



Πανεπιστήμιο Αιγαίου

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

Σχεδιασμός και Ανάπτυξη Εφαρμογών Κινητού Υπολογισμού

**Εργασία: Ανάπτυξη εφαρμογής για τη διαχείριση υπηρεσιών παροχής
πρώτων βοηθειών με τη χρήση απινιδωτών σε εξωτερικούς δημόσιους χώρους**

Αναφορά εφαρμογής DigiAED

Τσότρας Στέφανος 321/2013189

12/01/2019

1. Τεχνολογίες που χρησιμοποιήθηκαν

Η υλοποίηση της εφαρμογής smartphone έγινε με native κώδικα για Android (Java) με την βοήθεια του IDE Android Studio.

Το back-end υλοποιήθηκε με τη βοήθεια του framework Google Firebase, όπου δίνει δυνατότητα για τη εκμετάλλευση πολλαπλών services back-end. Στην εφαρμογή συγκεκριμένα χρησιμοποιήθηκαν τα παρακάτω services.

- Firebase Auth

Χρησιμοποιείται για την γρήγορη εγγραφή και αυθεντικοποίηση των χρηστών και των διαχειριστών της εφαρμογής.

- Firebase FireStore

Χρησιμοποιείται για την αποθήκευση και την ανάκτηση των πληροφοριών και της ΓεωΔιεύθυνσης των απινιδωτών, όπως και τη αποθήκευση των εκάστοτε αναφορών προβλημάτων όπου αναφέρονται.

- Firebase Storage

Χρησιμοποιείται για την αποθήκευση και την ανάκτηση των εικόνων που συσχετίζονται με τους απινιδωτές.

Για την εγγραφή και είσοδο με Google Account, την δημιουργία χαρτών και την αναζήτηση τοποθεσίας χρησιμοποιήθηκε το Google API.

Στην εφαρμογή λόγω της εισαγωγής πολλαπλών δεσμεύσεων, χρειάστηκε η διαχείριση τους μέσω του multidex.

2. Απαιτήσεις εφαρμογής

Η εφαρμογή υποστηρίζεται σε τουλάχιστον SDK Version 21 και είναι υλοποιημένο για συσκευές με SDK Version 29. Οι συσκευές είναι απαραίτητο να υποστηρίζουν google-play services και επίσης απαραίτητη είναι η σύνδεση της συσκευής στο διαδίκτυο και η πρόσβαση στη τοποθεσία της.

Πρακτικά, οι απαιτήσεις αυτές ορίζονται στα παρακάτω αρχεία, όπου δημιουργούνται αυτόματα από android studio όταν δημιουργείται το project.

- Build.gradle (Project)

```
dependencies {  
    classpath 'com.android.tools.build:gradle:3.5.3'  
    classpath 'com.google.gms:google-services:4.3.3'
```

Σχήμα 1: project gradle dependencies

- Build.gradle(Module)

```

apply plugin: 'com.android.application'

android {
    compileSdkVersion 29
    buildToolsVersion "29.0.2"
    defaultConfig {
        applicationId "com.example.digiaed"
        minSdkVersion 21
        targetSdkVersion 29
        multiDexEnabled true
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled true
            shrinkResources true
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
}

```

Σχήμα 2: app gradle dependencies

```

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'androidx.appcompat:appcompat:1.1.0'
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'

    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'androidx.test.ext:junit:1.1.0'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.1.1'

    implementation 'com.android.support:multidex:1.0.3'

    //Firebase
    implementation 'com.google.firebase:firebase-auth:19.2.0'
    implementation 'com.google.firebase:firebase-database:19.2.0'
    implementation 'com.google.firebase:firebase-storage:19.1.0'
    implementation 'com.google.firebase:firebase-firestore:21.3.1'

    //Google
    implementation 'com.google.android.gms:play-services-auth:17.0.0'
    implementation 'com.google.android.gms:play-services-location:17.0.0'
    implementation 'com.google.android.gms:play-services-maps:17.0.0'
}

apply plugin: 'com.google.gms.google-services'

```

- Android Manifest

```

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

```

Σχήμα 3: απαραίτητα permissions εφαρμογής

Στο μανιφέστο αυτό ορίζονται και όλες η σελίδες όπου υλοποιούν τις λειτουργίες της εφαρμογής.

