



**Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Προηγμένα Συστήματα Πληροφορικής – Ανάπτυξη
Λογισμικού και Τεχνητής Νοημοσύνης»**

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ ANDROID ΜΕ ΑΝΤΙΚΕΙΜΕΝΟ ΤΗΝ ΚΑΤΑΧΩΡΗΣΗ ΚΑΙ ΑΝΑΖΗΤΗΣΗ ΧΑΜΕΝΩΝ Ή ΑΝΕΥΡΕΘΕΝΤΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ ANDROID APPLICATION DEVELOPMENT FOR REGISTRATION AND SEARCHING OF LOST OR FOUND OBJECTS
Όνοματεπώνυμο Φοιτητή	ΜΑΡΙΝΑΚΗΣ ΕΛΕΥΘΕΡΙΟΣ
Πατρώνυμο	ΘΩΜΑΣ
Αριθμός Μητρώου	ΜΠΣΠ-21031
Επιβλέπων Καθηγητής	ΑΛΕΠΗΣ ΕΥΘΥΜΙΟΣ

**Ημερομηνία Παράδοσης
Μάιος 2023**

Τριμελής Εξεταστική Επιτροπή

Ευθύμιος Αλέπης
Αναπληρωτής Καθηγητής

Μαρία Βίρβου
Καθηγήτρια

Κωνσταντίνος Πατσάκης
Αναπληρωτής Καθηγητής

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΡΟΛΟΓΟΣ	5
ΚΕΦΑΛΑΙΟ 1^ο: ΕΙΣΑΓΩΓΗ	6
1.1 ΘΕΜΑ ΕΡΓΑΣΙΑΣ	6
1.2 ΠΑΡΑΔΟΤΕΑ	6
ΚΕΦΑΛΑΙΟ 2^ο: ΠΡΟΒΛΗΜΑ ΚΑΙ ΛΥΣΗ	7
1.1 ΠΡΟΒΛΗΜΑ	7
1.2 ΛΥΣΗ	7
ΚΕΦΑΛΑΙΟ 3^ο: ΣΧΕΤΙΚΑ ΜΕ ΤΟ ANDROID	9
1.1 ΙΣΤΟΡΙΑ	9
1.2 ΤΕΧΝΟΛΟΓΙΑ	9
1.2.1 Components Εφαρμογής	9
1.2.2 Κύκλος Ζωής Activity	10
1.2.3 Κύκλος Ζωής Fragment	13
1.2.4 Layouts	15
1.2.5 Views	15
1.3 ΔΗΜΟΤΙΚΟΤΗΤΑ	16
1.4 ΤΟ ANDROID STUDIO ΚΑΙ Η ΑΝΑΠΤΥΞΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	18
ΚΕΦΑΛΑΙΟ 4^ο: ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ	19
1.1 ΣΚΟΠΟΣ ΚΑΙ ΧΡΗΣΙΜΟΤΗΤΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	19
1.2 ΑΠΑΙΤΗΣΕΙΣ ΛΟΓΙΣΜΙΚΟΥ	19
ΚΕΦΑΛΑΙΟ 5^ο: ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΑΛΛΑ ΕΡΓΑΛΕΙΑ	21
1.1 ΕΠΙΛΟΓΗ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ	21
1.2 ΣΧΕΔΙΑΣΜΟΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ	21
1.3 ΚΑΝΟΝΕΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ	26
1.4 ΑΠΟΘΗΚΕΥΤΙΚΟΣ ΧΩΡΟΣ ΜΕΣΩΝ (CLOUD STORAGE)	29
1.5 CLOUD ΣΥΝΑΡΤΗΣΕΙΣ (CLOUD FUNCTIONS)	30
1.6 CLOUD ΕΙΔΟΠΟΙΗΣΕΙΣ (FIREBASE CLOUD MESSAGING)	31
1.7 ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ (MACHINE LEARNING)	31
ΚΕΦΑΛΑΙΟ 6^ο: ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ ANDROID	33
1.1 ΑΡΧΙΤΕΚΤΟΝΙΚΗ	33
1.2 ΟΠΤΙΚΗ ΣΧΕΔΙΑΣΗ (FRONT-END)	34
1.3 ΛΕΙΤΟΥΡΓΙΚΗ ΣΧΕΔΙΑΣΗ (BACK-END)	47
1.3.1 Σύνδεση Βάσης Δεδομένων	47
1.3.2 Σύνδεση-Εγγραφή Χρήστη	47
1.3.3 Καταχώρηση και Προβολή Αντικειμένων	48
1.3.4 Αναζήτηση και Προβολή Αντικειμένων	48
1.3.5 Έξυπνη Αναζήτηση Αντικειμένων	49
1.3.6 Προσωπικά Μηνύματα Μεταξύ Χρηστών	49
1.3.7 Firebase Cloud Functions	50
25.1.1 Δομή	51
25.2 ΔΙΑΓΡΑΜΜΑ ΠΙΘΑΝΩΝ ΔΙΑΔΡΟΜΩΝ (ΣΥΝΔΕΔΕΜΕΝΟΥ ΧΡΗΣΤΗ)	52
ΚΕΦΑΛΑΙΟ 7^ο: ΣΥΜΠΕΡΑΣΜΑΤΑ	53
ΒΙΒΛΙΟΓΡΑΦΙΑ	54

ΠΡΟΛΟΓΟΣ

Με την όλο αυξανόμενη ανάπτυξη της τεχνολογίας, πολλές καθημερινές μας ανάγκες έχουν ενταχθεί στον ψηφιακό κόσμο με σκοπό την διευκόλυνσή μας. Παράλληλα την τελευταία δεκαετία έχει εκτοξευθεί ο αριθμός χρηστών που χρησιμοποιούν την έξυπνη συσκευή κινητού τηλεφώνου (smartphone) καθώς αυτή πλέον μπορεί να διατελέσει πολλαπλές λειτουργικότητες οι οποίες παλαιότερα γίνονταν από διαφορετικές συσκευές. Ο συνδυασμός των δύο παραπάνω γεγονότων, οδήγησε σταδιακά στην ανάπτυξη κάθε είδους εφαρμογής για τα κινητά τηλέφωνα θέτοντας τα έξυπνα κινητά τηλέφωνα εκτός ανταγωνισμού από άλλες συσκευές. Επίσης, ένας ακόμα παράγοντας που εκτόξευσε την συγκεκριμένη τεχνολογία είναι η ευκολία των προγραμματιστών, κυρίως σε Android εφαρμογές, να δημιουργούν τις δικές τους εφαρμογές και να τις μοιράζονται με τους υπόλοιπους χρήστες.

ABSTRACT

With the ever-increasing development of technology, many of our daily needs have joined the digital world for our convenience. At the same time, in the last decade, the number of users using the smartphone has skyrocketed, as it can now carry out multiple functions that were previously done by different devices. The combination of the above two facts gradually led to the development of all kinds of applications for mobile phones putting smart mobile phones out of competition from other devices. Also, another factor that helped this technology to distinguish is the ease of developers, mainly in Android applications, to create their own applications and share them with other users.

ΚΕΦΑΛΑΙΟ 1^ο: ΕΙΣΑΓΩΓΗ

1.1 Θέμα Εργασίας

Η παρούσα εργασία αποτελεί το έντυπο παραδοτέο μέρος της μεταπτυχιακής διατριβής με θέμα την ανάπτυξη εφαρμογής για καταχωρήσεις και αναζητήσεις χαμένων ή ανευρεθέντων αντικειμένων. Η εφαρμογή θα αναπτυχθεί για συσκευές που υποστηρίζουν το λειτουργικό σύστημα Android από την έκδοση 24 ή νεότερη (Έκδοση Android New York Cheesecake 7.0 22 Αυγούστου 2016)¹. Στη συνέχεια θα αναπτυχθούν και θα επεξηγηθούν βασικές έννοιες τόσο του συγκεκριμένου θέματος όσο και της τεχνολογίας που θα αναπτυχθεί η εφαρμογή. Έπειτα, θα πραγματοποιηθεί ανάλυση της διαδικασίας υλοποίησης των εφαρμογών, του τρόπου λειτουργίας αυτών και τελικά θα παραχθούν ορισμένα συμπεράσματα για τον σκοπό και την σημαντικότητα της υλοποίησης της εφαρμογής.

1.2 Παραδοτέα

Στην ενότητα αυτή θα αναφερθούν επιγραμματικά τα παραδοτέα της τελικής διατριβής. Με τον όρο παραδοτέα εννοούμε τα άυλα προϊόντα που παράχθηκαν με σκοπό την δημιουργία της εφαρμογής. Τα παραδοτέα της συγκεκριμένης εργασίας είναι:

1. Το παρόν έντυπο κείμενο της Μεταπτυχιακής διατριβής, το οποίο περιλαμβάνει την επισκόπηση της σχετικής βιβλιογραφίας, τα συμπεράσματα που προέκυψαν από τη μελέτη των αποτελεσμάτων και την ανάλυση του σχεδιασμού και της υλοποίησης του συνοδευτικού λογισμικού.
2. Το λογισμικό της android εφαρμογής που υλοποιήθηκε μέσω του εργαλείου Android Studio
3. Το λογισμικό της εφαρμογής που υλοποιήθηκε μέσω του εργαλείου Visual Studio Code για την εκτέλεση των Cloud συναρτήσεων
4. Ένα οπτικοακουστικό αρχείο τύπου mp4 με όνομα «Παρουσίαση Μεταπτυχιακής Διατριβής (Android)» διάρκειας 31 λεπτών και 48 δευτερολέπτων στο οποίο παρουσιάζονται όλες οι δυνατότητες της Android εφαρμογής.
5. Ένα οπτικοακουστικό αρχείο τύπου mp4 με όνομα «Παρουσίαση Μεταπτυχιακής Διατριβής (Firebase)» διάρκειας 15 λεπτών και 32 δευτερολέπτων στο οποίο παρουσιάζεται το λογισμικό και οι κανόνες της Firebase.

1 Wikipedia. (n.d.). *Android version history*. Ανάκτηση από Wikipedia: https://en.wikipedia.org/wiki/Android_version_history

ΚΕΦΑΛΑΙΟ 2^ο: ΠΡΟΒΛΗΜΑ ΚΑΙ ΛΥΣΗ

1.1 Πρόβλημα

Οι περισσότεροι άνθρωποι κατά τη διάρκεια της ζωής τους βρίσκονται σε θέση ως ιδιοκτήτες πολλών πραγμάτων έμψυχων ή άψυχων, μεγάλων ή μικρότερων διαστάσεων. Ειδικά τον τελευταίο αιώνα με την ανάπτυξη της τεχνολογίας, τα αντικείμενα που φέρει πάνω του ένας άνθρωπος είναι περισσότερα σε σχέση με παλαιότερες εποχές. Εμπειρικά και με βάση ορισμένα άρθρα, είναι αρκετά μικρή η πιθανότητα κατά τη διάρκεια της ζωής ενός ανθρώπου ο ίδιος να μην χάσει τουλάχιστον ένα αντικείμενο. Μάλιστα, από σχετικές έρευνες που έχουν διεξαχθεί, ένας μέσος άνθρωπος χάνει περισσότερα από 3000 αντικείμενα κατά τη διάρκεια της ζωής του.²

Ορισμένες φορές η απώλεια ενός αντικειμένου μπορεί να μην επιφέρει σημαντικές συνέπειες, για παράδειγμα η απώλεια μιας ομπρέλας κατά τη διάρκεια του καλοκαιριού. Ωστόσο, στις περισσότερες περιπτώσεις όταν χάνεται ένα αντικείμενο, θα είναι ικανής αξίας. Η αξία ενός αντικειμένου μπορεί να μην καθορίζεται μόνο από το ποσό που κοστίζει το ίδιο το αντικείμενο αλλά και την συναισθηματική του αξία. Ακόμη, η αξία ενός χαρτοφύλακα για παράδειγμα μπορεί να μην είναι μεγάλη αλλά το περιεχόμενό του να έχει αποκτηθεί ή δημιουργηθεί με αρκετό κόπο ή αρκετές εργατοώρες.

1.2 Λύση

Αξίζει να σημειωθεί ότι κάποια μόνιμη λύση στο παραπάνω πρόβλημα δεν υπάρχει. Πάντοτε θα χάνονται αντικείμενα καθώς ένας άνθρωπος δεν μπορεί να είναι συνέχεια συγκεντρωμένος στο να μην χάσει κάθε τι που έχει στην κατοχή του. Ενδέχεται από κάποια ασχολία που θα προκύψει, να αφαιρεθεί το μυαλό του από το αντικείμενο που δεν επιθυμεί να χάσει. Αυτό που μπορεί να λεχθεί ως λύση είναι η ικανότητα εύρεσης των χαμένων αντικειμένων.

Ορισμένες εταιρείες έχουν δημιουργήσει έξυπνες συσκευές οι οποίες συνήθως είναι μικρών διαστάσεων και λειτουργούν ως GPS trackers (συσκευές παρακολούθησης γεωγραφικού στίγματος). Με τέτοιου είδους συσκευές είναι εύκολο κάποιος να εντοπίσει το στίγμα τους οπότε και του αντικειμένου που έχει τοποθετηθεί. Για παράδειγμα η εταιρεία Apple έχει δημιουργήσει το Apple AirTag³, όπου από ένα κινητό τηλέφωνο μπορεί ανά πάσα στιγμή να εντοπισθεί που βρίσκεται το AirTag. Παρόμοιες συσκευές εντοπισμού υπάρχουν και για πολλά διαφορετικά αντικείμενα όπως οχήματα, σκάφη, κατοικίδια κ.ά..

2 Maughan, R. (2016). *Mirror*. Retrieved from How many items you lose in a lifetime revealed - and what are the most common: <https://www.mirror.co.uk/news/uk-news/how-many-items-you-lose-8246553>

3 Apple. (2023). *AirTag*. Ανάκτηση από Apple: <https://www.apple.com/gr/airtag/>

Ωστόσο, με τον παραπάνω τρόπο είναι περιορισμένα τα αντικείμενα που μπορούν να εντοπισθούν και εκτός αυτού απαιτείται ένα αρκετά μεγάλο χρηματικό ποσό. Μια πιο πρόσφορη λύση είναι η συνεισφορά όλων των ανθρώπων που μπορούν να βοηθήσουν στην εύρεση ενός αντικειμένου με παρατηρητικότητα τους. Σήμερα, υπάρχουν διαδικτυακές υπηρεσίες και ομάδες σε μέσα μαζικής δικτύωσης τα οποία σχετίζονται με την εύρεση χαμένων αντικειμένων. Μάλιστα, αυτές οι υπηρεσίες/ομάδες έχουν αυξημένη κινητικότητα περισσότερο για τις κατηγορίες των κατοικίδιων και των ηλεκτρονικών συσκευών. Το μειονέκτημά τους είναι ότι οι χρήστες δεν μπορούν να αναζητήσουν εύκολα και έξυπνα χαμένα ή ανευρεθέντα αντικείμενα, πράγμα το οποίο είναι μείζονος σημασίας.

ΚΕΦΑΛΑΙΟ 3^ο: ΣΧΕΤΙΚΑ ΜΕ ΤΟ ANDROID

1.1 Ιστορία

Το Android είναι λειτουργικό σύστημα για έξυπνες κινητές συσκευές το οποίο βασίζεται στο λειτουργικό του Linux. Η πρώτη του ανάπτυξη πραγματοποιήθηκε από την Google και αργότερα από την Open Handset Alliance. Αρχικά σχεδιάστηκε για την κάλυψη των αναγκών λειτουργικού συστήματος στις συσκευές με οθόνη αφής όπως τα smartphones και τα tablet, όμως αναπτύχθηκε και σε άλλες συσκευές όπως έξυπνες τηλεοράσεις, οθόνες αυτοκινήτων και έξυπνα ρολόγια χειρός. Η πρώτη παρουσίαση της πλατφόρμας έγινε την 5 Νοεμβρίου 2007 και το λογότυπό του είναι ένα πράσινο ρομπότ.⁴



Εικόνα 3.1 Λογότυπο Android

1.2 Τεχνολογία

1.2.1 Components Εφαρμογής

Υπάρχουν τεσσάρων ειδών components μιας εφαρμογής Android.⁵ Αυτά είναι τα:

- Activities (Δραστηριότητες)
- Services (Υπηρεσίες)
- Broadcast Receivers (Δέκτες Εκπομπής) και
- Content Providers (Πάροχοι Περιεχομένου)

Κάθε component έχει τον δικό του σκοπό στην εφαρμογή και έχει το δικό του κύκλο ζωής στην εξέλιξή της, καθώς δημιουργείται και καταστρέφεται όταν αυτό είναι επιθυμητό.

4 Wikipedia. (2023). *Android (operating system)*. Ανάκτηση από [https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system))

5 Android. *Application fundamentals*. Ανάκτηση από Android Developers: <https://developer.android.com/guide/components/fundamentals>

Το Activity πρόκειται το σημείο αλληλεπίδρασης με τον χρήστη. Αντιπροσωπεύει μία μόνο οθόνη με διεπαφή χρήστη. Μία εφαρμογή συνήθως χρειάζεται πολλαπλά Activities για να μπορεί να είναι λειτουργική και εύκολη στη χρήση της. Παρόλο που οι δραστηριότητες συνεργάζονται για να δημιουργήσουν μια συνεκτική εμπειρία στον χρήστη, κάθε μία είναι ανεξάρτητη από τις άλλες.

Μία Υπηρεσία είναι ένα σημείο εισόδου γενικού σκοπού για την διατήρηση μιας εφαρμογής στο παρασκήνιο για κάθε είδους λόγο. Είναι ένα στοιχείο που εκτελείται στο παρασκήνιο για την ολοκλήρωση μακροχρόνιων λειτουργιών ή για την εκτέλεση εργασιών για απομακρυσμένες διαδικασίες. Για παράδειγμα μια υπηρεσία ενδέχεται να αναπαράγει μουσική στο παρασκήνιο ενώ ο χρήστης βρίσκεται σε διαφορετική εφαρμογή.

Ένας δέκτης εκπομπής είναι ένα στοιχείο που επιτρέπει στο σύστημα να παραδίδει συμβάντα στην εφαρμογή έξω από την κανονική ροή που ακολουθεί ο χρήστης, επιτρέποντας στην εφαρμογή να ανταποκρίνεται σε ανακοινώσεις εκπομπής σε όλο το σύστημα. Για παράδειγμα μια εφαρμογή μπορεί να προγραμματίσει να εμφανίσει μια ειδοποίηση για ένα επερχόμενο συμβάν και για αυτό το λόγο η εφαρμογή δεν είναι απαραίτητο να εκτελείται μέχρι να εμφανιστεί η ειδοποίηση.

