

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ  
ΕΦΑΡΜΟΓΗ ΔΙΑΧΕΙΡΙΣΗΣ ΑΠΟΣΥΡΣΗΣ ΟΧΗΜΑΤΩΝ



**Των φοιτητών**

**Συμεωνίδα Στυλιανού  
Αρ. Μητρώου: 154540**

**Συμεωνίδα Δημήτριου  
Αρ. Μητρώου: 174909**

**Επιβλέπων**

**Δεληγιάννης Ιγνάτιος  
Βαθμίδα: Καθηγητής**

**Ημερομηνία 4/6/2022**

Τίτλος Π.Ε.: Εφαρμογή διαχείρισης απόσυρσης οχημάτων

Κωδικός Π.Ε.: 21390

Όνοματεπώνυμο φοιτητών: Συμεωνίδης Στυλιανός – Συμεωνίδης Δημήτριος

Όνοματεπώνυμο εισηγητή: Δεληγιάννης Ιγνάτιος

Ημερομηνία ανάληψης Π.Ε.: 20/10/2021

Ημερομηνία περάτωσης Π.Ε.: 4/6/2022

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

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

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



## Πρόλογος

Η παρούσα πτυχιακή εργασία, με τίτλο: «Εφαρμογή διαχείρισης απόσυρσης οχημάτων», πραγματοποιήθηκε στο πλαίσιο των σπουδών του προπτυχιακού προγράμματος του τμήματος «Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων» του Διεθνούς Πανεπιστημίου της Ελλάδος (ΔΙΠΑΕ) κατά το ακαδημαϊκό έτος 2021 – 2022. Το τμήμα εδρεύει στη Σύνδο της Περιφερειακής Ενότητας Θεσσαλονίκης.

Γενικός σκοπός της εργασίας είναι η δημιουργία εφαρμογής να αποτελέσει χρήσιμο και αξιόπιστο εργαλείο στις υπηρεσίες των Δημοτικών Αρχών, οι οποίες ασχολούνται με τη διαδικασία διαχείρισης απόσυρσης οχημάτων. Η υλοποίηση της εφαρμογής εμπεριέχει σύνολο λειτουργιών.

## Περίληψη

Η ανάπτυξη της εφαρμογής διαχείρισης απόσυρσης οχημάτων υλοποιήθηκε στο προγραμματιστικό περιβάλλον Android Studio. Περιλαμβάνει τις ακόλουθες λειτουργίες:

- Φωτογράφιση εγκαταλελειμμένου οχήματος (με ή δίχως αριθμό κυκλοφορίας) και καταχώρηση στο κεντρικό σύστημα, μαζί με τις συντεταγμένες θέσης που βρίσκεται.
- Επαλήθευση των στοιχείων από υπάλληλο του Δήμου.
- Έγκριση για απόσυρση.
- Ανάθεση απόσυρσης σε ειδική ομάδα.
- Ολοκλήρωση της απόσυρσης.
- Παρακολούθηση των σταδίων της όλης διαδικασίας από διάφορους χρήστες.
- Παρακολούθηση σε χάρτη των θέσεων οχημάτων.
- Οργάνωση των ημερησίων αποσύρσεων.
- Παρακολούθηση των διεκπεραιώσεων των ημερησίων αποσύρσεων.

# Vehicle withdrawal management application

Symeonidis Stylianos

Symeonidis Dimitrios

## **Abstract**

The development of the vehicle withdrawal management application was implemented in the Android Studio development environment. Includes the following functions:

- Photograph of an abandoned vehicle (with or without registration number) and registration in the central system, together with the position coordinates located.
- Verification of the data by an employee of the Municipality.
- Approval for withdrawal.
- Assigning a withdrawal to a special group.
- Completion of the withdrawal.
- Monitoring of the stages of the whole process by various users.
- Map tracking of vehicle locations.
- Organization of daily withdrawals.
- Monitoring the processing of daily withdrawals.

## **Ευχαριστίες**

Ευχαριστούμε θερμά τον επιβλέποντα καθηγητή κ. Δελιγιάννη Ιγνάτιο για τη στήριξη που μας προσέφερε καθ' όλη τη διάρκεια της πτυχιακής μας εργασίας.

Επίσης, ευχαριστούμε όλους τους καθηγητές καθώς και το διοικητικό προσωπικό της σχολής για τη συνεχή τους συμπαράσταση και υποστήριξη των σπουδών μας.

# Περιεχόμενα

Πρόλογος.....	iv
Περίληψη.....	v
Abstract .....	vi
Ευχαριστίες .....	vii
Περιεχόμενα .....	viii
Κατάλογος Εικόνων .....	xi
Κατάλογος Πινάκων.....	xi
Παραδείγματα Κώδικα.....	xi
Συντομογραφίες.....	xii
Κεφάλαιο 1ο: Λειτουργικό σύστημα Android .....	1
1.1 Εισαγωγή.....	1
1.2 Τι είναι το Android.....	1
1.3 Η εξέλιξη του Android.....	1
1.4 Δομή Android.....	2
1.4.1 Πυρήνας (Linux Kernel).....	2
1.4.2 Βιβλιοθήκες (Libraries).....	2
1.4.3 Πλαίσιο εφαρμογών (Application Framework).....	3
1.4.4 Εφαρμογές (Applications).....	3
1.5 Δομικά συστατικά των εφαρμογών.....	3
1.5.1 Δραστηριότητες (Activities).....	3
1.5.2 Υπηρεσίες (Services).....	4
1.5.3 Πάροχοι περιεχομένου (Content Providers).....	4
1.5.4 Παραλήπτες μηνυμάτων πανεκπομπών (Broadcast Receivers) .....	4
1.6 Επίλογος.....	4
Κεφάλαιο 2ο: Android Studio .....	5
2.1 Εισαγωγή.....	5
2.2 Τι είναι το Android Studio .....	5
2.3 Γλώσσες Προγραμματισμού .....	5
2.3.1 Java.....	5
2.3.2 Kotlin.....	6
2.3.3 C++.....	6
2.4 Επίλογος.....	6

Κεφάλαιο 3ο: Χάρτες Google (Google Maps) .....	7
3.1 Εισαγωγή .....	7
3.2 Τι είναι οι Χάρτες Google .....	7
3.2.1 Street View .....	8
3.3 Γεωεντοπισμός (Geolocation) .....	8
3.4 Επίλογος .....	8
Κεφάλαιο 4ο: Οι τεχνολογίες της εφαρμογής .....	9
4.1 Εισαγωγή .....	9
4.2 Firebase .....	9
4.2.1 Αυθεντικοποίηση (Authentication) .....	9
4.2.2 Βάση δεδομένων πραγματικού χρόνου (Realtime Database).....	9
4.3 XML (eXtensible Markup Language) .....	10
4.4 Επίλογος .....	10
Κεφάλαιο 5ο: Μεθοδολογία ανάπτυξης λογισμικού .....	11
5.1 Εισαγωγή .....	11
5.2 Ο ορισμός του Scrum .....	11
5.3 Η Ομάδα Scrum.....	11
5.3.1 Scrum Master .....	11
5.3.2 Product Owner.....	11
5.3.3 Ομάδα Ανάπτυξης.....	12
5.4 Product Backlog .....	12
5.5 Ιστορίες Χρηστών (User Stories).....	13
5.6 Εκτίμηση προσπάθειας (Planning Poker).....	15
5.7 Επίλογος.....	16
Κεφάλαιο 6ο: Τεχνική αναφορά της εφαρμογής.....	17
6.1 Εισαγωγή.....	17
6.2 Ανάπτυξη και Προγραμματισμός.....	17
6.2.1 Περιγραφή του πίνακα Vehicle .....	22
6.2.2 Ανέβασμα εικόνας στο FirebaseStorage .....	23
6.2.3 Λειτουργία Geolocation .....	24
6.2.4 Λειτουργία MapView .....	25
6.2.5 Ειδοποιήσεις.....	26
6.2.6 Περιγραφή του πίνακα Notification .....	26
6.3 Πρόσβαση Χρηστών και Firebase Authentication Panel .....	27
6.4 Προβλήματα κατά το testing, προσθήκες, διορθώσεις και αλλαγές.....	29

6.5	Επίλογος.....	30
	ΒΙΒΛΙΟΓΡΑΦΙΑ.....	31
	ΠΑΡΑΡΤΗΜΑ: ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΤΗ.....	33

## Κατάλογος Εικόνων

Εικόνα 1.1 Διάγραμμα αρχιτεκτονικής Android.....	2
Εικόνα 1.2 Κύκλος ζωής μιας δραστηριότητας .....	4
Εικόνα 3.1 Χάρτης ΔΙΠΑΕ Σίνδου .....	7
Εικόνα 5.1 Πρότυπο Ιστορίας Χρήστη .....	13
Εικόνα 5.2 Αξιολόγηση των US.....	15
Εικόνα 5.3 Σύνοψη βαθμολογίας παιχνιδιού .....	15
Εικόνα 6.1 Δημιουργία νέας εφαρμογής στην online πλατφόρμα Google Developers Console .....	18
Εικόνα 6.2 Το αρχείο google-services.json.....	18
Εικόνα 6.3 Συγχρονισμός της κλάσης Role με το Firebase .....	20
Εικόνα 6.4 Δημιουργία API key για χρήση Google Maps στην εφαρμογή .....	26
Εικόνα 6.5 Σύνδεση στο Project vehicleapp του Firebase Console για Αυθεντικοποίηση χρηστών ....	28
Εικόνα 6.6 Λίστα των εγγεγραμμένων χρηστών προς Αυθεντικοποίηση.....	28

## Κατάλογος Πινάκων

Πίνακας 1.1 Εκδόσεις Android .....	1
Πίνακας 5.1 Ιστορίες Χρηστών (User Stories).....	13

## Παραδείγματα Κώδικα

Κώδικας 4.1 Στιγμιότυπο ενός XML κειμένου .....	10
Κώδικας 6.1 Φόρτωση βιβλιοθηκών Firebase .....	17
Κώδικας 6.2 Κλήση της βάσης δεδομένων Firebase κατά την εγγραφή.....	19
Κώδικας 6.3 Χρήση του Firebase Authentication για τον έλεγχο ταυτότητας του εγγεγραμμένου χρήστη στο Firebase.....	19
Κώδικας 6.4 Έλεγχος του ρόλου στην κλάση MainActivity .....	21
Κώδικας 6.5 Μεταφόρτωση των εικόνων στο Firebase .....	23
Κώδικας 6.6 Αποθήκευση των δεδομένων του οχήματος στον πίνακα "Vehicle" του Firebase .....	24
Κώδικας 6.7 Άδειες Χρήστη .....	24
Κώδικας 6.8 LocationManager .....	25
Κώδικας 6.9 Google Maps API key στο αρχείο AndroidManifest.xml .....	26
Κώδικας 6.10 Η κλάση MyNotificationPublisher.....	27
Κώδικας 6.11 Η κλάση MyAppCompatActivity .....	27

## Συντομογραφίες

ΔΠΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία
κ.λπ.	και λοιπά
π.χ.	παραδείγματος χάρη
GPS	Global Positioning System
HTML	HyperText Markup Language
iOS	iPhone Operating System
I/O	Input/Output
IDE	Integrated Development Environment
JDK	Java Development Kit
JRE	Java Runtime Environment
JSON	JavaScript Object Notation
JVM	Java Virtual Machine
NDK	Native Development Kit
PP	Planning Poker
SDK	Software Development Kit
SSL	Secure Sockets Layer
UI	User Interface
US	User Stories
W3C	World Wide Web Consortium
XML	eXtensible Markup Language

## Κεφάλαιο 1ο: Λειτουργικό σύστημα Android

### 1.1 Εισαγωγή

Στο παρόν κεφάλαιο γίνεται αναφορά στο λειτουργικό σύστημα Android και την εξέλιξή του. Αναλύεται η δομή του που περιλαμβάνει τον πυρήνα, τις βιβλιοθήκες, καθώς και ορισμένες εφαρμογές. Τέλος, περιγράφονται τα βασικά δομικά συστατικά των εφαρμογών του.

### 1.2 Τι είναι το Android

Το Android είναι ένα ευρέως διαδεδομένο λειτουργικό σύστημα το οποίο βασίζεται κυρίως σε κινητές συσκευές με οθόνη αφής (π.χ. έξυπνα τηλέφωνα και ταμπλέτες) και στηρίζεται στο λειτουργικό Linux. Η Google ήταν η πρώτη εταιρεία που το ανέπτυξε και στη συνέχεια η Open Handset Alliance. Επιτρέπει στους κατασκευαστές λογισμικού να προγραμματίζουν με τη χρήση της γλώσσας Java, ελέγχοντας έτσι τη συσκευή μέσω βιβλιοθηκών λογισμικού. [1]

### 1.3 Η εξέλιξη του Android

Η πλατφόρμα Android παρουσιάστηκε για πρώτη φορά στις 5 Νοεμβρίου 2007, παράλληλα με την ίδρυση του οργανισμού Open Handset Alliance. Η αρχική έκδοση 1.6 του Android – με την κωδική ονομασία «Donut» – κυκλοφόρησε στις 15 Σεπτεμβρίου 2009, ενώ η τελευταία έκδοση 11 στις 8 Σεπτεμβρίου 2020. [1]

Πίνακας 1.1 Εκδόσεις Android

Έκδοση	Κωδική Ονομασία	Ημερομηνία	API Level
11	11	8 Σεπτεμβρίου 2020	30
10	10	3 Σεπτεμβρίου 2019	29
9	Pie	6 Αυγούστου 2018	28
8.1	Oreo	5 Δεκεμβρίου 2017	27
8.0		21 Αυγούστου 2017	26
7.1	Nougat	4 Οκτωβρίου 2016	25
7.0		22 Αυγούστου 2016	24
6.0	Marshmallow	5 Οκτωβρίου 2015	23
5.1	Lollipop	9 Μαρτίου 2015	22
5.0		3 Νοεμβρίου 2014	21
4.4	KitKat	31 Οκτωβρίου 2013	19
4.3	Jelly Bean	24 Ιουλίου 2013	18
4.2		13 Νοεμβρίου 2012	17
4.1		9 Ιουλίου 2012	16
4.0	Ice Cream Sandwich	16 Δεκεμβρίου 2011	15
3.2	Honeycomb	15 Ιουλίου 2011	13
2.3	Gingerbread	9 Φεβρουαρίου 2011	10
2.2	Froyo	20 Μαΐου 2010	8
2.0	Eclair	26 Οκτωβρίου 2009	7
1.6	Donut	15 Σεπτεμβρίου 2009	4

## 1.4 Δομή Android

Η δομή του Android απαρτίζεται από τέσσερα επίπεδα. Κάθε ανώτερο επίπεδο εξαρτάται από τις παρεχόμενες υπηρεσίες του κατώτερου επιπέδου του.



Εικόνα 1.1 Διάγραμμα αρχιτεκτονικής Android

### 1.4.1 Πυρήνας (Linux Kernel)

Είναι ένα από τα θεμελιώδη κομμάτια της αρχιτεκτονικής του Android. Είναι αρμόδιος για την εκτέλεση βασικών υπηρεσιών του συστήματος, όπως η διαχείριση διεργασιών, η διαχείριση μνήμης, καθώς επίσης και η χρήση προγραμμάτων οδήγησης (drivers) για διάφορα τμήματα σχετικά με το υλικό. Ο πυρήνας συγκροτεί το «μεσολαβητή» μεταξύ λογισμικού και υλικού. [2]

### 1.4.2 Βιβλιοθήκες (Libraries)

Πάνω από τον πυρήνα του Linux υπάρχει ένα σύνολο βιβλιοθηκών. Οι βιβλιοθήκες αποτελούν βασικά εργαλεία ανοικτού κώδικα και βασίζονται στη γλώσσα προγραμματισμού C/C++. Είναι χρήσιμες για διάφορα τμήματα του συστήματος. Μερικά από αυτά είναι: **1)** ο χρόνος εκτέλεσης Android (Android RunTime), **2)** η μηχανή προγράμματος περιήγησης WebKit, **3)** το πλαίσιο πολυμέσων (Media Framework), **4)** το σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων SQLite και **5)** το πρωτόκολλο Secure Sockets Layer (SSL). [2]

### 1.4.3 Πλαίσιο εφαρμογών (Application Framework)

Στο αμέσως ανώτερο επίπεδο βρίσκεται το πλαίσιο εφαρμογών. Παρέχει κλάσεις και διεπαφές για την ανάπτυξη εφαρμογών Android από τους προγραμματιστές. Περιλαμβάνει τα ακόλουθα βασικά συστατικά: **1)** Διαχειριστής Δραστηριότητας (Activity Manager), **2)** Πάροχος Περιεχομένου (Content Provider), **3)** Διαχειριστής Πόρων (Resource Manager), **4)** Διαχειριστής Τοποθεσίας (Location Manager), **5)** Διαχειριστής Ενημερώσεων (Notification Manager) καθώς και άλλες εφαρμογές. [2]

### 1.4.4 Εφαρμογές (Applications)

Στην κορυφή της αρχιτεκτονικής του Android υπάρχουν οι εφαρμογές οι οποίες χρησιμοποιούνται από τους χρήστες. Παραδείγματα τέτοιων εφαρμογών είναι οι επαφές του τηλεφώνου, οι συσκευές αναπαραγωγής πολυμέσων κ.λπ.. Επιπλέον, χρησιμοποιούνται εφαρμογές και από τρίτα άτομα εγκατεστημένες στη συσκευή των χρηστών. [2]

## 1.5 Δομικά συστατικά των εφαρμογών

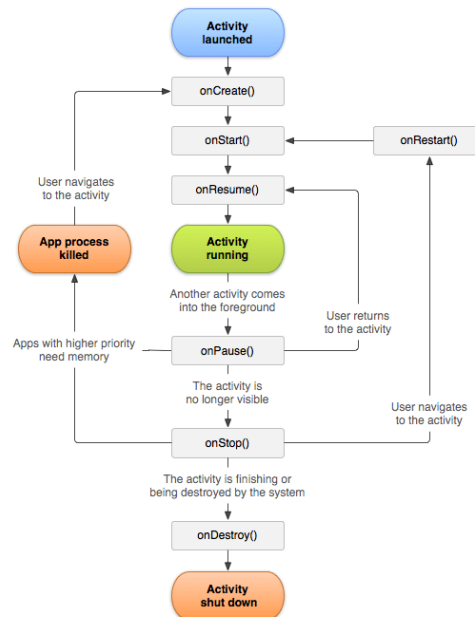
Τα τέσσερα (4) βασικά δομικά συστατικά των εφαρμογών Android είναι: **1)** οι δραστηριότητες (Activities), **2)** οι υπηρεσίες (Services), **3)** οι πάροχοι περιεχομένου (Content Providers) και **4)** οι παραλήπτες μηνυμάτων πανεκπομπών (Broadcast Receivers). Κάθε είδος συστατικού παρέχει έναν ιδιαίτερο σκοπό και έχει έναν ιδιαίτερο κύκλο ζωής ο οποίος προσδιορίζει τον τρόπο με τον οποίο ένα συστατικό δημιουργείται και καταστρέφεται. [3]

### 1.5.1 Δραστηριότητες (Activities)

Ένα activity σχετίζεται με το στρώμα παρουσίασης (presentation layer) μιας εφαρμογής που αναπαρίσταται σε μία οθόνη (screen) και με την οποία αλληλεπιδρά ο χρήστης. Μία εφαρμογή είναι πιθανό να περιλαμβάνει περισσότερα από ένα activity με σκοπό τη διαχείριση πολλαπλών οθονών της εφαρμογής. Κάθε δραστηριότητα είναι υπεύθυνη για την αποθήκευση της κατάστασής της στη διάρκεια του κύκλου ζωής της. [3]

Σε ένα κύκλο ζωής ενός activity υπάρχουν οι παρακάτω έξι (6) καταστάσεις:

- **onCreate()** : Καλείται όταν δημιουργείται το activity και εκτελείται μόνο μία φορά.
- **onStart()** : Κάνει ορατή τη διεπαφή του activity στον χρήστη.
- **onResume()** : Είναι το στάδιο αλληλεπίδρασης με τον χρήστη.
- **onPause()** : Εκτελείται όταν έχει σταματήσει η αλληλεπίδραση με τον χρήστη και το activity περνά στο παρασκήνιο.
- **onStop()** : Καλείται όταν το activity είναι αόρατο στον χρήστη.
- **onDestroy()** : Καλείται όταν καταστρέφεται το activity.



