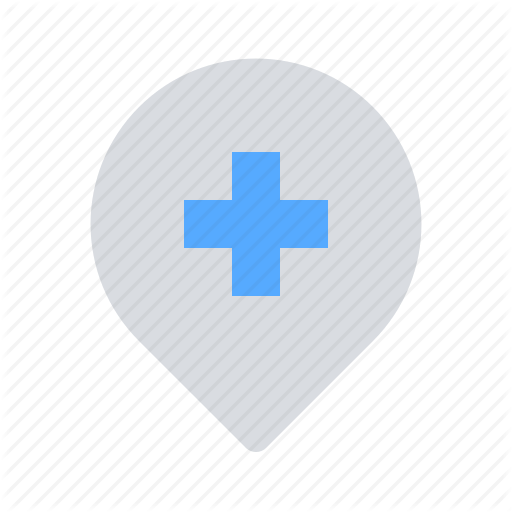
|  |  |
| --- | --- |
|  | **Πανεπιστήμιο Αιγαίου**  **Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων** |



Σχεδιασμός και Ανάπτυξη Εφαρμογών Κινητού Υπολογισμού

ΕΡΓΑΣΙΑ

«Crowdsourcing σύστημα αναφορών πληροφοριών τοποθεσίας Απινιδωτή με χρήση Firebase»

Τριανταφύλλης Πέτρος Ευάγγελος

Κωνσταντίνος Κοντογεωργόπουλος

Φάνης Μπινιάκου

ΥΠΕΥΘΥΝΟΣ ΚΑΘΗΓΗΤΗΣ: Χρήστος Γκουμόπουλος

ΣΑΜΟΣ 2019

Crowdsourcing σύστημα αναφορών πληροφοριών τοποθεσίας Απινιδωτή με χρήση Firebase

Τριανταφύλλης Πέτρος Ευάγγελος

Κωνσταντίνος Κοντογεωργόπουλος

Φάνης Μπινιάκου

**Περίληψη:**Η παρούσα αναφορά συνοδεύει την υλοποίηση του ομαδικού project το μάθημα ”Σχεδιασμός και ανάπτυξη και Ανάπτυξη Εφαρμογών Κινητού Υπολογισμού” και πραγματέυεται **μόνο τη διαδικασία υλοποίησης του συστήματος.**

**Λέξεις-κλειδιά:** android , firebase , croudsourcing ,health , defibrillator , js , real-time , nosql , android-studio , is .

**Περιεχόμενα**

[1. Client Development (native android) 5](#__RefHeading___Toc796_285580365)

[Αρχεία διαχείρησης Εφαρμογής 5](#__RefHeading___Toc772_285580365)

[Αρχεία .java 7](#__RefHeading___Toc774_285580365)

[Οθόνες Εφαρμογής 18](#__RefHeading___Toc794_285580365)

[Αναφορές 20](#__RefHeading___Toc717_285580365)

[Παράρτημα Α: τίτλος 20](#__RefHeading___Toc719_285580365)

[Παράρτημα Β: τίτλος 20](#__RefHeading___Toc721_285580365)

