

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ – ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ****Πρόγραμμα Μεταπτυχιακών Σπουδών****«Πληροφορική»****Μεταπτυχιακή Διατριβή**

Τίτλος Διατριβής	Εφαρμογή Android «My Doctor» για ηλεκτρονικά ραντεβού με γιατρό υλοποιημένη με Android Studio και Firebase Android application 'My Doctor' for online doctor appointments, developed using Android Studio and Firebase
Όνοματεπώνυμο Φοιτητή	Δρούκα Παναγιώτα
Πατρώνυμο	Ιωάννης
Αριθμός Μητρώου	ΜΠΠΛ20016
Επιβλέπων	Ευθύμιος Αλέπης, Καθηγητής

Ημερομηνία Παράδοσης **Νοέμβριος 2024**

Τριμελής Εξεταστική Επιτροπή

Ευθύμιος Αλέπης,
Καθηγητής

Μαρία Βίρβου
Καθηγήτρια

Διονύσιος Σωτηρόπουλος
Επίκουρος Καθηγητής

Περιεχόμενα

Περίληψη.....	5
Κεφάλαιο 1: Εισαγωγή.....	6
1.1 Περιγραφή του προβλήματος	6
1.2 Παρόμοιες εφαρμογές στην αγορά.....	7
1.2.1 Doctor Anytime.....	7
1.2.2 finddoctors.gov.gr (ΗΔΙΚΑ)	7
1.2.3 AnyDoctor	8
1.2.4 Iatrica.....	8
1.2.5 Πλατφόρμα 1535.gr.....	9
1.2.6 Εφαρμογές ιατρικών κέντρων (π.χ., ιδιωτικών νοσοκομείων):	9
1.2.7 Viva.gr (Viva Health).....	9
1.3 Σκοπός και στόχοι.....	10
Κεφάλαιο 2: Screenshots και Περιγραφή Λειτουργικότητας ...	12
2.1 Ασθενής (user view)	12
2.1.1 Αρχική σελίδα / Login Page	12
2.1.2 Δημιουργία νέου λογαριασμού	12
2.1.4 Προβολή σελίδας γιατρού / Κλείσιμο ραντεβού	18
2.1.5 Μετάβαση στο προφίλ του ασθενή.....	20
2.2 Γιατρός (doctor view)	22
2.2.1 Αρχική σελίδα / Login Page	22
2.2.2 Τροποποίηση στοιχείων	23
2.2.3 Διαχείριση διαθεσιμότητας	28
2.2.4 Προβολή διαθεσιμότητας	34
2.2.5 Προβολή επερχόμενων ραντεβού	36
2.2.5 Προβολή παρελθοντικών ραντεβού.....	37
2.2.5 Προβολή στατιστικών	38
Κεφάλαιο 3: Αρχιτεκτονική Εφαρμογής.....	40
3.1 Γλώσσα προγραμματισμού: Java.....	40
3.2 Βάση δεδομένων: Firebase Realtime Database	40
3.2.1 Λόγοι επιλογής της Firebase Realtime Database	40

3.2.2 Δομή της βάσης δεδομένων.....	40
3.2.3 Χρήση και Ενημέρωση Δεδομένων σε Πραγματικό Χρόνο.....	41
3.2.4 Ασφάλεια και Authentication	41
3.3 Δομή εφαρμογής	42
3.3.1 Activity-based UI Design.....	42
3.3.2 Αρχιτεκτονική MVC.....	42
3.3.3 Επικοινωνία Front-end με Firebase.....	43
3.4 Διαγράμματα	45
3.4.1 Διάγραμμα περίπτωσης χρήσης ασθενή	45
3.4.2 Διάγραμμα περίπτωσης χρήσης γιατρού.....	47
Κεφάλαιο 4: Συμπεράσματα και Μελλοντικές Επεκτάσεις.....	48
Κεφάλαιο 5: Βιβλιογραφία	50

Περίληψη

Η παρούσα εργασία αφορά την ανάπτυξη μίας Android εφαρμογής «My Doctor» για εύρεση και προγραμματισμό ιατρικών ραντεβού ηλεκτρονικά, η οποία επιτρέπει σε ασθενείς να κλείνουν ραντεβού με γιατρούς βάσει ειδικότητας και γεωγραφικής τοποθεσίας. Οι γιατροί έχουν τη δυνατότητα να διαχειρίζονται τη διαθεσιμότητά τους, να βλέπουν τα προγραμματισμένα ραντεβού τους, να ακυρώνουν ραντεβού και να εξετάζουν στατιστικά στοιχεία όπως η εβδομάδα του μήνα με τα περισσότερα ραντεβού και η πιο δημοφιλής ώρα για ραντεβού. Η εφαρμογή αναπτύχθηκε χρησιμοποιώντας Java για το front-end και Firebase Realtime Database για τη διαχείριση των δεδομένων, προσφέροντας συγχρονισμό σε πραγματικό χρόνο.

Σκοπός της εφαρμογής «My Doctor» είναι να απλοποιήσει τη διαδικασία εύρεσης και προγραμματισμού ραντεβού, συμβάλλοντας στην αποτελεσματικότερη διαχείριση χρόνου τόσο για τους ασθενείς όσο και για τους γιατρούς.

Η εργασία αποτελείται από 4 βασικά μέρη. Στο πρώτο μέρος, γίνεται μια εισαγωγή για την εφαρμογή, περιγραφή του προβλήματος, αναφορά σε παρόμοιες εφαρμογές στην αγορά και αξιολόγησή τους, και αναφορά στον σκοπό και στόχους της συγκεκριμένης εφαρμογής «My Doctor». Στο δεύτερο μέρος, γίνεται παρουσίαση και περιγραφή της λειτουργικότητας της εφαρμογής. Στο τρίτο μέρος, γίνεται μια σύντομη αναφορά στην αρχιτεκτονική της εφαρμογής, και στο τέταρτο κεφάλαιο, αναφέρονται τα συμπεράσματα και σκέψεις για μελλοντικές επεκτάσεις.

Abstract

This thesis focuses on the development of an Android application called "My Doctor" for finding and scheduling medical appointments online. The application allows patients to book appointments with doctors based on their specialty and geographical location. Doctors can manage their availability, view scheduled appointments, cancel appointments, and analyze statistics such as the busiest week of the month and the most popular appointment hour. The app is built using Java for the front-end and Firebase Realtime Database for data management, providing real-time synchronization.

The purpose of "My Doctor" is to simplify the process of finding and scheduling appointments, enhancing time management for both patients and doctors.

The thesis consists of four main sections. The first section introduces the application, describes the problem, discusses similar applications available in the market and evaluates them, and explains the purpose and goals of "My Doctor." The second section presents and describes the application's functionality. The third section provides a brief overview of the application's architecture, and the fourth section concludes with findings and suggestions for future extensions.

Κεφάλαιο 1: Εισαγωγή

1.1 Περιγραφή του προβλήματος

Η εύρεση γιατρών και ο προγραμματισμός ραντεβού παραμένουν καθημερινές προκλήσεις για πολλούς ασθενείς. Παρότι υπάρχουν διαθέσιμες ιστοσελίδες και εφαρμογές που προσφέρουν την δυνατότητα να κλείσεις online ραντεβού, οι ανάγκες της σύγχρονης κοινωνίας και η ταχύτητα εξέλιξης της τεχνολογίας απαιτούν νέες, πιο αποτελεσματικές λύσεις στη διαχείριση υπηρεσιών υγείας.

Ένα μεγάλο ποσοστό του πληθυσμού στην Ελλάδα πλέον χρησιμοποιεί smartphones, προτιμώντας τις κινητές συσκευές για καθημερινές εργασίες αντί για υπολογιστές. Συγκεκριμένα, σύμφωνα με έρευνα του PEW Research Center (2022), το 95% των ατόμων ηλικίας 18-34 διαθέτει smartphone, ενώ το ποσοστό στις ηλικίες άνω των 50 ετών, αν και αρκετά χαμηλότερο (29%), αυξάνεται σταθερά. Η εξοικείωση με τις κινητές συσκευές, σε συνδυασμό με την ευκολία και την ταχύτητα που προσφέρουν οι εφαρμογές, αναδεικνύει τη δυναμική τους ως εργαλείων για την οργάνωση ραντεβού, προσφέροντας απλούστερη και πιο φιλική εμπειρία χρήσης σε σχέση με τις ιστοσελίδες.

Οι εφαρμογές για smartphones παρέχουν πολλά πλεονεκτήματα, όπως:

- Ευκολότερη πρόσβαση: Οι κωδικοί σύνδεσης μπορούν να αποθηκευτούν στη συσκευή, διευκολύνοντας τη σύνδεση και μειώνοντας την ανάγκη για επαναλαμβανόμενη εισαγωγή στοιχείων.
- Διατήρηση ιστορικού: Οι χρήστες μπορούν να βλέπουν τα προηγούμενα και μελλοντικά ραντεβού τους, καθώς και σημαντικές πληροφορίες που σχετίζονται με αυτά.
- Ταχύτητα: Οι εφαρμογές συχνά έχουν ταχύτερη απόκριση σε σχέση με τις ιστοσελίδες σελίδα, μέσω χρήσης τοπικής αποθήκευσης δεδομένων (cache) και είναι βελτιστοποιημένες για τη χρήση στις συσκευές όπου είναι εγκατεστημένες.
- Πρόσθετες λειτουργίες: Μια εφαρμογή μπορεί να στέλνει υπενθυμίσεις για επερχόμενα ραντεβού μέσω ειδοποιήσεων (push notifications), ενισχύοντας τη λειτουργικότητα και την εμπειρία χρήστη.

Τέτοιες δυνατότητες είναι ιδιαίτερα χρήσιμες για:

- Άτομα με περιορισμένο χρόνο, όπως επαγγελματίες, που χρειάζονται γρήγορη πρόσβαση σε γιατρό.
- Επείγουσες καταστάσεις που απαιτούν άμεση εύρεση διαθέσιμου γιατρού.
- Ηλικιωμένους ή τεχνολογικά λιγότερο εξοικειωμένους χρήστες, για τους οποίους μπορεί η χρήση μιας εφαρμογής να είναι πιο απλή και κατανοητή από την περιήγηση σε ιστοσελίδες.

Οι παραδοσιακές μέθοδοι, όπως η τηλεφωνική κράτηση ραντεβού, συχνά απαιτούν περισσότερο χρόνο, ενώ μπορεί να οδηγήσουν σε λάθη ή παρανοήσεις. Επιπλέον, οι χρήστες δεν έχουν τη δυνατότητα να διατηρούν ιστορικό των ραντεβού τους ή να βλέπουν άμεσα διαθεσιμότητα σε πραγματικό χρόνο. Αντίθετα, μια εφαρμογή αυτοματοποιεί τη διαδικασία, μειώνει τις πιθανότητες σφαλμάτων και παρέχει πρόσβαση σε σημαντικές πληροφορίες ανά πάσα στιγμή.

Επιπλέον, ενώ πολλές υπάρχουσες λύσεις επικεντρώνονται κατά βάση στις ανάγκες των ασθενών, παρατηρείται έλλειψη εργαλείων που εξυπηρετούν εξίσου τους γιατρούς. Μια εξειδικευμένη εφαρμογή μπορεί να προσφέρει διαχείριση διαθεσιμότητας, αυτοματοποιημένη οργάνωση ραντεβού και στατιστικά δεδομένα δραστηριότητας.

Οι εφαρμογές υγείας αποτελούν κομμάτι ενός ευρύτερου κινήματος ψηφιοποίησης των υπηρεσιών. Σε αυτό το πλαίσιο, η συγκεκριμένη εφαρμογή προσαρμόζεται στις ανάγκες των σύγχρονων χρηστών, προσφέροντας όχι μόνο ευκολία αλλά και μια βιώσιμη λύση για τη βελτίωση της προσβασιμότητας στις υπηρεσίες υγείας. Έχει σχεδιαστεί για να είναι φιλική προς όλους τους

χρήστες, ασθενείς και γιατρούς, και να τους επιτρέπει να ολοκληρώνουν τις καθημερινές τους ανάγκες χωρίς περιττές δυσκολίες. Παρέχει οπτικά ευχάριστο και λειτουργικό περιβάλλον, επιτρέποντας ακόμα και σε χρήστες χωρίς προηγούμενη εμπειρία με εφαρμογές να πλοηγούνται εύκολα.

Η δημιουργία μιας εφαρμογής που θα καλύπτει όλες αυτές τις απαιτήσεις, προσφέροντας βελτιωμένη εμπειρία χρήστη και για τις δύο πλευρές, αποτελεί τον κύριο στόχο της εργασίας.

1.2 Παρόμοιες εφαρμογές στην αγορά

Στην ελληνική αγορά υπάρχουν ορισμένες γνωστές εφαρμογές και ιστοσελίδες που προσφέρουν υπηρεσίες προγραμματισμού ιατρικών ραντεβού. Ανάμεσά τους, οι πιο δημοφιλείς είναι:

1.2.1 Doctor Anytime

Η πλατφόρμα Doctor Anytime είναι από τις πιο δημοφιλείς υπηρεσίες προγραμματισμού ραντεβού για ιατρικές εξετάσεις στην Ελλάδα, κυρίως στην περιοχή της Αθήνας. Παρέχει τη δυνατότητα αναζήτησης γιατρών βάσει ειδικότητας και περιοχής, ενώ διαθέτει και την επιλογή της βιντεοκλήσης για τηλέιατρική.

Θετικά:

Η πλατφόρμα είναι ιδιαίτερα φιλική προς τον χρήστη, προσφέροντας εύκολη αναζήτηση γιατρών με βάση την ειδικότητα και την περιοχή. Ταυτόχρονα, παρέχει αξιολογήσεις από άλλους χρήστες, κάτι που ενισχύει την εμπιστοσύνη στη διαδικασία επιλογής. Η δυνατότητα τηλεϊατρικής, μέσω βιντεοκλήσεων, είναι ιδιαίτερα χρήσιμη για ασθενείς που χρειάζονται άμεση βοήθεια από απόσταση.

Αρνητικά:

Οι λειτουργίες της πλατφόρμας είναι επικεντρωμένες στους ασθενείς. Δεν προσφέρει στατιστικά δεδομένα ή εργαλεία παρακολούθησης της δραστηριότητας στους γιατρούς και επαγγελματίες υγείας.

Παρά την πληθώρα επιλογών σε γιατρούς ειδικά σε μεγαλύτερες πόλεις, ορισμένοι χρήστες εκφράζουν απογοήτευση από την ποιότητα της περιθάλψης αναφέροντας πως οι γιατροί του δικτύου δεν παρέχουν το επίπεδο φροντίδας που αναμένουν από μία τέτοια πλατφόρμα.

Επιπλέον, οι ακυρώσεις ή επαναπρογραμματισμοί των ραντεβού χωρίς επαρκή προειδοποίηση έχουν δημιουργήσει αρκετές αρνητικές εμπειρίες για τους χρήστες, οι οποίοι έρχονται αντιμέτωποι με αβεβαιότητα και καθυστερήσεις. Ένα άλλο κοινό πρόβλημα αφορά τη λειτουργία αναζήτησης γιατρών ανά περιοχή. Οι χρήστες έχουν παρατηρήσει ότι η αναζήτηση δεν αποδίδει πάντα ακριβή αποτελέσματα και σε ορισμένες περιπτώσεις εμφανίζονται γιατροί από περιοχές εκτός της ζητούμενης.

1.2.2 finddoctors.gov.gr (ΗΔΙΚΑ)

Η FindDoctors.gov.gr είναι μια κυβερνητική πλατφόρμα που επιτρέπει την κράτηση ραντεβού με γιατρούς σε δημόσιες και ιδιωτικές μονάδες υγείας, με ιδιαίτερη έμφαση στην πρωτοβάθμια φροντίδα. Είναι πιο διαδεδομένη σε όλη την Ελλάδα, συμπεριλαμβανομένων των μικρότερων πόλεων και περιοχών, προσφέροντας μια πιο ολοκληρωμένη κάλυψη σε σχέση με άλλες πλατφόρμες.

Θετικά:

Η πλατφόρμα παρέχει τη δυνατότητα online κράτησης ραντεβού με γιατρούς σε δημόσιες δομές πρωτοβάθμιας φροντίδας (π.χ. Κέντρα Υγείας, TOMY) ή και πιστοποιημένους ιδιώτες γιατρούς. Διαθέτει εύχρηστο σύστημα για την αναζήτηση γιατρών, με δυνατότητες φιλτραρίσματος βάσει ειδικότητας και περιοχής.

Αρνητικά:

Ένα από τα μεγαλύτερα προβλήματα που αντιμετωπίζουν οι χρήστες είναι οι μεγάλοι χρόνοι αναμονής για ραντεβού, ιδιαίτερα σε κεντρικές περιοχές. Υπάρχει περιορισμένη συνδεσιμότητα μεταξύ δημόσιων νοσοκομείων και πρωτοβάθμιων δομών.

Επίσης, τα δημόσια νοσοκομεία δεν ενημερώνουν πάντα άμεσα τη βάση δεδομένων για τη διαθεσιμότητα των γιατρών, με αποτέλεσμα να υπάρχει ελλιπής ενημέρωση των διαθέσιμων ραντεβού ή να πραγματοποιούνται αλλαγές χωρίς προειδοποίηση. Τέλος, η διαδικασία εισόδου απαιτεί πιστοποίηση μέσω του προσωπικού γιατρού, που μπορεί να δυσχεραίνει την πρόσβαση.

1.2.3 AnyDoctor

Το AnyDoctor είναι μία εύχρηστη πλατφόρμα που επιτρέπει στους χρήστες να κλείνουν ραντεβού με γιατρούς μέσω ενός απλού συστήματος αναζήτησης με βάση την ειδικότητα και την περιοχή. Επίσης παρέχει βασικές πληροφορίες για τις υπηρεσίες και θεραπείες.

Θετικά:

Η διαδικασία της κράτησης είναι γρήγορη και απλή, επιτρέποντας στους χρήστες να βρουν εύκολα τους γιατρούς που χρειάζονται. Παράλληλα, το σύστημα παρέχει βασικές πληροφορίες για τις θεραπείες που καλύπτονται από τους γιατρούς, κάτι που βοηθάει στην καλύτερη κατανόηση των υπηρεσιών τους.

Αρνητικά:

Αν και η πλατφόρμα είναι γενικά εύχρηστη, οι δυνατότητες αναζήτησης και φιλτραρίσματος είναι περιορισμένες. Δεν προσφέρει δυνατότητες τηλεϊατρικής, γεγονός που περιορίζει τις επιλογές για ασθενείς που επιθυμούν να εξυπηρετηθούν απομακρυσμένα. Επιπλέον, πολλές εγγραφές γιατρών δεν συνοδεύονται από αξιολογήσεις ή οι αξιολογήσεις είναι περιορισμένες, κάτι που καθιστά τη διαδικασία επιλογής πιο δύσκολη και λιγότερο διαφανή για τους χρήστες. Έχουν παρατηρηθεί επίσης προβλήματα με την ακρίβεια των δεδομένων, όπως η έλλειψη πλήρων πληροφοριών για τις υπηρεσίες που προσφέρονται από κάθε γιατρό ή η μη επικαιροποιημένη διαθεσιμότητα.

1.2.4 Iatrica

Το Iatrica είναι μία πλατφόρμα που συνδέει τους ασθενείς με γιατρούς και προσφέρει δυνατότητες online πληρωμών και τηλεϊατρικής. Εστιάζει κυρίως στους ιδιωτικούς γιατρούς.

Θετικά:

Η χρήση της πλατφόρμας είναι εύκολη και παρέχει λειτουργίες τηλεϊατρικής, κάτι που είναι ιδιαίτερα χρήσιμο για ασθενείς που θέλουν να αποφύγουν τη φυσική επίσκεψη στον γιατρό. Η δυνατότητα online πληρωμών επίσης διευκολύνει τις συναλλαγές και αποφεύγει την ανάγκη φυσικής παρουσίας.