Εικόνα 1.2 Κύκλος ζωής μιας δραστηριότητας

### 1.5.2 Υπηρεσίες (Services)

Οι υπηρεσίες αφορούν διεργασίες που εκτελούνται στο παρασκήνιο (background) και είναι χρήσιμες για την εκτέλεση μακροχρόνιων εργασιών. Μία υπηρεσία δεν προσφέρει γραφικό περιβάλλον (User Interface) στο χρήστη. Παραδείγματος χάρη, μία τέτοια υπηρεσία είναι η αναπαραγωγή ενός μουσικού κομματιού από τον χρήστη ενώ βρίσκεται παράλληλα σε κάποια άλλη εφαρμογή της συσκευής του. Επίσης, μπορεί να κατεβάσει δεδομένα μέσω του διαδικτύου χωρίς να διακοπεί η αλληλεπίδρασή του από άλλες δραστηριότητες. [3]

### 1.5.3 Πάροχοι περιεχομένου (Content Providers)

Οι πάροχοι περιεχομένου δίνουν τη δυνατότητα σε μία εφαρμογή να εισέρχεται στα δεδομένα άλλων εφαρμογών και να τα διαχειρίζεται. Επιπλέον, της επιτρέπουν να διανέμει τα δεδομένα της σ' αυτές. Οι χώροι όπου αποθηκεύονται αυτά τα δεδομένα μπορεί να είναι μία βάση δεδομένων SQLite ή ακόμα και φάκελοι αρχείων του κινητού. [2] [3]

### 1.5.4 Παραλήπτες μηνυμάτων πανεκπομπών (Broadcast Receivers)

Οι Broadcast Receivers, ένα από τα βασικά δομικά συστατικά του Android, θέτονται σε λειτουργία όταν δέχονται κάποια ειδοποίηση (π.χ. χαμηλή στάθμη μπαταρίας του κινητού, κατέβαση αρχείου από το Διαδίκτυο κ.λπ.) από το σύστημα. [2] Εκτός αυτού, ένας broadcast receiver μπορεί να ενεργοποιηθεί και από οποιονδήποτε χρήστη μέσω της έναρξης ή ολοκλήρωσης μίας διαδικασίας. Παράλληλα, το λειτουργικό σύστημα έχει σκοπό να ενημερώνει τις αδρανείς εφαρμογές και αυτό επιτυγχάνεται με τη δημιουργία ασύγχρονων μηνυμάτων – intents. [3]

## 1.6 Επίλογος

Στο κεφάλαιο αυτό πραγματοποιήθηκε μια σύντομη αναφορά στην εξέλιξη του λειτουργικού συστήματος Android. Τέλος, παρουσιάστηκαν η δομή καθώς και τα βασικά συστατικά των εφαρμογών του με τις αναλύσεις τους.

## Κεφάλαιο 2ο: Android Studio

### 2.1 Εισαγωγή

Το παρόν κεφάλαιο εστιάζει στο περιβάλλον προγραμματισμού Android Studio. Περιγράφει συνοπτικά και ορισμένες γλώσσες προγραμματισμού που μας παρέχει με σκοπό την υλοποίηση εξελιγμένων εφαρμογών.

### 2.2 Τι είναι το Android Studio

Το Android Studio είναι ένα ολοκληρωμένο προγραμματιστικό περιβάλλον (Integrated Development Environment – IDE) το οποίο χρησιμοποιείται για την ανάπτυξη εφαρμογών στην πλατφόρμα Android. Δημοσιοποιήθηκε στις 16 Μαΐου 2013 στο ετήσιο συνέδριο της Google I/O (Input/Output) στο Σαν Φρανσίσκο της Καλιφόρνιας. Κυκλοφόρησε ανεπίσημα με την έκδοση 0.1 τον Μάιο του 2013. Η πρώτη επίσημη έκδοση 1.0 βγήκε το Δεκέμβριο του 2014.

Το Android Studio παρουσιάζει αρκετές δυνατότητες, όπως:

- Ένα προσαρμόσιμο σύστημα κατασκευής βασισμένο στο λογισμικό Gradle.
- Ένα γρήγορο και πλούσιο σε δυνατότητες εξομοιωτή για δοκιμές εφαρμογών.
- Ένα ενοποιημένο περιβάλλον ανάπτυξης για όλες τις συσκευές Android.
- Εκτενή εργαλεία για αποσφαλμάτωση (debugging) και δοκιμή (test).
- Υποστήριξη της γλώσσας C++ και της εργαλειοθήκης Native Development Kit (NDK).
- Ενσωματωμένη υποστήριξη για την πλατφόρμα Google Cloud. [4]

### 2.3 Γλώσσες Προγραμματισμού

#### 2.3.1 Java

Η Java είναι μια δημοφιλής αντικειμενοστρεφής γλώσσα προγραμματισμού και πλατφόρμα λογισμικού που χρησιμοποιείται σε πολλές συσκευές. Τέτοιες είναι οι φορητοί υπολογιστές, κονσόλες παιχνιδιών, ιατρικές συσκευές κ.λπ.. Επινόηθηκε το 1991 από τον επιστήμονα James Gosling της εταιρείας Sun Microsystems. Οι κανόνες και η σύνταξη της στηρίζονται στις γλώσσες C και C++.

Ένα αξιόλογο προτέρημα της ανάπτυξης λογισμικού με Java είναι η φορητότητα που παρουσιάζει. Δηλαδή, μπορεί ένας ειδικός αφού γράψει κώδικα για ένα πρόγραμμα Java σε φορητό υπολογιστή, πολύ εύκολα να τον μετακινήσει σε μια φορητή συσκευή. Είναι σημαντικό να γίνει κατανοητή και η διαφορά μεταξύ των γλωσσών Java και JavaScript. Η Javascript δεν είναι απαραίτητο να μεταγλωττιστεί και εκτελείται μόνο σε φυλλομετρητές ιστοσελίδων. Αντιθέτως, ο κώδικας Java πρέπει να μεταγλωττιστεί και μπορεί να εκτελεστεί οπουδήποτε.

Για τη δημιουργία μίας εφαρμογής με χρήση της Java, πρέπει να γίνει λήψη του Java Development Kit (JDK), το οποίο διατίθεται για λειτουργικά συστήματα Windows, macOS και Linux. Ο προγραμματιστής κωδικοποιεί την εφαρμογή σε γλώσσα Java και στη συνέχεια ένας μεταγλωττιστής μετατρέπει το πρόγραμμα σε bytecode Java [σύνολο εντολών για την Java Virtual Machine (JVM) που αποτελεί μέρος του Java Runtime Environment (JRE)]. [5]

### 2.3.2 Kotlin

Η Kotlin είναι μια στατική αντικειμενοστρεφής γλώσσα προγραμματισμού που σχεδιάστηκε από την εταιρεία JetBrains και εκτελείται πάνω στην εικονική μηχανή της Java (JVM). Είναι συμβατή με την Java και το Android. Απο τον Φεβρουάριο του 2012 η Kotlin εξελίσσεται ως γλώσσα ανοικτού κώδικα. Η πρώτη επίσημη έκδοση 1.0 δημοσιεύτηκε τον Φεβρουάριο του 2016. Τον Μάιο του 2018 η Google ανακοίνωσε ότι η Kotlin θα είναι επίσημη γλώσσα του Android. Ένα πρόγραμμα Kotlin μπορεί να μεταγλωττιστεί σε πηγαίο κώδικα JavaScript. [6]

### 2.3.3 C++

Η C++ είναι μια δυναμική και ευέλικτη γλώσσα προγραμματισμού γενικού σκοπού. Έχει αρκετά κοινά χαρακτηριστικά με τη γλώσσα C καθώς έχει κληρονομήσει το μεγαλύτερο μέρος της σύνταξής της. Χρησιμοποιείται για την ανάπτυξη λειτουργικών συστημάτων, προγραμμάτων περιήγησης, παιχνιδιών και εφαρμογών σε σταθερούς υπολογιστές. Τέλος, υποστηρίζει διαφορετικούς τρόπους προγραμματισμού όπως διαδικαστικούς, αντικειμενοστρεφείς, λειτουργικούς κ.λπ.. [7]

## 2.4 Επίλογος

Στο παρόν κεφάλαιο έγινε αναφορά στο προγραμματιστικό περιβάλλον Android Studio. Με λίγα λόγια, συνοψίστηκαν και ορισμένες γλώσσες προγραμματισμού που αυτό μας παρέχει βοηθώντας έτσι στην ανάπτυξη εφαρμογών.

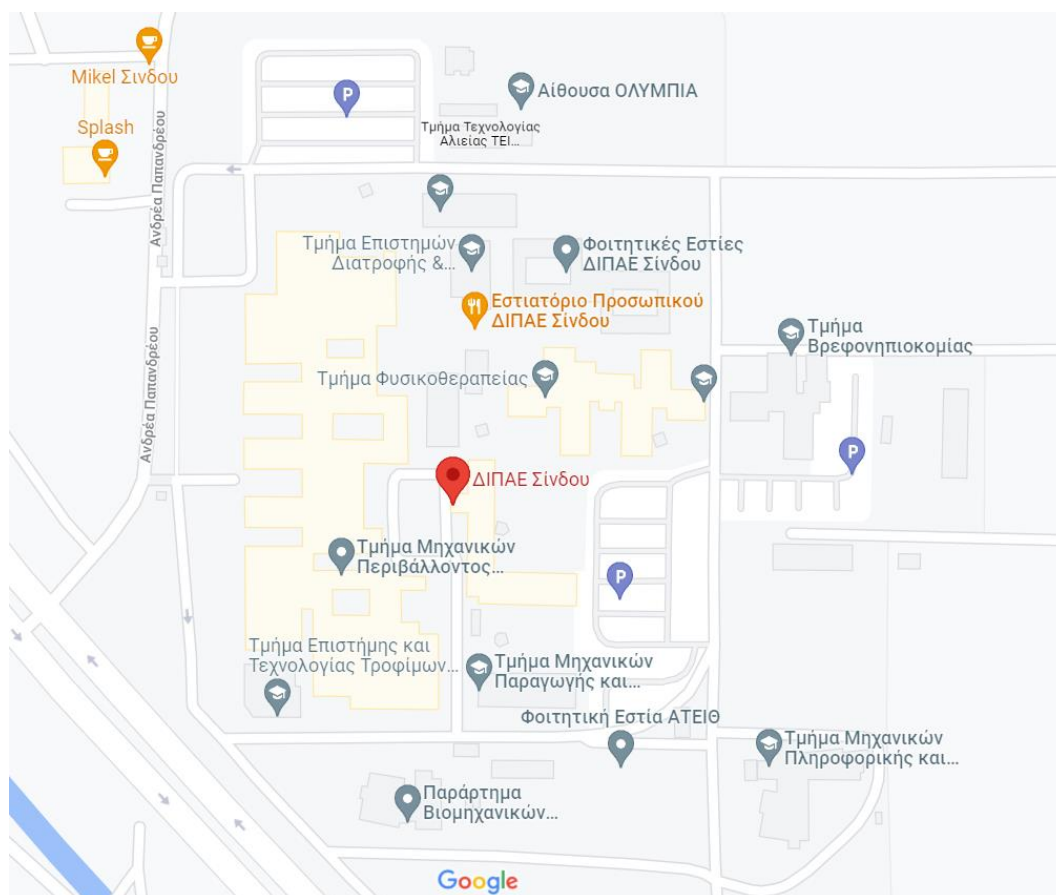
## Κεφάλαιο 3ο: Χάρτες Google (Google Maps)

### 3.1 Εισαγωγή

Το Διαδίκτυο, πέρα από τις απεριόριστες λειτουργίες που μας προσφέρει, περιλαμβάνει και τους Χάρτες Google. Με τη βοήθεια των Χαρτών αναπτύσσεται και μια σημαντική δυνατότητα αυτή του γεωεντοπισμού.

### 3.2 Τι είναι οι Χάρτες Google

Οι Χάρτες Google είναι υπηρεσία απεικόνισης των γεωγραφικών, διοικητικών και άλλων στοιχείων μιας τοποθεσίας στο Διαδίκτυο. Είναι διαθέσιμοι σε Android συσκευές από την Google Maps API v2. Παρέχουν στο χρήστη ένα διαδραστικό χάρτη του πλανήτη στον οποίο μπορεί πολύ εύκολα να περιηγηθεί εντοπίζοντας επιχειρήσεις που βρίσκονται σε διάφορες πόλεις των κρατών του κόσμου. Οι δορυφορικές εικόνες των Χαρτών Google δεν ανανεώνονται σε πραγματικό χρόνο, ωστόσο η Google προσθέτει δεδομένα στη Βάση Δεδομένων της σε τακτικά χρονικά διαστήματα και οι περισσότερες από τις εικόνες δεν είναι πάνω από τριών ετών. [2] [8]



Εικόνα 3.1 Χάρτης ΔΙΠΑΕ Σίνδου

### 3.2.1 Street View

Ένα χρήσιμο τεχνολογικό γνώρισμα αποτελεί το Street View. Ξεκίνησε να χρησιμοποιείται το 2001. Αναπαριστά με ψηφιακό τρόπο τον κοινωνικό μας χώρο στους Χάρτες Google. Προσφέρει εκατομμύρια πανοραμικές εικόνες από θέσεις σε πολλούς δρόμους του κόσμου. Δίνει την ευχέρεια στον καθένα να εξερευνήσει εικονικά κάθε περιοχή του πλανήτη. Το περιεχόμενό του πηγάζει από την Google και τους χρήστες. [9]

### 3.3 Γεωεντοπισμός (Geolocation)

Ο γεωεντοπισμός (geolocation) χρησιμοποιεί συστήματα εντοπισμού θέσης (π.χ. μιας φυσικής διεύθυνσης). Δίνει τη δυνατότητα στους χρήστες να παρακολουθούν την ακριβή γεωγραφική θέση τους μέσω συσκευών πλοήγησης GPS (Global Positioning System), κινητών τηλεφώνων, ταμπλετών και σημείων πρόσβασης Wi-Fi. Οι κινητές και επιτραπέζιες συσκευές μπορούν να βρίσκουν μια γεωγραφική τοποθεσία και να μας προμηθεύουν σημαντικές πληροφορίες. [10]

### 3.4 Επίλογος

Οι έννοιες Google Maps, Street View και Geolocation αποτελούν συνδυαστικά ωφέλιμο εργαλείο στα χέρια κάθε ενδιαφερόμενου. Οι γεωγραφικές θέσεις, κοντινές και απομακρυσμένες, διαμέσου αυτών των εφαρμογών συνυπάρχουν στην καθημερινότητά μας.

## Κεφάλαιο 4ο: Οι τεχνολογίες της εφαρμογής

### 4.1 Εισαγωγή

Για τις ανάγκες υλοποίησης της εφαρμογής της πτυχιακής εργασίας χρησιμοποιήθηκαν συγκεκριμένες τεχνολογίες. Τέτοιες τεχνολογίες είναι η απομακρυσμένη βάση δεδομένων Firebase και η γλώσσα σήμανσης XML.

### 4.2 Firebase

Το Firebase θεωρείται πλατφόρμα ανάπτυξης εφαρμογών ιστού σε πραγματικό χρόνο στηριζόμενο στην υποδομή της Google. Έχει σχεδιαστεί με σκοπό να βοηθήσει τους προγραμματιστές να αναπτύξουν τόσο ποιοτικές όσο και αξιόλογες εφαρμογές. Τα δεδομένα αποθηκεύονται σε έγγραφα που φέρουν τη μορφή JavaScript Object Notation (JSON), η οποία δεν χρησιμοποιεί ερωτήματα για την εισαγωγή, ενημέρωση, διαγραφή ή προσθήκη άλλων δεδομένων στη βάση. [11] [12]

Αλλαγές μεταξύ τοπικών αντιγράφων πληροφοριών που είναι αποθηκευμένες στις συσκευές των χρηστών με εκδόσεις που διατηρούνται στο cloud, μπορεί διαρκώς να τις διαδίδει και να τις συγχρονίζει. Το Firebase διασφαλίζει ότι τα σωστά bits είναι τα ίδια σε όλο το σύστημα και εξαλείφει πολλές από τις προκλήσεις της μίξης ελέγχου ταυτότητας, συγχρονισμού και διαχωρισμού. [11]

Το 2014 η Google απέκτησε την εταιρεία Firebase. Σήμερα, αποτελεί κεντρικό μέρος της εργαλειοθήκης ανάπτυξης του Google Cloud και το προϊόν της αποτελεί ένα αποκορύφωμα ετών εξέλιξης που τοποθετήθηκε στο κέντρο μιας προσφοράς υποστήριξης για κινητά. Διαθέσιμες βιβλιοθήκες και εργαλεία ανοιχτού κώδικα διασυνδέονται με το Firebase. [11]

#### 4.2.1 Αυθεντικοποίηση (Authentication)

Ένα βασικό χαρακτηριστικό της Firebase είναι η αυθεντικοποίηση. Σχεδόν όλες οι εφαρμογές είναι σημαντικό να χρησιμοποιούν κάποια μορφή ταυτοπροσωπίας του χρήστη. Με αυτόν τον τρόπο μια εφαρμογή είναι σε θέση να αποθηκεύει με ασφάλεια δεδομένα ενός χρήστη στο cloud καθώς και να παρέχει την ίδια ειδική γνώση σε όλες τις συσκευές του. Επιπλέον, αρκετοί είναι οι χρήστες που δε θέλουν να διαδίδουν προσωπικές πληροφορίες τους, όπως για παράδειγμα το ονοματεπώνυμο, την τοποθεσία, την ημερομηνία γέννησης, τον αριθμό τηλεφώνου και οτιδήποτε άλλο. Επομένως, συχνά επιλέγουν να χρησιμοποιούν στοιχεία που έχουν ήδη παραχωρήσει σε τρίτους και τους δίνουν δικαίωμα να διαχειρίζονται τη σύνδεση για λογαριασμό των χρηστών. [13] [14]

#### 4.2.2 Βάση δεδομένων πραγματικού χρόνου (Realtime Database)

Η Firebase Realtime Database είναι μια βάση δεδομένων που βρίσκεται στο cloud στην οποία αποθηκεύονται δεδομένα ως JSON. Συγχρονίζονται σε πραγματικό χρόνο σε κάθε συνδεδεμένο πελάτη και αποτελούν ένα μεγάλο αντικείμενο το οποίο οι προγραμματιστές μπορούν να διαχειριστούν. Ο συγχρονισμός διευκολύνει τους χρήστες στην πρόσβαση των δεδομένων τους από οποιαδήποτε συσκευή, είτε αυτή είναι ιστού είτε κινητής τηλεφωνίας. Δημιουργώντας εφαρμογές πολλαπλών πλατφορμών iOS (Operating System) και JavaScript SDK (Software Development Kit), αρκετοί πελάτες χρησιμοποιούν ένα στιγμιότυπο της βάσης και δέχονται αυτόματα ενημερώσεις νεότερων δεδομένων. Η βάση δεδομένων Realtime βοηθά στη συνεργασία μεταξύ των χρηστών. Όταν οι χρήστες αποσυνδέονται από το ίντερνετ, τα SDK βάσης δεδομένων χρησιμοποιούν

τοπική προσωρινή μνήμη στη συσκευή για την προβολή και την αποθήκευση αλλαγών. Με τη σύνδεση της συσκευής στο διαδίκτυο τα τοπικά δεδομένα συγχρονίζονται αυτόματα. [15]

### 4.3 XML (eXtensible Markup Language)

Η XML είναι κάτι παραπάνω από μία γλώσσα σήμανσης, η οποία περιέχει ένα σύνολο κανόνων για την ηλεκτρονική κωδικοποίηση εγγράφων. Αναπτύχθηκε το 1996 και από το Φεβρουάριο του 1998 απαρτίζει σύνθεση του W3C (World Wide Web Consortium).

Είναι σημαντικό να τονιστεί ότι η XML συμπληρώνει και δεν αντικαθιστά την HTML. Μεταξύ αυτών των δύο γλωσσών παρουσιάζονται σημαντικές διαφορές. Η XML σχεδιάστηκε με σκοπό να περιγράψει, να αποστέλλει και να αποθηκεύει δεδομένα που δεν έχουν αυστηρό τρόπο δόμησης. Παρέχει τη δυνατότητα στους χρήστες να ορίζουν τις δικές τους σημάνσεις (tags) αλλά και τις δομημένες μεταξύ τους σχέσεις. Αντίθετα, η HTML δημιουργήθηκε για να παρουσιάζει δεδομένα και εστιάζει στον τρόπο εμφάνισής τους. Οι σημάνσεις της είναι προκαθορισμένες από την ίδια τη γλώσσα προγραμματισμού. [16]

Παρακάτω αναπαρίσταται η δομή ενός XML εγγράφου:

```
<Books>
  <Book ISBN="0553212419">
    <title>Sherlock Holmes: Complete Novels...
    <author>Sir Arthur Conan Doyle</author>
  </Book>
  <Book ISBN="0743273567">
    <title>The Great Gatsby</title>
    <author>F. Scott Fitzgerald</author>
  </Book>
  <Book ISBN="0684826976">
    <title>Undaunted Courage</title>
    <author>Stephen E. Ambrose</author>
  </Book>
  <Book ISBN="0743203178">
    <title>Nothing Like It In the World</title>
    <author>Stephen E. Ambrose</author>
  </Book>
</Books>
```

Κώδικας 4.1 Στιγμιότυπο ενός XML κειμένου

### 4.4 Επίλογος

Η απομακρυσμένη βάση δεδομένων Firebase με χαρακτηριστικά την αυθεντικοποίηση και τη δημιουργία βάσης δεδομένων πραγματικού χρόνου, βοηθά τους προγραμματιστές να αναπτύξουν ποιοτικές και αξιόλογες εφαρμογές. Η XML ως γλώσσα σήμανσης περιλαμβάνει ένα σύνολο από κανόνες για την ηλεκτρονική κωδικοποίηση κειμένων.

## Κεφάλαιο 5ο: Μεθοδολογία ανάπτυξης λογισμικού

### 5.1 Εισαγωγή

Στο κεφάλαιο αυτό παρουσιάζεται η μεθοδολογία ανάπτυξης λογισμικού Scrum που ακολουθήθηκε. Περιγράφεται ο πίνακας ιστοριών των χρηστών (User Stories) καθώς και η εκτίμηση προσπάθειάς τους με την τεχνική του Planning Poker.

