# MISRIMAL NAVAJEE MUNOTH JAIN ENGINEERING COLLEGE

(Managed By Tamil Nadu Educational and Medical Trust)
Thoraipakkam, Chennai – 600097.

## DEPARTMENT
## OF
## COMPUTER SCIENCE AND ENGINEERING

# MOBILE APPLICATION DEVELOPMENT LABORATORY

# RECORD

**NAME :**

**REGISTER NUMBER :**

**YEAR III**

**SEMESTER VI**

## MISRIMAL NAVAJEE MUNOTH JAIN ENGINEERING COLLEGE

(Managed By Tamil Nadu Educational and Medical Trust)
Thoraipakkam, Chennai – 600097.

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Register Number

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **3** | **1** | **1** | **6** | **1** | **7** | **1** | **0** | **4** | | | |

# BONAFIDE CERTIFICATE

This is to certify that this is a bonafide record of work done by ______________________________________________of III Year B.E Computer Science and Engineering in the CS-8662 Mobile Application Development Laboratory during the Academic year 2019-2020.

**Staff In-Charge** **Head of the Department**

Submitted for the University Practical Examination held on__________________

**Internal Examiner** **External Examiner**

## MISRIMAL NAVAJEE MUNOTH JAIN ENGINEERING COLLEGE, CHENNAI – 97

## DEPARTMENT OFCOMPUTER SCIENCE AND ENGINEERING

- **VISION**

  Producing competent Computer Engineers with a strong background in the latest trends and technology to achieve academic excellence and to become pioneer in software and hardware products with an ethical approach to serve the society.

- **MISSION**

  To provide quality education in Computer Science and Engineering with the state of the art facilities. To provide the learning audience that helps the students to enhance problem solving skills and to inculcate in them the habit of continuous learning in their domain of interest. To serve the society by providing insight solutions to the real world problems by employing the latest trends of computing technology with strict adherence to professional and ethical responsibilities.

# INDEX

## BASIC LAYOUT IN ANDROID

**Ex. No.: 1 (a)**

**Date:**

**AIM:**

To develop an application to understand the basic views needed to create a layout in Android.

**ALGORITHM:**

- Create a new project in androidstudio.
- Add a ImageViews and a TextViews to the application in the activity_main.xmlfile.
- Run the project and test it in a mobiledevice.

**PROGRAM:**

- **MainAcivity.java**

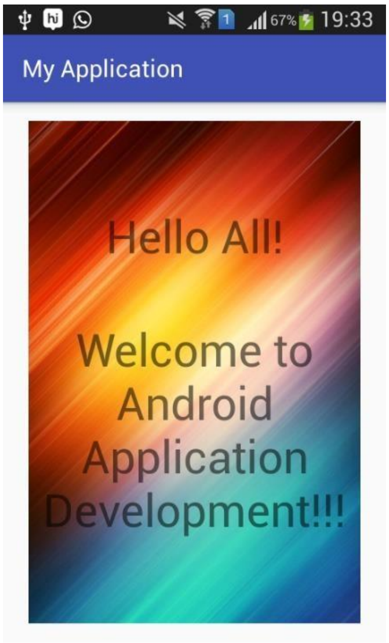
```
package com.example.user.myapplication;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
public class MainActivity extends AppCompatActivity {
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);


      setContentView(R.layout.activity_main);
   }
}
```

- **activity_main.xml**

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

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools"
   android:id="@+id/activity_main"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:paddingBottom="@dimen/activity_vertical_margin"
   android:paddingLeft="@dimen/activity_horizontal_margin"
   android:paddingRight="@dimen/activity_horizontal_margin"
   android:paddingTop="@dimen/activity_vertical_margin"
   tools:context="com.example.user.myapplication.MainActivity">
```

```
  <ImageView
    android:id="@+id/back"
    android:src="@drawable/background"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />

    <TextView android:id="@+id/hello"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="45sp"
    android:textAlignment="center"
    android:text="Hello All!"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_marginTop="70dp"/>

  <TextView
    android:id="@+id/welcome"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="45sp"
    android:textAlignment="center"
    android:text="Welcome to Android Application Development!!!"
    android:layout_alignParentBottom="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_marginBottom="70dp" />


</RelativeLayout>
```

**OUTPUT:**



Hello All!

Welcome to Android Application Development!!!

**RESULT:**

Thus the basic layout in android was successfully developed.

## GUI COMPONENTS, FONTS AND COLORS

**Ex. No.: 1 (b)**

**Date:**

**AIM:**

To develop an application that uses GUI components, font and colors.

**ALGORITHM:**

- Create a new project in androidstudio.
- Add a button and a textview to the application in the activity_main.xmlfile.
- Declare the button and TextView variables and initialize them to the correspondingviews.
- Set up an onClickListener to the button which changes the color of the text in thetextView.
- Run the project and test it in a mobiledevice.

**PROGRAM:**

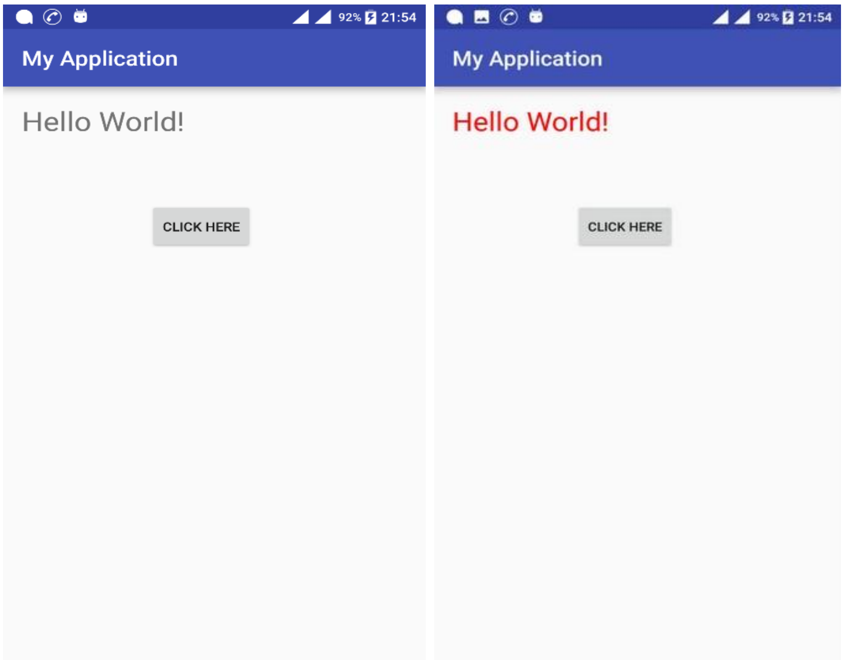
- **MainAcivity.java**

```
package com.example.harishma.colorchange;

import android.graphics.Color;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
private TextView t;
    private Button b;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        b = (Button)findViewById(R.id.button);
        t = (TextView) findViewById(R.id.textView);
        b.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                t.setTextColor(Color.RED);
            } });
            }}
```

- **activity_main.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.harishma.colorchange.MainActivity">


    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        android:textSize="30sp"
        android:id="@+id/textView" />
    <Button
        android:text="CLICK HERE"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="108dp"
        android:layout_marginTop="62dp"
        android:id="@+id/button"
        android:layout_below="@+id/textView"
        android:layout_alignParentStart="true" />
</RelativeLayout>
```

**OUTPUT:**



**RESULT:**

Thus the GUI components, fonts and colors were successfully developed.

## LAYOUT MANAGERS AND EVENT LISTENERS

**Ex. No.: 2**

**Date:**

**AIM:**

To develop an application that uses layout manger and event listeners.

**ALGORITHM:**

- Create a new project in androidstudio.
- Add a button and a textview to the application in the activity_main.xmlfile.
- Declare the button and TextView variables and initialize them to the correspondingviews.
- Set up an onClickListener to the button which changes the size of the text in thetextView.
- Run the project and test it in a mobiledevice.

