DIPLOMA IN COMPUTER

ENGINEERING

CSM 65– MOBILE COMPUTING PRACTICAL

M – SCHEME

(VI SEM)

2019 – 2020

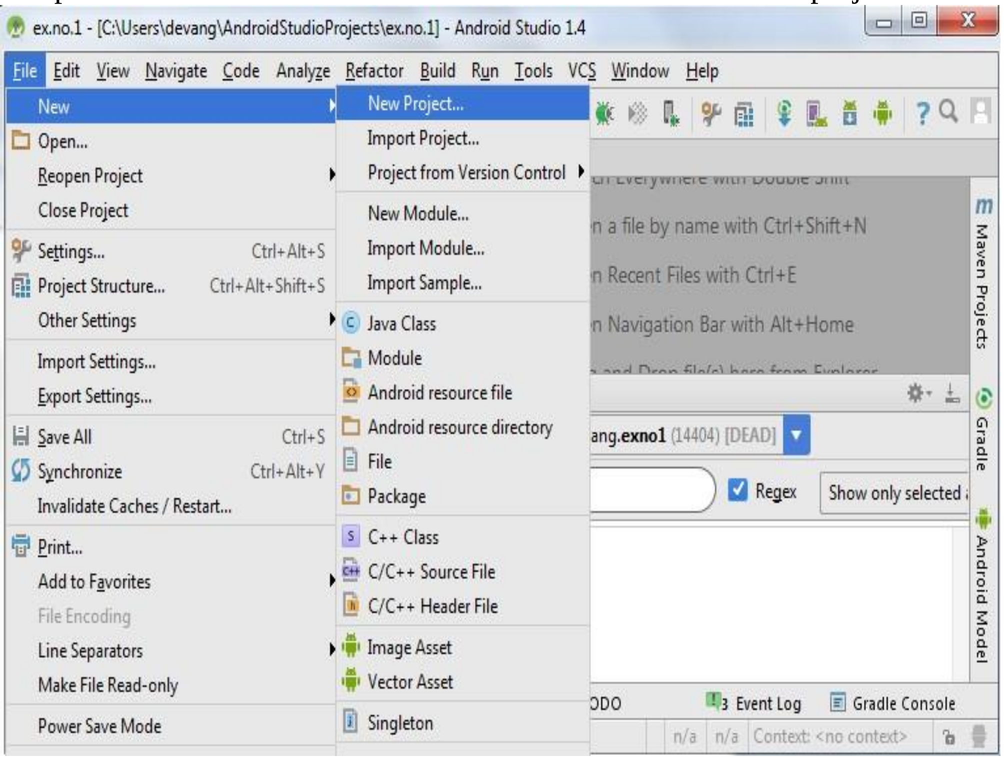
NAME	:	……………………………………………...
ROLL NO	:	……………………………………………...
CLASS	:	……………………………………………...

INDEX

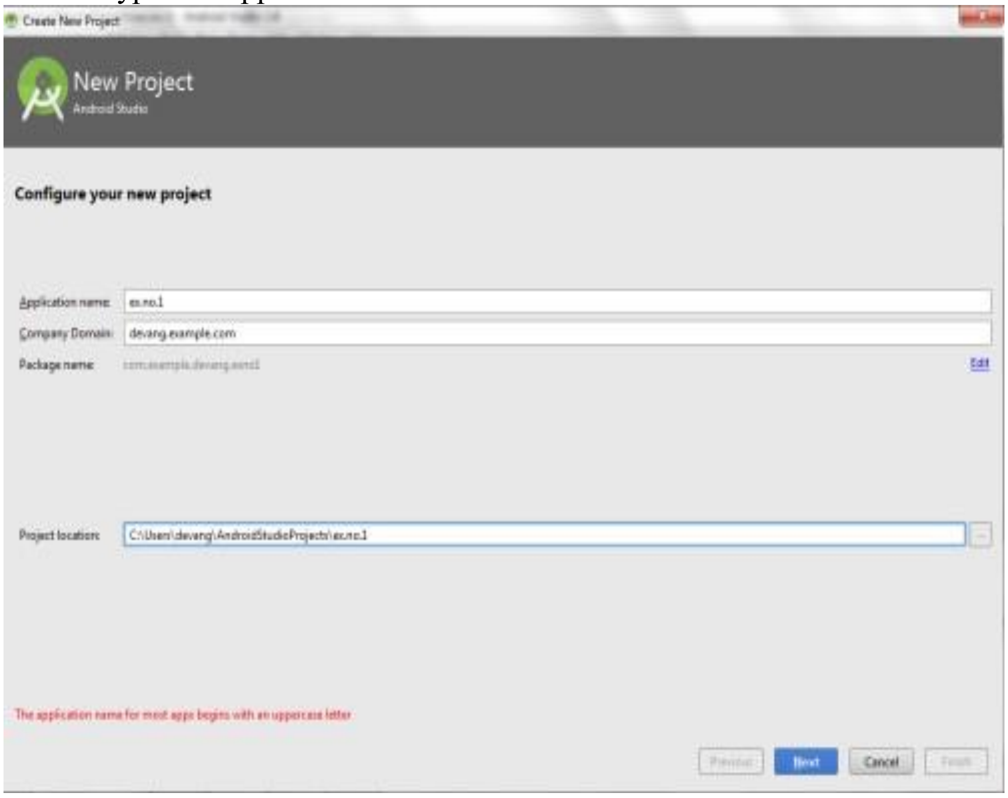
SL. NO	DATE	NAME OF THE EXPERIMENT	MARKS	SIGN
1.		APPLICATION LIFE CYCLE		
2.		LAYOUTS		
3.		SIMPLE CALCULATOR		
4.		LIST VIEW		
5.		PHOTO GALLERY		
6.		DATE PICKER AND TIME PICKER		
7.		CONTEXT MENU AND OPTION MENU		
8.		SMS APPLICATION		
9.		CONTACTS APP		
10.		E-MAIL APP		
11.		SERVICE APP		
12.		WEB VIEW APP		
13.		MAP VIEW APP		
14.		INTENT CLASS		
15.		EXTERNAL MEMORY		

STEPS TO CREATE A NEW PROJECT

1. Open Android Studio and then click on File -> New -> New project.



2. Then type the Application name as “ex1″ and click Next.



Configure your new project

Application name: ex.no.1

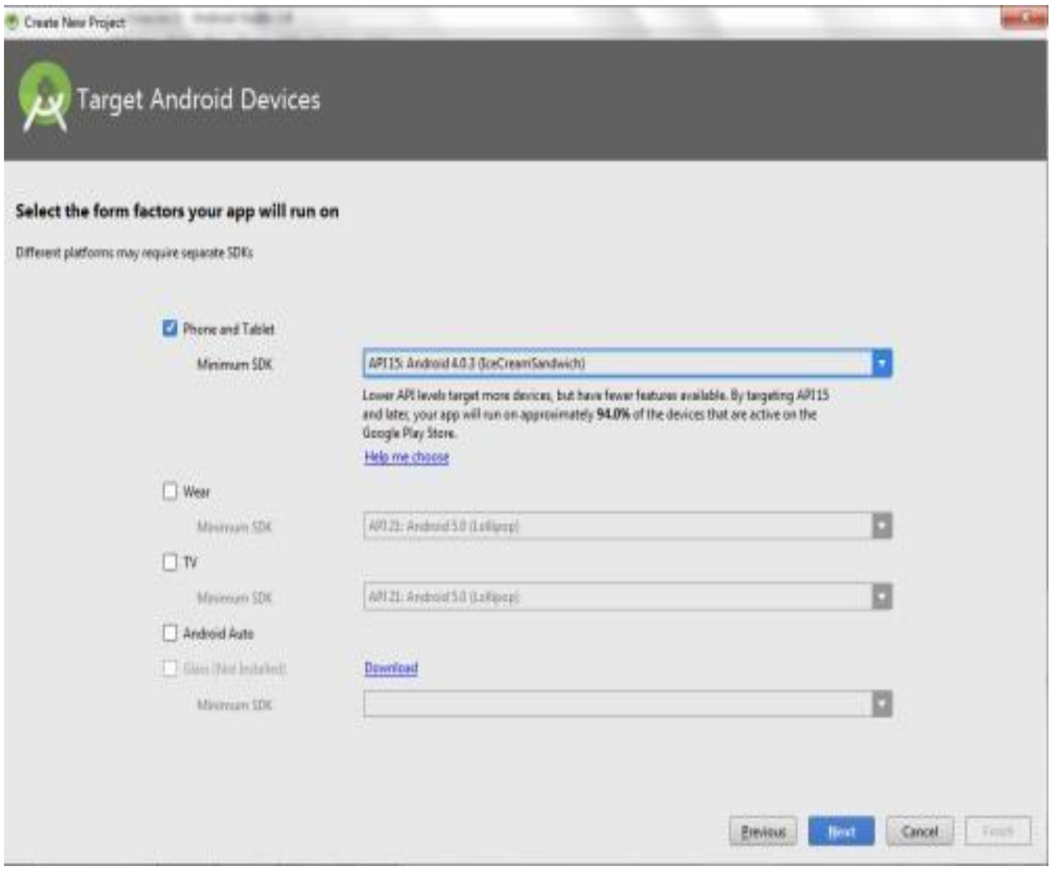
Company Domain: devang.example.com

Package name: com.example.devang.exno1

Project location: C:\Users\devang\AndroidStudioProjects\ex.no.1

The application name for most apps begins with an uppercase letter

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



Create New Project

Target Android Devices

Select the form factors your app will run on

Different platforms may require separate SDKs

☑ Phone and Tablet

Minimum SDK: API 15: Android 4.0.3 (IceCreamSandwich)

Lower API levels target more devices, but have fewer features available. By targeting API 15 and later, your app will run on approximately **94.0%** of the devices that are active on the Google Play Store.

Help me choose

☐ Wear

Minimum SDK: API 21: Android 5.0 (Lollipop)

☐ TV

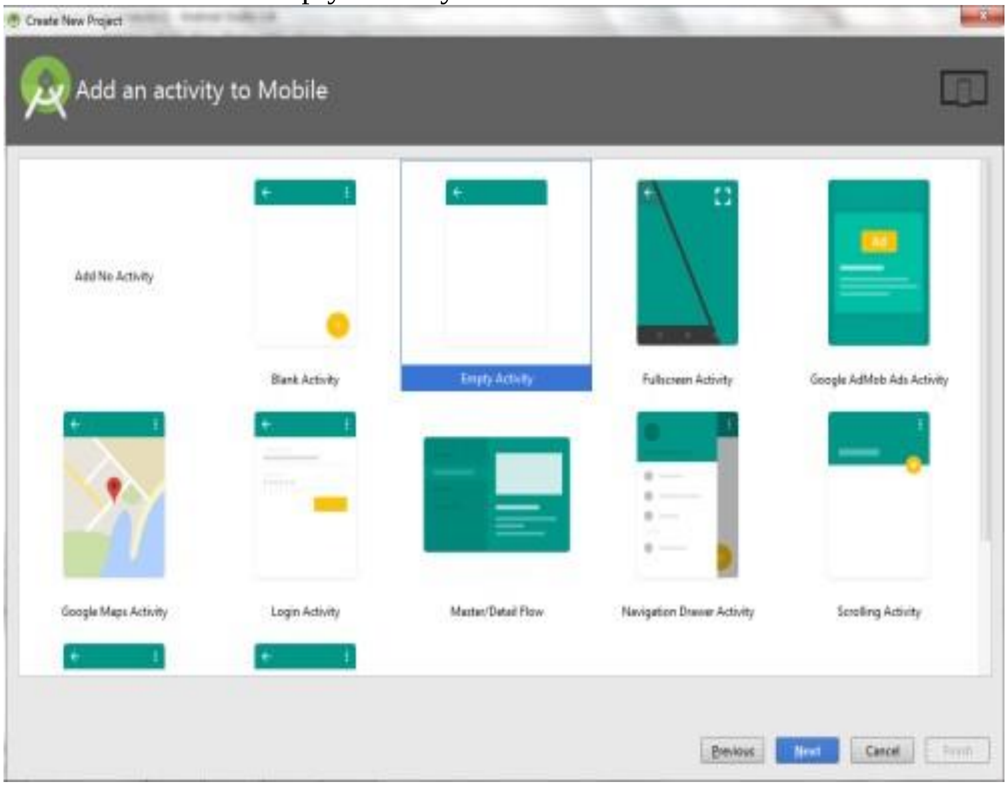
Minimum SDK: API 21: Android 5.0 (Lollipop)

☐ Android Auto

☐ Glass (Not Installed) Download

Minimum SDK

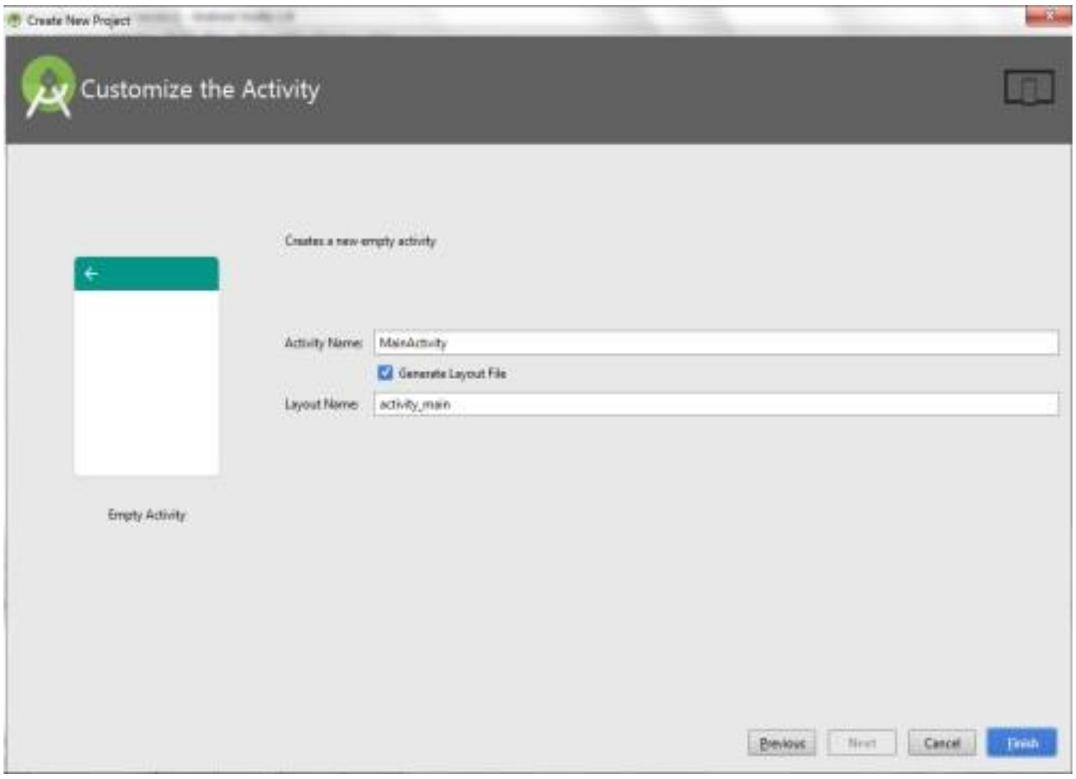
4. Then select the Empty Activity and click Next.



Create New Project

Add an activity to Mobile

5. Finally click Finish.



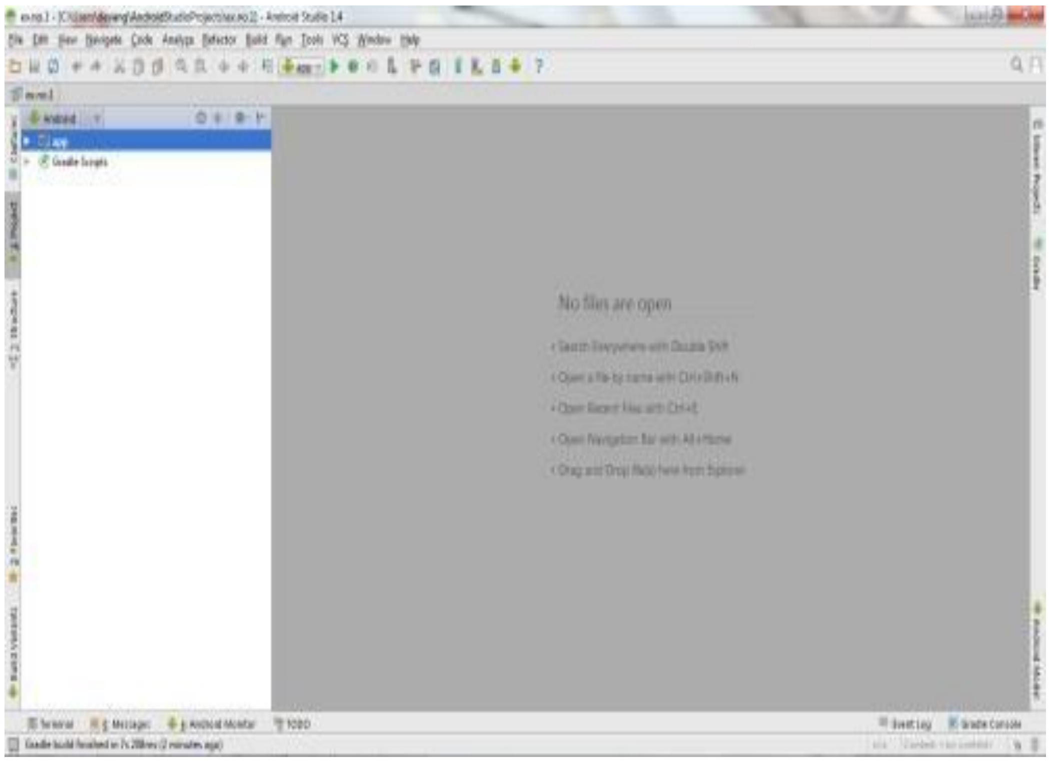
Creates a new empty activity

Activity Name: MainActivity

☑ Generate Layout File

Layout Name: activity_main

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



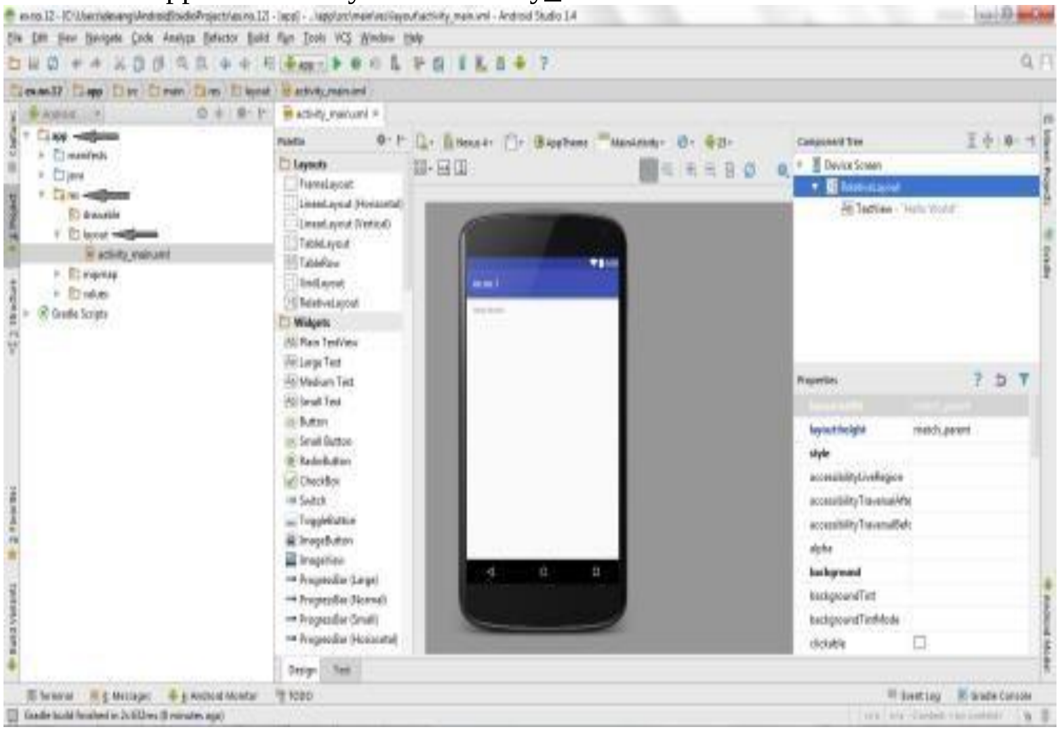
No files are open

- Search Everywhere with Double Shift
- Open a file by name with Ctrl+Shift+N
- Open Recent Files with Ctrl+E
- Open Navigation Bar with Alt+Home
- Drag and Drop file(s) here from Explorer