### 5.2 Ο ορισμός του Scrum

Το Scrum δεν είναι μια τεχνική για τη δημιουργία προϊόντων. Ορίζεται ως ένα πλαίσιο με τη συμβολή του οποίου μπορεί οποιοσδήποτε να λύσει σύνθετα εξελικτικά προβλήματα, παραδίδοντας συγχρόνως προϊόντα μεγάλης αξίας, τόσο με παραγωγικό όσο και δημιουργικό τρόπο. Παρουσιάζει χαμηλή πολυπλοκότητα, είναι απλό στην κατανόησή του και απαιτητικό να το κατακτήσει κάποιος. [17]

### 5.3 Η Ομάδα Scrum

Η Ομάδα Scrum περιλαμβάνει τον Scrum Master, τον Product Owner και την Ομάδα Ανάπτυξης. Η ίδια η ομάδα αυτοοργανώνεται έχοντας δεξιότητες από διαφορετικές λειτουργικές περιοχές. Η απόφαση ολοκλήρωσης ενός έργου με τον καλύτερο δυνατό τρόπο λαμβάνεται από τις ίδιες τις ομάδες χωρίς να εξαρτώνται από άλλους εκτός της ομάδας. Το μοντέλο της ομάδας στο Scrum έχει σχεδιαστεί έτσι ώστε να μεγιστοποιούνται η ευελιξία, η δημιουργικότητα και η παραγωγικότητα. [18]

#### 5.3.1 Scrum Master

Ο Scrum Master εγγυάται υπεύθυνα ότι το Scrum είναι κατανοητό και μπορεί να εφαρμοστεί από την Ομάδα Scrum. Συμβάλλει στην προσπάθεια των ατόμων εκτός της Ομάδας Scrum να κατανοήσουν ποιες από τις αλληλεπιδράσεις τους με την ομάδα είναι σημαντικές και ωφέλιμες και ποιες όχι. Επιπλέον, τους βοηθάει να τις αλλάξουν με απώτερο στόχο να αυξήσουν την αξία της Ομάδας Scrum. [17]

#### 5.3.2 Product Owner

Ο Product Owner είναι αρμόδιος για τη βελτιστοποίηση της αξίας του προϊόντος και του έργου που παράγει η Ομάδα Ανάπτυξης. Ο τρόπος με τον οποίο γίνεται αυτό διαφέρει αρκετά από οργανισμό σε οργανισμό, από ομάδα σε ομάδα και από άνθρωπο σε άνθρωπο.

Ο Product Owner είναι το μοναδικό άτομο που διαχειρίζεται το Product Backlog. Η διαχείριση του Product Backlog μπορεί:

- Να εκφράζει ξεκάθαρα τα στοιχεία του Product Backlog.
- Να ταξινομεί τα στοιχεία του Product Backlog με σκοπό την εκπλήρωση των στόχων και αποστολών, με τον βέλτιστο δυνατό τρόπο.
- Να διασφαλίζει ότι το Product Backlog είναι ορατό, διαφανές και ξεκάθαρο σε όλους και ότι δείχνει τα επόμενα πράγματα στα οποία θα δουλέψει η Ομάδα Scrum.

- Να διασφαλίζει ότι η Ομάδα Ανάπτυξης κατανοεί τα στοιχεία του Product Backlog στο βαθμό που απαιτείται.

Ο Product Owner λαμβάνει υπόψιν του όλες τις προτεινόμενες απόψεις άλλων ατόμων όσον αφορά τη φάση του Product Backlog. Όποιος επιθυμεί να τροποποιήσει τη σειρά σπουδαιότητας ενός στοιχείου, θα πρέπει να απευθυνθεί στον Product Owner. [18]

### 5.3.3 Ομάδα Ανάπτυξης

Η Ομάδα Ανάπτυξης συγκροτείται από επαγγελματίες οι οποίοι εκτελούν τις αναγκαίες εργασίες, ώστε στο τέλος κάθε Sprint να παραδώσουν μία εν δυνάμει παραδοτέα Επαύξηση "Ετοιμου" προϊόντος. Η Επαύξηση πραγματοποιείται μόνο από μέλη της Ομάδας Ανάπτυξης.

Είναι δομημένη και ενδυναμώνεται από τον οργανισμό, κατά τέτοιο τρόπο ώστε να οργανώνει και να διευθετεί η ίδια τη δουλειά της. Η συνεργασία που προκύπτει ως αποτέλεσμα καλυτερεύει στο μέγιστο βαθμό τη συνολική αποδοτικότητα και αποτελεσματικότητα της Ομάδας Ανάπτυξης.

Κάθε Ομάδα Ανάπτυξης παρουσιάζει συνήθως τα ακόλουθα χαρακτηριστικά:

- ✓ Αυτοοργανώνεται. Κανείς (ούτε και ο Scrum Master) δεν λέει στην Ομάδα Ανάπτυξης πώς να μετατρέψει το Product Backlog σε Επαυξήσεις εν δυνάμει παραδοτέου προϊόντος.
- ✓ Έχει δεξιότητες από διαφορετικές λειτουργικές περιοχές, και συνολικά ως ομάδα όλες τις δεξιότητες που απαιτούνται για τη δημιουργία της Επαύξησης του προϊόντος.
- ✓ Το Scrum δεν αναγνωρίζει κανέναν τίτλο για τα μέλη Ομάδας Ανάπτυξης, πέρα από αυτόν του Developer, ανεξάρτητα από τη δουλειά που κάνει ο καθένας στην ομάδα. Σε αυτόν τον κανόνα δεν υπάρχουν εξαιρέσεις.
- ✓ Το Scrum δεν αναγνωρίζει υποομάδες στην Ομάδα Ανάπτυξης, ανεξάρτητα του εάν υπάρχουν συγκεκριμένοι τομείς με τους οποίους θα ασχοληθεί η Ομάδα, όπως οι δοκιμές και η επιχειρησιακή ανάλυση. Και εδώ δεν υπάρχουν εξαιρέσεις.
- ✓ Τα επιμέρους μέλη της Ομάδας Ανάπτυξης μπορεί να έχουν εξειδικευμένες ικανότητες και τομείς στους οποίους επικεντρώνονται, αλλά η ευθύνη ανήκει συλλογικά σε ολόκληρη την Ομάδα Ανάπτυξης. [17]

## 5.4 Product Backlog

Στην ευέλικτη μεθοδολογία Scrum, το Product Backlog (συνγά αναφέρεται και ως ανεκτέλεστο προϊόν) είναι μια εξελισσόμενη, ταξινομημένη λίστα όλων των προτεινόμενων εργασιών που πρέπει να υλοποιήσει μια ομάδα προϊόντων στα πλαίσια ενός έργου. Στόχος αυτής της ομάδας είναι να δώσει προτεραιότητα σ' αυτές που θεωρεί σημαντικές και να τις παραδώσει με τη σειρά.

Το ανεκτέλεστο προϊόν στην πρόωρη μορφή του εμπεριέχει τις πιο γνωστές και κατανοητές προδιαγραφές. Βελτιώνεται ταυτόχρονα με το προϊόν και το περιβάλλον εντός του οποίου αυτό χρησιμοποιείται. Είναι δυναμικό και μεταβάλλεται συνεχώς για να υποδείξει τι χρειάζεται το προϊόν ώστε να είναι κατάλληλο, ανταγωνιστικό, και χρήσιμο.

Το Product Backlog περιέχει όλα τα χαρακτηριστικά και τις λειτουργίες που δείχνουν τις μελλοντικές αλλαγές που πρόκειται να συμβούν σε εκδόσεις του προϊόντος. Τα στοιχεία του έχουν τις εξής ιδιότητες: περιγραφή, σειρά προτεραιότητας, εκτίμηση και αξία. [17]

## 5.5 Ιστορίες Χρηστών (User Stories)

Το 1999, ο Αμερικανός μηχανικός λογισμικού Kent Beck επινόησε τον όρο User Stories για τις δυνατότητες του προϊόντος. Οι ιστορίες χρηστών αντιπροσωπεύουν απλές, συνοπτικές και κατανοητές περιγραφές για τη λειτουργικότητα ενός γνωρίσματος από τη μεριά του χρήστη ή του πελάτη. [19]

Το πρότυπο που ακολουθείται για τη δημιουργία ενός User Story έχει ως εξής:

**“ Ως (τύπος χρήστη),  
θέλω να (εκτελέσω κάποια ενέργεια)  
έτσι ώστε να (επιτύχω κάποιο στόχο/όφελος). ”**

Εικόνα 5.1 Πρότυπο Ιστορίας Χρήστη

Σύμφωνα με τον οδηγό SCRUM, υπεύθυνος για την κατάρτιση, διαθεσιμότητα και ταξινόμηση των ιστοριών χρηστών (User Stories) είναι ο Product Owner. Στο συγκεκριμένο έργο οι ρόλοι είναι:

**Όνομα Scrum Master:** Δεληγιάννης Ιγνάτιος

**Όνομα 1ου μέλους:** Συμεωνίδης Στυλιανός

**Όνομα 2ου μέλους:** Συμεωνίδης Δημήτριος

Πίνακας 5.1 Ιστορίες Χρηστών (User Stories)

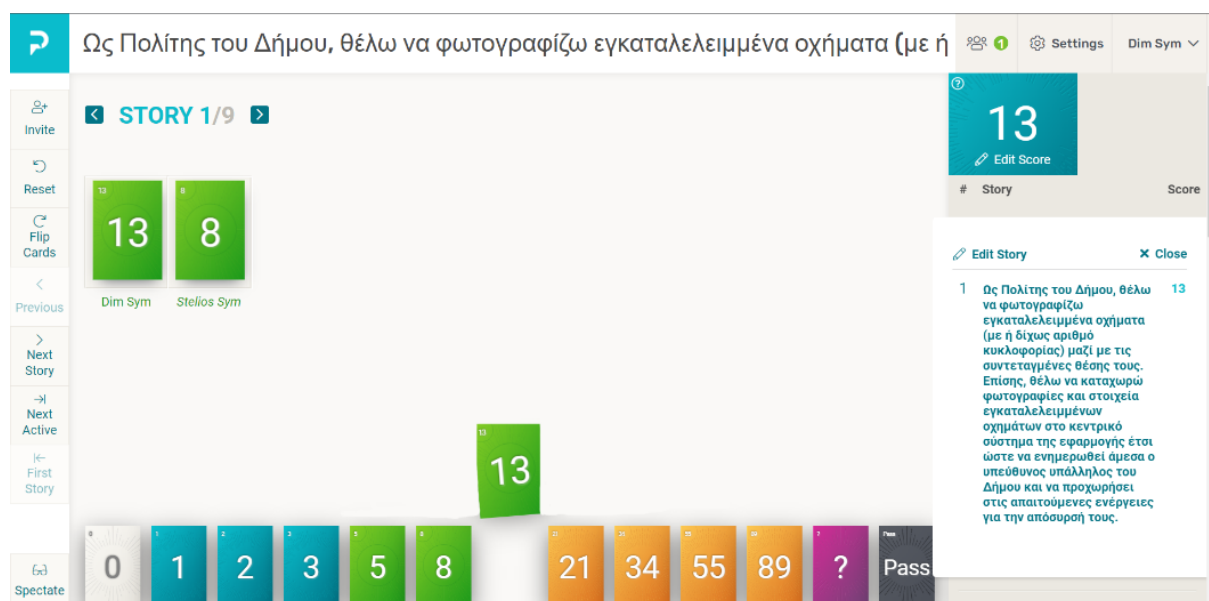
ΕΦΑΡΜΟΓΗ ΑΠΟΣΥΡΣΗΣ ΕΓΚΑΤΑΛΕΙΜΜΕΝΩΝ ΟΧΗΜΑΤΩΝ (Vehicle App)				
ID	USER STORY	Περιγραφή	Estimation - Story Points	Priority 1 - 5
	(Product Owner - PO)		(Scrum Team - ST)	(PO)
US01	Ως <b>Πολίτης του Δήμου</b> , θέλω να <b>φωτογραφίζω</b> εγκαταλελειμμένα οχήματα (με ή δίχως αριθμό κυκλοφορίας) μαζί με τις συντεταγμένες θέσης τους. Επίσης, θέλω να <b>καταχωρώ</b> φωτογραφίες και στοιχεία εγκαταλελειμμένων οχημάτων στο κεντρικό σύστημα της εφαρμογής <b>έτσι ώστε να</b> ενημερωθεί άμεσα ο υπεύθυνος υπάλληλος του Δήμου και να προχωρήσει στις απαιτούμενες ενέργειες για την απόσυρσή τους.	<p>Ανοίγοντας την εφαρμογή, ο Πολίτης του Δήμου κάνει είσοδο συμπληρώνοντας τα προσωπικά του στοιχεία. Εφόσον δεν είναι εγγεγραμμένος, δημιουργεί ένα νέο προφίλ.</p> <p>Χρησιμοποιώντας την κάμερα και την τοποθεσία του κινητού του, φωτογραφίζει τα εγκαταλελειμμένα οχήματα που επιθυμεί (με ή δίχως αριθμό κυκλοφορίας) μαζί με τις συντεταγμένες θέσης τους. Μόλις τραβηχτεί η φωτογραφία, αποθηκεύεται αυτόματα στη "Συλλογή" του κινητού.</p> <p>Μέσα από το drawer μενού της εφαρμογής, επιλέγει "Καταχώρηση Οχημάτων" και μεταφέρεται σε μια νέα οθόνη όπου ανεβάζει τη φωτογραφία πατώντας το κουμπί "Μεταφόρτωση φωτογραφίας".</p>	13	1

US02	Ως <b>υπάλληλος της αρμόδιας υπηρεσίας του Δήμου</b> , θέλω να <b>επαληθεύω</b> τα στοιχεία οποιουδήποτε εγκαταλελειμένου οχήματος, μέσα από την εφαρμογή, <b>για να</b> βεβαιωθώ ότι πρόκειται για όχημα με πραγματικά στοιχεία.	Επαλήθευση στοιχείων σχετικά με ... <ul style="list-style-type: none"> <li>- την ταυτότητα των ιδιοκτητών</li> <li>- τον αριθμό πλαισίου της μηχανής</li> <li>- την άδεια κυκλοφορίας</li> <li>- τις πινακίδες (εφόσον αυτό φέρει)</li> </ul>	21	2
US03	Ως <b>υπάλληλος του Δήμου</b> , θέλω να <b>εγκρίνω</b> ένα οποιοδήποτε εγκαταλελειμένο όχημα <b>για</b> νόμιμη απόσυρση.	Από το κεντρικό σύστημα της εφαρμογής, η αρμόδια υπηρεσία του Δήμου αποφασίζει και επιλέγει τα εγκαταλελειμμένα οχήματα που θα εγκρίνει για απόσυρση. Αυτή η λειτουργία εκτελείται πατώντας το κουμπί "Έγκριση απόσυρσης".	8	3
US04	Ως <b>υπάλληλος του Δήμου</b> , θέλω να <b>αναθέτω</b> την απόσυρση του εγκαταλελειμένου οχήματος σε ειδική ομάδα <b>ώστε να</b> την διεκπεραιώσουν με νόμιμες διαδικασίες.	Κατόπιν, τα παραπάνω οχήματα αναθέτονται για απόσυρση σε ειδική ομάδα η οποία και ολοκληρώνει τη διαδικασία πατώντας το κουμπί "Ολοκλήρωση απόσυρσης".	13	3
US05	Ως <b>υπεύθυνος της ομάδας απόσυρσης</b> , θέλω να <b>αποσύρω</b> και να <b>καταχωρώ</b> την απόσυρση του εγκαταλελειμένου οχήματος <b>με σκοπό να</b> διεκπεραιώνεται η απόσυρση σύμφωνα με το νόμο.		21	3
US06	Ως <b>χρήστης</b> , θέλω να <b>παρακολουθώ</b> τα στάδια της όλης διαδικασίας <b>για να</b> βελτιωθεί ο χρόνος απόσυρσης των οχημάτων.	Μέσα από το drawer μενού, υπάρχει η δυνατότητα παρακολούθησης των σταδίων της όλης διαδικασίας επιλέγοντας "Πρόσδος". Έπειτα, εμφανίζεται μια λίστα όλων των σταδίων στην οποία ο χρήστης μπορεί να δει ποιες εργασίες/ενέργειες έχουν ολοκληρωθεί.	21	4
US07	Ως <b>αρμόδια υπηρεσία του Δήμου</b> , θέλω να <b>παρακολουθώ</b> σε χάρτη τις θέσεις των οχημάτων <b>ώστε να</b> έχω καλύτερη οπτική εικόνα σχετικά με τα σημεία στα οποία βρίσκονται.	Με την χρήση των "Χαρτών" (Google Maps) εμφανίζονται σε μια νέα οθόνη οι συντεταγμένες θέσης των οχημάτων.	8	4
US08	Ως <b>υπάλληλος του Δήμου</b> , θέλω να <b>οργανώνω</b> τις ημερήσιες αποσύρσεις <b>με σκοπό να</b> πραγματοποιούνται σε αριθμό όσες μπορεί να διεκπεραιώσει το αρμόδιο τμήμα.	Μέσα από το drawer μενού, ο χρήστης επιλέγει την κατηγορία "Οργάνωση". Έπειτα, εμφανίζεται μια νέα οθόνη στην οποία μπορεί να οργανώσει τις ημερήσιες αποσύρσεις διαχωρίζοντας τα οχήματα σε κατηγορίες (π.χ. αυτοκίνητα Ι.Χ, μηχανές, φορτηγά, λεωφορεία, κ.λπ.).	21	5
US09	Ως <b>υπάλληλος του Δήμου</b> , θέλω να <b>παρακολουθώ</b> τις διεκπεραιώσεις των ημερησίων αποσύρσεων <b>για να</b> έχω πλήρη εικόνα της διαδικασίας τους.	Με την χρήση των ειδοποιήσεων στο κινητό, θα υπάρχει η δυνατότητα να παρακολουθεί ποιες ημερήσιες αποσύρσεις έχουν διεκπεραιωθεί.	8	5

## 5.6 Εκτίμηση προσπάθειας (Planning Poker)

Το Planning Poker, γνωστό και ως "Scrum Poker", είναι μια ευέλικτη τεχνική εκτίμησης και σχεδιασμού που βασίζεται στη συναίνεση. Χρησιμοποιείται κυρίως από ομάδες με σκοπό την εκτίμηση του χρόνου και της προσπάθειας που χρειάζονται για την ολοκλήρωση κάθε απόφασης στο ανεκτέλεστο προϊόν τους. Η εκτίμηση στηρίζεται στις ιστορίες χρηστών και το μέγεθος του προϊόντος υπολογίζεται σε Πόντους (Score). [20]

Στις παρακάτω εικόνες παρουσιάζεται το περιβάλλον του παιχνιδιού μέσω της ιστοσελίδας του PP καθώς και συγκεντρωτικά αποτελέσματα που προέκυψαν έπειτα από ψηφοφορία μεταξύ των μελών της ομάδας.



Εικόνα 5.2 Αξιολόγηση των US

**Your Planning Poker® Game Summary**

**Vehicle App**

Story	Story Title	Score
1	Ως Πολίτης του Δήμου, θέλω να φωτογραφίζω εγκαταλελειμμένα οχήματα (με ή δίχως αριθμό κυκλοφορίας) μαζί με τις συντεταγμένες θέσης τους. Επίσης, θέλω να καταχωρώ φωτογραφίες και στοιχεία εγκαταλελειμμένων οχημάτων στο κεντρικό σύστημα της εφαρμογής έτσι ώστε να ενημερωθεί άμεσα ο υπεύθυνος υπάλληλος του Δήμου και να προχωρήσει στις απαιτούμενες ενέργειες για την απόσυρσή τους.	13
2	Ως υπάλληλος της αρμόδιας υπηρεσίας του Δήμου, θέλω να επαληθεύω τα στοιχεία οποιουδήποτε εγκαταλελειμμένου οχήματος, μέσα από την εφαρμογή, για να βεβαιωθώ ότι πρόκειται για όχημα με πραγματικά στοιχεία.	21
3	Ως υπάλληλος του Δήμου, θέλω να εγκρίνω ένα οποιοδήποτε εγκαταλελειμμένο όχημα για νόμιμη απόσυρση.	8
4	Ως υπάλληλος του Δήμου, θέλω να αναθέτω την απόσυρση του εγκαταλελειμμένου οχήματος σε ειδική ομάδα ώστε να την διεκπεραιώσουν με νόμιμες διαδικασίες.	13
5	Ως υπεύθυνος της ομάδας απόσυρσης, θέλω να αποσύρω και να καταχωρώ την απόσυρση του εγκαταλελειμμένου οχήματος με σκοπό να διεκπεραιώνεται η απόσυρση σύμφωνα με το νόμο.	21
6	Ως χρήστης, θέλω να παρακολουθώ τα στάδια της όλης διαδικασίας για να βελτιωθεί ο χρόνος απόσυρσης των οχημάτων.	21
7	Ως αρμόδια υπηρεσία του Δήμου, θέλω να παρακολουθώ σε χάρτη τις θέσεις των οχημάτων ώστε να έχω καλύτερη οπτική εικόνα σχετικά με τα σημεία στα οποία βρίσκονται.	8
8	Ως υπάλληλος του Δήμου, θέλω να οργανώνω τις ημερήσιες αποσύρσεις με σκοπό να πραγματοποιούνται σε αριθμό όπως μπορεί να διεκπεραιώσει το αρμόδιο τμήμα.	21
9	Ως υπάλληλος του Δήμου, θέλω να παρακολουθώ τις διεκπεραιώσεις των ημερησίων αποσύρσεων για να έχω πλήρη εικόνα της διαδικασίας τους.	8

Εικόνα 5.3 Σύνοψη βαθμολογίας παιχνιδιού

## 5.7 Επίλογος

Η μεθοδολογία ανάπτυξης λογισμικού Scrum βοήθησε στην υλοποίηση της εφαρμογής της πτυχιακής. Έγινε περιγραφή του πίνακα ιστοριών των χρηστών. Με την τεχνική του Planning Poker εκτιμήθηκε ο βαθμός δυσκολίας των US.