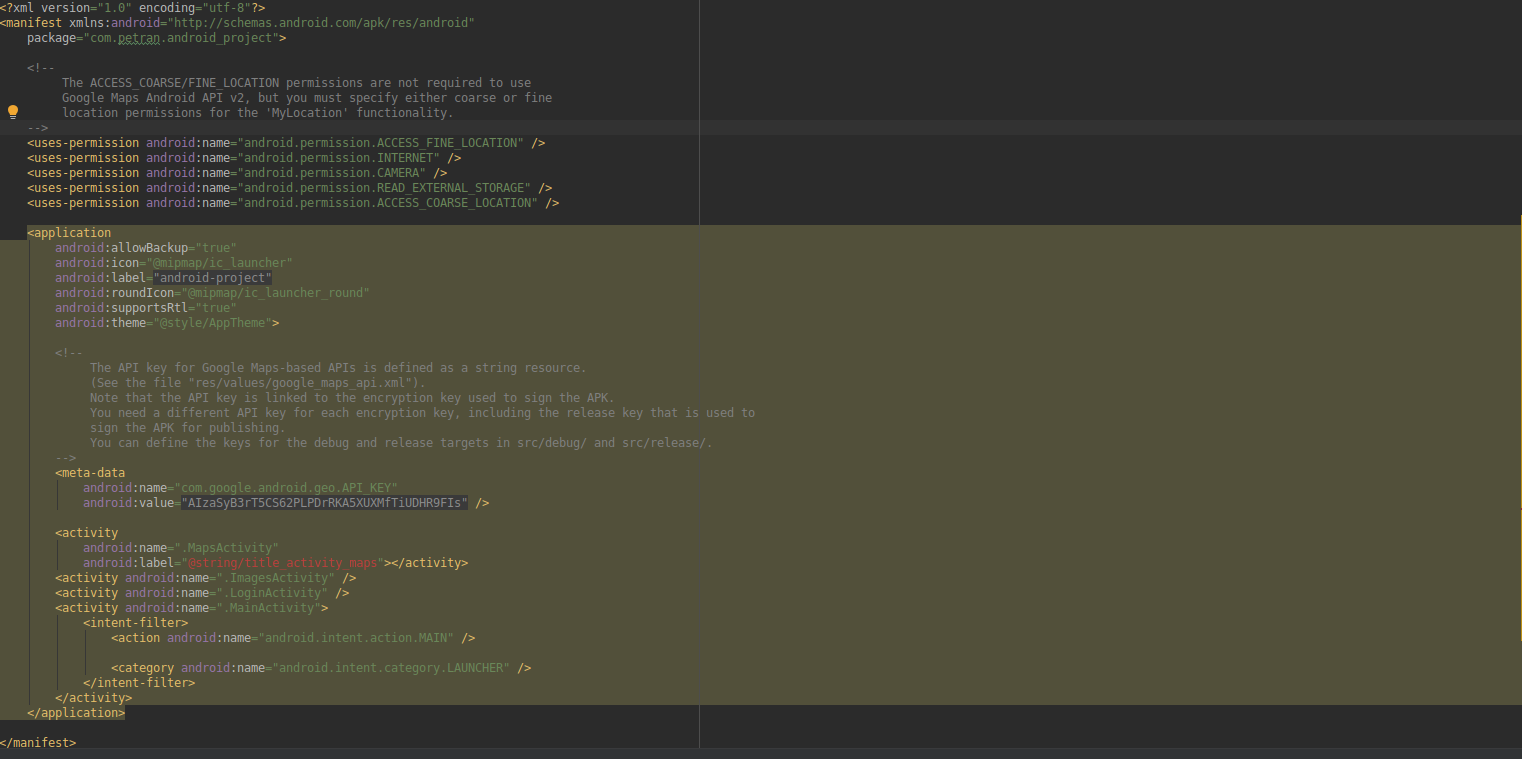
# 

Η παρακάτω υλοποίηση όπως και η παρούσα αναφορά βρίσκόνται ακόμα σε στάδιο ανάπτυξης , δεν αποτελούν την release έκδοση του συστήματος.

# Client Development (native android)

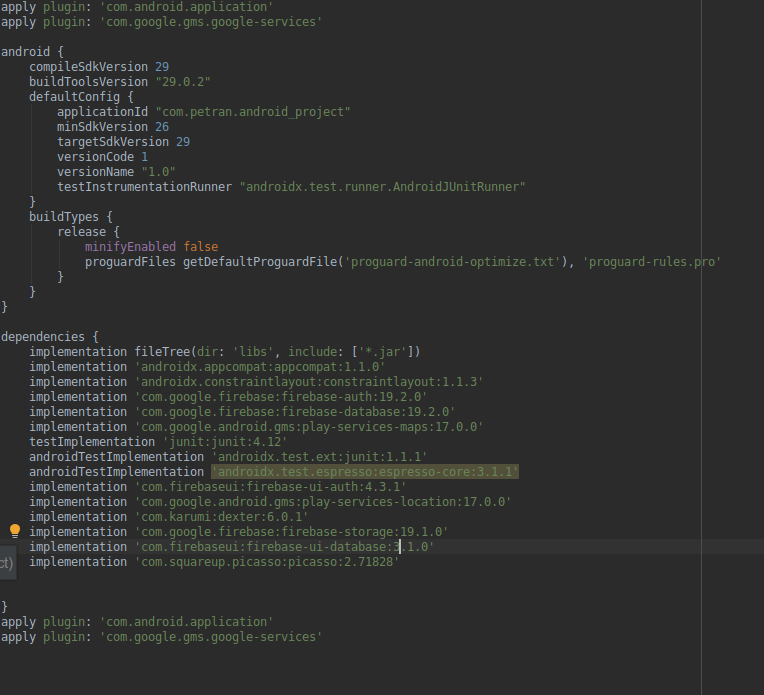
## Αρχεία διαχείρησης Εφαρμογής

* Στό **AndroidManifest.xml** δήλωσάμε στο λειτουργίκό μας τους πόρους που θα χρειαστούμε απο αυτό.

