

**Name: Lakshmana Kumar Mettu**

**Class ID:11**

### **Introduction:**

- This ICP is got to know about usage of API's like google maps and usage of components like sensors, accelerometer sensor in android devices.

### **Programming elements:**

- Android Studio
- Google Maps API

### **Objective:**

- Display the user current location
- Providing line from source to destination on maps using google API
- Using Accelerometer sensor providing readings
- The corresponding code snippets for this ICP is given below.
- My maps activity which is having google API along with key is shown below.

```
// Building the parameters to the web service
String parameters = str_origin + "&" + str_dest + "&" + sensor + "&" + mode + "&key=AIzaSyCEPaRwc2fKch2iLfxjpXfSHfZ8_vH6tXs";

// Output format
String output = "json";

// Building the url to the web service
String url = "https://maps.googleapis.com/maps/api/directions/" + output + "?" + parameters;

return url;
}
```



- Accelerometer activity to obtain the parameters like x,y,z readings shown below.

```
private void getAccelerometer(SensorEvent event) {
    float[] values = event.values;
    // Movement
    float x = values[0];
    float y = values[1];
    float z = values[2];

    float accelerationSquareRoot = (x * x + y * y + z * z)
        / (SensorManager.GRAVITY_EARTH * SensorManager.GRAVITY_EARTH);
    long actualTime = event.timestamp;
    if (accelerationSquareRoot >= 2) //
    {
        if (actualTime - lastUpdate < 200) {
            return;
        }
        lastUpdate = actualTime;
        Toast.makeText(this, "Device was shuffled", Toast.LENGTH_SHORT)
            .show();
        if (color) {
            view.setBackgroundColor(Color.GREEN);
        } else {
            view.setBackgroundColor(Color.RED);
        }
        color = !color;
    }
}

@Override
```

- Main Activity which uses the onclick location method to display user current location and providing the line from source to destination given below.

```
public class MainActivity extends AppCompatActivity {

    Button button_map, button_photo;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        if (ActivityCompat.checkSelfPermission(this, android.Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(this, android.Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
            return;
        }
    }

    public void onPhotoClick(View v) {
        //This code redirects the from main page to the maps page.
        Intent redirect = new Intent(MainActivity.this, CameraActivity.class);
        startActivity(redirect);
    }

    public void onLocationClick(View v) {
        //This code redirects to the photo activity.
        Intent redirect = new Intent(MainActivity.this, MyMapsActivity.class);
        startActivity(redirect);
    }

    public void onSensorClick(View v) {
        Intent redirect = new Intent(MainActivity.this, Accelerometer.class);
        startActivity(redirect);
    }
}
```



- It has additional feature like usage of camera in order to get snaps.

```
import android.os.Bundle;
import android.os.Bundle;
import android.os.Bundle;
import android.os.Bundle;

public class CameraActivity extends AppCompatActivity {

    int TAKE_PHOTO_CODE = 0;
    ImageView userImage ;
    private static final int MY_CAMERA_REQUEST_CODE = 100;

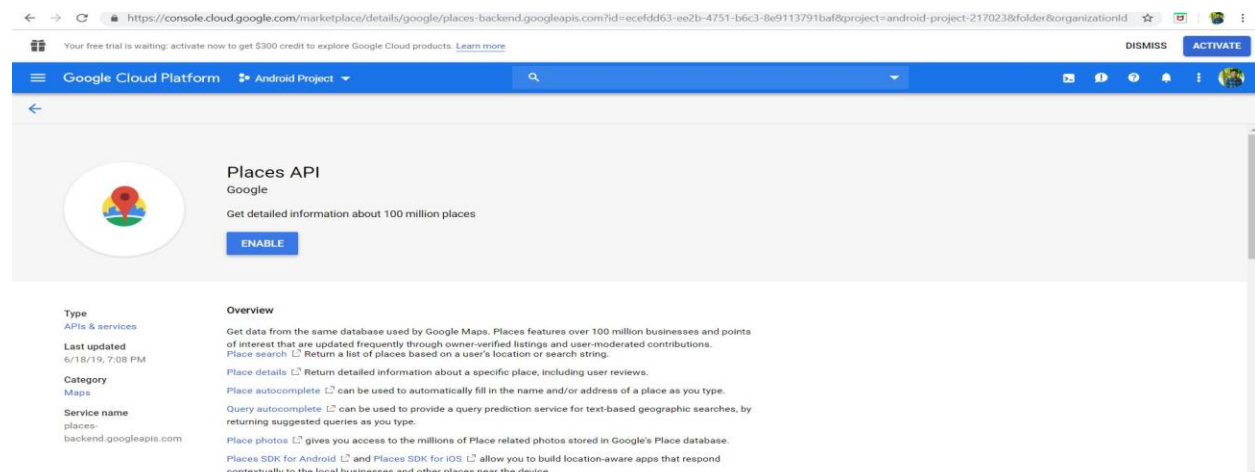
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_camera);
        Button capture = (Button) findViewById(R.id.btn_take_photo);
        userImage = (ImageView) findViewById(R.id.view_photo);
        //Button click eventlistener. Initializes the camera.
        Toast.makeText(this, "In camera activity", Toast.LENGTH_SHORT).show();
        if (checkSelfPermission(android.Manifest.permission.CAMERA) != PackageManager.PERMISSION_GRANTED) {
            requestPermissions(new String[]{android.Manifest.permission.CAMERA}, MY_CAMERA_REQUEST_CODE);
        }
    }

    public void callCamera(View v) {
        //Toast.makeText(getApplicationContext(), "on take photo click", Toast.LENGTH_SHORT).show();
        Intent cameraIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        if (cameraIntent.resolveActivity(getPackageManager()) != null) {
            startActivityForResult(cameraIntent, TAKE_PHOTO_CODE);
        }
    }

    //If the photo is captured then set the image view to the photo captured.
    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
    }
}
```

