# ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

## ΣΧΟΛΗ ΨΗΦΙΑΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ
## ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ

*ΣΥΣΤΗΜΑ ΚΑΤΑΓΡΑΦΗΣ ΤΟΠΟΘΕΣΙΑΣ ΣΥΣΚΕΥΩΝ*

*Πτυχιακή εργασία*

Δανάη Καμβουσιώρα

Αθήνα, 2021

# ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΣΧΟΛΗ ΨΗΦΙΑΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ

**Τριμελής Εξεταστική Επιτροπή**

Επιβλέπων

Τσερπές Κωνσταντίνος

Επίκουρος Καθηγητής, Τμήμα Πληροφορικής και Τηλεματικής, Χαροκόπειο Πανεπιστήμιο

Μέλη

Δαλάκας Βασίλειος

Καθηγητής, Τμήμα Πληροφορικής και Τηλεματικής, Χαροκόπειο Πανεπιστήμιο

Δημητρακόπουλος Γεώργιος

Επίκουρος Καθηγητής, Τμήμα Πληροφορικής και Τηλεματικής, Χαροκόπειο Πανεπιστήμιο

Η Δανάη Καμβουσιώρα δηλώνω υπεύθυνα ότι:

1. Είμαι ο κάτοχος των πνευματικών δικαιωμάτων της πρωτότυπης αυτής εργασίας και από όσο γνωρίζω η εργασία μου δε συκοφαντεί πρόσωπα, ούτε προσβάλει τα πνευματικά δικαιώματα τρίτων.

2. Αποδέχομαι ότι η ΒΚΠ μπορεί, χωρίς να αλλάξει το περιεχόμενο της εργασίας μου, να τη διαθέσει σε ηλεκτρονική μορφή μέσα από τη ψηφιακή Βιβλιοθήκη της, να την αντιγράψει σε οποιοδήποτε μέσο ή/και σε οποιοδήποτε μορφότυπο καθώς και να κρατά περισσότερα από ένα αντίγραφα για λόγους συντήρησης και ασφάλειας.

# Ευχαριστίες

Η παρούσα πτυχιακή εργασία υλοποιήθηκε στα πλαίσια του προγράμματος προπτυχιακών σπουδών του τμήματος Πληροφορικής και Τηλεματικής στο Χαροκόπειο Πανεπιστήμιο Αθηνών. Η υλοποίηση της πτυχιακής εργασίας ξεκίνησε τον Μάρτιο του 2020 και ολοκληρώθηκε τον Δεκέμβριο του 2020. Με την ολοκλήρωση της πτυχιακής μου εργασίας θα ήθελα να ευχαριστήσω όλους όσους με βοήθησαν και με στήριξαν σε όλη τη διάρκεια των σπουδών μου αλλά και της εκπόνησης της εργασίας.

Ευχαριστώ θερμά τον επιβλέποντα καθηγητή της πτυχιακής μου εργασίας, κύριο Τσερπέ Κωνσταντίνο, για την υποστήριξη και τις πολύτιμες συμβουλές του.

Θα ήθελα, επίσης, να ευχαριστήσω τα μέλη της εξεταστικής επιτροπής κ. Δαλάκα Βασίλειο, και κ. Δημητρακόπουλο Γεώργιο που μου έκαναν την τιμή να αξιολογήσουν την προσπάθεια μου.

Θέλω να ευχαριστήσω τους συμφοιτητές και πλέον φίλους μου Χάρη, Χρήστο, Τζώρτζη, Σοφία, Αλέκο και Σταύρο, που συνεργαστήκαμε σε εργασίες και μαθήματα σε όλη την διάρκεια σπουδών μου.

Τέλος, δε θα μπορούσα να παραλείψω την οικογένεια μου, και το φιλικό μου περιβάλλον που με στήριξαν και με βοήθησαν σε όλη τη διάρκεια των σπουδών μου.

# Περιεχόμενα

**Περίληψη** **8**

**Abstract** **9**

**1 Εισαγωγή** **14**

1.1 Σχετικές Εφαρμογές . . . . . 15

1.2 Σκοπός της εργασίας . . . . . 15

1.3 Δομή πτυχιακής . . . . . 17

**2 Πλατφόρμα Android** **18**

2.1 Ιστορική Αναδρομή . . . . . 18

2.2 Αρχιτεκτονική του Android . . . . . 19

2.2.1 Linux Kernel . . . . . 20

2.2.2 Hardware Abstraction Layer | HAL . . . . . 20

2.2.3 Android Runtime . . . . . 20

2.2.4 Native C/C++ Libraries . . . . . 20

2.2.5 Java API Framework . . . . . 21

2.2.6 System Apps . . . . . 21

2.3 Δομικά Συστατικά Εφαρμογών | App Components . . . . . 22

2.3.1 Activity . . . . . 22

2.3.2 Service . . . . . 23

2.3.3 Content Provider . . . . . 25

2.3.4 Broadcast Receiver . . . . . 25

2.4 Εκτέλεση συστατικών | Μηχανισμός Intent . . . . . 26

2.5 Διεπαφή Χρήστη | User Interface . . . . . 26

2.5.1 User Interface Layout . . . . . 26

2.6 Προγραμματισμός Εφαρμογών Android . . . . . 27

**3 Firebase** **28**

**4 Σχεδιασμός Εφαρμογής** **31**

4.1 Εξήγηση της κατεύθυνσης της υλοποίησης . . . . . 31

4.2 Εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση . . . . . 31

4.2.1 Λειτουργίες Firebase . . . . . 31

4.2.2 Google APIs . . . . . 32

4.3 Διεπαφή χρήστη | User Interface . . . . . 33

4.4 Πρώτη εφαρμογή Android: Καταγραφή τοποθεσίας του χρήστη . . . . . 36

4.4.1 App Components . . . . . 36

4.4.2 Αρχιτεκτονική Εφαρμογής | Component Diagram . . . . . 37

4.4.3 Υπηρεσίες Λογαριασμού . . . . . 37

4.4.4 Καταγραφή τοποθεσίας και εμφάνιση σε χάρτη . . . . . 38

4.4.5 Ειδοποιήσεις | Notifications . . . . . 39

4.4.6 Sequence Diagrams για βασικές λειτουργίες της εφαρμογής . . . . . 40

4.5 Δεύτερη εφαρμογή: Εύρεση τοποθεσίας χαμένης συσκευής | Android . . . . . 42

4.5.1 App Components . . . . . 42

4.5.2 Αρχιτεκτονική Εφαρμογής | Component Diagram . . . . . 42

4.5.3 Υπηρεσίες Λογαριασμού . . . . . 43

4.5.4 Εύρεση τοποθεσίας και εμφάνιση σε χάρτη . . . . . 44

4.5.5 Οδηγίες προς την τοποθεσία . . . . . 44

4.5.6 Αποστολή ειδοποίησης στην πρώτη εφαρμογή . . . . . 45

4.5.7 Sequence Diagrams για βασικές λειτουργίες της εφαρμογής . . . . . 46

4.6 Τρίτη εφαρμογή: Εύρεση τοποθεσίας χαμένης συσκευής | Web . . . . . 48

4.6.1 Αρχιτεκτονική Εφαρμογής | Component Diagram . . . . . 48

4.6.2 Υπηρεσίες Λογαριασμού . . . . . 48

4.6.3 Εύρεση τοποθεσίας και εμφάνιση σε χάρτη . . . . . 48

4.6.4 Αποστολή ειδοποίησης στην πρώτη εφαρμογή . . . . . 49

4.6.5 Οδηγίες προς την τοποθεσία . . . . . 50
4.6.6 Sequence Diagrams για βασικές λειτουργίες της εφαρμογής . . . . . 50
4.6.7 Δημοσίευση διαδικτυακής πλατφόρμας . . . . . 51
4.6.8 Εικόνες web εφαρμογής . . . . . 52

**5 Πειράματα και Αξιολόγηση 54**

5.1 Παρατηρήσεις . . . . . 54
5.2 Android Apps Performance Monitoring & Testing . . . . . 55
5.2.1 Συσκευές που επιλέχθηκαν . . . . . 55
5.2.2 Κριτήρια Αξιολόγησης . . . . . 55
5.2.3 Μέθοδοι και εργαλεία αξιολόγησης . . . . . 56
5.2.4 Αποτελέσματα μετρήσεων . . . . . 57
5.2.5 Συμπεράσματα μετρήσεων . . . . . 60

**6 Συμπεράσματα 61**

6.1 Σύνοψη . . . . . 61
6.2 Μελλοντικοί Στόχοι . . . . . 61

# Περίληψη

Αντικείμενο της παρούσας πτυχιακής εργασίας είναι να προσφέρει τη δυνατότητα στους χρήστες της παραχθείσας πλατφόρμας να γνωρίζουν ανά πάσα στιγμή την τοποθεσία της έξυπνης κινητής συσκευής τους με λειτουργικό Android. Έτσι, ακολούθησε η δημιουργία μιας εφαρμογής για Android κινητές συσκευές, η οποία καταγράφει την τοποθεσία του χρήστη, καθώς και η δημιουργία μιας δεύτερης εφαρμογής, από την οποία μπορεί ο χρήστης να δει την τοποθεσία της συσκευής του, σε περίπτωση που την χάσει. Η δεύτερη εφαρμογή έχει δημιουργηθεί για Android κινητές συσκευές, αλλά και ως διαδικτυακή πλατφόρμα.

Η βασική εφαρμογή εγκαθίσταται στην συσκευή του χρήστη, και του δίνει τη δυνατότητα να καταγράφει την τοποθεσία του είτε κατ' απαίτηση, είτε αδιάκοπα ανά διαστήματα ενός λεπτού, όσο η εφαρμογή βρίσκεται στο παρασκήνιο. Με κάθε ανανέωση της τοποθεσίας, ο χρήστης λαμβάνει μία ειδοποίηση, και μπορεί να δει την τοποθεσία του στον χάρτη της εφαρμογής. Ο χάρτης ανανεώνεται σε πραγματικό χρόνο.

Η δεύτερη εφαρμογή εγκαθίσταται σε μία άλλη συσκευή, σε περίπτωση που ο χρήστης χάσει την πρώτη συσκευή του, και μέσω αυτής έχει τη δυνατότητα να δει σε χάρτη την τελευταία καταγεγραμμένη τοποθεσία της χαμένης συσκευής. Ο χάρτης ανανεώνεται σε πραγματικό χρόνο, και αν η τοποθεσία της χαμένης συσκευής αλλάξει, ο χρήστης θα δει αμέσως τη νέα τοποθεσία στον χάρτη. Με αυτή την εφαρμογή, ο χρήστης μπορεί επίσης να στείλει μία ειδοποίηση στην χαμένη συσκευή. Τέλος, προσφέρει στον χρήστη οδηγίες από την τρέχουσα τοπεθεσία του, προς την τοποθεσία της χαμένης συσκευής. Η web-based έκδοση αυτής της εφαρμογής παρέχει τις ίδιες υπηρεσίες στον χρήστη, καλύπτωντας έτσι την περίπτωση που ο χρήστης δεν έχει κάποια διαθέσιμη συσκευή.

Μετά τη δημιουργία τους, έγιναν δοκιμές επίδοσης (performance testing) των Android εφαρμογών σε low-end και σε high-end συσκευές, ώστε να επιβεβαιωθεί πως οι εφαρμογές λειτουργούν αποδοτικά ανεξαρτήτως των πόρων των συσκευών, αλλά και πως αυτές αντίστοιχα δεν επιβαρύνουν αρνητικά τις συσκευές και την εμπειρία του χρήστη.

**Λέξεις κλειδιά**: [Android, εφαρμογές κινητών συσκευών, εύρεση τοποθεσίας]

# Abstract

The purpose of this dissertation is to offer the users of the developed platform the opportunity to know at any time the location of their smart Android mobile device. Thus, followed the development of an Android application for mobile devices, which tracks the user's location, as well as the development of a second application, from which the user can see the location of his device, in case he loses it. The second application has been developed for both Android mobile devices and web browsers.

The first application is installed on the user's device, enabling him to track his location either on demand or seamlessly while the application is in the background, at intervals of one minute. With every location update, the user gets a push notification, and can also see their current location on the map of the application. The map is updated in real-time.

The second application is installed on another device, in case the user loses his main device, and with it the user has the ability to see the last tracked location of the lost device. The map is updated in real-time, and if the location of the lost device changes, the user using the second app will see the new location immediately on the map. The user is also able to send a push notification to the lost device. Finally, the app provides the user with directions from his current location, to the location of the lost device. The web-based version of this application provides the same services to the user, thus covering the case where the user does not have an available device.

After their development, performance tests of the Android applications, on both low-end and high-end devices, were performed to ensure that the apps operate efficiently regardless of the devices' resources, but that they also don't burden negatively the devices and the user's experience.

**Keywords**: [Android, mobile applications, location tracking]

# Κατάλογος σχημάτων


1 Android Platform Architecture . . . 19
2 Navigation Back Stack. . . . 22
3 Activity Lifecycle. . . . 23
4 Service Lifecycle. . . . 24
5 Content Provider Overview. . . . 25
6 Intent starting another activity. . . . 26
7 View Hierarchy. . . . 27
8 Firebase as a backend. . . . 28
9 Firebase technologies overview. . . . 29
10 Firestore database της εφαρμογής. . . . 32
11 Logo . . . 33
12 Modified Status Bar. . . . 34
13 User login (first app). . . . 34
14 User register (first app). . . . 34
15 User main page (first app). . . . 35
16 User settings (first app). . . . 35
17 Update info dialog fragment (first app). . . . 36
18 First Android App Component Diagram. . . . 37
19 Notification Builder. . . . 39
20 Notification Channel. . . . 39
21 User Register Sequence Diagram. . . . 40
22 User Login Sequence Diagram. . . . 40
23 Location Tracking Sequence Diagram. . . . 41
24 Second Android App Component Diagram. . . . 42
25 User login (second app). . . . 43

26 Password reset dialog (second app). . . . . 43
27 User main page (second app) . . . . . 44
28 Directions question (second app). . . . . 45
29 Directions with polylines (second app). . . . . 45
30 FCM notification. . . . . 46
31 User Login Sequence Diagram. . . . . 46
32 Second Android App Sequence Diagram. . . . . 47
33 Web App Component Diagram. . . . . 48
34 Web app message post request (code snippet). . . . . 49
35 Web app user ID verification (code snippet). . . . . 49
36 Web app message (code snippet). . . . . 49
37 Web app notification front-end (code snippet). . . . . 50
38 User Login Sequence Diagram. . . . . 50
39 Web App Sequence Diagram. . . . . 51
40 Web app login. . . . . 52
41 Web app main page. . . . . 52
42 Web app directions. . . . . 53
43 Web app message. . . . . 53
44 Fields check. . . . . 54
45 Email check. . . . . 54

# Κατάλογος πινάκων


1 Συντομογραφίες . . . . . 13

2 Device Selection . . . . . 55

3 Measurement Tools . . . . . 56

4 App start time (in milliseconds) . . . . . 57

5 Memory Usage during launch (in MB) . . . . . 57

6 Memory Usage right after launch (in MB) . . . . . 57

7 Memory Usage while idle in background (in MB) . . . . . 57

8 Memory Usage while map Activity loads (in MB) . . . . . 57

9 Memory Usage in background while collecting location updates (in MB) . . . . . 58

10 Memory Usage while tracking current user location (in MB) . . . . . 58

11 Memory Usage while directions are calculated (in MB) . . . . . 58

12 Memory Usage notification is sent (in MB) . . . . . 58

13 CPU Usage during launch (in % of total cpu) . . . . . 58

14 CPU Usage right after launch (in % of total cpu) . . . . . 58

15 CPU Usage while idle in background (in % of total cpu) . . . . . 58

16 CPU Usage while map Activity loads (in % of total cpu) . . . . . 58

17 CPU Usage in background while collecting location updates (in % of total cpu) . . 59

18 CPU Usage while tracking current user location (in % of total cpu) . . . . . 59

19 CPU Usage while directions are calculated (in % of total cpu) . . . . . 59

20 CPU Usage while notification is sent (in % of total cpu) . . . . . 59

21 Disk Space (in MB) . . . . . 59

22 Battery Consumption (in % of battery usage) . . . . . 60

23 Battery Consumption (in mAh) . . . . . 60

# Συντομογραφίες

| | |
|---|---|
| GPS | Global Positioning System |
| LBS | Location Based Service |
| OS | Operating System |
| HAL | Hardware Abstraction Layer |
| AOT | Ahead of time |
| JIT | Just in time |
| VM | Virtual Machine |
| UI | User Interface |
| SDK | Software Development Kit |
| IDE | Integrated Development Environment |
| CDN | Content Delivery Network |
| FCM | Firebase Cloud Messaging |
| CPU | Central Processing Unit |
| RAM | Random Access Memory |
| API | Application Programming Interface |

Πίνακας 1: Συντομογραφίες

# 1 Εισαγωγή

Στην εποχή μας είναι σχεδόν δεδομένο πως, αν όχι όλοι, τουλάχιστον η πλειοψηφία των ανθρώπων, έχει στην κατοχή της μία, ή και περισσότερες, έξυπνες κινητές συσκευές (smartphones). Στην αρχή, οι συσκευές αυτές προορίζονταν κυρίως για επαγγελματική χρήση, για την οποία ήταν αναγκαία η σύνδεση στο διαδίκτυο (για παράδειγμα για επικοινωνία μέσω e-mail). Ένα μεγάλο πρόβλημα που αντιμετώπιζαν τότε οι κινητές συσκευές, ήταν η δυσκολία πρόσβασης και πλοήγησης τους σε πλατφόρμες ή εφαρμογές που ήταν ανεπτυγμένες για την πρόσβασή τους μέσω ηλεκτρονικών υπολογιστών, και το πρόβλημα αυτό οφειλόταν κυρίως στο ότι οι εφαρμογές αυτές δεν ήταν σωστά προσαρμοσμένες για το μικρό μέγεθος της οθόνης των κινητών συσκευών.

Για τη λύση των προβλημάτων αυτών, οι περισσότερες ιστοσελίδες ξεκίνησαν την ανάπτυξη ειδικών εκδόσεων, προσαρμοσμένων κατάλληλα στις απαιτήσεις των κινητών συσκευών, πράγμα που προσέφερε μεγαλύτερη επισκεψιμότητα σε αυτές και σίγουρα καλύτερη και ευκολότερη πλοήγηση στους χρήστες.

Πλέον, οι κινητές συσκευές χρησιμοποιούνται από το ευρύ κοινό, και πιο συγκεκριμένα, ο αριθμός των χρηστών έξυπνων κινητών συσκευών παγκοσμίως σήμερα, ξεπερνά τα τρία δισεκατομμύρια και προβλέπεται να αυξηθεί περαιτέρω κατά αρκετές εκατοντάδες εκατομμύρια τα επόμενα χρόνια [1].

Μέρα με τη μέρα, οι συσκευές αυτές γίνονται ολοένα και περισσότερο τεχνολογικά προηγμένες, και προσφέρουν πολλές παραπάνω δυνατότητες από την απλή επικοινωνία με άλλους ανθρώπους. Αναμεταδίδουν συνεχώς τη θέση τους στους πύργους κινητής τηλεφωνίας που εξυπηρετούν το δίκτυό τους, παρέχοντας στον πάροχο κινητής τηλεφωνίας μία συχνά αρκετά ακριβή εικόνα της θέσης του τηλεφώνου [2]. Στα κινητά τηλέφωνα η καταγραφή τοποθεσίας με τη χρήση του Παγκόσμιου Συστήματος Τοποθεσίας (GPS) και ο διαμοιρασμός της, είναι κάποιες από τις τεχνολογικές προόδους που παρέχουν άπλετες διευκολύνσεις στην σύγχρονη καθημερινότητα ενός ανθρώπου. Μερικές τέτοιες διευκολύνσεις αποτελούν η εύρεση της ακριβούς τοποθεσίας του χρήστη, η δυνατότητα αποστολής αυτής της τοποθεσίας στην περίπτωση μιας κρίσιμης κατάστασης, η εύρεση πολλαπλών διαδρομών μεταξύ δύο τοποθεσιών, καθώς και η εύρεση χαμένων συσκευών.

