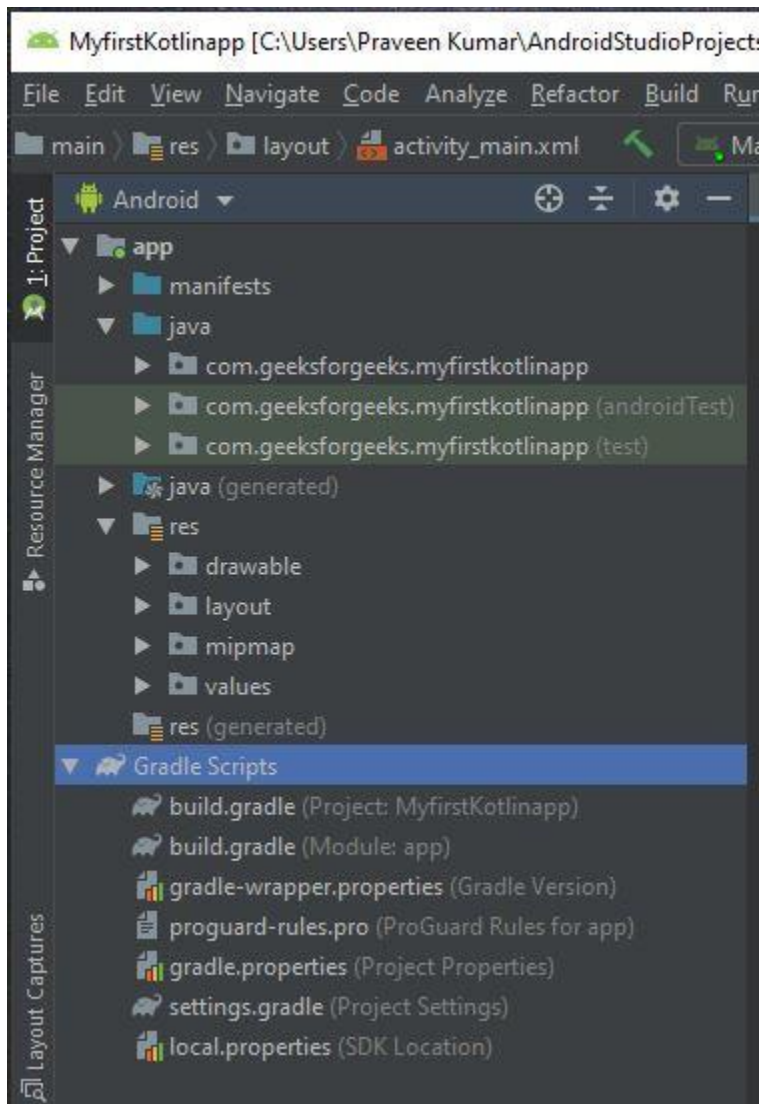


## Lab Exercise-11: Location based Services (ATM, Branch and Caller Location)

### Project Structure (Example)

---



Open `app > res > layout > activity_main.xml`. This file defines the layout for the user interface (UI). A UI in Android is defined in XML files.

### `activity_main.xml`

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    xmlns:app="http://schemas.android.com/apk/res-auto"
```

```
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">
```

```
<fragment
    android:id="@+id/mapFragment"
    android:name="com.google.android.gms.maps.SupportMapFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_above="@+id/listView"/>
```

```
<Button
    android:id="@+id/buttonFindAtms"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Find ATMs"
    android:layout_alignParentBottom="true"/>
```

```
<ListView
    android:id="@+id/listView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_above="@+id/buttonFindAtms"
    android:visibility="gone" />
```

```
</RelativeLayout>
```

**Open app > res > values> colors.xml**

## **colors.xml**

```
<?xml version="1.0" encoding="utf-8"?>

<resources>

    <color name="black">#FF000000</color>

    <color name="white">#FFFFFFF</color>

</resources>
```

## **strings.xml**

```
<resources>

    <string name="app_name"> Atm-nearme</string>

</resources>
```

## **Root folder of the application**

### **AndroidManifest.xml**

```
<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:tools="http://schemas.android.com/tools">


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

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


    <application

        android:allowBackup="true"

        android:dataExtractionRules="@xml/data_extraction_rules"

        android:fullBackupContent="@xml/backup_rules"

        android:icon="@mipmap/ic_launcher"
```

```
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Atmearme"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

        <meta-data
            android:name="com.google.android.geo.API_KEY"
            android:value="AIzaSyDRYczTDT6kSz-bWGrLGh6WCaBgUSL-1Dk"/>

    </application>

</manifest>
```

**Application root folder → java →**

## **MainActivity.java**

```
package com.idrft.atm_nearme; // We're using special tools to build an app that finds nearby ATMs.
import android.Manifest;
import android.content.Intent;
```

