Finding nearby Places (using Google Places API and Maps API in Android)

Rajat Sharma[1] Bhawna Sharma[2] Sheetal Gandotra[3]
[1,2,3]Department of Computer Engineering
[1,2,3]Government College of Engineering and Technology, Chak Bhalwal, Jammu, J&K, India

Abstract— This paper is about using the Google places API and Google maps API to locate and visually display all the places requested by the user around his current location on the map. The user is provided with the feature of selecting the radius or range in meters within which it wants to search the places. App quickly identifies user position and allows user to choose the nearest Bank, ATM, Airport, Restaurant, Police Station, Bus Stand, etc. It shows user a complete list of all the facilities in the category the user has tapped on along with the distance from where he is. App is a great navigation tool anyone can take on a vacation which can help him find his way around an unfamiliar city.

Key words: Advance Programming Interface (API), API Key, Android, GMS

I. Introduction

World is expanding rapidly with the growth of mobile handset technology. As the number of users is increasing day by day, facilities are also increasing and it is estimated that this number will increase to 50 million by the end of 2020. Starting with basic early handsets which were used only to make phone calls, now they have become a vital part of our life. Now they are not only used just for making phone calls but they have plenty of uses and can be used as a Computing device, Camera, Music player, Tablet, T.V., Web browser etc. And with the new ever growing technologies, new software and, operating systems the possibilities are endless. And with the use of this app, a person will never find himself lost ever.

Find near Me is an easy to use app which quickly identifies our position and helps us find or virtually anything near us. It shows a complete list of all the facilities in the category user have tapped on along with the distance from where they are.

- ATM
- Airport
- Bank
- Bar
- Bus Station
- Café
- Car Wash
- Court House
- Dentist
- Doctors
- Gas Station
- Gym
- Hardware Store
- Hospital
- Hotel
- Movie Theatre
- Park
- Police
- Restaurant
- Salon
- Shopping Mall
- Taxi Stand
- Taxi
- Train Station
- Travel Agency
- University
- Zoo,

How many times have you had a need to find something in the middle of a situation like to locate the nearest ATM around you, in a city which you are new to or to find a Doctor? Nearby Places application can help anyone to do this and much more in just a few taps.

II. Introduction to the Google Maps API

With the Google Maps Android API, we can incorporate maps to android apps and operate upon them the way we want. The API included in android Google Mobile Service (GMS) have functionality to handle access to Google Maps servers for downloading map related data, and perform action to map gestures. We can also use API calls to add markers (Little Icons for places), polygons, and overlays to a basic map, and to change the requested view of a particular map area. It also provides additional information for map locations, and also allows user interaction with the map. The API allows us to add these graphics to a map:

- Icons for marking particular area (Markers).
- Lines (Polylines).
- Marked Area (Polygons).
- Bitmap graphics (Ground Overlays).
- Images set which are displayed on top of the map (Tile Overlays).

III. Building App

The overall process of displaying the places around user current location is divided in several parts given below.

A. *Creating API Key*

To use the Google Places API and Google Maps API for Android, we must incorporate a Google API key to our app. The type of API key we need is an Android key.

All Android apps are signed with a digital certificate (SHA-1 fingerprint) for which signee hold the private key.

Getting a key for app requires several steps. These steps are outlined below.

1) First open google console and then add goto credential.
2) Click on creates credential and then clicks on API key.
3) Add fingerprint and package name to fields and click save.
4) Add the key to app by adding anmeta-data tag to app manifest.

New Android API key appears in the list of API keys for our project. An API key is a string of characters, something like this:

AIzaSyBdVl-cTICSwYKrZ95SuvNw7adMuDt1KG0

Adding API key to manifest:

```
<application>
 ...
<meta-data
android:name="com.google.android.geo.API_KEY"
android:value="YOUR_API_KEY"/>
</application>
```

B. *Displaying Map in Android Activity*

A fragment is used to display the google map in the android activity. The fragment code in the xml form in RelativeView is given as:

```
<fragment
android:id="@+id/map"
android:layout_width="fill_parent"
android:layout_height="match_parent"
class="com.google.android.gms.maps.SupportMapFragment" />
```

C. *Detecting Current Location*

Getting user current location in Android works by means of implementing a callback. For that it is mandatory to specify request to location updates from the LocationManager class by calling requestLocationUpdates() method and passing it to a LocationListener object. LocationListener interface must implement different callback methods that the Location Manager calls when the user location changes or when the type of the service changes.

The example showing how to implement a LocationListener and request location updates is given below:

```
LocationManagerlocationManager = (LocationManager)
this.getSystemService(Context.LOCATION_SERVICE);
// Define a listener that responds to location updates
LocationListenerlocationListener = new LocationListener()
{
        //callback method 1
           public void onLocationChanged(Location
        location) {
        //handle location change event here
          }
        //callback method 2
         public void onStatusChanged(String provider, int
        status, Bundle extras) {}
        //callback method 3
        public void onProviderEnabled(String provider) {
                //handle event for provider enable
        }
        //callback method 4
           public void onProviderDisabled(String provider)
        {
                //handle event to disable the  provider
        }
 };
```

In order to receive location updates from GPS_PROVIDER, it must request user’s permission by declaring either the ACCESS_COARSE_LOCATION or ACCESS_FINE_LOCATION permission, respectively, in your Android manifest file. For example:

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

Without the presence of these permissions, application will fail at runtime when request for location updates are made.

