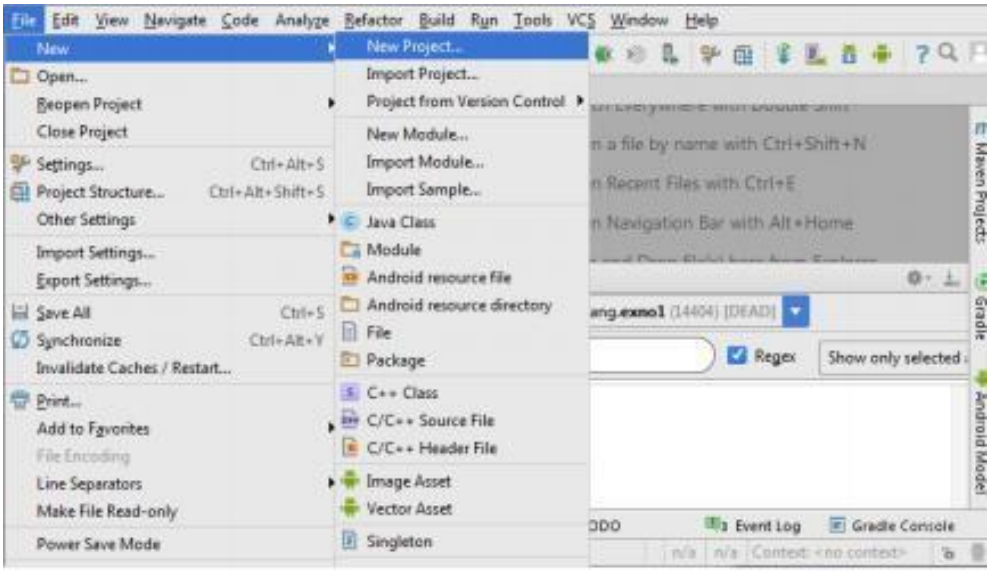
Experiment No. 01

Name-Rebecca Dias Roll No-19 BECMPN A PID-182027 Batch- 2

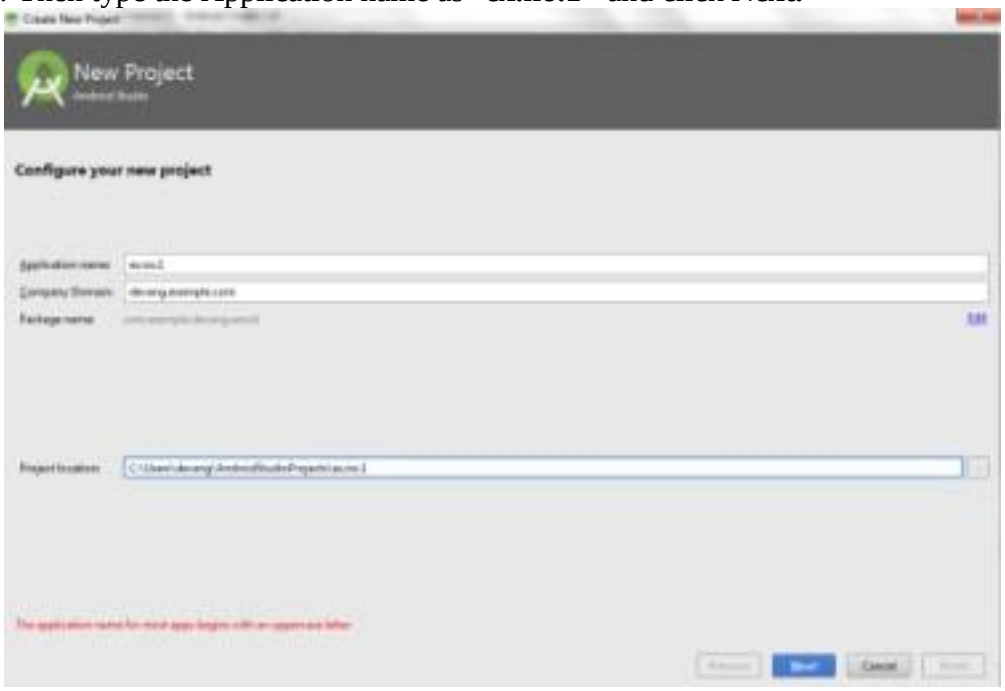
Aim: Develop an application that uses GUI components.

