

ACTIVITY-INTENT

Android Activity Lifecycle

Android Activity Lifecycle is controlled by 7 methods of android.app.Activity class. The android Activity is the subclass of ContextThemeWrapper class.

An activity is the single screen in android. It is like window or frame of Java.

By the help of activity, you can place all your UI components or widgets in a single screen.

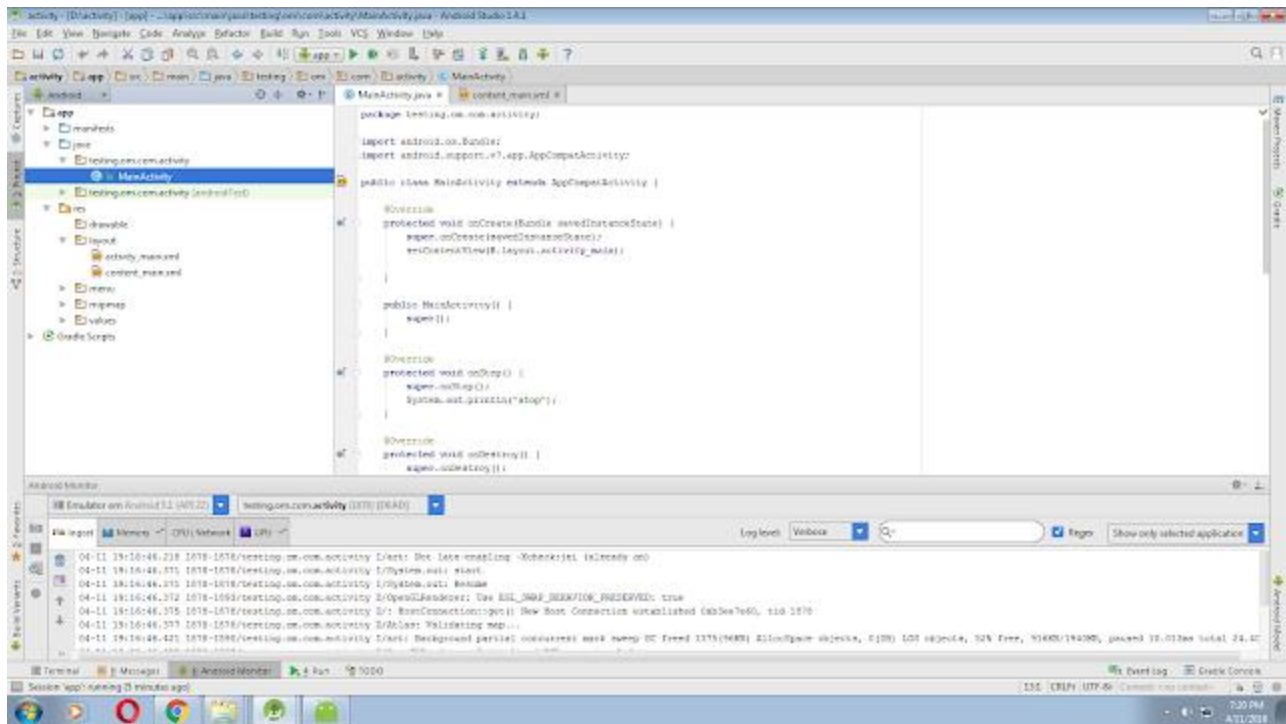
The 7 lifecycle method of Activity describes how activity will behave at different states.

Android Activity Lifecycle methods

Let's see the 7 lifecycle methods of android activity.

Method	Description
onCreate	called when activity is first created.
onStart	called when activity is becoming visible to the user.
onResume	called when activity will start interacting with the user.
onPause	called when activity is not visible to the user.
onStop	called when activity is no longer visible to the user.
onRestart	called after your activity is stopped, prior to start.
onDestroy	called before the activity is destroyed.

Example on Activity Lifecycle see below directory structure :



TO check activity life cycle follow given code :

(1)write code for MainActivity.java file:

```

package testing.om.com.activity;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

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

    public MainActivity() {
        super();
    }

    @Override
    protected void onStop() {
        super.onStop();
        System.out.println("stop");
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        System.out.println("destroy");
    }
}

```

```

@Override
protected void onPause() {
    super.onPause();
    System.out.println("pause");
}

@Override
protected void onResume() {
    super.onResume();
    System.out.println("Resume");
}

@Override
protected void onStart() {
    super.onStart();
    System.out.println("start");
}

@Override
protected void onRestart() {
    super.onRestart();
    System.out.println("Restart");
}
}

```

(2)write code for activity_main.xml file :

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

```

<android.support.design.widget.CoordinatorLayout

    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:fitsSystemWindows="true"

    tools:context=".MainActivity">

    <android.support.design.widget.AppBarLayout android:layout_height="wrap_content"

        android:layout_width="match_parent"

        android:theme="@style/AppTheme.AppBarOverlay">

        <android.support.v7.widget.Toolbar android:id="@+id/toolbar"

            android:layout_width="match_parent"

            android:layout_height="?attr/actionBarSize"

            android:background="?attr/colorPrimary"

            app:popupTheme="@style/AppTheme.PopupOverlay" />

    </android.support.design.widget.AppBarLayout>

