



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	ΠΡΟΓΡΑΜΜΑ ΠΡΩΤΩΝ ΒΟΗΘΕΙΩΝ ΜΕ ΧΡΗΣΗ ΤΗΣ ONLINE ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ FIREBASE ΚΑΙ ΤΟΥ WEB SERVICE SPRING BOOT. FIRST AID PROGRAM USING THE ONLINE DATA BASE FIREBASE AND THE SPRING BOOT WEB SERVICE.
Όνοματεπώνυμο Φοιτητή	Γεράσιμος Φλέγκας
Πατρώνυμο	Θεόδωρος
Αριθμός Μητρώου	ΜΠΠΛ17057
Επιβλέπων	Ευθύμιος Αλέπης, Αναπληρωτής Καθηγητής

Ημερομηνία Παράδοσης **Σεπτέμβριος 2020**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

Ευθύμιος Αλέπης
Αναπληρωτής Καθηγητής

(υπογραφή)

Βίρβου Μαρία
Καθηγητής

(υπογραφή)

Πατσάκης
Κωνσταντίνος
Επίκουρος Καθηγητής

Περίληψη

Στόχος της παρούσας μεταπτυχιακής διατριβής είναι η ανάπτυξη ενός συστήματος πρώτων βοηθειών, που περιλαμβάνει μία εφαρμογή Android για τον τελικό χρήστη και ένα Web Service για την επικοινωνία με την βάση δεδομένων. Η εφαρμογή διαθέτει πληροφορίες για παροχή πρώτων βοηθειών ανάλογα με την κατάσταση, καθώς και δυνατότητα κοινοποίησης της τρέχουσας τοποθεσίας του χρήστη στην βάση δεδομένων προκειμένου να φτάσει εξειδικευμένη βοήθεια. Η εφαρμογή δεν επικοινωνεί απευθείας με την βάση, αλλά το Web Service εξασφαλίζει αυτή την επικοινωνία ως ενδιάμεσος.

Για την ανάπτυξη του παραπάνω λογισμικού, έπρεπε να μελετηθούν τεχνολογίες full-stack, που να δίνουν δηλαδή την δυνατότητα υλοποίησης όλων των λειτουργιών από το front-end, την γραφική διεπιφάνεια του χρήστη, ως το back-end, την επικοινωνία με την βάση και την μεταφορά δεδομένων. Το κομμάτι της επαφής με τον χρήστη, όπως αναφέρθηκε, επιλέχθηκε να γίνει μέσω μιας εφαρμογής Android. Το λογισμικό του Web Service χρησιμοποιεί την αρχιτεκτονική REST για την επικοινωνία με την βάση δεδομένων και τέλος για την αποθήκευση των δεδομένων επιλέχθηκε η πλατφόρμα Google Firebase, η οποία μεταξύ των πολλών υπηρεσιών που προσφέρει, παρέχει και μη σχεσιακή βάση δεδομένων. Τα δεδομένα αποθηκεύονται σε μορφή JSON και είναι διαθέσιμα σε πραγματικό χρόνο.

Στην συγγραφή του κειμένου της παρούσας βιβλιογραφικής μελέτης συναντήθηκαν ορισμένες γλωσσικές δυσκολίες. Η βιβλιογραφία στα ελληνικά που αφορά τις τεχνολογίες που χρησιμοποιήθηκαν είναι περιορισμένη, και για αυτόν τον λόγο η μετάφραση των όρων είναι υποκειμενική. Για την αποφυγή συγχύσεων αλλά και την διευκόλυνση των αναγνωστών πολλοί όροι συμπεριλαμβάνονται σε παρενθέσεις με την αγγλική τους ονομασία.

Abstract

The purpose of the present thesis is the development of a first aid system, that includes an Android application for the end user and a Web Service for the communication with the database. The application contains information on providing first aid regarding the emergency, as well as the capability of persisting the user's location on the database in order for special assistance to arrive. The application does not communicate directly with the database, but the Web Service ensures this communication as an intermediate.

For the development of the aforementioned software, full-stack technologies needed to be studied, meaning, ones that provide the capabilities of developing all functions, starting from the front-end, the user's graphical interface, all the way to the back-end, the communication with the database and the data transportation. For the user part, as mentioned, an Android application was decided to be developed. The Web Service software uses the REST architecture for the communication with the database and lastly for the storage of the data the Google Firebase platform was chosen, which among others services, provides a non relational database. The data are saved in JSON format and are available in real time.

On writing the content of this study certain language difficulties were met. The greek bibliography regarding the technologies that were used is limited, and for that reason the translation of many terms is subjective. In order to avoid confusion as well as to facilitate the readers, the english naming of many such terms are also included inside parenthesis.

Πίνακας Περιεχομένων

1	Εισαγωγή – Σύντομη περιγραφή του προβλήματος	6
1.1	Οργάνωση κειμένου	6
2	Το λειτουργικό σύστημα Android	8
2.1	Χρήση – Στατιστικά	8
2.2	Προγραμματισμός σε Android	8
2.2.1	Εργαλεία & Γλώσσες Προγραμματισμού	9
2.2.2	Βασική δομή εφαρμογής σε Android	9
3	Η Αρχιτεκτονική REST	11
3.1	Λειτουργίες CRUD	12
3.2	RESTful Web Services	12
3.3	Το framework Spring Boot	13
4	Google Firebase	14
4.1	Υπηρεσίες της πλατφόρμας	14
4.2	Βάσεις Δεδομένων στην Firebase	15
4.2.1	Realtime Database	15
4.2.2	Cloud Firestore	16
5	Ανασκόπηση πεδίου Πρώτων Βοηθειών	17
5.1	Αντίστοιχες εφαρμογές σε Ελλάδα & εξωτερικό για Android	17
5.1.1	Η εφαρμογή : First Aid – IFRC	19
6	Η εφαρμογή Πρώτων Βοηθειών	21
6.1	Χρήση της εφαρμογής – Γραφική Διεπιφάνεια	21
6.1.1	Απλός Χρήστης	21
6.1.2	Διαχειριστής	27
6.1.3	Αποθήκευση νέας κατάστασης – συμπτώματος	27
6.1.4	Επεξεργασία κατάστασης - συμπτώματος	29
6.1.5	Διαγραφή κατάστασης – συμπτώματος	31
6.1.6	Αποτυχία επικοινωνίας με διακομιστή – μηνύματα λάθους	31
6.2	Υλοποίηση	32
6.2.1	Λειτουργίες – Business Logic	32
6.2.2	Αρχεία Πόρων Γραφικής Διεπιφάνειας (UI Resource files)	34
6.2.3	Το αρχείο AndroidManifest.xml	35
6.2.4	Build με το Gradle	36
7	First Aid RESTful Web Service	37
7.1	Υλοποίηση	37
7.1.1	Δημιουργία Βάσης στην Google Firebase	37
7.1.2	Δημιουργία πρότζεκτ Spring Boot, Java	39
7.1.3	Λειτουργίες CRUD	41
7.2	Επικοινωνία με Firebase	44
8	Επίλογος	45

8.1	Συμπεράσματα	45
8.2	Μελλοντικές επεκτάσεις	45
9	Πίνακας Εικόνων.....	47
10	Ευρετήριο Πινάκων.....	48
11	Βιβλιογραφία	49

1 Εισαγωγή – Σύνομη περιγραφή του προβλήματος

Η ραγδαία ανάπτυξη του διαδικτύου τα τελευταία χρόνια έχει στρέψει μεγάλη μερίδα του ενδιαφέροντος της προγραμματιστικής κοινότητας στις web τεχνολογίες. Η επικοινωνία είναι απαραίτητη σε όλο το φάσμα των ανθρώπινων δραστηριοτήτων και το διαδίκτυο έχει διευρύνει τις δυνατότητες επικοινωνίας σε μεγέθη πρωτόγνωρα για την ιστορία της ανθρωπότητας. Οι περισσότερες υπηρεσίες προσφέρονται πλέον διαδικτυακά και το κλειδί για της επιτυχημένης διαδικτυακής παρουσίας είναι η συνεχής βελτίωση ταυτόχρονα με τις προσφερόμενες τεχνολογίες.

Ένα επιπλέον βήμα της τεχνολογικής επανάστασης ήταν η εξέλιξη των έξυπνων κινητών συσκευών. Την τελευταία δεκαετία τα smartphones και τα τάμπλετ κατέχουν πολύ μεγάλο μερίδιο της αγοράς τεχνολογίας αλλά και της καθημερινότητας. Αυτό δεν μπορεί να σημαίνει παρά μια συνεχόμενη απαίτηση για βελτίωση του λογισμικού των κινητών συσκευών, των οποίων οι ανάγκες είναι διαφορετικές από αυτών των καθιερωμένων σταθερών υπολογιστών. Οι μικρότερες οθόνες σε συνδυασμό με την μικρότερη υπολογιστική δύναμη θέτουν περιορισμούς στην ανάπτυξη του λογισμικού, τόσο των λειτουργικών τους συστημάτων όσο και των εφαρμογών. Παρ' όλαυτά, οι αγορές εφαρμογών για κινητές συσκευές διαθέτουν apps για οτιδήποτε μπορεί κανείς να φανταστεί. Δωρεάν είτε πληρώνοντας κάποιο αντιτίμο, οι χρήστες μπορούν να κατεβάσουν παιχνίδια, κοινωνικά μέσα, εργαλεία εργασίας, εφαρμογές με μαθησιακό περιεχόμενο (tutorials), υπηρεσίες συγκοινωνιών, εφαρμογές κρατήσεων καταλυμάτων, εστιατορίων και πολλά άλλα.

Οι πρώτες βοήθειες δεν θα μπορούσαν να λείπουν από μία τέτοια αγορά, καθώς το ενδιαφέρον για την ανθρώπινη ζωή είναι διαχρονικά υψίστης σημασίας. Έτσι, σημαντικοί φορείς όπως ο Ερυθρός Σταυρός, έχουν προχωρήσει στην ανάπτυξη εφαρμογών με περιεχόμενο πρώτων βοηθειών, αλλά ταυτόχρονα υπάρχει και πληθώρα ιδιωτικών πρωτοβουλιών που συνεισφέρουν στο αντικείμενο. Οι περισσότερες εφαρμογές είναι φιλικές προς τον χρήστη, λίγες όμως από αυτές προσφέρουν την πληροφορία σε μορφή ευανάγνωστη για μία έκτακτη ανάγκη. Επιπλέον πολλές από αυτές περιέχουν πάρα πολλές διαφημίσεις, που δυσχεραίνουν ακόμα περισσότερο την πιθανή χρήση σε επείγουσα κατάσταση.

Στην παρούσα μεταπτυχιακή διατριβή, επιλέχθηκε να αναπτυχθεί ένα ολοκληρωμένο σύστημα με πληροφορίες πρώτων βοηθειών, με την διάθεσή τους μέσω μιας εφαρμογής Android. Η διαφοροποίηση αυτής της εφαρμογής σε σχέση με τις υπόλοιπες είναι η υπηρεσία κοινοποίησης της τοποθεσίας του χρήστη. Μέσω μιας απλής γραφικής διεπιφάνειας, ο χρήστης μπορεί να καλέσει βοήθεια, πατώντας ένα κουμπί και δηλώνοντας το σύμπτωμά του. Η τοποθεσία του καταγράφεται στην βάση δεδομένων και είναι διαθέσιμη σε οποιαδήποτε υπηρεσία θα μπορούσε να αναλάβει την εξυπηρέτηση της επείγουσας κλήσης.

Δόθηκε ιδιαίτερο βάρος στην μελέτη και την σωστή εφαρμογή των τεχνολογιών που χρησιμοποιήθηκαν, ενώ επιλέχθηκαν τεχνολογίες αιχμής, προκειμένου να συμβαδίζουν με τις τρέχουσες εξελίξεις και τα σύγχρονα εργαλεία. Το Web Service που δημιουργήθηκε για την επικοινωνία της βάσης με την εφαρμογή, μπορεί να χρησιμοποιηθεί όχι μόνο από την παρούσα εφαρμογή αλλά και από οποιαδήποτε άλλη. Έπειτα από διάθεσή του στο διαδίκτυο, οποιαδήποτε εφαρμογή πελάτη μπορεί να το χρησιμοποιήσει για να αποκτήσει πρόσβαση στις πληροφορίες της βάσης. Αυτό σημαίνει ότι μπορούν εύκολα να αναπτυχθούν άλλες εφαρμογές, π.χ. desktop ή iOS, που χρησιμοποιούν την ίδια βάση δεδομένων.

1.1 Οργάνωση κειμένου

Η μελέτη και η εργασία εκπονήθηκαν από το γενικό προς το ειδικό. Έπειτα από διεξοδική μελέτη των τεχνολογιών που επρόκειτο να χρησιμοποιηθούν αλλά και του γνωστικού πεδίου που θα αφορούσε η εφαρμογή, εκπονήθηκε η υλοποίηση, όπου η μελέτη μπήκε σε εφαρμογή. Με αυτόν τον τρόπο διαρθρώνεται και το κείμενο της βιβλιογραφικής ανασκόπησης.

Στα πρώτα τρία κεφάλαια παρουσιάζονται οι τεχνολογίες που μελετήθηκαν για την εκπόνηση της εργασίας και τα σημεία – κλειδιά τους. Ξεκινώντας από τον χρήστη προς τα πιο τεχνικά μέρη και τον προγραμματιστή, το κεφάλαιο δύο αφορά το λειτουργικό σύστημα Android, στατιστικά χρήσης του και προγραμματιστικά στοιχεία, το τρίτο την αρχιτεκτονική REST και το τέταρτο τις υπηρεσίες της Google Firebase, μία εκ τω οποίων χρησιμοποιήθηκε στην εργασία ως η βάση

δεδομένων. Στο πέμπτο κεφάλαιο γίνεται μία ανασκόπηση του θέματος του αντικειμένου των πρώτων βοηθειών και των αντίστοιχων εφαρμογών που υπάρχουν σε Ελλάδα και εξωτερικό. Στο έκτο κεφάλαιο παρουσιάζεται η εφαρμογή που αναπτύχθηκε, πρώτα η χρήση της από την μεριά του χρήστη και έπειτα το κομμάτι της υλοποίησης, ενώ στο έβδομο αναλύεται η δημιουργία του Web Service με REST και ο τρόπος επικοινωνίας του με την βάση και την εφαρμογή. Τέλος, το κεφάλαιο οκτώ περιλαμβάνει συμπεράσματα και μελλοντικές επεκτάσεις.

2 Το λειτουργικό σύστημα Android

Το λειτουργικό σύστημα Android αναπτύχθηκε αρχικά από την Google και αργότερα από την Open Handset Alliance και κυκλοφόρησε εμπορικά πρώτη φορά το 2008, ενώ η τελευταία του έκδοση (Android 10) βγήκε τον Σεπτέμβριο του 2019. Πρόκειται για linux-based λειτουργικό σύστημα, που σημαίνει ότι ο πυρήνας του είναι βασισμένος στο linux, και είναι ελεύθερο λογισμικό ανοικτού κώδικα. Είναι σχεδιασμένο κυρίως για κινητές συσκευές με οθόνες αφής, π.χ. smartphones, tablets, αλλά διανομές του χρησιμοποιούνται και σε συσκευές όπως smart τηλεοράσεις και smartwatches. Αξίζει να σημειωθεί ότι είναι το πιο ευρέως διαδεδομένο λογισμικό στον κόσμο, με τον αριθμό των συσκευών να ξεπερνά σε πωλήσεις όλες τις συσκευές Windows, iOS και Mac OS X μαζί [1] [2].

2.1 Χρήση – Στατιστικά

Το ενδιαφέρον της παγκόσμιας τεχνολογικής και επιχειρηματικής κοινότητας έχει στραφεί στο Android λόγω της δημοτικότητάς του και του μεριδίου της αγοράς που κατέχουν οι συσκευές που χρησιμοποιούν το συγκεκριμένο λειτουργικό σύστημα. Συγκεκριμένα τον Ιούνιο του 2020, πάνω από το 74% των κινητών συσκευών σε παγκόσμιο επίπεδο έτρεχαν σε λειτουργικό σύστημα Android, ακολουθεί το λειτουργικό σύστημα της Apple, iOS με 25% και το υπόλοιπο 1% μοιράζονται λιγότερο γνωστά λειτουργικά συστήματα [3]. Στην Ελλάδα οι χρήστες Android ξεπερνούν ακόμη περισσότερο τους χρήστες iOS, με τους πρώτους να φτάνουν το 83% και τους δεύτερους μόλις το 12%. Πρακτικά αυτό σημαίνει ότι το τεράστιο κοινό του Android, δημιουργεί έμμεσα μία απαίτηση για συνεχείς βελτιώσεις, την οποία υποστηρίζει μία μεγάλη κοινότητα προγραμματιστών και οικονομικών φορέων. Έτσι, μπορούμε να μιλάμε για ένα λειτουργικό σύστημα που ενισχύεται συνεχώς και από την χρόνια κυκλοφορία του έως σήμερα έχουμε τουλάχιστον μία νέα έκδοση του λογισμικού ανά έτος [4].

Χαρακτηριστικό είναι ακόμη το σύνολο των διαθέσιμων εφαρμογών στα διαφορετικά λειτουργικά συστήματα για κινητές συσκευές. Το Google Play, η διαδικτυακή αγορά εφαρμογών Android, βρίσκεται πρώτο στο πλήθος των διαθέσιμων εφαρμογών (Ιούνιος 2020) με τον αριθμό τους να φτάνει τα 2.560.000, ενώ η αμέσως επόμενη αγορά, το Apple App Store διαθέτει 1.847.000 εφαρμογές για τους χρήστες [5]. Τέλος, αξίζει να αναφέρουμε την ανοδική πορεία ζήτησης εφαρμογών για κινητές συσκευές, αφενός λόγω της αύξησης των συνολικών χρηστών smart συσκευών, αφετέρου λόγω του αυξανόμενου χρόνου που δαπανούν οι χρήστες στα κινητά τους τηλέφωνα. Ενδεικτικά, αναφέρουμε ότι το 2016 υπήρχαν 2.5 εκατομμύρια χρήστες smart phone παγκοσμίως, το 2020 ο αριθμός αυξήθηκε κατά 1 εκατομμύριο ενώ υπολογίζεται ότι το 2021 θα έχει φτάσει τα 3.8 εκατομμύρια [6].

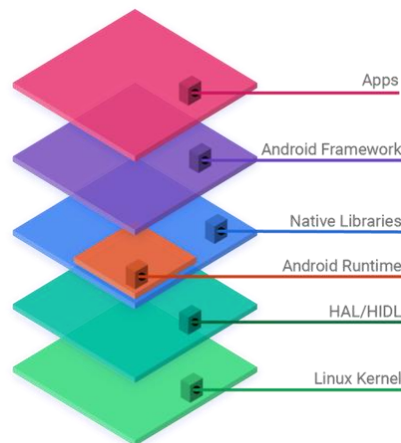
2.2 Προγραμματισμός σε Android

Με βάση τα παραπάνω είναι εύκολα κατανοητό το μεγάλο ενδιαφέρον προγραμματιστών, επιχειρήσεων πληροφορικής αλλά και φορέων για ανάπτυξη εφαρμογών που απευθύνεται στο κοινό των χρηστών Android. Ο προγραμματισμός σε Android, έχει κάποιες ιδιαιτερότητες που δεν αντιμετωπίζει κάποιος που αναπτύσσει λογισμικό για σταθερούς υπολογιστές και που πρέπει να ληφθούν υπόψη τόσο από τον τελικό προγραμματιστή αλλά και από αυτούς που αναπτύσσουν το λειτουργικό και τις διάφορες βιβλιοθήκες. Συνοπτικά πρέπει να δοθεί προσοχή στα παρακάτω:

- Οι οθόνες είναι μικρές και αφής. Αυτό σημαίνει ότι οι λειτουργίες που μπορούν να συμπεριληφθούν στην γραφική διεπιφάνεια του χρήστη είναι σχετικά περιορισμένες και πρέπει να υπάρχει πρόβλεψη για τα χαρακτηριστικά των στοιχείων που μπορούν να «πατηθούν» (clickable), μιας και το κλικ γίνεται με το δάχτυλο.
- Η μνήμη και η ταχύτητα των επεξεργασιών των κινητών συσκευών είναι σχετικά περιορισμένη. Σε σύγκριση με έναν σταθερό υπολογιστή, οι κινητές συσκευές έχουν προς το παρόν λιγότερη υπολογιστική δύναμη, άρα οι εφαρμογές τους θα πρέπει να είναι πιο «οικονομικές» ως προς την διαχείριση των πόρων του συστήματος.

