή παρέχει και την επιλογή για τον ορισμό ενός PIN. Ο κωδικός αυτός θα είναι αποθηκευμένος μόνο τοπικά για εκείνη τη συσκευή [8], δηλαδή θα χρειαστεί να οριστεί ξανά σε περίπτωση που ο χρήστης αλλάξει συσκευή, αν αδειάσει τις μοιραζόμενες προτιμήσεις της εφαρμογής, ή αν διαγράψει και εγκαταστήσει ξανά την εφαρμογή. Με τη χρήση του κωδικού PIN, ο χρήστης μπορεί να προσπεράσει το βιομετρικό έλεγχο, που έχει ενεργοποιήσει, ενώ ταυτόχρονα πιστοποιείται από τη γνώση του PIN. Η επιλογή για χρήση PIN παρέχεται ανεξάρτητα από το αν η συσκευή υποστηρίζει τη χρήση πιστοποίησης μέσω βιομετρικών μετρήσεων.

5.2.3 Υλοποίηση διαχείρισης ιατρικών δεδομένων

Από την φύση της, η εφαρμογή υλοποιεί ένα σύστημα που αποθηκεύει και διαχειρίζεται ιατρικά δεδομένα, που καταθέτει ο χρήστης και τα προβάλλει με διάφορους τρόπους, όπως αυτός επιθυμεί. Η κεντρική αυτή λειτουργία σημαίνει ότι η εφαρμογή αποκτά πρόσβαση σε μερικά ιδιαίτερα εμπιστευτικά δεδομένα του χρήστη, όλα συγκεντρωμένα σε μία βάση. Η φύση αυτή της εφαρμογής γίνεται γραπτά γνωστή στο χρήστη, πριν το πρώτο άνοιγμά της και απαιτεί τη συναίνεσή του για τη συνέχεια της δράσης της. Επιπλέον, τα δεδομένα που αποθηκεύει είναι μόνο αυτά τα οποία καταθέτει ο ίδιος ο χρήστης, δίνοντάς του όλη την αρμοδιότητα ως προς το είδος και το πλήθος των δεδομένων, που θα αποθηκευτούν στη βάση, καθώς και το πότε θα διαγραφτούν.

Με την επιτυχή είσοδό του στο λογαριασμό του ή, σε περίπτωση εκκίνησης της εφαρμογής ενώ ο χρήστης είναι ήδη συνδεδεμένος, εκτελείται αναζήτηση όλων των σχετικών δεδομένων του χρήστη από τη βάση [5], τα οποία αποθηκεύονται προσωρινά σε τοπικές λίστες, για όσο διάστημα η εφαρμογή παραμένει ανοιχτή. Τα δεδομένα αυτά δεν αποθηκεύονται μόνιμα στην εφαρμογή, καθώς αυτό θα μπορούσε να παρουσιάσει μεγαλύτερο ρίσκο κλοπής τους.

Εξάιρεση στην αναζήτηση αυτή αποτελούν συλλογές στις οποίες δεν πρέπει να υπάρχει άμεση πρόσβαση, όπως οι συλλογές μηνιαίων αναφορών και ο κατάλογος ενεργειών του χρήστη, ή ο πίνακας με το προφίλ του χρήστη, που υπάρχει κυρίως για λόγους καταχώρησης δεδομένων, που αφορούν την ταυτότητα του χρήστη στο σύστημα και δεν βρίσκεται λόγος να αποθηκευτεί τοπικά, ακόμα και προσωρινά.

Ως προς τη μέθοδο προσωρινής αποθήκευσης των δεδομένων στην εφαρμογή, τα έγγραφα από κάθε συλλογή χρησιμοποιούνται για τη δημιουργία αντικειμένων για διάφορες λίστες, χρησιμοποιώντας το αντίστοιχο μοντέλο δεδομένων για την κάθε συλλογή. Έτσι, τα αντικείμενα τύπου «Παθήσεων», για παράδειγμα, θα αποθηκευτούν σε μία λίστα αντικειμένων «Παθήσεων», με τη λίστα να είναι αποθηκευμένη σε ένα αρχείο στην εφαρμογή, το οποίο είναι προορισμένο μόνο για δεδομένα αυτού του τύπου. Τα έγγραφα από τις υπόλοιπες συλλογές αντιμετωπίζονται με παρόμοιο τρόπο.

Για συλλογές που μοιράζονται ένα μοντέλο δεδομένων με μικρές παραλλαγές στην εφαρμογή του, όπως οι συλλογές στόχων και μετρήσεων ζωτικών σημείων, τα έγγραφα της κάθε συλλογής αποθηκεύονται σε διαφορετικές λίστες του ίδιου αντικειμένου, οι οποίες με τη σειρά τους είναι αποθηκευμένες στο ίδιο αρχείο δεδομένων. Η μέθοδος αυτή εφαρμόζεται και στις συλλογές αποτελεσμάτων εξετάσεων και αποτελεσμάτων εγχειρήσεων, που μοιράζονται το ίδιο μοντέλο, αλλά τα αντικείμενα που δημιουργούνται από τα έγγραφά τους παίρνουν διαφορετική τιμή στο πεδίο τύπου αποτελέσματος, ανάλογα με τη συλλογή από την οποία προήλθαν.

Το σκεπτικό αυτό δεν εφαρμόζεται στη συλλογή εξετάσεων, παρόλο που το αντίστοιχο μοντέλο δεδομένων της έχει ομοιότητες με το μοντέλο δεδομένων για τα αποτελέσματα, εφόσον γίνεται διάκριση μεταξύ δύο ειδών εξετάσεων. Η επιλογή αυτή στο σχεδιασμό γίνεται με βάση το γεγονός ότι τα ραντεβού εξετάσεων αποθηκεύονται στην ίδια συλλογή, με μοναδική διαφορά τον τύπο εξετάσεων στον οποίο αναφέρονται.

Όσον αφορά τη βάση δεδομένων [5], τα δεδομένα που αποθηκεύονται σε αυτή παραμένουν χωρίς αλλαγή. Αυτό σημαίνει ότι ένα έγγραφο που καταχωρήθηκε στη βάση [5] θα παραμείνει σε αυτή, εκτός εάν ο χρήστης που το καταχώρησε προχωρήσει στο να το διαγράψει μέσω της εφαρμογής. Η διαγραφή παλιών και άχρηστων δεδομένων ανατίθεται στο χρήστη, επιτρέποντάς του να οργανώσει τα δεδομένα του με τον τρόπο που θέλει, χωρίς αυτοματοποιημένες διαγραφές παλιών εγγράφων.

Η παραπάνω φιλοσοφία δεν περιλαμβάνει μερικές συλλογές όπου ο χρήστης δεν έχει άμεση πρόσβαση, όπως οι συλλογές μηνιαίας αναφοράς και ο κατάλογος ενεργειών χρήστη. Στην περίπτωση της πρώτης συλλογής, η εφαρμογή δεν αποθηκεύει κάθε αναφορά που αντιστοιχεί στο χρήστη, μόνο την πιο πρόσφατη και μόνο αν δεν την έχει ήδη αποθηκευμένη. Με αυτόν τον τρόπο βεβαιώνεται ότι η συσκευή του χρήστη δεν επιβαρύνεται άσκοπα από παλιές και άχρηστες αναφορές.

Σχετικά με τον κατάλογο ενεργειών, η κεντρική του λειτουργία είναι να καταγράφει τις ενέργειες που εκτελεί ο χρήστης, όταν αυτές επηρεάζουν τη βάση [5], ή όταν αυτός συνδέεται στο λογαριασμό του. Επομένως, ο κατάλογος ενεργειών του χρήστη δεν είναι σχεδιασμένος να είναι διαθέσιμος σε αυτόν και δεν αποθηκεύεται τοπικά. Και στις δύο περιπτώσεις, τα έγγραφα που σχετίζονται με το χρήστη παραμένουν χωρίς αυτόματη διαγραφή, αλλά μπορούν ακόμα να διαγραφτούν από το χρήστη, εάν αυτός προχωρήσει σε διαγραφή του λογαριασμού του. Σε αυτήν την περίπτωση θα διαγραφτούν όλα τα δεδομένα σχετικά με αυτόν από κάθε συλλογή της βάσης.

5.2.4 Υλοποίηση μέτρων ασφαλείας

Καθώς τα ιατρικά δεδομένα που καταθέτει ο χρήστης στη βάση δεδομένων [5] είναι προσωπικά δεδομένα με κρίσιμη σημασία γι' αυτόν, το σύστημα απαιτείται να λαμβάνει μέτρα ασφαλείας στη διαχείρισή τους. Σε αυτόν τον τομέα, την περισσότερη δουλειά την αναλαμβάνει η εφαρμογή.

Από την πλευρά της βάσης δεδομένων [5], τα μέτρα ασφαλείας που εφαρμόζει παραμένουν βασικά. Με τη χρήση του Firebase [6] για την υλοποίησή της βάσης, παρέχεται στο σύστημα ένα έτοιμο μέτρο ασφαλείας με τις υπηρεσίες SSL (Secure Sockets Layer) και TLS (Transport Layer Security) που είναι

διαθέσιμες και αυτόματα ενεργοποιημένες σε κάθε εργασία, που υλοποιείται στο Firebase. Οι υπηρεσίες αυτές κρυπτογραφούν αυτόματα κάθε επικοινωνία προς ή από το Firebase [5]. Ο χρήστης δεν χρειάζεται να κάνει κάποια ενέργεια από την πλευρά του, για να ενεργοποιήσει αυτές τις υπηρεσίες, αφού χρησιμοποιούνται αυτόματα, όποτε η εφαρμογή εκτελεί μια οποιαδήποτε διαδικασία, που θα τη φέρει σε επαφή με τη βάση δεδομένων, ή θα καλέσει κάποια άλλη υπηρεσία του Firebase.

Πέρα από αυτό, όμως, η ίδια η βάση [5] εφαρμόζει μόνο ένα ακόμα μέτρο ασφάλειας, την πιστοποίηση χρήσης, που παρέχεται από το Firebase Authentication [12]. Εφόσον ο χρήστης έχει συνδεθεί και πιστοποιηθεί με επιτυχία, η βάση [5] δεν περιέχει κανόνες, περιορισμούς, ή μέτρα για να περιορίσει την πρόσβασή του στα δεδομένα. Αυτό το καθήκον, ανάμεσα σε άλλα, το αναλαμβάνει η εφαρμογή.

Ένας ιδιαίτερα απλός, αλλά αποτελεσματικός, τρόπος, με τον οποίο περιορίζεται η πρόσβαση σε δεδομένα από την εφαρμογή είναι με την προσκόλληση ενός περιορισμού στο «εύρος» των αιτημάτων αναζήτησης δεδομένων, που στέλνονται στη βάση. Συγκεκριμένα, η εφαρμογή είναι προγραμματισμένη να χρησιμοποιεί μοναδικά αναγνωριστικά του χρήστη, που διαθέτουν οι διάφορες συλλογές της βάσης, για να περιορίσει τα δεδομένα στα οποία έχει πρόσβαση ο χρήστης, αφήνοντάς του μόνο πρόσβαση σε αυτά, που ο ίδιος δημιούργησε.

Αναγνωριστικά που χρησιμοποιούνται για αυτό το σκοπό είναι κυρίως η ταυτότητα χρήστη του χρήστη, με το email του να είναι ένα άλλο αναγνωριστικό, που βλέπει χρήση πιο σπάνια. Με αυτή τη μέθοδο αποφεύγεται ο κίνδυνος ότι ένας χρήστης μπορεί να αποκτήσει πρόσβαση στα προσωπικά και εμπιστευτικά ιατρικά δεδομένα ενός άλλου χρήστη.

Ο κίνδυνος κλοπής δεδομένων από άλλους χρήστες, όμως, δεν είναι ο μόνος κίνδυνος που αντιμετωπίζει η εφαρμογή. Υπάρχει πάντα και η πιθανότητα ότι τα δεδομένα θα υποκλαπούν κατά τη μεταφορά τους, ή ότι η ίδια η βάση θα παραβιαστεί και κάποιος θα αποκτήσει πρόσβαση στα δεδομένα που στάλθηκαν σε αυτή. Η εφαρμογή αντιμετωπίζει αυτό το πρόβλημα με την επιπλέον κρυπτογράφηση όλων των δεδομένων που μεταφέρονται, τα οποία είναι προσωπικής, ή εμπιστευτικής φύσης.

Για αυτό το λόγο χρησιμοποιείται κρυπτογράφηση δεδομένων AES-256. Η μέθοδος αυτή, γνωστή για την ασφάλειά της, κρυπτογραφεί τα προσωπικά δεδομένα του χρήστη, αφήνοντας μόνο την ταυτότητα χρήστη και την ημερομηνία δημιουργίας εγγράφου με την αρχική τους μορφή. Στην περίπτωση της ταυτότητας χρήστη, η έλλειψη περεταίρω κρυπτογράφησης οφείλεται σε ένα συνδυασμό δύο παραγόντων.

Ο πρώτος από αυτούς είναι ότι η ταυτότητα χρήστη, αν και είναι μοναδικό αναγνωριστικό, δεν φανερώνει προσωπικές πληροφορίες του και δεν μπορεί να δράσει ως κλειδί εισβολής στη βάση δεδομένων, καθώς οι προγραμματισμένες μέθοδοι αναζήτησης λαμβάνουν την ταυτότητα χρήστη που χρησιμοποιούν κατευθείαν από την ταυτότητα του πιστοποιημένου χρήστη, που είναι συνδεδεμένος στην εφαρμογή, όχι από κάποιο πεδίο εισαγωγής κειμένου. Ο δεύτερος λόγος είναι η προσπάθεια διευκόλυνσης του συστήματος στη αναζήτηση για τον κατάλληλο χρήστη και στην εντόπιση και διόρθωση σφαλμάτων σε περίπτωση που παρουσιαστεί κάποιο πρόβλημα με αυτό.

Όσο για το πεδίο της ημερομηνίας δημιουργίας εγγράφου, ο πρώτος λόγος για την αποφυγή κρυπτογράφησης της τιμής του είναι ότι, όπως και το πεδίο ταυτότητας χρήστη, η πληροφορία που παρουσιάζει θεωρείται χαμηλής κρισιμότητας, ως προς την πληροφορία που παρουσιάζει. Ακόμα και σε περίπτωση υποκλοπής του εγγράφου, το πεδίο απλά περιέχει το πότε δημιουργήθηκε το έγγραφο. Ο δεύτερος λόγος έχει να κάνει με τον τύπο της πληροφορίας και τη σημασία του πεδίου σε άλλες λειτουργίες της εφαρμογής. Το πεδίο ημερομηνίας δημιουργίας δρα ως ένα δεύτερο συστατικό ενός

μοναδικού αναγνωριστικού για ένα έγγραφο, που θα χρησιμοποιηθεί, σε συνδυασμό με την ταυτότητα του χρήστη, για τη σύνθεση τιμής μιας ταυτότητας εγγράφου.

Η τιμή αυτή θα είναι πρακτικά μοναδική, καθώς οι τιμές Timestamp [7] περιέχουν ιδιαίτερα ακριβείς πληροφορίες για την ημερομηνία και ώρα που παρουσιάζουν, μέχρι και στα μιλιδευτερόλεπτα της ώρας. Η λεπτομερής αυτή τιμή επιτρέπει στην εφαρμογή να ξεχωρίζει μεταξύ εγγράφων, ακόμα και αν έχουν ίδιες τιμές σε όλα τα άλλα πεδία τους, καθώς είναι αδύνατον για έναν άνθρωπο να καταγράψει δύο αρχείο στο ίδιο μιλιδευτερόλεπτο. Επομένως, η τιμή πρέπει να παραμείνει τύπου Timestamp [7] στη βάση δεδομένων [5], που περιέχει την απαραίτητα λεπτομερή πληροφορία χωρίς πρόβλημα και μπορεί να μετατραπεί με ευκολία σε τιμή DateTime πίσω στην εφαρμογή [7], καθώς έτσι γίνεται πιο εύκολη η σύγκριση δύο τιμών όταν αυτό χρειάζεται. Καθώς χρειάζεται να δρα ως μοναδικό αναγνωριστικό για την αναζήτηση εγγράφων, η ταυτότητα εγγράφου δεν κρυπτογραφείται.

5.2.5 Υλοποίηση αποθήκευσης δεδομένων στο Firebase

Η αποθήκευση δεδομένων στο Firebase [6] γίνεται μέσω της υπηρεσίας Firebase Firestore [5]. Η υπηρεσία αυτή παρέχει μία βάση δεδομένων, στην οποία μπορούν να δημιουργηθούν διαφορετικές συλλογές για κάθε είδος αντικείμενου που σχεδιάζεται να διαχειρίζεται η εφαρμογή. Κάθε αντικείμενο που δημιουργεί ο χρήστης αποθηκεύεται στην αντίστοιχη συλλογή της βάσης ως ένα καινούργιο έγγραφο της, με τα πεδία του να καθορίζονται από ένα χάρτη με κλειδιά κειμένου και δυναμικές τιμές, δηλαδή τιμές που μπορούν να είναι οποιοδήποτε τύπου.

Το ίδιο το Firestore [5] μπορεί να αποθηκεύσει τιμές τύπου κειμένου, ακέραιου ή δεκαδικού αριθμού (int64 ή double), τιμές λογικής Boolean, χάρτες, λίστες, πίνακες, γεωγραφικά σημεία, τιμές χρόνου Timestamp, κενές τιμές ή και αναφορές σε άλλα έγγραφα. Από αυτά τα είδη τιμών, η εφαρμογή χρησιμοποιεί κυρίως τιμές κειμένου, αλλά υπάρχουν περιπτώσεις όπου βρίσκει χρήσεις και για τις περισσότερες από τα υπόλοιπα είδη τιμών, με εξαίρεση των αριθμών, των γεωγραφικών σημείων και των αναφορών. Στις περισσότερες περιπτώσεις, αν χρησιμοποιούνται λίστες, πρόκειται για λίστες κειμένου.

Αυτά τα δεδομένα θα εισαχθούν στη βάση από την εφαρμογή, η οποία με τη σειρά της τα λαμβάνει από το χρήστη, κυρίως από τα πλαίσια εισαγωγής κειμένου που έχουν αναφερθεί σε προηγούμενες ενότητες. Όσο για την ίδια διαδικασία εισαγωγής δεδομένων, η ακριβής υλοποίησή της διαφέρει ανάλογα με το ρόλο που παίζει στην εφαρμογή.

Για τις περισσότερες διαδικασίες εισαγωγής δεδομένων, κυρίως σε αυτές που εκτελούνται στις οθόνες εισαγωγής, η διαδικασία πρέπει να γίνεται αντιληπτή από το χρήστη, δίνοντάς του έτσι άμεση ενημέρωση σε απάντηση στις ενέργειες που έκανε. Η οθόνη πρέπει να εμφανίζει κάτι σε αυτόν, που θα δηλώνει την κατάσταση του αιτήματός του. Αυτό επιτυγχάνεται με δύο συναρτήσεις εμφάνισης μπάρας ενημέρωσης, οι οποίες καλούνται ανάλογα με την επιτυχία ή αποτυχία της απόπειρας εισαγωγής.

Εάν η απόπειρα αποτύχει, ανεξάρτητα αν αυτό οφείλεται στο ότι η σύνδεση με τη βάση χάθηκε, αν η ίδια η βάση αντιμετωπίζει προβλήματα, ή για κάποιον άλλο λόγο, η ενημέρωση που θα λάβει ο χρήστης θα τον πληροφορεί απλά για την αποτυχία της απόπειράς του και η διαδικασία θα τελειώσει. Η ενημέρωση αυτή, λοιπόν, παρουσιάζεται σε περίπτωση που εμφανιστεί το οποιοδήποτε σφάλμα, από τη στιγμή που αρχίζει η διαδικασία καταχώρησης και αποθήκευσης δεδομένων στη βάση.

Από την άλλη, εάν η απόπειρα ολοκληρωθεί με επιτυχία, οι ακόλουθες αντιδράσεις της εφαρμογής είναι αρκετά διαφορετικές. Πρώτον, αντί να σταλθεί αμέσως ενημέρωση στο χρήστη, θα ξεκινήσει μία δεύτερη διαδικασία καταχώρησης δεδομένων, αυτή τη φορά για την καταγραφή της τελευταίας πράξης

του χρήστη στον κατάλογο ενεργειών του χρήστη. Η δεύτερη αυτή διαδικασία θα είναι ασύγχρονη, δηλαδή η πρώτη διαδικασία δε θα περιμένει μετά την εκκίνηση της δεύτερης, αλλά θα προχωρήσει στην επόμενη ενέργειά της, δηλαδή στην ενημέρωση του χρήστη για την επιτυχία της απόπειρας εισαγωγής.

Σε ορισμένες οθόνες εισαγωγής, η διαδικασία εισαγωγής στον κατάλογο ενεργειών δεν θα είναι η τελευταία διαδικασία εισαγωγής, που θα ενεργοποιηθεί με την επιτυχία της απόπειρας του χρήστη. Μεταξύ της εκκίνησης της δεύτερης διαδικασίας και της ενημέρωσης του χρήστη για την επιτυχία της απόπειρας, θα ξεκινήσει και μία τρίτη διαδικασία, η οποία θα είναι η καταγραφή του αντικειμένου που δημιουργήθηκε και στο ιστορικό του χρήστη.

Σε αντίθεση με τη δεύτερη διαδικασία, η τρίτη διαδικασία θα αναγκάσει την πρώτη να περιμένει, μέχρι την ολοκλήρωσή της, προτού προχωρήσει στην ενημέρωση του χρήστη για την επιτυχία της απόπειρας εισαγωγής. Τα αντικείμενα των οποίων η δημιουργία καλεί αυτή τη διαδικασία είναι τα αντικείμενα τύπου «Παθήσεις», «Φάρμακο», «Εξέταση», «Εγχείρηση», «Επίσκεψη», «Αποτελέσματα εξετάσεων», «Αποτελέσματα εγχείρησης». Με την ολοκλήρωση όλων αυτών των διαδικασιών και την ενημέρωση του χρήστη για αυτό, δημιουργείται και ένα τοπικό αντικείμενο, για την αντίστοιχη λίστα δεδομένων στην εφαρμογή.

Πρέπει να αναφερθεί σε αυτό το σημείο ότι, λόγω της σειράς διαδικασιών εισαγωγής που καλούνται, υπάρχει μία πιθανότητα η εφαρμογή να παρουσιάσει στο χρήστη ενημέρωση σφάλματος, χωρίς αυτό να ισχύει, και ενώ η πρώτη εισαγωγή ολοκληρώθηκε με επιτυχία, καθώς η σύνδεση του χρήστη μπορεί να διακοπεί σε λάθος στιγμή. Παρομοίως, δεν είναι αδύνατον, υπό ορισμένες συνθήκες, η πρώτη διαδικασία να ολοκληρωθεί, χωρίς να ολοκληρωθούν η δεύτερη και η τρίτη.

Σε περίπτωση που προκύψει ένα σφάλμα αυτού του τελευταίου τύπου, ο χρήστης έχει τρόπους να εντοπίσει το νέο αντικείμενο που δημιούργησε, ή την έλλειψη του αντίστοιχου αντικειμένου στο ιστορικό του. Υπάρχουν διάφορες οθόνες, όπως αναφέρθηκε στο κεφάλαιο 4, που προβάλλουν τα έγγραφα μιας συλλογής και επιτρέπουν στο χρήστη να τα τροποποιήσει ή να τα διαγράψει. Με αυτό τον τρόπο, ο χρήστης μπορεί να παρατηρήσει την ύπαρξη του αντικειμένου, ακόμα και σε περίπτωση που προκύψει σφάλμα.

Παρόμοιες διαδικασίες εισαγωγής δεδομένων στον κατάλογο ενεργειών του χρήστη υπάρχουν και στις οθόνες τροποποίησης και διαγραφής, ενημερώνοντας τη βάση [5] για κάθε δράση του χρήστη κατά την επιτυχία τους. Σε αντίθεση, όμως, με τις οθόνες εισαγωγής, υπάρχει μια σημαντική διαφορά στην τρίτη διαδικασία, η οποία καλείται στις οθόνες τροποποίησης και διαγραφής δεδομένων, αφού ολοκληρωθεί επιτυχώς η τροποποίηση, ή η διαγραφή δεδομένων από την αντίστοιχη συλλογή της οθόνης.

Εάν πρόκειται για διαδικασία τροποποίησης που εκτελέστηκε, η τρίτη διαδικασία τροποποιεί το αντίστοιχο αντικείμενο στο ιστορικό του χρήστη, αντί να δημιουργήσει καινούργιο. Παρομοίως, εάν η διαδικασία που εκτελέστηκε ήταν η διαγραφή ενός αντικειμένου, η τρίτη διαδικασία θα διαγράψει το αντίστοιχο αντικείμενο από το ιστορικό. Όπως και στις οθόνες εισαγωγής, τρίτη διαδικασία καλείται μόνο όταν αφορά τα συγκεκριμένα είδη αντικειμένων, που αναφέρθηκαν παραπάνω.

Η μέθοδος αυτή εφαρμόζεται μόνο στις αλλαγές, που αφορούν τις συλλογές της βάσης [5], τις οποίες εκτελεί ο ίδιος ο χρήστης. Το σύστημα διαθέτει και άλλες λειτουργίες, που εκτελούνται αυτόματα από την εφαρμογή και επομένως είναι σχεδιασμένες να τρέχουν στα παρασκήνια, επομένως δε γίνονται αντιληπτές από το χρήστη. Το πιο ξεκάθαρο παράδειγμα αυτού του είδους διαδικασίας αφορά τη δημιουργία νέας μηνιαίας αναφοράς.

Οι μηνιαίες αναφορές δημιουργούνται αυτόματα από την εφαρμογή, όταν υπολογίζεται ότι έχει περάσει ένας μήνας από την τελευταία. Η διαδικασία, που υπολογίζει τη διαφορά χρόνου, εκτελείται κατά την

εκκίνηση της εφαρμογής, αν ο χρήστης είναι ήδη συνδεδεμένος, ή κατά την επιτυχή σύνδεση, αν δεν είναι. Η εκπόνηση της νέας αναφοράς γίνεται χωρίς επέμβαση του χρήστη και μάλιστα χωρίς καν να έχει ο ίδιος γνώση, μέχρις ότου παρουσιαστεί η νέα αναφορά. Αυτή η αναφορά, αφού δημιουργηθεί με επιτυχία, θα σταλθεί στο χρήστη ως ειδοποίηση [22] και θα είναι διαθέσιμη και από την οθόνη τελευταίας αναφοράς.

Το παράδειγμα αυτό δείχνει τη διαφορά μεταξύ διαδικασιών αποθήκευσης, οι οποίες είναι σχεδιασμένες ώστε να είναι προφανείς και εκείνων, που είναι σχεδιασμένες ώστε να είναι αόρατες. Η πρώτη κατηγορία κρατά το χρήστη ενήμερο για την κατάσταση της διαδικασίας που εκτελείται, ανεξάρτητα από το αν η απόπειρά του καταλήξει σε επιτυχία ή σε αποτυχία. Επιπλέον, ο χρήστης έχει πλήρη γνώση ότι εκτελείται η διαδικασία, καθώς είναι αυτός που την εκκίνησε.

Η δεύτερη κατηγορία, από την άλλη, δεν ενημερώνει το χρήστη παρά μόνο σε περίπτωση επιτυχίας. Ο χρήστης δεν έχει γνώση του πότε εκτελείται η διαδικασία ακριβώς και δεν έχει ξεκάθαρη γνώση της αιτίας ή της λογικής που ακολουθείται στην πορεία. Πληροφορείται για την εκτέλεσή της μόνο από τα αποτελέσματα της επιτυχίας της.

5.2.6 Υλοποίηση λειτουργιών ημερολογίου και ραντεβού

Η υλοποίηση της λειτουργίας ημερολογίου έγινε με τη χρήση του πακέτου Flutter «table_calendar» [20], ένα έτοιμο πακέτο σχεδιασμένο να επιτρέπει τη δημιουργία εξειδικευμένων ημερολογίων, για κάθε είδος χρήσης. Το πακέτο παρέχει ένα Widget ημερολόγιο, το οποίο μπορεί μετά να τροποποιηθεί με διάφορους τρόπους, πέρα από τη βασική μορφή και λειτουργία με την οποία ξεκινάει.

Σε αυτή την περίπτωση, το ημερολόγιο χρησιμοποιείται για την προβολή των ραντεβού, που έχει προγραμματίσει ο χρήστης. Για το σκοπό αυτό, στο άνοιγμα της οθόνης, η εφαρμογή διαβάζει τις τοπικά αποθηκευμένες λίστες, με τα διαφόρων ειδών ραντεβού, που έχει κανονίσει ο χρήστης, ελέγχοντας για ραντεβού εξέτασης, εγχείρησης, ή οποιαδήποτε άλλη καταγεγραμμένη επίσκεψη, που έχει προγραμματιστεί από αυτόν.