3. Λειτουργίες της εφαρμογής

Η εφαρμογή υλοποιεί τις παρακάτω λειτουργίες.

- Εμφάνιση της τοποθεσίας απινιδωτών σε χάρτη

Η κλάση που υλοποιεί τον κύριο χάρτη της εφαρμογής είναι η AEDMapActivity.java, στην οποία γίνονται αρκετές διεργασίες πριν εμφανιστεί το τελικό αποτέλεσμα.

Αρχικά, ζητείται πρόσβαση στη τοποθεσία του χρήστη, όπου στη συνέχεια εφόσον δωθεί βρίσκεται η τοποθεσία του χρήστη

```

private void getDeviceLocation(){

    mFusedLocationProviderClient = LocationServices.getFusedLocationProviderClient(this);

    try{
        if(mLocationPermissionGranted){
            Task location = mFusedLocationProviderClient.getLastLocation();
            location.addOnCompleteListener(new OnCompleteListener() {
                @Override
                public void onComplete(@NonNull Task task) {
                    if(task.isSuccessful()){
                        currentLocation = (Location) task.getResult();

                        if(currentLocation != null){
                            lati = currentLocation.getLatitude();
                            loni = currentLocation.getLongitude();
                            Log.d(TAG,"komple "+lati+" "+loni);
                            moveCamera(new LatLng(lati,loni),DEFAULT_ZOOM);
                        }
                        else{
                            getDeviceLocation();
                            Log.d(TAG,"godaaaaammit!!");
                        }
                    }
                }
            });
        }
    }catch(SecurityException e){
        Toast.makeText(AEDMapActivity.this,"unable to get current location",Toast.LENGTH_SHORT).show();
    }
}

```

Σχήμα 4: Εύρεση τοποθεσίας χρήστη.

γίνεται δημιουργία του χάρτη και μετακίνηση της κάμερας στη τοποθεσία του χρήστη όπου πήραμε προηγουμένως.

```
@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
    mLocationPermissionGranted = false;

    switch(requestCode){
        case LOCATION_PERMISSION_REQUEST_CODE:{
            if(grantResults.length > 0){
                for(int i=0;i<grantResults.length;i++){
                    if(grantResults[i] != PackageManager.PERMISSION_GRANTED){
                        mLocationPermissionGranted = false;
                        return;
                    }
                }
                mLocationPermissionGranted = true;
                //intialize map
                initMap();
            }
        }
    }
}
```

Σχήμα 5: Δημιουργία του χάρτη με επιτυχής πρόσβαση στη τοποθεσία του χρήστη.

Αφού ο χάρτης είναι έτοιμος γίνεται ανάλογο αίτημα στη βάση για την ανάκτηση των καταχωρημένων απινιδωτών.

Το αίτημα γίνεται μέσω ενός αντικειμένου FirestoreInstance db όπου αποτελεί reference στη βάση όπου είναι αποθηκευμένες οι πληροφορίες των απινιδωτών.

```
private void getDataFromDatabase(){

    db.collection("AEDMap")
        .get()
        .addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {
            @Override
            public void onComplete(@NonNull Task<QuerySnapshot> task) {
                if (task.isSuccessful()) {
                    for (QueryDocumentSnapshot document : task.getResult()) {
                        Log.d(TAG, document.getId() + " => " + document.getData());

                        //Get data and save on list

                        Map<String, Object> markerEntry = document.getData();
                        markerEntry.put("Id", document.getId());
                        Log.d(TAG, "List object : "+markerEntry);
                        markerMap.add(markerEntry);
                    }

                    setMarkersOnMap();
                } else {
                    Log.w(TAG, "Error getting documents.", task.getException());
                }
            }
        });
}
```

