MY LOCATION FINDER APP

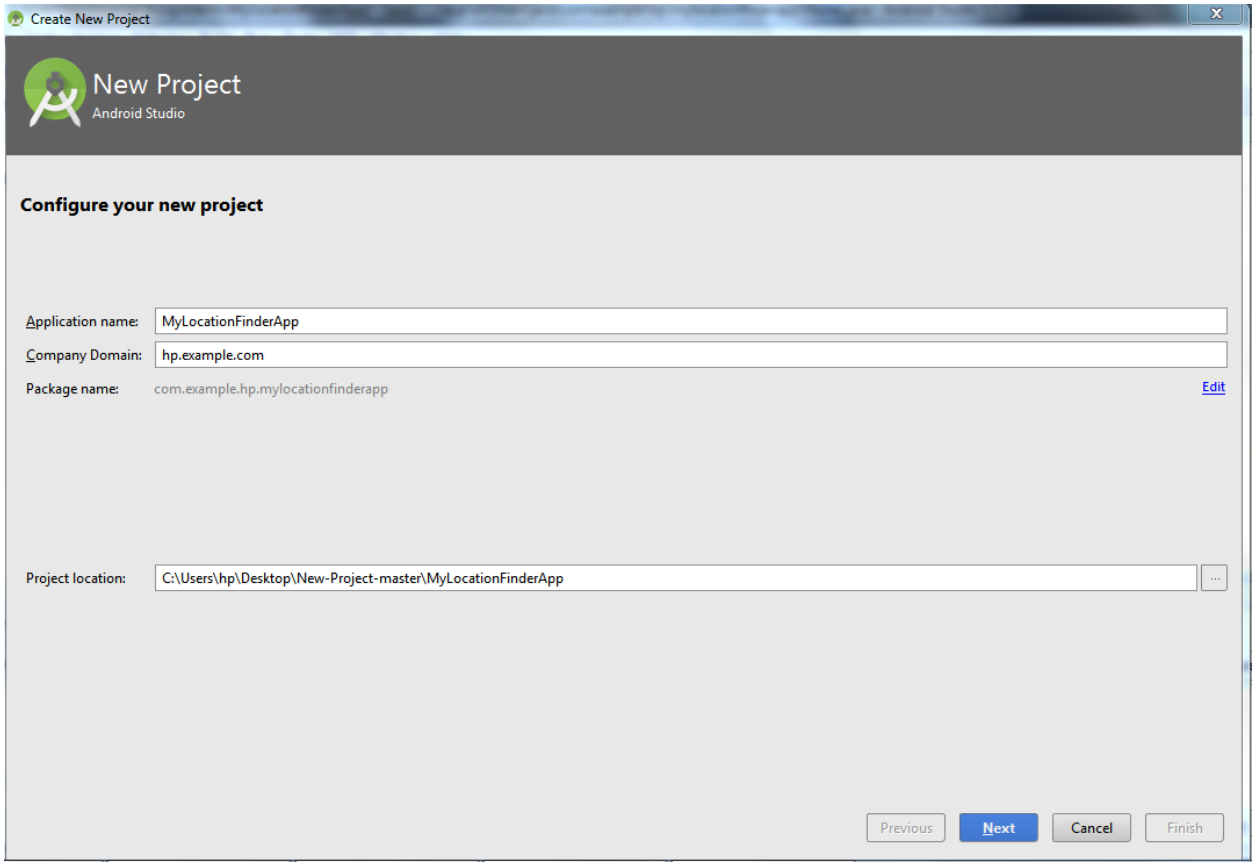
Introduction:

My Location Finder app is an app to display your current location with an option to display a photo by using camera api. On launching the app it opens sign-up page and after successful sign up it redirects to Map Page.