```
import android.content.pm.PackageManager;

import android.net.Uri;

import android.os.Bundle;

import android.view.View;

import android.widget.AdapterView;

import android.widget.AdapterView.OnItemClickListener;

import android.widget.ArrayAdapter;

import android.widget.ListView;


import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.app.ActivityCompat;

import androidx.core.content.ContextCompat;


import com.google.android.gms.location.FusedLocationProviderClient;

import com.google.android.gms.location.LocationServices;

import com.google.android.gms.maps.CameraUpdateFactory;

import com.google.android.gms.maps.GoogleMap;

import com.google.android.gms.maps.OnMapReadyCallback;

import com.google.android.gms.maps.SupportMapFragment;

import com.google.android.gms.maps.model.LatLng;

import com.google.android.gms.maps.model.MarkerOptions;


import java.util.ArrayList;

import java.util.List;


// Our main activity starts here.

public class MainActivity extends AppCompatActivity implements OnMapReadyCallback {

    // We need to ask for permission to access location.

    private static final int LOCATION_PERMISSION_REQUEST_CODE = 1;
```

```

private GoogleMap mMap;

private FusedLocationProviderClient fusedLocationProviderClient;

private ListView listView;

private List<AtmLocation> atmLocations;

@Override

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_main);

    fusedLocationProviderClient = LocationServices.getFusedLocationProviderClient(this);

    // We're using Google Maps to show the locations.

    SupportMapFragment mapFragment = (SupportMapFragment)
getSupportFragmentManager().findFragmentById(R.id.mapFragment);

    mapFragment.getMapAsync(this);

    // We're setting up the list view.

    listView = findViewById(R.id.listView);

    listView.setVisibility(View.GONE);

    // Here are the names and coordinates of ATMs.

    atmLocations = new ArrayList<>();

    atmLocations.add(new AtmLocation("State Bank Of India ATM", new LatLng(17.390747309709223,
78.44114807143532)));

    atmLocations.add(new AtmLocation("State Bank Of India ATM", new LatLng(17.390747309709223,
78.44114807143532)));

    atmLocations.add(new AtmLocation("Karur Vysya Bank ATM", new LatLng(17.390778691098642,
78.44119223074773)));

```

```
atmLocations.add(new AtmLocation("Icici Bank Atm", new LatLng(17.39086461619988,
78.44223976006604)));

atmLocations.add(new AtmLocation("HDFC Bank ATM", new LatLng(17.388992835701426,
78.4425735055428)));

atmLocations.add(new AtmLocation("ICICI Bank Atm", new LatLng(17.392685314894678,
78.43992304664734)));

atmLocations.add(new AtmLocation("ICICI Bank ATM", new LatLng(17.392359438315882,
78.43778408775188)));

atmLocations.add(new AtmLocation("State Bank Of India ATM", new LatLng(17.39315807762303,
78.44216364420214)));

atmLocations.add(new AtmLocation("State Bank ATM", new LatLng(17.39437099550929,
78.4404153877519)));

atmLocations.add(new AtmLocation("Axis Bank ATM", new LatLng(17.38903711457186,
78.43771072885643)));

atmLocations.add(new AtmLocation("Axis Bank ATM", new LatLng(17.392913436141797,
78.44183137670638)));
```

```
// We're making a list of ATM names.
```

```
ArrayAdapter<String> adapter = new ArrayAdapter<>(this, android.R.layout.simple_list_item_1,
getAtmNames());
```

```
listView.setAdapter(adapter);
```

```
// When an ATM is clicked, we show its location on the map.
```

```
listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {

    @Override

    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

        showAtmLocationOnMap(atmLocations.get(position));

    }

});
```

```
// When the "Find ATMs" button is clicked, we look for nearby ATMs.
```

```
findViewById(R.id.buttonFindAtms).setOnClickListener(new View.OnClickListener() {
```

```
        @Override  
        public void onClick(View v) {  
            findNearbyAtms();  
        }  
    }  
};  
}
```

```
// When the map is ready, we set up things.
```

```
@Override  
public void onMapReady(GoogleMap googleMap) {  
    mMap = googleMap;  
    requestLocationPermission();  
}
```

```
// We request permission to access the device's location.
```

```
private void requestLocationPermission() {  
    // If we don't have permission, we ask for it.  
    if (ContextCompat.checkSelfPermission(this, android.Manifest.permission.ACCESS_FINE_LOCATION)  
        != PackageManager.PERMISSION_GRANTED) {  
        ActivityCompat.requestPermissions(this, new  
String[]{android.Manifest.permission.ACCESS_FINE_LOCATION},  
LOCATION_PERMISSION_REQUEST_CODE);  
    } else {  
        // If we have permission, we show the user's location on the map.  
        mMap.setMyLocationEnabled(true);  
        getLastKnownLocation();  
    }  
}
```

```
// We get the last known location of the user.
```



```

private void getLastKnownLocation() {

    // We check if we have permission to access location.

    if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {

        // TODO: Handle the situation where permissions are not granted.

        return;
    }

    // If we have permission, we get the last known location.
    fusedLocationProviderClient.getLastLocation()

        .addOnSuccessListener(this, location -> {

            if (location != null) {

                LatLng currentLocation = new LatLng(location.getLatitude(), location.getLongitude());

                // We move the camera to the user's location.

                mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(currentLocation, 15));

            }

        });
}

// We handle the response when the user grants or denies permission.