****

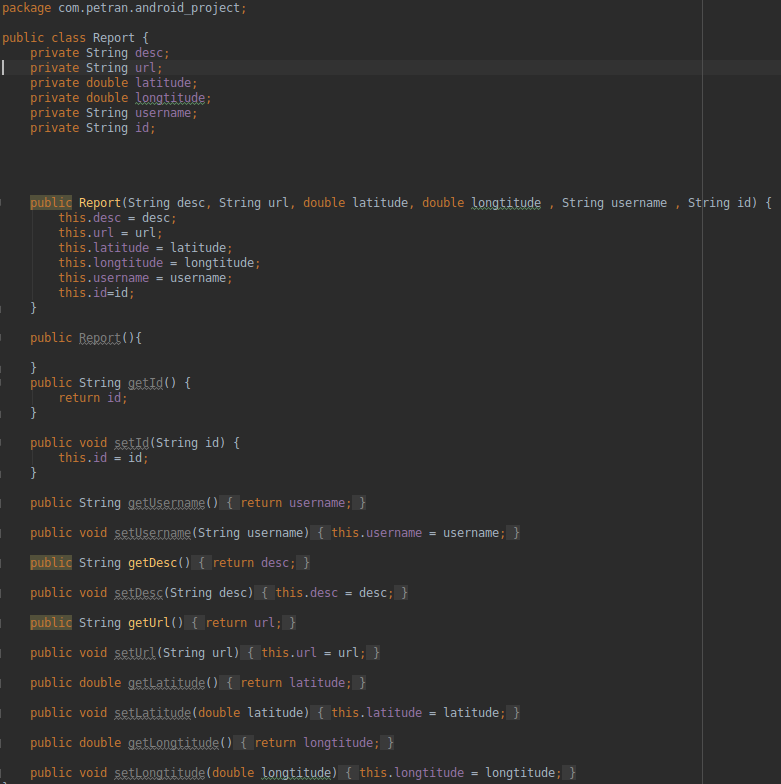
**Σχήμα 1:**AndroidManifest.xml

* Οι βιβλιοθήκες τρίτών που χρειαστήκαμε δηλώθηκαν στο **build.gradle** σε επίπεδο **app.**

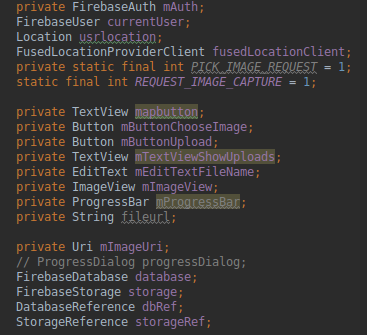


**Σχήμα 2:**/app/builde.gradle

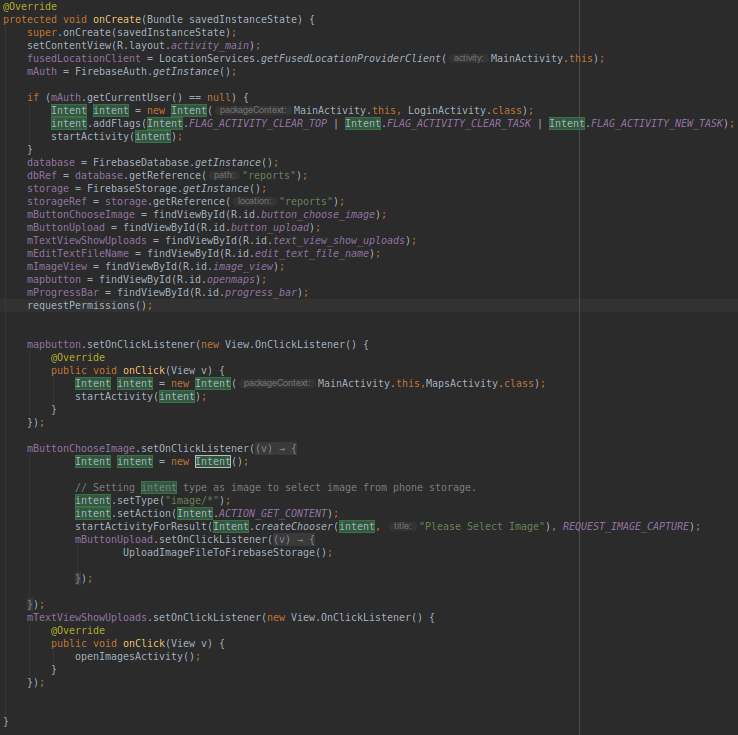
## Αρχεία .java

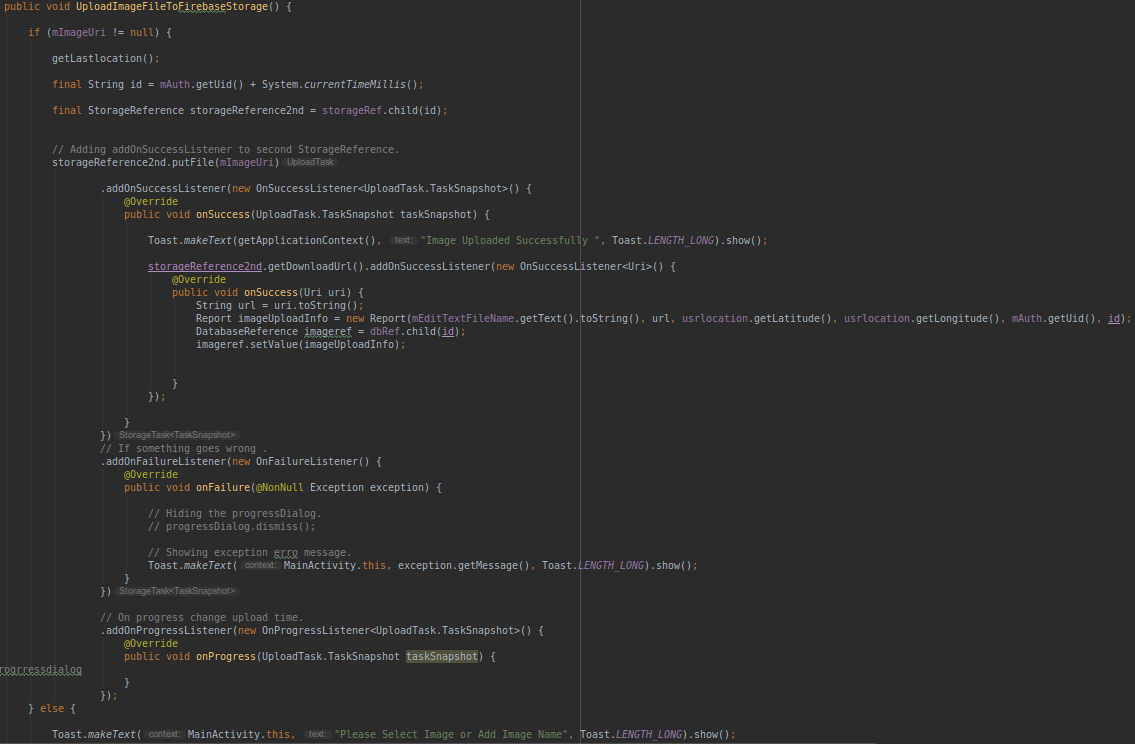
* Το μοντέλο αναφόρας απινιδωτή σε **java** class με χρήση default constructor , get-set μεθόδους.