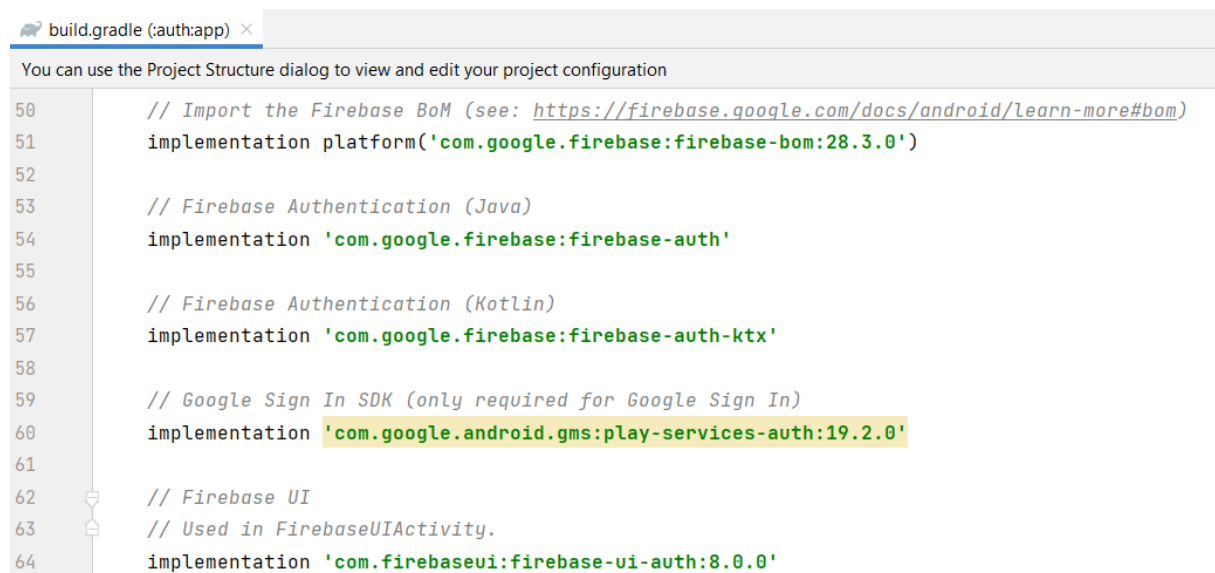
## Κεφάλαιο 6ο: Τεχνική αναφορά της εφαρμογής

### 6.1 Εισαγωγή

Στο κεφάλαιο αυτό αναφέρονται τα βήματα που ακολουθήθηκαν για τη δημιουργία της εφαρμογής. Ο σχεδιασμός της σχετίζεται με τη διαχείριση απόσυρσης οχημάτων.

### 6.2 Ανάπτυξη και Προγραμματισμός

Για την ανάπτυξη της εφαρμογής της πτυχιακής εργασίας χρησιμοποιήθηκε το περιβάλλον Android Studio και συγκεκριμένα η έκδοση Arctic Fox (2020.3.1). Αρχικά, δημιουργήσαμε μια νέα εφαρμογή Android στο Android Studio με Gradle build και Java. Η αρχιτεκτονική της εφαρμογής στηρίζεται στις δυνατότητες που μας δίνουν οι βιβλιοθήκες Firebase Authentication, Database και Storage τις οποίες καλούμε στον κώδικα. Στον σύνδεσμο Google Developers Console: <https://firebase.google.com/docs/android/setup#console> υπάρχει το επίσημο έγγραφο τεκμηρίωσης που περιέχει οδηγίες σχετικά με τον τρόπο εγκατάστασης και σύνδεσης του Firebase με το Android, τις οποίες και ακολουθήσαμε.

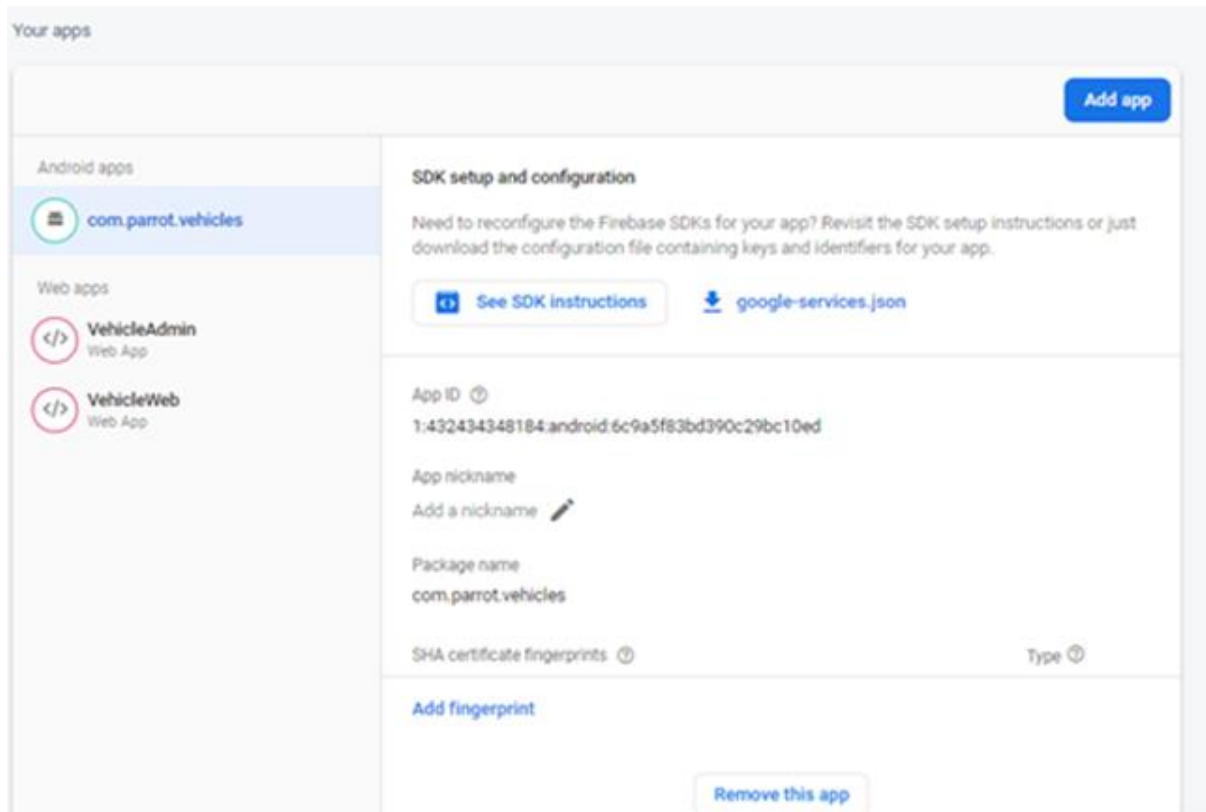


```
build.gradle (:auth.app) x
You can use the Project Structure dialog to view and edit your project configuration
50 // Import the Firebase BoM (see: https://firebase.google.com/docs/android/learn-more#bom)
51 implementation platform('com.google.firebase:firebase-bom:28.3.0')
52
53 // Firebase Authentication (Java)
54 implementation 'com.google.firebase:firebase-auth'
55
56 // Firebase Authentication (Kotlin)
57 implementation 'com.google.firebase:firebase-auth-ktx'
58
59 // Google Sign In SDK (only required for Google Sign In)
60 implementation 'com.google.android.gms:play-services-auth:19.2.0'
61
62 // Firebase UI
63 // Used in FirebaseAuthUIActivity.
64 implementation 'com.firebaseui:firebase-ui-auth:8.0.0'
```

Κώδικας 6.1 Φόρτωση βιβλιοθηκών Firebase

Το αρχείο build.gradle που φαίνεται στην παραπάνω εικόνα είναι ένα αρχείο που παράγεται αυτόματα και έχει διάφορα συστατικά (components), όπως plugins που περιέχουν όλες τις πληροφορίες όπως οι υπηρεσίες που χρησιμοποιούνται σε αυτή την εφαρμογή. Έχουμε τα com.android.application, com.google.gms.google-services, compileSdkVersion (που εδώ είναι 31), minSdkVersion που είναι 19, versioning κλπ. Η ενότητα Dependencies περιέχει τις απαιτούμενες βιβλιοθήκες: firebase-bom, firebase-auth, firebase-ui-auth, firebase-database, firebase-storage κ.α. που είναι όλα μερικά παραδείγματα των εξαρτήσεων.

Παράλληλα, δημιουργήσαμε τη mobile εφαρμογή μας στην κονσόλα Firebase και έτσι παράχθηκε το αρχείο google-service.json που τοποθετήσαμε στον αρχικό φάκελο του κώδικα της εφαρμογής.



Εικόνα 6.1 Δημιουργία νέας εφαρμογής στην online πλατφόρμα Google Developers Console

Name	Date modified	Type	Size
build	12/18/2021 10:46 AM	File folder	
src	12/15/2021 8:41 AM	File folder	
.gitignore	11/9/2021 12:05 AM	Git Ignore Source ...	1 KB
build	1/6/2022 10:32 PM	Gradle Source File	3 KB
google-services	12/18/2021 10:49 PM	JSON Source File	2 KB
proguard-rules.pro	11/9/2021 12:05 AM	PRO File	2 KB

Εικόνα 6.2 Το αρχείο google-services.json

Το google-services.json είναι το αρχείο ρυθμίσεων Firebase Android που δημιουργείται αυτόματα με τα βήματα που περιγράφονται στο Firebase Console στον παρακάτω σύνδεσμο: <https://firebase.google.com/docs/android/setup#console>.

Στο αρχείο εξόδου βλέπουμε διάφορα κύρια στοιχεία όπως firebase\_url, project\_id, storage\_bucket τα οποία δεν είναι τίποτα άλλο από το νέο δημιουργημένο Firebase Project, με το URL και το αντίστοιχο όνομα.

Στην ενότητα client βλέπουμε το όνομα του πακέτου Android και το αναγνωριστικό client oauth για τον έλεγχο ταυτότητας με το api\_key. Όλα αυτά δημιουργούνται αυτόματα δίνοντας το input στην κονσόλα Firebase. Ακόμα, η λογική ελέγχου ταυτότητας (Authentication) και εξουσιοδότησης (Authorization) της εφαρμογής μας στηρίζεται στο Firebase. Για την εξουσιοδότηση βάσει ρόλων δημιουργήσαμε έναν πίνακα με το όνομα "roles" όπου προσθέσαμε τα εξής 3 πεδία:

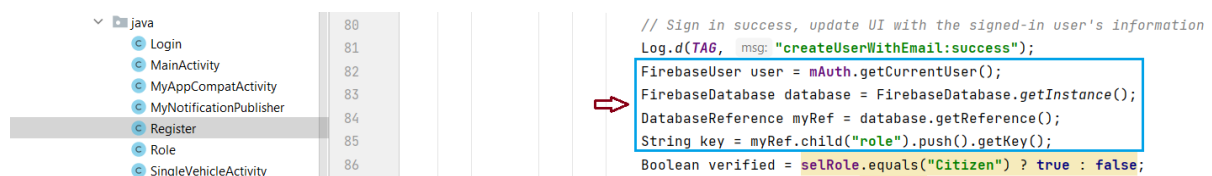
**email** – Λειτουργεί και ως username για τη σύνδεση.

**role** – Μπορεί να είναι: Πολίτης του Δήμου (Citizen) / Υπάλληλος του Δήμου (Employee) / Υπεύθυνος Ομάδας Απόσυρσης (Leader).

**verified** – Λαμβάνει τιμές true/false.

Μόλις η ταυτότητα ενός χρήστη ελεγχθεί μέσω του Firebase, πραγματοποιείται query στο πεδίο role και ο χρήστης ανακατευθύνεται στην αντίστοιχη αρχική οθόνη με βάση το ρόλο του. Για να κάνουμε το Role Query δημιουργήσαμε την κλάση Role.java. Επίσης, για την υποστήριξη των Vehicle Queries δημιουργήσαμε την κλάση Vehicle.java, όπως και τις κλάσεις Register και Login για την εγγραφή και την σύνδεση αντίστοιχα.

## Εγγραφή



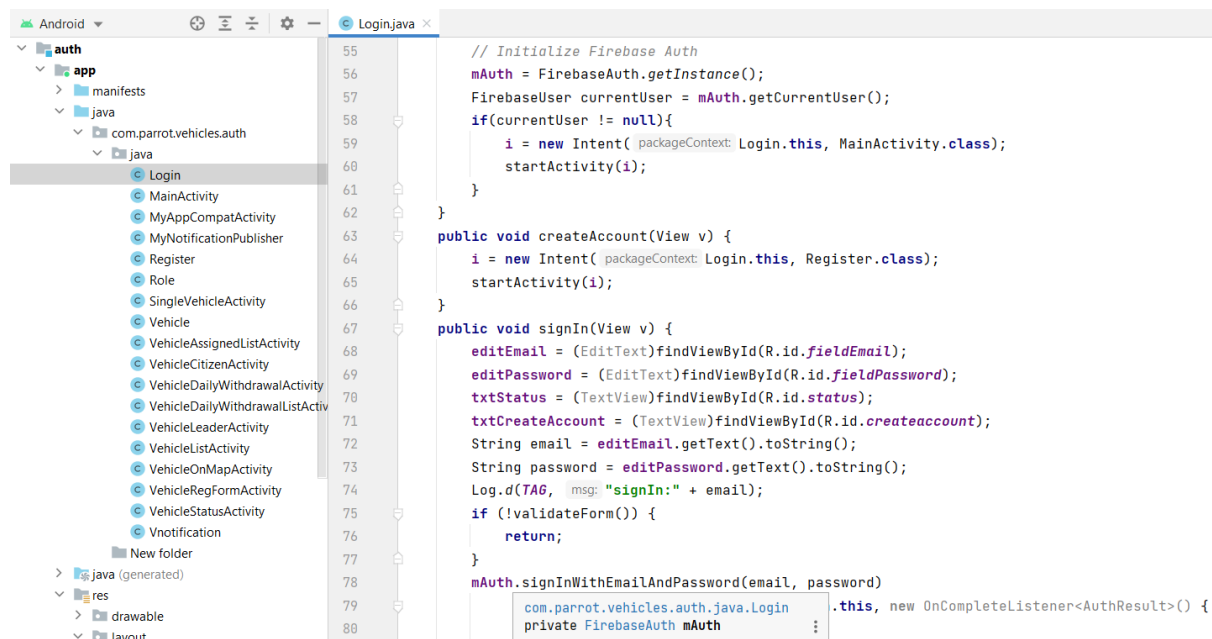
```

80
81
82 // Sign in success, update UI with the signed-in user's information
83 Log.d(TAG, msg: "createUserWithEmail:success");
84 FirebaseUser user = mAuth.getCurrentUser();
85 FirebaseDatabase database = FirebaseDatabase.getInstance();
86 DatabaseReference myRef = database.getReference();
87 String key = myRef.child("role").push().getKey();
88 Boolean verified = selRole.equals("Citizen") ? true : false;

```

Κώδικας 6.2 Κλήση της βάσης δεδομένων Firebase κατά την εγγραφή

## Σύνδεση



```

55 // Initialize Firebase Auth
56 mAuth = FirebaseAuth.getInstance();
57 FirebaseUser currentUser = mAuth.getCurrentUser();
58 if(currentUser != null){
59     i = new Intent( packageContext: Login.this, MainActivity.class);
60     startActivity(i);
61 }
62 }
63 public void createAccount(View v) {
64     i = new Intent( packageContext: Login.this, Register.class);
65     startActivity(i);
66 }
67 public void signIn(View v) {
68     editTextEmail = (EditText)findViewById(R.id.fieldEmail);
69     editTextPassword = (EditText)findViewById(R.id.fieldPassword);
70     txtStatus = (TextView)findViewById(R.id.status);
71     txtCreateAccount = (TextView)findViewById(R.id.createaccount);
72     String email = editTextEmail.getText().toString();
73     String password = editTextPassword.getText().toString();
74     Log.d(TAG, msg: "signIn:" + email);
75     if (!validateForm()) {
76         return;
77     }
78     mAuth.signInWithEmailAndPassword(email, password)
79     .this, new OnCompleteListener<AuthResult>() {
80     private FirebaseAuth mAuth

```

Κώδικας 6.3 Χρήση του Firebase Authentication για τον έλεγχο ταυτότητας του εγγεγραμμένου χρήστη στο Firebase

Το FirebaseAuth είναι μια κλάση Java της βιβλιοθήκης FirebaseAuth. Στο παραπάνω απόσπασμα κώδικα δημιουργούμε ένα αντικείμενο του FirebaseAuth με το όνομα mAuth και στη συνέχεια προσπαθούμε να πάρουμε τα τρέχοντα στοιχεία του χρήστη. Αν δεν πάρουμε τα τρέχοντα στοιχεία

του χρήστη, τότε ανακατευθύνεται στην οθόνη σύνδεσης. Έτσι, ο μη πιστοποιημένος χρήστης δε θα μπορέσει να αποκτήσει ποτέ πρόσβαση στην εφαρμογή. Αν βρεθούν τα στοιχεία του χρήστη, τότε θα γίνει άλλη μια κλήση για τη βάση δεδομένων FirebaseDatabase και θα δημιουργηθεί ένα αντικείμενο αναφοράς της βάσης δεδομένων. Στη συνέχεια θα ληφθεί το αντικείμενο του πίνακα "Role" και μετά από αυτό θα επιλεγεί ο ρόλος για τον τρέχοντα χρήστη. Με βάση τον ρόλο, ο χρήστης θα αποκτήσει πρόσβαση στην αντίστοιχη οθόνη (δηλαδή ως «Πολίτης του Δήμου», «Υπάλληλος του Δήμου» ή «Υπεύθυνος Ομάδας Απόσυρσης»).

Κατά την εγγραφή καλούνται 2 μέθοδοι η μία μετά την άλλη. Πρώτα δημιουργείται λογαριασμός με το Email και τον κωδικό πρόσβασης που δίνονται, χρησιμοποιώντας το FirebaseAuth. Μόλις δημιουργηθεί ο λογαριασμός καλείται το αντικείμενο FirebaseDatabase για να δημιουργήσει μια νέα εγγραφή στον πίνακα Role με στήλες: Email, Role, verified (true/false). Από προεπιλογή όλοι οι χρήστες έχουν οριστεί σε verified = false εκτός από τον Citizen που είναι από προεπιλογή true.