@Override

public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull
int[] grantResults) {

    super.onRequestPermissionsResult(requestCode, permissions, grantResults);

    if (requestCode == LOCATION_PERMISSION_REQUEST_CODE) {

        // If permission is granted, we show the user's location on the map.

        if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {

            if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {

```

```

        // TODO: Handle the situation where permissions are not granted.
        return;
    }
    mMap.setMyLocationEnabled(true);
    getLastKnownLocation();
} else {
    // If permission is denied, we handle it here.
    // TODO: Handle the case where permission is denied.
}
}
}
}

```

// We find and display nearby ATM locations on the map.

```

private void findNearbyAtms() {
    mMap.clear(); // Clear existing markers on the map
    for (AtmLocation location : atmLocations) {
        mMap.addMarker(new MarkerOptions().position(location.getLatLng()).title(location.getName()));
    }
    // We move the camera to show the nearby ATMs.
    mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(atmLocations.get(0).getLatLng(), 12));
    listView.setVisibility(View.VISIBLE); // Show the list of ATM names.
}

```

// We show the location of a specific ATM on the map and provide directions.

```

private void showAtmLocationOnMap(AtmLocation atmLocation) {
    mMap.clear(); // Clear existing markers on the map
    mMap.addMarker(new
MarkerOptions().position(atmLocation.getLatLng()).title(atmLocation.getName()));
    mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(atmLocation.getLatLng(), 15));
}

```

```

// We create a link to open Google Maps for navigation.

LatLng currentLocation = new LatLng(17.39765403313162, 78.44976611797014); // User's current
location

String directionsUrl = "http://maps.google.com/maps?saddr=" + currentLocation.latitude + "," +
currentLocation.longitude +

    "&daddr=" + atmLocation.getLatLng().latitude + "," + atmLocation.getLatLng().longitude;

Intent intent = new Intent(android.content.Intent.ACTION_VIEW, Uri.parse(directionsUrl));

startActivity(intent); // Open Google Maps for navigation.

}

```

```

// We get a list of ATM names for the list view.

```

```

private List<String> getAtmNames() {

    List<String> atmNames = new ArrayList<>();

    for (AtmLocation location : atmLocations) {

        atmNames.add(location.getName());

    }

    return atmNames;

}

```

```

// We handle the back button press.

```

```

@Override

public void onBackPressed() {

    if (listView.getVisibility() == View.VISIBLE) {

        listView.setVisibility(View.GONE); // Hide the list view.

        mMap.clear(); // Clear existing markers on the map.

    } else {

        super.onBackPressed(); // Handle the back button as usual.

    }

}

```

```
}  
}  
  
// We define a class to hold ATM location information.  
class AtmLocation {  
    private String name;  
    private LatLng latLng;  
  
    // Constructor to create an ATM location.  
    AtmLocation(String name, LatLng latLng) {  
        this.name = name;  
        this.latLng = latLng;  
    }  
  
    // Method to get the ATM name.  
    String getName() {  
        return name;  
    }  
  
    // Method to get the latitude and longitude of the ATM.  
    LatLng getLatLng() {  
        return latLng;  
    }  
}
```

## **build.gradle** (1<sup>st</sup> build.gradle file – application gradle file)

```
plugins {  
    id 'com.android.application'  
}  
  
android {  
    namespace 'com.idrbt.atm_nearme'  
    compileSdk 33  
  
    defaultConfig {  
        applicationId "com.idrbt.atm_nearme"  
        minSdk 24  
        targetSdk 33  
        versionCode 1  
        versionName "1.0"  
  
        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"  
    }  
  
    buildTypes {  
        release {  
            minifyEnabled false  
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'  
        }  
    }  
  
    compileOptions {  
        sourceCompatibility JavaVersion.VERSION_1_8  
        targetCompatibility JavaVersion.VERSION_1_8  
    }  
}
```

```
}  
}
```

```
dependencies {
```

```
    implementation 'androidx.appcompat:appcompat:1.6.1'
```

```
    implementation 'com.google.android.material:material:1.5.0'
```

```
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
```

```
    implementation 'com.google.android.gms:play-services-maps:18.1.0'
```

```
    implementation 'com.google.android.gms:play-services-maps:17.0.1'
```

```
    implementation 'com.google.android.gms:play-services-location:18.0.0'
```

```
    testImplementation 'junit:junit:4.13.2'
```

```
    androidTestImplementation 'androidx.test.ext:junit:1.1.5'
```

```
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'
```

```
}
```