**PROGRAM:**

- **MainActivity.java**

```
package com.example.geekymad.experiment2;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.TextView;


public class MainActivity extends AppCompatActivity {

   Button b1,b2,b3;

@Override

   protected void onCreate(Bundle savedInstanceState) {

      super.onCreate(savedInstanceState);

      setContentView(R.layout.activity_main);

      b1=(Button)findViewById(R.id.button1);

      b2=(Button)findViewById(R.id.button2);

      b3 = (Button) findViewById(R.id.resetButton);

      b1.setOnClickListener(new View.OnClickListener() {

         @Override

         public void onClick(View v) {
             TextView txtView = (TextView) findViewById(R.id.helloTextView);

             txtView.setTextSize(15);

         }

     });

      b2.setOnClickListener(new View.OnClickListener() {

         @Override

         public void onClick(View v) {

             TextView txtView = (TextView) findViewById(R.id.helloTextView);

             txtView.setTextSize(55);

         }

      });

      b3.setOnClickListener(new View.OnClickListener() {

         @Override

         public void onClick(View v) {

                     TextView hello = (TextView) findViewById(R.id.helloTextView);
   hello.setTextSize(25);

         }

      });

   }

}
```

- **activity_main.xml**

```
<?xml version="1.0"encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <ImageView
        android:id="@+id/image"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:contentDescription="@string/sample_image"
        android:src="@drawable/image" />

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/event_handling"
        android:textSize="30sp" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="20dp"
        android:text="@string/mnm_jec"
        android:textColor="#000"
        android:textSize="30sp" />

    <Button
        android:id="@+id/resetButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/reset_to_default_size" />

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/small_font" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/large_font" />

    <TextView
        android:id="@+id/helloTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="30dp"
        android:text="Hello World!" />

</LinearLayout>
```

- **colors.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#7cb342</color>
    <color name="colorPrimaryDark">#558b2f</color>
    <color name="colorAccent">#FF4081</color>
</resources>
```

- **dimens.xml**

```
<resources>
    <!-- Default screen margins, per the Android Design guidelines. -->
    <dimen name="activity_horizontal_margin">16dp</dimen>
    <dimen name="activity_vertical_margin">16dp</dimen>
```

```
</resources>
```

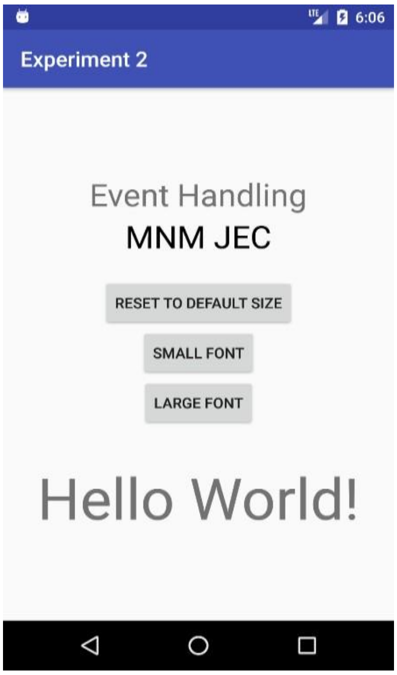
- **strings.xml**

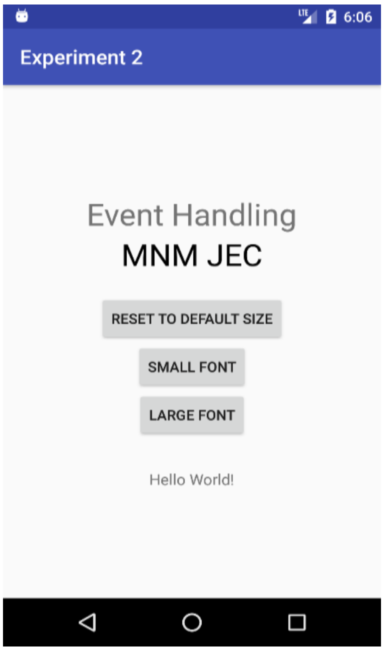
```
<resources>
    <string name="app_name">Experiment 2</string
    <string name="event_handling">Event Handling</string>
    <string name="mnm_jec">MNM JEC</string>
    <string name="reset_to_default_size">Reset to default size</string>
    <string name="small_font">Smallfont</string>
    <string name="large_font">LargeFont</string>
    <string name="sample_image">image</string>
</resources>
```

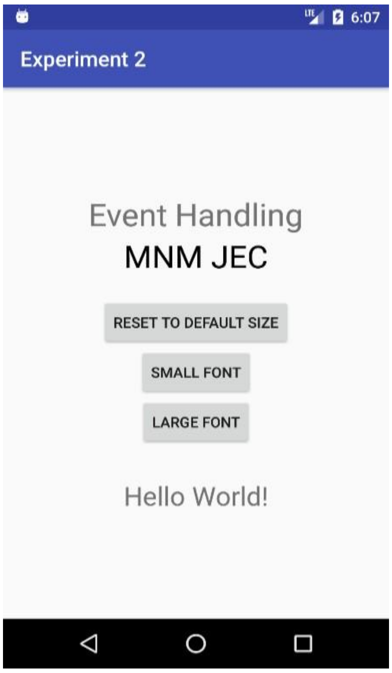
- **styles.xml**

```
<resources>
    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
    </style></resources>
```

**OUTPUT:**







**RESULT:**

Thus the layout manager and event listerner was successfully developed.

## BASIC GRAPHICAL PRIMITIVES

**Ex. No: 3**

**Date:**

**AIM:**

To write an application to draw basic graphical primitives on the screen.

**ALGORITHM:**

- Create a new project in androidstudio.
- Create a canvas on which different shapes can bedrawn.
- Declare a paint object which can be used to change the color of the shapedrawn.
- Draw a circle and a rectangle and paint it in differentcolors.
- Run the project and test it in a mobiledevice.

**PROGRAM:**

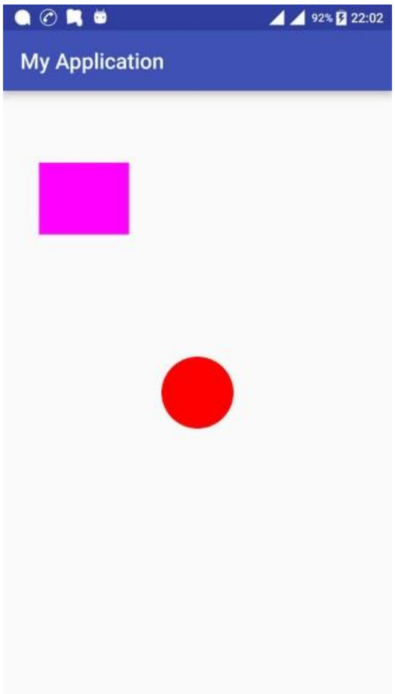
- **MainActivity.java**

```
package com.example.admin.graphics;

import android.support.v7.app.AppCompatActivity; import
android.graphics.Color;
import android.graphics.Paint; import
android.os.Bundle; import
android.view.View; import
android.content.Context; import
android.graphics.Canvas;

public class MainActivity extends AppCompatActivity {
    SampleViewDraw mSampleViewDraw;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState); mSampleViewDraw = new
        SampleViewDraw(this); setContentView(mSampleViewDraw);
    }
    public class SampleViewDraw extends View{ public
        SampleViewDraw(Context context) {
            super(context);
        }
        @Override
        protected void onDraw(Canvas canvas) {
            super.onDraw(canvas);
            Paint paint = new Paint();
            paint.setStyle(Paint.Style.FILL);
            paint.setColor(Color.RED); int x =
            getWidth();
            int y = getHeight();
            canvas.drawCircle(x / 2, y / 2, 100, paint);
            paint.setColor(Color.MAGENTA); canvas.drawRect(100,
            200, 350, 400, paint);}}}
```

- **activity_main.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.admin.graphics.MainActivity">
</RelativeLayout>
```

**OUTPUT:**



**RESULT:**

Thus the basic graphical primitives was successfully developed.

