Πανεπιστήμιο Αιγαίου

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτχων

321-4002 Σχεδιασμός και Ανάπτυξη Εφαρμογών Κινητού Υπολογισμού

Διδάσκων: Γκουμόπουλος Χρήστος

Ανάπτυξη εφαρμογής πληθοπορισμού (crowdsourcing) για αναφορά προβλημάτων καθημερινότητας σε Δημοτική Αρχή

Εργαστηριακοί Συνεργάτες: Χρυσολωράς Γεώργιος

Βασίλης Κασσικός

icsd13074

Αντώνης Αντωνόπουλος

icsd14009

Ευκαρπίδης Κωνσταντίνος

icsd15051

Σάμος, <ημέρα παράδοσης>, 2018

Βασίλης Κασσικός icsd13074, Αντώνης Αντωνόπουλος icsd14009, Ευκαρπίδης Κωνσταντίνος – icsd15051

1 Εισαγωγή

Το project αφορά την τεχνική του πληθοπορισμού. Την πράξη δηλαδή της εξωτερικής ανάθεσης καθηκόντων σε εθελοντές που σε άλλη περίπτωση θα εκτελούντουσαν από κάποιον υπάλληλο. Συγκεκριμένα ζητήθηκε από την ομάδα μας να αναπτύξουμε μία εφαρμογή ,σε κινητό περιβάλλον, για αναφορά προβλημάτων καθημερινότητας σε Δημοτική Αρχή. Ταυτόχρονα με την ανάπτυξη της εφαρμογής ,όπου θα χρησιμοποιήσουν οι εθελοντές (client-side) , έπρεπε να δημιουργήσουμε και την πλευρά του διαχειριστή (server-side). Το πανελ στο οποίο θα εισέρχεται ο διαχειριστής (υπάλληλος της δημοτικής αρχής) και θα επιβλέπει τις πιο πρόσφατες αναφορές σε λίστα καθώς και σε χάρτη. Το 3ο κομμάτι του project αφορά την αξιολόγηση της εφαρμογής. Η αξιολόγηση θα πραγματοποιείτε μέσα από ένα test mode που δημιουργήσαμε και θα επιλέγετε κατα την εκκίνηση της εφαρμογής. Το τελευταίο κομμάτι αυτής περιλαμβάνει ένα ερωτημάτολόγιο στο οποίο περγιγράφεται η συνολική εμπειρία του χρήστη ως αποτέλεσμα των απαντήσεων που δώσανε.

2 Τεχνολογίες & οδηγίες εγκατάστασης

2.1 Client Side

Στην πλευρά του client έχουμε χρησιμοποιήσει το **Android Studio** για την ανάπτυξη του **back-end** και του **front-end** της εφαμοργής. Για την βάση δεδομένων χρησιμοποιήσαμε το **Google Firebase**.

2.2 Server Side

Για την πλευρά της ιστοσελίδας , μεταφορτώσαμε τον ιστότοπο στο **000webhost** , το οποίο χρησιμοποιεί για **web server apache** και ως γλώσσα προγραμματισμού **back-end** γράψαμε σε **php**.

2.3 Οδηγίες εγκατάστασης

Αφού έχετε εγκαταστήσει τους **adb** drivers του κινητού σας στον υπολογιστή τότε μπορείτε να συνεχίσετε κάνοντας import του project στο Android Studio και μετέπειτα να κάνετε build & run από αυτό. Έτσι καταλήγετε με την εφαρμογή μας εγκατεστημένη στην κινητή σας συσκευή.

3 Android Manifest

Το Android Manifest xml αρχείο αποτελεί ένα πληροφοριοδότη για τον debugger καθώς και για την συσκευή που εκτελεί την εφαρμογή. Συγκεκριμένα περιέχει πληροφορίες σχετικά με τις απαιτούμενες άδειες από τον χρήστη για την έγκυρη εκτέλεση της εφαρμογής. Επίσης γνωστοποιεί στο λειτουργικό ποιο υλικό θα χρησιμοποιήσει καθώς και πληροφορίες για την εφαρμογή όπως θέμα, υπηρεσίες , δραστηριότητες και meta-data.

```
<application android:theme="@style/Theme.AppCompat.Light.NoActionBar">
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <service android:name=".utilities.GPSTracker" />

    <activity android:name=".StartActivity">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <activity android:name=".Register" />
    <activity android:name=".FormActivity" />
    <activity android:name=".ListForm" />
    <activity android:name=".MainActivity" />
    <activity android:name=".TestFormActivity" />
    <!--
         The API key for Google Maps-based APIs is defined as a string resource.
         (See the file "res/values/google_maps_api.xml").
         Note that the API key is linked to the encryption key used to sign the APK.
         You need a different API key for each encryption key, including the release key that is used to
         sign the APK for publishing.
         You can define the keys for the debug and release targets in src/debug/ and src/release/.
    -->
    <meta-data
        android:name="com.google.android.geo.API_KEY"
        android:value="AIzaSyA-Vo6DnGLv2Y8KWXV13i2x5GUwU16dGi0" />

    <activity
        android:name=".Map_reports"
        android:label="Map"></activity>
</application>
```

Εικόνα 3.1.1 : Κομμάτι από manifest (1)

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

Εικόνα 3.1.2 : Σχετικές απαιτούμενες άδειες της εφαρμογής (ίντερνετ , τοποθεσία)

```
<uses-feature
    android:name="android.hardware.camera"
    android:required="true" />
```

Εικόνα 3.1.3 : Υλικό προς χρήση από την εφαρμογή (κάμερα)

4 Java αρχεία

Η εφαρμογή αποτελείται από 12 συνολικά java αρχεία τα οποία θα εξετάσουμε το καθένα ξεχωριστά. Αυτά είναι :

- Form
- FormActivity
- GPSTracker
- ListAdapter
- ListForm
- MainActivity
- Map_reports
- Register
- SpinnerItemSelected
- StartActivity
- TestFormActivity
- User

4.1 Form

Στην κλάση Form έχουμε τα πεδία που θα έχει μία φόρμα (αναφορά). Τους setters & getters καθώς και τους constructors. Επίσης κάνουμε Override την συνάρτηση toString για μελλοντική χρήση.

```
public class Form {
    private int id;
    private String type;
    private String description;
    private String email;
    private double latitude;
    private double longitude;
    private String timestamp;
    private String url;
    private int count_up;
    private int count_down;
    private double xronos_ekt;


    public Form(){

    }

    public Form(int id, String type, String description, String email,double latitude,double longitude,String timestamp,String url,int count
        this.id = id;
        this.type = type;
        this.description = description;
        this.email = email;
        this.latitude = latitude;
        this.longitude = longitude;
        this.timestamp = timestamp;
        this.url = url;
        this.count_down = count_down;
        this.count_up = count_up;
    }

    public Form(int id, String type,double latitude,double longitude,String description,String timestamp,double xronos_ekt) {
        this.id = id;
        this.type = type;
        this.description = description;
        this.latitude = latitude;
        this.longitude = longitude;
        this.timestamp = timestamp;
        this.xronos_ekt = xronos_ekt;
    }
```

Εικόνα 4.1.1 : Πεδία και κατασκευαστές κλάσης Form

```
    @Override
    public String toString() { return getType()+" : " + getDescription() +" : "+getTimestamp(); }
}
```

Εικόνα 4.1.2 : Override toString

```
public int getId() { return id; }

public void setId(int id) { this.id = id; }

public String getType() { return type; }

public void setType(String type) { this.type = type; }

public String getDescription() { return description; }

public void setDescription(String description) { this.description = description; }

public String getEmail() { return email; }

public void setEmail(String email) { this.email = email; }

public double getLatitude() { return latitude; }

public void setLatitude(double latitude) { this.latitude = latitude; }

public double getLongitude() { return longitude; }

public void setLongitude(double longitude) { this.longitude = longitude; }

public String getTimestamp() { return timestamp; }

public void setTimestamp(String timestamp) { this.timestamp = timestamp; }

public String getUrl() { return url; }

public void setUrl(String url) { this.url = url; }

public double getXronos_ekt() { return xronos_ekt; }

public void setXronos_ekt(double xronos_ekt) { this.xronos_ekt = xronos_ekt; }

public int getCount_up() { return count_up; }

public void setCount_up(int count_up) { this.count_up = count_up; }

public int getCount_down() { return count_down; }

public void setCount_down(int count_down) { this.count_down = count_down; }
```