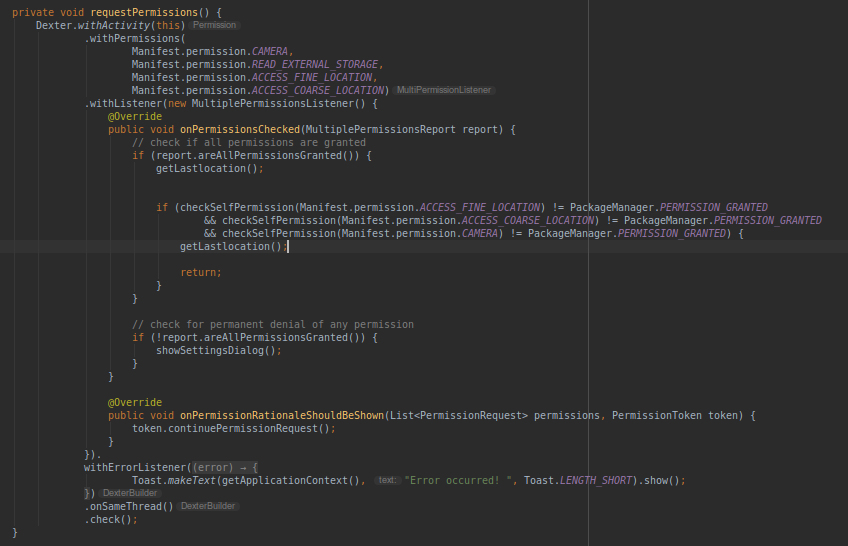
**Σχήμα 3:**Report.java

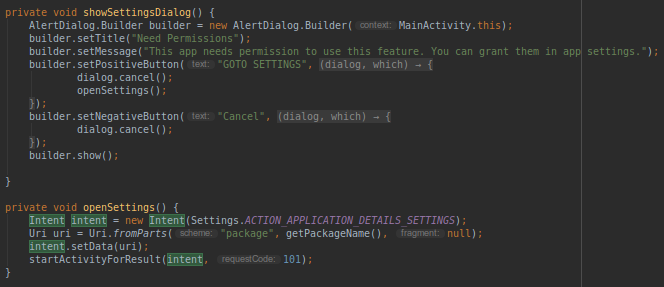
* Η MainActivity απαρτίζει την κύρια οθόνη του client καθώς και ελέγχει την ροή του.

**Σχήμα 4:** Δηλώσεις Μεταβλητών στην MainActivity.

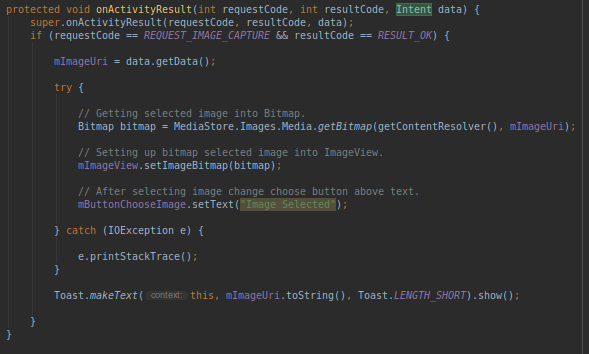
* Στήν onCreate method αρχικοποιούμε αντικείμενα κλάσεων που θα χρειαστούμε στην MainActivity καθώς και το components του GUI. Επίσης ελέγχούμε την κατάσταση του login μεταφέροντας το χρήστη στο LoginActivity ετσι ώστε να αυθεντικοποιηθεί.  
  **Σχήμα 5:** onCreate της MainActivity
* H μέθοδος UploadFileToFirebaseStorage στέλνει το αρχείο που επιλέξαμε σε έναν κατάλογο στο FirebaseStorage . Η επιτυχή ολοκλήρωση του Task οδηγεί σε απόπειρα ανάκτησης του downloadUrl της εικόνας έτσι ώστε Τελικά να γραφτεί σαν Attribute του μοντέλου Report.java στη βάση δεδομένων μας.