Η κλάση Role συγχρονίζεται με τον πίνακα «role» του Firebase:



Εικόνα 6.3 Συγχρονισμός της κλάσης Role με το Firebase

Γίνεται επίσης ο έλεγχος του ρόλου στο MainActivity.

```

95     if(currentUser != null) {
96         FirebaseDatabase database = FirebaseDatabase.getInstance();
97         DatabaseReference myRef = database.getReference();
98         myRef.child("role")
99         .addListenerForSingleValueEvent(new ValueEventListener() {
100             @Override
101             public void onDataChange(DataSnapshot dataSnapshot) {
102                 for (DataSnapshot snapshot : dataSnapshot.getChildren()) {
103                     Role role = snapshot.getValue(Role.class);
104                     if(role.email.equalsIgnoreCase(currentUser.getEmail()) && role.verified) {
105                         if(role.role.equals("Citizen")) {
106                             Menu nav_Menu = navigationView.getMenu();
107                             nav_Menu.findItem(R.id.vehicle_registration).setVisible(true);
108                             nav_Menu.findItem(R.id.status).setVisible(true);
109                             txtRole.setText("Ρόλος: Πολίτης του Δήμου");
110                         }
111                         if(role.role.equals("Employee")) {
112                             Menu nav_Menu = navigationView.getMenu();
113                             nav_Menu.findItem(R.id.vehicle_map).setVisible(true);
114                             nav_Menu.findItem(R.id.vehicle_list).setVisible(true);
115                             nav_Menu.findItem(R.id.daily_withdrawal).setVisible(true);
116                             txtRole.setText("Ρόλος: Υπάλληλος του Δήμου");
117                         }
118                         if(role.role.equals("Leader")) {
119                             Menu nav_Menu = navigationView.getMenu();
120                             nav_Menu.findItem(R.id.vehicle_list_assigned).setVisible(true);
121                             txtRole.setText("Ρόλος: Υπεύθυνος Ομάδας Απόσυρσης");
122                         }
123                     }
124                 }
125             }
126         }
127     }

```

Κώδικας 6.4 Έλεγχος του ρόλου στην κλάση MainActivity

Το παραπάνω κομμάτι κώδικα αφορά την εμφάνιση της Αρχικής Σελίδας / Οθόνης Καλωσορίσματος. Εδώ δημιουργείται και πάλι ένα αντικείμενο της κλάσης FirebaseAuth με το όνομα mAuth και ένα άλλο αντικείμενο της FirebaseDatabase με το όνομα database, το οποίο στη συνέχεια δέχεται το DatabaseReference ως myRef όπου επιλέγεται ο πίνακας Role. Ακόμα, εάν ο ρόλος του χρήστη είναι Πολίτης του Δήμου (Citizen) θα εμφανιστεί στην αρχική σελίδα καλωσορίσματος το μήνυμα "Ρόλος: Πολίτης του Δήμου", για τον υπάλληλο του δήμου (Employee) το μήνυμα "Ρόλος: Υπάλληλος του Δήμου" και για τον υπεύθυνο της ομάδας (Leader) το μήνυμα "Ρόλος: Υπεύθυνος Ομάδας Απόσυρσης". Με βάση τους παραπάνω ρόλους τα στοιχεία μενού μέσα στο Drawer Menu ορίζονται επίσης σε True ή False. Για τον Πολίτη του Δήμου, τα vehicle\_registration (εγγραφή οχήματος) και η vehicle\_status (κατάσταση οχήματος) ορίζονται ως true. Όταν ένας Υπάλληλος του Δήμου συνδεθεί, κάποια πρόσθετα στοιχεία μενού ενεργοποιούνται με τα ονόματα vehicle\_list (λίστα οχημάτων), vehicle\_map (χάρτης οχημάτων) και daily\_withdrawal (καθημερινή απόσυρση/οργάνωση), βάσει του Product Backlog. Για τους Υπεύθυνους Ομάδας Απόσυρσης προστίθεται ένα στοιχείο μενού που ονομάζεται "vehicle\_list\_assigned" που προσθέτει το κουμπί «Λίστα Καταχωρημένων Οχημάτων». Εκτός από αυτά, το Home (Αρχική Σελίδα) στην κορυφή και το Logout (Αποσύνδεση) στο κάτω μέρος ορίζεται πάντα ως true για όλους τους ρόλους ώστε να εμφανίζονται στο μενού.

### 6.2.1 Περιγραφή του πίνακα Vehicle

Στον πίνακα Vehicle, τα πεδία ορίζονται και λαμβάνουν ως τιμές τα παρακάτω:

**assignee** – Email του Υπεύθυνου Ομάδας Απόσυρσης. Φέρει τη μορφή: "leader\_name@gmail.com".

**carLicense** – Άδεια του οχήματος. Συμπληρώνεται από τον Υπάλληλο του Δήμου κατά τη στιγμή της επαλήθευσης.

**category** – Κατηγορία οχήματος: Αυτοκίνητο I.X. / Μηχανή / Φορτηγό / Λεωφορείο / Τροχόσπιτο / Αυτοκινούμενο / Βαν. Ορίζεται κατά τη φάση ημερήσιας απόσυρσης.

**characteristics** – Περιγραφή των χαρακτηριστικών οχήματος. Συμπληρώνεται από τους Πολίτες κατά την εγγραφή του οχήματος. Π.χ.: "Λευκό βανάκι με γκράφιτι".

**latitude** – Γεωγραφικό πλάτος για τον χάρτη (αυτόματο).

**longitude** – Γεωγραφικό μήκος για τον χάρτη (αυτόματο).

**ownerIdentity** – Ταυτότητα του ιδιοκτήτη. Συμπληρώνεται από τον Υπάλληλο του Δήμου κατά την επαλήθευση.

**pdf** – Πλήρες URL του ειδοποιητηρίου PDF στο Firebase.  
Π.χ.: "https://firebasestorage.googleapis.com/v0/b/...".

**picture** – Το τοπικό path της φωτογραφίας. Π.χ.: "/storage/emulated/0/1644767033106.jpg".

**regPlate** – Αριθμός Πλαισίου Μηχανής. Συμπληρώνεται από τον Υπάλληλο του Δήμου κατά την επαλήθευση.

**status** – Υπάρχουν 9 πιθανές καταστάσεις και ο δείκτης κάθε κατάστασης αποθηκεύεται στη συγκεκριμένη μεταβλητή ως αριθμός μεταξύ 0 και 8.

**uid** – Email του Πολίτη του Δήμου. Φέρει τη μορφή: "citizen\_name@gmail.com".

**vin** – Άδεια Κυκλοφορίας. Συμπληρώνεται από τον Υπάλληλο του Δήμου κατά την επαλήθευση.

**vnumber** – Αριθμός Πινακίδων. Π.χ.: "YAN7593".

## 6.2.2 Ανέβασμα εικόνας στο FirebaseStorage

Ο χειρισμός των μεταφορτώσεων εικόνων στο Firebase αλλά και η αποθήκευση των στοιχείων των οχημάτων γίνεται από την κλάση VehicleRegFormActivity.java.

```

VehicleRegFormActivity.java x
342 public void saveVehicleReg(View v) {
343     mAuth = FirebaseAuth.getInstance();
344     FirebaseUser currentUser = mAuth.getCurrentUser();
345     if(currentUser != null){
346         FirebaseStorage storage = FirebaseStorage.getInstance();
347         StorageReference storageRef = storage.getReference();
348         if(imagePath != null) {
349             StorageReference mountainsRef = storageRef.child(imagePath);
350             Bitmap bitmap = ((BitmapDrawable) imageView.getDrawable()).getBitmap();
351             ByteArrayOutputStream baos = new ByteArrayOutputStream();
352             bitmap.compress(Bitmap.CompressFormat.JPEG, quality: 100, baos);
353             byte[] data = baos.toByteArray();
354             UploadTask uploadTask = mountainsRef.putBytes(data);
355             uploadTask.addOnFailureListener(new OnFailureListener() {
356                 @Override
357                 public void onFailure(@NonNull Exception exception) {
358                     // Handle unsuccessful uploads
359                 }
360             }).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
361                 @Override
362                 public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
363                     System.out.println("currentUser picture saved : "+currentUser.getEmail());
364                 }
365             });

```

Κώδικας 6.5 Μεταφόρτωση των εικόνων στο Firebase

Για κάθε Πολίτη του Δήμου είναι ενεργοποιημένη η επιλογή «Εγγραφή Οχήματος», όπου ο ιδιοκτήτης μπορεί να προσθέσει τα στοιχεία του οχήματος, ξεκινώντας με τον αριθμό άδειας. Στη συνέχεια, μπορεί να επιλέξει από τη γκαλερί ή να τραβήξει μια φωτογραφία. Για αυτή τη δυνατότητα, έχουμε χρησιμοποιήσει την επιλογή Android AlertDialog που εμφανίζει 3 επιλογές: "Κάμερα", "Συλλογή" και "Ακύρωση". Για να γίνει λήψη μίας φωτογραφίας από την κάμερα χρησιμοποιείται το Android MediaStore.ACTION\_IMAGE\_CAPTURE, ενώ για να πάρουμε μία εικόνα από τη γκαλερί έχουμε χρησιμοποιήσει το Intent.ACTION\_PICK. Μόλις πατηθεί το κουμπί Save (Αποθήκευση), το αρχείο bitmap αποστέλλεται στο FirebaseStorage χρησιμοποιώντας την κλάση UploadTask όπου το πεδίο addOnSuccessListener λαμβάνει τη διεύθυνση URL της αποθηκευμένης εικόνας στο FirebaseStorage και στη συνέχεια αποθηκεύει τη διαδρομή στον πίνακα Vehicle ως νέα εγγραφή μαζί με άλλα στοιχεία του οχήματος.

```

VehicleRegFormActivity.java
366     EditText editvnumber = (EditText)findViewById(R.id.fieldVehicleNumber);
367     EditText editcharacteristics = (EditText)findViewById(R.id.fieldCharacteristics);
368     String vnumber = editvnumber.getText().toString().toUpperCase();
369     String characteristics = editcharacteristics.getText().toString();
370     if(!vnumber.equals("")) {
371         FirebaseDatabase database = FirebaseDatabase.getInstance();
372         DatabaseReference myRef = database.getReference();
373         String key = myRef.child("vehicles").push().getKey();
374         Vehicle vehicle = new Vehicle(currentUser.getEmail(), picturePath, latitude, longitude, vnumber, characteristics, status: 0,
375         Map<String, Object> vehicleValues = vehicle.toMap();
376         Map<String, Object> childUpdates = new HashMap<>();
377         childUpdates.put(" /vehicles/" + key, vehicleValues);
378         myRef.updateChildren(childUpdates);
379         Toast.makeText( context: this, text: "Το όχημα καταχωρήθηκε.", Toast.LENGTH_SHORT).show();
380         System.out.println("currentUser saved : "+currentUser.getEmail());
381     } else {
382         Toast.makeText( context: this, text: "Εισαγάγετε τον αριθμό οχήματος.", Toast.LENGTH_SHORT).show();
383     }

```

Κώδικας 6.6 Αποθήκευση των δεδομένων του οχήματος στον πίνακα "Vehicle" του Firebase

Για να προσθέσουμε μια νέα εγγραφή στη βάση δεδομένων FirebaseDatabase στον πίνακα Vehicles, θα πρέπει πρώτα να δημιουργήσουμε ένα αντικείμενο Java της κλάσης Vehicles και να ορίσουμε όλα τα δεδομένα εισόδου στην κλάση Vehicle. Στη συνέχεια, με τη μέθοδο myRef.child("vehicles").push().getKey() παίρνουμε ένα νέο κλειδί που χρησιμοποιείται ως αναφορά για την εισαγωγή νέας εγγραφής στη βάση δεδομένων με τον ίδιο τρόπο. Τέλος, με τη μέθοδο myRef.updateChildren(childUpdates) ενημερώνουμε τις τιμές στη FirebaseDatabase.

### 6.2.3 Λειτουργία Geolocation

Για να χρησιμοποιήσουμε τη λειτουργία Geolocation εκμεταλλευτήκαμε τη λειτουργικότητα του Android GPS που απαιτεί την αντίστοιχη αδειοδότηση. Οι άδειες που ζητούνται από τον χρήστη είναι: **1)** Άδεια χρήσης της κάμερας, **2)** Άδεια πρόσβασης στα αρχεία και **3)** Άδεια γνωστοποίησης της τοποθεσίας, όπως φαίνεται στο παρακάτω τμήμα κώδικα της MainActivity.

```

MainActivity.java
32     protected void onCreate(Bundle savedInstanceState) {
33         mAuth = FirebaseAuth.getInstance();
34         FirebaseUser currentUser = mAuth.getCurrentUser();
35         if(currentUser == null){
36             i = new Intent( packageContext: MainActivity.this, Login.class);
37             startActivity(i);
38         }
39         ActivityCompat.requestPermissions( activity: this, new String[]{ android.Manifest.permission.ACCESS_FINE_LOCATION,
40             android.Manifest.permission.READ_EXTERNAL_STORAGE, android.Manifest.permission.WRITE_EXTERNAL_STORAGE,
41             android.Manifest.permission.CAMERA}, requestCode: 1);
42         super.onCreate(savedInstanceState);
43         ActivityMainBinding binding = ActivityMainBinding.inflate(getLayoutInflater());
44         setContentView(R.layout.activity_main);

```

Κώδικας 6.7 Άδειες Χρήστη

Στη συνέχεια ανακτώνται από τον παρακάτω κώδικα LocationManager.

```

VehicleRegFormActivity.java x
205 private void getLocation() {
206     if (ActivityCompat.checkSelfPermission(
207         context: VehicleRegFormActivity.this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED &&
208         context: VehicleRegFormActivity.this, Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
209         ActivityCompat.requestPermissions( activity: this, new String[]{Manifest.permission.ACCESS_FINE_LOCATION}, requestCode: 1);
210     } else {
211         LocationRequest = LocationRequest.create();
212         LocationRequest.setInterval(1000);
213         LocationRequest.setFastestInterval(500);
214         LocationRequest.setPriority(LocationRequest.PRIORITY_BALANCED_POWER_ACCURACY);
215
216         LocationCallback = new LocationCallback() {
217             @Override
218             public void onLocationResult(@NonNull LocationResult locationResult) {
219                 super.onLocationResult(locationResult);
220                 locationResult.getLastLocation();
221                 Location lastLocation = locationResult.getLastLocation();
222                 double lat = lastLocation.getLatitude();
223                 double longi = lastLocation.getLongitude();
224                 latitude = String.valueOf(lat);
225                 longitude = String.valueOf(longi);
226                 showLocation = findViewById(R.id.fieldLocation);
227                 showLocation.setText("Latitude: " + latitude + " , " + "Longitude: " + longitude);
228             }
229
230             @Override
231             public void onLocationAvailability(@NonNull LocationAvailability locationAvailability) {
232                 super.onLocationAvailability(locationAvailability);
233             }
234         };

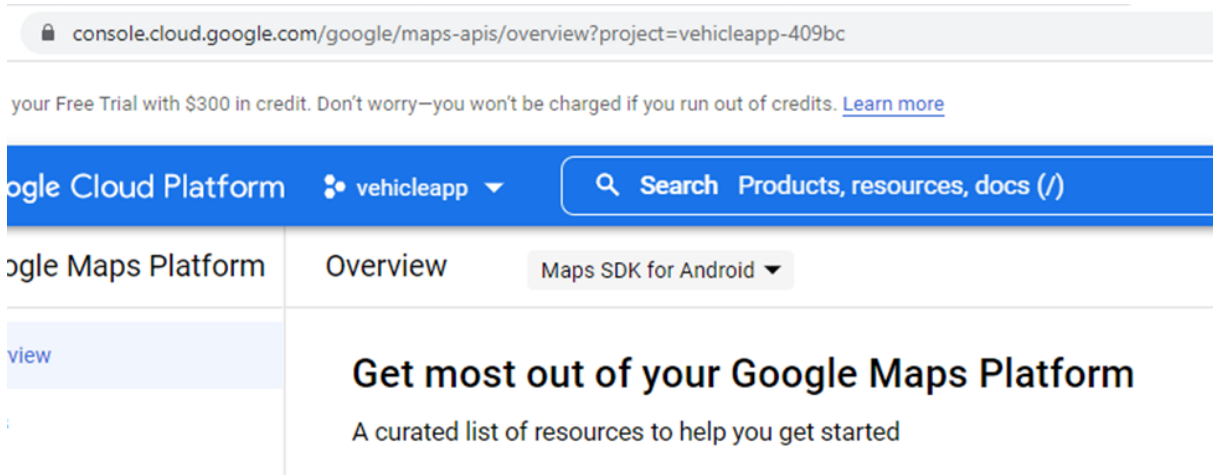
```

Κώδικας 6.8 LocationManager

Για όλες τις εγγραφές οχημάτων που συμπεριλαμβάνουν την τοποθεσία του οχήματος είναι πολύ σημαντική η λήψη του σωστού γεωγραφικού πλάτους και μήκους. Έτσι, κατά την πρώτη σύνδεση στην Αρχική Οθόνη η εφαρμογή ζητά άδεια για την ενεργοποίηση της τοποθεσίας, την ενεργοποίηση της κάμερας και την άδεια εξωτερικού αποθηκευτικού χώρου. Η λειτουργία locationManager του Android λαμβάνει την τοποθεσία χρησιμοποιώντας την υπηρεσία LocationManager.GPS\_PROVIDER, η οποία δίνει το γεωγραφικό πλάτος και το γεωγραφικό μήκος.

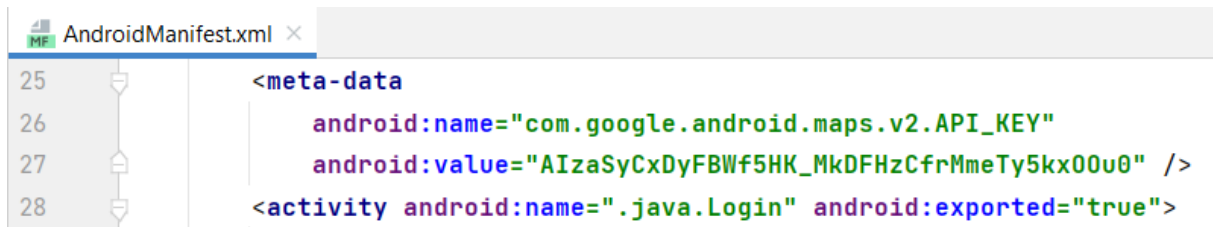
## 6.2.4 Λειτουργία MapView

Για τη λειτουργία του Χάρτη στην εφαρμογή μας το πρώτο που χρειάζεται είναι ένα νέο κλειδί API για Google Maps από την πλατφόρμα Google Developers Console. Αρχικά δημιουργούμε ένα νέο Project και έπειτα ακολουθούμε τις οδηγίες έκδοσης του API key.



Εικόνα 6.4 Δημιουργία API key για χρήση Google Maps στην εφαρμογή

Μόλις πάρουμε το API key το τοποθετούμε στο πεδίο android:value της ετικέτας metadata στο αρχείο AndroidManifest.xml.



Κώδικας 6.9 Google Maps API key στο αρχείο AndroidManifest.xml

## 6.2.5 Ειδοποιήσεις

Για την εμφάνιση ειδοποιήσεων χρησιμοποιείται το Android AlarmManager και η κλάση BroadcastReceiver. Για να λάβουμε το query της ειδοποίησης αναπτύξαμε την κλάση Notification.java.

## 6.2.6 Περιγραφή του πίνακα Notification

**received** – Λαμβάνει τιμές true/false. Το "False" αλλάζει σε "True" μόλις εμφανιστεί ειδοποίηση σε μια οθόνη χρήστη.

**sender** – Email του Υπευθύνου Ομάδας Απόσυρσης.

**vin** – Αριθμός Αδείας του Οχήματος.

**vntype** – Λαμβάνει τιμές στο διάστημα [0, 4] που αντιστοιχούν σε διαφορετικούς τύπους ειδοποιήσεων. Π.χ.: "Accepted", "Rejected", "Withdrawal image uploaded" κ.λπ.