Εικόνα 4.1.3 : Setters & Getters

4.2 User

Αντίστοιχα με την προηγούμενη κλάση έχουμε και την κλάση του χρήστη.

```java
public class User {

    private String username;
    private String email;
    private String password;


    public User(String username, String email, String password) {
        this.username = username;
        this.email = email;
        this.password = password;
    }
    public User(){

    }

    public String getUsername() { return username; }

    public void setUsername(String username) { this.username = username; }

    public String getEmail() { return email; }

    public void setEmail(String email) { this.email = email; }

    public String getPassword() { return password; }

    public void setPassword(String password) { this.password = password; }
    @Override
    public String toString() { return getUsername()+" : "+getEmail(); }
}
```

Εικόνα 4.2.1 : Κλάση User (Constructor, Πεδία , Setters, Getters , toString)

4.3 GPSTracker

Σ ’αυτήν την κλάση υλοποιούμε τον κώδικα για την λειτουργία του GPS. Συγκεκριμένα εκτός από την λήψη των συντεταγμένων , με την χρήση του API , ελέγχουμε αν υπάρχει δυνατότητα λήψης αυτών . Δηλαδή αν το GPS ή το δίκτυο είναι συνδεδεμένα , τα μέσα για να γίνει η λήψη.

```
//orismos ths klashs ws Service
public class GPSTracker extends Service {

    private  Context mContext;
    // Flag for GPS status
    boolean isGPSEnabled = false;
    // Flag for network status
    boolean isNetworkEnabled = false;
    // Flag for GPS status
    boolean canGetLocation = false;
    Location location; // Location
    double latitude; // Latitude
    double longitude; // Longitude

    // The minimum distance to change Updates in meters
    private static final long MIN_DISTANCE_CHANGE_FOR_UPDATES = 100; // 100 meters
    // The minimum time between updates in milliseconds
    private static final long MIN_TIME_BW_UPDATES = 1000 * 60 * 1; // 1 minute
    // Declaring a Location Manager
    protected LocationManager locationManager;
    Activity activity;

    public GPSTracker() {
    }

    public GPSTracker(Context context, Activity activity) {
        this.mContext = context;
        this.activity = activity;
        getLocation();//me to poy dhmioyrgeitai to antikeimeno, kaleitai h getLocation()
    }
```

Εικόνα 4.3.1 : GPSTracker fields, constructor

```
private final LocationListener mLocationListener = new LocationListener() {
    @Override
    public void onLocationChanged(final Location location) {
        //enhmerwsh twn syntetagmenwn
        if (location != null) {
            latitude = location.getLatitude();
            longitude = location.getLongitude();
        }
    }
```

Εικόνα 4.3.2 : Listener για να τραβήξει τις συντεταγμένες

```
@SuppressLint("MissingPermission")
public Location getLocation() {
    try {


        locationManager = (LocationManager) mContext.getSystemService(LOCATION_SERVICE);
        // Getting GPS status
        isGPSEnabled = locationManager
                .isProviderEnabled(LocationManager.GPS_PROVIDER);
        // Getting network status
        isNetworkEnabled = locationManager
                .isProviderEnabled(LocationManager.NETWORK_PROVIDER);
        if (!isGPSEnabled && !isNetworkEnabled) {
            // No network provider is enabled
        } else {
            this.canGetLocation = true;
            if (isNetworkEnabled) {
                int requestPermissionsCode = 50;

                locationManager.requestLocationUpdates(
                        LocationManager.NETWORK_PROVIDER,
                        MIN_TIME_BW_UPDATES,
                        MIN_DISTANCE_CHANGE_FOR_UPDATES, mLocationListener);
                Log.d( tag: "Network",  msg: "Network");
                if (locationManager != null) {
                    location = locationManager
                            .getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
                    if (location != null) {
                        latitude = location.getLatitude();
                        longitude = location.getLongitude();
                    }
                }
            }
        }
```

Εικόνα 4.3.3 : Συνάρτηση λήψης τοποθεσίας (1ο κομμάτι)

```
    // If GPS enabled, get latitude/longitude using GPS Services
    if (isGPSEnabled) {
        if (location == null) {
            if (ContextCompat.checkSelfPermission(activity, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED &&
                ActivityCompat.requestPermissions(activity, new String[]{Manifest.permission.ACCESS_FINE_LOCATION},  requestCode: 50);

            } else {
                locationManager.requestLocationUpdates(
                        LocationManager.GPS_PROVIDER,
                        MIN_TIME_BW_UPDATES,
                        MIN_DISTANCE_CHANGE_FOR_UPDATES, mLocationListener);
                Log.d( tag: "GPS Enabled",  msg: "GPS Enabled");
                if (locationManager != null) {

                    location = locationManager
                            .getLastKnownLocation(LocationManager.GPS_PROVIDER);
                    if (location != null) {
                        latitude = location.getLatitude();
                        longitude = location.getLongitude();
                    }
                }
            }
        }

    }
} catch (Exception e) {
    e.printStackTrace();
}

return location;
}
```

Εικόνα 4.3.4 : Συνάρτηση λήψης τοποθεσίας (2ο κομμάτι)

```
public double getLatitude(){
    if(location != null){
        latitude = location.getLatitude();
    }
    // return latitude
    return latitude;
}
```

Εικόνα 4.3.4 : Κλήση μέσω REST για γεωγραφικό πλάτος

```
public double getLongitude(){
    if(location != null){
        longitude = location.getLongitude();
    }

    // return longitude
    return longitude;
}
```

Εικόνα 4.3.5 : Κλήση μέσω REST για γεωγραφικό μήκος

```
public boolean canGetLocation() { return this.canGetLocation; }


/**
 * Function to show settings alert dialog.
 * On pressing the Settings button it will launch Settings Options.
 * */
public void showSettingsAlert() {
    AlertDialog.Builder alertDialog = new AlertDialog.Builder(mContext);

    // Setting Dialog Title
    alertDialog.setTitle("Enable GPS in settings");

    // Setting Dialog Message
    alertDialog.setMessage("GPS is not enabled. Do you want to go to settings menu?");

    // On pressing the Settings button.
    alertDialog.setPositiveButton( text: "Settings", (dialog, which) → {
            Intent intent = new Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
            mContext.startActivity(intent);
    });

    // On pressing the cancel button
    alertDialog.setNegativeButton( text: "Cancel", (dialog, which) → {
            dialog.cancel();
    });

    // Showing Alert Message
    alertDialog.show();
}


@Override
public IBinder onBind(Intent arg0) { return null; }
}
```

Εικόνα 4.3.6 : Συνάρτηση επίδειξης κατάλληλων μηνυμάτων

4.4 ListAdapter

Στην κλάση ListAdapter αυτό που κάνουμε είναι να περιγράφουμε ,βάση των δικών μας αναγκών , την λειτουργία ενός ViewHolder που ανήκει στην κατηγορία των Adapters. Έτσι γίνεται πιο εύκολη και προσιτή η υιοθέτηση της αποθήκευσης δεδομένων κάνοντας την προβολή αυτών πιο εύκολη.

```
public class ListAdapter extends RecyclerView.Adapter<ListAdapter.ViewHolder> {

    private ArrayList table;

    public ListAdapter(ArrayList table) {
        this.table=table;
    }

    @NonNull
    @Override
    public ListAdapter.ViewHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int viewtype) {
        Context context = viewGroup.getContext();
        LayoutInflater inflater = LayoutInflater.from(
                viewGroup.getContext());
        View v =
                inflater.inflate(R.layout.list_item, viewGroup,  attachToRoot: false);
        ViewHolder vh = new ViewHolder(v);
        return vh;
    }

    @Override
    public void onBindViewHolder(@NonNull ListAdapter.ViewHolder viewHolder, int position) {

        viewHolder.textView.setText(table.get(position).toString());
    }

    @Override
    public int getItemCount() { return table.size(); }
    class ViewHolder extends RecyclerView.ViewHolder {
        TextView textView;

        public ViewHolder(View itemView) {
            super(itemView);
            textView = (TextView) itemView.findViewById(R.id.viewItem);
        }
    }
}
```