So what happens when our Location Listener runs when our app is launched is shown below:

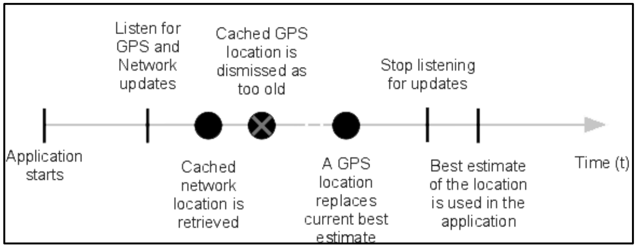
Listen for GPS and Network updates
Cached GPS location is dismissed as too old
Stop listening for updates
Application starts
Cached network location is retrieved
A GPS location replaces current best estimate
Best estimate of the location is used in the application
Time (t)

Fig. 1: Launching Steps of App.

D. *Procedure to Retrieve Data from Service*

Google APIlink is used to get the desired information around the user current location detected using Location Listener. HTTP GET is used to send the latitude and longitude information along with the type of place required like restaurant, ATM, airport, zoo, etc. After placing the request successfully, Google servers send the data in JSON format containing information about the requested place type. JSON data contains the name of the establishments, vicinity of establishments, ratings of establishment, and the location where the establishments are located. JSON data can be parsed into the required fields in the android using JSON parser library. After parsing the JSON data google map markers can be used to mark the information about the establishment onto the map. Below is the format of the google URL used to retrieve the information about establishment type.

https://maps.googleapis.com/maps/api/place/nearbysearch/{xml_or_json}?location={Latitude},{Longitude}&radius={value}&types={eg:airport,atm,etc}&sensor=true&key={API_KEY}

Example of retrieving all cafe using GET request is given below:

URL :
https://maps.googleapis.com/maps/api/place/nearbysearch/json?
location=32.67256602639742,74.893 69913582729
&radius=20000
&types=café
&sensor=true
&key=##

Response from google servers received by our application in JSON format is:

```
{
  "html_attributions”: [],
  "results”: [
    {
      "geometry”: {
        "location”: {
          "lat”: 32.705341,
 "lng”: 74.87308100000001
        }
      },
"icon”:
"https://maps.gstatic.com/mapfiles/place_api/icons/cafe71.png",
       "id”:
"c183311d510b8f5ee8bf5c684a87060a28ab78ea",
       "name”: "Cafe Coffee Shop",
       "opening_hours”: {
         "open_now”: true,
         "weekday_text”: []
       },
       "place_id”: "ChIJhzxBkaeEHjkRNyio90nlCXg",
       "rating”: 3.9,
       "scope”: "GOOGLE",
       "types”: [ "cafe", "food", "point_of_interest",
"establishment”],
       "vicinity”: "Inside BahuChowk, Chitkara Singh
Chowk, Near Railway
Road, North Block"
     },
]  "status" : "OK"
}
```

The received in JSON data is parsed and the information about the establishment like name, vicinity, image, reviews, latitude, longitude, etc. are taken out of the JSON data and are marked onto the map in our activity.

```
Marker m = mGoogleMap.addMarker(markerOptions);
//marker information //contains the establishment information
```

The overall process of getting the places information around current location is:

1) User clicks on place type.
2) App sends the request to the google servers.
3) Google servers return the data in JSON format.
4) App parses the JSON format and marks the information received onto the map in the app.

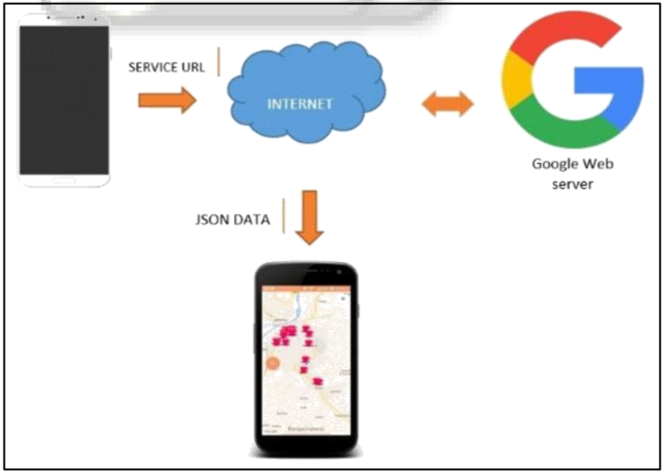

Fig. 2: Diagrammatical View.

IV. Conclusion

This navigational tool Finding Nearby Places shall be very useful to locate all the facilities available in the vicinity of the particular place along with its distances. It quickly identifies the person’s position and one shall never find him lost in the unknown places.

References

[1] Google methods references: https://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwiOkObdw9nMAhXMKo8KHakQDS0QFggcMAA&url=https%3A%2F%2Fdevelopers.google.com%2Fplaces%2Fandroid-api%2F&usg=AFQjCNF-2SV2jcHBO1HTnOrAwuF_lnNH6w&sig2=FDttY6pHqua_RXDtNnUvxg

[2] Google Maps reference: https://developers.google.com/android/reference/com/google/android/gms/maps/package-summary

[3] Places API reference: https://developers.google.com/android/reference/com/google/android/gms/location/places/PlaceDetectionApi