Οι τιμές αυτές θα χρησιμοποιηθούν από το ημερολόγιο [20] κατά τη διαδικασία του χτισίματός του. Συγκεκριμένα, οι ημέρες που παρουσιάζει χρωματίζονται με διαφορετικό τρόπο, ανάλογα με τα ραντεβού που περιέχουν. Τα ραντεβού που κρίνονται πιο κρίσιμα στη σημασία τους από τη φύση τους, έχουν χρώματα που τραβούν το βλέμμα περισσότερο και σημάνουν την αυξημένη σημασία τους, ενώ οι μέρες χωρίς κανένα προγραμματισμένο ραντεβού διατηρούν το αρχικό τους λευκό χρώμα.

Εάν υπάρχουν πολλά ραντεβού, διαφορετικών ειδών, προγραμματισμένα για την ίδια μέρα, προτεραιότητα για το χρώμα που θα πάρει η ημέρα δίνεται στην πιο κρίσιμη κατηγορία, με τις εγχειρήσεις να είναι στην κορυφή της λίστας και τις γενικές επισκέψεις τελευταίες. Αυτό γίνεται προκειμένου ο χρήστης να μπορεί να δει με μια ματιά το πιο σημαντικό ραντεβού που έχει, για μια οποιαδήποτε μέρα.

Καθώς ένα ημερολόγιο χρησιμοποιείται για να προβάλει πληροφορίες, με τρόπο που γίνεται εύκολα κατανοητός απλά κοιτώντας το, είναι απαραίτητο οι οπτική παρουσίαση αυτών των ημερών να είναι ικανοποιητικά σχεδιασμένη. Η συνηθισμένη κλίμακα χρωμάτων για μία τέτοια περίπτωση θα ήταν τα λιγότερο επείγοντα ραντεβού να χρωματίζονται με ψυχρά χρώματα, όπως το μπλε, ενώ τα πιο επείγοντα ραντεβού να έχουν περισσότερο θερμά χρώματα, με το κόκκινο να είναι το χρώμα που χρησιμοποιείται παραδοσιακά, για τις πιο κρίσιμες περιπτώσεις. Όμως, με βάση το σκοπό της εφαρμογής και την εντύπωση που θα κάνουν τα χρώματα αυτά σε ένα χρήστη, οι τελικές επιλογές για χρώματα ακολουθούν μια παραλλαγή αυτής της λογικής.

Το πρώτο πρόβλημα είναι η χρήση του μπλε χρώματος στην οθόνη. Λόγω της εμφάνισης του UI της εφαρμογής, το χρώμα θα μπορούσε εύκολα να μην τραβήξει το μάτι, ή να μην είναι προφανές στο χρήστη ότι σημάνει ένα είδος ραντεβού, για εκείνη την ημέρα. Για αυτό το λόγο, το χρώμα για τις επισκέψεις, το λιγότερο κρίσιμο είδος ραντεβού στη λογική του σχεδιασμού, αποφασίστηκε ότι θα είναι το πράσινο. Το χρώμα αυτό τραβάει την προσοχή περισσότερο από το μπλε, αλλά είναι επίσης ένα χρώμα με θετικές αντιλήψεις από το μέσο άτομο, δείχνοντας ότι το ραντεβού δεν είναι κάτι κρίσιμο.

Στην ακριβώς αντίθετη πλευρά της κλίμακας, η επιλογή των χρωμάτων που χρησιμοποιούνται για τα πιο επείγοντα ραντεβού, τις εγχειρήσεις, δεν μπορούσε να είναι το κόκκινο. Η ιδέα μιας εγχείρησης από μόνη της προκαλεί άγχος σε ένα άτομο, οπότε η χρήση του κόκκινου ως το χρώμα για το ραντεβού θεωρήθηκε υπερβολικά αρνητική και πιθανόν να χειροτερέψει το άγχος του χρήστη. Με αυτή τη λογική, το χρώμα που χρησιμοποιείται για αυτή την κατηγορία ορίστηκε ότι θα είναι το πορτοκαλί, που μεταφέρει ακόμα την ένδειξη του επείγοντος αλλά δεν σημάνει άμεσο κίνδυνο όπως το κόκκινο.

Με τις δύο ακραίες περιπτώσεις της κλίμακας επιλεγμένες, το τελευταίο χρώμα για τα ραντεβού εξετάσεων, που πέφτουν κάπου στο ενδιάμεσο ως προς την κρισιμότητά τους, αποφασίστηκε ότι θα είναι ένα κίτρινο χρώμα.

Στο θέμα των χρωμάτων των ημερών, το Widget παρουσιάζει μία συγκεκριμένη απαίτηση για τη σωστή παρουσίασή τους. Αυτή η απαίτηση είναι ότι ο κώδικας χτισίματος του διακρίνει μεταξύ της εστιασμένης ημέρας, της τρέχουσας ημέρας και τη γενική λογική χτισίματος για τις ημέρες. Αυτό σημαίνει ότι η λογική για το χρωματισμό των ημερών πρέπει να εφαρμοστεί και στις τρεις για το αποτέλεσμα που προσπαθεί να επιτύχει η εφαρμογή.

Αφού ολοκληρωθεί το χτίσιμό του, το ημερολόγιο παρέχει στο χρήστη τη δυνατότητα να επιλέξει οποιαδήποτε ημέρα που προβάλλεται, ακόμη και να προχωρήσει και σε άλλους μήνες και να κάνει την επιλογή του από εκεί [20]. Το ημερολόγιο καλύπτει τις ημερομηνίες από την αρχή του 1920 μέχρι το τέλος του έτους 20 χρόνια μετά το τρέχον έτος, με την τιμή αυτή να υπολογίζεται δυναμικά στο άνοιγμα της οθόνης. Η επιλογή αυτή, για το όριο των ημερομηνιών, που μπορεί να προβάλει το ημερολόγιο στο μέλλον, γίνεται κυρίως για πρακτικούς λόγους, καθώς θεωρείται λογικό ότι ένας χρήστης δε θα θέλει να κλείσει κάποιο ραντεβού περισσότερο από 20 χρόνια στο μέλλον.

Από τη στιγμή που ο χρήστης επιλέξει μια ημέρα, το ημερολόγιο εκτελεί μία αναζήτηση στις τοπικά αποθηκευμένες λίστες για ραντεβού, που λαμβάνουν χώρα εκείνη την ημέρα και τοποθετεί τα αποτελέσματά της αναζήτησης σε προσωρινές λίστες, ανάλογα με το είδος του ραντεβού. Η διάκριση αυτή θα βοηθήσει αργότερα στην κατάλληλη προβολή των πληροφοριών, όταν υπάρχουν αντικείμενα για να προβληθούν.

Αν οι προσωρινές λίστες παραμείνουν άδειες μετά την αναζήτηση, το ημερολόγιο συμπεραίνει ότι δεν υπάρχουν ραντεβού για εκείνη την ημέρα. Με αυτή την πληροφορία, προχωρά στο να αντιμετωπίσει τον πιθανότερο λόγο για τον οποίο ο χρήστης θα μπορούσε να επιλέξει μια ημέρα που, σε αυτόν, θα ήταν ξεκάθαρο ότι δεν περιέχει ραντεβού. Με άλλα λόγια, υποθέτει ότι ο χρήστης σημάνει με την επιλογή του ότι θέλει να προγραμματίσει καινούργιο ραντεβού, με την επιλεγμένη μέρα ως την ημερομηνία του. Επομένως, θα σημάνει στην εφαρμογή να ανοίξει ένα παράθυρο για τη γρήγορη εισαγωγή ενός ραντεβού.

Η διαδικασία, η οποία διαχειρίζεται αυτό το παράθυρο, περιμένει μέχρι να εισάγει ο χρήστης με επιτυχία το καινούργιο έγγραφο στην κατάλληλη συλλογή της βάσης δεδομένων [5], προσέχοντας για το σωστό κλείσιμο του παραθύρου εισαγωγής. Ο λόγος για αυτό είναι ότι περιμένει για ένα σήμα από το παράθυρο κατά το κλείσιμό του. Εφόσον το λάβει, θα προχωρήσει στο να εξετάσει ξανά τις λίστες ραντεβού,

αναζητώντας για το καινούργιο αντικείμενο που προστέθηκε, για να το προσθέσει στην κατάλληλη προσωρινή λίστα. Με την ολοκλήρωση της αναζήτησής του, αναγκάζει ένα καινούργιο χτίσιμο της οθόνης προκειμένου να εμφανίζεται το νέο ραντεβού στο ημερολόγιο.

Αυτή είναι η λειτουργία του ημερολογίου εάν δε βρεθεί ραντεβού για την επιλεγμένη ημέρα. Η συμπεριφορά του όμως αλλάζει στην περίπτωση που οι προσωρινές λίστες δεν μείνουν άδειες μετά την αναζήτηση. Αντί να σημάνει για την εμφάνιση του παραθύρου εισαγωγής, το ημερολόγιο προκαλεί τη δημιουργία και εμφάνιση μίας λίστας των ραντεβού από κάτω του. Στη λίστα αυτή διοχετεύονται όλες οι προσωρινές λίστες, των οποίων το περιεχόμενο θα χρησιμοποιηθεί για τη δημιουργία των αντικειμένων της.

Το κάθε αντικείμενο της λίστας δημιουργείται με βάση την προσωρινή λίστα από την οποία προήλθαν τα δεδομένα που χρησιμοποιεί. Το αντικείμενο παρουσιάζει πάντα την τοποθεσία σημειωμένη για το ραντεβού, ανεξάρτητα από το είδος του, αλλά η προφανής διαφορά φαίνεται στο χρωματισμένο κουτάκι, που βρίσκεται δίπλα σε αυτό το κείμενο. Τα χρώματα που χρησιμοποιούνται και οι κατηγορίες αντικειμένων στα οποία αντιστοιχούν, παραμένουν όπως αναφέρθηκε νωρίτερα. Με αυτό τον τρόπο, σε συνδυασμό με την ονομασία της τοποθεσίας, ο χρήστης θεωρητικά πληροφορείται για το είδος του ραντεβού και τους κανόνες χρήσης των χρωμάτων από την εφαρμογή χωρίς αυτή η πληροφορία να παρουσιάζεται γραπτά.

Πέρα από την τοποθεσία του ραντεβού και το χρωματισμένο τετράγωνο, που δηλώνει το είδος του ραντεβού, τα αντικείμενα της λίστας αυτής παρέχουν πληροφορίες και για την ώρα για την οποία αυτό έχει προγραμματιστεί. Χρησιμοποιώντας εικονίδια δίπλα από το κείμενο του κάθε πεδίου, ώστε να είναι προφανές στο χρήστη, παρουσιάζονται επίσης και η ημερομηνία και η ώρα κατά την οποία θα λάβει χώρα το ραντεβού.

Βέβαια, ένας περιορισμός αυτού του σχεδίου, είναι ότι εξαρτάται από το χρήστη να καταγράψει την τοποθεσία, με εύκολα κατανοητό τρόπο, ή τουλάχιστον με τρόπο που ο ίδιος θα καταλαβαίνει. Για παράδειγμα, εάν έχει δύο ραντεβού την ίδια ημέρα, ένα για εξέταση και ένα για εγχείρηση στην ίδια τοποθεσία, η παρουσίαση των δεδομένων δεν είναι εξίσου αποτελεσματική, στην περίπτωση που δεν έχει προσθέσει κάποιο αναγνωριστικό, ή αν δε θυμάται την ώρα του κάθε ραντεβού. Η βασική πληροφορία, η ύπαρξη των ραντεβού και η ώρα στην οποία πρέπει να βρίσκεται εκεί, μεταφέρεται ως ένα σημείο, αλλά όχι όσο αποτελεσματικά θα μπορούσε.

Καθώς, όμως, υπάρχει η πιθανότητα ο χρήστης να επιθυμεί να εισάγει καινούργιο ραντεβού, σε μία ημέρα όπου έχει ήδη προγραμματίσει κάποιο άλλο, είναι απαραίτητο να υπάρχει μέθοδος εισαγωγής και γι' αυτήν την περίπτωση. Η μέθοδος που αναφέρθηκε νωρίτερα δεν είναι κατάλληλη, καθώς καλείται μόνο αν δεν υπάρχει ραντεβού για την επιλεγμένη ημέρα. Επομένως, η μέθοδος για την εισαγωγή καινούργιου αντικειμένου πρέπει να είναι προφανής στο χρήστη και αρκεί να είναι διαθέσιμη μόνο όταν πρόκειται για ημέρα με προγραμματισμένο ραντεβού.

Για αυτό το λόγο, πάνω από την λίστα με τα παρουσιασμένα ραντεβού, υπάρχει και ένα κουμπί για την εισαγωγή καινούργιων. Καθώς το στοιχείο αυτό δεν δημιουργείται παρά μόνο αν η επιλεγμένη μέρα δεν είναι κενή, η ύπαρξη του κουμπιού δεν αποσπάει την προσοχή από την παρουσίαση του ημερολογίου [20]. Ταυτόχρονα, η διαδικασία, που εκτελείται στο πάτημά του, μπορεί να είναι ίδια με αυτήν που εκτελείται στην περίπτωση που επιλεγθεί κενή ημέρα, μια και το πρόβλημα ήταν η ενεργοποίησή της μόνο όταν κρίνεται κατάλληλο, είτε από το χρήστη είτε από τη λογική της εφαρμογής. Η λογική της ίδιας της διαδικασίας δε χρειάζεται αλλαγές.

Εκτός από τη χρήση του ημερολογίου [20], η εφαρμογή παρέχει και άλλους τρόπους για τον χρήστη να δει και να διαχειριστεί τα ραντεβού του. Οι οθόνες τροποποίησης και διαγραφής κάθε κατηγορίας παρουσιάζουν τα δεδομένα ως λίστες, επιτρέποντάς του να δει όλα τα ραντεβού που έχει προγραμματίσει. Η μεγαλύτερη διαφορά, πέρα από τον τρόπο παρουσίασής τους, είναι ότι η κάθε οθόνη τροποποίησης ή διαγραφής παρουσιάζει μόνο τα αντικείμενα που ανήκουν στην κατηγορία της, ενώ η οθόνη ιατρικού ημερολογίου παρουσιάζει όλα τα ραντεβού κάθε είδους σε μία οθόνη.

Η άλλη διαφορά είναι ότι ο χρήστης δεν μπορεί να τροποποιήσει ή να διαγράψει ραντεβού από το ημερολόγιο και πρέπει να περάσει από την αρμόδια οθόνη τροποποίησης ή διαγραφής για να το κάνει. Αυτό γίνεται επειδή η οθόνη ιατρικού ημερολογίου έχει ως βασικό σκοπό την προβολή των ραντεβού και όχι τη διαχείρισή τους. Ακόμα και η λειτουργία εισαγωγής νέου ραντεβού της οθόνης αυτής, υπάρχει περισσότερο, επειδή θεωρείται πιθανόν για τον χρήστη να βρει πιο εύκολα κατανοητή τη διαδικασία εισαγωγής, όταν μπορεί να δει την ημέρα που προγραμματίζει το ραντεβού σε μορφή ημερολογίου, παρά γιατί η προσθήκη της αποτελεί σημαντικό τμήμα της λειτουργίας της οθόνης.