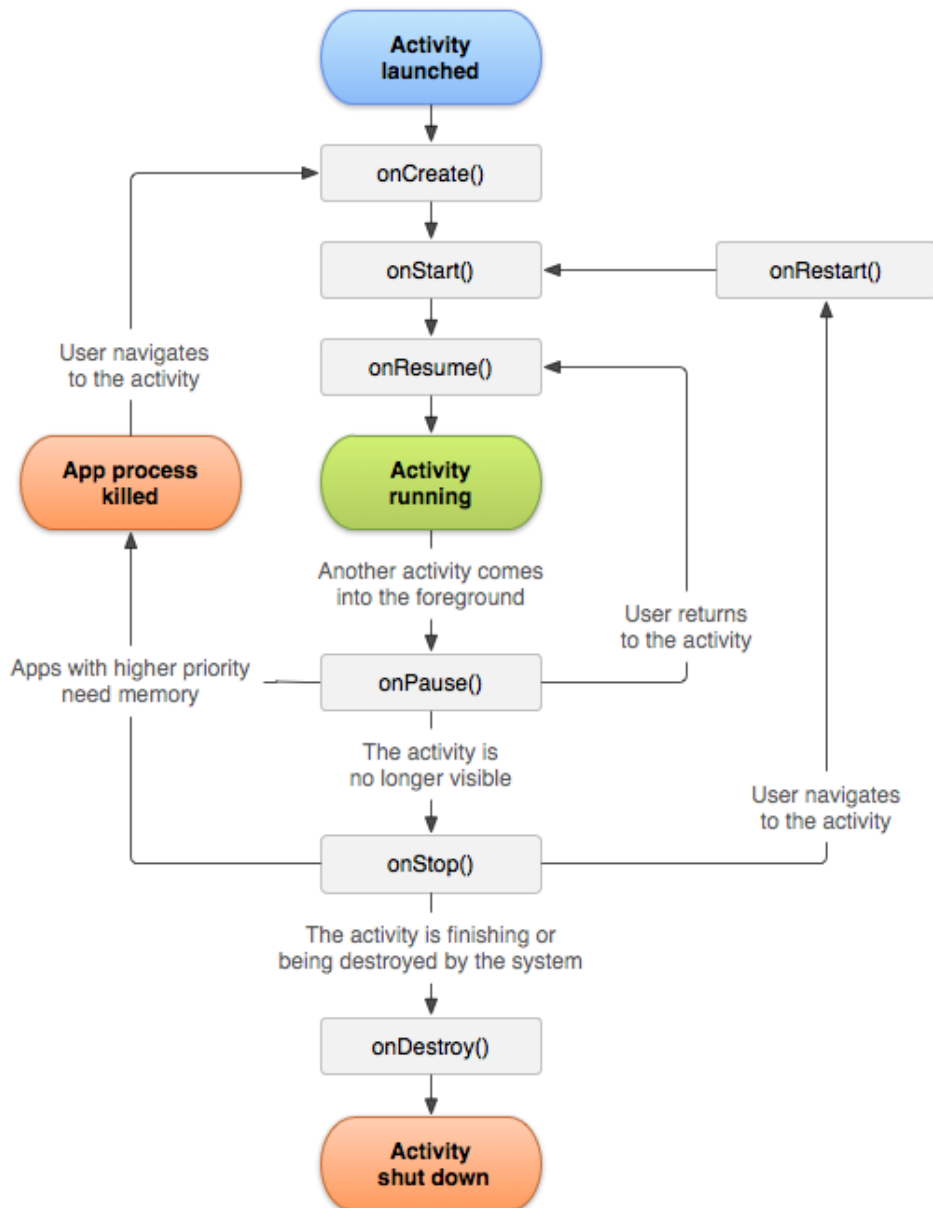
Ο πάροχος περιεχομένου διαχειρίζεται ένα κοινόχρηστο σύνολο δεδομένων εφαρμογών που μπορεί να αποθηκευτούν στο σύστημα αρχείων της συσκευής, σε μια βάση δεδομένων, στον διαδίκτυο ή σε οποιαδήποτε άλλη μόνιμη τοποθεσία αποθήκευσης στην οποία μπορεί να έχει πρόσβαση μια εφαρμογή. Μέσω του παρόχου μπορεί να υποβληθούν ερωτήματα ή να τροποποιηθούν τα δεδομένα αν φυσικά επιτρέπεται αυτή η πρόσβαση. Για παράδειγμα ένας πάροχος περιεχομένου θα μπορούσε να είναι η αναζήτηση στο Google, δηλαδή ένα διαδικτυακό πρόγραμμα στο οποίο αφού δώσουμε κάποια είσοδο (ερώτημα) θα μας επιστρέψει τα δεδομένα που χρειαζόμαστε (αναζήτηση στην βάση δεδομένων της Google).

1.2.2 Κύκλος Ζωής Activity

Καθώς ένας χρήστης βρίσκεται σε ένα συγκεκριμένο Activity και για κάποιο λόγο αποφασίσει να βγει από την εφαρμογή και να επιστρέψει σε αυτή μετά από λίγο, το Activity μεταβαίνει σε διαφορετικές καταστάσεις του κύκλου ζωής του. Το κάθε Activity παρέχει έναν αριθμό μεθόδων που του επιτρέπουν να γνωρίζει ότι μια κατάσταση έχει αλλάξει. Συγκεκριμένα, ένα Activity μπορεί να ξέρει πότε το σύστημα το έχει δημιουργήσει, σταματήσει, επαναφέρει ή καταστρέψει.⁶

6 Android. *The activity lifecycle*. Ανάκτηση από Android Developer: <https://developer.android.com/guide/components/activities/activity-lifecycle>

Παρακάτω θα πραγματοποιηθεί επεξήγηση του κύκλου ζωής ενός activity από την δημιουργία του έως τον τερματισμό του. Για να γίνει αυτό θα χρησιμοποιηθούν οι μέθοδοι: δημιουργίας «onCreate()», έναρξης «onStart()», επαναφοράς «onResume()», παύσης/διακοπής «onPause()», επανεκκίνησης «onRestart()», τερματισμός «onStop()» και καταστροφής «onDestroy()». Στην εικόνα 3.2 φαίνεται σχηματικά πως συνδέονται όλες αυτές οι μέθοδοι.



Εικόνα 3.2 Κύκλος ζωής ενός Activity

Αν κριθεί απαραίτητο σε μια εφαρμογή να ενεργοποιηθεί ένα Activity είτε λόγω επιλογής του χρήστη είτε λόγω καθορισμένης λειτουργίας της εφαρμογής, αυτό γίνεται με καθορισμένες κλήσεις των μεθόδων που αναφέρθηκαν

προηγουμένως. Για να λειτουργήσει ένα Activity δεν απαιτείται να έχουν δημιουργηθεί προγραμματιστικά όλες οι μέθοδοι εκτός από την `onCreate()`.

- Κατά την ενεργοποίηση ενός Activity, πρώτα καλείται η μέθοδος **`onCreate()`** η οποία δεσμεύει τα δεδομένα που χρησιμοποιεί ο χρήστης, περιέχει την σύνδεση του πηγαίου κώδικα με το μοντέλο εμφάνισης και δημιουργεί μεταβλητές οι οποίες έχουν προκαθοριστεί εξωτερικά από αυτή.
- Έπειτα, καλείται η μέθοδος **`onStart()`** ακόμα και αν δεν έχει δημιουργηθεί προγραμματιστικά, η οποία πυροδοτείται αμέσως μετά την `onCreate()`.
- Στη συνέχεια καλείται η μέθοδος **`onResume()`** ακόμα και αν δεν έχει δημιουργηθεί προγραμματιστικά, η οποία πυροδοτείται αμέσως μετά την `onStart()`.

Μέχρι αυτό το σημείο, η εφαρμογή πρέπει να λειτουργεί κανονικά και να έχει εμφανιστεί το μοντέλο εμφάνισης στην οθόνη του χρήστη. Υπάρχουν όμως δύο περιπτώσεις να διακοπεί η συνέχεια του Activity. Αυτές είναι αν ο χρήστης ή το σύστημα τερματίσουν το Activity ή αν ο χρήστης ή το σύστημα διακόψουν το Activity. Αντίστοιχα για τις παραπάνω περιπτώσεις υπάρχουν οι μέθοδοι που καλούνται πριν από κάθε τερματισμό ή διακοπή οι οποίες είναι οι `onStop()` και `onPause()`.

Καθώς το Activity εκτελείται, όταν αυτό διακοπεί προσωρινά για οποιονδήποτε λόγο χωρίς να χαθεί η ορατότητά του, τότε καλείται η μέθοδος `onPause()` η οποία τοποθετεί σε κατάσταση παύσης το Activity. Για παράδειγμα ένα άλλο activity δημιουργείται πάνω σε αυτό που ήδη εκτελείται χωρίς όμως το πρώτο να χάσει την ορατότητά του. Για να επανέλθει το Activity στην αρχική του κατάσταση, πρέπει να κληθεί η μέθοδος `onResume()` και αυτό γίνεται όταν ο χρήστης επανέλθει στο συγκεκριμένο Activity.

Ωστόσο, αν το Activity εκτελείται και ο χρήστης επιλέξει να εξέλθει από την εφαρμογή χωρίς να την τερματίσει (πατώντας το κουμπί Home της συσκευής) ή αν δημιουργηθεί νέο Activity, χωρίς όμως να είναι ορατό και στις δύο περιπτώσεις, τότε πρώτα καλείται η μέθοδος `onPause()` που είδαμε προηγουμένως και μετά καλείται η μέθοδος `onStop()`. Στην τελευταία μέθοδο το Activity δεν έχει ακόμα τερματίσει οπότε μπορούμε καλώντας την μέθοδο `onRestart()` να το επαναφέρουμε στην αρχική του κατάσταση, καλώντας με τη σειρά της τις μεθόδους `onStart()` και `onResume()`. Η μέθοδος `onRestart()` καλείται όταν το Activity επαναφέρεται σε ορατή κατάσταση.

Αξίζει να σημειωθεί ότι αν το Activity βρίσκεται σε κατάσταση παύσης/διακοπής (Pause) ή τερματισμού (Stop) και η συσκευή αντιληφθεί ότι η φυσική μνήμη εξαντλείται λόγω ύπαρξης άλλων προγραμμάτων που εκτελούνται, τότε η συσκευή μπορεί να πάρει την πρωτοβουλία και να τερματίσει το Activity. Αυτό σημαίνει ότι αν ο χρήστης πάει να επιστρέψει στο Activity, αυτό θα πρέπει

να εκτελεσθεί από το πρώτο βήμα το οποίο είναι η κλήση της μεθόδου onCreate().

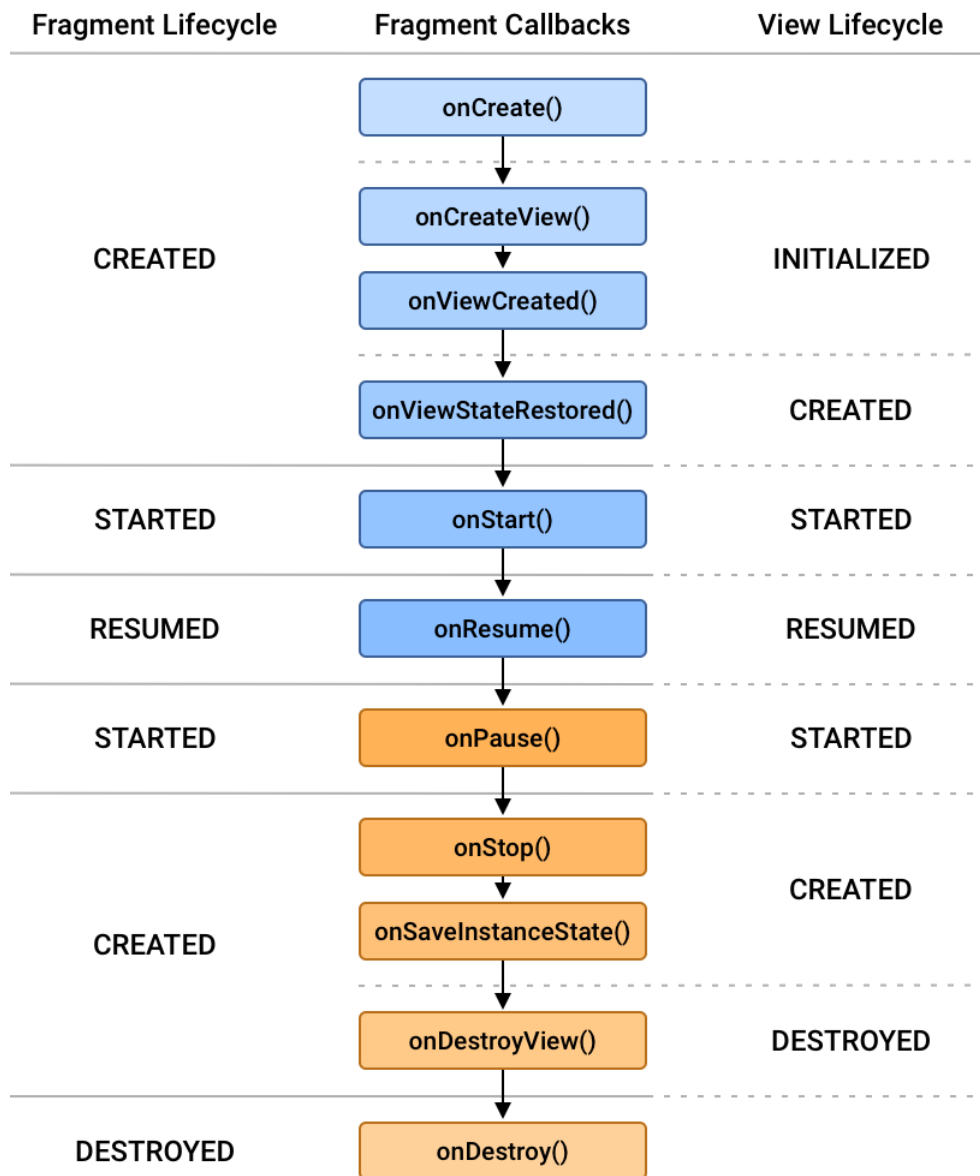
Τέλος, όταν ο χρήστης ή το σύστημα τερματίσουν το Activity τότε καλείται η μέθοδος onDestroy(), αφού κληθούν οι μέθοδοι onPause() και onStop(), όπου οι τιμές των μεταβλητών που βρίσκονται στο συγκεκριμένο Activity απελευθερώνονται από τη φυσική μνήμη της συσκευής.

1.2.3 Κύκλος Ζωής Fragment

Το Fragment αντιπροσωπεύει ένα επαναχρησιμοποιήσιμο τμήμα της διεπαφής χρήστη μίας εφαρμογής. Καθορίζει και διαχειρίζεται τη δική του οπτική απεικόνιση, έχει τον δικό του κύκλο ζωής και μπορεί να χειριστεί τα δικά του συμβάντα εισόδου. Τα Fragments δεν μπορούν να εκτελούνται μόνα τους αλλά πρέπει να περιέχονται σε ένα Activity. Η ιεραρχία προβολής των Fragments γίνεται μέρος ή συνδέεται με την ιεραρχία προβολής του Activity που περιέχονται. Αφού κάθε Fragment έχει το δικό του κύκλο ζωής, όταν ένας χρήστης πλοηγείται και αλληλεπιδρά με την εφαρμογή, τα Fragments μεταβαίνουν σε διάφορες καταστάσεις στον κύκλο ζωής τους καθώς προστίθενται, αφαιρούνται και εισέρχονται ή εξέρχονται από την οθόνη.⁷

Συγκεκριμένα, ένα Fragment μπορεί να βρεθεί στις εξής βασικές καταστάσεις: αρχικοποιημένο (Initialized), δημιουργημένο (created), έναρξη (started), επαναφορά (resumed), κατεστραμμένο (destroyed).

7 Android. *Fragment lifecycle*. Ανάκτηση από Android Developer: <https://developer.android.com/guide/fragments/lifecycle>



Εικόνα 3.3 Κύκλος ζωής ενός Fragment

Οι καταστάσεις αυτές των Fragments συνδυάζονται άμεσα με τις καταστάσεις του Activity το οποίο εξυπηρετούν. Η κατάσταση ενός Fragment δεν μπορεί ποτέ να είναι μεγαλύτερη χρονικά από την μητρική του (κατάσταση του Activity). Δηλαδή για να μεταβεί σε κατάσταση έναρξης (started) ένα Fragment, θα πρέπει το Activity το οποίο εξυπηρετεί να έχει ήδη δημιουργηθεί (created) με βάση αυτά που αναφέρθηκαν στην ενότητα 3.2.2. Αντίστοιχα η κατάσταση επαναφοράς (resumed) πρέπει πρώτα να ολοκληρωθεί στο Activity και έπειτα στο Fragment. Το ίδιο ισχύει και με τις υπόλοιπες καταστάσεις οι οποίες έχουν κοινά χαρακτηριστικά με τις καταστάσεις ενός Activity.

1.2.4 Layouts

Ένα Layout καθορίζει τη δομή της διεπαφής χρήστη σε μία εφαρμογή. Το Layout περιλαμβάνει είτε ορισμένα στοιχεία τα οποία ο χρήστης μπορεί να δει όπως ένα κουμπί ή ένα πεδίο κειμένου (Views) είτε μία ομάδα αντικειμένων η οποία είναι αόρατη και καθορίζει τη δομή διάταξης για τα Views (ViewGroup).⁸ Κατά τη δημιουργία ενός View ή ViewGroup, είναι απαραίτητο να συμπληρωθεί το ύψος και το πλάτος. Υπάρχουν πολλών ειδών Layouts όμως στην παρούσα εργασία θα παρουσιασθούν τα βασικότερα που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής.

- **RelativeLayout:** Το συγκεκριμένο Layout, είναι ένα ViewGroup που περιέχει παιδιά-Views σε σχετικές θέσεις μέσα στην οθόνη προβολής. Η θέση αυτή μπορεί να προσδιοριστεί ως σχετική με τα αδέρφια-Views μέσα στο συγκεκριμένο ViewGroup (όπως από κάτω από άλλο View) ή σε θέσεις σε σχέση με την περιοχή του πατέρα (όλου του ViewGroup).
- **ConstraintLayout:** Αυτό το Layout, έχει κατασκευασθεί για βοηθητικό σκοπό όπου επιτρέπεται να τοποθετηθούν και να διαχειριστούν άλλα Layouts μέσα σε αυτό.
- **LinearLayout:** Το LinearLayout, πρόκειται για ένα ViewGroup που ευθυγραμμίζει όλα τα παιδιά-Views σε μία κατεύθυνση (κάθετη ή οριζόντια). Η έκταση κάθε παιδιού-View μπορεί να καθοριστεί είτε με βάση συγκεκριμένη απόσταση είτε με βάση κάποια βάρη που θα έχει κάθε παιδί-View σε σχέση με το συνολικό βάρος του ViewGroup.

1.2.5 Views

Τα Views είναι το βασικό δομικό στοιχείο για την διεπαφή του χρήστη. Κάθε View καταλαμβάνει μια ορθογώνια περιοχή στην οθόνη και είναι υπεύθυνο για τον σχεδιασμό γραφικών στοιχείων και την διαχείριση ανάλογα με το έργο που εκτελεί. Όπως και στα Layouts, υποχρεωτικά στοιχεία που πρέπει να έχουν προκαθοριστεί είναι το ύψος και το πλάτος του View.⁹ Κάποια από τα βασικότερα Views παρουσιάζονται παρακάτω.

- **Button** απλό κουμπί
- **TextView** πεδίο κειμένου χωρίς δυνατότητα επεξεργασίας από τον χρήστη
- **EditText** πεδίο κειμένου με δυνατότητα επεξεργασίας από τον χρήστη
- **ImageView** πεδίο προβολής ψηφιακής εικόνας
- **ImageButton** πεδίο προβολής ψηφιακής εικόνας με την δυνατότητα συμπεριφοράς του ως κουμπί

8 Android. *Layouts*. Ανάκτηση από Android Developer: <https://developer.android.com/develop/ui/views/layout/declaring-layout>

9 Android. *View*. Ανάκτηση από Android Developer: <https://developer.android.com/reference/android/view/View>

- **ScrollView** πρόκειται για ένα ViewGroup το οποίο επιτρέπει την κατακόρυφη κύλιση των Views που βρίσκονται μέσα σε αυτό. Αξίζει να σημειωθεί ότι αυτά τα πολλαπλά Views πρέπει να βρίσκονται μέσα σε άλλο ViewGroup καθώς το ScrollView μπορεί να περιέχει μέχρι ένα ViewGroup ή View.

Επίσης, όπως συμβαίνει με το ScrollView, υπάρχουν και άλλα Views τα οποία έχουν δημιουργηθεί με τη βοήθεια των βασικότερων Views που αναφέραμε παραπάνω. Κάποια από αυτά είναι το CheckBox το οποίο είναι επέκταση του Button, το Spinner το οποίο είναι συνδυασμός πολλαπλών Views (ViewGroup) και στην πιο απλή του μορφή παριστάνει μια λίστα με στοιχεία, το DatePicker το οποίο είναι και αυτό ViewGroup και παριστάνει ένα ημερολόγιο που ο χρήστης μπορεί να επιλέξει συγκεκριμένη ημερομηνία κ.ά.. Τέλος, υπάρχει η δυνατότητα δημιουργίας View προσαρμοσμένου στις ανάγκες του προγραμματιστή.

