TYBSC IT SEMESTER 6

Advanced Mobile Programming

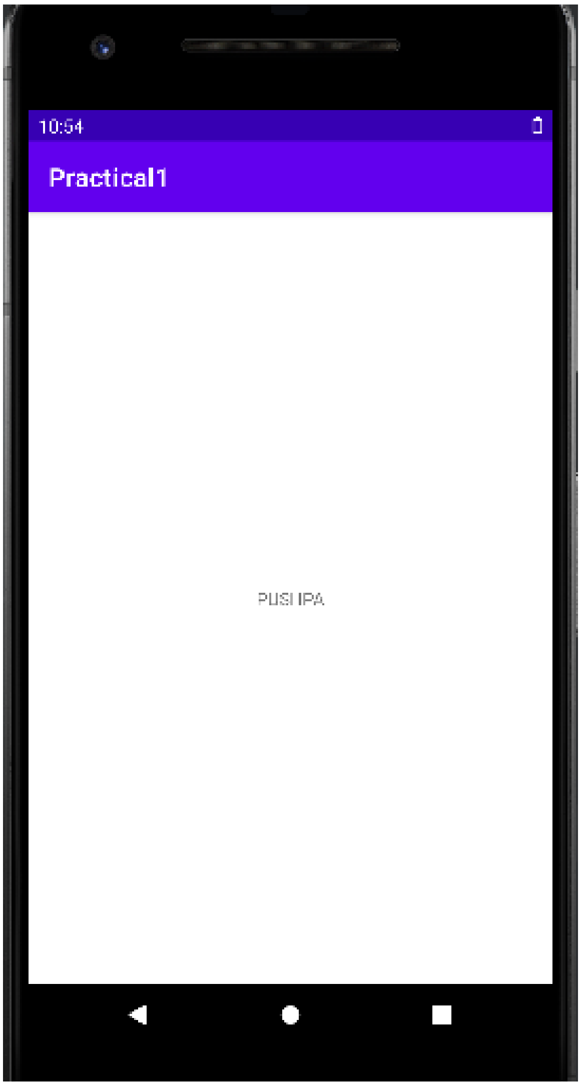
Practical No.:1

Introduction of Android Studio

Aim: Download and install Android studio. Setup SDK manager and create virtual device. Create android application to display your name.

activity_main.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="PUSHPA"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```



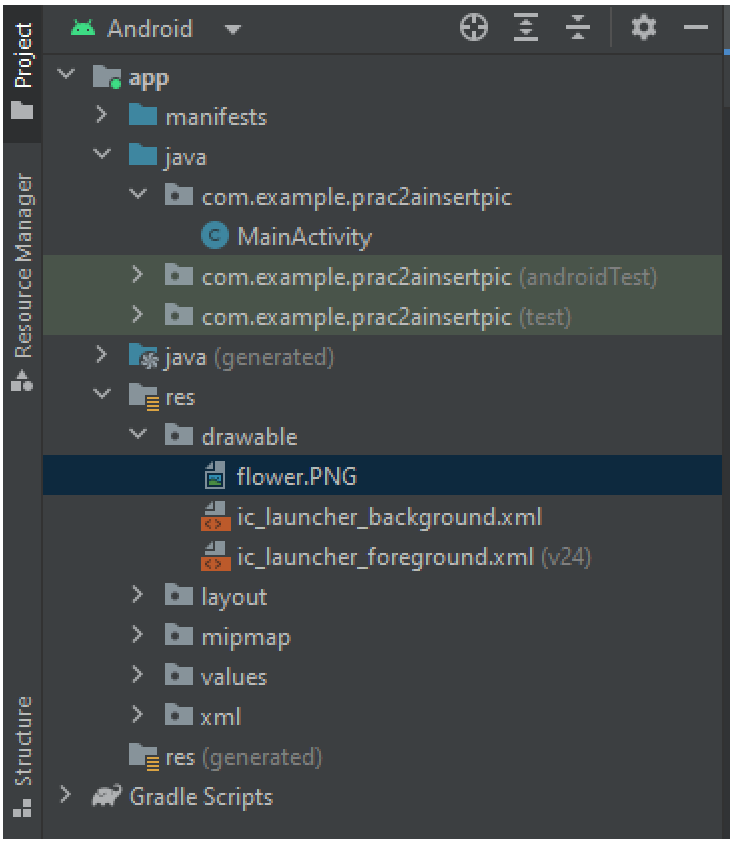
PUSHPA

Practical No. 2

Programming Resources

Aim: Create an android application with student name and insert one picture.

Step 1: Download any picture from google in png then copy the image and paste in drawable

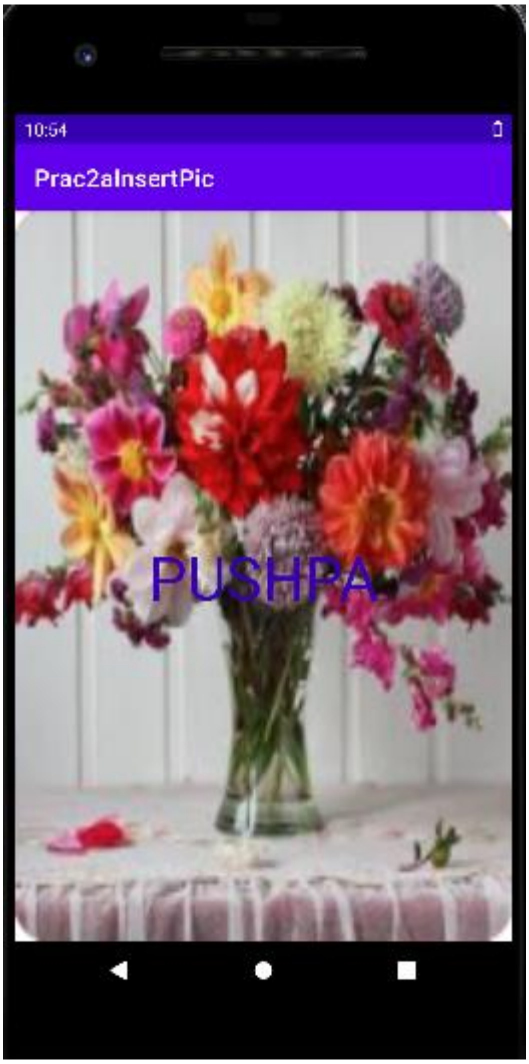

Android
Project
Resource Manager
Structure
app
manifests
java
com.example.prac2ainsertpic
MainActivity
com.example.prac2ainsertpic (androidTest)
com.example.prac2ainsertpic (test)
java (generated)
res
drawable
flower.PNG
ic_launcher_background.xml
ic_launcher_foreground.xml (v24)
layout
mipmap
values
xml
res (generated)
Gradle Scripts

Step 2:

activity_main.xml

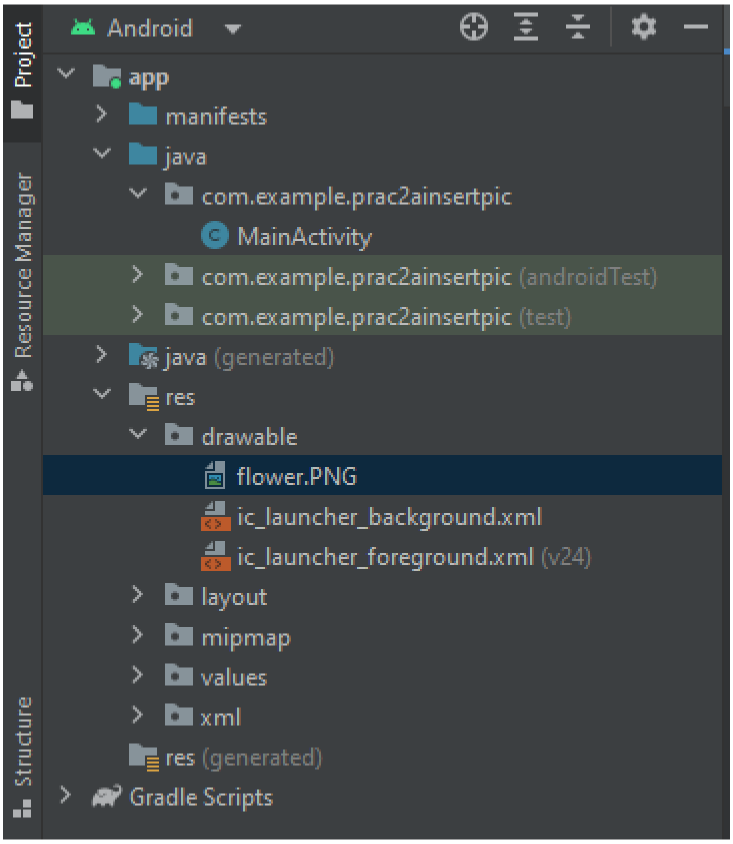
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/flower"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="PUSHPA"
        android:textColor="@color/purple_700"
        android:textSize="50dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Output:



Aim: Create an app with freedom fighter and its info

Step 1: Download any picture from google in png then copy the image and paste in drawable