Εικόνα 4.4.1 : ViewHolder συναρτήσεις

4.5 ListForm

Η κλάση List είναι το δεύτερο κομμάτι της υλοποίησης , όσον αφορά την προβολή της λίστας. Περιλαμβάνει τον κώδικα που λαμβάνουμε τα δεδομένα , από την βάση δεδομένων και έναν listener για την προσθήκη κάθε μίας φόρμας από την βάση στην λίστα.

```
public class ListForm extends AppCompatActivity {

    private RecyclerView recyclerView;
    private RecyclerView.Adapter mAdapter;
    private FirebaseDatabase mFirebaseDatabase;
    private DatabaseReference mMessagesDatabaseReference;
    private ChildEventListener childEventListener;
    private ArrayList<Object> arrayList = new ArrayList<~>();
    private static final int NUM_LIST_ITEMS = 100;

    //ensomatosi recycle view sto activity
    //kai listas mesa se auto.
    ///Travame ta data apo thn db kai ta kanoume add se ena arraylist
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_list_form);
        recyclerView = (RecyclerView) findViewById(R.id.rview);
        LinearLayoutManager layoutManager = new LinearLayoutManager( context: this);
        recyclerView.setLayoutManager(layoutManager);
        mFirebaseDatabase = FirebaseDatabase.getInstance();
        mMessagesDatabaseReference = mFirebaseDatabase.getReference().child("forms");

                mMessagesDatabaseReference.addValueEventListener(new ValueEventListener() {
                    @Override
                    public void onDataChange(DataSnapshot dataSnapshot) {
                        for (DataSnapshot data:dataSnapshot.getChildren())
                        {
                            Form form = data.getValue(Form.class);
                            arrayList.add(form);
                        }
                        recyclerView.setHasFixedSize(true);
                        mAdapter = new ListAdapter(arrayList);
                        recyclerView.setAdapter(mAdapter);
                    }

                    @Override
                    public void onCancelled(@NonNull DatabaseError databaseError) {

                    }
                });

    }

}
```

Εικόνα 4.5.1 : Firebase instances & data retrieve

4.6 MainActivity

Η κλάση MainActivity αποτελεί την δραστηριότητα της οθόνης όπου ο χρήστης έχει επιλέξει την πλήρη έκδοση της εφαρμογής. Στην οθόνη αυτή έχει δυνατότητα εισόδου και εγγραφής.

Αρχικά γίνεται η σύνδεση στην βάση για την επικύρωση των διαπιστευτηρίων σε περίπτωση εισόδου. Έχουμε υλοποιήσει 2 listener για αντίστοιχα κουμπιά.

```
public class MainActivity extends AppCompatActivity {

    private static final String EMAIL = "email";
    private EditText username;
    private EditText pwd;
    private Button login;
    private Button register;
    private FirebaseDatabase mFirebaseDatabase;
    private DatabaseReference mMessagesDatabaseReference;
    private FirebaseAuth firebaseAuth;

    @Override
    protected void onCreate(final Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        username = (EditText) findViewById(R.id.emailIgn);
        pwd = (EditText) findViewById(R.id.pwdIgn);
        login = (Button) findViewById(R.id.login);
        register = (Button) findViewById(R.id.MainRegister);

        mFirebaseDatabase = FirebaseDatabase.getInstance();
        mMessagesDatabaseReference = mFirebaseDatabase.getReference().child("users");
        firebaseAuth = FirebaseAuth.getInstance();
```

Εικόνα 4.6.1 : Πεδία κλάσης, συνάρτηση onCreate & instances για την βάση

```
register.setOnClickListener((v) -> {
        Context context = MainActivity.this;
        Intent intent = new Intent(context, Register.class);
        startActivity(intent);

});
```

Εικόνα 4.6.2 : Listener για κουμπί εγγραφής

```
//listener gia sundesi
login.setOnClickListener((v) → {
        mMessagesDatabaseReference.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {

                boolean flag1 = false;
                //anazitisi user stin vasi dedomenon
                for (DataSnapshot data : dataSnapshot.getChildren()) {
                    User user = data.getValue(User.class);
                    String email = user.getEmail();
                    String password = user.getPassword();
                    if (email.equals(username.getText().toString()) && password.equals(pwd.getText().toString())) {
                        flag1 = true;
                        Context context = MainActivity.this;
                        CharSequence message = "Welcome " + user.getUsername();
                        Toast toast = Toast.makeText(context, message, Toast.LENGTH_LONG);
                        toast.show();
                        Intent intent = new Intent(context, FormActivity.class);
                        Bundle bundle = new Bundle();
                        bundle.putString(EMAIL, user.getUsername());
                        //kratame data kai ta pername se allo activity
                        if (bundle != null) {
                            intent.putExtras(bundle);
                            startActivity(intent);
                            break;
                        } else {
                            startActivity(intent);
                            break;
                        }
                    }
                }

                if (flag1 == false) {
                    CharSequence message = "Invalid credits";
                    Context context = MainActivity.this;
                    Toast toast = Toast.makeText(context, message, Toast.LENGTH_LONG);
                    toast.show();
                    username.setText("");
                    pwd.setText("");
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {

            }
        });
});
```

Εικόνα 4.6.3 : Listener για κουμπί εισόδου

4.7 Map_reports

Η κλάση Map_reports περιέχει την συνάρτηση onCreate , όπως κάθε java αρχείο και περιέχει όλες τις αρχικοποιήσεις που χρειάζονται κατά την δημιουργία της σελίδας. Η 2η και μοναδικά επιπλέον συνάρτηση είναι η onMapReady. Αφού έχει φορτωθεί ο χάρτης στην δραστηριότητα , είναι η ώρα να γεμίσει με markers που αφορούν τα σημεία αναφοράς του προβλήματος από τον εθελοντή. Επιπλέον έχουμε επιλέξει να βάλουμε διαφορετικά χρώματα στους markers ανάλογα με την κατηγορία του προβλήματος.

```java
public class Map_reports extends FragmentActivity implements OnMapReadyCallback {

    private GoogleMap mMap;
    private FirebaseDatabase mFirebaseDatabase;
    private DatabaseReference mMessagesDatabaseReference;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_map_reports);
        mFirebaseDatabase = FirebaseDatabase.getInstance();
        mMessagesDatabaseReference = mFirebaseDatabase.getReference().child("forms");
        // Obtain the SupportMapFragment and get notified when the map is ready to be used.
        SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
                .findFragmentById(R.id.map);
        mapFragment.getMapAsync( onMapReadyCallback: this);
    }


    // dimiourgia map me ola ta reports pou einai stin vasi
    //analoga tin katigoria exoume allo xroma marker
    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;

        mMessagesDatabaseReference.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                for (DataSnapshot data:dataSnapshot.getChildren())
                {
                    Form form = data.getValue(Form.class);
                    LatLng sydney = new LatLng(form.getLatitude(), form.getLongitude());
                    if(form.getType().equals("Road")){
                        mMap.addMarker(new MarkerOptions().position(sydney).title(form.getDescription()).icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_AZURE)));
                    }else if(form.getType().equals("General")){
                        mMap.addMarker(new MarkerOptions().position(sydney).title(form.getDescription()).icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_ORANGE)));
                    }else if(form.getType().equals("Lambs")){
                        mMap.addMarker(new MarkerOptions().position(sydney).title(form.getDescription()).icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_YELLOW)));
                    }else{
                        mMap.addMarker(new MarkerOptions().position(sydney).title(form.getDescription()).icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_GREEN)));
                    }
                   // mMap.addMarker(new MarkerOptions().position(sydney).title(form.getDescription()));
                    mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {

            }
        });
```

Εικόνα 4.7.1 : onCreate & συμπλήρωμα του χάρτη με markers

4.8 Register

Όπως φαίνεται και στον τίτλο της υποενότητας , είμαστε στο σημείο εγγραφής του χρήστη. Για τα τυπικά του project ζητείται ένα username, ένα email και ένας κωδικός. Για την κατοχύρωση της εγγραφής έχουμε γράψει έναν listener στον οποίο φτιάχνουμε ένα αντικείμενο της κλάσης user που αναφέραμε προηγουμένως. Έπειτα το κάνουμε push στην βάση και εμφανίζουμε ανάλογο μήνυμα στον χρήστη για την καλύτερη διεπαφή του με την εφαρμογή.

```java
public class Register extends AppCompatActivity {
    private EditText username;
    private EditText email;
    private EditText password;
    private Button registerbtn;
    private FirebaseDatabase mFirebaseDatabase;
    private DatabaseReference mMessagesDatabaseReference;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);

        username = (EditText) findViewById(R.id.username);
        email = (EditText) findViewById(R.id.email);
        password =(EditText) findViewById(R.id.pwd);
        registerbtn = (Button) findViewById(R.id.register);
        mFirebaseDatabase = FirebaseDatabase.getInstance();
        mMessagesDatabaseReference = mFirebaseDatabase.getReference().child("users");
    }

    @Override
    protected void onStart() {
        super.onStart();
        registerbtn.setOnClickListener((v) -> {
                User user = new User(username.getText().toString(),email.getText().toString(),password.getText().toString());
                mMessagesDatabaseReference.push().setValue(user);
                Context context = Register.this;
                CharSequence message = "Registed successfully";
                Toast toast = Toast.makeText(context, message, Toast.LENGTH_LONG);
                toast.show();
                Intent intent = new Intent(context,MainActivity.class);
                startActivity(intent);

        });
    }
}
```

Εικόνα 4.8.1 : Δημιουργία χρήστη και πέρασμα στην βάση

4.9 SpinnerItemSelected

Spinner είναι ένα αντικείμενο που θυμίζει το dropdown. Πρόκειται για μία λειτουργία του android που σου δίνει την επιλογή να επιλέξεις μία τιμή από ένα σετ επιλογών που εμφανίζονται κάνοντας κλικ πάνω σαυτό.

```java
public class SpinnerItemSelected  implements AdapterView.OnItemSelectedListener {


    @Override
    public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
        Toast.makeText(parent.getContext(), text: "OnItemSelectedListener : " + parent.getItemAtPosition(position).toString(), Toast.LENGTH_SHORT).show();
    }

    @Override
    public void onNothingSelected(AdapterView<?> parent) {

    }
}
```

Εικόνα 4.9.1 : Κλάση spinner

4.10 StartActivity

Ως StartActivity έχουμε ορίσει την πρώτη δραστηριότητα που τρέχει κατά το άνοιγμα της εφαρμογής. Η επιλογή αυτή φαίνεται και στο manifest.

```
<activity android:name=".StartActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

Εικόνα 4.10.1 : Ορισμός της δραστηριότητας ως κύριας

Σ ‘ αυτήν έχουμε προσθέσει 2 απλά κουμπιά για την επιλογή του χρήστη για το αν θέλει να ανοίξει το test mode ή την πλήρη έκδοση της εφαρμογής. Η διαφορά είναι ότι στην δοκιμαστική έκδοση δεν περιέχεται η λειτουργία της κάμερας ,αφού η προβολή της εικόνας γίνεται από μας προς τον χρήστη. Αυτός απλός θα επιλέγει την κατηγορία του προβλήματος που αναγνωρίζει από την φωτογραφία , και θα γράφει μία περιγραφή για το πρόβλημα και θα αποστέλλει την αναφορά.

```
public class StartActivity extends AppCompatActivity {

    private Button usr_exp;
    private Button complete;

    @Override
    protected void onCreate(final Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_start);

        usr_exp = (Button) findViewById(R.id.button);
        complete = (Button) findViewById(R.id.button2);

        complete.setOnClickListener((v) -> {
                Context context = StartActivity.this;
                Intent intent = new Intent(context, MainActivity.class);
                startActivity(intent);
        });
        usr_exp.setOnClickListener((v) -> {
                Context context = StartActivity.this;
                Intent intent = new Intent(context, TestFormActivity.class);
                startActivity(intent);
        });
    }
```

Εικόνα 4.10.2 : Κουμπιά & listeners

```
@Override
public void onBackPressed() {
    new AlertDialog.Builder( context: this)
            .setTitle("Really Exit?")
            .setMessage("Are you sure you want to exit?")
            .setNegativeButton(android.R.string.no,  listener: null)
            .setPositiveButton(android.R.string.yes, (dialogInterface, i) → {
                    StartActivity.super.onBackPressed();
            }).create().show();


}
```

Εικόνα 4.10.3 : Εμφάνιση διαλόγου με μήνυμα επιβεβαίωσης εξόδου

4.11 FormActivity

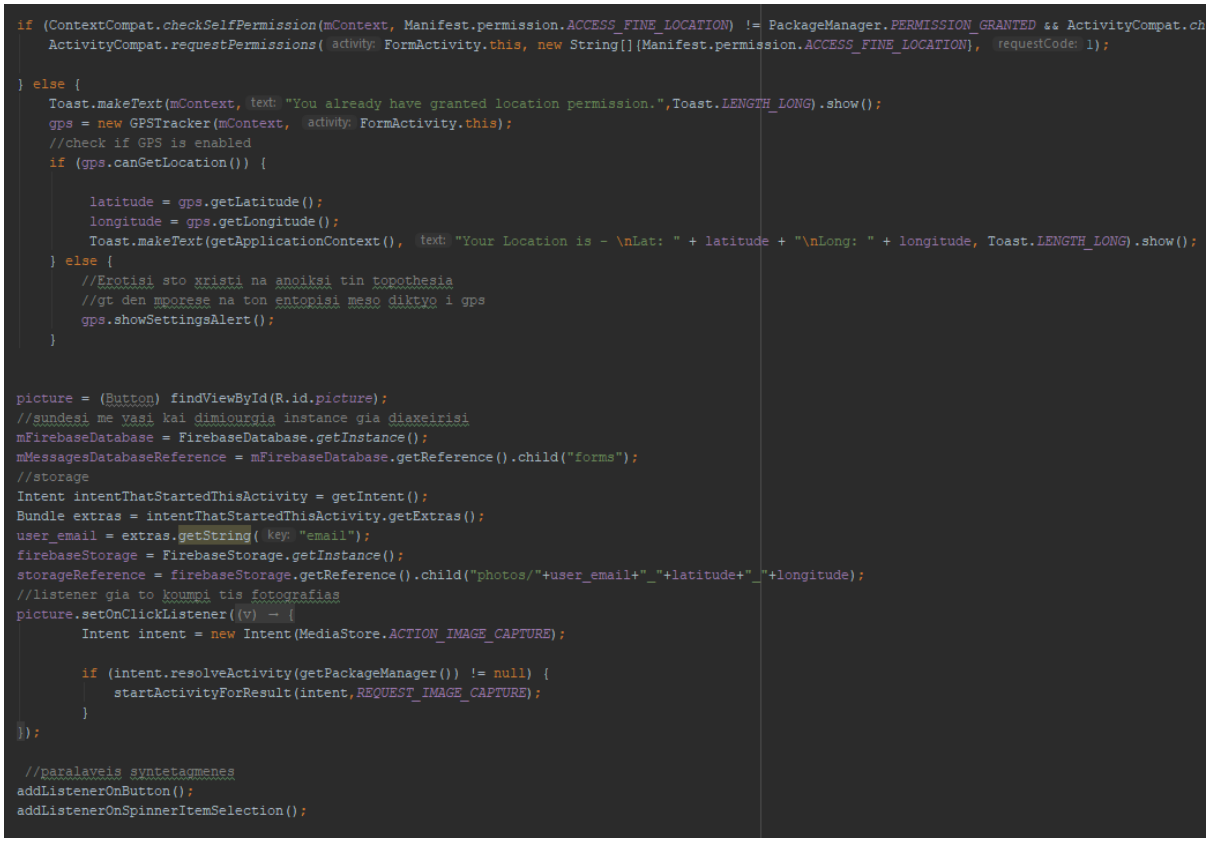
Αποτελεί την κλάση πυρήνα τις εφαρμογής αφού το μεγαλύτερο πλήθος λειτουργιών της εφαρμογής μας υλοποιούνται και ολοκληρώνονται σαυτή.

```
public class FormActivity extends AppCompatActivity {
    private FirebaseDatabase mFirebaseDatabase;
    private DatabaseReference mMessagesDatabaseReference;
    private FirebaseStorage firebaseStorage;
    private StorageReference storageReference;
    private DrawerLayout mDrawerLayout;

    static final int REQUEST_IMAGE_CAPTURE = 1;
     Context mContext;

    //antikeimeno ths prostheths klashs GPSTracker
     GPSTracker gps ;

     TextView text;
    private Spinner spinner;
    private Button send;
    private EditText description;
    private String user_email;
    private Button picture;
    private double longitude;
    private double latitude;
    private String url;
    private int count_down,id;
    private int count_up;
```

Εικόνα 4.11.1 : Πεδία κλάσης

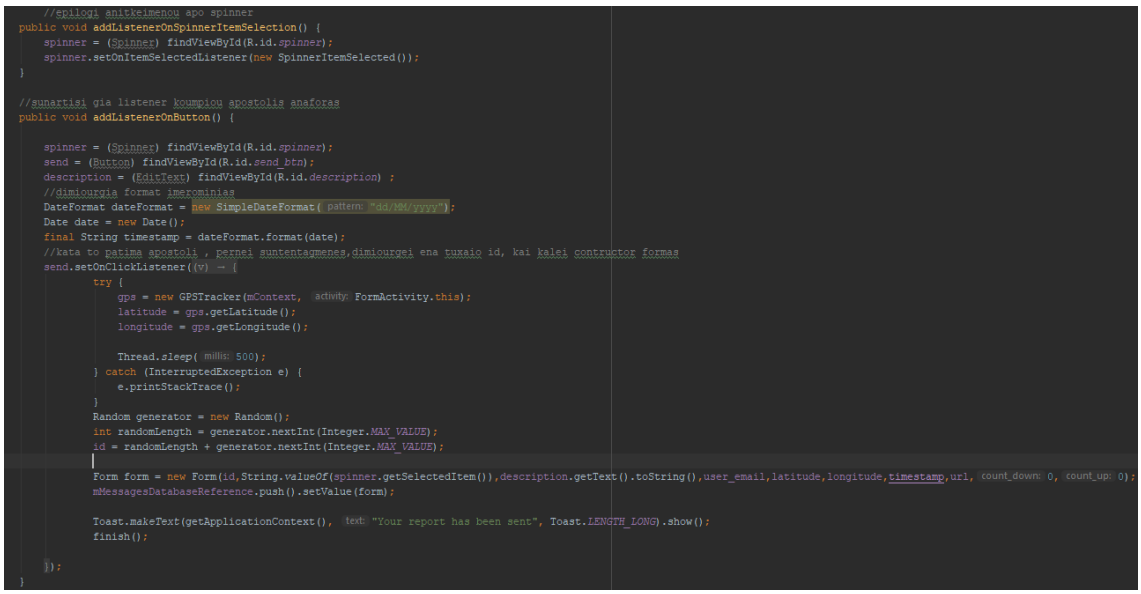
```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_form);
    mContext = this;
    //layouts kai toolbar
    mDrawerLayout = findViewById(R.id.drawer_layout);
    Toolbar toolbar = findViewById(R.id.toolbar);
    String extras2 = getIntent().getExtras().getString( key: "email");
    toolbar.setTitle(extras2);
    setSupportActionBar(toolbar);
    ActionBar actionbar = getSupportActionBar();
    actionbar.setDisplayHomeAsUpEnabled(true);
    actionbar.setHomeAsUpIndicator(R.drawable.ic_menu);
    //diaxeirisi ton antikeimenon tou navbar
    NavigationView navigationView = findViewById(R.id.nav_view);

    navigationView.setNavigationItemSelectedListener(
            (menuItem) -> {
                    //highlight to epilegmeno antikeimeno
                    menuItem.setChecked(true);
                    Intent intent;
                    if(menuItem.getItemId() == 2131296408){
                         intent = new Intent(mContext, Map_reports.class);
                    }else if (menuItem.getItemId() == 2131296409){
                         intent = new Intent(mContext, ListForm.class);
                    }else{
                        intent = new Intent(mContext, StartActivity.class);
                        finish();
                    }
                    //ekkinisi activity analoga tin epilogi kai
                    //klisimo toy drawer
                    startActivity(intent);
                    mDrawerLayout.closeDrawers();
                    return true;
            });
```

Εικόνα 4.11.2 : Εισαγωγή toolbar και κώδικας υπερσύνδεσης του με δραστηριότητες



```
if (ContextCompat.checkSelfPermission(mContext, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED && ActivityCompat.ch
    ActivityCompat.requestPermissions( activity: FormActivity.this, new String[]{Manifest.permission.ACCESS_FINE_LOCATION}, requestCode: 1);

} else {
    Toast.makeText(mContext, text: "You already have granted location permission.",Toast.LENGTH_LONG).show();
    gps = new GPSTracker(mContext, activity: FormActivity.this);
    //check if GPS is enabled
    if (gps.canGetLocation()) {

        latitude = gps.getLatitude();
        longitude = gps.getLongitude();
        Toast.makeText(getApplicationContext(), text: "Your Location is - \nLat: " + latitude + "\nLong: " + longitude, Toast.LENGTH_LONG).show();
    } else {
        //Erotisi sto xristi na anoiksi tin topothesia
        //gt den mporese na ton entopisi meso diktyo i gps
        gps.showSettingsAlert();
    }


picture = (Button) findViewById(R.id.picture);
//sundesi me vasi kai dimiourgia instance gia diaxeirisi
mFirebaseDatabase = FirebaseDatabase.getInstance();
mMessagesDatabaseReference = mFirebaseDatabase.getReference().child("forms");
//storage
Intent intentThatStartedThisActivity = getIntent();
Bundle extras = intentThatStartedThisActivity.getExtras();
user_email = extras.getString( key: "email");
firebaseStorage = FirebaseStorage.getInstance();
storageReference = firebaseStorage.getReference().child("photos/"+user_email+"_"+latitude+"_"+longitude);
//listener gia to koumpi tis fotografias
picture.setOnClickListener((v) -> {
        Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);

        if (intent.resolveActivity(getPackageManager()) != null) {
            startActivityForResult(intent,REQUEST_IMAGE_CAPTURE);
        }
});

 //paralaveis syntetagmenes
addListenerOnButton();
addListenerOnSpinnerItemSelection();
```

Εικόνα 4.11.3 : Λήψη συντεταγμένων , σύνδεση στη βάση & listener για άνοιγμα κάμερας



```
    //epilogi anitkeimenou apo spinner
public void addListenerOnSpinnerItemSelection() {
    spinner = (Spinner) findViewById(R.id.spinner);
    spinner.setOnItemSelectedListener(new SpinnerItemSelected());
}

//sunartisi gia listener koumpiou apostolis anaforas
public void addListenerOnButton() {

    spinner = (Spinner) findViewById(R.id.spinner);
    send = (Button) findViewById(R.id.send_btn);
    description = (EditText) findViewById(R.id.description) ;
    //dimiourgia format imerominias
    DateFormat dateFormat = new SimpleDateFormat( pattern: "dd/MM/yyyy");
    Date date = new Date();
    final String timestamp = dateFormat.format(date);
    //kata to patima apostoli , pernei suntentagmenes,dimiourgei ena tuxaio id, kai kalei contructor formas
    send.setOnClickListener((v) -> {
            try {
                gps = new GPSTracker(mContext, activity: FormActivity.this);
                latitude = gps.getLatitude();
                longitude = gps.getLongitude();

                Thread.sleep( millis: 500);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            Random generator = new Random();
            int randomLength = generator.nextInt(Integer.MAX_VALUE);
            id = randomLength + generator.nextInt(Integer.MAX_VALUE);

            Form form = new Form(id,String.valueOf(spinner.getSelectedItem()),description.getText().toString(),user_email,latitude,longitude,timestamp,url, count_down: 0, count_up: 0);
            mMessagesDatabaseReference.push().setValue(form);

            Toast.makeText(getApplicationContext(), text: "Your report has been sent", Toast.LENGTH_LONG).show();
            finish();

    });
}
```

Εικόνα 4.11.4 : Σχηματισμός timestamp, τυχαίο id, δημιουργία αναφοράς και push στην βαση

```
//epistrefei to epilegmeno antikeimeno apo to menu
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case android.R.id.home:
            mDrawerLayout.openDrawer(GravityCompat.START);
            return true;
    }
    return super.onOptionsItemSelected(item);
}

//sunartisi gia anevasma fotografias sto firebase
//pername apo to activity tis cameras tin eikona , kai tin metatrepoyme se bytes
//tin kanoume push kai kratame to link gia na to pername stin forma
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode == RESULT_OK) {
        Bundle extras = data.getExtras();
        Bitmap imageBitmap = (Bitmap) extras.get("data");
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        imageBitmap.compress(Bitmap.CompressFormat.JPEG,  quality: 100, baos);
        byte[] data1 = baos.toByteArray();
        Task<Uri> uploadTask = storageReference.putBytes(data1).continueWithTask((task) -> {
                if (!task.isSuccessful()) {
                    throw task.getException();
                }
                return storageReference.getDownloadUrl();
        }).addOnCompleteListener((task) -> {
                if (task.isSuccessful()) {
                    Uri downloadUri = task.getResult();
                     url = downloadUri.toString();
                    Toast.makeText( context: FormActivity.this, url, Toast.LENGTH_SHORT).show();
                } else {
                    Toast.makeText( context: FormActivity.this,  text: "upload failed: " + task.getException().getMessage(), Toast.LENGTH_SHORT).show();
                }
        });

    }
}
```

Εικόνα 4.11.5 : Συνάρτηση αποστολής φωτογραφίας στο Firebase Storage

```
//aithsh permission apo to xrhsth
@Override
public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);

    switch (requestCode) {
        case 1: {
            // If request is cancelled, the result arrays are empty.
            if (grantResults.length > 0
                    && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                // permission was granted, yay! Do the
                // contacts-related task you need to do.
                gps = new GPSTracker(mContext,  activity: FormActivity.this);
                // Check if GPS enabled
                if (gps.canGetLocation()) {

                    latitude = gps.getLatitude();
                    longitude = gps.getLongitude();
                    // \n is for new line
                    Toast.makeText(getApplicationContext(),  text: "Your Location is - \nLat: " + latitude + "\nLong: " + longitude, Toast.LENGTH_LONG).show();
                } else {
                    // Can't get location.
                    // GPS or network is not enabled.
                    // Ask user to enable GPS/network in settings.
                    gps.showSettingsAlert();
                }

            } else {
                // permission denied, boo! Disable the
                // functionality that depends on this permission.
                Toast.makeText(mContext,  text: "You need to grant location permission", Toast.LENGTH_SHORT).show();
            }
            return;
        }
```

Εικόνα 4.11.6 : Λήψη συντεταγμένων μετά την επικύρωση αδειών

4.12 TestFormActivity

Πρόκειται για την τελευταία κλάση του project , και είναι παρόμοια της FormActivity. Η βασική διαφορά των 2 κλάσεων είναι ότι σ 'αυτήν δεν πραγματοποιείται λήψη συντεταγμένων αλλά γίνεται αυτόματη τυχαία επιλογή μέσα στα πραγματικά μήκη και πλάτη. Ταυτόχρονα έχουμε προσθέσει μία μικρή καθυστέρηση (Thread.sleep) της τάξης των 0.3 milliseconds για την ομαλότερη έξοδο από την δραστηριότητα. Στην συνάρτηση addListenerOnSpinnerItemSelection παίρνουμε την επιλογή του χρήστη από το spinner. Ταυτόχρονα γίνεται υπολογισμός του χρόνου εκτέλεσης από την στιγμή που δημιουργείται η δραστηριότητα μέχρι πριν φτιαχτεί η φόρμα αφού πάτησε το κουμπί.

```
public class TestFormActivity extends AppCompatActivity {
    private FirebaseDatabase mFirebaseDatabase;
    private DatabaseReference mMessagesDatabaseReference;
    private Spinner spinner;
    private Button send;
    private EditText description;
    private int id;
    private double longitude;
    private double latitude;
    private double start;
    private double runTime;
    Context mContext;
```

Εικόνα 4.12.1 : Πεδία κλάσης

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_test_form);
    mContext = this;
    start = System.currentTimeMillis();
    mFirebaseDatabase = FirebaseDatabase.getInstance();
    mMessagesDatabaseReference = mFirebaseDatabase.getReference().child("t_forms");
    Intent intentThatStartedThisActivity = getIntent();
    addListenerOnButton();
    addListenerOnSpinnerItemSelection();
}
```

Εικόνα 4.12.2 : Αρχικοποίηση instances & κλήση συναρτήσεων

```
public void addListenerOnSpinnerItemSelection() {
    spinner = (Spinner) findViewById(R.id.spinner);
    spinner.setOnItemSelectedListener(new SpinnerItemSelected());

}
```

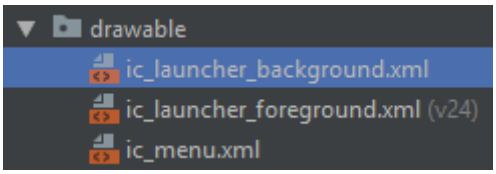
Εικόνα 4.12.3 : Επιλογή χρήστη από το spinner

```
public void addListenerOnButton() {
    spinner = (Spinner) findViewById(R.id.spinner);
    send = (Button) findViewById(R.id.send_btn);
    description = (EditText) findViewById(R.id.description);
    DateFormat dateFormat = new SimpleDateFormat( pattern: "dd/MM/yyyy");
    Date date = new Date();
    final String timestamp = dateFormat.format(date);

    send.setOnClickListener((v) → {
            Random random = new Random();
            latitude = (180 * random.nextDouble()) - 90;
            longitude = (360 * random.nextDouble()) - 180;

            Random generator = new Random();
            int randomLength = generator.nextInt(Integer.MAX_VALUE);
            id = randomLength + generator.nextInt(Integer.MAX_VALUE);
            runTime = System.currentTimeMillis() - start;
            Form form = new Form( id, String.valueOf(spinner.getSelectedItem()), latitude, longitude,description.getText().toString(),  timestamp,  xronos_ekt: runTime/1000.0);
            mMessagesDatabaseReference.push().setValue(form);

            Toast.makeText(getApplicationContext(),  text: "Your report has been sent", Toast.LENGTH_LONG).show();
            try {
                Thread.sleep( millis: 300);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            finish();
    });
```

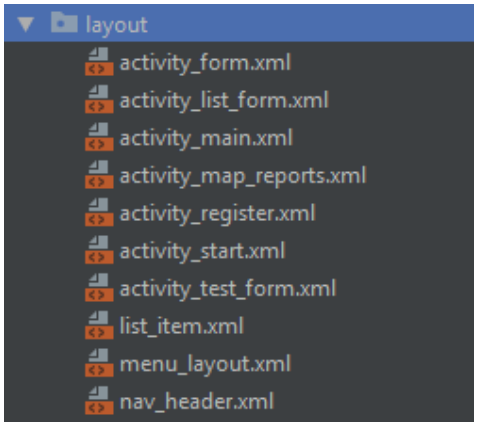
Εικόνα 4.12.4 : Listener κουμπιού αποστολής της αναφοράς

5 Λοιπά resources

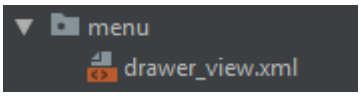
Μιλώντας για resources μιας εφαρμογής αναφερόμαστε σε drawable , layouts , values & custom widgets. Στην κατηγορία mipmap έχουμε την επαναληψιμότητα ενός αρχείου με σκοπό την συμβατότητα της εφαρμογής μας σε διαφορετικής διαγώνιου οθόνες κινητών. Στο κομμάτι των layouts έχουμε μία xml σελίδα σχεδόν για κάθε αρχείο java και όχι μόνο. Συγκεκριμένα αυτά τα αρχεία αφορούν γραφικό περιβάλλον , είτε πρόκειται για μία ολόκληρη σελίδα , είτε για ένα custom γραφικό (πχ widget). Τέλος έχουμε τα values που αφορούν strings κατά κύριο λόγο.



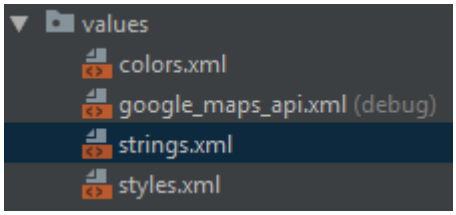
Εικόνα 5.1 : Drawable



Εικόνα. 5.2 : Λίστα layouts του project



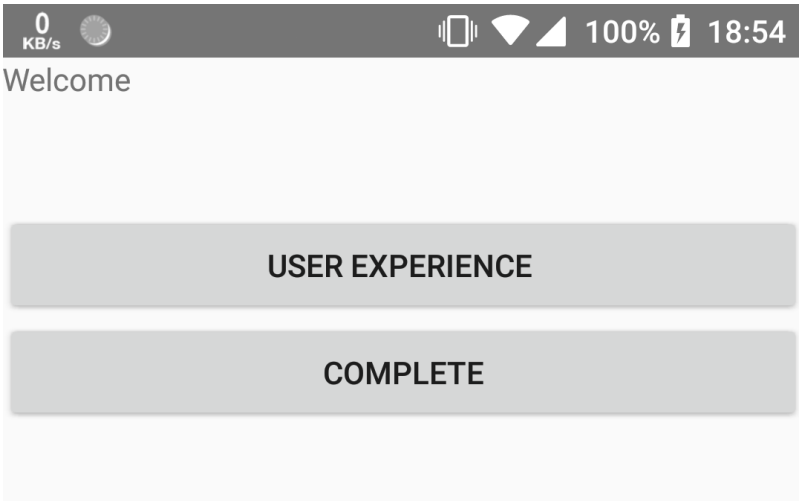
Εικόνα 5.3 : Γραφικό για το menu που ανοίγει κατά το πάτημα του navbar



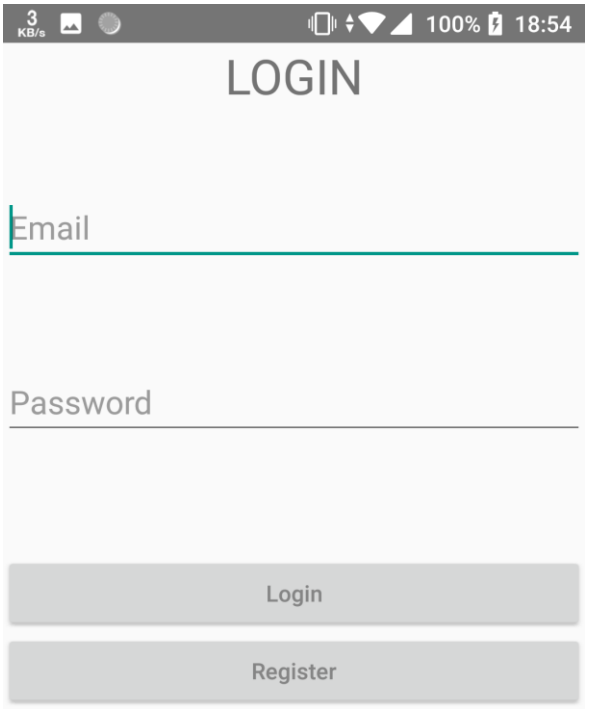
Εικόνα 5.4 : Ιδιότητες

6 Οθόνες

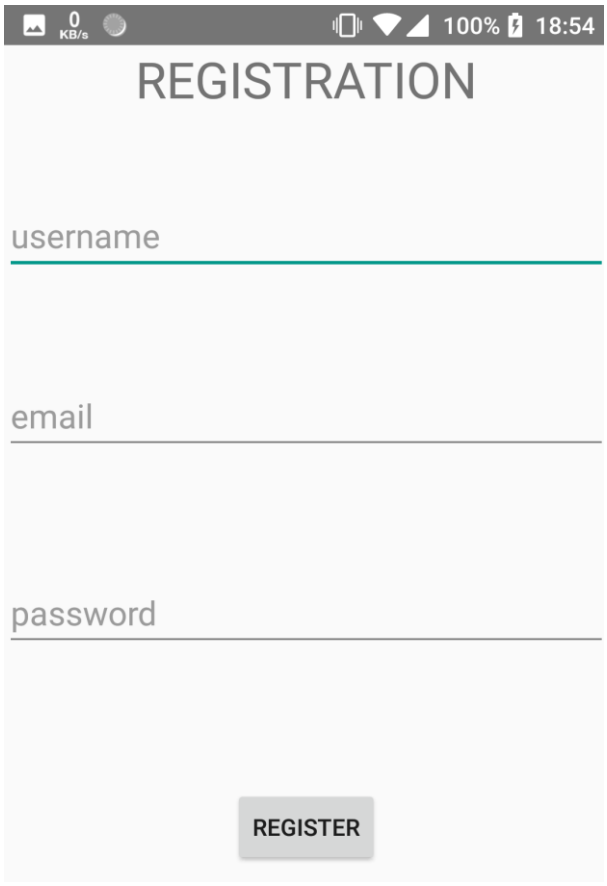
Για να προβάλουμε τις οθόνες τις εφαρμογής μας τραβήξαμε screenshots κατά την λειτουργία της. Στην πρώτη εικόνα έχουμε την πρώτη δραστηριότητα StartActivity. Ο χρήστης μπορεί να επιλέξει ανάμεσα στις δύο εκδόσεις της εφαρμογής.



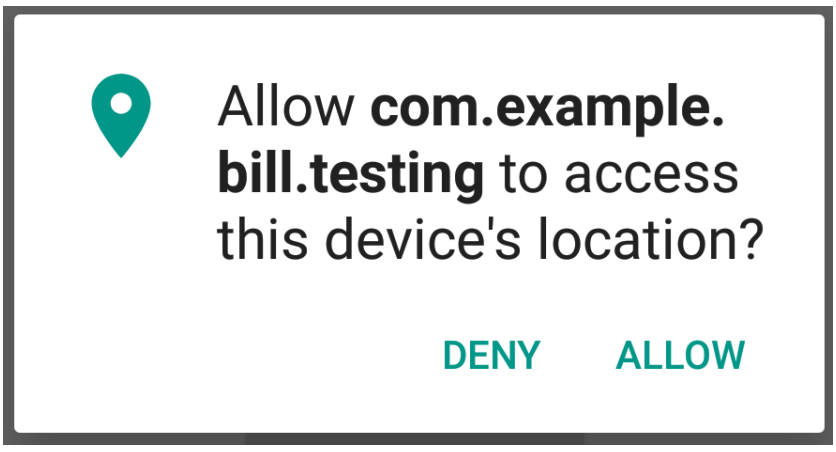
Εικόνα 6.1 : Δοκιμαστική & πλήρης έκδοση



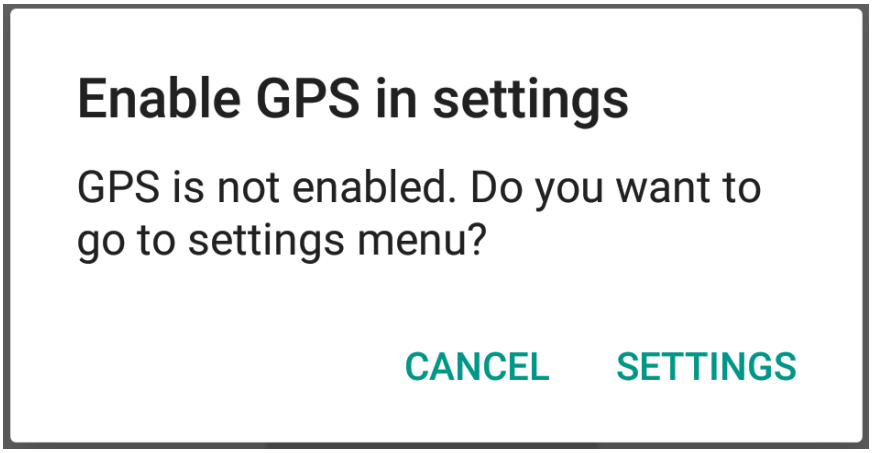
Εικόνα 6.2 : Είσοδος & δυνατότητα εγγραφής