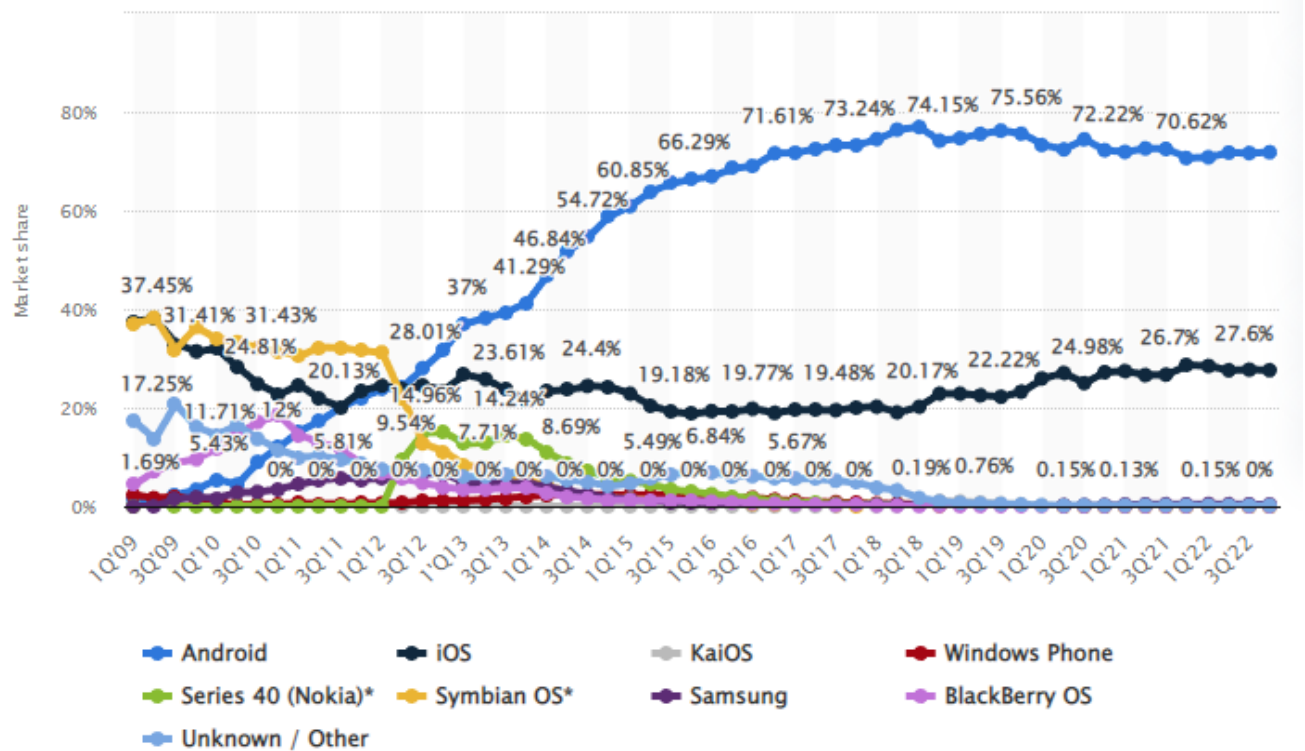
1.3 Δημοτικότητα

Παρότι το Android δεν ήταν το πρώτο που εμφανίστηκε ως λειτουργικό σύστημα σε smartphone συσκευές, γρήγορα κατέκτησε τα ηνία της αγοράς από τα πρώτα χρόνια κυκλοφορίας του. Συγκεκριμένα η πρώτη παρουσίαση της πλατφόρμας Android έγινε την 5 Νοεμβρίου 2007 και το 2010 βρέθηκε πρώτο στις προτιμήσεις των καταναλωτών.¹⁰

Ως προς τα στατιστικά δεδομένα, το τέταρτο τρίμηνο του 2022 το Android είχε 71.80% μερίδιο αγοράς ως προς τα λειτουργικά συστήματα που χρησιμοποιούνται σε κινητές συσκευές, έναντι του ποσοστού 27.60% του ανταγωνιστή του iOS.¹¹

10 aayushitaded2000. (2022). *History of Android*. Ανάκτηση από GeekForGeeks: <https://www.geeksforgeeks.org/history-of-android/>

11 Statista. (2023). *Mobile operating systems' market share worldwide from 1st quarter 2009 to 4th quarter 2022*. Ανάκτηση από Statista: <https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/>



Εικόνα 3.4 Ποσοστά μεριδίου αγοράς ως προς το λειτουργικό σύστημα smartphones

Ένας από τους σημαντικότερους λόγους αυτής της εξέλιξης εκτός από την ευκολία χρήσης του, είναι ο αριθμός των συσκευών που το υποστηρίζουν και το εύρος τιμών αυτών. Η τιμή αγοράς μίας συσκευής smartphone είναι καθοριστικός παράγοντας για τους περισσότερους χρήστες και πολλές φορές η αγορά μίας κινητής συσκευής με λειτουργικό σύστημα Android είναι μονόδρομος χωρίς όμως αυτό να καθιστά αρνητική την εμπειρία των χρηστών. Ακόμα, οι χρήστες έχουν ποικιλία επιλογών συσκευών μεταξύ πολλών κατασκευαστών και δεν περιορίζονται σε μια εταιρεία που παράγει έξυπνες συσκευές με το δικό της λογισμικό.

Μία ακόμα δυνατότητα που έχει βοηθήσει στην κυριαρχία του Android είναι ο τεράστιος αριθμός εφαρμογών για οποιαδήποτε χρήση που παρέχεται από το επίσημο κατάστημα εφαρμογών Play Store. Αυτό έχει πραγματοποιηθεί λόγω της ευκολίας και χωρίς ιδιαίτερους περιορισμούς, καταχώρησης μιας εφαρμογής στο Play Store από τους προγραμματιστές. Αντίθετα, σε άλλα λειτουργικά συστήματα, η δυνατότητα αποστολής μιας εφαρμογής από προγραμματιστές σε αντίστοιχες εφαρμογές λήψης λογισμικού όπως το Play Store, διέπεται από πολλούς περιορισμούς και πολλές φορές δημιουργεί εμπόδια διαμοιρασμού νέων εφαρμογών που μπορεί να είναι χρήσιμες για πολλούς χρήστες.

1.4 Το Android Studio και η ανάπτυξη της εφαρμογής

Το Android Studio είναι το επίσημο ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment IDE) για το λειτουργικό σύστημα Android της Google. Βασίζεται στο λογισμικό IntelliJ IDEA της εταιρείας JetBrains και έχει σχεδιαστεί ειδικά για ανάπτυξη Android εφαρμογών. Είναι διαθέσιμο για λήψη σε λειτουργικά συστήματα που βασίζονται σε Windows, macOS και Linux ή ως συνδρομητική υπηρεσία από το 2020. Το Android Studio ανακοινώθηκε ως IDE την 16 Μαΐου 2013 και η πρώτη σταθερή έκδοση κυκλοφόρησε τον Δεκέμβριο του 2014 ξεκινώντας από την έκδοση 1.0. Η ανάπτυξη Android εφαρμογών στο συγκεκριμένο περιβάλλον γίνεται σε γλώσσα προγραμματισμού Java, C++ ή Kotlin. Η γλώσσα Kotlin έχει σχεδιαστεί από την Google ειδικά για το Android Studio έως σήμερα και την 7 Μαΐου 2019 η Google ανακοίνωσε ότι η Kotlin είναι η προτιμώμενη γλώσσα προγραμματισμού για την ανάπτυξη εφαρμογών Android.¹²

Η εφαρμογή εκτός από την δυνατότητα ανάπτυξης λογισμικού με τα παραπάνω χαρακτηριστικά, παρέχει εικονικούς προσομοιωτές στους οποίους μπορούν να εκτελεσθούν οι εφαρμογές που υλοποιούνται. Ο προγραμματιστής μπορεί να εγκαταστήσει όσους προσομοιωτές επιθυμεί από κάθε υποστηριζόμενο είδος (smartphone, tablet, smartwatch κτλ.). Οι προσομοιωτές αυτοί είναι απόλυτα λειτουργικοί και η λειτουργία τους δεν διαφέρει σε τίποτα από μια φυσική συσκευή με τη δυνατότητα παραμετροποίησης ορισμένων παραμέτρων (π.χ. ποσοστό μπαταρίας).

Η ανάπτυξη της εφαρμογής android στην παρούσα εργασία πραγματοποιήθηκε σε γλώσσα προγραμματισμού Kotlin 1.8.0 στην έκδοση Android Studio Electric Eel με στοχευμένο επίπεδο API 33 και ελάχιστο 24. Χρησιμοποιήθηκε το build system Gradle 7.4 το οποίο υποστηρίζεται από το Android Studio.

12 Wikipedia. (2023). *Android Studio*. Ανάκτηση από Wikipedia: https://en.wikipedia.org/wiki/Android_Studio

ΚΕΦΑΛΑΙΟ 4^ο: ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ

1.1 Σκοπός και Χρησιμότητα της Εφαρμογής

Η κύρια χρησιμότητα της εφαρμογής είναι να δίνει την δυνατότητα στους χρήστες να καταχωρούν χαμένα αντικείμενα και να αναζητούν με έξυπνο τρόπο τα ήδη καταχωρημένα από άλλους χρήστες χαμένα ή ανευρεθέντα αντικείμενα. Με αυτόν τον τρόπο οι χρήστες θα μπορούν να αναζητούν με τα κριτήρια και τα φίλτρα της επιλογής τους χωρίς περιορισμούς ώστε να εντοπίσουν ευκολότερα το αντικείμενο που επιθυμούν. Όσον αφορά τον σκοπό της εφαρμογής, αυτός είναι να παρέχει τη μέγιστη βοήθεια στους χρήστες να εντοπίσουν τα αντικείμενα που έχουν χάσει.

1.2 Απαιτήσεις Λογισμικού

Οι απαιτήσεις λογισμικού σε μία εφαρμογή διακρίνονται σε λειτουργικές και μη λειτουργικές. Οι λειτουργικές απαιτήσεις ορίζουν ποιες υπηρεσίες θα πρέπει να παρέχει το σύστημα, πως θα πρέπει να αντιδρά σε συγκεκριμένες εισόδους και πως θα πρέπει να συμπεριφέρεται σε συγκεκριμένες καταστάσεις. Μη λειτουργικές απαιτήσεις είναι οι περιορισμοί στις υπηρεσίες ή τις λειτουργίες που προσφέρει το σύστημα, όπως χρονικοί περιορισμοί και θέματα απόδοσης.

Παρακάτω αναλύονται οι **λειτουργικές απαιτήσεις** της εφαρμογής.

1. Δυνατότητα δημιουργίας λογαριασμού ή σύνδεσης στην εφαρμογή με υπάρχον λογαριασμό.
2. Δυνατότητα σύνδεσης στην εφαρμογή με λογαριασμό Google.
3. Η εφαρμογή μπορεί να υποστηρίξει τα ελληνικά τα αγγλικά και τα ισπανικά σε όλες τις λειτουργίες.
4. Καταχώρηση αντικειμένων τα οποία έχουν χαθεί ή ανευρεθεί με τη συμπλήρωση των απαιτούμενων δεδομένων όπως κατηγορία, ημερομηνία περιστατικού, τοποθεσία κτλ.
5. Προβολή ιστορικού καταχωρημένων αντικειμένων και δυνατότητα επεξεργασία ή διαγραφής αυτών.
6. Επεξεργασία προφίλ όπου ο χρήστης θα μπορεί να τροποποιήσει το όνομα χρήστη, τον κωδικό πρόσβασης, την φωτογραφία του και να αλλάξει το καταχωρημένο φύλο. Επίσης, μπορεί να διαγράψει πλήρως τον λογαριασμό του μαζί με ότι καταχωρήσεις έχει δημιουργήσει.
7. Προβολή αντικειμένων άλλων χρηστών με δυνατότητα αναζήτησης.
8. Αναζήτηση αντικειμένων μέσω φίλτρου λέξης κλειδιού, κατηγορίας, είδους, ημερομηνίας και τοποθεσίας.
9. Έξυπνη αναζήτηση αντικειμένων με εισαγωγή φωτογραφίας από τον χρήστη (χρήση τεχνητής νοημοσύνης).
10. Αποστολή προσωπικών μηνυμάτων μεταξύ χρηστών.

11. Ειδοποιήσεις νέων εισερχόμενων μηνυμάτων κατά τη διάρκεια εκτέλεσης της εφαρμογής ή και χωρίς να εκτελείται η εφαρμογή.
12. Αλλαγή θέματος μεταξύ του φωτεινού και σκούρου θέματος.
13. Αναφορά προβλήματος προς τους διαχειριστές της εφαρμογής όπου ο χρήστης πληκτρολογεί και στέλνει το μήνυμά του για την παροχή βοήθειας.
14. Αποσύνδεση χρήστη κατόπιν εισόδου του στην εφαρμογή.
15. Ενεργοποίηση ή απενεργοποίηση όλων των ειδοποιήσεων της εφαρμογής.
16. Δυνατότητα επίλυσης ενός χαμένου αντικειμένου με το email του χρήστη που βρήκε και παρέδωσε το αντικείμενο στον ιδιοκτήτη του.

Παρακάτω αναλύονται οι **μη λειτουργικές απαιτήσεις**.

1. Ο χρόνος απόκρισης της εφαρμογής μεταξύ των λειτουργιών πρέπει να είναι μικρός χωρίς καθυστερήσεις.
2. Ο χρήστης αν δεν αποσυνδεθεί από τον λογαριασμό του ή αν δεν τερματίσει την εφαρμογή, αυτή θα τοποθετείται στο παρασκήνιο έως ότου ο χρήστης την ανοίξει πάλι. Αυτό θα γίνει για τη μείωση του χρόνου και δεδομένων για την εκ νέου σύνδεση.
3. Όλες οι λειτουργίες απαιτούν την σύνδεση στο διαδίκτυο είτε μέσω δεδομένων είτε μέσω WiFi.
4. Ορισμένες λειτουργίες απαιτούν την χρήση GPS μέσω των ρυθμίσεων της συσκευής (Καταχώρηση αντικειμένου, φίλτρο τοποθεσίας).
5. Τα προσωπικά στοιχεία των χρηστών πρέπει να προστατεύονται και να μην είναι διαθέσιμα σε άλλους χρήστες.
6. Τα προσωπικά μηνύματα που ανταλλάσσονται μεταξύ των χρηστών, πρέπει να είναι κωδικοποιημένα και να διαγράφονται μετά την παρέλευση 30 ημερών.

ΚΕΦΑΛΑΙΟ 5^ο: ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΑΛΛΑ ΕΡΓΑΛΕΙΑ

1.1 Επιλογή Βάσης Δεδομένων

Η βάση δεδομένων που επιλέχθηκε για να αποθηκεύονται οι βασικές πληροφορίες των χρηστών, οι καταχωρήσεις αντικειμένων και τα προσωπικά μηνύματα που ανταλλάσσουν οι χρήστες μεταξύ τους, είναι η Firebase της Google. Η επιλογή της βάσης δεδομένων που θα συνδεθεί η Android Εφαρμογή, αποφασίσθηκε λόγω εύκολης πρόσβασης και διαχείρισης που παρέχεται μέσω του Android Studio. Πρόκειται για μια Cloud NoSQL βάση δεδομένων, πράγμα που σημαίνει ότι δεν είναι εφικτή η διαχείρισή της μέσω SQL queries. Το σημαντικότερο προτέρημα είναι ότι η συγκεκριμένη βάση δεδομένων βρίσκεται σε cloud αποθηκευτικό χώρο και για αυτό το λόγο δεν χρειάζεται εγκατάσταση και δημιουργία τοπικής βάσης δεδομένων σε κάθε συσκευή που θα χρησιμοποιηθεί η εφαρμογή.

Η Firebase παρέχει πολλά είδη αποθήκευσης για κάθε ανάγκη. Για τις ανάγκες της παρούσας εργασίας, χρησιμοποιήθηκε η λειτουργία Authentication για την αποθήκευση και είσοδο των καταχωρημένων χρηστών στην εφαρμογή, καθώς και η Realtime Database στην οποία θα περιέχονται όλα τα αποθηκευμένα στοιχεία.^{13, 14}

1.2 Σχεδιασμός Βάσης Δεδομένων

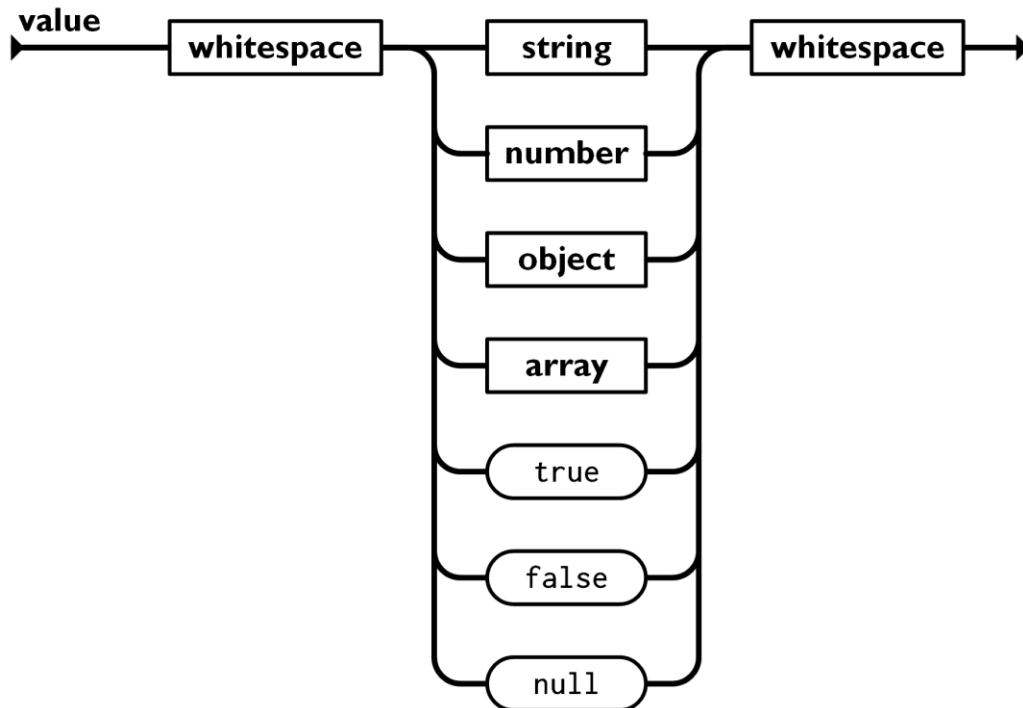
Για την ορθή και ασφαλή δημιουργία λογαριασμού και είσοδο εντός της εφαρμογής, χρησιμοποιείται η λειτουργία Authentication στην οποία απαιτούνται: το μοναδικό email και ο κωδικός του χρήστη τόσο για την δημιουργία όσο και για την αυθεντικοποίηση ενός λογαριασμού. Παράλληλα, με την δημιουργία του λογαριασμού, παράγεται και αποθηκεύεται ένα μοναδικό κλειδί το οποίο ονομάζεται User UID. Το User UID χρησιμοποιείται για την εύρεση της εγγραφής που περιέχει τα στοιχεία του χρήστη και τον έλεγχο προσβασιμότητας στα στοιχεία αυτά.

Εκτός από το email και τον κωδικό πρόσβασης, η αυθεντικοποίηση μπορεί να γίνει με κάποιον λογαριασμό Google, όπου η μοναδική προϋπόθεση είναι ο χρήστης να είναι ήδη συνδεδεμένος με τον λογαριασμό του στη συσκευή ή να συνδεθεί εκείνη τη στιγμή μέσω του αναδυόμενου παραθύρου της Google. Με αυτόν τον τρόπο, νέοι χρήστες δεν είναι απαραίτητο να δημιουργήσουν νέο λογαριασμό μέσω της εφαρμογής.