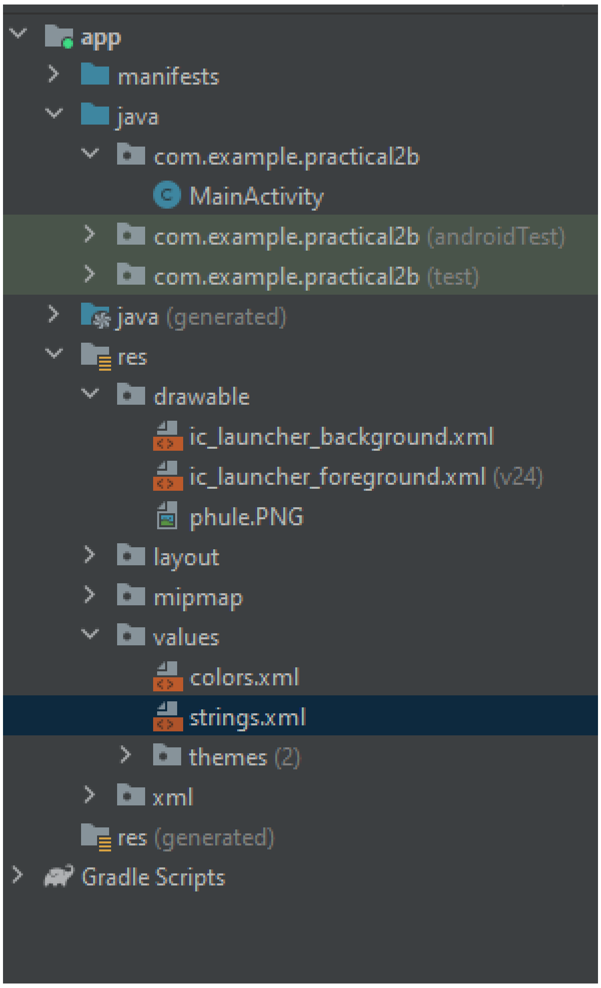
```
Android
app
  manifests
  java
    com.example.prac2ainsertpic
      MainActivity
    com.example.prac2ainsertpic (androidTest)
    com.example.prac2ainsertpic (test)
  java (generated)
  res
    drawable
      flower.PNG
      ic_launcher_background.xml
      ic_launcher_foreground.xml (v24)
    layout
    mipmap
    values
    xml
  res (generated)
Gradle Scripts
```

Step 2 : Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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"
android:background="#FBFBF0"
tools:context=".MainActivity">

<TextView
    android:id="@+id/Title"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/Title"
    android:textColor="#A9091C83"
    android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintVertical_bias="0.407"
    app:layout_constraintTop_toTopOf="parent" />
```

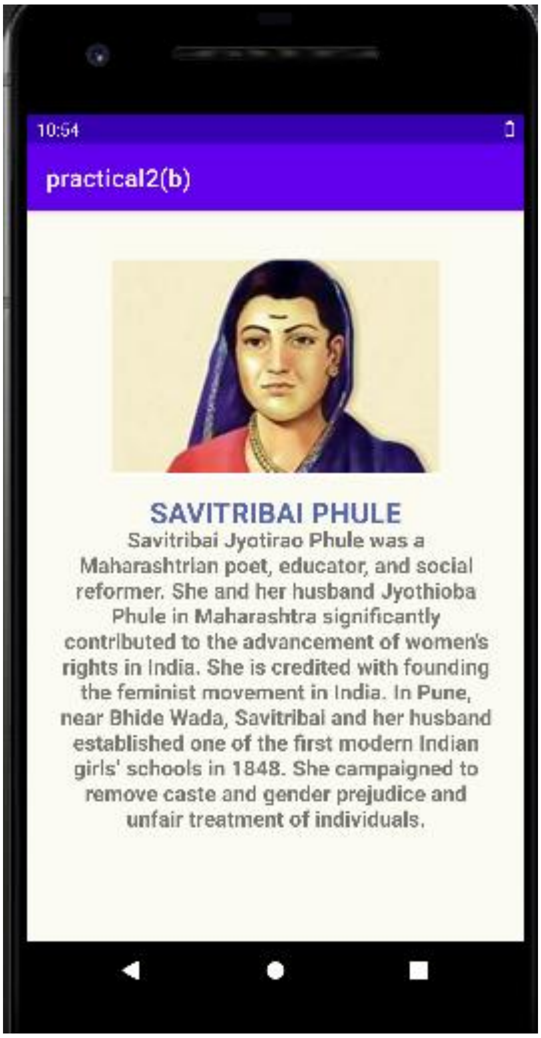
```
<ImageView
    android:id="@+id/imageView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="40dp"
    android:src="@drawable/phule"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
<TextView
    android:id="@+id/Info"
    android:layout_width="361dp"
    android:layout_height="389dp"
    android:text="@string/Info"
    android:textSize="18sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.506"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/Title"
    app:layout_constraintVertical_bias="0.121"
    android:gravity="center_horizontal"
    />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Step 3: Give Title and content



string.xml

```
resources>
    <string name="app_name">practical2(b)</string>
    <string name="Title">Savitribai Phule</string>
    <string name="Info">Savitribai Jyotirao Phule was a Maharashtrian poet,
educator, and social reformer. She and her husband Jyothioba Phule in
Maharashtra significantly contributed to the advancement of women’s rights in
India.
She is credited with founding the feminist movement in India. In Pune, near
Bhide Wada, Savitribai and her husband established one of the first modern
Indian girls’ schools in 1848.
She campaigned to remove caste and gender prejudice and unfair treatment of
individuals.</string>
</resources>
```



10:54

practical2(b)

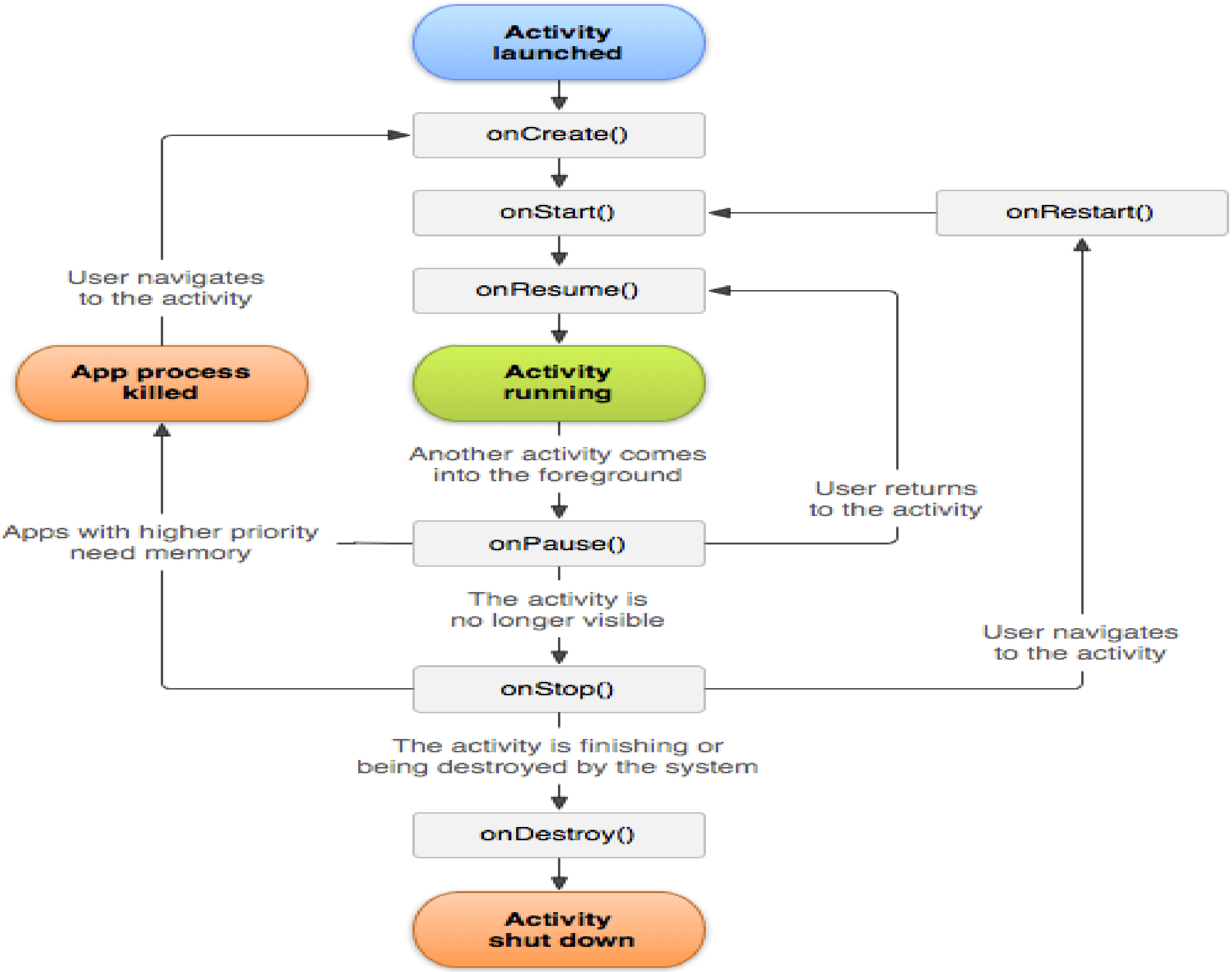
SAVITRIBAI PHULE

Savitribai Jyotirao Phule was a Maharashtrian poet, educator, and social reformer. She and her husband Jyothioba Phule in Maharashtra significantly contributed to the advancement of women's rights in India. She is credited with founding the feminist movement in India. In Pune, near Bhide Wada, Savitribai and her husband established one of the first modern Indian girls' schools in 1848. She campaigned to remove caste and gender prejudice and unfair treatment of individuals.