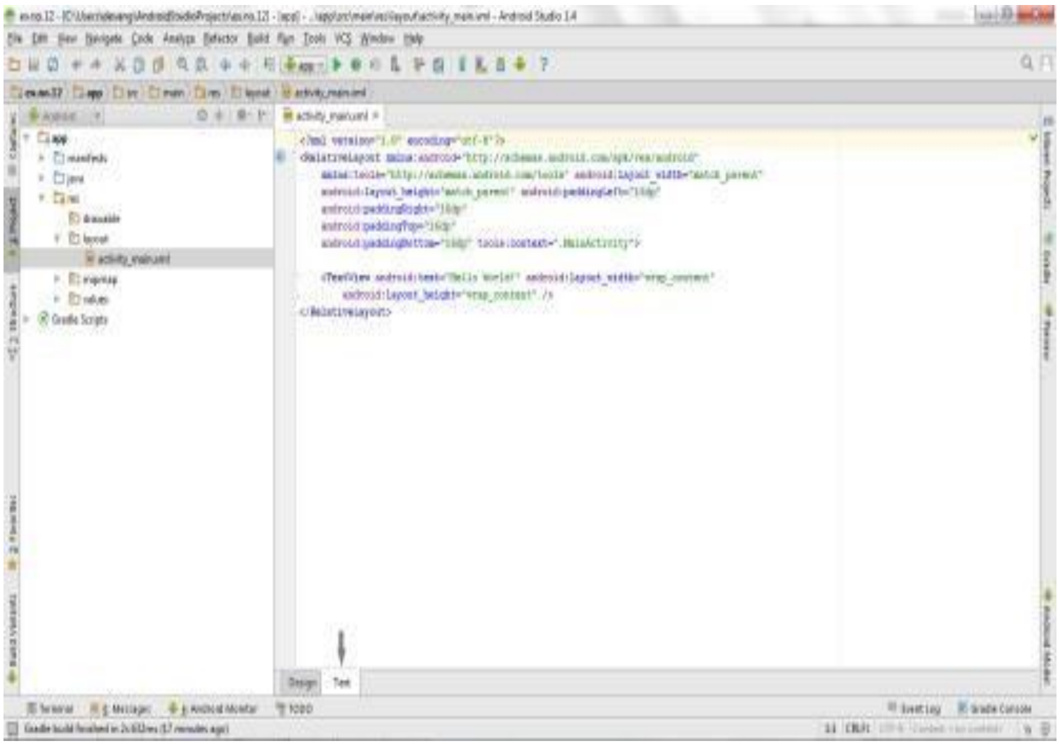
Gradle build finished in 7s 28ms (2 minutes ago)

DESIGNING LAYOUT FOR THE ANDROID APPLICATION:

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



2. Now click on Text as shown below.



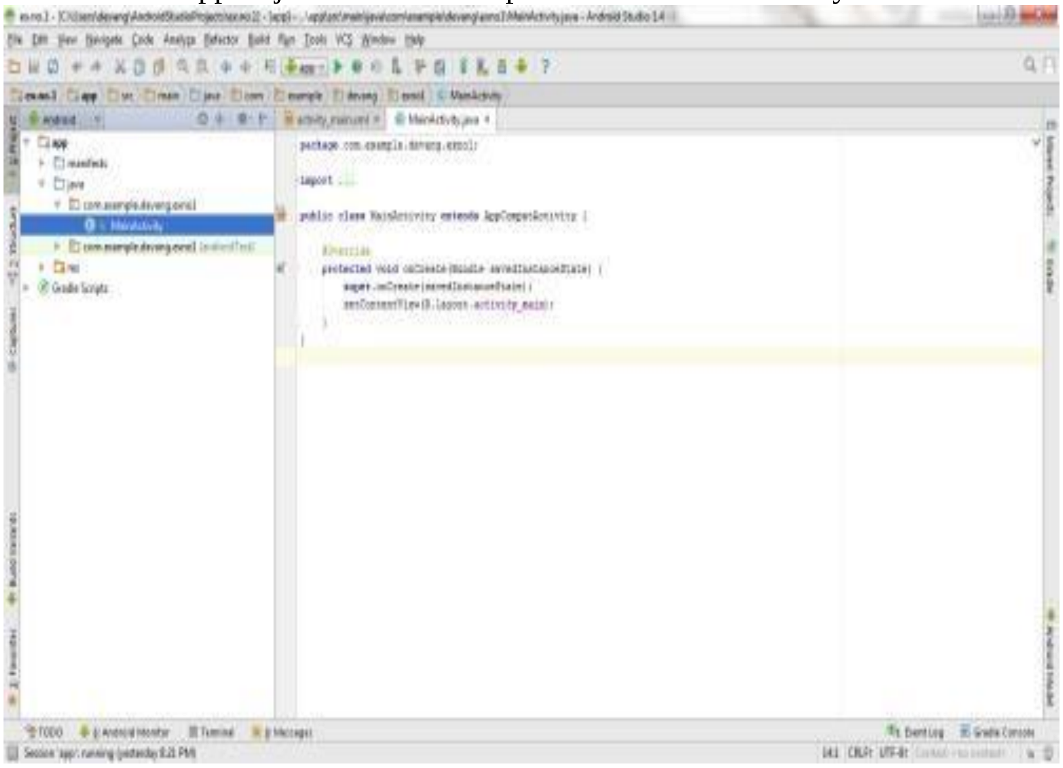
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent" android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp" tools:context=".MainActivity">

    <TextView android:text="Hello World!" android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</RelativeLayout>
```

3. Then type the code required for the program.

JAVA CODING FOR THE ANDROID APPLICATION:

1. Click on app -> java -> com.example.exno1 -> MainActivity.



```
package com.example.devang.exno1;

import ...

public class MainActivity extends AppCompatActivity {

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

2. Then type the required code for the program.

EX. NO : 1 **Date:**	**APPLICATION LIFE CYCLE**

AIM:

To develop an android program to demonstrate activity.

PROCEDURE:

Following are the 6 States in the life cycle of Android Application.

i. onCreate()
   This event is called when an activity is created.
ii. onStart()
   This event is called when an activity is started.
iii. onResume()
   This event is called when an activity is resumed.
iv. onPause()
   This event is called when an activity is paused.
v. onStop()
   This event is called when an activity is stopped.
vi. onDestroy()
   This event is called when an activity is destroyed.

PROGRAM:

<u>MainActivity.Java</u>

```
Public class MainActivity extends AppCompatActivity {
   @Override
        protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Toast.makeText(this,"Activity Created",Toast.LENGTH_LONG).show();
   }
   @Override
protected void onPause() {    super.onPause();
        Toast.makeText(this,"Activity Paused",Toast.LENGTH_LONG).show();
   }
   @Override
protected void onResume() {  super.onResume();
        Toast.makeText(this,"Activity resumed",Toast.LENGTH_LONG).show();
   }
   @Override
protected void onStop() {         super.onStop();
        Toast.makeText(this,"Activity Stopped",Toast.LENGTH_LONG).show();
   }
   @Override
protected void onDestroy() {    super.onDestroy();
              Toast.makeText(this,"Activity Destroyed",Toast.LENGTH_LONG).show();
   }}
```

OUTPUT:

RESULT:

Thus, an android program to demonstrate activity lifecycle is executed successfully.

EX. NO : 2 Date:	LAYOUTS

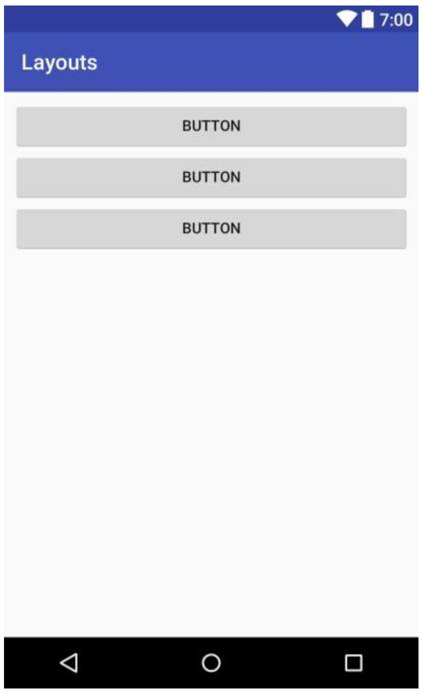
AIM:

To develop an android program to demonstrate different types of layouts.

PROCEDURE:

- A layout defines the visual structure for a user interface. The different layouts are:
  1. Linear Layout
     The LinearLayout arranges views in a single column or a single row. Child views can be arranged either vertically or horizontally
  2. Relative Layout
     RelativeLayout is a view group that displays child views in relative positions.
  3. Table Layout
     TableLayout is a view that groups views into rows and columns.
  4. Grid View
     GridView is a ViewGroup that displays items in a two-dimensional, scrollable grid.
  5. Table Layout
     TableLayout is a view that groups views into rows and columns.
  6. Frame Layout
     The FrameLayout is a placeholder on screen that you can use to display a single view.

UI DESIGN:



PROGRAM:

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Previous" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="OK" />

    <Button
        android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="next" />
</LinearLayout>
```

MainActivity.java

```
public class MainActivity extends Activity {
   @Override
   protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);       setContentView(R.layout.activity_main);
   }
}
```

OUTPUT:

RESULT:

Thus, an android program to demonstrate different types of layouts is executed successfully.

EX. NO : 3 Date:	SIMPLE CALCULATOR

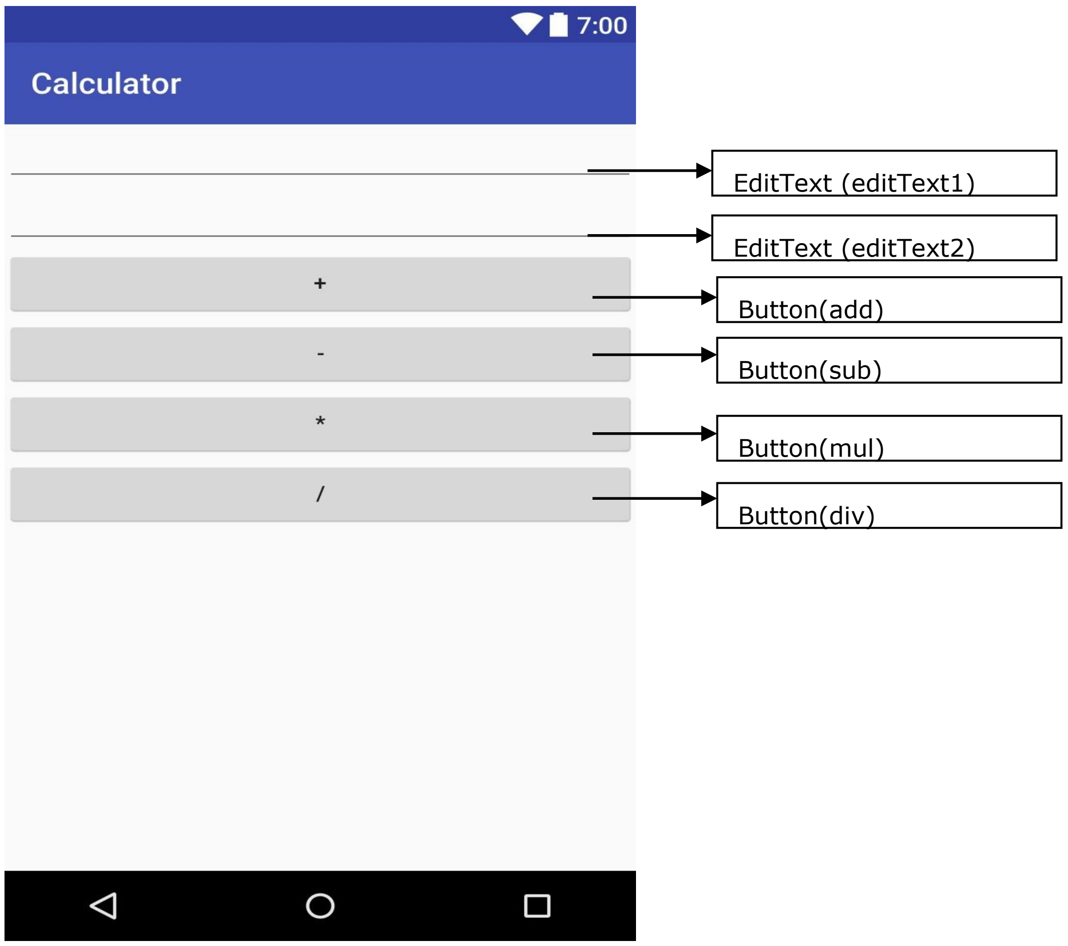
AIM:

To develop an android program to implement simple calculator using text view, edit view, option button and button.

PROCEDURE:

1. Create a Linearlayout vertical
2. Add 2 Edittext for entering numbers in it.
3. Then, add four button (+, -, *, /)
4. Call the Calc() method in the onClick attribute of the above buttons.

<u>UI DESIGN:</u>



PROGRAM:

<u>MainActivity.java</u>

```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;
```

```
public class MainActivity extends AppCompatActivity {



    EditText num1,num2;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        num1=findViewById(R.id.editText);
        num2=findViewById(R.id.editText2);
    }
    public void calc(View v){
        int val1=Integer.parseInt(num1.getText().toString());
        int val2=Integer.parseInt(num2.getText().toString());
        int result=0;
        switch(v.getId())
        {
            case R.id.button:
                result=val1+val2;
                break;
            case R.id.button2:
                result=val1-val2;
                break;
            case R.id.button3:
                result=val1*val2;
                break;
            case R.id.button4:
                result=val1/val2;
                break;
        }
        Toast.makeText(this,"The Result is "+result,Toast.LENGTH_LONG).show();
    }
}
```

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"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/editText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight=""
        android:ems="10"
        android:inputType="textPersonName" />

    <EditText
        android:id="@+id/editText2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight=""
        android:ems="10"
        android:inputType="textPersonName" />

    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="calc"
        android:text="+" />

    <Button
        android:id="@+id/button2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="calc"
        android:text="-" />

    <Button
        android:id="@+id/button3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="calc"
        android:text="*" />

    <Button
        android:id="@+id/button4"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="calc"
        android:text="/" />

</LinearLayout>
```

OUTPUT:

RESULT:

Thus, an android program to implement simple calculator is executed successfully.

EX. NO : 4 **Date:**	**LIST VIEW**

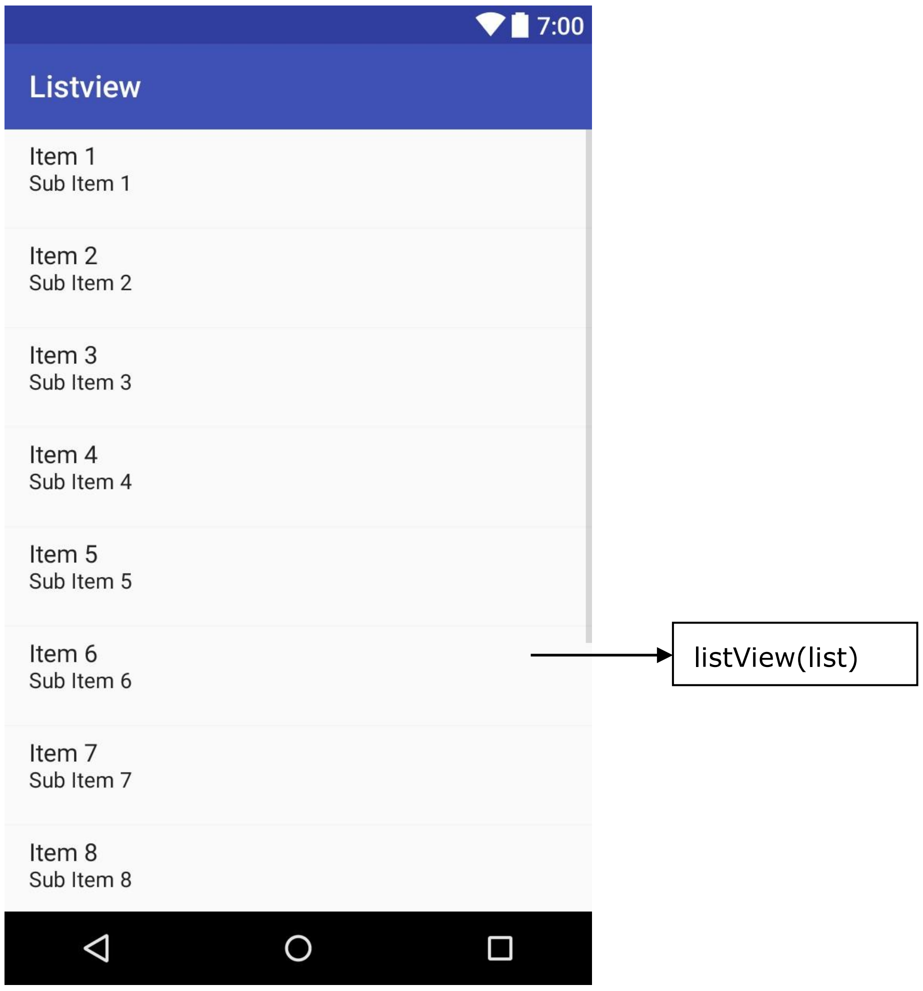
AIM:

To develop an android program to demonstrate ListView.

PROCEDURE:

1. Select Listview from the palette. Drag and drop on virtual mobile screen to create the UI.
2. An adapter is a bridge between UI component and data source that helps us to fill data in UI component.

UI DESIGN:



PROGRAM:

MainActivity.java

```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.ListView;

public class MainActivity extends AppCompatActivity {
    ListView mListView;
    String[] Countries=new
String[]{"India","Australia","Newzealand","Indonesia","China","Russia","Japan","South
Korea"};
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mListView=(ListView)findViewById(R.id.list);
        ArrayAdapter<String>countryAdapter=new
ArrayAdapter<String>(this,android.R.layout.simple_list_item_1,Countries);
        mListView.setAdapter(countryAdapter);
    }
}}
```

ACTIVITY_MAIN.XML

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
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:id="@+id/list"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</android.support.constraint.ConstraintLayout>
```

OUTPUT:

RESULT:

Thus, an android program to demonstrate ListView is executed successfully.

EX. NO : 5 Date:	PHOTO GALLERY

AIM:

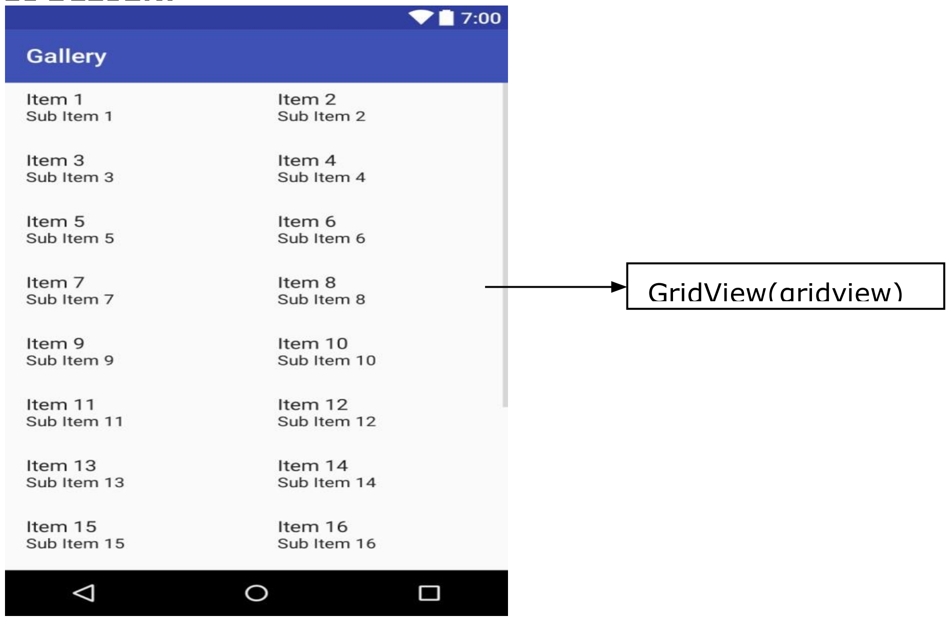
To develop an android program to demonstrate Photo Gallery.

PROCEDURE:

A Gallery is an android widget that helps us to display items (images) in 2-dimensional grid.

1. Add Gridview widget to the UI design.
2. Copy 5 images (p1.jpg, p2.jpg, p3.jpg, p4.jpg, p5.jpg, p6.jpg) to /main/res/drawable folder.

UI DESIGN:



PROGRAM:

MainActivity.java