Σχήμα 6: Query στη firestore βάση για την ανάκτηση απινιδωτών

Αφού οι πληροφορίες των απινιδωτών ανακτηθούν και σωθούν τοπικά σε μία λίστα από HashMaps καλείται η συνάρτηση όπου βάζει τους απινιδωτές στο χάρτη ως markers.

```
private void setMarkersOnMap(){
    if(!markerMap.isEmpty()){
        for(int i=0; i<markerMap.size(); i++){
            String desc = (String) markerMap.get(i).get("Description");
            String name = (String) markerMap.get(i).get("Name");
            String imgUrl = (String) markerMap.get(i).get("ImageUrl");
            GeoPoint geoPoint = (GeoPoint) markerMap.get(i).get("Geolocation");
            String id = (String) markerMap.get(i).get("Id");

            Log.d(TAG,"name: "+name+" desc: "+desc+" geo: "+geoPoint+ " imgurl: "+imgUrl);

            String snippet = ""+desc+"";

            MarkerOptions options = new MarkerOptions().position(new LatLng(geoPoint.getLatitude(),geoPoint.getLongitude()))
                .title(name)
                .snippet(snippet)
                .icon(BitmapDescriptorFactory.fromResource(R.drawable.aedmap6));

            mMap.addMarker(options).setTag(id);
        }
    }
    else{
        Log.d(TAG,"markerMap is Empty");
    }
}
```

Σχήμα 7: Εισαγωγή markers στο χάρτη αναπαριστώντας τη τοποθεσία των απινιδωτών.

Στη συνέχεια, υλοποιούνται click listeners για το menu της εφαρμογής

```
ic_acc.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        startActivity(new Intent(AEDMapActivity.this, LoginActivity.class));

    }
});

ic_cpr.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Uri uri = Uri.parse("https://www.youtube.com/watch?v=fb29LCjX4-E&t=81s");
        startActivity(new Intent(Intent.ACTION_VIEW, uri));
    }
});

ic_reports.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        Intent i =new Intent(AEDMapActivity.this, ShowReportActivity.class);
        i.putExtra("Lat",lati);
        i.putExtra("Lon",loni);
        startActivity(i);

    }
});
```

Σχήμα 8: click listeners κουμπιών menu, για login, cpr, report.

Για το κουμπί add marker ο listener υλοποιείται αφού ετοιμαστεί ο χάρτης έτσι ώστε να εμφανιστεί μετακινήσιμος marker στη τοποθεσία του χρήστη

```
ic_plus.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        DEFAULT_ZOOM = 17f;
        getDeviceLocation();

        LatLng currlatlon = new LatLng(currentLocation.getLatitude()+0.0002,currentLocation.getLongitude()+0.0002);

        curraddmarker = mMap.addMarker(new MarkerOptions().position(currlatlon).draggable(true));
        currentMarkerLat=curraddmarker.getPosition().latitude;
        currentMarkerLon=curraddmarker.getPosition().longitude;
        hideUI();
        showConfirm();
    }
});
```

Σχήμα 9: Εισαγωγή draggable marker και εμφάνιση UI επιβεβαίωσης.

Αφού επιβεβαιωθεί η τοποθεσία του marker στέλνονται οι συντεταγμένες του στο addMarkerActivity όπου υλοποιείται η εισαγωγή του απινιδωτή στη βάση.

Όταν ο χρήστης πατάει τους markers εμφανίζονται ο τίτλος, οδηγίες μετάβασης προς σαυτόν και εμφανίζεται κουμπί κάτω απτό μενού για την προβολή περαιτέρω πληροφοριών.

Όταν ο χρήστης πατάει το χάρτη το menu κρύβεται και εμφανίζεται ξανά αν πατήσει πάλι το χάρτη.

- Εγγραφή και είσοδος χρήστη μέσω Email, Google Account

```
mAuth.createUserWithEmailAndPassword(email, pass)
    .addOnCompleteListener(RegisterActivity.this, new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {

            progressbar.setVisibility(View.GONE);

            if (task.isSuccessful()) {
                // Sign in success, update UI with the signed-in user's information
                Log.d(TAG, "createUserWithEmail:success");
                FirebaseUser user = mAuth.getCurrentUser();

                startActivity(new Intent(RegisterActivity.this, LoginActivity.class));
            } else {
                // If sign in fails, display a message to the user.
                Log.w(TAG, "createUserWithEmail:failure", task.getException());
                Toast.makeText(RegisterActivity.this, "Authentication failed.",
                    Toast.LENGTH_SHORT).show();
            }
        }
    });
```