Η κλάση MyNotificationPublisher που επεκτείνει την BroadcastReceiver, χειρίζεται τις ειδοποιήσεις με κατάλληλη ρύθμιση των ιδιοτήτων των αντικειμένων των κλάσεων NotificationManager και NotificationChannel.

```

1 package com.parrot.vehicles.auth.java;
2
3 import ...
9 public class MyNotificationPublisher extends BroadcastReceiver {
10     public static String NOTIFICATION_ID = "notification-id" ;
11     public static String NOTIFICATION = "notification" ;
12     public void onReceive (Context context , Intent intent) {
13         NotificationManager notificationManager = (NotificationManager) context.getSystemService(Context. NOTIFICATION_SERVICE )
14         Notification notification = intent.getParcelableExtra( NOTIFICATION ) ;
15         if (android.os.Build.VERSION.SDK_INT >= android.os.Build.VERSION_CODES. O ) {
16             int importance = NotificationManager. IMPORTANCE_HIGH ;
17             NotificationChannel notificationChannel = new NotificationChannel( id: "10001" , name: "NOTIFICATION_CHANNEL_NAME" ,
18                 assert notificationManager != null;
19                 notificationManager.createNotificationChannel(notificationChannel) ;
20         }
21         int id = intent.getIntExtra( NOTIFICATION_ID , defaultValue: 0 ) ;
22         assert notificationManager != null;
23         //notificationManager.notify(id , notification) ;
24     }
25 }

```

Κώδικας 6.10 Η κλάση MyNotificationPublisher

```

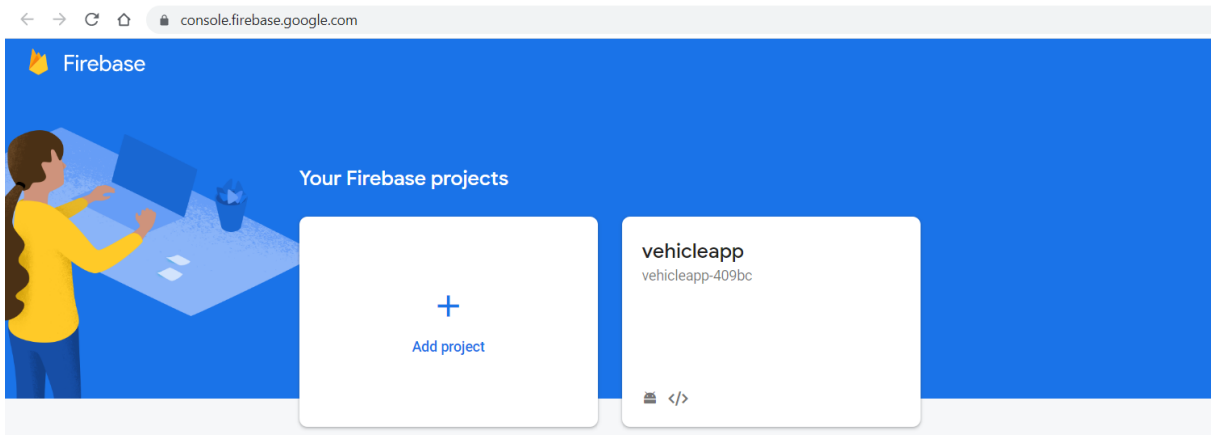
36 protected void onCreate(Bundle savedInstanceState) {...}
139 private void scheduleNotification (String content, String vinStatus, Intent resultIntent) {
140     Intent notificationIntent = new Intent( packageContext: this, MyNotificationPublisher. class ) ;
141     notificationIntent.putExtra( NOTIFICATION_ID , value: 1 ) ;
142     pendingIntent = PendingIntent.getBroadcast ( context: this, requestCode: 0 , notificationIntent , PendingIntent.FLAG_CANCEL
143     long futureInMillis = SystemClock.elapsedRealtime () ;
144     AlarmManager alarmManager = (AlarmManager) getSystemService(Context. ALARM_SERVICE ) ;
145     assert alarmManager != null;
146     alarmManager.set(AlarmManager. ELAPSED_REALTIME_WAKEUP , futureInMillis , pendingIntent) ;
147
148     resultIntent.putExtra( name: "vNumber", vinStatus);
149     TaskStackBuilder stackBuilder = TaskStackBuilder.create(this);
150     stackBuilder.addNextIntentWithParentStack(resultIntent);
151     PendingIntent resultPendingIntent = stackBuilder.getPendingIntent( requestCode: 0 , PendingIntent.FLAG_UPDATE_CURRENT);
152     NotificationCompat.Builder builder = new NotificationCompat.Builder( context: this, NOTIFICATION_CHANNEL_ID);
153     builder.setContentIntent(resultPendingIntent);
154     builder.setContentTitle( "Ειδοποίηση Οχήματος" ) ;
155     builder.setContentText(content) ;
156     builder.setSmallIcon(R.drawable.ic_person_white) ;
157     builder.setAutoCancel( false ) ;
158     builder.setChannelId( NOTIFICATION_CHANNEL_ID ) ;
159     NotificationManagerCompat notificationManager = NotificationManagerCompat.from(this);
160     notificationManager.notify( id: 1, builder.build());
161 }
162 }

```

Κώδικας 6.11 Η κλάση MyAppCompatActivity

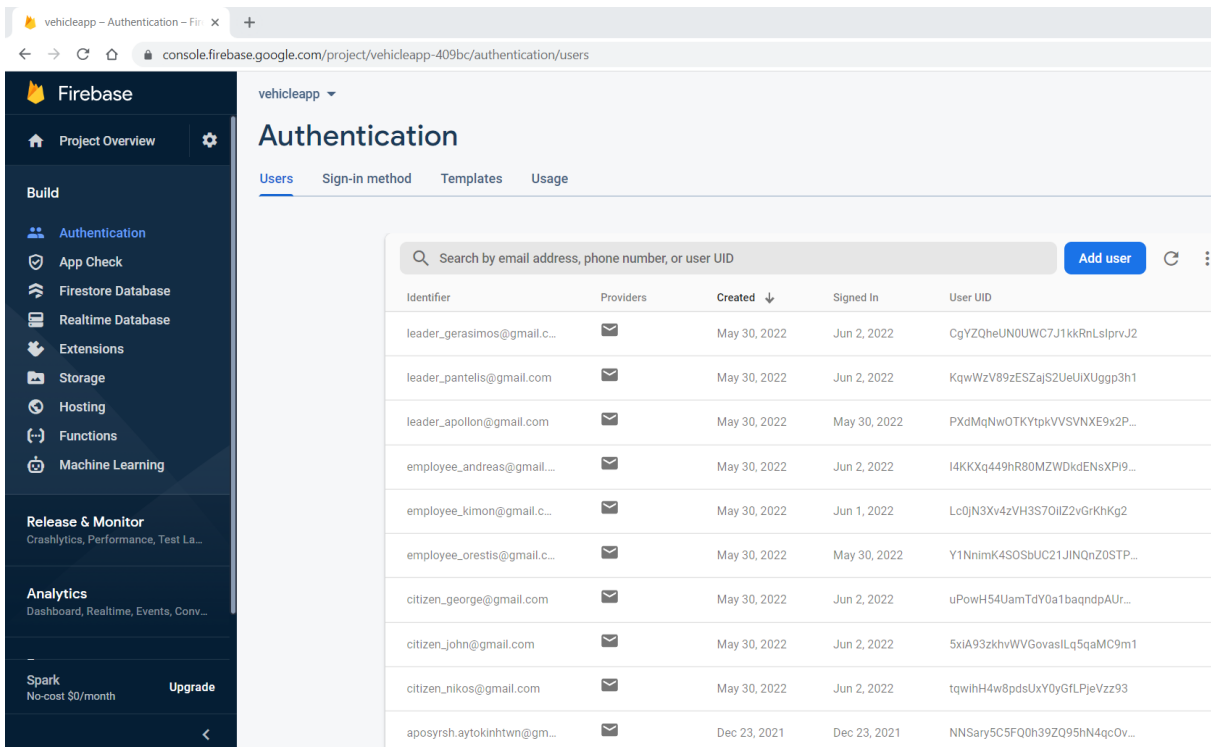
### 6.3 Πρόσβαση Χρηστών και Firebase Authentication Panel

Για την αυθεντικοποίηση των χρηστών από κάποιον Super Admin, η αρχική ιδέα ήταν να δημιουργηθεί ένα Web-admin Panel. Παρόλα αυτά προτιμήθηκε να χρησιμοποιηθεί το Firebase ως μία πιο ασφαλής και συγκεντρωτική λύση. Για να λάβει πρόσβαση στο Firebase Console, ο Super Admin θα πρέπει πρώτα να κάνει μία εγγραφή στο Google Cloud (που είναι δωρεάν κατά τη συγγραφή της εργασίας). Το email σύνδεσης θα πρέπει να είναι αυτό που θα θεωρείται το email του ιδιοκτήτη της εφαρμογής στο Firebase Console. Έτσι, μόλις γίνει η εγγραφή στο <https://cloud.google.com/> και η σύνδεση στο <https://console.firebase.google.com/> μπορούμε να δούμε το Project μας (vehicleApp), όπως φαίνεται και στην παρακάτω εικόνα.



Εικόνα 6.5 Σύνδεση στο Project vehicleapp του Firebase Console για Αυθεντικοποίηση χρηστών

Στην αριστερή πλαϊνή μπάρα του μενού πατάμε κλικ στο κουμπί «Authentication», ώστε να εμφανιστεί μια λίστα με όλους τους χρήστες που έχουν εγγραφεί στην εφαρμογή.



Εικόνα 6.6 Λίστα των εγγεγραμμένων χρηστών προς Αυθεντικοποίηση

Από εκεί ο Super Admin έχει τη δυνατότητα να διαγράψει, να απενεργοποιήσει ή να επαναφέρει ένα λογαριασμό χρήστη, όπως φαίνεται στην παραπάνω εικόνα σε κάθε γραμμή χρήστη.

Για να επαληθεύσουμε έναν χρήστη, π.χ. κάποιον Υπάλληλο του Δήμου ή Υπεύθυνο Ομάδας, αρχικά πηγαίνουμε στην επιλογή "Realtime Database" και κάτω από τη βάση δεδομένων Role βλέπουμε μία λίστα χρηστών με τα email, τους ρόλους και το status τους στο πεδίο "verified". Το τελευταίο παίρνει τιμές true/false. Απλώς αλλάζοντας αυτό το πεδίο, επιτρέπουμε ή αποτρέπουμε την πρόσβαση στον κάθε χρήστη.

## 6.4 Προβλήματα κατά το testing, προσθήκες, διορθώσεις και αλλαγές

Σε διάφορα σημεία ανάπτυξης της εφαρμογής έγιναν δοκιμές λειτουργίας της. Επίσης, έγινε και ένας τελικός έλεγχος για πιθανά μικροπροβλήματα που μπορεί να διέφυγαν από τις προηγούμενες δοκιμές. Κατά το testing εντοπίστηκαν και διορθώθηκαν διάφορα θέματα. Για την ολοκλήρωση της τελικής εφαρμογής χρειάστηκαν διάφορες προσθήκες, αλλαγές και διορθώσεις τόσο όσον αφορά τη λογική, όσο και τεχνικές που αφορούσαν τη λειτουργικότητα. Πιο συγκεκριμένα:

- **Geolocation:** Ένα πρόβλημα λογικής που αντιμετωπίσαμε ήταν ότι οι ρυθμίσεις δικαιωμάτων για το Geolocation. Αρχικά, τοποθετήθηκαν στην οθόνη «Καταχώρησης Οχήματος» ενώ έπρεπε να τοποθετηθούν στην αρχική οθόνη της εφαρμογής. Ως εκ τούτου, η εγγραφή οχήματος δε λειτουργούσε. Το πρόβλημα διορθώθηκε μετά από εκτενέστερο έλεγχο και κατάλληλη αλλαγή της λογικής της εφαρμογής.
- **Εικόνες / Μεταφορτώσεις / Κάμερα / Γκαλερί**

**Λήψη:** Η αποθήκευση των φωτογραφιών οχήματος έπρεπε να γίνει στον server. Χρησιμοποιήσαμε για αποθήκευση το Firebase Storage, γι' αυτό και αποθηκεύαμε το όνομα αρχείου στη Βάση Δεδομένων. Όμως για να συγχρονιστούν με όλους τους χρήστες έπρεπε να διατηρήσουμε το path επίσης αποθηκευμένο στη Βάση Δεδομένων. Επομένως, επιλέξαμε η διεύθυνση URL να μπαίνει στη Βάση Δεδομένων και τώρα η λήψη εικόνων λειτουργεί κανονικά.

**Μεταφόρτωση:** Το ζητούμενο ήταν να ανεβάζουμε τη φωτογραφία που έβγαζε ο χρήστης σε μια κεντρική Βάση Δεδομένων, έτσι ώστε να είναι προσβάσιμη από τους άλλους χρήστες, άρα και να έχει μία πλήρη διαδρομή με μοναδικό όνομα. Το Firebase Storage παρέχει αυτή τη λειτουργία για την επιτυχή μεταφόρτωση. Έτσι, χρησιμοποιήσαμε αυτό το χαρακτηριστικό του Firebase. Νωρίτερα προσπαθήσαμε με το Fileoutput Stream για να αποθηκεύσουμε στον server, τεχνική που αποδείχτηκε μη αποτελεσματική.

- **Διεπαφή Χρήστη**

**SlideBars:** Όταν κατά το testing, ο αριθμός των οχημάτων αυξανόταν, τότε παρατηρούσαμε ότι κάποια οχήματα κρύβονταν στην οθόνη. Αναγνωρίσαμε ότι το πρόβλημα ήταν η μη υλοποίηση του scrolling. Για την αντιμετώπιση αυτού του προβλήματος έχουμε κάνει wrap όλη την οθόνη καταχώρησης οχημάτων σε ένα Scrollview. Έτσι, τώρα όλα λειτουργούν κανονικά και οι εικόνες μπορούν να προβληθούν.

**Το User Interface (UI) προβολής της κατάστασης για τους Πολίτες:** Ο Πολίτης έπρεπε να μπορεί να παρακολουθεί την κατάσταση των οχημάτων. Προσπαθήσαμε να αναπτύξουμε ένα UI που να είναι απλό και ενημερωτικό. Δοκιμάσαμε πολλά διαφορετικά UI και καταλήξαμε σε ένα UI όπου δείχνουμε όλη την κατάσταση και τονίζουμε ποια είναι αυτή τη στιγμή ενεργή.

**Μπάρα προόδου:** Αναλόγως με το μέγεθος αρχείου, η λήψη μπορούσε να είναι είτε γρήγορη είτε αργή. Έτσι, προσθέσαμε μια μπάρα προόδου που όταν επιλέγεται από τον χρήστη να ανοίγει ένα πρόγραμμα εξερεύνησης αρχείων, έτσι ώστε να είναι πλέον εύκολο για τον ίδιο να δει ότι έχει γίνει λήψη.

**Τιμές τρέχουσας κατάστασης οχημάτων:** Κατά την ανάπτυξη της εν λόγω λειτουργικότητας, αντιμετωπίσαμε δυσκολίες με το status των οχημάτων επειδή με τη ροή δεδομένων χρήστη δεν ήταν σαφές πόσες καταστάσεις απαιτούνται στην αρχή. Υπολογίσαμε ότι θα υπάρχουν πέντε (5) πιθανές τιμές κατάστασης, αλλά αργότερα, καθώς εξελισσόταν η εφαρμογή, καταλάβαμε ότι η λογική θα ήταν σωστότερη να γίνει με συνολικά εννέα (9) τιμές κατάστασης και με βάση την κατάσταση να εμφανίζονται διαφορετικές επιλογές και οθόνες για όλους τους χρήστες.

**Ειδοποιήσεις:** Αντιμετωπίσαμε διάφορα προβλήματα με τις ειδοποιήσεις οι οποίες στην αρχή ήταν δύσκολο να λειτουργήσουν και έπειτα υπήρχε δυσκολία φόρτωσης των πληροφοριών οχημάτων. Με το πάτημα ειδοποίησης έπρεπε να εμφανίζεται η σωστή οθόνη και ο αριθμός άδειας του οχήματος έπρεπε να περάσει στην οθόνη από την ειδοποίηση ώστε να εμφανιστούν τα σωστά δεδομένα, πράγμα που τελικά έγινε.

**Οργάνωση / Καθημερινή απόσυρση:** Αρχικά, η λογική της εφαρμογής ήταν διαφορετική. Πλέον υπάρχει ξεχωριστή οθόνη για την Οργάνωση των καθημερινών αποσύρσεων.

## 6.5 Επίλογος

Στο τελευταίο κεφάλαιο περιγράφηκαν τα στάδια του προγράμματος για τη διαχείριση απόσυρσης οχημάτων. Ο γενικός σκοπός της εργασίας, που είναι η εφαρμογή να αποτελέσει ουσιαστικό εργαλείο στις αρμόδιες υπηρεσίες των Δημοτικών Αρχών, μπορεί να εκπληρωθεί με την ορθή χρήση του συγκεκριμένου προγράμματος από τους υπεύθυνους. Ο κάθε χρήστης μπορεί εύκολα να το χρησιμοποιήσει σύμφωνα με τις οδηγίες που παρατίθενται στο παράρτημα.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Android, «[el.wikipedia.org](https://el.wikipedia.org/wiki/Android)» [Ηλεκτρονικό]. Available: <https://el.wikipedia.org/wiki/Android>.
- [2] Γαβαλάς Δαμιανός, Κασαπάκης Βλάσης, Χατζηδημήτρης Θωμάς, Κινητές Τεχνολογίες: Κινητός Ιστός - Κινητές Εφαρμογές στην Πλατφόρμα Android - Επαυξημένη Πραγματικότητα, Εκδόσεις Νέων Τεχνολογιών: Έκδοση 1η, Αθήνα, 2015.
- [3] Μπαλαούρας Παντελής, Εισαγωγή στην πλατφόρμα Android, Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών [Ηλεκτρονικό]. Available: <https://docplayer.gr/59346217-Dialexi-eisagogi-stin-platforma-android-eisigitis-pantelis-mpalaouyras.html>.
- [4] Android Studio, «[JavaTpoint](https://www.javatpoint.com/android-studio)» [Ηλεκτρονικό]. Available: <https://www.javatpoint.com/android-studio>.
- [5] IBM Cloud Education, What is Java?, 8 May, 2019 [Ηλεκτρονικό]. Available: <https://www.ibm.com/cloud/learn/java-explained>.
- [6] Subham B., Aditi K., Madhuleena M., Madhurima B., A comparative study: Java vs kotlin Programming in Android Application Development, May-June 2018 [Ηλεκτρονικό]. Available: <https://pdfs.semanticscholar.org/c0ee/43434064520cdde7222318bf6c4d2db69177.pdf>.
- [7] Learn C++ Programming, «[Programiz](https://www.programiz.com/cpp-programming)» [Ηλεκτρονικό]. Available: <https://www.programiz.com/cpp-programming>.
- [8] Χάρτες Google, «[el.wikipedia.org](https://el.wikipedia.org/wiki/Χάρτες_Google)» [Ηλεκτρονικό]. Available: [https://el.wikipedia.org/wiki/Χάρτες\\_Google](https://el.wikipedia.org/wiki/Χάρτες_Google).
- [9] Street View, «[Google Χάρτες](https://www.google.gr/intl/el/streetview/)» [Ηλεκτρονικό]. Available: <https://www.google.gr/intl/el/streetview/>.
- [10] Frankenfield Jake, Geolocation, 25 Jan, 2021 [Ηλεκτρονικό]. Available: <https://www.investopedia.com/terms/g/geolocation.asp>.
- [11] Wayner Peter, What is Firebase?, 2 Aug, 2021 [Ηλεκτρονικό]. Available: <https://venturebeat.com/2021/08/02/what-is-firebase/>.
- [12] Chunnu K., Pritam S., Application of Firebase in Android App Development - A Study, Volume 179 – No.46, June 2018 [Ηλεκτρονικό]. Available: [https://www.researchgate.net/profile/Chunnu-Khawas/publication/325791990\\_Application\\_of\\_Firebase\\_in\\_Android\\_App\\_Development-A\\_Study/links/5bab55ed45851574f7e6801e/Application-of-Firebase-in-Android-App-Development-A-Study.pdf](https://www.researchgate.net/profile/Chunnu-Khawas/publication/325791990_Application_of_Firebase_in_Android_App_Development-A_Study/links/5bab55ed45851574f7e6801e/Application-of-Firebase-in-Android-App-Development-A-Study.pdf).
- [13] Moroney Laurence, Using Authentication in Firebase, Seattle, 11 Nov, 2017 [Ηλεκτρονικό]. Available: [https://link.springer.com/chapter/10.1007/978-1-4842-2943-9\\_2](https://link.springer.com/chapter/10.1007/978-1-4842-2943-9_2).
- [14] Firebase Authentication, «[Firebase](https://firebase.google.com/docs)» [Ηλεκτρονικό]. Available: <https://firebase.google.com/docs>.
- [15] Firebase: Realtime Database, «[JavaTpoint](https://www.javatpoint.com/firebase-realtime-database)» [Ηλεκτρονικό]. Available: <https://www.javatpoint.com/firebase-realtime-database>.
- [16] Σαλαμπάσης Μιχάλης, Εισαγωγή στον Προγραμματισμό Διαδικτυακών Εφαρμογών, Έκδοση 2η, Feb 2017.

[17] Schwaber Ken, Sutherland Jeff, Ο Οδηγός του Scrum - Ο απόλυτος οδηγός για το Scrum: Οι κανόνες του παιχνιδιού, Jul 2013 [Ηλεκτρονικό]. Available: <https://scrumguides.org/docs/scrumguide/v1/Scrum-Guide-GR.pdf>.

[18] Schwaber Ken, Sutherland Jeff, Ο Οδηγός του Scrum - Ο απόλυτος οδηγός για το Scrum: Οι κανόνες του παιχνιδιού, Nov 2020 [Ηλεκτρονικό]. Available: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Greek.pdf>.

[19] Narechania Nishal, The Fundamentals of User Stories and Product Backlogs, 15 Jul, 2019 [Ηλεκτρονικό]. Available: <https://medium.com/nsquared-labs/user-stories-and-the-product-backlog-in-scrum-c87d36df4b96>.

[20] What is Planning Poker in Agile?, «Visual-Paradigm» [Ηλεκτρονικό]. Available: <https://www.visual-paradigm.com/scrum/what-is-agile-planning-poker/>.

# ΠΑΡΑΡΤΗΜΑ: ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΤΗ

## Οθόνη Εγγραφής



Απόσυρση Οχημάτων

Απόσυρση Οχημάτων

Δημιουργία Λογαριασμού

Email

Password

Πολίτης του Δήμου

ΕΓΓΡΑΦΗ

Έχετε λογαριασμό; Συνδεθείτε εδώ!



Απόσυρση Οχημάτων

Απόσυρση Οχημάτων

Πολίτης του Δήμου

Υπάλληλος του Δήμου

Υπεύθυνος Ομάδας Απόσυρσης

Πολίτης του Δήμου

ΕΓΓΡΑΦΗ

Έχετε λογαριασμό; Συνδεθείτε εδώ!

Για να δημιουργήσετε έναν δικό σας λογαριασμό, στην οθόνη εγγραφής εισάγετε το email και τον κωδικό πρόσβασης που επιθυμείτε και επιλέξτε έναν από τους ρόλους που εμφανίζονται στο αναδυόμενο μενού («Πολίτης του Δήμου», «Υπάλληλος του Δήμου» ή «Υπεύθυνος Ομάδας Απόσυρσης»). Έπειτα, πατήστε «ΕΓΓΡΑΦΗ». Για τους χρήστες που επιλέγουν το ρόλο του **Πολίτη του Δήμου**, ο λογαριασμός δημιουργείται αυτόματα και μπορούν να χρησιμοποιήσουν την εφαρμογή. Για τους χρήστες που είναι **Υπάλληλοι του Δήμου** ή **Υπεύθυνοι Ομάδας Απόσυρσης** η διαδικασία είναι ελαφρώς διαφοροποιημένη, καθώς για να είναι σε θέση να χρησιμοποιήσουν το λογαριασμό τους θα πρέπει πρώτα η εγγραφή να επαληθευτεί από τον διαχειριστή της εφαρμογής.

