

User Guide

Navigation

We all know what navigation is, it displays the position, location, and direction of a user at a particular instant. For example, if the user wants to know his/her location, or wants to travel from point A to point B, the first thing a person does is take out his/her smartphone and check their location and start navigating (google maps, apple maps or other maps available in the market).

As we know for outdoor navigation, GPS works fine, but when it comes to indoor navigation, GPS fails to give the exact position. So, to overcome this problem, WIFI or EMF sensors of the phone are used to determine the location.

This application has two parts:

Outdoor navigation

Indoor navigation

To load and install the application follow the steps below:

1. Use <https://www.dropbox.com/s/dhd1n8va636tytj/app-debug.apk?dl=0> link to download and load the application on your smartphone.
2. Click on the .apk file. It will prompt for a user permission to Install or cancel. Click on install app and wait for the installation to complete (1-3 seconds).
3. Click on done or open. It will prompt for the user permission to access the location of the device. Enable the location services for the app.
4. Re-launch the app.
5. The app is ready to use.

Some benefits of using this application:

- Navigation (Indoor + Outdoor)
- Ad Free
- Easy calibration
- Clean and clear designs
- Shows information of a place when used in outdoor mode
- Small size
- Tells the Position of the user (Co-ordinates) with a blue dot on the map
- Simple to use, use it like a normal navigation app
- Tells the position of user by displaying coordinates according to the WIFI routers / access points.
- Internet connection, WIFI and GPS are required
- Detailed World map (Outdoor Location)
- Easy Readability
- Fast
- Free
- Good GUI

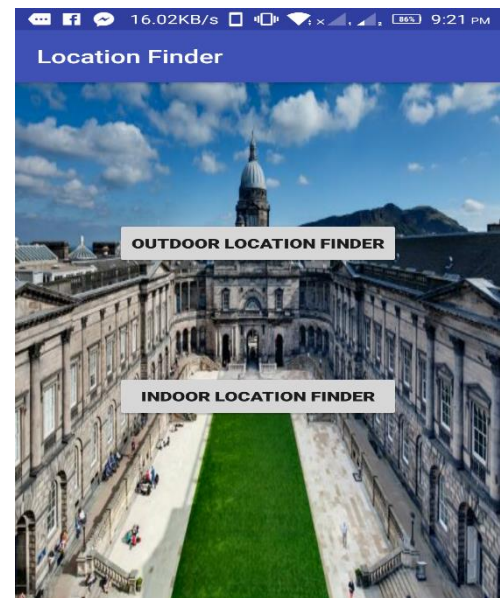


Figure 1: Application Menu: Outdoor and Indoor location finder

This app can be used by anyone, but it will be particularly beneficial for those who are frequent travellers. Tourists can also use this app to find a particular place.

People in an office/building can use this app to locate their position (indoor tracking) where GPS won't work with much accuracy. Students can find this app apt as they can find a particular room inside their campus building while being connected to their WIFI.

This app can run on devices with OS 4- Ice Cream Sandwich and above and API level 14 For this particular app, it has been tested on OS 7- Nougat with the API level 26. The minimum SDK level the app can run smoothly is 16, and the target/compiled SDK version is 26.

Outdoor navigation

As the name suggests, outdoor navigation is for navigating outside.

In this, you can see a blue dot in figure 1 which tells the user his/her current location. If a user is checking other location and wants to know his/her location, then he/she can touch the top right button (just below the search box), and he/she will be directed back to their current location.

There is a search-bar feature also, where the user can check the location of any place on earth with a marker on it. This marker tells some details about the place. There is another feature available with the app which is the information feature. The icon (i icon) on the top left. When the user searches for a place and clicks on this info button, then it will tell the user about the information of the place. This information includes address, phone number, email id, price rating, etc. These information's are collected from the Google and information available on the internet.

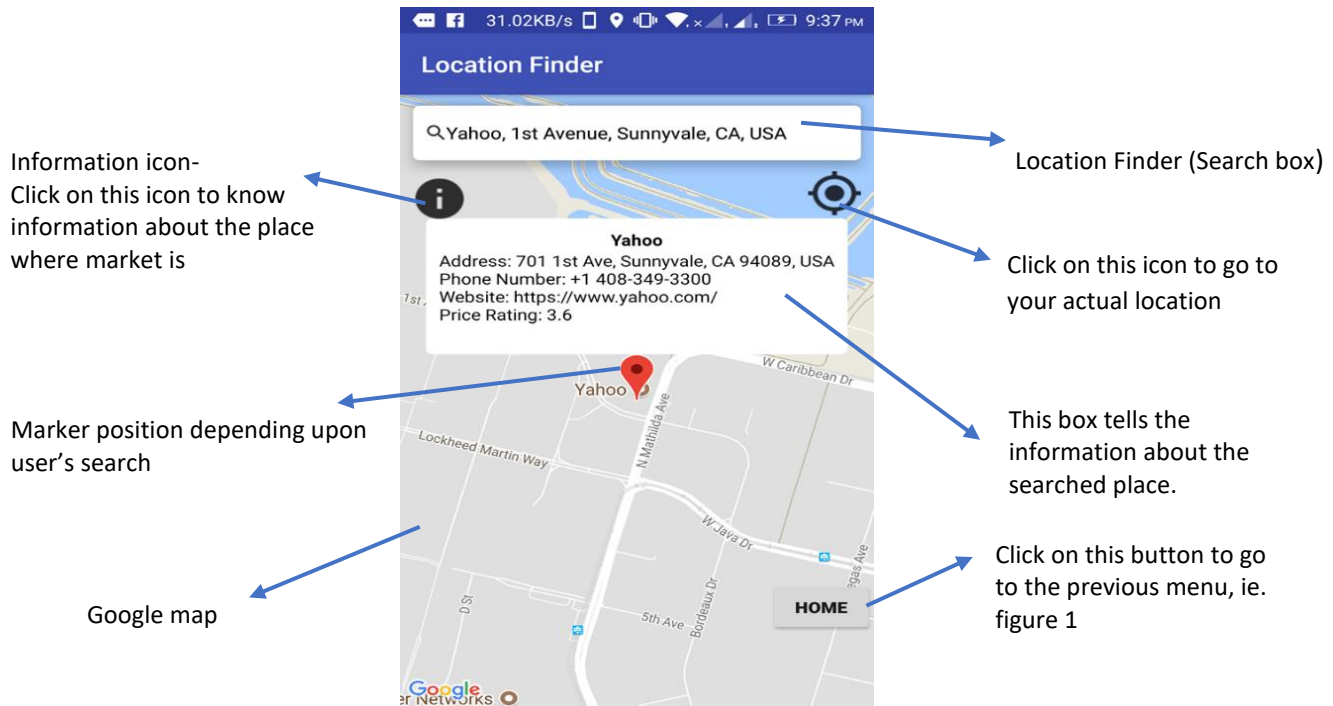


Figure 2: In Outdoor location when Yahoo is searched, it displays information about the place collected from the google.

Indoor Navigation

If the user clicks on indoor location finder in figure 1, then he is redirected to figure 3 shown on the right side.

Here, there are three buttons, Point1, Point2, Point3 which collect the x and y positions of the WIFI access points available inside the building. Click on the three buttons (wait for at least 10-20 mili seconds to let it save the data), then the position of x and y axis of the router(WIFI access points) will appear on the text box (shown in blue colour). After the points appear, click on the Display button. This tells the exact position of the user inside the building by calculating it. This is shown in the green text box. Again, to go back to figure 1, there is a home button at the bottom right corner which will redirect the user to figure 1 or the main menu.

Requirements

A smartphone should have android version 4 or above with WIFI connectivity and EMF sensors.

Google play services is must, otherwise it will prompt to download the google play service

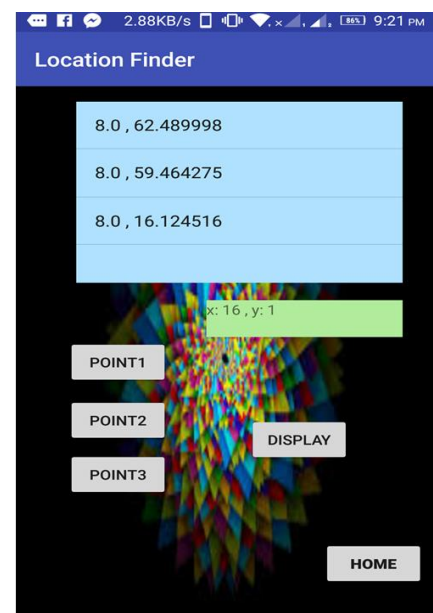


Figure 3: Indoor navigation, Points collected from different WIFI access points

Permissions

GPS and network location: Needed by the Location Finder (name of application) to calculate the GPS coordinates.

WIFI connectivity: Needed by Location Finder to calculate the location by computing the coordinates of WIFI access points.

Prevent phone from sleeping: Needed by the application to keep the display on while using the navigation.

Note ** The app might work differently on different devices depending upon the quality of sensors used by the device

Programmer Guide

The input for Location Finder for outdoor navigation is taken from internet / WIFI and GPS. It helps the user to find his exact location with a blue marker on the map.

The input for indoor location finder is basically from WIFI access points, because GPS fails to tell the exact position of the user when he/she is inside a building.

The major classes used in this application by the developer are

Outdoor Navigation Classes

MainActivity.class
MapsActivity.class
PlaceAutocompleteAdapter.class
PlaceInfo.class

Indoor Navigation Classes

IndoorsActivity.class
InMap.class

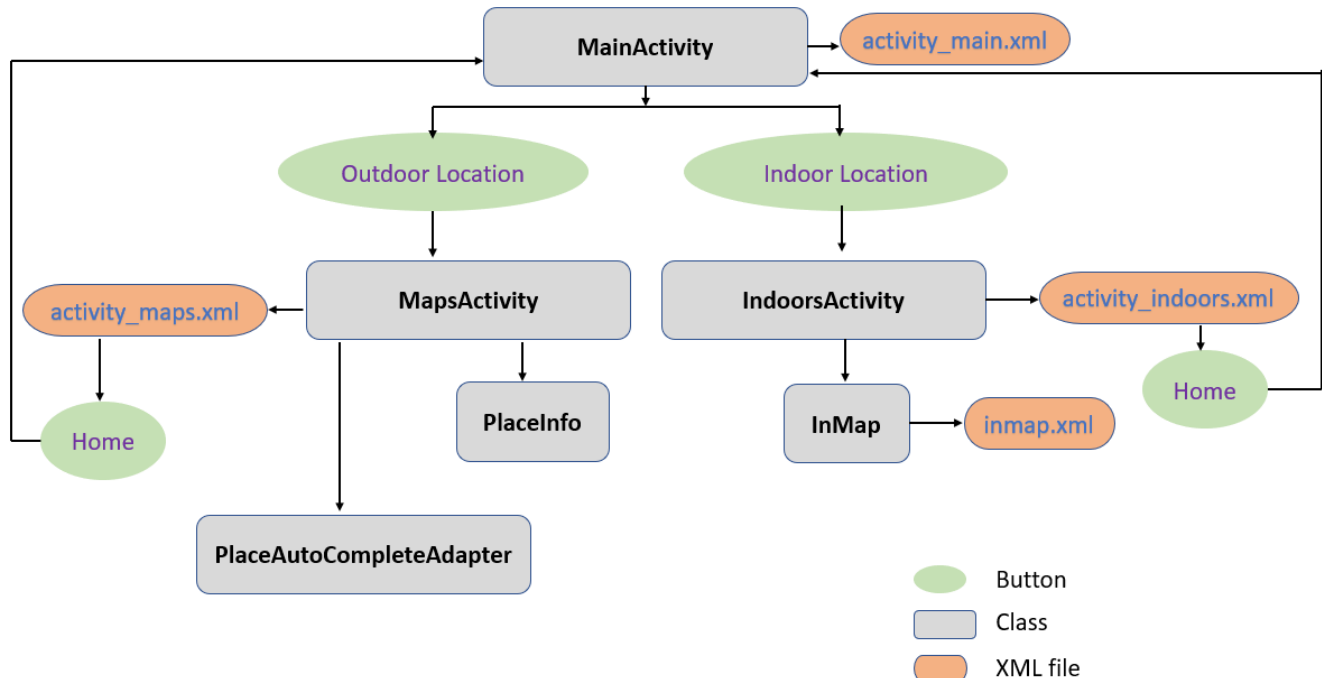


Figure 4: Full cycle of Location Finder Application

Sensors used: WIFI , GPS(location sensor)

Key listener's used :

- OnClickListener() for buttons.
- OnItemClickListener() for clicking on a particular value.
- LocationListener() to determine the current location of the user
- OnCompleteListener()

The key functions and variable used in the whole code are:-

In **MainActivity** class the first thing is checking for the google API by **isServicesOK()**. This function checks whether Google play services is available or not. If it is available then it goes to the initialize **init()** function. Where it goes to two main classes – **MapsActivity** and **IndoorActivity** class.