13 Google. *Firestore Realtime Database*. Ανάκτηση από <https://firebase.google.com/docs/database>

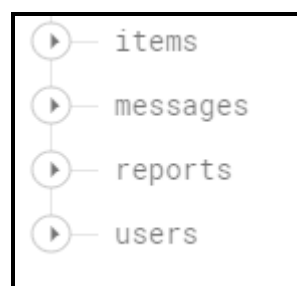
14 Google. *Firestore Authentication*. Ανάκτηση από <https://firebase.google.com/docs/auth>

Για την σωστή λειτουργία της εφαρμογής και για την άμεση ανταπόκριση στις καταχωρήσεις και τροποποιήσεις των δεδομένων, χρησιμοποιήθηκαν τέσσερα βασικά παιδιά-κλαδιά της βάσης δεδομένων Realtime Database. Η εμφάνιση των αποθηκευμένων στοιχείων στην βάση δεδομένων, έχουν την μορφή JSON (JavaScript Object Notation) αρχείων. Δηλαδή έχουν την μορφή ενός κατάλληλα ιεραρχημένου δέντρου όπως φαίνεται στην εικόνα 5.1 όπου σε κάθε επίπεδο παιδιού-κλαδιού μπορεί να δεχτεί οποιαδήποτε μορφή δεδομένων όπως είναι ένα κείμενο, ένα νούμερο ή μια λίστα.¹⁵



Εικόνα 5.1 Τιμές που μπορεί να δεχθεί το JSON

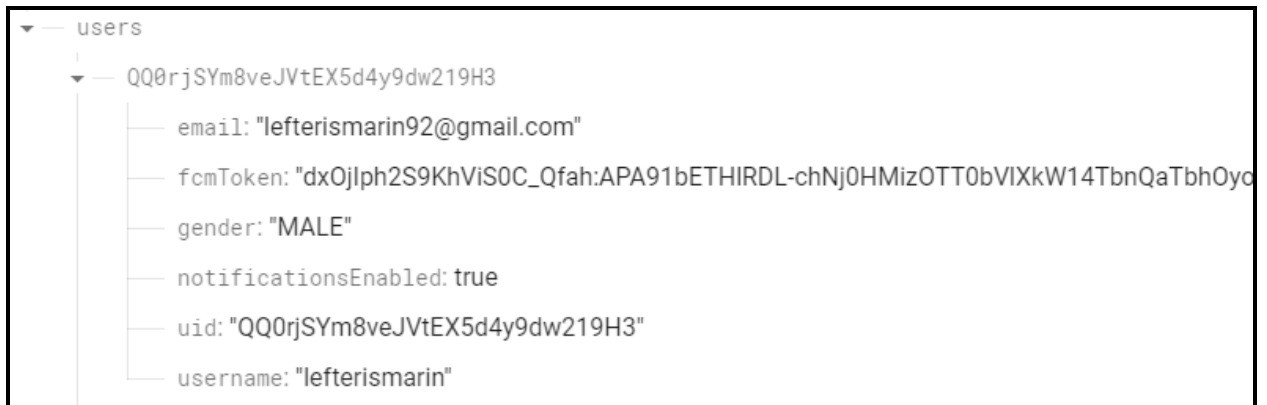
Τα τέσσερα παιδιά-κλαδιά που χρησιμοποιήθηκαν έχουν ως κύρια ονόματα κλαδιού τα εξής:



Εικόνα 5.2 Παιδιά-Κλαδιά της Realtime Database που χρησιμοποιήθηκε

15 JSON. Εισαγωγή στο JSON. Ανάκτηση από JSON: <https://www.json.org/json-el.html>

1. **Users:** Περιέχει τα στοιχεία του κάθε χρήστη τα οποία είναι προσβάσιμα μόνο από τον ίδιο εκτός του username όπου μπορούν να το βλέπουν και άλλοι χρήστες. Αυτά είναι τα εξής:
 - 1.1. Κλειδί εγγραφής
 - 1.2. Όνομα χρήστη
 - 1.3. Φύλο
 - 1.4. Email
 - 1.5. Σκοτεινό θέμα
 - 1.6. Ενεργοποίηση ειδοποιήσεων
 - 1.7. FCM Token (μοναδικό κλειδί ανά συσκευή εγκατάστασης για αποστολή ειδοποιήσεων όταν η εφαρμογή δεν εκτελείται)



Εικόνα 5.3 Σχήμα βάσης δεδομένων χρηστών

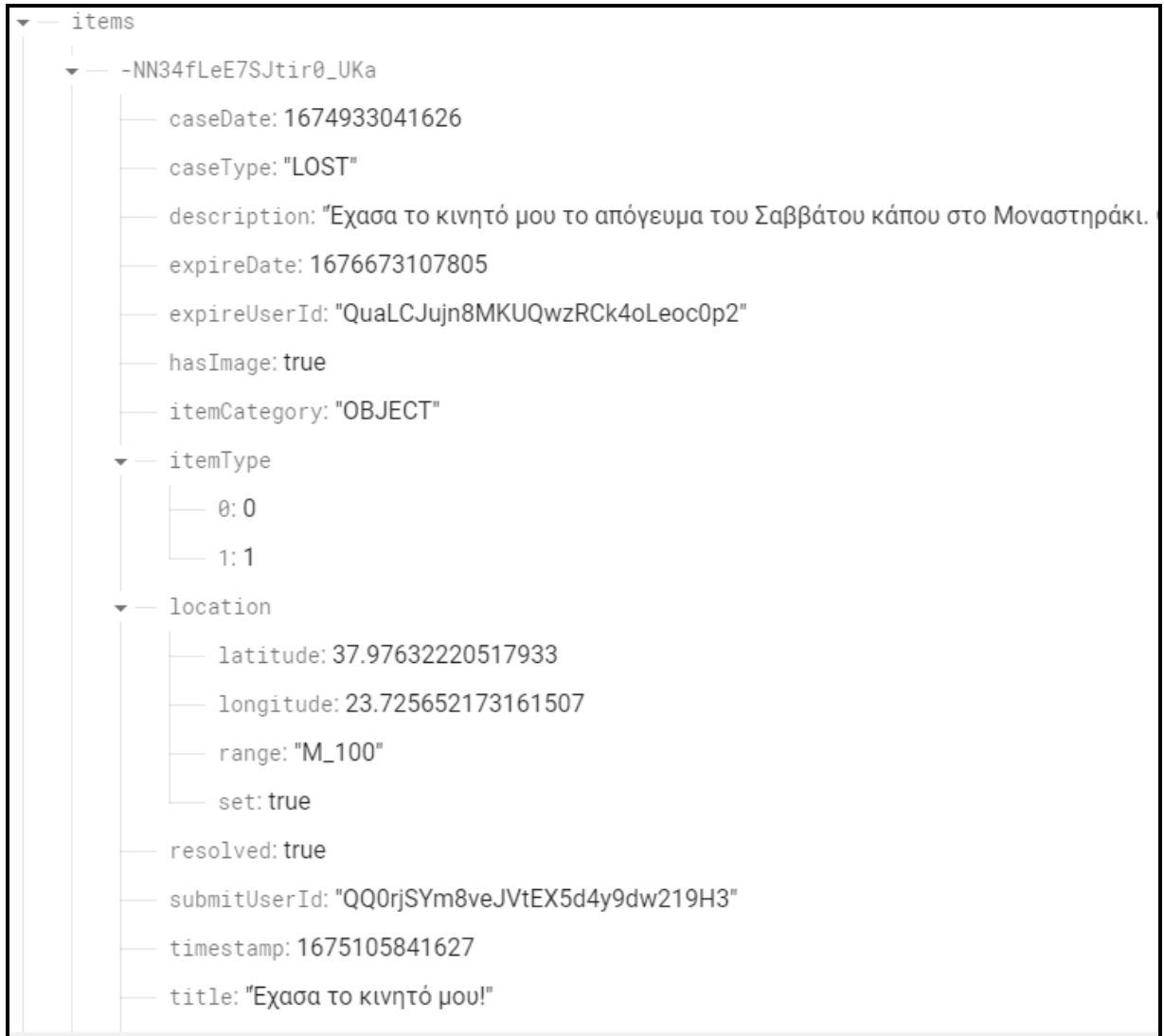
2. **Messages:** Περιέχει τα βασικά στοιχεία των προσωπικών μηνυμάτων που ανταλλάσσονται μεταξύ των χρηστών. Αυτά είναι τα εξής:
 - 2.1. Κλειδί εγγραφής
 - 2.2. Κλειδί αποστολέα
 - 2.3. Κλειδί παραλήπτη
 - 2.4. Ημερομηνία μηνύματος
 - 2.5. Αναγνώστηκε από παραλήπτη
 - 2.6. Περιεχόμενο μηνύματος σε κωδικοποιημένη μορφή (χρησιμοποιείται ο αλγόριθμος κρυπτογράφησης AES-CBC με μέγεθος block 128bits)
 - 2.7. Σχέση μεταξύ αποστολέα-παραλήπτη (χρησιμοποιείται για λόγους εύρεσης συνομιλιών)

- 2.8.** Ορατό στον αποστολέα (χρησιμοποιείται σε περίπτωση διαγραφής μηνύματος από τον αποστολέα)
- 2.9.** Ορατό στον παραλήπτη (χρησιμοποιείται σε περίπτωση διαγραφής μηνύματος από τον παραλήπτη)



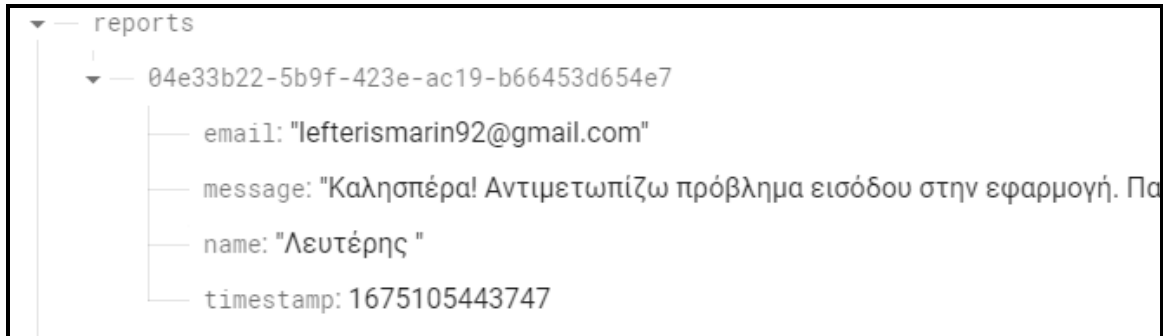
Εικόνα 5.4 Σχήμα βάσης δεδομένων προσωπικών μηνυμάτων

- 3. Items:** Περιέχει τα στοιχεία που απαιτούνται για κάθε καταχώρηση περιστατικού απώλειας ή ανεύρεσης αντικειμένου. Αυτά είναι τα εξής:
- 3.1.** Κλειδί εγγραφής
 - 3.2.** Κατηγορία περιστατικού (Χάθηκε – Βρέθηκε)
 - 3.3.** Κατηγορία αντικειμένου (Αντικείμενο – Κατοικίδιο – Όχημα)
 - 3.4.** Είδος Αντικειμένου ανάλογα την κατηγορία
 - 3.5.** Τίτλος περιστατικού
 - 3.6.** Περιγραφή περιστατικού
 - 3.7.** Ημερομηνία περιστατικού
 - 3.8.** Ημερομηνία καταχώρησης
 - 3.9.** Έχει καταχωρηθεί φωτογραφία
 - 3.10.** Τοποθεσία (γεωγραφικό μήκος, γεωγραφικό πλάτος και ακτίνα)
 - 3.11.** Κλειδί χρήστη καταχώρησης
 - 3.12.** Επιλύθηκε
 - 3.13.** Κλειδί χρήστη επίλυσης (expireUserId)
 - 3.14.** Ημερομηνία επίλυσης (expireDate)



Εικόνα 5.5 Σχήμα βάσης δεδομένων αντικειμένων

4. **Reports:** Περιέχει όλες τις καταχωρημένες αναφορές προβλημάτων από τους χρήστες μέσω της εφαρμογής με τα βασικά στοιχεία αυτών. Αυτές είναι οι εξής:
 - 4.1. Email
 - 4.2. Ημερομηνία αναφοράς
 - 4.3. Συνταχθέν κείμενο
 - 4.4. Όνομα



Εικόνα 5.6 Σχήμα βάσης δεδομένων αναφορών

1.3 Κανόνες Βάσης Δεδομένων

Όλα τα παιδιά-κλαδιά που δημιουργήθηκαν, μαζί με τα δεδομένα που φέρουν, πρέπει με κάποιο τρόπο να προστατεύονται. Για τον λόγο αυτό στην Firebase και συγκεκριμένα στην Realtime Database, υπάρχουν οι κανόνες οι οποίοι επιτρέπουν την πρόσβαση σε συγκεκριμένα δεδομένα και σε συγκεκριμένους χρήστες ώστε τα δεδομένα αυτά να μην είναι επεξεργάσιμα ή ορατά από οποιονδήποτε. Οι κανόνες αυτοί μπορεί να είναι απλοί ή σύνθετοι, ανάλογα με τους τρόπους που επιθυμεί ο προγραμματιστής να έχουν πρόσβαση διάφοροι χρήστες.

Για τις ανάγκες της εργασίας, χρησιμοποιήθηκαν κανόνες που αφορούν την ανάγνωση (read), την επεξεργασία (write) και την ευρετηρίαση (indexing) από κάθε παιδί-κλαδί. Όπως χρησιμοποιείται η μορφή JSON για την αναπαράσταση των δεδομένων στην Realtime Database, έτσι και στους κανόνες χρησιμοποιείται η ίδια μορφή. Κάθε παιδί-κλαδί της βάσης δεδομένων έχει δικούς του κανόνες για την ανάγνωση και την επεξεργασία αντίστοιχα.¹⁶ Παρακάτω θα δούμε όλους τους κανόνες:

16 Google. *Understand Firebase Realtime Database Security Rules*. Ανάκτηση από Firebase: <https://firebase.google.com/docs/database/security>

i. Users

```
/*rules for users table*/
"users": {
  /*allow read only if user is authenticated*/
  ".read": "auth != null",
  "$uid": {
    /*allow write only if user is authenticated and his uid matches the given one*/
    ".write": "auth != null && $uid == auth.uid"
  }
},
```

Εικόνα 5.7 Κανόνες χρηστών

Για τους χρήστες ο κανόνας της ανάγνωσης μιας εγγραφής είναι απλός. Αρκεί ο χρήστης που επιχειρεί να αποκτήσει πρόσβαση, να έχει εξουσιοδοτηθεί από την λειτουργία Authenticate. Για να μπορεί να έχει πρόσβαση σε μία από τις εγγραφές και να την επεξεργαστεί πρέπει εκτός από το να έχει εξουσιοδοτηθεί από την λειτουργία Authenticate, το `userId` του να ταιριάζει αυτό της εγγραφής. Επειδή, όπως αναφέρθηκε προηγουμένος, το `userId` είναι μοναδικό για κάθε χρήστη, κάθε ένας από αυτούς, θα έχει δυνατότητα τροποποίησης μόνο της δικιάς του εγγραφής. Σε κάθε άλλον χρήστη δεν θα επιτρέπεται η πρόσβαση.

ii. Messages

```
/*rules for messages table*/
"messages": {
  ".indexOn": "ownerId",
  /*allow read for all authenticated users*/
  ".read": "auth != null",
  "$key": {
    /*allow write only for authenticated user and
    * if his uid matches the one on exist userKey column or new userId*/
    ".write": "auth != null && (newData.child('ownerId').val() == auth.uid ||
    data.child('ownerId').val() == auth.uid)"
  }
},
```

Εικόνα 5.8 Κανόνες μηνυμάτων

Αντίστοιχα, το παιδί-κλαδί `messages` αφού πρέπει να είναι προσβάσιμο για ανάγνωση από όλους τους χρήστες, ελέγχεται μόνο αν ο συνδεδεμένος χρήστης έχει εξουσιοδοτηθεί από την λειτουργία Authentication. Όσον αφορά την επεξεργασία του κάθε μηνύματος, για την επεξεργασία μιας εγγραφής ακολουθείται η εξής πρόταση: *“ο χρήστης μπορεί να επεξεργαστεί την εγγραφή αν έχει εξουσιοδοτηθεί από την λειτουργία Authentication και το `OwnerId` του ταιριάζει με αυτό της υπάρχουσας εγγραφής ή αυτό της εγγραφής που πρόκειται*

να δημιουργηθεί". Στη συγκεκριμένη περίπτωση ο κανόνας αυτός περιλαμβάνει και τις εγγραφές που πρόκειται να καταχωρηθούν, διαφορετικά κανένας χρήστης δεν θα μπορούσε να καταχωρήσει επιτυχώς μια νέα υπόθεση διότι δεν θα είχε το δικαίωμα. Επίσης, σε αυτή τη περίπτωση υπάρχει και ευρετηρίαση (indexOn) των messages με βάση το κλειδί του αποστολέα (ownerId) για υποβοήθηση των queries που πρόκειται να πραγματοποιηθούν.

iii. Items

```
/*rules for items table*/
"items":{
  ".indexOn": "submitUserId",
  /*allow read for all*/
  ".read": true,
  "$key":{
    /*allow write only for authenticated user and
    * if his uid matches the one on exist userKey column or new userId*/
    ".write": "auth != null && (newData.child('submitUserId').val() == auth.uid ||
    data.child('submitUserId').val() == auth.uid)"
  }
},
```

Εικόνα 5.9 Κανόνες αντικειμένων

Η ανάγνωση των καταχωρημένων περιστατικών δεν διαφέρει από την ανάγνωση των μηνυμάτων πλην του σημείου ανάγνωσης καθώς πρέπει τα καταχωρημένα αντικείμενα να είναι προσβάσιμα από όλους τους χρήστες ακόμα και αν αυτοί δεν είναι συνδεδεμένοι. Για την επεξεργασία των εγγραφών πρέπει εκτός από την αυθεντικοποίηση του χρήστη, να ταιριάζει και το SubmitUserId του με αυτό της υπάρχουσας ή νέας εγγραφής, όπως λέχθηκε στα messages. Επίσης, υπάρχει και σε αυτή τη περίπτωση ευρετηρίαση με βάση το κλειδί submitUserId.

iv. Reports

```
/*rules for reports table*/
"reports": {
  /*allow read only if user is authenticated and has exact email.
  * Write permission have all users even if they are not authenticated*/
  ".read": "auth != null && auth.token.email == 'lefterismarin92@gmail.com'",
  ".write": true
},
```

Εικόνα 5.10 Κανόνες αναφορών

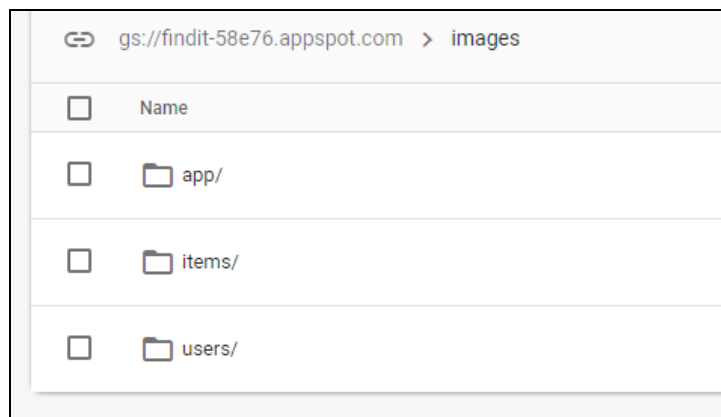
Στα Reports το δικαίωμα ανάγνωσης το έχει μόνο ο χρήστης με συγκεκριμένο email (αυτό το email θα είναι του διαχειριστή της εφαρμογής). Επίσης, υπάρχει η δυνατότητα να προστεθούν και άλλα email σε περίπτωση που υπάρχουν και άλλοι διαχειριστές. Το δικαίωμα της καταχώρησης θα το έχουν όλοι οι χρήστες ακόμα και αυτοί που δεν έχουν συνδεθεί στην εφαρμογή.

1.4 Αποθηκευτικός Χώρος Μέσων (Cloud Storage)

Η εφαρμογή εκτός από την αποθήκευση των δεδομένων, χρειάζεται και αποθήκευση μέσων όπως η εικόνα του χρήστη και η εικόνα του αντικειμένου που καταχωρείται. Η Realtime Database της Firebase δεν υποστηρίζει τέτοιο είδος αποθήκευσης (εκτός αν αποθηκευτεί το αρχείο σε Base64 μορφή). Για το λόγο αυτό, υπάρχει μια άλλη back-end υπηρεσία της Firebase που ονομάζεται Cloud Storage και έχει δημιουργηθεί από της Google για ανέβασμα και κατέβασμα αρχείων όπως εικόνες, βίντεο, αρχεία ήχου κ.ά..¹⁷ Ως υπηρεσία, ξεχωρίζει από τρία χαρακτηριστικά:

- τις ισχυρές λειτουργίες (robust operations) δηλαδή την ικανότητα να συνεχίζει να εκτελείται ανεξάρτητα από την ποιότητα του δικτύου (παύση της διαδικασίας όταν το δίκτυο είναι απασχολημένο και συνέχιση όταν απελευθερωθεί)
- Ισχυρή ασφάλεια (strong security) καθώς σε συνδυασμό με την υπηρεσία Authentication και κανόνων περιορισμού, δημιουργούν μια ασφαλή υπηρεσία
- Υψηλή επεκτασιμότητα (high scalability) αφού η συγκεκριμένη υπηρεσία έχει δημιουργηθεί για την κάλυψη δεδομένων της τάξης exabyte (10^{18} bytes)

Στην εφαρμογή υπάρχει ένας κάδος (bucket) ο οποίος έχει μοναδικό όνομα και μέσα του περιέχει όλους τους φακέλους και τα αρχεία που αποθηκεύουν οι χρήστες. Στην εικόνα 5.11 φαίνεται το περιεχόμενο του κάδου.



Εικόνα 5.11 Περιεχόμενα κάδου Cloud Storage

17 Google. *Cloud Storage for Firebase*. Ανάκτηση από Firebase: <https://firebase.google.com/docs/storage>

Όπως και με την Realtime Database, έτσι και με την Cloud Storage υπάρχουν κανόνες ασφάλειας οι οποίοι δηλώνονται για να προστατεύονται τα αρχεία που αποθηκεύονται. Για τις ανάγκες ανάπτυξης της συγκεκριμένης εφαρμογής, η προβολή των εικόνων είναι διαθέσιμη για όλους τους χρήστες ακόμα και μη συνδεδεμένους. Η επεξεργασία των εικόνων γίνεται μόνο από εγγεγραμμένους χρήστες.

```
1  rules_version = '2';
2  service firebase.storage {
3    match /b/{bucket}/o {
4      match /{allPaths=**} {
5        allow read;
6        allow write: if request.auth != null;
7      }
8    }
9  }
```

Εικόνα 5.12 Κανόνες Cloud Storage

1.5 Cloud Συναρτήσεις (Cloud Functions)

Σχεδόν σε κάθε android εφαρμογή που έχει δυνατότητα διαμοιρασμού δεδομένων, υπάρχει ένας αριθμός διαδικασιών που μπορούν να διεξαχθούν εκτός της ίδιας της εφαρμογής χωρίς να επιβαρύνουν την επεξεργαστική ισχύ της συσκευής του χρήστη. Αυτές οι διαδικασίες μπορεί να γίνουν σε ένα cloud σύστημα που παρέχεται από την Firebase που ονομάζεται Cloud Functions.