```
import android.content.Context;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.GridView;
import android.widget.ImageView;

import java.util.Vector;

public class MainActivity extends AppCompatActivity {
   GridView gridView;

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

```
        gridView.setAdapter(new myadapter(this));
    }
    public class myadapter extends BaseAdapter{
        private Context mcontext;
        private int[]
img={R.drawable.p1,R.drawable.p2,R.drawable.p3,R.drawable.p4,R.drawable.p5,R.drawable.p6,R.d
rawable.p7,R.drawable.p8};
        public myadapter(Context c)
        {

            mcontext=c;
        }
        public int getCount(){

            return img.length;
        }
        public View getView(int i, View v, ViewGroup vg){
            ImageView imgview=new ImageView(mcontext);
            imgview.setImageResource(img[i]);
            imgview.setLayoutParams(new GridView.LayoutParams(250,250));
            return imgview;
        }
        public Object getItem(int position){

            return position;
        }
        public long getItemId(int position)
        {
            return position;
        }
    }
}
```

ACTIVITY_MAIN.XML

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
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">

<GridView
    android:id="@+id/grid"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />

</android.support.constraint.ConstraintLayout>
```

OUTPUT:

RESULT:

Thus, an android program to demonstrate PhotoGallery is executed successfully.

EX. NO : 6 **Date:**	**DATE TIME PICKER**

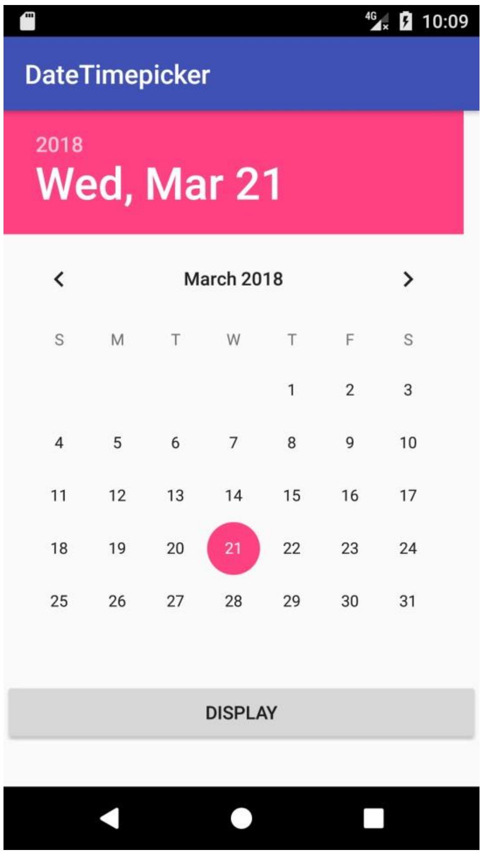
AIM:

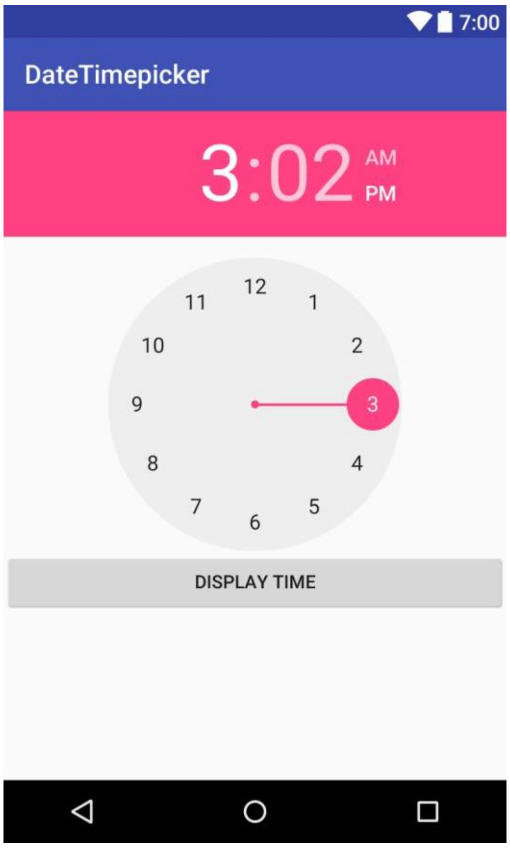
To develop an android program to demonstrate Date Picker and Time Picker.

PROCEDURE:

DateTime Picker components are used to select date and time in a customized user interface.

1. Add a DatePicker and button. Similarly, design for Timepicker.
2. The UI is shown below:





3. Call the display() method in the onclick attribute of the button.

PROGRAM:

<u>DATEPICKER</u> <u>MainActivity.java</u>

```
package com.example.balashanmugam.datepicker;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
DatePicker d1;
Button b1;
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        d1=(DatePicker)findViewById(R.id.simpledp);
        b1=(Button)findViewById(R.id.button2);
    }
    public void display(View v)
    {
        int a=d1.getMonth()+1;
        Toast.makeText(this,"Date
selected:"+d1.getDayOfMonth()+"/"+a+"/"+d1.getYear(),Toast.LENGTH_LONG).show();
    }
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <DatePicker
        android:id="@+id/simpledp"
        android:layout_width="404dp"
        android:layout_height="413dp"></DatePicker>

    <Button
        android:id="@+id/button2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="display"
        android:text="Display" />

</LinearLayout>
```

TIME PICKER

MainActivity.java

```
package com.example.balashanmugam.timepicker;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TimePicker;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    TimePicker t1;
    Button b1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        t1=(TimePicker)findViewById(R.id.timepick);
        b1=(Button)findViewById(R.id.button);
    }
    public void display(View v)
    {
        Toast.makeText(this,"TIME 
SELECTED:"+t1.getHour()+":"+t1.getMinute(),Toast.LENGTH_LONG).show();
    }
}
```

Activity_Main.Xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <TimePicker
        android:id="@+id/timepick"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"></TimePicker>

    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="display"
        android:text="DISPLAY TIME" />


</LinearLayout>
```

OUTPUT:

RESULT:

Thus, an android program to demonstrate Date Picker and Time Picker is executed successfully.

EX. NO : 7 **Date:**	**MENU APPLICATION**

AIM:

To develop an simple android application with context and option menu.

PROCEDURE:

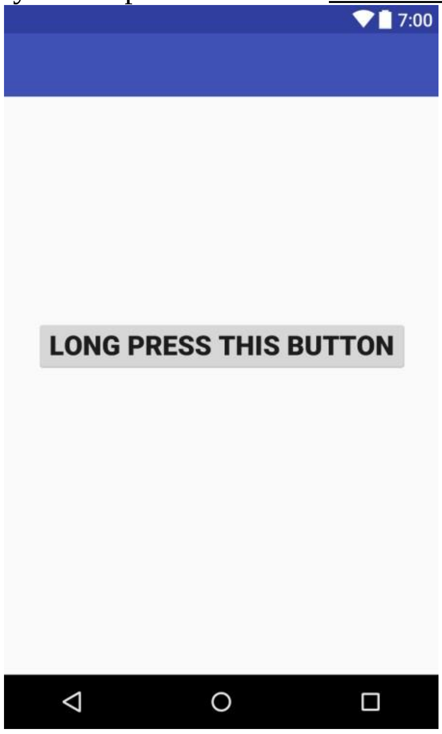
Menus are useful for displaying extra options that are not directly visible of the main UI of the application.

Types:

- Android Option Menu can be used for settings, search, delete etc..
- Android Context Menu appears when user press long click on the element. It is also known as floating menu Steps:

1. Drag a button to the design view.
2. Create activity_menu.xml under *src/main/res/menu directory*
3. Create the required menu item in the activity_menu.xml.
4. The event handling on clicking the menu item is performed in the MainActivity.java

5. Run the application to display the output in emulator <u>UI DESIGN</u>



PROGRAM:

<u>MainActivity.java</u>

```
public class MainActivity extends AppCompatActivity {

   @Override
   protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
Button btn = (Button) findViewById(R.id.button1);
btn.setOnCreateContextMenuListener(this);
   }
   @Override
   public boolean onCreateOptionsMenu(Menu menu) {
getMenuInflater().inflate(R.menu.activity_menu, menu);
return true;     }
   @Override
   public boolean onOptionsItemSelected(MenuItem item)
   {
      Toast.makeText(this, item.getTitle(),
Toast.LENGTH_LONG).show();         return true;     }
   @Override
   public void onCreateContextMenu(ContextMenu menu, View view,
ContextMenu.ContextMenuInfo menuInfo)
   {
      getMenuInflater().inflate(R.menu.activity_menu, menu);
   }
   @Override
   public boolean onContextItemSelected(MenuItem item)
   {
      Toast.makeText(this, item.getTitle(), Toast.LENGTH_LONG).show();
return true;

}
}
```

activity_menu.xml (*src/main/res/menu/activity_main*)