Practical 3:

Programming Activities

Activity Lifecycle:


Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

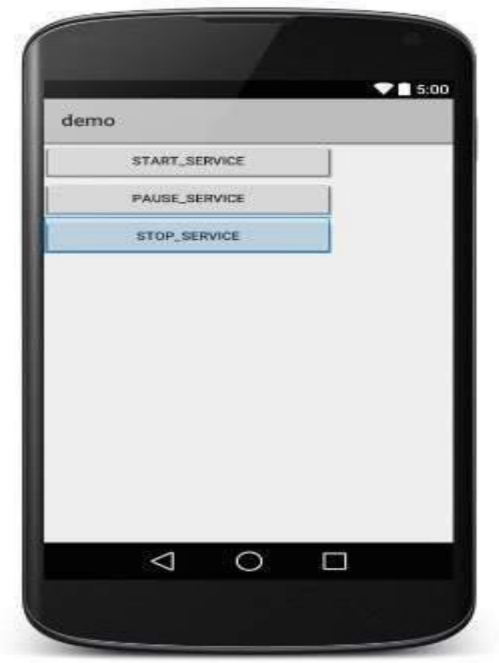
MainActivity.java

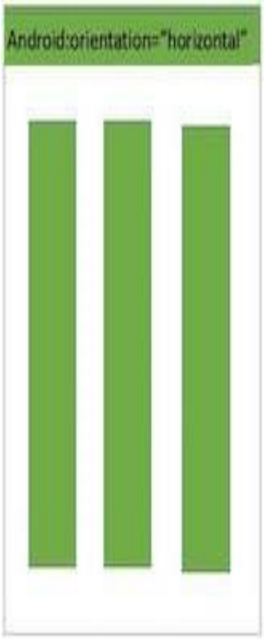
```
package com.example.myapplication;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Toast.makeText(getApplicationContext(), "onCreate
Called", Toast.LENGTH_LONG).show();
    }
    protected void onStart() {
        super.onStart();
        Toast.makeText(getApplicationContext(), "onStart
Called", Toast.LENGTH_LONG).show();
    }

    @Override
    protected void onRestart() {
        super.onRestart();
        Toast.makeText(getApplicationContext(), "onRestart
Called", Toast.LENGTH_LONG).show();
    }

    protected void onPause() {
        super.onPause();
        Toast.makeText(getApplicationContext(), "onPause
Called", Toast.LENGTH_LONG).show();
    }

    protected void onResume() {
        super.onResume();
        Toast.makeText(getApplicationContext(), "onResume
Called", Toast.LENGTH_LONG).show();
    }

    protected void onStop() {
        super.onStop();
        Toast.makeText(getApplicationContext(), "onStop
Called", Toast.LENGTH_LONG).show();
    }

    protected void onDestroy() {
        super.onDestroy();
        Toast.makeText(getApplicationContext(), "onDestroy
Called", Toast.LENGTH_LONG).show();
    }
}
```

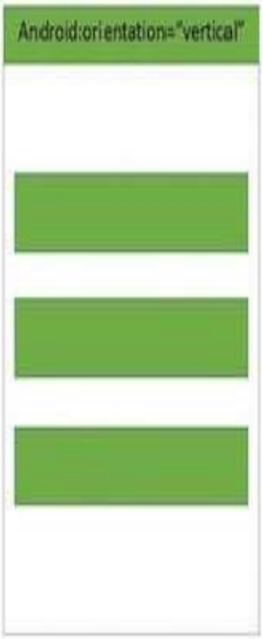
Practical No: 4

Android Layouts: Linear, Relative

Android Linear Layout:





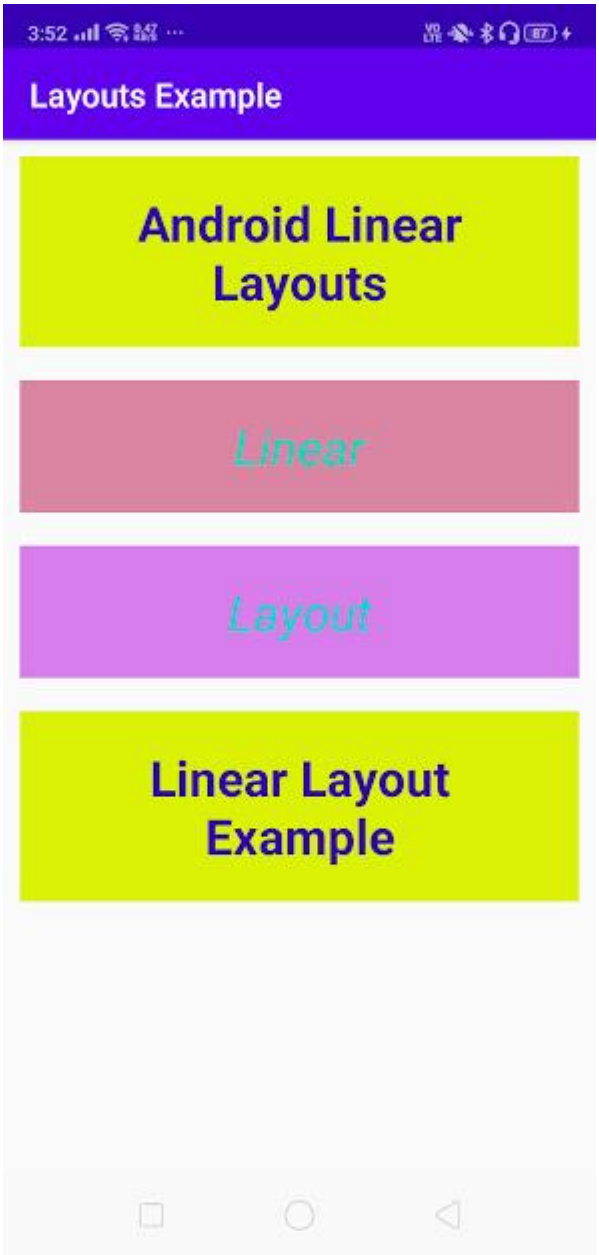


activity_main.xml

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

  <Button
    android:id="@+id/btnStartService"
    android:layout_width="270dp"
    android:layout_height="wrap_content"
    android:text="start_service"/>

  <Button
    android:id="@+id/btnPauseService"
    android:layout_width="270dp"
    android:layout_height="wrap_content"
    android:text="pause_service"/>
  <Button
    android:id="@+id/btnStopService"
    android:layout_width="270dp"
    android:layout_height="wrap_content"
    android:text="stop_service"/>

</LinearLayout>
```

Example 2:

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

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/coloryellow"
        android:layout_margin="10dp"
        android:layout_marginTop="30dp"
        android:padding="20dp"
        android:text="Android Linear Layouts"
        android:textAlignment="center"
        android:textColor="@color/colorPrimaryDark"
        android:textSize="30sp"
        android:textStyle="bold" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:background="@color/colorpink"
        android:padding="20dp"
        android:text="Linear"
        android:textAlignment="center"
        android:textColor="@color/colorAccent"
        android:textSize="30sp"
        android:textStyle="italic" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:padding="20dp"
        android:background="@color/colorpurple"
        android:text="Layout"
        android:textAlignment="center"
        android:textColor="@color/colorAccent"
        android:textSize="30sp"
        android:textStyle="italic" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:padding="20dp"
        android:background="@color/coloryellow"
        android:text="Linear Layout Example"
        android:textAlignment="center"
        android:textColor="@color/colorPrimaryDark"
        android:textSize="30sp"
        android:textStyle="bold" />

</LinearLayout>
```



Android Linear Layouts

Linear

Layout

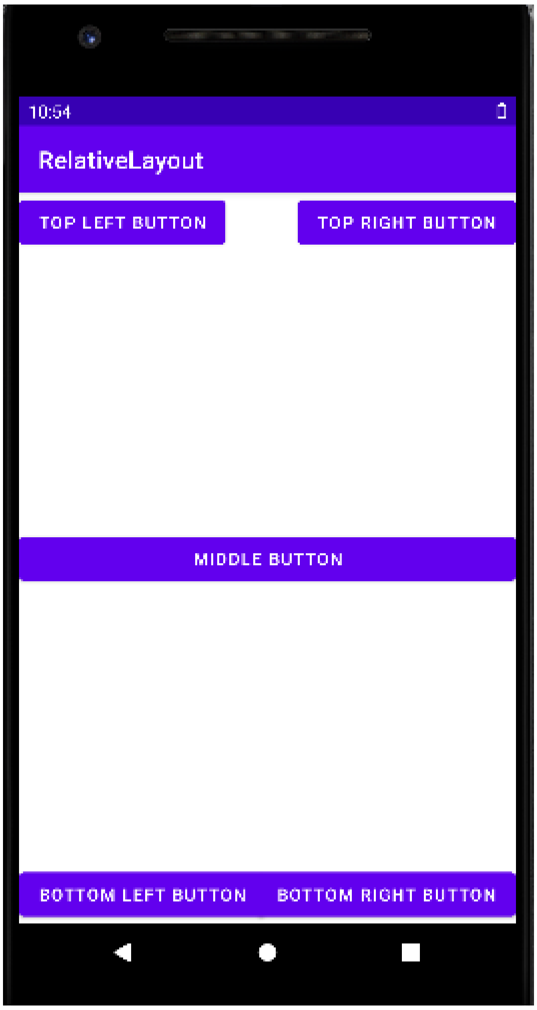
Linear Layout Example

Android Relative Layout

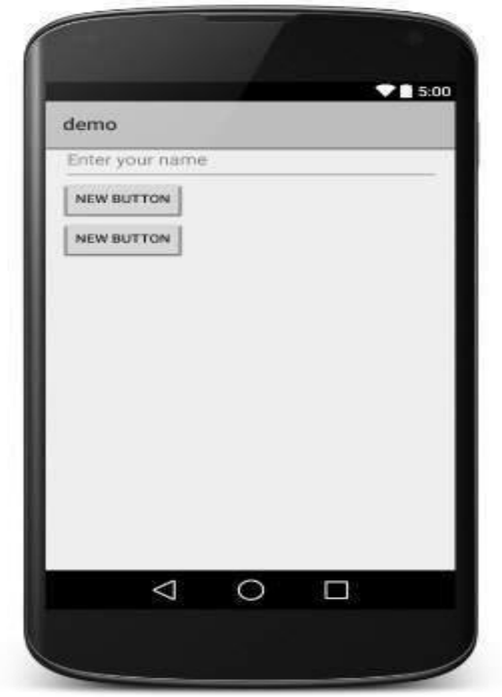
(1)

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

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Top Left Button"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"/>

    <Button
```

```
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Top Right Button"
        android:layout_alignParentTop="true"
        android:layout_alignParentRight="true"/>
    <Button
        android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Bottom Left Button"
        android:layout_alignParentLeft="true"
        android:layout_alignParentBottom="true"/>