Η αρχιτεκτονική της πλατφόρμας είναι μία στοίβα (stack) με τον πυρήνα Linux και όλους τους σχετικούς drivers (Camera, USB, Audio κλπ.) στο χαμηλότερο επίπεδο της. Ακολουθεί ένα επίπεδο αφαίρεσης υλικού (Hardware Abstraction Layer) το οποίο παρέχει στα υψηλότερα

επίπεδα της στοίβας διεπιφάνειες (Interfaces) που κάνουν διαθέσιμα τα στοιχεία του υλικού (hardware) στο επίπεδο του framework της Java. Έπειτα, βρίσκονται το περιβάλλον εκτέλεσης του Android (Android runtime) μαζί με κάποιες εγγενείς βιβλιοθήκες σε C και C++ και αμέσως μετά το Java API Framework το οποίο παρέχει στους προγραμματιστές όλες τις βιβλιοθήκες για την ανάπτυξη των εφαρμογών και την επικοινωνία με τα χαμηλότερα στρώματα. Τέλος, στο υψηλότερο επίπεδο βρίσκονται οι εφαρμογές που είναι διαθέσιμες στον τελικό χρήστη, εφαρμογές συστήματος, αυτές δηλαδή που έρχονται προεγκατεστημένες και αφορούν λειτουργίες του συστήματος, όπως το τηλέφωνο, η αποστολή μηνυμάτων, η κάμερα κ.α., αλλά και εφαρμογές που έχουν φτιαχτεί από τρίτους, όπως παιχνίδια, εφαρμογές με πληροφορίες, συστήματα online κρατήσεων κ.α. [7] [8]



Εικόνα 1: Αρχιτεκτονική Android (Android stack)

2.2.1 Εργαλεία & Γλώσσες Προγραμματισμού

Το Android SDK (Software Development Kit) είναι το σημείο εκκίνησης προγραμματισμού μιας εφαρμογής Android. Το kit αναπτύχθηκε από την Google και έγινε για πρώτη φορά διαθέσιμο τον Οκτώβριο του 2009, είναι γραμμένο στην γλώσσα υψηλού επιπέδου Java και μπορεί να χρησιμοποιηθεί από διάφορα λειτουργικά συστήματα (Cross-Platform). Περιλαμβάνει αποσφαλματωτή (debugger), βιβλιοθήκες, ένα προσομοιωτή του android os, τεκμηρίωση (documentation), εκπαιδευτικό υλικό (tutorials) και παραδείγματα κώδικα.

Έως το 2014, το επίσημο περιβάλλον ανάπτυξης (IDE) ήταν το Eclipse με την χρήση του πρόσθετου (plugin) Android Development Tools (ADT), όμως και άλλα περιβάλλοντα ανάπτυξης υποστήριζαν τον προγραμματισμό σε Android, όπως το IntelliJ IDEA και το NetBeans IDE. Από το 2015 ωστόσο, η Google σε συνεργασία με το IntelliJ διέθεσε το Android Studio ως το επίσημο περιβάλλον ανάπτυξης και σταμάτησε να υποστηρίζει το ADT.

Η ανάπτυξη λογισμικού στο Android SDK μπορεί να γίνει χρησιμοποιώντας τις γλώσσες Java, Kotlin και C++, όμως είναι δυνατή και η χρήση άλλων γλωσσών, εκτός JVM (Java Virtual Machine) όπως JavaScript, Go, C με τη χρήση εργαλείων για την σύνδεση με το JVM. Ακόμα, για την ανάπτυξη ολόκληρης της στοίβας της εφαρμογής (full-stack) χρησιμοποιούνται στο front-end κομμάτι, γλώσσες σήμανσης όπως η XML (eXtensible Markup Language).

2.2.2 Βασική δομή εφαρμογής σε Android

Ένα πρότζεκτ Android περιλαμβάνει ένα δομημένο σύνολο φακέλων, υποφακέλων και αρχείων, που ορίζουν όλη την εφαρμογή από τον πηγαίο κώδικα και όλα τα στατικά στοιχεία (assets) όπως εικόνες, ως τον κώδικα για τεστ και τις απαραίτητες ρυθμίσεις για τη δημιουργία (build) της εφαρμογής. Η δομή αυτή μπορεί εύκολα να δημιουργηθεί, μέσα από τα επίσημα περιβάλλοντα ανάπτυξης ξεκινώντας ένα καινούριο πρότζεκτ, χωρίς να χρειάζεται ο προγραμματιστής να την γνωρίζει ακριβώς. Η εξοικείωση ωστόσο τουλάχιστον με τα βασικά στοιχεία της είναι απαραίτητη για την ανάπτυξη του λογισμικού.

Μία δομική μονάδα (module) είναι το βασικό στοιχείο ανάπτυξης και επιτρέπει στον προγραμματιστή να διαιρέσει το πρότζεκτ σε διακριτές λειτουργικές μονάδες, η καθεμία με τα δικά της αρχεία πηγαίου κώδικα και ρυθμίσεων. Ένα πρότζεκτ μπορεί να έχει ένα ή περισσότερα modules, τα οποία είναι δυνατό να αλληλοεξαρτώνται εάν χρειάζεται, ενώ κάθε ένα από αυτά μπορεί να δημιουργηθεί, να τεσταριστεί και να αποσφαλματωθεί αυτόνομα. Ξεκινώντας ένα νέο πρότζεκτ στο Android Studio, το προκαθορισμένο module που δημιουργείται αυτόματα είναι αυτό της εφαρμογής, ονομάζεται Android app module και μπορεί να αφορά τηλέφωνα και τάμπλετ, Wear OS (λειτουργικό σύστημα για συσκευή που φοριέται όπως τα smart watch), Android TV ή να είναι Glass module. Πέρα από το app module, υπάρχει η δυνατότητα δημιουργίας module δυναμικών χαρακτηριστικών (Dynamic feature module), module βιβλιοθήκης (Library module) για επαναχρησιμοποίηση κώδικα, ή Google Cloud module για σύνδεση με λειτουργίες του Google Cloud [9].

Τα αρχεία που δημιουργούνται με την εκκίνηση ενός πρότζεκτ (project files) ακολουθούν μία συγκεκριμένη δομή σε φακέλους και υποφακέλους. Τα σημαντικότερα από αυτά που αφορούν μία εφαρμογή Android βρίσκονται στον φάκελο με το όνομα του module στο οποίο ανήκουν. Παρακάτω παρατίθενται οι βασικοί φάκελοι με το όνομα και τον τύπο αρχείων που περιέχουν.

module-name/

build/

περιέχει αρχεία με το αποτέλεσμα της διαδικασίας δημιουργίας (build)

libs/

περιέχει αρχεία ιδιωτικών βιβλιοθηκών

src/

περιέχει όλα τα αρχεία πηγαίου κώδικα αλλά και άλλους πόρους (resource files) της εφαρμογής, όπως εικόνες

main/

AndroidManifest.xml : σ' αυτό το αρχείο περιγράφεται η φύση της εφαρμογής και κάθε ένα από τα στοιχεία της. Εδώ υπάρχουν πληροφορίες για την εκκίνηση της εφαρμογής, τον τίτλο της κ.α.

java/

περιέχει τα αρχεία java

res/

περιέχει τους πόρους της εφαρμογής (resource files) που είναι συνήθως αρχεία xml, τα οποία καθορίζουν την γραφική διεπιφάνεια του χρήστη (User Interface)

test/

περιέχει κώδικα για τοπικά τεστ της εφαρμογής

build.gradle (module)

περιέχει πληροφορίες σχετικά με το build του συγκεκριμένου module

build.gradle (project)

περιέχει πληροφορίες σχετικά με το build ολόκληρου του πρότζεκτ (που μπορεί να συμπεριλαμβάνει περισσότερα από ένα modules)

Η παραπάνω δένδρική δομή είναι πολύ χρήσιμη για όποιον ξεκινάει να χτίζει μια εφαρμογή αλλά και στην συνέχεια για όποιον θα χρειαστεί να την τροποποιήσει, μιας και το πρότυπο παρέχει έναν εύκολο και αυτοματοποιημένο τρόπο ακόμα και για έναν νεό προγραμματιστή να φτιάξει την εφαρμογή του πάνω σε αυτό.

3 Η Αρχιτεκτονική REST

Η λέξη REST είναι ένα ακρωνύμιο για το Representational State Transfer και πρόκειται για ένα πρότυπο συνόλου κανόνων που σκοπό έχουν να ορίσουν την επικοινωνία κόμβων μέσω του ίντερνετ. Η αρχιτεκτονική REST δημιουργήθηκε το 2000, όταν ο Roy Fielding στην διπλωματική του διατριβή μαζί με συνεργάτες εντόπισαν ότι τα μέχρι τότε πρότυπα για την μεταφορά δεδομένων μέσω διαδικτύου ήταν ιδιαίτερα πολύπλοκα στην κατανόηση, στην χρήση αλλά και στην αποσφαλμάτωση. Έτσι οδηγήθηκαν στην ανάπτυξη του REST το οποίο επιβάλλει ορισμένους αρχιτεκτονικούς περιορισμούς, έτσι ώστε με τον τρόπο που οι εξυπηρετητές (servers) επεξεργάζονται και απαντούν σε αιτήματα πελατών (clients), το σύστημα τελικά να αποκτά ιδιότητες (όχι σχετικές με την λειτουργία) όπως η απόδοση (performance), η επεκτασιμότητα (scalability), η απλότητα (simplicity), η τροποποιεσιμότητα (modifiability), η φορητότητα (portability), η αξιοπιστία (reliability) κ.α. Εάν κάποιο σύστημα παραβιάζει οποιοδήποτε από τους παρακάτω περιορισμούς δεν μπορεί να θεωρηθεί RESTful :

1. Αρχιτεκτονική Πελάτη – Διακομιστή (Client – Server Architecture): θα πρέπει να υπάρχει σαφής διαχωρισμός των δύο μερών που πραγματοποιούν την επικοινωνία. Το ένα σύστημα θα πρέπει να λειτουργεί ως ο διακομιστής που καλείται και το άλλο ως ο πελάτης που αιτείται. Με αυτόν τον τρόπο βελτιώνεται η φορητότητα των εφαρμογών καθώς και η επεκτασιμότητα, αφού από τη μία οι πελάτες δεν χρειάζεται να γνωρίζουν συστατικά στοιχεία των εξυπηρετητών αλλά και οι εξυπηρετητές απλοποιούν τις δικές τους λειτουργίες. Επιπλέον, με αυτόν τον περιορισμό, τα συστήματα που δρουν ως πελάτες ή εξυπηρετητές έχουν την δυνατότητα να εξελίσσονται αυτόνομα χωρίς αλληλοεξάρτηση.
2. Έλλειψη κατάστασης (Statelessness): το συγκεκριμένο χαρακτηριστικό αναφέρεται στην απαίτηση να μην είναι απαραίτητη η αποθήκευση ενδιάμεσων πληροφοριών μεταξύ των αιτημάτων. Πιο συγκεκριμένα, κάθε αίτημα πελάτη φέρει όλες τις πληροφορίες που είναι απαραίτητες στον διακομιστή για να ετοιμάσει την απάντηση, χωρίς να χρειάζεται ο διακομιστής να έχει διατηρήσει πληροφορίες (state) από προηγούμενη επικοινωνία. Αυτό απαλλάσσει τον διακομιστή από την ανάγκη συγχρονισμού των δεδομένων με μια εξωτερική εφαρμογή.
3. Cacheability: απαιτείται οι απαντήσεις (responses) των συστημάτων – εξυπηρετητών να φέρουν επισήμανση για το αν είναι αποθηκεύσιμες (cacheable) ή μη-αποθηκεύσιμες (non-cacheable). Στην πρώτη περίπτωση ο πελάτης που λαμβάνει την απάντηση έχει την δυνατότητα να αποθηκεύσει τα δεδομένα για επαναχρησιμοποίησή τους σε πιθανό ίδιο αίτημα. Έτσι, αποφεύγονται αχρείαστες διαδράσεις πελάτη – διακομιστή βελτιώνοντας την απόδοση.
4. Ενοποιημένη διασύνδεση (Uniform Interface): αυτή η αρχή σχεδίασης είναι στον πυρήνα των υλοποιήσεων που θέλουν να είναι RESTful και υποδεικνύει πως για την επικοινωνία θα χρησιμοποιείται το πρωτόκολλο HTTP και θα υπάρχει ένα-προς-ένα αντιστοίχιση των μεθόδων του HTTP με τις λειτουργίες CRUD (Create, Read, Update, Delete). Έτσι, για να δημιουργήσουμε έναν πόρο (resource) στον διακομιστή χρησιμοποιούμε την μέθοδο POST του HTTP, για να ανασύρουμε έναν πόρο την μέθοδο GET, για να τροποποιήσουμε την PUT και για να διαγράψουμε την DELETE. Επιπλέον, το URI δεν περιλαμβάνει τα δεδομένα αλλά αυτά βρίσκονται στο σώμα (Request Body) της κάθε μεθόδου.

Η έννοια του πόρου (resource) είναι η βασική αφαίρεση για την πληροφορία στο REST. Οποιαδήποτε πληροφορία που μπορεί να ονομαστεί μπορεί να θεωρηθεί πόρος, όπως φωτογραφίες και αρχεία, μία προσωρινή υπηρεσία, ακόμα και μία συλλογή άλλων πόρων. Το REST χρησιμοποιεί προσδιοριστικά πόρων (resource identifiers) για να εντοπίσει τους εκάστοτε απαραίτητους πόρους για την πραγματοποίηση μιας επικοινωνίας. Η κατάσταση ενός πόρου σε μία οποιαδήποτε χρονική στιγμή είναι γνωστή ως αναπαράσταση πόρου (resource representation) και αποτελείται από δεδομένα, μεταδεδομένα που τα περιγράφουν και συνδέσμους υπερμέσων. Η μορφή των δεδομένων μιας αναπαράστασης είναι γνωστή ως τυπός μέσου (media type) και ορίζει τον τρόπο με τον οποίο τα δεδομένα πρέπει να επεξεργαστούν. Ο τύπος του μέσου είναι σύνηθες να είναι JSON ή XML. [10] [11] [12]

3.1 Λειτουργίες CRUD

Οι λειτουργίες **Create**, **Read**, **Update**, **Delete** είναι οι βασικές λειτουργίες που επιτελούνται επί μίας βάσης δεδομένων (persistent database) [Ως persistence στην επιστήμη των υπολογιστών αναφέρεται η ιδιότητα μίας κατάστασης (state) να διαρκεί περισσότερο από την διεργασία που την δημιούργησε. Οι βάσεις δεδομένων ως persistent, αποτελούν μονάδες αποθήκευσης που διατηρούν τα δεδομένα ασχέτως της διάρκειας ζωής τως διεργασιών που τα δημιούργησαν [13]]. Κάθε λειτουργία χωριστά μπορεί να αντιστοιχιστεί στις βασικές λειτουργίες που πραγματοποιούνται στις εφαρμογές βάσεων δεδομένων.

Στην αρχιτεκτονική REST, λόγω της αρχής της Ενοποιημένης Διασύνδεσης (Uniform Interface), οι λειτουργίες αυτές πρέπει να αντιστοιχίζονται μία προς μία σε συγκεκριμένα αιτήματα (requests) του πρωτοκόλλου HTTP, σε αντίθεση με τις SQL βάσεις όπου ορίζονται λειτουργίες επί αυτών ως συναρτήσεις. Παρακάτω φαίνονται οι λειτουργίες CRUD όπως υλοποιούνται στα πιο διαδεδομένα συστήματα :

Operation	SQL	HTTP	RESTfull WS	MongoDB
Create	INSERT	POST / PUT	POST	Insert
Read	SELECT	GET	GET	Find
Update	UPDATE	POST / PUT / PATCH	PUT	Update
Delete	DELETE	DELETE	DELETE	Remove

Πίνακας 1: Λειτουργίες CRUD σε διάφορα συστήματα

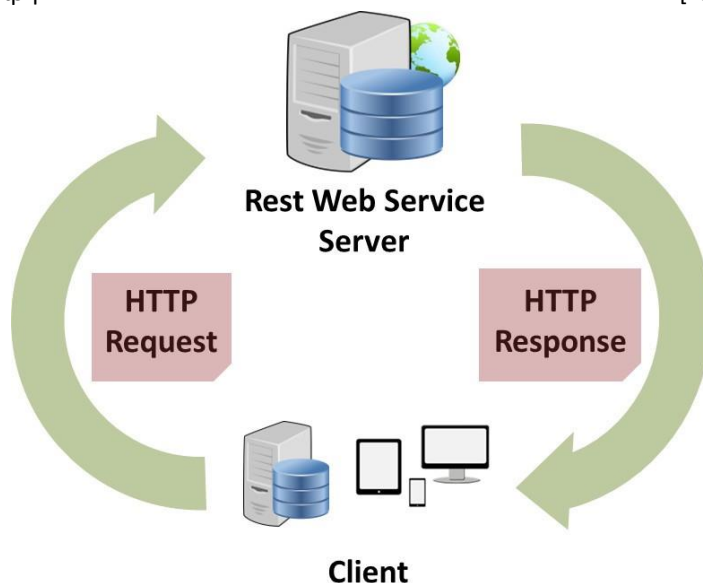
Η σήμανση των λέξεων συχνά υποδηλώνει την λειτουργία και είναι σχετικά εύκολο να κατανοήσουμε σε κάθε σύστημα ποια συνάρτηση αντιστοιχεί σε ποια λειτουργία, ενώ μερικές λέξεις παραμένουν ίδιες σε όλες τις πλατφόρμες. Για παράδειγμα, η δημιουργία μιας εγγραφής (Create) γίνεται με την εντολή Insert σε SQL και MongoDB πλατφόρμες, ενώ η διαγραφή έχει την ίδια εντολή σε SQL, HTTP και REST.

Το σημαντικό σε σχέση με την αρχιτεκτονική REST είναι η παρατήρηση πως, ενώ με το πρωτόκολλο HTTP μπορούμε να πραγματοποιήσουμε λειτουργίες δημιουργίας (Create) και τροποποίησης (Update) με παραπάνω από δύο εντολές, κάθε Web Service που υλοποιεί το REST πρέπει υποχρεωτικά να χρησιμοποιεί μία συγκεκριμένη. Δηλαδή, κάθε Create πρέπει να γίνεται με ένα POST request και κάθε Update με ένα PUT request.