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
   <item
      android:id="@+id/menu1"
      android:icon="@mipmap/ic_launcher_round"
android:title="Item 1">
      <menu>
<item
            android:id="@+id/menu1.1"
android:title="Item 1.1"/>
         <item
               android:id="@+id/menu1.1"
android:title="Item 1.2"/>
      </menu>
   </item>
   </menu>
```

OUTPUT:

RESULT:

Thus, a simple android application with context and option menu is developed and executed successfully.

EX. NO : 8 Date:	SMS APPLICATION

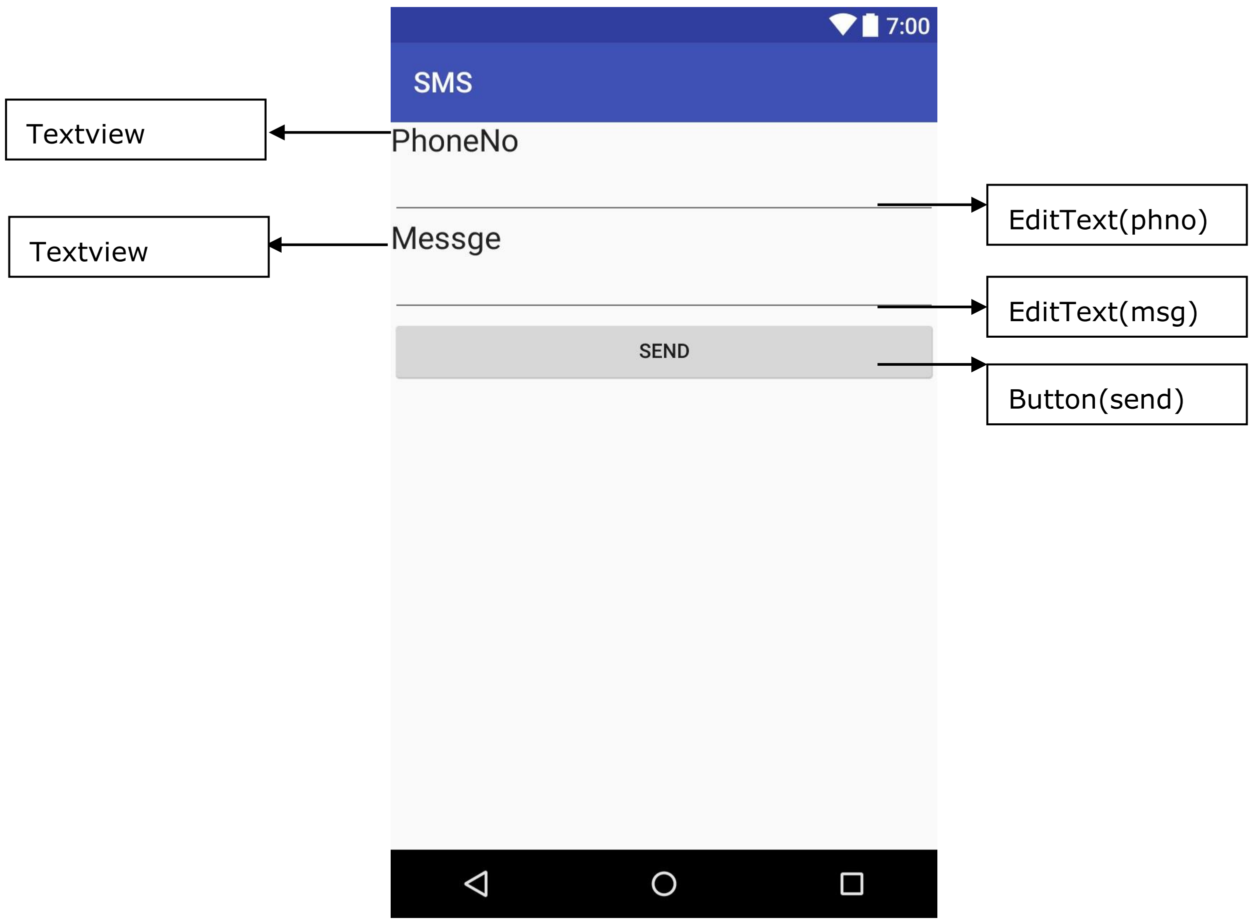
AIM:

To develop an android application to send SMS.

PROCEDURE:

1. Create new Android Application.
2. Open Activity_main.xml. Add 2 EditText view and a button to it.
3. Add the required permission to send SMS in AndroidManifest.java
4. Open MainActivity.java.
5. SMSManager API is used to send SMS.
6. Run the application to display the output.

UI DESIGN



PROGRAM:

MainActivity.java

```
package com.example.balashanmugam.ex8sms;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
Button send;
EditText phone_Number,message;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      send=(Button)findViewById(R.id.button);
      phone_Number=(EditText)findViewById(R.id.editText);
      message=(EditText)findViewById(R.id.editText2);
   }
   public void sendSMS(View v){
      String phone_Num=phone_Number.getText().toString();
      String send_msg=message.getText().toString();
      SmsManager sms=SmsManager.getDefault();
      sms.sendTextMessage(phone_Num,null,send_msg,null,null);
      Toast.makeText(getApplicationContext(),"SMS Sent!",Toast.LENGTH_LONG).show();
   }
}
```

Android.Manifest File

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

Activity_Main.Xml File

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
   android:onClick="sendSMS"
   android:orientation="vertical"
   tools:context=".MainActivity">

   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="@string/phone_no"
      android:textSize="25dp"
      app:layout_constraintBottom_toBottomOf="parent"
      app:layout_constraintLeft_toLeftOf="parent"
      app:layout_constraintRight_toRightOf="parent"
      app:layout_constraintTop_toTopOf="parent" />

   <EditText
      android:id="@+id/editText"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
```

```
        android:ems="10"
        android:inputType="textPersonName" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Message"
        android:textSize="25dp" />

    <EditText
        android:id="@+id/editText2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPersonName" />

    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="sendSMS"
        android:text="send" />

</LinearLayout>
```

<u>AndroidManifest.xml</u>

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

OUTPUT:

RESULT:

Thus, an android application to send SMS is developed and executed successfully.

EX. NO : 9 **Date:**	**CONTACTS APP**

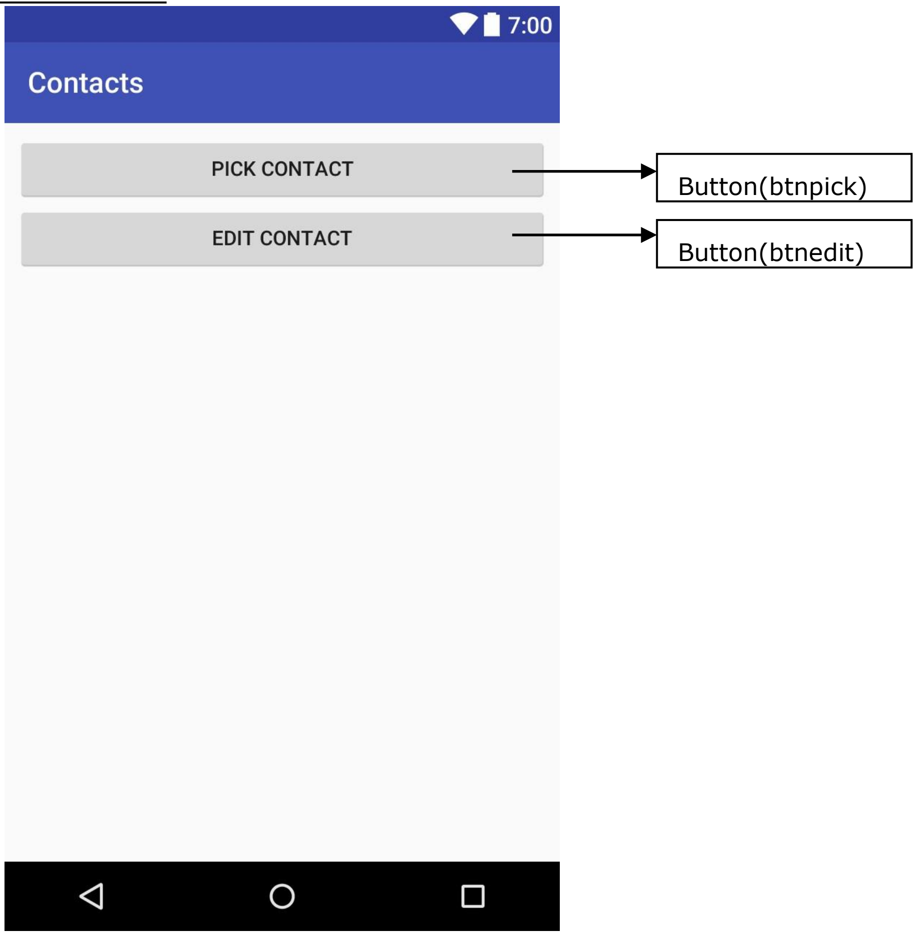
AIM:

To develop an android program to edit, view contacts.

PROCEDURE:

1. Create new Android Application.
2. Open Activity_main.xml.
3. Add 2 button to it.
4. Add the required permission to read and write contacts in AndroidManifest.java
5. Open MainActivity.java.
6. With Intent, an object is created which calls the Contact Manager to edit and view the contact.
7. Run the application to display the output.

UI DESIGN:



PROGRAM:

Activity_Main.Xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
   android:orientation="vertical"
   tools:context=".MainActivity">
```

```
    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Pick contact"
        tools:layout_editor_absoluteX="148dp"
        tools:layout_editor_absoluteY="30dp"
        android:onClick="pickContact"/>
</LinearLayout>
```

Mainactivity.Java

```
import android.content.ContentUris;
import android.content.Intent;
import android.provider.ContactsContract;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;

import java.net.DatagramPacket;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    public void pickContact(View v){
        Intent pick=new
Intent(Intent.ACTION_PICK,ContactsContract.CommonDataKinds.Phone.CONTENT_URI);
        startActivity(pick);
    }
}
```

AndroidManifest.xml