Αρνητικά:

Η υπηρεσία έχει περιορισμένη κάλυψη σε μικρότερες περιοχές. Κάποιοι χρήστες ανέφεραν ότι η πλατφόρμα δεν είναι πλήρως βελτιστοποιημένη για όλους τους τύπους ιατρικών ειδικοτήτων, πράγμα που ενδέχεται να περιορίζει τη χρήση της για ορισμένα ιατρικά ραντεβού. Επίσης, το σύστημα προσφέρει περιορισμένα εργαλεία διαχείρισης για τους γιατρούς και επαγγελματίες υγείας.

1.2.5 Πλατφόρμα 1535.gr

Η πλατφόρμα 1535.gr είναι η επίσημη ελληνική πλατφόρμα για την κράτηση ραντεβού σε εξωτερικά ιατρεία δημόσιων νοσοκομείων. Προσφέρει τη δυνατότητα αναζήτησης γιατρού και κράτησης ραντεβού σε δημόσιες δομές υγείας.

Θετικά:

Παρέχει τη δυνατότητα ηλεκτρονικών ραντεβού για τα εξωτερικά και απογευματινά ιατρεία σε δημόσια νοσοκομεία. Εξυπηρετεί κυρίως μεγάλες και κεντρικές υγειονομικές μονάδες, διευκολύνοντας την πρόσβαση σε υπηρεσίες υγείας σε επίπεδο εθνικό.

Αρνητικά:

Δεν συμμετέχουν όλα τα νοσοκομεία της Αττικής και της επαρχίας στο σύστημα 1535, με αποτέλεσμα η κάλυψη του συστήματος να είναι περιορισμένη. Αν και καλύπτονται αρκετά νοσοκομεία, κυρίως τα μεγάλα και κεντρικά, η συμμετοχή εξαρτάται από την απόφαση των υγειονομικών περιφερειών και των νοσοκομείων να ενταχθούν στην πλατφόρμα. Ειδικά τα μικρότερα νοσοκομεία στην επαρχία δεν έχουν ενταχθεί πλήρως, περιορίζοντας έτσι την εθνική κάλυψη και την ισότιμη πρόσβαση σε υγειονομικές υπηρεσίες.

Επιπλέον, η αποτελεσματικότητα του συστήματος εξαρτάται από την επικαιροποίηση των δεδομένων από κάθε νοσοκομείο, κάτι που συχνά δημιουργεί προβλήματα. Οι χρήστες αναφέρουν συχνά καθυστερήσεις στην ανανέωση των πληροφοριών, γεγονός που επηρεάζει αρνητικά τη διαθεσιμότητα των ραντεβού και οδηγεί σε μεγάλες αναμονές για τους ασθενείς. Αναφέρεται επίσης ότι η διαδικασία εύρεσης και κράτησης ραντεβού είναι πιο δύσκολη σε συγκεκριμένες περιοχές, όπως σε μικρές επαρχιακές πόλεις ή σε νοσοκομεία που εξυπηρετούν μεγάλο αριθμό ασθενών σε μεγαλύτερες πόλεις.

1.2.6 Εφαρμογές ιατρικών κέντρων (π.χ., ιδιωτικών νοσοκομείων):

Οι εφαρμογές των ιατρικών κέντρων προσφέρουν τις δυνατότητες κράτησης ραντεβού, παρακολούθησης της υγείας και τηλεϊατρικής, συνήθως για περιορισμένο κύκλο ασθενών που εξυπηρετούνται από συγκεκριμένα κέντρα.

Θετικά:

Προσφέρουν εύκολη διαδικασία κράτησης ραντεβού και δυνατότητες πληρωμής online, ενώ οι ασθενείς μπορούν να παρακολουθούν την υγεία τους και να κλείνουν ραντεβού μέσω της πλατφόρμας.

Αρνητικά:

Οι εφαρμογές αυτές συχνά δεν συνδέονται με το σύστημα του ΕΟΠΥΥ, καθιστώντας δύσκολη τη χρήση τους για όσους καλύπτονται από δημόσιο ασφαλιστικό φορέα. Επίσης, η κάλυψη είναι περιορισμένη στην περιοχή του συγκεκριμένου νοσοκομείου ή ιατρικού κέντρου.

1.2.7 Viva.gr (Viva Health)

Το "Viva.gr", αν και αρχικά ξεκίνησε ως πλατφόρμα για την αγορά εισιτηρίων και άλλων υπηρεσιών, έχει πλέον επεκταθεί και στο χώρο της υγείας. Το Viva.gr προσφέρει την υπηρεσία "Viva Health", που επιτρέπει την online κράτηση ραντεβού με γιατρούς σε διάφορες περιοχές της Ελλάδας.

Θετικά:

Η πλατφόρμα έχει ένα εύχρηστο σύστημα αναζήτησης γιατρών, το οποίο επιτρέπει στους χρήστες να βρίσκουν γιατρούς σύμφωνα με τις ειδικότητες και την περιοχή τους. Επιπλέον, προσφέρει τη δυνατότητα online πληρωμής για τις υπηρεσίες, διευκολύνοντας την ολοκλήρωση των διαδικασιών. Εξυπηρετεί τόσο γιατρούς του δημοσίου όσο και ιδιώτες.

Αρνητικά:

Η διαθεσιμότητα γιατρών του δημοσίου είναι συχνά περιορισμένη σε σχέση με τους ιδιώτες. Επίσης, έχουν αναφερθεί προβλήματα με την επικαιροποίηση των δεδομένων στην πλατφόρμα, ειδικά όσον αφορά τη διαθεσιμότητα των ραντεβού. Σε κάποιες περιπτώσεις, οι πληροφορίες δεν ενημερώνονται εγκαίρως, προκαλώντας καθυστερήσεις ή δυσκολίες στη διαδικασία κράτησης ραντεβού.

1.3 Σκοπός και στόχοι

Ο κύριος σκοπός της συγκεκριμένης εφαρμογής είναι να διευκολύνει και να διασφαλίσει την έγκυρη διαδικασία κλεισίματος ραντεβού με γιατρούς για τους ασθενείς, ενώ ταυτόχρονα να προσφέρει στους γιατρούς τη δυνατότητα αποτελεσματικής διαχείρισης του χρόνου και της διαθεσιμότητάς τους. Στόχος είναι να δημιουργηθεί μία εύχρηστη, αξιόπιστη και ασφαλής εμπειρία και για τους δύο χρήστες, ασθενή και γιατρό, που θα ανταποκρίνεται πλήρως στις ανάγκες και των δύο ρόλων.

Πιο συγκεκριμένα, η εφαρμογή επιτρέπει στους ασθενείς να κλείνουν ραντεβού με γιατρούς εύκολα και γρήγορα, μέσω ενός φιλικού προς τον χρήστη περιβάλλοντος. Το σύστημα αναζήτησης είναι όσο το δυνατόν πιο ακριβές και αποτελεσματικό, προκειμένου να επιστρέφει τα σωστά αποτελέσματα για τον χρήστη. Η εφαρμογή φιλτράρει τα αποτελέσματα σύμφωνα με τα κριτήρια που έχει επιλέξει ο χρήστης, όπως η ειδικότητα του γιατρού και η περιοχή, εξασφαλίζοντας ότι οι επιλογές είναι ακριβείς και ταιριαστές με τις ανάγκες του κάθε χρήστη.

Προσφέρει διαφορετικές δυνατότητες και εργαλεία ανάλογα με τον τύπο χρήστη (ασθενείς ή γιατροί). Οι ασθενείς έχουν εύκολη πλοήγηση και γρήγορη αναζήτηση γιατρού, ενώ οι γιατροί μπορούν να διαχειρίζονται τη διαθεσιμότητά τους, να παρακολουθούν τα ραντεβού τους σε πραγματικό χρόνο και να έχουν πρόσβαση σε χρήσιμα για εκείνους στατιστικά. Αυτή η διαφοροποίηση δημιουργεί μια εξατομικευμένη εμπειρία για κάθε τύπο χρήστη.

Παρέχει χρήσιμα στατιστικά δεδομένα στους γιατρούς, όπως π.χ. την πιο πολυάσχολη εβδομάδα ή ώρα, τις πιο συχνές αιτήσεις για ραντεβού και την πληρότητα του προγράμματος. Αυτά τα δεδομένα βοηθούν τους γιατρούς να οργανώνουν καλύτερα το πρόγραμμά τους και να προσαρμόζουν τις υπηρεσίες τους στις ανάγκες των ασθενών.

Χρησιμοποιώντας την τεχνολογία Firebase Realtime Database, η εφαρμογή επιτρέπει την άμεση ενημέρωση των δεδομένων σε πραγματικό χρόνο, εξασφαλίζοντας την άμεση επικοινωνία μεταξύ χρηστών και συστήματος. Αυτό διασφαλίζει ότι οι ασθενείς βλέπουν πάντα ενημερωμένες πληροφορίες σχετικά με τη διαθεσιμότητα των γιατρών.

Ιδιαίτερη έμφαση δίνεται στην ασφάλεια των προσωπικών δεδομένων των χρηστών. Η εφαρμογή πληροί τα πρότυπα ασφαλείας που απαιτούνται για την προστασία των ευαίσθητων πληροφοριών, όπως τα δεδομένα των ραντεβού και τα προσωπικά στοιχεία των χρηστών.

Παράλληλα, σε δεύτερο χρόνο, η εφαρμογή στοχεύει στη βελτίωση της πρόσβασης στις υπηρεσίες υγείας και στις μικρότερες πόλεις ή περιοχές της χώρας, που παραδοσιακά αντιμετωπίζουν μεγαλύτερες δυσκολίες στην εύρεση γιατρών και την πρόσβαση σε ραντεβού. Μέσω της πλατφόρμας, οι χρήστες σε επαρχιακές περιοχές θα έχουν την ίδια ευχέρεια και δυνατότητες με εκείνους στις μεγαλύτερες πόλεις.

Η εφαρμογή θα αναβαθμίζεται τακτικά, λαμβάνοντας υπόψη τις αξιολογήσεις των χρηστών και τις τελευταίες τάσεις στην αγορά. Οι νέες δυνατότητες και η βελτίωση της εμπειρίας χρήστη θα είναι προτεραιότητα, ώστε η εφαρμογή να παραμένει σύγχρονη και να εξυπηρετεί τους χρήστες της με τον καλύτερο δυνατό τρόπο.

Εν κατακλείδι, η εφαρμογή στοχεύει στην επίλυση προβλημάτων που παρατηρούνται σε άλλες πλατφόρμες κράτησης ραντεβού προσφέροντας μια πιο ολοκληρωμένη λύση. Στόχος της είναι να παρέχει ακριβείς, ενημερωμένες και εξατομικευμένες πληροφορίες σχετικά με τους γιατρούς και τη διαθεσιμότητά τους, ενώ παράλληλα θα διευκολύνει τους χρήστες με απλή και γρήγορη πλοήγηση. Θα εξασφαλίζει την άμεση ενημέρωση σε περιπτώσεις ακύρωσης ή υπενθύμισης

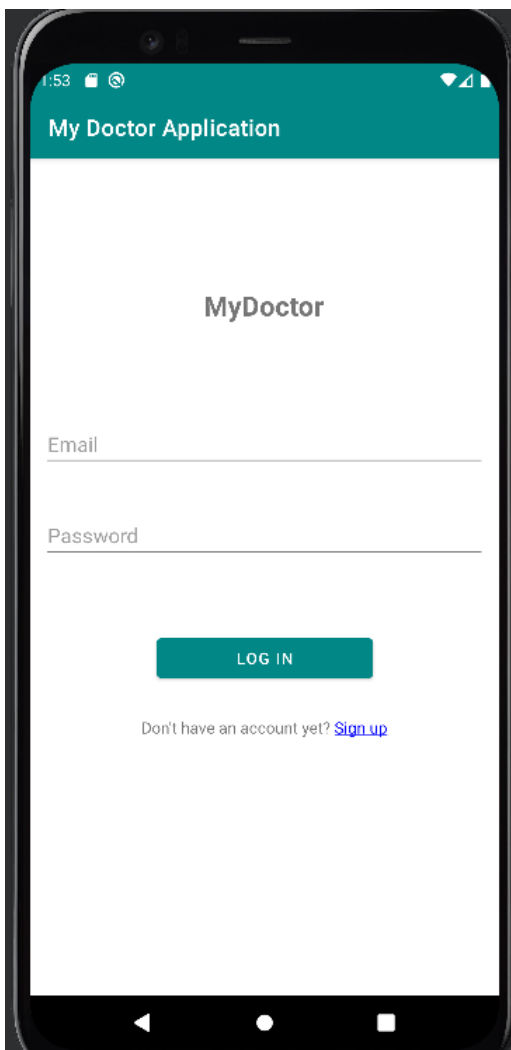
επερχόμενων ραντεβού, ενώ θα ενισχύει τη διαθεσιμότητα των υπηρεσιών υγείας ακόμη και στις πιο απομακρυσμένες περιοχές.

Κεφάλαιο 2: Screenshots και Περιγραφή Λειτουργικότητας

2.1 Ασθενής (user view)

2.1.1 Αρχική σελίδα / Login Page

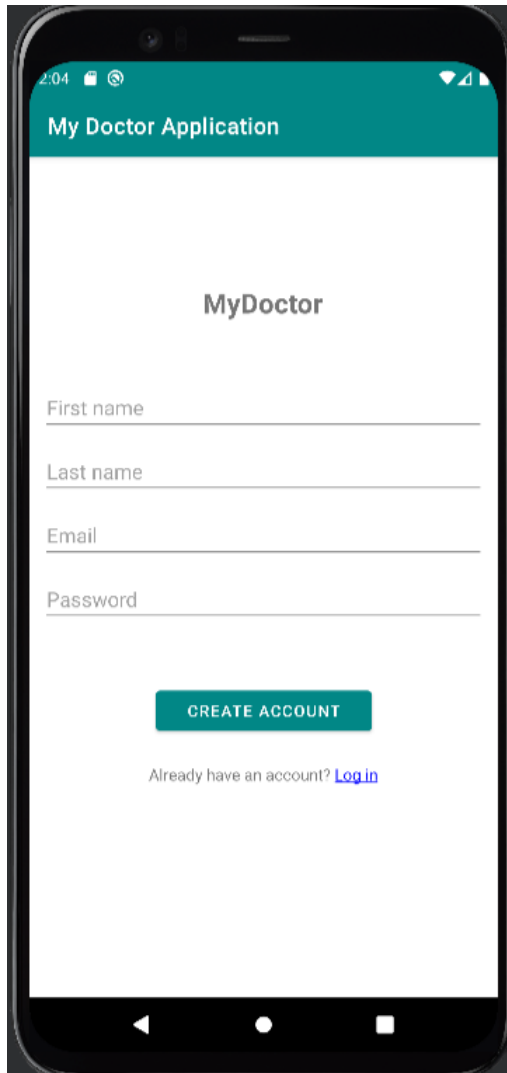
Με την εκκίνηση της εφαρμογής, ο χρήστης μεταβαίνει αυτόματα στην αρχική οθόνη (Login page) (Εικόνα 2.1.1: Αρχική). Από εδώ, μπορεί να συνδεθεί στον λογαριασμό του ή, εάν δεν διαθέτει ήδη λογαριασμό, να δημιουργήσει έναν νέο πατώντας στο “Sign up”. Η σύνδεση είναι απαραίτητη για την πρόσβαση στις λειτουργίες της εφαρμογής, όπως η δυνατότητα κλεισίματος ραντεβού με γιατρό.



Εικόνα 2.1.1: Αρχική

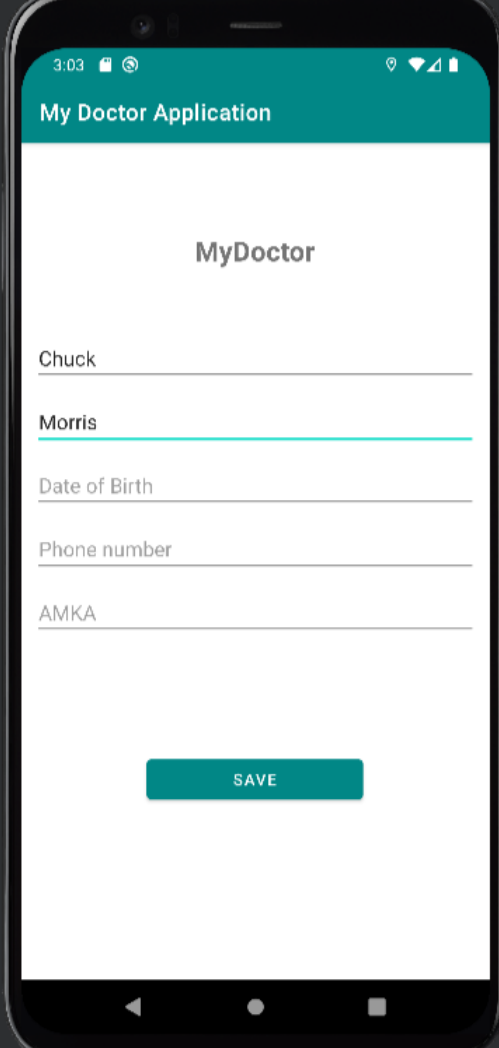
2.1.2 Δημιουργία νέου λογαριασμού

Στην οθόνη εγγραφής (Εικόνα 2.1.2: Εγγραφή), ο χρήστης μπορεί να δημιουργήσει νέο λογαριασμό συμπληρώνοντας τα απαιτούμενα πεδία, όπως όνομα, επώνυμο, διεύθυνση email και κωδικό πρόσβασης. Αφού εισάγει τα στοιχεία του, μπορεί να επιβεβαιώσει την εγγραφή του πατώντας το κουμπί “Create Account”. Η διαδικασία αυτή είναι απαραίτητη για να αποκτήσει πρόσβαση στις λειτουργίες της εφαρμογής.