Είναι συνεπώς φανερή η ανάγκη ύπαρξης κατάλληλων εφαρμογών που να μπορούν να προσφέρουν τις προαναφερθείσες διευκολύνσεις με εύκολο και εύχρηστο τρόπο για τους ανθρώπους.

## 1.1 Σχετικές Εφαρμογές

Σήμερα, υπάρχουν πολλές εφαρμογές που εστιάζουν στην εύρεση της τοποθεσίας μιας συσκευής. Στα πλαίσια του σχεδιασμού αυτής της πτυχιακής, διερευνήθηκαν μερικές άλλες εφαρμογές για Android κινητά τηλέφωνα, που βρίσκονται στο κατάστημα εφαρμογών Play Store και προσφέρουν στους χρήστες παρόμοιες υπηρεσίες. Οι εφαρμογές που δοκιμάστηκαν είναι:

- **Find My Phone: Find Lost Phone**: Η εφαρμογή αυτή παρέχει ουσιαστικά τις απαραίτητες υπηρεσίες για να βρει ο χρήστης τη συσκευή του σε περίπτωση που η συσκευή χαθεί/κλαπεί, καθώς και κάποιες λειτουργίες για να αποτρέψουν την κλοπή της συσκευής. Η εφαρμογή είναι διαθέσιμη εδώ.
- **GPS Phone Tracker**: Η εφαρμογή αυτή εστιάζει στη χρήση της μεταξύ πολλών ατόμων ώστε να γίνεται άμεσος διαμοιρασμός της τοποθεσίας τους. Ο κάθε χρήστης δημιουργεί έναν "κύκλο" ατόμων με τους οποίους επιθυμεί να μοιράζεται την τοποθεσία του και αντίστοιχα να έχει ενημέρωση και για τη δική τους τοποθεσία. Η εφαρμογή είναι διαθέσιμη εδώ.

## 1.2 Σκοπός της εργασίας

Σκοπός αυτής της πτυχιακής εργασίας είναι η δημιουργία ενός συστήματος που λειτουργεί σε Android κινητές συσκευές για την καταγραφή της τοποθεσίας του χρήστη, αυτόματα ανά τακτά χρονικά διαστήματα είτε κατ' απαίτηση του χρήστη. Όλες οι λειτουργίες θα δίνονται με έναν εύκολο και κατανοητό στον χρήστη τρόπο, καθώς θα απευθύνεται στον μέσο χρήστη κινητών συσκευών.

Επίσης, ακολούθησε η ανάπτυξη μιας πλατφόρμας και για Android συσκευές αλλά και για το διαδίκτυο, ώστε ο χρήστης να μπορεί να εντοπίσει την τελευταία καταγεγραμμένη τοποθεσία της συσκευής του. Η απεικόνιση της τοποθεσίας αυτής γίνεται σε πραγματικό χρόνο, και ο χάρτης θα ανανεώνεται απευθείας στην καινούργια τοποθεσία, εφόσον αυτή αλλάξει. Από αυτήν την εφαρμογή, ο χρήστης θα έχει επίσης τη δυνατότητα να στείλει μία ειδοποίηση στην χαμένη συσκευή, καθώς και να δει οδηγίες από την τωρινή του τοποθεσία προς την τελευταία καταγεγραμμένη τοποθεσία της χαμένης συσκευής.

Τέλος, μέρος της εργασίας αποτέλεσε και το performance testing των Android εφαρμογών, ώστε να επιβεβαιωθεί η καλή τους απόδοση και σε high-end και σε low-end συσκευές.

Στόχος ήταν η εφαρμογή να:

- Κάνει ακριβή καταγραφή της τοποθεσίας
- Είναι εύχρηστη, να έχει φιλικό προς τον χρήστη περιβάλλον το οποίο θα τον διευκολύνει κατά την χρήση της εφαρμογής. Να είναι δηλαδή μία εφαρμογή που θα μπορεί να τη χρησιμοποιήσει ο χρήστης κάθε επιπέδου χωρίς να έχει ιδιαίτερες γνώσεις και να είναι εξοικειωμένος με την τεχνολογία
- Είναι ευπαρουσίαστη, να έχει δηλαδή όμορφο περιβάλλον που θα ευχαριστεί αισθητικά τον χρήστη
- Έχει καλή απόδοση, ανεξαρτήτως των πόρων των συσκευών, και να μην καταναλώνει πολλή μπαταρία
- Μην επιβαρύνει αρνητικά τη συσκευή και την εμπειρία του χρήστη.

## 1.3 Δομή πτυχιακής

Η παρούσα πτυχιακή εργασία είναι διαρθρωμένη ως εξής:

- Στο 1ο Κεφάλαιο αναλύθηκε ο σκοπός της εργασίας.
- Στο 2ο Κεφάλαιο αναλύεται η πλατφόρμα Android και τα βασικά χαρακτηριστικά της.
- Στο 3ο Κεφάλαιο περιγράφονται οι βασικές έννοιες του Firebase.
- Στο 4ο Κεφάλαιο αναλύεται ο σχεδιασμός των εφαρμογών.
- Στο 5ο Κεφάλαιο αναφέρεται ο τρόπος αξιολόγησης επιδόσεων των εφαρμογών.
- Στο 6ο Κεφάλαιο αναφέρονται τα συμπεράσματα της εργασίας.

# 2 Πλατφόρμα Android

## 2.1 Ιστορική Αναδρομή

Το 2003, αρκετά χρόνια δηλαδή πριν τη χρήση του όρου "smartphone" και την κυκλοφορία του iPhone από την Apple (2007) , η εταιρεία Android Inc ιδρύθηκε στην Καλιφόρνια από τους Rich Miner, Nick Sears, Chris White, και Andy Rubin. Ο Rubin δήλωσε σε ένα συνέδριο στο Τόκιο το 2013, πως το λειτουργικό σύστημα Android αρχικά προοριζόταν για τη βελτίωση των λειτουργικών συστημάτων που χρησιμοποιούσαν οι ψηφιακές κάμερες [3]. Λίγο καιρό αργότερα έκριναν πως η αγορά για ψηφιακές φωτογραφικές μηχανές δεν ήταν μεγάλη, και έτσι η Android Inc επικεντρώθηκε στα κινητά τηλέφωνα. Ο ίδιος ο Rubin είχε δηλώσει πως η εταιρεία Android Inc επρόκειτο να αναπτύξει «πιο έξυπνες κινητές συσκευές που γνωρίζουν περισσότερο την τοποθεσία και τις προτιμήσεις του κατόχου της» [4].

Το 2005 η Android Inc εξαγοράστηκε από την Google. Ο Rubin και άλλα ιδρυτικά μέλη παρέμειναν για να συνεχίσουν να αναπτύσσουν το λειτουργικό σύστημα υπό τους νέους ιδιοκτήτες τους. Η απόφαση λήφθηκε ώστε να χρησιμοποιηθεί το Linux ως βάση για το λειτουργικό σύστημα Android, πράγμα που σήμαινε ότι το ίδιο το Android θα μπορούσε να προσφέρεται δωρεάν σε κατασκευαστές κινητών τηλεφώνων. Θεώρησαν λοιπόν ότι θα ήταν κερδοφόρο να προσφέρουν άλλες υπηρεσίες που χρησιμοποιούν το λειτουργικό σύστημα, συμπεριλαμβανομένων των εφαρμογών [5].

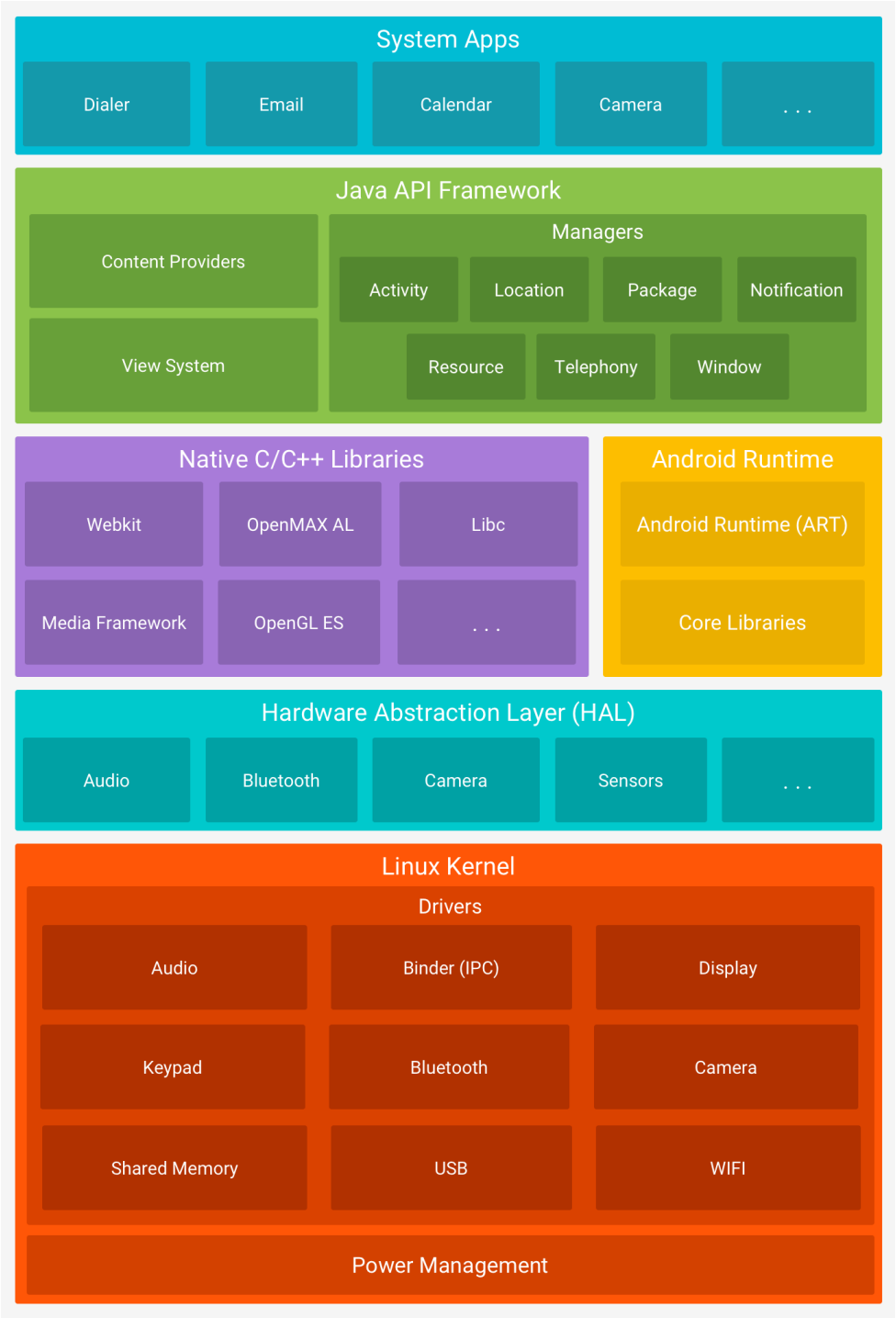
Το 2007, η Apple κυκλοφόρησε το πρώτο iPhone. Την ίδια χρονιά, η Google για την καταπολέμηση της Apple χρησιμοποίησε τον σχηματισμό Open Handset Alliance, ο οποίος ήταν μία συνεργασία πολλών μεγάλων εταιρειών των τηλεπικοινωνιών (μεταξύ των οποίων οι T-Mobile, HTC, Motorola και φυσικά η Google) και αποσκοπούσε στην ανάπτυξη ανοιχτών προδιαγραφών για κινητές συσκευές [5].

Έχουν περάσει 12 χρόνια από τότε που το πρώτο επίσημο τηλέφωνο με λειτουργικό Android έκανε την εμφάνιση του στην αγορά. Το πρώτο κινητό τηλέφωνο που βγήκε στην αγορά με λειτουργικό Android ήταν το HTC Dream (γνωστό επίσης ως T-Mobile G1 στις Ηνωμένες Πολιτείες και σε κάποιες χώρες της Ευρώπης) με την έκδοση 1.0, τον Οκτώβρη του 2008 [6].

Λίγα χρόνια μετά την κυκλοφορία του Android 1.0, τα smartphone που είχαν εγκατεστημένο αυτό το λειτουργικό σύστημα ήταν παντού. Σήμερα έχει γίνει το πιο δημοφιλές λειτουργικό σύστημα φορητών συσκευών στον κόσμο, με πάνω από το 70% των χρηστών κινητών συσκευών της παγκόσμιας αγοράς [7], νικώντας τους πολλούς ανταγωνιστές του, όπως BlackBerry και Windows Phone. Το iOS της Apple είναι η μόνη πλατφόρμα που εξακολουθεί να αποτελεί σοβαρό ανταγωνιστή του Android και αυτή η κατάσταση δεν φαίνεται ότι θα αλλάξει σύντομα [5].

## 2.2 Αρχιτεκτονική του Android

Ουσιαστικά το Android είναι μία στοίβα λογισμικού ανοιχτού κώδικα, βασισμένη σε Linux. Στο σχήμα αναπαρίστανται τα επίπεδα αυτής της στοίβας, τα οποία περιγράφονται εν συντομία παρακάτω [8].




Σχήμα 1: Android Platform Architecture

### 2.2.1 Linux Kernel

Ο πυρήνας Linux είναι η υποδομή της πλατφόρμας Android, και σε αυτόν στηρίζονται όλα τα πάνω επίπεδα της στοίβας. Η χρήση αυτού επιτρέπει στο Android να εκμεταλλευτεί βασικά χαρακτηριστικά ασφαλείας, διαχείρισης μνήμης και οδηγούς συσκευών. Ο πυρήνας λειτουργεί επίσης ως ένα ενδιάμεσο επίπεδο αφαίρεσης μεταξύ της στοίβας λογισμικού και του υλικού.

### 2.2.2 Hardware Abstraction Layer | HAL

Το επίπεδο αυτό παρέχει διεπαφές (standard interfaces) για χρήση των δυνατοτήτων των συστατικών hardware από το Java API framework. Αποτελείται από πολλαπλές βιβλιοθήκες, κάθε μία υλοποιεί μία συγκεκριμένη διεπαφή για ένα τύπο συστατικού hardware. Όταν ένα framework API καλεί για χρήση μία συσκευή, το Android φορτώνει τη βιβλιοθήκη για αυτή τη συσκευή.

### 2.2.3 Android Runtime

Το επίπεδο αυτό τρέχει τον εκτελέσιμο κώδικα των εφαρμογών.

- Για εκδόσεις Android κάτω από 5.0 (API level 21): χρησιμοποιείται το Dalvik VM, το οποίο είναι βελτιστοποιημένο Java VM με χαμηλές απαιτήσεις μνήμης. Βασίζεται σε Just-in-time (JIT) μεταγλώττιση, δηλαδή ο DEX JIT compiler (dx tool) μεταγλωττίζει τα Java bytecodes σε γλώσσα μηχανής κατά την εκτέλεση του κώδικα (run time) και όχι πρωτύτερα.
- Για εκδόσεις Android πάνω από 5.0 (API level 21): χρησιμοποιείται το ART VM. Βασίζεται σε Ahead Of Time (AOT) μεταγλώττιση, δηλαδή μετατρέπει το bytecode ενός VM σε γλώσσα μηχανής πριν την εκτέλεση (κατά την εγκατάσταση, δηλαδή, της εφαρμογής με το dex2oat tool), πράγμα που προσφέρει καλύτερη απόδοση και μικρότερη κατανάλωση ενέργειας. Επιπλέον, έχει καλύτερο περιβάλλον ανάπτυξης και debugging.

Το ART και το Dalvik είναι συμβατά, έτσι εφαρμογές που αναπτύχθηκαν για το Dalvik θα πρέπει να τρέχουν και με ART, αλλά το αντίθετο μπορεί να μην αληθεύει.

### 2.2.4 Native C/C++ Libraries

Το επίπεδο αυτό περιλαμβάνει ένα σύνολο από βιβλιοθήκες γραμμένες σε C/C++ οι οποίες χρησιμοποιούνται σε πολλά συστατικά του Android system, π.χ., ART και HAL. Το Android παρέχει Java Framework APIs για να διαθέτει τη λειτουργικότητα των βιβλιοθηκών στις εφαρμογές.

### 2.2.5 Java API Framework

Όλο το σύνολο των δυνατοτήτων και χαρακτηριστικών του Android OS είναι διαθέσιμο στον προγραμματιστή μέσω APIs γραμμένα σε Java. Τα APIs είναι τα δομικά στοιχεία που χρειάζεται ένας προγραμματιστής για να δημιουργήσει Android εφαρμογές επαναχρησιμοποιώντας βασικά δομικά στοιχεία και υπηρεσίες.

Κυριότερα δομικά συστατικά και υπηρεσίες:

- View System για την ανάπτυξη της διεπαφής χρήσης της εφαρμογής (app's UI) π.χ., lists, grids, text boxes, buttons κλπ.
- Content Providers: επιτρέπει στις εφαρμογές να έχουν πρόσβαση σε δεδομένα από άλλες εφαρμογές (π.χ. εφαρμογή επαφών) ή να μοιράζουν τα δικά τους δεδομένα.
- Ο Resource Manager παρέχει πρόσβαση σε πόρους, αρχεία πλην των αρχείων κώδικα, π.χ. τοπικές συμβολοσειρές (localized strings), γραφικά (graphics), και αρχεία που ορίζουν τη διάταξη (layout files).
- Ο Notification Manager επιτρέπει σε όλες τις εφαρμογές να εμφανίζουν μηνύματα στη γραμμή κατάστασης (custom alerts in the status bar).
- Ο Activity Manager διαχειρίζεται το κύκλο ζωής των εφαρμογών και παρέχει την κοινή στοίβα πλοήγησης από εφαρμογή σε εφαρμογή με το back (navigation back stack).

### 2.2.6 System Apps

Το Android συνοδεύεται από ένα σύνολο από βασικές εφαρμογές: email, SMS messaging, calendar, internet browsing, contacts, κλπ. Επιπλέον, παρέχουν βασικές δυνατότητες που μπορεί να χρησιμοποιήσει ένας προγραμματιστής σε μία νέα εφαρμογή. Π.χ. εάν η εφαρμογή σας θέλετε να στείλει ένα SMS, δεν απαιτείται να αναπτύξετε εξ αρχής τη λειτουργικότητα αλλά να καλέσετε την SMS εφαρμογή που είναι ήδη εγκατεστημένη.

## 2.3 Δομικά Συστατικά Εφαρμογών | App Components

Υπάρχουν τέσσερις διαφορετικοί τύποι συστατικών:

- Activity
- Service
- Content Provider
- Broadcast Receiver

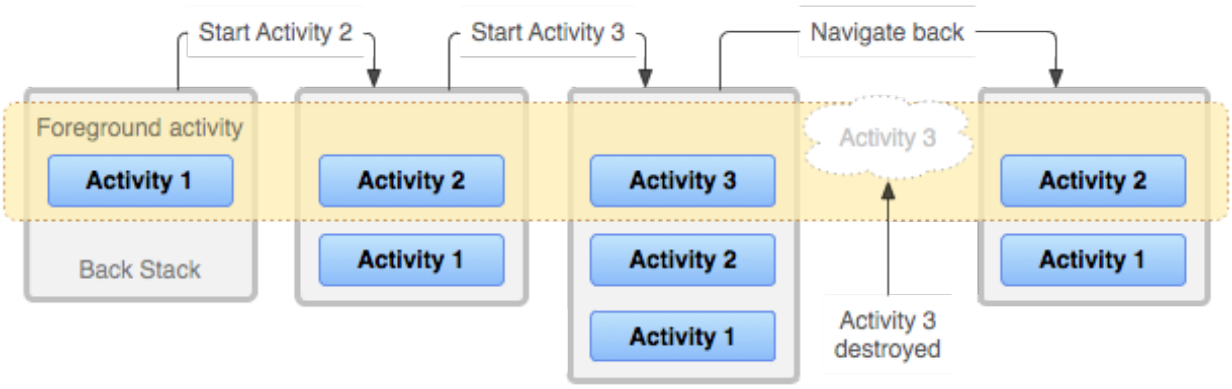
Κάθε τύπος συστατικού εξυπηρετεί έναν διακριτό σκοπό και έχει εναν διακριτό κύκλο ζωής, που ορίζει το πώς κάθε συστατικό δημιουργείται (created) και καταστρέφεται (destroyed).