Theory:

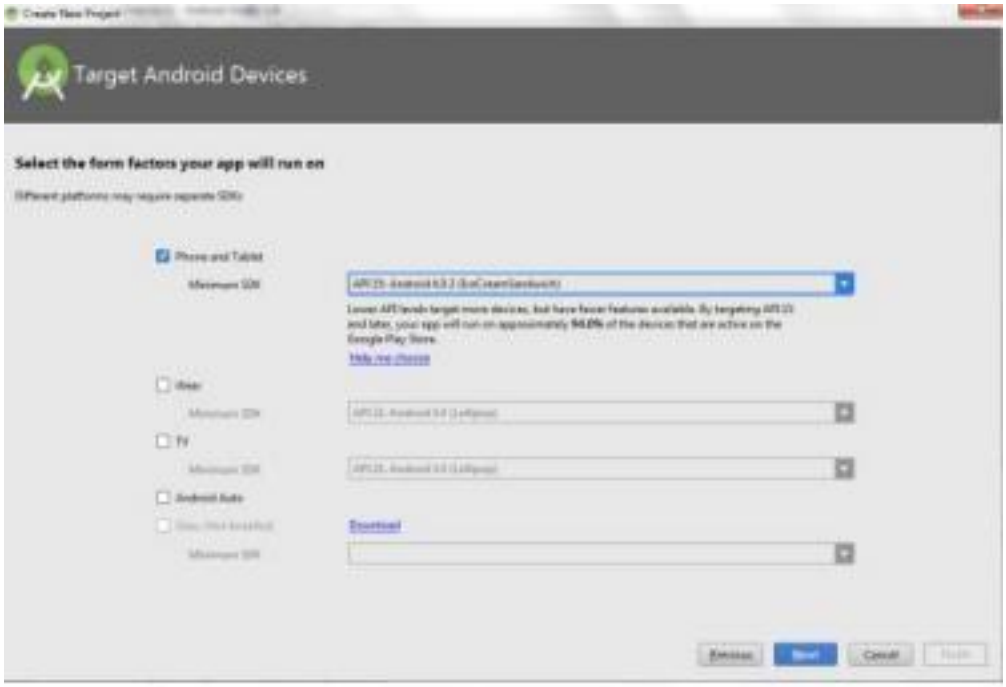
1. Open android studio and select new android project
   File -> New -> New project



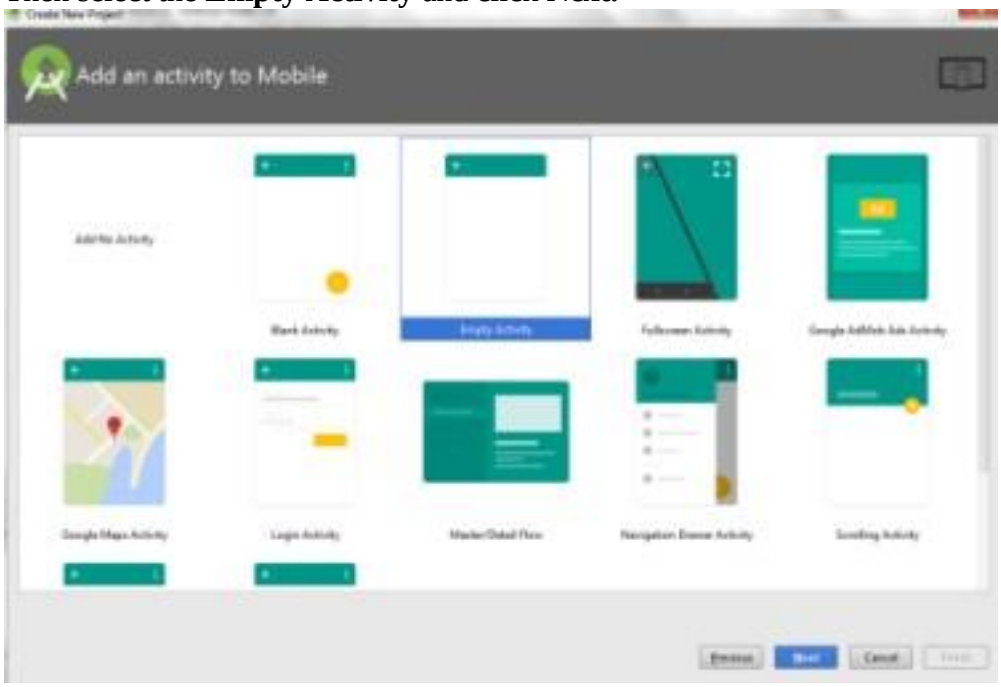
2. Then type the Application name as "**ex.no.1"** and click **Next.**