Το βασικό του προτέρημα είναι ότι το περιβάλλον του είναι ξεχωριστό από αυτό της android εφαρμογής, δηλαδή αναπτύσσεται σε ένα διαφορετικό project χωρίς φυσικά να χάνεται η επαφή του με τις λειτουργικότητες των άλλων υπηρεσιών της Firebase. Επίσης, ως cloud υπηρεσία, δεν θα χρειαστεί ποτέ από τον προγραμματιστή να κάνει κάποια ενέργεια για την συντήρηση ή την κλιμάκωση των ανεβασμένων διαδικασιών καθώς η Firebase αναλαμβάνει αυτό το καθήκον. Τέλος, η απομόνωση και η ασφάλεια των διαδικασιών από τους χρήστες, οδηγεί σε βέλτιστες τεχνικές ανάπτυξης μιας εφαρμογής.¹⁸

Όπως αναφέρθηκε, το περιβάλλον ανάπτυξης των Cloud Functions είναι διαφορετικό από αυτό της android εφαρμογής. Για τον λόγο αυτό θα επεξηγηθούν αναλυτικά οι διαδικασίες στο επόμενο κεφάλαιο.

18 Google. *Cloud Functions for Firebase*. Ανάκτηση από Firebase: <https://firebase.google.com/docs/functions>

1.6 Cloud Ειδοποιήσεις (Firebase Cloud Messaging)

Στο κεφάλαιο 4, αναφέρθηκε στις λειτουργικές απαιτήσεις ότι θα υπάρχει δυνατότητα ανταλλαγής μηνυμάτων μεταξύ των χρηστών. Όταν ένας χρήστης θα αποστέλλει προσωπικό μήνυμα σε κάποιον άλλο χρήστη, ο δεύτερος πρέπει με κάποιο τρόπο να ειδοποιηθεί ακόμα και αν δεν εκτελείται εκείνη τη χρονική στιγμή η εφαρμογή. Το Firebase Cloud Messaging ή εν συντομία FCM, παρέχει αυτή τη δυνατότητα, δηλαδή αποστέλλεται ένα μήνυμα με μέγιστο μέγεθος δεδομένων 4000 bytes, σε μια συγκεκριμένη συσκευή. Η προαναφερόμενη συσκευή καθορίζεται από ένα μοναδικό κλειδί το οποίο αλλάζει μόνο αν αφαιρεθεί και εγκατασταθεί ξανά η εφαρμογή στην ίδια συσκευή ή παρέλθει ένα μεγάλο χρονικό διάστημα (περίπου 2 μήνες) όπου το κλειδί επανακαθορίζεται από την υπηρεσία (key rotation) για λόγους ασφάλειας.¹⁹

Η ειδοποίηση στη συσκευή ενός χρήστη γίνεται καλώντας την συγκεκριμένη υπηρεσία με το περιεχόμενο του μηνύματος και το μοναδικό κλειδί. Το μοναδικό κλειδί έχει την επίσημη ονομασία FCM Token και περιέχεται στα στοιχεία του χρήστη εντός της Realtime Database. (βλέπε παράγραφο 5.2) Την κλήση της υπηρεσίας αναλαμβάνει να την εκτελέσει μια διαδικασία που υπάρχει στα Cloud Functions, οπότε ουσιαστικά γίνεται αυτόματα και λειτουργεί ως πυροδότηση (trigger) με την αποστολή ενός μηνύματος.

1.7 Τεχνητή Νοημοσύνη (Machine Learning)

Μας λειτουργικές απαιτήσεις αναφέρθηκε ότι η εφαρμογή θα υποστηρίζει αναζήτηση αντικειμένων με εισαγωγή φωτογραφίας από τον χρήστη με τη χρήση τεχνητής νοημοσύνης. Η Firebase παρέχει μια υπηρεσία Machine Learning η οποία διαθέτει μια γκάμα από λειτουργικότητες τεχνητής νοημοσύνης όπως αναγνώριση κειμένου, μετάφραση, αναγνώριση προσώπου, σάρωση barcode κ.ά.. Υπάρχει ένας διαχωρισμός της συγκεκριμένης υπηρεσίας σε Cloud και σε ενσωματωμένης εντός της εφαρμογής. Ο κάθε τρόπος έχει τα πλεονεκτήματα και τα μειονεκτήματα οπότε η επιλογή πρέπει να γίνεται πάντα με το ζητούμενο αποτέλεσμα.²⁰

Στην εφαρμογή της παρούσας εργασίας θα εφαρμοστεί η λειτουργικότητα επισήμανσης εικόνας (Image labeling). Αυτή η λειτουργικότητα απαιτεί την εισαγωγή ενός αρχείου εικόνας και επιστρέφει σε λέξεις τι περιέχει αυτή η εικόνα. Για παράδειγμα αν υπάρχει μια τσάντα στην εικόνα, θα μας επιστρέψει τη λέξη τσάντα.

19 Google. *Firebase Cloud Messaging*. Ανάκτηση από Firebase: <https://firebase.google.com/docs/cloud-messaging/>

20 Google. *Firebase Machine Learning*. Ανάκτηση από Firebase: <https://firebase.google.com/docs/ml>

Φυσικά οι λέξεις που μπορεί να επιστραφούν από το σύστημα είναι συγκεκριμένες και μάλιστα έως την ημερομηνία σύνταξης της μεταπτυχιακής διατριβής ο αριθμός τους ανέρχεται σε 446.²¹

Για τις ανάγκες της εφαρμογής, επιλέχθηκε η εκτέλεση του image labeling ML στην συσκευή του κάθε χρήστη και όχι στο cloud διότι κάθε αίτημα χρήστη θα έκανε αρκετό χρόνο μέχρι να επιστραφούν οι επισημάνσεις. Η συγκεκριμένη πλατφόρμα ονομάζεται ML Kit.²²

21 Google. *ML Kit image labeling: Labels for default model*. Ανάκτηση από Firebase: <https://developers.google.com/ml-kit/vision/image-labeling/label-map>

22 Google. *Image Labeling*. Ανάκτηση από Firebase: <https://firebase.google.com/docs/ml-kit/label-images>

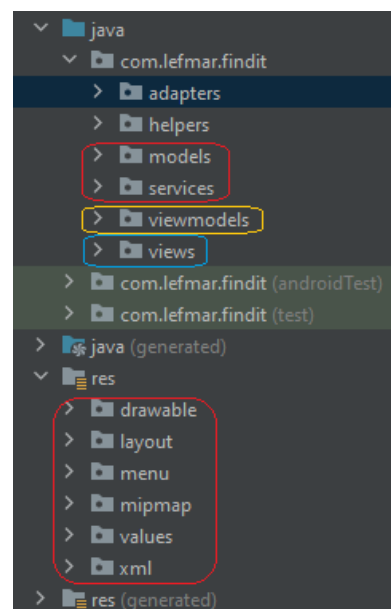
ΚΕΦΑΛΑΙΟ 6^ο: ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ ANDROID

1.1 Αρχιτεκτονική

Η αρχιτεκτονική σχεδίασης που χρησιμοποιήθηκε είναι η MVVM (Model View ViewModel).²³ Περιέχει τρεις διαφορετικές λειτουργίες:

1. **Model:** η δομή και ανάκτηση των δεδομένων που χρησιμοποιεί η εφαρμογή. Το model δεν επικοινωνεί απευθείας με το view αντ' αυτού η επικοινωνία γίνεται μέσω του ViewModel.
2. **View:** αντιπροσωπεύει την διεπαφή χρήστη (User Interface). Επικοινωνεί με το ViewModel με τη βοήθεια μεταβλητών οι οποίες κάθε φορά που τροποποιούνται από το ViewModel, ενημερώνεται το View (Observables). Views είναι τα Activities και τα Fragments.
3. **ViewModel:** Είναι ο ενδιάμεσος κρίκος μεταξύ View και Model. Οι δυνατότητές του είναι να αιτείται δεδομένα και να τα μετατρέπει σε κατάλληλα Objects μέσω του Model. Επίσης, παρέχει τη δυνατότητα μεταφοράς των δεδομένων που έχουν ανακτηθεί ή τροποποιηθεί στα Views.

Αξίζει να σημειωθεί ότι το model για λόγους “καθαρού” κώδικα έχει διαχωριστεί σε δομές δεδομένων (entities, data, enums και interfaces) και λειτουργικότητα (services). Επίσης, στην αρχιτεκτονική του project υπάρχουν και άλλα βοηθητικά πακέτα (φάκελοι) που περιέχουν επιπλέον λειτουργικότητα χωρίς να είναι ενσωματωμένη σε κάποια από τα τρία κυρίως κομμάτια του MVVM. Αυτά τα πακέτα είναι τα adapters και helpers. Στην εικόνα 6.1 φαίνεται η δομή του project. Ότι περιέχεται στο πακέτο “res”, περιλαμβάνει επέκταση του View.



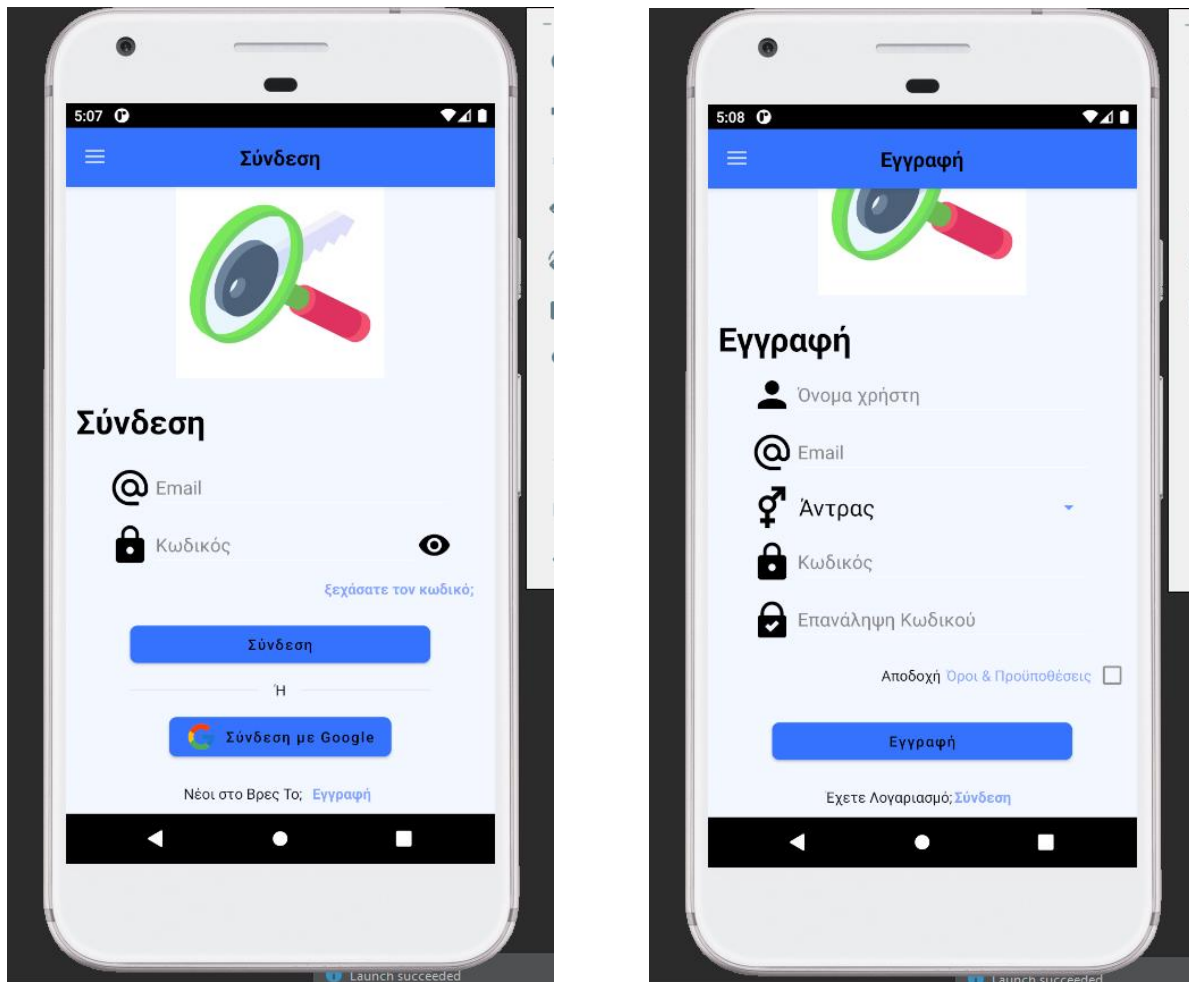
Εικόνα 6.1 Δομή Project

23 Chugh, A. (2022). *Android MVVM Design Pattern*. Ανάκτηση από DigitalOcean: <https://www.digitalocean.com/community/tutorials/android-mvvm-design-pattern>

1.2 Οπτική Σχεδίαση (Front-end)

Με βάση τις λειτουργικές απαιτήσεις που αναλύθηκαν στο κεφάλαιο 4, θα γίνει παρουσίαση και ανάλυση της οπτικής σχεδίασης της εφαρμογής Android. Η γλώσσα της εφαρμογής κατά τη παρουσίαση θα είναι η ελληνική. Κατά την επιλογή αγγλικής γλώσσας δεν υπάρχει καμία οπτική διαφορά εκτός από τη γλώσσα.

Σύνδεση-Εγγραφή Χρήστη:

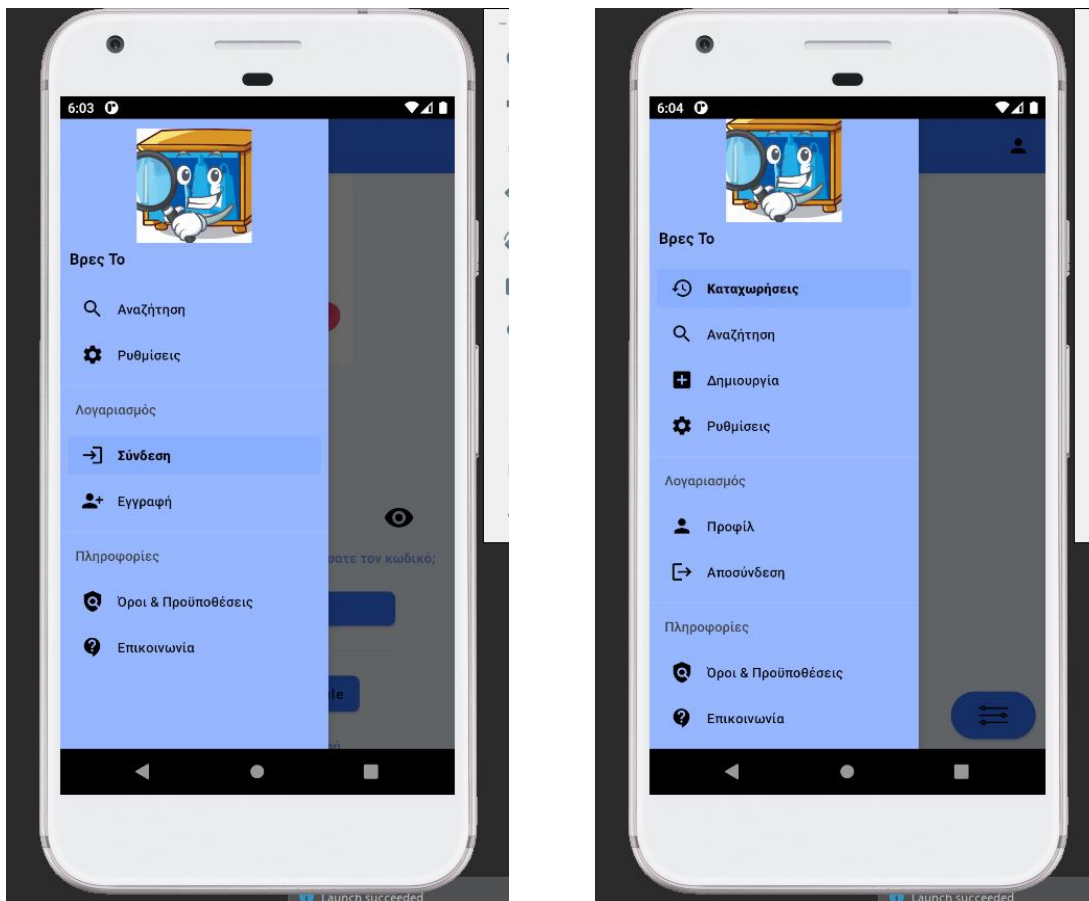


Εικόνα 6.2 Αριστερά η οθόνη σύνδεσης και δεξιά η οθόνη εγγραφής

Για την σύνδεση ή εγγραφή ενός χρήστη υπάρχουν οι οθόνες που φαίνονται στην εικόνα 6.2, οι οποίες μπορούν να εναλλάσσονται μεταξύ των επιλογών «Νέοι στο Βρες Το;» και «Έχετε Λογαριασμό;». Ο χρήστης έχει επιλογή να συνδεθεί με το email και τον κωδικό του αφότου έχει δημιουργήσει λογαριασμό ή να συνδεθεί με έναν λογαριασμό Google. Οι δύο παραπάνω οθόνες φαίνονται μόνο στην περίπτωση που ο χρήστης δεν έχει συνδεθεί ή αποσυνδεθεί από την εφαρμογή.

Κεντρικό Μενού

Στο πάνω αριστερά μέρος της οθόνης υπάρχει το λεγόμενο “hamburger menu” δηλαδή το μενού χάμπουργκερ. Ονομάζεται έτσι διότι οι τρεις οριζόντιες παύλες φαίνονται σαν δύο ψωμάκια και στη μέση το μπιφτέκι του χάμπουργκερ. Με το πάτημα αυτού του κουμπιού, εμφανίζονται όλες οι επιλογές ενός χρήστη στην εφαρμογή. Το μενού αυτό εμφανίζεται σε στυλ συρταριού από τα αριστερά προς τα δεξιά. Οι επιλογές που έχει ένας συνδεδεμένος χρήστης είναι διαφορετικές από έναν μη συνδεδεμένο χρήστη, όπως φαίνεται στην εικόνα 6.3.



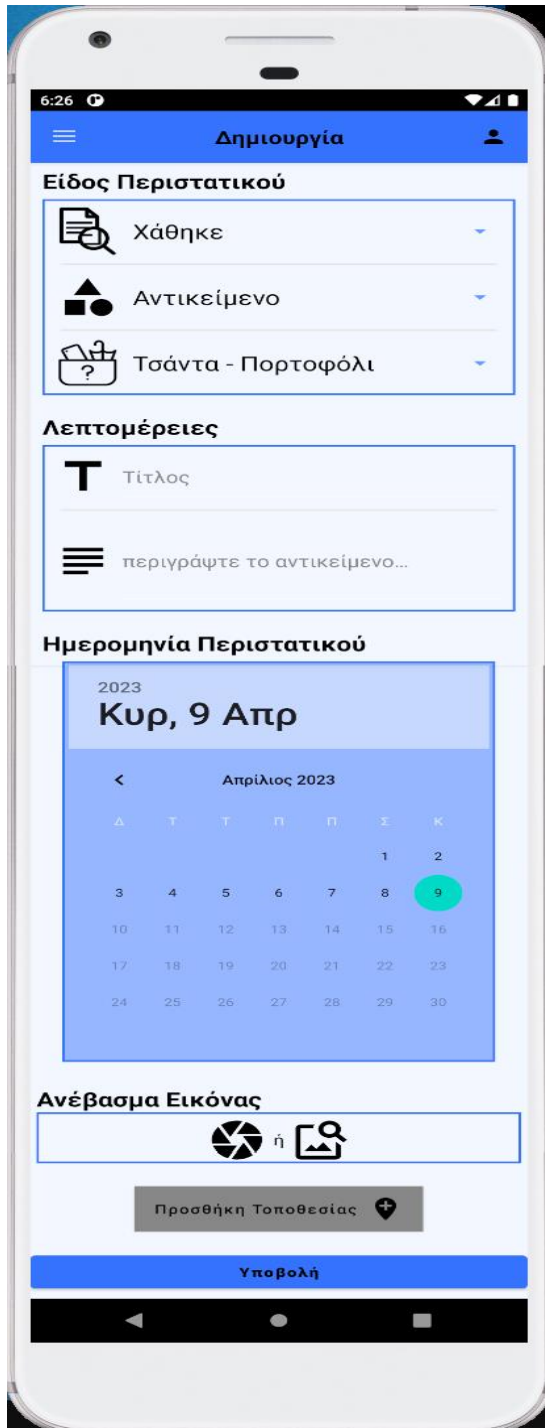
Εικόνα 6.3 Αριστερά το μενού ενός μη συνδεδεμένου χρήστη και δεξιά το μενού ενός συνδεδεμένου χρήστη

Όπως είναι λογικό ο μη συνδεδεμένος χρήστης έχει λιγότερες δυνατότητες από τον συνδεδεμένο χρήστη. Συγκεκριμένα ο συνδεδεμένος χρήστης μπορεί να δει τις καταχωρήσεις του, να δημιουργήσει νέα καταχώρηση, να μεταβεί στο προφίλ του και να αποσυνδεθεί από την εφαρμογή. Ο μη συνδεδεμένος χρήστης μπορεί να συνδεθεί ή να εγγραφεί στην εφαρμογή. Οι κοινές δυνατότητες και των δύο είναι ότι μπορούν να αναζητήσουν αντικείμενα (ο μη συνδεδεμένος χρήστης με περιορισμούς), να μεταβούν στις ρυθμίσεις της εφαρμογής, να δουν τους όρους

και τις προϋποθέσεις και να επικοινωνήσουν με τους διαχειριστές της εφαρμογής μέσω μηνύματος.