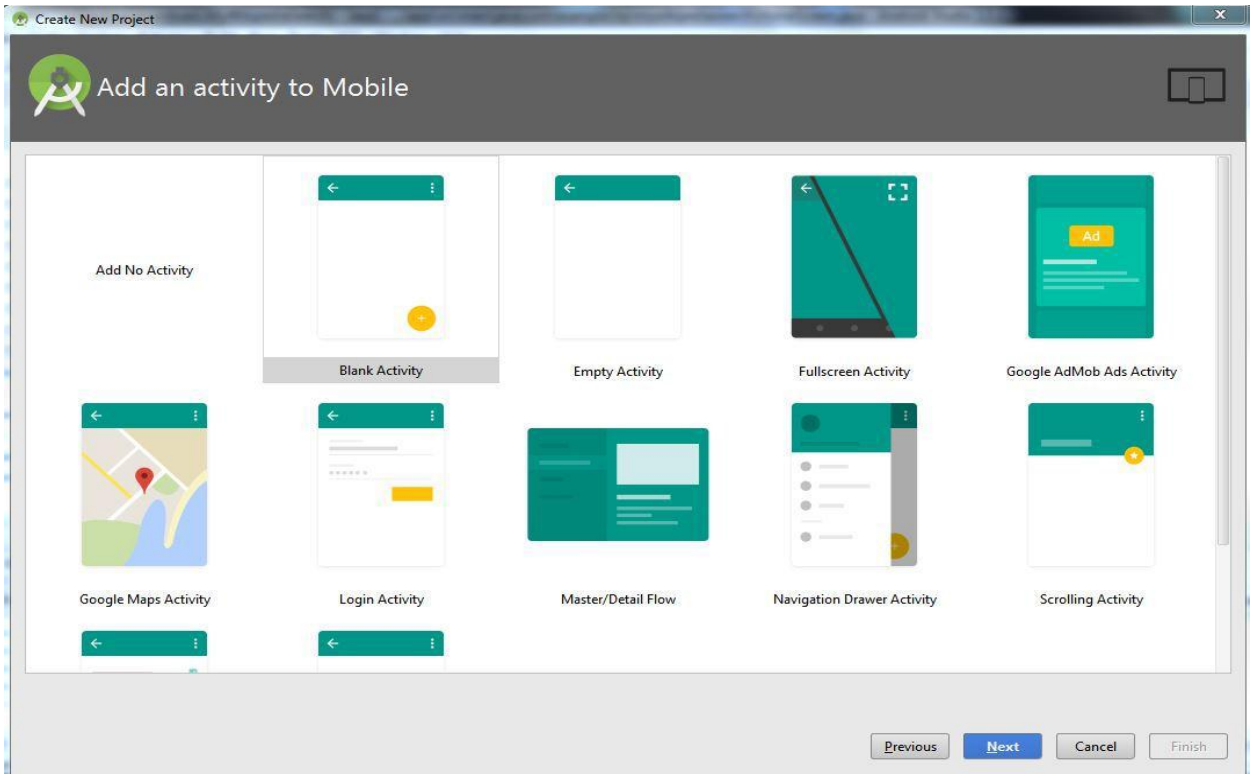
Documentation:

The tool used for the development of this android application is ANDROID STUDIO.

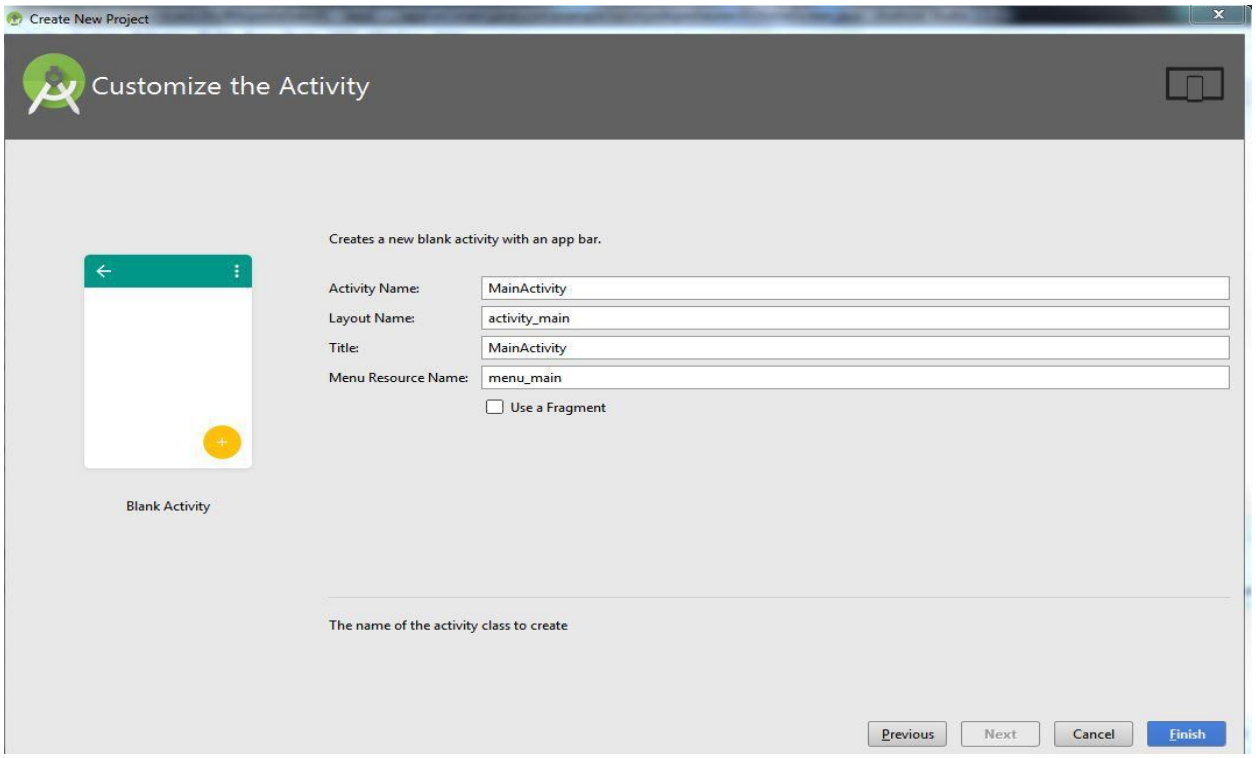
After opening Android Studio, Click on File and select New>New Project. Fill the application Name and choose the Project location.