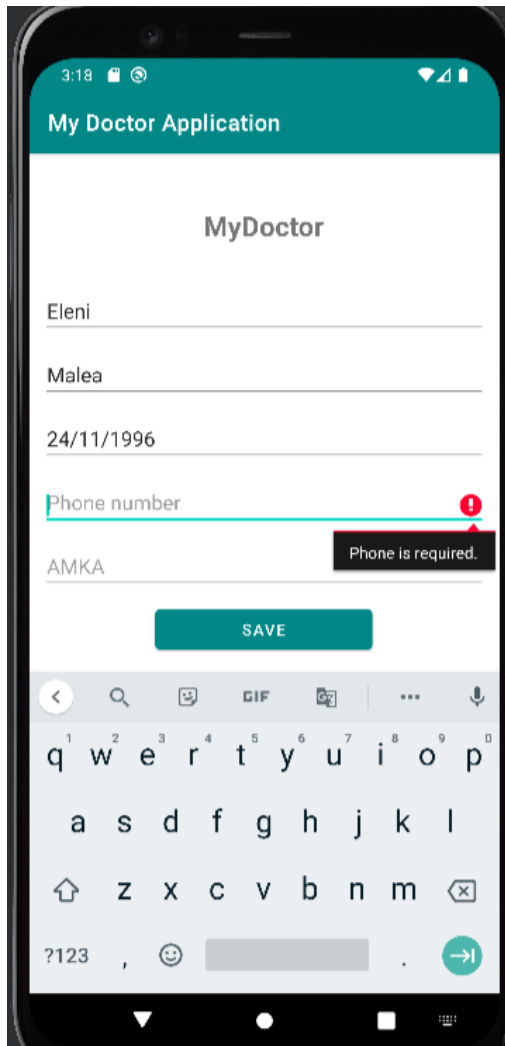
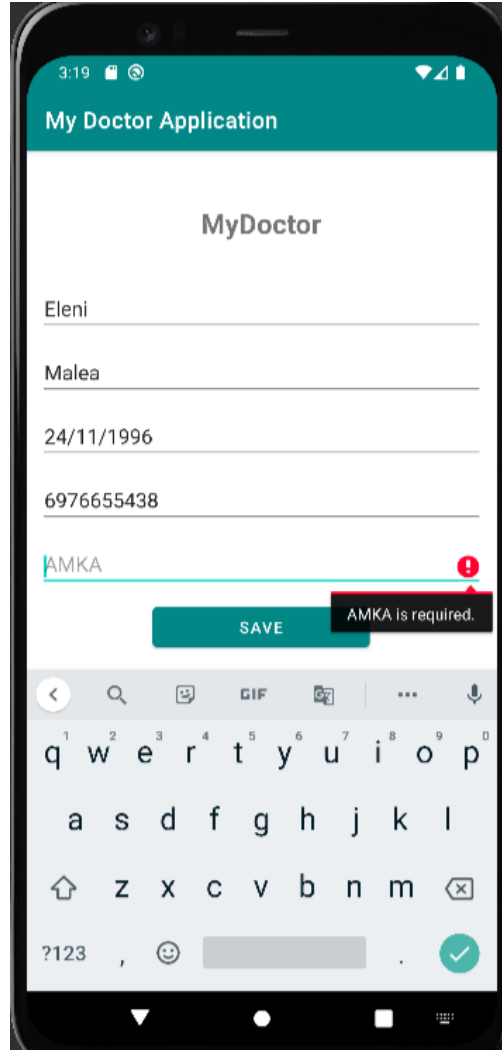
Εικόνα 2.1.2: Εγγραφή

Μετά τη συμπλήρωση αυτών των στοιχείων, η εφαρμογή ζητάει επιπλέον πληροφορίες απαραίτητες για τη δημιουργία του λογαριασμού, όπως την ημερομηνία γέννησης, το τηλέφωνο επικοινωνίας και τον ΑΜΚΑ (Εικόνα 2.1.3: Συμπλήρωση στοιχείων). Μόλις ολοκληρωθεί η καταχώρηση όλων των πεδίων, ο χρήστης μπορεί να επιβεβαιώσει τη δημιουργία του λογαριασμού πατώντας το κουμπί “Save”.

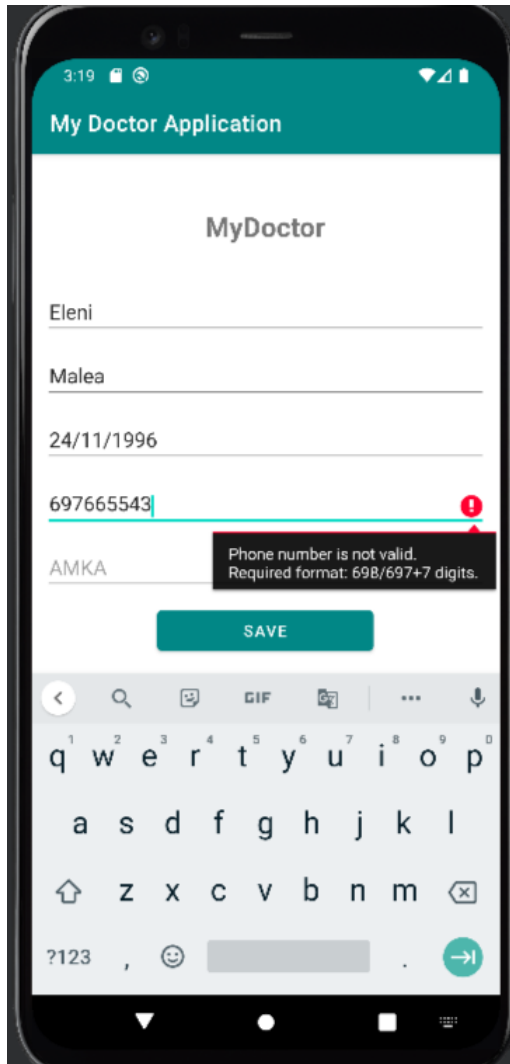


Εικόνα 2.1.3: Συμπλήρωση στοιχείων

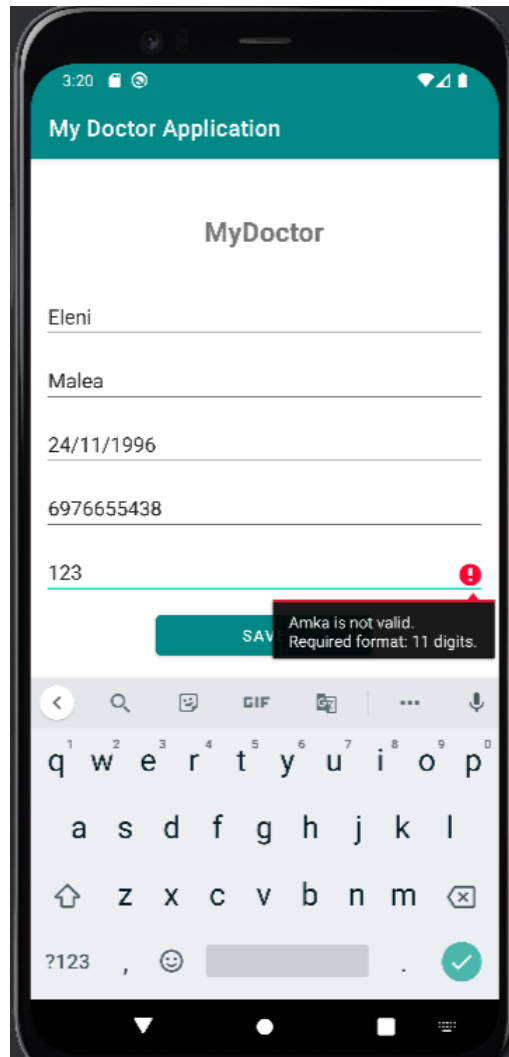
Κατά τη διαδικασία εγγραφής, πραγματοποιούνται διάφοροι έλεγχοι για να διασφαλιστεί ότι τα απαιτούμενα στοιχεία έχουν καταχωρηθεί σωστά. Όταν ο χρήστης εισάγει τα προσωπικά του στοιχεία (όνομα, επώνυμο, email, κωδικό πρόσβασης), η εφαρμογή δεν επιτρέπει τη μετάβαση στην κεντρική σελίδα του ασθενή αν δεν συμπληρωθούν τα απαραίτητα πεδία, όπως η ημερομηνία γέννησης, το τηλέφωνο επικοινωνίας και ο ΑΜΚΑ. Σε περίπτωση που λείπει κάποιο από αυτά τα στοιχεία, εμφανίζονται κατάλληλα μηνύματα στην οθόνη που ενημερώνουν τον χρήστη για την απουσία των συγκεκριμένων πληροφοριών (Εικόνα 2.1.4: Σφάλμα 1, Εικόνα 2.1.5: Σφάλμα 2).

**Εικόνα 2.1.4:** Σφάλμα 1**Εικόνα 2.1.5:** Σφάλμα 2

Επιπλέον, πραγματοποιούνται έλεγχοι για το σωστό format των δεδομένων που εισάγονται. Ειδικότερα, το τηλέφωνο και ο ΑΜΚΑ ελέγχονται για να διασφαλιστεί ότι πληρούν το απαιτούμενο format. Αν κάποιος από τους δύο αυτούς αριθμούς δεν πληροί τις προδιαγραφές, εμφανίζεται μήνυμα που καθοδηγεί τον χρήστη για το σωστό format, ώστε να μπορέσει να διορθώσει τα δεδομένα πριν προχωρήσει στην εγγραφή (Εικόνα 2.1.6: Σφάλμα 3, Εικόνα 2.1.7: Σφάλμα 4).



Εικόνα 2.1.6: Σφάλμα 3

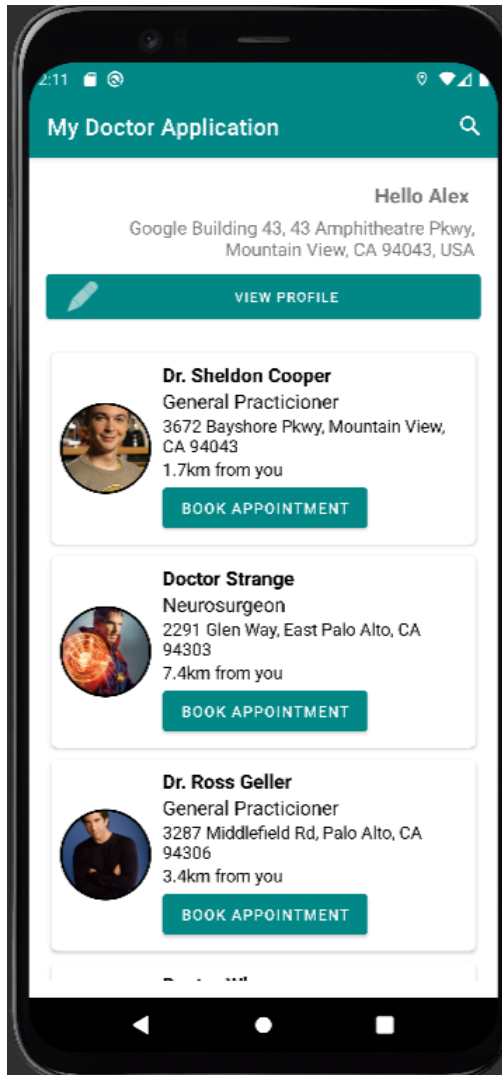


Εικόνα 2.1.7: Σφάλμα 4

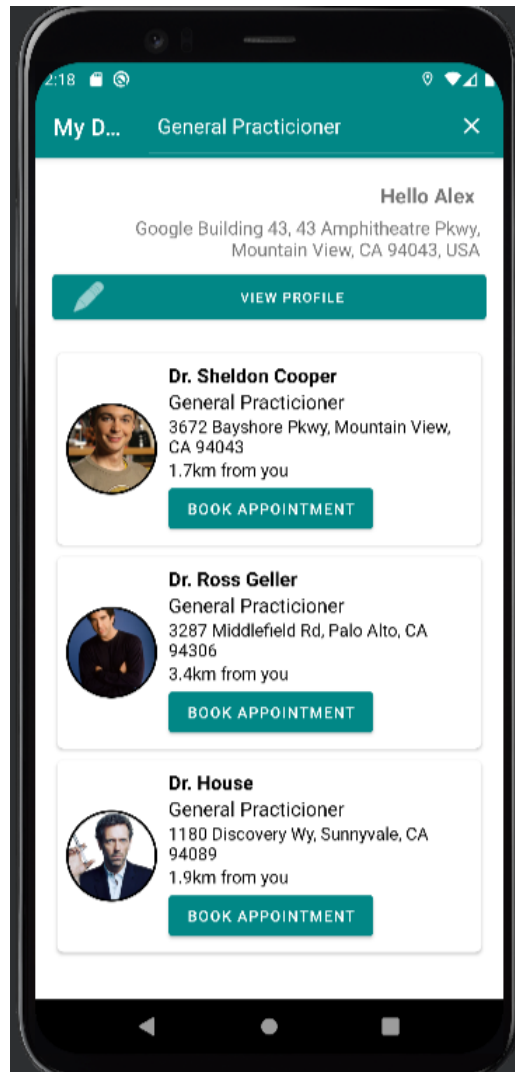
2.1.3 Είσοδος στην κεντρική σελίδα του ασθενή

Αφού δημιουργηθεί καινούριο account ασθενή ή αφού κάποιος εγγεγραμμένος ασθενής κάνει log in από την αρχική σελίδα, στην κεντρική σελίδα του ασθενή (Εικόνα 2.1.8: Κεντρική ασθενή), εμφανίζεται η διεύθυνση που έχει καταχωρήσει ο ίδιος, καθώς και μια λίστα με τους διαθέσιμους γιατρούς. Για κάθε γιατρό παρουσιάζονται η ειδικότητά του και η απόσταση από τον χρήστη. Ο ασθενής μπορεί να επιλέξει έναν γιατρό για να κλείσει ραντεβού πατώντας το κουμπί 'Book Appointment'. Εναλλακτικά, χρησιμοποιώντας τον μεγεθυντικό φακό στο πάνω μέρος της οθόνης, μπορεί να πραγματοποιήσει αναζήτηση για συγκεκριμένο γιατρό ή ειδικότητα (Εικόνα 2.1.9: Αναζήτηση ειδικότητα, Εικόνα 2.1.10 Αναζήτηση γιατρού).

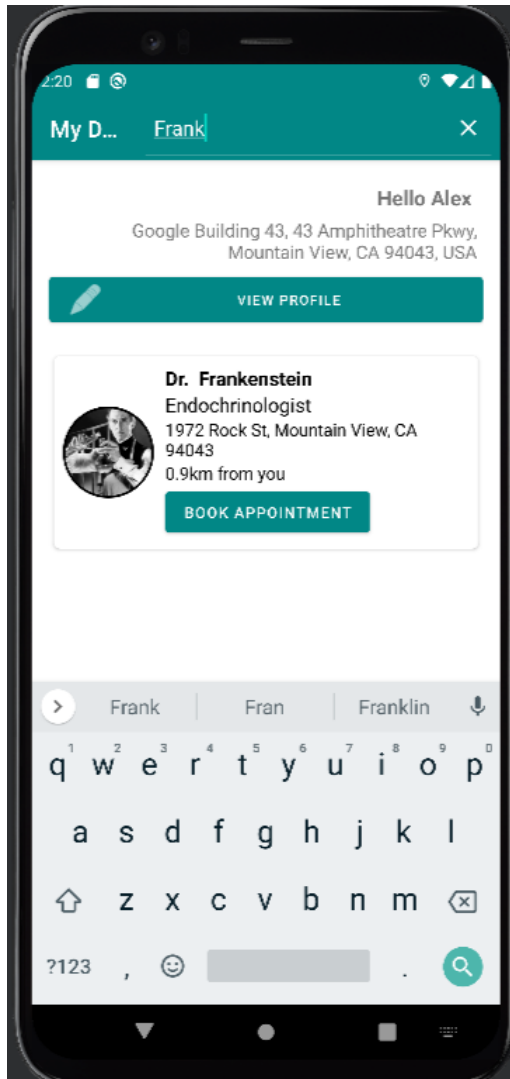
Επιπλέον, από την ίδια σελίδα, ο ασθενής έχει τη δυνατότητα να μεταβεί στο προφίλ του πατώντας το κουμπί "View Profile", όπου μπορεί να διαχειριστεί τις προσωπικές του πληροφορίες και ρυθμίσεις. Οι λειτουργίες του προφίλ θα περιγραφούν αναλυτικά στη συνέχεια.



Εικόνα 2.1.9: Κεντρική ασθενή



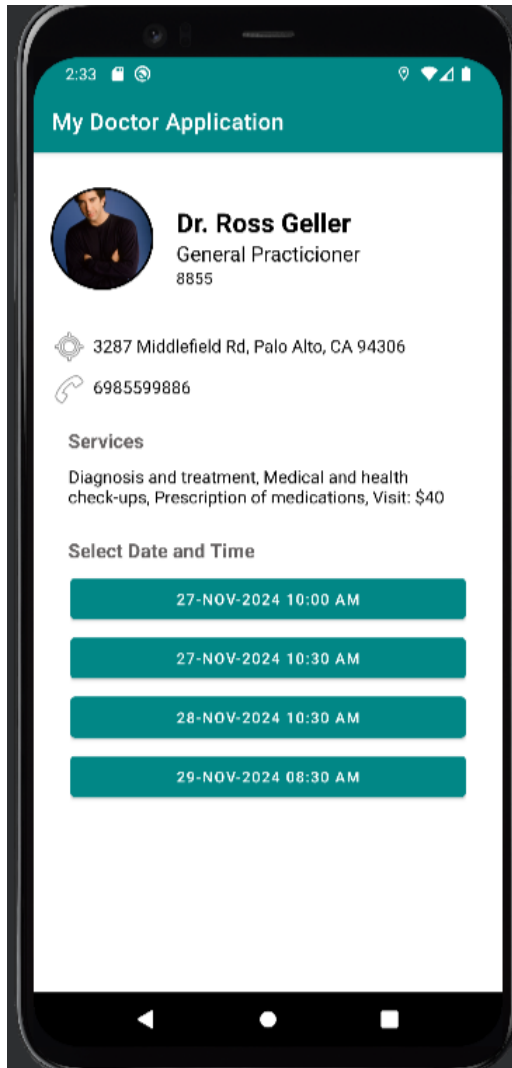
Εικόνα 2.1.9: Αναζήτηση ειδικότητα



Εικόνα 2.1.10 Αναζήτηση γιατρού

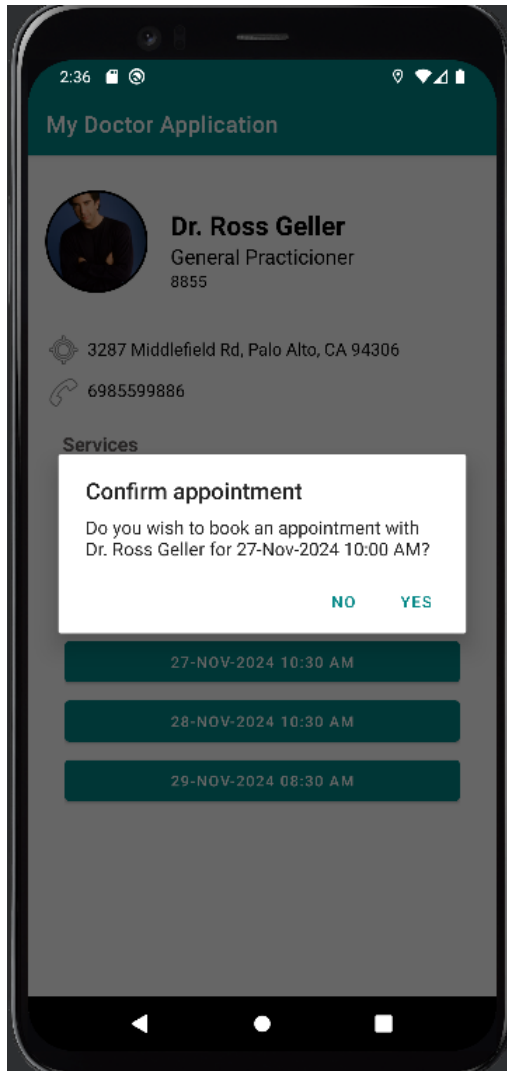
2.1.4 Προβολή σελίδας γιατρού / Κλείσιμο ραντεβού

Στην οθόνη κλεισίματος ραντεβού (Εικόνα 2.1.11: Κλείσιμο Ραντεβού), ο χρήστης βλέπει τις λεπτομέρειες του επιλεγμένου γιατρού, όπως το όνομα, το επώνυμο, τον αριθμό άδειας, την ειδικότητα, τη διεύθυνση του ιατρείου, το τηλέφωνο επικοινωνίας, τις παρεχόμενες υπηρεσίες και το κόστος επίσκεψης. Επιπλέον, εμφανίζεται η λίστα με τις διαθέσιμες ώρες ραντεβού. Ο ασθενής μπορεί να επιλέξει την ώρα που προτιμά από τις διαθέσιμες και να προχωρήσει με την επιβεβαίωση του ραντεβού.