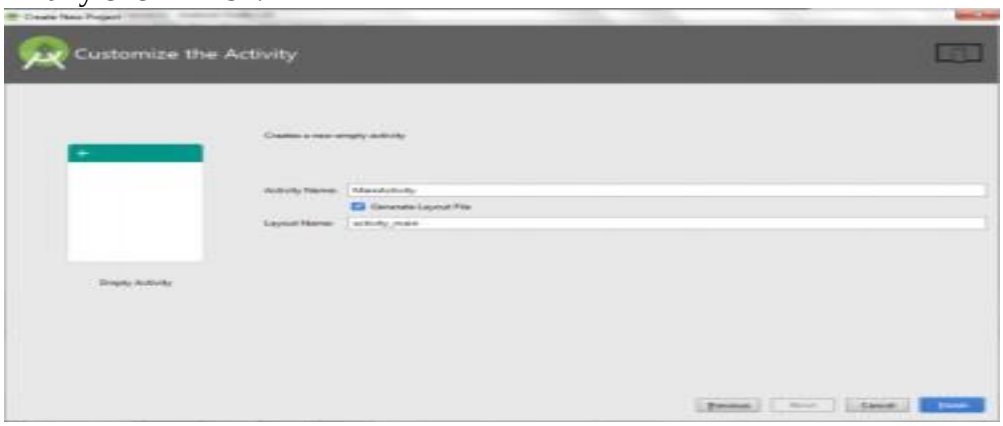
3. Then select the **Minimum SDK** as shown below and click **Next**.



4. Then select the **Empty Activity** and click **Next.**



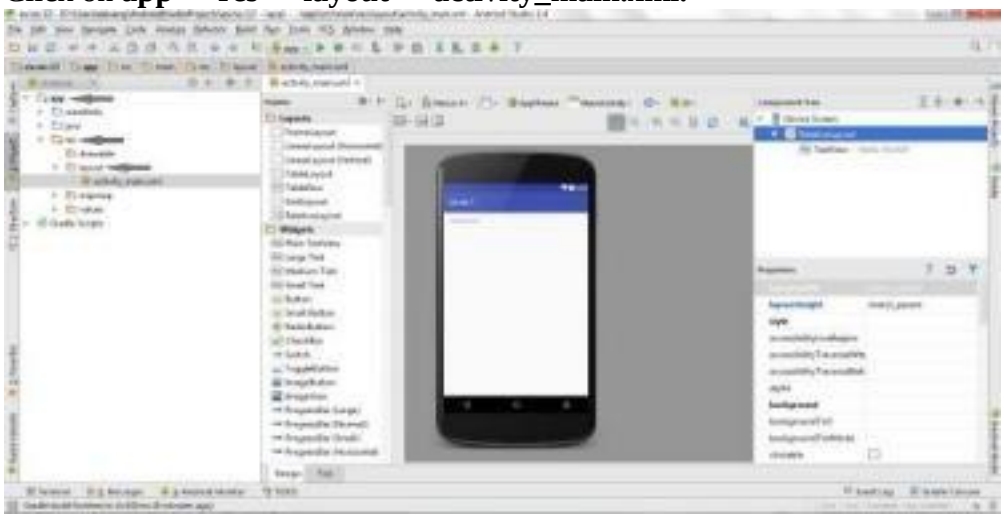
5. Finally click F**inish**.



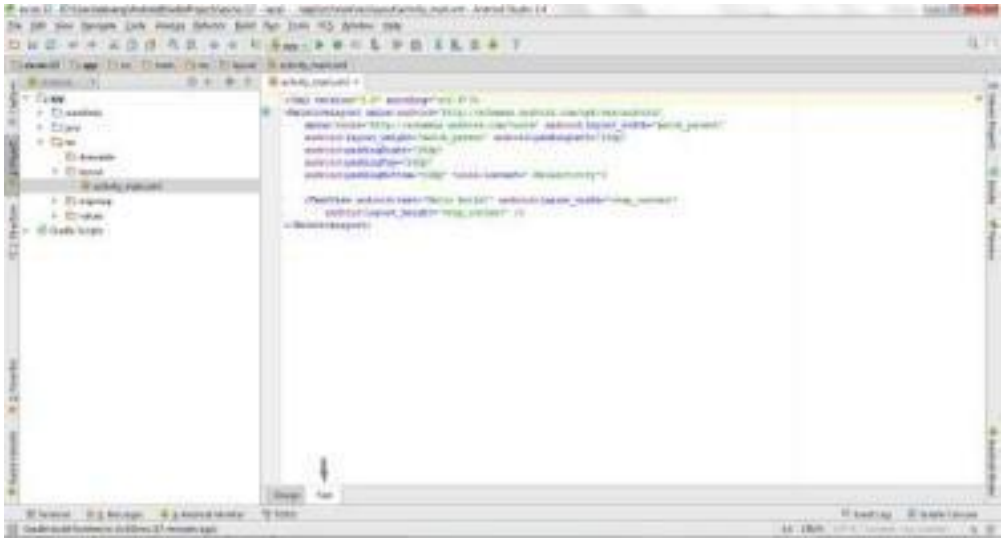
6. It will take some time to build and load the project.
7. After completion it will look as given below.