Μία εφαρμογή αποτελείται από ένα σύνολο συστατικών. Πιο συγκεκριμένα, αποτελείται από ένα ή περισσοτέρα Activities και από **κανένα** ή περισσότερα Services, Broadcast Receivers και Content Providers [9].

### 2.3.1 Activity

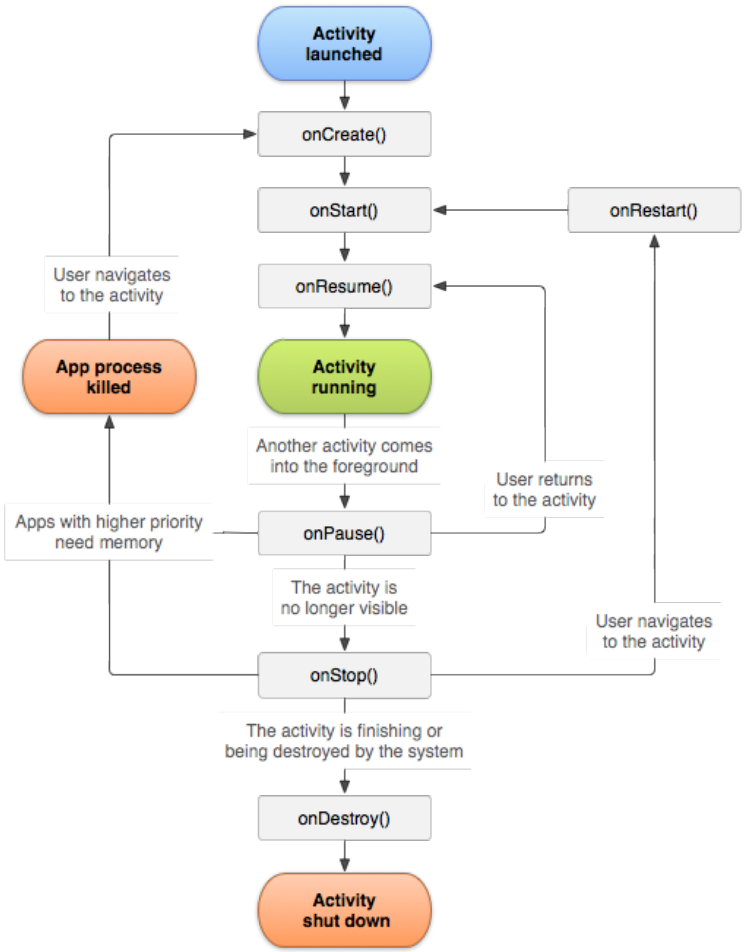
Είναι το σημείο εισόδου για διάδραση με το χρήστη, αφορά το στρώμα παρουσίασης (presentation layer) της εφαρμογής και αντιστοιχεί σε μία οθόνη (screen) της εφαρμογής. Μία εφαρμογή δηλαδή μπορεί να έχει περισσότερα του ενός activities για διαχείριση διαφορετικών οθονών της εφαρμογής.

Κάθε activity έχει το δικό του κύκλο ζωής και είναι υπεύθυνο να αποθηκεύει την δική του κατάσταση κατά το κύκλο ζωής του. Ο Activity Manager που αναφέρθηκε προηγουμένως, παρακολουθεί τα ενεργά Activities, τις καταστάσεις στις οποίες βρίσκονται, και παρέχει την κοινή στοίβα πλοήγησης από εφαρμογή σε εφαρμογή με το back (navigation back stack).




Σχήμα 2: Navigation Back Stack.

Μία εφαρμογή έχει ένα ή περισσότερα activities και μία διεργασία Linux που τα περιέχει. Κάθε διεπαφή χρήστη (οθόνη) αναπαρίσταται από μία **κλάση** Activity (π.χ. MainActivity extends Activity), η οποία περιέχει τις απαραίτητες μεθόδους για να διατηρηθεί ο κύκλος ζωής ενός Activity (Activity Lifecycle), ο οποίος φαίνεται στην παρακάτω εικόνα.




Σχήμα 3: Activity Lifecycle.

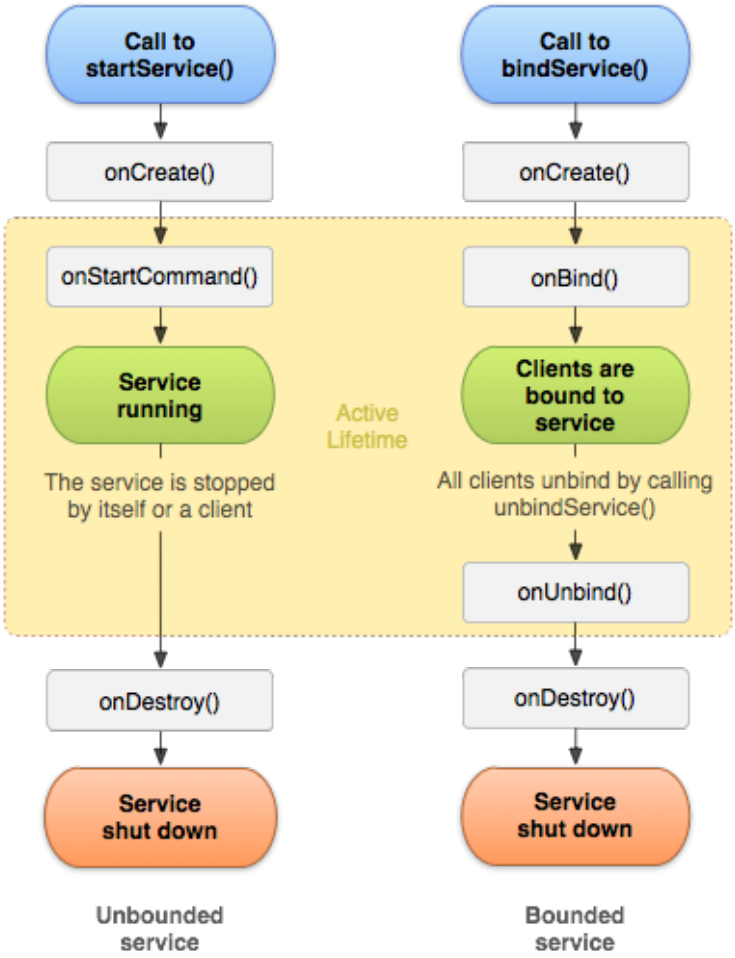
### 2.3.2 Service

Είναι διεργασία που τρέχει στο παρασκήνιο (background) χωρίς τη παρέμβαση του χρήστη, για να εκτελέσει, είτε λειτουργίες μακράς διάρκειας, είτε λειτουργίες για άλλες απομακρυσμένες διεργασίες και **δεν** παρέχει διεπαφή χρήσης (UI) στον χρήστη. Μόλις ξεκινήσει, μπορεί να συνεχίσει να λειτουργεί ακόμα και αν ο χρήστης ανοίξει κάποια άλλη εφαρμογή. Υπάρχουν πολλά built-in services σε Android συσκευές (π.χ Αναπαραγωγή μουσικής στο παρασκήνιο). Αυτές είναι οι τρεις κατηγορίες των services [10]:

- **Foreground Service:** εκτελεί κάποια λειτουργία η οποία είναι αντιληπτή από τον χρήστη, καθώς πρέπει να απεικονίζει μία επίμονη ειδοποίηση (persistent notification) στο status bar. Αυτού του τύπου services συνεχίζουν να λειτουργούν στο παρασκήνιο ακόμα και όταν χρήστης δεν αλληλεπιδρά με την εφαρμογή.

- **Background Service:** εκτελεί κάποια λειτουργία που δεν παρατηρείται άμεσα από τον χρήστη.
- **Bound Service:** ένα bound service προσφέρει μία διεπαφή πελάτη-διακομιστή που επιτρέπει σε στοιχεία (components) να αλληλεπιδρούν με το service, να στέλνουν αιτήματα και να λαμβάνουν αποτελέσματα. Ένα bound service εκτελείται μόνο εφόσον ένα άλλο στοιχείο εφαρμογής είναι δεσμευμένο (bound) σε αυτό. Πολλαπλά στοιχεία μπορούν να δεσμευτούν στο ίδιο service ταυτόχρονα, αλλά όταν αποδεσμευτούν όλα, το service καταστρέφεται.

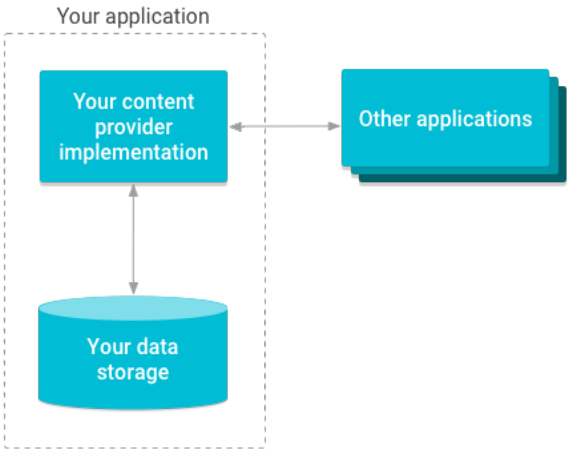
Η **κλάση** Service είναι η βασική κλάση για διαδικασίες που εκτελούνται στο παρασκήνιο (π.χ. HelloService extends Service). Δεν είναι προκαθορισμένα μία ξεχωριστή διεργασία ούτε ένα ξεχωριστό thread. **Αποτελεί τμήμα του main thread**, αλλά αν επιθυμούμε μπορούμε να καθορίσουμε ότι θα τρέχει σε ξεχωριστή διεργασία.




Σχήμα 4: Service Lifecycle.

### 2.3.3 Content Provider

Οι Content Providers διαμοιράζονται αποθηκευτικούς χώρους που είναι προσπελάσιμοι από περισσότερες εφαρμογές, βοηθούν μία εφαρμογή να διαχειριστεί την πρόσβαση σε δεδομένα που είναι αποθηκευμένα από μόνα τους ή αποθηκευμένα από άλλες εφαρμογές, και παρέχει έναν τρόπο κοινοποίησης δεδομένων με άλλες εφαρμογές. Ενσωματώνουν τα δεδομένα και παρέχουν μηχανισμούς για τον ορισμό της ασφάλειας αυτών. Υπάρχει μία ενσωματωμένη βάση δεδομένων στο λειτουργικό Android (SQLite database), στην οποία μπορούν να αποθηκεύουν ή να διαβάζουν δεδομένα οι Content Providers. Το πιο σημαντικό είναι ότι ένας Content Provider μπορεί να διαμορφωθεί ώστε να επιτρέπεται σε άλλες εφαρμογές να έχουν ασφαλή πρόσβαση και να τροποποιούν τα δεδομένα μιας εφαρμογής.



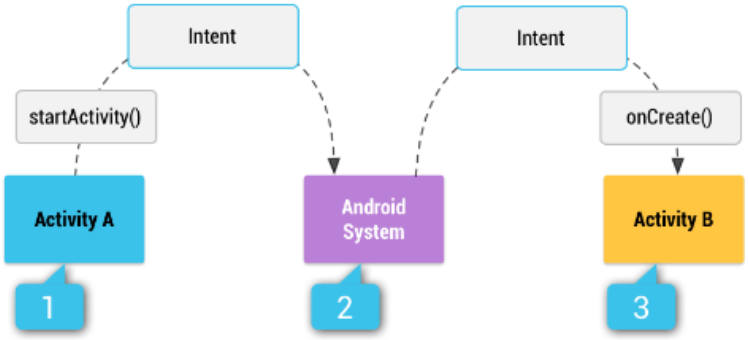

Σχήμα 5: Content Provider Overview.

### 2.3.4 Broadcast Receiver

Οι ανακοινώσεις (broadcasts) μπορεί είτε να παράγονται από το λειτουργικό (**OS-generated**), π.χ. χαμηλή στάθμη μπαταρίας, είτε να παράγονται από τον χρήστη (**User-generated**), π.χ. ενεργοποίηση κάποιου χαρακτηριστικού, όπως σύνδεση σε Wi-fi. Η εφαρμογή λαμβάνει αυτά τα broadcasts με τη χρήση των Broadcast Receivers και αντιδρά κατάλληλα, προκαλώντας την εκτέλεση κώδικα εκτός της κανονικής ροής. Το σύστημα μπορεί να στείλει broadcasts ακόμα και σε εφαρμογές που δεν τρέχουν. Δεν διαθέτουν διεπαφή χρήστη (UI) αλλά μπορούν να δημιουργήσουν ένα status bar notification για να ειδοποιήσουν το χρήστη εάν ένα broadcast event συμβαίνει.

## 2.4 Εκτέλεση συστατικών | Μηχανισμός Intent

Στο Android σχεδόν όλες οι αλληλεπιδράσεις γίνονται μέσω intents. Intent είναι ένας μηχανισμός που καθορίζει ποια συγκεκριμένη ενέργεια πρέπει να εκτελεστεί, στέλνοντας ασύγχρονα μηνύματα. Ο ρόλος ενός intent είναι να ειδοποιηθεί το ActivityManagerService ότι ο χρήστης επιθυμεί να γίνει κάποια ενέργεια. Ένα intent είναι ένα αντικείμενο της κλάσης Intent, το οποίο περιέχει κάποιο περιεχόμενο. Μέσω των intents μπορούμε να ξεκινήσουμε την εκτέλεση των Activities, των Services και των Broadcast Receivers, αλλά όχι των Content Providers. [11]



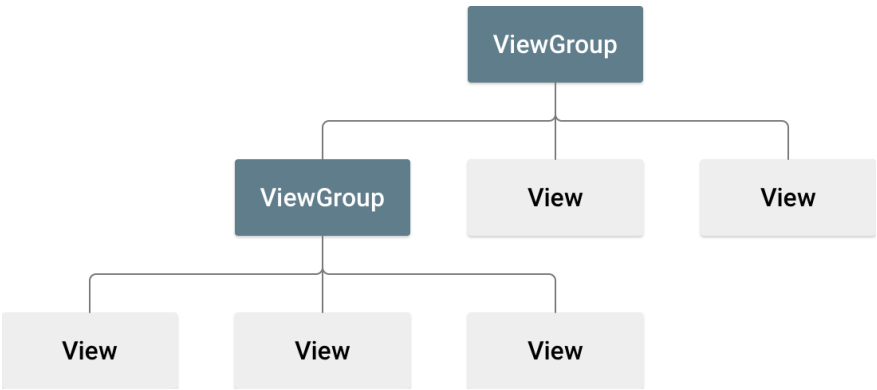

Σχήμα 6: Intent starting another activity.

## 2.5 Διεπαφή Χρήστη | User Interface

Ο τελικός χρήστης συνήθως δεν γνωρίζει τι συμβαίνει πίσω από την εφαρμογή που τρέχει, και τον ενδιαφέρει μόνο το γραφικό περιβάλλον και η ευχρηστία της εφαρμογής. Γι'αυτόν τον λόγο, User Interface και λειτουργικότητα της εφαρμογής είναι δύο αλληλένδετοι όροι, και γι'αυτό μεγάλες εταιρείες δίνουν τόση βάση στον σχεδιασμό της διεπαφής χρήστη. Πολλές φορές μάλιστα, είναι δυσκολότερος ο σχεδιασμός της διεπαφής μιας εφαρμογής από την υλοποίησή της.

### 2.5.1 User Interface Layout

Το layout καθορίζει τη δομή για μία διεπαφή χρήστη σε μία εφαρμογή. Όλα τα στοιχεία σε ένα layout χτίζονται χρησιμοποιώντας μία ιεραρχία από αντικείμενα View (στοιχείο το οποίο βλέπει ο χρήστης και μπορεί να αλληλεπιδράσει με αυτό) ή αντικείμενα ViewGroup (αόρατο "δοχείο" που οργανώνει τα Views).




Σχήμα 7: View Hierarchy.

Υπάρχουν διάφορες κατηγορίες layouts, εκ των οποίων οι πιο χρησιμοποιούμενες είναι:

- **Linear Layout:** εδώ η διάταξη των στοιχείων γίνεται είτε οριζόντια είτε καταφόρυφα.
- **Relative Layout:** εδώ υπάρχει μεγαλύτερη ευελιξία στη διάταξη των στοιχείων, καθώς η θέση ενός στοιχείου ορίζεται με βάση τη θέση του προηγούμενου στοιχείου.
- **Constraint Layout:** μοιάζει με το Relative Layout εφόσον η θέση ενός στοιχείου ορίζεται με βάση την θέση κάποιου άλλου στοιχείου, αλλά προσφέρει ακόμα μεγαλύτερη ευελιξία, καθώς προσφέρει περισσότερες επιλογές τοποθέτησης. Βελτιστοποιεί την ιεραρχία των View και αποφεύγει το nesting στην ιεραρχία.

Η δήλωση των layouts γίνεται είτε μέσω των αρχείων xml, είτε απευθείας μέσα σε ένα Activity. Τα αρχεία xml είναι ένας στατικός τρόπος δημιουργίας των γραφικών στοιχείων, αποθηκεύονται σε συγκεκριμένο φάκελο του project και τα καλούμε μέσω των activities.

## 2.6 Προγραμματισμός Εφαρμογών Android

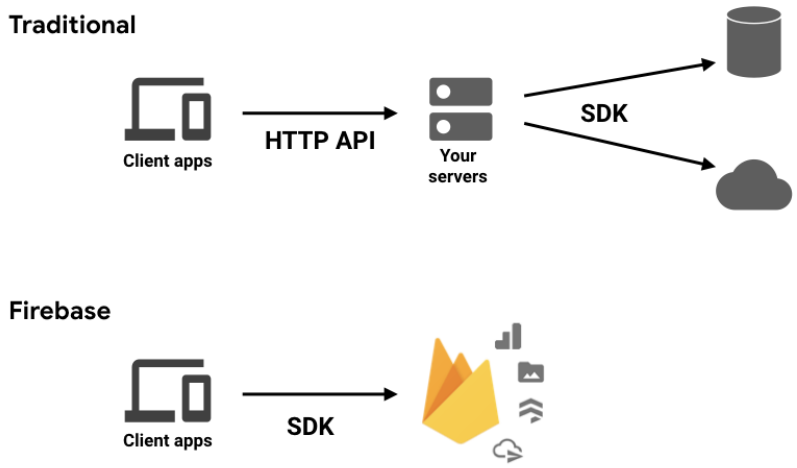
Μερικές **γλώσσες προγραμματισμού** που χρησιμοποιούνται για την ανάπτυξη τέτοιων εφαρμογών είναι:

- **Java:** είναι η επίσημη γλώσσα προγραμματισμού Android, έχει δηλαδή την περισσότερη στήριξη από την Google.
- **Kotlin:** είναι μία καινούρια και πιο μοντέρνα γλώσσα προγραμματισμού. Έχει αναγνωριστεί τα τελευταία χρόνια ως η δεύτερη επίσημη γλώσσα για Android και η χρήση της αυξάνεται όλο και περισσότερο χάρη στην απλότητά της και στο γεγονός ότι με μικρότερα κομμάτια κώδικα μπορεί κανείς να πετύχει το ίδιο αποτέλεσμα που θα πετύχαινε με αισθητά μεγαλύτερα κομμάτια κώδικα στην Java.
- **C/C++**
- **C#**

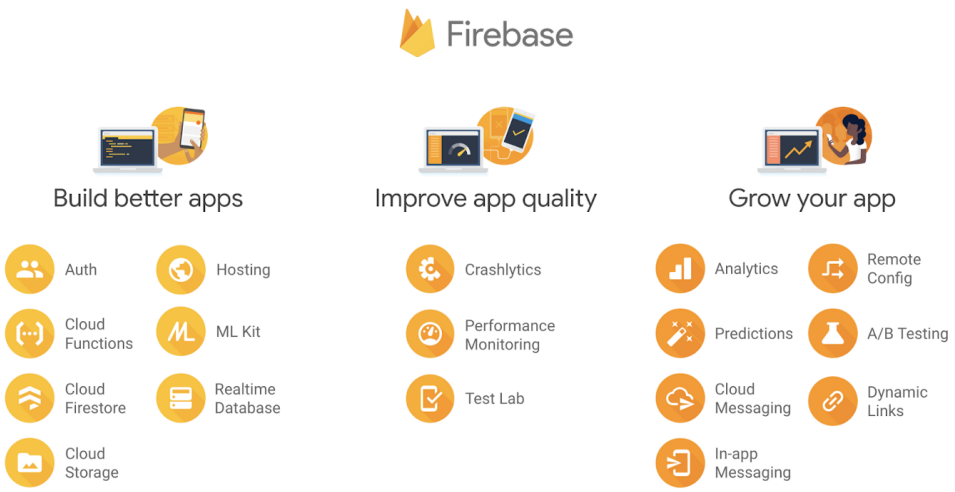
Για να αναπτύξει κάποιος μία εφαρμογή για Android συσκευές χρειάζεται το Android SDK (software development kit), ένα σύνολο εργαλείων, απαραίτητο για την δημιουργία εφαρμογών. Μπορεί να χρησιμοποιηθεί μόνο του με τη χρήση του command-prompt, ή με τη χρήση κάποιου γραφικού περιβάλλοντος (IDE - Integrated Development Environment), όπως το Android Studio ή το Eclipse. Για να δει την εφαρμογή που έχει αναπτυχθεί μπορεί κάποιος είτε να χρησιμοποιήσει έναν emulator ή μία κανονική κινητή συσκευή.