Τέλος, ανάμεσα στις λειτουργίες ραντεβού, περιλαμβάνεται και η υπενθύμιση, προς το χρήστη, για αυτά, όταν έρχεται η κατάλληλη ημέρα. Προς αυτό το σκοπό, εκδίδεται ειδοποίηση προς το χρήστη, κατά την εκκίνηση της εφαρμογής. Αφού συγχρονιστούν οι τοπικές λίστες της εφαρμογής με αυτές της βάσης δεδομένων, εκτελείται μία αναζήτηση στις λίστες ραντεβού και ελέγχεται ποια από αυτά λαμβάνουν χώρα κατά την τρέχουσα ημέρα, για εκείνο το μήνα και χρόνο. Αν βρεθούν αντικείμενα που πληρούν αυτές τις προϋποθέσεις, εκδίδεται ειδοποίηση για κάθε ένα από αυτά [22].

Η ενημέρωση αυτή γίνεται συγκεκριμένα κατά την εκκίνηση ή την επιτυχημένη είσοδο του χρήστη στο λογαριασμό του και ελέγχει μόνο αν τα ραντεβού στη λίστα αντιστοιχούν στην τρέχουσα ημέρα. Αυτό σημαίνει ότι η κάθε ειδοποίηση μπορεί να σταλθεί πολλές φορές, αν ο χρήστης κλείσει και ανοίξει πάλι την εφαρμογή. Επιπλέον, δεν ελέγχει την ώρα του ραντεβού, μόνο την ημέρα, οπότε μπορεί να σταλθούν ειδοποιήσεις [22] για ραντεβού που αντιστοιχούν σε εκείνη την ημέρα, αλλά ώρες πριν ή μετά την προγραμματισμένη ώρα.

Η λειτουργία αυτή θεωρείται αποδεκτό κομμάτι του σχεδιασμού, καθώς πρέπει να ενημερώνει το χρήστη χωρίς γνώση του αν αυτός έχει ήδη λάβει ειδοποίηση, αν ξέχασε το ραντεβού, ή αν υπάρχει κάποιος άλλος παράγοντας που δεν υπολογίστηκε κατά την υλοποίηση της λειτουργίας.

5.2.7 Υλοποίηση γραφημάτων και οπτικοποίησης δεδομένων

Ο βασικός σκοπός μιας λειτουργίας οπτικοποίησης δεδομένων είναι να παρουσιάσει μία συλλογή δεδομένων με τρόπο που γίνεται εύκολα κατανοητός από το χρήστη, συνήθως σε σχέση με κάποιο άλλο μέγεθος στον άλλον άξονα. Η πιο συνηθισμένη επιλογή για αυτό το άλλο μέγεθος είναι ο χρόνος, επιτρέποντας στο χρήστη να παρατηρήσει την πρόοδο των τιμών στην πάροδο του χρόνου. Το τυπικό είδος γραφήματος για αυτό είναι ένα γραμμικό γράφημα [21].

Στην περίπτωση αυτής της εφαρμογής, γραμμικά γραφήματα χρησιμοποιούνται για την οπτικοποίηση των διαφόρων μετρήσεων ζωτικών σημείων, που έχει καταγράψει ο χρήστης [21]. Το κάθε είδος μέτρησης παρουσιάζεται χωριστά από τα υπόλοιπα, διακρίνοντας έτσι μεταξύ μετρήσεων θερμοκρασίας, παλμού, σακχάρου, καθώς και συστολικής και διαστολικής πίεσης.

Παρόλο που υπάρχει διάκριση μεταξύ τους, οι διάφορες κατηγορίες μετρήσεων δεν παρουσιάζονται με την κάθε μία σε διαφορετικό γράφημα, ούτε εμφανίζονται στην οθόνη όλες μαζί. Η επιλογή της κατηγορίας που θα προβληθεί ανατίθεται κυρίως στο χρήστη, ο οποίος μπορεί να αλλάξει μεταξύ των επιλογών με ελευθερία. Μοναδική εξαίρεση στο σχεδιασμό αυτό, είναι το γράφημα που προβάλλεται

στο άνοιγμα της οθόνης γραφημάτων, το οποίο είναι πάντα η προεπιλεγμένη κατηγορία των μετρήσεων θερμοκρασίας.

Το γράφημα που χρησιμοποιείται, για την οπτικοποίηση αυτών των μετρήσεων, είναι πάντα το ίδιο, απλά χτίζεται ξανά κάθε φορά που αλλάζει η επιλεγμένη κατηγορία μετρήσεων [21]. Οι τιμές και η κλίμακά του αλλάζουν κάθε φορά που χτίζεται, με βάση την μέγιστη τιμή ανάμεσα στις μετρήσεις και το μέγεθος της λίστας συντεταγμένων που του δίνεται.

Στον κατακόρυφο άξονα, η τοποθεσία ενός σημείου στο γράφημα εξαρτάται καθαρά από την τιμή της αντίστοιχης μέτρησης [21]. Οι τιμές αυτές παρέχονται στο γράφημα από την οθόνη γραφήματος, η οποία με τη σειρά της συγκεντρώνει αυτές τις τιμές σε μία λίστα, διαβάζοντας την τοπική λίστα μετρήσεων. Προσοχή δίνεται ώστε να καταγράφονται μόνο οι μετρήσεις της κατηγορίας που έχει επιλεγεί και η διαδικασία αυτή εκτελείται κάθε φορά που ο χρήστης αλλάζει την επιλεγμένη του κατηγορία για οπτικοποίηση, μεταξύ των επιλογών που αναφέρθηκαν παραπάνω.

Στον οριζόντιο άξονα, η κάθε τιμή τοποθετείται μία μονάδα μακριά από την προηγούμενη προς τα δεξιά. Με αυτό τον τρόπο δημιουργείται μια αίσθηση του χρόνου από το χρήστη, χωρίς να χρειάζεται να χρησιμοποιηθούν οι ημερομηνίες καταγραφής των μετρήσεων ως τίτλοι ή τιμές στον άξονα. Η διαδικασία αυτή συνεχίζεται για όλες τις τιμές συντεταγμένων που κληρονομεί το γράφημα, άρα το μέγεθος του άξονα εξαρτάται από το πλήθος των μετρήσεων.

Η φιλοσοφία αυτή όμως δημιουργεί μία νέα απαίτηση προκειμένου το σύστημα να δουλέψει σωστά. Αν οι μετρήσεις απλά τοποθετούνται στον οριζόντιο άξονα με βάση της θέση τους στη λίστα, πρέπει η λίστα αυτή να είναι οργανωμένη με βάση τις ημερομηνίες των μετρήσεων, αλλιώς το γράφημα θα παρουσιάζει τις σωστές τιμές αλλά με εντελώς λανθασμένη σειρά.

Καθώς όμως οι συντεταγμένες προέρχονται από την τοπική λίστα, η οργάνωση των τιμών δεν είναι εξασφαλισμένη. Η τοπική λίστα λαμβάνει το περιεχόμενό της μέσω αναζήτησης από το Firebase [5], που σημαίνει ότι τα αντικείμενά της μπορεί να είναι με οποιαδήποτε σειρά αυτά ανακαλύφθηκαν στη βάση δεδομένων. Επιπλέον, ο χρήστης μπορεί να έχει προσθέσει, διαγράψει, ή τροποποιήσει οποιαδήποτε από αυτά, οπότε δεν θα μπορούσε να θεωρηθεί δεδομένο ότι η λίστα μετρήσεων θα ήταν οργανωμένη, ακόμα και αν τα δεδομένα που έλαβε στην αρχή είχαν ληφθεί με αυτή τη σειρά.

Επομένως, η εφαρμογή εκτελεί μία διαδικασία ταξινόμησης των αντικειμένων της τοπικής λίστας μετρήσεων κατά το άνοιγμα της οθόνης γραφημάτων. Η διαδικασία αυτή οργανώνει τα αντικείμενα της λίστας συγκρίνοντας τις τιμές των πεδίων ημερομηνιών τους. Καθώς η οθόνη γραφημάτων δεν περιέχει κάποια άλλη μέθοδο, που μπορεί να αλλάξει τη λίστα μετρήσεων, αρκεί μία μόνο εκτέλεση αυτής της διαδικασίας, για να είναι σίγουρο ότι τα αντικείμενά της είναι οργανωμένα, ανεξάρτητα από οτιδήποτε άλλο μπορεί να κάνει ο χρήστης, κατά την παραμονή του στην οθόνη.

Το σχέδιο αυτό παρουσιάζει ένα σημαντικό περιορισμό, συγκεκριμένα ως προς την παρουσίαση του χρόνου μέσω του γραφήματος. Αν και οι τιμές παρουσιάζονται σε χρονολογική σειρά, το διάστημα μεταξύ τους δε διαφέρει, ανεξάρτητα του χρόνου που μπορεί να έχει περάσει μεταξύ μετρήσεων. Αυτό αποτελεί έναν περιορισμό της υλοποίησης, ο οποίος δεν έχει ξεπεραστεί ακόμα στην τρέχουσα έκδοση της εφαρμογής.

Πέρα από την ταξινόμηση των μετρήσεων, υπάρχει και ένα άλλο θέμα που πρέπει να αντιμετωπιστεί προκειμένου να εκτελείται ικανοποιητικά η οπτικοποίηση δεδομένων. Παρόλο που τα αντικείμενα είναι οργανωμένα και παρουσιάζονται σε χρονολογική σειρά, το γράφημα που παρουσιάζεται δεν έχει ιδιαίτερη χρησιμότητα αν χρησιμοποιεί όλες τις μετρήσεις του χρήστη χωρίς περιορισμούς.

Συγκεκριμένα, υπάρχει η πιθανότητα οι μετρήσεις τις λίστες να προέρχονται από πολλές διαφορετικές χρονιές, με αρκετές ή και όλες από αυτές να ανήκουν σε προηγούμενα έτη.

Γίνεται έτσι απαραίτητο για το χρήστη να μπορεί να ξεχωρίσει μεταξύ χρόνων στον οποίον έχει καταγράψει μετρήσεις. Η επιλογή του αυτή πρέπει με τη σειρά της να περιορίζει τις μετρήσεις που θα γίνουν δεκτές με βάση τις ημερομηνίες λήψης τους. Για διευκρίνηση, η ημερομηνία αυτή αναφέρεται στην ημερομηνία που παρατηρήθηκαν οι μετρήσεις αυτές, όχι την ημερομηνία δημιουργίας του εγγράφου στη βάση δεδομένων.

Κάθε φορά που η οθόνη γραφημάτων ανοίγει, η εφαρμογή εξακριβώνει τα έτη στα οποία έχουν γίνει μετρήσεις από το χρήστη αναζητώντας για αυτά στην τοπική λίστα μετρήσεων, σχηματίζοντας μία λίστα από έτη. Για κάθε μέτρηση στην τοπική λίστα μετρήσεων, ελέγχεται αν το έτος κατά το οποίο έλαβε χώρα η μέτρηση είναι καταγεγραμμένο ήδη και, αν δεν είναι, προστίθεται στη λίστα. Η λίστα είναι οργανωμένη ώστε οι πιο πρόσφατες χρονιές να είναι πιο μπροστά στη λίστα, για τη διευκόλυνση του χρήστη στην επιλογή μιας χρονιάς που θα είναι άμεσα σχετική με την τρέχουσα κατάστασή του. Ανεξάρτητα από αν υπάρχουν μετρήσεις για αυτό ή όχι, το τρέχον έτος είναι πάντα στη λίστα και λειτουργεί ως το προεπιλεγμένο έτος στο άνοιγμα της οθόνης.

Οι καταγεγραμμένες ημερομηνίες παρουσιάζονται στο χρήστη ως δυναμικά δημιουργημένες επιλογές σε ένα κουμπί `DropDownButton`. Οι μετρήσεις που θα χρησιμοποιηθούν από το γράφημα, για την επιλεγμένη κατηγορία, θα περιορίζονται μόνο σε αυτές που έγιναν στο έτος που δείχνει αυτό το κουμπί.

Για την υλοποίηση του ίδιου του γραφήματος, χρησιμοποιείται ένα `Widget` τύπου `LineChart` από το πακέτο `Flutter «fl_chart»` [21]. Οι ελάχιστες τιμές του για τον οριζόντιο και τον κατακόρυφο άξονα ορίζονται ως το 0, ενώ η μέγιστη τιμή του κατακόρυφου άξονα ορίζεται ως 5 μονάδες παραπάνω από τη μέγιστη τιμή στη λίστα συντεταγμένων που παρέχονται στο γράφημα. Δεν ορίζεται μέγιστη τιμή για τον οριζόντιο άξονα.

Οι μετρήσεις της επιλεγμένης κατηγορίας δεν είναι η μόνη γραμμή που περιέχει το γράφημα. Κατά τη διαδικασία κατά την οποία συγκεντρώνονται οι τιμές αυτές σε μία λίστα συντεταγμένων, ορίζεται επίσης και μία ακόμα τιμή, η τυπική τιμή μέτρησης. Η τυπική τιμή μέτρησης καθορίζεται από την επιλεγμένη κατηγορία και αντιστοιχεί σε μία μέτρησή της που θα θεωρούταν «τυπική». Για παράδειγμα, στην κατηγορία της θερμοκρασίας, η τυπική τιμή μέτρησης είναι πάντα ορισμένη στο 37, που θεωρείται μία συνηθισμένη θερμοκρασία σώματος, που σημαίνει ότι δεν υπάρχει τίποτα ανησυχητικό με την υγεία ενός ατόμου.

Η τυπική τιμή μέτρησης δεν επηρεάζεται από τις τιμές μετρήσεις που έχει καταγράψει ο χρήστης, μόνο από την επιλεγμένη κατηγορία. Δεν είναι ούτε ο μέσος όρος των τιμών των σχετικών μετρήσεων, ούτε η ελάχιστη τιμή από αυτές. Είναι μία σταθερή τιμή που καθορίστηκε με βάση έρευνα στις μέσες τιμές για κάθε κατηγορία που θα μπορούσαν να θεωρηθούν κανονικές.

Πέρα από αυτές τις κατηγορίες, υπάρχει και μία επιπλέον που χρησιμοποιεί διαφορετική λογική κώδικα και ένα εντελώς διαφορετικό `Widget` γραφήματος. Αυτή είναι η κατηγορία παθήσεων, η οποία αντλεί τα δεδομένα της από την τοπική λίστα παθήσεων του χρήστη.