**Σχήμα 6:** UploadFileToFirebaseStorage της MainActivity

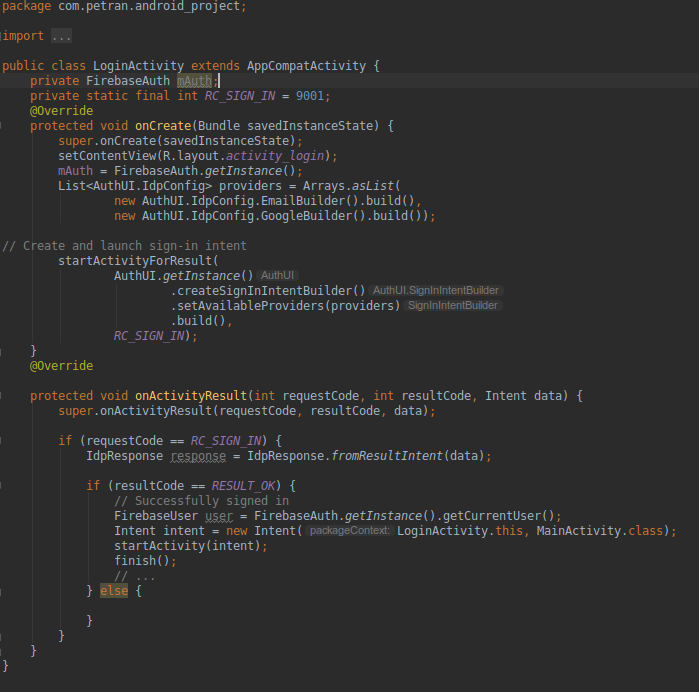
* Η μέθοδος requestPermissions κάνει χρήση του Dexter για να πάρουμε τα δικαιώματα για την χρήση των πόρων που έχουμε δηλώσει στο AndroidManifest.xml . H επιτυχή απόκτηση των δικαιωμάτων οδηγεί στην ανανέωση της μεταβλητής της τοποθεσίας του gps.  
    
  **Σχήμα 7:** requestPermissions της MainActivity
* Η μόνιμη άρνηση των δικαιωμάτων στον Default διάλογο του Android οδήγει σε στο χτίσιμο προσαρμοσμένου παραθύρου διαλόγου και τελικά στις ρυθμίσεις του κινητού έτσι ώστε να επιτραπεί χειροκίνητα η πρόσβαση του κινητού στους συγκεκριμένους πόρους όπως φαίνεται στο σχήμα 8.

**Σχήμα 8:** showSettingDialog() - openSettings()

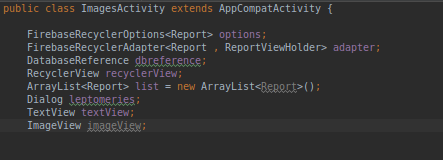
* H onActivityResult εκμεταλεύεται την εικόνα που μας επιστρέφει το intent της Δραστηριότητας στην οποία επιλέγουμε εικόνα προς upload.

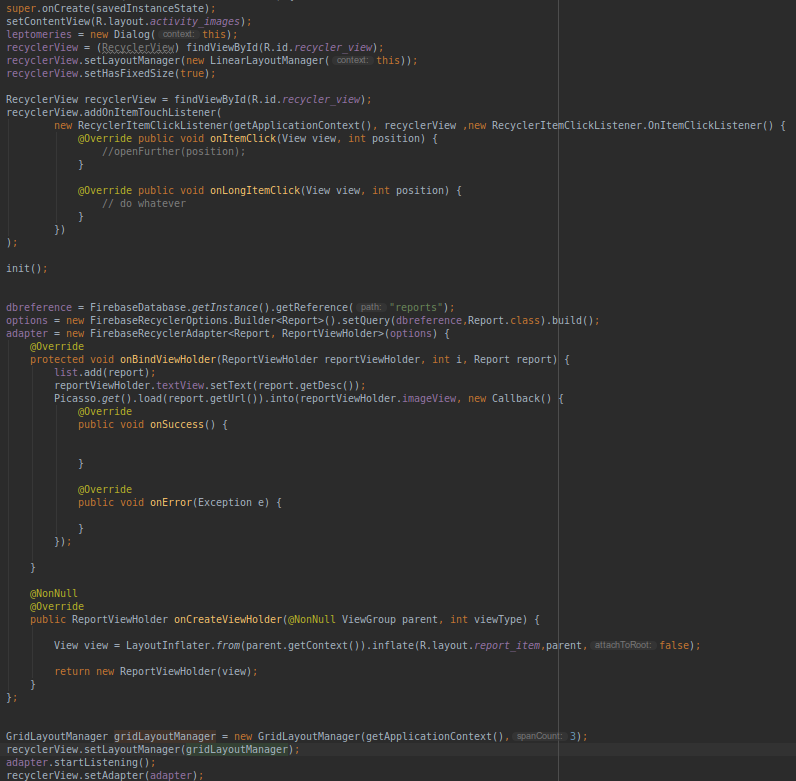
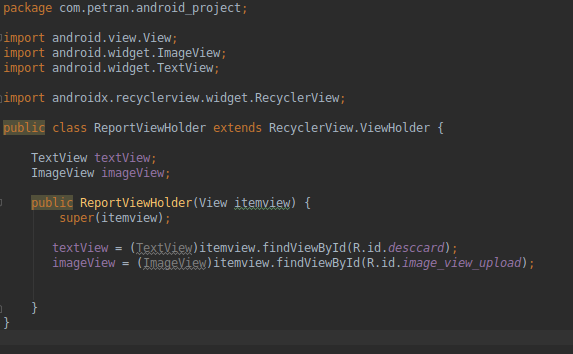
**Σχήμα 9 :** OnActivityResult()