```

```
<include layout="@layout/content_main" />
```

```
<android.support.design.widget.FloatingActionButton android:id="@+id/fab"  
    android:layout_width="wrap_content" android:layout_height="wrap_content"  
    android:layout_gravity="bottom|end" android:layout_margin="@dimen/fab_margin"  
    android:src="@android:drawable/ic_dialog_email" />
```

```
</android.support.design.widget.CoordinatorLayout>
```

(3)write code for content_main.xml file:

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

```
<RelativeLayout 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:layout_width="match_parent"
```

```
    android:layout_height="match_parent"  
    android:paddingLeft="@dimen/activity_horizontal_margin"
```

```
    android:paddingRight="@dimen/activity_horizontal_margin"
```

```
    android:paddingTop="@dimen/activity_vertical_margin"
```

```
    android:paddingBottom="@dimen/activity_vertical_margin"
```

```
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
```

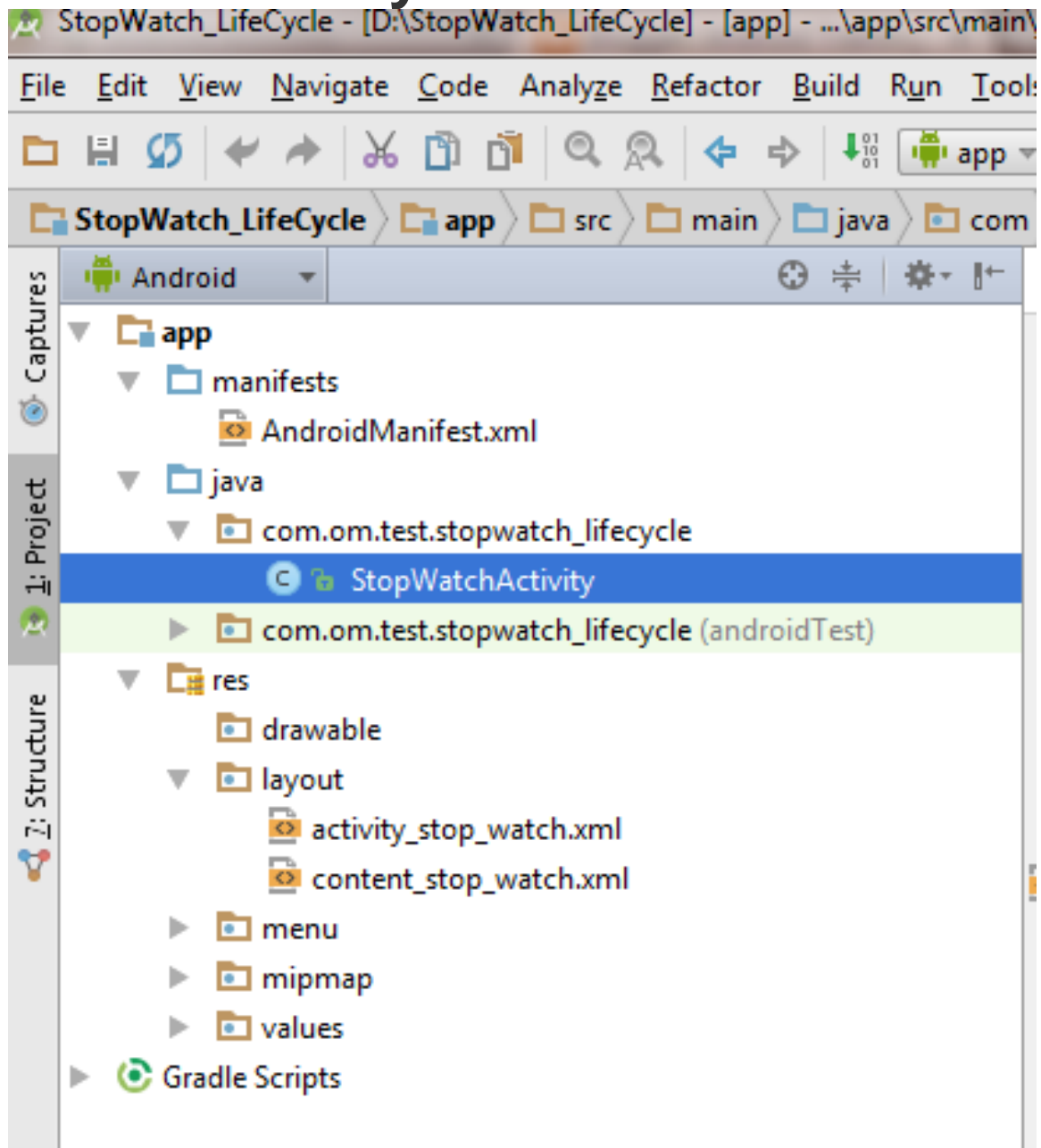
```
    tools:showIn="@layout/activity_main" tools:context=".MainActivity">
```

```
<TextView android:text="Hello World!" android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content" />
```

```
</RelativeLayout>
```

Example on Activity Lifecycle see below directory structure :