    <Button
        android:id="@+id/button4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Bottom Right Button"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"/>
    <Button
        android:id="@+id/button5"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Middle Button"
        android:layout_centerVertical="true"
        android:layout_centerHorizontal="true"/>
</RelativeLayout>
```



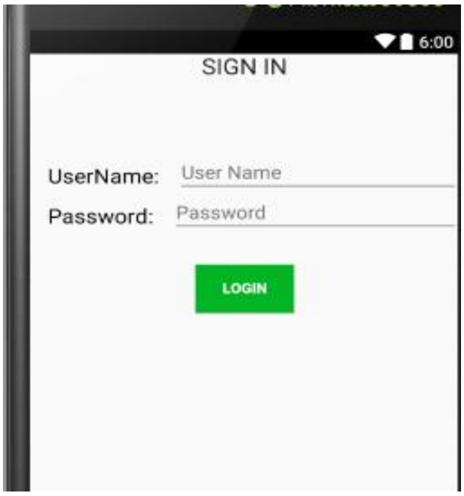
(2)



activity_main.xml

```
<RelativeLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:paddingLeft="16dp"
  android:paddingRight="16dp" >

  <EditText
    android:id="@+id/name"
    android:layout_width="wrap_parent"
    android:layout_height="wrap_content"
    android:hint="@string/name" />

  <LinearLayout
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_below="@+id/name">

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

    <Button
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Button 2"
      android:id="@+id/button2" />

  </LinearLayout>
</RelativeLayout>
```

(3)



```
<?xml version="1.0" encoding="utf-8"?>
<!--Relative Layout Is Used-->
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <!--Text View for Displaying SIGN IN Text At Top of UI-->
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="SIGN IN"
        android:id="@+id/textView3"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true" />
    <!--Text View for Displaying Username-->
    <TextView
        android:id="@+id/userName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="@dimen/activity_horizontal_margin"
        android:layout_marginTop="110dp"
        android:text="UserName:"
        android:textColor="#000000"
        android:textSize="20sp" />
```

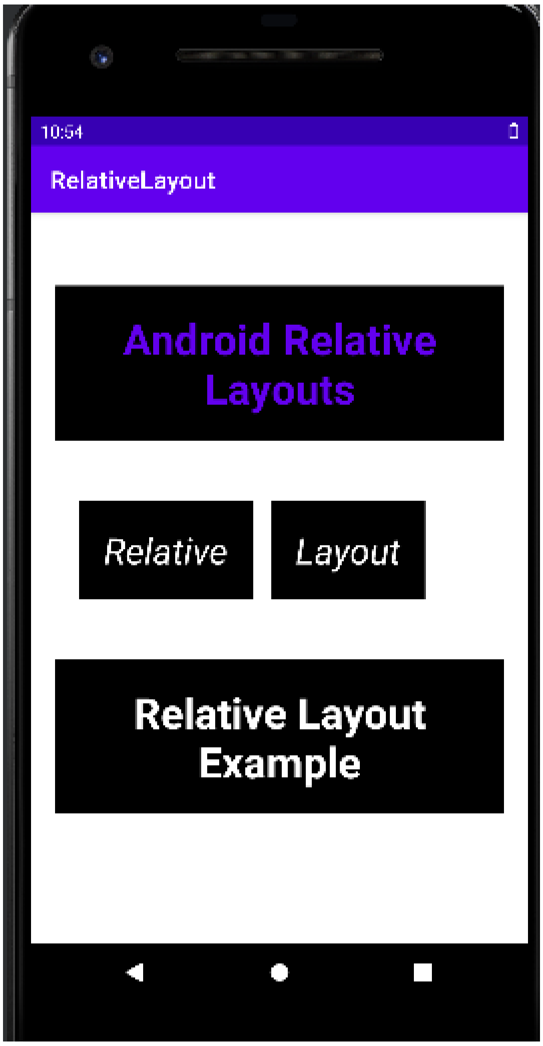
```
<!--Text View for Displaying Password-->
<TextView
    android:id="@+id/password"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/userName"
    android:layout_margin="@dimen/activity_horizontal_margin"
    android:text="Password:"
    android:textColor="#000000"
    android:textSize="20sp" />
<!--Edit Text for Filling Username-->
<EditText
    android:id="@+id/edt_userName"
    android:layout_width="fill_parent"
    android:layout_height="40dp"
    android:layout_marginLeft="@dimen/activity_horizontal_margin"
    android:layout_marginTop="100dp"
    android:layout_toRightOf="@+id/userName"
    android:hint="User Name" />
<!--Edit Text for Filling Password-->
<EditText
    android:layout_width="fill_parent"
    android:layout_height="40dp"
    android:layout_below="@+id/edt_userName"
    android:layout_centerVertical="true"
    android:layout_toRightOf="@+id/password"
    android:hint="Password" />
<!--Button for Clicking after filling details-->

<Button
    android:id="@+id/btnLogin"
    android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:layout_below="@+id/password"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="20dp"
        android:background="#03B424"
        android:text="Login"
        android:textColor="#ffffff"
        android:textStyle="bold" />
</RelativeLayout>
```

(4)

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
android:id="@+id/relative_layout"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_marginLeft="10dp"
android:layout_marginTop="10dp"
android:layout_marginRight="10dp"
android:layout_marginBottom="10dp"
android:layout_weight="1">

<TextView
    android:id="@+id/text_view"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_marginLeft="10dp"
    android:layout_marginTop="50dp"
    android:layout_marginRight="10dp"
    android:padding="20dp"
    android:background="@color/black"
    android:text="Android Relative Layouts"
    android:textAlignment="center"
    android:textColor="@color/design_default_color_primary"
    android:textSize="35sp"
    android:textStyle="bold" />

<TextView
```

```
    android:id="@+id/text_view1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/text_view"
    android:background="@color/design_default_color_on_secondary"
    android:layout_marginLeft="30dp"
    android:layout_marginTop="50dp"
    android:layout_marginRight="10dp"
    android:padding="20dp"
    android:text="Relative"
    android:textAlignment="center"
    android:textColor="@color/cardview_light_background"
    android:textSize="30sp"
    android:textStyle="italic" />

<TextView
    android:id="@+id/text_view2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/text_view"
    android:layout_marginLeft="5dp"
    android:layout_marginTop="50dp"
    android:layout_toRightOf="@id/text_view1"
    android:padding="20dp"
    android:background="@color/black"
    android:text="Layout"
    android:textAlignment="center"
    android:textColor="@color/cardview_light_background"
    android:textSize="30sp"
    android:textStyle="italic" />

<TextView
    android:id="@+id/text_view3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/text_view1"
    android:layout_centerInParent="true"
    android:layout_marginLeft="10dp"
    android:layout_marginTop="50dp"
    android:background="@color/black"
    android:layout_marginRight="10dp"
    android:padding="20dp"
    android:text="Relative Layout Example"
    android:textAlignment="center"
    android:textColor="@color/cardview_light_background"
    android:textSize="35sp"
    android:textStyle="bold" />
</RelativeLayout>
```



10:54

RelativeLayout

Android Relative Layouts

Relative

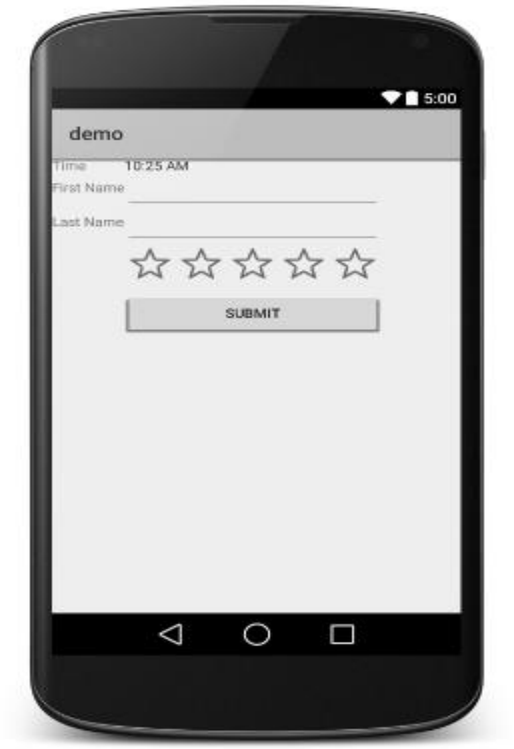
Layout

Relative Layout Example

Practical No: 5

Android Layouts: Table, Absolute, Frame, List View, Grid View.