Εικόνα 2.1.11: Κλείσιμο Ραντεβού

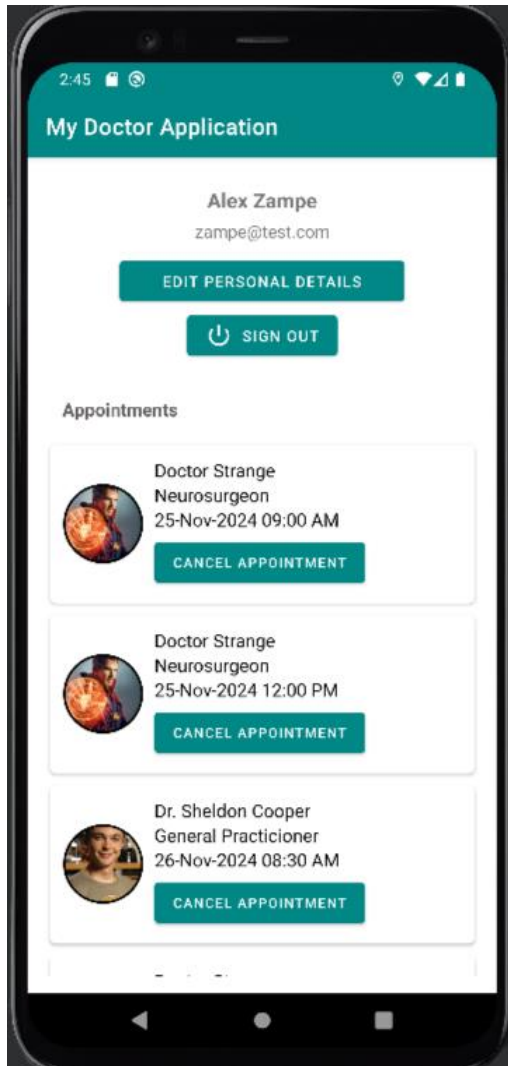
Αφού ο ασθενής επιλέξει μία συγκεκριμένη ώρα για το ραντεβού, εμφανίζεται ένα pop-up για επιβεβαίωση (Εικόνα 2.1.12: Επιβεβαίωση Ραντεβού). Το παράθυρο αυτό περιέχει τις λεπτομέρειες του ραντεβού, όπως το όνομα του γιατρού, την ημερομηνία και την ώρα της συνάντησης. Ο ασθενής καλείται να επιβεβαιώσει ότι επιθυμεί να κλείσει το ραντεβού πατώντας στο κουμπί “YES”. Εάν αλλάξει γνώμη, μπορεί να επιλέξει “NO” και να επιστρέψει στην προηγούμενη οθόνη.



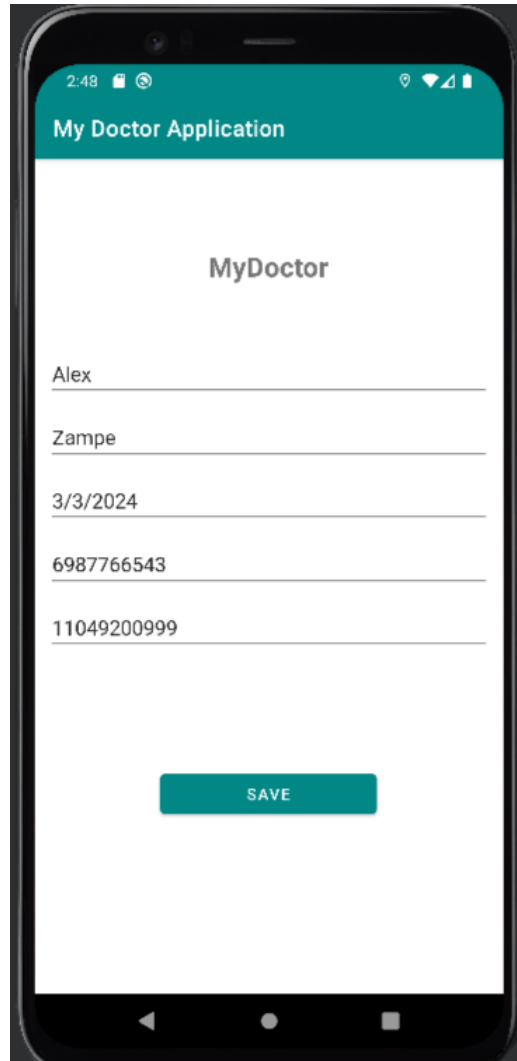
Εικόνα 2.1.12: Επιβεβαίωση Ραντεβού

2.1.5 Μετάβαση στο προφίλ του ασθενή

Όταν ο ασθενής επιλέξει 'View Profile' από την κεντρική σελίδα, μεταβαίνει στη σελίδα του προφίλ του (Εικόνα 2.1.13: Προφίλ Χρήστη). Στην οθόνη αυτή, ο χρήστης μπορεί να δει και να διαχειριστεί τα προσωπικά του στοιχεία. Με την επιλογή 'Edit Personal Details', μπορεί να τροποποιήσει πληροφορίες όπως το όνομα, το επώνυμο, την ημερομηνία γέννησης, το τηλέφωνο και το ΑΜΚΑ, μεταβαίνοντας σε ξεχωριστό activity για την επεξεργασία των δεδομένων του (Εικόνα 2.1.14: Τροποποίηση στοιχείων). Παράλληλα, ο χρήστης έχει τη δυνατότητα να δει τα επερχόμενα ραντεβού που έχει κλείσει, τα οποία εμφανίζονται με όλες τις απαραίτητες λεπτομέρειες (γιατρός, ειδικότητα, ημερομηνία, ώρα). Εάν το επιθυμεί, μπορεί να ακυρώσει κάποιο ραντεβού απευθείας από αυτήν την οθόνη πατώντας το κουμπί "Cancel Appointment". Τέλος, παρέχεται η επιλογή "Sign Out", μέσω της οποίας ο χρήστης αποσυνδέεται από την εφαρμογή και επιστρέφει στην αρχική σελίδα σύνδεσης/εγγραφής.

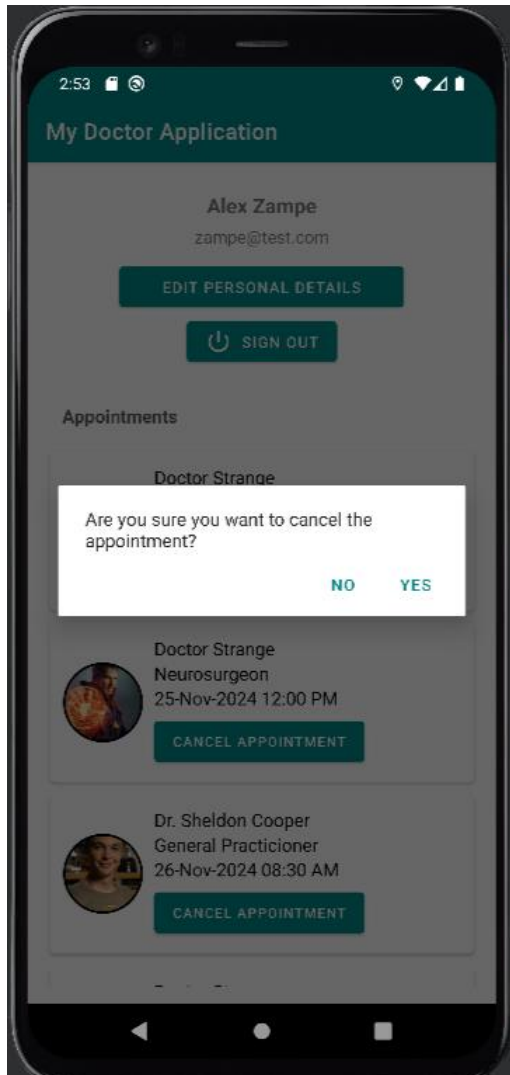


Εικόνα 2.1.13: Προφίλ Χρήστη



Εικόνα 2.1.14: Τροποποίηση στοιχείων

Όταν ο ασθενής πατήσει το κουμπί "Cancel Appointment" για κάποιο ραντεβού, εμφανίζεται ένα popup για επιβεβαίωση της ακύρωσης (Εικόνα 2.1.15: Επιβεβαίωση Ακύρωσης Ραντεβού). Ο ασθενής καλείται να επιβεβαιώσει την ακύρωση πατώντας "YES", αν επιθυμεί να ακυρώσει το ραντεβού, ή "NO", αν θέλει να επιστρέψει στην προηγούμενη οθόνη χωρίς καμία αλλαγή.

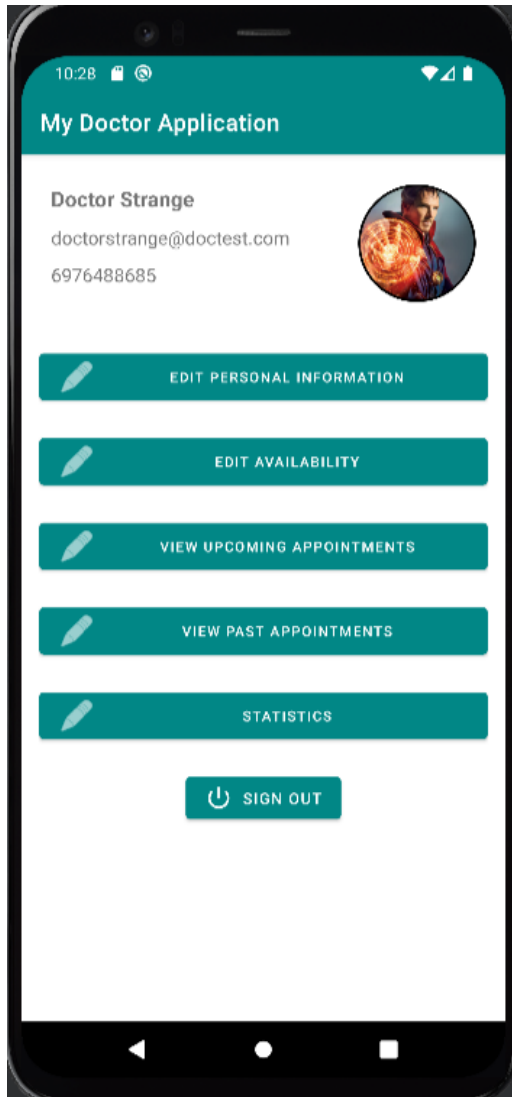


Εικόνα 2.1.15: Επιβεβαίωση Ακύρωσης Ραντεβού

2.2 Γιατρός (doctor view)

2.2.1 Αρχική σελίδα / Login Page

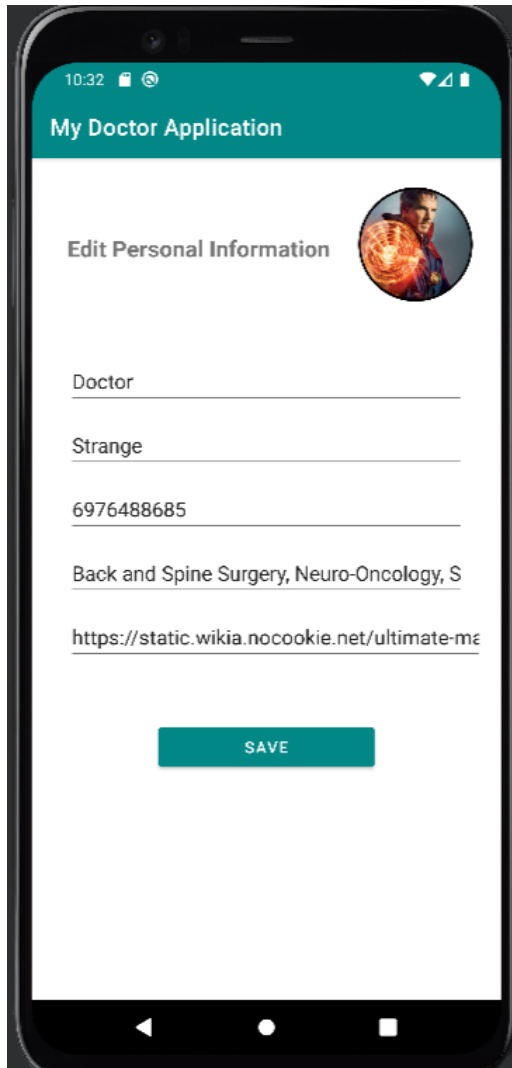
Εφόσον κάποιος χρήστης στο προφίλ του οποίου φαίνεται ότι είναι γιατρός (`isPatient = false`) κάνει log in στην εφαρμογή, θα εμφανιστεί η αρχική σελίδα του γιατρού (Εικόνα 2.2.1: Αρχική σελίδα γιατρού). Στην αρχική σελίδα, ο γιατρός βλέπει τα προσωπικά του στοιχεία, όπως το ονοματεπώνυμο, τη φωτογραφία προφίλ, τη διεύθυνση email και τον αριθμό τηλεφώνου που έχει καταχωρίσει. Επιπλέον, η σελίδα περιλαμβάνει διάφορες λειτουργικές επιλογές, μέσω κουμπιών, για τη διαχείριση της δραστηριότητάς του στην εφαρμογή: όπως η τροποποίηση των προσωπικών του στοιχείων, η διαχείριση της διαθεσιμότητάς του για ραντεβού, η προβολή των επερχόμενων και ολοκληρωμένων ραντεβού, η πρόσβαση σε στατιστικά στοιχεία για τη δραστηριότητά του, και τέλος η επιλογή αποσύνδεσης από την εφαρμογή.



Εικόνα 2.2.1: Αρχική σελίδα γιατρού

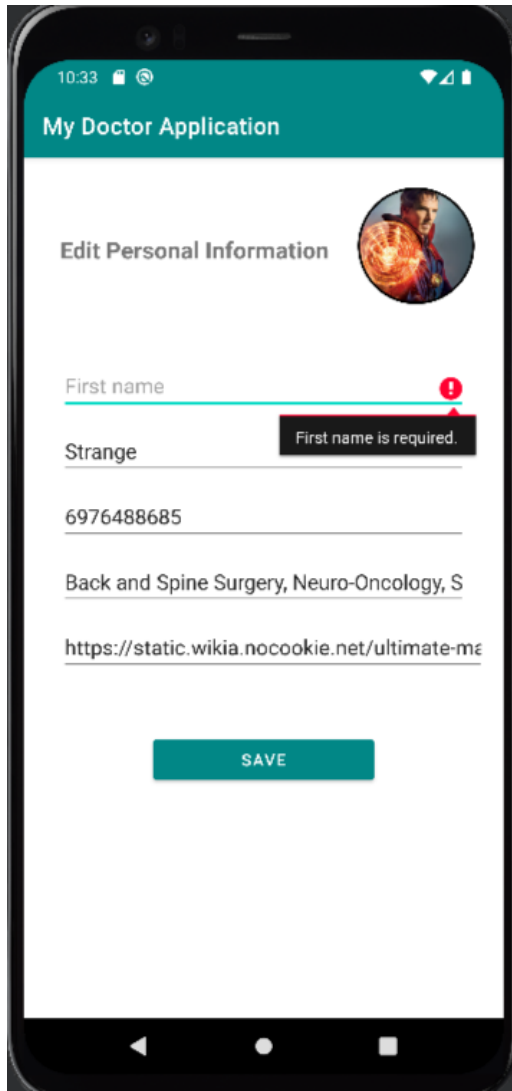
2.2.2 Τροποποίηση στοιχείων

Όταν ο γιατρός επιλέξει “Edit Personal Information” από την κεντρική σελίδα, μεταβαίνει στη σελίδα του προφίλ του (Εικόνα 2.2.2: Στοιχεία γιατρού). Στην οθόνη αυτή, μπορεί να δει και να τροποποιήσει τα προσωπικά του στοιχεία, όπως το ονοματεπώνυμο, το τηλέφωνο, τις υπηρεσίες που προσφέρει και την φωτογραφία του.

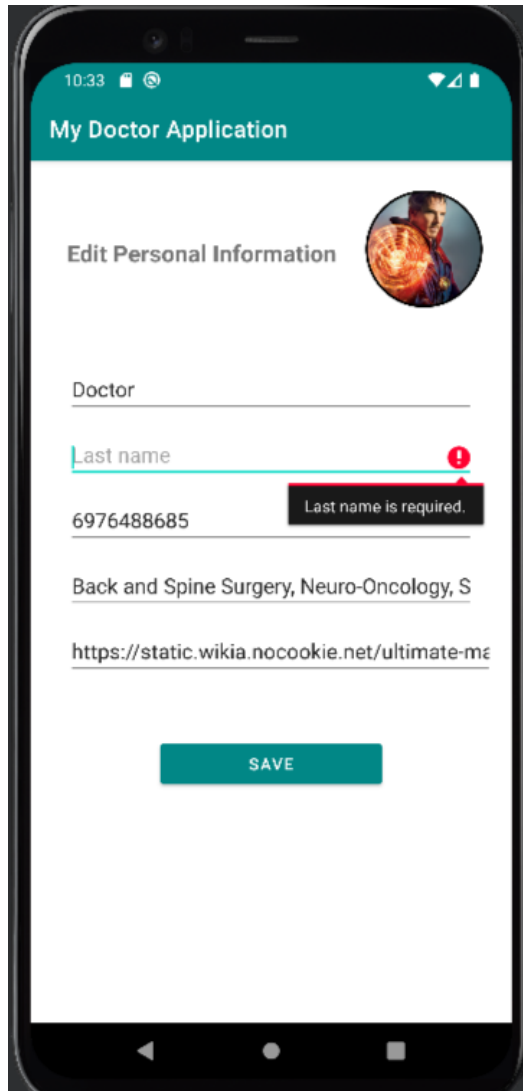


Εικόνα 2.2.2: Στοιχεία γιατρού

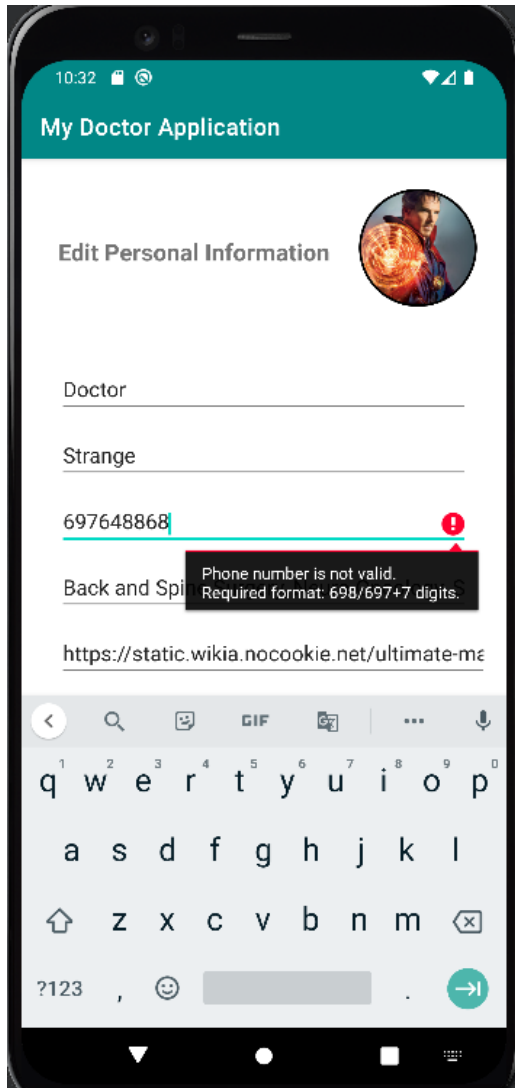
Προτού αποθηκευτούν οποιεσδήποτε αλλαγές, πραγματοποιούνται διάφοροι έλεγχοι για να διασφαλιστεί ότι τα απαιτούμενα στοιχεία έχουν καταχωρηθεί σωστά, δηλαδή ότι δεν έχει μείνει κάποιο πεδίο κενό (Εικόνα 2.2.3: Σφάλμα 1, Εικόνα 2.2.4: Σφάλμα 2) και ότι χρησιμοποιείται το απαιτούμενο format (Εικόνα 2.2.5. Σφάλμα 3).