Click Next and the below screen appears.

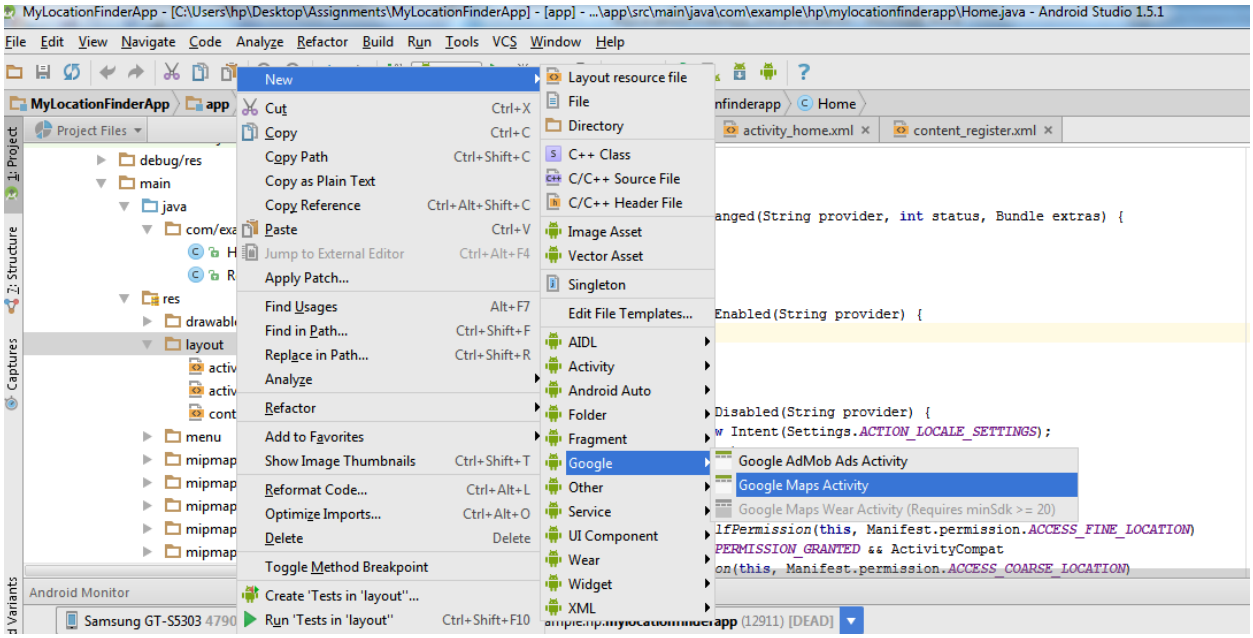


Complete the following details and Click Finish.

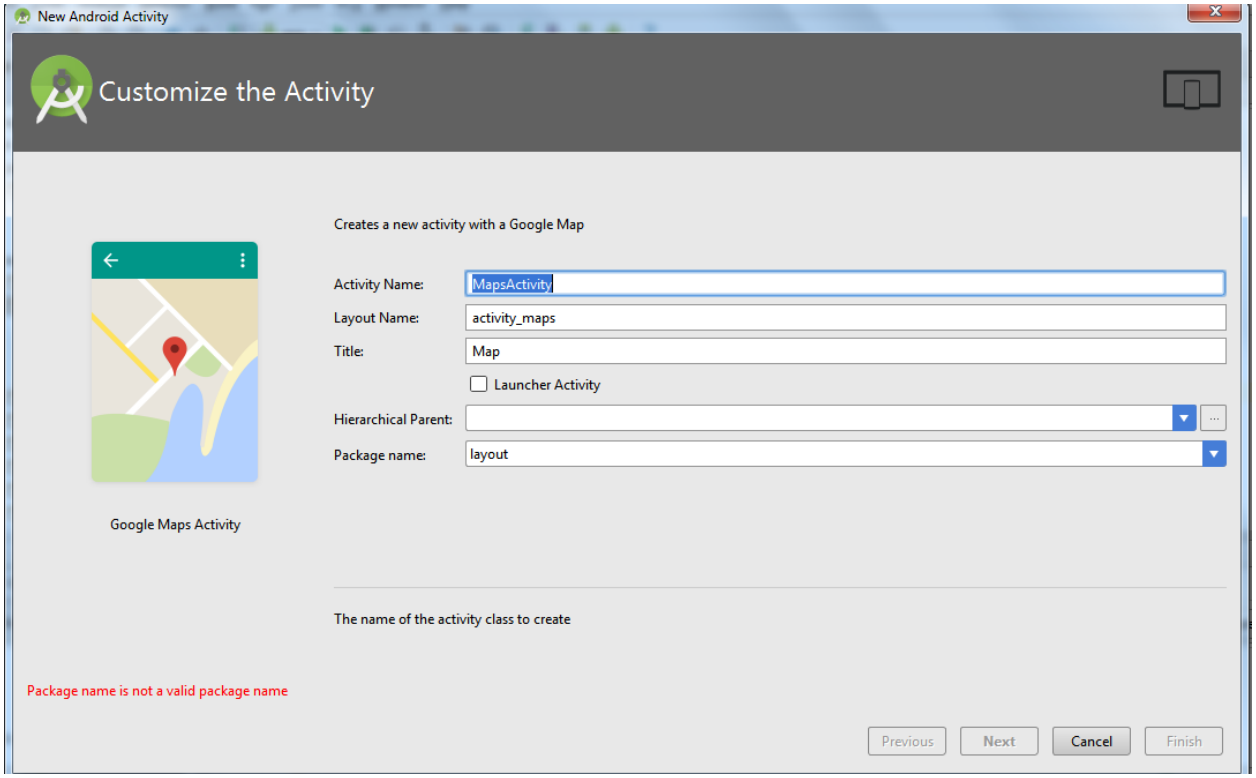


To create a google Maps Activity

Select a google map activity as shown below.