Σε αντίθεση με τις προηγούμενες κατηγορίες, η κατηγορία παθήσεων δεν προσπαθεί να παρουσιάσει την πρόοδο μίας κατάστασης του χρήστη στην πάροδο του χρόνου. Ο σκοπός της είναι να προβάλλει το σύνολο των συμβάντων που συνέβησαν στο χρήστη για μία χρονιά, η οποία θα είναι αυτή που επέλεξε ο χρήστης.

Όταν ο χρήστης επιλέξει την κατηγορία παθήσεων, εκτελείται μία αναζήτηση στην τοπική λίστα παθήσεων, για όλα τα αντικείμενα με ημερομηνία παρατήρησης παθήσεως μέσα στο επιλεγμένο έτος. Οι ημερομηνίες αυτές θα συγκεντρωθούν σε μία λίστα και θα περαστούν σε ένα διαφορετικό γράφημα από αυτό που χρησιμοποιούσαν οι υπόλοιπες κατηγορίες, το οποίο θα αντικατασταθεί από το γράφημα συγκεκριμένα για αυτή την κατηγορία.

Το γράφημα αυτό είναι ένα γράφημα με μπάρες, υλοποιημένο με τη χρήση του Widget BarChar του πακέτου Flutter «fl_chart» [21]. Δημιουργεί 12 μπάρες, μία για κάθε μήνα, οι οποίες απέχουν μία μονάδα μεταξύ τους στον οριζόντιο άξονα. Το ύψος μίας μπάρας ορίζεται από το πλήθος ημερομηνιών που είναι καταχωρημένες τον αντίστοιχο μήνα, με το μέγιστο αποδεκτό ύψος του γραφήματος να καθορίζεται δυναμικά από το μεγαλύτερο πλήθος συμβάντων καταγεγραμμένο ανάμεσα σε όλους τους μήνες του έτους.

Για την ευκρίνεια των δεδομένων και της σημασίας τους, κάτω από κάθε μπάρα υπάρχει και το όνομα του μήνα στον οποίο αντιστοιχεί. Οι τίτλοι αυτοί δημιουργούνται με μία διαδικασία που προσαρμόστηκε από τους δημιουργούς του πακέτου «fl_chart» [21] [35], η οποία αντιστοιχεί τη θέση του μήνα σε ένα χάρτη με το κατάλληλο όνομα.

Έτσι, ο χρήστης μπορεί να δει με εύκολα κατανοητό τρόπο το πότε είχε τις περισσότερες παθήσεις σε ένα συγκεκριμένο έτος.

5.2.8 Υλοποίηση ειδοποιήσεων

Μία εφαρμογή διαχείρισης ιατρικών δεδομένων δεν μπορεί να περιορίζεται μόνο στο να αποθηκεύει και να προβάλλει αυτά τα δεδομένα. Σημαντική λειτουργία, που αναμένεται από μία εφαρμογή σαν αυτή, είναι το να μπορεί να ειδοποιεί το χρήστη για διάφορα θέματα, σχετικά με τα δεδομένα που αυτός εισήγαγε. Υπενθυμίσεις για ραντεβού, ή για την έγκαιρη λήψη φαρμακευτικής αγωγής είναι η πιο συνηθισμένη και ίσως αναμενόμενη λειτουργία, που μπορεί να έχει η εφαρμογή.

Για την υλοποίηση αυτής της λειτουργίας, καθώς και για κάθε διαδικασία που θα βασιστεί σε αυτή, χρησιμοποιείται το πακέτο Flutter «flutter_local_notification» [22]. Αυτό ενσωματώνεται σε ένα αρχείο υπηρεσίας ειδοποιήσεων, το οποίο θα κληθεί από την εφαρμογή κατά την εκκίνησή της. Σε εκείνο το σημείο, θα εκτελεστεί η ενεργοποίηση της υπηρεσίας, που θα επιτρέψει την εκπόνηση ειδοποιήσεων από κάθε άλλη διαδικασία της εφαρμογής που το χρειάζεται.

Η χρήση της υπηρεσίας αυτής απαιτεί τη συναίνεση του χρήστη για να λειτουργήσει. Στο πρώτο άνοιγμα της εφαρμογής, παρουσιάζεται ένα παράθυρο, που του ζητάει άδεια να στέλνει ειδοποιήσεις. Η απόφασή του αποθηκεύεται αυτόματα ώστε να μην απαιτείται ο χρήστης να απαντά αυτή την ερώτηση ξανά.

Όσο για το περιεχόμενο της υπηρεσίας που χρησιμοποιείται από άλλες διαδικασίες, αυτό γίνεται καλώντας μία μέθοδό της [22]. Αυτή δέχεται έναν αριθμό ταυτότητας για την ειδοποίηση, καθώς και τον τίτλο και το περιεχόμενο της ειδοποίησης. Με βάση αυτά, θα δημιουργήσει το μήνυμα που θα σταλθεί στη συσκευή και το εμφανίζει. Το πιο απλό παράδειγμα χρήσης ειδοποίησης σε αυτήν την περίπτωση είναι όταν ο χρήστης εισάγει ένα νέο αντικείμενο πληροφορίας εκτάκτου ανάγκης, ή όταν τροποποιεί τις τιμές ενός αντικειμένου που υπάρχει ήδη. Και στις δύο περιπτώσεις, με την επιτυχία της διαδικασίας εισαγωγής ή τροποποίησης, στέλνεται μία καινούργια ειδοποίηση στη συσκευή του χρήστη με τις πληροφορίες του αντικειμένου [22].

Το μήνυμα της ειδοποίησης θα αποτελείται από τον τίτλο, που είναι ένα στατικό μήνυμα, για τη δήλωση ότι η ειδοποίηση αυτή αφορά τις πληροφορίες εκτάκτου ανάγκης του χρήστη, καθώς και ένα δυναμικά

συναρμολογημένο κείμενο ως το περιεχόμενό της. Το κείμενο αυτό δημιουργείται συνδυάζοντας μαζί πολλά μικρότερα μηνύματα, ανάλογα με τα πεδία που έχει συμπληρώσει ο χρήστης και τον τύπο της μεταβλητής που τους αντιστοιχεί.

Για τις μεταβλητές που μπορεί να περιέχουν πάνω από ένα αντικείμενο, εμφανίζεται το όνομα του πεδίου, για παράδειγμα «Αλλεργίες», σε συνδυασμό με σειρές από όλα τα αντικείμενα της αντίστοιχης μεταβλητής. Τα μηνύματα αυτά έχουν τουλάχιστον δύο σειρές κειμένου, καθώς το όνομα του πεδίου είναι σε διαφορετική σειρά από το πρώτο αντικείμενο της λίστας.

Αν το πεδίο δεν ανήκει σε κατηγορία που αναμένεται να περιέχει πολλαπλά αντικείμενα, δηλαδή αν πρόκειται για το πεδίο ομάδας αίματος, το κείμενο είναι μόνο μία γραμμή. Το όνομα του πεδίου και η μοναδική τιμή του τοποθετούνται μαζί και η διαδικασία εκπόνησης προχωρά στο να ελέγξει τα υπόλοιπα πεδία, για να δει αν ο χρήστης έχει προσθέσει αντικείμενα σε ένα από αυτά.

Στην περίπτωση της ειδοποίησης μετά από εισαγωγή ή τροποποίηση αντικειμένου πληροφοριών εκτάκτου ανάγκης, ο σκοπός και το σημείο στη λειτουργία της εφαρμογής που βρίσκεται αυτή η διαδικασία επιτρέπει την απευθείας εντολή έκδοσης ειδοποίησης με πλήρης γνώση του χρήστη για τη λογική που ακολουθείται. Η ειδοποίηση λειτουργεί σαν κάρτα πληροφοριών εκτάκτου ανάγκης, παρέχοντας εύκολη πρόσβαση σε πληροφορίες που μπορεί να σώσουν τη ζωή του χρήστη, ακόμα και όταν η εφαρμογή έχει κλείσει ή η συσκευή έχει κλειδωθεί.

Η ίδια ειδοποίηση στέλνεται επίσης σε μία άλλη περίπτωση, η οποία πραγματοποιείται αμέσως μετά την εκκίνηση της εφαρμογής. Ανάλογα με το αν ο χρήστης συνδεθεί πρώτα στο λογαριασμό του, ή αν η εφαρμογή ξεκινήσει με αυτόν ήδη συνδεδεμένο, μια από τις πρώτες ενέργειες που εκτελούνται, είναι η αναζήτηση των δεδομένων στη βάση δεδομένων, τα οποία μετά αποθηκεύονται στις τοπικές λίστες. Μετά την ολοκλήρωση αυτής της διαδικασίας, γίνεται έλεγχος αν υπάρχει αντικείμενο στη λίστα πληροφοριών εκτάκτου ανάγκης. Αν αποδειχτεί ότι η λίστα δεν είναι άδεια, η ειδοποίηση στέλνεται [22].

Παρόμοια λογική ακολουθείται και για δύο ακόμα περιπτώσεις. Η πρώτη από αυτές έχει να κάνει με τα ραντεβού που μπορεί να έχει προγραμματίσει ο χρήστης για την ημέρα. Με την ολοκλήρωση της αναζήτησης των σχετικών δεδομένων από τη βάση κατά την εκκίνηση, στέλνονται ειδοποιήσεις για κάθε ραντεβού που είναι προγραμματισμένο για εκείνη την ημέρα [22]. Η διαφορά είναι ότι η διαδικασία αυτή στέλνει μία ειδοποίηση για κάθε ξεχωριστό ραντεβού, αντί να στέλνει μία ειδοποίηση με όλα τα ραντεβού.

Το δεύτερο παράδειγμα αυτής της λογικής είναι η μηνιαία αναφορά που δημιουργεί η εφαρμογή για το χρήστη. Η ειδοποίηση αυτή διαφέρει σχετικά από τις υπόλοιπες, κυρίως στο ότι είναι σχεδιασμένη να στέλνεται μόνο μία φορά κάθε μήνα. Χρησιμοποιούνται επίσης περισσότεροι κανόνες για τον έλεγχο αυτής της λειτουργίας, κυρίως προκειμένου να ενημερώνονται άμεσα η βάση και η συσκευή για κάθε αναφορά, αλλά και επειδή πρέπει να μπορεί το σύστημα να αναγνωρίσει με σιγουριά την τελευταία ημερομηνία που γράφτηκε μηνιαία αναφορά, ακόμα και αν ο χρήστης αλλάξει συσκευή ή διαγράψει την εφαρμογή για ένα διάστημα. Πέρα από αυτό όμως, η λογική της ειδοποίησης παραμένει ίδια με αυτήν των παραπάνω παραδειγμάτων.

Το σύστημα έκδοσης ειδοποιήσεων, όμως, δεν περιορίζεται μόνο στο να στέλνει ειδοποιήσεις κατά την εκκίνηση, ή μετά από την εκτέλεση μιας προκαθορισμένης διαδικασίας. Διαθέτει επίσης και τη δυνατότητα να στέλνει ειδοποιήσεις στο χρήστη αυτόματα, με βάση την ώρα, ακόμα και όταν η εφαρμογή δεν είναι ανοιχτή. Αυτό γίνεται με τη βοήθεια του πακέτου Flutter «workmanager»[29], το

οποίο είναι σχεδιασμένο ώστε να επιτρέπει σε εφαρμογές να εκτελούν κώδικα στο παρασκήνιο, ακόμα και όταν αυτές παραμένουν κλειστές.

Το πακέτο αυτό, όπως και το προηγούμενο πακέτο Flutter που αναφέρθηκε, ενσωματώνεται σε ένα αρχείο, το οποίο καλείται και τίθεται σε εκκίνηση από την εφαρμογή στο άνοιγμά της, ανεξάρτητα από το αν ο χρήστης είναι συνδεδεμένος ή όχι. Από εκεί, αρχίζει να εκτελεί συγκεκριμένες διαδικασίες, που έχουν οριστεί στο πρόγραμμα, αφήνοντας ένα ενδιάμεσο χρονικό διάστημα μεταξύ εκτελέσεων που καθορίζεται από τον προγραμματιστή. Οι διαδικασίες αυτές δεν είναι απαραίτητο να αφορούν τη δημιουργία ειδοποιήσεων, αλλά αυτή είναι η λειτουργία που χρειάζεται η εφαρμογή και που ορίζεται για εκτέλεση.

Ως προς τη χρήση του πακέτου αυτού από την εφαρμογή, ο ρόλος που παίζει είναι να επιτρέπει σε αυτή να στέλνει ειδοποιήσεις για την έγκαιρη λήψη φαρμακευτικής αγωγής [29] [22]. Κάθε λίγα λεπτά, ελέγχει αν υπάρχουν φάρμακα σημειωμένα στην αντίστοιχη τοπική λίστα και αν για αυτά έχει σημειωθεί ότι πρέπει να σταλθεί ειδοποίηση.

Για όσα αντικείμενα πληρούν αυτές τις προϋποθέσεις, υπολογίζει τις ώρες στις οποίες πρέπει να παίρνει τα φάρμακά του ο χρήστης και προχωράει στο να κρίνει αν υπάρχει ανάγκη για εκπόνηση μιας ειδοποίησης. Αν για κάποια από τις ώρες έχει σημειωθεί ότι δεν έχει σταθεί ειδοποίηση και η τρέχουσα ώρα είναι ίδια με αυτή ή μετά από αυτήν, στέλνεται αυτόματα ειδοποίηση. Η διαδικασία αυτή συνεχίζεται κάθε λίγα λεπτά, αλλά δεν στέλνονται ενημερώσεις μετά από μία ώρα που μπορεί να ορίσει ο χρήστης.

5.2.9 Διαχείριση διαγραφών και τροποποιήσεων δεδομένων

Η τροποποίηση και η διαγραφή δεδομένων στην εφαρμογή αποτελεί μία ιδιαίτερη πρόκληση. Η εφαρμογή προσπαθεί να προσφέρει στο χρήστη τη δυνατότητα να διαχειρίζεται τα δεδομένα του με μεγάλη ελευθερία και καλύπτοντας όσα περισσότερα πιθανά σενάρια και περιπτώσεις χρήσης γίνεται. Στην πράξη όμως, κάθε φορά που ολοκληρώνει μία οριστική αλλαγή στα τοπικά δεδομένα, ανεξάρτητα από το μέγεθος της αλλαγής, χρειάζεται μετά να έρθει σε επικοινωνία με τη βάση δεδομένων [5] και να περάσει τις νέες εκδόσεις των αντικειμένων σε αυτή.

Βασική λειτουργία της εφαρμογής είναι να παρέχει διάφορες οθόνες, που προβάλλουν λίστες αντικειμένων από κάποια συλλογή της βάσης δεδομένων. Για να επιταχύνεται το χτίσιμό τους, χρησιμοποιούνται οι τοπικές λίστες δεδομένων. Το μόνο σημείο στο οποίο δημιουργείται σύνδεση με τη βάση [5] είναι αφού η αλλαγή έχει οριστικοποιηθεί τοπικά, είτε πρόκειται για τροποποίηση των δεδομένων είτε για διαγραφή τους.

Προς αυτό το σκοπό, οι οθόνες που επιτρέπουν τη διαγραφή ή την τροποποίηση των αντικειμένων τους χρησιμοποιούν ένα σύστημα επιβεβαίωσης των προθέσεων του χρήστη, προκειμένου να είναι βέβαιο ότι όντως αυτό επιθυμεί. Έτσι, αποφεύγεται το να χαθεί χρόνος δημιουργώντας σύνδεση με τη βάση [5] ενώ ο χρήστης θα θέλει να πάρει πίσω τις πράξεις του αμέσως μετά.

Για την τροποποίηση των δεδομένων, η ίδια η φόρμα τροποποίησης αρκεί στις περισσότερες περιπτώσεις. Ο χρήστης αναγκάζεται να εισάγει ο ίδιος τις νέες τιμές που επιθυμεί, προτού τις καταθέσει. Ο χρόνος και η σκέψη που θα πάρει για αυτό το έργο αρκούν συνήθως, για να είναι σίγουρος για τις προθέσεις του.

Οι διαγραφές δεδομένων, από την άλλη, απαιτούν περισσότερη προσοχή, λόγω της απλής μορφής με την οποία παρουσιάζονται για τα περισσότερα είδη αντικειμένων. Είναι ιδιαίτερα εύκολο και γρήγορο για ένα άτομο να σύρει ένα αντικείμενο στο πλάι και να το διαγράψει. Αυτό, αν και βολικό για το

χρήστη, σημαίνει ότι υπάρχει κάθε πιθανότητα αυτός να μετανιώσει τη διαγραφή που έκανε και να θέλει να την αναίρεσει.

Η λύση για αυτό το πρόβλημα είναι η χρήση μιας μπάρας ενημέρωσης του χρήστη. Η μπάρα αυτή εμφανίζεται για 3 δευτερόλεπτα από την στιγμή που εκτελεστεί η τοπική διαγραφή από τη λίστα, ενημερώνοντας το χρήστη για αυτό και παρέχοντάς του ένα κουμπί για την αναίρεση αυτής της ενέργειας. Στο τέλος αυτού του περιθωρίου, η μπάρα σβήνεται και η διαγραφή θεωρείται οριστική από την εφαρμογή. Σε αυτό το σημείο, δημιουργείται η σύνδεση με τη βάση [5] και στέλνεται το αίτημα για τη διαγραφή του αντίστοιχου εγγράφου από την κατάλληλη συλλογή της.

Ανεξάρτητα από το αν πρόκειται για διαγραφή ή τροποποίηση δεδομένων, σημαντικό είναι η εφαρμογή να μπορεί να αναγνωρίσει το σωστό έγγραφο στη βάση [5] για την αλλαγή, αλλιώς μπορεί να επηρεαστούν έγγραφα που δεν σχετίζονται με αυτό. Ταυτόχρονα, το αναγνωριστικό που θα χρησιμοποιηθεί πρέπει να είναι αρκετά μοναδικό ώστε να μην είναι λογικά πιθανόν να αντιγραφεί ή να προκύψει πολλαπλές φορές.

Για τα περισσότερα έγγραφα της βάσης δεδομένων [5], το ρόλο αυτό παίζουν δύο διαφορετικά πεδία, η ταυτότητα χρήστη του χρήστη και η ημερομηνία δημιουργίας του εγγράφου. Η πρώτη τιμή περιορίζει τα αποτελέσματα, που «βλέπει» η εφαρμογή με την αναζήτησή της, μόνο στα έγγραφα που δημιουργήθηκαν από το συνδεδεμένο χρήστη. Η δεύτερη τιμή, από την άλλη, αποτελεί μία τιμή που είναι δύσκολο να αντιγράψει κάποιος, καθώς είναι η ημερομηνία και η ώρα στην οποία δημιουργήθηκε το έγγραφο με ακρίβεια μιλιδευτερολέπτων.

Μαζί, τα δύο αυτά αναγνωριστικά δρουν ως ικανοποιητικά μοναδικά για τους σκοπούς της εφαρμογής. Οι τιμές τους συνδυάζονται σε μία ταυτότητα εγγράφου, η οποία αποθηκεύεται στη βάση δεδομένων [5] και χρησιμοποιείται, μαζί με την ταυτότητα χρήστη, ως στόχος αναζήτησης, όταν η εφαρμογή ψάχνει για συγκεκριμένο έγγραφο

Εξαίρεση στην παραπάνω μεθοδολογία αποτελούν οι συλλογές στόχων και πληροφοριών εκτάκτου ανάγκης. Σε αντίθεση με τις υπόλοιπες συλλογές, αυτές οι δύο είναι σχεδιασμένες να διατηρούν μόνο ένα έγγραφο από κάθε χρήστη, το οποίο θα τροποποιείται κάθε φορά που ο χρήστης θα εκτελεί εντολές εισαγωγής σε αυτές μετά τη δημιουργία του. Η διαφορά αυτή επιτρέπει τη χρήση μιας εναλλακτικής μεθοδολογίας ως προς την τροποποίηση και διαγραφή τιμών από το έγγραφο, ή και το ίδιο το έγγραφο. Συγκεκριμένα, η ύπαρξη μόνο ενός αντικείμενου στη λίστα σημαίνει ότι αρκεί η χρήση της ταυτότητας χρήστη σαν αναγνωριστικό.

Επιπλέον, όσον αφορά τη διαγραφή εγγράφων, διατίθεται στο χρήστη και η ικανότητα να επιλέξει μεταξύ της ολοκληρωτικής διαγραφής του εγγράφου, ή της διαγραφής συγκεκριμένων τιμών ή πεδίων από αυτό. Εάν επιλέξει τη διαγραφή του εγγράφου, το αντικείμενο και όλα τα πεδία του διαγράφονται από την τοπική λίστα και τη βάση δεδομένων, ακριβώς όπως γίνεται και στα υπόλοιπα είδη αντικειμένων.

Η άλλη επιλογή που μπορεί να διαλέξει είναι η διαγραφή συγκεκριμένων τιμών ή πεδίων. Ο παράγοντας που διακρίνει μεταξύ του αν θα μπορεί να διαγραφεί ολόκληρο το πεδίο, ή συγκεκριμένες τιμές που θα καταθέτει ο χρήστης, είναι ο τύπος του πεδίου. Αν το πεδίο δέχεται μόνο μία τιμή κάθε φορά, ο χρήστης παρουσιάζεται με τη δυνατότητα να το διαγράψει ολόκληρο. Πεδία που δέχονται και αποθηκεύουν πολλαπλές τιμές, σε αυτά τα αντικείμενα, επιτρέπουν μόνο τη διαγραφή τιμών που ορίζει ο χρήστης, αλλά δεν είναι δυνατόν να διαγραφούν ολόκληρα.

Η περίπτωση αυτή δε θεωρείται προβληματική, καθώς τα μόνα πεδία αυτού του τύπου ανήκουν στην κατηγορία αντικειμένων πληροφοριών εκτάκτου ανάγκης. Κεντρικό τμήμα αυτών των αντικειμένων

είναι ότι οι τιμές των πεδίων τους είναι εύκολα διαθέσιμες στο χρήστη μέσω της ειδοποίησης που λαμβάνει για αυτά. Επομένως, έχει πάντα τη δυνατότητα να δει τις τιμές ακριβώς όπως έχουν εισαχθεί, που σημαίνει ότι μπορεί εύκολα να τις καταθέσει για διαγραφή.

Σε περίπτωση που το πεδίο αδειάσει λόγω αυτών των εντολών, θα διαγραφεί και αυτό αυτόματα. Παρομοίως, αν όλα τα πεδία του αντικειμένου είναι σημειωμένα να διαγραφούν, θα διαγραφεί ολόκληρο το αντικείμενο.

Τελευταίο θέμα, της διαχείρισης διαγραφής και τροποποίησης δεδομένων στην εφαρμογή, είναι ο σύνδεσμος που υπάρχει μεταξύ μερικών κατηγοριών αντικειμένων. Ο σύνδεσμος αυτός υπάρχει λόγω του ιστορικού του χρήστη, το οποίο καταγράφει τα διάφορα αντικείμενα παθήσεων, φαρμάκων, αποτελεσμάτων εξετάσεων και εγχειρήσεων, καθώς και τα προγραμματισμένα ραντεβού κάθε είδους του χρήστη.

Αυτό δημιουργεί ένα πρόβλημα όμως στην τροποποίηση ή διαγραφή ενός αντικειμένου από αυτές τις κατηγορίες. Το ιστορικό του χρήστη δεν μπορεί να παραμείνει ίδιο μετά από μία αλλαγή σαν αυτή, αλλιώς οι πληροφορίες που προβάλλει θα παρουσιάζουν κάποιο σφάλμα. Το λογικό συμπέρασμα για την αντιμετώπιση αυτού του προβλήματος είναι ότι κάθε ενέργεια που επηρεάζει ένα αντικείμενο στις παρατηρούμενες κατηγορίες πρέπει να αντικατοπτρίζεται και στο αντίστοιχο αντικείμενο του ιστορικού.

Με την επιτυχία της τροποποίησης ή της διαγραφής αντικειμένου από τις παρατηρούμενες συλλογές, τίθεται σε εκκίνηση και μία ακόμα διαδικασία του ανάλογου τύπου ενέργειας. Οι διαδικασίες αυτές δέχονται ως παραμέτρους όλες τις σχετικές τιμές, που χρειάζονται για την αναφορά σε ένα αντικείμενο εκείνης της κατηγορίας, καθώς και αναγνωριστικά, για να διακρίνουν την κατηγορία του αντικειμένου. Με αυτές τις τιμές, αναζητούν στο ιστορικό για το ακριβές αντικείμενο, που πρέπει να αλλαχτεί και εκτελούν την κατάλληλη ενέργεια, είτε για την τροποποίηση των τιμών του είτε για τη διαγραφή του.

5.3 Επίλογος

Παρά τις δυσκολίες που συναντήθηκαν στην πορεία, η υλοποίηση της εφαρμογής αποδείχτηκε αρκετά ομαλή, κυρίως επειδή οι τεχνολογίες που χρησιμοποιήθηκαν για αυτό ήταν ικανές να συνεργαστούν με ευκολία. Το Visual Studio Code [31] παρείχε εκπληκτική ευελιξία στο πλήθος υπηρεσιών και πακέτων με τα οποία μπορούσε να συνεργαστεί. Παρομοίως, το Firebase [6] ενσωματώθηκε με ευκολία στην εφαρμογή, έχοντας ήδη έτοιμες μεθόδους εγκατάστασης.

Κύρια πρόκληση σε όλη τη διάρκεια της υλοποίησης της εφαρμογής ήταν ο καθαρός ορισμός της λογικής που θα ήταν δεμένη με τα στοιχεία της. Η κάθε υπηρεσία απαιτεί τις τοπικές λίστες της εφαρμογής για την ελαχιστοποίηση των επικοινωνιών με τη βάση [5], ενώ κάθε αλλαγή σε αυτές καθιστούσε απαραίτητη το άνοιγμα νέας επικοινωνίας.

Επιπλέον, η εφαρμογή παρέχει διάφορες υπηρεσίες που απαιτούν τον προγραμματισμό ενημερώσεων, ή άλλες λειτουργίες που απαιτούν την εφαρμογή να «θυμάται» κάποια πληροφορία, όπως οι αναφορές. Καθώς όμως δεν μπορεί να είναι σίγουρο ότι τοπικά αποθηκευμένες πληροφορίες θα παραμείνουν αποθηκευμένες, ή ότι η εφαρμογή δε θα διαγραφεί προσωρινά από το χρήστη, γίνεται απαραίτητο να εκτελούνται έλεγχοι στη βάση δεδομένων για τη σχετική πληροφορία. Αυτό προσθέτει έναν ακόμα βαθμό περιπλοκότητας στην υλοποίηση.

Προκλήσεις σαν αυτές, που βασίζονται περισσότερο στη λογική παρά στις τεχνολογίες διαθέσιμες, εμφανίζονται σε κάθε τομέα της ανάπτυξης της εφαρμογής. Όμως, με την κατάλληλη χρήση περιορισμών και λογικής, οι λειτουργίες αυτές υλοποιούνται με επιτυχία.

Κεφάλαιο 6ο: Αξιολόγηση της Εφαρμογής

6.1 Εισαγωγή

Πέρα από τη σχεδίαση και υλοποίησή της, είναι απαραίτητη και η σωστή αξιολόγηση της εφαρμογής. Αυτό γίνεται μέσω λεπτομερούς εξέτασης και πολλαπλών δοκιμών της εφαρμογής, για διάφορα σενάρια χρήσης και υποθέτοντας μικρή ή και ανύπαρκτη γνώση για τον αναμενόμενο τρόπο χρήσης, που οραματίστηκε ο δημιουργός της. Η εφαρμογή πρέπει να μπορεί να ικανοποιεί τους λειτουργικούς της στόχους μέσω μίας διεπαφής που μπορεί να χρησιμοποιηθεί με ευκολία από οποιονδήποτε.

6.2 Δοκιμές Εφαρμογής

6.2.1 Δοκιμές Λειτουργικότητας

Πρώτη σε σημασία από τις λειτουργίες της εφαρμογής είναι η χρήση της απομακρυσμένης βάσης δεδομένων, χωρίς την οποία δεν μπορεί να δουλέψει η υπόλοιπη δομή. Επομένως, η λειτουργικότητά της και οι λειτουργίες δεμένες σε αυτή, όπως η ταυτοποίηση του χρήστη, είναι το θέμα που πρέπει να αναφερθεί πριν από οτιδήποτε άλλο.

Κατά τις δοκιμές της εφαρμογής, η σύνδεση στη βάση δεδομένων [5] από τη συσκευή δεν παρουσίασε κάποιο πρόβλημα. Απόπειρες δημιουργίας λογαριασμού υπό διαφορετικά email συναντήθηκαν με επιτυχία, με το χρήστη να μπορεί να αποσυνδεθεί και να συνδεθεί ξανά με ευκολία [12]. Επιπλέον δοκιμές απέδειξαν ότι το ίδιο ίσχυε και για τη σύνδεση μέσω λογαριασμού Google, η οποία ολοκληρώθηκε με επιτυχία χωρίς παρατηρούμενες διαφορές στη συμπεριφορά της εφαρμογής ή στη λειτουργικότητα της βάσης [12].