Σχήμα 10: Εγγραφή χρήστη μέσω FirebaseAuth αντικειμένου.

```

login_btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        progressbar2.setVisibility(View.VISIBLE);

        String email = log_email.getText().toString().trim();
        String pass = log_password.getText().toString().trim();

        mAuth.signInWithEmailAndPassword(email, pass)
            .addOnCompleteListener(LoginActivity.this, new OnCompleteListener<AuthResult>() {
                @Override
                public void onComplete(@NonNull Task<AuthResult> task) {

                    progressbar2.setVisibility(View.GONE);
                    if (task.isSuccessful()) {
                        // Sign in success, update UI with the signed-in user's information
                        Log.d(TAG, "signInWithEmail:success");
                        FirebaseUser user = mAuth.getCurrentUser();
                        startActivity(new Intent(LoginActivity.this, AEDMapActivity.class));
                    } else {
                        // If sign in fails, display a message to the user.
                        Log.w(TAG, "signInWithEmail:failure", task.getException());
                        Toast.makeText(LoginActivity.this, "Authentication failed.",
                                    Toast.LENGTH_SHORT).show();
                    }
                }
            });
    }
});

```

Σχήμα 11: Σύνδεση χρήστη με email μέσω FirebaseAuth αντικειμένου.

Όταν ο χρήστης πατάει το κουμπί Login στο menu ενώ δεν είναι συνδεδεμένος μεταφέρεται στο LoginActivity στο οποίο μπορεί να συνδεθεί με το προφίλ του, είτε με το Google account του, ενώ αν δεν έχει προφίλ μπορεί να πλοηγηθεί από UI στο RegisterActivity όπου μπορεί να φτιάξει προφίλ.

Σε περίπτωση που ο χρήστης είναι συνδεδεμένος ήδη μεταφέρεται στο LogoutActivity όπου φαίνεται το email με το οποίο είναι συνδεδεμένο και κουμπί για αποσύνδεση.

- Εισαγωγή απινιδωτή στο χάρτη

Αρχικά, παίρνει τις συντεταγμένες όπου θα καρφитσωθεί το marker από προηγούμενο Activity μέσω extras.

Για την εισαγωγή του απινιδωτή στη βάση ο χρήστης καλείται να συμπληρώσει ένα τίτλο, μία περιγραφή και προαιρετικά φωτογραφία για τον απινιδωτή.

Η φωτογραφία πρώτα σώζεται στο FirebaseStorage, όταν ολοκληρωθεί το ανέβασμα και επιστραφεί link προς αυτή καλείται η συνάρτηση που σώζει στη βάση τα παράνω.

Αν ο χρήστης δεν έχει ανεβάσει φωτογραφία τότε το link μένει κενό.


```

//Get Image Extension
public String getExtension(Uri uri){
    ContentResolver cr = getContentResolver();
    MimeTypeMap mimeTypeMap = MimeTypeMap.getSingleton();
    return mimeTypeMap.getExtensionFromMimeType(cr.getType(uri));
}

//Upload Image to Firebase Storage
private void FileUploader(){

    if (imguri != null) {

        StorageReference ref = mStorageRef.child(System.currentTimeMillis() + "." + getExtension(imguri));

        ref.putFile(imguri)
            .addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
                @Override
                public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {

                    Task<Uri> downloadUrl = taskSnapshot.getStorage().getDownloadUrl();
                    downloadUrl.addOnSuccessListener(new OnSuccessListener<Uri>() {
                        @Override
                        public void onSuccess(Uri uri) {

                            if (uri.toString() != null) {
                                imgUrl = uri.toString();
                            }
                            Log.d(TAG, "Image Upload Success, uri.toString: " + uri.toString());
                            Log.d(TAG, "Image Upload Success, imgUrl: " + imgUrl);

                            addCollection();
                        }
                    });
                }
            })
            .addOnFailureListener(new OnFailureListener() {
                @Override
                public void onFailure(@NonNull Exception exception) {
                    // Handle unsuccessful uploads
                    Log.d(TAG, "Image Failed to Upload");
                }
            });
    }
    else{
        imgUrl = "";
        addCollection();
    }
}

```

Σχήμα 12: Ανέβασμα φωτογραφίας στο Firebase Storage και link της ανεβασμένης.