MapsActivity class is outdoor location finder class. It implements **OnMapReadyCallback** and **GoogleApiClient**. It is a method that is triggered when the map is ready to be used and provides a non-null instance of **GoogleMap**. In the method **onMapReady()**, the permissions are checked (GPS). In **onCreate()** various buttons are initialized and xml file is linked. In **init()**(initialize method) Google client is initialized, Marker is set, location is searched and set to a point. Here another class is called **AutocompletePrediction** is implemented which helps to predict the names of the places automatically by users entered words. Another class **Placeinfo** is also implemented where the information about the place is stored. These information are Address, phone number, Email ID, Latitude and longitude of that place are stored accordingly. A method called **goHome()** is also there with a button called **HOME**. So everytime where the user press the HOME button it will call **launchMain()** that will eventually bring the user back to the **MainActivity** class.

AutocompletePrediction class is directly taken from GoogleAPI. **PlaceInfo** is a class that stores the value of the information collected from the marker.

IndoorsActivity class is the indoor location finder class. It extends **AppCompatActivity** class. Principally, this offers a backport of the action bar. In this class the first thing which is done here is checking the permissions of WIFI state and WIFI manager. In **initiallizeIndoors()** method **startScan()** method is called which starts scanning the **WIFI**. Then the three methods **storeP1()**, **storeP2()**, **StoreP3()** methods are used to store the level of signal coming from the WIFI. Then the distance is calculated by the method **getDistance()**. It checks the signal level between the wifi reference points and subtracts them accordingly to find the particular location of the user. All these are done in trainng phase (**getRouterPosition()**) and positioning phase(**getuserPosition()**). After calculating the required values of “x” and “y” coordinates of the user through the WIFI routers, this is sent to **UpdateCurrentPosition()** which updates the position of the user in the APP. Again like Outdoor location finder, there is a HOME button to go back to the main menu. **onResume()** – registering both the sensors
onPause() – unregistering both the sensors to save the battery life

InMap class is pretty similar to the built in class for google maps with some minute changes to it. It functions similar to the MapsActivity class.

Tools used:

Android Studio 3.0.1

JRE 1.8.0_152-release-915-b01 amd64

Device used : Lenovo k33a42 with OS – 7 The app can run on devices with OS 4- Ice Cream Sandwich and above and API level 16

For this particular app has been tested on OS 7- Nougat with the API level 26.

The minimum SDK level the app can run smoothly is 16 and the target/compiled SDK version is 26.

Runtime permissions – with 6.0 Android Marshmallow, Google has made it compulsory to tell the users what all special permissions are requesting by the app rather than blindly accepting all the permissions at install time. Now the user is prompted to accept the permissions which are required during the app.

Following are my manifest.xml snippet for requesting the permission.

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

Extra features:

Outdoor location: - Prediction search for places.

Detailed information of the searched place.

HOME button functionality for the whole app.

Magnify icon + current location finder icon.

Indoor location: - Buttons for saving the data from the WIFI access points.

Display button and a text box (coloured) to show the x and y coordinates.

To reduce the App size –

1) Shrink the image.png.

The following is the snippet in the build.gradle that have been used to shrink the app size.

```
defaultConfig
{
    multiDexEnabled true
}
buildTypes
{
    release
    { minifyEnabled true
      shrinkResources true
      proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
    }
}
```

- minify is an Android tool that helps to decrease the size of the application when we build it. It's a powerful code/statement which means smaller apk files. It detects the unused code and libraries and ignores them while building the project.
- Shrinkresources also helps in shrinking the resources which your application is not using.

In the above snippet multiDexEnabled is true because the developer has used over 65,536 methods including the methods in all the libraries. Google API uses many libraries due to which the developer has to enable the multiDexEnable feature so that he can include more methods in his code.

****This navigation app can be integrated with device's camera for geo tagging and floor maps also. That can be included in updated versions of the application.**

References:

- For google Auto place prediction <https://developers.google.com/places/android-api/autocomplete>
- For google place API <https://developers.google.com/places/web-service/details>
- For Indoor Positioning H.-H. L. I. a. C. Liu, "Implementation of Wi-Fi Signal Sampling on an Android Smartphone for Indoor Positioning Systems," *sensors*, pp. 1-16, 2017.
- And some GitHub based tutorials