3.2 RESTful Web Services

Υπάρχουν πολλοί ορισμοί του Web Service λόγω του μεγάλου αριθμού υπηρεσιών που παρέχονται στο διαδίκτυο αλλά και της ταχείας ανάπτυξης αυτού και των τεχνολογιών του. Σε γενικές γραμμές, ως Web Service νοούνται οι υπηρεσίες που είναι διαθέσιμες στο διαδίκτυο (ή ακόμα και σε ιδιωτικά δίκτυα), χρησιμοποιούνται για την μεταφορά δεδομένων και είναι ανεξάρτητες από λειτουργικά συστήματα ή προγραμματιστικές πλατφόρμες καθώς χρησιμοποιούν κοινές γλώσσες σήμανσης για τα δεδομένα, όπως οι XML και JSON. Πέρα από το REST, ένα άλλο πολύ διαδεδομένο πρωτόκολλο για Web Services είναι το SOAP και η τεχνολογία AJAX (Asynchronous JavaScript And XML).

Η αρχιτεκτονική REST έχει αρχίσει να επικρατεί στην υλοποίηση των Web Services λόγω των πλεονεκτημάτων που αναφέρθηκαν στο προηγούμενο κεφάλαιο. Ουσιαστικά, ένα RESTful Web Service είναι κι αυτό ένα Web Service που όμως βασίζεται στην αρχιτεκτονική REST και χρησιμοποιεί τις μεθόδους του πρωτοκόλλου HTTP για να υλοποιήσει τις λειτουργίες του. [13] [14] [15]



Εικόνα 2: RESTful Web Service

Αποτελεί και αυτό μία εφαρμογή (web application), γραμμένη σε οποιαδήποτε γλώσσα προγραμματισμού, η οποία διαθέτει τις μεθόδους της στο διαδίκτυο εφόσον τρέχει σε κάποιον διακομιστή. Η διαδικασία «ανεβάσματος» ενός Web Service σε κάποιον διακομιστή και η προετοιμασία του ώστε να εκτελείται λέγεται deploy, και είναι απαραίτητη για να είναι το service προσβάσιμο από οποιαδήποτε άλλη εφαρμογή. Συνεπώς, πέρα από τον πελάτη και το service, είναι αναγκαίο και το λογισμικό του διακομιστή.

Το σημείο στο οποίο η υπηρεσία είναι προσβάσιμη λέγεται API endpoint. Ουσιαστικά, ένα endpoint είναι το ένα άκρο της επικοινωνίας μεταξύ του service και της εφαρμογής - πελάτη. Όταν ένας πελάτης κάνει ένα «αίτημα» (request) σε ένα service, ή αντιστοίχως όταν το service στέλνει μία «απάντηση» (response) στο αίτημα ενός πελάτη, το σημείο σύνδεσής τους είναι το endpoint, και συνήθως αποτελείται από το URL του διακομιστή ή και του service. Το endpoint είναι επίσης η τοποθεσία στην οποία το service διαθέτει τους πόρους (resources) που είναι απαραίτητοι για την διεξαγωγή μίας λειτουργίας. Ένα RESTful Web Service, λέμε ότι διαθέτει REST endpoints, ακριβώς για υποδείξουμε ότι πρόκειται για μια υπηρεσία που χρησιμοποιεί την συγκεκριμένη αρχιτεκτονική.

3.3 Το framework Spring Boot

Ένα RESTful Web Service μπορεί να αναπτυχθεί σε διάφορες γλώσσες προγραμματισμού χρησιμοποιώντας ποικίλα frameworks. Ένα από αυτά είναι το Spring Boot το οποίο χρησιμοποιείται για την ανάπτυξη εφαρμογών στην γλώσσα Java. Δημιουργήθηκε από την εταιρεία Pivotal Software, η οποία το 2002 ανέπτυξε το Spring Framework. Σύμφωνα με το documentation, το Spring Boot είναι ένα ανοιχτού κώδικα framework βασισμένο στην Java, το οποίο μπορεί να χρησιμοποιηθεί για να δημιουργηθεί μία αυτόνομη (stand-alone) και έτοιμη για την παραγωγή (production ready) spring εφαρμογή. Παρέχει το πλαίσιο για την διευκόλυνση της ανάπτυξης του λογισμικού από την δημιουργία της εφαρμογής ως την εκτέλεσή της και απαιτεί ελάχιστες ρυθμίσεις για την εκκίνηση.

Η έκδοση 2.3.1.RELEASE έχει ως ελάχιστες απαιτήσεις συστήματος την έκδοση 8 της Java ενώ υποστηρίζει μέχρι και Java 14. Ακόμα, παρέχει υποστήριξη για τα εργαλεία build, Gradle και Maven, ενώ συμπεριλαμβάνει και τους Servlet Containers Tomcat, Jetty και Undertow.

Τα πλεονεκτήματα χρήσης του Spring Boot framework είναι η ευκολία στην κατανόηση και την ανάπτυξη εφαρμογών, η αύξηση της παραγωγικότητας και του χρόνου ανάπτυξης με την ελαχιστοποίηση των ρυθμίσεων που απαιτούνται.

4 Google Firebase

Η Firebase είναι μία πλατφόρμα για ανάπτυξη λογισμικού web και κινητών εφαρμογών. Δημιουργήθηκε από την εταιρεία Firebase το 2011, και το 2014 εξαγοράστηκε από την Google. Μέχρι σήμερα (Ιούλιος 2020) παρέχει 18 προϊόντα, τα οποία χρησιμοποιούνται από περισσότερες από 1,5 εκατομμύριο εφαρμογές. Μπορεί να βρίσκεται ενσωματωμένη σε μία εφαρμογή, χωρίς αυτό να επηρεάζει την εμπειρία του χρήστη, και είναι ιδιαίτερα χρήσιμη για τον εντοπισμό και την καταγραφή των κλικς και των μικροσφαλμάτων (glitches) [17].

4.1 Υπηρεσίες της πλατφόρμας

Η πλατφόρμα Firebase παρέχει μία πληθώρα υπηρεσιών, για την κάλυψη όλων των αναγκών μίας σύγχρονης εφαρμογής. Οι ανάγκες αυτές αφορούν το ίδιο το λογισμικό των εφαρμογών και την ποιότητά του, την ταχύτητα, την ασφάλεια και τον εντοπισμό σφαλμάτων. Ακόμα, μέσω της Firebase μπορεί να παρακολουθηθεί και να αναπτυχθεί η δυναμική της εφαρμογής με υπηρεσίες ανάλυσης δεδομένων, ένας ταχέως αναπτυσσόμενος κλάδος της πληροφορικής. Συγκεκριμένα οι υπηρεσίες που προσφέρονται είναι:

1. Cloud Firestore : αποθήκευση και συγχρονισμός δεδομένων εφαρμογής
2. Firebase ML : προσθήκη δυνατοτήτων Machine Learning στις εφαρμογές
3. Cloud Functions : ανάπτυξη backend λειτουργιών χωρίς διακομιστές, π.χ. δημιουργία συναρτήσεων που πυροδοτούνται από άλλες λειτουργίες της πλατφόρμας Firebase
4. Authentication : ασφαλής και απλή αυθεντικοποίηση χρηστών σε διάφορες πλατφόρμες
5. Hosting : φιλοξενία ιστοσελίδων και εφαρμογών
6. Cloud Storage : αποθήκευση και διάθεση αρχείων, σχεδιασμένο για γρήγορη και άμεση διάθεση περιεχομένου που έχει δημιουργηθεί από χρήστες, όπως φωτογραφίες και βίντεο
7. Realtime Database : αποθήκευση και συγχρονισμός δεδομένων σε πραγματικό χρόνο
8. Crashlytics : αναφορές σφαλμάτων σε πραγματικό χρόνο για την άμεση αντιμετώπισή τους
9. Performance Monitoring : πληροφορίες για την απόδοση των εφαρμογών από την οπτική του χρήστη, όπως η ταχύτητα και το responsiveness
10. Test Lab : περιβάλλον για δοκιμές (testing) των εφαρμογών σε φυσικές και εικονικές συσκευές, πριν η εφαρμογή φτάσει στον τελικό χρήστη
11. App Distribution : διάθεση εκδόσεων εφαρμογής πριν την τελική τους μορφή σε έμπιστους δοκιμαστές (testers)
12. In-App messaging : στοχευμένα μηνύματα σε συχνούς χρήστες της εφαρμογής με προτροπή για δράση μέσα σε αυτήν
13. Google Analytics : παροχή δεδομένων χρήσης για πάνω από 500 ξεχωριστά γεγονότα (events)
14. Predictions : ομαδοποίηση των χρηστών βάσει της προβλεπόμενης συμπεριφοράς τους
15. A/B Testing : περιβάλλον πειραματισμού (experimenting) της εφαρμογής και πιθανών αλλαγών της πριν αυτές διατεθούν ευρέως
16. Cloud Messaging : αποστολή ειδοποιήσεων σε διάφορες πλατφόρμες και συσκευές εκτός της εφαρμογής
17. Remote Config : εύκολη και γρήγορη παραμετροποίηση της εφαρμογής «εν κινήσει» (on the fly), χωρίς να είναι απαραίτητο το «ανέβασμα» (deploy) νέα έκδοσης
18. Dynamic Links : δυναμικοί σύνδεσμοι, ή αλλιώς deep links, για μεταφορά του χρήστη σε οποιοδήποτε σημείο μέσα στην εφαρμογή ακόμα και με την πρώτη τους είσοδο

Όλες οι υπηρεσίες της πλατφόρμας είναι διαθέσιμες μέσω της κονσόλας της Firebase (Firebase console) διευκολύνοντας την χρήση τους. Ακόμα, παρέχεται ένα σημείο στο οποίο ο

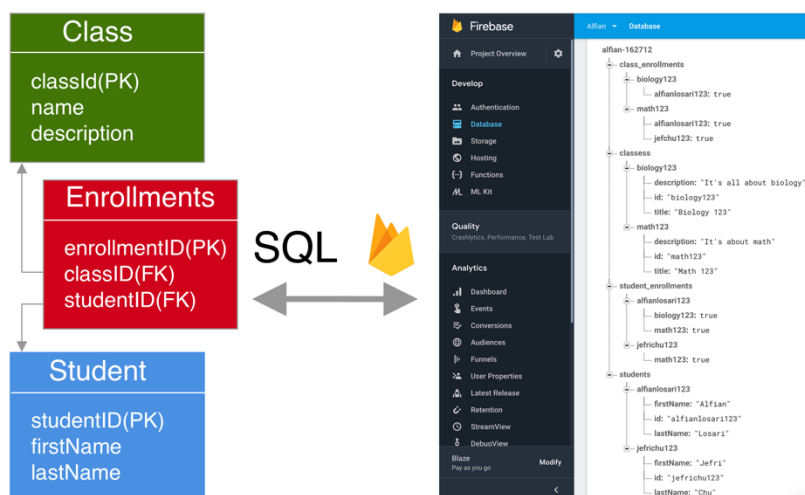
διαχειριστής ή ο developer μπορούν να αντρέξουν για να προσδώσουν στην εφαρμογή όλα τα απαραίτητα χαρακτηριστικά και τις δυνατότητες με τεχνολογίες αιχμής. [18]

4.2 Βάσεις Δεδομένων στην Firebase

Οι υπηρεσίες της βάσης δεδομένων αναλύονται περισσότερο, μιας και αποτελούν κεντρικό μέρος της παρούσας εργασίας. Η Firebase προσφέρει δύο τύπους βάσης δεδομένων : την Realtime Database και το Cloud Firestore. Η πρώτη υπήρξε η μοναδική βάση δεδομένων της Firebase μέχρι το 2019 όπου και εισήχθη και το Cloud Firestore.

4.2.1 Realtime Database

Όπως υποδηλώνει και το όνομα, η Realtime Database προσφέρει αποθήκευση και συγχρονισμό δεδομένων σε πραγματικό χρόνο. Φιλοξενείται (hosted) στο cloud και είναι μία NoSQL βάση δεδομένων. Αυτό σημαίνει ότι η αποθήκευση των δεδομένων δεν γίνεται σχεσιακά, με την συνηθισμένη οργάνωση πινάκων και σχέσεων μεταξύ τους, αλλά με άλλες μορφές, όπως ζευγάρι κλειδί – τιμή ή έγγραφο/αντικείμενο (document). Όλα τα δεδομένα αποθηκεύονται σε ένα και μόνο JSON Tree. Η προσπέλαση των δεδομένων είναι εύκολη από οποιαδήποτε συσκευή, όπου κάθε συσκευή μπορεί να χρησιμοποιεί διαφορετικό λειτουργικό σύστημα. Λόγω του συγχρονισμού σε πραγματικό χρόνο, κάθε εφαρμογή που είναι συνδεδεμένη με την βάση λαμβάνει οποιαδήποτε αλλαγή στα δεδομένα μέσα σε milliseconds. Στην παρακάτω εικόνα φαίνεται η δομή ενός JSON Tree (μέσα στην κονσόλα της Firebase) σε αντίθεση με μία σχεσιακή βάση δεδομένων (SQL).



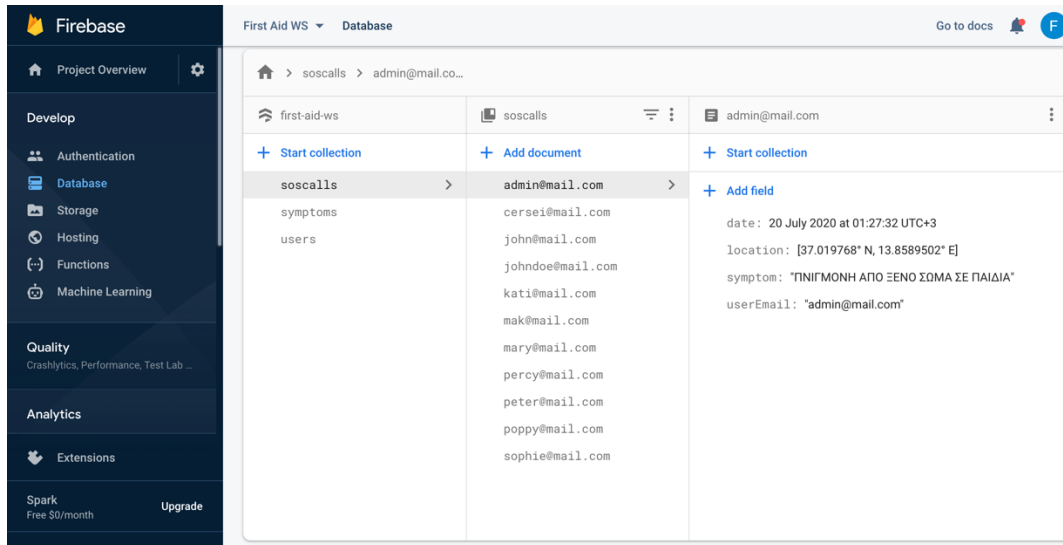
Εικόνα 3: Αποθήκευση Δεδομένων σε JSON Tree

Οι εφαρμογές που χρησιμοποιούν την Realtime Database μπορούν να λειτουργούν κανονικά ακόμα και αν αποσυνδεθούν από το διαδίκτυο. Αυτό συμβαίνει γιατί το SDK της Realtime Database αποθηκεύει τα δεδομένα στον δίσκο για την περίπτωση που χαθεί η σύνδεση. Μόλις επανασυνδεθεί η εφαρμογή, πραγματοποιείται ξανά συγχρονισμό έτσι ώστε να υπάρχει συνέπεια στα δεδομένα λόγω πιθανών τροποποιήσεων του κατά το χρονικό διάστημα της αποσύνδεσης.

Επιπλέον παρέχονται SDKs για την ανάπτυξη εφαρμογών χωρίς την διαμεσολάβηση διακομιστών. Τα δεδομένα της βάσης είναι προσπελάσιμα απευθείας από τις εφαρμογές, web ή κινητές, χωρίς να είναι απαραίτητος ένας application server. Η ασφάλεια και η επικύρωση των δεδομένων σε αυτή την περίπτωση επιτυγχάνεται μέσω κανόνων (Security Rules) εγγεγραμμένων στην βάση, που εκτελούνται κάθε φορά που υπάρχει εγγραφή ή ανάγνωση. [19]

4.2.2 Cloud Firestore

Το Cloud Firestore είναι η επόμενη γενιά της Realtime Database με πολλά κοινά χαρακτηριστικά αλλά σημαντικές βελτιώσεις. Πρόκειται και πάλι για μία NoSQL βάση δεδομένων που όμως αποθηκεύει τα δεδομένα σε συλλογές με έγγραφα/αντικείμενα (documents). Και εδώ η μορφή των δεδομένων είναι JSON όμως οι συλλογές είναι ξεχωριστές, δίνοντας την δυνατότητα η βάση να δομηθεί, και κατ'επέκταση να προσπελαύνεται, με μεγαλύτερη ευκολία. Ουσιαστικά, τα δεδομένα αναπτύσσονται σε ιεραρχίες και δομούνται ανεξάρτητα, ανάλογα με τις ανάγκες των εφαρμογών και των αιτημάτων που θα χρειαστούν. Τα αιτήματα (queries) προς την βάση μπορούν να είναι πιο εκφραστικά και περιεκτικά και να εξυπηρετούν αμεσότερα τον σκοπό χρήσης τους. Στην παρακάτω εικόνα φαίνεται η δομή των δεδομένων σε συλλογές και έγγραφα (documents) όπως παρουσιάζεται στην κονσόλα της Firebase :



Εικόνα 4: Αποθήκευση δεδομένων σε collections

Φυσικά, και στην περίπτωση του Google Firestore τα δεδομένα και οι αλλαγές τους είναι άμεσα διαθέσιμα σε πραγματικό χρόνο. Επιπλέον, το offline χαρακτηριστικό της Realtime Database παραμένει, δίνοντας και εδώ την δυνατότητα προσωρινής αποθήκευσης των δεδομένων σε κρυφή μνήμη και επαναπροσδιορισμού τους τη στιγμή της επανασύνδεσης.

Η ανάπτυξη εφαρμογών χωρίς διακομιστές (serverless apps), είναι και εδώ ένα από τα σημαντικά πλεονεκτήματα. Το Cloud Firestore συνεργάζεται άμεσα με την υπηρεσία Cloud Functions της Firebase, επιτρέποντας αλλαγές στο backend μέρος της εφαρμογής σε πραγματικό χρόνο χωρίς να είναι απαραίτητος ένας server στο ενδιάμεσο. Τέλος, η συγκεκριμένη βάση δεδομένων είναι σχεδιασμένη για να παρέχει εύκολη επεκτασιμότητα (scalability) στις εφαρμογές, προσφέροντας συνέπεια δεδομένων (data consistency) και αντιγραφή σε διάφορες περιοχές σε παγκόσμιο επίπεδο (multi-region replicas). [20]

5 Ανασκόπηση πεδίου Πρώτων Βοηθειών

Οι πρώτες βοήθειες αποτελούν ένα πεδίο με μεγάλο ενδιαφέρον για πολλούς ανθρώπους, λόγω της άμεσης σχέσης τους με την ανθρώπινη ζωή. Όλο και περισσότερος κόσμος παρακολουθεί σεμινάρια για παροχή πρώτων βοηθειών σε έκτακτες ανάγκες, ενώ τα τελευταία χρόνια με την ραγδαία ανάπτυξη του παγκόσμιου ιστού και των έξυπνων κινητών συσκευών οι πληροφορίες μπορούν να εντοπιστούν άμεσα και γρήγορα με το πάτημα λίγων κουμπιών. Στο διαδίκτυο υπάρχουν σε όλες τις γλώσσες χρήσιμες πληροφορίες από έγκυρους φορείς αλλά και από προσωπικά blogs και ιστοσελίδες. Ωστόσο σε μία έκτακτη ανάγκη, η αναζήτηση της ενέργειας που πρέπει να πραγματοποιηθεί για να σωθεί η ζωή ενός ανθρώπου μέσα σε ένα χείμαρρο πληροφοριών δεν είναι πρακτική.

Με την επικράτηση των έξυπνων κινητών συσκευών και κατ' επέκταση των εφαρμογών που δημιουργούνται για αποκλειστική και εξειδικευμένη χρήση από αυτές, ένα πολύ μεγάλο μέρος του γνωστικού εύρους της ανθρώπινης δραστηριότητας κωδικοποιήθηκε και εισάχθηκε σε apps έτοιμα και φιλικά προς τον χρήστη, για να τα κατεβάσει στην συσκευή του μέσα από τα αντίστοιχα stores. Οι εφαρμογές Πρώτων Βοηθειών δεν θα μπορούσαν να λείπουν από τις αγορές εφαρμογών, αφού μέσα από μία εφαρμογή, σε αντίθεση με μία ιστοσελίδα (που συνήθως είναι σχεδιασμένη για σταθερό υπολογιστή) ή ένα μεγάλο δυσανάγνωστο αρχείο, οι πληροφορίες μπορούν να οργανωθούν άρτια και να εξυπηρετούν αμεσότερα των σκοπών της έκτακτης ανάγκης. Επιπλέον, σε ένα τόσο κρίσιμο ζήτημα που η εγκυρότητα της πληροφορίας παίζει κυριολεκτικά ζωτικό ρόλο, η αξιολόγηση της πηγής που την παρέχει είναι πολύ ευκολότερη, μιας και στα stores αναφέρεται υποχρεωτικά και σε εμφανές σημείο το όνομα του εκδότη του λογισμικού. Έτσι, ο χρήστης μπορεί να ταυτοποιήσει πιο άμεσα τον φορέα που παρέχει τις πληροφορίες.

5.1 Αντίστοιχες εφαρμογές σε Ελλάδα & εξωτερικό για Android

Η επίσημη αγορά εφαρμογών στο λειτουργικό σύστημα Android είναι το Google Play. Για την εύρεση εφαρμογών με περιεχόμενο Πρώτων Βοηθειών μπορούν να χρησιμοποιηθούν οι όροι "First Aid", "Red Cross", "Emergencies", και στα ελληνικά «Πρώτες Βοήθειες», «Ερυθρός Σταυρός» κ.α. Η αναζήτηση του όρου "First Aid" επιστρέφει πάνω από 200 αποτελέσματα καλύπτοντας ένα ευρύτατο γνωστικό πεδίο. Στις κορυφαίες θέσεις βρίσκονται λίγες συγκεκριμένες εφαρμογές, οι οποίες και έχουν τις υψηλότερες βαθμολογίες αλλά και τον μεγαλύτερο αριθμό downloads.

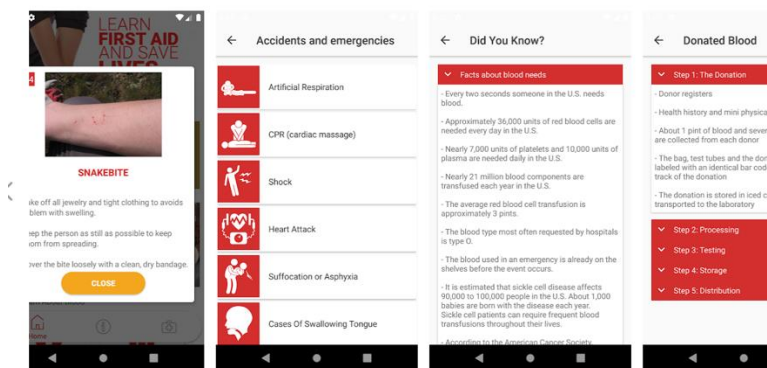
Τα γνωστικά πεδία που καλύπτουν οι εφαρμογές παρουσιάζουν ιδιαίτερο ενδιαφέρον. Στα πρώτα αποτελέσματα των αναζητήσεων βρίσκονται εφαρμογές που παρέχουν δομημένες τις πληροφορίες των Πρώτων Βοηθειών για γενικές και πιο συνηθισμένες καταστάσεις, οργανωμένες σε κατηγορίες ανάλογα με την επείγουσα κατάσταση. Αρκετές από αυτές περιλαμβάνουν και οπτικό υλικό, όπως φωτογραφίες, εικονίδια (icons) και βίντεο. Παρακάτω παρουσιάζονται ενδεικτικά μερικές από αυτές.

Η εφαρμογή "First Aid and Emergency Techniques" περιλαμβάνει περιγραφικά εικονίδια για διάφορες κατηγορίες επείγουσών αναγκών, όπως επείγοντα εσωτερικών και εξωτερικών χώρων, επείγοντα στον δρόμο, κ.α., τα οποία πατώντας επάνω τους εμφανίζουν γενικές πληροφορίες και μέτρα που πρέπει να ληφθούν σε κάθε μία. Είναι στα κορυφαία αποτελέσματα των αναζητήσεων τόσο στα αγγλικά όσο και στα ελληνικά και οι πληροφορίες είναι διαθέσιμες μόνο στα αγγλικά :



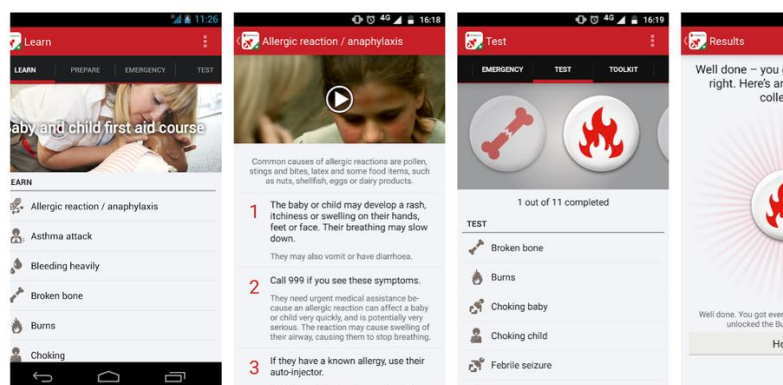
Εικόνα 5: Εφαρμογή "First Aid and Emergency Techniques"

Στην ίδια λογική κινείται και η εφαρμογή “First Aid” από την “ESTEPS”, στην οποία το υλικό οργανώνεται σε κατηγορίες με βάση την επείγουσα ανάγκη αυτή καθεαυτή, όπως πληροφορίες για έμφραγμα ή πνιγμό. Οι πληροφορίες, διαθέσιμες στα αγγλικά και στα αραβικά, είναι πολύ περιεκτικές και ίσως αρκετά πυκνογραμμένες για μία επείγουσα κατάσταση. Ένα μέρος της εφαρμογής αφιερώνεται στην αιμοδοσία, ενώ και εδώ χρησιμοποιούνται εικονίδια και πολύ λίγες φωτογραφίες :



Εικόνα 6: Εφαρμογή "First Aid" από "ESTEPS"

Στα κορυφαία αποτελέσματα και μία από τις πιο ενδιαφέρουσες εφαρμογές είναι το “Baby and Child First Aid”. Έχει αναπτυχθεί από τον Βρετανικό Ερυθρό Σταυρό και οι πληροφορίες εστιάζουν εξ’ολοκλήρου στην παροχή πρώτων βοηθειών σε μωρά και παιδιά. Παρουσιάζει ιδιαίτερο ενδιαφέρον λόγω της οργάνωσης των λειτουργιών οι οποίες χωρίζονται σε 5 κατηγορίες : εκμάθηση, προετοιμασία, επείγουσα κατάσταση, τεστ, εργαλειοθήκη. Οι πρώτες δύο, της εκμάθησης και της προετοιμασίας, περιέχουν αρκετό μαθησιακό υλικό, το οποίο ο χρήστης μπορεί να μελετήσει εκτενώς εκ των προτέρων για να είναι έτοιμος σε περίπτωση έκτακτης ανάγκης. Η τρίτη κατηγορία, αυτή της επείγουσας κατάστασης, περιέχει τις πληροφορίες πολύ πιο συμπυκνωμένες, ακριβώς όπως είναι αναγκαίο σε μία δύσκολη και στρεσογόνα επείγουσα κατάσταση. Τέλος, τα τεστ είναι συμπληρωματικά για την εκμάθηση και η εργαλειοθήκη είναι ένας χώρος αρχειοθέτησης προσωπικού ιστορικού υγείας του ενδιαφερόμενου. Επιπλέον, η εφαρμογή παρέχει επαρκές φωτογραφικό υλικό αλλά και βίντεο που διευκολύνει την κατανόηση των πληροφοριών :



Εικόνα 7: Εφαρμογή "Baby and Child First Aid"

Ο βρετανικός Ερυθρός Σταυρός έχει αναπτύξει ακόμα μία εφαρμογή για πρώτες βοήθειες, πολύ παρόμοια στον σχεδιασμό με την προηγούμενη με επιπλέον κάποιες λειτουργίες σχετικές με την τοποθεσία, όπως παρακολούθηση συγγενών ατόμων, ειδοποιήσεις σχετικά με τον καιρό κ.α., η οποία ωστόσο είναι διαθέσιμη μόνο στην επικράτεια του Ηνωμένου Βασιλείου.

Ο αμερικανικός Ερυθρός Σταυρός διαθέτει ένα ολόκληρο kit εφαρμογών για πρώτες βοήθειες προσαρμοσμένες σε ειδικότερες καταστάσεις. Δεδομένων των συχνών ακραίων καιρικών φαινομένων στην αμερικανική ήπειρο, ο Ερυθρός Σταυρός δημιούργησε εφαρμογές με

πληροφορίες για σεισμούς, τυφώνες και ανεμοστρόβιλους. Ακόμα, έχει ξεχωριστές εφαρμογές για την κολύμβηση, τις πλυμμήρες και τα κατοικίδια ζώα, κ.α.



Εικόνα 8: Εφαρμογές Αμερικανικού Ερυθρού Σταυρού

Από ελληνικές υλοποιήσεις, μία από τις πιο ολοκληρωμένες και από τις ελάχιστες που προσφέρουν πληροφορίες στα ελληνικά, είναι η εφαρμογή «Πρώτες Βοήθειες», η οποία πρόκειται για πρωτοβουλία ιδιώτη. Και εδώ, τα δεδομένα πρώτων βοηθειών είναι οργανωμένα σε κατηγορίες ανάλογα με την έκτακτη ανάγκη και υπάρχει ηχητικό και φωτογραφικό υλικό για διευκόλυνση της κατανόησης των οδηγιών. Μία χρήσιμη προσθήκη της εφαρμογής είναι οι πληροφορίες για τηλέφωνα άμεσης ανάγκης, όπως εφημερεύοντα φαρμακεία και νοσοκομεία, το Κέντρο Δηλητηριάσεων, το Εθνικό Κέντρο Άμεσης Βοήθειας κ.α. :



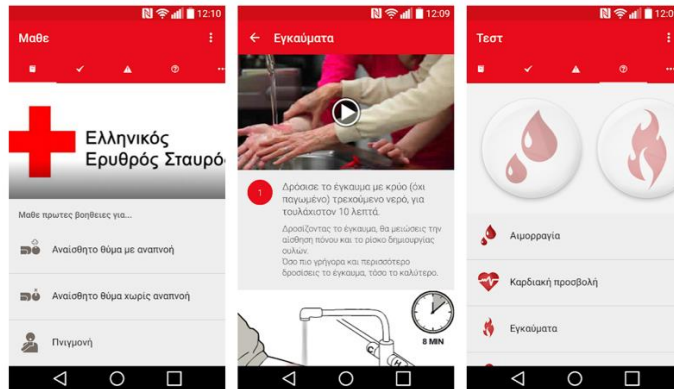
Εικόνα 9: Εφαρμογή "Πρώτες Βοήθειες"

Πέρα από εφαρμογές για τις πιο συνηθισμένες έκτακτες ανάγκες, όπως αυτές που παρουσιάστηκαν παραπάνω, αξίζει να αναφερθεί ότι υπάρχουν εφαρμογές που αφορούν αποκλειστικά παροχή πρώτων βοηθειών σε κατοικίδια ζώα, αρκετές που εστιάζουν αποκλειστικά στην εκμάθηση των τεχνικών και άλλες που αναφέρονται σε έκτακτες ανάγκες που μπορεί να προκύψουν σε συγκεκριμένες περιοχές του κόσμου ή από συγκεκριμένους παράγοντες (π.χ. φίδια). Παραδείγματα είναι το “Australian Bites and Stings” [21], “First Aid Africa” [22] και “Snakebite First Aid Africa” [23].

5.1.1 Η εφαρμογή : First Aid – IFRC

Η εφαρμογή “First Aid - IFRC” [24], έχει αναπτυχθεί από την «Διεθνή Ομοσπονδία Συλλόγων Ερυθρού Σταυρού και Ερυθράς Ημισελήνου» (International Federetaion for Red Cross) και αυτή τη στιγμή (Ιούλιος 2020) αποτελεί την πιο σταθερή, πλήρη και έγκυρη εφαρμογή πρώτων βοηθειών στο Google Play. Διατίθεται στις γλώσσες των χωρών της Ομοσπονδίας, ανάμεσα τους και τα ελληνικά και με το κατέβασμά της οι πληροφορίες της εφαρμογής ρυθμίζονται μέσω μιας επιλογής στην επιθυμητή γλώσσα. Οι πληροφορίες είναι γεωγραφικά προσδιορισμένες όχι μόνο όσον αφορά την γλώσσα αλλά και τα τηλεφωνα των υπηρεσιών έκτακτης ανάγκης που εξυπηρετούν την εκάστοτε περιοχή.

Τα δεδομένα οργανώνονται και εδώ και εκμάθηση, προετοιμασία, επείγοντα, τεστ και γενικές πληροφορίες. Το τμήμα της οργάνωσης και της προετοιμασίας παρέχουν εξαιρετικά χρήσιμες και οργανωμένες πληροφορίες και σε συνδυασμό με το τμήμα των τεστ, είναι ένας ολοκληρωμένος οδηγός για κάποιον που επιθυμεί να είναι σε ετοιμότητα σε καταστάσεις έκτακτης ανάγκης. Το πλεονέκτημα αυτής της εφαρμογής, όπως και αυτών του βρετανικού ερυθρού σταυρού (ο σχεδιασμός των εφαρμογών είναι σχεδόν πανομοιότυπος) είναι το ξεχωριστό μέρος της εφαρμογής που παρέχει τις πληροφορίες ειδικά στην περίπτωση έκτακτης ανάγκης. Έτσι, ο χρήστης μπορεί να ανατρέξει εύκολα και να βρει αυτό που χρειάζεται σε μία επείγουσα κατάσταση. Το υλικό περιλαμβάνει κείμενο, εικονίδια για την αναγνώριση της κατάστασης, φωτογραφίες απλές αλλά και κινούμενες, και βίντεο.



Εικόνα 10: Εφαρμογή "First Aid - IFRC"

6 Η εφαρμογή Πρώτων Βοηθειών

Έχοντας μελετήσει τα εργαλεία και τις μεθόδους προγραμματισμού για Android, ακολούθησε η ανάπτυξη της εφαρμογής. Πρόκειται για μια εφαρμογή εστιασμένη στην παροχή πληροφοριών για πρώτες βοήθειες αλλά και με δυνατότητα κοινοποίησης της τοποθεσίας του χρήστη. Για τον λόγο αυτό ο χρήστης κατά την πρώτη είσοδό του στην εφαρμογή καλείται να δημιουργήσει έναν λογαριασμό ή εάν έχει ήδη να πραγματοποιήσει πιστοποιημένη είσοδο, έτσι ώστε η εφαρμογή να γνωρίζει κάποια βασικά στοιχεία για αυτόν. Η απαίτηση εγγραφής του χρήστη στο σύστημα πριν την χρήση των υπηρεσιών εξασφαλίζει την γνωστοποίηση ορισμένων στοιχείων του στην περίπτωση επείγουσας κλήσης. Η ημερομηνία γέννησης κρίθηκε απαραίτητη, μιας και η ηλικία ενός ατόμου που πραγματοποιεί μία επείγουσα κλήση μπορεί να είναι χρήσιμη πληροφορία προς την αρμόδια υπηρεσία.

Όλα τα δεδομένα σχετικά με τις πρώτες βοήθειες που εισήχθησαν στην βάση δεδομένων και χρησιμοποιούνται από την εφαρμογή προέρχονται από έγκυρες πηγές, όπως το ΕΚΑΒ ή ο Ερυθρός Σταυρός. Προαπαιτούμενο για να μπορέσει να τρέξει η εφαρμογή, να γεμίσει με τα δεδομένα αλλά και να μπορέσει ο χρήστης να συνδεθεί είναι να τρέχει και το Web Service το οποίο παρέχει την επικοινωνία με την βάση δεδομένων.

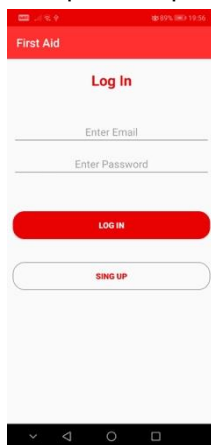
6.1 Χρήση της εφαρμογής – Γραφική Διεπιφάνεια

Η χρήση της εφαρμογής και οι λειτουργίες που παρέχονται είναι διαφορετικές για έναν απλό χρήστη και για έναν διαχειριστή. Οι δυνατότητες ενός απλού χρήστη περιορίζονται στην ανάγνωση των πληροφοριών των πρώτων βοηθειών, ενώ ο διαχειριστής μπορεί να τις τροποποιήσει, να προσθέσει νέες, ή να διαγράψει κάποιες από αυτές. Η υπηρεσία της επείγουσας κλήσης παραμένει ίδια και για τις δύο ομάδες χρηστών. Παρακάτω παρουσιάζονται αναλυτικά όλες οι λειτουργίες και τα σενάρια χρήσης για κάθε έναν από αυτούς.

6.1.1 Απλός Χρήστης

Η χρήση της εφαρμογής είναι αρκετά απλή και η γραφική διεπιφάνεια του χρήστη είναι σχεδιασμένη για να παρέχει την υπηρεσία κοινοποίησης της τοποθεσίας και τις πληροφορίες πρώτων βοηθειών εύκολα και κατανοητά.

Κατά την εκκίνηση της εφαρμογής για πρώτη φορά, ο χρήστης οδηγείται στην φόρμα εισόδου στην εφαρμογή (login) όπου καλείται να συμπληρώσει τα πεδία email και password και πατώντας το κουμπί “LOG IN” για να πιστοποιηθεί ότι πρόκειται για έγκυρο χρήστη:



Εικόνα 11: Οθόνη Login

Στην περίπτωση που ο χρήστης δεν έχει ακόμα έγκυρο λογαριασμό στην εφαρμογή, πρέπει πρώτα να συμπληρώσει την φόρμα εγγραφής. Πατώντας το κουμπί “SIGN UP” οδηγείται στην φόρμα εγγραφής, στην οποία καλείται να συμπληρώσει το όνομα, το email, έναν κωδικό και την ημερομηνία γέννησής του.

Εικόνα 12: Εγγραφή χρήστη

Η συμπλήρωση όλων των πεδίων της φόρμας είναι απαραίτητη για την εγγραφή. Σε περίπτωση μη συμπλήρωσης κάποιων από αυτά εμφανίζεται το μήνυμα λάθους *“Please fill the form, don’t leave any field blank”* κατά το πάτημα του κουμπιού *“REGISTER”*, όπως φαίνεται παρακάτω:

Εικόνα 13: Εγγραφή, μήνυμα λάθους

Το μήνυμα λάθους *“Please enter valid email”* εμφανίζεται σε περίπτωση που ο χρήστης εισάγει μη έγκυρη διεύθυνση email, δηλαδή διεύθυνση που δεν ταιριάζει στον κανόνα *<προσδιοριστικό>@<φορέας-εμαι>.<αναγνωριστικά-χώρας>*

Εικόνα 14: Εγγραφή, λάθος δομή email

Ένας τελευταίος έλεγχος στην είσοδο των δεδομένων κατά την εγγραφή νέου χρήστη, είναι η επικύρωση ότι η δοθείσα διεύθυνση email δεν υπάρχει ήδη στην βάση δεδομένων. Στην

περίπτωση που υπάρχει εμφανίζεται το μήνυμα λάθους “*Email already exists*” και ο χρήστης θα πρέπει να εισάγει μία διαφορετική διεύθυνση:



Εικόνα 15: Εγγραφή, το email υπάρχει ήδη

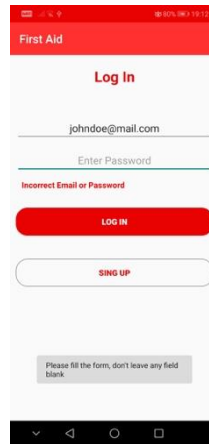
Τέλος, αφού η εισαγωγή των δεδομένων περάσει όλα τα στάδια ελέγχου που αναφέρθηκαν εμφανίζεται μήνυμα επιτυχούς εγγραφής “*You are successfully registered*” και τα στοιχεία διαγράφονται από τα αντίστοιχα πεδία :



Εικόνα 16: Εγγραφή επιτυχής

Το επόμενο βήμα είναι η είσοδος στην εφαρμογή με τα credentials που δημιουργήθηκαν. Πατώντας το κουμπί “*LOGIN*”, ο χρήστης οδηγείται εκ νέου στην φόρμα εισόδου, όπου και εδώ υπάρχουν στάδια ελέγχου για την εγκυρότητα των στοιχείων.

Στην φόρμα εισόδου πρέπει να συμπληρωθούν σωστά τα στοιχεία email και password στα αντίστοιχα πεδία. Η συμπλήρωση και των δύο πεδίων είναι απαραίτητη για την πιστοποιημένη είσοδο του χρήστη στο σύστημα. Σε περίπτωση μη συμπλήρωσης κάποιου από τα πεδία εμφανίζεται μήνυμα λάθους “*Please fill the form, don't leave any field blank*”:



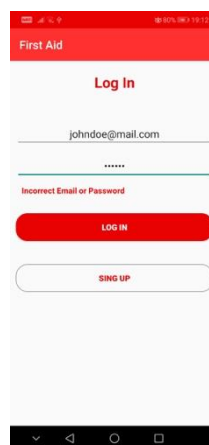
Εικόνα 17: Είσοδος, κενό πεδίο

Κι εδώ υπάρχει έλεγχος από την εφαρμογή για την συμπλήρωση διεύθυνσης email που δεν ακολουθεί το πρότυπο, οπότε και εμφανίζεται το μήνυμα *"Please enter valid email"*.



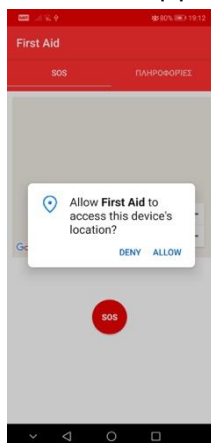
Εικόνα 18: Είσοδος, λάθος δομή email

Τέλος, εάν τα στοιχεία περάσουν από τον έλεγχο της φόρμας και αποσταλούν στον server για πιστοποίηση, εάν η διεύθυνση email δεν βρέθηκε ή εάν βρέθηκε αλλά ο κωδικός δεν είναι ο ίδιος με τον καταγεγραμμένο στην βάση για την συγκεκριμένη διεύθυνση email, εμφανίζεται το μήνυμα λάθους *"Incorrect email or password"* με κόκκινα γράμματα κάτω από το πεδίο για την εισαγωγή του κωδικού:



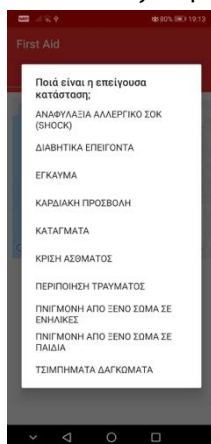
Εικόνα 19: Είσοδος, λανθασμένα δεδομένα

Εφόσον τα στοιχεία συμπληρωθούν σωστά, εμφανίζεται το μήνυμα “*You are successfully logged in*” και ο χρήστης έχει πλέον πρόσβαση στις υπηρεσίες και τα δεδομένα της εφαρμογής. Κατά την είσοδό του, εμφανίζεται ένα διαλογικό παράθυρο που ζητάει πρόσβαση στην τοποθεσία, προκειμένου να μπορεί να χρησιμοποιηθεί η υπηρεσία της επείγουσας κλήση. Εάν σε αυτό το σημείο ο χρήστης απορρίψει το αίτημα, επιλέξει δηλαδή “*Deny*” στο παράθυρο, δεν θα μπορεί να χρησιμοποιήσει την υπηρεσία επείγουσας κλήσης. Για να αλλάξει αυτή την επιλογή, μπορεί μετέπειτα να το κάνει μέσω των ρυθμίσεων της συσκευής του. Ακόμα, προαπαιτούμενο για την χρήση της συγκεκριμένης υπηρεσίας είναι να είναι ενεργοποιημένες οι υπηρεσίες τοποθεσίας:



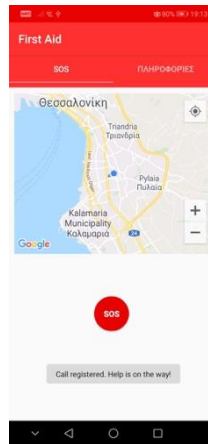
Εικόνα 20: Αίτημα πρόσβασης στην τοποθεσία

Η εφαρμογή έχει δύο καρτέλες (Tabs) με τα ονόματα “SOS” και “ΠΛΗΡΟΦΟΡΙΕΣ”. Στην καρτέλα “SOS”, που είναι και αυτή την οποία βλέπει πρώτη ο χρήστης, υπάρχει ένας χάρτης με την τρέχουσα τοποθεσία (εφόσον οι υπηρεσίες τοποθεσίας είναι ενεργοποιημένες και η εφαρμογή έχει πρόσβαση στην τοποθεσία της συσκευής) καθώς και ένα κουμπί με την ένδειξη “SOS”. Με το πάτημα του κουμπιού εμφανίζεται ένα διαλογικό παράθυρο με μία λίστα συμπτωμάτων – καταστάσεων στο οποίο ο χρήστης πρέπει να επιλέξει την επείγουσα κατάσταση:



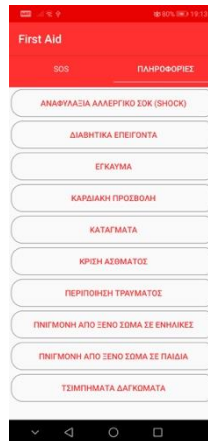
Εικόνα 21: Επιλογή επείγουσας κατάστασης

Έπειτα από την επιλογή από την λίστα, η κλήση μαζί με το σύμπτωμα, το email του χρήστη και την τοποθεσία του αποθηκεύονται στην βάση δεδομένων και εμφανίζεται στην εφαρμογή το μήνυμα “*Call registered. Help is on the way!*”:



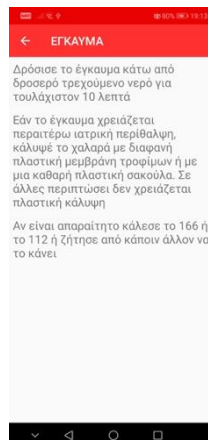
Εικόνα 22: Επιτυχής καταγραφή επείγουσας κλήσης

Η εναλλαγή μεταξύ των δύο καρτελών μπορεί να γίνει είτε πατώντας επάνω στο όνομα της είτε με κύληση προς τα αριστερά ή τα δεξιά. Έτσι, στην καρτέλα "ΠΛΗΡΟΦΟΡΙΕΣ" ο χρήστης μπορεί να μεταβεί είτε πατώντας επάνω στο όνομα της καρτέλας είτε με κύληση προς τα δεξιά. Η καρτέλα περιέχει μία λίστα με κουμπιά που αντιστοιχούν σε επείγουσες καταστάσεις:



Εικόνα 23: Καρτέλα "ΠΛΗΡΟΦΟΡΙΕΣ"

Πατώντας σε κάθε ένα από αυτά, ο χρήστης μεταβαίνει σε μία νέα οθόνη όπου μπορεί να διαβάσει πληροφορίες για το πώς να διαχειριστεί την επείγουσα κατάσταση. Για παράδειγμα, πατώντας το κουμπί "ΕΓΚΑΥΜΑ", εμφανίζεται η παρακάτω οθόνη με πληροφορίες για παροχή πρώτων βοηθειών σε περίπτωση εγκαύματος:



Εικόνα 24: Πληροφορίες Πρώτων Βοηθειών για έγκαυμα

Η οθόνη περιέχει τις πληροφορίες πρώτων βοηθειών και στο επάνω μέρος τον τίτλο της επείγουσας κατάστασης καθώς και ένα κουμπί – βέλος για μετάβαση στην προηγούμενη οθόνη. Με το πάτημά του εμφανίζεται και πάλι η λίστα κουμπιών με τις έκτακτες ανάγκες, απ' όπου ο χρήστης μπορεί να μεταβεί εκ νέου στις πληροφορίες πρώτων βοηθειών μίας άλλης κατάστασης ή στη καρτέλα “SOS”.

6.1.2 Διαχειριστής

Η είσοδος στην εφαρμογή με email και κωδικό διαχειριστή, δίνει πρόσβαση σε επιπλέον λειτουργίες επεξεργασίας των δεδομένων πρώτων βοηθειών. Συγκεκριμένα, ο διαχειριστής μπορεί να προσθέσει, να επεξεργαστεί και να διαγράψει πληροφορίες από την βάση. Η διαφοροποίηση στις λειτουργίες βρίσκεται μόνο στα δεδομένα των πρώτων βοηθειών ενώ η υπηρεσία επείγουσας κλήσης παραμένει ίδια με αυτή του απλού χρήστη.

6.1.3 Αποθήκευση νέας κατάστασης – συμπτώματος

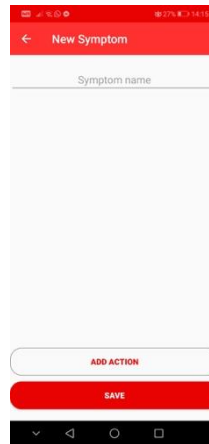
Κατά την είσοδο του διαχειριστή στην εφαρμογή, είναι ορατή η οθόνη με τις δύο καρτέλες “SOS” και “ΠΛΗΡΟΦΟΡΙΕΣ”. Η καρτέλα “SOS” παραμένει ίδια, ενώ μεταβαίνοντας στην καρτέλα “ΠΛΗΡΟΦΟΡΙΕΣ” εκτός από τα κουμπιά που αντιστοιχούν στην λίστα με τις επείγουσες καταστάσεις, υπάρχει στο πάνω μέρος της οθόνης ένα επιπλέον κουμπί με την επιγραφή “NEW SYMPTOM”:



Εικόνα 25: Καρτέλα "Πληροφορίες", προσθήκη νέου συμπτώματος

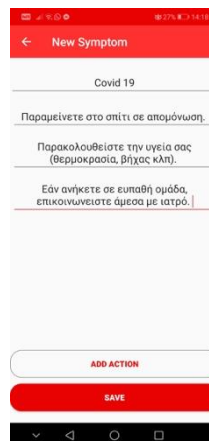
Πατώντας το, ο διαχειριστής οδηγείται σε μια νέα οθόνη με τα πεδία για την καταγραφή νέας κατάστασης. Η οθόνη από πάνω προς τα κάτω περιλαμβάνει:

1. Μπάρα με τον τίτλο της δραστηριότητας, “New Symptom”, και κουμπί – βέλος για μετάβαση στην προηγούμενη οθόνη. Πατώντας το κουμπί ο χρήστης οδηγείται στην καρτέλα των πληροφοριών
2. Πεδίο εισαγωγής κειμένου “Symptom Name”. Εδώ καταχωρείται το όνομα της νέας κατάστασης.
3. Λευκό κουμπί “Add action”
4. Κόκκινο κουμπί “Save”



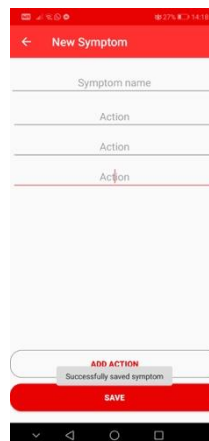
Εικόνα 26: Οθόνη καταγραφής νέας κατάστασης

Προκειμένου ο διαχειριστής να καταχωρίσει μία νέα κατάσταση – σύμπτωμα, πρέπει να συμπληρώσει το όνομά της στο πεδίο “Symptom name” και έπειτα να προσθέσει κατάλληλες ενέργειες. Πατώντας το κουμπί “ADD ACTION” εμφανίζεται ένα νέο πεδίο κειμένου για την καταγραφή μίας ενέργειας. Ο διαχειριστής μπορεί να προσθέσει όσες ενέργειες θέλει. Στο παρακάτω παράδειγμα, έχει δημιουργηθεί μία νέα κατάσταση – σύμπτωμα με όνομα “Covid 19” και έχουν προστεθεί τρεις ενέργειες :



Εικόνα 27: Νέα κατάσταση, συμπλήρωση πεδίων

Πατώντας το κουμπί “SAVE” οι πληροφορίες καταγράφονται στην βάση δεδομένων, το κείμενο διαγράφεται από τα πεδία και εμφανίζεται μήνυμα επιτυχούς καταγραφής “Successfully saved symptom”:



Εικόνα 28: Επιτυχής αποθήκευση κατάστασης

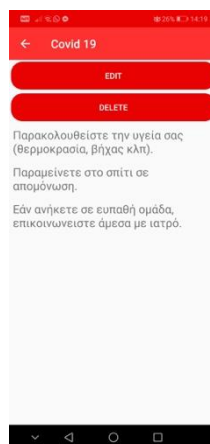
Τα νεοεισαχθέντα δεδομένα είναι πλέον διαθέσιμα από την βάση δεδομένων στην εφαρμογή. Πατώντας το κουμπί – βέλος για πλοήγηση πίσω στην καρτέλα των πληροφοριών ο διαχειριστής μπορεί να δει ότι έχει προστεθεί ένα νέο κουμπί με τον τίτλο του συμπτώματος, εδώ “Covid 19” το οποίο πατώντας το οδηγεί στην οθόνη με τις ενέργειες που εισήχθησαν από τον διαχειριστή:



Εικόνα 29: Νέα κατάσταση στην καρτέλα "Πληροφορίες"

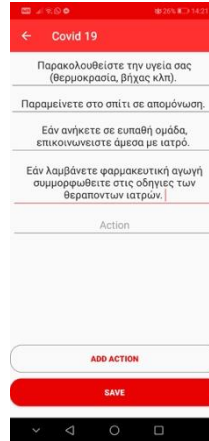
6.1.4 Επεξεργασία κατάστασης - συμπτώματος

Ο διαχειριστής έχει την δυνατότητα να επεξεργαστεί τις πληροφορίες των ενεργειών κάποιας από τις υπάρχουσες καταστάσεις. Πατώντας το κουμπί της κατάστασης που θέλει να επεξεργαστεί από την καρτέλα των πληροφοριών οδηγείται στην οθόνη με τις ενέργειες που αντιστοιχούν στην κατάσταση. Στο πάνω μέρος της οθόνης υπάρχουν τα κουμπιά “EDIT” και “DELETE”:



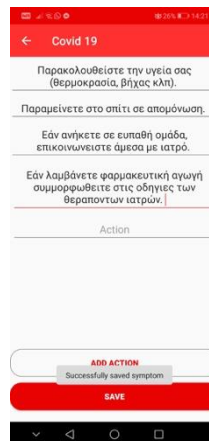
Εικόνα 30: Διαχειριστής, Οθόνη ενεργειών

Πατώντας το κουμπί “EDIT” οδηγείται σε μια νέα οθόνη, όπου η λίστα των ενεργειών βρίσκεται σε μορφή πεδίων κειμένου. Εκεί οι πληροφορίες μπορούν να επεξεργαστούν, αλλά και να προστεθούν νέες ενέργειες πατώντας το κουμπί “ADD ACTION”. Στο παράδειγμα έχει προστεθεί μία τέταρτη ενέργεια στην κατάσταση “Covid 19” αλλά και μία πέμπτη που είναι κενή. Οι ενέργειες που δεν έχουν κείμενο δεν αποθηκεύονται στην βάση, έτσι για την διαγραφή κάποιας ενέργειας, αρκεί στο πεδίο κειμένου να μην υπάρχει καταγεγραμμένο κείμενο:



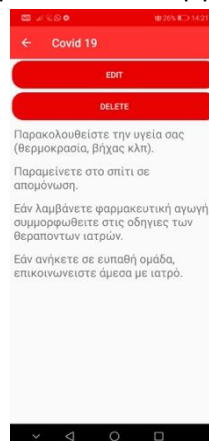
Εικόνα 31: Επεξεργασία κατάστασης – συμπτώματος

Πατώντας το κουμπί “SAVE” εμφανίζεται μήνυμα επιτυχούς αποθήκευσης “Successfully saved symptom”. Σε αυτήν την περίπτωση, δεν γίνεται αυτόματη διαγραφή των πληροφοριών, έτσι ώστε ο διαχειριστής να μπορεί να συνεχίσει με την επεξεργασία των πεδίων αν το επιθυμεί και να επαναλάβει την αποθήκευση :



Εικόνα 32: Επιτυχής αποθήκευση αλλαγών

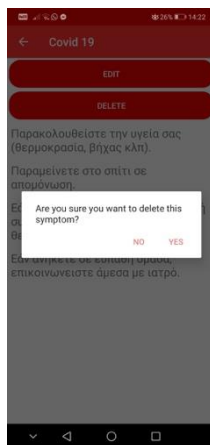
Πατώντας το κουμπί – βέλος για μετάβαση στην προηγούμενη οθόνη οι αποθηκευμένες πληροφορίες είναι πλέον διαθέσιμες στην λίστα των ενεργειών της συγκεκριμένης κατάστασης :



Εικόνα 33: Εμφάνιση αλλαγών στην λίστα ενεργειών

6.1.5 Διαγραφή κατάστασης – συμπτώματος

Στην οθόνη ενεργειών, που ανοίγει με το πάτημα ενός κουμπιού συμπτώματος από την καρτέλα “ΠΛΗΡΟΦΟΡΙΕΣ” υπάρχει επιπλέον το κουμπί “DELETE” κάτω από το “EDIT”, όπως φαίνεται και στις προηγούμενες εικόνες, που είναι διαθέσιμο όπως και το “EDIT” μόνο στον διαχειριστή. Πατώντας το, εμφανίζεται ένα διαλογικό παράθυρο που ζητά επιβεβαίωση για την διαγραφή του συμπτώματος:



Εικόνα 34: Επιβεβαίωση διαγραφής

Στο παράθυρο υπάρχουν οι επιλογές “No” και “Yes”. Με το πάτημα του “No” το παράθυρο κλείνει, ο χρήστης επιστρέφει στην οθόνη των ενεργειών και ακυρώνεται η διαγραφή. Με το πάτημα του “Yes” η κατάσταση – σύμπτωμα διαγράφεται από την βάση δεδομένων, εμφανίζεται το μήνυμα “Successfully deleted symptom” και ο χρήστης οδηγείται πίσω στην καρτέλα πληροφοριών, αφού το σύμπτωμα δεν είναι πλέον διαθέσιμο. Η καρτέλα δεν περιέχει πια το σύμπτωμα που έχει διαγραφεί. Στο παράδειγμα, ο χρήστης βρίσκεται στην οθόνη ενεργειών του “Covid 19” και επιλέγοντας “Yes” στο διαλογικό παράθυρο, το συγκεκριμένο σύμπτωμα παύει να βρίσκεται στην καρτέλα “ΠΛΗΡΟΦΟΡΙΕΣ” :



Εικόνα 35: Επιτυχής διαγραφή

6.1.6 Αποτυχία επικοινωνίας με διακομιστή – μηνύματα λάθους

Απαραίτητη προϋπόθεση για την ολοκληρωμένη λειτουργία της εφαρμογής είναι να είναι ενεργό το web service για την επικοινωνία με την βάση δεδομένων αλλά και να μπορεί η εφαρμογή να συνδεθεί με αυτό. Οι περιπτώσεις που η εφαρμογή επικοινωνεί με το web service και κατ’ επέκταση με τη βάση δεδομένων είναι οι παρακάτω :

1. Εγγραφή νέου χρήστη (Register)
2. Είσοδος υπάρχοντος χρήστη (Login)
3. Πάτημα του κουμπιού “SOS” για επείγουσα κλήση

4. Μετάβαση στην καρτέλα “ΠΛΗΡΟΦΟΡΙΕΣ”
5. Δημιουργία νέας κατάστασης – συμπτώματος
6. Επεξεργασία κατάστασης – συμπτώματος
7. Διαγραφή κατάστασης – συμπτώματος

Σε κάθε μία από τις παραπάνω λειτουργίες, εμφανίζονται μηνύματα λάθους ανάλογα με το πρόβλημα που υπάρχει εάν η σύνδεση ή η επικοινωνία με το Web Service δεν είναι επιτυχής:

1. “Requested resource not found”. Το συγκεκριμένο μήνυμα εμφανίζεται σε περίπτωση αποτυχίας σύνδεσης με τον διακομιστή, πιθανά λόγω εσφαλμένων ρυθμίσεων δικτύου. Αντιστοιχεί στο μήνυμα σφάλματος 404 του πρωτοκόλλου HTTP.
2. “Something went wrong at server end”. Αυτό το μήνυμα υποδεικνύει ότι η εφαρμογή έχει συνδεθεί με το web service αλλά το αίτημα δεν μπόρεσε να επεξεργαστεί ή κάτι άλλο πήγε στραβά στον διακομιστή. Αντιστοιχεί στο μήνυμα σφάλματος 500 του πρωτοκόλλου HTTP.
3. “Unexpected Error occurred! [Most common Error: Device might not be connected to Internet or remote server is not up and running]”. Όπως υποδεικνύει και το μήνυμα, αυτό το λάθος μπορεί να εμφανιστεί συχνά σε περίπτωση που η συσκευή δεν είναι συνδεδεμένη στο διαδίκτυο ή ο διακομιστής δεν είναι σε λειτουργία. Πρόκειται για γενίκευση και δεν αντιστοιχεί σε κάποιο συγκεκριμένο μήνυμα σφάλματος του πρωτοκόλλου HTTP, αλλά περιλαμβάνει ο,τιδήποτε πέραν των κωδικών 404 και 500.

6.2 Υλοποίηση

Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκε το επίσημο περιβάλλον ανάπτυξης για Android εφαρμογές, το Android Studio. Η κύρια γλώσσα προγραμματισμού είναι η Java, ενώ για το κομμάτι της γραφικής διεπιφάνειας του χρήστη χρησιμοποιήθηκε και η γλώσσα σήμανσης XML. Η εφαρμογή αναπτύχθηκε πάνω στο πρότυπο ενός πρότζεκτ που διαθέτει το Android Studio, χρησιμοποιώντας την έτοιμη δομή φακέλων και υποφακέλων για την δημιουργία των απαραίτητων αρχείων. Ακολουθήθηκαν πρότυπα ονοματοδοσίας για ευκολότερη κατανόηση του κώδικα, όπως οι κλάσεις που κληρονομούν την κλάση Activity έχουν στο τέλος του ονόματός τους την λέξη “Activity” και αντίστοιχα για τις κλάσεις που κληρονομούν την κλάση Fragment και χειρίζονται κάποιο fragment στην γραφική διεπιφάνεια, ενώ τα αρχεία πόρων (resource files) της γραφικής διεπιφάνειας ξεκινούν με το activity ή fragment, και ακολουθεί το όνομα της κλάσης που χειρίζεται την λειτουργία τους. Τέλος, για το build της εφαρμογής χρησιμοποιήθηκε το λογισμικό Gradle, όπως προτείνεται κατά την δημιουργία ενός νέου πρότζεκτ στο Android Studio.

6.2.1 Λειτουργίες – Business Logic

Οι λειτουργίες της εφαρμογής, ή αλλιώς η επιχειρηματική λογική (Business Logic) υλοποιούνται στις Java κλάσεις που βρίσκονται στο κυρίως πακέτο (package) της εφαρμογής. Εκεί υπάρχουν κλάσεις που αντιστοιχούν ένα-προς-ένα στα αρχεία layout της γραφικής διεπιφάνειας και χειρίζονται ό,τι συμβαίνει στην διεπαφή με τον χρήστη, αλλά και ένα πακέτο με βοηθητικές κλάσεις. Κάθε κλάση που αντιστοιχεί σε αρχείο διεπιφάνειας υλοποιεί την μέθοδο onCreate, η οποία καλείται κατά την δημιουργία της διεπιφάνειας και πραγματοποιεί βασικές λειτουργίες εκκίνησης, όπως συμπλήρωση και προετοιμασία δεδομένων και δομών.

Αναλυτικά, στο κυρίως package με όνομα com.example.firstaid υπάρχουν οι κλάσεις:

Class Name	Functionalities
LoginActivity	διαχείριση των γεγονότων της γραφικής διεπιφάνειας activity_login.xml
	Στην περίπτωση που ο χρήστης έχει πραγματοποιήσει ήδη είσοδο στην εφαρμογή, μέσω αυτής της κλάσης ανακατευθύνεται στην MainActivity.java και στη αντίστοιχη διεπιφάνεια.
	Σε αντίθετη περίπτωση στην κλάση διαβάζονται τα δεδομένα των πεδίων της φόρμας εισόδου, αποστέλλονται στο web service για

	<p>ταυτοποίηση και εμφανίζονται τα κατάλληλα μηνύματα επιτυχίας ή αποτυχίας εισόδου.</p> <p>Επιπλέον μέθοδοι για ανακατεύθυνση στην κεντρική οθόνη της εφαρμογής ή στην φόρμα εγγραφής.</p>
RegisterActivity	<p>διαχείριση των γεγονότων της γραφικής διεπιφάνειας activity_register.xml.</p> <p>Διάβασμα των δεδομένων των πεδίων της φόρμας εγγραφής και αποστολή στο web service για αποθήκευση.</p> <p>Εμφάνιση μηνυμάτων επιτυχίας ή αποτυχίας της εγγραφής.</p> <p>Επιπλέον μέθοδοι για ανακατεύθυνση στην κεντρική οθόνη της εφαρμογής ή στην φόρμα εισόδου.</p>
MainActivity	<p>διαχείριση των γεγονότων της γραφικής διεπιφάνειας activity_register.xml.</p> <p>Ρύθμιση των κυλιόμενων καρτελών της κεντρικής σελίδας της εφαρμογής και του τρόπου λειτουργίας τους.</p>
SosTab	<p>διαχείριση των γεγονότων της γραφικής διεπιφάνειας fragment_sos_tab.xml.</p> <p>Προετοιμασία και ρύθμιση του χάρτη και της τοποθεσίας του χρήστη.</p> <p>Διάβασμα από το web service όλων των καταστάσεων – συμπτωμάτων.</p> <p>Χειρισμός του κουμπιού επείγουσας κλήσης.</p> <p>Με το πάτημα του κουμπιού εμφανίζονται τα συμπτώματα και μετά την επιλογή ενός από αυτά, αποστέλλονται στο web service η τοποθεσία, το σύμπτωμα και τα στοιχεία του χρήστη για αποθήκευση της κλήσης στη βάση δεδομένων.</p>
InfoTab	<p>διαχείριση των γεγονότων της γραφικής διεπιφάνειας fragment_info_tab.xml.</p> <p>Προετοιμασία των δεδομένων καταστάσεων – συμπτωμάτων και πρώτων βοηθειών.</p> <p>Κατά την εκκίνηση, διαβάζονται τα δεδομένα από το web service και δημιουργούνται τα κατάλληλα κουμπιά με τίτλους τις καταστάσεις – συμπτώματα, που οδηγούν στα δεδομένα πρώτων βοηθειών.</p> <p>Εάν ο χρήστης είναι διαχειριστής, δημιουργείται επιπλέον το κουμπί για προσθήκη νέας κατάστασης – συμπτώματος, που οδηγεί στην συμπλήρωση της αντίστοιχης φόρμας.</p>
NewSymptomActivity	<p>διαχείριση των γεγονότων της γραφικής διεπιφάνειας activity_new_symptom.xml.</p> <p>Μόνο στην περίπτωση που ο χρήστης είναι διαχειριστής, η κλάση αναλαμβάνει το διάβασμα των πεδίων δεδομένων της φόρμας δημιουργίας ενός νέου συμπτώματος, την προσθήκη ενεργειών, και την αποστολή στο web service για αποθήκευση του νέου συμπτώματος στην βάση δεδομένων.</p>
ActionsActivity	<p>διαχείριση των γεγονότων της γραφικής διεπιφάνειας activity_actions.xml.</p> <p>Προετοιμασία του κειμένου για τις ενέργειες πρώτων βοηθειών και τοποθέτησή του στην γραφική διεπιφάνεια.</p> <p>Εάν ο χρήστης είναι διαχειριστής, δημιουργούνται επιπλέον τα κουμπιά για επεξεργασία και διαγραφή συμπτώματος.</p> <p>Στην περίπτωση που πατηθεί το κουμπί για διαγραφή, η κλάση εμφανίζει το διαλογικό παράθυρο και εφόσον επιβεβαιωθεί η</p>

	<p>διαγραφή, καλεί το web service για πραγματοποίηση της διαγραφής του συμπτώματος από την βάση.</p> <p>Στην περίπτωση που πατηθεί το κουμπί για επεξεργασία, υπάρχει σχετική μέθοδος για ανακατεύθυνση στην οθόνη της επεξεργασίας του συμπτώματος.</p>
EditSymptomActivity	<p>διαχείριση των γεγονότων της γραφικής διεπιφάνειας activity_edit_symptom.xml.</p> <p>Μόνο στην περίπτωση που ο χρήστης είναι διαχειριστής, η κλάση αναλαμβάνει την παρουσίαση των δεδομένων των ενεργειών σε επεξεργάσιμη μορφή, την προσθήκη ενεργειών, και την αποστολή στο web service των τροποποιημένων δεδομένων για την αποθήκευση στην βάση δεδομένων.</p>

Πίνακας 2: Κλάσεις εφαρμογής στο com.example.firstaid

Στο πακέτο helper μέσα στο com.example.firstaid υπάρχουν όπως αναφέρθηκε βοηθητικές κλάσεις. Αναλυτικά αυτές είναι:

Class Name	Functionalities
ApiParam	Περιέχει μεθόδους που επιστρέφουν σε String όλα τα URLs που χρησιμοποιούνται για την επικοινωνία με το web service.
StringConstants	Περιέχει String που χρησιμοποιούνται στην εφαρμογή μόνο για το κομμάτι της επικοινωνίας με το web service, όπως ονόματα πεδίων.
Utility	Περιέχει μεθόδους για τον έλεγχο πεδίων email και για έλεγχο κενών ή άδειων String.
MyPagerAdapter	Απαραίτητη κλάση για να λειτουργήσει η κύλιση των καρτελών στην κεντρική οθόνη της εφαρμογής.

6.2.2 Αρχεία Πόρων Γραφικής Διεπιφάνειας (UI Resource files)

Η γραφική διεπιφάνεια του χρήστη υλοποιείται σε αρχεία xml που περιλαμβάνουν τα στοιχεία και την διαμόρφωσή τους στην οθόνη. Ονομάζονται resource files και στην δομή του πρότζεκτ τοποθετούνται στον φάκελο res που βρίσκεται στον φάκελο src με τα αρχεία πηγαίου κώδικα. Στον υποφάκελο values του φακέλου res υπάρχουν ακόμα αρχεία που περιέχουν σταθερές τιμές για strings, χρώματα και styles της εφαρμογής.

Για τις ανάγκες της εφαρμογής Πρώτων Βοηθειών δημιουργήθηκαν τα παρακάτω αρχεία πόρων στον φάκελο /res/layout :

File Name	Περιεχόμενα
activity_login.xml	Περιέχει τα στοιχεία της οθόνης χρήστη για είσοδο στην εφαρμογή.
activity_register.xml	Περιέχει τα στοιχεία της οθόνης χρήστη για εγγραφή στην εφαρμογή.
main_activity.xml	Περιέχει τις δύο καρτέλες της κύριας οθόνης της εφαρμογής.
fragment_sos_tab.xml	<p>Η πρώτη από τις δύο καρτέλες της κύριας οθόνης.</p> <p>Περιέχει τον χάρτη με την τοποθεσία του χρήστη και το κουμπί επείγουσας κλήσης.</p>
fragment_info_tab.xml	<p>Η δεύτερη από τις δύο καρτέλες της κύριας οθόνης.</p> <p>Περιέχει κουμπιά με τους τίτλους των καταστάσεων – συμπτωμάτων, που δημιουργούνται δυναμικά.</p>

	Εάν ο χρήστης είναι διαχειριστής, περιέχει και το κουμπί δημιουργίας νέας κατάστασης – συμπτώματος. Δημιουργείται και αυτό δυναμικά.
activity_actions.xml	Περιέχει την λίστα με τις ενέργειες Πρώτων Βοηθειών σε μορφή απλού κειμένου. Η λίστα και τα πεδία κειμένου δημιουργούνται δυναμικά. Εάν ο χρήστης είναι διαχειριστής, περιέχει και τα κουμπιά επεξεργασίας και διαγραφής της κατάστασης – συμπτώματος και των ενεργειών. Δημιουργούνται και αυτά δυναμικά.
activity_new_symptom.xml	Περιέχει το πεδίο επεξεργάσιμου κειμένου για τον τίτλο του νέου συμπτώματος, ένα κουμπί για προσθήκη ενέργειας και ένα κουμπί για αποθήκευση. Περιέχει πεδία επεξεργάσιμου κειμένου για καταχώριση ενεργειών, που δημιουργούνται δυναμικά με το πάτημα του κουμπιού προσθήκης νέας ενέργειας.
activity_edit_symptom.xml	Περιέχει πεδία επεξεργάσιμου κειμένου για επεξεργασία των ενεργειών ενός συμπτώματος, ένα κουμπί για προσθήκη ενέργειας και ένα κουμπί για αποθήκευση των αλλαγών. Πεδία επεξεργάσιμου κειμένου για προσθήκη ενέργειας δημιουργούνται δυναμικά με το πάτημα του αντίστοιχου κουμπιού.

Πίνακας 3: Αρχεία πόρων της γραφικής διεπιφάνειας στο src/main/res/layout

Ακόμα, στον υποφάκελο drawable του φακέλου res δημιουργήθηκαν για την σχεδίαση (design) της εφαρμογής τα εξής αρχεία ορισμού styling :

File Name	
circular_button.xml	Ορίζει ένα στρόγγυλο σχήμα συγκεκριμένου χρώματος. Χρησιμοποιείται από το κουμπί επείγουσας κλήσης.
rounded_button.xml	Ορίζει ένα παραλληλόγραμμο σχήμα με στρογγυλεμένες γωνίες, συγκεκριμένου χρώματος. Χρησιμοποιείται από τα κουμπιά δημιουργίας νέου συμπτώματος, επεξεργασίας, συμπτώματος, διαγραφής συμπτώματος και εγγραφής και εισόδου στην εφαρμογή.
rounded_button_transparent.xml	Ορίζει ένα παραλληλόγραμμο σχήμα με στρογγυλεμένες γωνίες, διάφανου χρώματος με επιπλέον χρώμα περιγράμματος. Χρησιμοποιείται από τα κουμπιά που αντιστοιχούν σε συμπτώματα στην βάση δεδομένων και τα κουμπιά εγγραφής και εισόδου στην εφαρμογή.

Πίνακας 4: Αρχεία πόρων ορισμού σχημάτων στο src/main/res/drawable

6.2.3 Το αρχείο AndroidManifest.xml

Το αρχείο AndroidManifest.xml είναι απαραίτητο να υπάρχει σε κάθε πρότζεκτ Android μέσα στον υποφάκελο main στον φάκελο με τα αρχεία πηγαίου κώδικα με ακριβώς αυτό το όνομα. Περιέχει πληροφορίες απαραίτητες για τα εργαλεία που κάνουν το build της εφαρμογής (εδώ το Gradle), την εκκίνησή της, το λειτουργικό σύστημα Android καθώς και για το Google Play.

Μεταξύ άλλων, σε αυτό το αρχείο είναι υποχρεωτικό να υπάρχει το όνομα του πακέτου (package) της εφαρμογής αλλά και το αναγνωριστικό της. Αυτό χρησιμοποιείται για να μπορούν τα διάφορα μέρη να εντοπίζουν τις κλάσεις ή τα αρχεία πόρων. Για παράδειγμα στο αρχείο

main_activity.xml δηλώνεται ότι η κλάση που είναι υπεύθυνη για τον χειρισμό των λειτουργιών της είναι η MainActivity.java, χρησιμοποιώντας την παρακάτω δήλωση `<activity android:name=".MainActivity">` η οποία αναλύεται στο `<package-name>.MainActivity` δηλαδή εδώ στο `com.example.firstaid.MainActivity`, επιτρέποντας τον εντοπισμό της κλάσης.

Ακόμα, στο συγκεκριμένο αρχείο πρέπει υποχρεωτικά να συμπεριλαμβάνονται όλες οι κλάσεις που κληρονομούν την κλάση Activity και συνεπώς αντιστοιχούν σε μια οθόνη στην εφαρμογή και ποια από αυτές δημιουργείται κατά την εκκίνηση της εφαρμογής.

Εμπριέχονται επίσης πληροφορίες για τις άδειες που χρειάζεται η εφαρμογή. Στη συγκεκριμένη περίπτωση είναι απαραίτητη η χρήση του ίντερνετ και της τοποθεσίας του χρήστη, συνεπώς στο αρχείο υπάρχουν οι γραμμές

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

Τέλος, για την χρήση των χαρτών της Google είναι απαραίτητη η δήλωση στο AndroidManifest.xml του κλειδιού API σε ένα metadata tag ως εξής:

```
<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="key value" />
```

6.2.4 Build με το Gradle

Τα παραπάνω μέρη κώδικα, πόρων και επιπλέον πληροφοριών, μαζί με τις εξωτερικές βιβλιοθήκες και άλλες ρυθμίσεις, αναλαμβάνει να συναρμολογήσει σε μία ενιαία εφαρμογή το εργαλείο για build, Gradle. Κατά την δημιουργία του πρότζεκτ, δημιουργούνται αυτόματα δύο αρχεία build.gradle, ένα στο επίπεδο της εφαρμογής, δηλαδή μέσα στον φάκελο app και ένα στο επίπεδο του πρότζεκτ, δηλαδή στον root φάκελό του. Σε αυτά τα αρχεία περιέχονται απαραίτητες πληροφορίες για το build της εφαρμογής, την συναρμολόγησή του δηλαδή σε μία λειτουργική Android εφαρμογή έτοιμη για να «τρέξει». Μεταξύ άλλων εδώ υπάρχει το αναγνωριστικό της εφαρμογής (id), οι εκδόσεις του παρόντος λογισμικού, η έκδοση του SDK που χρησιμοποιεί και οι διάφορες εξαρτήσεις από εξωτερικές βιβλιοθήκες (dependencies).

7 First Aid RESTful Web Service

Όπως προαναφέρθηκε, για τις ανάγκες της εργασίας δημιουργήθηκε ένα Web Service, ως ενδιάμεσος μεταξύ της εφαρμογής Android και της βάσης δεδομένων στην Google Firebase. Το λογισμικό που αναπτύχθηκε είναι υπεύθυνο για την μεταφορά των δεδομένων από και προς την βάση δεδομένων και την εφαρμογή των Πρώτων Βοηθειών, ενώ η εφαρμογή σε καμία περίπτωση δεν έχει πρόσβαση απευθείας στην βάση και στα δεδομένα. Για την υλοποίηση χρησιμοποιήθηκε η αρχιτεκτονική REST και το Spring Boot Framework με γλώσσα προγραμματισμού Java. Αποτελεί μία λύση έτοιμη για διάθεση στο διαδίκτυο (deploy), η οποία μπορεί να χρησιμοποιηθεί από οποιαδήποτε εφαρμογή χρειάζεται πρόσβαση στην συγκεκριμένη βάση Πρώτων Βοηθειών.

7.1 Υλοποίηση

Τα στάδια της ανάπτυξης κινήθηκαν από το πιο «πίσω», που είναι η δημιουργία της βάσης, προς τα «μπροστά», την δημιουργία του Web Service.

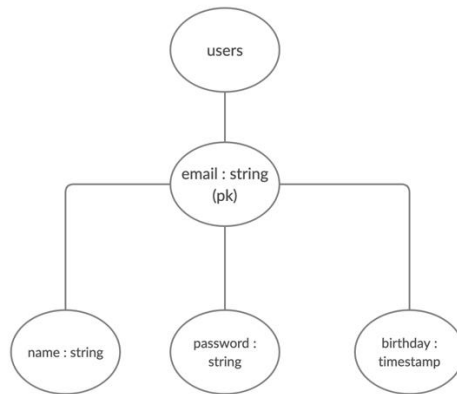
7.1.1 Δημιουργία Βάσης στην Google Firebase

Η δημιουργία νέα βάσης δεδομένων στην Google Firebase είναι μία σχετικά απλή διαδικασία και το μόνο που χρειάζεται είναι ένας λογαριασμός στην Google και ένας φυλλομετρητής (browser). Δημιουργήθηκε ένας καινούριος λογαριασμός Google, με διεύθυνση fristaidprj@gmail.com, που χρησιμοποιείται αποκλειστικά για τους σκοπούς της παρούσας εργασίας. Έπειτα, μέσα από την κονσόλα της πλατφόρμας <https://console.firebase.google.com/>, ακολουθώντας τα προτεινόμενα βήματα φτιάχτηκε ένα καινούριο πρότζεκτ στο οποίο φιλοξενείται και η βάση δεδομένων. Το πρότζεκτ ονομάστηκε “First Aid WS” και για την βάση επιλέχθηκε ο τύπος βάσης Cloud Firestore, έναντι της Realtime Database, μιας και αποτελεί την πιο σύγχρονη λύση της πλατφόρμας.

Όπως αναλύεται στο κεφάλαιο για την Google Firebase, πρόκειται για μία μη-σχεσιακή βάση δεδομένων (NoSQL). Δεν υπάρχουν πίνακες (tables) και εγγραφές (records) αλλά τα δεδομένα αποθηκεύονται σε JSON Trees και δομούνται ανάλογα με τις ανάγκες του πρότζεκτ. Έτσι, αναλύθηκαν οι λειτουργίες που θα έπρεπε να επιτελεί η εφαρμογή, οι τύποι δεδομένων που θα χρειαζόταν και οι σχέσεις μεταξύ τους και προέκυψαν τα παρακάτω:

- Για την χρήση της εφαρμογής είναι απαραίτητη η πιστοποιημένη είσοδος σε αυτήν (login) με έναν καταγεγραμμένο λογαριασμό στην βάση. Άρα χρειάζεται δεδομένα δομημένα ως έναν χρήστη. Ο χρήστης προσδιορίζεται με βάση την διεύθυνση email του, που υποχρεωτικά είναι μοναδική στον παγκόσμιο ιστό. Επιπλέον, θα πρέπει να αποθηκεύεται το όνομά του και ένας κωδικός. Λόγω της φύσης της εφαρμογής, που αφορά πρώτες βοήθειες, επιλέχθηκε να αποθηκεύεται επίσης και η ημερομηνία γέννησής του.
- Η λειτουργία της επείγουσας κλήσης μέσω του κουμπιού της εφαρμογής πρέπει να καταγράφει στην βάση α) τον χρήστη, β) την τοποθεσία του, γ) το σύμπτωμά του και δ) την ημερομηνία που πραγματοποιήθηκε η κλήση. Για να είναι ευκολότερα προσβάσιμα τα δεδομένα των κλήσεων από πιθανά μελλοντικές υπηρεσίες που θα ήθελαν να τα χρησιμοποιήσουν, επιλέχθηκε να αποθηκεύονται σε ξεχωριστή δομή.
- Οι πληροφορίες Πρώτων Βοηθειών αποτελούνται από μία κατάσταση – σύμπτωμα και μία λίστα ενεργειών που πρέπει να κάνει κάποιος σε περίπτωση που προκύψει η συγκεκριμένη έκτακτη ανάγκη. Τα συγκεκριμένα δεδομένα δεν σχετίζονται άμεσα με τους χρήστες, παρα μόνο για την καταγραφή της επείγουσας κλήσης, γι’ αυτό και επιλέχθηκε να αποθηκεύονται σε ξεχωριστή δομή.

Με βάση τα παραπάνω δημιουργήθηκαν στην βάση οι εξής συλλογές (collections), κατ’ αντιστοιχία των πινάκων μίας σχεσιακής βάσης δεδομένων, οι τύποι δεδομένων είναι αυτοί που προσφέρονται από την Firebase :

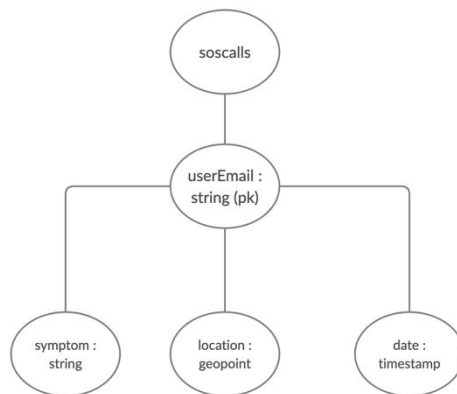
**Εικόνα 36: Collection "users"**

Συλλογή "users" : για την αποθήκευση των δεδομένων χρήστη

Κλειδί : email, τύπος δεδομένων string

Πεδία :

1. name, τύπος δεδομένων string
2. password, τύπος δεδομένων string
3. birthday, τύπος δεδομένων timestamp

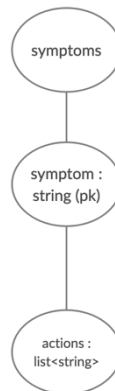
**Εικόνα 37: Collection "soscalls"**

Συλλογή "soscalls" : για την αποθήκευση των δεδομένων επείγουσας κλήσης

Κλειδί : userEmail, τύπος δεδομένων string

Πεδία :

1. symptom, τύπος δεδομένων string
2. location, τύπος δεδομένων geopoint
3. date, τύπος δεδομένων timestamp



Εικόνα 38: Collection "symptoms"

Συλλογή "symptoms" : για την αποθήκευση των δεδομένων πρώτων βοηθειών

Κλειδί : symptom, τύπος δεδομένων string

Πεδία :

1. actions, τύπος δεδομένων λίστα string

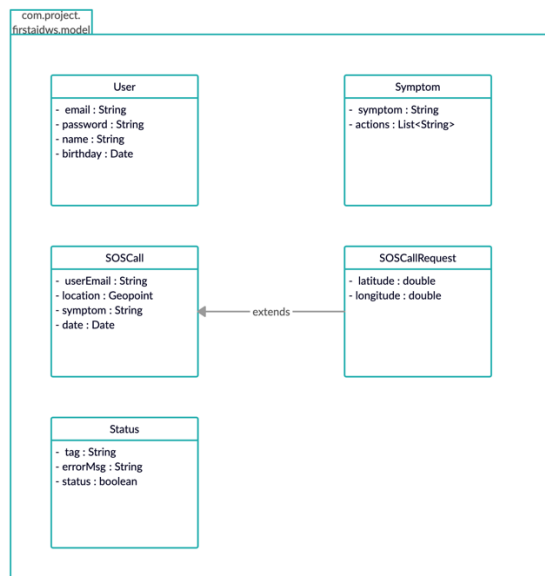
7.1.2 Δημιουργία πρότζεκτ Spring Boot, Java

Το περιβάλλον ανάπτυξης IntelliJIDEA, προσφέρει αυτοματοποιημένο τρόπο για την δημιουργία ενός Spring Boot Project με Java. Το εργαλείο build σε αυτήν την περίπτωση είναι το Maven, και οι εξωτερικές εξαρτήσεις (dependencies) καθώς και άλλες πληροφορίες που χρειάζονται για το build αποθηκεύονται στο αρχείο pom.xml.

Το σημαντικό πλεονέκτημα του αυτοματοποιημένου τρόπου δημιουργίας ενός Web Service με Spring Boot είναι ότι το configuration περιλαμβάνει ενσωματωμένο το λογισμικό του διακομιστή, σε αυτήν την περίπτωση έναν Tomcat server. Έτσι, κατά την εκκίνηση του πρότζεκτ, ξεκινάει ο διακομιστής και αυτόματα το web service τρέχει (deploy) σε αυτόν. Η προεπιλεγμένη θύρα (port) είναι η :8080 και το web service γίνεται απευθείας διαθέσιμο στην διεύθυνση <http://localhost:8080/>, υποθέτοντας ότι η εφαρμογή τρέχει τοπικά.

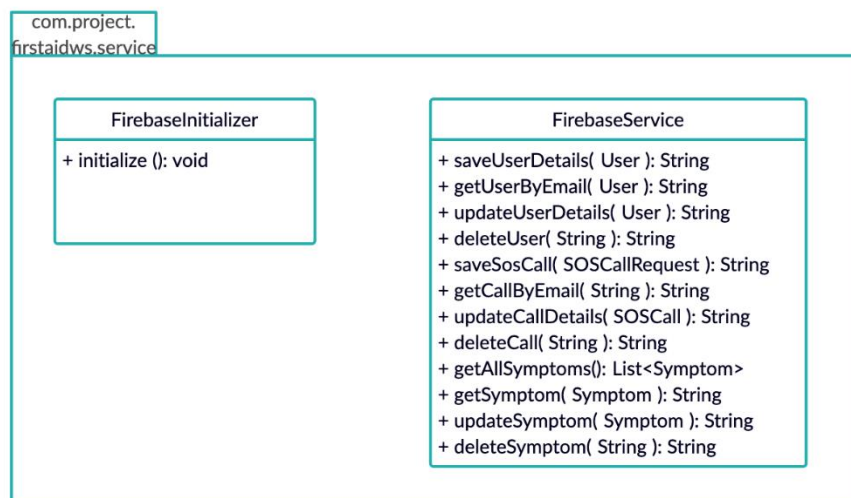
Ο κώδικας της εφαρμογής βρίσκεται στον υποφάκελο src/main/java στο package com.project.firstaidws και οι κλάσεις οργανώθηκαν σε άλλα packages για την καλύτερη δόμηση του πρότζεκτ.

Για το διάβασμα των δεδομένων από την βάση, στο package .model προστέθηκαν οι κλάσεις που αντιστοιχούν στις συλλογές της βάσης, ως το μοντέλο των δεδομένων, αλλά και δύο βοηθητικές. Η κλάση Status χρησιμοποιείται για να αποσταλλεί ως απάντηση (response) σε κάθε αίτημα (request) που γίνεται προς το service και η κλάση SOSCallRequest χρησιμοποιείται για την αντιστοίχιση του γεωγραφικού μήκους και πλάτους σε Geopoint, που είναι ο τύπος δεδομένων της τοποθεσίας στην βάση. Ακολουθεί το διάγραμμα UML με τις κλάσεις του πακέτου :



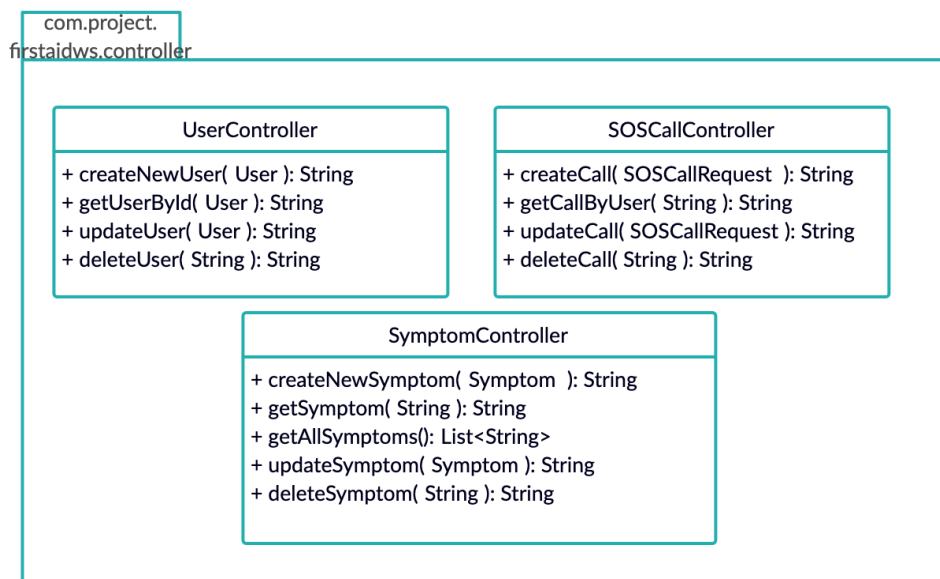
Εικόνα 39: UML com.project.firstaidws.model package

Την επικοινωνία με την βάση αναλαμβάνουν οι κλάσεις στο πακέτο `.service`. Η κλάση `FirebaseInitializer` περιλαμβάνει μία μόνο μέθοδο, που είναι υπεύθυνη για τις ρυθμίσεις παραμέτρων για την επικοινωνία με την βάση και την αρχικοποίηση, με την εκκίνηση του απαραίτητου στιγμιότυπου (`Instance`). Η κλάση `FirebaseService` περιλαμβάνει όλες τις μεθόδους που καλούνται από τα REST endpoints και πραγματοποιούν τις συναλλαγές με την βάση δεδομένων. Ακολουθεί το διάγραμμα UML με τις κλάσεις του πακέτου :



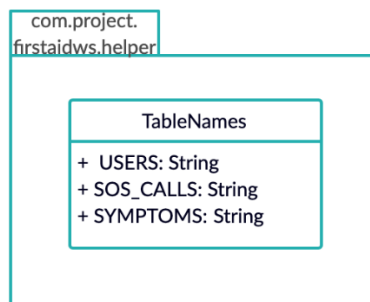
Εικόνα 40: UML com.project.firstaidws.service package

Οι κλάσεις που λειτουργούν ως REST endpoints βρίσκονται στο πακέτο `.controller`. Αναλογικά με τις συλλογές της βάσης, δημιουργήθηκαν τρεις κλάσεις `controllers` για τις λειτουργίες επί των χρηστών, των επείγουσων κλήσεων και των συμπτωμάτων. Οι μέθοδοι χρησιμοποιούν τις αντίστοιχες μεθόδους της κλάσης `FirebaseService`. Ακολουθεί το διάγραμμα UML με τις κλάσεις του πακέτου :



Εικόνα 41: UML com.project.firstaidws.controller package

Τέλος, στο πακέτο .helper υπάρχει μόνο μία βοηθητική κλάση που περιέχει τα ονόματα των συλλογών στην βάση σε static final Strings, για την διευκόλυνση της επανειλημμένης χρήσης τους από άλλα μέρη του κώδικα.



Εικόνα 42: UML com.project.firstaidws.helper package

7.1.3 Λειτουργίες CRUD

CRUD, όπως αναφέρθηκε και στο κεφάλαιο της αρχιτεκτονικής REST είναι το ακρωνύμιο για τις λέξεις Create, Read, Update, Delete και αναφέρονται στις τέσσερις βασικές λειτουργίες μίας βάσης δεδομένων (persistent database). Επιπλέον, είδαμε ότι ένα RESTful Web Service πρέπει υποχρεωτικά να υλοποιεί αυτές τις τέσσερις λειτουργίες μέσω συγκεκριμένων αιτημάτων του πρωτοκόλλου HTTP, με αντιστοίχιση ένα προς ένα λόγω της αρχής της Ενοποιημένης Διασύνδεσης (Uniform Interface).

Για τις ανάγκες της εφαρμογής, οι λειτουργίες αυτές έπρεπε να υλοποιηθούν και για τις τρεις συλλογές της βάσης δεδομένων, δηλαδή για τις συλλογές users, soscalls και symptoms. Η κάθε μία μπορεί να πραγματοποιηθεί χρησιμοποιώντας διάφορα κλειδιά, π.χ. read μιας εγγραφής χρήστη με βάση το όνομά του, όμως υλοποιήθηκαν μόνο αυτές που ήταν αναγκαίες για την λειτουργία της εφαρμογής Android.

Στον κώδικα σε Java, όπως φαίνεται στο προηγούμενο κεφάλαιο, οι λειτουργίες αναπτύχθηκαν σε δύο επίπεδα: α) στις κλάσεις Controller που είναι τα REST endpoints που επικοινωνούν με τον «εξωτερικό κόσμο», και κατ' επέκταση και με την εφαρμογή Πρώτων

Βοηθειών και β) στην κλάση `FirebaseService` που πραγματοποιεί την σύνδεση με την βάση Google Firebase και στην ουσία και τα αιτήματα προς αυτήν.

Σε ένα RESTful Web Service σε Java, τα REST endpoints ορίζονται ως οι μέθοδοι μίας κλάσης που αναλαμβάνουν το κάθε αίτημα (request). Η υλοποίησή τους γίνεται με συγκεκριμένα Annotations πάνω από την εκάστοτε μέθοδο που είναι μέρος της βιβλιοθήκης `org.springframework.web.bind.annotation` [17]. Το annotation ορίζει και τί είδους αντιστοίχιση θα γίνεται με τα αιτήματα του πρωτοκόλλου HTTP. Έτσι, σε κάθε μία από τις κλάσεις Controller της εφαρμογής, υπάρχουν μέθοδοι για τις λειτουργίες CRUD. Αναλυτικά, οι μέθοδοι και οι λειτουργίες τους φαίνονται παρακάτω, όπου στην στήλη URI είναι το URI (Uniform Resource Identifier) από το οποίο η κάθε μέθοδος είναι προσπελάσιμη :

UserController:

Όνομα μεθόδου	Annotation	URI	Λειτουργία
getUserById	@GetMapping	/getUser	Ως ID του χρήστη θεωρείται το email του. Η μέθοδος λαμβάνει ως παράμετρο έναν χρήστη με τον κωδικό και το email του και επιστρέφει ένα String με το αποτέλεσμα σε μορφή JSON, ανάλογα με το αν ο χρήστης υπάρχει με τα συγκεκριμένα πιστοποιητικά ή όχι στην συλλογή users.
createUser	@PostMapping	/createUser	Η μέθοδος λαμβάνει ως παράμετρο έναν χρήστη με email, κωδικό, όνομα και ημερομηνία γέννησης και τον αποθηκεύει στην βάση δεδομένων στην συλλογή users. Επιστρέφει ένα String σε μορφή JSON ανάλογα με το αποτέλεσμα της λειτουργίας (επιτυχής ή όχι).
updateUser	@PutMapping	/updateUser	Η μέθοδος λαμβάνει ως παράμετρο έναν χρήστη με το email του, και πιθανά κάποιο από τα άλλα πεδία του και τον αποθηκεύει στην βάση δεδομένων στην συλλογή users. Επιστρέφει ένα String σε μορφή JSON ανάλογα με το αποτέλεσμα της λειτουργίας (επιτυχής ή όχι).
deleteUser	@DeleteMapping	/deleteUser	Η μέθοδος λαμβάνει ως παράμετρο το email του χρήστη σε μορφή String και τον διαγράφει από την συλλογή users. Επιστρέφει ένα String σε μορφή JSON ανάλογα με το αποτέλεσμα της λειτουργίας (επιτυχής ή όχι).

Πίνακας 5: Μέθοδοι της κλάσης `UserController`

SOSCallController:

Όνομα μεθόδου	Annotation	URI	Λειτουργία
getCallByUser	@GetMapping	/getCallByUser	Η μέθοδος λαμβάνει ως παράμετρο το email ενός χρήστη και επιστρέφει ένα String σε μορφή JSON, που περιέχει τα δεδομένα της επείγουσας κλήσης (SOSCall) από την συλλογή soscalls.

createCall	@PostMapping	/createCall	Η μέθοδος λαμβάνει ως παράμετρο ένα αντικείμενο της κλάσης SOSCallRequest, και το αποθηκεύει στην βάση δεδομένων στην συλλογή soscalls. Επιστρέφει ένα String σε μορφή JSON ανάλογα με το αποτέλεσμα της λειτουργίας (επιτυχής ή όχι).
updateCall	@PutMapping	/updateCall	Η μέθοδος λαμβάνει ως παράμετρο ένα αντικείμενο της κλάσης SOSCallRequest, και το τροποποιεί στην βάση δεδομένων στην συλλογή soscalls. Επιστρέφει ένα String σε μορφή JSON ανάλογα με το αποτέλεσμα της λειτουργίας (επιτυχής ή όχι).
deleteCall	@DeleteMapping	/deleteCall	Η μέθοδος λαμβάνει ως παράμετρο το email του χρήστη σε μορφή String και διαγράφει την εγγραφή που του αντιστοιχεί στην συλλογή soscalls. Επιστρέφει ένα String σε μορφή JSON ανάλογα με το αποτέλεσμα της λειτουργίας (επιτυχής ή όχι).

Πίνακας 6: Μέθοδοι της κλάσης SOSCallController

SymptomController:

Όνομα μεθόδου	Annotation	URI	Λειτουργία
getSymptom	@GetMapping	/getSymptom	Η μέθοδος λαμβάνει ως παράμετρο τον τίτλο ενός συμπτώματος και επιστρέφει ένα String σε μορφή JSON, που περιέχει τα δεδομένα του (Symptom) από την συλλογή symptoms.
getAllSymptoms	@GetMapping	/getAllSymptoms	Η μέθοδος επιστρέφει μία λίστα με όλα τα συμπτώματα που υπάρχουν στην συλλογή symptoms.
createSymptom	@PostMapping	/createSymptom	Η μέθοδος λαμβάνει ως παράμετρο ένα αντικείμενο της κλάσης Symptom, και το αποθηκεύει στην βάση δεδομένων στην συλλογή symptoms. Επιστρέφει ένα String σε μορφή JSON ανάλογα με το αποτέλεσμα της λειτουργίας (επιτυχής ή όχι).
updateSymptom	@PutMapping	/updateSymptom	Η μέθοδος λαμβάνει ως παράμετρο ένα αντικείμενο της κλάσης Symptom, και το τροποποιεί στην βάση δεδομένων στην συλλογή symptoms. Επιστρέφει ένα String σε μορφή

			JSON ανάλογα με το αποτέλεσμα της λειτουργίας (επιτυχής ή όχι).
deleteSymptom	@DeleteMapping	/deleteSymptom	Η μέθοδος λαμβάνει ως παράμετρο τον τίτλο ενός συμπτώματος σε μορφή String και διαγράφει την εγγραφή που του αντιστοιχεί στην συλλογή symptoms. Επιστρέφει ένα String σε μορφή JSON ανάλογα με το αποτέλεσμα της λειτουργίας (επιτυχής ή όχι).

Πίνακας 7: Μέθοδοι της κλάσης SymptomController

7.2 Επικοινωνία με Firebase

Προκείμενου το Web Service να συνδεθεί με την βάση δεδομένων Firebase για μεταφορά δεδομένων από και προς αυτήν, χρησιμοποιείται το Admin SDK και ορισμένες ρυθμίσεις. Το Admin SDK δίνει δικαιώματα διαχειριστή στην εφαρμογή που το χρησιμοποιεί, τα οποία μπορούν και να περιοριστούν ανάλογα με τις ανάγκες. Προσθέτοντας την αντίστοιχη εξωτερική βιβλιοθήκη (dependency) στο pom.xml, οι λειτουργίες του Admin SDK είναι διαθέσιμες στο πρότζεκτ. Επιπλέον είναι απαραίτητη η δημιουργία ενός κλειδιού (service account key) μέσω της κονσόλας της Firebase που πιστοποιεί την εφαρμογή η οποία ζητάει πρόσβαση στην βάση. Το κλειδί αυτό δημιουργείται σε μορφή JSON και πρέπει να συμπεριληφθεί στα αρχεία του πρότζεκτ.

Η σύνδεση πραγματοποιείται στην κλάση FirebaseInitializer, όπου

1. διαβάζεται το κλειδί που υπάρχει αποθηκευμένο ως αρχείο στο πρότζεκτ
2. ορίζεται το URI στο οποίο βρίσκεται η βάση δεδομένων
3. αρχικοποιείται η σύνδεση με την Firebase με βάση τις παραπάνω παραμέτρους

Εφόσον οι παράμετροι είναι σωστές και η αυθεντικοποίηση της εφαρμογής μέσω του κλειδιού επιτυχημένη, το Web Service μπορεί να πραγματοποιήσει αιτήματα προς την βάση.

8 Επίλογος

Στο κεφάλαιο αυτό συνοψίζεται η παρουσίαση της διπλωματικής εργασίας, εξάγονται κάποια συμπεράσματα και τέλος, προτείνονται μελλοντικές επεκτάσεις.

8.1 Σύνοψη & Συμπεράσματα

Στα προηγούμενα κεφάλαια έγινε μία πλήρης και εμπειριστατωμένη έρευνα και υλοποίηση ενός συστήματος παροχής πληροφοριών πρώτων βοηθειών. Συζητήθηκαν εκτενώς οι ανάγκες ύπαρξης ενός τέτοιου συστήματος σε συνάρτηση με την ζήτηση από το κοινό των κινητών συσκευών. Ακόμα, παρουσιάστηκαν αναλυτικά οι τεχνολογίες που επιλέχθηκαν καθώς και οι λόγοι για τους οποίους αποτελούν πλεονέκτημα ενός σύγχρονου πληροφοριακού συστήματος. Το λειτουργικό σύστημα Android όπως και η αρχιτεκτονική REST, εμφανίστηκαν τις τελευταίες δύο δεκαετίες και το ενδιαφέρον προς αυτές όχι μόνο δεν έχει φθίνει αλλά αυξάνεται συνεχώς. Από την άλλη, η Google Firebase προσφέρει όλες τις υπηρεσίες που είναι απαραίτητες σε μία σύγχρονη εφαρμογή με βασικότερο χαρακτηριστικό αυτό της ταχύτητας και της διάθεσης των δεδομένων σε πραγματικό χρόνο.

Στην συνέχεια, έγινε μία ανασκόπηση του πεδίου των πρώτων βοηθειών και της παρουσίας του στις εφαρμογές για κινητές συσκευές. Υπάρχει πληθώρα διαθέσιμων apps, λίγες όμως από αυτές είναι εμπειριστατωμένες και πιστοποιείται η εγκυρότητα των πληροφοριών τους από φορείς όπως ο Ερυθρός Σταυρός. Εκτός από αυτό, οι περισσότερες έχουν πολλές ενοχλητικές διαφημίσεις και δεν παρουσιάζουν τις πληροφορίες με τέτοιο τρόπο ώστε να διευκολύνουν τον χρήστη σε μία επείγουσα κατάσταση. Στην ελληνική αγορά δε, εντοπίστηκαν μόνο δύο εφαρμογές που περιέχουν τις πληροφορίες στα ελληνικά, μία εκ των οποίων είναι αυτή του Ερυθρού Σταυρού. Τέλος, σε καμία από αυτές που ερευνήθηκαν δεν δίνεται η δυνατότητα κοινοποίησης της παρουσίας του χρήστη.

Στα επόμενα κεφάλαια, παρουσιάστηκαν διεξοδικά η εφαρμογή Android, ο τρόπος χρήσης της και το web service που υλοποιήθηκε για την επικοινωνία με την βάση. Η εφαρμογή έχει διαφορετικές λειτουργίες για δύο ομάδες χρηστών, αυτή του απλού χρήστη και αυτή του διαχειριστή. Ο διαχειριστής μπορεί να τροποποιήσει όλα τα δεδομένα στην βάση που αφορούν τις πρώτες βοήθειες. Εκτός από αυτό το user manual, αναλύθηκε και ο τρόπος υλοποίησης του λογισμικού, τόσο της εφαρμογής όσο και του web service, έτσι ώστε ο πηγαίος κώδικας της εφαρμογής να είναι κατανοητός και διαθέσιμος σε προγραμματιστές που θα τον αναθεωρήσουν στο μέλλον.

8.2 Μελλοντικές επεκτάσεις

Η ανάπτυξη του web service ως ενδιάμεσο της επικοινωνίας με την βάση είναι ένα σημαντικό πλεονέκτημα που δίνει στην εργασία εξαιρετικές δυνατότητες επέκτασης. Εφόσον τα δεδομένα είναι διαθέσιμα στην βάση και υπάρχει λογισμικό που τα διαθέτει με τρόπο προσβάσιμο από πάρα πολλές τεχνολογίες (λόγω της αρχιτεκτονικής REST), είναι πολύ εύκολο να αναπτυχθούν αντίστοιχες εφαρμογές και για άλλες πλατφόρμες. Μπορεί για παράδειγμα να δημιουργηθεί εφαρμογή για λειτουργικό iOS αλλά και για desktop, όπου οι διαχειριστές θα μπορούν να αξιοποιήσουν πληρέστερα τα δικαιώματα τροποποίησης των δεδομένων.

Όσον αφορά την εφαρμογή και τις λειτουργίες της, θα ήταν χρήσιμο να προστεθεί οπτικοακουστικό υλικό για καλύτερη επεξήγηση των πληροφοριών. Θα μπορούσαν επιπλέον να επεκταθούν οι πληροφορίες και να συμπεριλαμβάνουν μαθησιακό υλικό όπως και τεστ, αντίστοιχα με τις εφαρμογές του Ερυθρού Σταυρού που παρουσιάστηκαν στο κεφάλαιο πέντε.

Συνολικά η εφαρμογή μαζί με πιθανές επεκτάσεις της μπορεί να διαθεθεί σε έναν πάροχο υπηρεσιών υγείας, που με την σειρά του θα την παρέχει στους συνδρομητές του. Κατ' αυτόν τον τρόπο οι συνδρομητές θα μπορούν να καλέσουν βοήθεια γνωστοποιώντας την παρουσία τους και την επείγουσα κατάσταση και διασφαλίζοντας πως θα φτάσει έγκαιρα βοήθεια στο σημείο. Σε αυτή την περίπτωση προτείνεται επιπλέον δημιουργία προφίλ χρήστη στην εφαρμογή, που θα περιλαμβάνει ένα πληρέστερο ιατρικό ιστορικό και διευκολύνει την υπηρεσία διάσωσης. Με τα δεδομένα ενός ιατρικού ιστορικού, η εφαρμογή θα μπορούσε να προτείνει στοχευμένες πληροφορίες σε κάθε χρήστη, και να τον προτρέψει να μάθει πώς να διαχειριστεί επείγουσες

κατάστασεις που είναι πιθανότερο να συμβούν σε εκείνον. Τέλος, με την πιθανή ύπαρξη μίας desktop εφαρμογής (χωρίς όμως να είναι απαραίτητο) ο πάροχος που διαθέτει την εφαρμογή στους συνδρομητές, μπορεί εύκολα να ανανεώσει τα δεδομένα των πρώτων βοηθειών χωρίς την διαμεσολάβηση προγραμματιστών για πρόσβαση στην βάση.

9 Πίνακας Εικόνων

Εικόνα 1: Αρχιτεκτονική Android (Android stack)	9
Εικόνα 2: RESTful Web Service	13
Εικόνα 3: Αποθήκευση Δεδομένων σε JSON Tree	15
Εικόνα 4: Αποθήκευση δεδομένων σε collections	16
Εικόνα 5: Εφαρμογή "First Aid and Emergency Techniques"	17
Εικόνα 6: Εφαρμογή "First Aid" από "ESTEPS"	18
Εικόνα 7: Εφαρμογή "Baby and Child First Aid"	18
Εικόνα 9: Εφαρμογές Αμερικανικού Ερυθρού Σταυρού	19
Εικόνα 8: Εφαρμογή "Πρώτες Βοήθειες"	19
Εικόνα 10: Εφαρμογή "First Aid - IFRC"	20
Εικόνα 11: Οθόνη Login	21
Εικόνα 12: Εγγραφή χρήστη	22
Εικόνα 13: Εγγραφή, μήνυμα λάθους	22
Εικόνα 14: Εγγραφή, λάθος δομή email	22
Εικόνα 15: Εγγραφή, το email υπάρχει ήδη	23
Εικόνα 16: Εγγραφή επιτυχής	23
Εικόνα 17: Είσοδος, κενό πεδίο	24
Εικόνα 18: Είσοδος, λάθος δομή email	24
Εικόνα 19: Είσοδος, λανθασμένα δεδομένα	24
Εικόνα 20: Αίτημα πρόσβασης στην τοποθεσία	25
Εικόνα 21: Επιλογή επείγουσας κατάστασης	25
Εικόνα 22: Επιτυχής καταγραφή επείγουσας κλήσης	26
Εικόνα 23: Καρτέλα "ΠΛΗΡΟΦΟΡΙΕΣ"	26
Εικόνα 24: Πληροφορίες Πρώτων Βοηθειών για έγκαιρα	26
Εικόνα 25: Καρτέλα "Πληροφορίες", προσθήκη νέου συμπτώματος	27
Εικόνα 26: Οθόνη καταγραφής νέας κατάστασης	28
Εικόνα 27: Νέα κατάσταση, συμπλήρωση πεδίων	28
Εικόνα 28: Επιτυχής αποθήκευση κατάστασης	28
Εικόνα 29: Νέα κατάσταση στην καρτέλα "Πληροφορίες"	29
Εικόνα 30: Διαχειριστής, Οθόνη ενεργειών	29
Εικόνα 31: Επεξεργασία κατάστασης – συμπτώματος	30
Εικόνα 32: Επιτυχής αποθήκευση αλλαγών	30
Εικόνα 33: Εμφάνιση αλλαγών στην λίστα ενεργειών	30
Εικόνα 34: Επιβεβαίωση διαγραφής	31
Εικόνα 35: Επιτυχής διαγραφή	31
Εικόνα 36: Collection "users"	38
Εικόνα 37: Collection "soscalls"	38
Εικόνα 38: Collection "symptoms"	39
Εικόνα 39: UML com.project.firstaidws.model package	40
Εικόνα 40: UML com.project.firstaidws.service package	40
Εικόνα 41: UML com.project.firstaid.controller package	41
Εικόνα 42: UML com.project.firstaidws.helper package	41

10 Ευρετήριο Πινάκων

Πίνακας 1: Λειτουργίες CRUD σε διάφορα συστήματα	12
Πίνακας 4: Κλάσεις εφαρμογής στο com.example.firstaid.....	34
Πίνακας 5: Αρχεία πόρων της γραφικής διεπιφάνειας στο src/main/res/layout	35
Πίνακας 6: Αρχεία πόρων ορισμού σχημάτων στο src/main/res/drawable	35
Πίνακας 7: Μέθοδοι της κλάσης UserController	42
Πίνακας 8: Μέθοδοι της κλάσης SOSCallController	43
Πίνακας 9: Μέθοδοι της κλάσης SymptomController	44

11 Βιβλιογραφία

- [1] Google, "Android," 2020. [Online]. Available: <https://www.android.com/>.
- [2] Wikipedia, 2020. [Online]. Available: [https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system)).
- [3] "statcounter, GlobalStats," 2020. [Online]. Available: <https://gs.statcounter.com/os-market-share/mobile/worldwide>.
- [4] "Wikipedia," 2020. [Online]. Available: https://en.wikipedia.org/wiki/Android_version_history.
- [5] "statista," 2020. [Online]. Available: <https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/>.
- [6] "statista," 2020. [Online]. Available: <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/>.
- [7] "Android Developers," 2020. [Online]. Available: <https://developer.android.com/guide/platform>.
- [8] "Wikipedia," 2020. [Online]. Available: https://en.wikipedia.org/wiki/Android_software_development.
- [9] "Android Development," 2020. [Online]. Available: <https://developer.android.com/studio/projects>.
- [10] R. Fielding, 2008. [Online]. Available: <https://roy.gbiv.com/untangled/2008/rest-apis-must-be-hypertext-driven>.
- [11] R. Fielding, 2000. [Online]. Available: https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm.
- [12] "Wikipedia," 2020. [Online]. Available: https://en.wikipedia.org/wiki/Representational_state_transfer.
- [13] "Wikipedia," 2020. [Online]. Available: [https://en.wikipedia.org/wiki/Persistence_\(computer_science\)](https://en.wikipedia.org/wiki/Persistence_(computer_science)).
- [14] "Wikipedia," 2020. [Online]. Available: https://en.wikipedia.org/wiki/Web_service.
- [15] "W3org," 2004. [Online]. Available: <https://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/#webservice>.
- [16] "Oracle," [Online]. Available: <https://docs.oracle.com/javaee/6/tutorial/doc/gijqy.html>.
- [17] Wikipedia, 2020. [Online]. Available: <https://en.wikipedia.org/wiki/Firebase>.
- [18] "Spring docs," 2020. [Online]. Available: <https://docs.spring.io/spring/docs/current/javadoc-api/org/springframework/web/bind/annotation/package-summary.html>.