Δημιουργείται HashMap με τα δεδομένα μας και στη συνέχεια προστίθενται στο collection AEDMap όπου αποθηκεύονται οι απινιδωτές.

Με την επιτυχή πρόσθεση των δεδομένων στη βάση γίνεται ανακατεύθυνση στον αρχικό χάρτη όπου μπορούμε να δούμε τον καινούργιο απινιδωτή που προσθέσαμε.

```

//Save to Database
Map<String, Object> aed = new HashMap<>();
aed.put("Description",AEDDescr);
aed.put("Geolocation",geoloc);
aed.put("ImageUrl",imgUrl);
aed.put("Name",AEDName);

Log.d(TAG,"Image Url after putting it in the database"+imgUrl);

db.collection("AEDMap").add(aed).addOnSuccessListener(new OnSuccessListener<DocumentReference>() {
    @Override
    public void onSuccess(DocumentReference documentReference) {
        Log.d(TAG, "DocumentSnapshot added with ID: " + documentReference.getId());

        progressBar3.setVisibility(View.GONE);
        startActivity(new Intent(addMarkerActivity.this, AEDMapActivity.class));
    }
})

.addOnFailureListener(new OnFailureListener() {
    @Override
    public void onFailure(@NonNull Exception e) {
        Log.w(TAG, "Error adding document", e);
        progressBar3.setVisibility(View.GONE);
    }
});

```

Σχήμα 13: Ανέβασμα του απινιδωτή στη βάση.

- Προβολή και επεξεργασία υπάρχοντος απινιδωτή

Όταν ο χρήστης πατάει πάνω σε απινιδωτή εμφανίζεται κάτω από μενού κουμπί marker info με το οποίο μεταφέρεται στο MarkerInfoActivity στο οποίο εμφανίζονται οι πληροφορίες του συγκεκριμένου marker όπου πάτησε, σταλμένες με ως extras από προηγούμενο activity.

Τα πεδία είναι επεξεργάσιμα και ο χρήστης μπορεί να εισάγει καινούργιο τίτλο, περιγραφή και φωτογραφία. Στη βάση τα πεδία ανανεώνονται στο ήδη υπάρχον document μέσω του id.

```

db.collection("AEDMap").document(marker_id).set(aed).addOnSuccessListener(new OnSuccessListener<Void>() {
    @Override
    public void onSuccess(Void aVoid) {
        progressBar8.setVisibility(View.GONE);
        startActivity(new Intent(MarkerInfoActivity.this, AEDMapActivity.class));
    }
})

.addOnFailureListener(new OnFailureListener() {
    @Override
    public void onFailure(@NonNull Exception e) {
        Log.w(TAG, "Error adding document", e);
        progressBar8.setVisibility(View.GONE);
    }
});

```

Σχήμα 14: Ανανέωση πεδίων σε συγκεκριμένο document.

- Αναφορά προβλήματος απινιδωτή

Στο MarkerInfoActivity υπάρχει κουμπί για την αναφορά προβλήματος στο συγκεκριμένο απινιδωτή, όπου μεαφέρει το χρήστη στο ReportActivity όπου συμπληρώνει φόρμα για το πρόβλημα που υπάρχει.

Τα δεδομένα ύστερα αποθηκεύονται σε καινούργιο collection AEDReport με τρία πεδία το κάθε document, το ID του απινιδωτή (το id του document από AEDMap collection), ένα σχόλιο και ο τύπος του προβλήματος.

```
Map<String, Object> aedreport = new HashMap<>();
aedreport.put("AEDMapID", marker_id);
aedreport.put("Comment", com);
aedreport.put("Type", choice);

db.collection("AEDReport").add(aedreport).addOnSuccessListener(new OnSuccessListener<DocumentReference>() {
    @Override
    public void onSuccess(DocumentReference documentReference) {
        Log.d(TAG, "DocumentSnapshot added with ID: " + documentReference.getId());

        progressBar9.setVisibility(View.GONE);
        startActivity(new Intent(ReportActivity.this, AEDMapActivity.class));
    }
})

.addOnFailureListener(new OnFailureListener() {
    @Override
    public void onFailure(@NonNull Exception e) {
        Log.w(TAG, "Error adding document", e);
        progressBar9.setVisibility(View.GONE);
    }
});
```

Σχήμα 15: Ανέβασμα προβλήματος απινιδωτή.