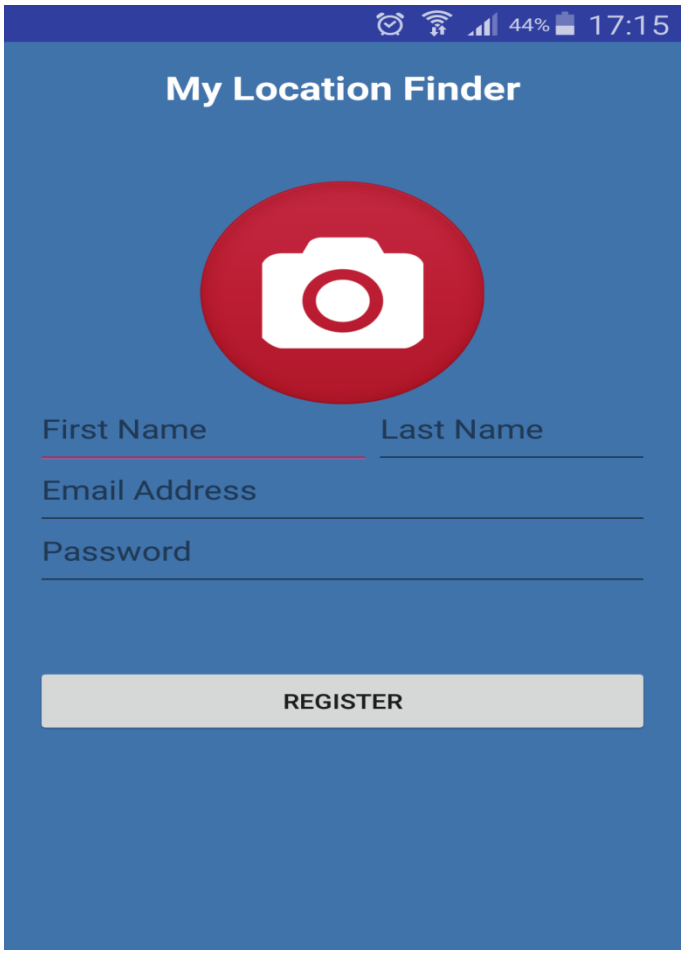
Fill up the following details.



Design:

Home Screen : (Content_Register)

The following screen on successful launch of the application.



This is home screen(Content_Register)

This screen contains a textview which contains the title.

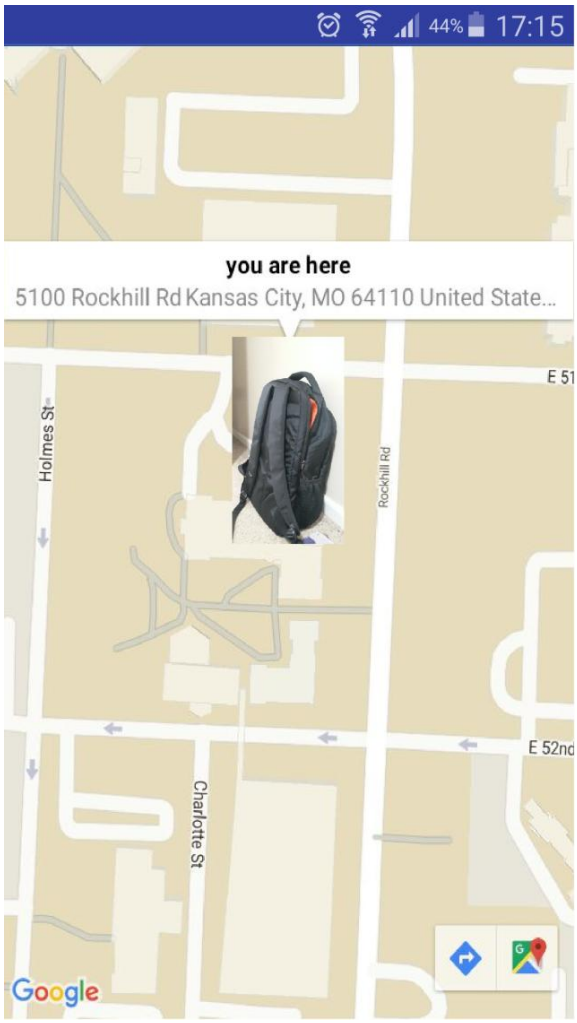
An image view icon with onClick which acts as camera button.

Four Edit text widgets for Signup. One each for First Name, Last name, Email Address and Password.

A button to Register,on being successful redirects to Map page.

Map Screen : (Activity_home)

This screen after being directed from Content_Register Screen



Implementation

The main implementation of the application is done in two java files

1)RegisterActivity

The below screenshot contains declaration of all files

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_register);
    //Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
    //setSupportActionBar(toolbar);

    etFirstName=(EditText) findViewById(R.id.etFirstName);
    etLastName=(EditText) findViewById(R.id.etLastName);
    etEmail=(EditText) findViewById(R.id.etEmail);
    etPassword=(EditText) findViewById(R.id.etPassword);
    btRegister=(Button) findViewById(R.id.btRegister);
    ivCamera=(ImageView) findViewById(R.id.ivCamera);


}
```

This below screenshot contains camera implementation. On clicking Image view screen is redirected to camera.