Designing layout for the Android Application:

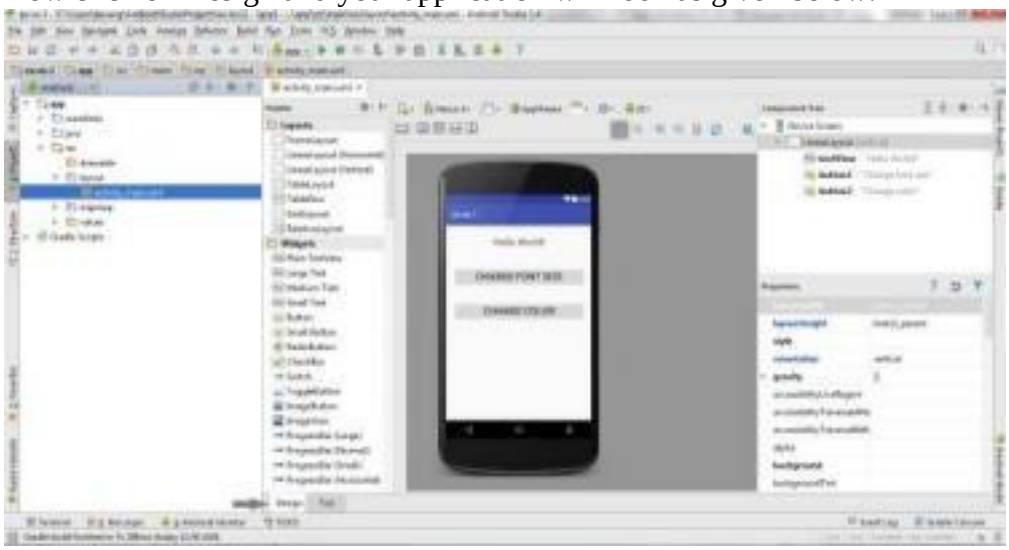
1. Click on **app -> res -> layout -> activity_main.xml.**



2. Now click on **Text** as shown below.



3. Now click on Design and your application will look as given below.



4. So now the designing part is completed.

Implementation:

JAVA:

```
public void updateButtonOnCLickCurrent(View
  view) { locationListener = new
  LocationListener() {
    @Override
    public void onLocationChanged(@NonNull Location
      location) { try {
        LatLng latLng = new LatLng(location.getLatitude(),
        location.getLongitude()); mMap.clear();
        mMap.addMarker(new MarkerOptions().position(latLng).title("Current Position"));
```

```
                mMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));
                databaseReference.child("latitude").push().setValue(location.getLatitude());
                databaseReference.child("longitude").push().setValue(location.getLongitude());
                //Send
                notification
                notifyNow();
            }
            catch(Exception e){
                e.printStackTrace()
                ;}
        }
          @Override
          public void onStatusChanged(String s, int i, Bundle
          bundle) {} @Override
          public void onProviderEnabled(String
          s) {} @Override
          public void onProviderDisabled(String s) {}
      };
      locationManager = (LocationManager) getSystemService(LOCATION_SERVICE);
      if (ActivityCompat.checkSelfPermission(this,
      Manifest.permission.ACCESS_FINE_LOCATION)
!= PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
          return;
      }
      try {
          locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER,
MIN_TIME, MIN_DIST, locationListener);
          locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER ,
MIN_TIME, MIN_DIST, locationListener);
      }
      catch (Exception e){
          e.printStackTrace(
          );
      }
  }
```

XML:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:map="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=".MapsActivity">
        <Button
           android:id="@+id/button2"
           android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:layout_weight="0.5"
        android:onClick="updateButtonOnCLickCurr
        ent" android:text="@string/current" />
</RelativeLayout>
```

Output:

Figure 1: Example of Button

Conclusion:

From this experiment, we learnt how to create GUI components in android. We used android studio where we created an empty activity, in the xml activity, we created a Button with onClick functionality. The on click function was added to the MainActivity.java file to execute its function. The button was used to calculate the current location of the user.