Δημιουργία Καταχώρησης

Από την αντίστοιχη επιλογή του μενού μόνο για τους συνδεδεμένους χρήστες, υπάρχει η δυνατότητα να καταχωρήσουν ένα νέο αντικείμενο.



Εικόνα 6.4 Οθόνη δημιουργίας περιστατικού

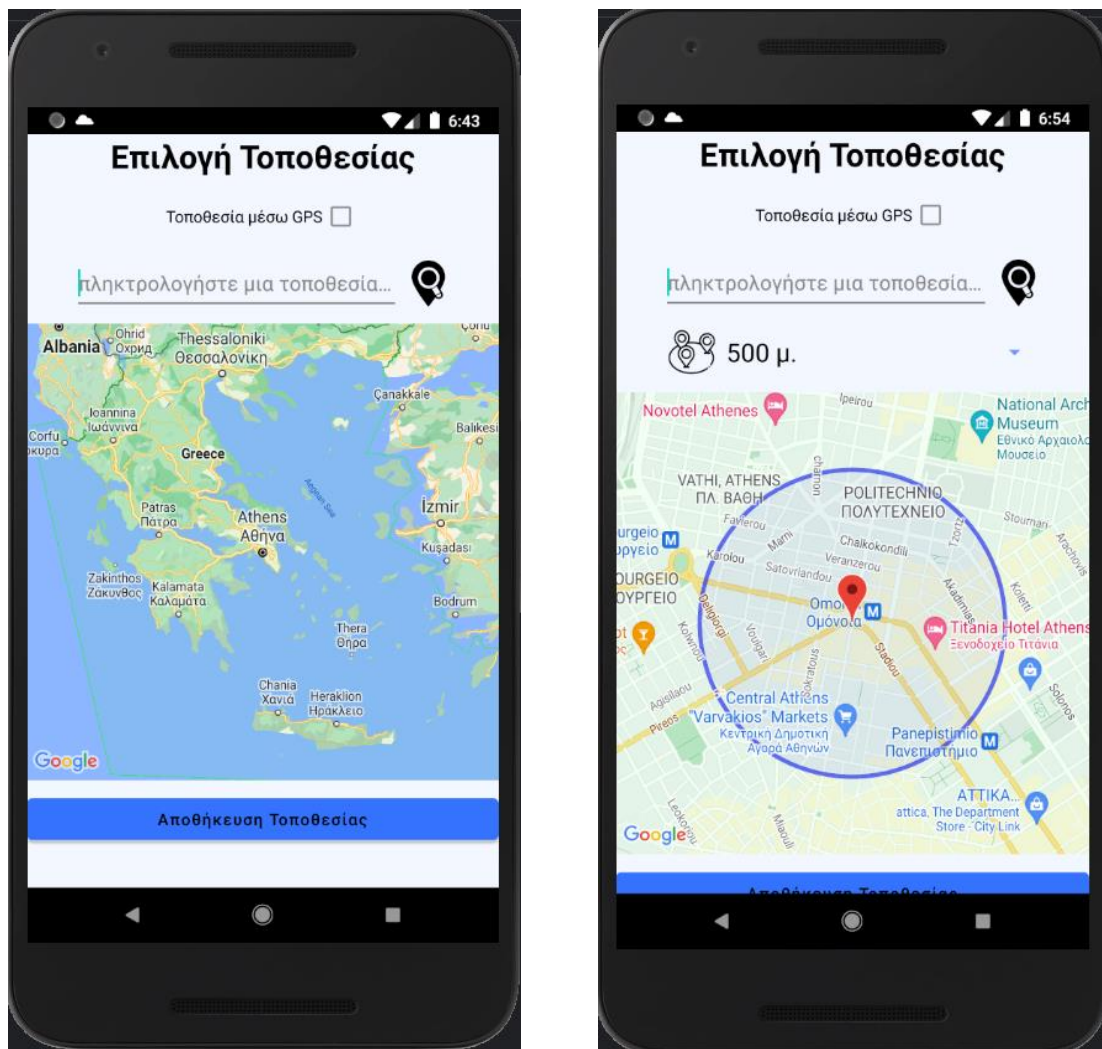
Στην οθόνη δημιουργίας περιστατικού, ο χρήστης πρέπει να συμπληρώσει τα παρακάτω πεδία:

- Κατηγορία Περιστατικού (Χάθηκε - Βρέθηκε)
- Κατηγορία Αντικειμένου (Αντικείμενο – Όχημα – Κατοικίδιο)
- Είδος Αντικειμένου
- Τίτλος Περιστατικού
- Περιγραφή Αντικειμένου
- Ημερομηνία Περιστατικού
- Ανέβασμα εικόνας από την φωτογραφική μηχανή ή από τον αποθηκευτικό χώρο (προαιρετικό)
- Προσθήκη Τοποθεσίας

Η προσθήκη της τοποθεσίας γίνεται σε αναδυόμενο παράθυρο έτσι ώστε να είναι περισσότερο ορατές οι λεπτομέρειες του τόπου. Ο χρήστης μπορεί να επιλέξει το σημείο όπου έγινε το περιστατικό με τρεις τρόπους.

1. Μέσω του αισθητήρα GPS της συσκευής του αφού επιτρέψει την πρόσβαση
2. Με την εισαγωγή της διεύθυνσης και αναζήτηση της τοποθεσίας
3. Με την επιλογή του σημείου πάνω στον χάρτη

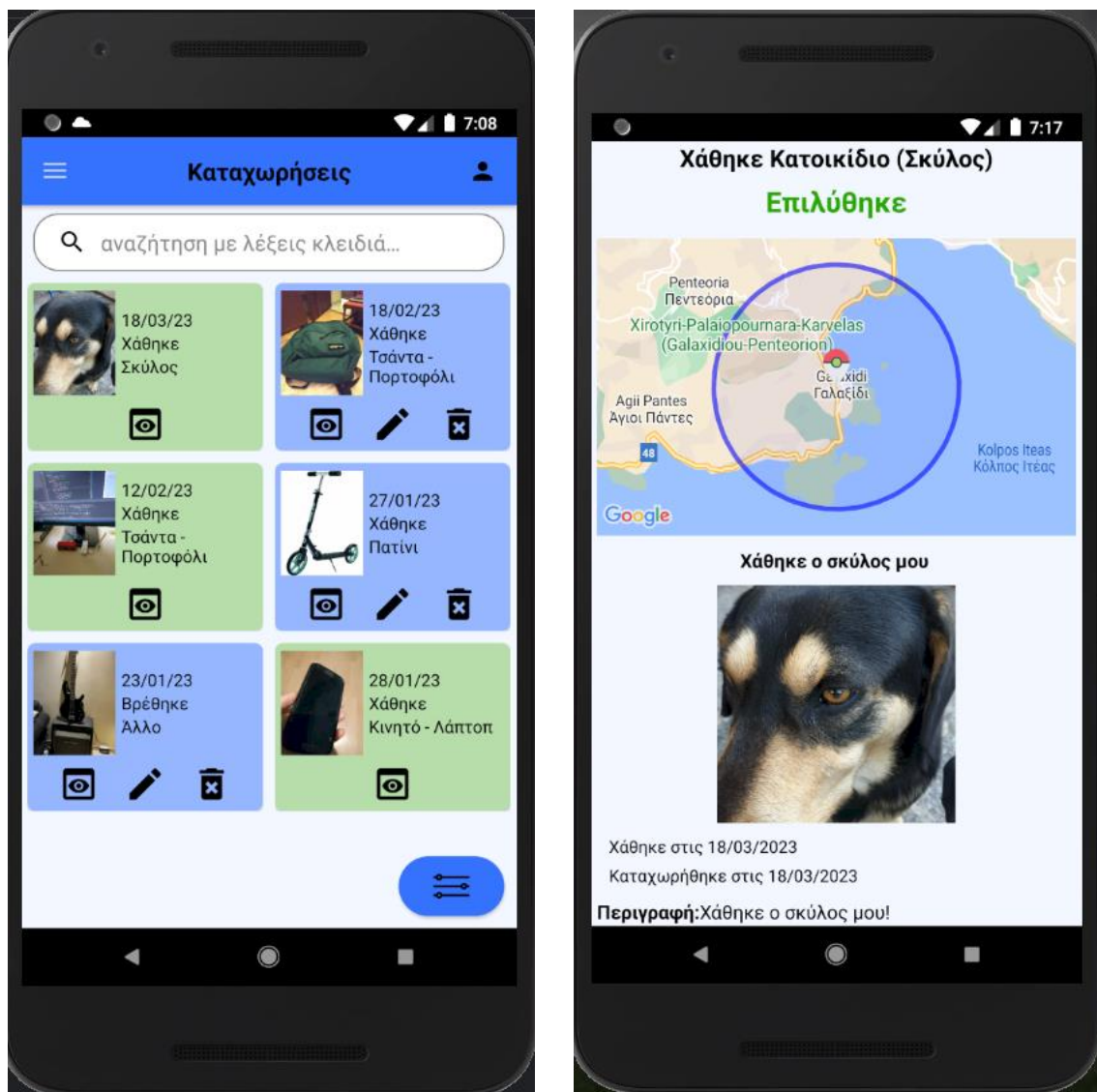
Ακόμα, ο χρήστης μπορεί να επιλέξει την ακτίνα που έγινε το περιστατικό γιατί ενδέχεται να μην γνωρίζει που ακριβώς έχασε το αντικείμενο.



Εικόνα 6.5 Οθόνη επιλογής τοποθεσίας

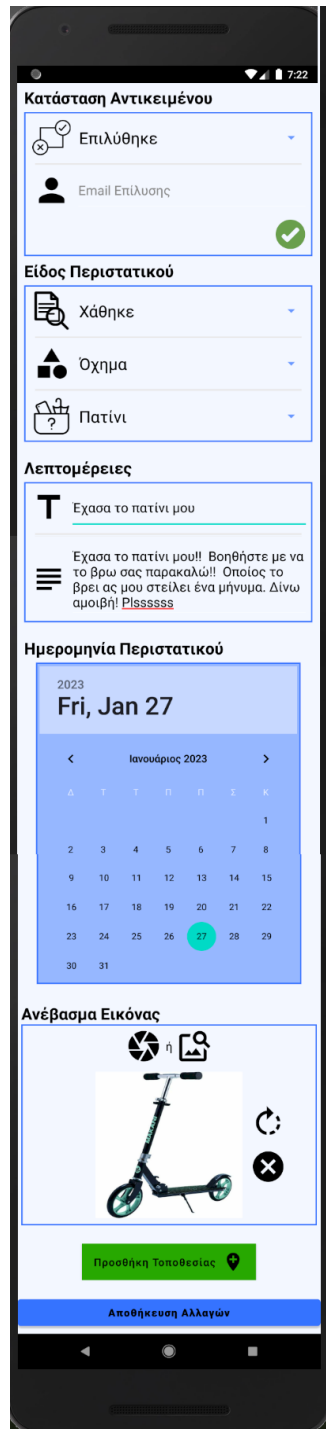
Ιστορικό Καταχωρήσεων

Πρόκειται για την παρουσίαση των ήδη καταχωρημένων αντικειμένων ταξινομημένα σε φθίνουσα σειρά κατά ημερομηνία καταχώρησης. Όπως φαίνεται στην εικόνα 6.6 υπάρχουν ορισμένες καταχωρήσεις με μπλε φόντο και άλλες με πράσινο φόντο. Η καταχωρήσεις με πράσινο φόντο είναι αυτές που έχουν επιλυθεί, δηλαδή τα αντικείμενα αυτά έχουν βρεθεί από τον ιδιοκτήτη τους. Για τον λόγο αυτό, στις καταχωρήσεις που έχουν επιλυθεί, δίνεται η δυνατότητα μόνο προβολής του περιστατικού. Στα μη επιλυμένα περιστατικά υπάρχει η δυνατότητα προβολής, επεξεργασίας και διαγραφής. Επίσης, μπορεί να γίνει αναζήτηση των καταχωρήσεων με λέξεις κλειδιά ή με ορισμένα φίλτρα τα οποία θα αναλυθούν στην λειτουργικότητα της εφαρμογής.



Εικόνα 6.6 Αριστερά οθόνη καταχωρημένων αντικειμένων και δεξιά προβολή καταχώρησης

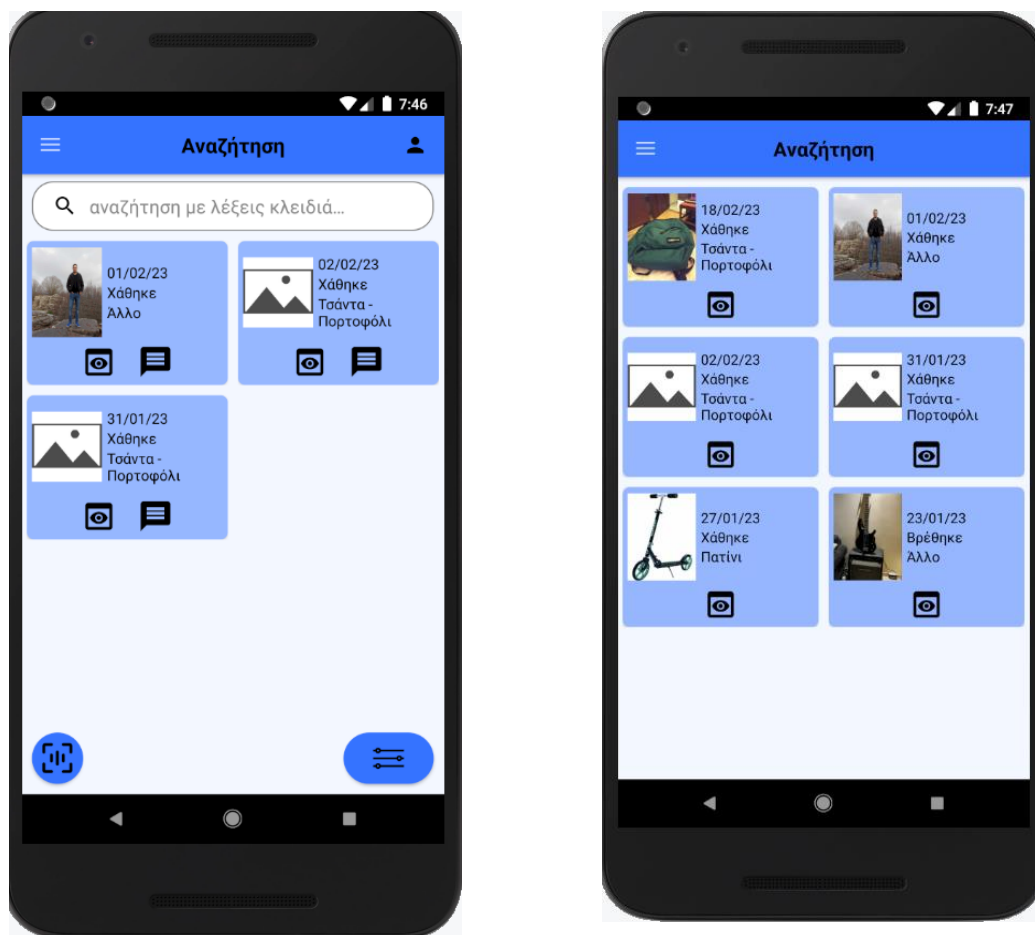
Στην επεξεργασία ενός περιστατικού ο χρήστης έχει τη δυνατότητα να τροποποιήσει όλες τις επιλογές που είχε κατά τη διαδικασία καταχώρησης. Επίσης, μπορεί να επιλέξει την επίλυση του περιστατικού εισάγοντας το email του χρήστη που βρήκε ή επέστρεψε το αντικείμενο.



Εικόνα 6.7 Οθόνη επεξεργασίας αντικειμένου

Αναζήτηση Αντικειμένων

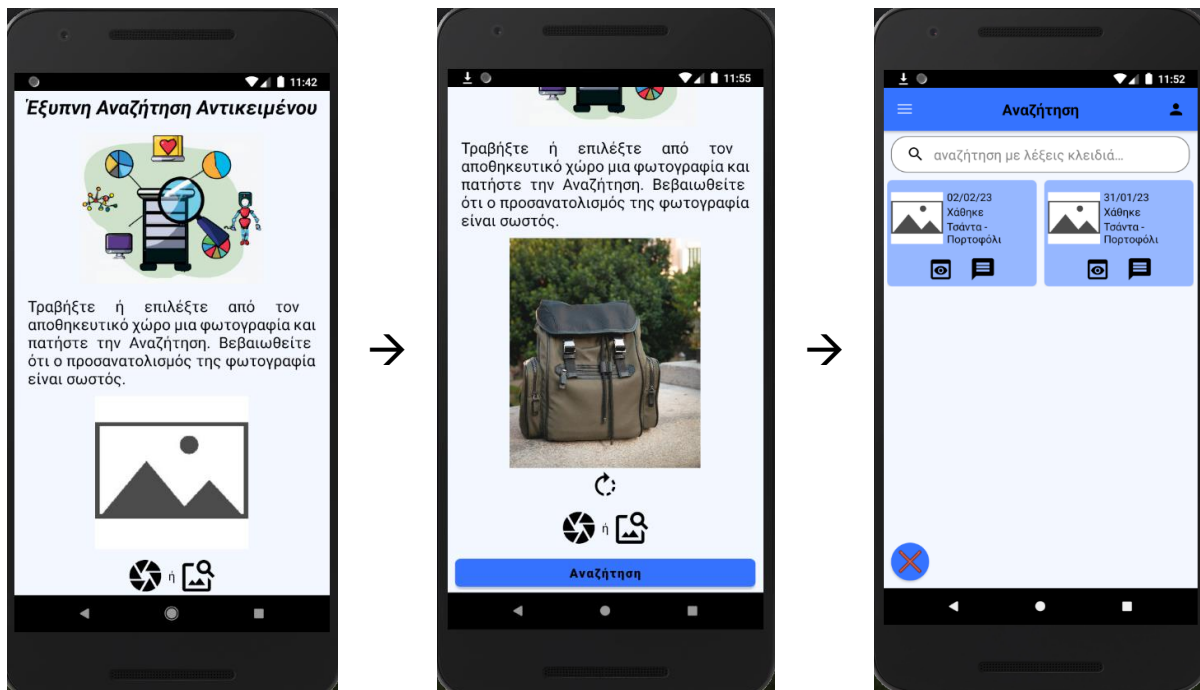
Στην αναζήτηση αντικειμένων φαίνονται τα αντικείμενα που έχουν καταχωρηθεί από άλλους χρήστες. Η οθόνη αυτή μπορεί να προβληθεί και από τους μη συνδεδεμένους χρήστες και από αυτούς που έχουν συνδεθεί με τη μόνη διαφορά ότι οι συνδεδεμένοι χρήστες έχουν περισσότερες δυνατότητες. Συγκεκριμένα, οι μη συνδεδεμένοι χρήστες μπορούν μόνο να προβάλουν τα αντικείμενα των άλλων χρηστών, ενώ οι συνδεδεμένοι χρήστες εκτός από την προβολή, μπορούν να στείλουν προσωπικό μήνυμα στον χρήστη καταχώρησης του αντικειμένου, φιλτράρισμα των αντικειμένων με λέξεις κλειδιά ή ορισμένα φίλτρα και τέλος μπορούν να επιλέξουν την λειτουργικότητα αναζήτησης αντικειμένων εισάγοντας μια εικόνα από τη συσκευή τους με τη χρήση τεχνητής νοημοσύνης (έξυπνη αναζήτηση).



Εικόνα 6.8 Οθόνη αναζήτησης αντικειμένων

Έξυπνη Αναζήτηση Αντικειμένων

Με την έξυπνη αναζήτηση αντικειμένου, ο χρήστης εισάγει μια εικόνα από τον αποθηκευτικό χώρο της συσκευής ή απευθείας με λήψη φωτογραφίας από την κάμερα και αφού πατήσει το κουμπί αναζήτησης, του εμφανίζονται όλα τα αντικείμενα που σχετίζονται με την εικόνα που επέλεξε. Αξίζει να σημειωθεί ότι ο χρήστης μπορεί να περιστρέψει την εικόνα σε περίπτωση που για κάποιο λόγο αυτή είναι αναστραμμένη. Στην εικόνα 6.9 φαίνονται οι οθόνες της διαδικασίας έξυπνης αναζήτησης αντικειμένου.