```
<uses-permission android:name="android.permission.READ_CONTACTS"/>
<uses-permission android:name="android.permission.WRITE_CONTACTS"/>
```

OUTPUT:

RESULT:

Thus, an android program to edit, view contacts is developed and executed successfully.

EX. NO : 10 **Date:**	**E-MAIL APP**

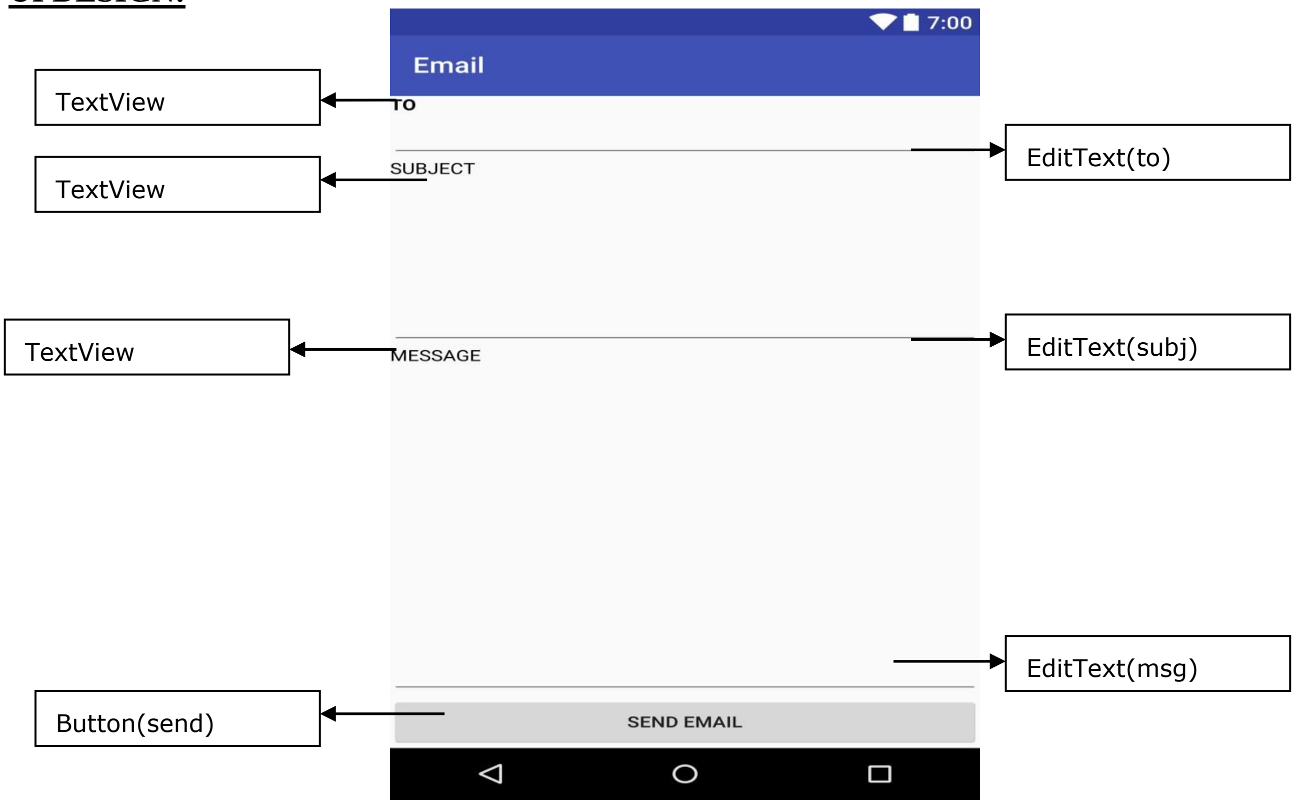
AIM:

To develop an android program to send e-mail.

PROCEDURE:

1. Create a new Android Application
2. Open activity_main.xml
3. Drag 2 EditTexts, 1 MultiLine EditText, 3 TextViews and 1 Button from the palette to the design part of xml file.
4. In MainActivity.java, create an intent to send the email.
5. The intent Object "ACTION_SEND" is used to launch an email client installed on the android device.
6. The intent Object "setData()" is used to specify the URI and "setType()" is set to message/rfc822.
7. The intent Object "EXTRA" is used to add TO, SUBJECT, CC, etc..
8. Run the application to launch Android emulator.

UI DESIGN:



PROGRAM:

MainActivity.java

```
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;
```

```
public class MainActivity extends AppCompatActivity {
EditText e_to,e_subj,e_msg;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        e_to=(EditText)findViewById(R.id.editText);
        e_subj=(EditText)findViewById(R.id.editText2);
        e_msg=(EditText)findViewById(R.id.editText3);
    }
    public void sendmail(View view)
    {
        String to=e_to.getText().toString();
        String subj=e_subj.getText().toString();
        String msg=e_msg.getText().toString();
        Intent emailApp=new Intent(Intent.ACTION_SEND);
        emailApp.putExtra(Intent.EXTRA_EMAIL,new String[]{to});
        emailApp.putExtra(Intent.EXTRA_SUBJECT,subj);
        emailApp.putExtra(Intent.EXTRA_TEXT,msg);
        emailApp.setType("message/rfc822");
        try {
            startActivity(Intent.createChooser(emailApp,"Send mail..."));
        }
        catch (android.content.ActivityNotFoundException ex){
            Toast.makeText(MainActivity.this,"There are no email clients
installed.",Toast.LENGTH_LONG).show();
        }
        }
    }
```

Activity_Main.Xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/to" />

    <EditText
        android:id="@+id/editText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPersonName" />
    <TextView
        android:id="@+id/textView2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/subject" />

    <EditText
        android:id="@+id/editText2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPersonName" />

    <TextView
        android:id="@+id/textView3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/msg" />

    <EditText
        android:id="@+id/editText3"
        android:layout_width="match_parent"
        android:layout_height="277dp"
        android:ems="10"
        android:inputType="textMultiLine" />

    <Button
        android:id="@+id/button2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="sendmail"
        android:text="@string/send" />

</LinearLayout>
```

AndroidManifest.xml

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

OUTPUT:

RESULT:

Thus, an android program to send e-mail is developed and executed successfully.

EX. NO : 11 **Date:**	**SERVICE APP**

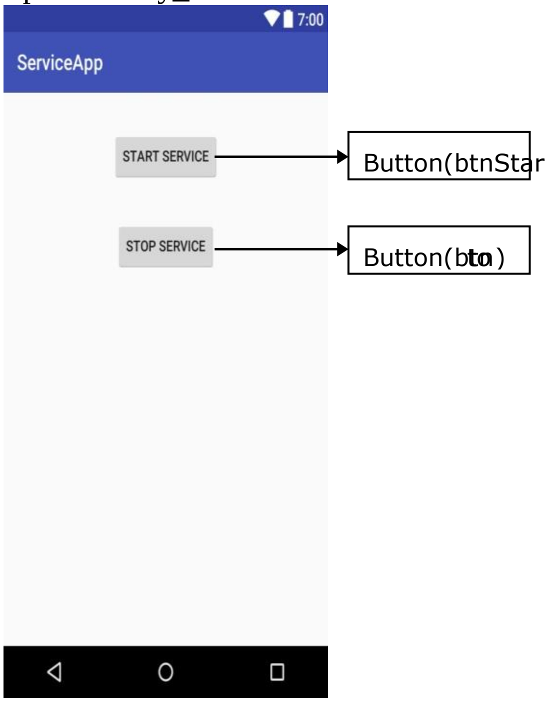
AIM:

To develop an android program to demonstrate a service.

PROCEDURE:

Android service is a component that is used to perform operations on the background such as playing music, handle network transactions, interacting content providers etc. It doesn't has any UI (user interface).

1. Create new Android Application.
2. Open activity_main.xml and add two buttons as the UI shown below:



3. Add the required permission in AndroidManifest.xml
4. In MainActivity.java, Create onclick event for the two buttons. These events start and stops service respectively.
5. The service calls method in myService.java and displays a toast message.

PROGRAM:

Activity_Main.Xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
            android:onClick="StartService"
            android:text="START SERVICE"
            tools:layout_editor_absoluteX="123dp"
            tools:layout_editor_absoluteY="172dp" />
        <Button
            android:id="@+id/button2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:onClick="StopService"
            android:text="STOP SERVICE"
            tools:layout_editor_absoluteX="141dp"
            tools:layout_editor_absoluteY="492dp" />
    </LinearLayout>
```

Main_Activity.Java

```
package com.example.ex11;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.Button;
import android.content.Intent;
import android.view.View;
import android.app.Activity;
import android.util.Log;
public class MainActivity extends AppCompatActivity {
Button start,stop;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        start = (Button) findViewById(R.id.button);
        stop = (Button) findViewById(R.id.button2);
    }
    public void StartService(View v) {
        startService(new Intent(getBaseContext(), MyService.class));
    }
    public void StopService(View v) {
        stopService(new Intent(getBaseContext(), MyService.class));
    }

}
```

AndroidMainfest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.ex11">

    <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>
        <service android:name=".MyService"></service>
    </application>

</manifest>
```

MyService.Java

```
package com.example.ex11;
import android.os.IBinder;
import android.widget.Toast;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.Button;
import android.content.Intent;
import android.view.View;
import android.app.Service;
import android.support.annotation.Nullable;
import android.widget.Toast;

import static android.app.Service.START_STICKY;

public class MyService extends Service
{
    @Nullable
    @Override
    public IBinder onBind(Intent intent)
    {        return null;
    }
    @Override
    public int onStartCommand(Intent intent, int flags, int startId)
    {
        Toast.makeText(this,"Service Started",Toast.LENGTH_SHORT).show();
        return START_STICKY;
    }
    @Override
    public void onDestroy()
    {       MyService.super.onDestroy();
        Toast.makeText(this,"Service Destroyed",Toast.LENGTH_SHORT).show();
    }

}
```

OUTPUT:

RESULT:

Thus, an android program is developed to demonstrate a service.

EX. NO : 12 **Date:**	**WEB VIEW APP**

AIM:

To develop an android program to demonstrate web view to display website.

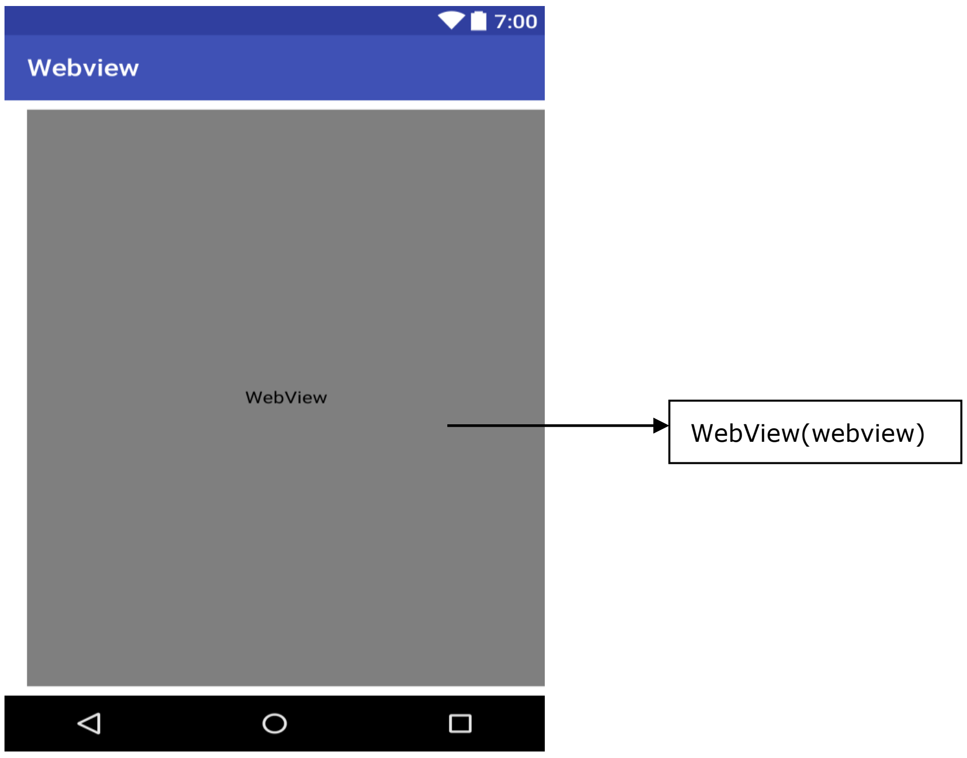
PROCEDURE:

- Create a new project in Android Studio and name it WebViewAp
- Open activity_main.xml, create the application interface and add webview element to it.
- Open src -> package -> MainActivity.java.
- Declare a webview variable, make JavaScript enable and load the URL of the website
- Open AndroidManifest.xml file and add internet permission to it just after the package name. It is required because the App will load data directly from the website

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

- To open the links in browser and not in application itself, the below line of code is added in MainActivity.java class.

  mywebView.setWebViewClient(new WebViewClient());



PROGRAM:

MainAcitivty.java

```
package com.example.ex12;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.webkit.WebViewClient;
```

```
public class MainActivity extends AppCompatActivity {
   private WebView wv1;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      wv1 = (WebView) findViewById(R.id.webview);
      WebSettings webSettings= wv1.getSettings();
      webSettings.setJavaScriptEnabled(true);
      wv1.loadUrl("http://gmail.com/");
      wv1.setWebViewClient(new WebViewClient());
   }
}
```

AndroidManifest.xml

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

OUTPUT:

RESULT:

Thus, an android program is developed to demonstrate web view to display website.