Table Layout



5:00

demo

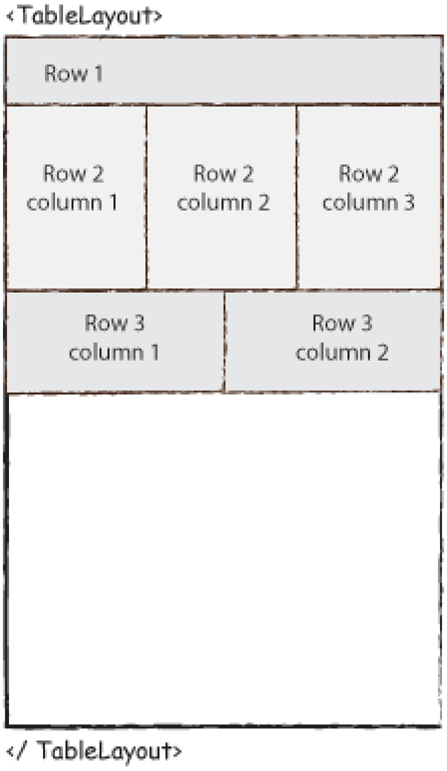
Time 10:25 AM

First Name

Last Name

☆ ☆ ☆ ☆ ☆

SUBMIT



<TableLayout>

Row 1

Row 2 column 1 | Row 2 column 2 | Row 2 column 3

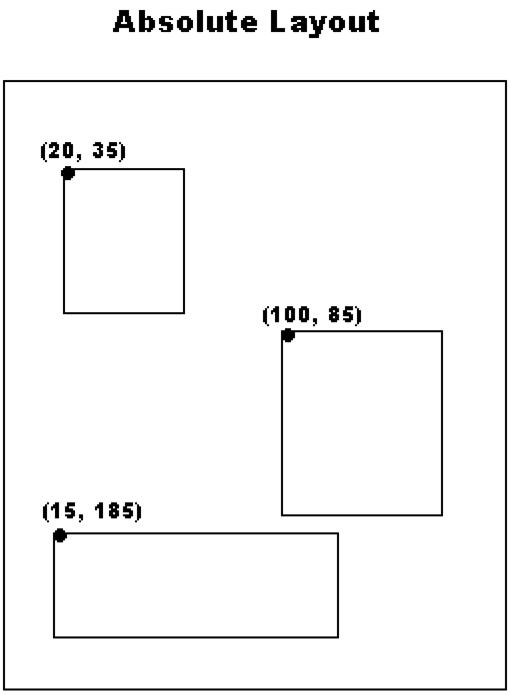
Row 3 column 1 | Row 3 column 2

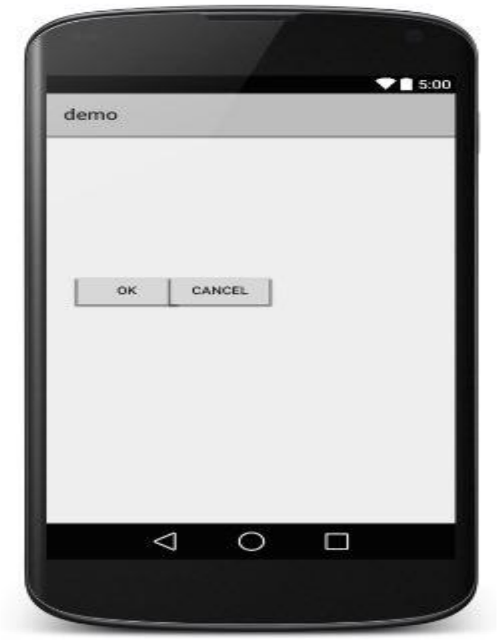
</ TableLayout>

```
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent">

  <TableRow
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <TextView
      android:text="Time"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_column="1" />

    <TextClock
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:id="@+id/textClock"
      android:layout_column="2" />

  </TableRow>

  <TableRow>
```

```
    <TextView
      android:text="First Name"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_column="1" />

    <EditText
      android:width="200px"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content" />
  </TableRow>


 <TableRow>

    <TextView
      android:text="Last Name"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_column="1" />

    <EditText
      android:width="100px"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content" />
  </TableRow>

  <TableRow
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <RatingBar
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:id="@+id/ratingBar"
      android:layout_column="2" />
  </TableRow>

  <TableRow
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"/>

  <TableRow
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <Button
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Submit"
      android:id="@+id/button"
      android:layout_column="2" />
  </TableRow>

</TableLayout>
```

Android Absolute Layout





activity_main.xml

```
<AbsoluteLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent">

  <Button
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:text="OK"
    android:layout_x="50px"
    android:layout_y="361px" />
  <Button
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:text="Cancel"
    android:layout_x="225px"
    android:layout_y="361px" />
```

```
</AbsoluteLayout>
```

List Layout

Step 1: Create a new project Listexample and activity Main Activity. Here we will create a ListView in LinearLayout

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

    <ListView
        android:id="@+id/simpleListView"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:divider="@color/material_blue_grey_800"
        android:dividerHeight="1dp" />
</LinearLayout>
```

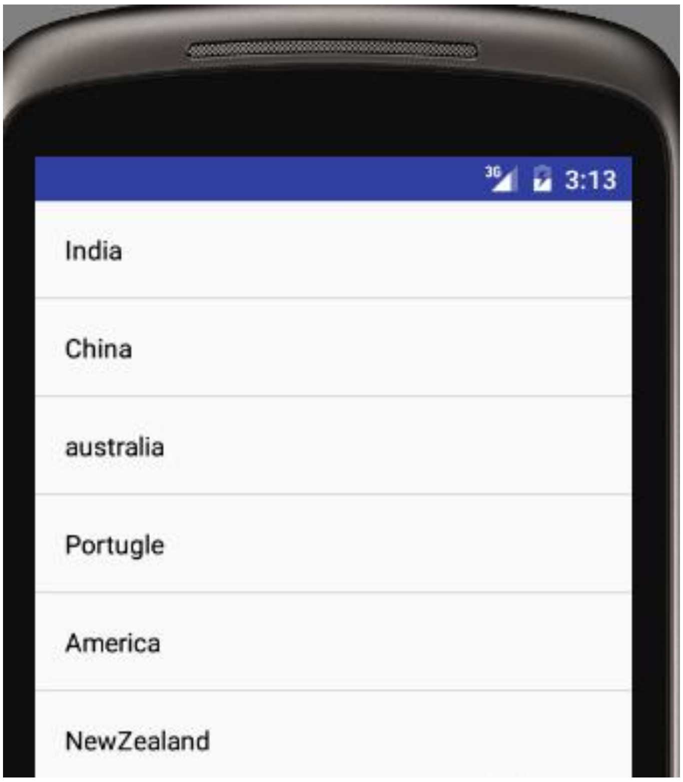
Step 2: Create a new activity name Listview

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

    <TextView
        android:id="@+id/textView"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:padding="2dp"
        android:textColor="@color/black" />
</LinearLayout>
```

Step 3: Now in this final step we will use ArrayAdapter to display the country names in UI. **Below is the code of MainActivity.java**

```
package com.example.list;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.widget.ArrayAdapter;
import android.widget.ListView;

public class MainActivity extends AppCompatActivity {

    ListView simpleList;
    String countryList[] = {"India", "China", "australia", "Portugle", "America",
"NewZealand"};

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        simpleList = (ListView)findViewById(R.id.simpleListView);
        ArrayAdapter<String> arrayAdapter = new ArrayAdapter<String>(this,
R.layout.listview, R.id.textView, countryList);
        simpleList.setAdapter(arrayAdapter);
    }
}
```

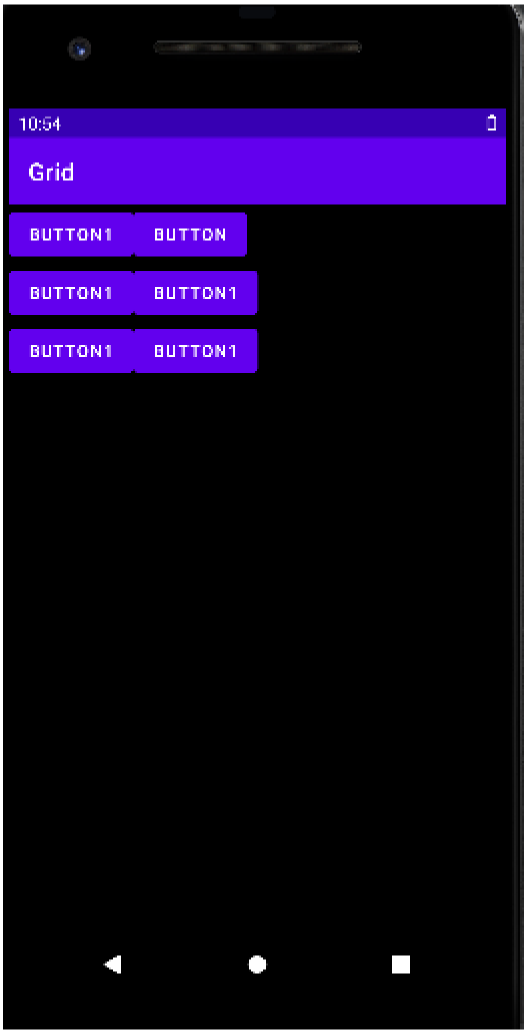
Output:



```
India
China
australia
Portugle
America
NewZealand
```

Grid Layout

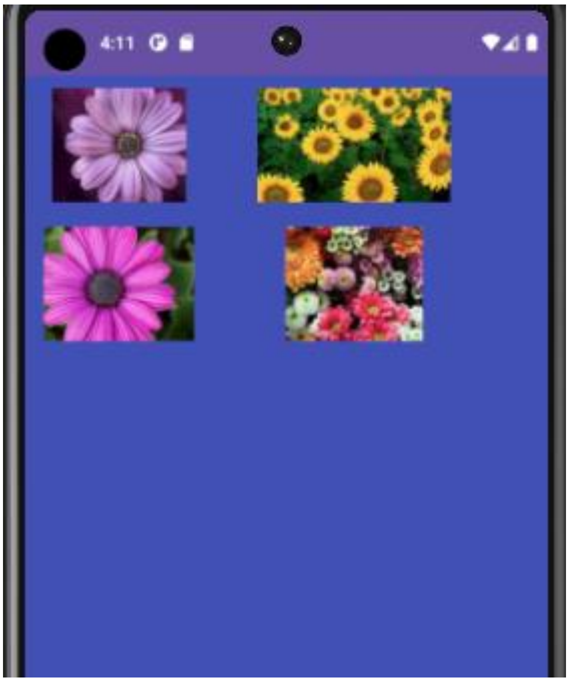
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/black"
    tools:context=".MainActivity">

    <GridLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10sp"
        android:rowCount="3"
        android:columnCount="2"
        tools:ignore="MissingConstraints">
        <Button
            android:id="@+id/button1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button1"/>
        <Button
            android:id="@+id/button2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button"/>
        <Button
            android:id="@+id/button3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button1"/>
        <Button
            android:id="@+id/button4"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button1"/>
        <Button
            android:id="@+id/button5"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button1"/>
        <Button
            android:id="@+id/button6"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
        android:text="Button1"/>
    </GridLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```



Example :

```
<?xml version="1.0" encoding="utf-8"?>
<GridLayout
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:rowCount="2"
android:columnCount="2"
android:background="#3F51B5"
tools:context=".MainActivity">

<ImageView
   android:layout_height="90dp"
   android:layout_width="130dp"
   android:layout_margin="10dp"
   android:src="@drawable/image_1"
   />
```

```
<ImageView
   android:layout_height="90dp"
   android:layout_width="200dp"
   android:layout_margin="10dp"
   android:src="@drawable/image_2"
   />

<ImageView
   android:layout_height="90dp"
   android:layout_width="130dp"
   android:layout_margin="10dp"
   android:src="@drawable/image_3"
   />

<ImageView
   android:layout_height="90dp"
   android:layout_width="200dp"
   android:layout_margin="10dp"
   android:src="@drawable/image_4"
   />


</GridLayout>
```



Practical No. 6

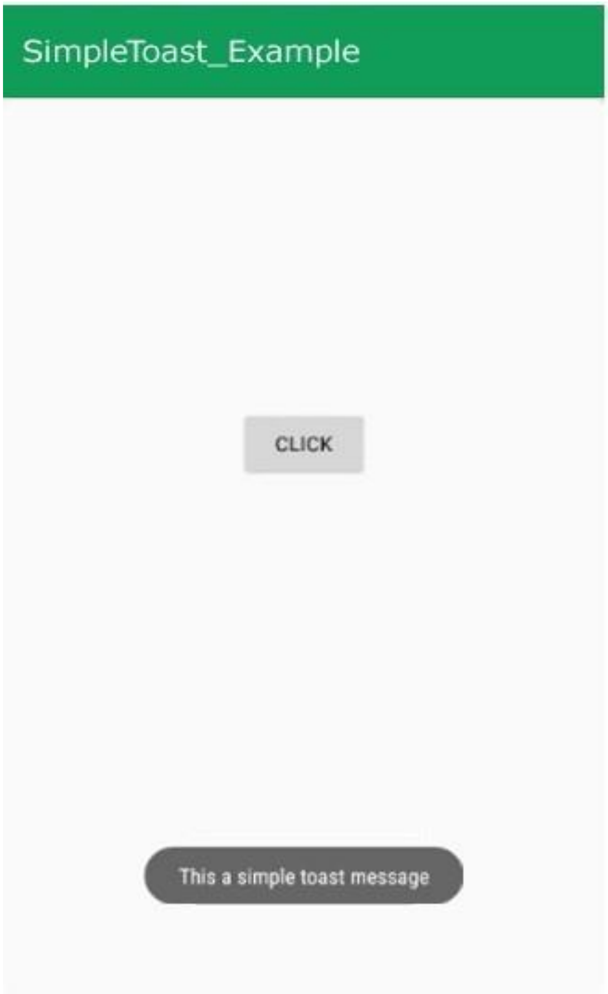
Android | What is Toast and How to use it with Examples

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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">

    <Button
        android:id="@+id/Button01"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_marginTop="209dp"
        android:onClick="onClick"
        android:text="Click"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        tools:ignore="OnClick" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.toast;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    // Defining the object for button
    Button btn;

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

        // Bind the components to their respective objects
        // by assigning their IDs
        // with the help of findViewById() method
        Button btn = (Button)findViewById(R.id.Button01);

        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v)
            {
                // Displaying simple Toast message
                Toast.makeText(getApplicationContext(),
                                "This a toast message",
                                Toast.LENGTH_LONG)
                        .show();
            }
        });
    }
}
```



This a simple toast message

Note:

1. To display the Toast at the center:

toast.setGravity(Gravity.CENTER, 0, 0);

2. To display the Toast at the top, centered horizontally:

toast.setGravity(Gravity.TOP | Gravity.CENTER_HORIZONTALLY, 0, 0);

3. To display the Toast at the top, centered horizontally, but 30 pixels down from the top:

toast.setGravity(Gravity.TOP | Gravity.CENTER_HORIZONTALLY, 0, 30);

4. To display the Toast at the bottom, rightmost horizontally:

toast.setGravity(Gravity.BOTTOM | Gravity.RIGHT, 0, 0);

Syntax:

```
Toast t = Toast.makeText(getApplicationContext(),
                         "This a positioned toast message",
                         Toast.LENGTH_LONG);
t.setGravity(Gravity.BOTTOM | Gravity.RIGHT, 0, 0);
t.show();
```

MainActivity_java

```
package com.example.toast;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Gravity;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    // Defining the object for button
    Button btn;

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // Bind the components to their respective objects
        // by assigning their IDs
        // with the help of findViewById() method
        Button btn = (Button)findViewById(R.id.Button01);

        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v)
            {
                // Displaying posotioned Toast message
                Toast t = Toast.makeText(getApplicationContext(),
                        "This a positioned toast message",
                        Toast.LENGTH_LONG);
                t.setGravity(Gravity.BOTTOM | Gravity.RIGHT, 0, 0);
                t.show();
            }
        });
    }
}
```

Android: Menus

Way to create menu directory and menu resource file:

1. To create the menu directory just right-click on **res** folder and navigate to **res->New->Android Resource Directory.** Give resource directory name as menu and resource type also menu. One directory will be created under res folder.

2. To create XML resource file simply right-click on menu folder and navigate to **New->Menu Resource File.** Give name of file as options_menu. One options_menu.xml will be created under menu folder.

Aim:

Create an android application for the following menu items, the appropriate toast should appear by clicking on the item :

- Search
- Upload
- Copy
- Print
- Share
- Bookmark

options_menu.xml

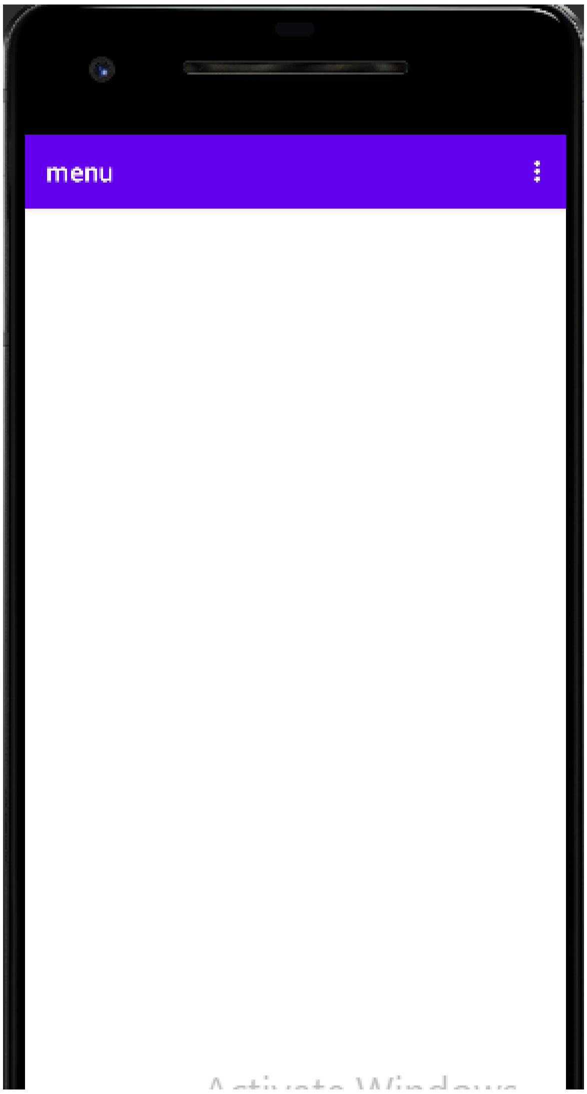
```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/search_item"
        android:title="Search" />

    <item android:id="@+id/upload_item"
        android:title="Upload" />

    <item android:id="@+id/copy_item"
        android:title="Copy" />

    <item android:id="@+id/print_item"
        android:title="Print" />

    <item android:id="@+id/share_item"
        android:title="Share" />

    <item android:id="@+id/bookmark_item"
        android:title="BookMark" />
</menu>
```

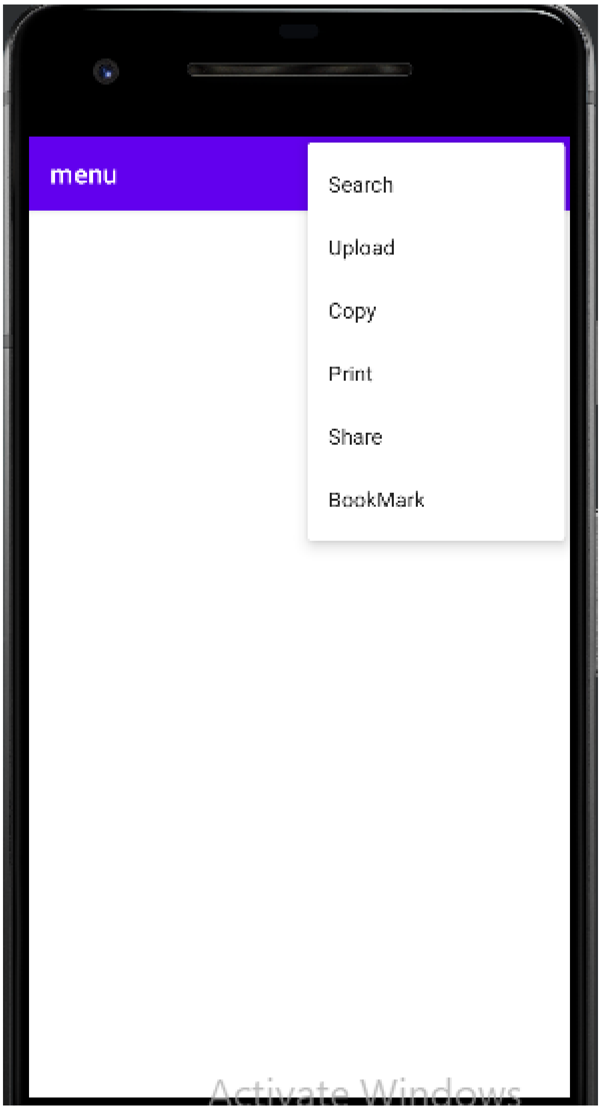
MainActivity.java

```
package com.example.menu;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater()
                .inflate(R.menu.options_menu, menu);
        return true;
    }
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        Toast.makeText(this, "Selected Item: " + item.getTitle(), Toast.LENGTH_SHORT)
```

```
                .show();
        switch (item.getItemId()) {
            case R.id.search_item:
                return true;

            case R.id.upload_item:
                return true;

            case R.id.copy_item:
                return true;

            case R.id.print_item:
                return true;

            case R.id.share_item:
                return true;

            case R.id.bookmark_item:
                return true;

            default:
                return super.onOptionsItemSelected(item);
        }
    }
}
```

output :


menu


menu
Search
Upload
Copy
Print
Share
BookMark

Aim:

Create an android application for the following menu items, the appropriate toast should appear by clicking on the item :

- message
- picture
- mode
- about
- calculator
- exit

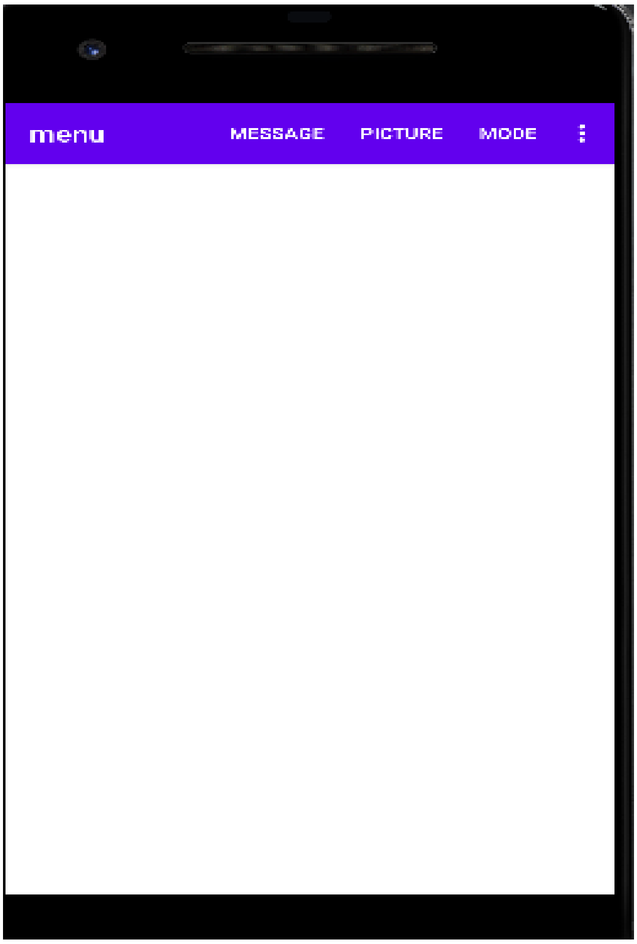
  (make calculator and exit items disabled)

options_menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu
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"
tools:context=".MainActivity">

<item
   android:id="@+id/message"
   app:showAsAction="always"
   android:title="message"/>

<item
   android:id="@+id/picture"
   app:showAsAction="always"
   android:title="picture"/>

<item
   android:id="@+id/mode"
   app:showAsAction="always"
   android:title="mode"/>

<item
   android:id="@+id/about"
   app:showAsAction="never"
   android:title="calculator"/>

<item
   android:id="@+id/exit"
   app:showAsAction="never"
   android:title="exit"/>
</menu>
```

MainActivity.java

```
package com.example.menu;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Toast;

import static android.widget.Toast.LENGTH_LONG;

public class MainActivity extends AppCompatActivity {

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

    public boolean onCreateOptionsMenu(Menu menu)
    {
        getMenuInflater().inflate(R.menu.options_menu, menu);
        return true;
    }

    public boolean onOptionsItemSelected(MenuItem item)
    {
        switch (item.getItemId()) {
            case R.id.message:
                Toast
                        .makeText(
                                getApplicationContext(),
                                "Shows share icon",
                                Toast.LENGTH_SHORT)
                        .show();
                return true;

            case R.id.picture:
                Toast
                        .makeText(
                                getApplicationContext(),
                                "Shows image icon",
                                Toast.LENGTH_SHORT)
                        .show();
                return (true);

            case R.id.mode:
                Toast
                        .makeText(
                                getApplicationContext(),
                                "Shows call icon",
                                Toast.LENGTH_SHORT)
                        .show();
                return (true);

            case R.id.about:
                Toast
                        .makeText(
                                getApplicationContext(),
                                "calculator menu",
                                Toast.LENGTH_SHORT)
                        .show();
                return (true);

            case R.id.exit:
                finish();
                return (true);
        }
        return (super.onOptionsItemSelected(item));
    }
}
```



Practical No.: 7

Alert Dialog

Aim: Create an android application to display Alert Dialog on pressing the Back button.

Step 1: Create a New Project in Android Studio

Step 2: Working with the XML Files

```
<?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"
   tools:context=".MainActivity">

   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginTop="180dp"
      android:gravity="center_horizontal"
      android:text="Press The Back Button of Your Phone."
      android:textSize="30dp"
      android:textStyle="bold" />
</RelativeLayout>
```

Step 3: Working with the MainActivity File

```
package com.example.myapplication;


import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.content.DialogInterface;
import androidx.appcompat.app.AlertDialog;
public class MainActivity extends AppCompatActivity {
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
   }
```

```
    @Override
    public void onBackPressed() {
        AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);
        builder.setMessage("Do you want to exit ?");
        builder.setTitle("Alert !");
        builder.setPositiveButton("Yes", (DialogInterface.OnClickListener) (dialog, which) ->
        {
            finish();
        });
        builder.setNegativeButton("No", (DialogInterface.OnClickListener) (dialog, which) ->
        {
            dialog.cancel();
        });
        AlertDialog alertDialog = builder.create();
        alertDialog.show();
    }
}
```

Date Picker Dialog:

Aim: Create an android application to display current system date

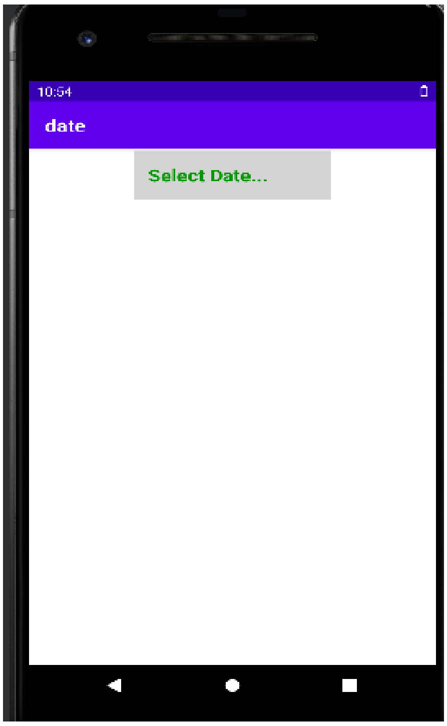
activity_main.xml

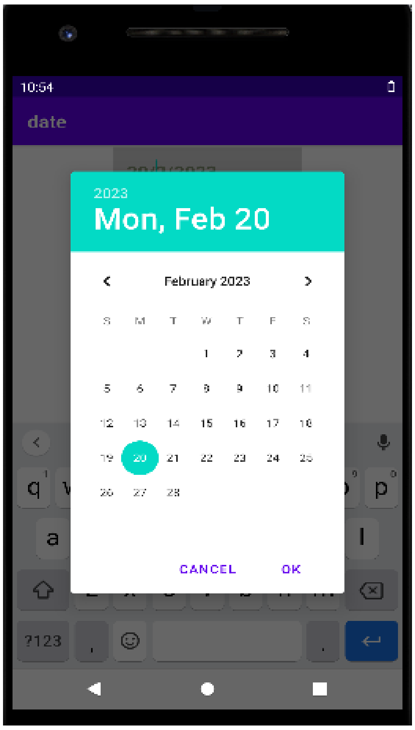
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="2dp"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/date"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:background="#d4d4d4"
        android:hint="Select Date..."
        android:padding="15dp"
        android:textColor="#897"
        android:textColorHint="#090"
        android:textSize="20sp"
        android:textStyle="bold" />

</RelativeLayout>
```

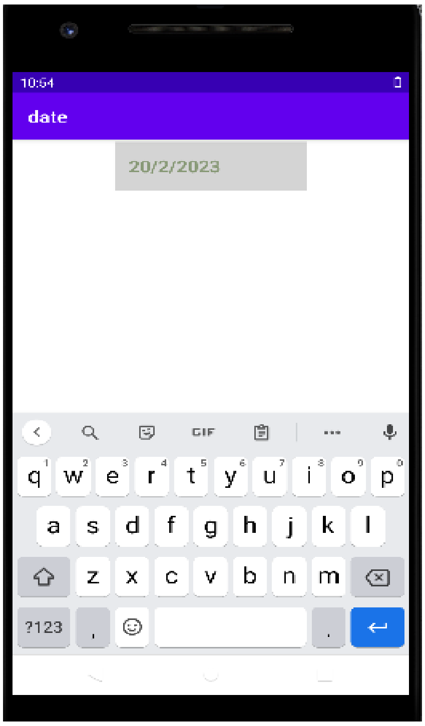
main_activity.java

```
package com.example.date;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.app.DatePickerDialog;
import android.view.View;
import android.widget.DatePicker;
import android.widget.EditText;
import java.util.Calendar;

public class MainActivity extends AppCompatActivity {

    EditText date;
    DatePickerDialog datePickerDialog;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // initiate the date picker and a button
        date = (EditText) findViewById(R.id.date);
        // perform click event on edit text
        date.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // calender class's instance and get current date , month and year from calender
                final Calendar c = Calendar.getInstance();
                int mYear = c.get(Calendar.YEAR); // current year
                int mMonth = c.get(Calendar.MONTH); // current month
                int mDay = c.get(Calendar.DAY_OF_MONTH); // current day
                // date picker dialog
                datePickerDialog = new DatePickerDialog(MainActivity.this,
                        new DatePickerDialog.OnDateSetListener() {

                            @Override
                            public void onDateSet(DatePicker view, int year,
                                                  int monthOfYear, int dayOfMonth) {
                                // set day of month , month and year value in the edit text
                                date.setText(dayOfMonth + "/"
                                        + (monthOfYear + 1) + "/" + year);

                            }
                        }, mYear, mMonth, mDay);
                datePickerDialog.show();
            }
        });

    }


}
```

Output:







Time Picker dialog

Aim: Create an android application to display current system time.

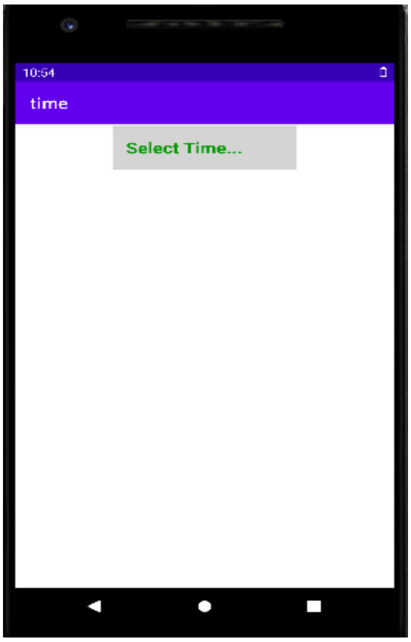
Activity_main.xml

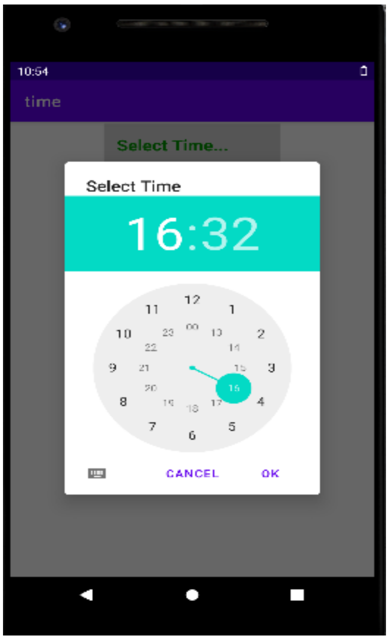
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="2dp"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/time"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:hint="Select Time..."
        android:textColor="#008799"
        android:textColorHint="#090"
        android:background="#d4d4d4"
        android:padding="15dp"
        android:textSize="20sp"
        android:textStyle="bold" />

</RelativeLayout>
```

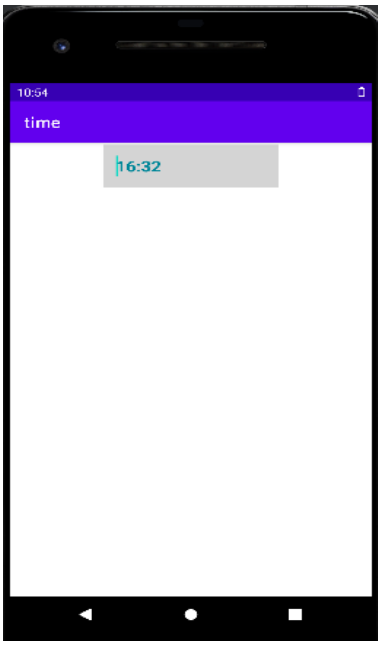
MainActivity.java

```
package com.example.time;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.app.TimePickerDialog;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.TimePicker;
import android.widget.Toast;
import java.util.Calendar;

public class MainActivity extends AppCompatActivity {

    EditText time;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        //  initiate the edit text
        time = (EditText) findViewById(R.id.time);
        // perform click event listener on edit text
        time.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Calendar currentTime = Calendar.getInstance();
                int hour = currentTime.get(Calendar.HOUR_OF_DAY);
                int minute = currentTime.get(Calendar.MINUTE);
                TimePickerDialog mTimePicker;
                mTimePicker = new TimePickerDialog(MainActivity.this, new
TimePickerDialog.OnTimeSetListener() {
                    @Override
                    public void onTimeSet(TimePicker timePicker, int selectedHour, int
selectedMinute) {
                        time.setText(selectedHour + ":" + selectedMinute);
                    }
                }, hour, minute, true);//Yes 24 hour time
                mTimePicker.setTitle("Select Time");
                mTimePicker.show();

            }
        });
    }


}
```

Output:





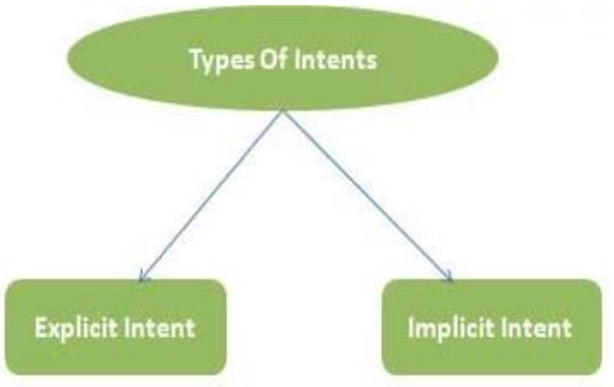


Practical No. 8

Android: Intent

- Android uses Intent for communicating between the components of an Application and also from one application to another application.
- Intent are the objects which is used in android for passing the information among Activities in an Application and from one app to another also.
- Intent are used for communicating between the Application components and it also provides the connectivity between two apps.

Types of Intent:



Explicit Intent:

- Explicit Intents are used to connect the application internally.

Implicit Intent:

- In Implicit Intents we do need to specify the name of the component. We just specify the Action, which has to be performed, and further this action is handled by the component of another application.

Aim: Create an android application to pass the data from current application to another application using intent

Activity_main.xml

```
<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:padding="2dp"
   tools:context=".MainActivity">

<TextView
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:textAppearance="?android:attr/textAppearanceMedium"
   android:text="If you click on Explicit example we will navigate to second activity within App and if you click on Implicit example Android Homepage will open in Browser"
   android:id="@+id/textView2"
   android:clickable="false"
   android:layout_alignParentTop="true"
   android:layout_alignParentStart="true"
   android:layout_marginTop="42dp"
   android:background="#3e7d02"
   android:textColor="#ffffff" />

<Button
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="Explicit Intent Example"
   android:id="@+id/explicit_Intent"
   android:layout_alignParentTop="true"
   android:layout_centerHorizontal="true"
   android:layout_marginTop="147dp" />

<Button
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="Implicit Intent Example"
   android:id="@+id/implicit_Intent"
   android:layout_centerVertical="true"
   android:layout_centerHorizontal="true" />

</RelativeLayout>
```

Activity_second.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:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="2dp"
    android:background="#CCEEAA"
    tools:context=".SecondActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="This is Second Activity"
        android:id="@+id/textView"
        android:layout_centerVertical="true"
        android:layout_centerHorizontal="true" />
</RelativeLayout>
```

MainActivity.java

```
package com.example.intent;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.content.Intent;
import android.net.Uri;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {
    Button explicit_btn, implicit_btn;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);


        explicit_btn = (Button)findViewById(R.id.explicit_Intent);
        implicit_btn = (Button) findViewById(R.id.implicit_Intent);

        //implement Onclick event for Explicit Intent

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

                Intent intent = new  Intent(getBaseContext(), SecondActivity.class);
                startActivity(intent);



            }
        });

        //implement onClick event for Implicit Intent

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

                Intent intent = new Intent(Intent.ACTION_VIEW);
                intent.setData(Uri.parse("https://www.androidstudio.com"));
                startActivity(intent);
            }
        });


    }
}
```

<u>SecondActivity.java</u>

```
ackage com.example.intent;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.Toast;
public class SecondActivity extends AppCompatActivity {

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

        Toast.makeText(getApplicationContext(), "We are moved to second 
Activity",Toast.LENGTH_LONG).show();
    }
}
```

AndroidManifest.xml

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

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/Theme.Intent"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

Practical No. 9

Broadcast Receiver in Android

- Broadcast in android is the system-wide events that can occur when the device starts, when a message is received on the device or when incoming calls are received, or when a device goes to airplane mode, etc.
- Broadcast Receivers are used to respond to these system-wide events. Broadcast Receivers allow us to register for the system and application events, and when that event happens, then the register receivers get notified.
- Intent:

Intent	Description Of Event
android.intent.action.BATTERY_LOW :	Indicates low battery condition on the device.
android.intent.action.BOOT_COMPLETED	This is broadcast once after the system has finished booting
android.intent.action.CALL	To perform a call to someone specified by the data
android.intent.action.DATE_CHANGED	Indicates that the date has changed
android.intent.action.REBOOT	Indicates that the device has been a reboot
android.net.conn.CONNECTIVITY_CHANGE	The mobile network or wifi connection is changed(or reset)
android.intent.ACTION_AIRPLANE_MODE_CHANGED	This indicates that airplane mode has been switched on or off.

Creating the Broadcast Receiver:

Step 1: Create a New Project

Step 2: Working with the activity_main.xml file

Step 3: Working with the MainActivity file

```
package com.example.myapplication1;

import android.content.Intent;
import android.content.IntentFilter;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
    AirplaneModeChangeReceiver airplaneModeChangeReceiver = new
AirplaneModeChangeReceiver();

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

    @Override
    protected void onStart() {
        super.onStart();
        IntentFilter filter = new IntentFilter(Intent.ACTION_AIRPLANE_MODE_CHANGED);
        registerReceiver(airplaneModeChangeReceiver, filter);
    }

    @Override
    protected void onStop() {
        super.onStop();
        unregisterReceiver(airplaneModeChangeReceiver);
    }
}
```

Step 4: Create a new class

Go to **app** > **java** > **MainActicity** > **right-click** > **New** > **Java Class** and name the files as **AirplaneModeChangeReceiver**.

```
package com.example.myapplication1;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.provider.Settings;
import android.widget.Toast;

public class AirplaneModeChangeReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {

        if (isAirplaneModeOn(context.getApplicationContext())) {
            Toast.makeText(context, "AirPlane mode is on", Toast.LENGTH_SHORT).show();
        } else {
            Toast.makeText(context, "AirPlane mode is off", Toast.LENGTH_SHORT).show();
        }
    }

    private static boolean isAirplaneModeOn(Context context) {
        return Settings.System.getInt(context.getContentResolver(),
Settings.Global.AIRPLANE_MODE_ON, 0) != 0;
    }
}
```