Εικόνα 2.2.3: Σφάλμα 1

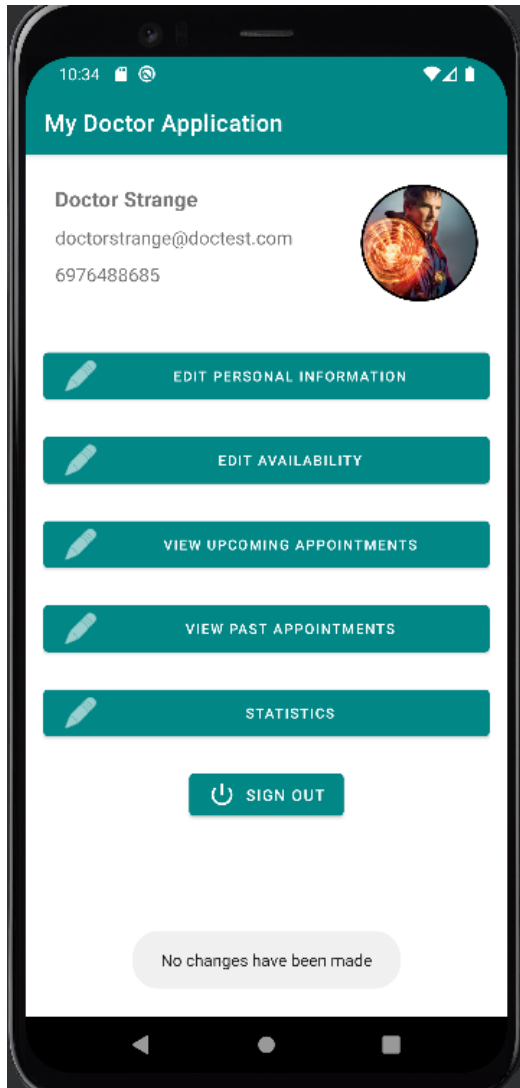


Εικόνα 2.2.4: Σφάλμα 2



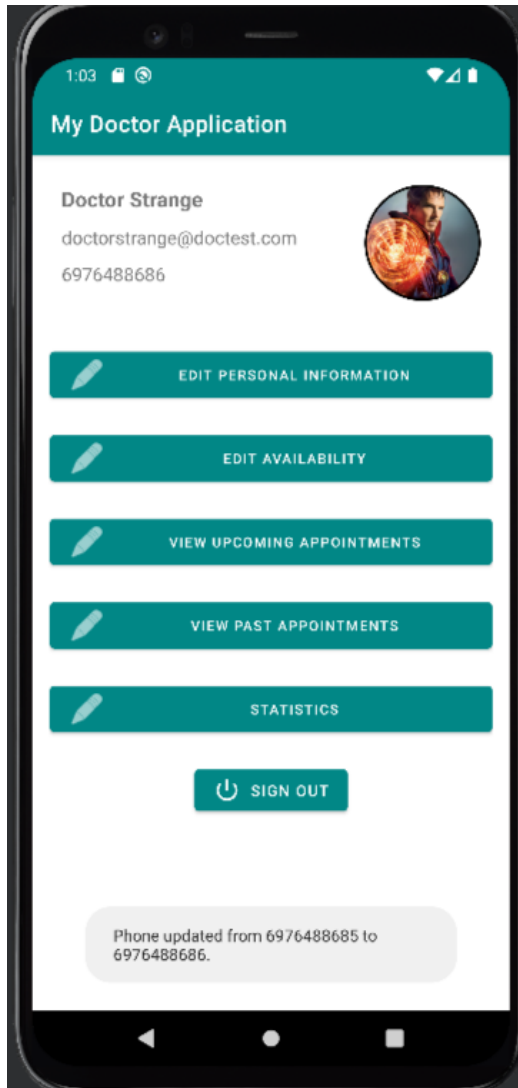
Εικόνα 2.2.5: Σφάλμα 3

Αν ο γιατρός δεν προβεί σε καμία αλλαγή στα στοιχεία του και παρόλα αυτά πατήσει την επιλογή "Save", δεν πραγματοποιείται ενημέρωση της βάσης δεδομένων. Σε αυτή την περίπτωση, εμφανίζεται μήνυμα που πληροφορεί τον γιατρό ότι δεν υπήρξε καμία τροποποίηση (Εικόνα 2.2.6: Καμία τροποποίηση). Μετά την εμφάνιση του μηνύματος, ο χρήστης επιστρέφει στην αρχική του σελίδα.



Εικόνα 2.2.6: Καμία τροποποίηση

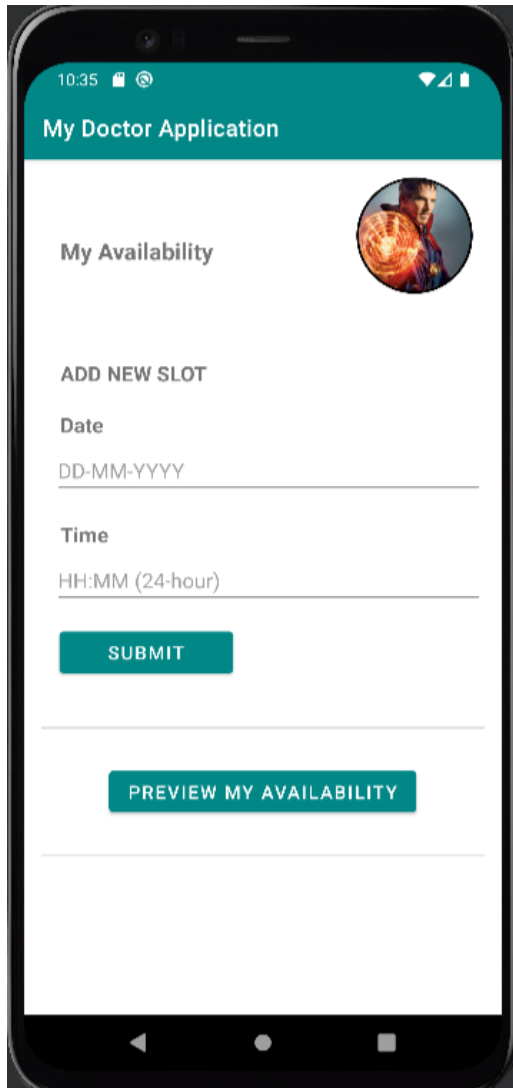
Αν ο γιατρός τροποποιήσει τα στοιχεία του και επιλέξει “Save”, η βάση δεδομένων ενημερώνεται με τις νέες πληροφορίες. Στη συνέχεια, εμφανίζεται μήνυμα που ενημερώνει τον γιατρό ότι η τροποποίηση ολοκληρώθηκε με επιτυχία (Εικόνα 2.2.7: Τροποποίηση στοιχείων). Μετά την εμφάνιση του μηνύματος, ο χρήστης επιστρέφει στην αρχική του σελίδα.



Εικόνα 2.2.6: Καμία τροποποίηση

2.2.3 Διαχείριση διαθεσιμότητας

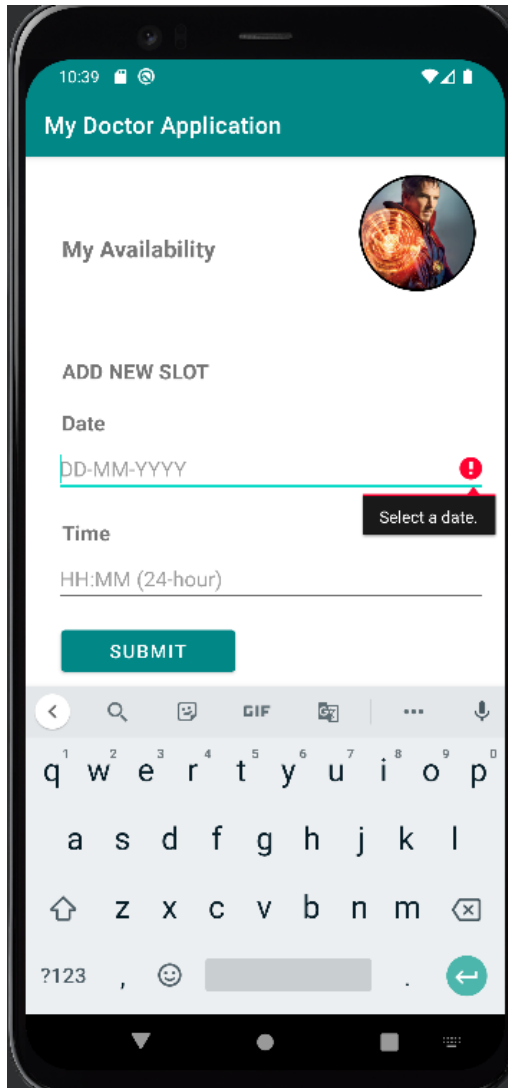
Όταν ο γιατρός επιλέξει “Edit Availability” από την κεντρική σελίδα, μεταβαίνει στη σελίδα όπου μπορεί να προσθέσει καινούρια slots και να δει την υπάρχουσα διαθεσιμότητά του (Εικόνα 2.2.7: Διαχείριση διαθεσιμότητας).



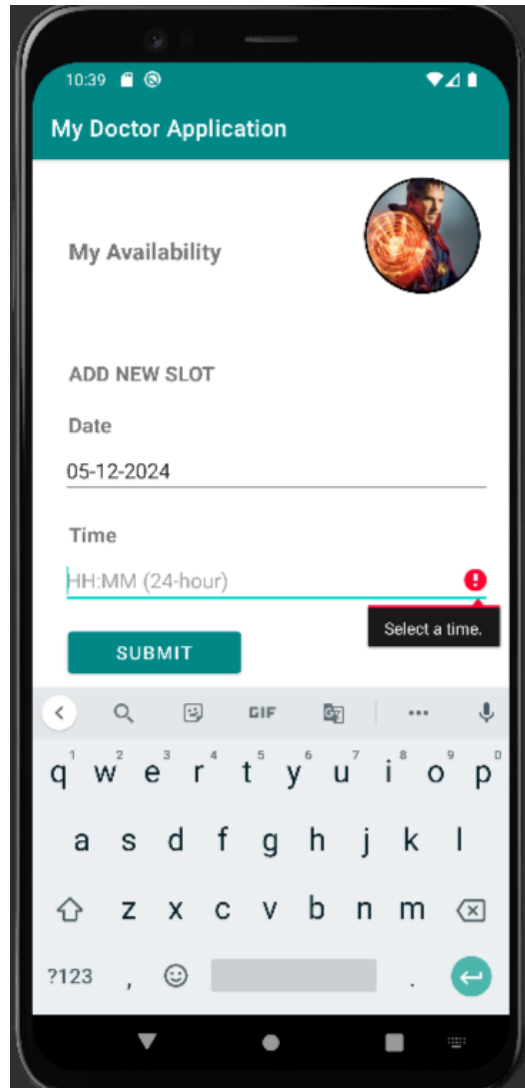
Εικόνα 2.2.7: Διαχείριση διαθεσιμότητας

Ο γιατρός πρέπει να εισαγάγει την ημερομηνία και την ώρα σε συγκεκριμένο format ώστε το slot να αποθηκευτεί επιτυχώς στη βάση. Υπάρχει υπόδειξη (hint) που δείχνει το απαιτούμενο format αλλά πραγματοποιούνται και επιπλέον έλεγχοι για να διασφαλιστεί ότι τα απαιτούμενα στοιχεία έχουν συμπληρωθεί σωστά πριν την αποθήκευση.

Εάν ο γιατρός δεν εισαγάγει την ημερομηνία (Εικόνα 2.2.8: Σφάλμα 4) ή την ώρα (Εικόνα 2.2.9: Σφάλμα 5), εμφανίζεται error που υποδεικνύει τη συγκεκριμένη παράλειψη.

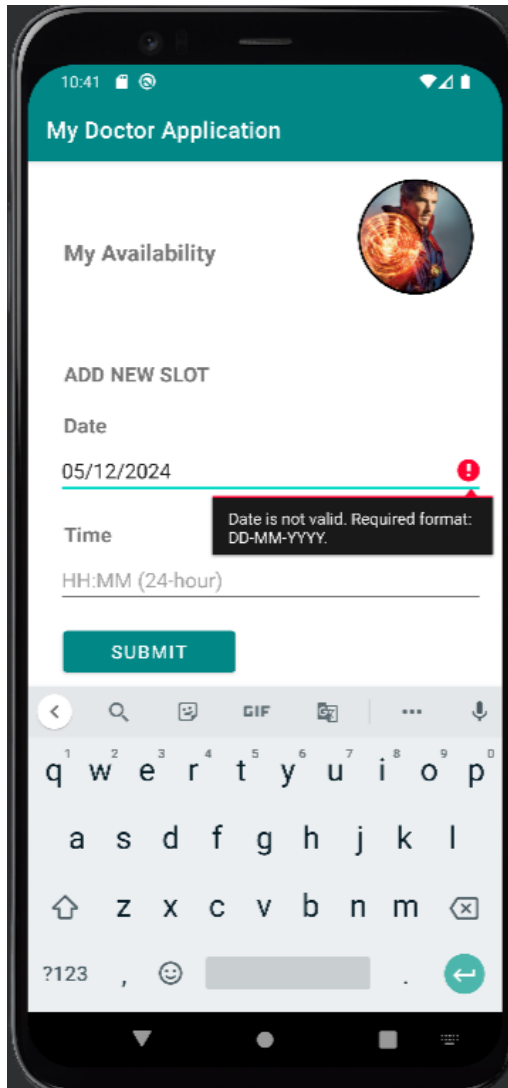


Εικόνα 2.2.8: Σφάλμα 4

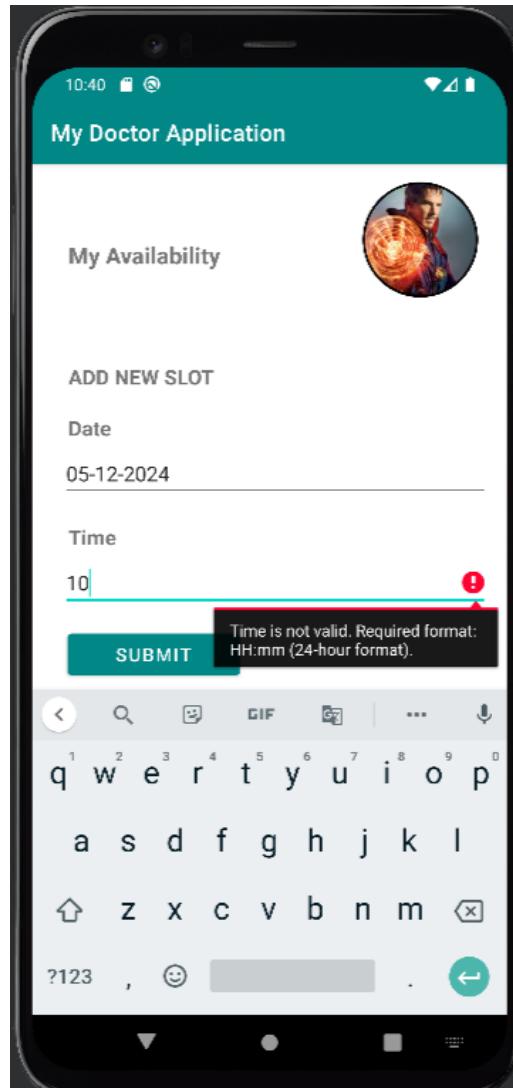


Εικόνα 2.2.9: Σφάλμα 5

Εάν ο γιατρός δεν χρησιμοποιήσει το απαιτούμενο format, θα εμφανιστεί σχετικό error (Εικόνα 2.2.10: Σφάλμα 6 και Εικόνα 2.2.11: Σφάλμα 7).

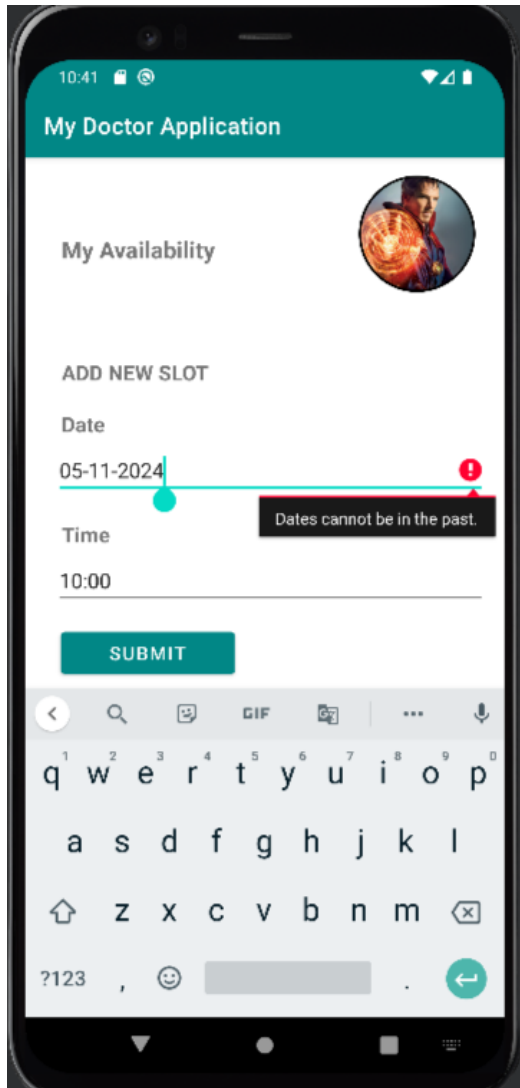


Εικόνα 2.2.10: Σφάλμα 6



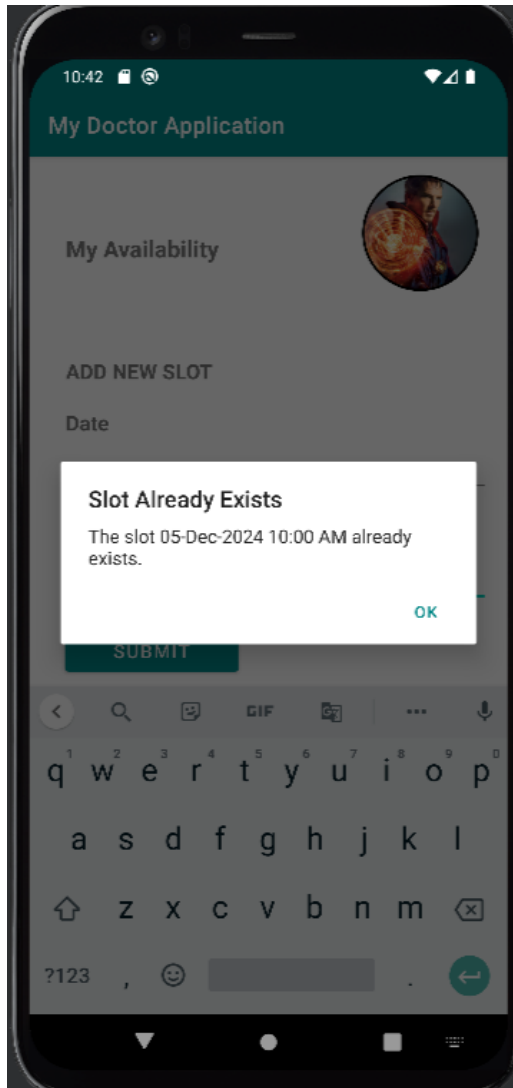
Εικόνα 2.2.11: Σφάλμα 7

Εάν ο γιατρός επιλέξει μια ημερομηνία προγενέστερη της τρέχουσας ημερομηνίας, θα εμφανιστεί σχετικό error (Εικόνα 2.2.12: Σφάλμα 8).



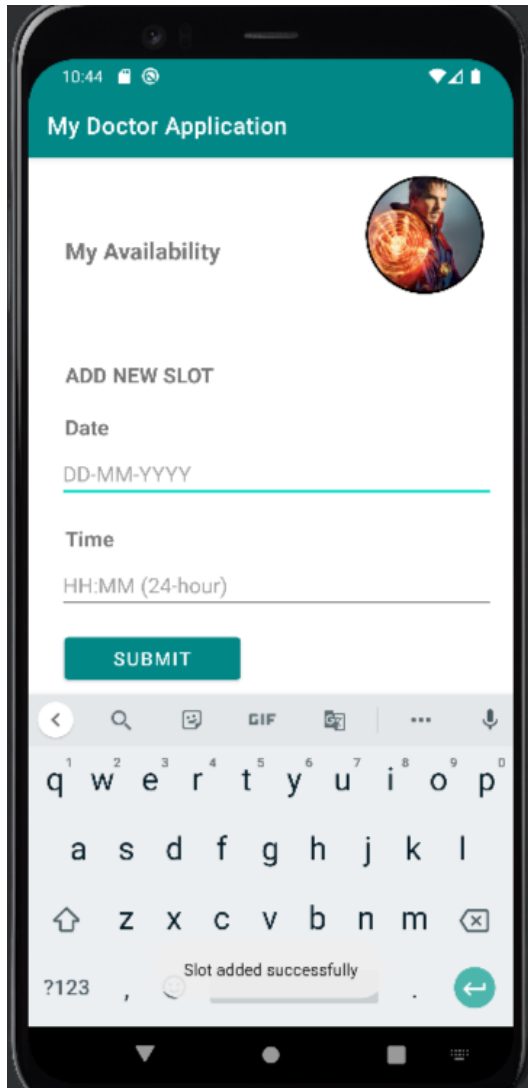
Εικόνα 2.2.12: Σφάλμα 8

Εάν το slot που προσπαθεί να προσθέσει ο χρήστης υπάρχει ήδη στη βάση, εμφανίζεται σχετικό error (Εικόνα 2.2.13: Σφάλμα 9).



Εικόνα 2.2.13: Σφάλμα 9

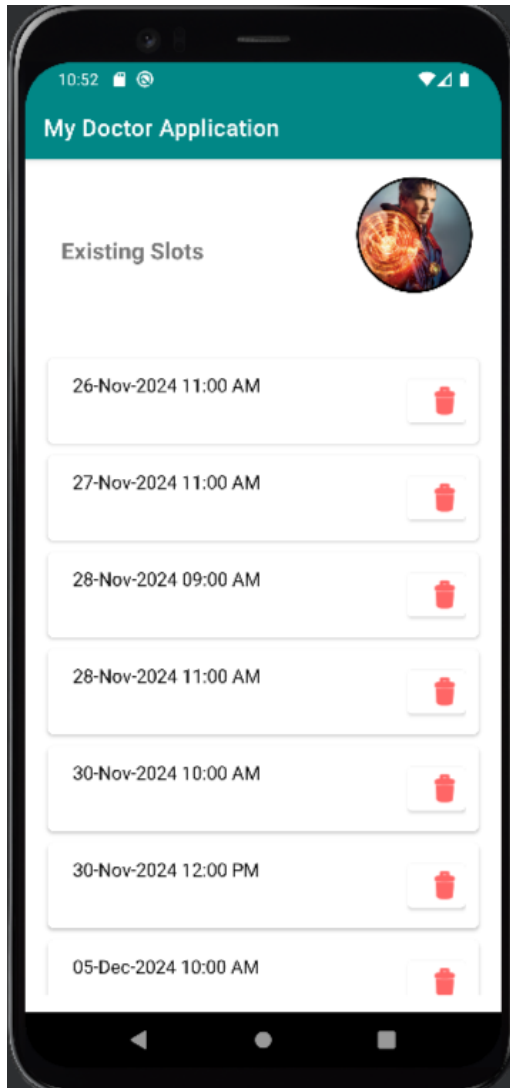
Εφόσον τα στοιχεία είναι σωστά και το slot καταχωρηθεί επιτυχώς, θα εμφανιστεί μήνυμα που ενημερώνει τον γιατρό ότι το slot καταχωρήθηκε στη βάση (Εικόνα 2.2.14: Καταχώρηση slot).



Εικόνα 2.2.14: Καταχώρηση slot

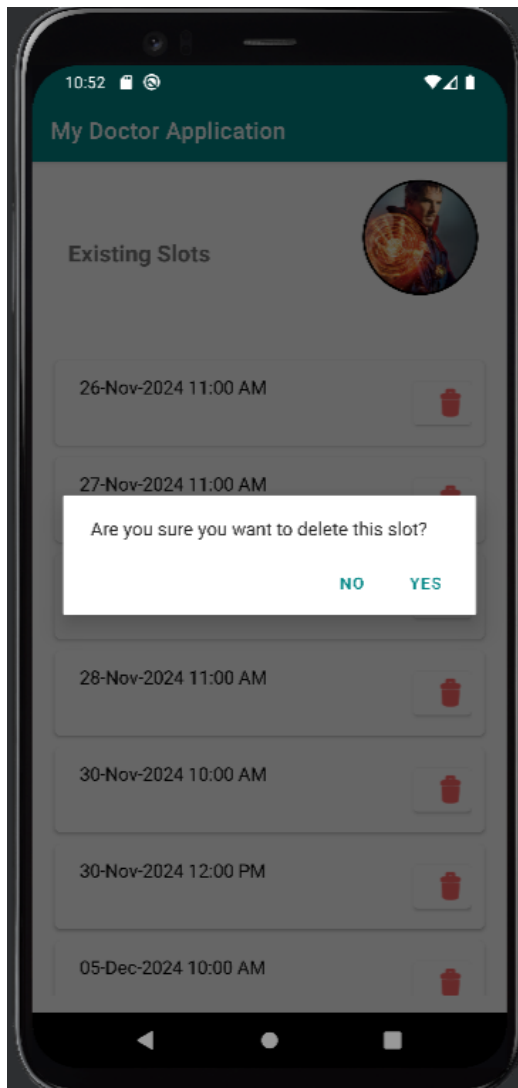
2.2.4 Προβολή διαθεσιμότητας

Όταν ο γιατρός επιλέξει την επιλογή "Preview my availability" από την σελίδα Διαχείρισης διαθεσιμότητας, μεταβαίνει στη σελίδα όπου εμφανίζονται τα slots που έχει ήδη προσθέσει (Εικόνα 2.2.15: Προβολή διαθεσιμότητας).



Εικόνα 2.2.15: Προβολή διαθεσιμότητας

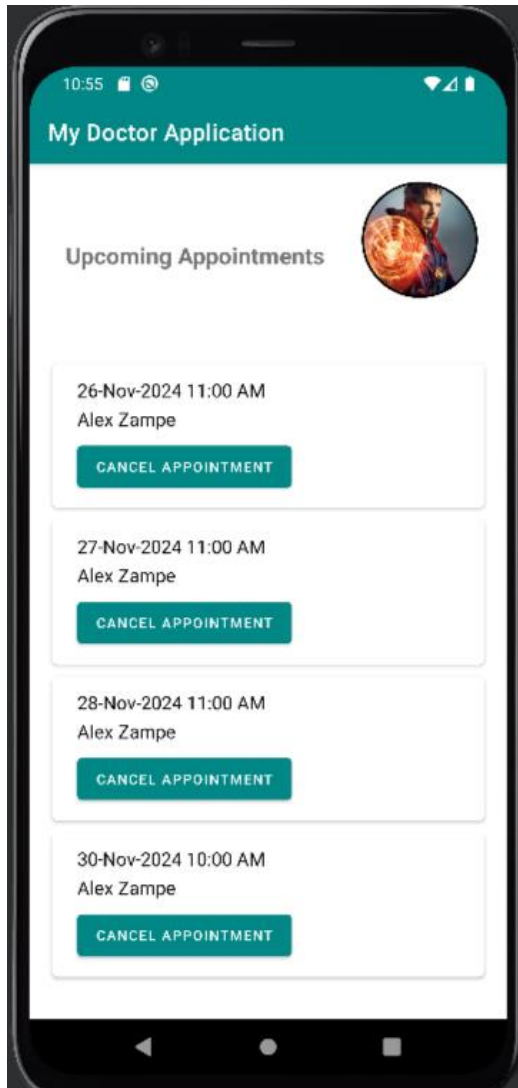
Σε αυτή τη σελίδα μπορεί να δει αναλυτικά τη διαθεσιμότητά του και να διαγράψει όποιο slot επιθυμεί (Εικόνα 2.2.16: Διαγραφή slot). Ο γιατρός καλείται να επιβεβαιώσει την διαγραφή πατώντας "YES", αν επιθυμεί να διαγράψει το slot, ή "NO", αν θέλει να επιστρέψει στην προηγούμενη οθόνη χωρίς καμία αλλαγή.



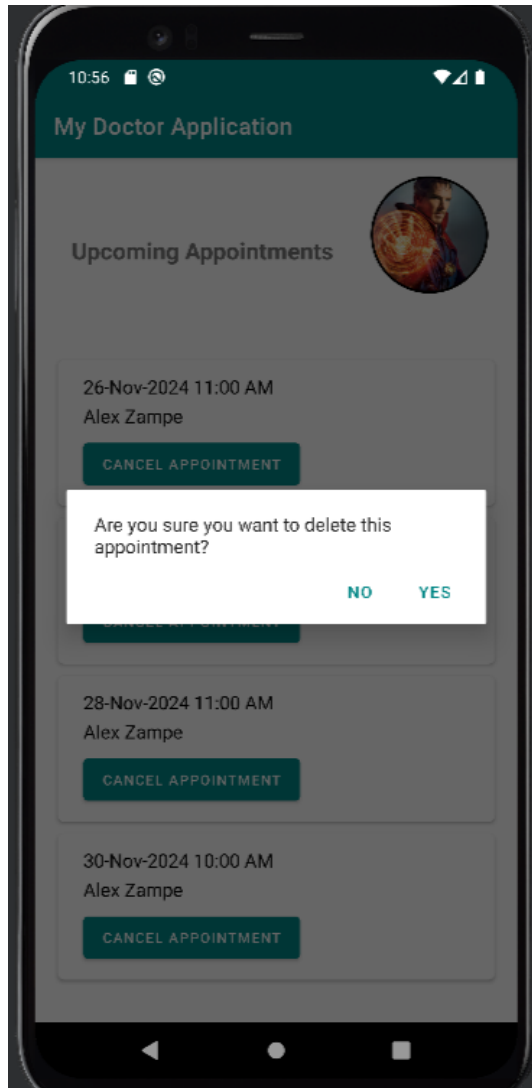
Εικόνα 2.2.16: Διαγραφή slot

2.2.5 Προβολή επερχόμενων ραντεβού

Όταν ο γιατρός επιλέξει την επιλογή «View upcoming appointments» από την κεντρική σελίδα, μεταφέρεται σε μια σελίδα όπου εμφανίζονται τα επερχόμενα ραντεβού που έχουν κλείσει οι ασθενείς (Εικόνα 2.2.17: Προβολή επερχόμενων ραντεβού). Στη σελίδα αυτή, ο γιατρός μπορεί να δει την ημερομηνία και ώρα κάθε ραντεβού, καθώς και το ονοματεπώνυμο του ασθενή που το έχει κλείσει. Επιπλέον, έχει τη δυνατότητα να ακυρώσει κάποιο ραντεβού, εφόσον χρειαστεί (Εικόνα 2.2.18: Διαγραφή ραντεβού).



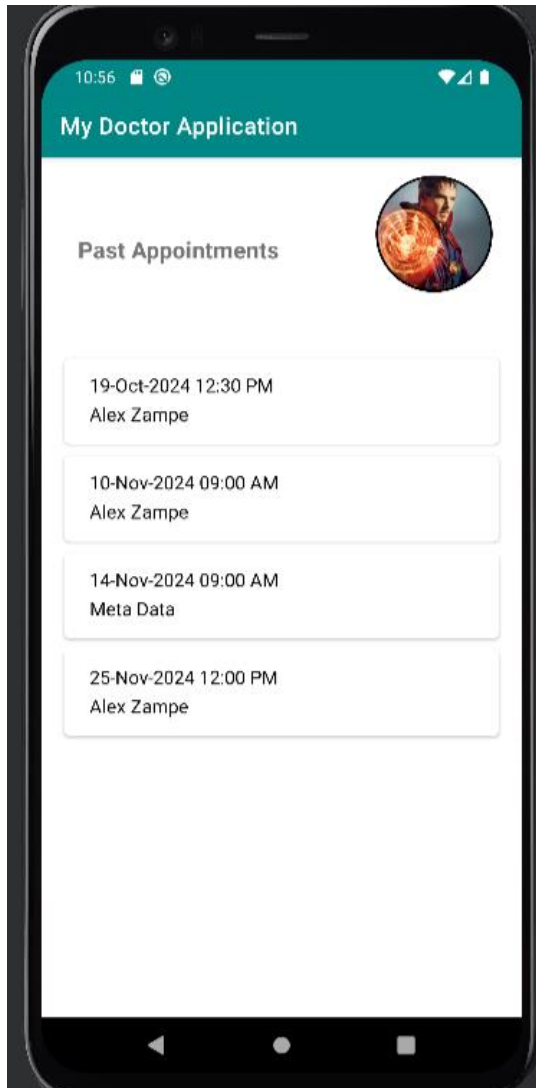
Εικόνα 2.2.17: Προβολή επερχόμενων ραντεβού



Εικόνα 2.2.18: Διαγραφή ραντεβού

2.2.5 Προβολή παρελθοντικών ραντεβού

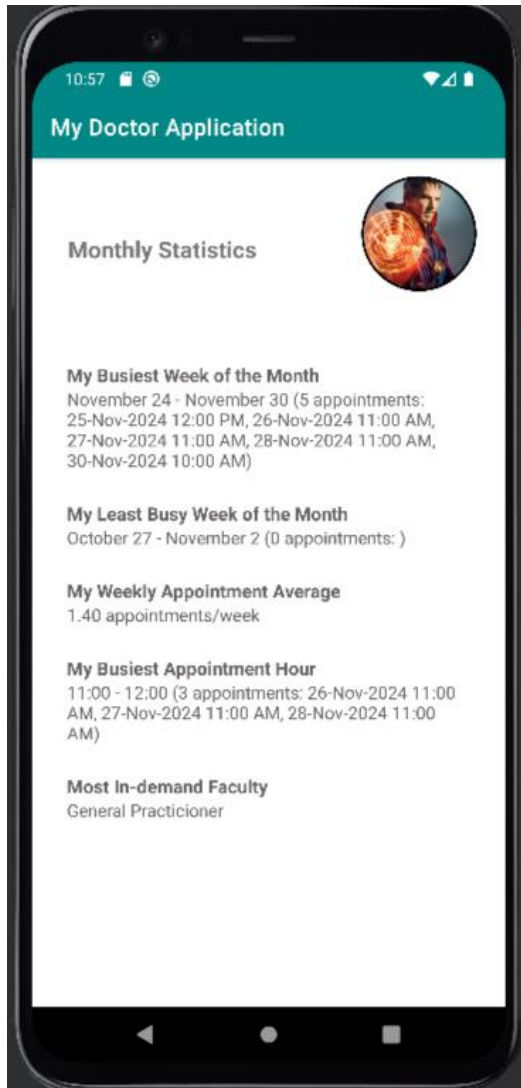
Όταν ο γιατρός επιλέξει την επιλογή «View past appointments» από την κεντρική σελίδα, μεταφέρεται σε μια σελίδα όπου εμφανίζονται τα παρελθοντικά ραντεβού που έχουν ήδη πραγματοποιηθεί (Εικόνα 2.2.19: Προβολή παρελθοντικών ραντεβού).



Εικόνα 2.2.19: Προβολή παρελθοντικών ραντεβού

2.2.5 Προβολή στατιστικών

Όταν ο γιατρός επιλέξει την επιλογή «Statistics» από την κεντρική σελίδα, μεταφέρεται σε μια σελίδα όπου εμφανίζονται μηνιαία στατιστικά στοιχεία που αφορούν την δραστηριότητά του. Τα στατιστικά περιλαμβάνουν την εβδομάδα του μήνα με τα περισσότερα ραντεβού, την εβδομάδα με τα λιγότερα ραντεβού, τον μέσο όρο ραντεβού ανά εβδομάδα για τον τελευταίο μήνα, καθώς και την πιο δημοφιλή ώρα για ραντεβού. Επιπλέον, αναφέρονται αναλυτικά τα ραντεβού ανά κατηγορία. Τέλος, παρουσιάζονται γενικά στατιστικά στοιχεία, όπως η πιο δημοφιλής ειδικότητα, δηλαδή η ειδικότητα με τον μεγαλύτερο αριθμό ραντεβού. (Εικόνα 2.2.20: Προβολή στατιστικών).



Εικόνα 2.2.20: Προβολή στατιστικών

Κεφάλαιο 3: Αρχιτεκτονική Εφαρμογής

3.1 Γλώσσα προγραμματισμού: Java

Η εφαρμογή αναπτύχθηκε εξ ολοκλήρου σε Java, μία από τις πλέον καθιερωμένες γλώσσες προγραμματισμού για την ανάπτυξη εφαρμογών Android. Κατόπιν έρευνας, οι πιο βασικοί λόγοι για τους οποίους επέλεξα την γλώσσα Java είναι για την σταθερότητα και ασφάλεια που προσφέρει. Η Java έχει σχεδιαστεί ώστε να παρέχει ασφάλεια μέσω του συστήματος διαχείρισης μνήμης, αποτρέποντας προβλήματα όπως οι διαρροές μνήμης (memory leaks), που μπορούν να επηρεάσουν την απόδοση της εφαρμογής.

Ένας εξίσου σημαντικός παράγοντας είναι η εκτεταμένη υποστήριξη από την κοινότητα προγραμματιστών. Με εκατομμύρια προγραμματιστές να χρησιμοποιούν τη Java, υπάρχουν άφθονοι πόροι, όπως φόρουμ, τεχνικά άρθρα, και tutorials, που διευκολύνουν την επίλυση προβλημάτων και την ανάπτυξη της εφαρμογής.

Τέλος, η Java προσφέρει αντικειμενοστραφή σχεδίαση, που διευκολύνει τη διαχείριση του κώδικα, ενισχύει την αναγνωσιμότητα και προάγει την επαναχρησιμοποίηση των κώδικα (code reusability). Αυτά τα χαρακτηριστικά συντελούν στη δημιουργία μιας πιο οργανωμένης και συντηρήσιμης εφαρμογής.

Συνολικά, η επιλογή της Java διασφαλίζει ότι η εφαρμογή είναι αξιόπιστη, ασφαλής, και εύκολα επεκτάσιμη, παρέχοντας ταυτόχρονα μία εξαιρετική εμπειρία τόσο στους προγραμματιστές όσο και στους τελικούς χρήστες.

3.2 Βάση δεδομένων: Firebase Realtime Database

3.2.1 Λόγοι επιλογής της Firebase Realtime Database

Η επιλογή της Firebase Realtime Database βασίστηκε στη δυνατότητά της να παρέχει συγχρονισμό δεδομένων σε πραγματικό χρόνο και στην απλοποιημένη αρχιτεκτονική της. Πρόκειται για μία βάση δεδομένων NoSQL που αποθηκεύει τα δεδομένα σε μορφή JSON και επιτρέπει τη συγχρονισμένη ενημέρωση μεταξύ των πελατών που είναι συνδεδεμένοι. Αυτή η λειτουργικότητα είναι ιδιαίτερα σημαντική για την εφαρμογή "My Doctor", καθώς διασφαλίζει την άμεση επικοινωνία και ενημέρωση μεταξύ γιατρών και ασθενών.

Επιπλέον, προσφέρει εξαιρετικό integration με το Android, παρέχοντας εργαλεία για authentication. Με τη χρήση των παρεχόμενων SDKs και εργαλείων, η διασύνδεση της εφαρμογής με τη βάση δεδομένων είναι αρκετά εύκολη και απλή. Επίσης, υποστηρίζεται από την Google, το οποίο σημαίνει ταυτόχρονα εγγυημένη ασφάλεια και προστασία των προσωπικών δεδομένων των χρηστών. Τέλος, ένας ακόμα λόγος επιλογής της συγκεκριμένης βάσης είναι η δυνατότητα επεκτασιμότητας. Καθώς η εφαρμογή μεγαλώνει και οι ανάγκες αυξάνονται, η Firebase είναι σε θέση να διαχειριστεί μεγαλύτερους όγκους δεδομένων και περισσότερους ταυτόχρονους χρήστες, χωρίς να απαιτείται εκτεταμένη αναδιάρθρωση της υποδομής.

3.2.2 Δομή της βάσης δεδομένων

Η βάση δεδομένων είναι οργανωμένη κάτω από τον κεντρικό κόμβο Users, ο οποίος περιέχει τέσσερις βασικούς υποκόμβους (sub-nodes):

Doctors: Περιλαμβάνει πληροφορίες για όλους τους εγγεγραμμένους γιατρούς.

Patients: Περιέχει δεδομένα σχετικά με τους εγγεγραμμένους ασθενείς.

Metadata: Αποθηκεύει τον ρόλο (γιατρός ή ασθενής) κάθε χρήστη.

AppointmentsFaculty: Καταγράφει στατιστικά σχετικά με τα ραντεβού ανά ειδικότητα γιατρών.

Κάτω από τον κόμβο Users > Doctors, κάθε γιατρός έχει το δικό του UID ως μοναδικό κλειδί. Εκεί αποθηκεύονται προσωπικές πληροφορίες όπως το όνομα, η ειδικότητα, η διεύθυνση, ο αριθμός αδείας, το τηλέφωνο, οι υπηρεσίες, οι τιμές του και ένα sUrl που περιέχει το link της φωτογραφίας προφίλ του. Επίσης, υπάρχει ένα boolean πεδίο “patient” το οποίο δείχνει ότι το συγκεκριμένο προφίλ είναι γιατρός (“false”). Οι γιατροί μπορούν να προσθέτουν τις διαθέσιμες ώρες τους μέσω του υποκόμβου “available_slots”. Κάθε διαθέσιμο slot αποθηκεύεται ως ξεχωριστό node με πληροφορίες όπως η ημερομηνία, η ώρα, το όνομα του γιατρού, το UID του και ένα boolean πεδίο “isBooked” που δείχνει αν το slot είναι διαθέσιμο ή έχει κλειστεί. Όταν δημιουργείται ένα νέο slot, υπάρχουν επίσης κάποια πεδία που αφορούν τον ασθενή (patient’s UID και patient’s full name) τα οποία αρχικά παραμένουν κενά. Όταν ένας ασθενής κλείσει ένα ραντεβού, το αντίστοιχο slot αντιγράφεται στον κόμβο “booked”, και τα πεδία που σχετίζονται με τον ασθενή ενημερώνονται. Το πεδίο “isBooked” ενημερώνεται αυτόματα σε true.

Κάτω από τον κόμβο Users > Patients, κάθε ασθενής έχει επίσης το δικό του UID ως μοναδικό κλειδί. Αποθηκεύονται προσωπικά στοιχεία όπως όνομα, τηλέφωνο και διεύθυνση. Επίσης, το boolean πεδίο “patient” δείχνει αντίστοιχα ότι το συγκεκριμένο προφίλ είναι ασθενής (“true”). Όταν ένας ασθενής κλείνει ραντεβού, τα σχετικά δεδομένα αποθηκεύονται στον κόμβο των κλεισμένων ραντεβού του “appointments”. Αυτά περιλαμβάνουν πληροφορίες για τον γιατρό, την ημερομηνία και την ώρα του ραντεβού.

Ο κόμβος Users > Metadata αποθηκεύει ένα απλό boolean πεδίο “patient” για κάθε χρήστη, καθορίζοντας τον ρόλο του. Κατά την είσοδο στην εφαρμογή, γίνεται έλεγχος αυτού του πεδίου για να καθοριστεί ο ρόλος του χρήστη. Αν το “patient” είναι true, ο χρήστης είναι ασθενής και ενεργοποιούνται τα αντίστοιχα activities. Αν είναι false, ο χρήστης είναι γιατρός.

Ο κόμβος Users > AppointmentsFaculty καταγράφει τα στατιστικά ραντεβού ανά ειδικότητα. Για κάθε ειδικότητα (π.χ., General Practitioner, Dentist), αποθηκεύεται ένας μετρητής που αυξάνεται κατά 1 κάθε φορά που κλείνεται ένα ραντεβού με γιατρό αυτής της ειδικότητας ή αντίστοιχα μειώνεται κατά 1 κάθε φορά που ακυρώνεται ένα ραντεβού. Αυτή η λειτουργία επιτρέπει στους γιατρούς να βλέπουν ποια ειδικότητα έχει τη μεγαλύτερη ζήτηση μέσω ενός χαρακτηριστικού στατιστικού (most in-demand faculty).

3.2.3 Χρήση και Ενημέρωση Δεδομένων σε Πραγματικό Χρόνο

Η χρήση της Firebase Realtime Database επιτρέπει τον συγχρονισμό δεδομένων σε πραγματικό χρόνο. Για παράδειγμα, όταν ένας γιατρός προσθέσει νέα διαθέσιμα slots ή τροποποιήσει τη διαθεσιμότητά του, οι αλλαγές εμφανίζονται άμεσα στους ασθενείς που αναζητούν ραντεβού. Όταν ένας ασθενής κλείσει ένα ραντεβού, η αλλαγή αυτή αντικατοπτρίζεται σε πραγματικό χρόνο τόσο στον κόμβο του γιατρού όσο και σε αυτόν του ασθενούς, εξασφαλίζοντας ομαλή και γρήγορη λειτουργία.

3.2.4 Ασφάλεια και Authentication

Η ασφάλεια της βάσης δεδομένων διασφαλίζεται μέσω των Security Rules της Firebase. Κάθε χρήστης έχει πρόσβαση μόνο στα δικά του δεδομένα ή σε ό,τι είναι απαραίτητο για τη λειτουργία της εφαρμογής. Επιπλέον, το σύστημα Authentication της Firebase εξασφαλίζει ότι μόνο οι εξουσιοδοτημένοι χρήστες μπορούν να εισέλθουν στην εφαρμογή.

3.3 Δομή εφαρμογής

Η εφαρμογή My Doctor έχει σχεδιαστεί με γνώμονα την απλότητα και την ευχρηστία, αξιοποιώντας βασικά components του Android για τη διαχείριση του UI και των δεδομένων. Όλη η λογική της εφαρμογής είναι οργανωμένη με χρήση ξεχωριστών Activities, καθένα από τα οποία αναλαμβάνει μια συγκεκριμένη λειτουργία.

3.3.1 Activity-based UI Design

Το περιβάλλον χρήστη (UI) της εφαρμογής υλοποιείται αποκλειστικά με Activities, τα οποία εξυπηρετούν τις παρακάτω λειτουργίες:

Login: Χειρίζεται τη διαδικασία ταυτοποίησης του χρήστη (ασθενής ή γιατρός).

Register: Χειρίζεται τη διαδικασία εγγραφής νέου χρήστη (ασθενή).

MainActivity: Εμφανίζει την κεντρική σελίδα του ασθενή.

DoctorProfile: Επιτρέπει στους ασθενείς να κλείσουν ραντεβού επιλέγοντας διαθέσιμες ώρες γιατρών.

PatientProfile: Προβάλλει την καρτέλα του ασθενή και τα επερχόμενα ραντεβού με γιατρούς, δίνοντας τη δυνατότητα ακύρωσης αυτών.

EditProfile: Επιτρέπει στον ασθενή να δει και να επεξεργαστεί τα προσωπικά του στοιχεία.

DoctorsMain: Εμφανίζει την κεντρική σελίδα του γιατρού.

DocEditProfile: Επιτρέπει στους γιατρούς να προβάλλουν και να επεξεργαστούν τα προσωπικά τους στοιχεία.

DocAddSlots: Επιτρέπει στους γιατρούς να προσθέσουν διαθέσιμες ώρες για ραντεβού.

DocPreviewSlots: Προβάλλει και επιτρέπει τη διαγραφή των διαθέσιμων ωρών του γιατρού.

DocUpcomingAppnts: Προβάλλει τα επερχόμενα ραντεβού με ασθενείς και δίνει τη δυνατότητα ακύρωσης αυτών.

DocPastAppnts: Εμφανίζει τα παρελθοντικά ραντεβού με ασθενείς.

DocStatistics: Παρουσιάζει στατιστικά στοιχεία στους γιατρούς.

Κάθε Activity είναι υπεύθυνο για τη φόρτωση δεδομένων και την αλληλεπίδραση του χρήστη, χρησιμοποιώντας τα διαθέσιμα δεδομένα από τη Firebase.

3.3.2 Αρχιτεκτονική MVC

Η δομή της εφαρμογής βασίζεται στο πρότυπο Model-View-Controller (MVC):

- **Model:** Περιλαμβάνει τα δεδομένα της εφαρμογής, όπως ασθενείς (Patient), γιατρούς (Doctor, MainModel), διαθέσιμες ώρες (Slot) και πληροφορίες για τον ρόλο του χρήστη (Metadata). Πιο συγκεκριμένα:

Patient: Περιέχει τα δεδομένα των ασθενών, όπως όνομα, επώνυμο, τηλέφωνο, ΑΜΚΑ, και ημερομηνία γέννησης, που είναι υποχρεωτικά πεδία κατά την εγγραφή.

Doctor: Δημιουργήθηκε για να ανακτά δεδομένα όπως αριθμός άδειας, email, πλήρες όνομα, τοποθεσία, τηλέφωνο, υπηρεσίες, ειδικότητα, σύνδεσμος φωτογραφίας (sUrl), καθώς και γεωγραφικό πλάτος και μήκος (latitude, longitude). Τα δεδομένα αυτά εμφανίζονται στο RecyclerView της δραστηριότητας DoctorProfile Activity.

MainModel: Δημιουργήθηκε για να ανακτά δεδομένα σχετικά με τους γιατρούς, όπως πλήρες όνομα, ειδικότητα, τοποθεσία, σύνδεσμος φωτογραφίας (sUrl), γεωγραφικό πλάτος και μήκος (latitude, longitude). Αυτά τα δεδομένα εμφανίζονται στο RecyclerView της MainActivity, όπου προβάλλονται συγκεκριμένα μόνο στοιχεία που αφορούν τους γιατρούς.

Slot: Δημιουργήθηκε για να αποθηκεύει στη βάση δεδομένων πληροφορίες σχετικά με την ημερομηνία και την ώρα μιας διαθέσιμης ώρας, καθώς και στοιχεία του γιατρού που τη δημιούργησε (όπως όνομα, ειδικότητα και φωτογραφία) και του πιθανού ασθενούς που μπορεί να την κλείσει. Περιλαμβάνει πεδία που καταγράφουν τη χρονική στιγμή του slot, αν έχει κλειστεί, το ID του γιατρού και του ασθενούς, καθώς και άλλα στοιχεία απαραίτητα για την επικοινωνία μεταξύ τους.

Metadata: Αποθηκεύει πληροφορίες για το ρόλο του χρήστη μέσω του boolean πεδίου isPatient, το οποίο ελέγχει αν ο χρήστης που κάνει login είναι ασθενής ή γιατρός. Αυτός ο έλεγχος πραγματοποιείται στη δραστηριότητα Login Activity, ώστε να ανοίγει το αντίστοιχο σύνολο από activities, ανάλογα με τον ρόλο του χρήστη. Ένας ξεχωριστός κόμβος με την ονομασία "Metadata" δημιουργήθηκε στη Firebase, για να διευκολύνει και να επιταχύνει την αναζήτηση κατά τη διαδικασία του login.

- View: Αποτελεί το UI που προβάλλεται στον χρήστη μέσω των XML Layouts των Activities. Στην εφαρμογή μου περιλαμβάνει τα UI layouts που σχεδιάστηκαν σε XML για κάθε Activity και επιπλέον XML αρχεία που χρησιμοποιήθηκαν για την προσαρμογή των στοιχείων του RecyclerView.

main_item: Το card layout δημιουργήθηκε για να χρησιμοποιηθεί από το RecyclerView στη δραστηριότητα MainActivity. Εμφανίζει λίστα με τους γιατρούς, προβάλλοντας συγκεκριμένα στοιχεία για τον καθένα, όπως το όνομα, την ειδικότητα και την τοποθεσία τους.

patients_appointments: Το layout δημιουργήθηκε για να εμφανίζει τη λίστα με τα κλεισμένα ραντεβού των ασθενών. Παρέχει στους ασθενείς τη δυνατότητα να ακυρώσουν τα ραντεβού τους και εμφανίζει λεπτομέρειες, όπως την ημερομηνία, την ώρα, το όνομα του γιατρού και την ειδικότητά του. Χρησιμοποιείται στη δραστηριότητα PatientProfile Activity.

doctors_past_appnts: Το layout δημιουργήθηκε για να εμφανίζει τη λίστα με τα ραντεβού που έχουν ήδη πραγματοποιηθεί μεταξύ του γιατρού και των ασθενών του. Χρησιμοποιείται στη δραστηριότητα DocPastAppnts Activity.

doctors_slots: Το layout δημιουργήθηκε για να εμφανίζει τη λίστα με τις διαθέσιμες ώρες που έχει προσθέσει ο γιατρός. Παρέχει τη δυνατότητα διαγραφής συγκεκριμένων slots, αν ο γιατρός το επιθυμεί. Χρησιμοποιείται στη δραστηριότητα DocPreviewSlots Activity.

doctors_upcoming_appnts: Το layout δημιουργήθηκε για να εμφανίζει τη λίστα με τα κλεισμένα ραντεβού των ασθενών του γιατρού. Παρέχει τη δυνατότητα στον γιατρό να ακυρώσει κάποιο από τα ραντεβού. Χρησιμοποιείται στη δραστηριότητα DocUpcomingAppnts Activity.

Η σύνδεση αυτών των XML αρχείων με τον κώδικα γίνεται μέσω των adapters του RecyclerView, οι οποίοι μεταφέρουν δεδομένα από το Model στο View.

- Controller: Υλοποιείται μέσα στις Activities και αναλαμβάνει την επικοινωνία μεταξύ του View και του Model. Οι Activities περιλαμβάνουν τη λογική για την ανάκτηση, επεξεργασία και αποθήκευση δεδομένων μέσω της Firebase.

3.3.3 Επικοινωνία Front-end με Firebase

Η σύνδεση μεταξύ του UI και της Firebase Realtime Database πραγματοποιείται μέσω των Activities, που χρησιμοποιούν τις Firebase SDK APIs για την ανάκτηση και αποστολή δεδομένων. Παραδείγματα:

Ανάκτηση δεδομένων γιατρών από τη βάση δεδομένων Firebase σε ένα RecyclerView:

```
// pull the data from firebase database, from doctors child in the form of the MainModel
FirebaseRecyclerOptions<MainModel> options = new FirebaseRecyclerOptions.Builder<MainModel>()
    .setQuery(FirebaseDatabase.getInstance().getReference().child(USERS_PATH).child(DOCTORS
PATH), MainModel.class)
```

```

        .build();

        FirebaseRecyclerAdapter<MainModel,myViewHolder> adapter =
            new FirebaseRecyclerAdapter<MainModel, myViewHolder>(options) {
                @Override
                protected void onBindViewHolder(@NonNull myViewHolder holder, int position,
                @NonNull MainModel model) {

                    // display the data from firebase database into the relevant textviews on the
                    main_item layout
                    String fullName = model.getFirst_name() + " " + model.getLast_name();
                    holder.name.setText(fullName);
                    holder.specialty.setText(model.getSpecialty());
                    holder.location.setText(model.getLocation());
                    holder.distance.setText("");

                    // call method calculateDistance
                    float finalDistance = calculateDistance(model.getDoc_latitude(),
                    model.getDoc_longitude());
                    // Show only one decimal point
                    String strDouble = String.format("%.1f", finalDistance);
                    holder.distance.setText(strDouble + "km from you");

                    // get the photo url from the database and use glide to display the photo
                    Glide.with(holder.img.getContext())
                        .load(model.getGurl())
                        .placeholder(com.firebase.ui.database.R.drawable.common_google_signin_b
                    tn_icon_dark)
                        .circleCrop()
                        // if any error occurs it will display the selected image
                    n_dark_normal)
                        .error(com.firebase.ui.database.R.drawable.common_google_signin_btn_ico
                        .into(holder.img);

                    holder.btnBook.setOnClickListener(new View.OnClickListener() {
                        @Override
                        public void onClick(View v) {
                            // we can't use the position argument to handle user clicks because if
                            any of the items in the recycler view change (deleted/added/changed)
                            // RecyclerView will not call onBindViewHolder method and update all
                            items' positions, it will only update the position of the new ones
                            // the getAdapterPosition will always have the updated adapter's
                            position of this holder
                            int currentPosition = holder.getAdapterPosition();
                            String doctorID = getRef(currentPosition).getKey();
                            // pass the doctor's uid that the patient selected to the DoctorProfile
                            activity
                            Intent profileIntent = new Intent(MainActivity.this,
                            DoctorProfile.class);
                            profileIntent.putExtra("doctorID",doctorID);
                            startActivity(profileIntent);
                        }
                    });

                    @NonNull
                    @Override
                    public myViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
                        // bind the main_item card layout that we created with the recyclerview
                        View view =
                        LayoutInflater.from(parent.getContext()).inflate(R.layout.main_item,parent,false);
                        myViewHolder viewHolder = new myViewHolder(view);
                        return viewHolder;
                    }
                }
            };

            recyclerView.setAdapter(adapter);
            adapter.startListening();
    }
}

```

Ενημέρωση της βάσης κατά το κλείσιμο ραντεβού:

```

String selectedTimeslot = getRef(currentPosition).getKey();

        databaseReferenceDoctor.child("available_slots").child(selectedTimeslot)
            .addListenerForSingleValueEvent(new ValueEventListener() {
                @Override
                public void onDataChange(@NonNull DataSnapshot snapshot) {
                    if (snapshot.exists()) {
                        Slot selectedSlot = snapshot.getValue(Slot.class);
                        if (selectedSlot != null) {