(1)First write code for "StopWatchActivity.java" Activity :

```
package com.om.test.stopwatch_lifecycle;

import android.os.Bundle;
import android.os.Handler;
import android.support.design.widget.FloatingActionButton;
```

```

import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.View;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.TextView;

public class StopwatchActivity extends AppCompatActivity {

    int seconds=0;
    boolean running = false;
    boolean wasRunning = false;

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

        if (savedInstanceState!=null)
        {
            seconds=savedInstanceState.getInt("seconds");
            running=savedInstanceState.getBoolean("isRunning");
        }

        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
                public void onClick(View view) {
                    Snackbar.make(view, "Replace with your own action",
Snackbar.LENGTH_LONG)
                        .setAction("Action", null).show();
                }
        });

        runTimer();
    }

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

    @Override    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will //
        automatically handle clicks on the Home/Up button, so long // as you specify a
        parent activity in AndroidManifest.xml.        int id = item.getItemId();

        //noinspection SimplifiableIfStatement        if (id == R.id.action_settings)
    {
        return true;
    }

    return super.onOptionsItemSelected(item);
    }

    public void onClickStart(View view)
    {

```

```

        running =true;
    }

    public void onClickStop(View view)
    {
        running =false;
    }

    public void onClickReset(View view)
    {
        running =false;
        seconds=0;
    }

    public void runTimer()
    {
        final TextView tv = (TextView) findViewById(R.id.timeDisplay);

        final Handler handler=new Handler();

        handler.post(new Runnable() {
            @Override          public void run() {

                int hrs = seconds / 3600;
                int min = (seconds % 3600) / 60;
                int sec = seconds % 60;
                String time = String.format("%d:%02d:%02d", hrs, min, sec);
                tv.setText(time);
                if (running) {
                    seconds++;
                }

                handler.postDelayed(this,1000);
            }
        });
    }

    @Override    protected void onSaveInstanceState(Bundle outState) {
        super.onSaveInstanceState(outState);
        outState.putInt("seconds", seconds);
        outState.putBoolean("isRunning", running);
    }

    @Override    protected void onStart() {
        super.onStart();

        if(wasRunning)
            running=true;
        else
            running= false;
    }

    @Override    protected void onStop() {
        super.onStop();
        wasRunning = running;
        running=false;
    }

```

```
}
```

(2)write code for activity_stop_watch.xml:

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

<android.support.design.widget.CoordinatorLayout

    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:fitsSystemWindows="true"

    tools:context=".StopWatchActivity">

    <android.support.design.widget.AppBarLayout android:layout_height="wrap_content"

        android:layout_width="match_parent"

        android:theme="@style/AppTheme.AppBarOverlay">

        <android.support.v7.widget.Toolbar android:id="@+id/toolbar"

            android:layout_width="match_parent"

            android:layout_height="?attr/actionBarSize"

            android:background="?attr/colorPrimary"

            app:popupTheme="@style/AppTheme.PopupOverlay" />

    </android.support.design.widget.AppBarLayout>

    <include layout="@layout/content_stop_watch" />

    <android.support.design.widget.FloatingActionButton android:id="@+id/fab"

        android:layout_width="wrap_content" android:layout_height="wrap_content"

        android:layout_gravity="bottom|end" android:layout_margin="@dimen/fab_margin"

        android:src="@android:drawable/ic_dialog_email" />

</android.support.design.widget.CoordinatorLayout>
```

(3)write code for content_stop_watch.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"
xmlns:app="http://schemas.android.com/apk/res-auto"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
android:paddingBottom="@dimen/activity_vertical_margin"
app:layout_behavior="@string/appbar_scrolling_view_behavior"
tools:showIn="@layout/activity_stop_watch" tools:context=".StopWatchActivity">
```

```
<TextView
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text=""
    android:id="@+id/timeDisplay"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="41dp"
    android:textSize="92sp"
/>
```

```
<Button
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Start"
    android:id="@+id/start"
    android:layout_below="@+id/timeDisplay"
    android:layout_centerHorizontal="true"
    android:onClick="onClickStart"
    android:layout_marginTop="45dp" />
```

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Stop"
    android:id="@+id/stop"
    android:onClick="onClickStop"
    android:layout_below="@+id/start"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="47dp" />
```

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Reset"
    android:onClick="onClickReset"
    android:id="@+id/reset"
    android:layout_below="@+id/stop"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="56dp" />
```

```
</RelativeLayout>
```

OUTPUT:



ANDROID INTENT :

Android Intent is the *message* that is passed between components such as activities, content providers, broadcast receivers, services etc.

It is generally used with `startActivity()` method to invoke activity, broadcast receivers etc.

The **dictionary meaning** of intent is *intention or purpose*. So, it can be described as the intention to do action.

The LabeledIntent is the subclass of android.content.Intent class.

Android intents are mainly used to:

- Start the service
 - Launch an activity
 - Display a web page
 - Display a list of contacts
 - Broadcast a message
 - Dial a phone call etc.
-

Types of Android Intents

There are two types of intents in android: implicit and explicit.

1) Implicit Intent

Implicit Intent doesn't specify the component. In such case, intent provides information of available components provided by the system that is to be invoked.

For example, you may write the following code to view the webpage.

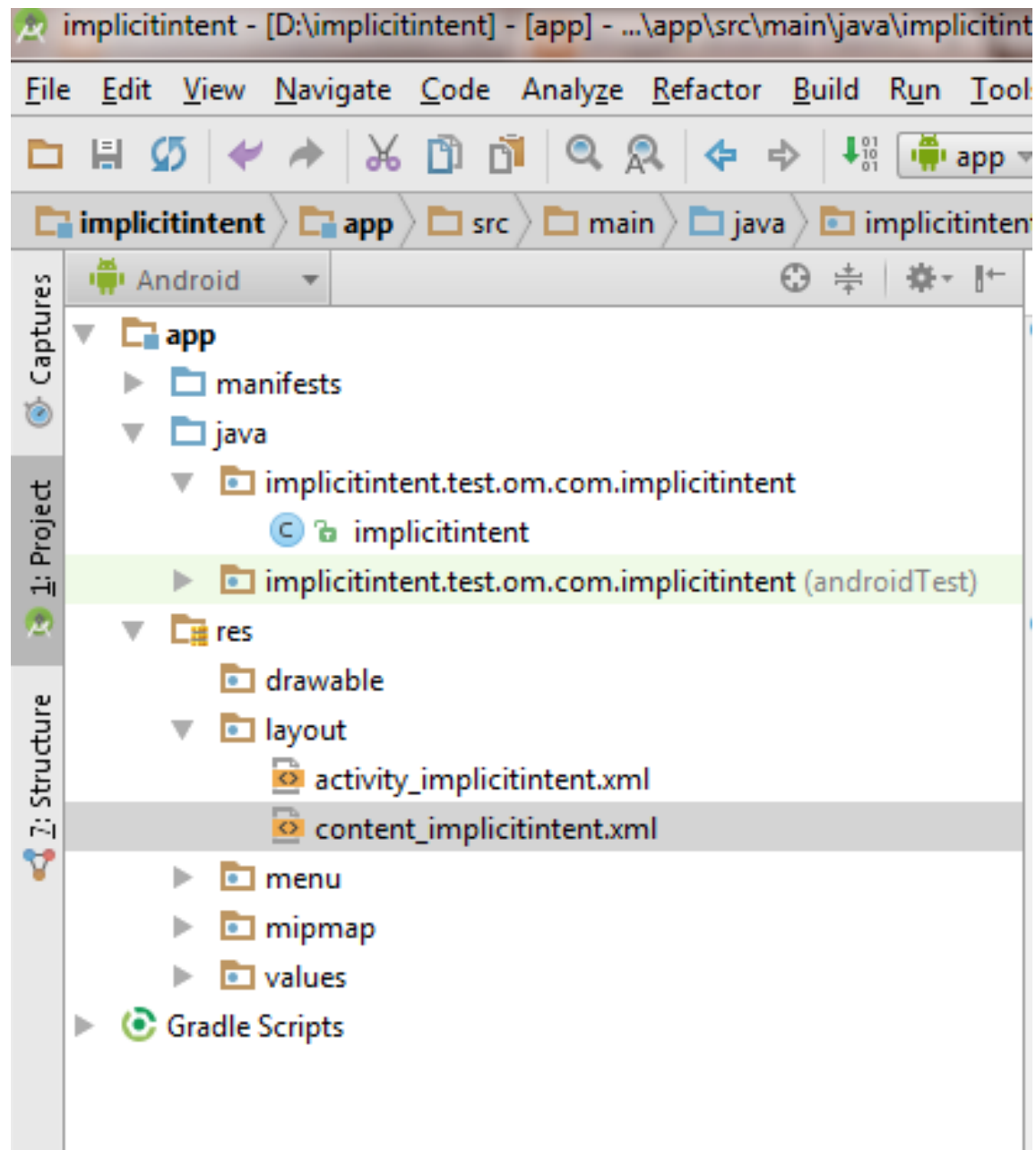
1. Intent intent=**new** Intent(Intent.ACTION_VIEW);
2. intent.setData(Uri.parse("http://www.vissicomp.com"));
3. startActivity(intent);
- 4.

2) Explicit Intent

Explicit Intent specifies the component. In such case, intent provides the external class to be invoked.

1. Intent i = **new** Intent(getApplicationContext(), ActivityTwo.**class**);
2. startActivity(i);

IMPLICIT INTENT EXAMPLE see below
Directory Structure :



(1) First write code for implicitinten.java Activity:

```
package implicitintent.test.om.com.implicitintent;  
  
import android.content.Intent;
```

```

import android.net.Uri;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;

public class implicitintent extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_implicitintent);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Snackbar.make(view, "Replace with your own action",
                    Snackbar.LENGTH_LONG)
                    .setAction("Action", null).show();
            }
        });

    }

    public void onClick(View view)
    {
        Intent intent=new Intent(Intent.ACTION_VIEW);
        intent.setData(Uri.parse("http://www.vissicomp.com"));
        startActivity(intent);
    }

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

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {
            return true;
        }

        return super.onOptionsItemSelected(item);
    }
}

```

(2)Then write code for activity_implicitintent.xml :

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

<android.support.design.widget.CoordinatorLayout

    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:fitsSystemWindows="true"

    tools:context=".implicitintent">

    <android.support.design.widget.AppBarLayout android:layout_height="wrap_content"

        android:layout_width="match_parent"

        android:theme="@style/AppTheme.AppBarOverlay">

        <android.support.v7.widget.Toolbar android:id="@+id/toolbar"

            android:layout_width="match_parent"

            android:layout_height="?attr/actionBarSize"

            android:background="?attr/colorPrimary"

            app:popupTheme="@style/AppTheme.PopupOverlay" />

    </android.support.design.widget.AppBarLayout>

    <include layout="@layout/content_implicitintent" />

    <android.support.design.widget.FloatingActionButton android:id="@+id/fab"

        android:layout_width="wrap_content" android:layout_height="wrap_content"

        android:layout_gravity="bottom|end" android:layout_margin="@dimen/fab_margin"

        android:src="@android:drawable/ic_dialog_email" />

</android.support.design.widget.CoordinatorLayout>
```

(3)Then write code for content_implicitintent.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"

    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"

    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"

    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"

    tools:showIn="@layout/activity_implicitintent" tools:context=".implicitintent">

    <Button

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:text="vissicomp"

        android:id="@+id/button"

        android:onClick="onClick"

        android:layout_marginTop="144dp"

        android:layout_alignParentTop="true"

        android:layout_centerHorizontal="true" />

</RelativeLayout>
```

OUTPUT:



AFTER CLICK ON VISSICOMP BUTTON :



2) Explicit Intent

Explicit Intent specifies the component. In such case, intent provides the external class to be invoked.

1. Intent i = **new** Intent(getApplicationContext(), ActivityTwo.**class**);
2. startActivity(i);

Android Explicit Intent Example:

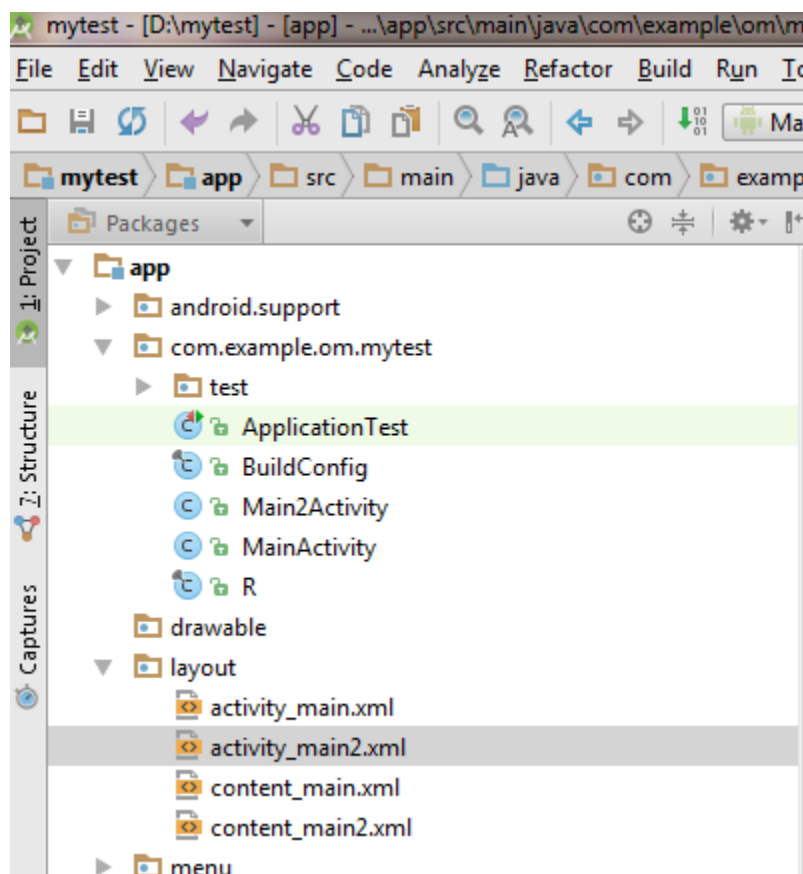
Android Explicit intent specifies the component to be invoked from activity. In other words, we can call another activity in android by explicit intent.

We can also pass the information from one activity to another using explicit intent.

Here, we are going to see an example to call one activity from another and vice-versa.

Android calling one activity from another activity example using Explicit Intent.

Let's see the simple example of android explicit example that calls one activity from another and vice versa. SEE DIRECTORY STRUCTURE OF EXAMPLE IN GIVEN IMAGE..



MainActivity.java :

```
package com.example.om.mytest;

import android.content.Intent;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
```

```

import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.View;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

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

        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Snackbar.make(view, "Replace with your own action",
Snackbar.LENGTH_LONG)
                    .setAction("Action", null).show();
            }
        });

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

        b1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v)
            {

                Intent i = new Intent(getApplicationContext(), Main2Activity.class);
                startActivity(i);

            }
        });

    }

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

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

```

```

        //noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {
            return true;
        }

        return super.onOptionsItemSelected(item);
    }
}

```

activity_main.xml:

```

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

<android.support.design.widget.CoordinatorLayout

    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:fitsSystemWindows="true"

    tools:context=".MainActivity">

    <android.support.design.widget.AppBarLayout android:layout_height="wrap_content"

        android:layout_width="match_parent"

        android:theme="@style/AppTheme.AppBarOverlay">

        <android.support.v7.widget.Toolbar android:id="@+id/toolbar"

            android:layout_width="match_parent"

            android:layout_height="?attr/actionBarSize"

            android:background="?attr/colorPrimary"

            app:popupTheme="@style/AppTheme.PopupOverlay" />

    </android.support.design.widget.AppBarLayout>

    <include layout="@layout/content_main" />

    <android.support.design.widget.FloatingActionButton android:id="@+id/fab"

        android:layout_width="wrap_content" android:layout_height="wrap_content"

```

```
    android:layout_gravity="bottom|end" android:layout_margin="@dimen/fab_margin"
    android:src="@android:drawable/ic_dialog_email" />
```

```
</android.support.design.widget.CoordinatorLayout>
```

content_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"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:showIn="@layout/activity_main" tools:context=".MainActivity">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="submit"
        android:id="@+id/button1"
        android:layout_centerVertical="true"
        android:layout_toRightOf="@+id/textView"
        android:layout_toEndOf="@+id/textView" />

</RelativeLayout>
```

Main2Activity.java:

```

package com.example.om.mytest;

import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.View;

public class Main2Activity extends AppCompatActivity {

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main2);

        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View view) {

                Snackbar.make(view, "Replace with your own action",
Snackbar.LENGTH_LONG)

                    .setAction("Action", null).show();

            }

        });

    }

}

```

activity_main2.xml:

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

```

<android.support.design.widget.CoordinatorLayout
    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:fitsSystemWindows="true"
    tools:context="com.example.om.mytest.Main2Activity">

    <android.support.design.widget.AppBarLayout android:layout_height="wrap_content"

        android:layout_width="match_parent"
        android:theme="@style/AppTheme.AppBarOverlay">

        <android.support.v7.widget.Toolbar android:id="@+id/toolbar"

            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"

            android:background="?attr/colorPrimary"
            app:popupTheme="@style/AppTheme.PopupOverlay" />

    </android.support.design.widget.AppBarLayout>

    <include layout="@layout/content_main2" />

    <android.support.design.widget.FloatingActionButton android:id="@+id/fab"

        android:layout_width="wrap_content" android:layout_height="wrap_content"

        android:layout_gravity="bottom|end" android:layout_margin="@dimen/fab_margin"

        android:src="@android:drawable/ic_dialog_email" />

</android.support.design.widget.CoordinatorLayout>

```

content_main2.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"

```



```
xmlns:app="http://schemas.android.com/apk/res-auto"
android:layout_width="match_parent"

    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"

    android:paddingRight="@dimen/activity_horizontal_margin"

    android:paddingTop="@dimen/activity_vertical_margin"

    android:paddingBottom="@dimen/activity_vertical_margin"

    app:layout_behavior="@string/appbar_scrolling_view_behavior"

    tools:showIn="@layout/activity_main2"
tools:context="com.example.om.mytest.Main2Activity">
```

```
<TextView
```

```
    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:textAppearance="?android:attr/textAppearanceLarge"

    android:text="it is second activity example"

    android:id="@+id/textView2"

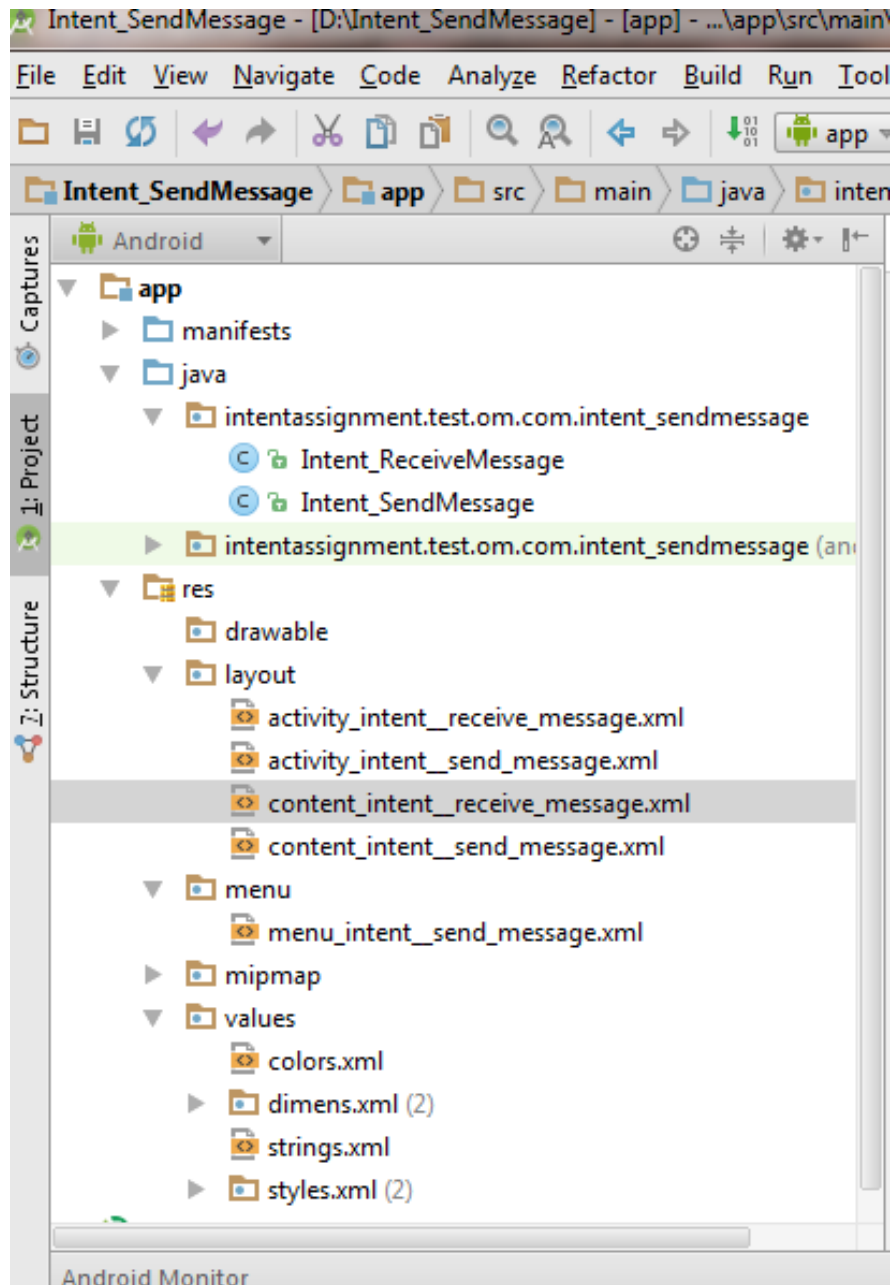
    android:layout_alignParentTop="true"

    android:layout_centerHorizontal="true"

    android:layout_marginTop="119dp" />
```

```
</RelativeLayout>
```

One more example on Intent to send message from one activity to another activity follow below directory structure:



**First write code
for Intent_SendMessage activity
which send message to
Intent_ReceiveMessage activity:**

(1)see code for activity `Intent_SendMessage.java` :

```
package intentassignment.test.om.com.intent_sendmessage;
```

```

import android.content.Intent;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.View;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.EditText;

public class Intent_SendMessage extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_intent_send_message);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Snackbar.make(view, "Replace with your own action",
Snackbar.LENGTH_LONG)
                    .setAction("Action", null).show();
            }
        });
    }

    public void sendMessage(View view)
    {
        EditText e = (EditText) findViewById(R.id.sendMessage);
        Intent in = new Intent(this, Intent_ReceiveMessage.class);
        in.putExtra("message",e.getText().toString());
        startActivity(in);
    }

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

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {
            return true;
        }
    }
}

```

```

        return super.onOptionsItemSelected(item);
    }
}

```

(2)Then see code for activity_intent_send_message.xml file:

```

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

<android.support.design.widget.CoordinatorLayout

    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:fitsSystemWindows="true"

    tools:context=".Intent_SendMessage">

    <android.support.design.widget.AppBarLayout android:layout_height="wrap_content"

        android:layout_width="match_parent"

        android:theme="@style/AppTheme.AppBarOverlay">

        <android.support.v7.widget.Toolbar android:id="@+id/toolbar"

            android:layout_width="match_parent"

            android:layout_height="?attr/actionBarSize"

            android:background="?attr/colorPrimary"

            app:popupTheme="@style/AppTheme.PopupOverlay" />

    </android.support.design.widget.AppBarLayout>

    <include layout="@layout/content_intent_send_message" />

    <android.support.design.widget.FloatingActionButton android:id="@+id/fab"

        android:layout_width="wrap_content" android:layout_height="wrap_content"

        android:layout_gravity="bottom|end" android:layout_margin="@dimen/fab_margin"

        android:src="@android:drawable/ic_dialog_email" />

</android.support.design.widget.CoordinatorLayout>

```

(3)then see code for content_intent_send_message.xml file:

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

<RelativeLayout 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:layout_width="match_parent"

    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"

    android:paddingRight="@dimen/activity_horizontal_margin"

    android:paddingTop="@dimen/activity_vertical_margin"

    android:paddingBottom="@dimen/activity_vertical_margin"

    app:layout_behavior="@string/appbar_scrolling_view_behavior"

    tools:showIn="@layout/activity_intent__send_message"
    tools:context=".Intent_SendMessage">
```

```
<Button

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:text="Send Message"

    android:onClick="sendMessage"

    android:id="@+id/button"

    android:layout_below="@+id/sendMessage"

    android:layout_centerHorizontal="true" />
```

```
<EditText

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:id="@+id/sendMessage"

    android:hint="Enter message..."

    android:layout_alignLeft="@+id/button"

    android:layout_alignStart="@+id/button"

    android:layout_alignRight="@+id/button"

    android:layout_alignEnd="@+id/button" />
```

```
</RelativeLayout>
```

Now code for Intent_ReceiveMessage Activity means second activity which receives messages :

(1)write code for Intent_ReceiveMessage.java :

```
package intentassignment.test.om.com.intent_sendmessage;
```

```
import android.content.Intent;
```

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

```
import android.support.design.widget.FloatingActionButton;
```

```
import android.support.design.widget.Snackbar;
```

```
import android.support.v7.app.AppCompatActivity;
```

```
import android.support.v7.widget.Toolbar;
```

```
import android.view.View;
```

```
import android.widget.TextView;
```

```
public class Intent_ReceiveMessage extends AppCompatActivity {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

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

```
        String message= getIntent().getStringExtra("message");
```

```
        TextView t = (TextView) findViewById(R.id.textView);
```

```

t.setText(message);

Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
setSupportActionBar(toolbar);

FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
fab.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View view) {

        Snackbar.make(view, "Replace with your own action",
Snackbar.LENGTH_LONG)

            .setAction("Action", null).show();

    }

});
}

public void back(View view)
{

    Intent i = new Intent(this, Intent_SendMessage.class);

    startActivity(i);

}

}

```

(2) Now write code for activity_intent_receive_message.xml :

```

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

<android.support.design.widget.CoordinatorLayout

    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:fitsSystemWindows="true"

        tools:context="intentassignment.test.om.com.intent_sendmessage.Intent_ReceiveMessa
ge">

        <android.support.design.widget.AppBarLayout android:layout_height="wrap_content"

                android:layout_width="match_parent"
android:theme="@style/AppTheme.AppBarOverlay">

                <android.support.v7.widget.Toolbar android:id="@+id/toolbar"

                        android:layout_width="match_parent"
android:layout_height="?attr/actionBarSize"

                                android:background="?attr/colorPrimary"
app:popupTheme="@style/AppTheme.PopupOverlay" />

        </android.support.design.widget.AppBarLayout>

        <include layout="@layout/content_intent_receive_message" />

        <android.support.design.widget.FloatingActionButton android:id="@+id/fab"

                android:layout_width="wrap_content" android:layout_height="wrap_content"

                android:layout_gravity="bottom|end" android:layout_margin="@dimen/fab_margin"

                android:src="@android:drawable/ic_dialog_email" />

</android.support.design.widget.CoordinatorLayout>

```

(3) Now write code for content_intent_receive_message.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"

        xmlns:app="http://schemas.android.com/apk/res-auto"
android:layout_width="match_parent"

        android:layout_height="match_parent"
android:paddingLeft="@dimen/activity_horizontal_margin"

        android:paddingRight="@dimen/activity_horizontal_margin"

        android:paddingTop="@dimen/activity_vertical_margin"

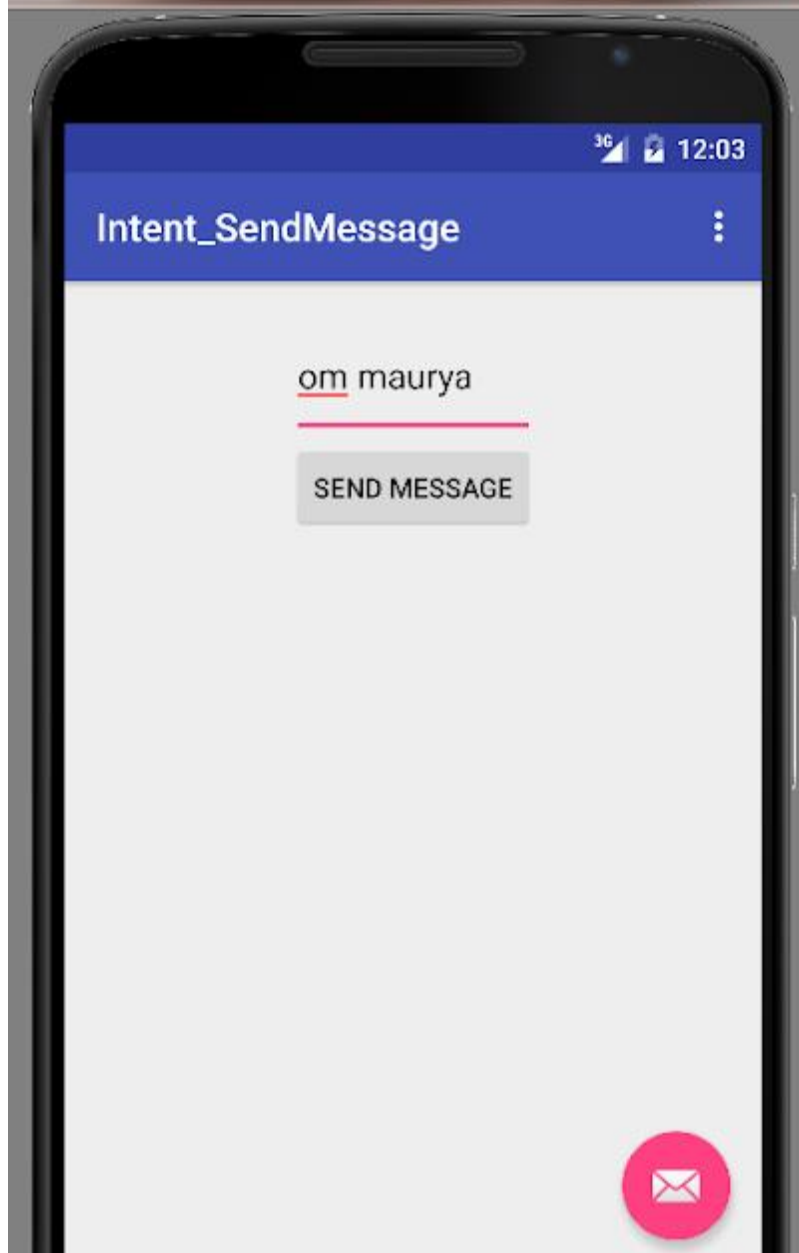
```



```
        android:paddingBottom="@dimen/activity_vertical_margin"
        app:layout_behavior="@string/appbar_scrolling_view_behavior"
        tools:showIn="@layout/activity_intent_receive_message"
        tools:context="intentassignment.test.om.com.intent_sendmessage.Intent_ReceiveMessage">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Message Received..."
        android:id="@+id/textView"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="45dp" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="BACK..."
        android:onClick="back"
        android:id="@+id/back"
        android:layout_below="@+id/textView"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="53dp" />
</RelativeLayout>
```

output:



Intent_SendMessage

om maurya

SEND MESSAGE