## CREATING A DATABASE IN ANDROID STUDIO

**Ex. No.: 4**

**Date:**

**AIM:**

To develop an application that makes use of database.

**ALGORITHM:**

- Create a new project in AndroidStudio.
- Design an initial layout to add and Display data
- Display the data available in thedatabase.
- Insert the data into database.
- Update the UI.
- Run the application in a mobiledevice.

**PROGRAM:**

- **Mainactivity.java**

```
package com.example.db;

import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;

import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {

    ArrayList<String> listData;

    @RequiresApi(api = Build.VERSION_CODES.P)
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);


        final DB db = new DB(getApplicationContext());

        ListView listView = findViewById(R.id.list);
        final EditText name = findViewById(R.id.name);

        Button save = findViewById(R.id.save);

        listData = db.getData();

        final ArrayAdapter adapter = new
ArrayAdapter(MainActivity.this,android.R.layout.simple_list_item_1,listData);

        listView.setAdapter(adapter);


        save.setOnClickListener(new View.OnClickListener() {
          @Override
          public void onClick(View v) {

            String value = name.getText().toString();

            if(db.setData(value)){

              Toast.makeText(MainActivity.this,"Success !!!",Toast.LENGTH_LONG).show();

              listData.clear();

              listData.addAll(db.getData());

              adapter.notifyDataSetChanged();

              name.setText("");
            }
            else {


              Toast.makeText(MainActivity.this,"Failure !!!",Toast.LENGTH_LONG).show();
              name.setText("");
            }
          }
        });

    }
}
```

- **Keys.java**

```
public class keys {

  public static final String NAME = "DETAILS";
  public static final String DBNAME = "DETAILS.DB";
  public static final String CREATE = "CREATE TABLE "+NAME+"(NAME VARCHAR(15))";
  public static final String DROP = "DROP IF EXISTS"+NAME;
}
```

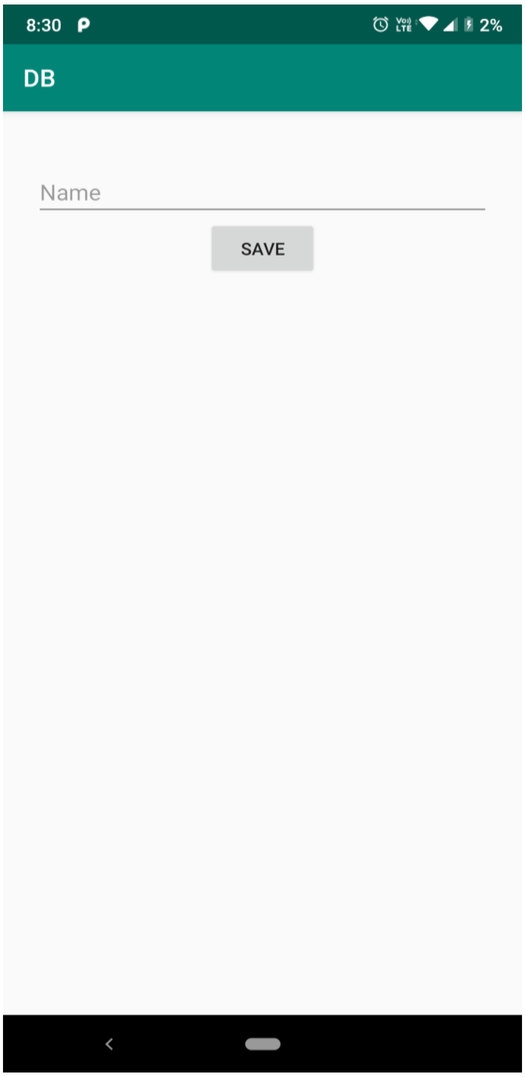
- DB.java

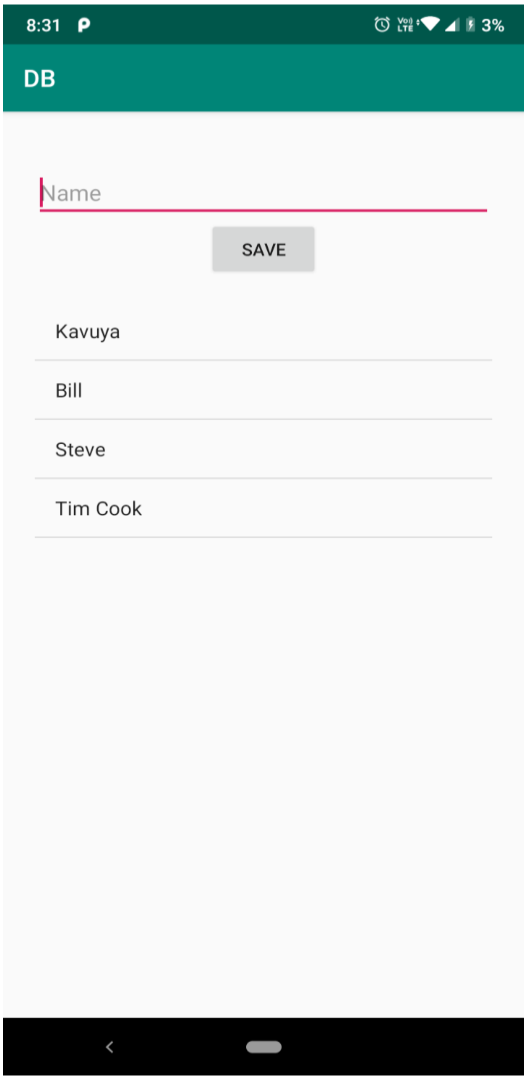
```
package com.example.db;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.os.Build;
import android.widget.ArrayAdapter;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.annotation.RequiresApi;

import java.util.ArrayList;
import java.util.List;

public class DB extends SQLiteOpenHelper {

    @RequiresApi(api = Build.VERSION_CODES.P)
    public DB(@Nullable Context context) {
        super(context,keys.DBNAME,null,1);

    }

    @Override
    public void onCreate(SQLiteDatabase db) {


        db.execSQL(keys.CREATE);

        System.out.println("SucessFully !!!");

    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

        db.execSQL(keys.DROP);

        onCreate(db);
    }

    public Boolean setData(String name){

        ContentValues values = new ContentValues();
        values.put("NAME",name);

        SQLiteDatabase db = this.getWritableDatabase();

        if (db.insert(keys.NAME,null,values)>0){

            db.close();
            return true;
        }
        else{

            db.close();

            return false;
        }



    }

    public ArrayList<String> getData(){


        SQLiteDatabase db = this.getReadableDatabase();
        String query = "SELECT * FROM "+keys.NAME;
        Cursor cursor =  db.rawQuery(query,null);

        ArrayList<String> data = new ArrayList<>();

        while (cursor.moveToNext()){

             data.add(cursor.getString(cursor.getColumnIndex("NAME")));
        }

        return data;
    }
}
```

- Activity_main.java

```
<?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"
    android:padding="25dp">

<EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/name"
        android:hint="Name"
        android:layout_marginTop="20dp"
        />
<Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Save"
        android:id="@+id/save"
        android:layout_below="@id/name"
        android:layout_centerHorizontal="true"/>

<ListView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/list"
    android:layout_below="@id/save"
    android:layout_marginTop="20dp"/>

</RelativeLayout>
```

- **Output**





```
Kavuya
Bill
Steve
Tim Cook
```

**RESULT:**

Thus the mobile application was successfully developed.

## NOTIFICATION MANAGER

**Ex. No.: 5**

**Date:**

**AIM:**

To develop an application that makes use of Notificatios.

**ALGORITHM:**

- Create a new project in AndroidStudio.
- Create a notification and display it on screen
- Run the application in a mobiledevice.

**PROGRAM:**

- **Mainactivity.java**

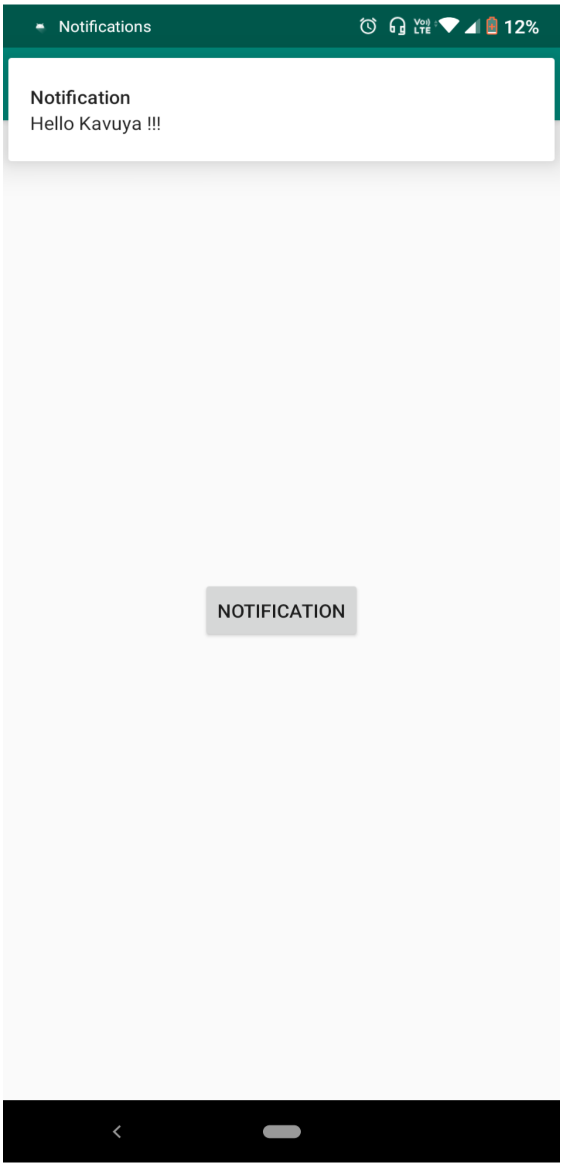
```
package com.example.notifications;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompat;
import androidx.core.app.NotificationManagerCompat;

import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);


        final Button Notification = findViewById(R.id.noti);

        Notification.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {


                NotificationCompat.Builder builder = new
NotificationCompat.Builder(MainActivity.this,"NOTID15")
                        .setContentTitle("Notification")
                        .setContentText("Hello Kavuya !!!")
                        .setSmallIcon(R.drawable.ic_launcher_foreground)
                        .setPriority(NotificationCompat.PRIORITY_HIGH);
                NotificationManager notificationManager = getSystemService(NotificationManager.class);


                if (android.os.Build.VERSION.SDK_INT >= android.os.Build.VERSION_CODES.O) {

                    NotificationChannel channel = new NotificationChannel("NOTID15","General",
NotificationManager.IMPORTANCE_HIGH);

                    notificationManager.createNotificationChannel(channel);
                }

                notificationManager.notify(15,builder.build());
            }
        });
    }
}
```

-

- **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">


<Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Notification"
        android:id="@+id/noti"
        android:layout_centerInParent="true"/>

</RelativeLayout>
```

- **Output**



Notification
Hello Kavuya !!!

**RESULT:**
Thus the notification manager is Deployed Successfully.

## Mutli-Threading

**Ex. No.: 6**

**Date:**

**AIM:**

To develop an application that makes use of threads.

**ALGORITHM:**

- Create a new project in androidstudio.
- Edit the activity_main.xml file to create thelayout.
- Edit the files necessary to implement the android application and run theprogram.

**PROGRAM:**

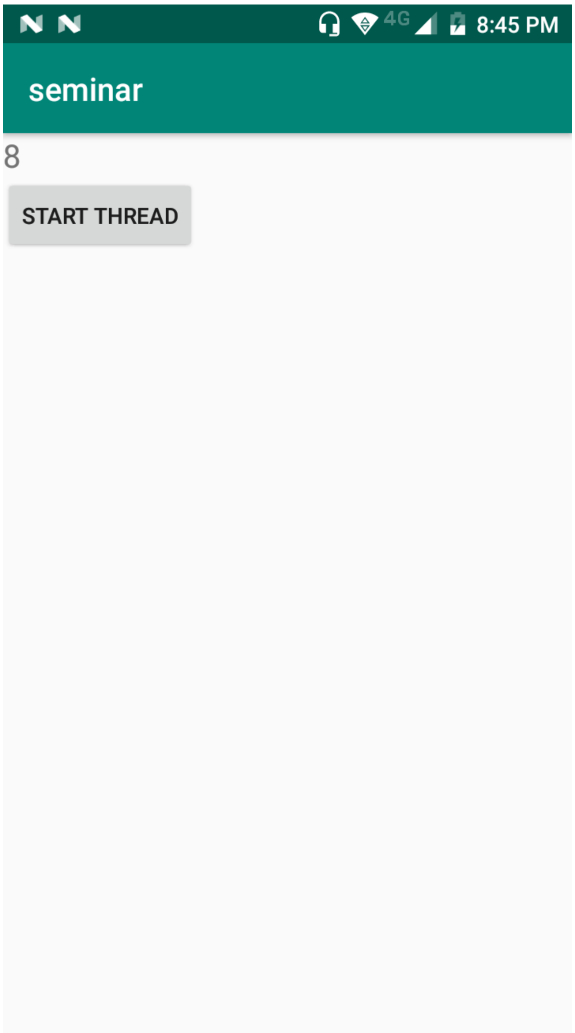
- **MainActivity.java**

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:orientation="vertical"
    android:layout_height="match_parent">

<TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="waiting to start!!"
        android:textSize="20dp"
        android:id="@+id/tv"
        />


<Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="create_thread"
        android:text="start thread"/>



</LinearLayout>
```

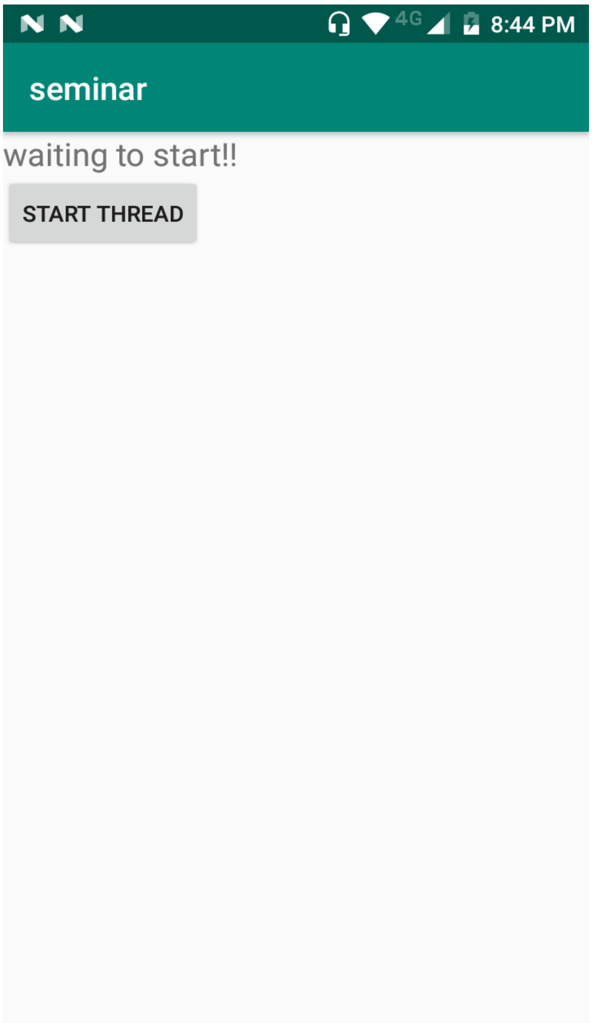
**MainActivity.java**

```
package com.example.seminar;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompat;

import android.app.NotificationManager;
import android.content.Context;
import android.os.Bundle;
import android.os.Handler;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import java.sql.Time;

public class MainActivity extends AppCompatActivity {
    TextView tv;
    int count = 10;
    Handler hand;
    thr obj;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        tv = findViewById(R.id.tv);

        hand = new Handler();
        obj = new thr();
    }

    public void create_thread(View view) {

       hand.postDelayed(obj,1000);

       count=10;

    }


    public class thr implements Runnable {

        @Override
        public void run() {
            if(count==0) {
               return;
            }
            else
            {
            hand.postDelayed(obj,1000);
            count--;
            tv.setText(count+"");}
    }
}}
```

- **Output**



8



waiting to start!!

**RESULT:**

Thus the multi-threading was successfully develop.

## GPS LOCATION INFORMATION

**Ex. No.: 7**

**Date:**

**AIM:**

To develop an application that uses GPS location information.

**ALGORITHM:**

- Create a new project in androidstudio.
- Edit the activity_main.xml file to create thelayout.
- Edit the files necessary to implement the android application and run theprogram.

**PROGRAM:**

- **MainActivity.java**

```
packagecom.siriusmicrotech.gpsplotter;

import android.location.Location;
import android.os.Bundle;
import android.support.v4.app.FragmentActivity;
import android.view.Menu;

import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.GooglePlayServicesClient;
import com.google.android.gms.location.LocationClient;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.maps.CameraUpdate;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.CameraPosition;
import com.google.android.gms.maps.model.LatLng;


public class MainActivity extends FragmentActivity
      implements GooglePlayServicesClient.ConnectionCallbacks,
      com.google.android.gms.location.LocationListener,
      GooglePlayServicesClient.OnConnectionFailedListener{
   privateGoogleMapmyMap;              // map reference
   private LocationClientmyLocationClient;
   private static final LocationRequest REQUEST = LocationRequest.create()
         .setInterval(5000)         // 5seconds
         .setFastestInterval(16)    // 16ms = 60fps
         .setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);

         @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      getMapReference();
   }
```

```
    @Override
    protected void onResume(){
        super.onResume();
        getMapReference();
        wakeUpLocationClient();
        myLocationClient.connect();
    }

     @Overrid
    e
    public void onPause(){
        super.onPause();
        if(myLocationClient != null){
            myLocationClient.disconnect();
        }
    }


    private void gotoMyLocation(double lat, double lng) {
        changeCamera(CameraUpdateFactory.newCameraPosition(new
CameraPosition.Builder().target(new LatLng(lat, lng))
                .zoom(15.5f)
                .bearing(0)
                .tilt(25)
                .build()
        ), new GoogleMap.CancelableCallback() {
            @Override
            public void onFinish() {
                // Your code here to do something after the Map is rendered
            }

            @Override
            public void onCancel() {
                // Your code here to do something after the Map rendering is cancelled
            }
        });
    }


    private void wakeUpLocationClient() {
        if(myLocationClient == null){
            myLocationClient = new LocationClient(getApplicationContext(),
                    this,       // Connection Callbacks
                    this);      // OnConnectionFailedListener
        }
    }


    private void getMapReference() {
        if(myMap == null){
            myMap = ((SupportMapFragment)
getSupportFragmentManager().findFragmentById(R.id.map))
                .getMap();
        }
        if(myMap != null){
            myMap.setMyLocationEnabled(true);
        }
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

     @Overrid
    e
    public void onConnected(Bundle bundle) {
        myLocationClient.requestLocationUpdates(
                REQUEST,
                this); // LocationListener
    }

     @Overrid
    e
    public void onDisconnected() {

    }

     @Overrid
    e
    public void onLocationChanged(Location location) {
        gotoMyLocation(location.getLatitude(), location.getLongitude());
    }

    @Override
    public void onConnectionFailed(ConnectionResult connectionResult) {
    }


    private void changeCamera(CameraUpdate update, GoogleMap.CancelableCallback callback)
{
        myMap.moveCamera(update);
    }
}
```

- **activity_main.xml**

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

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MapsActivity">

    <fragment

        class="com.google.android.gms.maps.SupportMapFragment"
        android:id="@+id/map"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:layout="@layout/abc_action_menu_layout" />
</RelativeLayout>
```

- **string.xml**

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

    <string name="app_name">GPSPlotter</string>
    <string name="action_settings">Settings</string>
    <string name="hello_world">Hello world!</string>

</resources>
```

- **AndroidManifest.xml**

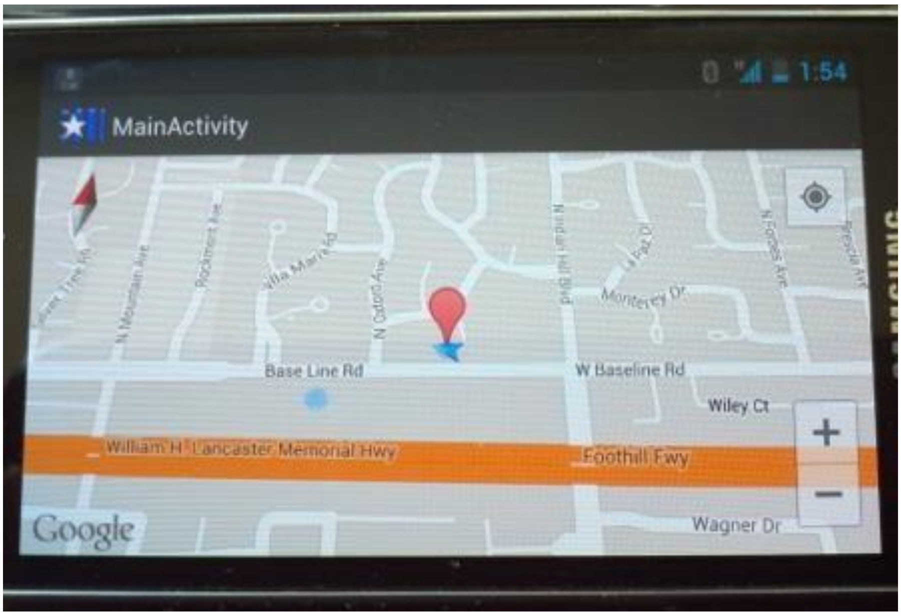
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.siriusmicrotech.gpsplotter"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="11"
        android:targetSdkVersion="17" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission
android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
    <!-- External storage for caching. -->
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <!-- My Location -->
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <!-- Maps API needs OpenGL ES 2.0. -->
    <uses-feature
        android:glEsVersion="0x00020000"
        android:required="true" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.siriusmicrotech.gpsplotter.MainActivity"
            android:label="@string/app_name" >
            <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.maps.v2.API_KEY"
            android:value="Your API Key Here!" />
    </application>

</manifest>
```

- **build.gradle**

```
buildscript {
    repositories {
        mavenCentral()
    }
    dependencies {
    classpath 'com.android.tools.build:gradle:0.5.+'
    }
}
apply plugin: 'android'

repositories {
    mavenCentral()
}

android {
    compileSdkVersion 18
    buildToolsVersion "18.0.1"

    defaultConfig {
        minSdkVersion 11
        targetSdkVersion 17
    }
}

dependencies {
    compile 'com.google.android.gms:play-services:3.2.65'
      compile 'com.android.support:appcompat-v7:18.0.0'
}
```

**OUTPUT:**



**RESULT:**
Thus the GPS location information was successfully developed.

## SD CARD WRITING

**Ex. No.: 8**

**Date:**

**AIM:**

To develop an application that writes data to the SD card.

**ALGORITHM:**

- Create a new project in androidstudio.
- Edit the activity_main.xml file to create thelayout.
- Edit the files necessary to implement the android application and run theprogram.

**PROGRAM:**

- **MainActivity.java**

```
package com.example.administrator.myapplication;

import  android.content.Context;
import android.graphics.Bitmap;
import android.os.Environment;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import java.io.File;
import java.io.FileOutputStream;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Random;
public class MainActivity extends AppCompatActivity {
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      Button b = (Button)findViewById(R.id.saveb);
      final EditText text = (EditText)findViewById(R.id.text);
      final EditText fname = (EditText)findViewById(R.id.fname);
      b.setOnClickListener(new View.OnClickListener(){
         @Override
         public void onClick(View v) {
            Save(" " + fname.getText().toString()," " + text.getText().toString());
         }
```

```
        });
    }

    private void Save(String sFileName, String sBody) {
        try {
            File root = new File(Environment.getExternalStorageDirectory(), "cse");
            if (!root.exists()) {
                root.mkdirs();
            }
            File textFile = new File(root, sFileName);
            FileWriter writer = new FileWriter(textFile);
            writer.append(sBody);
            writer.flush();
            writer.close();
            Toast.makeText(this.getApplicationContext(), "Saved", Toast.LENGTH_SHORT).show();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

- **activity_main.xml**

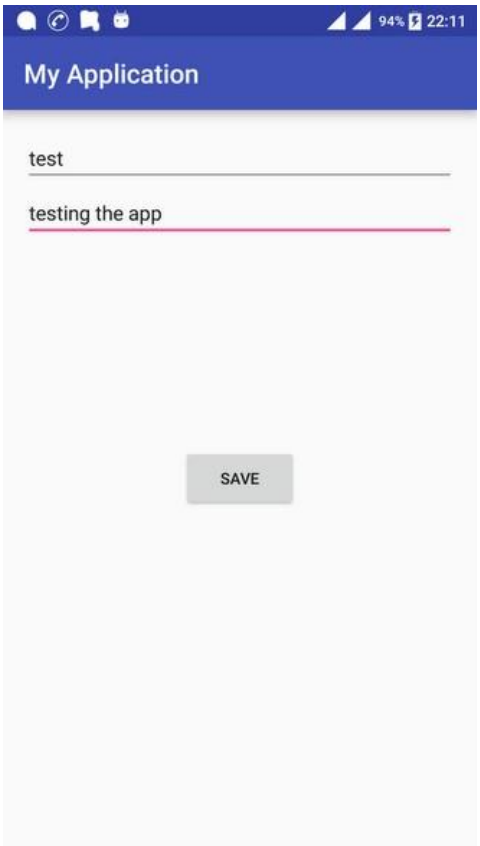
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context="com.example.administrator.myapplication.MainActivity"
    android:orientation="vertical"
    android:gravity="center|top">

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="File Name"
        android:maxLines="1"
        android:id="@+id/fname"/>
    <EditText
        android:layout_width="match_parent"
```

```
          android:layout_height="wrap_content"
          android:inputType="textMultiLine"
          android:hint="Write your content"
          android:id="@+id/text"/>

        <Button
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:text="save"
          android:id="@+id/saveb"/>

</LinearLayout>
```

- **AndroidManifest.xml**

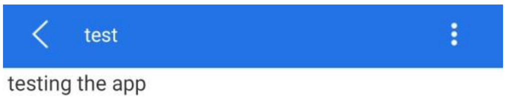
```
    <?xml version="1.0" encoding="utf-8"?>
    <manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.administrator.myapplication" >

    <application
      android:allowBackup="true"
      android:icon="@mipmap/ic_launcher"
      android:label="@string/app_name"
      android:supportsRtl="true"
      android:theme="@style/AppTheme">
      <activity android:name=".MainActivity" >
        <intent-filter>
          <action android:name="android.intent.action.MAIN" />

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

    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
</manifest>
```

## OUTPUT:



test

testing the app



testing the app

**RESULT:**

Thus the SD card writing was successfully developed.

## ALERT MESSAGE

**Ex. No.: 9**

**Date:**

**AIM:**

To develop an application that creates an alert upon receiving a message.

**ALGORITHM:**

- Create a new project in androidstudio.
- Create a broadcast receiver to listnen messages.
- Display the alert when a new message is received.
- Run the project and test it in a mobiledevice.

**PROGRAM:**

- **MainActivity.java**

```
package com.example.mcom;

import androidx.appcompat.app.AppCompatActivity;

import android.app.AlertDialog;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);


        AlertDialog.Builder j = new AlertDialog.Builder(this);
        j.setMessage("message received").setTitle("Title");
        AlertDialog k = j.create();
        k.show();

    }
}
```

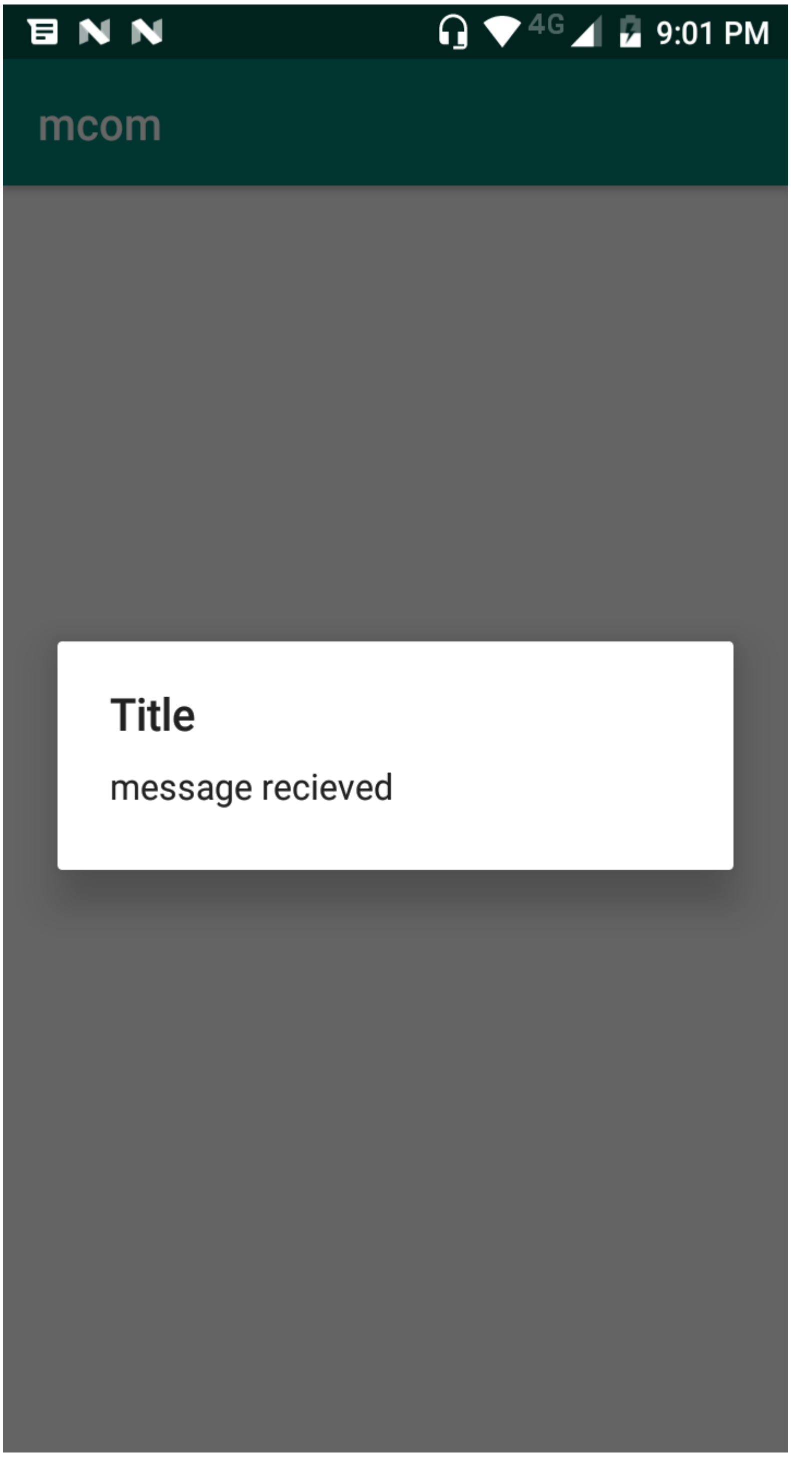
- **smsReciever.java**

```
package com.example.mcom;

import android.app.AlertDialog;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.view.WindowManager;
import android.widget.Toast;

public class smsReciever extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        try {
            Intent i = new Intent(context,MainActivity.class);
            i.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
            context.startActivity(i);
        }
        catch (Exception e) {
            Toast.makeText(context,e.getMessage().toString(),Toast.LENGTH_LONG).show();

        }
        }
    }
```

- **AndroidManifest.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.mcom">
<uses-permission android:name="android.permission.READ_SMS"/>
<uses-permission android:name="android.permission.RECEIVE_SMS"/>
<uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW"/>
<application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
<activity android:name=".MainActivity">
<intent-filter>
<action android:name="android.intent.action.MAIN" />

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

<receiver android:name=".smsReciever">
<intent-filter>
<action android:name="android.provider.Telephony.SMS_RECEIVED"/>
</intent-filter>
</receiver>
</application>

</manifest>
```

- **Output**



Title

message recieved

**RESULT:**

Thus the alert message was successfully developed.

## RSS FEED

**Ex. No.: 10**

**Date:**

**AIM:**

To develop an application that makes use of RSS feed.

**ALGORITHM:**

- Create a new project in androidstudio.
- Edit the activity_main.xml file to create thelayout.
- Edit the files necessary to implement the android application and run theprogram.

**PROGRAM:**

- **MainActivity.java**

```
package com.example.myapplication;

import androidx.appcompat.app.AppCompatActivity;
import android.os.AsyncTask;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.Toast;
import org.mcsoxford.rss.RSSFeed;
import org.mcsoxford.rss.RSSItem;
import org.mcsoxford.rss.RSSReader;
import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {

final ArrayList<String> list = new ArrayList<>();
ArrayAdapter adapter;


@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);

adapter = new ArrayAdapter(this,R.layout.listcell,R.id.txt,list);

ListView listView = findViewById(R.id.list);

listView.setAdapter(adapter);

bgTask task = new bgTask();

task.execute();

}
class bgTask extends AsyncTask{

@Override
protected Object doInBackground(Object[] objects) {

try{

RSSReader reader = new RSSReader();
String uri = "http://www.rssreader.com/englishfeeds.xml";
RSSFeed feed = reader.load(uri);

for(RSSItem article : feed.getItems()){

list.add(article.getTitle());
}

}
catch (Exception e){


}
return null;
}

@Override
protected void onPostExecute(Object o) {
super.onPostExecute(o);

adapter.notifyDataSetChanged();
}
}
}
```

- **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">

<ListView
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:id="@+id/list"
      />
</RelativeLayout>
```

- **listcell.xml**

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

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">

<TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="18dp"
        android:id="@+id/txt"
        android:padding="20dp"/>

</LinearLayout>
```

- **AndroidManifest.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.myapplication">

<application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
<activity android:name=".MainActivity">
<intent-filter>
<action android:name="android.intent.action.MAIN" />

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

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

</manifest>
```

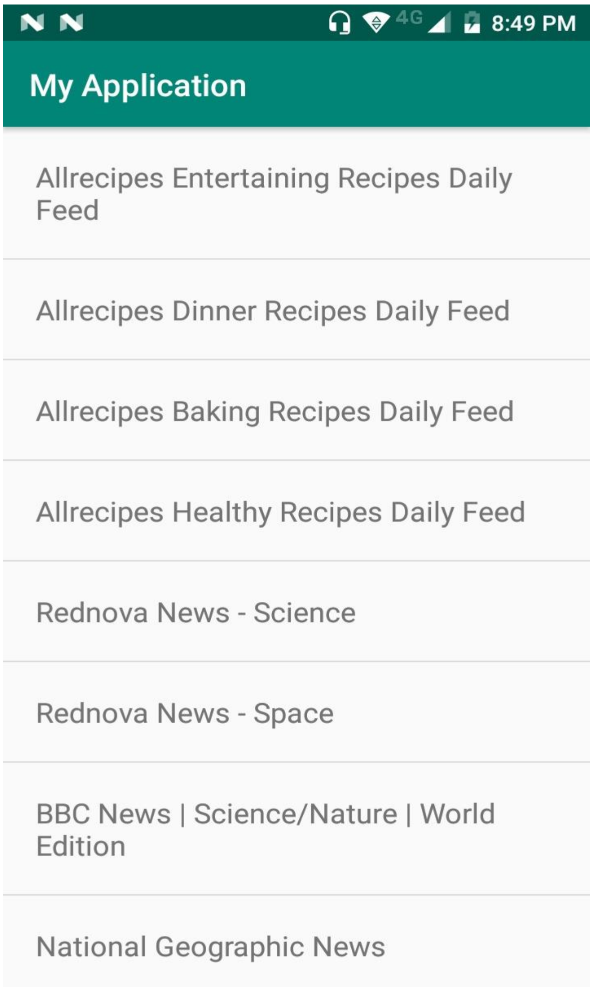
- **Add the Following Dependencie To Build.grade(Module: app) File**

```
implementation 'com.github.ahorn:android-rss:v1.0-rc1'
```

- **Add the Following Dependencie To Build.grade(Project : ApplicationName) File**

```
maven { url 'https://jitpack.io' }
```

- **Output**



8:49 PM

My Application

Allrecipes Entertaining Recipes Daily Feed

Allrecipes Dinner Recipes Daily Feed

Allrecipes Baking Recipes Daily Feed

Allrecipes Healthy Recipes Daily Feed

Rednova News - Science

Rednova News - Space

BBC News | Science/Nature | World Edition

National Geographic News

**RESULT:**

Thus the RSS feed was successfully developed.

## Email

**Ex. No.: 11**

**Date:**

**AIM:**

To develop an application to compose an Email.

**ALGORITHM:**

- Create a new project in androidstudio.
- Edit the activity_main.xml file to create thelayout.
- Edit the files necessary to implement the android application and run theprogram.

**PROGRAM:**

- **MainActivity.java**

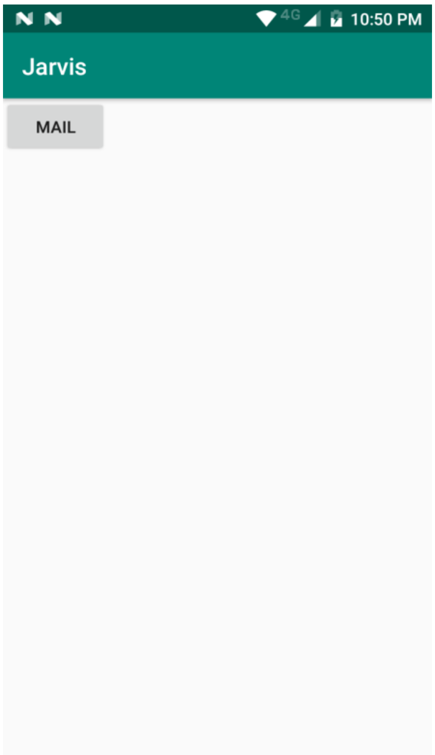
```
package com.example.jarvis;

import androidx.appcompat.app.AppCompatActivity;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.view.View;


public class MainActivity extends AppCompatActivity {


@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
}



public void composeEmail(View v) {
Intent emailIntent = new Intent(Intent.ACTION_SENDTO, Uri.parse("mailto:" + "psanirudh@yahoo.com"));
startActivity(Intent.createChooser(emailIntent, "Chooser Title"));
}
}
```

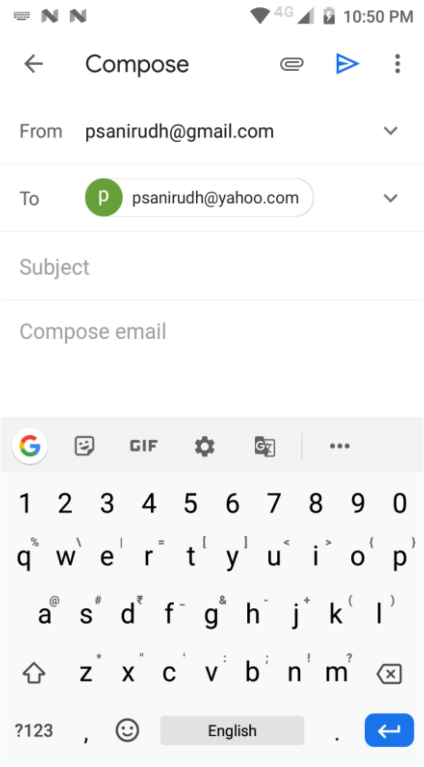
- **Activity_main.xml**

```
<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"
>
<Button
android:layout_width="wrap_content"
android:text="mail"
android:layout_height="wrap_content"
android:onClick="composeEmail"/>
</RelativeLayout>
```

- **Output**





From psanirudh@gmail.com

To psanirudh@yahoo.com

**RESULT:**

Thus the email was successfully developed.

## Mini Project

**Ex. No.: 12**

**Date:**

**AIM:**

To develop an mini project by using Android Studio.

**ALGORITHM:**

- Create a new project in AndroidStudio.
- Design an initial layout for User Details UI
- Edit the activity_main.xml file to create thelayout.
- Edit the files necessary to implement the android application and run theprogram.

- **PROGRAM**

```
package com.example.myapplication;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

import android.os.AsyncTask;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.Toast;

import com.google.firebase.database.ChildEventListener;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.FirebaseDatabase;

import org.mcsoxford.rss.RSSFeed;
import org.mcsoxford.rss.RSSItem;
import org.mcsoxford.rss.RSSReader;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.UUID;

public class MainActivity extends AppCompatActivity {

final ArrayList<String>list = new ArrayList<>();
    ArrayAdapter adapter;
    EditText msg;

    @Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_main);

    ListView listView = findViewById(R.id.list);
    Button send = findViewById(R.id.send);
msg = findViewById(R.id.msg);

adapter = new ArrayAdapter(MainActivity.this,R.layout.listcell,R.id.txt,list);

    listView.setAdapter(adapter);
    getmsg();

    send.setOnClickListener(new View.OnClickListener() {
      @Override
public void onClick(View v) {

        saveMsg();
      }
    });
  }

void saveMsg(){

    FirebaseDatabase firebaseDatabase = FirebaseDatabase.getInstance();


firebaseDatabase.getReference().child("msgs").child(UUID.randomUUID().toString()).child("msg").
setValue(msg.getText().toString());

msg.setText("");
  }

void getmsg(){

final FirebaseDatabase database = FirebaseDatabase.getInstance();

    database.getReference().child("msgs").addChildEventListener(new ChildEventListener() {
      @Override
public void onChildAdded(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {

        HashMap<String,Object> map = (HashMap<String, Object>) dataSnapshot.getValue();

list.add(map.get("msg").toString());

adapter.notifyDataSetChanged();



      }

      @Override
public void onChildChanged(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {

      }

      @Override
public void onChildRemoved(@NonNull DataSnapshot dataSnapshot) {
            }
            @Override
public void onChildMoved(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {
            }
            @Override
public void onCancelled(@NonNull DatabaseError databaseError) {
            }
        });
    }
}
```

- **activity_main.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="18dp"
        android:padding="5dp"
        android:id="@+id/txt"/>

</LinearLayout>
```

- **ACTIVITY MAIN**

```
<?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">

    <ListView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_above="@id/msg"
        android:id="@+id/list"
        android:dividerHeight="0dp"
        android:divider="@null"
        android:transcriptMode="alwaysScroll"
        />

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/msg"
        android:layout_alignParentBottom="true"
        android:layout_toLeftOf="@id/send"
        android:paddingTop="8dp"/>

    <Button
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Send"
        android:id="@+id/send"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        />
</RelativeLayout>
```

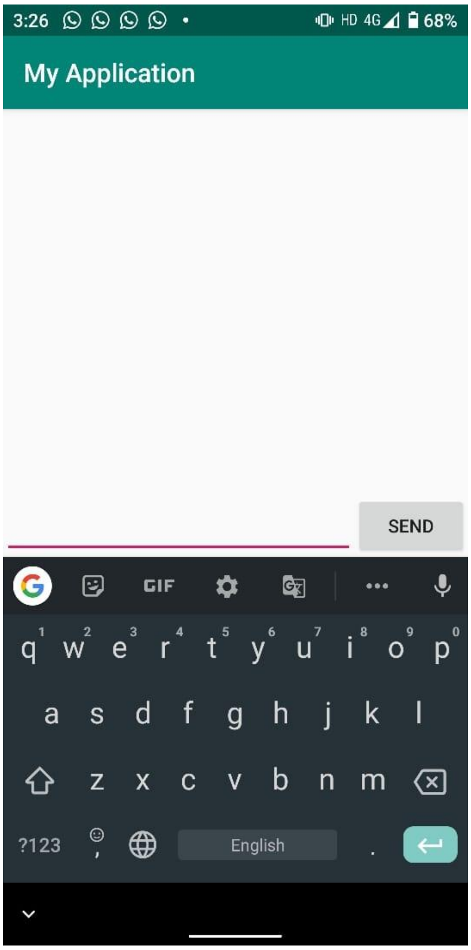
- **Add the Following Dependencie To Build.grade File**

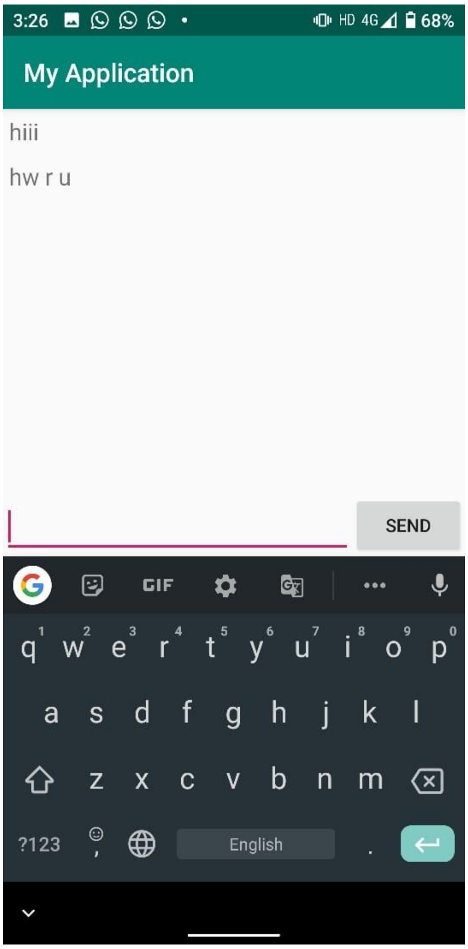
```
implementation 'com.google.firebase:firebase-database:16.0.4'
```

- **Add the Downloaded google-services.json file to**

  **Project -> App Directory**

- **Output**





hiii

hw r u

**RESULT:**

Thus the chat app as mini project was successfully developed.