



PROIECT DATM

Aplicație mobilă „Hotel Finder”

PROFESOR ÎNDRUMĂTOR

S.l.dr.ing. Adriana STAN

STUDENT

Adrian Dumitru ROȘCA

Cuprins

| | |
|--------------------------------|---|
| Introducere..... | 3 |
| Implementarea aplicației..... | 3 |
| Utilizarea aplicației | 4 |
| Secvențe de cod relevante..... | 6 |
| Concluzii..... | 7 |
| Bibliografie..... | 8 |

1. Introducere

Tema principală a acestui proiect a fost implementarea unei aplicații Android care să îi ofere utilizatorului informații despre localizarea sa pe hartă și să îi permită să găsească hotelurile aflate în proximitatea sa, precum și indicații despre cum poate să ajungă la hotelul dorit. După prezentarea pe scurt a tehnologiilor utilizate la implementarea aplicației, voi trece la descrierea modului de utilizare al acesteia, după care voi avea în vedere câteva secvențe de cod relevante sau care mi-au creat anumite probleme în timpul dezvoltării proiectului.

2. Implementarea aplicației

La implementarea aplicației, am utilizat mediul de dezvoltare Android Studio și limbajul de programare Java, precum și API-urile Google Maps și Google Places.

În primul rând, după crearea unui nou proiect în Android Studio, pentru a putea utiliza API-urile menționate anterior, am accesat <https://console.developers.google.com/apis/credentials> și am generat un API key pentru proiectul meu, pe care l-am copiat în fișierul AndroidManifest.xml. Tot în interiorul acestui fișier, am adăugat anumite permisiuni pentru a putea accesa locația dispozitivului mobil utilizat:

ACCES_NETWORK_STATE – pentru a verifica dacă sunt conectat sau nu la o rețea

INTERNET – pentru a verifica dacă sunt conectat la internet

ACCES_COARSE_LOCATION – pentru a determina locația mea aproximativă, utilizând WiFi și telefonul mobil

MAPS_RECEIVE și READ_GSERVICES – pentru a avea acces la Google Maps și Google Services

OpenGL ES V2 – necesar pentru a utiliza Google Maps V2

În continuare voi descrie pe scurt clasele java ale proiectului: MainActivity, MapsActivity, DownloadUrl și DataParser.

Clasei MainActivity îi corespunde layout-ul inițial al aplicației, activity_main.xml, care conține o componentă TextView, o componentă ImageView și o componentă Button. Ultima componentă este cea care face legătura cu layout-ul principal al aplicației, activity_maps.xml, care îi corespunde clasei MapsActivity. Layout-ul are ca și componentă principală un MapFragment, dar și două butoane și un EditText, care au rol de căutare.

În clasa principală MapsActivity.java, am setat pentru început starea inițială a hărții și am verificat dacă serviciile Google sunt disponibile. Fără acestea, aplicația se va opri.

Metoda geoLocate() are rolul de a localiza și de a marca pe hartă un anumit punct sau zonă, în funcție de textul introdus în câmpul de căutare.

```
public void geoLocate(View view) throws IOException {
    EditText text = (EditText) findViewById(R.id.editText);
    String location = text.getText().toString();
    Geocoder gc = new Geocoder(this);
    List<Address> addressList = gc.getFromLocationName(location, 1);
    Address address = addressList.get(0);
    String addressLine = address.getAddressLine(0);
    Toast.makeText(this, addressLine, Toast.LENGTH_LONG).show();
    double lat = address.getLatitude();
    double lng = address.getLongitude();
    goToLocationZoom(lat, lng, 15);
    setMarker(addressLine, lat, lng); [1]
}
```

Metoda principală a acestei clase este `getHotels()`. În primul rând, în această metodă sunt eliminate de pe harta toate markerurile existente, după care este apelată metoda `getUrl()`, folosită pentru a obține hotelurile aflate în apropiere (conform <https://developers.google.com/places/web-service/search>). Astfel am obținut un URL de tipul String, în funcție de anumite date precum latitudine, longitudine, proximitate și API key și l-am transmis ca și parametru împreună cu un obiect de tip `GoogleMap`. Într-un final, markerurile sunt adăugate pe hotelurile aflate în apropiere, folosind `getNearbyPlacesData.execute(DataTransfer)`. [8]

Clasa `DownloadURL` este folosită pentru a recepționa date de la URL folosind `HttpURLConnection` și metode pentru handling la fișiere. După primirea datelor sub formă de `googlePlacesData`, acestea sunt pasate metodei `onPostExecute`. Datele de la URL vor fi în format JSON, astfel că vor trebui parsate, lucru pentru care am creat clasa `DataParser`. Metoda `DataParser.parse(result)` este folosită pentru a parsea datele și rezultatele sunt stocate în lista `nearbyPlacesList`. Această listă are acum toate informațiile despre hotelurile aflate în apropiere, care pot fi accesate cu ușurință. Am adăugat de asemenea markere customizate, folosind metoda `ShowNearbyPlaces` din clasa `GetNearbyPlacesData`. [6]

3. Utilizarea aplicației

La pornirea aplicației, utilizatorul vede primul layout, care conține doar numele și titlul aplicației, un logo și butonul “Go To Map” (vezi figura 1).



Figura 1. Main layout

Dacă apasă pe acest buton, utilizatorul va vedea conținutul layout-ului principal al aplicației. Acesta conține în partea de jos harta pe care va fi afișată locația sa și a hotelurilor, iar deasupra acesteia se află un câmp text și două butoane, strict legate de operațiunea de căutare a unor locații/hoteluri.

Utilizatorul poate să vadă poziția sa actuală pe hartă, dar și să schimbe modul de vizualizare al hărții, prin utilizarea opțiunilor indicate în figura 2.

Prin introducerea unui anumit text sau cuvânt cheie în câmpul de deasupra hărții și apăsarea butonului “Search”, utilizatorul va vedea indicat cu un marker pe hartă locația respectivă. (vezi figura 3).

Pentru a vedea hotelurile aflate în apropierea sa, utilizatorul trebuie să apese butonul “Find Nearby Hotels”, moment în care locația sa actuală și hotelurile aflate în apropiere vor fi indicate cu markere customizate, după cum se poate vedea în figura 4.

La apăsarea pe markerul corespunzător hotelului dorit, vor apărea informații despre acel hotel (nume și adresă), precum și posibilitatea ca utilizatorul să vadă un traseu către destinația dorită (vezi figura 5).

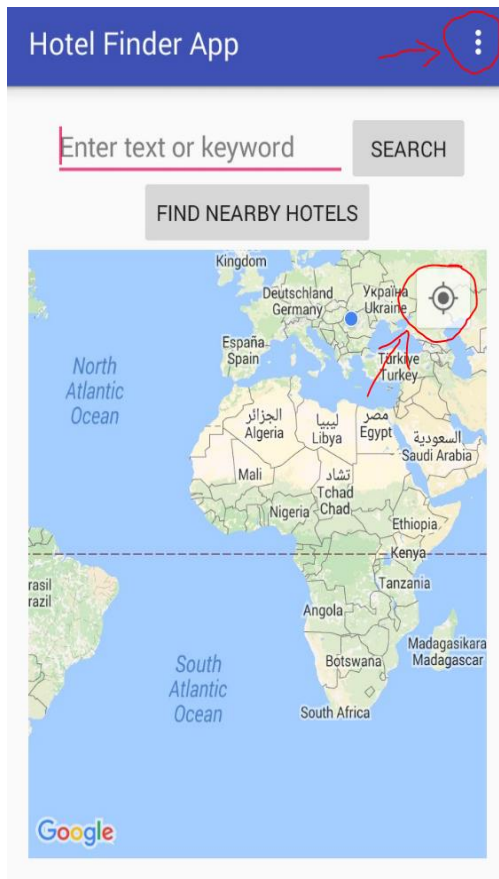


Figura 2. Maps layout

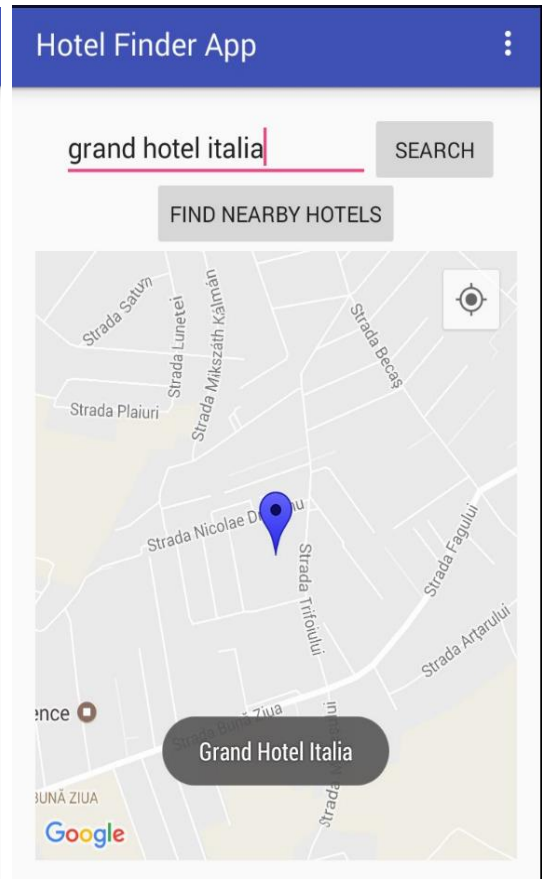


Figura 3. Search Function

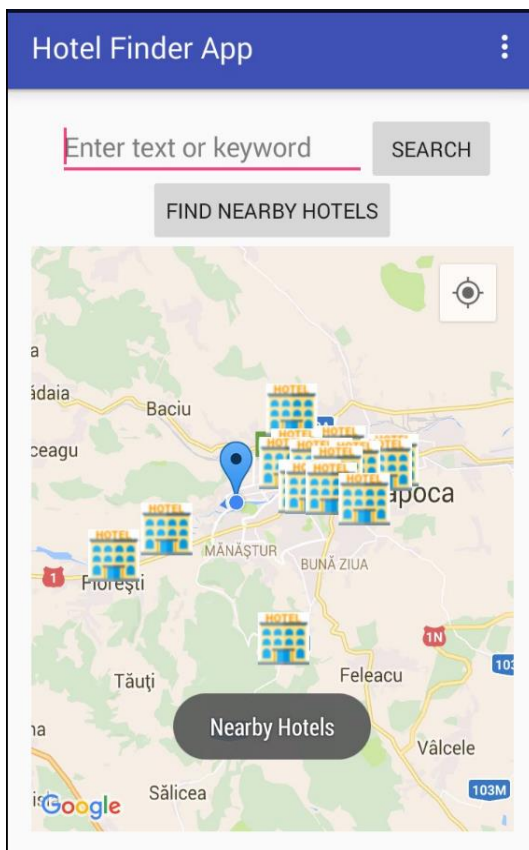


Figura 4. Find Nearby Hotels

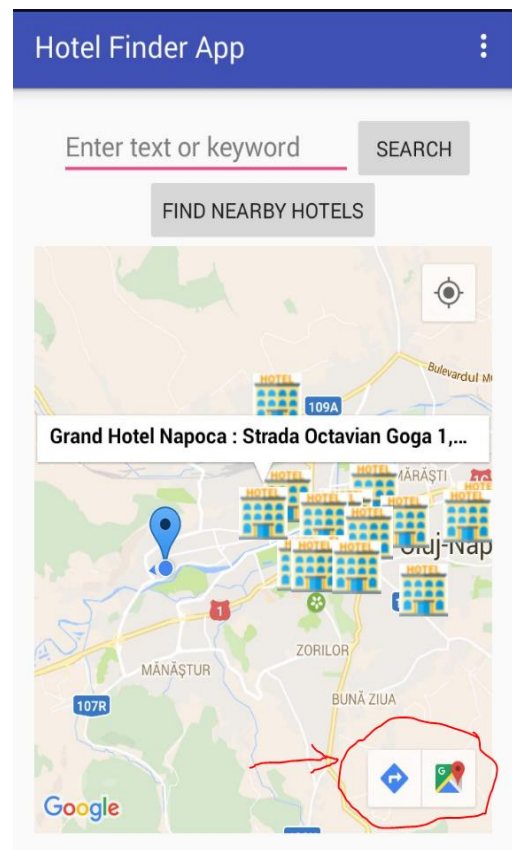


Figura 5. Markers & Indication

4. Secvențe de cod relevante

Referitor la implementarea aplicației, relevantă în realizarea acesteia și chiar cea mai interesantă mi s-a părut secvența de cod pentru căutarea în apropierea zonei actuale a utilizatorului, conform Google Places API. Aceasta se face în funcție de anumite cuvinte cheie, printr-un request HTTP URL de o anumită formă, outputul fiind sub formă de valori JSON care trebuie parsate pentru a obține informațiile dorite.

```
public void getHotels(View view) throws IOException {
    Log.d("onClick", "Button is Clicked");
    mGoogleMap.clear();
    Location myLocation = mGoogleMap.getMyLocation();
    drawMarkerCrtLocation(myLocation);
    String url = getUrl(myLocation.getLatitude(),
myLocation.getLongitude(), Hotel);
    Object[] DataTransfer = new Object[2];
    DataTransfer[0] = mGoogleMap;
    DataTransfer[1] = url;
    Log.d("onClick", url);
    GetNearbyPlacesData getNearbyPlacesData = new GetNearbyPlacesData();
    getNearbyPlacesData.execute(DataTransfer);
    Toast.makeText(MapsActivity.this, "Nearby Hotels",
Toast.LENGTH_LONG).show(); [6]
}
```

```
private String getUrl(double latitude, double longitude, String
nearbyPlace) {
    StringBuilder googlePlacesUrl = new
    StringBuilder("https://maps.googleapis.com/maps/api/place/nearbysearch/j
son?");
    googlePlacesUrl.append("location=" + latitude + "," + longitude);
    googlePlacesUrl.append("&radius=" + PROXIMITY_RADIUS);
    googlePlacesUrl.append("&type=" + nearbyPlace);
    googlePlacesUrl.append("&sensor=true");
    googlePlacesUrl.append("&key=" + "AIzaSy_QIomKWDzSUxhDg1Kln4jwCU");
    Log.d("getUrl", googlePlacesUrl.toString());
    return (googlePlacesUrl.toString()); [6]
}
```

Dacă ar fi să indic anumite secvențe de cod care mi-au ridicat probleme, aş face referire la cea în care am utilizat Geocoder pentru a obține o anumită zonă în funcție de un cuvânt cheie, zona sau locul dorit fiind greu de obținut cu exactitate, uneori textul introdus ducând chiar la oprirea aplicației.

```
public void geoLocate(View view) throws IOException {
    EditText text = (EditText) findViewById(R.id.editText);
    String location = text.getText().toString();
    Geocoder gc = new Geocoder(this);
    List<Address> addressList = gc.getFromLocationName(location, 1);
    Address address = addressList.get(0);
    String addressLine = address.getAddressLine(0);
    Toast.makeText(this, addressLine, Toast.LENGTH_LONG).show();
    double lat = address.getLatitude();
    double lng = address.getLongitude();
    goToLocationZoom(lat, lng, 15);
    setMarker(addressLine, lat, lng); [2]
}
```

5. Concluzii

Scopul acestui proiect a fost crearea unei aplicații Android pentru găsirea poziției actuale pe hartă a utilizatorului și a hotelurilor aflate în apropierea sa, precum și evidențierea avantajelor și utilității IDE-ului Android Studio și al limbajului de programare Java în realizarea unor aplicații pentru dispozitive mobile. După cum am putut observa din descrierea aplicației, putem cu ușurință obține informații relevante despre localizarea pe hartă a hotelurilor aflate în proximitate, precum și acces la traseul către acestea. Putem spune așadar că această versiune a aplicației, cu îmbunătățirile necesare, ar putea fi interesantă pentru posesorii de dispozitive mobile cu sistem de operare Android.

Bibliografie

- [1] <https://examples.javacodegeeks.com/android/android-google-places-api-example/>
- [2] <https://www.youtube.com/watch?v=k253ec4m33A&t=363s>
- [3] <https://www.youtube.com/watch?v=J3R4b-KauuI>
- [4] <http://stackoverflow.com/questions/5551028/using-google-places-api>
- [5] <http://stackoverflow.com/questions/32393134/getting-results-of-nearby-places-from-users-location-using-google-maps-api-in-a>
- [6] <https://www.androidtutorialpoint.com/intermediate/google-maps-search-nearby-displaying-nearby-places-using-google-places-api-google-maps-api-v2/>
- [7] <https://developers.google.com/maps/documentation/geocoding/start>
- [8] <https://developers.google.com/places/web-service/search>
- [9] <https://developers.google.com/maps/documentation/android-api/marker>