CameraActivity > onActivityResult()

- Google maps API obtaining method




← → ↺ https://console.cloud.google.com/marketplace/details/google/places-backend.googleapis.com?id=ecfd63-ee2b-4751-b6c3-8e9113791baf&project=android-project-217023&folder&organizationId ☆ | | |

Your free trial is waiting: activate now to get \$300 credit to explore Google Cloud products. [Learn more](#)

DISMISS ACTIVATE

Google Cloud Platform Android Project

←

 **Places API**  
Google  
Get detailed information about 100 million places  
[ENABLE](#)

Type	Overview
APIs & services	Get data from the same database used by Google Maps. Places features over 100 million businesses and points of interest that are updated frequently through owner-verified listings and user-moderated contributions.
Last updated 6/18/19, 7:08 PM	Place search ↗ Return a list of places based on a user's location or search string.
Category Maps	Place details ↗ Return detailed information about a specific place, including user reviews.
Service name places-backend.googleapis.com	Place autocomplete ↗ can be used to automatically fill in the name and/or address of a place as you type.
	Query autocomplete ↗ can be used to provide a query prediction service for text-based geographic searches, by returning suggested queries as you type.
	Place photos ↗ gives you access to the millions of Place related photos stored in Google's Place database.
	Places SDK for Android ↗ and Places SDK for iOS ↗ allow you to build location-aware apps that respond contextually to the local businesses and other places near the device.

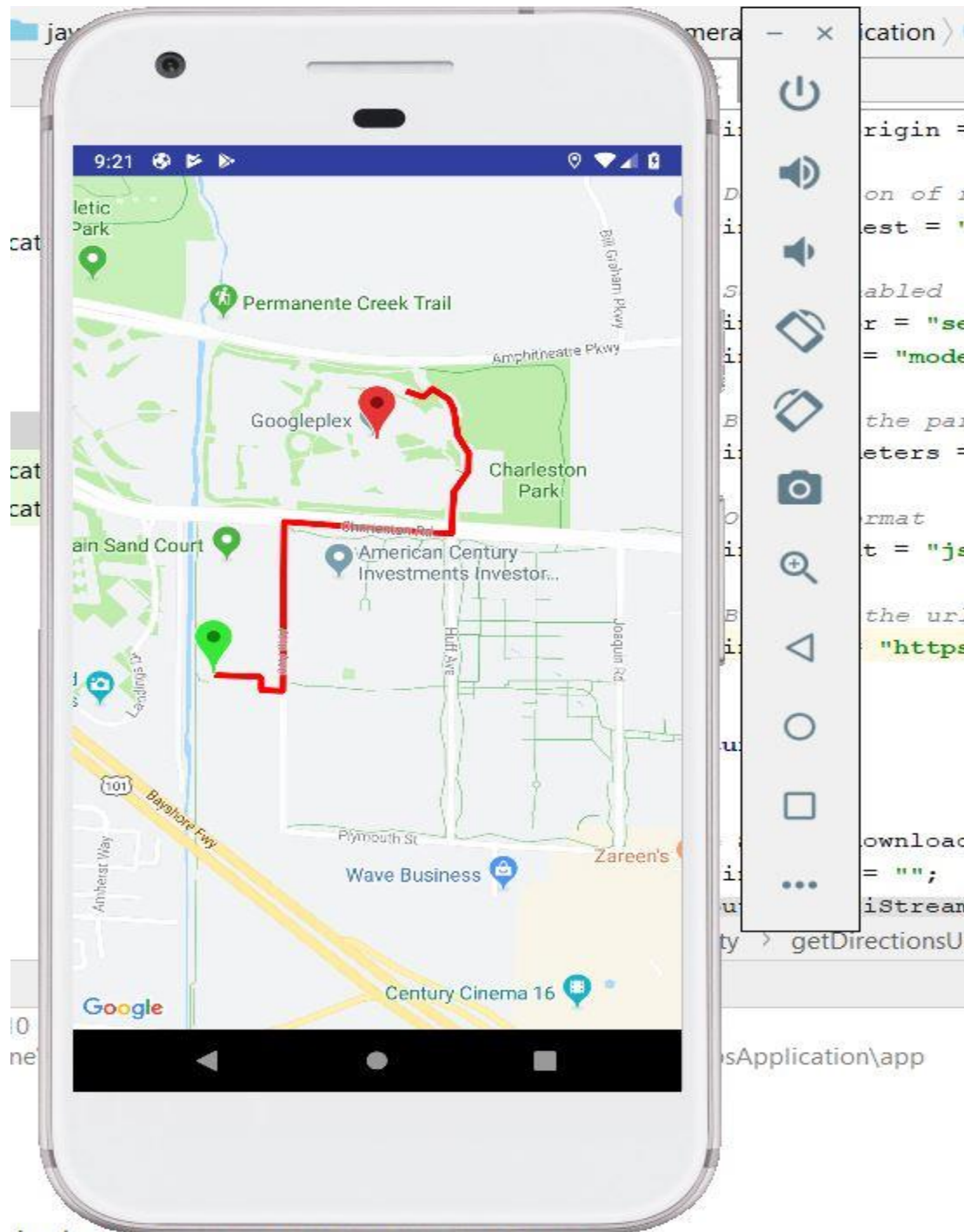


## Output:

- Displaying current location of user with marker



- Providing line from required point to location



has been replaced with Implementation and api.



- Displaying accelerometer sensor readings like x,y,z readings

### CameraMapsApplication

#### Accelerometer Readings

X: 0.10368347

Y: 5.537323

Z: 8.268463

### Conclusion:

- Hence learnt how to use API's and sensors in android devices and developed an application using android studio.