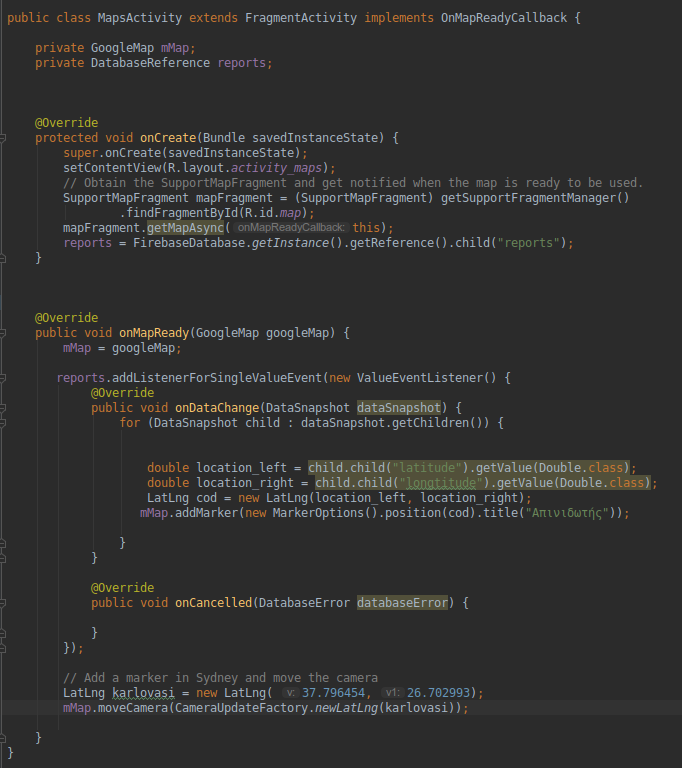
* Στην κλάση LoginActivity Αρχικοποιούμε το prebuild Login GUI του Firebase , για Email και Google Login

**Σχήμα 10**: LoginActivity.java

* Η κλάση ImagesActivity ανταποκρίνεται στην οθόνη οπου προβάλονται οι αναφορές των απινιδωτών καθώς και οι λεπτομέριες τους.

**Σχήμα 11**: Δηλώσεις μεταβλητών στην ImagesΑctivity.java

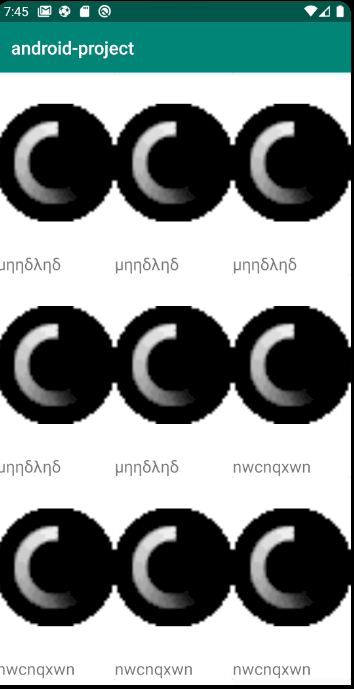
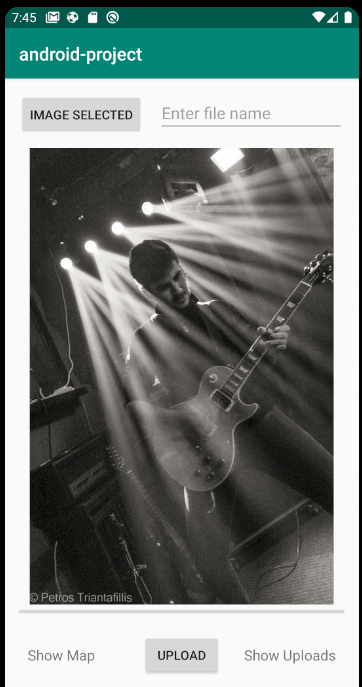
* Στην onCreate της ImagesActivity Αρχικοποιούμε Το RecyclerView στο Layout μας χρησιμοποιοώντας ώς Adapter τον FirebaseRecyclerAdapter (που υπάρχει στο SDK του Firebase) για να Χτίσουμε Αντικείμενα τύπου Rreport class απο τη JSON μορφή της βάσης δεδομένων μας. Τελικά τα αναπαριστούμε στον reportviewholder.
* Η κλάση ReportViewHolder μαζί με το αντιστοιχο layout αρχείο αναπαριστούν τίς κάρτες (Cardview widget) που θα εμφανίζει το RecyclerView **Σχήμα 13**: ReportViewHolder.java



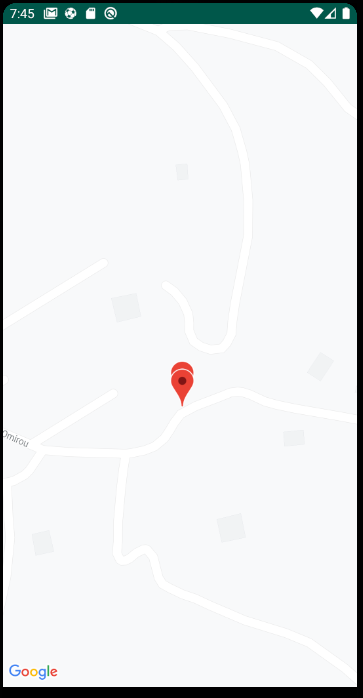
Η κλάση MapsActivity ανοίγει με το πάτημα ενος κουμπιού και αρχικοποιεί εναν χάρτη με pins τα οποία αντλούνται απο τη βάση δεδομένων με την κλήση ενος Datasnapshot της βάσης , η μέθοδος onDatachange εξασφαλίζει δυναμικη ανανέωση των pins.

**Σχήμα 14**: MapsActivity.java

## Οθόνες Εφαρμογής

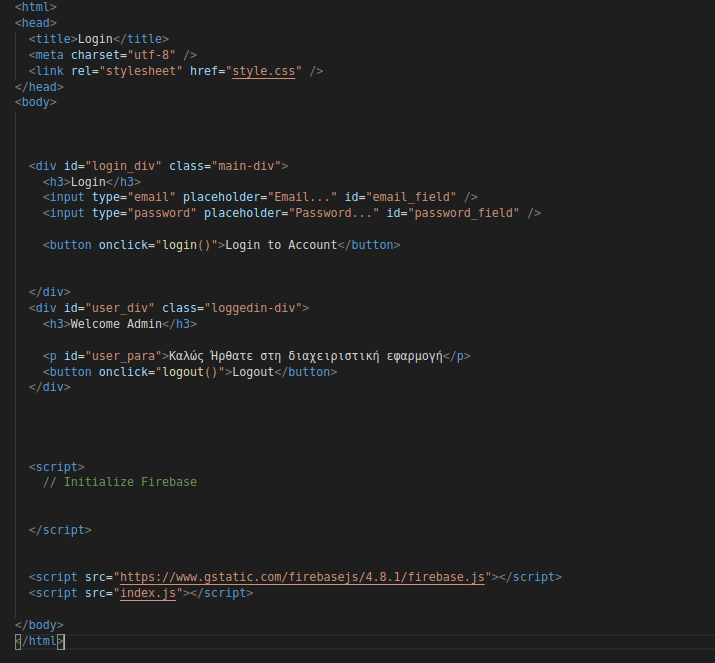


**Αριστερά :** Οθόνη Επιλογής φωτογραφίας **, Δεξιά :** Οθόνη προβολής clickable αναφορών.  
\*Το κλικ κρασαρε , δεν εχει υλοποιηθεί ακόμα.