- Οδηγίες μετάβασης σε απινιδωτή

Όταν ο χρήστης πατάει ένα marker κάτω δεξιά εμφανίζονται δύο επιλογές για οδηγίες μετάβασης σε αυτόν. Η λειτουργία αυτή περιέχεται στους markers του Google Maps.

- Οδηγίες εκμάθησης ΚΑΡΠΑ και χρήσης απινιδωτή μέσα από βίντεο

Στο μενού υπάρχει κουμπί learn cpr το οποίο στέλνει τον χρήστη στο σύνδεσμο του βίντεο εκμάθησης ΚΑΡΠΑ και χρήσης απινιδωτή.

- Είσοδος διαχειριστή

Οι διαχειριστές της εφαρμογής έχουν συγκεκριμένα προφίλ (με email ekav01@gmail.com, dimos01@gmail.com) όπου αναγνωρίζονται αυτόματα απτην εφαρμογή όταν συνδέονται.

Στους διαχειριστές εμφανίζεται έξτρα κουμπί report map για την προβολή χάρτη με απινιδωτές όπου έχει γίνει κάποιο report.

```
private void checkAdmin(){
    if(mAuth.getCurrentUser() != null){
        String userEmail = mAuth.getCurrentUser().getEmail();
        if(userEmail.equals(admins.get(0))){
            adminDimos = true;
            Log.d(TAG, "admin Dimos");

            showReportIcon();
        }
        else if(userEmail.equals(admins.get(1))){
            adminEkav = true;
            Log.d(TAG, "admin Ekav");

            showReportIcon();
        }
    }
}
```

- Προβολή των αναφορών προβλημάτων απτους διαχειριστές

Οι διαχειριστές έχουν πρόσβαση στον χάρτη όπου εμφανίζονται οι απινιδωτές που έχουν αναφερθεί.

```

db.collection("AEDReport")
    .get()
    .addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {
        @Override
        public void onComplete(@NonNull Task<QuerySnapshot> task) {
            if (task.isSuccessful()) {
                for (QueryDocumentSnapshot document : task.getResult()) {
                    Log.d(TAG, document.getId() + " => " + document.getData());

                    //Get data and save on list

                    Map<String, Object> markerEntry = document.getData();
                    markerEntry.put("Id", document.getId());
                    //Log.d(TAG, "List object : "+markerEntry);
                    markerMap.add(markerEntry);
                }

                db.collection("AEDMap")
                    .get()
                    .addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {
                        @Override
                        public void onComplete(@NonNull Task<QuerySnapshot> task) {
                            if (task.isSuccessful()) {
                                for (QueryDocumentSnapshot document : task.getResult()) {
                                    //Log.d(TAG, document.getId() + " => " + document.getData());

                                    //Get data and save on list
                                    for (int i=0; i<markerMap.size(); i++) {
                                        if (markerMap.get(i).containsValue(document.getId())) {
                                            markerMap.get(i).put("AEDName", document.get("Name"));
                                            markerMap.get(i).put("AEDGeo", document.get("Geolocation"));

                                            //Log.d(TAG, "List object AED : "+);
                                        }
                                    }
                                }

                                Log.d(TAG, "ALL AED REPORTS : "+markerMap);
                                setMarkersOnMap();
                            } else {
                                Log.w(TAG, "Error getting documents.", task.getException());
                            }
                        }
                    });
            } else {
                Log.w(TAG, "Error getting documents.", task.getException());
            }
        }
    });
}

```

Σχήμα 16: Ανάκτηση των απινιδωτών όπου έχουν report, απτη βάση.

Αρχικά, αποθηκεύουμε τα δεδομένα του collection AEDReport σε λίστα, έχοντας πρόσβαση στα id των απινιδωτών που έχουν αναφερθεί τους ανακτάμε απτο collection AEDMap και στη συνέχεια τους εμφανίζουμε στο χάρτη.

Βίντεο Αναπαράστασης Εφαρμογής: