Proiect de dezvoltarea de aplicații pentru telefoane mobile

Find your friends

**Table of contents**:

[1 Introducere 3](#_Toc474347883)

[2 Prezentarea aplicației 4](#_Toc474347884)

[2.1 Descrierea interfeței grafice 4](#_Toc474347885)

[2.2 Prezentarea implementării 4](#_Toc474347886)

[3 Concluzii 10](#_Toc474347887)

[4 Referințe 10](#_Toc474347888)

# Introducere

Scopul acestui proiect a fost dezvoltarea unei aplicații pentru telefoane mobile care folosesc sistemul de operare Android.

Idea din spate pentru aplicație a fost crearea unui instrument care ajută conectarea prietenilor. Friend your friends oferă o posibilitate pentru utilizator de a afla locația prietenilor, a pub-urilor și găsirea celui mai apropiat pub pentru toți prieteni. Aceasta ajută utilizatorului organizarea ieșirilor cu argumente reale în funcție de poziția participanților. Funcționalitatea de căutarea direcțiilor este implementat, astfel utilizator poate să ajungă la la destinație foarte ușor.

Proiectul a fost creat în Android Studio folosind API-ul de Google Maps.

Codul sursă a aplicației se găsește pe GitHub: https://github.com/antalagota92/FindFriends

# Prezentarea aplicației

## Descrierea interfeței grafice

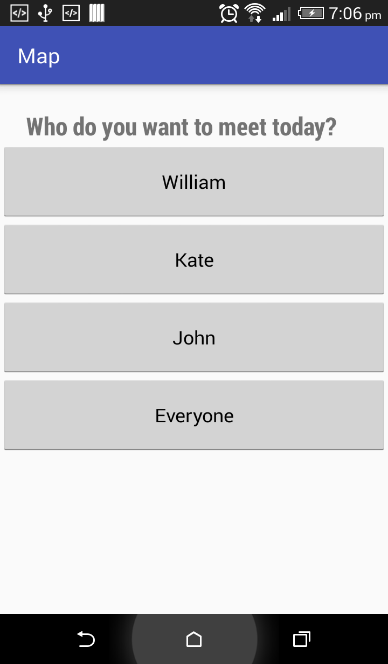
Aplicația conține două pagini. Pe pagina Home sunt niște butoane cu numele prietenilor și butonul ”Everyone”. Dacă utilizatorul apasă un buton, ajunge la pagina 2 unde se găsește harta. Pe această hartă va apărea prietenul (un marker pe poziția lui), iar și pub-urile în zonă. În cazul în care userul apasă butonul Everyone, toți prietenii apar pe hartă. Dacă userul dă click pe marker, numele prietenului este prezentat împreună cu cel mai apropiat pub.

Fig. 1. Pagina Home

## Prezentarea implementării

Aplicația folosește API-ul de Google Maps pentru afișarea hartei. Pentru aceasta a fost nevoie obținerea unor API key, introducerea lui in AnroidManifests și conectarea lui cu aplicației.

Structure proiectul e vizibil pe Figura 2.

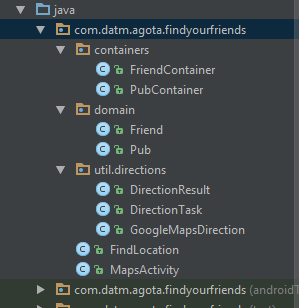


Fig. 2. Structura proiectului

Sunt două obiecte de bază, Friend și Pub. Pentru aceste avem câte un Container care conține niște instanțe și le returnează dacă e necesar. Cele două activități, FindLocation și MapsActivity sunt pentru paginile proiectului. În FindLocation se inițializează aplicația, se preia dorința utilizatorului de a găsi un prieten. Acest prieten este apoi injectat în aplicația de hartă unde apar și pub-urile din PubContainer.

Celălalt package stă pentru direcțiile, adică rutarea la un punct. Pentru această avem nevoie de DirectionResult, DirectionTask și GoogleMapsDirection. Logica se află în GoogleMapsDirection. Aceasta este o metodă asincronă având un callback în MapsActivity. Când sosește răspunsul (direcția dorită), pe punem pe hartă drumul către destinația, adică între punctul ”Me” și Pub.

Prezentarea unui Marker pentru un prieten e prezentat pe figura 3:



Fig. 3. Poziția unui Friend pe hartă

După cum se poate vedea, întotdeauna sunt prezentate cluburile. Acesta se întâmplă folosind code snippet-ul următor:

for (Pub pub : myPubs) {  
 mMap.addMarker(new MarkerOptions().position(pub.getLatLng()).title(pub.getName()).icon(BitmapDescriptorFactory  
 .*defaultMarker*(BitmapDescriptorFactory.*HUE\_AZURE*)));  
 mMap.moveCamera(CameraUpdateFactory.*newLatLng*(pub.getLatLng()));  
 mMap.animateCamera(CameraUpdateFactory.*zoomTo*(12.0f));  
}

Pe de altă parte se poate vedea că pubul cel mai apropiat apare în pop-upul markerului. Calcului celui mai apropiat club se face:

Location pubLocation = getLocation(pub.getLatLng(), pub.getName());  
 float distance = friendLocation.distanceTo(pubLocation);  
 if (distance < minDistance) {  
 closestPub = pub;  
 minDistance = distance;  
 }  
}



Fig. 4. Calculul celui mai apropiat pub tutoror prietenilor

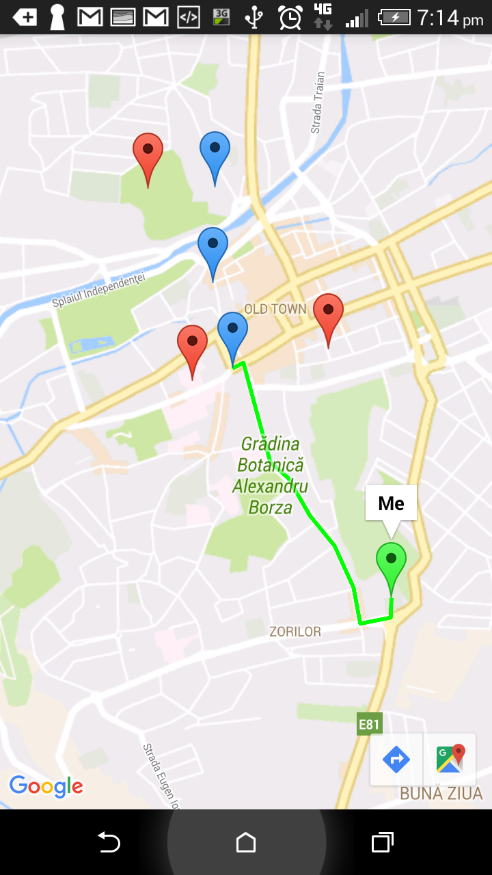
Pentru ca aplicația ajută să găsim un pub apropiat tuturor prietenilor, următorul calcul este necesar:

private Pub getClosestPubToAllFriends() {  
 final List<Friend> allFriends = FriendContainer.*getFriends*();  
 final List<Pub> allPubs = PubContainer.*getPubs*();  
 Pub closestPub = null;  
 float minTotalDistance = Float.*MAX\_VALUE*;  
 for (Pub pub : allPubs) {  
 float totalDistanceToPub = 0;  
 for (Friend friend : allFriends) {  
 final Location friendLocation = getLocation(friend.getLatLng(), friend.getName());  
 Location pubLocation = getLocation(pub.getLatLng(), pub.getName());  
 totalDistanceToPub += friendLocation.distanceTo(pubLocation);  
 }  
 if (totalDistanceToPub < minTotalDistance) {  
 closestPub = pub;  
 minTotalDistance = totalDistanceToPub;  
 }  
 }  
 return closestPub;  
}

Calculul drumului către o destinație

Pe hartă apare și poziția utilizatorului (markerul verde). Acest utilizator este „conectat” cu pub-ul ales (cel mai apropiat) cu o linie verde. Drumul este calculate cu ajutorului API-ului Google maps. Este nevoie de conexiune lla internet când folosim aplicația. O parte din acest cod destul de complex pentru acest calcul se găsește:

public ArrayList<LatLng> getDirection(final Document doc) {  
 NodeList nl1, nl2, nl3;  
 ArrayList<LatLng> listGeopoints = new ArrayList();  
 nl1 = doc.getElementsByTagName("step");  
 if (nl1.getLength() > 0) {  
 for (int i = 0; i < nl1.getLength(); i++) {  
 Node node1 = nl1.item(i);  
 nl2 = node1.getChildNodes();  
  
 Node locationNode = nl2.item(getNodeIndex(nl2, "start\_location"));  
 nl3 = locationNode.getChildNodes();  
 Node latNode = nl3.item(getNodeIndex(nl3, "lat"));  
 double lat = Double.*parseDouble*(latNode.getTextContent());  
 Node lngNode = nl3.item(getNodeIndex(nl3, "lng"));  
 double lng = Double.*parseDouble*(lngNode.getTextContent());  
 listGeopoints.add(new LatLng(lat, lng));  
  
 locationNode = nl2.item(getNodeIndex(nl2, "polyline"));  
 nl3 = locationNode.getChildNodes();  
 latNode = nl3.item(getNodeIndex(nl3, "points"));  
 ArrayList<LatLng> arr = decodePoly(latNode.getTextContent());  
 for (int j = 0; j < arr.size(); j++) {  
 listGeopoints.add(arr.get(j));  
 }  
  
 locationNode = nl2.item(getNodeIndex(nl2, "end\_location"));  
 nl3 = locationNode.getChildNodes();  
 latNode = nl3.item(getNodeIndex(nl3, "lat"));  
 lat = Double.*parseDouble*(latNode.getTextContent());  
 lngNode = nl3.item(getNodeIndex(nl3, "lng"));  
 lng = Double.*parseDouble*(lngNode.getTextContent());  
 listGeopoints.add(new LatLng(lat, lng));  
 }  
 }  
  
 return listGeopoints;  
}

public void handleGetDirectionsResult(final DirectionResult directionResult) {  
  
 final PolylineOptions rectLine = new PolylineOptions().width(5).color(Color.*GREEN*);  
 // move camera to zoom on map  
 final LatLngBounds.Builder builder = new LatLngBounds.Builder();  
 for (int i = 0; i < directionResult.getDirectionPoints().size(); i++) {  
 rectLine.add(directionResult.getDirectionPoints().get(i));  
 builder.include(directionResult.getDirectionPoints().get(i));  
 }  
  
 googleMap.addMarker(new MarkerOptions().position(directionResult.getDirectionPoints().get(0)).title("Me").icon(BitmapDescriptorFactory  
 .*defaultMarker*(BitmapDescriptorFactory.*HUE\_GREEN*)));  
 googleMap.addPolyline(rectLine);  
 // move camera to zoom on map  
 LatLngBounds bounds = builder.build();  
 // bound of points and offset from edges in pixels  
 CameraUpdate cu = CameraUpdateFactory.*newLatLngBounds*(bounds, 200, 200, 20);  
 googleMap.moveCamera(cu);  
 googleMap.moveCamera(CameraUpdateFactory.*newLatLngZoom*(directionResult.getDirectionPoints().get(0),  
 13));  
}

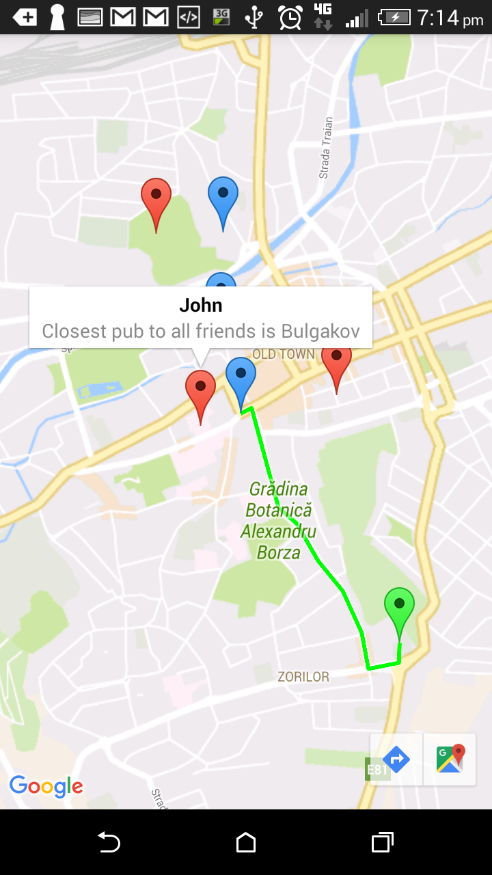


Fig. 4. Calculul celui mai apropiat pub tutoror prietenilor

# Concluzii

Acest proiect subliniază importanța și puterea API-ului Google Maps. Cu ajutorul lui construirea unei aplicații cu harta Google Maps este relativ simplă. Plasarea markerului ar fi o problem foarte complex, iar astfel se poate rezolva cu câteva linii de cod. Un alt punct forte este direcționare cu API, se poate obține direcțiile la o locație în câteva milisecunde.

Aplicația ajută pe utilizator să organizeze întâlniri, să afle unde sunt pritenii și să găsească un pub care este cel mai apropiat pentru toți.

Această implementare este numai un bun început, sunt multe direcții în care aplicația ar putea să dezvolte: prietenii și puburile ar trebui să fie luați dintr-o bază de date, să nu aibă poziții și nume hard-codați. Pe de altă parte, poziția lor ar putea să fie calculate tot de Google Maps, la fel ca și poziția utilizatorului.

# Referințe

[1] https://developers.google.com/maps/

[2] https://developer.android.com

[3] https://www.udacity.com/course/android-development-for-beginners--ud837