Εικόνα 6.9 Οθόνες έξυπνης αναζήτησης αντικειμένου

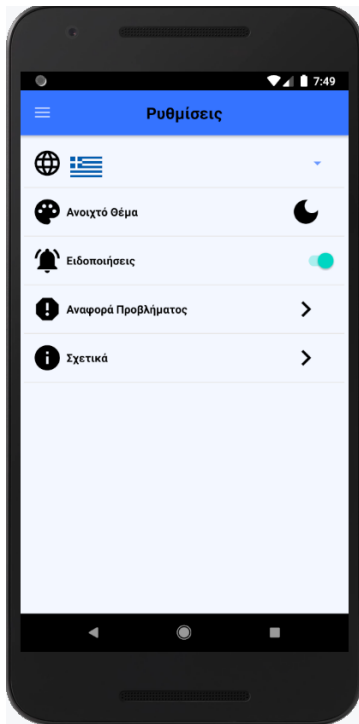
Ρυθμίσεις

Οι ρυθμίσεις της εφαρμογής είναι ορατές και κοινές για όλους τους χρήστες. Στις ρυθμίσεις κάθε χρήστης μπορεί να αλλάξει τις εξής επιλογές:

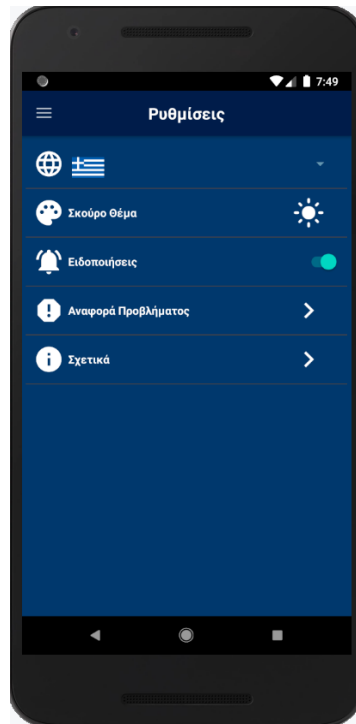
- Γλώσσα
- Χρωματικό Θέμα
- Ενεργοποίηση Ειδοποιήσεων
- Αναφορά Προβλήματος
- Σχετικά με την Εφαρμογή

Συγκεκριμένα, οι διαθέσιμες γλώσσες της εφαρμογής είναι ελληνικά, αγγλικά και ισπανικά. Υπάρχει ανοιχτό (φωτεινό) και σκούρο θέμα ανάλογα με τις προτιμήσεις του χρήστη (Εικόνα 6.11). Ο διακόπτης των ειδοποιήσεων αφορά τις ειδοποιήσεις μηνυμάτων που λαμβάνονται από άλλους χρήστες ακόμα και αν λαμβάνονται χωρίς να εκτελείται η εφαρμογή. Οι χρήστες μπορούν να

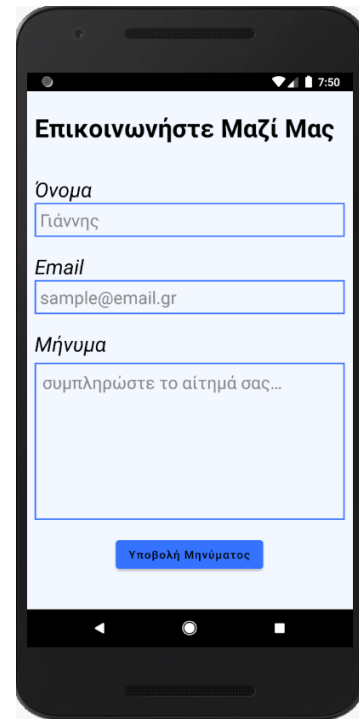
συμπληρώσουν μια φόρμα ώστε να αναφέρουν στους διαχειριστές της εφαρμογής ένα πρόβλημα (Εικόνα 6.12).. Τέλος, υπάρχει το κουμπί «Σχετικά» στο οποίο μπορούν οι χρήστες να μάθουν πληροφορίες για την εφαρμογή.



Εικόνα 6.10 Ρυθμίσεις



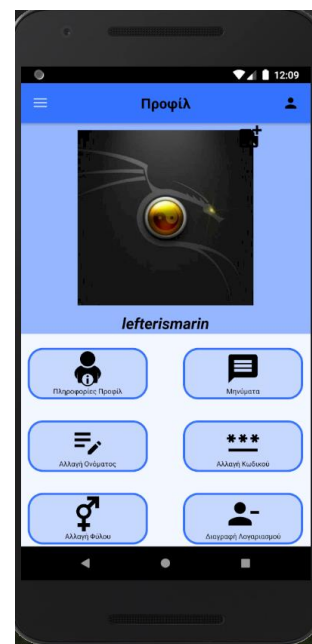
Εικόνα 6.11 Σκούρο θέμα



Εικόνα 6.12 Αναφορά Προβλήματος

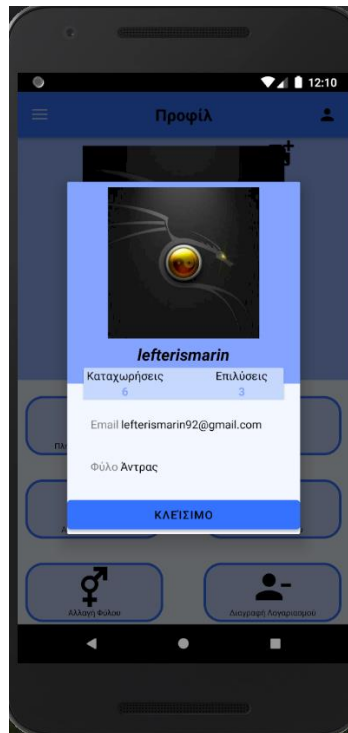
Προφίλ

Στο προφίλ του χρήστη όπως είναι αναμενόμενο, μπορούν να έχουν πρόσβαση μόνο οι συνδεδεμένοι χρήστες. Η πρόσβαση στο προφίλ μπορεί να γίνει μέσω του βασικού μενού ή από το εικονίδιο σε σχήμα ανθρώπου που εμφανίζεται στο πάνω δεξιά μέρος. Από αυτή την οθόνη μπορούν οι χρήστες να επεξεργαστούν τις βασικές τους πληροφορίες όπως τροποποίηση εικόνας προφίλ, αλλαγή ονόματος χρήστη, αλλαγή κωδικού πρόσβασης, αλλαγή επιλεγμένου φύλου και διαγραφή λογαριασμού.



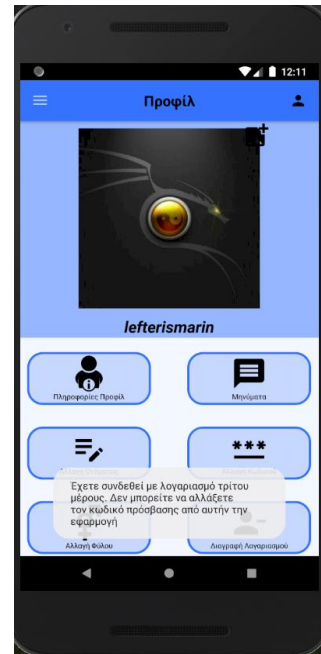
Εικόνα 6.13 Προφίλ

Επίσης, υπάρχει η δυνατότητα προβολής των βασικών στοιχείων του λογαριασμού όπως φαίνεται στην εικόνα 6.14. Στη συγκεκριμένη οθόνη, εκτός από τα βασικά στοιχεία, εμφανίζεται και ο συνολικός αριθμός καταχωρήσεων καθώς και ο αριθμός επιλύσεων υποθέσεων άλλων χρηστών. Ο τελευταίος αριθμός είναι ο δείκτης εμπιστοσύνης ενός χρήστη καθώς φαίνεται πόσους άλλους χρήστες έχει βοηθήσει κάποιος στο να βρεθεί κάποιο αντικείμενο.



Εικόνα 6.14 Στοιχεία συνδεδεμένου χρήστη

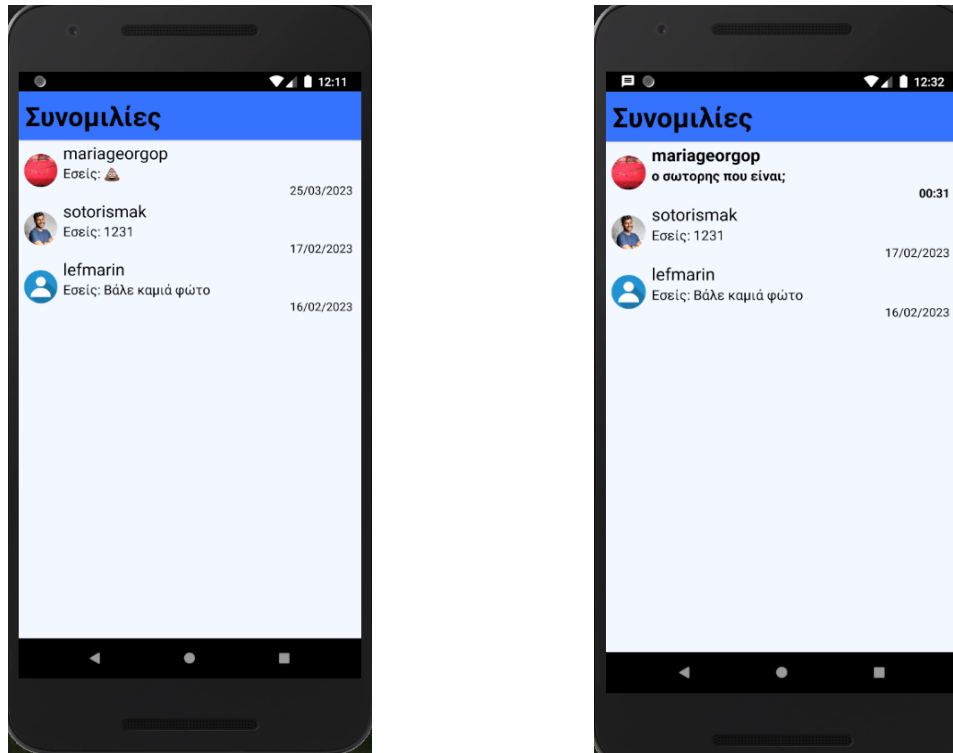
Τέλος, αξίζει να σημειωθεί ότι οι χρήστες που έχουν συνδεθεί στην εφαρμογή με λογαριασμό Google, δεν έχουν τη δυνατότητα αλλαγής του κωδικού πρόσβασης καθώς αυτός αφορά μόνο την Google και δεν απαιτείται για την είσοδο στην εφαρμογή (Εικόνα 6.15). Ακόμα, κατά την διάρκεια διαγραφής ενός χρήστη ο οποίος έχει συνδεθεί με έναν Google λογαριασμό, όπως είναι λογικό θα διαγραφούν μόνο οι πληροφορίες που έχουν αποθηκευτεί στην εφαρμογή χωρίς αυτό να επηρεάσει με κάποιον τρόπο τον λογαριασμό Google του χρήστη.



Εικόνα 6.15 Διαγραφή λογαριασμού

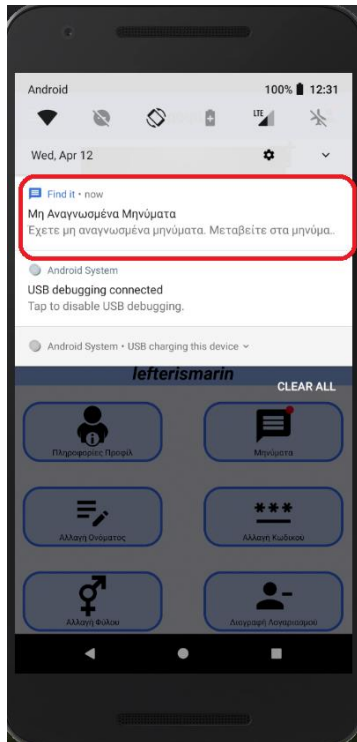
Μηνύματα

Το βασικό μέσο επικοινωνίας μεταξύ των χρηστών εντός της εφαρμογής είναι τα προσωπικά μηνύματα. Από τα μηνύματα ο χρήστης που έχει καταχωρήσει ένα αντικείμενο ως χαμένο ή ανευρεθέν θα μπορεί να έρθει σε επικοινωνία με τον χρήστη που θα εντοπίσει ή θα είναι ιδιοκτήτης του αντικειμένου αυτού αντίστοιχα. Η πρόσβαση στις συνομιλίες γίνεται μέσω του προφίλ. Μπορεί όμως κάποιος χρήστης να στείλει απευθείας μήνυμα σε κάποιον άλλο μέσω ενός καταχωρημένου αντικειμένου. Αφού σταλεί αυτό το μήνυμα, πλέον η συνομιλία θα εμφανίζεται στην αντίστοιχη λίστα συνομιλιών. Οι συνομιλίες μπορούν να διαγραφούν με το συνεχόμενο πάτημα πάνω σε μια από αυτές. Επίσης, όταν τουλάχιστον ένα μήνυμα δεν έχει διαβαστεί από τον χρήστη, τα στοιχεία της συνομιλίας θα εμφανίζονται με έντονη γραφή (Εικόνα 6.16).

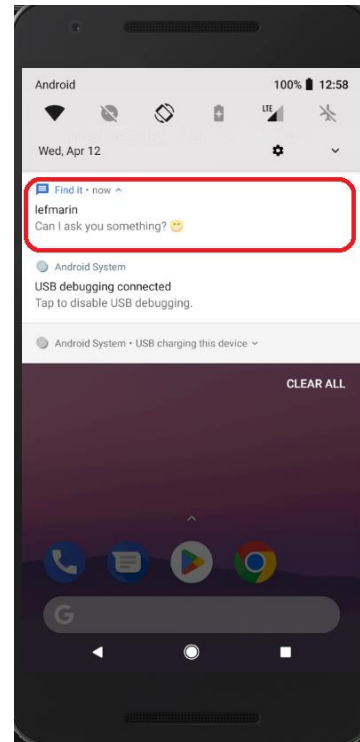


Εικόνα 6.16 Οθόνη συνομιλιών μεταξύ του συνδεδεμένου χρήστη και άλλων χρηστών χωρίς/με μη διαβασμένες συνομιλίες

Όταν αποστέλλεται ένα μήνυμα σε έναν άλλο χρήστη, αυτός πρέπει να ειδοποιείται με κάποιον τρόπο. Κάθε φορά λοιπόν που κάποιος χρήστης λαμβάνει ένα μήνυμα, εμφανίζεται μια σχετική ειδοποίηση στο κινητό του. Αυτή η ειδοποίηση θα εμφανιστεί ακόμα και αν ο παραλήπτης του μηνύματος δεν εκτελεί εκείνη τη στιγμή την εφαρμογή. Στην εικόνα 6.17 φαίνεται η ειδοποίηση που θα εμφανιστεί στον χρήστη ο οποίος εκτελεί εκείνη τη χρονική στιγμή την εφαρμογή και στην εικόνα 6.18 φαίνεται η ειδοποίηση που θα εμφανιστεί αν ο χρήστης δεν εκτελεί την εφαρμογή.



Εικόνα 6.17 Ειδοποίηση εντός εφαρμογής



Εικόνα 6.18 Ειδοποίηση εκτός εφαρμογής

Σε περίπτωση που ο χρήστης έχει μη αναγνωσμένα μηνύματα στις συνομιλίες του αλλά έχει διαγράψει τις ειδοποιήσεις, θα του εμφανίζεται μια κόκκινη παλλόμενη ένδειξη τόσο στο εικονίδιο με το ανθρωπάκι πάνω δεξιά της εφαρμογής όσο και στο εικονίδιο των μηνυμάτων εντός του προφίλ (Εικόνα 6.19)



Εικόνα 6.19 Ειδοποίηση εντός εφαρμογής

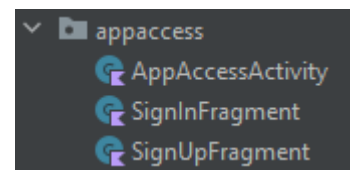
1.3 Λειτουργική Σχεδίαση (Back-end)

1.3.1 Σύνδεση Βάσης Δεδομένων

Όπως έχει αναφερθεί σε προηγούμενο κεφάλαιο, η βάση δεδομένων που θα χρησιμοποιηθεί είναι η Firebase της Google. Για την σύνδεση της βάσης δεδομένων στο project της εφαρμογής, πρέπει να συνδεθεί ο προγραμματιστής σε λογαριασμό της εφαρμογής Firebase και να τοποθετηθούν οι κατάλληλες εξαρτήσεις (dependencies) τόσο για την αυθεντικοποίηση των χρηστών όσο και για την καταχώρηση δεδομένων όπως τα περαιτέρω στοιχεία των χρηστών και τις πληροφορίες των αντικειμένων. Η διαδικασία σύνδεσης είναι πολύ εύκολη καθώς το πρόγραμμα Android Studio προσφέρει αυτοματοποιημένο σύστημα για το συγκεκριμένο λόγο.

1.3.2 Σύνδεση-Εγγραφή Χρήστη

Για την σύνδεση ή εγγραφή χρήστη, χρησιμοποιήθηκε ένα Activity το οποίο περιέχει δύο Fragments. Τα Fragments είναι αυτά που διαχειρίζονται την σύνδεση και την εγγραφή του χρήστη αντίστοιχα ενώ το Activity υποστηρίζει τα δύο Fragments με τις βασικές λειτουργικότητες.



Εικόνα 6.20 Σύνδεση χρηστών

Για την σύνδεση στην εφαρμογή ο χρήστης πρέπει να εισάγει το email και τον κωδικό του ώστε να γίνει επαλήθευση των στοιχείων. Κατά την πληκτρολόγηση του κωδικού του, μπορεί να πατήσει το εικονίδιο με το μάτι το οποίο κατά το πάτημά του, εμφανίζει ή εξαφανίζει τον κωδικό. Επίσης, αν ο χρήστης σταματήσει την εφαρμογή χωρίς να αποσυνδεθεί, την επόμενη φορά που θα εκτελέσει την εφαρμογή δεν θα χρειάζεται να ολοκληρώσει εκ νέου την διαδικασία σύνδεσης.

Τέλος, ο χρήστης μέσω του κουμπιού «Νέοι στο Βρες Το;» μπορεί να μεταβεί στην εγγραφή χρήστη και μέσω του κουμπιού «Ξεχάσατε τον κωδικό;» αφού συμπληρώσει το email του του αποστέλλεται email στην ηλεκτρονική διεύθυνση με τον σύνδεσμο για επαναφορά του κωδικού πρόσβασης.

Η διαδικασία εγγραφής του χρήστη είναι περισσότερο απλή αφού το μόνο που χρειάζεται είναι να συμπληρώσει τα απαιτούμενα στοιχεία, να πατήσει την επιλογή ότι αποδέχεται τους όρους και τις προϋποθέσεις (αφού τους διαβάσει) και να πατήσει το κουμπί Εγγραφή.

1.3.3 Καταχώρηση και Προβολή Αντικειμένων

Για την καταχώρηση ενός αντικειμένου, απαιτείται να έχουν συμπληρωθεί τα πεδία Τίτλος, Περιγραφή και Τοποθεσία περιστατικού. Επίσης, είναι απαραίτητα και τα πεδία Κατηγορία περιστατικού (Χάθηκε – Βρέθηκε), Κατηγορία Αντικειμένου (Αντικείμενο – Όχημα – Κατοικίδιο), Είδος Αντικειμένου ανάλογα την κατηγορία αντικειμένου που επέλεξε και την ημερομηνία του περιστατικού. Ακόμα συνιστάται στον χρήστη να προσθέσει μια εικόνα του αντικειμένου ώστε να διευκολύνει τον εντοπισμό του. Ως προς την τοποθεσία, ο χρήστης έχει τη δυνατότητα να επιλέξει την τοποθεσία του περιστατικού με τρεις τρόπους. Πατώντας στο σημείο χάρτη που εμφανίζεται στην οθόνη του, πληκτρολογώντας την τοποθεσία και πατώντας αναζήτηση αυτής και μέσω GPS όπου γίνεται εύρεση της τρέχουσας τοποθεσίας της συσκευής. Τέλος, είναι δυνατή η επιλογή εύρους τοποθεσίας και όχι αυστηρά ενός συγκεκριμένου σημείου.