## Σύνδεση Χρήστη

Απόσυρση Οχημάτων



Απόσυρση Οχημάτων

Είσοδος στην εφαρμογή

Email

Password


ΣΥΝΔΕΣΗ

Δεν έχετε λογαριασμό; Εγγραφείτε εδώ!

Στην οθόνη εισόδου της εφαρμογής εισάγετε το email και τον κωδικό πρόσβασης που επιλέξατε προηγουμένως. Ανάλογα με το ρόλο που έχετε και αν είστε επαληθευμένος χρήστης («Υπάλληλος του Δήμου» ή «Υπεύθυνος Ομάδας Απόσυρσης») θα μπορέσετε να συνδεθείτε.

## Αρχική Οθόνη

☰ Απόσυρση Οχημάτων



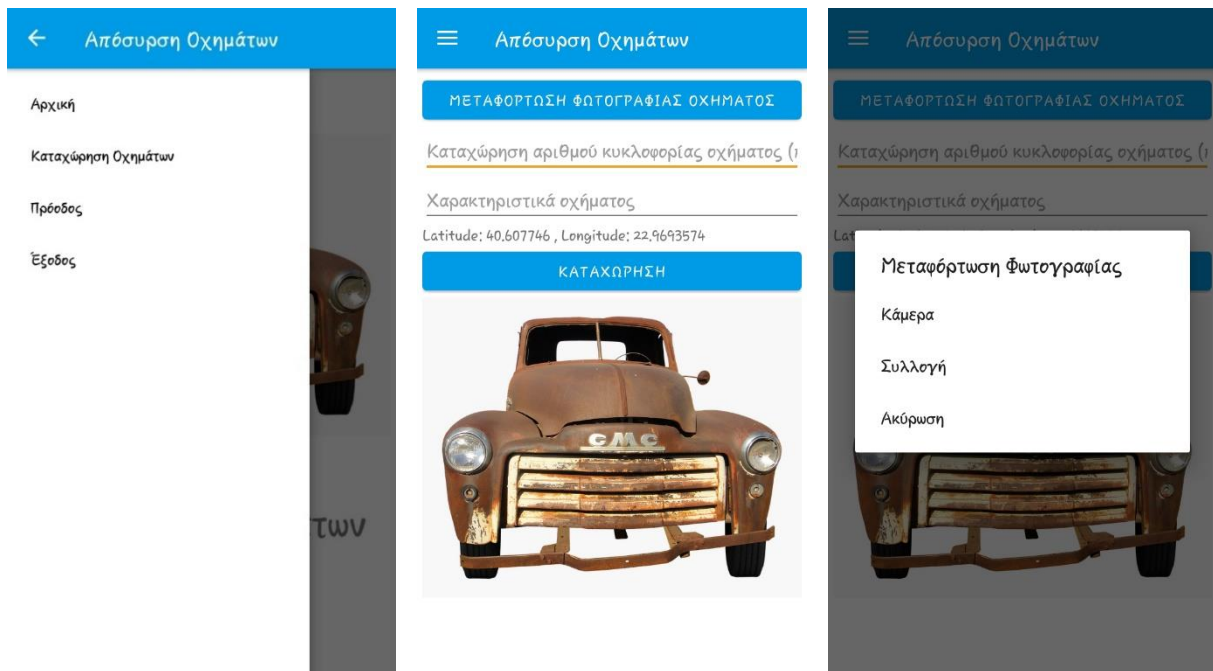
Απόσυρση Οχημάτων

Ρόλος: Πολίτης του Δήμου

Με βάση τον ρόλο που σας έχει ανατεθεί, θα εμφανίζεται η αντίστοιχη αρχική οθόνη.

## Πολίτης του Δήμου

### Καταχώρηση οχήματος



Από το Drawer Menu κάντε κλικ στην επιλογή «Καταχώρηση Οχημάτων» και συμπληρώστε τα πεδία. Κάντε κλικ στην επιλογή «Μεταφόρτωση Φωτογραφίας Οχήματος» και επιλέξτε μια εικόνα τραβώντας την από τη φωτογραφική μηχανή του κινητού Android ή τη συλλογή φωτογραφιών σας.



Ύστερα από την επιλογή της εικόνας και τη συμπλήρωση των πεδίων, θα εμφανιστούν το εγγεγραμμένο όχημα και το μήνυμα «Το όχημα καταχωρήθηκε» στην ίδια οθόνη λίγο παρακάτω.

## Πρόοδος της διαδικασίας απόσυρσης οχήματος

← Απόσυρση Οχημάτων

- Αρχική
- Καταχώρηση Οχημάτων
- Πρόοδος
- Έξοδος

☰ Απόσυρση Οχημάτων

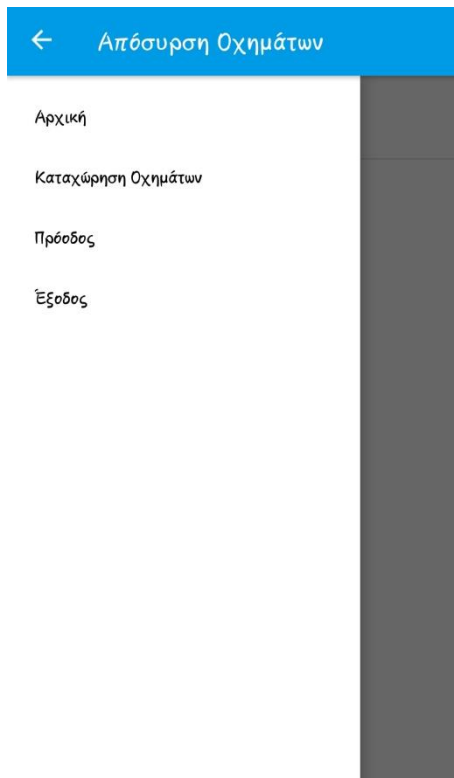
X026538 : Καταχωρήθηκε - Αναμένει  
Επαλήθευση στοιχείων

Από το Drawer Menu κάντε κλικ στην επιλογή «Πρόοδος» για να δείτε τη λίστα των καταχωρημένων οχημάτων και την τρέχουσα κατάστασή τους. Έπειτα, μπορείτε να επιλέξετε ένα οποιοδήποτε όχημα.

☰ Απόσυρση Οχημάτων	☰ Απόσυρση Οχημάτων	☰ Απόσυρση Οχημάτων
<b>X026538</b>	<b>X026538</b>	<b>X026538</b>
Καταχωρήθηκε - Αναμένει επαλήθευση στοιχείων	Καταχωρήθηκε - Αναμένει επαλήθευση στοιχείων	Καταχωρήθηκε - Αναμένει επαλήθευση στοιχείων
Επαλήθευση στοιχείων οχήματος	Επαλήθευση στοιχείων οχήματος	Επαλήθευση στοιχείων οχήματος
Έγκριση απόσυρσης	Έγκριση απόσυρσης	Έγκριση απόσυρσης
Ανάθεση απόσυρσης	Ανάθεση απόσυρσης	Ανάθεση απόσυρσης
Αποδοχή ανάθεσης απόσυρσης από ειδική ομάδα	Αποδοχή ανάθεσης απόσυρσης από ειδική ομάδα	Αποδοχή ανάθεσης απόσυρσης από ειδική ομάδα
Τελική έγκριση απόσυρσης	Τελική έγκριση απόσυρσης	Τελική έγκριση απόσυρσης
Ολοκλήρωση ειδοποίησης απόσυρσης	Ολοκλήρωση ειδοποίησης απόσυρσης	Ολοκλήρωση ειδοποίησης απόσυρσης
Το όχημα έχει αποσυρθεί	Το όχημα έχει αποσυρθεί	Το όχημα έχει αποσυρθεί
Ολοκλήρωση απόσυρσης	Ολοκλήρωση απόσυρσης	Ολοκλήρωση απόσυρσης

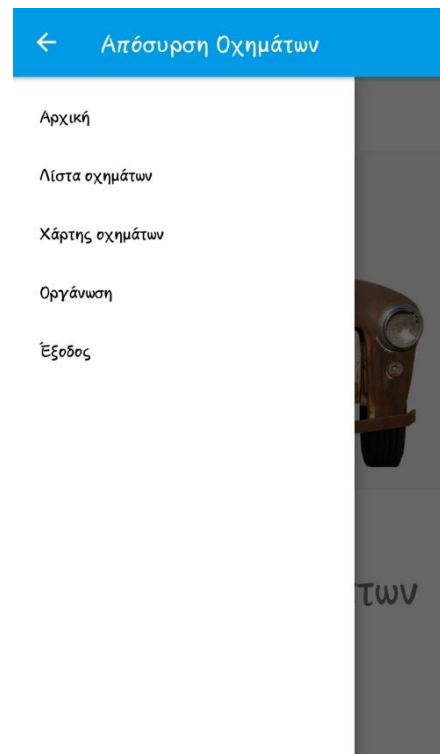
Ο Πολίτης έχει τη δυνατότητα να παρακολουθεί τη συνολική λίστα προόδου της απόσυρσης καθώς και την τρέχουσα κατάσταση του επιλεγμένου οχήματος, όπως φαίνεται παραπάνω.

## Αποσύνδεση



Μπορείτε να πραγματοποιήσετε «Έξοδο» από την εφαρμογή με την αντίστοιχη επιλογή από το Drawer Menu.

## Υπάλληλος του Δήμου



Μετά τη σύνδεση ως «Υπάλληλος του Δήμου» εμφανίζεται η αρχική οθόνη για τον Υπάλληλο. Από το Drawer Menu επιλέξτε «Λίστα οχημάτων».

☰ Απόσυρση Οχημάτων

Καταχωρήθηκε - Αναμένει Επαλήθευση σ., ▾

Χ026538 : Καταχωρήθηκε - Αναμένει Επαλήθευση στοιχείων

☰ Απόσυρση Οχημάτων

Καταχωρήθηκε - Αναμένει Επαλήθευση στοιχείων

Επαλήθευση στοιχείων οχήματος

Έγκριση Απόσυρσης

Ανάθεση Απόσυρσης

Αποδοχή Ανάθεσης Απόσυρσης από ειδική ομάδα

Τελική έγκριση απόσυρσης


Ολοκλήρωση Ειδοποίησης Απόσυρσης

Το όχημα έχει αποσυρθεί

Ολοκλήρωση απόσυρσης

Από το Drop Down μενού μπορείτε να επιλέξετε με ποιο στάδιο απόσυρσης αυτοκινήτων θέλετε να ασχοληθείτε.

☰ Απόσυρση Οχημάτων



Παρατημένο, σκουριασμένο, μπλε λεωφορείο

Ταυτότητα ιδιοκτήτη


Αριθμός πλαισίου μηχανής

Άδεια κυκλοφορίας

Πινακίδες (εφόσον το όχημα φέρει)

ΕΠΑΛΗΘΕΥΣΗ ΣΤΟΙΧΕΙΩΝ  
ΕΓΚΑΤΑΛΕΛΕΙΜΜΕΝΟΥ ΟΧΗΜΑΤΟΣ

☰ Απόσυρση Οχημάτων



Παρατημένο, σκουριασμένο, μπλε λεωφορείο

ΟΡΦΕΑΣ ΠΑΠΑΝΤΩΝΙΟΥ

26H0J1PF7GA438672

ΝΑΙ

Χ026538 Το όχημα επαληθεύτηκε.

ΕΠΑΛΗΘΕΥΣΗ ΣΤΟΙΧΕΙΩΝ  
ΕΓΚΑΤΑΛΕΛΕΙΜΜΕΝΟΥ ΟΧΗΜΑΤΟΣ

Πατώντας στην πρώτη επιλογή «Καταχωρήθηκε – Αναμένει Επαλήθευση» μπορείτε να δείτε τα νέα οχήματα που έχουν καταχωρηθεί από τους Πολίτες. Επιλέξτε ένα όχημα και προσθέστε τις απαραίτητες πληροφορίες προς επαλήθευση. Ύστερα, κάντε κλικ στο αντίστοιχο κουμπί «Επαλήθευση Στοιχείων».

☰ Απόσυρση Οχημάτων

X026538



Στη συνέχεια, μπορείτε να εγκρίνετε την απόσυρση του οχήματος από το αντίστοιχο μενού και με το αντίστοιχο κουμπί «Έγκριση Απόσυρσης Οχήματος», όπως φαίνεται στη διπλανή εικόνα.

Παρατημένο, σκουριασμένο, μπλε λεωφορείο  
ΟΡΦΕΑΣ ΠΑΠΑΝΤΩΝΙΟΥ  
26H0J1PF7GA438672  
ΝΑΙ

X026538 Το όχημα εγκρίθηκε προς απόσυρση.

ΕΓΚΡΙΣΗ ΑΠΟΣΥΡΣΗΣ ΟΧΗΜΑΤΟΣ

☰ Απόσυρση Οχημάτων



Στο επόμενο βήμα, που είναι η «Έγκριση Απόσυρσης», μπορείτε πλέον να επιλέξετε «Υπεύθυνο Ομάδας Απόσυρσης» και να του αναθέσετε την απόσυρση του οχήματος.

Παρατημένο, σκουριασμένο, μπλε λεωφορείο

lead Ο Υπεύθυνος Ομάδας Απόσυρσης έχει οριστεί.

ΑΝΑΘΕΣΗ ΑΠΟΣΥΡΣΗΣ  
ΕΓΚΑΤΑΛΕΛΕΙΜΜΕΝΟΥ ΟΧΗΜΑΤΟΣ ΣΕ  
ΕΙΔΙΚΗ ΟΜΑΔΑ

☰ Απόσυρση Οχημάτων

Ανάθεση Απόσυρσης ▾

X026538 : Ανάθεση Απόσυρσης

☰ Απόσυρση Οχημάτων

X026538



Παρατημένο, σκουριασμένο, μπλε λεωφορείο

leader\_apollo@gmail.com

Ο «Υπεύθυνος Ομάδας Απόσυρσης» που έχει επιλεγεί για το εκάστοτε όχημα, μπορεί να φανεί στο επόμενο βήμα-μενού το οποίο είναι η «Ανάθεση Απόσυρσης».

Μόλις ο «Υπεύθυνος Ομάδας Απόσυρσης» αποδεχτεί ή απορρίψει την απόσυρση του οχήματος, ο «Υπάλληλος του Δήμου» λαμβάνει ειδοποίηση και βλέπει την κατάσταση.

☰ Απόσυρση Οχημάτων 17:25

Ειδοποίηση Οχήματος

Ο Υπεύθυνος leader\_apollo@gmail.com αποδέχθηκε την..



Απόσυρση Οχημάτων

Ρόλος: Υπάλληλος του Δήμου

☰ Απόσυρση Οχημάτων

X026538

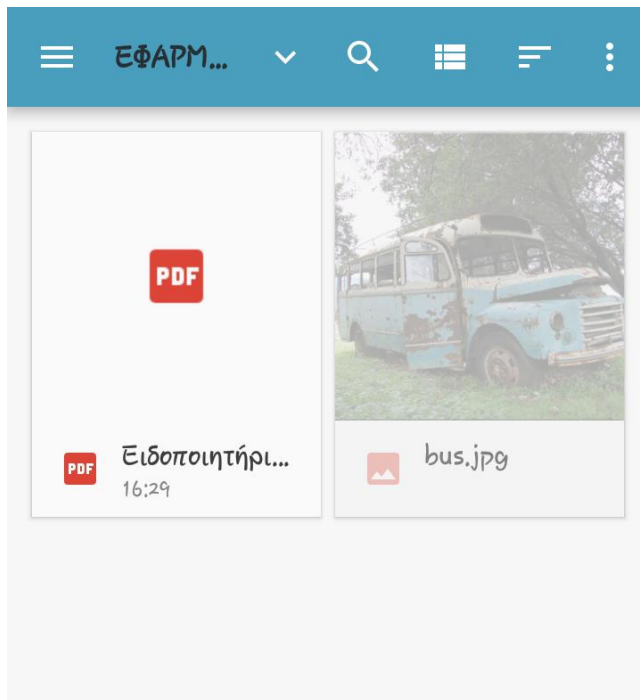


Παρατημένο, σκουριασμένο, μπλε λεωφορείο

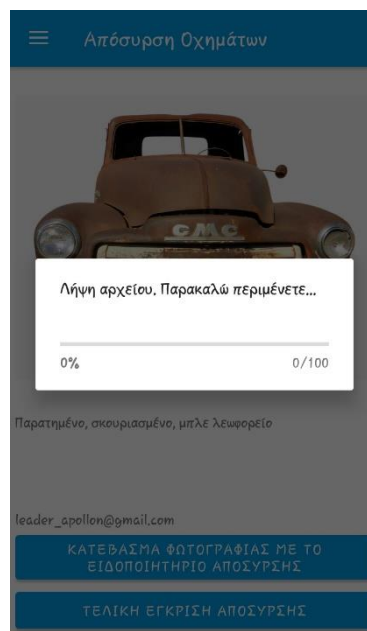
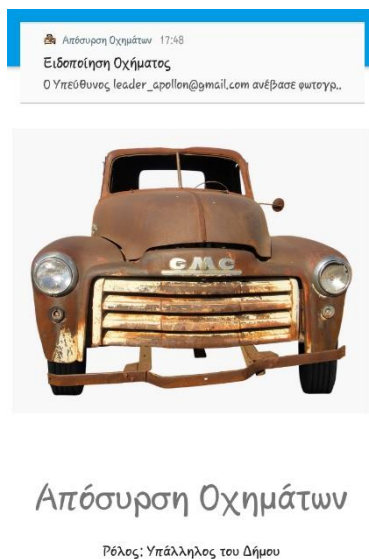
leader\_apollo@gmail.com

ΑΠΟΣΤΟΛΗ ΕΙΔΟΠΟΙΗΤΗΡΙΟΥ ΑΠΟΣΥΡΣΗΣ

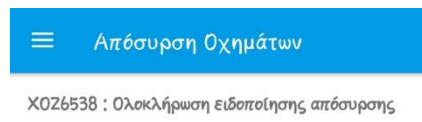
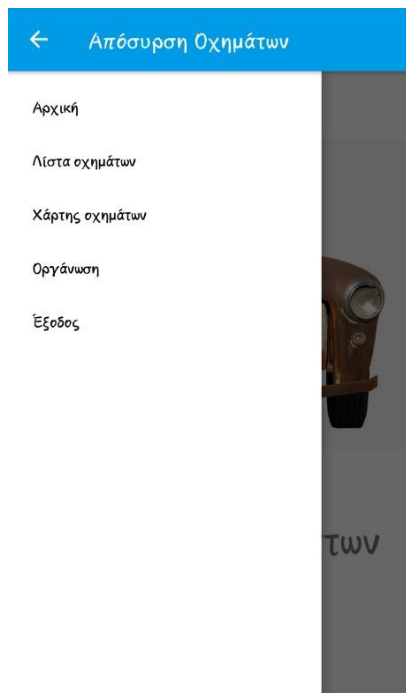
Σε αυτό το στάδιο, ο «Υπάλληλος του Δήμου» μπορεί να ανεβάσει το αντίστοιχο ειδοποιητήριο σε PDF, ώστε να μπορεί να το κατεβάσει ο «Υπεύθυνος Ομάδας Απόσυρσης».



Μόλις ο Υπάλληλος ανεβάσει το PDF, εμφανίζεται η ένδειξη «Μεταφορτώθηκε με επιτυχία». Τώρα, ο «Υπεύθυνος Ομάδας Απόσυρσης» μπορεί να κατεβάσει το PDF και να το κολλήσει στο όχημα, στη συνέχεια να πάρει τη φωτογραφία απόσυρσης μετά από μερικές ημέρες και να την ανεβάσει. Τότε, ο «Υπάλληλος του Δήμου» θα ειδοποιηθεί, όπως φαίνεται παρακάτω:



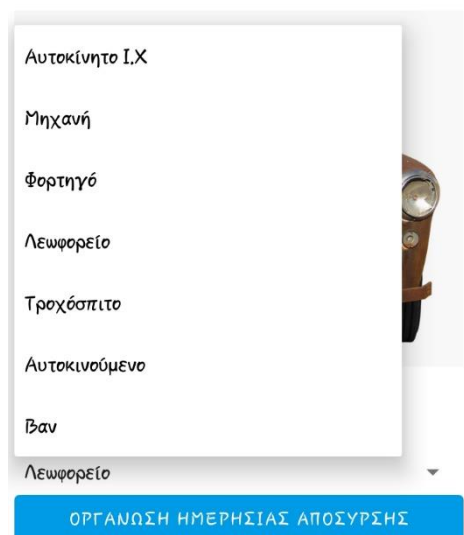
Σε αυτό το σημείο, ο Υπάλληλος κατεβάζει τη φωτογραφία με το ειδοποιητήριο της απόσυρσης περιμένοντας να ολοκληρωθεί η λήψη του αρχείου. Έπειτα, εγκρίνοντας την απόσυρση εμφανίζεται το μήνυμα «Η κατάσταση του οχήματος είναι έτοιμη για απόσυρση». Τέλος, κάνει κλικ στο κουμπί «Οργάνωση» από το Drawer Menu.



Επιλέξτε ένα από τα οχήματα που εμφανίζονται στη λίστα και επιλέξτε μια κατηγορία οχήματος, όπως φαίνεται παρακάτω. Έπειτα, πατώντας το κουμπί «Οργάνωση Ημερήσιας Απόσυρσης» ο «Υπεύθυνος Ομάδας Απόσυρσης» που έχει οριστεί ειδοποιείται κατάλληλα και μπορεί να κάνει τη φυσική απόσυρση στο επόμενο στάδιο.