Η ίδια η βάση λειτουργεί χωρίς πρόβλημα [5]. Δεδομένα που στέλνονται σε αυτήν αποθηκεύονται στις κατάλληλες συλλογές ως έγγραφα, με όλα τα απαραίτητα πεδία, των οποίων οι τιμές είναι συμπληρωμένες ακριβώς όπως προβλέπεται και αντιστοιχούν στις τιμές που εισήγαγε ο χρήστης στη συσκευή του. Παρομοίως, εντολές τροποποίησης ή και διαγραφής συγκεκριμένου εγγράφου εκτελέστηκαν δίχως σφάλμα και επηρεάζοντας μόνο το έγγραφο το οποίο «στόχευε» η εντολή, χωρίς να αλλάξουν άλλα έγγραφα στη βάση. Εντολές από ένα χρήστη δεν μπορούν να επηρεάσουν τα έγγραφα ενός άλλου.

Στο θέμα εισαγωγής δεδομένων στη βάση, η υπηρεσία κρυπτογράφησης των δεδομένων αποδείχτηκε επιτυχής [10]. Οι τιμές των πεδίων περνούν από κρυπτογράφηση με το αποθηκευμένο κλειδί [11] με επιτυχία, ανεξάρτητα από το αν η μεταβλητή του πεδίου είναι απλό κείμενο, λίστα ή χάρτης. Η επιστροφή των δεδομένων στην αναγνώσιμη μορφή τους, όταν διαβάζονται από τη συσκευή παρουσιάζει παρόμοια αποτελεσματικότητα, αποκρυπτογραφώντας τις τιμές χωρίς πρόβλημα ή σφάλμα [10].

Από την πλευρά της συσκευής, η ανάγνωση των δεδομένων από τη βάση και η παρουσίασή τους εκτελείται ικανοποιητικά ως προς την εκτέλεση της λειτουργίας τους. Τα διάφορα έγγραφα του χρήστη, και μόνο αυτού, μετατρέπονται στα αντίστοιχα είδη αντικειμένων και αποθηκεύονται σε τοπικά αρχεία χωρίς σφάλματα ή λανθασμένες τιμές. Όσο για την παρουσίαση αυτών των δεδομένων, ανεξάρτητα από το αν πρόκειται για λίστες, γραφήματα [21] ή ημερολόγιο [20], αυτά εμφανίζονται ξεκάθαρα και με όλες τις σχετικές πληροφορίες για το χρήστη, αν και πρέπει να αναφερθεί ότι τα γραφήματα [21]

παρουσιάζουν ένα σημαντικό περιορισμό στην υλοποίησή τους, που οδηγεί, στα περισσότερα από αυτά, να μην αντικατοπτρίζουν ικανοποιητικά το χρονικό διάστημα μεταξύ μετρήσεων.

Τα διάφορα στοιχεία UI της εφαρμογής ταιριάζουν στην οθόνη της συσκευής χωρίς πρόβλημα και εκτελούν όλες τις λειτουργίες για τις οποίες προορίζονται. Λίστες των οποίων τα αντικείμενα πρέπει να μπορούν να σύρονται στο πλάι για διαγραφή εφαρμόζονται με επιτυχία, ενώ πλαίσια και παράθυρα που πρέπει να εμφανίζονται μόνο μετά από δράση του χρήστη παραμένουν αόρατα μέχρι τότε και δημιουργούνται όταν έρχεται η κατάλληλη στιγμή, ακολουθώντας ακριβώς τη λογική του σχεδιασμού τους. Τα κουμπιά της κάθε οθόνης εμφανίζονται στις σωστές θέσεις και οι φόρμες της εφαρμογής διατηρούν τη δομή τους σύμφωνα με το σχέδιο. Όλα τα Widgets εμφανίζονται και λειτουργούν όπως θα έπρεπε.

Δοκιμές απέδειξαν επίσης ότι η εφαρμογή μπορεί να στείλει με επιτυχία ειδοποιήσεις [22]. Το σύστημα διαβάζει τα σχετικά δεδομένα, όπως τις πληροφορίες εκτάκτου ανάγκης, και στέλνει με επιτυχία ειδοποιήσεις στη συσκευή με σώμα ένα κείμενο που ενημερώνει το χρήστη για το είδος της πληροφορίας και τις σχετικές τιμές [22]. Η ειδοποίηση εμφανίζεται με αρκετά μεγάλο μέγεθος για να συμπεριλάβει το περιεχόμενο του σώματός του.

Τέλος, η εφαρμογή μπορεί να επιλέξει αρχεία PDF από τη συσκευή [24] και να δημιουργήσει καινούργια για αυτήν [23]. Η διαδικασία επιλογής αρχείου [24] δεν παρουσιάζει σφάλμα, ενώ το διάβασμα του κειμένου του ολοκληρώνεται επίσης με επιτυχία [25]. Όμως, το γράψιμο καινούργιου κειμένου για ένα δημιουργημένο αρχείο παρουσιάζει μία έλλειψη στην εμφάνισή του, αν και παραμένει αρκετά λειτουργικό ώστε να εκπληρώνει το σκοπό του [23].

6.2.2 Δοκιμές Χρηστικότητας

Πέρα από το αν η εφαρμογή λειτουργεί, είναι σημαντικό να είναι και εύκολη στη χρήση της. Στον τομέα αυτό, δυστυχώς, η εφαρμογή παρουσιάζει μερικά προβλήματα. Αν και η δομή του είναι αρκετά απλή, ώστε ένας χρήστης να καταλάβει πώς να χρησιμοποιήσει την εφαρμογή μετά από λίγες δοκιμές και τα δεδομένα παρουσιάζονται όντως με τρόπο ξεκάθαρο, ως προς την πληροφορία που μεταφέρουν, ένας νέος χρήστης θα δυσκολευτεί στην πλοήγησή του στην εφαρμογή.

Για τις οθόνες εισαγωγής δεδομένων, καθώς και για μερικές από τις οθόνες τροποποίησης, που δε χρησιμοποιούν λίστες για την προβολή των αντικειμένων τους, η βασική δομή για την εισαγωγή τιμών είναι η χρήση πλαισίων εισαγωγής κειμένου σε συνδυασμό με ατομικά κουμπιά για την αποθήκευση της αντίστοιχης τιμής, με ένα επιπλέον κουμπί για την κατάθεση των τιμών. Αυτή η ιδέα είναι αρκετά απλή, αλλά παρουσιάζει ένα πρόβλημα στο ότι δεν ακολουθεί τη συνηθισμένη δομή που περιμένει ο μέσος χρήστης. Η διαφορά αυτή, στην αναμενόμενη ροή ενεργειών για την ολοκλήρωση της εισαγωγής δεδομένων, μπορεί να οδηγήσει σε μερικές εσφαλμένες πρώτες απόπειρες προτού ο χρήστης συνειδητοποιήσει την μέθοδο που αναμένεται να ακολουθήσει.

Η περίοδος δοκιμών όμως παραμένει σύντομη χάρη στις μπάρες ενημερώσεων που χρησιμοποιεί η εφαρμογή. Με αυτές, ο χρήστης λαμβάνει επιβεβαίωση για την επιτυχή ολοκλήρωση μίας διαδικασίας, ώστε να του είναι προφανές τότε η απόπειρά του ήταν σωστά διατυπωμένη.

Παρόμοιο πρόβλημα εμφανίζει η οθόνη ημερολογίου. Η έλλειψη επεξηγηματικού υπομνήματος για τη χρήση χρωμάτων, στις κατηγορίες ραντεβού, αποτελεί πρόβλημα για το χρήστη και ελάττωμα στην υλοποίηση της εφαρμογής. Αν και ο σκοπός των χρωμάτων μπορεί να γίνει κατανοητός από το χρήστη αρκετά γρήγορα, η εφαρμογή δεν παρέχει στοιχείο που να επιβεβαιώνει τα συμπεράσματά του. Επομένως, η ακριβής λειτουργία των χρωμάτων δεν είναι όσο ξεκάθαρη θα έπρεπε.

Με εξαίρεση τα παραπάνω, το σύστημα που υλοποιείται ικανοποιεί κυρίως την ανάγκη για ευκολία χρήσης στις λειτουργίες του. Οι διάφορες επιλογές οθονών στις οποίες μπορεί να μεταβεί ο χρήστης είναι ξεκάθαρα ορισμένες στην οθόνη ιατρικών, όπως είναι και οι κατηγορίες αντικειμένων για εισαγωγή, τροποποίηση και διαγραφή. Παρομοίως, η κάθε οθόνη της εφαρμογής έχει ξεκάθαρα σημειωμένα κουμπιά με σωστή ονομασία για την κάθε λειτουργία, καθιστώντας τη χρήση αρκετά απλή, για να γίνεται γρήγορα κατανοητή από το χρήστη.

Τα κουμπιά επιστροφής σε προηγούμενη σελίδα είναι ξεκάθαρα σημειωμένα στο AppBar της κάθε οθόνης-στόχου, χρησιμοποιώντας τα τυπικά εικονίδια που θα περίμενε ο χρήστης. Επιπλέον, το κουμπί αποσύνδεσης χρήστη από το λογαριασμό του, τοποθετημένο στο ίδιο σημείο στην οθόνη ιατρικών όπως το κουμπί επιστροφής στις υπόλοιπες οθόνες, είναι επίσης προφανές στο χρήστη, καθώς χρησιμοποιεί το γνωστό εικονίδιο της εξόδου που υπάρχει σε πολλές άλλες εφαρμογές.

Μοναδικό πρόβλημα στην πλοήγηση αποτελεί το ότι η απαίτηση του συστήματος για το χρήστη να χρησιμοποιήσει αυτά τα κουμπιά για την επιστροφή του στην προηγούμενη οθόνη, καθώς η δομή της εφαρμογής, ορίζει ότι υπάρχει μόνο μία οθόνη στη στοίβα κάθε φορά. Αυτό σημαίνει ότι, αν ο χρήστης χρησιμοποιήσει το κουμπί επιστροφής της συσκευής του, η εφαρμογή θα ελαχιστοποιηθεί, το οποίο είναι σχεδόν σίγουρο να συμβεί τουλάχιστον μία φορά, έστω και από συνήθεια.

6.3 Αξιολόγηση απόδοσης

Στο θέμα της απόδοσης, η εφαρμογή έχει μεικτά αποτελέσματα. Τα στοιχεία της κάθε οθόνης, όπως έχει ήδη αναφερθεί, εμφανίζονται καθαρά και χωρίς σφάλματα στην οθόνη. Οι λίστες παρουσιάζονται χωρίς να καταλαμβάνουν περισσότερο κατακόρυφο χώρο από όσο χρειάζεται και επιτρέπουν στο χρήστη να περιηγηθεί προς τα κάτω στο πλαίσίό τους εάν δεν χωράνε όλα τα αντικείμενα σε αυτό. Τα πλαίσια εισαγωγής κειμένου και τα κείμενα που παρουσιάζονται έχουν όρια στο πλήθος χαρακτήρων που δέχονται, αποτρέποντας περιπτώσεις όπου «ξεχειλίζουν» από αυτά. Περιορισμοί και ικανός σχεδιασμός βεβαιώνουν ότι κανένα στοιχείο δεν παραλείπεται ή εξαφανίζεται χωρίς λόγο.

Επιπλέον, η κίνηση αυτών των στοιχείων στη συσκευή είναι και αυτή σταθερή και γρήγορη. Τα αντικείμενα στις λίστες σέρνονται στο πλάι και διαγράφονται ομαλά, χωρίς κολλήματα και επαναφέρονται εξίσου εύκολα. Οι ενημερώσεις εμφανίζονται και εξαφανίζονται με την ίδια άνεση, φεύγοντας από την οθόνη όταν περάσει ο ορισμένος τους χρόνος. Μέχρι και η αλλαγή μεταξύ ενεργών οθονών ολοκληρώνεται χωρίς αισθητή καθυστέρηση από το χρήστη. Σε κάθε τομέα που έχει να κάνει με αυτό, η οπτική της εφαρμογής είναι σίγουρα ικανοποιητική.

Η ταχύτητα απόκρισής της είναι επίσης ικανοποιητική. Εισαγόμενο κείμενο εμφανίζεται αμέσως και η κάθε λειτουργία εκτελείται αμέσως αφού ο χρήστης πατήσει το αντίστοιχο κουμπί. Μοναδική περίπτωση που αντιμετωπίζεται μια καθυστέρηση βρίσκεται σε περιπτώσεις όπου χρησιμοποιείται σύστημα εκτός του κώδικα της εφαρμογής, όπως όταν η εφαρμογή προσπαθεί να διαβάσει αρχείο αποθηκευμένο στη συσκευή. Η καθυστέρηση αυτή όμως είναι αναμενόμενο και τυπικό πρόβλημα σε μία εφαρμογή Flutter.

Το πρόβλημα εμφανίζεται στην ταχύτητα της εφαρμογής. Η κάθε επικοινωνία με τη βάση απαιτεί χρόνο για να ολοκληρωθεί με επιτυχία, απαιτώντας υπομονή από το χρήστη προτού μπορέσει να πάρει επιβεβαίωση για κάθε εισαγωγή που εκτελεί. Παρομοίως, ο χρόνος που χρειάζεται στην εκκίνηση για να ολοκληρώσει την ενημέρωση των τοπικών προσωρινών αρχείων σημαίνει ότι μερικές λειτουργίες που βασίζονται στο να γνωρίζουν το περιεχόμενο της βάσης, όπως οι οθόνες τροποποίησης, πρακτικά δεν λειτουργούν για ένα μικρό χρονικό διάστημα. Η αναμονή δεν είναι μεγάλη, αλλά καθιστά την

ταχύτητα της εφαρμογής, όσον αφορά την επικοινωνία με τη βάση και τις λειτουργίες βασισμένες σε αυτό, μη ικανοποιητική.

6.4 Συζήτηση αποτελεσμάτων

Μετά από ανάλυση των αποτελεσμάτων των δοκιμών της, η εφαρμογή θα μπορούσε να θεωρηθεί κυρίως ικανοποιητική. Εκπληρώνει την κεντρική της λειτουργία, την αποθήκευση, παρακολούθηση και διαχείριση ιατρικών δεδομένων, παρά τα μειονεκτήματά της, παρέχοντας εύκολα κατανοητές μεθόδους παρουσίασης και διαχείρισης αυτών.

Το γεγονός ότι η δομή των διαδικασιών εισαγωγής της διαφέρει από αυτή που περιμένει ο χρήστης μπορεί όντως να δημιουργήσει αρχικά προβλήματα στην χρήση της, αλλά όχι σε βαθμό που περιορίζει τη λειτουργικότητά της. Αυτό όμως αποτελεί απλά ένα εμπόδιο στα αρχικά στάδια της χρήσης και μπορεί να ξεπεραστεί γρήγορα από το χρήστη, καθώς διατηρεί ένα απλό σύστημα με ξεκάθαρα λειτουργίες και ενημερώσεις για την επιτυχία ή αποτυχία κάθε ενέργειας.

Παρομοίως, οι καθυστερήσεις που χρειάζεται για κάθε διαδικασία σύνδεσης με τη βάση δεδομένων δημιουργεί μια απαίτηση για υπομονή από το χρήστη, αλλά τίποτα παραπάνω. Ο χρήστης παραμένει ενήμερος για την επιτυχία ή αποτυχία της ολοκλήρωσής τους χάρη στις ενημερώσεις της εφαρμογής. Επιπλέον, αν και υπάρχει καθυστέρηση για κάθε επικοινωνία με τη βάση, ο χρόνος που παίρνουν να ολοκληρωθούν δεν είναι τόσος ώστε να επηρεάσει δραστικά την εμπειρία του χρήστη.

Κύριοι τομείς, στους οποίους απαιτείται μελλοντική βελτίωση, είναι τα γραφήματα [21] και η υλοποίηση της οθόνης ημερολογίου. Στον πρώτο τομέα, η υλοποίησή του παρουσιάζει περιορισμένη λειτουργικότητα, καθώς δεν αντικατοπτρίζει ικανοποιητικά το χρονικό διάστημα μεταξύ μετρήσεων, για τις περισσότερες κατηγορίες μετρήσεων που απεικονίζει. Ο περιορισμός αυτός δεν επηρεάζει όλη τη λειτουργικότητα των γραφημάτων, δηλαδή δεν αφορά την τοποθέτηση των ίδιων των μετρήσεων στον κατακόρυφο άξονα, μονάχα την απόσταση μεταξύ τους στον οριζόντιο άξονα.

Στην περίπτωση του δεύτερου τομέα, η έλλειψη επεξηγηματικού υπομνήματος για τη χρήση χρωμάτων, στην οθόνη ημερολογίου, αποτελεί ένα ελάττωμα στο σχεδιασμό της οθόνης. Αν και η λειτουργικότητά της δεν επηρεάζεται απευθείας από αυτή την έλλειψη, η χρησιμότητά της περιορίζεται, καθώς ο χρήστης δεν πληροφορείται για το σκοπό των χρωμάτων που χρησιμοποιούνται σε αυτή την οθόνη.

Επομένως, σύμφωνα με αυτή τη διατύπωση των αποτελεσμάτων, η εφαρμογή είναι κυρίως λειτουργική, αν και όχι πλήρως εύχρηστη.

6.5 Επίλογος

Σε γενικές γραμμές, η εφαρμογή μπορεί να θεωρηθεί ότι ικανοποιεί τους στόχους της. Η λειτουργικότητά της, μετά από δοκιμές, αποδεικνύεται αρκετή για το έργο της, ενώ η χρησιμότητά της απαιτεί μόνο λίγο χρόνο για να πειραματιστεί και να μάθει να τη χρησιμοποιεί ο χρήστης, καθώς και λίγη υπομονή για όποτε εκτελεί κάποια εντολή που οδηγεί σε επικοινωνία με τη βάση δεδομένων. Με λίγα λόγια, είναι αρκετά καλή αλλά έχει περιθώριο για βελτίωση.

Κεφάλαιο 7ο: Συμπεράσματα και Μελλοντικές Επεκτάσεις

Έχοντας δει όλα τα μέρη της εφαρμογής, από τις τεχνολογίες που χρησιμοποιήθηκαν στην ανάπτυξή της μέχρι και τα αποτελέσματα των δοκιμών της, σχηματίζεται μία καθαρή εικόνα για αυτήν. Η δομή της είναι απλή και βασίζεται σε μια υποδομή γνωστή για την ευελιξία της στο τι μπορεί να ενσωματώσει, αλλά η απλότητα αυτή διαφέρει από αυτό που περιμένει ο μέσος χρήστης. Διαθέτει τη λειτουργικότητα να ικανοποιήσει τις ανάγκες της, αλλά παρουσιάζει χαμηλή ταχύτητα σε αυτό και δεν είναι όσο εύχρηστη όσο θα μπορούσε να είναι. Δηλαδή, είναι λειτουργική, αλλά όχι ακριβώς ικανοποιητική, λόγω των περιορισμών αυτών.

Για μελλοντικές επεκτάσεις και βελτιώσεις, πέρα από την αντιμετώπιση των περιορισμών που αναφέρθηκαν ήδη, η κύρια ιδέα θα ήταν η διασύνδεση της εφαρμογής με συστήματα υγείας που χρησιμοποιούνται σε νοσοκομεία και άλλα ιατρικά ινστιτούτα. Η ικανότητα να στέλνονται ιατρικά δεδομένα εξ αποστάσεως στον αρμόδιο ειδικό, ανάλογα με τις ανάγκες της κατάστασης, θα ήταν ιδιαίτερα χρήσιμη. Η επέκταση αυτή θα μπορούσε να φτάσει και στο σημείο όπου ο χρήστης θα μπορούσε να κλείσει ραντεβού χρησιμοποιώντας μόνο την εφαρμογή, αντί απλά να καταγράφει τα ραντεβού του σε αυτήν.

Πέρα από αυτό, το σύστημα της ίδιας της εφαρμογής θα μπορούσε να επεκταθεί ώστε να ξεχωρίζει μεταξύ γιατρών και ασθενών για τον απευθείας ορισμό αποτελεσμάτων μέσω μόνο αυτού. Ο γιατρός θα μπορούσε να καταθέσει έτσι συνταγές ή αποτελέσματα εξετάσεων στη βάση δεδομένων και ο αντίστοιχος ασθενής του θα μπορούσε να λάβει ενημέρωση για αυτά με την ολοκλήρωση της αποθήκευσης.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Internet Site

- [1] Health Level Seven International (HL7), “HL7 Standard V2: Application Protocol for Electronic Data Exchange in Healthcare Environments,” *HL7 Standard*, 2003. [Online]. Internet: https://www.hl7.org/implement/standards/product_brief.cfm?product_id=185 [27 Φεβρουαρίου 2026].
- [2] Health Level Seven International (HL7), “FHIR R4 (HL7 Fast Healthcare Interoperability Resources),” *HL7 Standard*, 2019. [Online]. Internet: https://www.hl7.org/implement/standards/product_brief.cfm?product_id=491 [27 Φεβρουαρίου 2026].
- [3] fhirfli.dev (2026, Ιανουάριος). “fhir_r4”. [Online]. fhir_r4 0.4.4. Internet: https://pub.dev/packages/fhir_r4 [6 Μαρτίου 2026].
- [4] Ελληνική Δημοκρατία, Υπουργείο Ελληνικής Κυβέρνησης. (2021, Αύγουστος). “MyHealth.”. [Online]. Internet: <https://play.google.com/store/apps/details?id=gr.gov.myhealth&hl=el&pli=1> [1 Μαΐου 2026].
- [5] Cloud Firestore, Google LLC. (2019, Ιανουάριος). “Cloud Firestore.”. *Firebase Documentation*. [Online]. Internet: <https://firebase.google.com/docs/firestore> [30 Ιανουαρίου 2026].
- [6] Firebase, Google LLC. (2012, Απρίλιος). “Firebase,” *Firebase Documentation*. [Online]. Available: <https://firebase.google.com> [23 Ιανουαρίου 2026].
- [7] Cloud Firestore Flutter Πακέτο, firebase.google.com. [2026, Ιανουάριος]. “cloud_firestore”. [Online]. cloud_firestore 6.1.2. Internet: https://pub.dev/packages/cloud_firestore [30 Ιανουαρίου 2026].
- [8] Shared Preferences Flutter Πακέτο, flutter.dev. (2025, Δεκέμβριος) “Shared preferences plugin”. [Online]. shared_preferences 2.5.4. Internet: https://pub.dev/packages/shared_preferences [23 Ιανουαρίου 2026].
- [9] Google Health Flutter Package, carp.dk. (2026, Φεβρουάριος). “Health”. [Online]. health 13.3.1. Internet: <https://pub.dev/packages/health> [6 Μαρτίου 2026].
- [10] AES-256 Flutter Πακέτο, knottx.dev. (2025, Νοέμβριος). “AES256”. [Online]. aes256 2.2.2. [Online]. Internet: <https://pub.dev/packages/aes256> [6 Μαρτίου 2026].
- [11] Πακέτο Flutter DOTENV. (2025, Αύγουστος). “flutter_dotenv”. [Online]. flutter_dotenv 6.0.0. Internet: https://pub.dev/packages/flutter_dotenv [6 Μαρτίου 2026].
- [12] Firebase Authentication. Google LLC. (2014, Οκτώβριος). “Firebase Authentication”. [Online]. *Firebase Documentation*. Internet: <https://firebase.google.com/docs/auth> [23 Ιανουαρίου 2026].
- [13] Firebase UI Authentication Flutter Πακέτο, firebase.google.com. (2025, Οκτώβριος). “firebase_ui_auth”. [Online]. firebase_ui_auth 3.0.1. Internet: https://pub.dev/packages/firebase_ui_auth [23 Ιανουαρίου 2026].
- [14] Firebase UI OAuth Google Flutter Πακέτο, firebase.google.com. (2025, Οκτώβριος). “firebase_ui_oauth_google”. [Online]. firebase_ui_oauth_google 2.0.1. Internet: https://pub.dev/packages/firebase_ui_oauth_google [6 Μαρτίου 2026].

- [15] Firebase Authentication Flutter Πακέτο, firebase.google.com. (2026, Ιανουάριος). “Firebase Auth for Flutter”. [Online]. firebase_auth 6.1.4. Internet: https://pub.dev/packages/firebase_auth [23 Ιανουαρίου 2026].
- [16] Local Authentication Flutter Πακέτο, flutter.dev. (2026, Ιανουάριος). “local_auth”. 2026. [Online]. local_auth 3.0.1. Internet: https://pub.dev/packages/local_auth [23 Ιανουαρίου 2026].
- [17] Flutter, Google LLC. (2017, Μάιος). “Build for any screen”. [Online]. Internet: <https://flutter.dev> [7 Νοεμβρίου 2025].
- [18] Firebase CLI, Google LLC. (2026, Ιανουάριος). “Firebase CLI reference”. [Online]. Internet: <https://firebase.google.com/docs/cli> [23 Ιανουαρίου 2026].
- [19] Firebase Core Flutter Πακέτο, firebase.google.com. (2026, Ιανουάριος). “Firebase Core for Flutter”. [Online]. firebase_core 4,4,0. Internet: https://pub.dev/packages/firebase_core [23 Ιανουαρίου 2026].
- [20] Ημερολόγιο Flutter Πακέτο. (2025, Ιανουάριος). “TableCalendar”. [Online]. table_calendar 3.2.0. Internet: https://pub.dev/packages/table_calendar [13 Φεβρουαρίου 2026].
- [21] Γραφήματα Flutter Πακέτο, flchart.dev. (2025, Σεπτέμβριος). “FL Chart, Free and Open Source”. [Online]. fl_chart 1.1.1.: https://pub.dev/packages/fl_chart [20 Φεβρουαρίου 2026].
- [22] Τοπικές ενημερώσεις Flutter Πακέτο, dexterx.dev. (2026, Ιανουάριος). “flutter_local_notifications”. [Online]. flutter_local_notifications 20.0.0. Internet: https://pub.dev/packages/flutter_local_notifications [13 Φεβρουαρίου 2026].
- [23] Δημιουργία PDF Flutter Πακέτο, nfet.net. (2025, Φεβρουάριος). “Pdf creation library for dart/flutter”. [Online]. pdf 3.11.3. Internet: <https://pub.dev/packages/pdf> [6 Μαρτίου 2026].
- [24] Επιλογή αρχείου Flutter Πακέτο, miguelrui.com. (2026, Ιανουάριος). “File Picker”. [Online]. file_picker 10.3.10. Internet: https://pub.dev/packages/file_picker [6 Μαρτίου 2026].
- [25] Ανάγνωση κειμένου αρχείου PDF Flutter Πακέτο, fabi.online. (2025, Φεβρουάριος). “PDF Text Plugin”. [Online]. flutter_pdf_text 0.9.0. Internet: https://pub.dev/packages/flutter_pdf_text [6 Μαρτίου 2026].
- [26] Διαχείριση Αδειών Flutter Πακέτο, baseflow.com. (2025, Ιούλιος). “permission_handler”. [Online]. permission_handler 12.0.1. Internet: https://pub.dev/packages/permission_handler [6 Μαρτίου 2026].
- [27] Πληροφορίες Συσκευής Flutter Πακέτο, fluttercommunity.dev. (2026, Μάρτιος). “device_info_plus”. [Online]. device_info_plus 12.4.0. Internet: https://pub.dev/packages/device_info_plus [6 Μαρτίου 2026].
- [28] Πάροχος διαδρομής Flutter Πακέτο, flutter.dev. (2024, Ιούλιος). “path_provider”. [Online]. path_provider 2.1.5. Internet: https://pub.dev/packages/path_provider [6 Μαρτίου 2026].
- [29] Workmanager Flutter Πακέτο, fluttercommunity.dev. (2025, Αύγουστος). “Flutter Workmanager”. [Online]. workmanager 0.9.0+3. Internet: <https://pub.dev/packages/workmanager> [6 Μαρτίου 2026].
- [30] Intl Flutter Πακέτο, dart.dev. (2025, Ιανουάριος). “intl”. [Online]. intl 0.20.2. Internet: <https://pub.dev/packages/intl> [23 Ιανουαρίου 2026].

- [31] Visual Studio Code, Microsoft Corporation. (2016, Απρίλιος). “Visual Studio Code documentation”. [Online]. Internet: <https://code.visualstudio.com/docs> [7 Νοεμβρίου 2025].
- [32] Visual Studio Code Extension, Dart Code. (2016, Αύγουστος). “Dart”. [Online]. Internet: <https://marketplace.visualstudio.com/items?itemName=Dart-Code.dart-code> [7 Νοεμβρίου 2025].
- [33] Visual Studio Code Extension, Dart Code. (2018, Απρίλιος). “Flutter”. [Online]. Internet: <https://marketplace.visualstudio.com/items?itemName=Dart-Code.flutter> [7 Νοεμβρίου 2025].
- [34] Android Studio, Google LLC. (2013, Μάιος). “Android Studio”. [Online]. Internet: <https://developer.android.com/studio> [7 Νοεμβρίου 2025].
- [35] GitHub Repository, flchart.dev. (2023, Νοέμβριος) “imaNNeo/fl_chart”. [Online]. Internet: https://github.com/imaNNeo/fl_chart/blob/main/example/lib/main.dart [20 Φεβρουαρίου 2026].