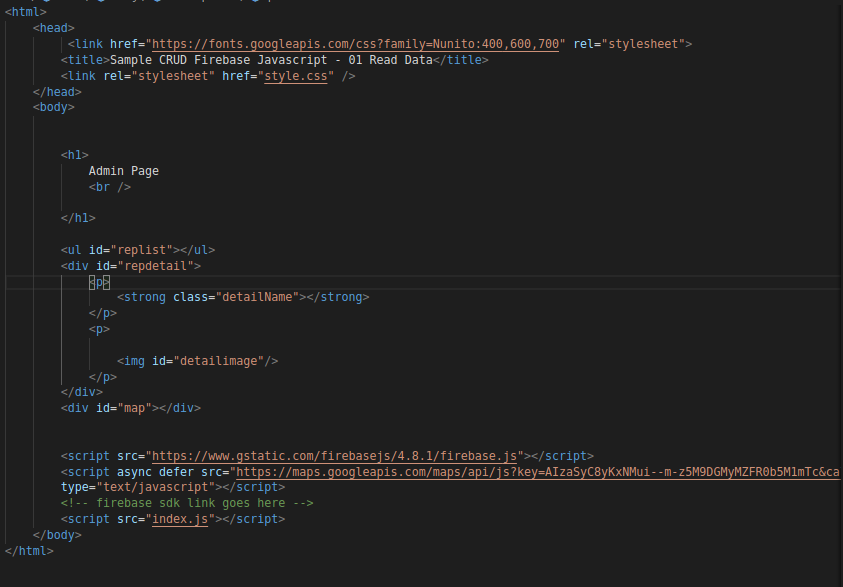


**Σχήμα 17:** Οθόνη MapsActivity

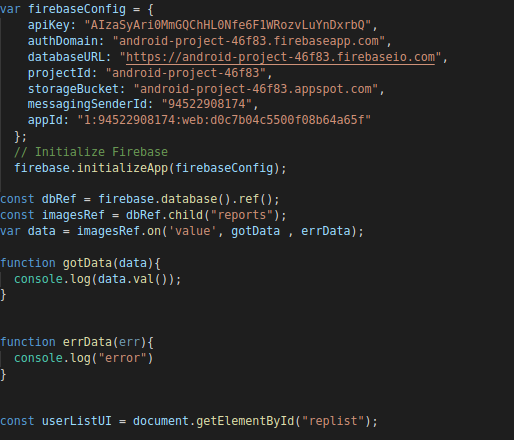
# 2.Admin UI

**Εικόνα 18**:Login page (index.html)

Στη Main Page δημιουργήσαμε μονο divs με id για να δημιουργούμε και να ανανεώνουμε το περιέχομενο δυναμικά με nodejs client.



**Εικόνα 19**: Kύρια σελίδα main.html

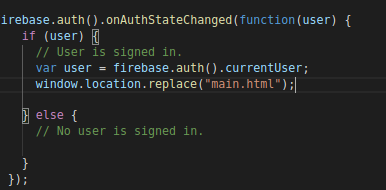


**Εικόνα 20 :**  Αρχικοποίηση Firebase και test functions στο αρχείο index.js

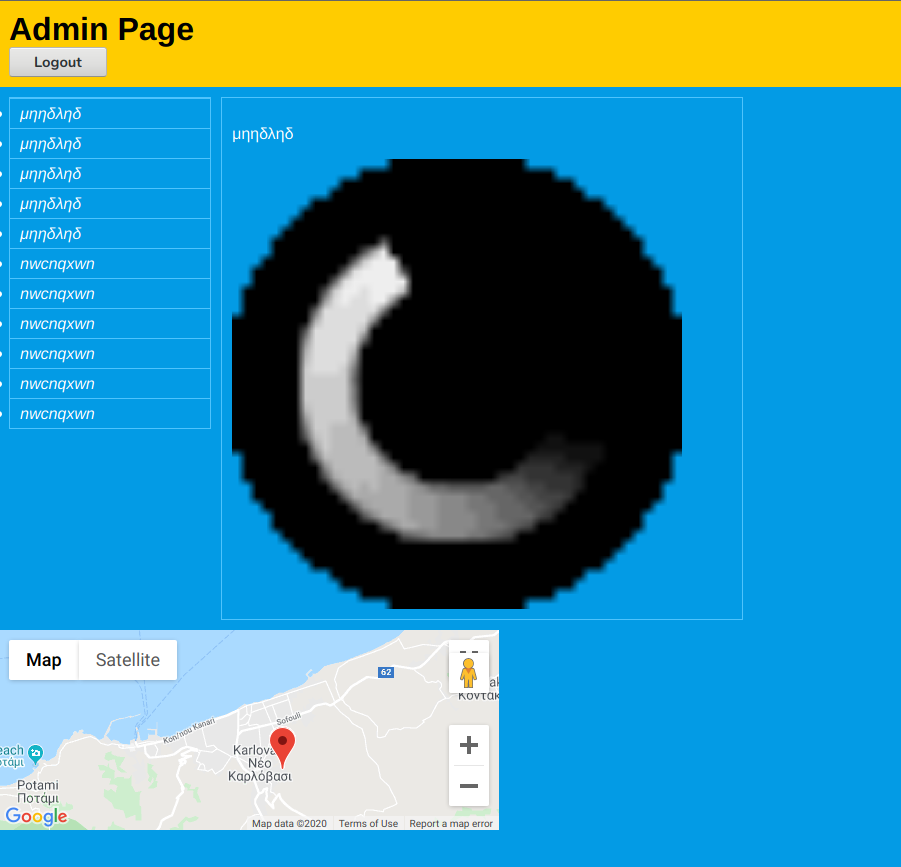
**Εικόνα 21 : l**ogin – logout functions

**Εικόνα 22 :** Αρχικοποίηση map με pins που αντλούνται απο data snapshot για καθε child του parent reference.

**Εικόνα 23**: Μέθοδος Δυναμικής ανανέωσης της λίστας των απινιδωτών και μέθοδος εκμετάλευσης του κλίκ σε όποια απο αυτές(Εμφάνιση λεπτομεριών).



**Εικονα 24**: Μέθοδος για redirect στο mainpage αφου γίνει το login.

**Εικονα 25**: Εμφάνιση Admin Page

# Παράρτημα Α: TODO

-Beautify Client (layout organization , labels , pictures , fonts ,themes)

-Beautify Admin UI( Better CSS)

-Εμφάνιση λεπτομεριών απινιδωτή στον client

-Beautify-Comment Code

-Ολοκλήρωση Αναφοράς (προσθήκη περιεχομένου , beautify , διόρθωση συνακτικών-γραμματικών λαθών.

-Επεξεργασία τοποθεσίας αναφοράς

# Παράρτημα Β: Δεν υλοποιήθηκαν

-Αναφορά προβλήματος απινιδωτή

-Διαχείρηση πρόσβασης στη βάση