Εικόνα 6.3 : Φόρμα εγγραφής

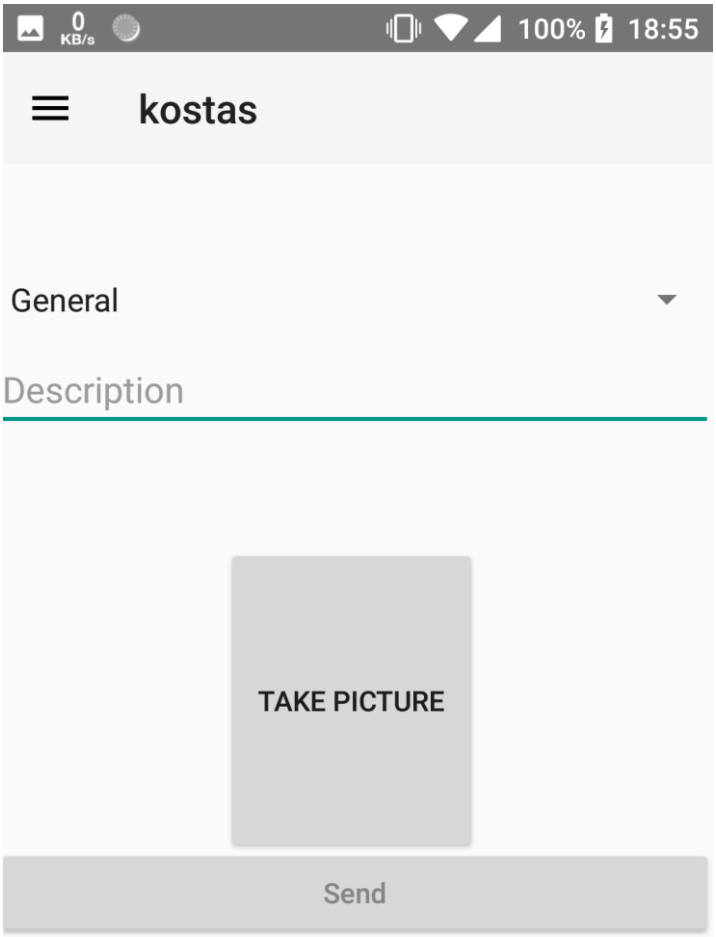


Εικόνα 6.4 : Ερώτηση για παραχώρηση άδειας τοποθεσίας



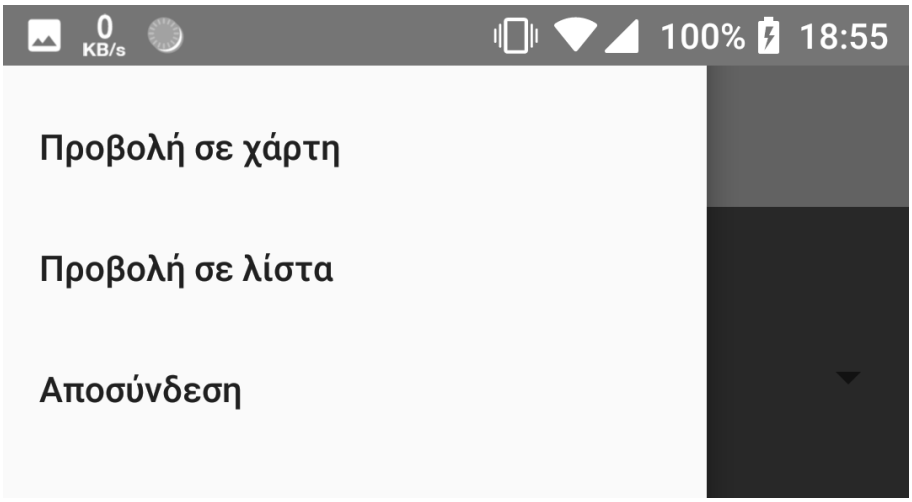
Εικόνα 6.5 : Υπενθύμιση άνοιγμα GPS

Μετά την επιτυχημένη είσοδο του χρήστη μεταφερόμαστε στην παρακάτω εικόνα.



Εικόνα 6.6 : Κυρίως οθόνη της εφαρμογής

Σε περίπτωση που ο χρήστης πατήσει το εικονίδιο της μπαρας με τις 3 γραμμές τότε του εμφανίζεται το custom toolbar μας, με δυνατότητες προβολής σε λίστα, χάρτη καθώς και αποσύνδεση.

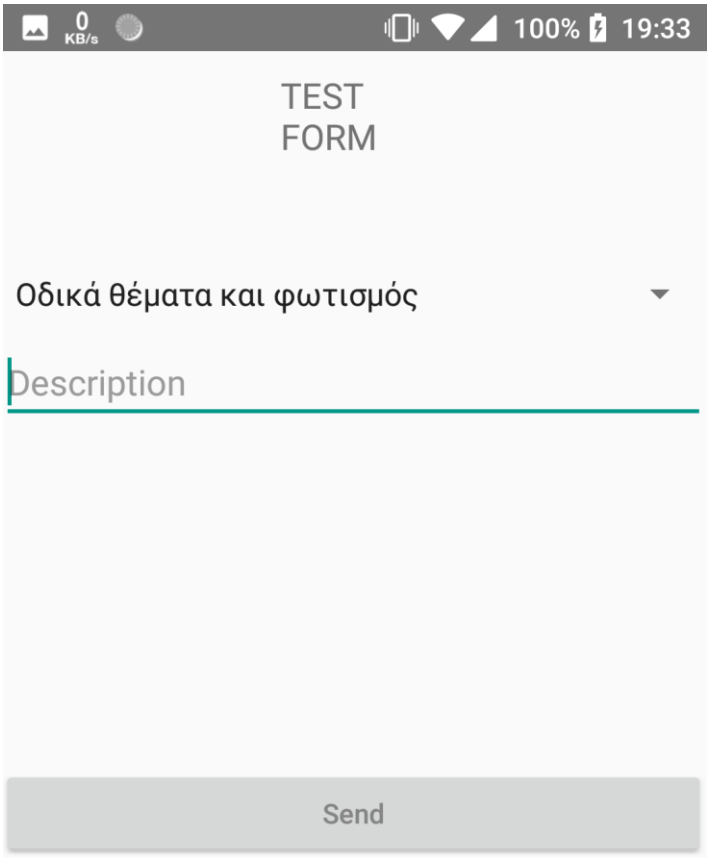


Εικόνα 6.7 : Toolbar

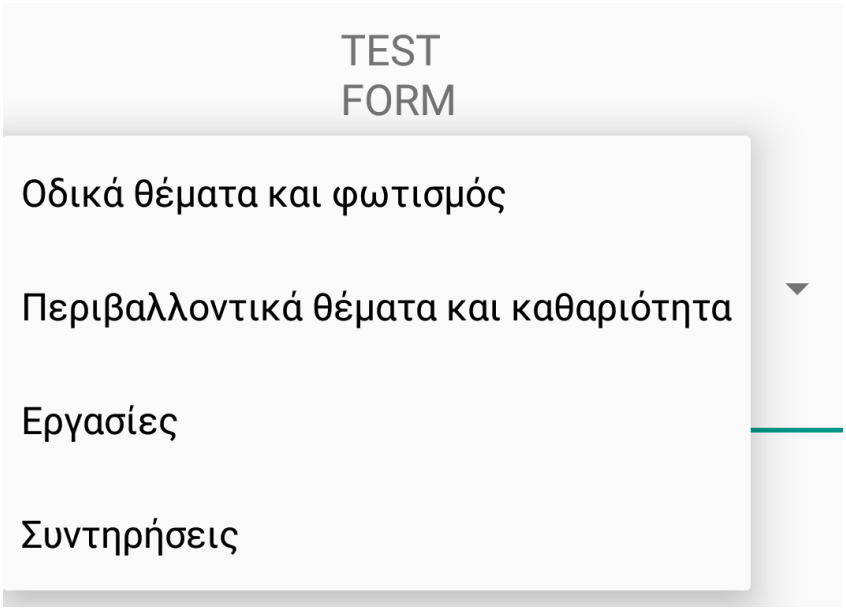
Εικόνα 6.8 : Προβολή αναφορών με markers

Εικόνα 6.9 : Προβολή σε λίστα

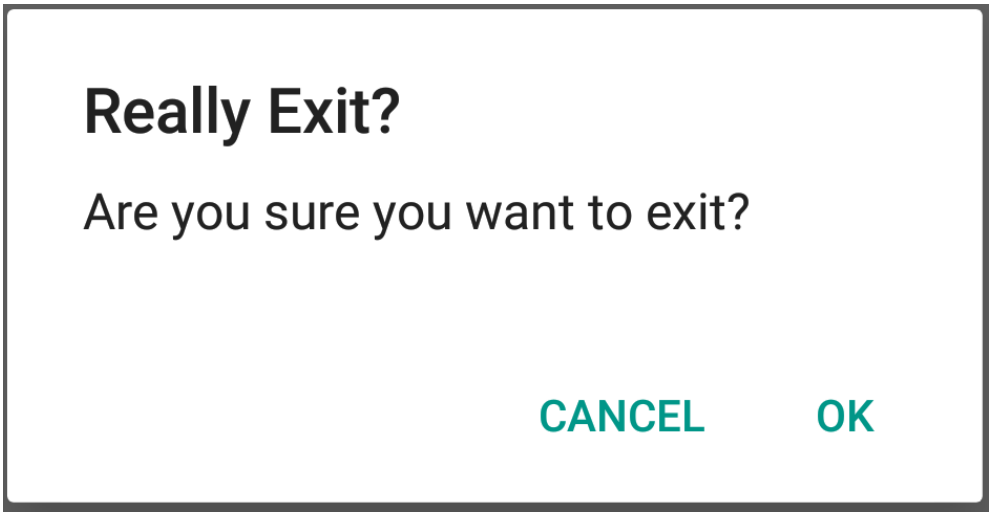
Σε περίπτωση που πατήσει ο χρήστης στο marker που θέλει να δει τότε του εμφανίζεται και η περιγραφή της αναφοράς. Στην δοκιμαστική έκδοση της εφαρμογής έχουμε αφαιρέσει το toolbar καθώς και την δυνατότητα λήψης φωτογραφίας. Η λήψη αποφεύχθηκε γιατί γίνεται προβολή φωτογραφιών του χρήστη όπως ζητήθηκε.



Εικόνα 6.9 : Δοκιμαστική έκδοση



Εικόνα 6.10 : Spinner



Εικόνα 6.11 : Μήνυμα εξόδου