EX. NO : 13 **Date:**	**MAP VIEW APP**

AIM:

To develop an android program to display map of given location/position using map view

PROCEDURE:

1. Open android studio and create a new android project.
2. Select google maps activity from the predefined templates.
3. Click next -> and finish
4. Copy the console url for api key "google_maps_api.xml"
5. Copy this and paste it to the browser.
6. Click on continue and click on create api key.
7. Copy the key & replace the "google_maps_key" string in the "google_maps_api.xml" file
8. Add the required permission in "AndroidManifest.xml"
9. Change the latitude and longitude values of the location in "mapsactivity.java"
10. Execute the project and display the output.

PROGRAM:

AndroidManifest.xml

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

MapsActivity.java

```
public class MapsActivity extends FragmentActivity implements OnMapReadyCallback {
private GoogleMap mMap;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);        setContentView(R.layout.activity_maps);
      SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
            .findFragmentById(R.id.map);
      mapFragment.getMapAsync(this);
   }
public void onMapReady(GoogleMap googleMap) {
mMap = googleMap;
 LatLng pos = new LatLng(13.0708988,80.2214289);
  mMap.addMarker(new MarkerOptions().position(pos).title("Murugappa"));
mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(pos,25));
mMap.animateCamera(CameraUpdateFactory.zoomTo(12), 2000, null); }
```

OUTPUT:

RESULT:

Thus, an android program is developed to display map of given location/position using map view

EX. NO : 14 **Date:**	**INTENT CLASS**

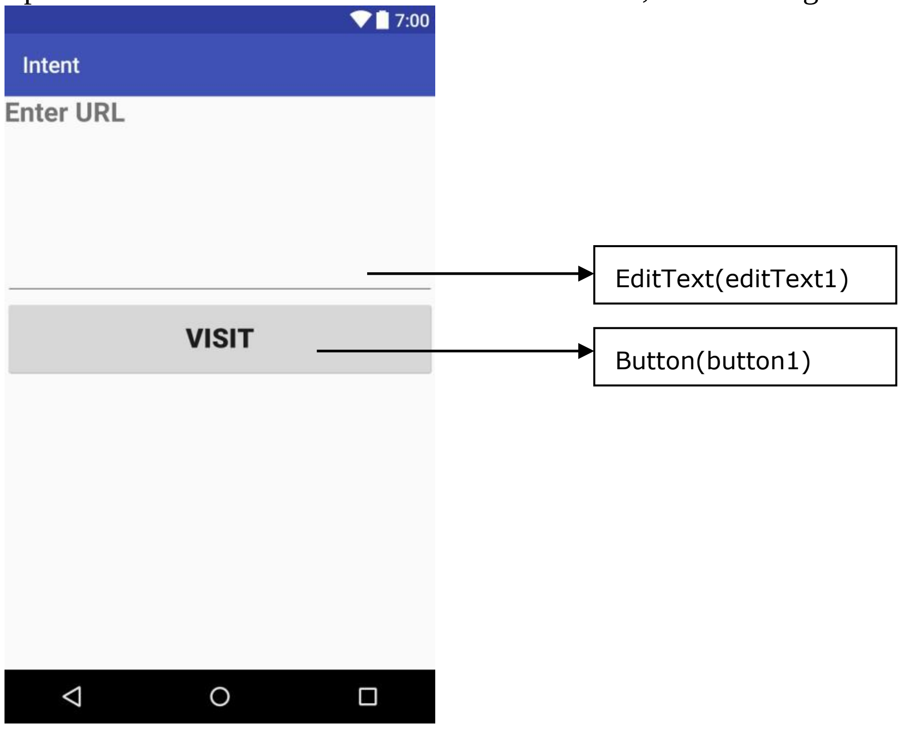
AIM:

To develop an android program to demonstrate the application of intent class.

PROCEDURE:

- Intents are used for communicating between the Application components and it also provides the connectivity between two apps.
- There are two types of intents in android: implicit and explicit.
  - Implicit Intent doesn't specifiy the component. In such case, intent provides information of available components provided by the system that is to be invoked
  - Explicit Intent specifies the component. In such case, intent provides the external class to be invoked.
- Follow the below steps to create intent in android, which opens a webpage in the browser:

1. Open new Android Project.
2. Open the XML file and add EditText and a button, like the UI given below:



3. Implement onClick event for Implicit Button inside MainActivity.java
4. Run the program in the emulator.

PROGRAM:

<u>Main_Activity.java</u>

```
package com.example.ex14a;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.content.Intent;
import android.net.Uri;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
public class MainActivity extends AppCompatActivity {
    Button button;
    EditText editText;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        button = findViewById(R.id.button);
        editText =  findViewById(R.id.editText);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String url=editText.getText().toString();
                Intent intent=new Intent(Intent.ACTION_VIEW, Uri.parse(url));
                startActivity(intent);
            }
        });
    }}
```

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
android:orientation="vertical"
    tools:context=".MainActivity">
    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="url" />
    <EditText
        android:id="@+id/editText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:ems="10"
        android:inputType="textPersonName" />
    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="visit" />
</LinearLayout>
```

OUTPUT:

RESULT:

Thus, an android program is developed to demonstrate the application of intent class

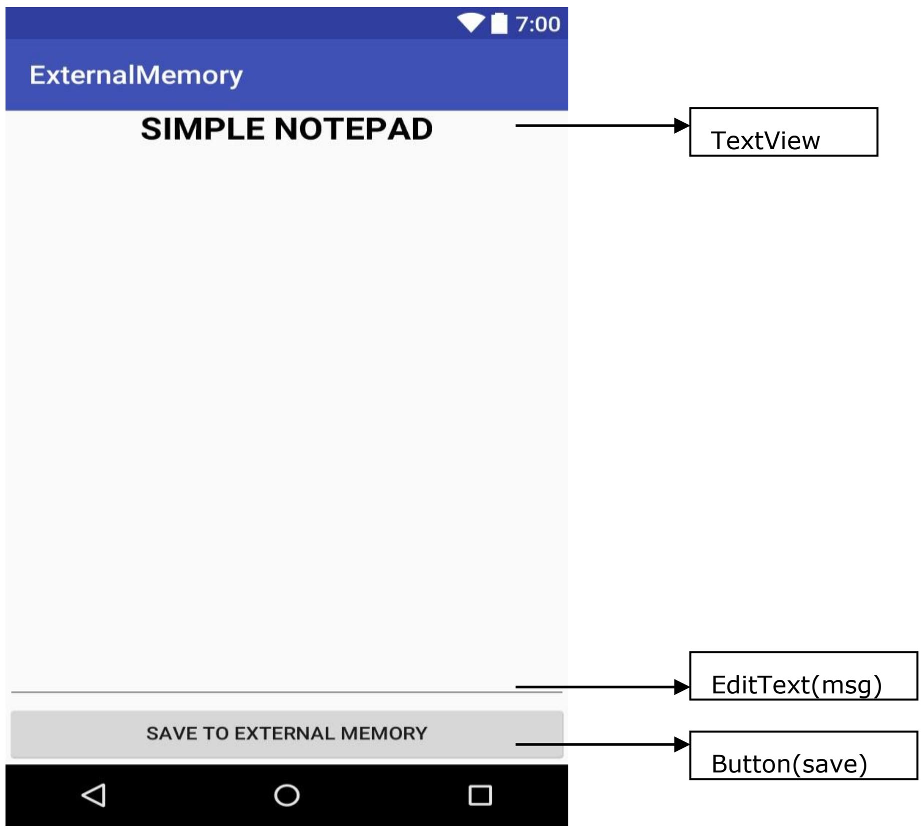
EX. NO : 15 **Date:**	**EXTERNAL MEMORY**

AIM:

To develop an android program to create a text file in a external memory.

PROCEDURE:

1. Create a new Android Application
2. Open activity_main.xml
3. Drag 1 EditTexts, 2 TextViews and 1 Button from the palette to the design part of xml file.
4. Add required permission to AndroidManifest.xml
5. Create a new folder, if not exists
6. Create a new text file and create an object for this file.
7. Append the text to this writer object
8. Close the object
9. Run the application to view the output in emulator



PROGRAM:

<u>AndroidManifest.xml</u>

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

<u>MainAcitivty.java</u>

```
package com.example.ex15;
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.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.io.FileWriter;
import android.os.Environment;
public class MainActivity extends AppCompatActivity {
    EditText msg;
    Button save;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        msg=(EditText)findViewById(R.id.editText3);
        save=(Button)findViewById(R.id.button);
    }
    public void save(View v)
    {       try {
        File folder = Environment.getExternalStorageDirectory();
        String path = folder.getAbsolutePath() + "/ppc.txt";

        File filepath = new File(path);
        FileWriter writer = new FileWriter(filepath);
        writer.append(msg.getText().toString());
        writer.flush();
        Toast.makeText(this, "file created", Toast.LENGTH_LONG).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: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"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="63dp"
        android:text="SIMPLE NOTEPAD"
        android:textSize="30sp" />
    <EditText
        android:id="@+id/editText3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:baselineAligned="false"
        android:ems="10"
        android:inputType="textPersonName" />
    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="save"
        android:text="SAVE" />

</LinearLayout>
```

Androidmainfest.Xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.ex15">
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <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>

</manifest>
```

OUTPUT:

RESULT:

Thus, an android program is developed to create a text file in an external memory.