# 3 Firebase

Το Firebase είναι η πλατφόρμα της Google για την ανάπτυξη εφαρμογών για κινητές συσκευές ή για διαδικτυακές εφαρμογές (web-based apps). Κυκλοφόρησε το καλοκαίρι του 2016 και στόχος του είναι να παρέχει τα εργαλεία και την υποδομή που χρειάζεται ένας προγραμματιστής για να δημιουργήσει εξαιρετικές εφαρμογές και να αναπτύξει μία επιτυχημένη επιχείρηση [12]. Δεν αντικαθιστά τα υπάρχοντα API για τη δημιουργία εφαρμογών Android, iOS ή Web. Είναι μία βελτίωση, που παρέχει κοινές υπηρεσίες που ίσως χρειαστούν στην ανάπτυξη μιας εφαρμογής - όπως μία βάση δεδομένων, τον ασφαλή έλεγχο ταυτότητας (authentication), ανταλλαγή μηνυμάτων, και άλλα. Επιπλέον, παρέχει τεχνολογίες που μπορούν να προστεθούν σε μία εφαρμογή, που βοηθούν στην ανάπτυξη της επιχείρησης.




Σχήμα 8: Firebase as a backend.




Σχήμα 9: Firebase technologies overview.

Πολύ συνοπτικά για τις σημαντικότερες από αυτές τις τεχνολογίες:

- **Authentication:** παρέχει με ασφάλεια σύνδεση (login) και αναγνώριση (authentication) των χρηστών στην εφαρμογή.

- **Realtime Database και Cloud Firestore:** παρέχουν υπηρεσίες βάσης δεδομένων (NoSQL) σε πραγματικό χρόνο. Έχουν ξεχωριστά προτερήματα και αδυναμίες, αλλά η επιλογή ανάμεσα τους εξαρτάται συνήθως από τις ανάγκες της εφαρμογής.

- **Cloud Storage:** παρέχει μαζικά επεκτάσιμη (scalable) αποθήκευση αρχείων.

- **Cloud Messaging:** επιτρέπει την αποστολή push μηνυμάτων στους χρήστες της εφαρμογής για να φανεί κάποια ενδιαφέρουσα πληροφορία.

- **In-App Messaging:** βοηθάει στην εμφάνιση στοχευμένων, προσαρμόσιμων μηνυμάτων στους χρήστες για να αλληλεπιδράσουν με βασικές δυνατότητες της εφαρμογής. Ενώ τα μηνύματα μέσω Cloud Messaging στέλνονται από τον server (ή το Firebase Console), στο In-App Messaging τα μηνύματα στέλνονται μέσα από την ίδια την εφαρμογή και εμφανίζονται όσο ο χρήστης την χρησιμοποεί.

- **Performance Monitoring:** είναι μία υπηρεσία που προσφέρει πληροφορίες σχετικά με τα χαρακτηριστικά απόδοσης εφαρμογών iOS, Android και ιστού. Μέσω αυτού, γίνεται συλλογή δεδομένων απόδοσης από την εφαρμογή, και στη συνέχεια, εμφανίζονται στο Firebase console. Οι βασικές μετρήσεις που προσφέρει είναι:
  - *App start time:* Εδώ μετριέται ο χρόνος μεταξύ της στιγμής που ο χρήστης ανοίγει την εφαρμογή, και της στιγμής της απόκρισης της εφαρμογής.
  - *App-in-foreground time:* Εδώ μετριέται ο χρόνος που η εφαρμογή εκτελείται στο προσκήνιο και είναι διαθέσιμη στον χρήστη (όταν δηλαδή είναι minimized αλλά δεν έχει τερματιστεί).
  - *App-in-background time:* Εδώ μετριέται ο χρόνος εκτέλεσης της εφαρμογής στο παρασκήνιο.

  Πέρα από αυτές, δίνεται η δυνατότητα στον χρήστη να κάνει custom μετρήσεις σε συγκεκριμένες ενέργειες της εφαρμογής του, με την προσθήκη μερικών γραμμών επιπλέον κώδικα.

# 4 Σχεδιασμός Εφαρμογής

## 4.1 Εξήγηση της κατεύθυνσης της υλοποίησης

Η εργασία χωρίζεται σε δύο ξεχωριστές ενότητες. Η πρώτη ενότητα περιλαμβάνει την ανάπτυξη της εφαρμογής για Android κινητές συσκευές, με σκοπό να καταγράφεται η τοποθεσία του χρήστη στη βάση δεδομένων. Η δεύτερη ενότητα, περιλαμβάνει την ανάπτυξη της εφαρμογής (και για Android κινητές συσκευές αλλά και σε διαδικτυακή πλατφόρμα) που δίνει τη δυνατότητα στον χρήστη να εντοπίσει την τελευταία καταγεγραμμένη τοποθεσία της συσκευής του σε περίπτωση που αυτή χαθεί ή κλαπεί, να στείλει μία ειδοποίηση στη συσκευή αυτή, καθώς και να προσφέρει στον χρήστη οδηγίες προς αυτή την τοποθεσία.

Αυτή η κατεύθυνση για τη δεύτερη εφαρμογή ακολουθήθηκε ώστε να κάνει εύκολη στον χρήστη τη διαδικασία εντοπισμού της χαμένης συσκευής. Σε περίπτωση που ο χρήστης δεν έχει διαθέσιμη κάποια άλλη Android κινητή συσκευή για να χρησιμοποιήσει την δεύτερη εφαρμογή, μπορεί να χρησιμοποιήσει τη διαδικτυακή εφαρμογή.

## 4.2 Εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση

Αρχικά, η ανάπτυξη των Android εφαρμογών έγινε σε γλώσσα Java και με τη χρήση του Android Studio. Η ελάχιστη έκδοση Android που απαιτείται για να λειτουργήσει η εφαρμογή είναι Android 7.0 (Nougat) - API 24. Χρησιμοποιήθηκαν για την δοκιμή των εφαρμογών και emulator (Pixel 2 API 27), αλλά και κανονικές κινητές συσκευές, που αναφέρονται πιο αναλυτικά στο Κεφάλαιο 5.

Για την web-based εφαρμογή χρησιμοποιήθηκαν HTML-CSS-Bootstrap 4, JavaScript και node.js στο backend. Η εφαρμογή έχει γίνει deployed στον Heroku server.

Χρησιμοποιήθηκαν λειτουργίες του Firebase και για τις Android εφαρμογές αλλά και για τη web-based εφαρμογή.

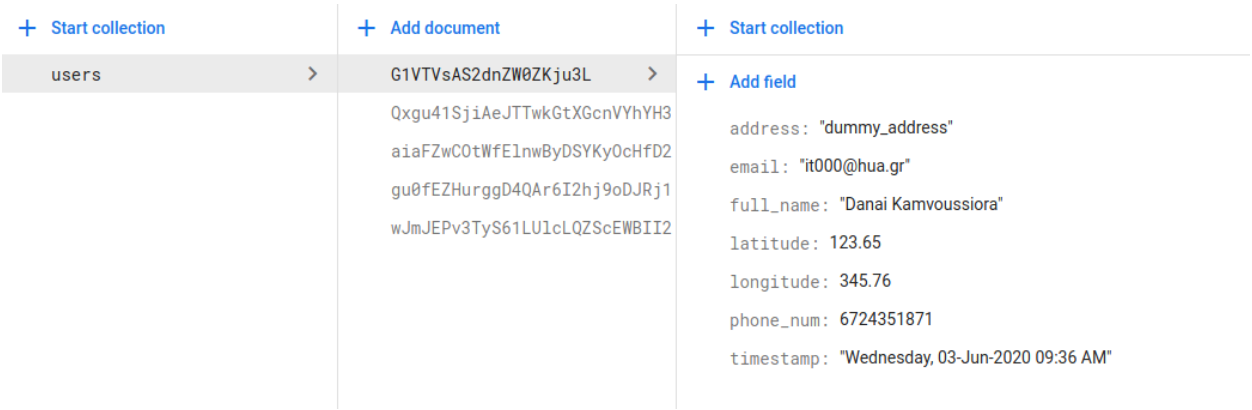
### 4.2.1 Λειτουργίες Firebase

Οι λειτουργίες του Firebase που έχουν χρησιμοποιηθεί στα πλαίσια της εργασίας είναι:

- **Authentication:** Σαν τρόπος εγγραφής και σύνδεσης έχει επιλεχθεί η συμπλήρωση email και password. Πολύ εύκολα μπορούν να προστεθούν και ως επιπλέον επιλογές η χρήση του λογαριασμού Google ή των στοιχείων του Facebook. Έχει χρησιμοποιηθεί και η λειτουργία reset password.

- **Cloud Messaging:** Έχει χρησιμοποιηθεί για να στέλνονται ειδοποιήσεις από την δεύτερη εφαρμογή στην πρώτη. Δίνεται η δυνατότητα, δηλαδή, στον χρήστη να στείλει μία ειδοποίηση στην χαμένη του συσκευή (π.χ. Αν βρέθηκε παρακαλώ επικοινωνήστε μαζί μου). Με τη χρήση αυτής της λειτουργίας, η ειδοποίηση θα εμφανιστεί στην συσκευή ακόμα και χωρίς να είναι η εφαρμογή ενεργή.

- **Database:** Για βάση δεδομένων έχει επιλεχθεί το **Firestore**. Έχει δημιουργηθεί μία συλλογή (collection) "users", που αποτελείται από πολλά έγγραφα (documents). Το κάθε έγγραφο απεικονίζει έναν χρήστη, και αποτελείται από τα απαραίτητα πεδία (fields) για την εφαρμογή μας, που φαίνονται στην παρακάτω ενδεικτική εικόνα.




Σχήμα 10: Firestore database της εφαρμογής.

- **Performance Monitoring:** Το Performance Monitoring χρησιμοποιήθηκε για να μετρηθεί το App start time των Android εφαρμογών.

### 4.2.2 Google APIs

Στα πλαίσια της εργασίας έχουν χρησιμοποιηθεί τα παρακάτω Google APIs:

- **Maps SDK for Android:** προσφέρει την χρήση χάρτη σε Android εφαρμογές.

- **Maps JavaScript API:** προσφέρει την χρήση χάρτη σε web-based εφαρμογές με χρήση JavaScript.

- **Directions API:** προσφέρει την δυνατότητα εύρεσης οδηγιών ανάμεσα σε δύο τοποθεσίες.

- **Geocoding API:** προσφέρει την δυνατότητα συσχέτισης συντεταγμένων (lat, lon) με την ονοματική διεύθυνση μιας τοποθεσίας.

- **Geolocation API:** προσφέρει πληροφορίες τοποθεσίας του χρήστη σε web εφαρμογές.

- **FusedLocationProviderClient API:** προσφέρει έναν έξυπνο συνδυασμό διαφορετικών δεδομένων για την παροχή πληροφοριών τοποθεσίας σε Android εφαρμογής.

## 4.3 Διεπαφή χρήστη | User Interface

Ένας από τους κύριους στόχους της εργασίας ήταν να είναι ευπαρουσίαστη η εφαρμογή. Έχει δοθεί έμφαση, λοιπόν, στο κομμάτι του UI.

Αρχικά, έχει χρησιμοποιηθεί Constraint Layout σε όλα τα Activities, καθώς είναι το πιο αποκριτικό (responsive) layout και προσέφερε μεγαλύτερη ευελιξία στον τρόπο τοποθέτησης των στοιχείων. Όλα τα activity layouts έχουν δημιουργηθεί και για προβολή landscape.

Για τις υπηρεσίες ρυθμίσεων του λογαριασμού έχουν χρησιμοποιηθεί Dialog Fragments, τα οποία έχουν προσαρμοστεί εμφανισιακά.

Μερικές λεπτομέριες στις οποίες αφιερώθηκε χρόνος:

- Δημιουργήθηκε ένα λογότυπο (logo), που φαίνεται στην παρακάτω εικόνα. Τα λογότυπα στις μέρες μας είναι άκρως απαραίτητα, καθώς ο χρήστης τα συσχετίζει με ένα προϊόν, μία υπηρεσία ή και μία ολόκληρη εταιρεία, και πολύ συχνά επηρεάζουν την επιλογή του.

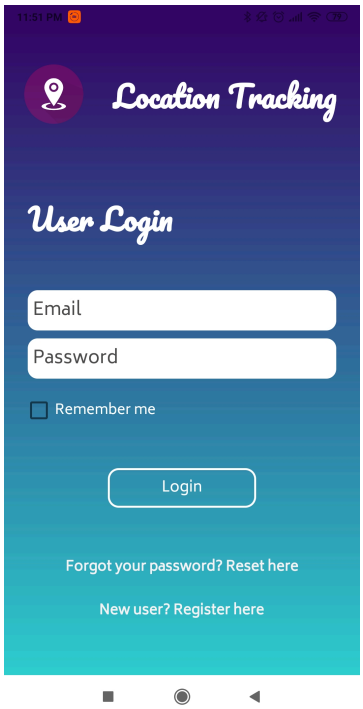
Σχήμα 11: Logo

- Δημιουργήθηκε **Splash Screen** και για τις δύο εφαρμογές με το παραπάνω λογότυπο. Το Splash Screen είναι ένα γραφικό στοιχείο που περιέχει μία εικόνα ή ένα λογότυπο (logo) και εμφανίζεται όσο η εφαρμογή ξεκινά. Ο κύριος στόχος του είναι να δείξει στον χρήστη ότι η εφαρμογή φορτώνει. Χωρίς αυτό, όσο η εφαρμογή φορτώνει, ο χρήστης βλέπει απλά μία λευκή οθόνη.

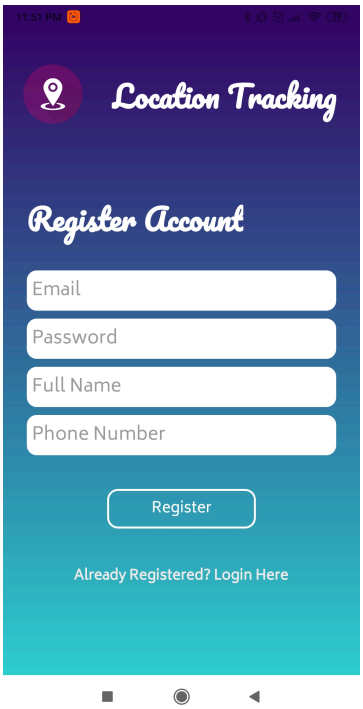
- Χρησιμοποιήθηκε ο gradient χρωματισμός που φαίνεται παραπάνω σαν φόντο σε όλα τα activities, πράγμα που προσδίδει ομοιομορφία στις εφαρμογές.

- Για εκδόσεις Android 5.0 και πάνω, δίνεται η δυνατότητα να αλλάξει το χρώμα του status bar κατά τη χρήση της εφαρμογής. Συνήθως το χρώμα του status bar ακολουθεί το θέμα που επιλέγει ο προγραμματιστής κατα την ανάπτυξη της εφαρμογής (π.χ Theme.AppCompat ή Theme.AppCompat.Light). Προτιμήθηκε όμως στην οθόνη της συσκευής να φαίνεται ενιαία η εφαρμογή και να μην διαχωρίζεται το status bar, γι'αυτό και επιλέχθηκε να χρησιμοποιηθεί χρώμα που να ταιριάζει με το φόντο της εφαρμογής.

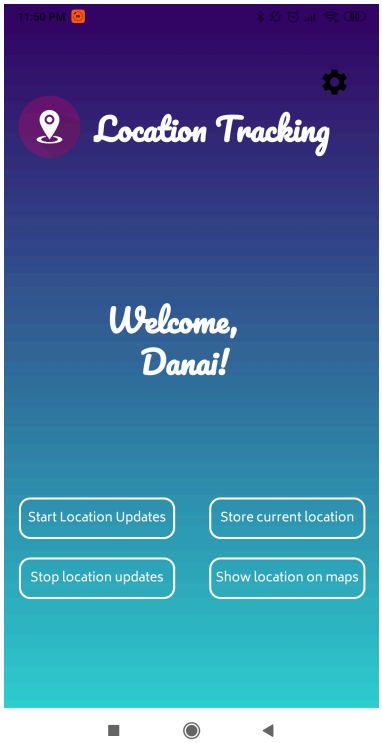
Σχήμα 12: Modified Status Bar.



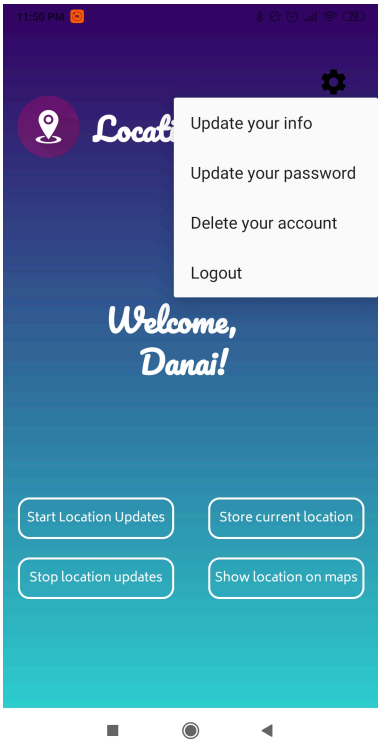

Σχήμα 13: User login (first app).



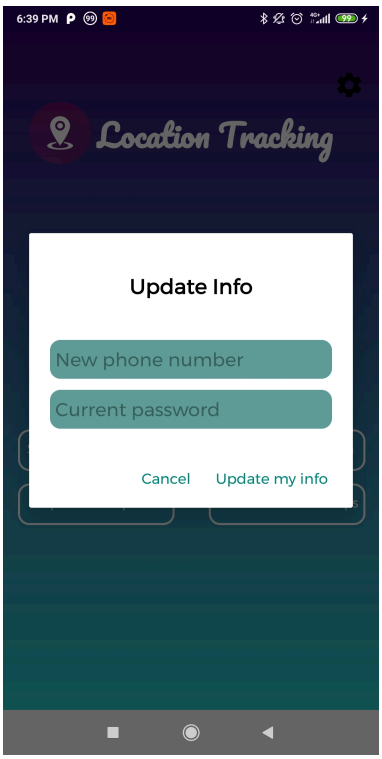

Σχήμα 14: User register (first app).




Σχήμα 15: User main page (first app).




Σχήμα 16: User settings (first app).




Σχήμα 17: Update info dialog fragment (first app).

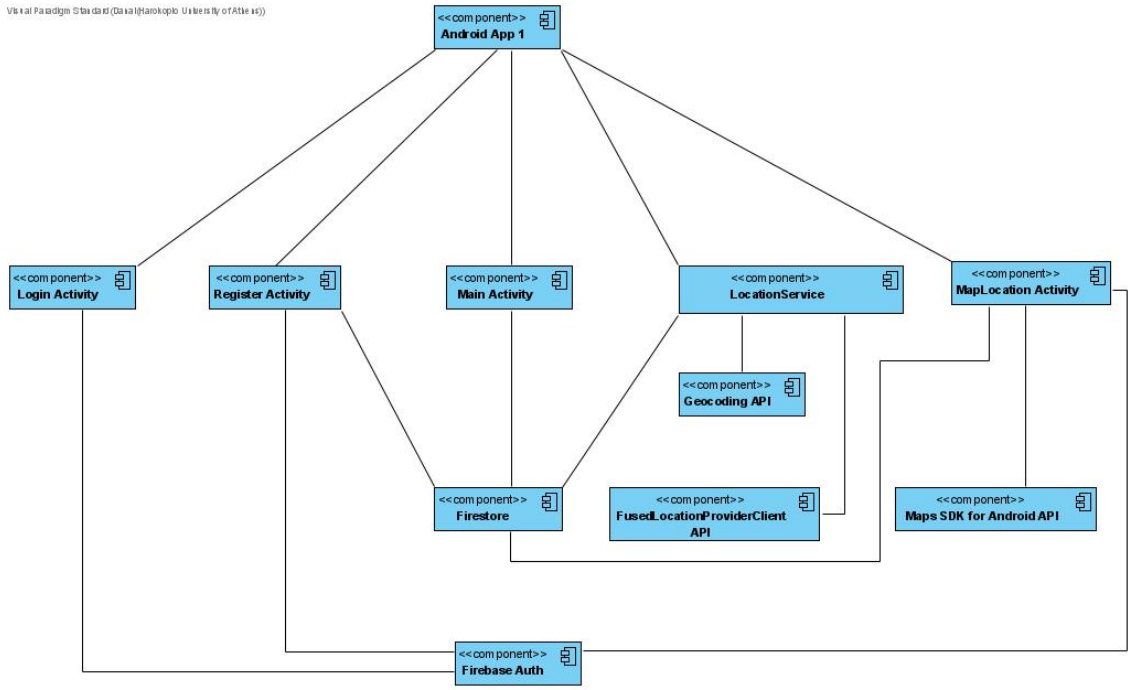
## 4.4 Πρώτη εφαρμογή Android: Καταγραφή τοποθεσίας του χρήστη

Αυτή είναι η εφαρμοφή που πρέπει να εγκαταστήσει ο χρήστης για να ξεκινήσει η διαδικασία καταγραφής της τοποθεσίας.

### 4.4.1 App Components

Έχουν δημιουργηθεί τέσσερα Activities, καθώς και ένα Service, συγκεκριμένα Foreground Service, το οποίο χρησιμοποιείται για την διαδικασία αδιάκοπης καταγραφής της τοποθεσίας. Παρατηρήθηκε πως με την χρήση απλού Background Service, το σύστημα σταματούσε αυτόματα τη λειτουργία του Service μετά από μερικά λεπτά, πράγμα που δεν έπρεπε να συμβαίνει στην περίπτωση μας.

### 4.4.2 Αρχιτεκτονική Εφαρμογής | Component Diagram




Σχήμα 18: First Android App Component Diagram.

### 4.4.3 Υπηρεσίες Λογαριασμού

Υπάρχουν όλες οι βασικές υπηρεσίες λογαριασμού που προσφέρει μία εφαρμογή, οι οποίες υλοποιήθηκαν με τη χρήση του Firebase. Ένας καινούριος χρήστης πρέπει να κάνει εγγραφή. Σε αυτή τη διαδικασία γίνονται οι απαραίτητοι έλεγχοι (π.χ συμπλήρωση των πεδίων, αρκετών χαρακτήρων στον κωδικό πρόσβασης, έλεγχος ότι συμπληρώνεται το email στη σωστή μορφή). Ένας ήδη εγγεγραμμένος χρήστης, απλά κάνει σύνδεση, και σε περίπτωση που έχει ξεχάσει τον κωδικό πρόσβασης μπορεί εύκολα να τον κάνει reset. Μπορεί επίσης να επιλέξει «να τον θυμάται» η εφαρμογή και να μην απαιτείται η σύνδεση του κάθε φορά που ανοίγει η εφαρμογή. Αφού έχει ήδη συνδεθεί ο χρήστης, μπορεί με εύκολο τρόπο να τροποποιήσει τα στοιχεία του (π.χ. καινούριος αριθμός τηλεφώνου, καινούριος κωδικός πρόσβασης), να αποσυνδεθεί από τον λογαριασμό, καθώς και να διαγράψει τελείως τον λογαριασμό του από το σύστημα. Να σημειωθεί πως για να κάνει ο χρήστης τροποποίηση στοιχείων, πρέπει να συμπληρώσει τον ισχύοντα κωδικό πρόσβασής του ώστε να γίνει re-authentication από το Firebase.

Στη διαδικασία εγγραφής του, απαιτείται να συμπλήρωθούν τα πεδία που φάνηκαν στο Σχήμα 10, ώστε να περαστούν τα απαραίτητα στοιχεία στο Cloud Firestore. Αυτά που συμπληρώνει ο ίδιος είναι το email του, έναν κωδικό πρόσβασης (τον οποίο δεν μπορούμε να δούμε εμείς στην βάση καθώς η διαχείρισή του γίνεται μέσω του authentication), το όνομά του και ο αριθμός του κινητού του. Τα πεδία address, latitude, longitude και timestamp συμπληρώνονται αυτόματα μόλις γίνει η πρώτη καταγραφή τοποθεσίας, και από εκείνη τη στιγμή, κάθε φορά που ανανεώνεται η τοποθεσία, ανανεώνονται και αυτά τα πεδία στη βάση.

### 4.4.4 Καταγραφή τοποθεσίας και εμφάνιση σε χάρτη

Όταν ο χρήστης φορτώσει την εφαρμογή για πρώτη φορά, είναι απαραίτητο να δώσει άδεια (permission) για τη χρήση της τοποθεσίας του, πράγμα που γίνεται εύκολα, καθώς η εφαρμογή του ζητάει τα απαραίτητα permissions εκείνη τη στιγμή. Γίνεται, επίσης, έλεγχος ότι έχει εγκατεστημένη την τελευταία έκδοση των Google Services και ότι έχει ανοιχτό το GPS. Σε περίπτωση που δεν πληρούνται αυτές οι προϋποθέσεις, ο χρήστης μεταφέρεται στις αντίστοιχες ρυθμίσεις για να τις ενεργοποιήσει.

Μόλις συνδεθεί ο χρήστης, έχει τη δυνατότητα να ξεκινήσει την αδιάκοπη καταγραφή τοποθεσίας, να ξεκινήσει το Foreground Service δηλαδή, το οποίο δεν σταματάει να δουλεύει μέχρι να το σταματήσει ο ίδιος, ακόμα και αν η εφαρμογή είναι στο background. Η κύρια διαφορά μεταξύ του foreground και του background service είναι πως το σύστημα Android αναγνωρίζει το foreground service ως μια διεργασία για την οποία γνωρίζει ενεργά ο χρήστης λόγω του persistent notification, και γι' αυτό δεν είναι υποψήφια να τη σκοτώσει όταν έχει χαμηλή μνήμη. Με αυτόν τον τρόπο εξασφαλίζουμε πως το service θα τρέχει αδιάκοπα μέχρι να το σταματήσει ο χρήστης.

Επιλέχθηκε ο FusedLocationProviderClient της Google αντί για τον LocationManager του Android, καθώς χρησιμοποιεί έναν έξυπνο συνδυασμό GPS, τοποθεσίας που παρέχεται από WiFi/Cell και αισθητήρες κίνησης της συσκευής. Το GPS, WiFi και Cell προσφέρουν πληροφορίες τοποθεσίας μια φορά/δευτερόλεπτο, ενώ οι αισθητήρες κίνησης (accelerator, gyrometer, magnetometer) προσφέρουν δεδομένα 100 φορές/δευτερόλεπτο. Ο συνδυασμός αυτός αποτελεί το Hybrid Bayesian Fusion [13].

Ένα χαρακτηριστικό παράδειγμα συνδυασμού αποτελεί η περίπτωση απώλειας σήματος GPS, κάτι που μπορεί να συμβεί περνώντας από ένα τούνελ ή χρησιμοποιώντας το μετρό. Ο fused location provider είναι ικανός να καλύψει το κενό που προκλήθηκε από την απώλεια σήματος GPS, όχι μόνο με χρήση Cell/WiFi όπου αυτά είναι διαθέσιμα, αλλά και με τους αισθητήρες κίνησης της συσκευής, όπως το επιταχυνσιόμετρο και το γυροσκόπιο, ώστε να προβλέψει την κίνηση της συσκευής μέχρι να αποκτήσει ξανά σήμα GPS [13]. Αυτό είναι σημαντικό για την συγκεκριμένη εφαρμογή, καθώς δεν είναι σπάνιο συσκευές να χάνονται σε συνθήκες κλειστού χώρου ή κάτω από το έδαφος.

Χρησιμοποιείται ακρίβεια PRIORITY_HIGH_ACCURACY, που παρέχει την πιο ακριβή δυνατή τοποθεσία (μεταξύ 3-10m [14]), η οποία υπολογίζεται χρησιμοποιώντας όσο το δυνατόν περισσότερες εισόδους (GPS, WiFi/Cell και ένα σύνολο αισθητήρων κίνησης της συσκευής) [15].

Το Service κάνει αυτόματη ανανέωση της τοποθεσίας κάθε ένα λεπτό με τη χρήση της μεθόδου requestLocationUpdates. Δίνεται, επίσης, στον χρήστη η δυνατότητα να κάνει και κατ'απαίτηση ανανέωση της τοποθεσίας του. Αυτό πραγματοποιείται με τη χρήση της μεθόδου getLastLocation. Στο πλαίσιο της εργασίας, κυρίως για δοκιμαστικούς λόγους, με κάθε ανανέωση της τοποθεσίας έρχεται στον χρήστη μία ειδοποίηση (notification), την οποία αν πατήσει, μεταφέρεται στον χάρτη της εφαρμογής για να δει την τοποθεσία του.

### 4.4.5 Ειδοποιήσεις | Notifications

Αξίζει να σχολιαστούν οι ειδοποιήσεις σε ξεχωριστή ενότητα καθώς έχουν μία «ιδιοτροπία» στον τρόπο γραφής του απαραίτητου κώδικα. Για εκδόσεις Android 8.0 - API 26 και πάνω, πρέπει να δημιουργηθεί ένα κανάλι ειδοποιήσεων [16] το οποίο θα περιέχει ένα Channel_ID. Αυτό το Channel_ID χρησιμοποιείται ως παράμετρος όταν χτίζεις μία ειδοποίηση, όπως φαίνεται και στο Σχήμα 19. Ανάλογα, λοιπόν, με τον τύπο ειδοποίησης, θα πρέπει να δημιουργηθούν τα κατάλληλα κανάλια. Στα πλαίσια της εργασίας χρειάστηκαν δύο κανάλια, ένα για τις δοκιμαστικές ειδοποιήσεις τοποθεσίας και ένα για την ειδοποίηση του Foreground Service. Ο κώδικας για τη δημιουργία ενός notification channel πρέπει να εκτελείται μόλις ξεκινάει η εφαρμογή.

Να σημειωθεί πως για προηγούμενες εκδόσεις Android, η παράμετρος Channel_ID αγνοείται όταν χτίζεται η ειδοποίηση, και τα notification channels δεν δημιουργούνται, όπως φαίνεται και στο Σχήμα 20, καθώς γίνεται έλεγχος της έκδοσης Android.

```
NotificationCompat.Builder builder = new NotificationCompat.Builder(this, CHANNEL_ID)
        .setSmallIcon(R.drawable.notification_icon)
        .setContentTitle(textTitle)
        .setContentText(textContent)
        .setPriority(NotificationCompat.PRIORITY_DEFAULT);
```

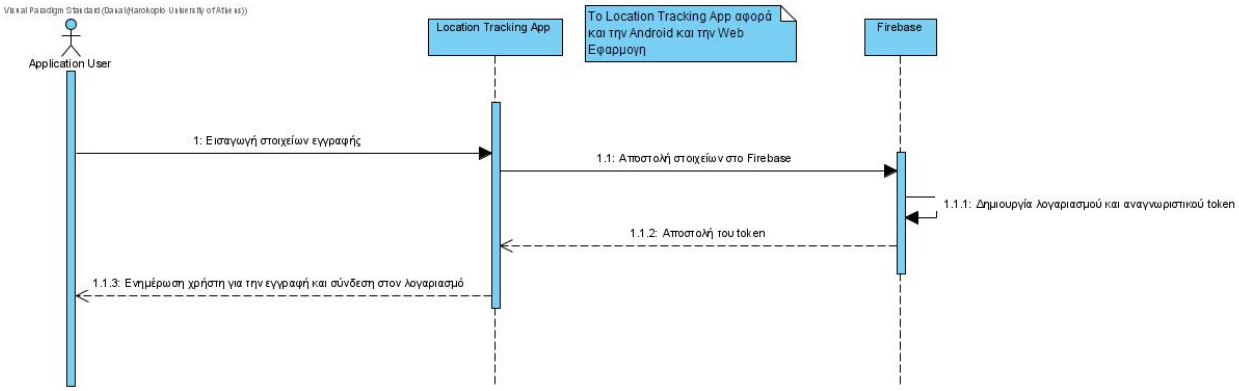
Σχήμα 19: Notification Builder.

```
private void createNotificationChannel() {
    // Create the NotificationChannel, but only on API 26+ because
    // the NotificationChannel class is new and not in the support library
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
        CharSequence name = getString(R.string.channel_name);
        String description = getString(R.string.channel_description);
        int importance = NotificationManager.IMPORTANCE_DEFAULT;
        NotificationChannel channel = new NotificationChannel(CHANNEL_ID, name, importance);
        channel.setDescription(description);
        // Register the channel with the system; you can't change the importance
        // or other notification behaviors after this
        NotificationManager notificationManager = getSystemService(NotificationManager.class);
        notificationManager.createNotificationChannel(channel);
    }
}
```

Σχήμα 20: Notification Channel.

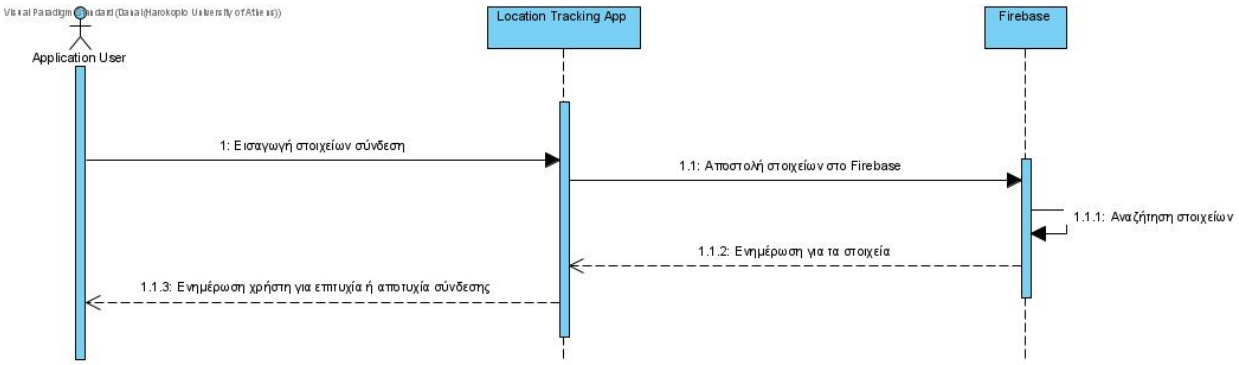
### 4.4.6 Sequence Diagrams για βασικές λειτουργίες της εφαρμογής

- **User Register Sequence Diagram**



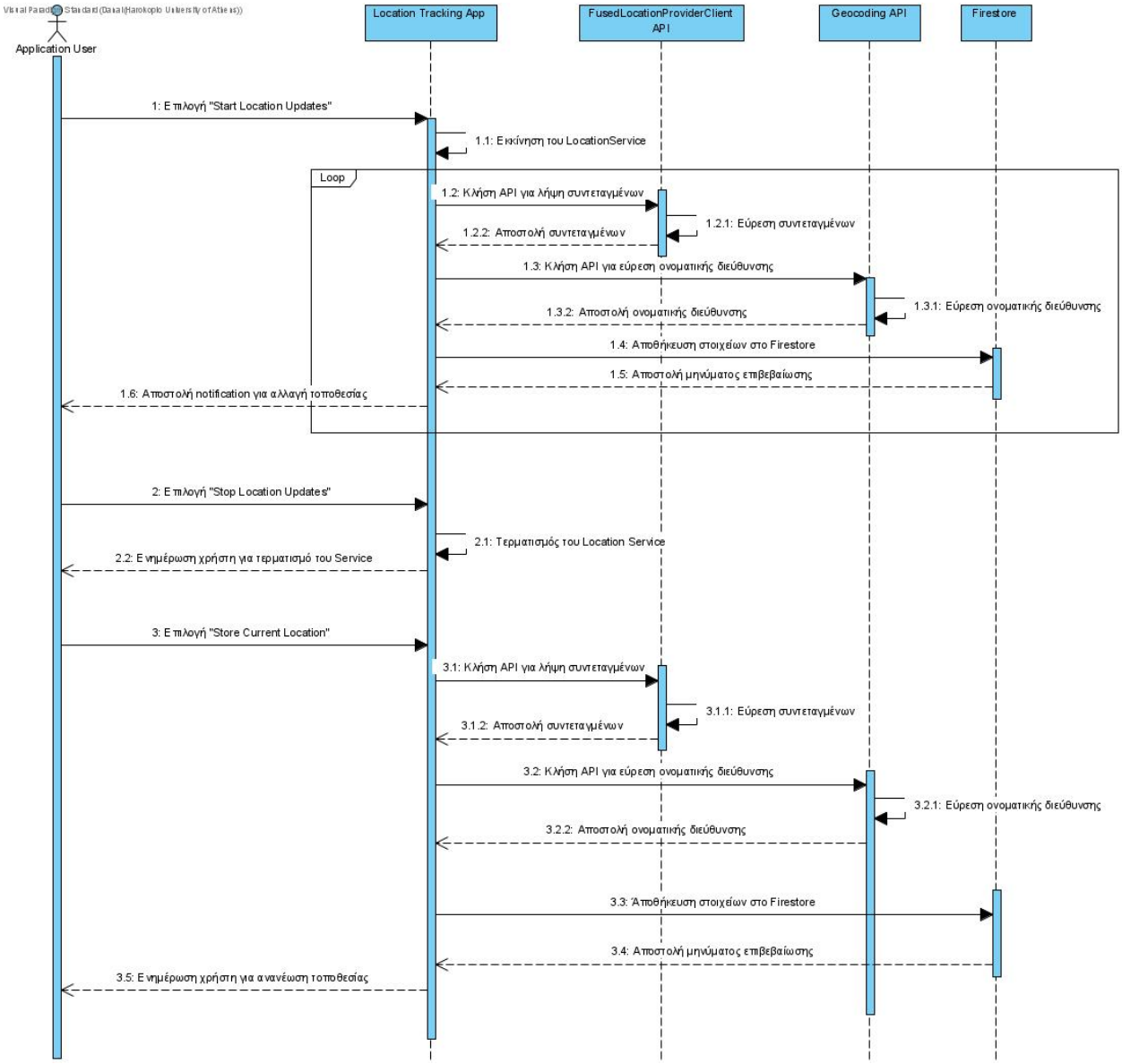

Σχήμα 21: User Register Sequence Diagram.

- **User Login Sequence Diagram**




Σχήμα 22: User Login Sequence Diagram.

- **Location Tracking Sequence Diagram**




Σχήμα 23: Location Tracking Sequence Diagram.

## 4.5 Δεύτερη εφαρμογή: Εύρεση τοποθεσίας χαμένης συσκευής | Android

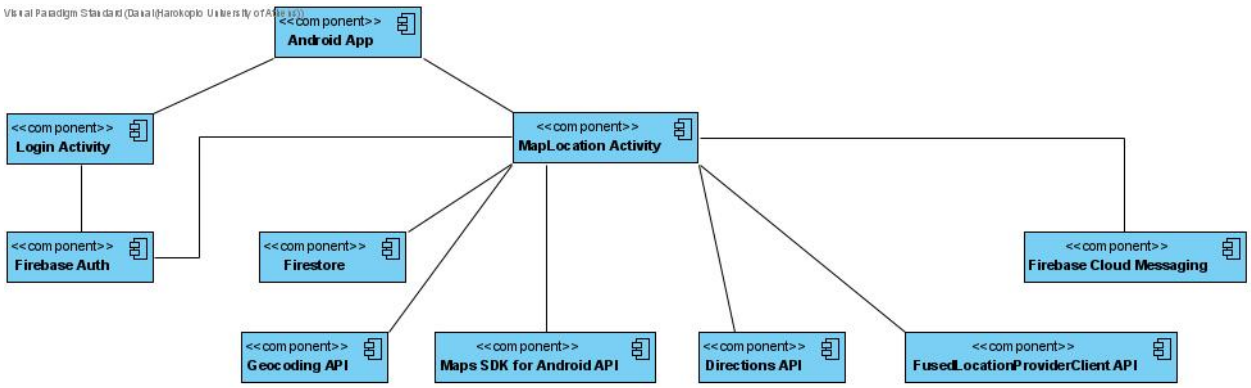
Αυτή είναι η Android εφαρμογή που πρέπει να χρησιμοποιήσει ο χρήστης για να βρει την χαμένη του συσκευή. Να σημειωθεί πως η πρώτη εφαρμογή πρέπει να είναι εγκατεστημένη στην χαμένη συσκευή, και ο χρήστης πρέπει να την έχει χρησιμοποιήσει ώστε να υπάρχουν τα στοιχεία του στο Firestore.

### 4.5.1 App Components

Η λειτουργία αυτής της εφαρμογής ήταν πιο απλή από αυτή της προηγούμενης εφαρμογής, γι'αυτό έχουν χρησιμοποιηθεί μόνο Activities, και συγκεκριμένα έχουμε δύο, το Login Activity και το MapLocation Activity. Δεν χρειαζόταν κάποιο Service, καθώς όλες οι πληροφορίες που χρειάζονταν υπήρχαν ήδη αποθηκευμένες στη βάση, και το μόνο που έπρεπε γίνει ήταν να τις πάρουμε.

### 4.5.2 Αρχιτεκτονική Εφαρμογής | Component Diagram

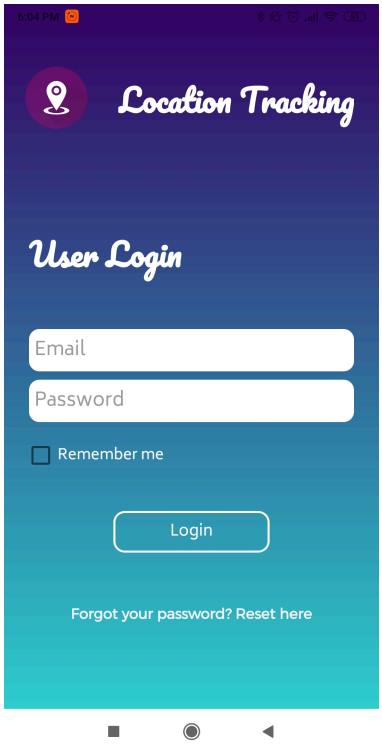
Στο παρακάτω σχήμα φαίνεται η αρχιτεκτονική της δεύτερης Android εφαρμογής.



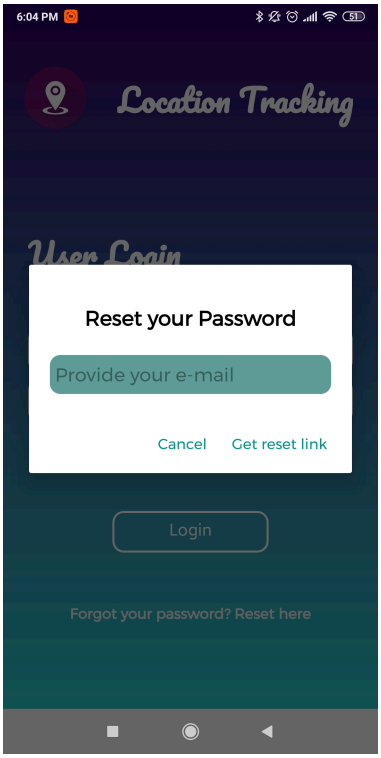

Σχήμα 24: Second Android App Component Diagram.

### 4.5.3 Υπηρεσίες Λογαριασμού

Όπως αναφέρθηκε και προηγουμένως, ο χρήστης πρέπει να υπάρχει ήδη στο σύστημα, έτσι υπάρχει μόνο η υπηρεσία σύνδεσης. Υπάρχει, επίσης, η λειτουργία reset password σε περίπτωση που ο χρήστης έχει ξεχάσει τον κωδικό πρόσβασής του.



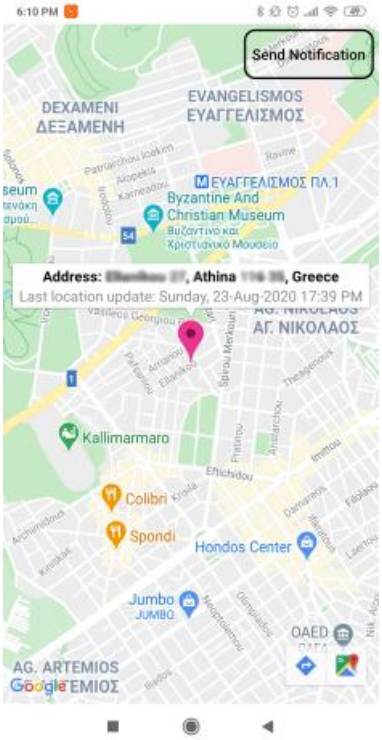

Σχήμα 25: User login (second app).




Σχήμα 26: Password reset dialog (second app).

### 4.5.4 Εύρεση τοποθεσίας και εμφάνιση σε χάρτη

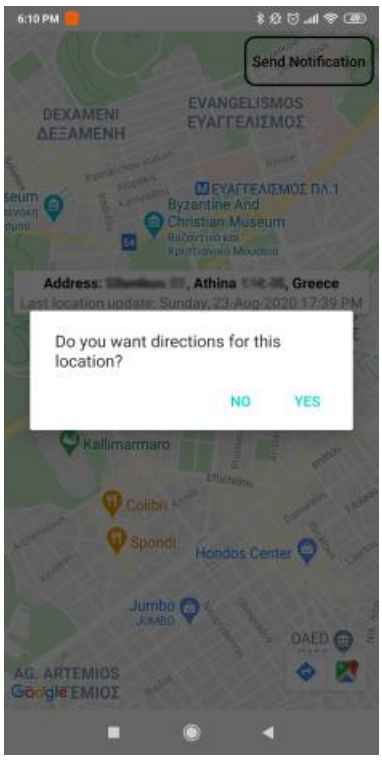
Μόλις ο χρήστης συνδεθεί, μεταφέρεται αμέσως στο Activity με τον χάρτη. Σε αυτό το Activity γίνεται η διαδικασία εύρεσης της καταγεγραμμένης τοποθεσίας από το Firestore, και η εμφάνισή της στον χάρτη. Όταν ο χρήστης πατήσει πάνω στο pin της τοποθεσίας, εμφανίζεται ένα info window στο οποίο αναγράφεται η διεύθυνση της τοποθεσίας καθώς και η ημερομηνία της καταγραφής της, δηλαδή η ημερομηνία που έγινε η τελευταία ανανέωση της τοποθεσίας. Να σημειωθεί πως η επικοινωνία με το Firestore γίνεται σε πραγματικό χρόνο, και στην περίπτωση που η τοποθεσία της πρώτης συσκευής αλλάξει ενώ ο χρήστης παρακολουθεί τον χάρτη αυτής της εφαρμογής, θα γίνει αμέσως ανανέωση στον χάρτη με τα νέα στοιχεία. Για την απεικόνιση της τοποθεσίας, έχουν χρησιμοποιηθεί οι χάρτες της Google με το *Maps SDK for Android*, όπως αναφέρθηκε και πριν.



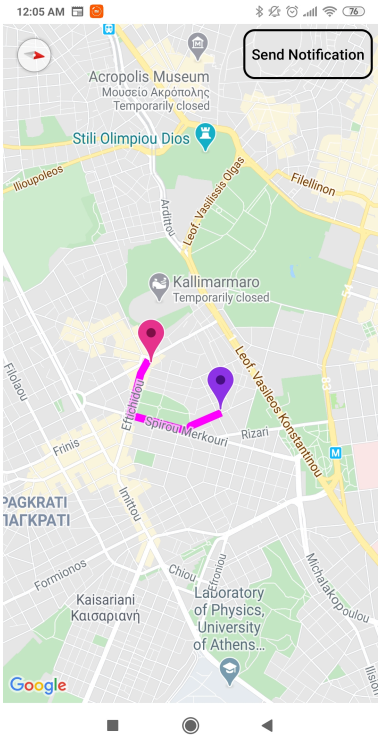

Σχήμα 27: User main page (second app)

### 4.5.5 Οδηγίες προς την τοποθεσία

Έχει υλοποιηθεί και η λειτουργία εύρεσης διαδρομής για την καταγεγραμμένη τοποθεσία. Αν ο χρήστης πατήσει το info Window των πληροφοριών, εμφανίζεται ένα alert dialog που τον ρωτάει αν επιθυμεί οδηγίες από την τωρινή του τοποθεσία προς την τοποθεσία της χαμένης συσκευής. Αρχικά, μόλις βρεθεί η τωρινή τοποθεσία του χρήστη (παλι με τη χρήση της μεθόδου getLastLocation), εμφανίζεται με το μωβ pin στον χάρτη. Η διαδικασία εύρεσης οδηγιών έχει υλοποιηθεί με το Directions API της Google, και με την χρήση μιας βοηθητικής μεθόδου DirectionsParser που φροντίζει για την αποκωδικοποίηση της απάντησης JSON που λαμβάνουμε από την κλήση του API. Τα αποτελέσματα εμφανίζονται στον χάρτη ως polylines.




Σχήμα 28: Directions question (second app).

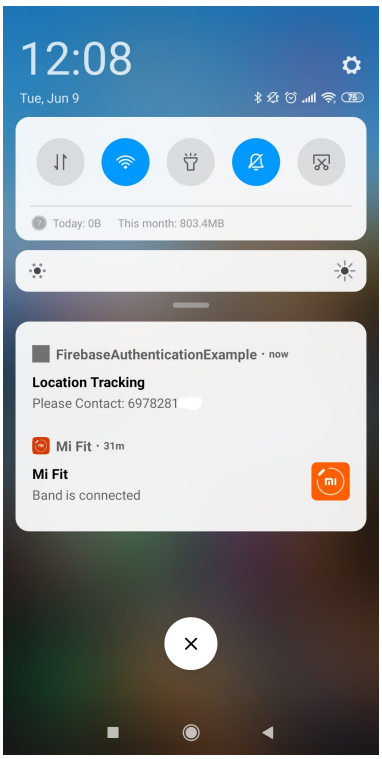



Σχήμα 29: Directions with polylines (second app).

### 4.5.6 Αποστολή ειδοποίησης στην πρώτη εφαρμογή

Δίνεται επίσης η δυνατότητα να αποστείλει ο χρήστης μία ειδοποίηση στην πρώτη εφαρμογή που είναι εγκατεστημένη στη χαμένη συσκευή. Η ειδοποίηση λέει "Please contact: phone_num", όπου phone_num ο αριθμός τηλεφώνου του χρήστη, που υπάρχει ήδη καταγεγραμμένος στη βάση μας και συμπληρώνεται αυτόματα μόλις πατηθεί το κουμπί Send Notification που φαίνεται στις παραπάνω εικόνες. Η λειτουργία αυτή έχει υλοποιηθεί με το Firebase Cloud Messaging (FCM) και τη βοήθεια της βιβλιοθήκης Volley.

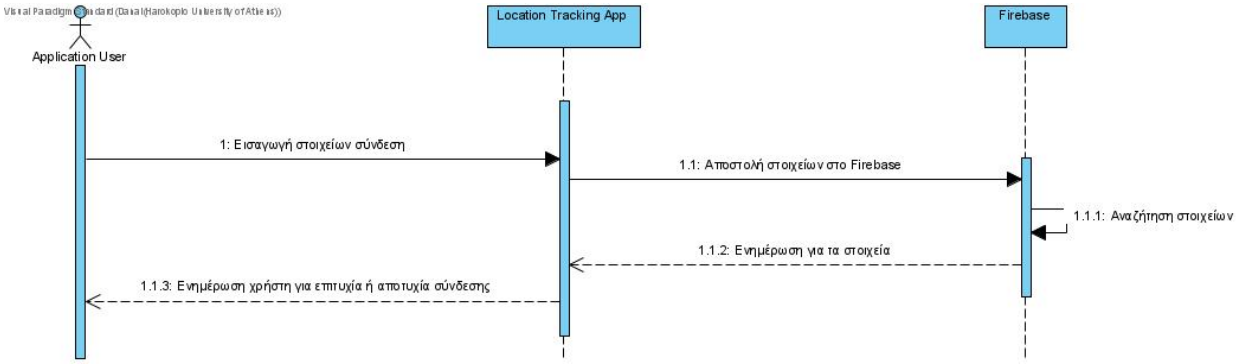
Με το FCM γίνεται η εγγραφή (subscribe) της πρώτης εφαρμογής σε ένα θέμα (topic), στο οποίο topic μετά στέλνουμε την συγκεκριμένη ειδοποίηση με την χρήση της βιβλιοθήκης Volley. Η επιλογή του topic έπρεπε να γίνει προσεκτικά, καθώς αν το topic ήταν κοινό για όλους τους χρήστες, θα λάμβαναν όλοι ειδοποιήσεις που μπορεί να μην προορίζονταν για αυτούς. Έτσι το topic δηλώνεται δυναμικά και χρησιμοποιεί το user_id (ξεχωριστό αναγνωριστικό που ανατίθεται από το Firebase την ώρα της εγγραφής) του κάθε χρήστη. Με αυτόν τον τρόπο οι ειδοποιήσεις φτάνουν μόνο στον χρήστη που πρέπει. Γίνεται κατάργηση εγγραφής (unsubscribe) από το topic μόνο όταν ο χρήστης αποσυνδεθεί από τον λογαριασμό του.




Σχήμα 30: FCM notification.

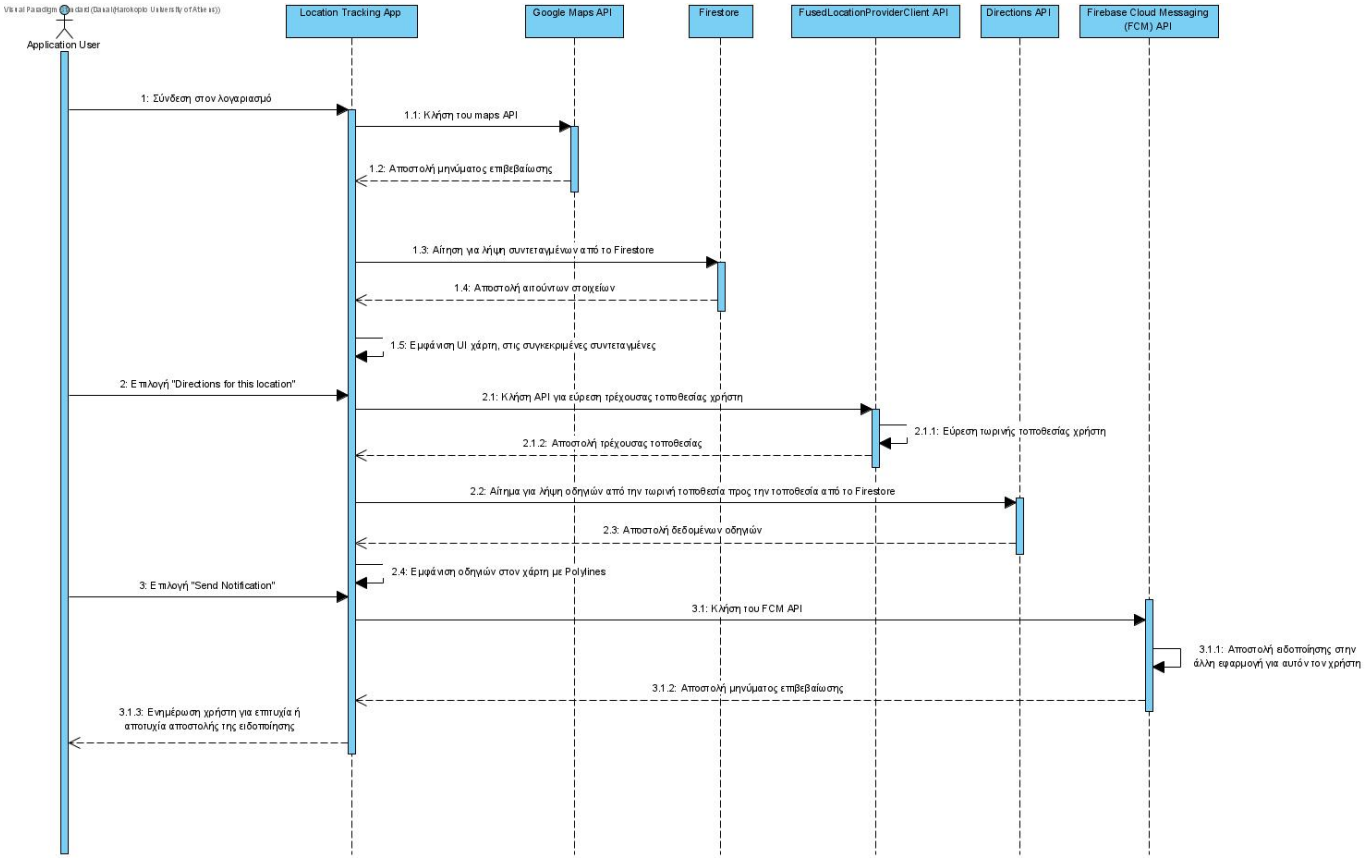
### 4.5.7 Sequence Diagrams για βασικές λειτουργίες της εφαρμογής

- **User Login Sequence Diagram**




Σχήμα 31: User Login Sequence Diagram.

- **All Features Sequence Diagram**



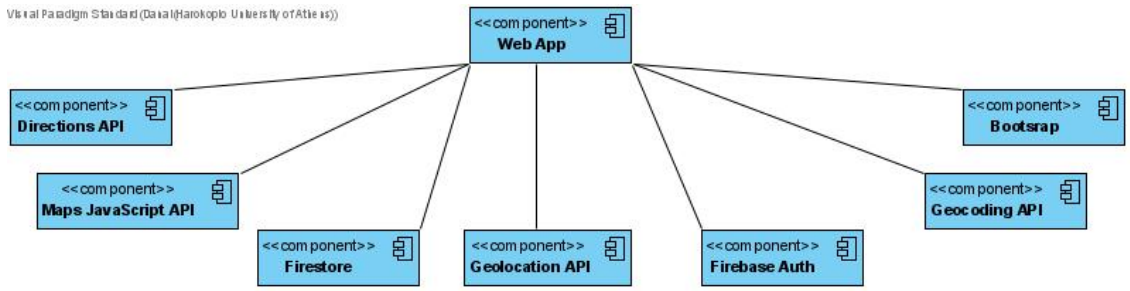

Σχήμα 32: Second Android App Sequence Diagram.

## 4.6 Τρίτη εφαρμογή: Εύρεση τοποθεσίας χαμένης συσκευής | Web

Αυτή είναι η διαδικτυακή εφαρμογή που πρέπει να χρησιμοποιήσει ο χρήστης για να βρει την χαμένη του συσκευή. Να σημειωθεί πως η πρώτη εφαρμογή πρέπει να είναι εγκατεστημένη στην χαμένη συσκευή, και ο χρήστης πρέπει να την έχει χρησιμοποιήσει ώστε να υπάρχουν τα στοιχεία του στο Firestore. Η web-based εφαρμογή παρέχει στον χρήστη ακριβώς τις ίδιες δυνατότητες με την Android εκδοχή της, και έχει ακολουθηθεί το ίδιο θέμα εμφάνισης με τις Android εφαρμογές. Χρησιμοποιούνται και εδώ οι υπηρεσίες του Firebase, σε JavaScript. Το front-end έχει υλοποιηθεί κυρίως με το Bootstrap.

Η εφαρμογή αποτελείται από μία μόνο html σελίδα, και ανάλογα με το στάδιο στο οποίο βρίσκεται ο χρήστης εμφανίζονται συγκεκριμένα σημεία της.

### 4.6.1 Αρχιτεκτονική Εφαρμογής | Component Diagram




Σχήμα 33: Web App Component Diagram.

### 4.6.2 Υπηρεσίες Λογαριασμού

Όπως αναφέρθηκε και προηγουμένως, ο χρήστης πρέπει να υπάρχει ήδη στο σύστημα, έτσι υπάρχει μόνο η υπηρεσία σύνδεσης. Υπάρχει επίσης η λειτουργία reset password σε περίπτωση που ο χρήστης έχει ξεχάσει τον κωδικό πρόσβασής του.

### 4.6.3 Εύρεση τοποθεσίας και εμφάνιση σε χάρτη

Μόλις ο χρήστης συνδεθεί, γίνεται η διαδικασία εύρεσης της καταγεγραμμένης τοποθεσίας από το Firestore, και η εμφάνισή της στον χάρτη. Όταν ο χρήστης πατήσει πάνω στο pin της τοποθεσίας, εμφανίζεται σε ένα info window η διεύθυνση της τοποθεσίας, καθώς και η ημερομηνία της καταγραφής της, δηλαδή η ημερομηνία που έγινε η τελευταία ανανέωση της τοποθεσίας στη βάση. Να σημειωθεί πως η επικοινωνία με το Firestore γίνεται σε πραγματικό χρόνο, και στην περίπτωση που η τοποθεσία της πρώτης συσκευής αλλάξει ενώ ο χρήστης παρακολουθεί τον χάρτη, θα γίνει αμέσως ανανέωση στον χάρτη με τα νέα στοιχεία.

### 4.6.4 Αποστολή ειδοποίησης στην πρώτη εφαρμογή

Για την αποστολή ειδοποίησης στην πρώτη εφαρμογή χρησιμοποιείται το Firebase Cloud Messaging μαζί με το node.js. Για την εγγραφή σε topic ακολουθείται η ίδια πρακτική που ακολουθήθηκε και στην Android εφαρμογή, που αναλύθηκε προηγουμένως. Για να γίνει η αποστολή της ειδοποίησης, εκτελείται ένα post request στον Node server. Στο σώμα του request περιλαμβάνεται το μήνυμα και το id του χρήστη. Με το Firebase-admin SDK επιβεβαιώνουμε την ταυτότητα του χρήστη, και αν η αναγνώριση είναι επιτυχής, δημοσιεύεται το μήνυμα (push notification) και ενημερώνεται ο χρήστης από το front-end για τα αποτελέσματα.

```
app.post('/', function(req, res) {

  var notification_message = req.body.message;
  var id = req.body.id;
```

Σχήμα 34: Web app message post request (code snippet).

```
admin.auth().verifyIdToken(id)
```

Σχήμα 35: Web app user ID verification (code snippet).

```
var message = {
  "notification": {
    "body": notification_message,
    "title": "FCM Web app message"
  },
  'topic': topic
};
```

Σχήμα 36: Web app message (code snippet).

```
// Send a message to devices subscribed to the provided topic.
admin.messaging().send(message)
  .then((response) => {
    // Response is a message ID string.
    console.log('Successfully sent message:', response);
    res.send({status:200,
    'message':`Your notification has been successfully sent to topic: ${message.topic}`});
  })
  .catch((error) => {
    res.send({status:404,
    'message':`Your message wasn't able to be published.Please try later.`})
    console.log('Error sending message:', error);
  });
```

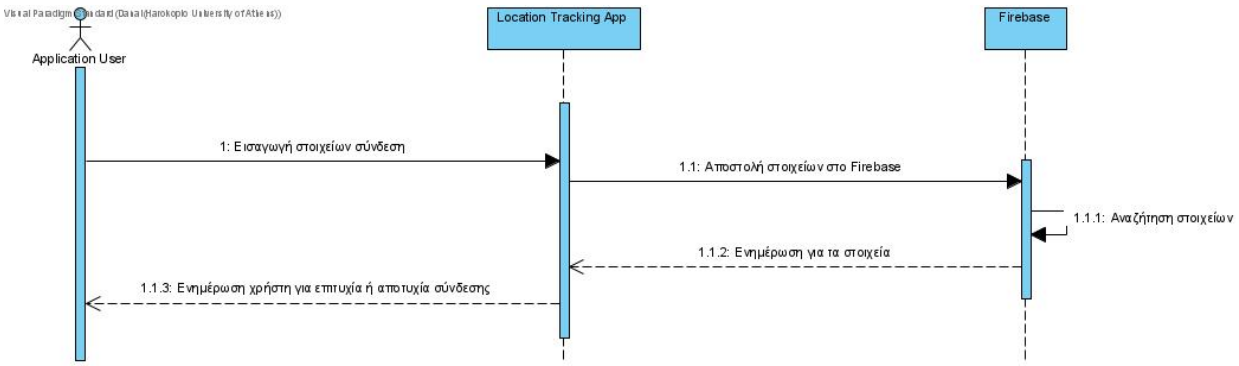
Σχήμα 37: Web app notification front-end (code snippet).

### 4.6.5 Οδηγίες προς την τοποθεσία

Έχει υλοποιηθεί και η λειτουργία εύρεσης διαδρομής για την καταγεγραμμένη τοποθεσία. Μόλις ο χρήστης πατήσει το αντίστοιχο κουμπί, βρίσκεται η τρέχουσα τοποθεσία του με τη χρήση του Geolocation API, και ανανεώνεται ο χάρτης ώστε να φαίνονται πλέον και οι δύο τοποθεσίες με pins. Έχει παρατηρηθεί πως ανάλογα με τον browser που χρησιμοποιεί ο χρήστης, η τωρινή τοποθεσία που επιστρέφεται μπορεί να διαφέρει. Η διαδικασία εύρεσης οδηγιών έχει υλοποιηθεί με το Directions API της Google. Με τη χρήση του Geocoding API βρίσκεται το όνομα της τωρινής τοποθεσίας του χρήστη (reverse geocoding) ώστε να εμφανίζεται σωστά στο pin. Τα αποτελέσματα εμφανίζονται στον χάρτη ως polylines.

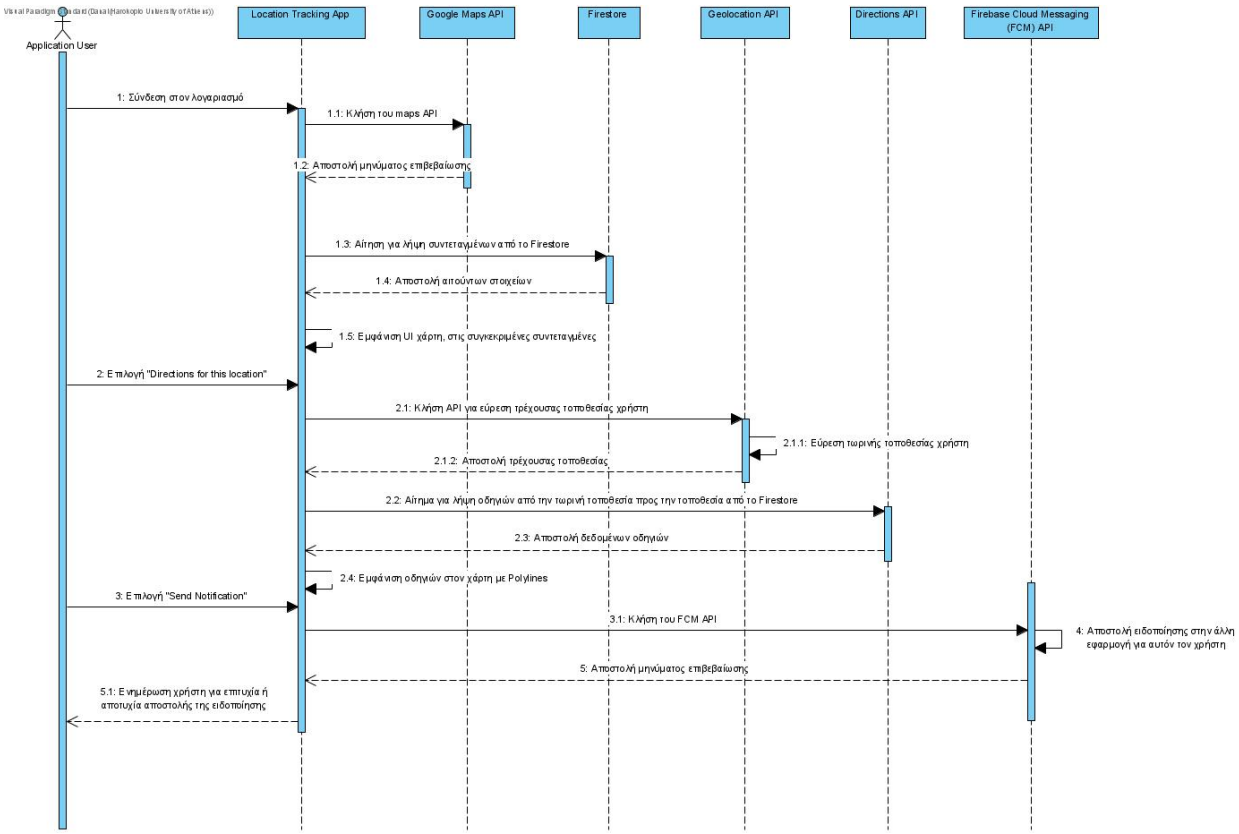
### 4.6.6 Sequence Diagrams για βασικές λειτουργίες της εφαρμογής

- **User Login Sequence Diagram**




Σχήμα 38: User Login Sequence Diagram.

- **Web App Sequence Diagram**

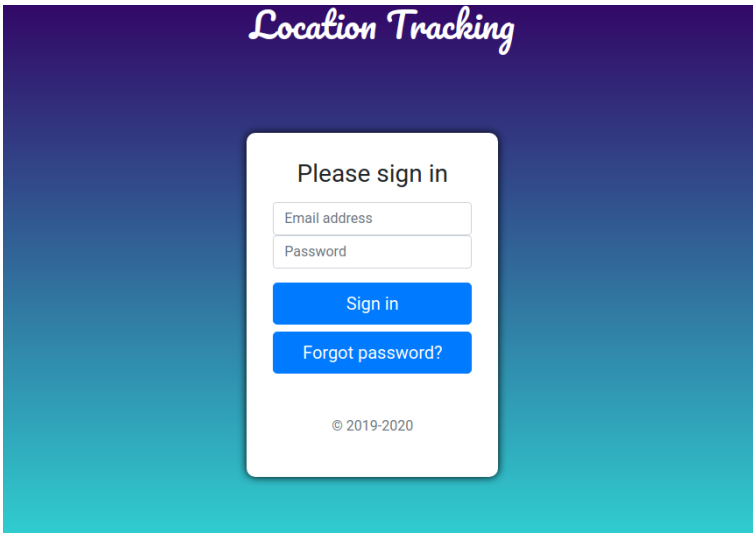



Σχήμα 39: Web App Sequence Diagram.

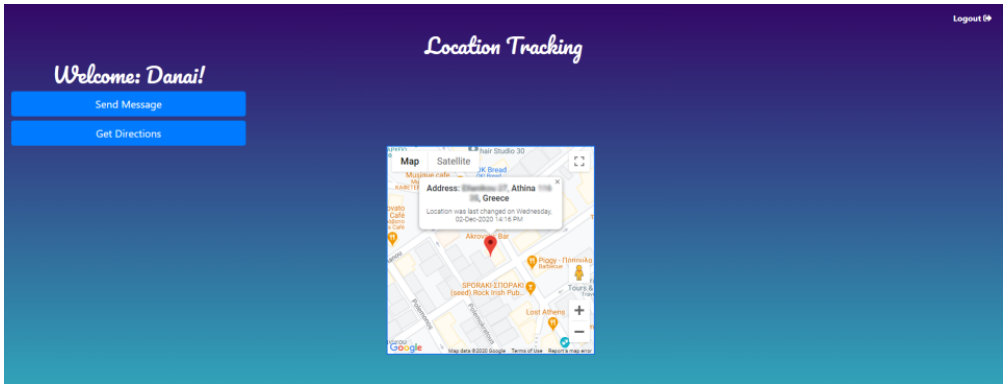
### 4.6.7 Δημοσίευση διαδικτυακής πλατφόρμας

Η web εφαρμογή δοκιμάστηκε για αρχή σε localhost και μετά έγινε deployed στον Heroku server. Είναι διαθέσιμη εδώ.

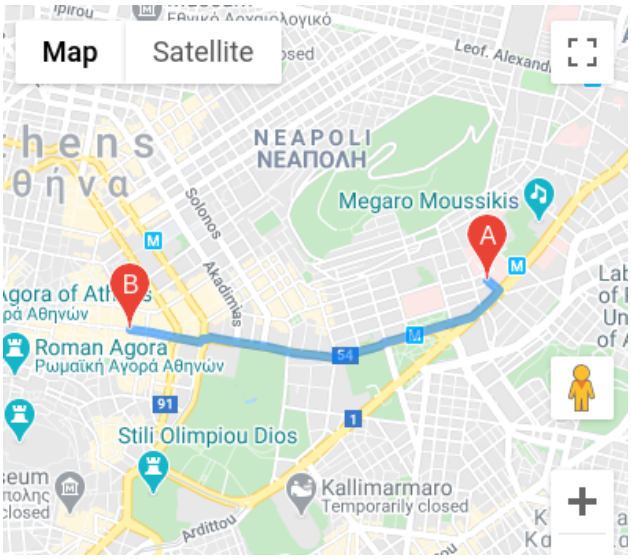
### 4.6.8 Εικόνες web εφαρμογής



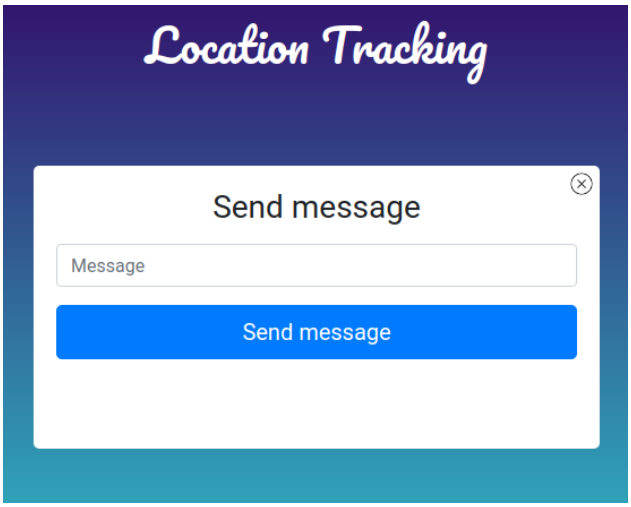

Σχήμα 40: Web app login.




Σχήμα 41: Web app main page.




Σχήμα 42: Web app directions.



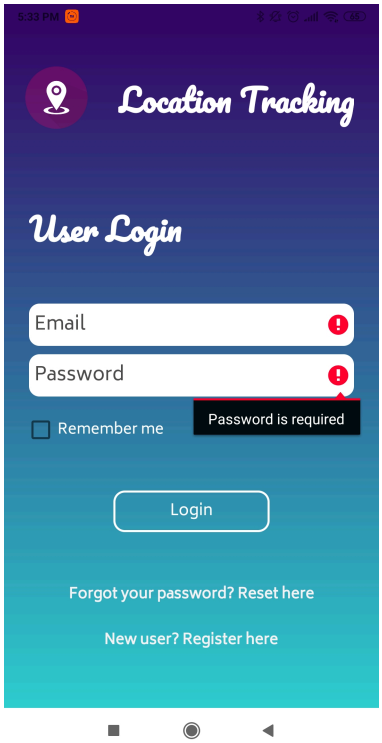

Σχήμα 43: Web app message.

# 5 Πειράματα και Αξιολόγηση

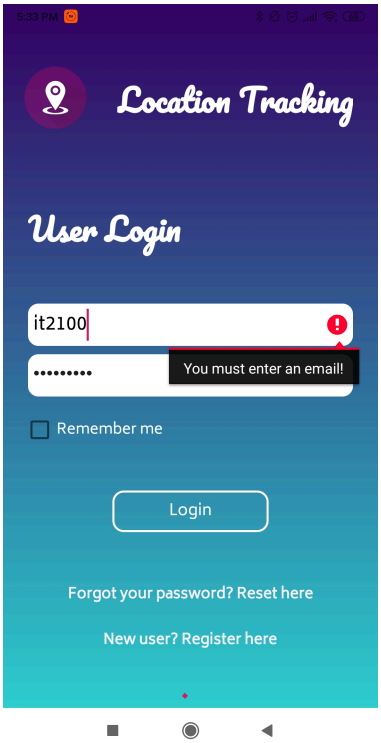
## 5.1 Παρατηρήσεις

Για την αξιολόγηση βασικός στόχος ήταν να επιβεβαιωθεί πως οι εφαρμογές θα μπορούν να χρησιμοποιηθούν με ευκολία από τον χρήστη, πως οι λειτουργίες που παρέχονται δουλεύουν σωστά και πως προσφέρεται στον χρήστη το προσδοκώμενο αποτέλεσμα.

Γι' αυτόν τον λόγο δοκιμάστηκαν σενάρια στα οποία μπορεί να έκανε λάθος ο χρήστης (π.χ. ότι συμπληρώνει κανονικό email, ότι το password τηρεί κάποιες προϋποθέσεις, ότι συμπληρώνονται σωστά τα απαραίτητα πεδία στις φόρμες) και έγιναν στον κώδικα οι απαραίτητοι έλεγχοι ώστε να αποτραπούν οι λάθος ενέργειες και να ενημερώνεται ο χρήστης για τη σωστή συμπλήρωση στοιχείων. Ενδεικτικά σημεία ελέγχου:




Σχήμα 44: Fields check.




Σχήμα 45: Email check.

Στην εύρεση της τωρινής τοποθεσίας του χρήστη στην web εφαρμογή, έχει παρατηρηθεί πως η τοποθεσία που μας επιστρέφεται από το Geolocation API δεν είναι ακριβής, καθώς εντοπίζεται μέσω του IP Address, και όχι με τη χρήση GPS, άρα έχουμε ακρίβεια ενός-δύο τετραγώνων. Έτσι, για να είναι πιο ακριβείς οι οδηγίες που επιστρέφονται από την εφαρμογή, ο χρήστης μπορεί να θέσει χειροκίνητα την ακριβή τοποθεσία του στον browser που χρησιμοποιεί, ακολουθώντας μερικά απλά βήματα από αυτό το άρθρο της βιβλιογραφίας [17].