Στο ιστορικό καταχωρήσεων εμφανίζονται όλες οι καταχωρήσεις που έχει πραγματοποιήσει ο συνδεδεμένος χρήστης όπου για κάθε καταχώρηση του δίνεται η δυνατότητα να κάνει προβολή περισσότερων λεπτομερειών, επεξεργασία αυτής ή διαγραφή της. Κατά τη προβολή εμφανίζονται όλες οι καταχωρημένες πληροφορίες του αντικειμένου καθώς και το σημείο αυτού πάνω στον χάρτη. Επίσης, υπάρχει δυνατότητα να φιλτράρει τα αντικείμενα με λέξεις κλειδιά ή με άλλα διαθέσιμα φίλτρα. Τα φίλτρα αυτά μπορεί να είναι ένα ή περισσότερα από τα παρακάτω:

- Επιλογή Κατηγορίας Περιστατικού (Χάθηκε ή Βρέθηκε)
- Επιλογή Είδους ανά κατηγορία (π.χ. Ταυτότητα και Τιμαλφή)
- Ημερομηνία Περιστατικού Από ή/και Ημερομηνία Περιστατικού Έως
- Τοποθεσία Περιστατικού εντός ενός επιλεγμένου εύρους

1.3.4 Αναζήτηση και Προβολή Αντικειμένων

Η αναζήτηση των αντικειμένων πρόκειται για αντικείμενα τα οποία έχουν καταχωρηθεί από άλλους χρήστες. Είναι η βασική λειτουργικότητα της εφαρμογής και ο λόγος που καθιστά την συγκεκριμένη εφαρμογή πρωτοπόρα. Οι χρήστες μπορούν να προβάλουν τις πληροφορίες για κάθε αντικείμενο καθώς και να στείλουν προσωπικό μήνυμα στον χρήστη καταχώρησης. Έχουν την ικανότητα να φιλτράρουν τα αντικείμενα με τους ίδιους τρόπους που μπορούν να το κάνουν στα δικά τους καταχωρημένα αντικείμενα με τη μόνη διαφορά ότι θα πρέπει να επιλέξουν το κουμπί «Εφαρμογή Φίλτρων».

Ο τρόπος με τον οποίο αναζητούνται τα αντικείμενα σε αυτή τη περίπτωση είναι διαφορετικός από αυτόν της προηγούμενης παραγράφου. Συγκεκριμένα, το αίτημα του χρήστη δεν επεξεργάζεται στο κινητό του αλλά αποστέλλεται στο Firebase Cloud Functions που αναφέρθηκε στην Ενότητα 5 παράγραφος 1.5. Αναλυτικότερα, το αίτημα του χρήστη περιλαμβάνει όλα τα

επιλεγμένα φίλτρα και αφού αποσταλούν στο Cloud Functions αυτό επεξεργάζεται και επιστρέφει μόνο τα αντικείμενα που προκύπτουν μετά την εφαρμογή των φίλτρων.

1.3.5 Έξυπνη Αναζήτηση Αντικειμένων

Στην έξυπνη αναζήτηση αντικειμένων, ο χρήστης μπορεί να επιλέξει μια εικόνα από τον αποθηκευτικό χώρο της συσκευής ή να τραβήξει μια φωτογραφία και αφού πατήσει το κουμπί «Αναζήτηση», μέσω της βιβλιοθήκης MLKit, θα βρεθούν παρόμοια αντικείμενα με αυτό της εικόνας που εισήγαγε. Ουσιαστικά το MLKit ανιχνεύει με τη βοήθεια της τεχνητής νοημοσύνης και ενός ήδη εκπαιδευμένου μοντέλου, τα αντικείμενα που υπάρχουν σε μια εικόνα επιστρέφοντας τις λέξεις που βρέθηκαν.

Ο κώδικας έχει μετατραπεί με κατάλληλο τρόπο ώστε να συμπληρώνονται όλα τα είδη αντικειμένων που αντιστοιχίζονται με αυτά του MLKit. Έτσι όταν επιστραφούν ορισμένες λέξεις και μετατραπούν σε υπάρχοντα είδη αντικειμένων της εφαρμογής, ακολουθείται η ίδια διαδικασία με την αναζήτηση αντικειμένων που επεξηγήθηκε στην προηγούμενη παράγραφο. Δηλαδή εισάγονται τα κατάλληλα φίλτρα και αποστέλλεται το αίτημα στο Firebase Cloud Functions για να επιστραφούν τα αντικείμενα που ταιριάζουν.

1.3.6 Προσωπικά Μηνύματα Μεταξύ Χρηστών

Κάθε συνδεδεμένος χρήστης μπορεί να ανταλλάξει προσωπικά μηνύματα με κάθε άλλον χρήστη για τις ανάγκες επικοινωνίας των χρηστών. Ο αποστολέας κάθε μηνύματος έχει τη δυνατότητα να διαγράψει ένα μήνυμά του και όταν το κάνει αυτό δεν θα φαίνεται ούτε στον παραλήπτη του. Το κάθε μήνυμα αν δεν διαγραφεί από τον αποστολέα, διαγράφεται μετά την παρέλευση 30 ημερών. Το περιεχόμενο των μηνυμάτων είναι κρυπτογραφημένο από άκρη σε άκρη (end-to-end encrypted) με τον αλγόριθμο κρυπτογράφησης AES-CBC με μέγεθος block 128bits.

Μέσω του προφίλ ενός χρήστη, μπορεί να γίνει μετάβαση στις συνομιλίες που έχει με άλλους χρήστες. Σε περίπτωση που έχει αδιάβαστα μηνύματα, οι πληροφορίες της συνομιλίας θα εμφανίζονται με έντονη γραφή. Οι συνομιλίες ταξινομούνται πρώτα με αν έχουν διαβαστεί ή όχι και μετά με την ημερομηνία αποστολής του τελευταίου μηνύματος είτε από τον αποστολέα είτε από τον παραλήπτη. Ο κάθε χρήστης έχει τη δυνατότητα να διαγράψει μια συνομιλία χωρίς πλέον να φαίνεται στην λίστα συνομιλιών του, όμως με αυτόν τον τρόπο τα μηνύματα δεν θα εξαφανιστούν και από τον συνομιλητή του. Στην παραπάνω περίπτωση αν και ο συνομιλητής διαγράψει την συνομιλία, τότε τα μηνύματα θα διαγραφούν μόνιμα. Μετά από κάθε διαγραφή συνομιλίας ή μηνύματος, δεν είναι δυνατή η ανάκτησή της/του.

Ειδοποιήσεις κατά την λήψη μηνύματος αποστέλλεται στον παραλήπτη ανεξάρτητα από το αν εκτελεί εκείνη τη χρονική στιγμή την εφαρμογή ή όχι. Αυτό γίνεται μέσω των ειδοποιήσεων της εφαρμογής αν εκτελείται αυτή ενώ αν δεν εκτελείται, αποστέλλεται ειδοποίηση μέσω της υπηρεσίας Firebase Cloud Messaging (FCM). Επίσης, όταν κάποιο μήνυμα είναι αδιάβαστο εμφανίζεται στο εικονίδιο του προφίλ πάνω δεξιά της εφαρμογής και στο εικονίδιο μηνυμάτων εντός του προφίλ, μια παλλόμενη κόκκινη κουκίδα.

1.3.7 *Firestore Cloud Functions*

Επιλέχθηκε αυτή η υπηρεσία της Firebase διότι σε βάθος χρόνου, όταν οι καταχωρήσεις αντικειμένων και τα μηνύματα που αποστέλλουν οι χρήστες είναι εκατοντάδες ή χιλιάδες, η επεξεργασία αυτών στην συσκευή των χρηστών θα είναι απαγορευτική. Αντίθετα τα Cloud Functions έχουν σταθερή και γρήγορη απόκριση στα αιτήματα των χρηστών ανεξαρτήτου συσκευής και μπορούν να ικανοποιούν χωρίς ιδιαίτερο πρόβλημα κάθε αίτημα. Επίσης, επειδή πρόκειται για cloud υπηρεσία της Google, σημαίνει ότι η συντήρηση και η κλιμάκωση των αιτημάτων δεν εμπίπτουν στις υποχρεώσεις των προγραμματιστών αλλά της ίδιας της εταιρείας Google.

Έχουν κατασκευαστεί συναρτήσεις για τις παρακάτω λειτουργικότητες:

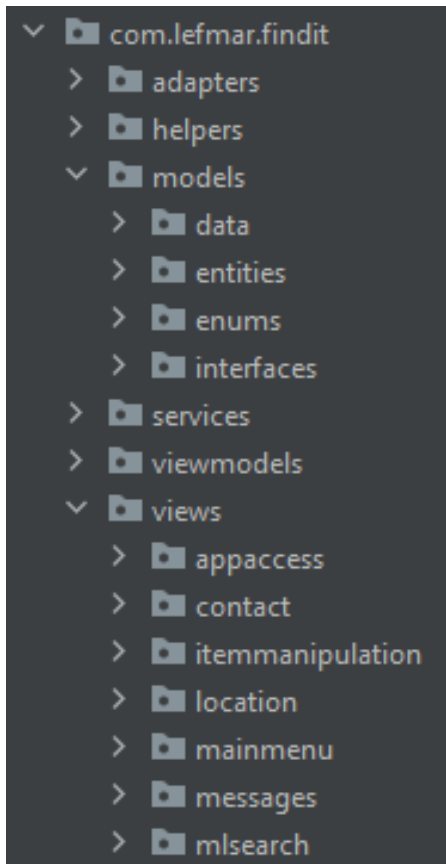
- Φιλτράρισμα Αντικειμένων
- Ανάκτηση όλων των συνομιλιών ενός χρήστη
- Διαγραφή μιας συνομιλίας
- Διαγραφή μηνυμάτων που έχουν αποσταλεί σε χρονικό διάστημα μεγαλύτερο των 30 ημερών από την σημερινή
- Ορισμός μηνυμάτων μιας συνομιλίας ως αναγνωσμένα
- Αποστολή ειδοποίησης νέου μηνύματος μέσω FCM Token

Όπως έχει ήδη ειπωθεί, τα Cloud Functions έχουν δημιουργηθεί σε ξεχωριστό project από αυτό της android εφαρμογής. Ο πηγαίος κώδικας αναπτύχθηκε στο IDE Visual Studio Code v.1.75.1 και με γλώσσα προγραμματισμού Typescript v.4.5.4.²⁴ Τέλος, τα cloud functions εκτελούνται στον Europe-West-1 server της firebase ο οποίος βρίσκεται στο Βέλγιο και είναι ο κοντινότερος server για την Ελλάδα.²⁵

²⁴ Microsoft. *Visual Studio Code in Action*. Ανάκτηση από Visual Studio Code: <https://code.visualstudio.com/docs>

²⁵ Google. *Cloud locations*. Ανάκτηση από Google Cloud: <https://cloud.google.com/about/locations#europe>

25.1.1 Δομή

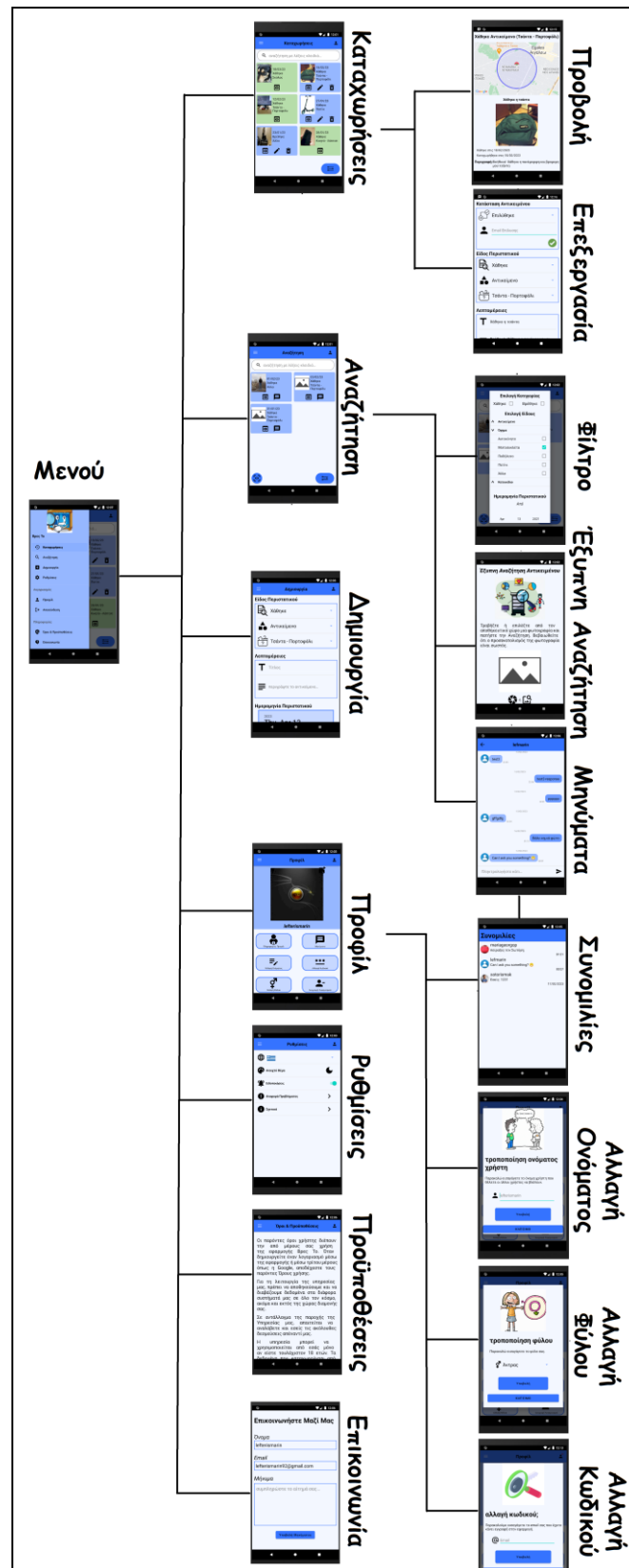


Εικόνα 6.21 Δομή android project

Η δομή του project είναι αυτή που εμφανίζεται στην εικόνα 6.21

- **Adapters:** περιλαμβάνονται οι προσαρμογείς οι οποίοι έχουν σχεδιαστεί για τις ανάγκες της εφαρμογής.
- **Helpers:** υπάρχουν οι βοηθητικές κλάσεις
- **Models:** Η δομή των αντικειμένων που περιέχουν μεταβλητές για την ευκολότερη διαχείριση των δεδομένων
- **Services:** Επέκταση του Model στην MVVM αρχιτεκτονική ως προς την αποστολή και λήψη πληροφοριών από τις Firebase υπηρεσίες
- **ViewModels:** Τα ViewModels της αρχιτεκτονικής MVVM
- **Views:** Τα Views της αρχιτεκτονικής MVVM τα οποία περιλαμβάνουν Activities και Fragments

25.2 Διάγραμμα πιθανών διαδρομών (Συνδεδεμένου Χρήστη)



Εικόνα 6.22 Διάγραμμα πιθανών διαδρομών

ΚΕΦΑΛΑΙΟ 7^ο: ΣΥΜΠΕΡΑΣΜΑΤΑ

Συμπερασματικά, η ανάπτυξη μιας εφαρμογής Android για την προσθήκη χαμένων ή ανευρεθέντων αντικειμένων αντιπροσωπεύει μια σημαντική πρόοδο στην τεχνολογία που έχει τη δυνατότητα να επηρεάσει θετικά πολλά άτομα και οργανισμούς. Με την αυξανόμενη επικράτηση των smartphones και της τεχνολογίας κινητής τηλεφωνίας, οι άνθρωποι αναζητούν πιο αποτελεσματικούς και βολικούς τρόπους για να αναφέρουν αντικείμενα που χάθηκαν ή βρέθηκαν. Αυτή η εφαρμογή παρέχει μια λύση σε αυτό το πρόβλημα αξιοποιώντας τη δύναμη της τεχνολογίας στον τομέα της κινητής τηλεφωνίας για να κάνει τη διαδικασία αναφοράς και ανάκτησης χαμένων ή ανευρεθέντων αντικειμένων πιο απρόσκοπτη και αποτελεσματική.

Οι δυνατότητες της εφαρμογής, όπως η γεωγραφική τοποθεσία και η προσθήκη εικόνων, επιτρέπουν στους χρήστες να παρέχουν λεπτομερείς πληροφορίες σχετικά με το χαμένο ή το ανευρεθέν αντικείμενο, αυξάνοντας τις πιθανότητες ανάκτησής του. Επιπλέον, το Cloud σύστημα διαχείρισης βάσεων δεδομένων της εφαρμογής επιτρέπει την εύκολη παρακολούθηση και αντιστοίχιση χαμένων και ανευρεθέντων αντικειμένων, απλοποιώντας τη διαδικασία και μειώνοντας τον χρόνο και την προσπάθεια που απαιτείται για την επανένωση των αντικειμένων με τους κατόχους τους.

Η δυνατότητα επεκτασιμότητας και προσαρμοστικότητας αυτής της εφαρμογής είναι σημαντική, καθώς μπορεί να προσαρμοστεί για να καλύψει τις συγκεκριμένες ανάγκες διαφόρων οργανισμών, όπως σχολεία, αεροδρόμια και συστήματα δημόσιων μεταφορών. Με τη φιλική προς το χρήστη διεπαφή της εφαρμογής, ο καθένας μπορεί εύκολα να αναφέρει ένα χαμένο ή ανευρεθέν αντικείμενο, καθιστώντας το προσβάσιμο σε ένα ευρύ φάσμα ατόμων, ανεξάρτητα από την τεχνολογική τους εμπειρία.

Συνοπτικά, η ανάπτυξη αυτής της εφαρμογής Android για την προσθήκη χαμένων ή ανευρεθέντων αντικειμένων έχει τη δυνατότητα να φέρει επανάσταση στον τρόπο με τον οποίο αναφέρουμε και ανακτούμε τα στοιχεία που χάθηκαν ή βρέθηκαν. Η ευκολία στη χρήση, η λειτουργικότητα και η προσαρμοστικότητά του το καθιστούν ιδανική λύση για διάφορες ρυθμίσεις, βελτιώνοντας την αποτελεσματικότητα της διαδικασίας και αυξάνοντας την πιθανότητα επιτυχούς ανάκτησης.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Wikipedia. (n.d.). *Android version history*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Android_version_history
2. Maughan, R. (2016). *Mirror*. Retrieved from How many items you lose in a lifetime revealed - and what are the most common: <https://www.mirror.co.uk/news/uk-news/how-many-items-you-lose-8246553>
3. Apple. (2023). *AirTag*. Ανάκτηση από Apple: <https://www.apple.com/gr/airtag/>
4. Wikipedia. (2023). *Android (operating system)*. Ανάκτηση από [https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system))
5. Android. *Application fundamentals*. Ανάκτηση από Android Developers: <https://developer.android.com/guide/components/fundamentals>
6. Android. *The activity lifecycle*. Ανάκτηση από Android Developer: <https://developer.android.com/guide/components/activities/activity-lifecycle>
7. Android. *Fragment lifecycle*. Ανάκτηση από Android Developer: <https://developer.android.com/guide/fragments/lifecycle>
8. Android. *Layouts*. Ανάκτηση από Android Developer: <https://developer.android.com/develop/ui/views/layout/declaring-layout>
9. Android. *View*. Ανάκτηση από Android Developer: <https://developer.android.com/reference/android/view/View>
10. aayushitated2000. (2022). *History of Android*. Ανάκτηση από GeekForGeeks: <https://www.geeksforgeeks.org/history-of-android/>
11. Statista. (2023). *Mobile operating systems' market share worldwide from 1st quarter 2009 to 4th quarter 2022*. Ανάκτηση από Statista: <https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/>
12. Wikipedia. (2023). *Android Studio*. Ανάκτηση από Wikipedia: https://en.wikipedia.org/wiki/Android_Studio

13. Google. *Firebase Realtime Database*. Ανάκτηση από Firebase:
<https://firebase.google.com/docs/database>
14. Google. *Firebase Authentication*. Ανάκτηση από Firebase:
<https://firebase.google.com/docs/auth>
15. JSON. *Εισαγωγή στο JSON*. Ανάκτηση από JSON:
<https://www.json.org/json-el.html>
16. Google. *Understand Firebase Realtime Database Security Rules*.
Ανάκτηση από Firebase:
<https://firebase.google.com/docs/database/security>
17. Google. *Cloud Storage for Firebase*. Ανάκτηση από Firebase:
<https://firebase.google.com/docs/storage>
18. Google. *Cloud Functions for Firebase*. Ανάκτηση από Firebase:
<https://firebase.google.com/docs/functions>
19. Google. *Firebase Cloud Messaging*. Ανάκτηση από Firebase:
<https://firebase.google.com/docs/cloud-messaging/>
20. Google. *Firebase Machine Learning*. Ανάκτηση από Firebase:
<https://firebase.google.com/docs/ml>
21. Google. *ML Kit image labeling: Labels for default model*. Ανάκτηση
από Firebase: <https://developers.google.com/ml-kit/vision/image-labeling/label-map>
22. Google. *Image Labeling*. Ανάκτηση από Firebase:
<https://firebase.google.com/docs/ml-kit/label-images>
23. Chugh, A. (2022). *Android MVVM Design Pattern*. Ανάκτηση από
DigitalOcean:
<https://www.digitalocean.com/community/tutorials/android-mvvm-design-pattern>
24. Microsoft. *Visual Studio Code in Action*. Ανάκτηση από Visual Studio
Code: <https://code.visualstudio.com/docs>
25. Google. *Cloud locations*. Ανάκτηση από Google Cloud:
<https://cloud.google.com/about/locations#europe>