```

```

        selectedSlot.setIsBooked(true);
        selectedSlot.setPatientId(uid);
        selectedSlot.setpFullName(pFullName);

        Map<String, Object> slotUpdates = new HashMap<>();
        slotUpdates.put("isBooked", true);
        slotUpdates.put("patientId", uid);
        slotUpdates.put("pFullName", pFullName);

        databaseReferenceDoctor.child("available_slots").child(selectedTimeslot)
    ).updateChildren(slotUpdates)
        .addOnCompleteListener(task -> {
            if (task.isSuccessful()) {
                databaseReferenceDoctor.child("booked").child(selectedTi
meslot).setValue(selectedSlot);
                databaseReferencePatient.child("appointments").child(sel
ectedTimeslot).setValue(selectedSlot);

                DatabaseReference facultyRef =
                FirebaseDatabase.getInstance().getReference()
                    .child("Users").child("AppointmentsFacul
ty").child(dSpecialty);

                facultyRef.addListenerForSingleValueEvent(new
                ValueEventListener() {
                    @Override
                    public void onDataChange(@NonNull DataSnapshot
                    snapshot) {
                        // get current count, default to 0 if not set
                        long currentCount = snapshot.exists() ?
                        snapshot.getValue(Long.class) : 0;

                        // increment by 1
                        facultyRef.setValue(currentCount + 1);
                    }

                    @Override
                    public void onCancelled(@NonNull DatabaseError
                    error) {
                        Toast.makeText(getApplicationContext(), "Failed
                        to update faculty count", Toast.LENGTH_LONG).show();
                    }
                });

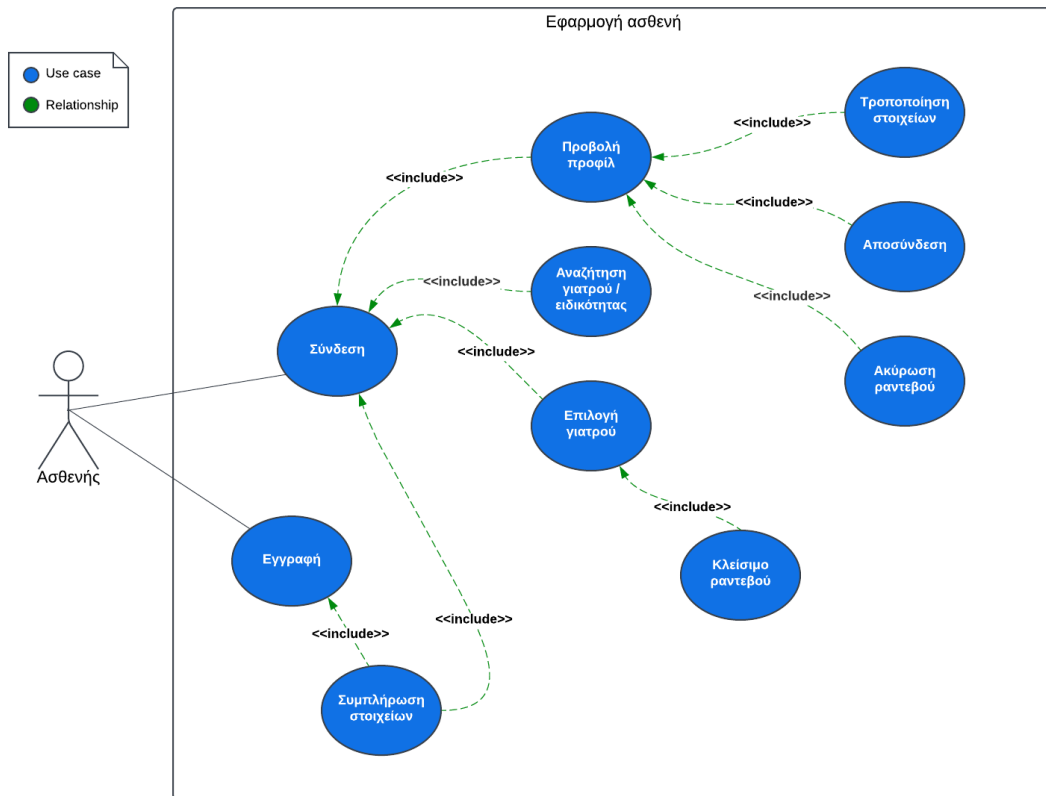
                Toast.makeText(getApplicationContext(), "Appointment
                booked successfully", Toast.LENGTH_LONG).show();
            } else {
                Toast.makeText(getApplicationContext(), "Failed to book
                appointment", Toast.LENGTH_SHORT).show();
            }
        });
    }

    @Override
    public void onCancelled(@NonNull DatabaseError error) {
    }
});

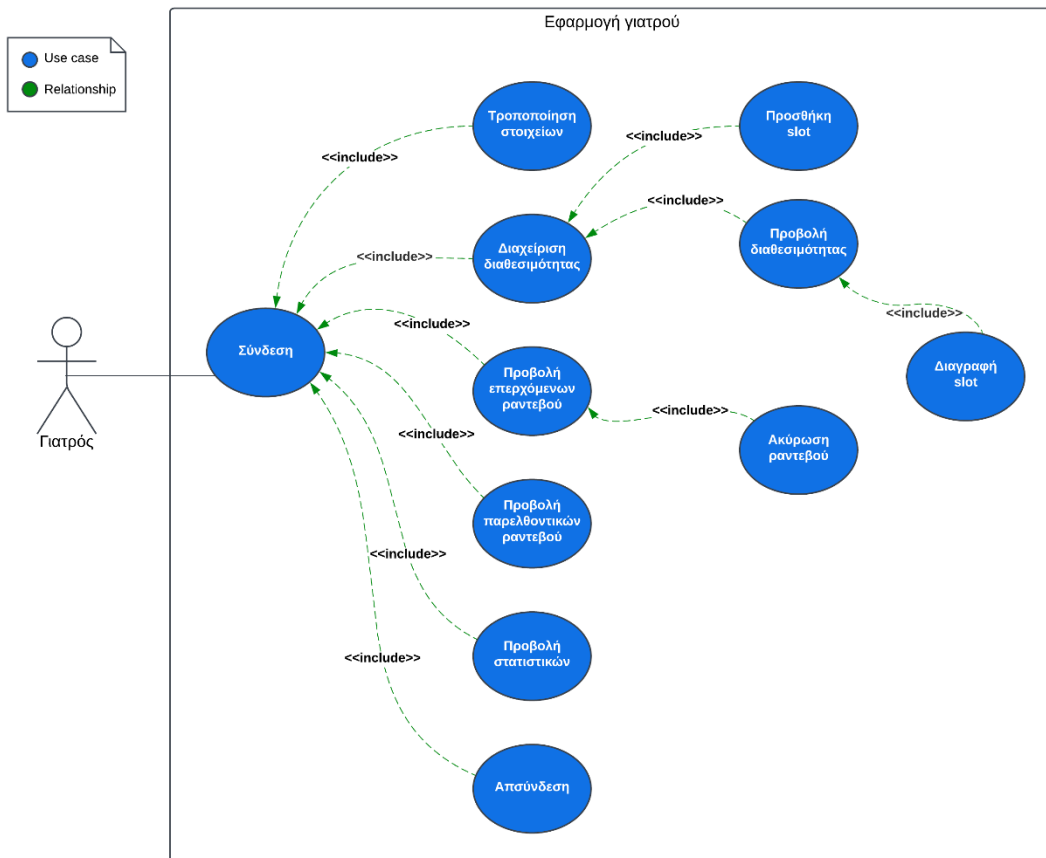
```

3.4 Διαγράμματα

3.4.1 Διάγραμμα περίπτωσης χρήσης ασθενή



3.4.2 Διάγραμμα περίπτωσης χρήσης γιατρού



Κεφάλαιο 4: Συμπεράσματα και Μελλοντικές Επεκτάσεις

Η εφαρμογή "My Doctor" έχει καταφέρει να δημιουργήσει μια αρκετά ολοκληρωμένη εμπειρία για τους χρήστες της, καλύπτοντας τόσο τις ανάγκες των ασθενών όσο και των γιατρών. Παρόλο που έχει καταφέρει να διευκολύνει τη διαδικασία κλεισίματος ραντεβού και διαχείρισης διαθεσιμότητας γιατρών, υπάρχουν ακόμη δυνατότητες βελτίωσης και επεκτάσεων που θα μπορούσαν να υλοποιηθούν σε μια τέτοια εφαρμογή. Η ανάπτυξη και βελτιστοποίηση εφαρμογών απαιτεί σημαντικό χρόνο και συνεχή αναβάθμιση, με αποτέλεσμα το "My Doctor" να μην είναι ακόμα "τέλειο". Ο κύριος στόχος της εφαρμογής παραμένει να προσφέρει όσο το δυνατόν πιο ακριβή και ενημερωμένα δεδομένα σχετικά με τους γιατρούς και τη διαθεσιμότητά τους, να προσφέρει ακριβέστερη αναζήτηση και φιλτράρισμα γιατρών, με βάση τις προτιμήσεις και τις ανάγκες των χρηστών, και να παραμείνει όσο το δυνατόν πιο εύχρηστη και απλή στη χρήση για μια εύκολη διαδικασία κλεισίματος ραντεβού.

Κατά την ανάπτυξη της εφαρμογής, αντιμετωπίστηκαν αρκετές προκλήσεις, τόσο από τεχνικής όσο και από οργανωτικής πλευράς. Μια από τις μεγαλύτερες προκλήσεις ήταν η οργάνωση της βάσης δεδομένων, η οποία απαιτούσε αρκετή έρευνα πριν από την ανάπτυξη της εφαρμογής για να διασφαλιστεί ότι θα μπορούσε να υποστηρίξει τον σωστό συγχρονισμό των δεδομένων και την διαχείριση των χρηστών. Ειδικότερα, έπρεπε να σχεδιαστεί η βάση με τέτοιο τρόπο ώστε να επιτρέπει τον έλεγχο του ρόλου του χρήστη (ασθενής ή γιατρός) κατά την είσοδό του στην εφαρμογή. Αυτό επιτεύχθηκε μέσω της προσθήκης μιας boolean μεταβλητής "patient" η οποία προσδιόριζε τον ρόλο του χρήστη και παρουσίαζε τα σχετικά activities.

Μια άλλη σημαντική και απαιτητική πρόκληση ήταν η σωστή αποθήκευση των διαθέσιμων slots στη βάση δεδομένων. Η διαδικασία αυτή ήταν κρίσιμη για την ανάκτηση, φιλτράρισμα και παρουσίαση των δεδομένων στο μέλλον, αλλά και για την υποστήριξη της συλλογής στατιστικών στοιχείων για τους γιατρούς. Το slotTime (χρονική σήμανση σε μορφή Unix), που αποθηκεύεται στην κλάση Slot, επέτρεπε την ταξινόμηση και τον φιλτράρισμα των δεδομένων, ενώ ταυτόχρονα ήταν καθοριστικό για την εξαγωγή στατιστικών, όπως για παράδειγμα το activity του τελευταίου ημερολογιακού μήνα. Με βάση το slotTime, οι γιατροί μπορούν να δουν ποια εβδομάδα του τρέχοντος μήνα είχε τα περισσότερα ή λιγότερα ραντεβού, χωρίς να περιορίζονται σε δεδομένα "all time". Επίσης ήταν απολύτως απαραίτητο να διασφαλιστεί ότι τα slots δεν επικαλύπτονται χρονικά και ότι δεν μπορεί να κλειστεί ραντεβού από δύο διαφορετικούς ασθενείς για την ίδια ακριβώς ώρα. Αυτό επιτεύχθηκε με την εισαγωγή του boolean isBooked στην κλάση Slot, το οποίο υποδεικνύει αν ένα slot έχει ήδη δεσμευτεί. Με αυτό τον τρόπο, κατά την προβολή των διαθέσιμων slots, το σύστημα ελέγχει την τιμή του isBooked και αποτρέπει την ταυτόχρονη κράτηση από δύο ασθενείς. Τέλος, με τους κατάλληλους ελέγχους και μηχανισμούς επαλήθευσης στη βάση δεδομένων Firebase, εξασφαλίστηκε ότι οι παρελθοντικές ημερομηνίες, οι διπλές εγγραφές ή τα μη έγκυρα slots δεν καταχωρούνται, υποστηρίζοντας τη σωστή λειτουργία της διαχείρισης ραντεβού στην εφαρμογή.

Στα πλαίσια της περαιτέρω βελτίωσης της εφαρμογής, ο στόχος είναι να τελειοποιηθεί η υπάρχουσα λειτουργικότητα και να προστεθούν νέες δυνατότητες. Μερικές από τις μελλοντικές επεκτάσεις της εφαρμογής περιλαμβάνουν την υλοποίηση ενός συστήματος ειδοποιήσεων για ακυρώσεις ραντεβού ή αλλαγές στη διαθεσιμότητα των γιατρών, καθώς και υπενθυμίσεις για επερχόμενα ραντεβού. Επιπλέον, η εφαρμογή θα μπορούσε να ενσωματώσει ένα σύστημα online πληρωμών, επιτρέποντας στους γιατρούς να χρεώνουν άμεσα τους ασθενείς μέσω της εφαρμογής (εφόσον οι ασθενείς έχουν καταχωρήσει τα στοιχεία της κάρτας τους) και να τους αποστέλλουν την απόδειξη μέσω της εφαρμογής, χωρίς την ανάγκη χρήσης POS ή εκτύπωσης αποδείξεων.

Μια άλλη χρήσιμη δυνατότητα θα ήταν η προσθήκη πεδίων στο προφίλ του χρήστη, ώστε να καταχωρούνται πληροφορίες σχετικά με τυχόν χρόνιες παθήσεις και τη συχνότητα των επισκέψεων που απαιτούνται για την παρακολούθησή τους. Αυτό θα επέτρεπε στην εφαρμογή να υπενθυμίζει στους χρήστες την ανάγκη για τακτικές επισκέψεις και να τους καθοδηγεί να κλείσουν ραντεβού με τον γιατρό που είχαν δει προηγουμένως, διευκολύνοντας την έγκαιρη και σωστή παρακολούθηση της υγείας τους.

Σημαντική βελτίωση θα μπορούσε να γίνει και στον τομέα των στατιστικών. Με τη χρήση δεδομένων από προηγούμενα χρόνια ή με την αξιοποίηση του AI, θα μπορούσαν να παρέχονται στους γιατρούς πιο αναλυτικές και χρήσιμες πληροφορίες σχετικά με τη δραστηριότητά τους. Θα μπορούσε να γίνεται ανάλυση των τάσεων της ζήτησης ανά ημέρα και ώρα, επιτρέποντας στους γιατρούς, είτε ατομικά είτε συνολικά σαν ειδικότητα, να προβλέπουν τις περιόδους με την περισσότερη ζήτηση και να οργανώνουν πιο αποδοτικά το πρόγραμμά τους. Το AI θα μπορούσε επίσης να προβλέπει την πιθανότητα ακύρωσης ενός ραντεβού με βάση την ιστορική συμπεριφορά των χρηστών, ή να αναγνωρίζει και να προβλέπει τις ανάγκες για επείγοντα περιστατικά, αξιοποιώντας εποχιακά δεδομένα ή ιστορικές τάσεις όπως αυξημένες επισκέψεις σε παιδίατρος το χειμώνα λόγω λοιμώξεων. Μια άλλη χρήσιμη δυνατότητα του AI θα ήταν η αυτόματη διαχείριση ακυρωμένων slots. Όταν ένα ραντεβού ακυρώνεται, η εφαρμογή θα μπορούσε να αναγνωρίζει ασθενείς που ενδέχεται να ενδιαφέρονται για το ελεύθερο slot, με βάση τη συχνότητα των προηγούμενων ραντεβού που έχουν κλείσει. Για παράδειγμα, εάν ένας ασθενής κλείνει συχνά ραντεβού μεταξύ 7:00 και 8:00 μ.μ., και έχει προγραμματίσει το επόμενο ραντεβού του στις 5:00 μ.μ., τότε αν κάποιο ραντεβού μεταξύ 7:00 και 8:00 μ.μ. ακυρωθεί, η εφαρμογή θα αποστέλλει αυτόματα ειδοποίηση στον ασθενή που ταιριάζει στο προφίλ του, σε περίπτωση που η καινούρια ώρα τον εξυπηρετεί καλύτερα. Τέλος, θα μπορούσε να συμβάλει στην αξιολόγηση της ικανοποίησης των χρηστών, αναλύοντας τα σχόλια και τις αξιολογήσεις τους. Μέσω αυτής της ανάλυσης, θα μπορούσε να εντοπίζει περιοχές για βελτίωση στην εμπειρία των χρηστών και να προτείνει αλλαγές τόσο σε επίπεδο εφαρμογής όσο και σε επίπεδο προσφερόμενων υπηρεσιών για την καλύτερη εξυπηρέτησή τους.

Όλες αυτές οι προσθήκες, σε συνδυασμό με τη συνεχή βελτίωση της εφαρμογής, θα επιτρέπουν στην εφαρμογή να καλύπτει πλήρως τις ανάγκες των χρηστών και να παραμένει μια σύγχρονη, εύχρηστη και λειτουργική λύση για τη διαχείριση ραντεβού με γιατρούς.

Σε έναν κόσμο που κινείται ολοένα και περισσότερο προς την ψηφιοποίηση, εφαρμογές όπως το "My Doctor" παίζουν καθοριστικό ρόλο στη βελτίωση του συστήματος υγείας. Η υλοποίηση των παραπάνω επεκτάσεων δεν θα διευκολύνει μόνο τη χρήση της εφαρμογής, αλλά θα μπορούσε να αποτελέσει οδηγό για τη βελτίωση της παροχής υπηρεσιών υγείας γενικότερα. Ειδικότερα, η ανάλυση δεδομένων και η πρόβλεψη ζήτησης μπορεί να οδηγήσουν σε καλύτερη κατανομή πόρων και στη μείωση της αναμονής για τους ασθενείς. Επιπλέον, η δυνατότητα σύνδεσης με άλλες υπηρεσίες, όπως ηλεκτρονικές συνταγές ή φαρμακεία, ανοίγει τον δρόμο για τη δημιουργία ενός ολοκληρωμένου οικοσυστήματος υγείας.

Κεφάλαιο 5: Βιβλιογραφία

L. Joshua Crotts: Learning Java - A Test-Driven Approach. Springer 2024, ISBN 978-3-031-66637-7

Eugenia A. Politou, Efthimios Alepis, Constantinos Patsakis: A survey on mobile affective computing. *Comput. Sci. Rev.* 25: 79-100 (2017)

Efthimios Alepis, Constantinos Patsakis: Hey Doc, Is This Normal?: Exploring Android Permissions in the Post Marshmallow Era. *SPACE 2017*: 53-73

Efthimios Alepis, Maria Virvou: Object-Oriented User Interfaces for Personalized Mobile Learning. *Intelligent Systems Reference Library 64*, Springer 2014, ISBN 978-3-642-53850-6, pp. 1-129

Margaretha Ohyver, Jurike V. Moniaga, Iwa Sungkawa, Bonifasius Edwin Subagyo, Ian Argus Chandra: The Comparison Firebase Realtime Database and MySQL Database Performance using Wilcoxon Signed-Rank Test. *ICCSCI 2019*: 396-405

Konstantinos Rammos, Spyros Papadimitriou, Maria Virvou, Efthimios Alepis: A Web-Based Application for Innovative Hospital Appointment Scheduling Using Neural Network. *IISA 2018*: 1-6

Alex Kuiper, Michel Mandjes, Jeroen de Mast, Ruben Brokkelkamp: A flexible and optimal approach for appointment scheduling in healthcare. *Decis. Sci.* 54(1): 85-100 (2023)

Heng Zhao, Sijia Zhou: Foot in both camps: How do activities on third-party online healthcare platforms affect doctors' demand on official online healthcare platforms? *Decis. Support Syst.* 188: 114350 (2025)

Catalina Valenzuela-Núñez, Guillermo Latorre-Núñez, Fredy Troncoso-Espinosa: Smart Medical Appointment Scheduling: Optimization, Machine Learning, and Overbooking to Enhance Resource Utilization. *IEEE Access* 12: 7551-7562 (2024)

Menglei Ji, Mohammad Mosaffa, Amir Ardestani-Jaafari, Jinlin Li, Chun Peng: Integration of text-mining and telemedicine appointment optimization. *Ann. Oper. Res.* 341(1): 621-645 (2024)

Jie Zhou, Peng Guo, Miaomiao Yu, Hairong Feng: Performance evaluation of an appointment-based medical service system with no-shows and emergency arrivals. *Comput. Ind. Eng.* 196: 110518 (2024)

What is Firebase? Available at: <https://www.educative.io/edpresso/what-is-firebase>

Firebase Realtime Database Documentation. Available at: <https://firebase.google.com/docs/database>

Mastering Firebase Databases: Real-time and Cloud Firestore. Available at: <https://ionicthemes.com/tutorials/mastering-firebase-databases-real-time-and-cloud-firestore>

Firebase Database Design for NoSQL Challenges. Available at: <https://www.restack.io/p/firebase-database-design-answer-nosql-challenges-ai-cat-ai>

Firebase, Firebase Realtime Database, and Firestore. Available at: <https://medium.com/@ramadan123sayed/firebase-firebase-realtime-database-and-firestore-8b4fcddb1059>

Πόσοι (και ποιοι) έχουν smartphone στην Ελλάδα – Πόσοι δεν έχουν καν κινητό. Available at: <https://www.news247.gr/ellada/posoi-kai-poioiden-exoun-smartphone-stin-ellada-posoi-den-exoun-kan-kinito/>