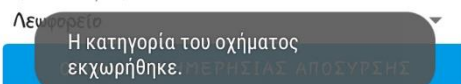
Χ026538



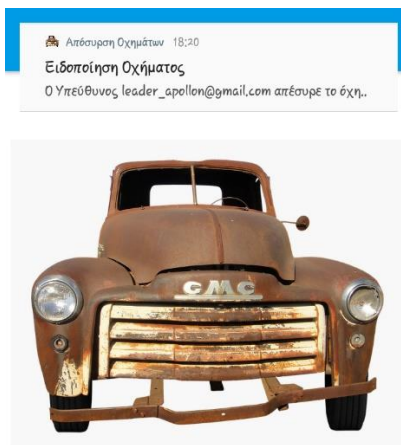
Χ026538



Παρατημένο, σκουριασμένο, μπλε λεωφορείο  
40.6077644, 22.9693772

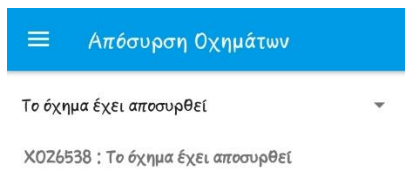


Μόλις το όχημα αποσυρθεί φυσικά από τον «Υπεύθυνο Ομάδας Απόσυρσης» και ενημερωθεί στην εφαρμογή, ο Υπάλληλος λαμβάνει την αντίστοιχη ενημέρωση και έπειτα μπορεί να βρει το συγκεκριμένο όχημα με βάση την παρακάτω κατάσταση:



## Απόσυρση Οχημάτων

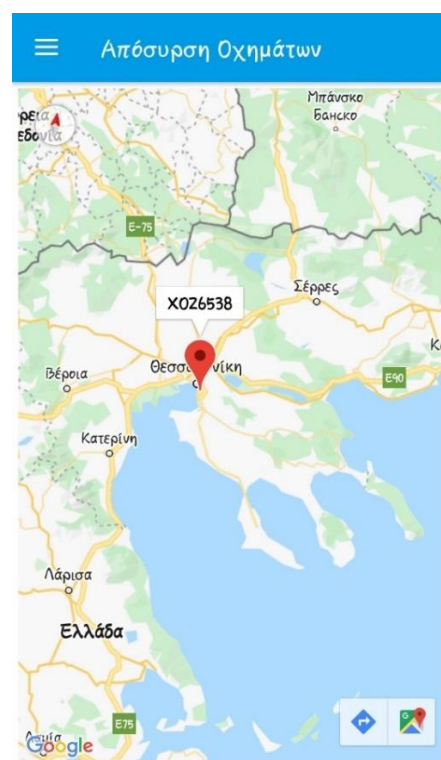
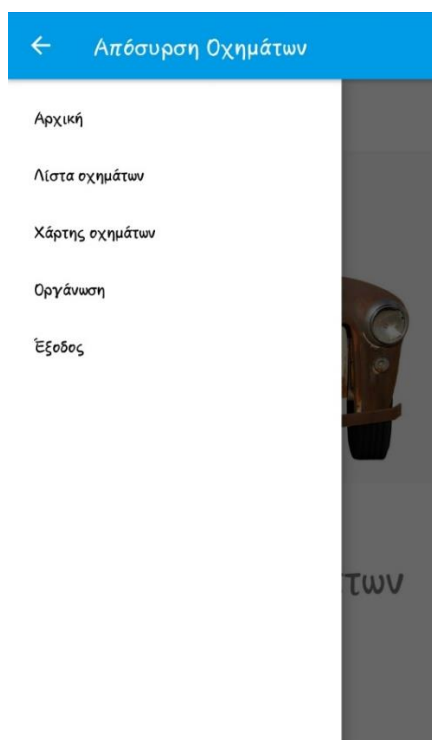
Ρόλος: Υπάλληλος του Δήμου



Τέλος, ο Υπάλληλος μπορεί να ανανεώσει το ολικό status του συγκεκριμένου οχήματος πατώντας «Κλείσιμο», καθώς όλη η διαδικασία απόσυρσής του έχει ολοκληρωθεί. Όπως φαίνεται στην εικόνα, εμφανίζεται το μήνυμα «Η διαδικασία απόσυρσης ολοκληρώθηκε».

## Προβολή Χάρτη

Ο «Υπάλληλος του Δήμου» μπορεί να δει όλες τις θέσεις των οχημάτων προς απόσυρση και τις πινακίδες τους (εφόσον έχουν) από το Drawer Menu και την επιλογή «Χάρτης οχημάτων».



## Υπεύθυνος Ομάδας Απόσυρσης

### Αρχική Οθόνη



## Απόσυρση Οχημάτων

Ρόλος: Υπεύθυνος Ομάδας Απόσυρσης



Αρχική

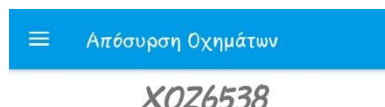
Λίστα οχημάτων προς απόσυρση

Έξοδος

Από το Drawer Menu ο «Υπεύθυνος Ομάδας Απόσυρσης» επιλέγει «Λίστα οχημάτων προς απόσυρση».



Χ026538 : Ανάθεση Απόσυρσης



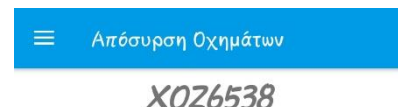
X026538



Παρατημένο, σκουριασμένο, μίτλε λεωφορείο  
40.6077644 , 22.9693772

ΑΠΟΔΟΧΗ ΑΝΑΘΕΣΗΣ ΑΠΟΣΥΡΣΗΣ

ΑΠΟΡΡΙΨΗ ΑΝΑΘΕΣΗΣ ΑΠΟΣΥΡΣΗΣ



X026538



Παρατημένο, σκουριασμένο, μίτλε λεωφορείο  
40.6077644 , 22.9693772

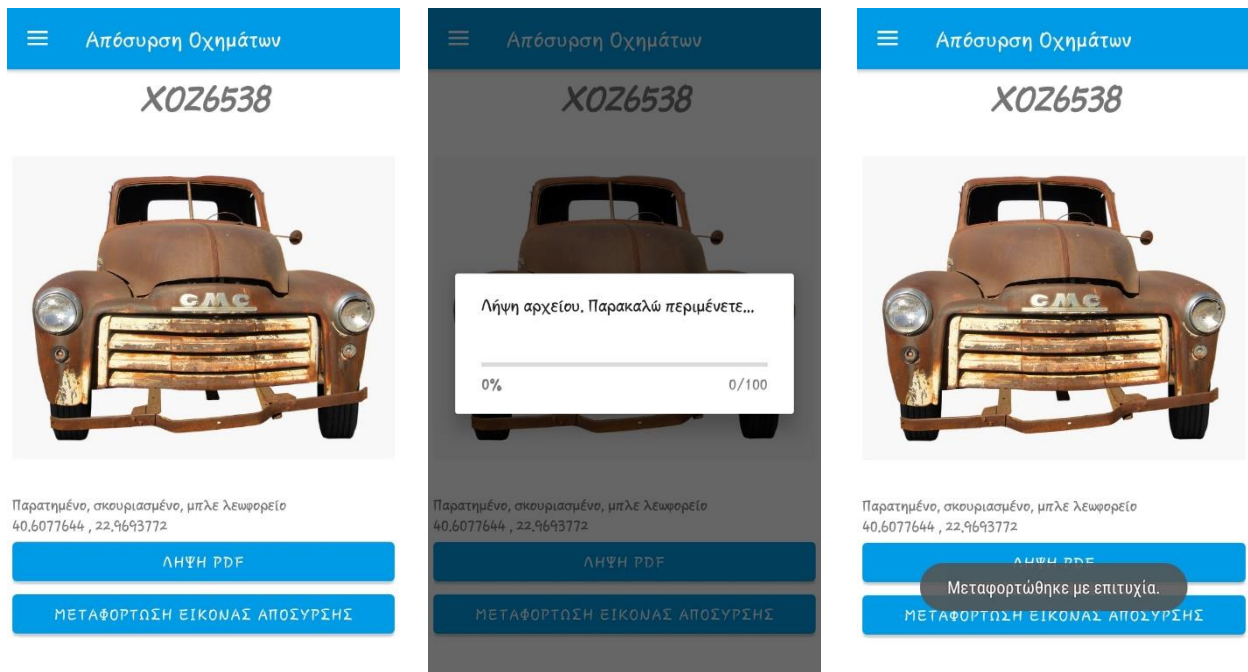
ΑΠΟΔΟΧΗ ΑΝΑΘΕΣΗΣ ΑΠΟΣΥΡΣΗΣ

Η ανάθεση απόσυρσης έγινε αποδεκτή.

ΑΠΟΡΡΙΨΗ ΑΝΑΘΕΣΗΣ ΑΠΟΣΥΡΣΗΣ

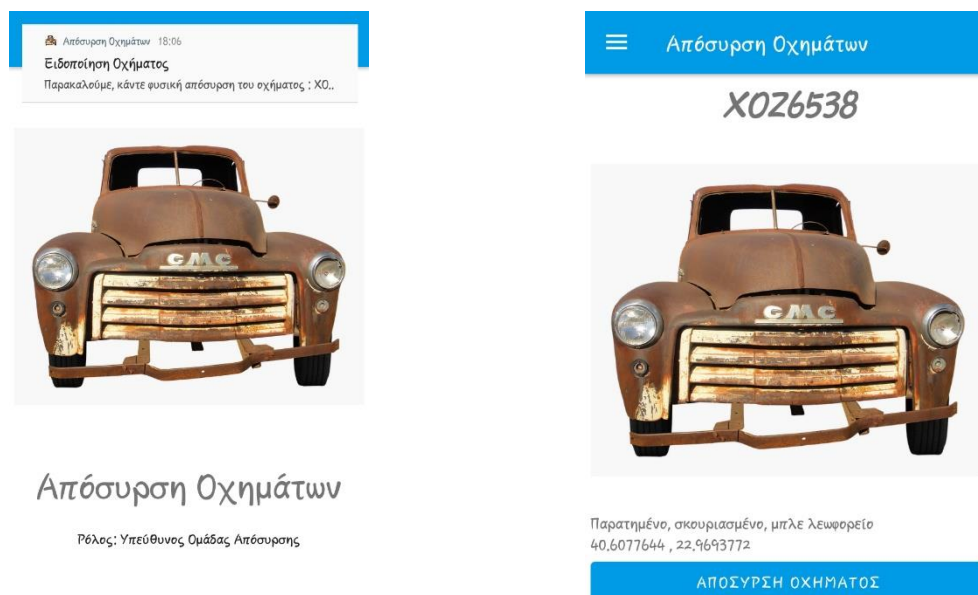
Ο «Υπεύθυνος Ομάδας Απόσυρσης» μπορεί να αποδεχθεί ή να απορρίψει την απόσυρση του οχήματος. Πατώντας το κουμπί «Αποδοχή Ανάθεσης Απόσυρσης» εμφανίζεται στην οθόνη το μήνυμα «Η ανάθεση απόσυρσης έγινε αποδεκτή», όπως φαίνεται παραπάνω.

Ο «Υπεύθυνος Ομάδας Απόσυρσης» μπορεί να επιστρέψει και να κάνει κλικ στη «Λίστα οχημάτων προς απόσυρση» για να δει την αλλαγμένη κατάσταση του οχήματος, όπως και να κατεβάσει το PDF του αντίστοιχου ειδοποιητηρίου.



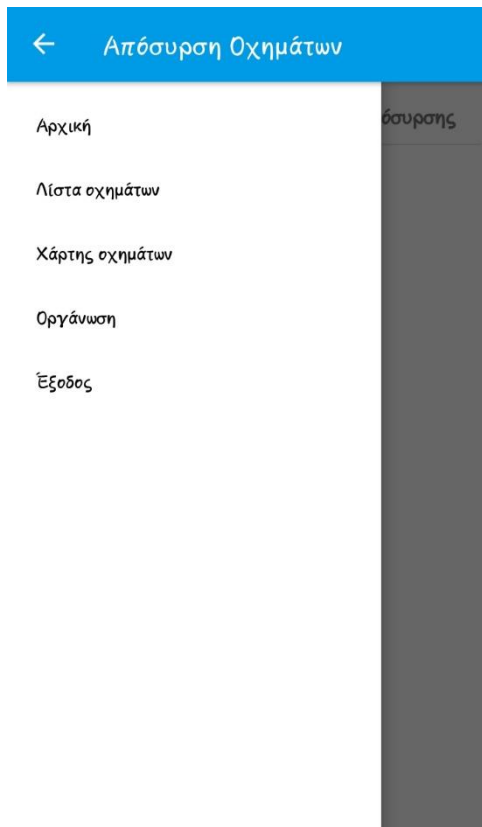
Ο «Υπεύθυνος Ομάδας Απόσυρσης» κατεβάζει το PDF και το τοποθετεί πάνω στο όχημα. Στη συνέχεια, ανεβάζει την αρχική φωτογραφία του οχήματος που πρόκειται να αποσυρθεί, τη μεταφορτώνει κάνοντας κλικ στο κουμπί «Μεταφόρτωση Εικόνας Απόσυρσης» και τότε εμφανίζεται το μήνυμα «Μεταφορτώθηκε με επιτυχία».

Μόλις φορτωθεί η εικόνα απόσυρσης, ο Υπάλληλος πρέπει να την επαληθεύσει και να προχωρήσει στην ημερήσια απόσυρση. Μόλις πραγματοποιηθεί αυτό το στάδιο, ο «Υπεύθυνος Ομάδας Απόσυρσης» θα ειδοποιηθεί για να αποσύρει το όχημα νόμιμα.



Σε αυτό το στάδιο, ο «Υπεύθυνος Ομάδας Απόσυρσης» κάνει τη φυσική απόσυρση του οχήματος πατώντας το αντίστοιχο κουμπί «Απόσυρση Οχήματος».

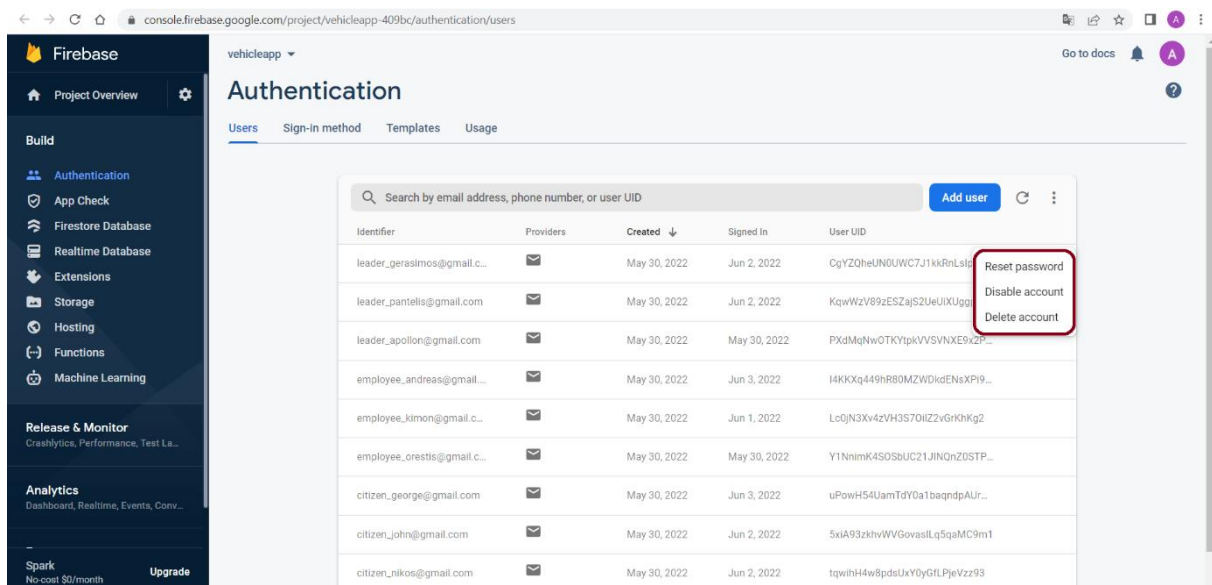
## Αποσύνδεση



Καθώς κάνετε κλικ στην τελευταία επιλογή του μενού, αποσυνδέεστε από την εφαρμογή.

## Super Admin

Ο Super Admin συνδέεται με τα διαπιστευτήριά του στην πλατφόρμα της Firebase για να δει τη λίστα όλων των εγγεγραμμένων χρηστών στην εφαρμογή. Σε αυτή την οθόνη, έχει τη δυνατότητα να επαναφέρει τον κωδικό πρόσβασης ενός χρήστη επιλέγοντας «**Reset password**», να απενεργοποιήσει το λογαριασμό ενός χρήστη επιλέγοντας «**Disable account**» και να διαγράψει το λογαριασμό του χρήστη επιλέγοντας «**Delete account**».



Για να επαληθεύσει έναν χρήστη ο οποίος έχει τον ρόλο «Υπάλληλος του Δήμου» ή «Αρχηγός Ομάδας Απόσυρσης» κάνει κλικ στο «**Realtime Database**» από το μενού και στη συνέχεια στον πίνακα «role». Επιλέγοντας το id ενός χρήστη, επεξεργάζεται το πεδίο **verified** και από **false** το αλλάζει σε **true**, όπως φαίνεται παρακάτω. Τώρα, ο συγκεκριμένος χρήστης μπορεί να έχει πρόσβαση στην εφαρμογή. Τέλος, μπορεί να αποκλείσει την πρόσβαση σε κάποιον χρήστη με τον ίδιο τρόπο.

### Ρόλος : Υπάλληλος του Δήμου

The image shows two screenshots of the Realtime Database console for the URL `https://vehicleapp-409bc-default-rtdb.firebaseio.com/`. Both screenshots show a tree view with a 'role' node expanded. The first screenshot shows a list of user IDs, with the selected user ID `-N3d22SR9HNKWizw092S` having the following properties: `email: "employee_periklis@gmail.com"`, `role: "Employee"`, and `verified: false`. The second screenshot shows the same user ID with the `verified` property set to `true`.

### Ρόλος : Αρχηγός Ομάδας Απόσυρσης

The image shows two screenshots of the Realtime Database console for the URL `https://vehicleapp-409bc-default-rtdb.firebaseio.com/`. Both screenshots show a tree view with a 'role' node expanded. The first screenshot shows a list of user IDs, with the selected user ID `-N3d2wqqHoEspvh-twEa` having the following properties: `email: "leader_alexandros@gmail.com"`, `role: "Leader"`, and `verified: false`. The second screenshot shows the same user ID with the `verified` property set to `true`.

Ο Super Admin, καθ' όλη τη διάρκεια εκτέλεσης της εφαρμογής στη συσκευή ενός χρήστη, έχει τη δυνατότητα να παρακολουθεί σε πραγματικό χρόνο όλες τις ειδοποιήσεις (notifications), εισαγωγές (imports) και αλλαγές (changes) που γίνονται στη Βάση Δεδομένων (Realtime Database) της Firebase. Παρακάτω φαίνονται ορισμένα "live" στιγμιότυπα της ΒΔ.

```
https://vehicleapp-409bc-default-rtdb.firebaseio.com/
└─ notifications
  └─ -N3j5G0sFZ9UfLdLm23Q
    └─ received: true
    └─ sender: "leader_apollon@gmail.com"
    └─ vin: "XOZ6538"
    └─ vntype: 1
  └─ -N3jAPtR0ICKd0LpCbU1
    └─ received: true
    └─ sender: "leader_apollon@gmail.com"
    └─ vin: "XOZ6538"
    └─ vntype: 2
  └─ -N3jEa3qdfItRadWWAjW
    └─ received: true
    └─ sender: "leader_apollon@gmail.com"
    └─ vin: "XOZ6538"
    └─ vntype: 3
  └─ -N3jGC2QVn3zNCH4hy1U
    └─ received: true
    └─ sender: "leader_apollon@gmail.com"
    └─ vin: "XOZ6538"
    └─ vntype: 4
```

```
https://vehicleapp-409bc-default-rtdb.firebaseio.com/
└─ notifications
└─ role
└─ vehicles
  └─ -N3izSRvDgTd8ljsxHJH
    └─ assignee: "leader_apollon@gmail.com"
    └─ carLicense: ""
    └─ category: "Λεωφορείο"
    └─ characteristics: "Παρατημένο, σκουριασμένο, μπλε λεωφορείο "
    └─ latitude: "40.6077644"
    └─ longitude: "22.9693772"
    └─ ownerIdentity: ""
    └─ pdf: "https://firebasestorage.googleapis.com/v0/b/vehicleapp-409bc.appspot.com/o/1654353109153.pdf?alt=media&token=c76a0ea0-f759-4d72-b7f4-..."
    └─ picture: "/storage/0403-0201/ΕΦΑΡΜΟΓΗ/bus.jpg"
    └─ regPlate: ""
    └─ status: 8
    └─ uid: "citizen_nikos@gmail.com"
    └─ vin: ""
    └─ vnumber: "XOZ6538"
    └─ withdrawalImage: "https://firebasestorage.googleapis.com/v0/b/vehicleapp-409bc.appspot.com/o/1654353928348.jpg?alt=media&token=78ffa020-39..."
```