## 5.2 Android Apps Performance Monitoring & Testing

Στα πλαίσια της παρούσας πτυχιακής, έγιναν μερικές μετρήσεις για την αξιολόγησή της ως προς το performance. [18]

### 5.2.1 Συσκευές που επιλέχθηκαν

Οι συσκευές αυτές επιλέχθηκαν καθώς είχα πρόσβαση σε αυτές ως φυσικές συσκευές οπότε τις επέλεξα για πιο ρεαλιστικά αποτελέσματα σε σχέση με emulators. Ο παρακάτω πίνακας παρέχει μία επισκόπηση των συσκευών. Οι περισσότερες εφαρμογές πρέπει να έχουν αποδεκτή επίδοση τόσο σε συσκευές υψηλών προδιαγραφών όσο και σε συσκευές χαμηλών προδιαγραφών. Ως εκ τούτου, επιλέχθηκε μία συσκευή υψηλών και μία συσκευή χαμηλών επιδόσεων για τις δοκιμές επίδοσης.

| | **High-end** | **Low-end** |
|---|---|---|
| Device | Xiaomi Mi 9T | Xiaomi Redmi Note 4 |
| Operating System | Android 10.0 API 29 | Android 7.1.1 API 25 |
| CPU | Octa-core 2.2 GHz | Octa-core 2.0 GHz |
| RAM Memory | 6GB | 3GB |
| Battery Capacity | 4000 mAh | 4100 mAh |
| Antutu (v7) score | 211915 | 75984 |

Πίνακας 2: Device Selection

### 5.2.2 Κριτήρια Αξιολόγησης

Αυτή η ενότητα περιλαμβάνει και καθορίζει τις μετρημένες παραμέτρους, μαζί με τη λογική πίσω από την επιλογή τους. Κάθε παράμετρος βασίζεται στο παραπάνω άρθρο της βιβλιογραφίας.

Οι χρόνοι απόκρισης (*response times*) αποτελούν σημαντικό παράγοντα στην εμπειρία χρήστη. Πιο συγκεκριμένα, στα πλαίσια της εργασίας, μετριέται ο χρόνος απόκρισης για την έναρξη της εφαρμογής (app start time), δηλαδή ο χρόνος που απαιτείται για να ξεκινήσει πλήρως η εφαρμογή (από το πάτημα του εικονιδίου της εφαρμογής, έως την εμφάνιση της πρώτης οθόνης της εφαρμογής).

Η χρήση της CPU (*CPU usage*) είναι το ποσοστό της συνολικής χωρητικότητας CPU της συσκευής που χρησιμοποιείται από την εφαρμογή, σε ένα καθορισμένο χρονικό διάστημα. Οι εφαρμογές που χρησιμοποιούν μεγάλο ποσοστό της CPU, ενδέχεται να επηρεάσουν αρνητικά άλλες διαδικασίες που εκτελούνται στη συσκευή, μειώνοντας την εμπειρία χρήστη. Για την αξιολόγησή μας, μετριέται η χρήση της CPU στις παρακάτω περιπτώσεις:

- Κατά την έναρξη της εφαρμογής (και στις δύο εφαρμογές)

- Αμέσως μετά την έναρξη της εφαρμογής (και στις δύο εφαρμογές)
- Κατά τη διάρκεια που βρίσκεται αδρανής στο background (και στις δύο εφαρμογές)
- Όσο φορτώνει το mapActivity (και στις δύο εφαρμογές)
- Κατά την περίοδο στην οποία η εφαρμογή βρίσκεται στο background όσο τρέχει το Location Service (πρώτη εφαρμογή)
- Όσο βρίσκεται η τωρινή τοποθεσία του χρήστη (πρώτη εφαρμογή)
- Όσο βρίσκονται οι οδηγίες για την τοποθεσία της συσκευής (δεύτερη εφαρμογή)
- Όσο γίνεται αποστολή notification στην συσκευή (δεύτερη εφαρμογή)

Η χρήση μνήμης (*memory usage*) υποδεικνύει την ποσότητα μνήμης RAM που καταλαμβάνεται από την εφαρμογή. Αυτή η παράμετρος μετριέται σε διαφορετικά σημεία στον κύκλο ζωής της εφαρμογής, και συγκεκριμένα στις περιπτώσεις που αναφέρθηκαν και στη μέτρηση της χρήσης της CPU.

Όσον αφορά τον αποθηκευτικό χώρο (*disk space*), μετριέται ο χώρος που καταλαμβάνεται από την εγκατεστημένη εφαρμογή στη συσκευή.

Η κατανάλωση μπαταρίας (*battery consumption*) είναι σημαντική σε όλες τις κινητές συσκευές καθώς οι χρήστες δεν θέλουν οι εφαρμογές να εξαντλήσουν τη μπαταρία των συσκευών τους. Στα πλαίσια της εργασίας επιλέχθηκε να μετρηθεί το ποσοστό της μπαταρίας που καταναλώνει η πρώτη εφαρμογή κατά την εύρεση της τοποθεσίας με το Location Service, και στις δύο συσκευές.

### 5.2.3 Μέθοδοι και εργαλεία αξιολόγησης

Ο παρακάτω πίνακας παρέχει μία επισκόπηση των παραμέτρων που επιλέχθηκαν για την αξιολόγηση, καθώς και των εργαλείων που χρησιμοποιήθηκαν για αυτές, με βάση το παραπάνω άρθρο της βιβλιογραφίας.

| Performance parameters | Measurement Tools |
|---|---|
| App start time | Firebase Performance Monitoring |
| Memory usage | Android Profiler |
| CPU usage | Android Profiler |
| Disk space | Visible on device |
| Battery consumption | Visible on device |

Πίνακας 3: Measurement Tools

Ο χρόνος εκκίνησης της εφαρμογής (*app start time*) μετριέται με τη χρήση του Firebase Performance Monitoring. Τα αποτελέσματα τα παίρνουμε μέσω του Firebase Console, και έχουμε τη δυνατότητα να τα δούμε ξεχωριστά για κάθε συσκευή στην οποία τρέχει η εφαρμογή μας.

Ο Android Profiler έχει επιλεχθεί για να τη μέτρηση της χρήσης CPU και μνήμης, καθώς παρέχει μία ξεκάθαρη εικόνα για τους πόρους που καταναλώνει η εφαρμογή σε κάθε συσκευή. Τα εργαλεία του Android Profiler παρέχουν δεδομένα σε πραγματικό χρόνο.

### 5.2.4 Αποτελέσματα μετρήσεων

Οι μετρήσεις έγιναν αρκετές φορές και στους παρακάτω πίνακες βρίσκονται οι μέσοι όροι των αποτελεσμάτων για λόγους σαφήνειας.

1. **App start time**
   Στον παρακάτω πίνακα βρίσκονται τα αποτελέσματα των χρόνων εκκίνησης των εφαρμογών, για τις δύο συσκευές που επιλέχθηκαν.

| Device | App1 Start Time | App2 Start Time |
|---|---|---|
| Xiaomi Redmi Note 4 | 318 | 331 |
| Xiaomi Mi 9T | 275 | 227 |

Πίνακας 4: App start time (in milliseconds)

   Οι χρόνοι απόκρισης των εφαρμογών κατά την εκκίνησή τους, δεν έχουν μεγάλη απόκλιση ανάμεσα στις δύο συσκευές. Όπως είναι αναμενόμενο, στην low-end συσκευή παρατηρείται μία μικρή επιβάρυνση στον χρόνο εκκίνησης της εφαρμογής, χωρίς όμως να επηρεάζει αρνητικά την εμπειρία του χρήστη. Αξίζει να σημειωθεί πως αναφερόμαστε στο cold start των εφαρμογών, δηλαδή ξεκινάνε για πρώτη φορά από την εκκίνηση της συσκευής. Οι χρόνοι είναι αποδεκτοί και για τις δύο συσκευές, καθώς δεν ξεπερνούν τα 500ms [19] .

2. **Memory usage**
   Στους παρακάτω πίνακες βρίσκονται τα αποτελέσματα της χρήσης μνήμης, για κάθε μία περίπτωση που έχει αναφερθεί.

| Device | App1 | App2 |
|---|---|---|
| Xiaomi Redmi Note 4 | 98.3 | 102.5 |
| Xiaomi Mi 9T | 97.7 | 104.2 |

Πίνακας 5: Memory Usage during launch (in MB)

| Device | App1 | App2 |
|---|---|---|
| Xiaomi Redmi Note 4 | 81.7 | 83.7 |
| Xiaomi Mi 9T | 76.9 | 79.9 |

Πίνακας 6: Memory Usage right after launch (in MB)

| Device | App1 | App2 |
|---|---|---|
| Xiaomi Redmi Note 4 | 77.1 | 87.2 |
| Xiaomi Mi 9T | 76.9 | 89.3 |

Πίνακας 7: Memory Usage while idle in background (in MB)

| Device | App1 | App2 |
|---|---|---|
| Xiaomi Redmi Note 4 | 152.4 | 155.6 |
| Xiaomi Mi 9T | 145.7 | 132.4 |

Πίνακας 8: Memory Usage while map Activity loads (in MB)

| Device | App1 |
|---|---|
| Xiaomi Redmi Note 4 | 49.2 |
| Xiaomi Mi 9T | 82.4 |

Πίνακας 9: Memory Usage in background while collecting location updates (in MB)

| Device | App1 |
|---|---|
| Xiaomi Redmi Note 4 | 51.3 |
| Xiaomi Mi 9T | 90.6 |

Πίνακας 10: Memory Usage while tracking current user location (in MB)

| Device | App2 |
|---|---|
| Xiaomi Redmi Note 4 | 143.3 |
| Xiaomi Mi 9T | 148.1 |

Πίνακας 11: Memory Usage while directions are calculated (in MB)

| Device | App2 |
|---|---|
| Xiaomi Redmi Note 4 | 144.2 |
| Xiaomi Mi 9T | 149.6 |

Πίνακας 12: Memory Usage notification is sent (in MB)

Παρατηρούνται σχετικά μικρές αποκλίσεις ανάμεσα στις δύο συσκευές. Αξίζει να σημειωθεί πως η high end συσκευή δεσμεύει περισσότερους πόρους μνήμης RAM καθώς η ελεύθερη μνήμη είναι σε αφθονία σε σχέση με την low end συσκευή. Αυτό γίνεται ώστε να φορτώνονται λιγότερες πληροφορίες από την εσωτερική μνήμη της συσκευής, προσφέροντας έτσι ταχύτερη απόκριση και καλύτερη εμπειρία χρήσης.

3. **CPU usage**
   Στους παρακάτω πίνακες βρίσκονται τα αποτελέσματα της χρήσης CPU, για κάθε μία περίπτωση που έχει αναφερθεί.

| Device | App1 | App2 |
|---|---|---|
| Xiaomi Redmi Note 4 | 36.3 | 38.2 |
| Xiaomi Mi 9T | 34.2 | 38.1 |

Πίνακας 13: CPU Usage during launch (in % of total cpu)

| Device | App1 | App2 |
|---|---|---|
| Xiaomi Redmi Note 4 | 9.2 | 14.5 |
| Xiaomi Mi 9T | 7.3 | 13.2 |

Πίνακας 14: CPU Usage right after launch (in % of total cpu)

| Device | App1 | App2 |
|---|---|---|
| Xiaomi Redmi Note 4 | 0 | 0 |
| Xiaomi Mi 9T | 0 | 0 |

Πίνακας 15: CPU Usage while idle in background (in % of total cpu)

| Device | App1 | App2 |
|---|---|---|
| Xiaomi Redmi Note 4 | 35.2 | 37.3 |
| Xiaomi Mi 9T | 37.1 | 38.2 |

Πίνακας 16: CPU Usage while map Activity loads (in % of total cpu)

| Device | App1 |
|---|---|
| Xiaomi Redmi Note 4 | 6.2 |
| Xiaomi Mi 9T | 5.8 |

Πίνακας 17: CPU Usage in background while collecting location updates (in % of total cpu)

| Device | App1 |
|---|---|
| Xiaomi Redmi Note 4 | 12.3 |
| Xiaomi Mi 9T | 8.5 |

Πίνακας 18: CPU Usage while tracking current user location (in % of total cpu)

| Device | App2 |
|---|---|
| Xiaomi Redmi Note 4 | 17.6 |
| Xiaomi Mi 9T | 22.2 |

Πίνακας 19: CPU Usage while directions are calculated (in % of total cpu)

| Device | App2 |
|---|---|
| Xiaomi Redmi Note 4 | 5.2 |
| Xiaomi Mi 9T | 6.2 |

Πίνακας 20: CPU Usage while notification is sent (in % of total cpu)

Και στη χρήση της CPU παρατηρούνται ελάχιστες αποκλίσεις στο ποσοστό χρήσης CPU μεταξύ των δύο συσκευών. Στη low-end συσκευή, ορισμένες διεργασίες απαιτούν ελαφρώς μεγαλύτερη χρήση, όπως η καταγραφή της τοποθεσίας. Όπως είναι λογικό, όταν η εφαρμογή είναι στο προσκήνιο και όσο περιμένει το interval για να καταγράψει την τοποθεσία, η χρήση είναι 0% και στις δύο συσκευές.

4. **Disk space**

   Στον παρακάτω πίνακα φαίνεται ο χώρος που καταλαμβάνουν οι εφαρμογές στις 2 συσκευές που επιλέχθηκαν.

| Device | App1 | App2 |
|---|---|---|
| Xiaomi Redmi Note 4 | 21.04 | 23.6 |
| Xiaomi Mi 9T | 7.47 | 7.45 |

Πίνακας 21: Disk Space (in MB)

Εδώ παρατηρείται διαφορά στο μέγεθος της εφαρμογής ανάμεσα στις 2 συσκευές. Αυτό συμβαίνει καθώς σε νεότερες εκδόσεις Android κάποια components και βιβλιοθήκες της εφαρμογής είναι ήδη εγκατεστημένα, ενώ σε παλιότερες εκδόσεις που δεν υπήρχαν, έπρεπε να εγκατασταθούν και αυτά.

5. **Battery consumption**
   Στους παρακάτω πίνακες φαίνεται η κατανάλωση της μπαταρίας με τη χρήση της πρώτης εφαρμογής κατά την εύρεση της τοποθεσίας, στις δύο συσκευές. Επιλέχθηκε να γίνει μόνο στην πρώτη εφαρμογή το συγκεκριμένο τεστ, καθώς εδώ έχουμε τη λειτουργία της αδιάκοπης εύρεσης της τοποθεσίας με τη χρήση Service. Να σημειωθεί πως το τεστ αυτό έγινε σε διάστημα δύο ωρών, και κάτω από τις ίδιες συνθήκες και για τις δύο συσκευές. Η εφαρμογή τοποθετήθηκε στο προσκήνιο μόλις ξεκίνησε το Service, και η οθόνη των συσκευών παρέμεινε κλειστή, ώστε να μην επηρεαστεί το ποσοστό της μπαταρίας που καταναλώθηκε.

| **Device** | **App1** |
|---|---|
| Xiaomi Redmi Note 4 | 2.04 |
| Xiaomi Mi 9T | 1.08 |

Πίνακας 22: Battery Consumption (in % of battery usage)

| **Device** | **App1** |
|---|---|
| Xiaomi Redmi Note 4 | 86.11 |
| Xiaomi Mi 9T | 61.14 |

Πίνακας 23: Battery Consumption (in mAh)

Όπως φαίνεται στους πίνακες, δεν υπάρχει μεγάλη απόκλιση στην κατανάλωση της μπαταρίας μεταξύ των δύο συσκευών. Και στις δύο συσκευές η κατανάλωση είναι αρκετά μικρή, ώστε η 24ωρη χρήση της εφαρμογής να μην αποτελεί πρόβλημα στην αυτονομία της συσκευής του χρήστη.

### 5.2.5 Συμπεράσματα μετρήσεων

Με το τέλος των μετρήσεων επίδοσης των εφαρμογών, καταλήγουμε στο συμπέρασμα πως και οι δύο εφαρμογές λειτουργούν αποδοτικά και σε high-end αλλά και σε low-end συσκευές, χωρίς να επηρεάζεται αρνητικά η επίδοσή τους από τους πόρους των συσκευών, και χωρίς αυτές αντίστοιχα να επιβαρύνουν τις συσκευές και την εμπειρία του χρήστη.

# 6 Συμπεράσματα

## 6.1 Σύνοψη

Στα πλαίσια της παρούσας πτυχιακής εργασίας δημιουργήθηκαν συνολικά τρεις εφαρμογές, δύο για κινητές συσκευές Android και μία web. Με την πρώτη εφαρμογή προσφέρεται στον χρήστη η λειτουργία της καταγραφής της τοποθεσίας της συσκευής του. Με τις δύο επόμενες εφαρμογές παρέχεται στον χρήστη η δυνατότητα να βρεί την τελευταία καταγεγραμμένη τοποθεσία της συσκευής του σε περίπτωση που χαθεί/κλαπεί.

Πέρα από τις λειτουργίες των εφαρμογών, δόθηκε έμφαση στο κομμάτι του UI του χρήστη και ακολουθήθηκε εννιαίος σχεδιασμός σε όλες τις εφαρμογές για ομοιομορφία. Ένας κύριος στόχος ήταν οι εφαρμογές να είναι φιλικές προς τον χρήστη και εύχρηστες, κάτι που επιτεύχθηκε.

Τέλος, υλοποιήθηκε performance testing στις δύο Android εφαρμογές ώστε να συγκριθούν οι επιδόσεις τους σε διάφορες συσκευές. Τα αποτελέσματα των μετρήσεων δείχνουν πως οι εφαρμογές λειτουργούν αποδοτικά ανεξαρτήτως των πόρων της συσκευής του χρήστη.

## 6.2 Μελλοντικοί Στόχοι

Μετά το τέλος της παρούσας πτυχιακής, σχεδιάζεται η υλοποίηση των Android εφαρμογών και για iOS συσκευές.

Επίσης, σχεδιάζεται και η προσθήκη επιπλέον λειτουργιών στην πρώτη εφαρμογη (όπως η λειτουργία anti-theft, ιστορικό τοποθεσιών), καθώς και η αναπροσαρμογή του UI των εφαρμογών και για tablets.

Κάτι επιπλέον που θα μελετηθεί, είναι το Flutter, δηλαδή το σετ εργαλείων διεπαφής χρήστη της Google, για την κατασκευή εγγενώς μεταγλωττισμένων εφαρμογών από μία βάση κώδικα. Ουσιαστικά παρέχει τη δυνατότητα να αναπτυχθούν δύο διαφορετικές εφαρμογές (για iOS και Android) σε μία γλώσσα προγραμματισμού και μία βάση κώδικα. Η γλώσσα προγραμματισμού που χρησιμοποιείται στο Flutter είναι η Dart.

# Αναφορές


[1] S. O'Dea, ``Number of smartphone users worldwide from 2016 to 2021," tech. rep., 2020.

[2] K. McLaughlin, ``The fourth amendment and cell phone location tracking: Where are we," *Hastings Comm. & Ent. LJ*, 2006.

[3] J. Alabaster, ``Android founder: We aimed to make a camera os," *PCWorld. Retrieved*, 2013.

[4] R. Sandberg and M. Rollins, ``The business of android apps development," 2013.

[5] J. Callaham, ``The history of android os: its name, origin and more," *Android Authority*, 2018.

[6] M. Wilson, ``T-mobile g1: Full details of the htc dream android phone," *Gizmodo-We come from the future*, 2008.

[7] S. G. Stats, ``Mobile operating system market share worldwide," *Dostopno prek* https://bit.ly/2AlUJ1N, 2020.

[8] ``Platform architecture." , published: https://cutt.ly/akgfSvt, 2020.

[9] ``Application fundamentals." , published: https://cutt.ly/ukggqqu, 2020.

[10] ``Services overview." , published: https://cutt.ly/4kggPlc, 2020.

[11] ``Intents and intent filters." , published: https://cutt.ly/DkghumO, 2020.

[12] D. Stevenson, ``What is firebase? the complete story, abridged.." , published: https://bit.ly/2MhJryd, 2018.

[13] ``Seamless and smooth location everywhere with the new fusedlocationprovider (google i/o'19)." , published: https://cutt.ly/Aka0xJq, 2019.

[14] ``Precise location using fused api." , published: https://cutt.ly/PkxvDzl, 2016.

[15] ``Optimize location for battery." , published: https://cutt.ly/ykgpLT0, 2020.

[16] ``Create and manage notification channels." , published: https://cutt.ly/VkgapCo, 2020.

[17] ``How to change your location in chrome and firefox." , published: https://bit.ly/37zuIbZ, 2020.

[18] M. Willocx, J. Vossaert, and V. Naessens, ``A quantitative assessment of performance in mobile app development tools," in *2015 IEEE International Conference on Mobile Services*, pp. 454--461, IEEE, 2015.

[19] ``App startup time." , published: https://bit.ly/38XqTiZ, 2020.