```
public void onClickRegister(View v){
    ivCamera.setDrawingCacheEnabled(true);
    Bitmap bm = ivCamera.getDrawingCache();
    Intent mapIntent = new Intent(RegisterActivity.this,Home.class);
    mapIntent.putExtra("PROFILEIMG", bm);
    startActivity(mapIntent);
}

public void onClickCamera(View v){
    Intent cameraIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    startActivityForResult(cameraIntent,TAKE_PHOTO_CODE);
}
```

The below piece of code captures the photo and compresses it.

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == TAKE_PHOTO_CODE && resultCode == RESULT_OK) {
        ByteArrayOutputStream stream = new ByteArrayOutputStream();
        Bitmap Takenimg = (Bitmap) data.getExtras().get("data");
        Takenimg.compress(Bitmap.CompressFormat.PNG, 100, stream);
        ivCamera.setImageBitmap(Takenimg);
        Log.d("CameraDemo", "Pic saved");
    }
}
```

Home(Maps Code)

```
public class Home extends FragmentActivity implements OnMapReadyCallback {

    private GoogleMap mMap;
    public Geocoder geocoder;
    RegisterActivity object = new RegisterActivity();
    Bitmap photoMarker=object.photo;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home);
        //getActionBar().setTitle("Map location");
        SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
                .findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);

    }
```

The above screenshot contains declaration of the variables.

```java
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;
    geocoder = new Geocoder(this);
    StringBuilder userAddress = new StringBuilder();

    LocationManager userCurrentLocation = (LocationManager) this
            .getSystemService(Context.LOCATION_SERVICE);
    LocationListener userCurrentLocationListener = new LocationListener() {
        @Override
        public void onLocationChanged(Location location) {
        }
        @Override
        public void onStatusChanged(String provider, int status, Bundle extras) {
        }
        @Override
        public void onProviderEnabled(String provider) {
        }
        @Override
        public void onProviderDisabled(String provider) {
            Intent intent = new Intent(Settings.ACTION_LOCALE_SETTINGS);
            startActivity(intent);
        }
    };
    LatLng userCurrentLocationCorodinates;
    if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION)
            != PackageManager.PERMISSION_GRANTED && ActivityCompat
            .checkSelfPermission(this, Manifest.permission.ACCESS_COARSE_LOCATION)
            != PackageManager.PERMISSION_GRANTED) {
        //show message or ask permissions from the user.
        return;
    }
```

This screen contains a predefined onMapReady method. The above methods can be overridden according to project requirements.

There is an if condition where in checks for the permissions(Manifest Permission) for Location access. If not granted, it returns without displaying.

This screenshot contains address, which is generated by the help of latitude and longitude of the current location.

```
userCurrentLocation.requestLocationUpdates(LocationManager.GPS_PROVIDER,0, 0, userCurrentLocationListener);
userCurrentLocationCorodinates = new LatLng(userCurrentLocation.getLastKnownLocation(LocationManager.GPS_PROVIDER)
        .getLatitude(),userCurrentLocation.getLastKnownLocation(LocationManager.GPS_PROVIDER).getLongitude() );
//Getting the address of the user based on latitude and longitude.
try {
    List<Address> addresses = geocoder.getFromLocation(userCurrentLocation.getLastKnownLocation(LocationManager.GPS_PROVIDER)
                    .getLatitude(),userCurrentLocation.getLastKnownLocation(LocationManager.GPS_PROVIDER).getLongitude(), 1);
    Address address = addresses.get(0);
    userAddress =  new StringBuilder();
    for (int i = 0; i < address.getMaxAddressLineIndex(); i++) {
        userAddress.append(address.getAddressLine(i)).append("\t");
    }
    userAddress.append(address.getCountryName()).append("\t");

}
catch(Exception ex)
{
    ex.printStackTrace();
}

Intent intent = getIntent();
Bitmap IBitmap = (Bitmap) intent.getParcelableExtra("PROFILEIMG");


//ProfileImage.setImageBitmap(IBitmap);

//Setting our image as the marker icon.
mMap.addMarker(new MarkerOptions().position(userCurrentLocationCorodinates)
        .title("My Current Location").snippet(userAddress.toString()))
        .setIcon(BitmapDescriptorFactory.fromBitmap(IBitmap));
//Setting the zoom level of the map.
mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(userCurrentLocationCorodinates, 7));
```

On the map, there is an image which is taken from the pic taken from the Register activity. The same picture is displayed on marker of the current location.

The final output will be: