**DEPARTMENT OF COMPUTER SCIENCE**

## Android Application Development

## Lab Manual



# RATHINAM COLLEGE OF ARTS & SCIENCE (AUTONOMOUS)

|  |  |  |  |
| --- | --- | --- | --- |
| **Prepared by:** | **Approved & Reviewed by:** | **Issued by:** | **W.e.f Date:** |
| S.KARTHIKEYAN |  |  | JUNE-2019 |

##### In-charge HOD Principal



## RATHINAM COLLEGE OF ARTS & SCIENCE

Rathinam Techzone Campus, Pollachi Road, Eachanari, Coimbatore - 641021, Tamil Nadu. [http://www.rathinamcollege.com](http://www.rathinamcollege.com/)

**Department of Computer Science**

Lab Manual for the Academic Year 2019-20 (in accordance with Computer Science syllabus)

SUBJECT : Android Application Development STREAM : BCA/B.Sc [ CS/CT/IT ]

H.O.D DEAN

**List of Lab Exercises**

1. Getting Started with Mobile App Development
2. Create and validate a login application using username as Email ID else login button must remain disabled
3. Creating a Hello World program Experiment with the most basic features and mobile application interaction concepts lists, text boxes, buttons, graphics, etc)
4. Create a following menu items in mobile application program a. cut b. copy c. pasted. delete e. select all f. unselect all
5. Create an application to change screen color as per the user choice from a menu
6. Create an Android application with a combo box, spinner, toast message get selected item
7. Create an application to call a phone number entered by the user and edit Text
8. Develop a native application that uses GPS location information.

***Android Application Development 1***

* 1. **GETTING STARTED WITH ANDROID**

##### Aim:

To know how to get started with Android Development Environment .

##### Procedure:

**Step 1:** Download the **Java Development Kit(jdk)** from the oracle Website**.**



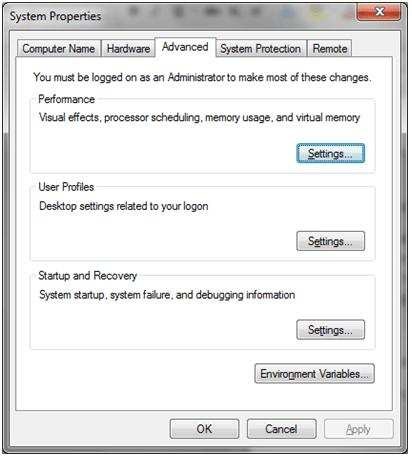
**Step 2:** Downloadable **jdk** will be shown.Click **Save file.**



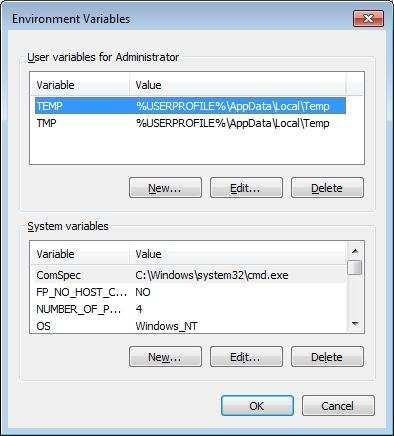
**Step 3:** Install the downloaded **Java Development Kit.**

**Step 4:** It will install all the required **Java Runtime Environment.**

**Step 5:** Select **Start** -> **Control Panel -> System** and select the **Advanced** tab and then click on **Environment Variables.**



**Step 6:** Select **Start** -> **Control Panel -> System -> Advanced System Settings.**In the **Advanced** tab and then click on **Environment Variables. Step 7 :** On the **Environment Variables,**In the **System Variables,**Select **Path.**

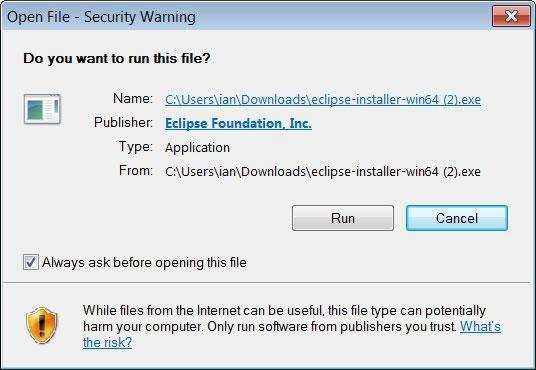


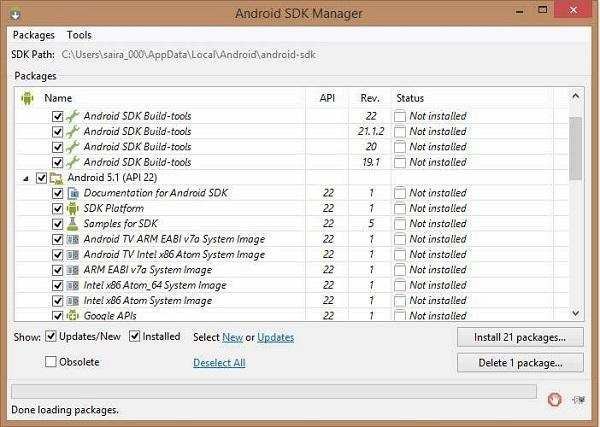
**Step 8 :** On the **Edit System Variables** window, Specify the following Path(Example)**.**

set PATH=C:\jdk1.8.0\_102\bin;%PATH% set JAVA\_HOME=C:\jdk1.8.0\_102

**Step 9 :** Download **Android SDK** from the website.

**Step 10 :** Download **Android SDK** from the website.

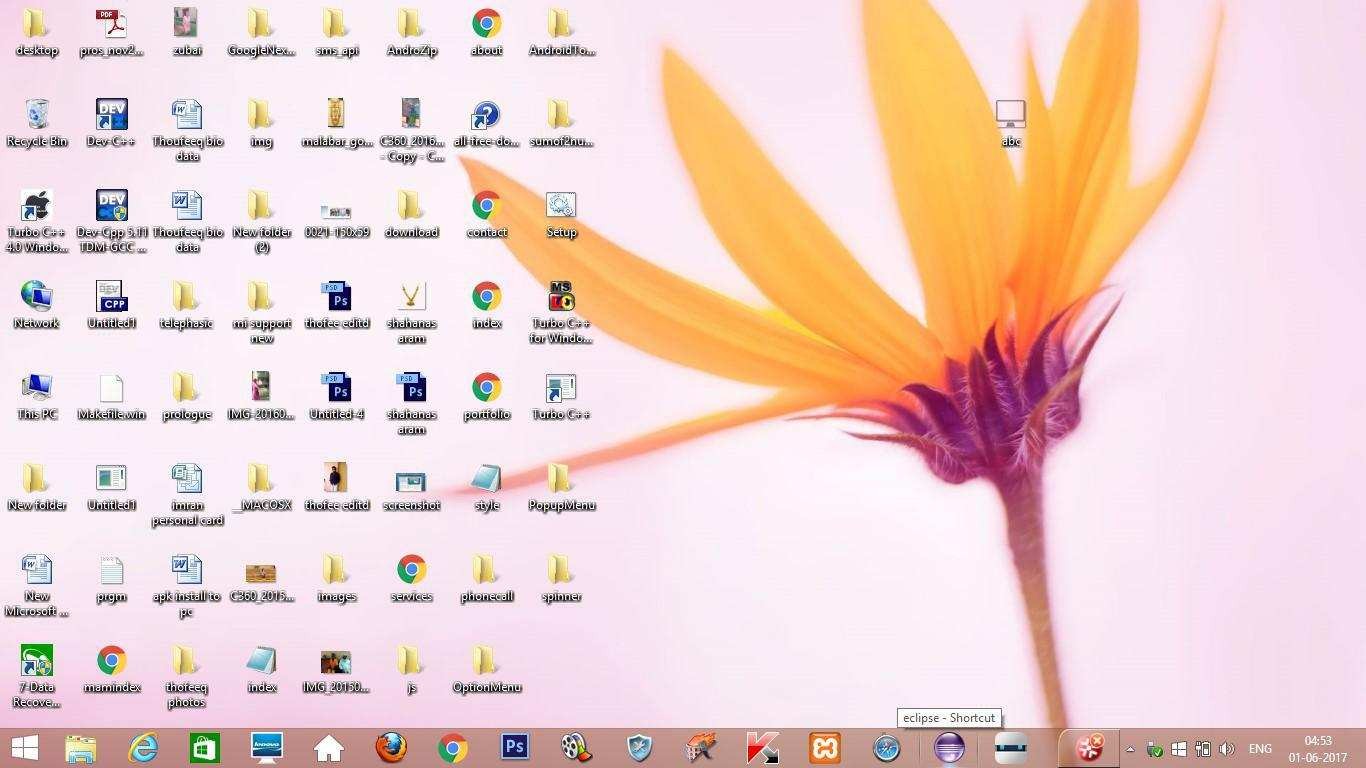


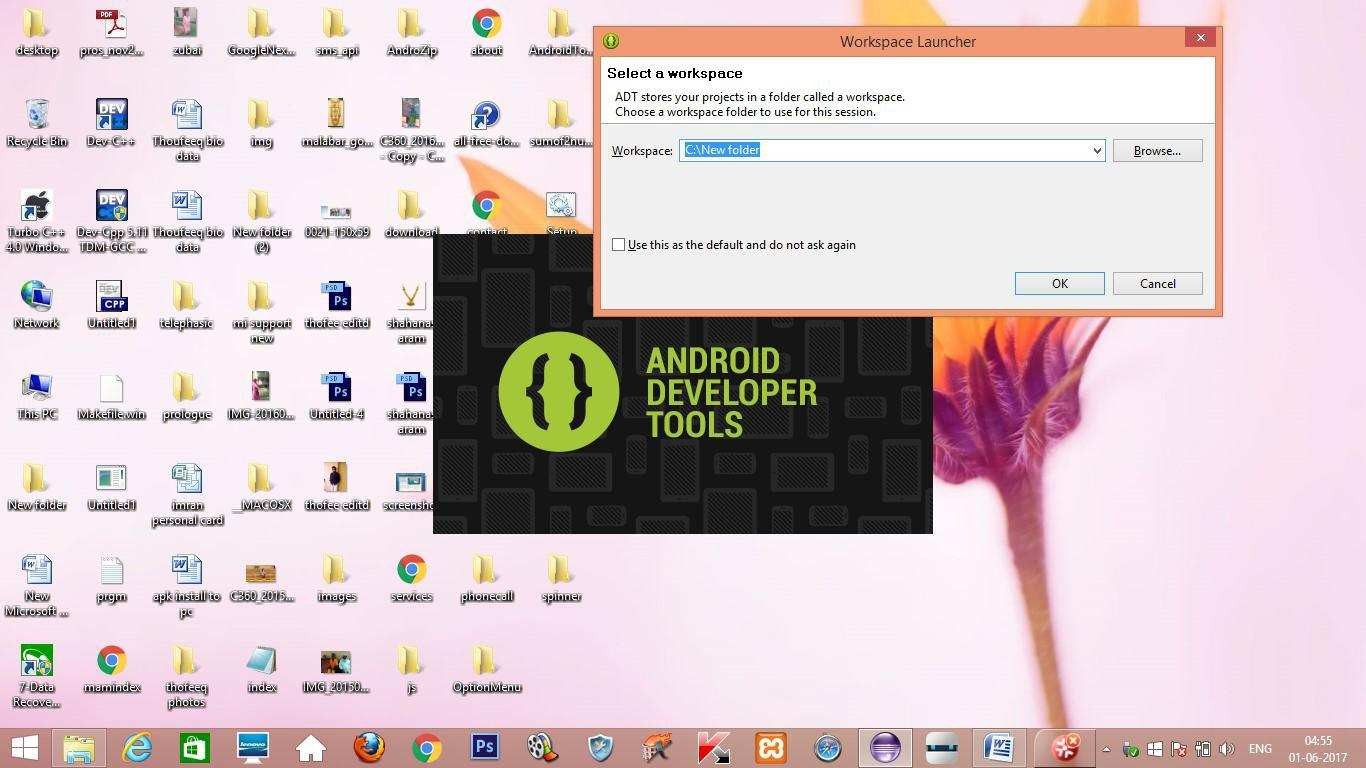
**Step 11 :** Select **All Programs > Android SDK Tools > SDK Manager**, this will give you following window.

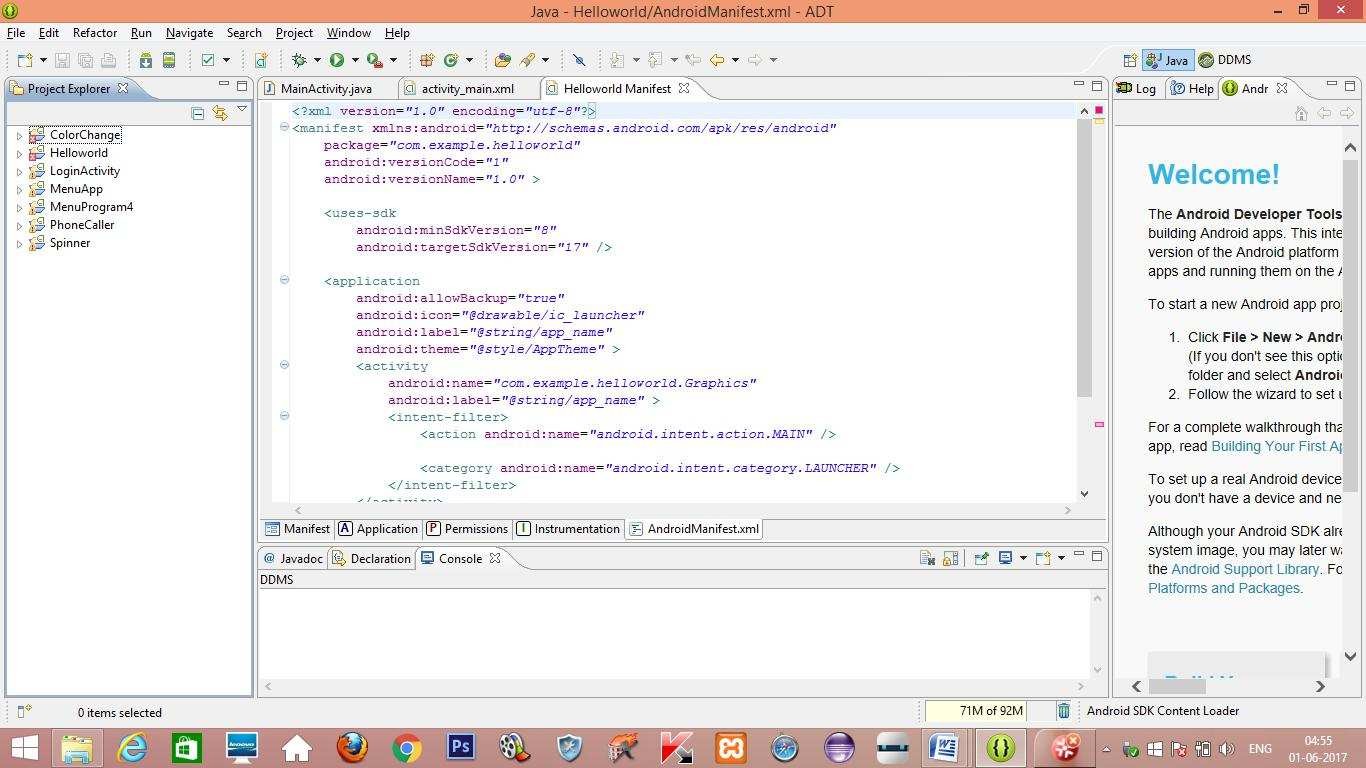
**Step 12 :** On the **Android SDK Manager** Window, Select **lnstall Packages**. **Step 13 :** On the **Choose Packages to Install** Window, Check **Accept,**and Click **Install**.

**Step 14 : Android SDK** Will be installed.

##### Output:







**Result:**

The above aim of the program has been achieved successfully.

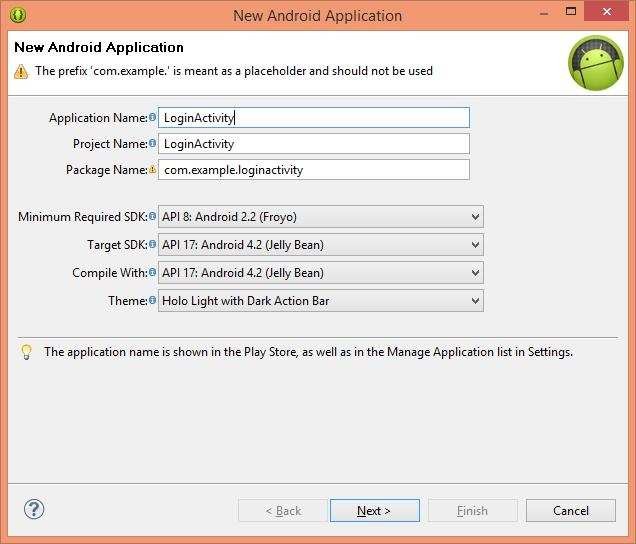
#### LOGIN ACTIVITY

##### Aim:

To create an android application with login operations.

##### Procedure:

**Step 1:** Open Eclipse IDE and go to **File -> New** -> **Project** -> **Android** -> **Android Application Project**. You have to specify the **Application Name**, the **Project Name** and the **Package name** in the appropriate text fields and then click **Next**.

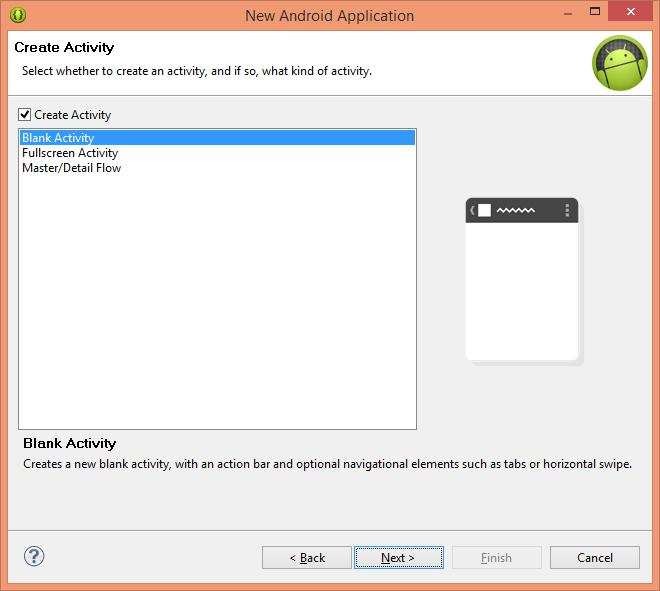


**Step 2:** In the next window make sure the select the **Application Launcher Icon**

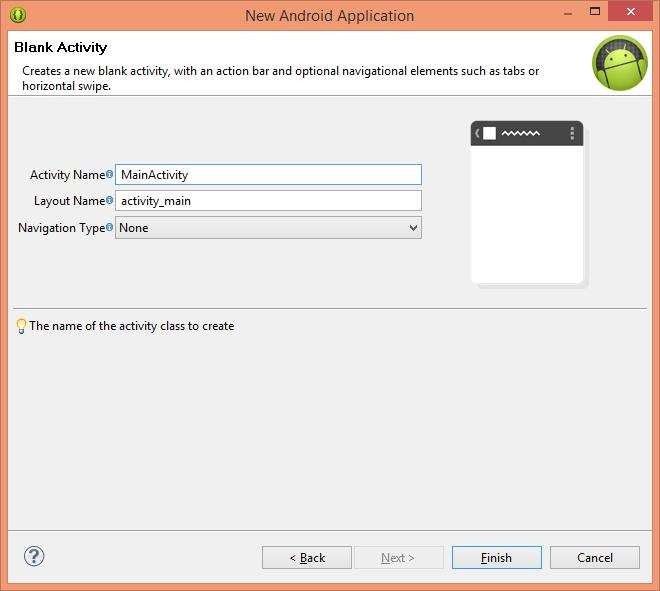
option is selected in order to create a new activity for your project, and click **Next**.



**Step 3:** On the next window click to select the **Blank Activity** and click **Next**.



**Step 4:** On the next window, name the specify **Activity Name, Layout Name and Navigation Type,** click **Next**.



**Step 5:** Drag and Drop All the needed Components from the Palette Window to the **Design View of** the **Activity main.xml.**

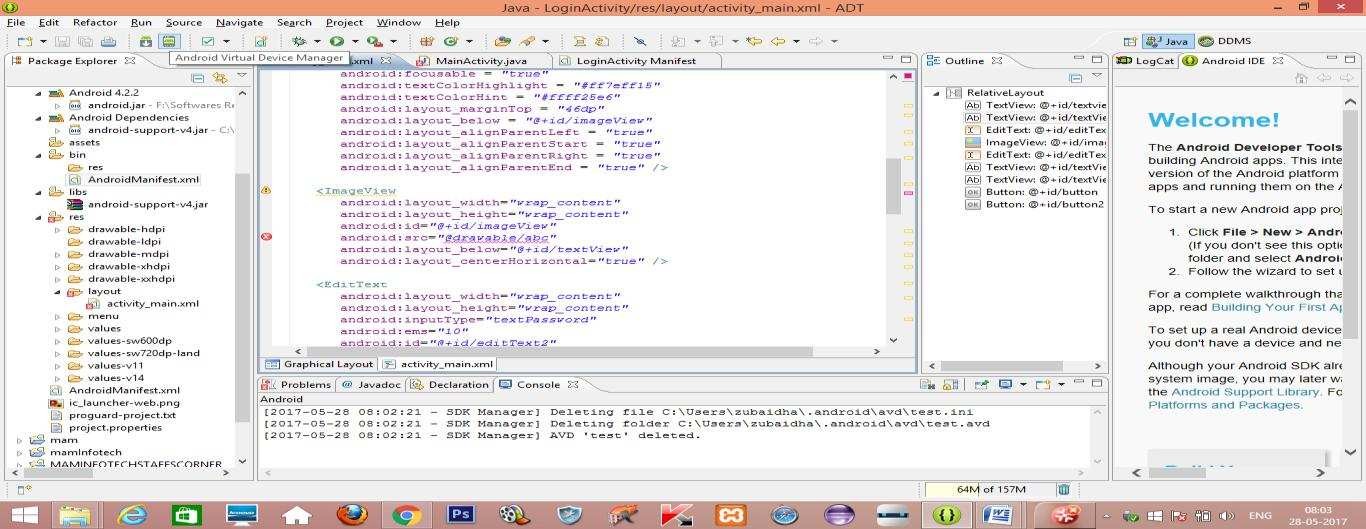
**Step 6:** Create a new activity named **Success.xml,** and add some content.

**Step 7:** Code the operations that are want to perform in the **MainActivity.java, of**

the preferred **.Xml** files.

**Step 8:** Create a new activity named **Success.xml,** and add some content for mobility.

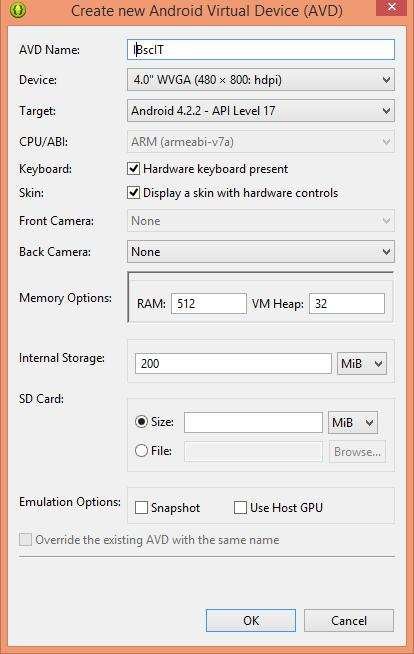
**Step 8:** Create an **AVD, by** Clicking the **Android Virtual Device Icon, on** the **Toolbar.**



**Step 8:** On the window, select **New.**

**Step 9:** On the next window, specify **AVD Name, Device, Target** and **Other Details**

and Click **OK.**



**Step 10:** Now a new **AVD** is created**.**

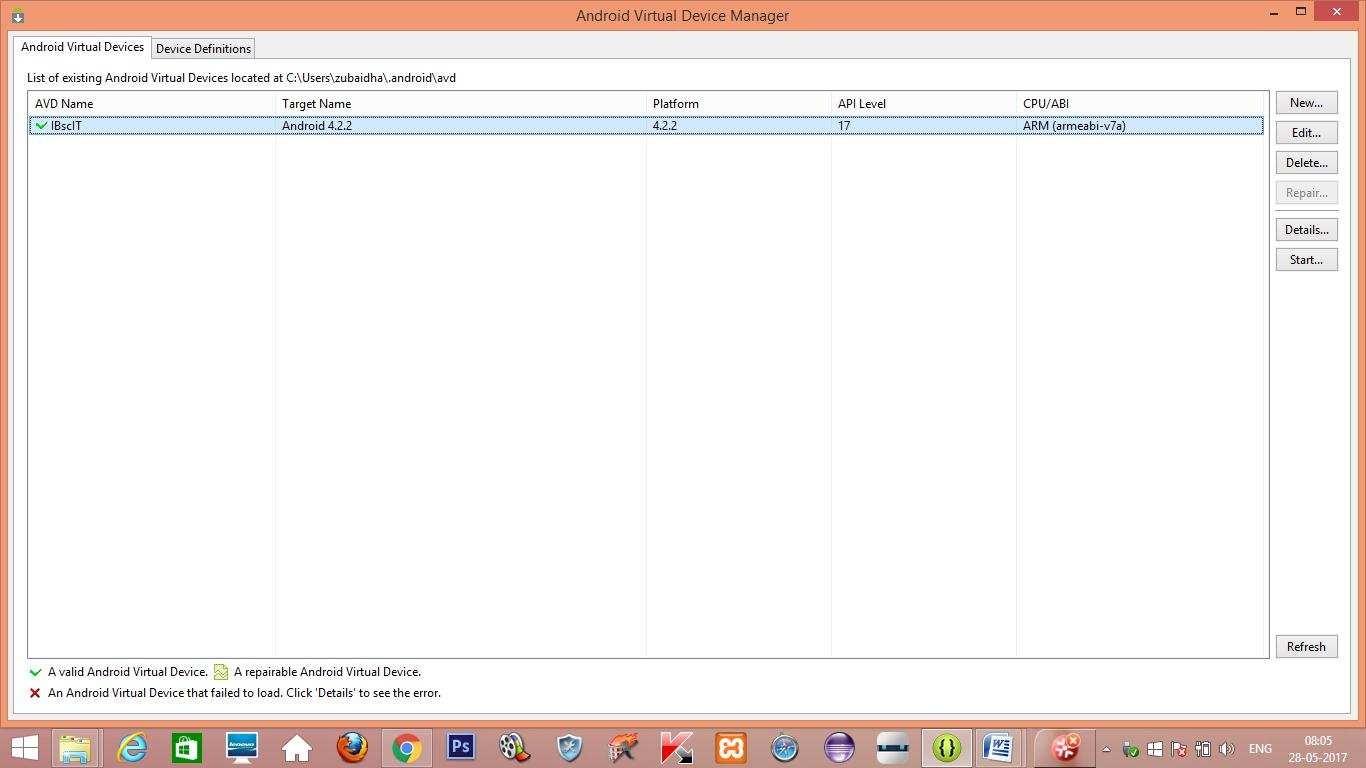
**Step 11:** On Completion of the **Project,** select **File ->Save All.**

**Step 12:** Select **Debug Icon** on the **Toolbar,** to debug the application.

**Step 13:** To run the application Select, **Run Icon** on the **Toolbar.**

**Step 14:** On the Android **Virtual Device Manager** window, select **AVD Name and**, Click

##### Start.



**Step 15:** On the Launch **Options** window, Check Wipe **User Data and**, Click **Launch.**

##### Program:

**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:paddingBottom="@dimen/activity\_vertical\_margin" android:paddingLeft="@dimen/activity\_horizontal\_margin" android:paddingRight="@dimen/activity\_horizontal\_margin" android:paddingTop="@dimen/activity\_vertical\_margin" tools:context=".MainActivity" >

<TextView android:id="@+id/textview"

android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignParentTop="true" android:layout\_centerHorizontal="true" android:text="Login Screen" android:textSize="35dp" />

<TextView android:id="@+id/textView" android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" android:layout\_below="@+id/textview" android:layout\_centerHorizontal="true" android:text="II B.sc IT" android:textColor="#ff7aff24" android:textSize="35dp" />

<EditText

android:layout\_width = "wrap\_content" android:layout\_height = "wrap\_content" android:id = "@+id/editText" android:hint = "Enter Name" android:focusable = "true" android:textColorHighlight = "#ff7eff15" android:textColorHint = "#ffff25e6" android:layout\_marginTop = "46dp"

android:layout\_below = "@+id/imageView" android:layout\_alignParentLeft = "true" android:layout\_alignParentStart = "true" android:layout\_alignParentRight = "true" android:layout\_alignParentEnd = "true" />

<ImageView android:id="@+id/imageView" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_below="@+id/textView"

android:layout\_centerHorizontal="true" android:src="@drawable/ic\_launcher" />

<EditText android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:inputType="textPassword" android:ems="10" android:id="@+id/editText2" android:layout\_below="@+id/editText" android:layout\_alignParentLeft="true" android:layout\_alignParentStart="true" android:layout\_alignRight="@+id/editText" android:layout\_alignEnd="@+id/editText" android:textColorHint="#ffff299f" android:hint="Password" />

<TextView android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="Attempts Left:" android:id="@+id/textView2" android:layout\_below="@+id/editText2" android:layout\_alignParentLeft="true" android:layout\_alignParentStart="true" android:textSize="25dp" />

<TextView android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="New Text" android:id="@+id/textView3" android:layout\_alignTop="@+id/textView2" android:layout\_alignParentRight="true" android:layout\_alignParentEnd="true" android:layout\_alignBottom="@+id/textView2" android:layout\_toEndOf="@+id/textview" android:textSize="25dp" android:layout\_toRightOf="@+id/textview" />

<Button android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="login" android:id="@+id/button" android:layout\_alignParentBottom="true" android:layout\_toLeftOf="@+id/textview"

android:layout\_toStartOf="@+id/textview" />

<Button android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"

android:text="Cancel" android:id="@+id/button2" android:layout\_alignParentBottom="true" android:layout\_toRightOf="@+id/textview" android:layout\_toEndOf="@+id/textview" />

</RelativeLayout>

##### MainActivity.java

package com.example.loginactivity;

import android.app.Activity; import android.graphics.Color; import android.os.Bundle; import android.view.View;

import android.widget.Button; import android.widget.EditText; import android.widget.TextView; import android.widget.Toast;

public class MainActivity extends Activity { Button b1,b2;

EditText ed1,ed2;

TextView tx1; int counter = 3;

@Override

protected void onCreate(Bundle savedInstanceState)

{ super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main);

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

ed1 = (EditText)findViewById(R.id.editText); ed2 = (EditText)findViewById(R.id.editText2);

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

tx1 = (TextView)findViewById(R.id.textView3); tx1.setVisibility(View.GONE);

b1.setOnClickListener(new View.OnClickListener()

{ @Override

public void onClick(View v) { if(ed1.getText().toString().equals("bscit") &&

ed2.getText().toString().equals("bscit")) { Toast.makeText(getApplicationContext(),

"Redirecting...",Toast.LENGTH\_SHORT).show();

}else{ Toast.makeText(getApplicationContext(), "Wrong Credentials",Toast.LENGTH\_SHORT).show();

tx1.setVisibility(View.VISIBLE); tx1.setBackgroundColor(Color.RED); counter--; tx1.setText(Integer.toString(counter));

if (counter == 0) { b1.setEnabled(false);

}

}

}

});

b2.setOnClickListener(new View.OnClickListener()

{ @Override

public void onClick(View v) { finish();

}

});

}

}

##### Android manifest.xml

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

<manifest xmlns:android="<http://schemas.android.com/apk/res/android>" package="com.example.loginactivity"

android:versionCode="1" android:versionName="1.0" >

<uses-sdk android:minSdkVersion="8" android:targetSdkVersion="17" />

<application android:allowBackup="true"

android:icon="@drawable/ic\_launcher" android:label="@string/app\_name" android:theme="@style/AppTheme" >

<activity android:name="com.example.loginactivity.MainActivity" android:label="@string/app\_name" >

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER"

/> </intent-filter>

</activity>

</application>

</manifest>

##### Strings.xml

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

<resources>

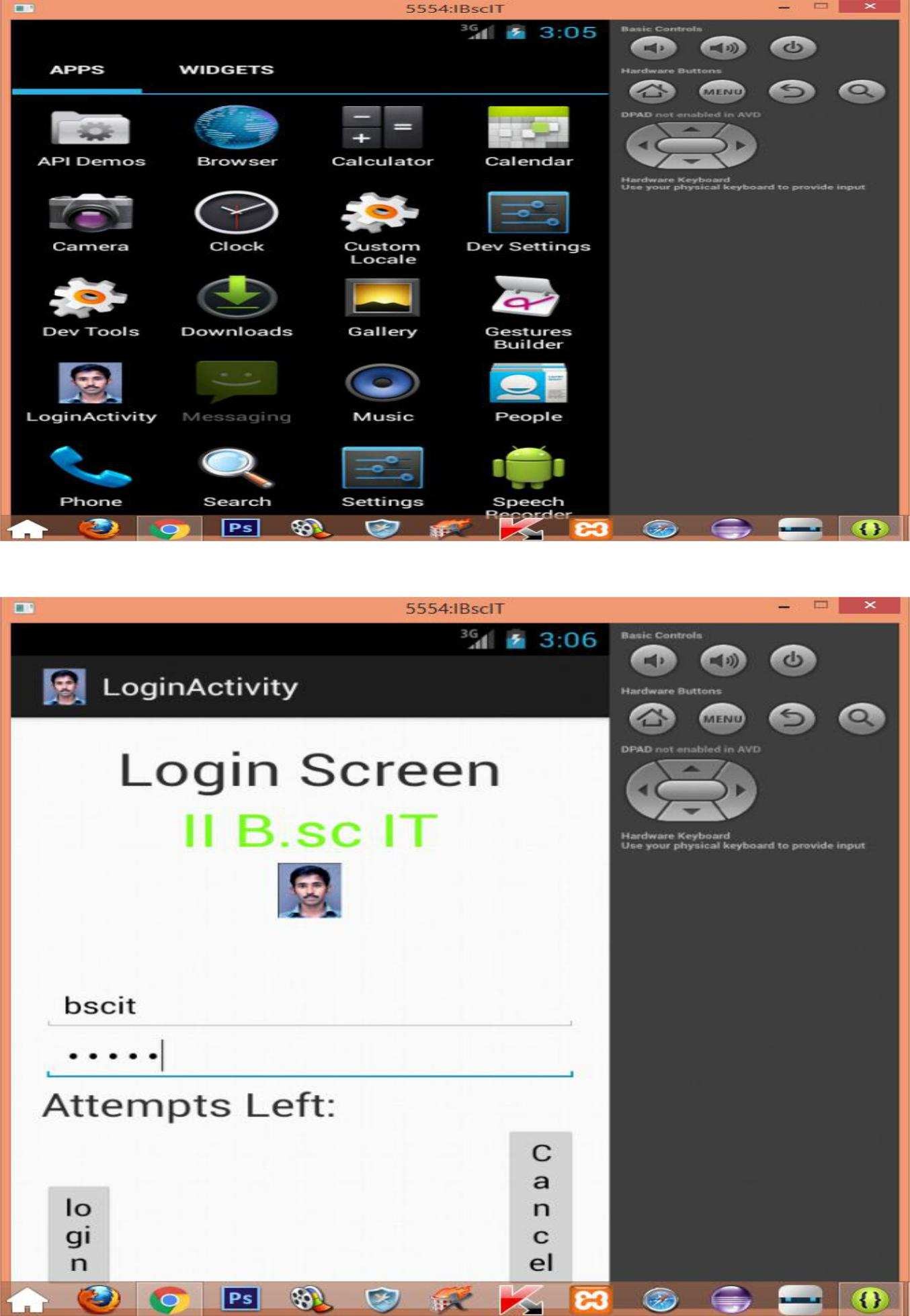
<string name="app\_name">LoginActivity</string>

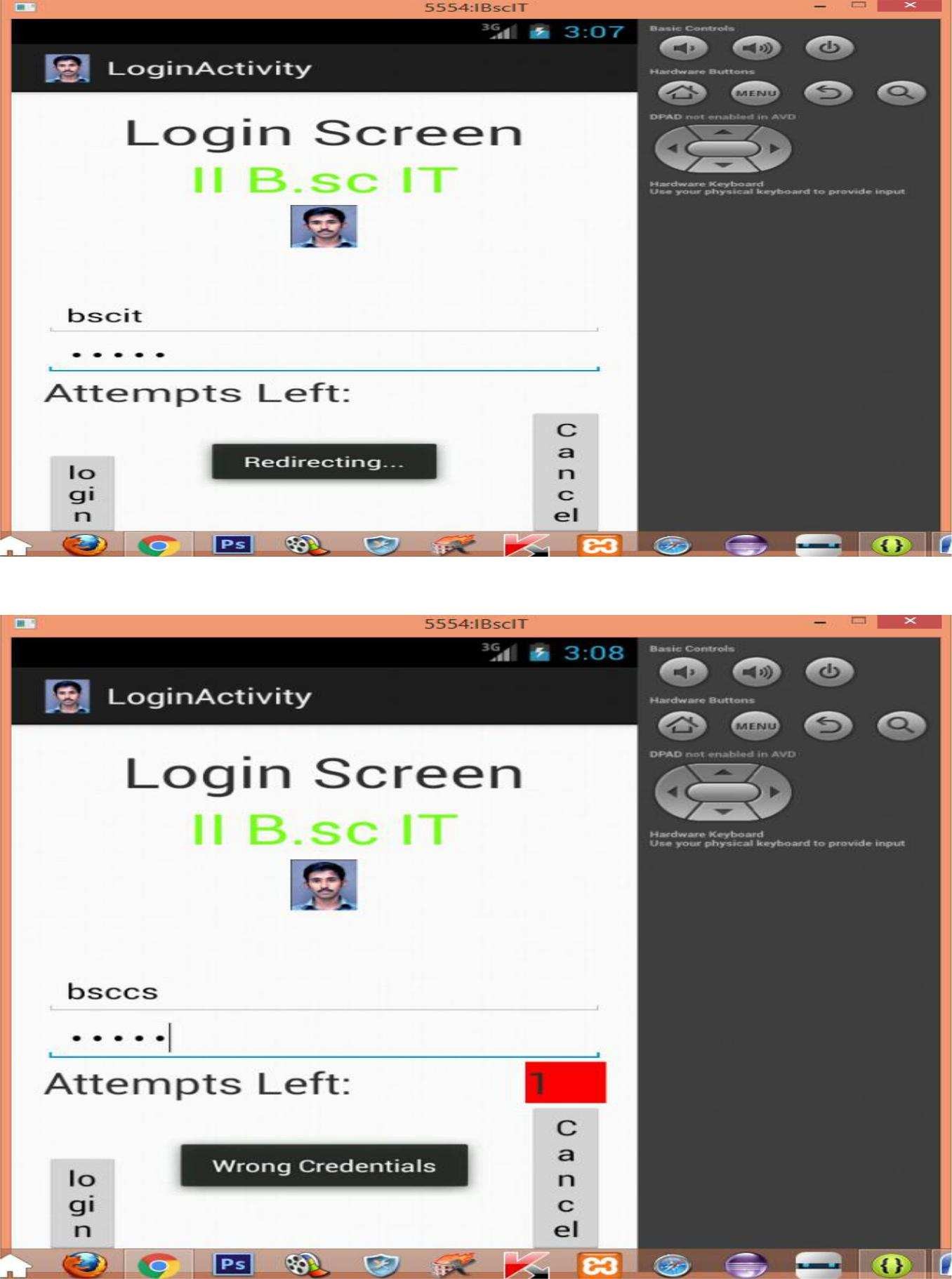
<string name="action\_settings">Settings</string>

<string name="hello\_world">Hello world!</string>

</resources>

##### Output:





**Result:**

The above aim of the program has been achieved successfully.

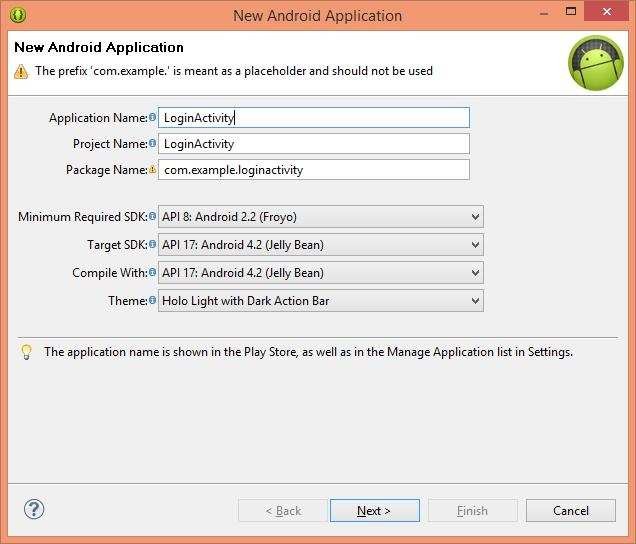
#### HELLO WORLD WITH BASIC INTERACTIONS

##### Aim:

To create an android application with login operations.

##### Procedure:

**Step 1:** Open Eclipse IDE and go to **File -> New** -> **Project** -> **Android** -> **Android Application Project**. You have to specify the **Application Name**, the **Project Name** and the **Package name** in the appropriate text fields and then click **Next**.

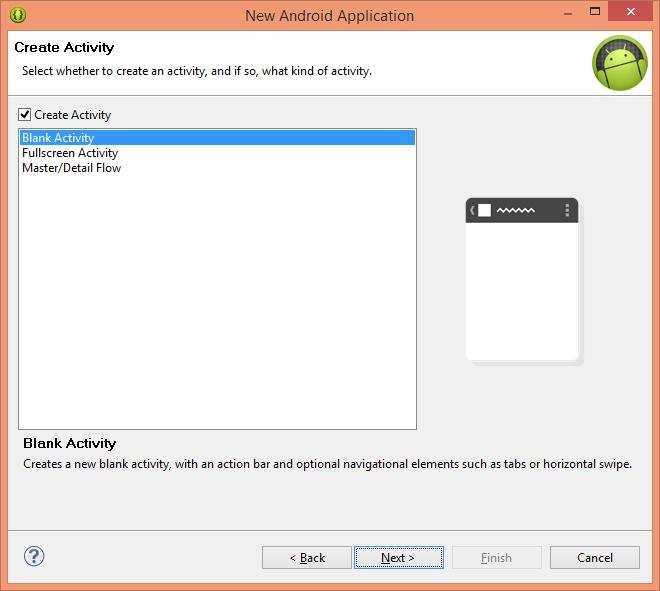


**Step 2:** In the next window make sure the select the **Application Launcher Icon**

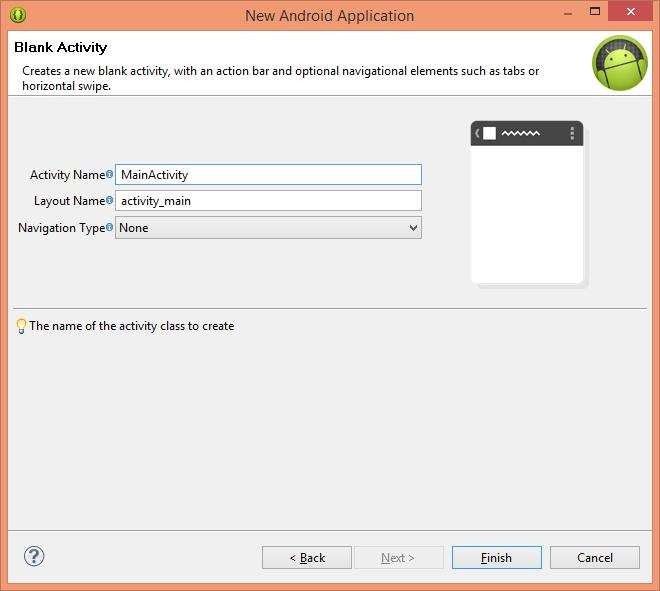
option is selected in order to create a new activity for your project, and click **Next**.



**Step 3:** On the next window click to select the **Blank Activity** and click **Next**.



**Step 4:** On the next window, name the specify **Activity Name, Layout Name and Navigation Type,** click **Next**.



**Step 5:** Drag and Drop All the needed Components from the Palette Window to the **Design View of** the **Activity main.xml.**

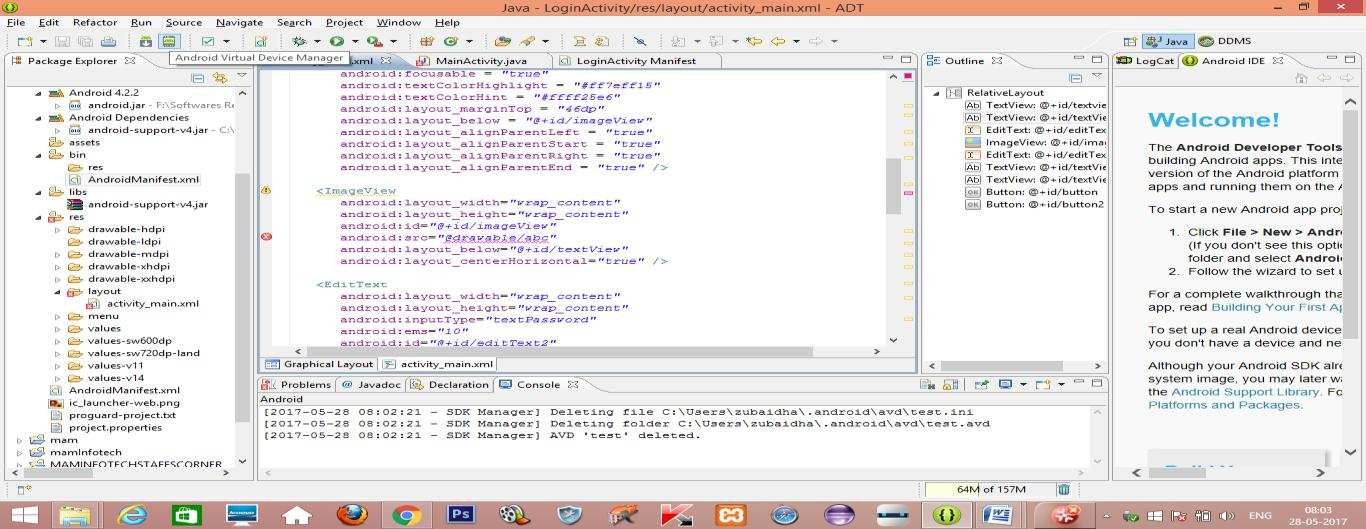
**Step 6:** Create a new activity named **Success.xml,** and add some content.

**Step 7:** Code the operations that are want to perform in the **MainActivity.java, of**

the preferred **.Xml** files.

**Step 8:** Create a new activity named **Success.xml,** and add some content for mobility.

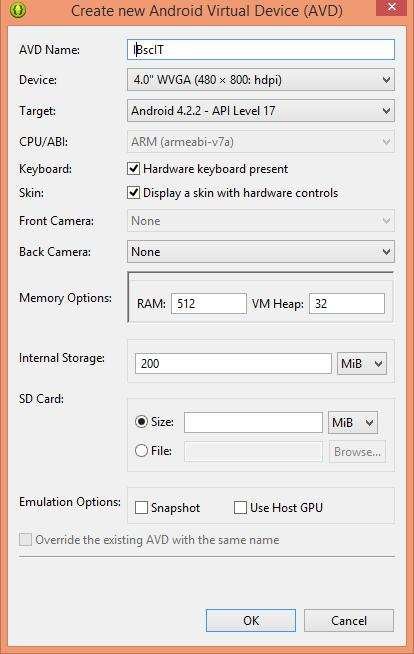
**Step 8:** Create an **AVD, by** Clicking the **Android Virtual Device Icon, on** the **Toolbar.**



**Step 8:** On the window, select **New.**

**Step 9:** On the next window, specify **AVD Name, Device, Target** and **Other Details**

and Click **OK.**



**Step 10:** Now a new **AVD** is created**.**

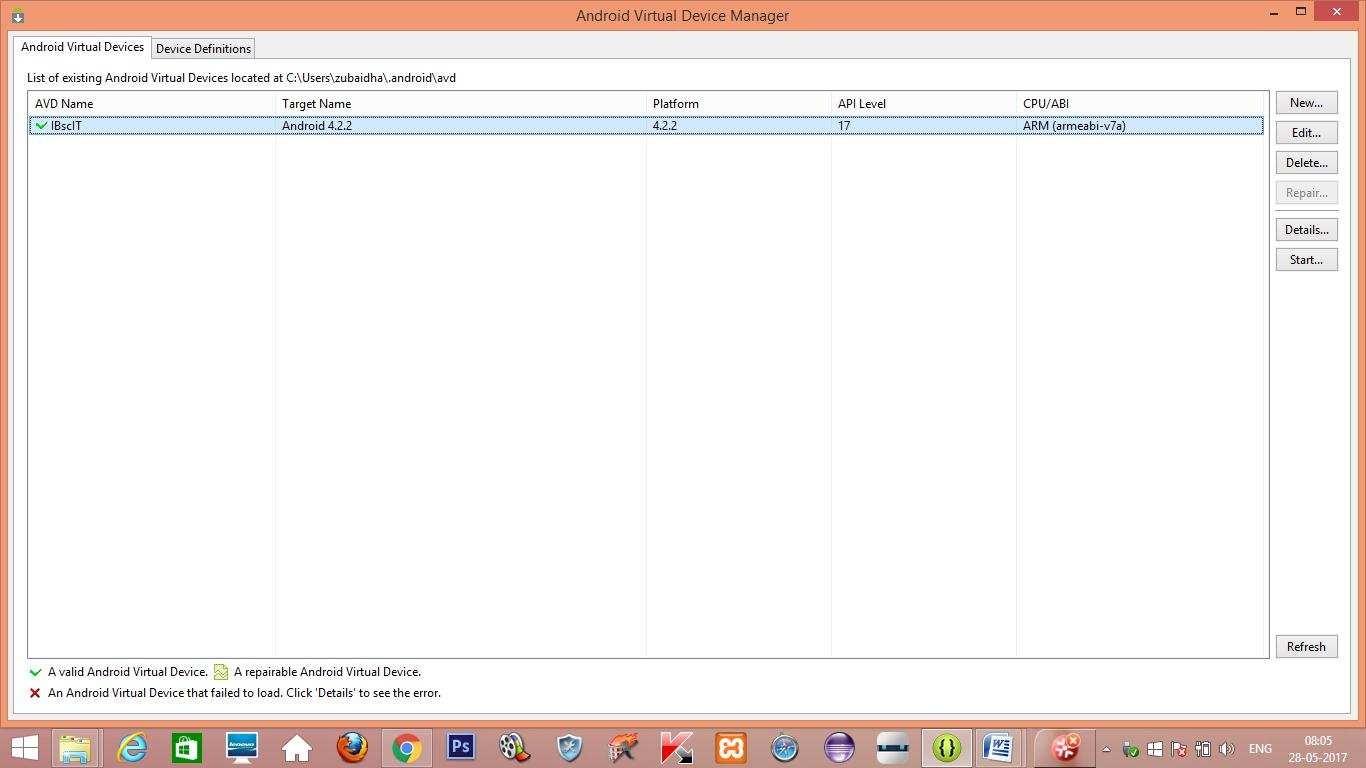
**Step 11:** On Completion of the **Project,** select **File ->Save All.**

**Step 12:** Select **Debug Icon** on the **Toolbar,** to debug the application.

**Step 13:** To run the application Select, **Run Icon** on the **Toolbar.**

**Step 14:** On the Android **Virtual Device Manager** window, select **AVD Name and**, Click

##### Start.



**Step 15:** On the Launch **Options** window, Check Wipe **User Data and**, Click **Launch.**

##### Program:

**Activity\_main.xml**

<RelativeLayout xmlns:android="<http://schemas.android.com/apk/res/android>" xmlns:tools="<http://schemas.android.com/tools>"

android:id="@+id/bu" android:layout\_width="match\_parent" android:layout\_height="match\_parent" android:paddingBottom="@dimen/activity\_vertical\_margin" android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin" android:paddingTop="@dimen/activity\_vertical\_margin" tools:context=".MainActivity" >

<TextView android:id="@+id/textView1" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="@string/hello\_world" />

<ListView android:id="@+id/mobile\_list" android:layout\_width="match\_parent" android:layout\_height="100dp" android:layout\_above="@+id/b1"

android:layout\_alignLeft="@+id/textView1" android:layout\_marginBottom="26dp" >

</ListView>

<Button

android:id="@+id/b1" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignParentBottom="true" android:layout\_marginBottom="68dp" android:layout\_toRightOf="@+id/textView" android:text="Next" />

<ImageView android:id="@+id/imageView" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"

android:layout\_above="@+id/mobile\_list" android:layout\_alignRight="@+id/textView" android:layout\_marginRight="29dp" android:src="@drawable/ic\_launcher" />

<TextView android:id="@+id/textView" android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" android:layout\_above="@+id/imageView"

android:layout\_centerHorizontal="true" android:text="II B.sc IT" android:textColor="#ff7aff24" android:textSize="35dp" />

<TextView android:id="@+id/textview2" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"

android:layout\_above="@+id/textView" android:layout\_alignRight="@+id/mobile\_list" android:layout\_marginBottom="14dp" android:text="Basic Interactions" android:textSize="35dp" />

</RelativeLayout>

##### MainActivity.java

package com.example.helloworld;

import android.os.Bundle; import android.app.Activity; import android.view.Menu;

import android.widget.ArrayAdapter; import android.widget.ListView;

public class MainActivity extends Activity {

private Button next;

String[] mobileArray = {"Bsc IT","BSc CS"};

@Override

protected void onCreate(Bundle savedInstanceState)

{ super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main);

//next = (Button)findViewById(R.id.b1);

//addListenerOnButton();

ArrayAdapter adapter = new ArrayAdapter<String>(this,

R.layout.activity\_listview, mobileArray);

ListView listView = (ListView) findViewById(R.id.mobile\_list); listView.setAdapter(adapter);

}

}

##### Activity\_main2.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:paddingBottom="@dimen/activity\_vertical\_margin" android:paddingLeft="@dimen/activity\_horizontal\_margin" android:paddingRight="@dimen/activity\_horizontal\_margin" android:paddingTop="@dimen/activity\_vertical\_margin" tools:context=".Main2Activity" >

<TextView android:id="@+id/textView2" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="@string/hello\_world" />

<TextView android:id="@+id/textView1" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"

android:layout\_below="@+id/textView2" android:layout\_marginLeft="58dp" android:layout\_marginTop="202dp" android:layout\_toRightOf="@+id/textView2" android:hint="Text Box"

android:text="II BSC IT" />

</RelativeLayout>

##### Main2Activity.java

package com.example.helloworld; import android.os.Bundle;

import android.app.Activity; import android.view.Menu; import android.content.Context; import android.graphics.Canvas; import android.graphics.Color; import android.graphics.Paint; import android.view.View; import android.app.Activity; import android.view.Menu;

public class Main2Activity extends Activity { DemoView demoview;

@Override

public void onCreate(Bundle savedInstanceState)

{ super.onCreate(savedInstanceState); demoview = new DemoView(this); setContentView(demoview);

}

private class DemoView extends View{ public DemoView(Context context){

super(context);

}

@Override protected void onDraw(Canvas canvas)

{ super.onDraw(canvas);

// custom drawing code here Paint paint = new Paint(); paint.setStyle(Paint.Style.FILL);

// make the entire canvas white paint.setColor(Color.WHITE); canvas.drawPaint(paint);

// draw blue circle with anti aliasing turned off paint.setAntiAlias(false); paint.setColor(Color.BLUE); canvas.drawCircle(20, 20, 15, paint);

// draw green circle with anti aliasing turned on paint.setAntiAlias(true); paint.setColor(Color.GREEN); canvas.drawCircle(60, 20, 15, paint);

// draw red rectangle with anti aliasing turned off paint.setAntiAlias(false); paint.setColor(Color.RED); canvas.drawRect(100, 5, 200, 30, paint);

// draw the rotated text canvas.rotate(-45);

paint.setStyle(Paint.Style.FILL); canvas.drawText("Graphics Rotation", 40, 180, paint);

//undo the rotate

canvas.restore();

}

}

@Override

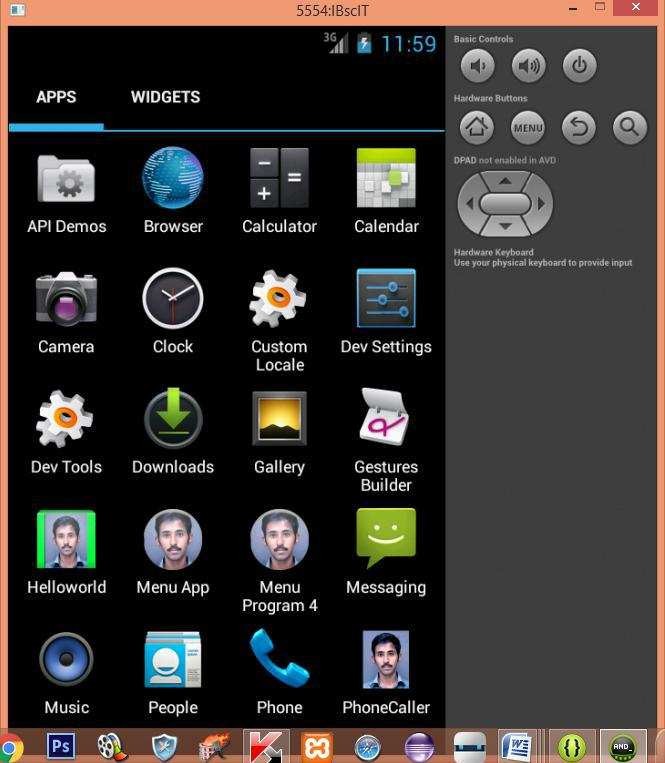
public boolean onCreateOptionsMenu(Menu menu) {

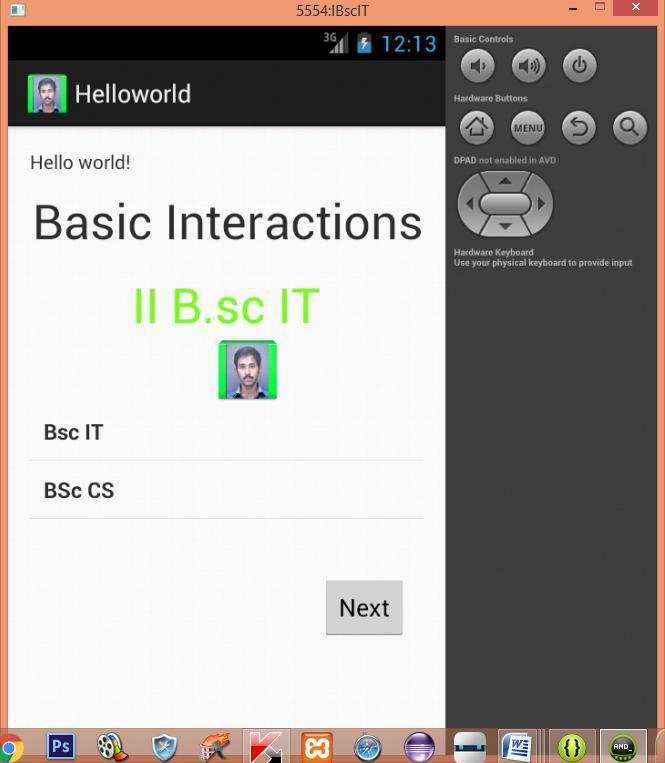
// Inflate the menu; this adds items to the action bar if it is present. getMenuInflater().inflate(R.menu.main2, menu); return true;

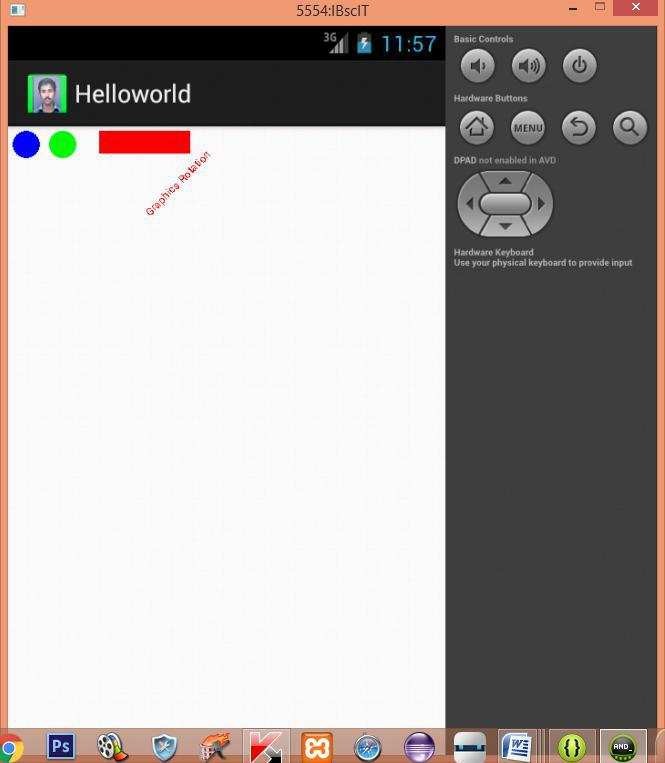
}

}

##### Output:







**Result:**

The above aim of the program has been achieved successfully.

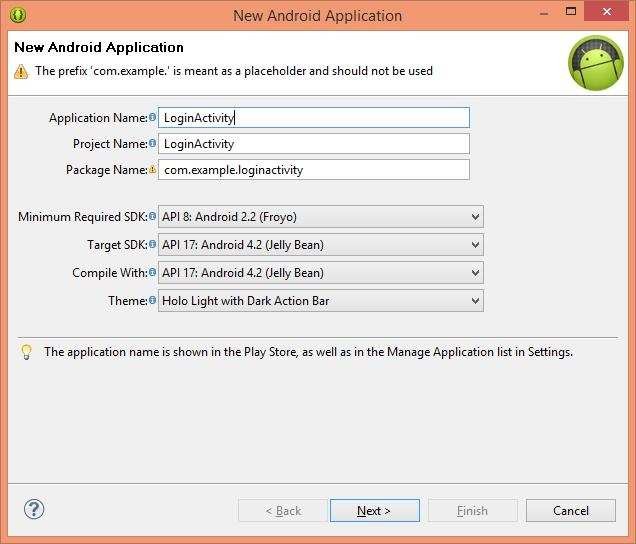
#### MENU ACTIVITY

##### Aim:

To create an android application with menu functions.

##### Procedure:

**Step 1:** Open Eclipse IDE and go to **File -> New** -> **Project** -> **Android** -> **Android Application Project**. You have to specify the **Application Name**, the **Project Name** and the **Package name** in the appropriate text fields and then click **Next**.

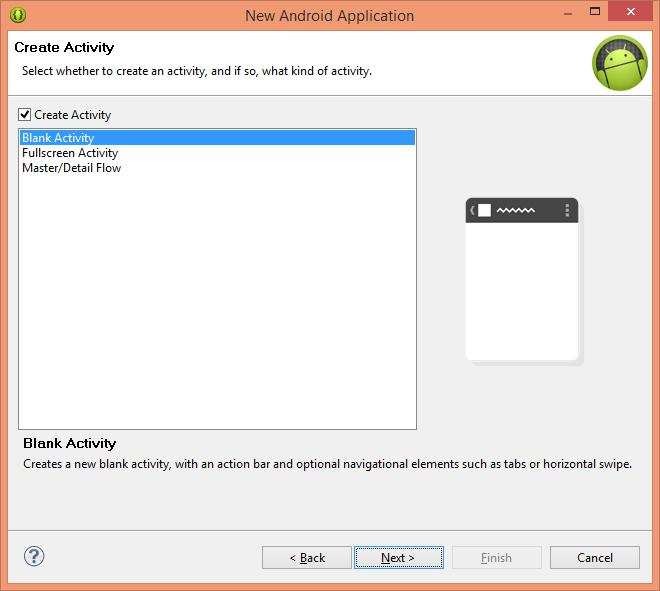


**Step 2:** In the next window make sure the select the **Application Launcher Icon**

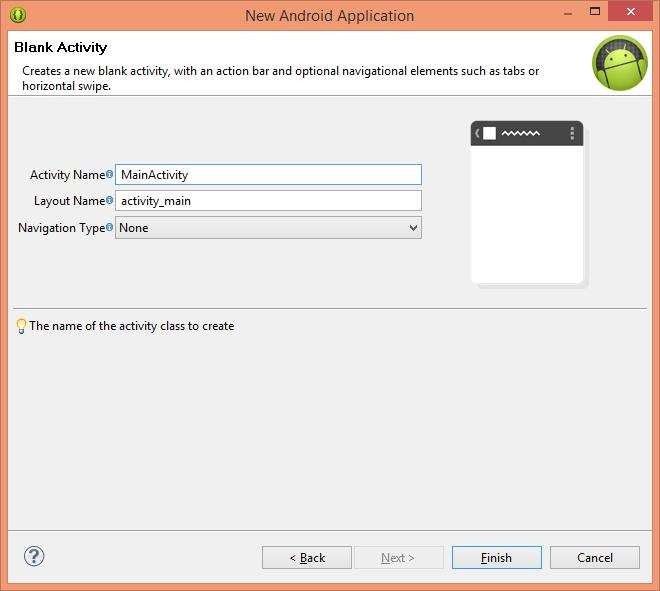
option is selected in order to create a new activity for your project, and click **Next**.



**Step 3:** On the next window click to select the **Blank Activity** and click **Next**.



**Step 4:** On the next window, name the specify **Activity Name, Layout Name and Navigation Type,** click **Next**.



**Step 5:** Drag and Drop All the needed Components from the Palette Window to the **Design View of** the **Activity main.xml.**

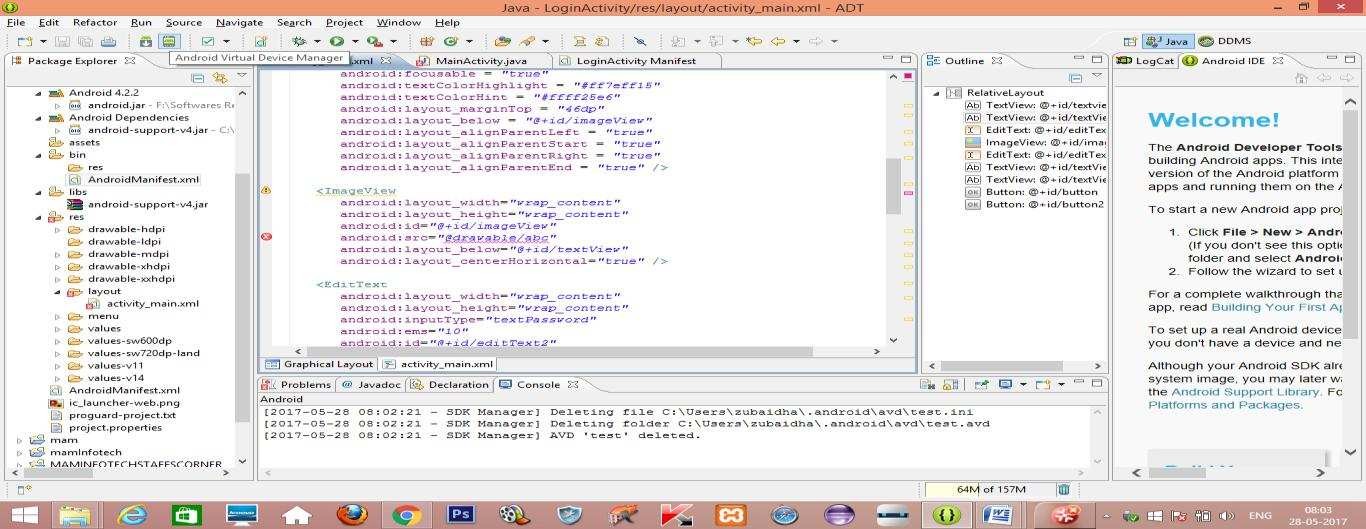
**Step 6:** Create a new activity named **Success.xml,** and add some content.

**Step 7:** Code the operations that are want to perform in the **MainActivity.java, of**

the preferred **.Xml** files.

**Step 8:** Create a new activity named **Success.xml,** and add some content for mobility.

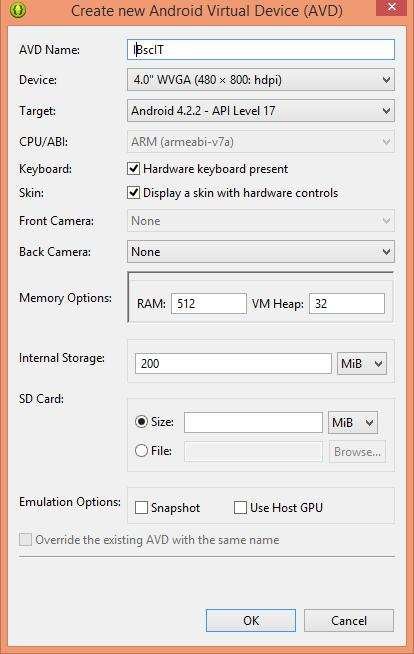
**Step 8:** Create an **AVD, by** Clicking the **Android Virtual Device Icon, on** the **Toolbar.**



**Step 8:** On the window, select **New.**

**Step 9:** On the next window, specify **AVD Name, Device, Target** and **Other Details**

and Click **OK.**



**Step 10:** Now a new **AVD** is created**.**

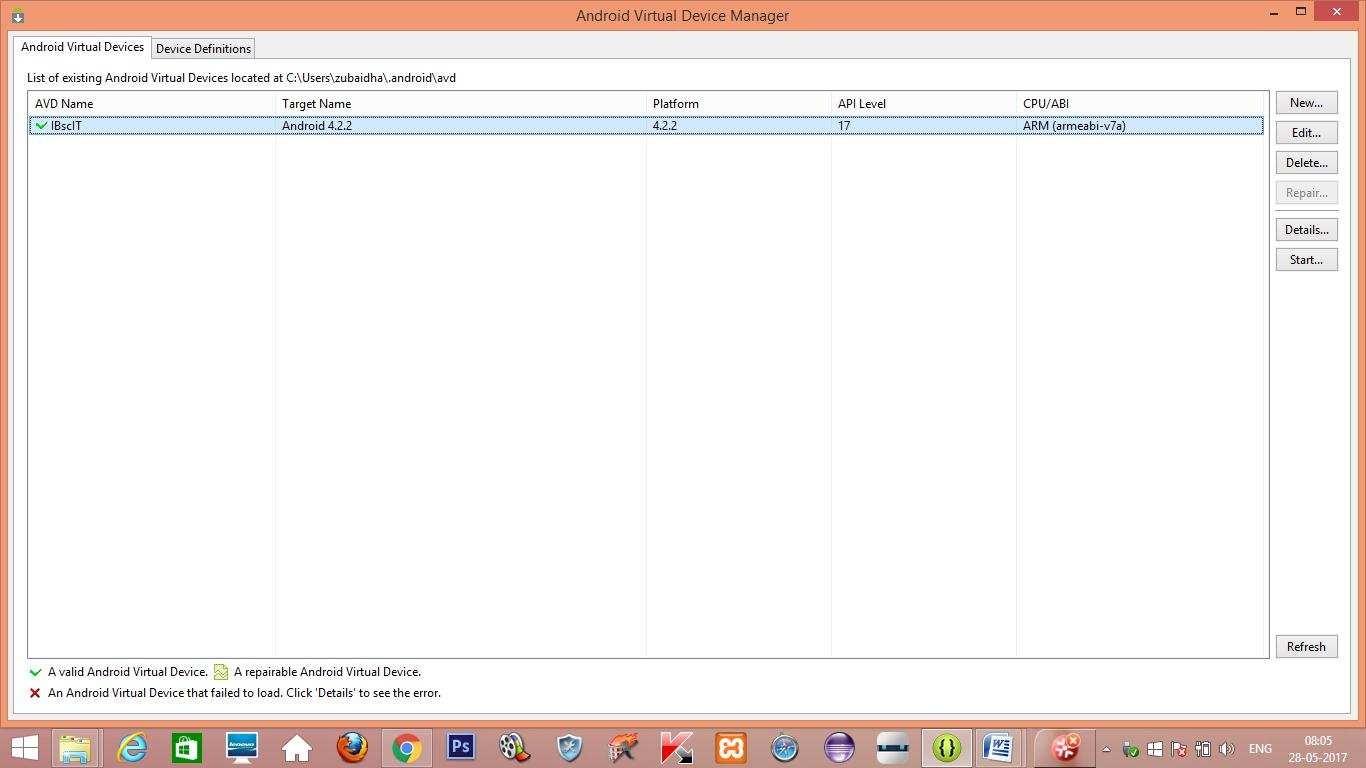
**Step 11:** On Completion of the **Project,** select **File ->Save All.**

**Step 12:** Select **Debug Icon** on the **Toolbar,** to debug the application.

**Step 13:** To run the application Select, **Run Icon** on the **Toolbar.**

**Step 14:** On the Android **Virtual Device Manager** window, select **AVD Name and**, Click

##### Start.



**Step 15:** On the Launch **Options** window, Check Wipe **User Data and**, Click **Launch.**

##### Program:

**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:paddingBottom="@dimen/activity\_vertical\_margin" android:paddingLeft="@dimen/activity\_horizontal\_margin" android:paddingRight="@dimen/activity\_horizontal\_margin" android:paddingTop="@dimen/activity\_vertical\_margin" tools:context=".MainActivity" >

<TextView android:id="@+id/textview" android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" android:layout\_alignParentTop="true" android:layout\_centerHorizontal="true" android:text="Menu Application" android:textSize="35dp" />

<TextView android:id="@+id/textView" android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" android:layout\_below="@+id/textview" android:layout\_centerHorizontal="true" android:text="II B.sc IT" android:textColor="#ff7aff24" android:textSize="35dp" />

<ImageView android:id="@+id/imageView" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"

android:layout\_alignLeft="@+id/textView" android:layout\_below="@+id/textView" android:layout\_marginLeft="28dp" android:src="@drawable/ic\_launcher" />

<Button

android:id="@+id/button1" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignLeft="@+id/imageView" android:layout\_below="@+id/imageView" android:layout\_marginTop="40dp" android:text="Menus" />

</RelativeLayout>

##### Popup\_menu File

<menu xmlns:android="<http://schemas.android.com/apk/res/android>" >

<item

android:id="@+id/one" android:title="Cut"/>

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

<item

android:id="@+id/three" android:title="Paste"/>

<item android:id="@+id/four" android:title="Delete"/>

<item android:id="@+id/five" android:title="Select All"/>

<item

android:id="@+id/six" android:title="Unselect All"/>

</menu>

##### MainActivity.java

package com.example.menuapp;

import android.os.Bundle; import android.app.Activity; import android.view.Menu; import android.view.MenuItem; import android.view.View;

import android.view.View.OnClickListener; import android.widget.Button;

import android.widget.PopupMenu; import android.widget.Toast;

public class MainActivity extends Activity { Button button1;

@Override

protected void onCreate(Bundle savedInstanceState)

{ super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main);

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

button1.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

//Creating the instance of PopupMenu

PopupMenu popup = new PopupMenu(MainActivity.this, button1);

//Inflating the Popup using xml file popup.getMenuInflater().inflate(R.menu.popup\_menu, popup.getMenu());

popup.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener() {

public boolean onMenuItemClick(MenuItem item) { Toast.makeText(MainActivity.this,"You Clicked : " +

item.getTitle(),Toast.LENGTH\_SHORT).show();

return true;

}

});

popup.show();//showing popup menu

}

});

}}

##### Android Manifest.xml

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

<manifest xmlns:android="<http://schemas.android.com/apk/res/android>" package="com.example.menuapp"

android:versionCode="1" android:versionName="1.0" >

<uses-sdk android:minSdkVersion="12" android:targetSdkVersion="17" />

<application android:allowBackup="true"

android:icon="@drawable/ic\_launcher" android:label="@string/app\_name" android:theme="@style/AppTheme" >

<activity

android:name="com.example.menuapp.MainActivity" android:label="@string/app\_name" > <intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER"

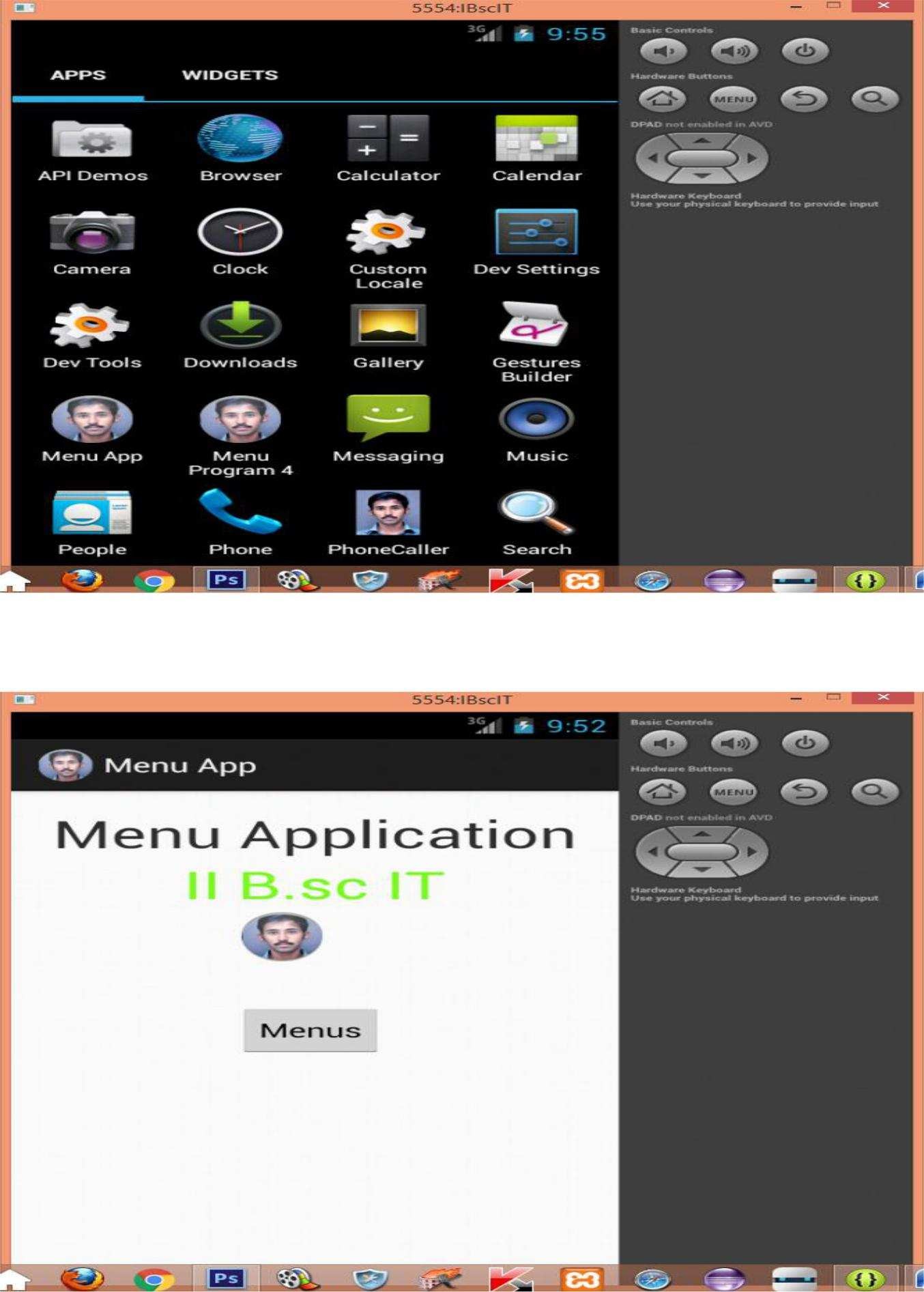
/> </intent-filter>

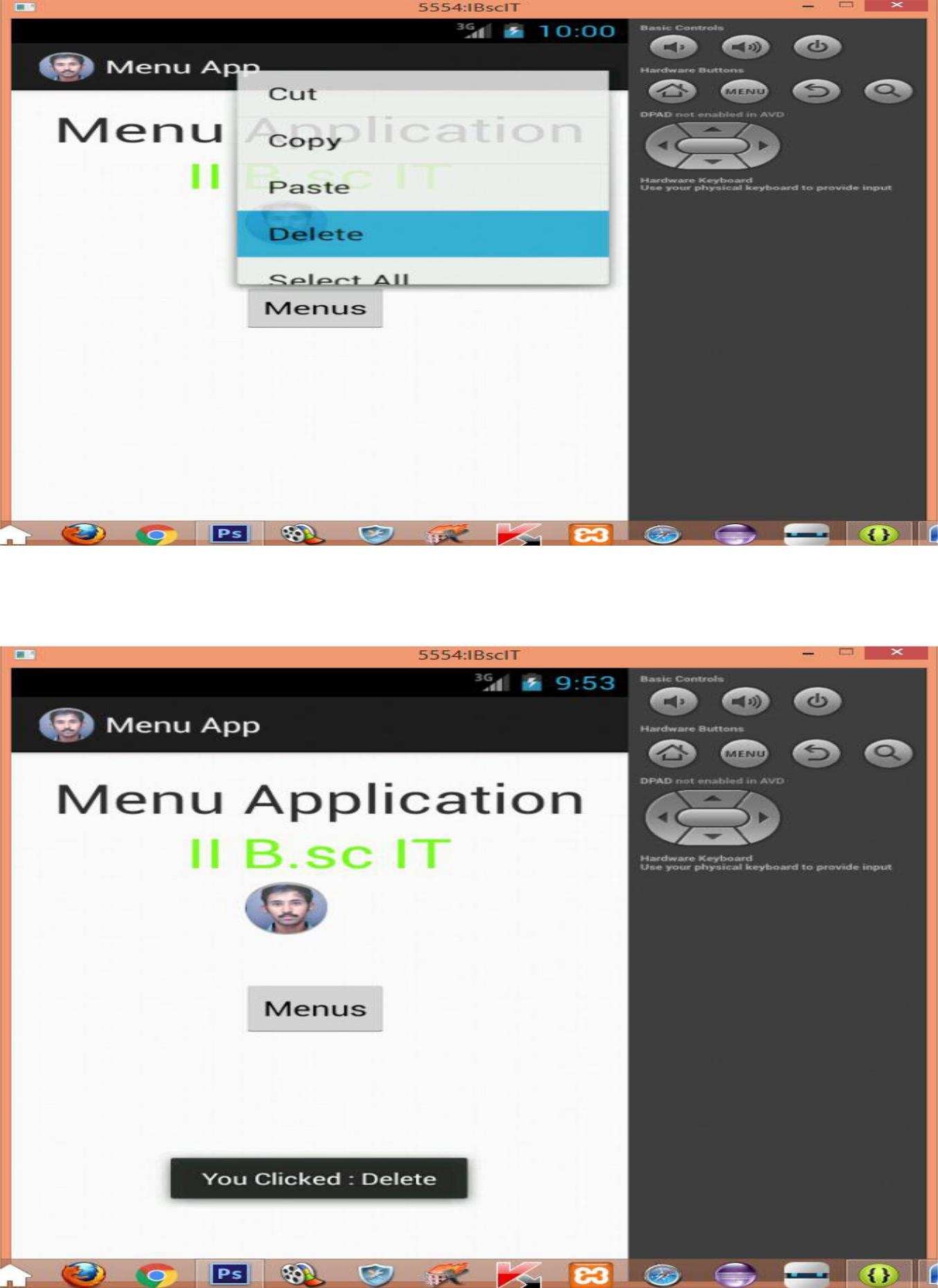
</activity>

</application>

</manifest>

##### Output:





**Result:**

The above aim of the program has been achieved successfully.

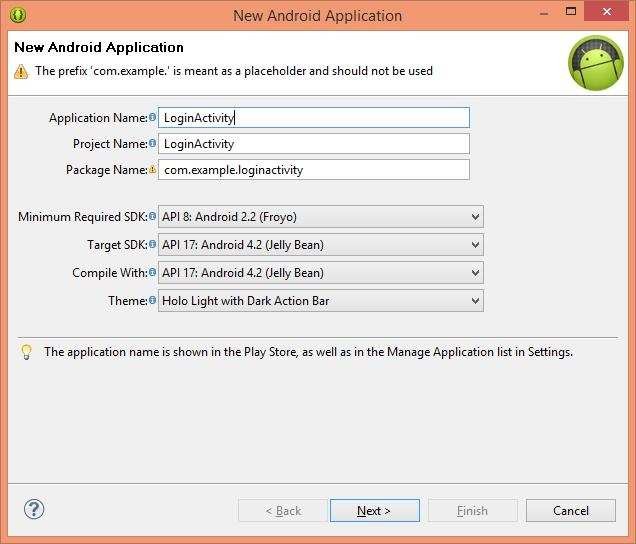
#### SCREEN COLOR CHANGE

##### Aim:

To create an android application with login operations.

##### Procedure:

**Step 1:** Open Eclipse IDE and go to **File -> New** -> **Project** -> **Android** -> **Android Application Project**. You have to specify the **Application Name**, the **Project Name** and the **Package name** in the appropriate text fields and then click **Next**.

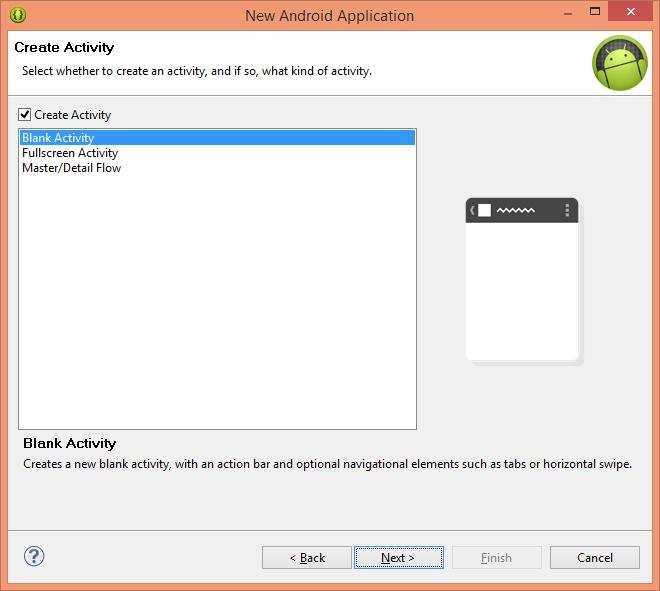


**Step 2:** In the next window make sure the select the **Application Launcher Icon**

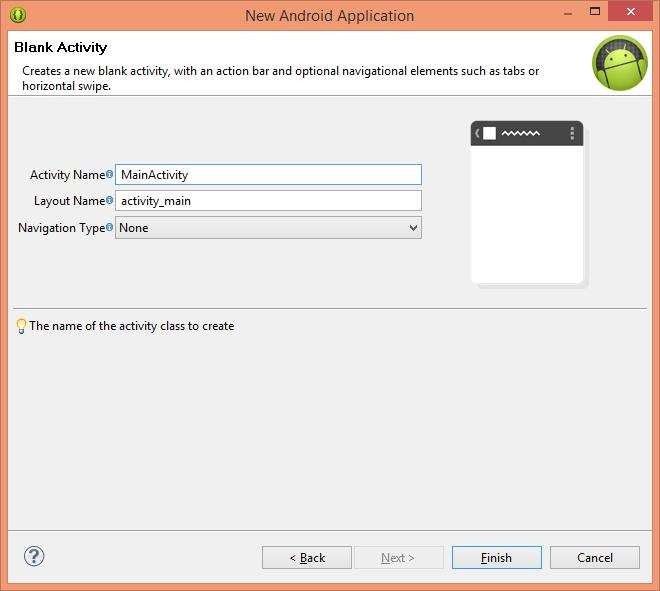
option is selected in order to create a new activity for your project, and click **Next**.



**Step 3:** On the next window click to select the **Blank Activity** and click **Next**.



**Step 4:** On the next window, name the specify **Activity Name, Layout Name and Navigation Type,** click **Next**.



**Step 5:** Drag and Drop All the needed Components from the Palette Window to the **Design View of** the **Activity main.xml.**

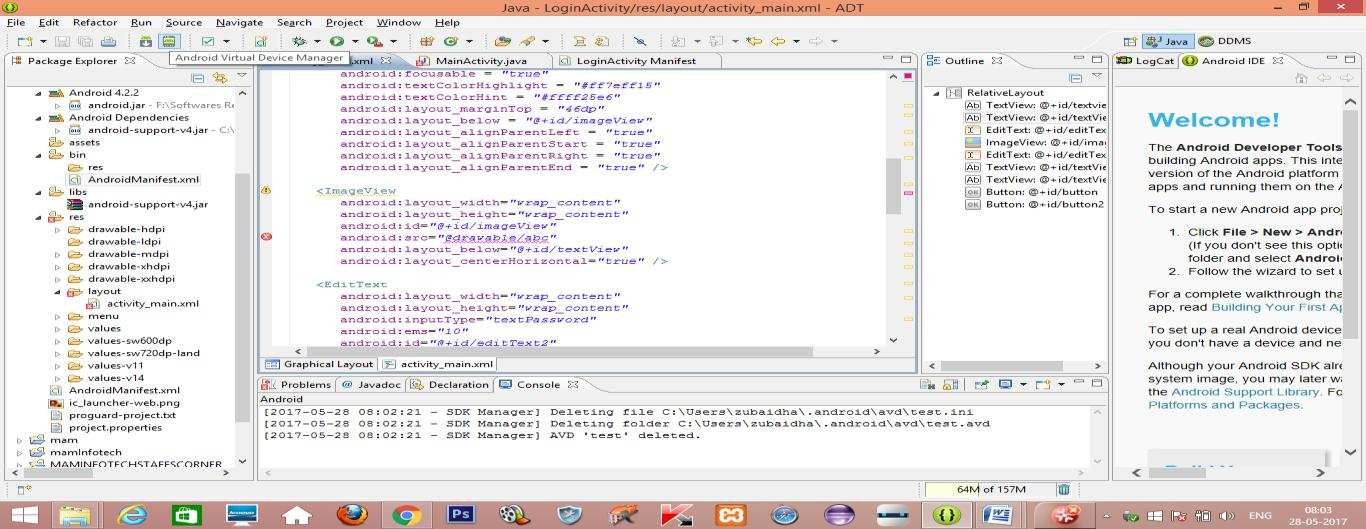
**Step 6:** Create a new activity named **Success.xml,** and add some content.

**Step 7:** Code the operations that are want to perform in the **MainActivity.java, of**

the preferred **.Xml** files.

**Step 8:** Create a new activity named **Success.xml,** and add some content for mobility.

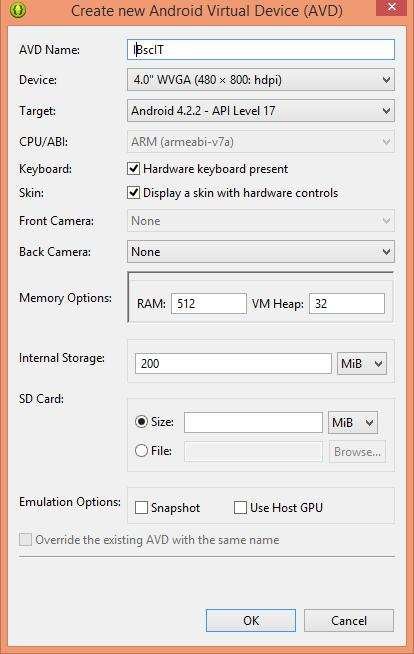
**Step 8:** Create an **AVD, by** Clicking the **Android Virtual Device Icon, on** the **Toolbar.**



**Step 8:** On the window, select **New.**

**Step 9:** On the next window, specify **AVD Name, Device, Target** and **Other Details**

and Click **OK.**



**Step 10:** Now a new **AVD** is created**.**

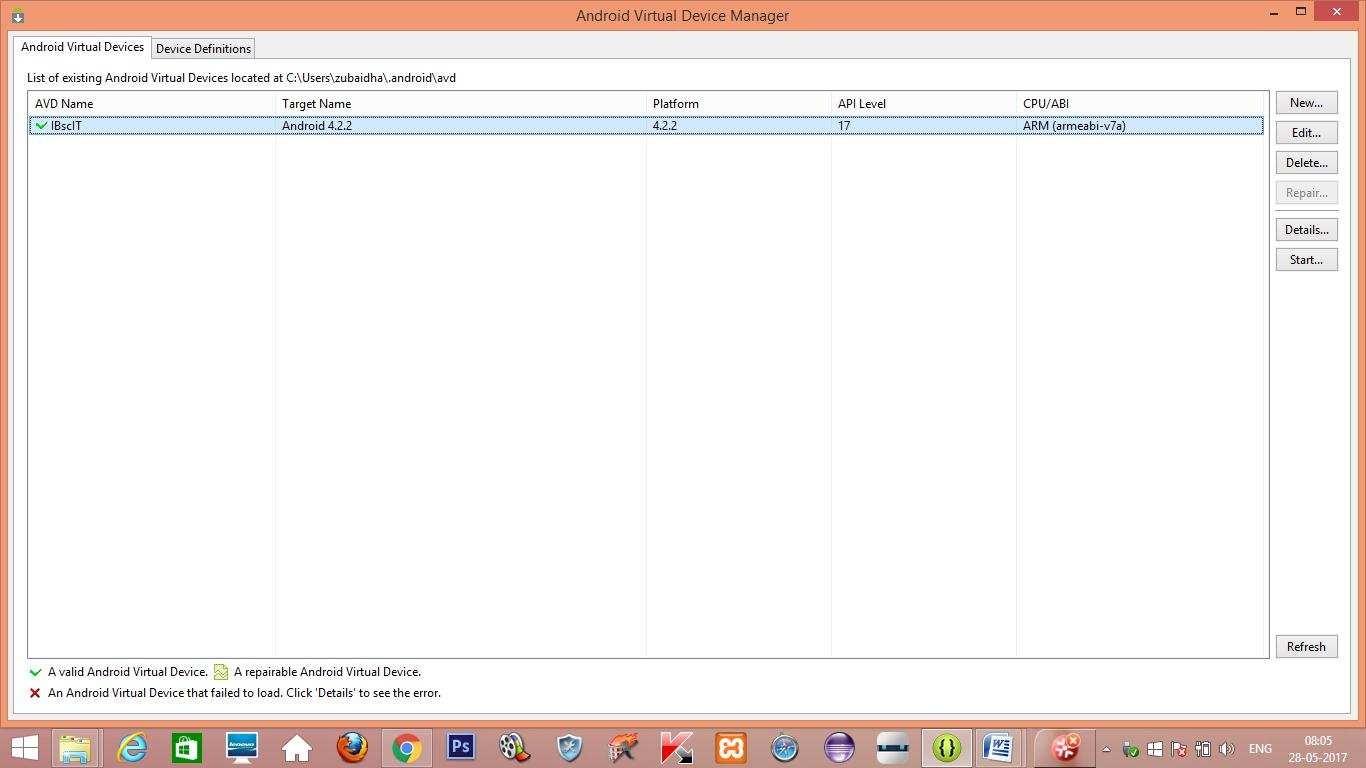
**Step 11:** On Completion of the **Project,** select **File ->Save All.**

**Step 12:** Select **Debug Icon** on the **Toolbar,** to debug the application.

**Step 13:** To run the application Select, **Run Icon** on the **Toolbar.**

**Step 14:** On the Android **Virtual Device Manager** window, select **AVD Name and**, Click

##### Start.



**Step 15:** On the Launch **Options** window, Check Wipe **User Data and**, Click **Launch.**

#### Program:

**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:paddingBottom="@dimen/activity\_vertical\_margin" android:paddingLeft="@dimen/activity\_horizontal\_margin" android:paddingRight="@dimen/activity\_horizontal\_margin" android:paddingTop="@dimen/activity\_vertical\_margin" tools:context=".MainActivity" >

<TextView

android:id="@+id/textview" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignParentTop="true" android:layout\_centerHorizontal="true" android:text="Screen Color Change" android:textSize="35dp" />

<TextView android:id="@+id/textView"

android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_below="@+id/textview" android:layout\_centerHorizontal="true" android:text="II B.sc IT" android:textColor="#ff7aff24" android:textSize="35dp" />

<Button

android:id="@+id/button2" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignLeft="@+id/button" android:layout\_below="@+id/button" android:layout\_marginTop="52dp" android:text="Blue" />

<Button

android:id="@+id/button" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignRight="@+id/textView" android:layout\_centerVertical="true"

android:layout\_marginRight="23dp" android:text="Red" />

<ImageView android:id="@+id/imageView" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_below="@+id/textView" android:layout\_centerHorizontal="true" android:src="@drawable/ic\_launcher" />

</RelativeLayout>

#### MainActivity.java

package com.example.colorchange; import android.app.Activity; import android.os.Bundle;

public class MainActivity extends Activity { @Override

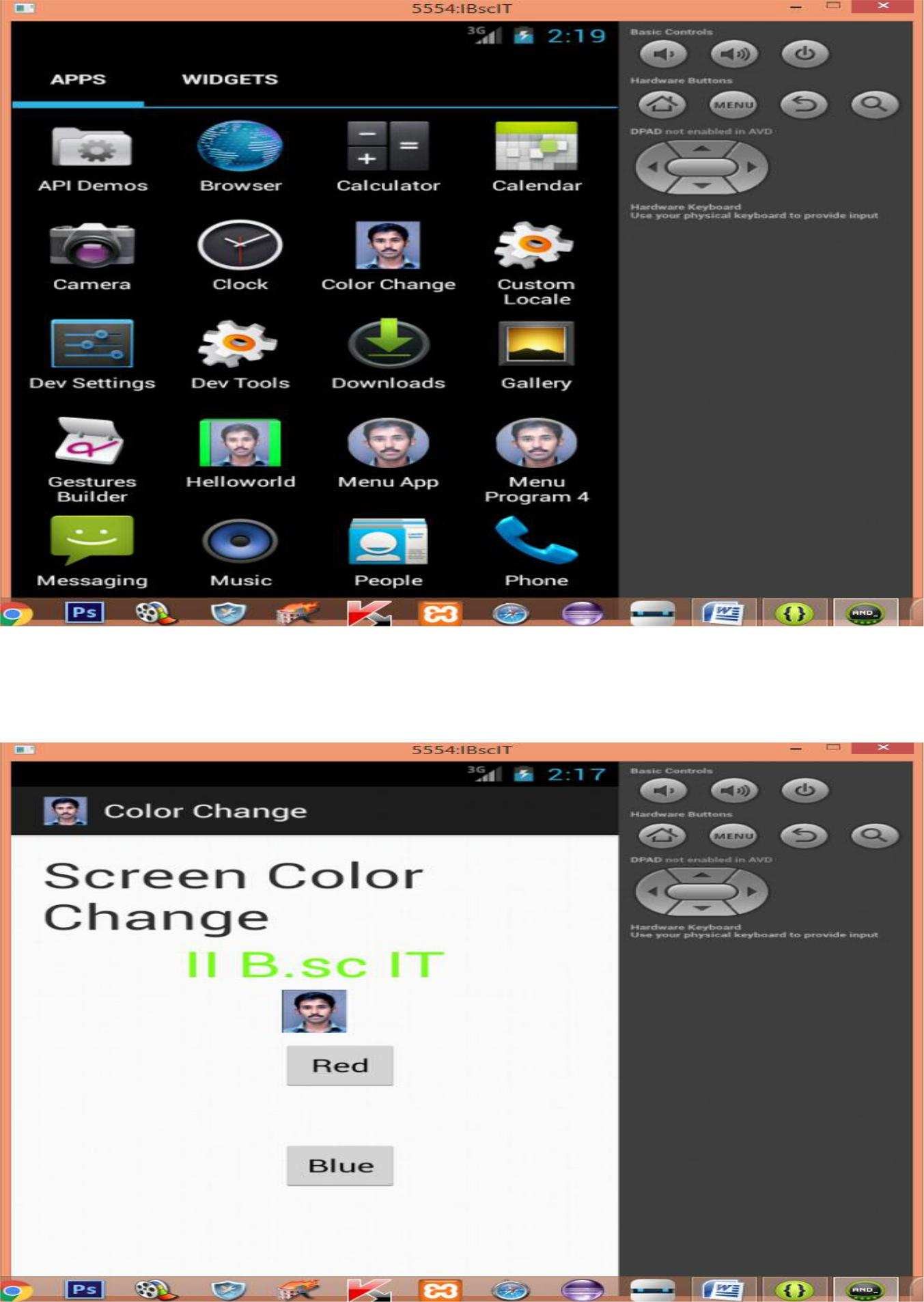
protected void onCreate(Bundle savedInstanceState)

{ super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main);

}

}

#### Output



##### Result:

The above aim of the program has been achieved successfully.

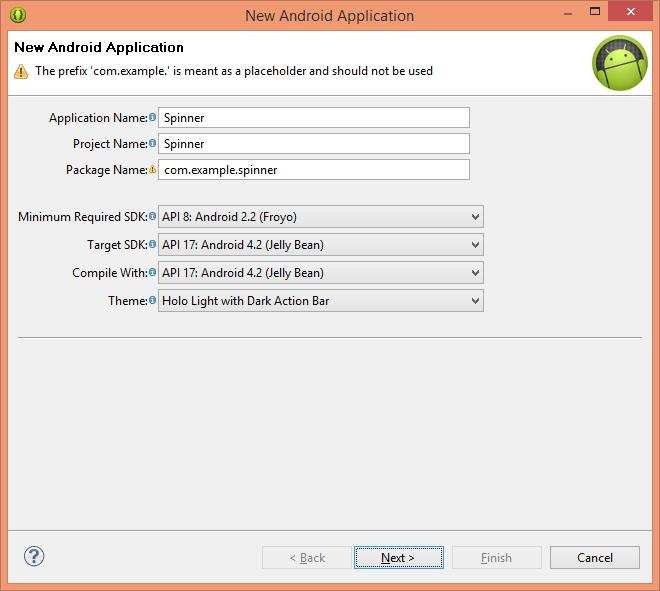
#### COMBO BOX,SPINNER ACTIVITY

##### Aim:

To create an android application to perform activities using combo box,spinner and Toast Message.

##### Procedure:

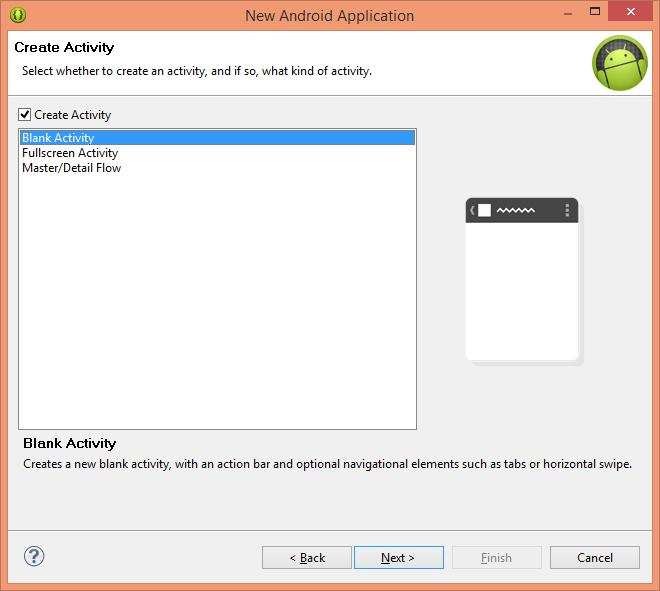
**Step 1:** Open Eclipse IDE and go to **File -> New** -> **Project** -> **Android** -> **Android Application Project**. You have to specify the **Application Name**, the **Project Name** and the **Package name** in the appropriate text fields and then click **Next**.



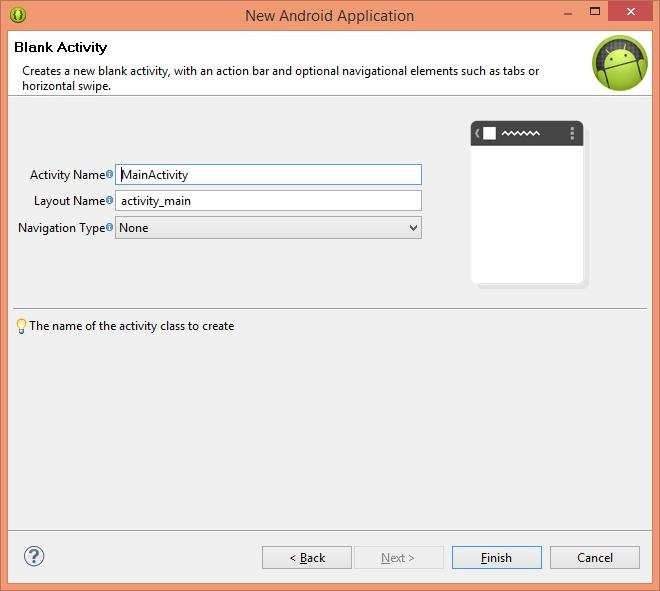
**Step 2:** In the next window make sure the select the **Application Launcher Icon**

option is selected in order to create a new activity for your project, and click **Next**.

**Step 3:** On the next window click to select the **Blank Activity** and click **Next**.



**Step 4:** On the next window, name the specify **Activity Name, Layout Name and Navigation Type,** click **Next**.



**Step 5:** Drag and Drop All the needed Components from the Palette Window to the **Design View of** the **Activity main.xml.**

**Step 6:** Create a new activity named **Success.xml,** and add some content.

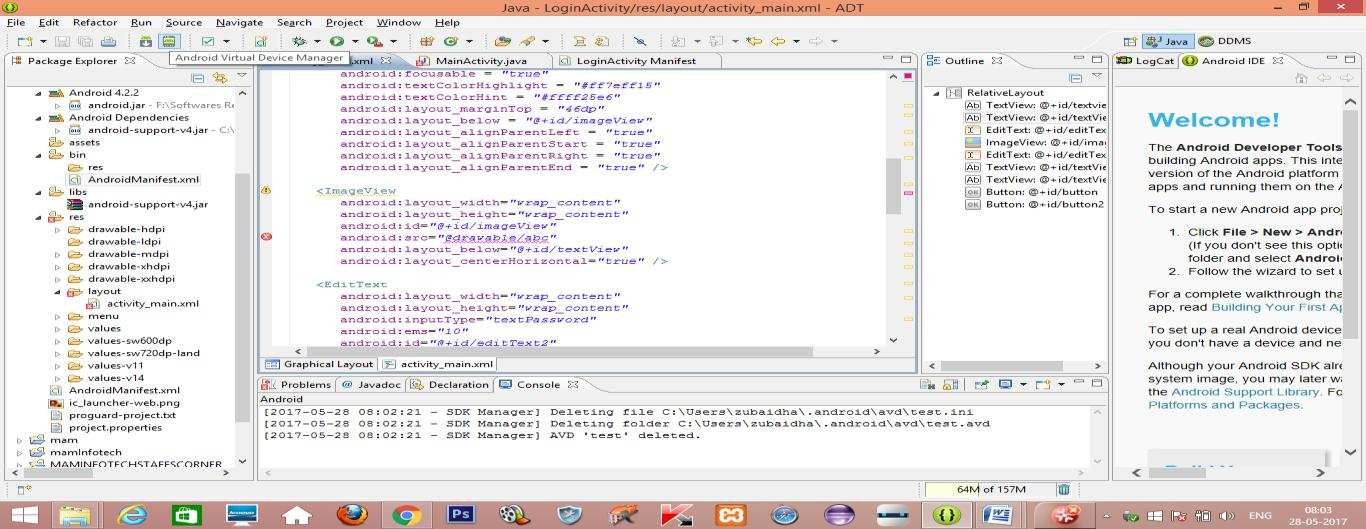
**Step 7:** Code the operations that are want to perform in the **MainActivity.java, of**

the preferred **.Xml** files.

**Step 8:** On the **Mainfest.xml,** file the permission for accessing phone dialer has to obtained.

**Step 9:** Create a new activity named **Success.xml,** and add some content for mobility.

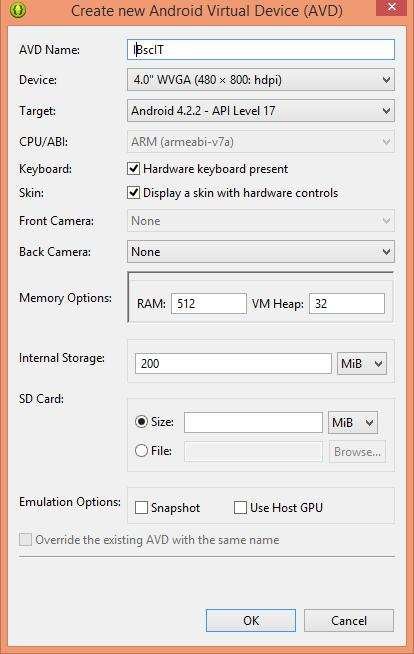
**Step 10:** Create an **AVD, by** Clicking the **Android Virtual Device Icon, on** the **Toolbar.**



**Step 11:** On the window, select **New.**

**Step 12:** On the next window, specify **AVD Name, Device, Target** and **Other Details**

and Click **OK.**



**Step 13:** Now a new **AVD** is created**.**

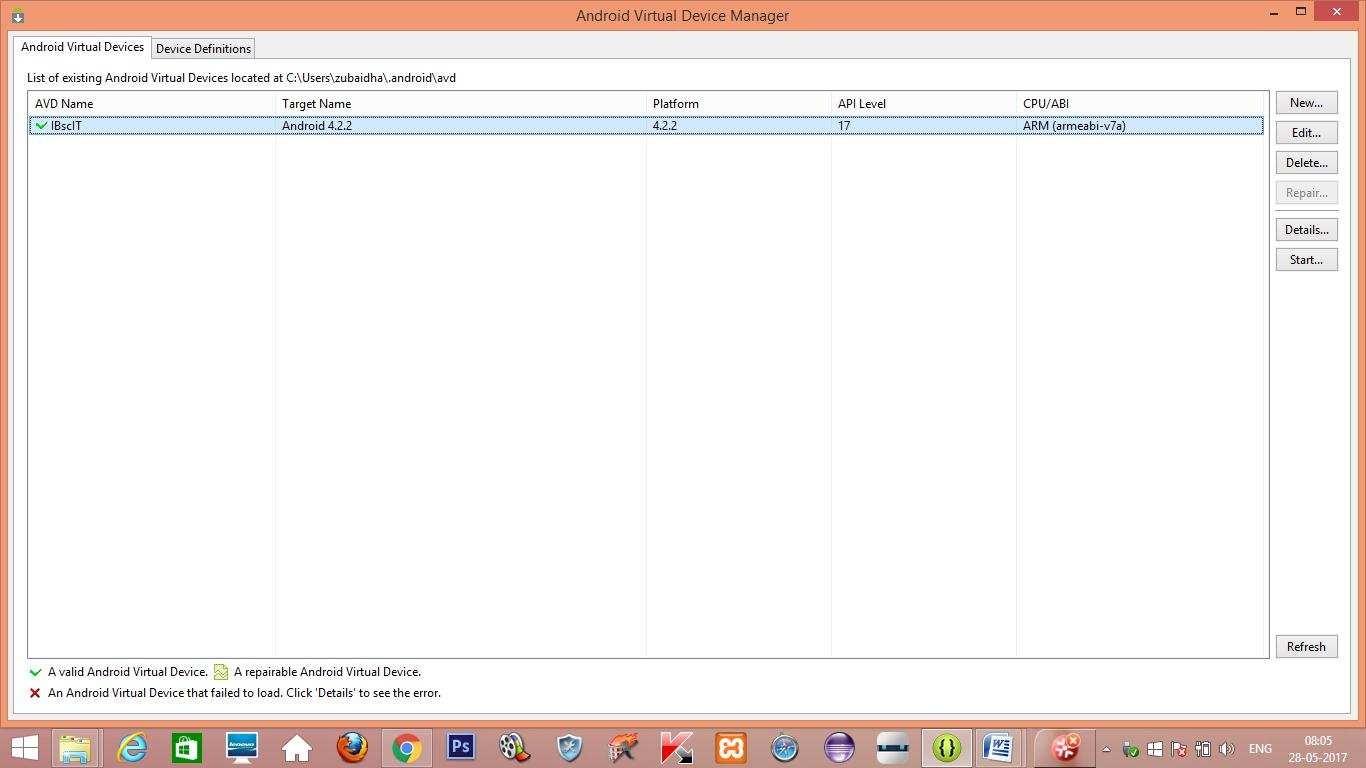
**Step 14:** On Completion of the **Project,** select **File ->Save All.**

**Step 15:** Select **Debug Icon** on the **Toolbar,** to debug the application.

**Step 16:** To run the application Select, **Run Icon** on the **Toolbar.**

**Step 17:** On the Android **Virtual Device Manager** window, select **AVD Name and**, Click

##### Start.



**Step 18:** On the Launch **Options** window, Check Wipe **User Data and**, Click **Launch.**

#### Program 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:paddingBottom="@dimen/activity\_vertical\_margin" android:paddingLeft="@dimen/activity\_horizontal\_margin" android:paddingRight="@dimen/activity\_horizontal\_margin" android:paddingTop="@dimen/activity\_vertical\_margin"

tools:context=".MainActivity" >

<TextView

android:id="@+id/textview" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignParentTop="true" android:layout\_centerHorizontal="true" android:text="Spinner With Toast" android:textSize="35dp" />

<TextView android:id="@+id/textView" android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" android:layout\_below="@+id/textview" android:layout\_centerHorizontal="true" android:text="II B.sc IT" android:textColor="#ff7aff24" android:textSize="35dp" />

<ImageView android:id="@+id/imageView" android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" android:layout\_below="@+id/textView" android:layout\_centerHorizontal="true" android:src="@drawable/ic\_launcher" />

<Spinner android:id="@+id/spinner1"

android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignParentTop="true" android:layout\_centerHorizontal="true" android:layout\_marginTop="175dp" />

</RelativeLayout>

#### MainActivity.java

package com.example.spinner; import android.app.Activity; import android.os.Bundle; import android.view.Menu; import android.view.View;

import android.widget.AdapterView; import android.widget.ArrayAdapter; import android.widget.Spinner;

import android.widget.TextView; import android.widget.Toast;

public class MainActivity extends Activity implements AdapterView.OnItemSelectedListener {

String[] country = { "RCAS", "RTC", "KPM", "RIPS", "RIM", };

@Override

protected void onCreate(Bundle savedInstanceState)

{ super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main);

//Getting the instance of Spinner and applying OnItemSelectedListener on it Spinner spin = (Spinner) findViewById(R.id.spinner1); spin.setOnItemSelectedListener(this);

//Creating the ArrayAdapter instance having the country list

ArrayAdapter aa = new ArrayAdapter(this,android.R.layout.simple\_spinner\_item,country);

aa.setDropDownViewResource(android.R.layout.simple\_spinner\_dropdown\_item);

//Setting the ArrayAdapter data on the Spinner spin.setAdapter(aa);

}

//Performing action onItemSelected and onNothing selected @Override

public void onItemSelected(AdapterView<?> arg0, View arg1, int position,long id) {

Toast.makeText(getApplicationContext(),country[position] , Toast.LENGTH\_LONG).show();

}

@Override

public void onNothingSelected(AdapterView<?> arg0)

{ // TODO Auto-generated method stub

}

@Override

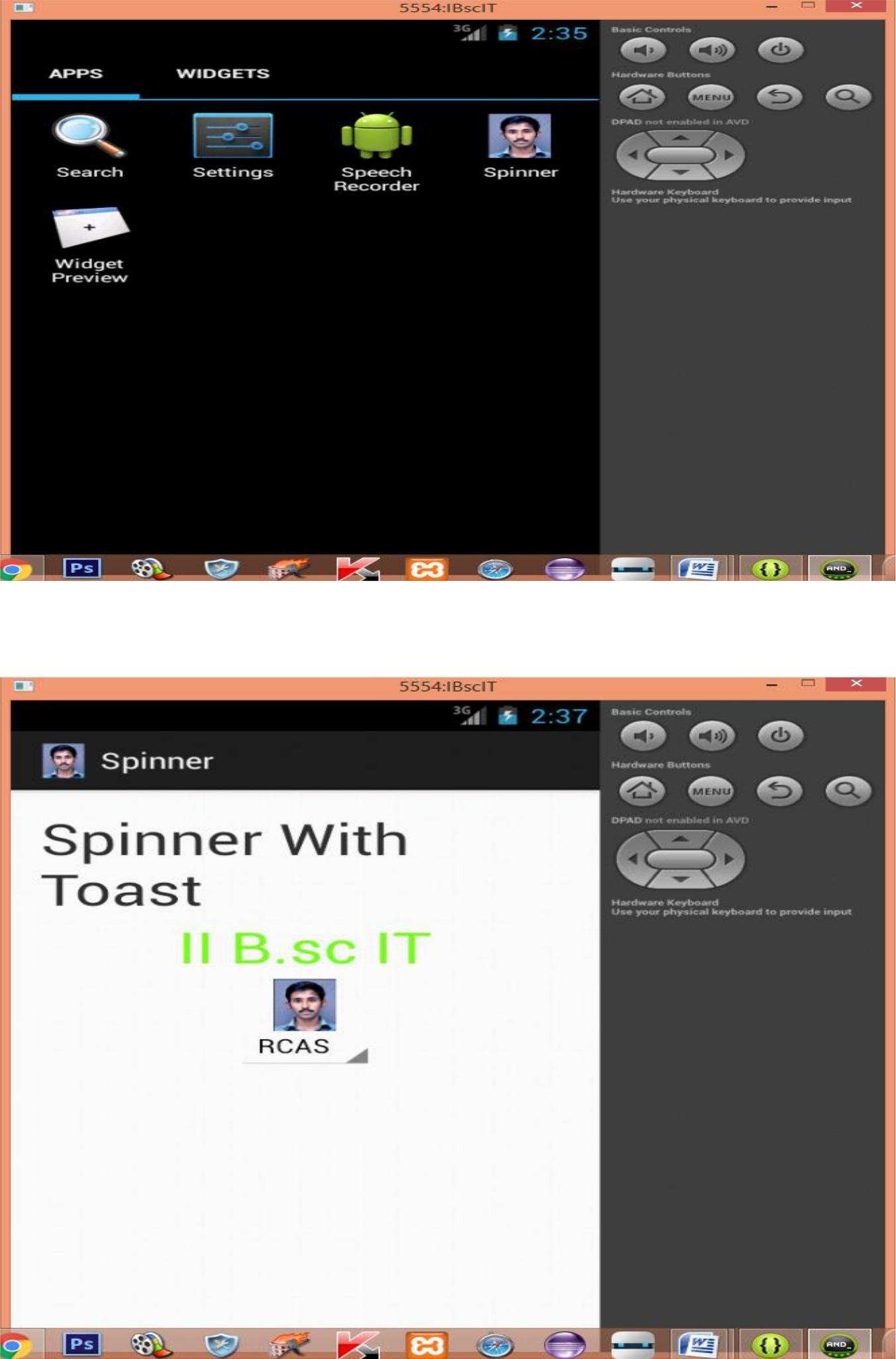
public boolean onCreateOptionsMenu(Menu menu) {

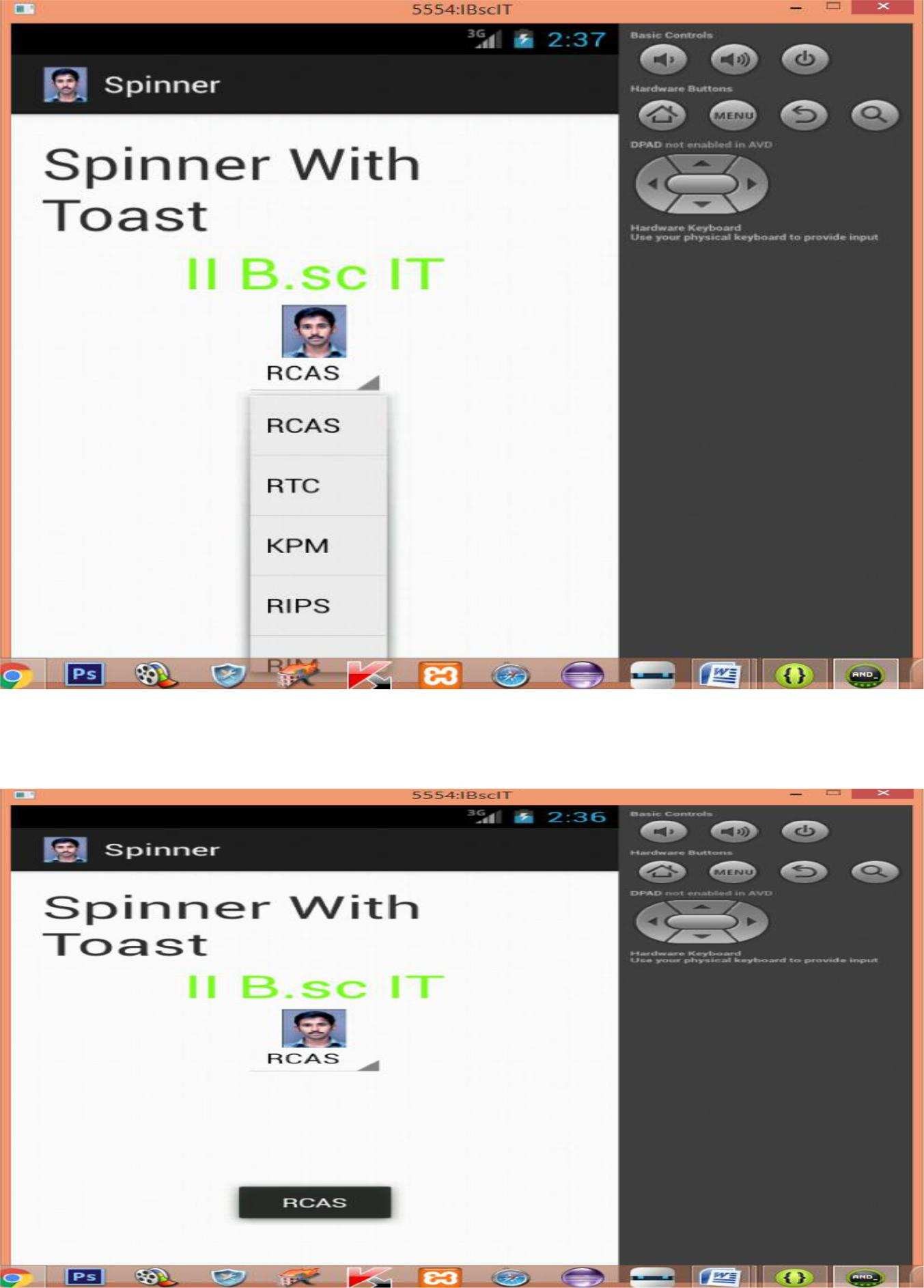
// Inflate the menu; this adds items to the action bar if it is present. //getMenuInflater().inflate(R.menu.activity\_main, menu); return true;

}

}

##### Output:





**Result:**

The above aim of the program has been achieved successfully.

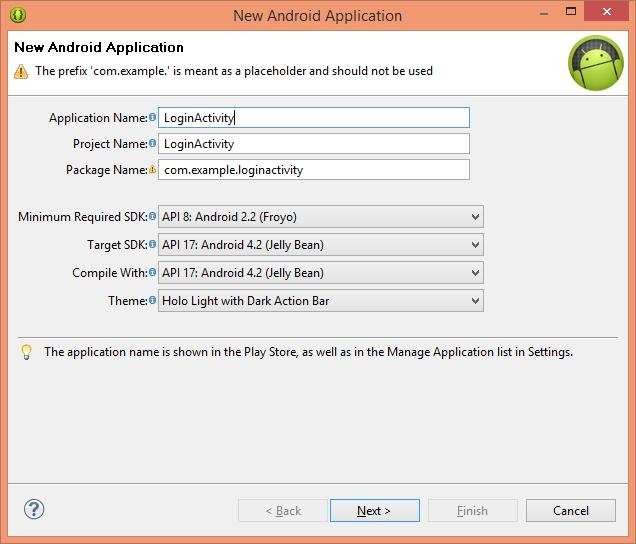
#### PHONE CALL ACTIVITY

##### Aim:

**To create an android application to perform phone call to the user entered number.**

##### Procedure:

**Step 1: Open Eclipse IDE and go to File -> New -> Project -> Android -> Android Application Project. You have to specify the Application Name, the Project Name and the Package name in the appropriate text fields and then click Next.**

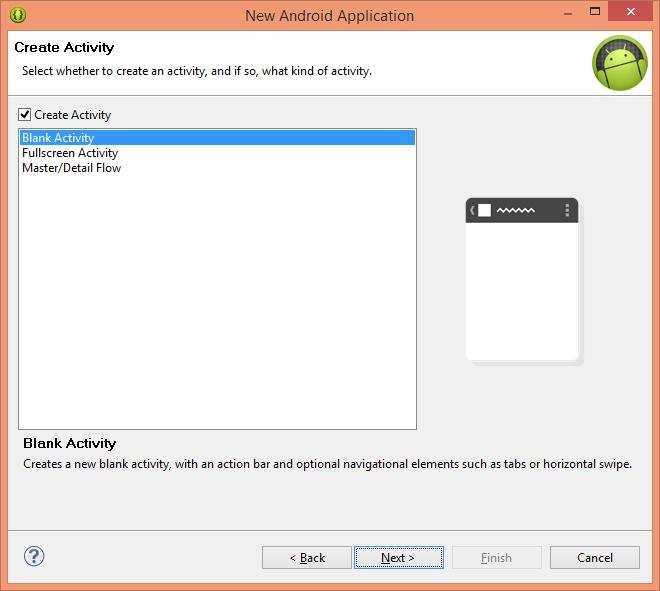


**Step 2:** In the next window make sure the select the **Application Launcher Icon**

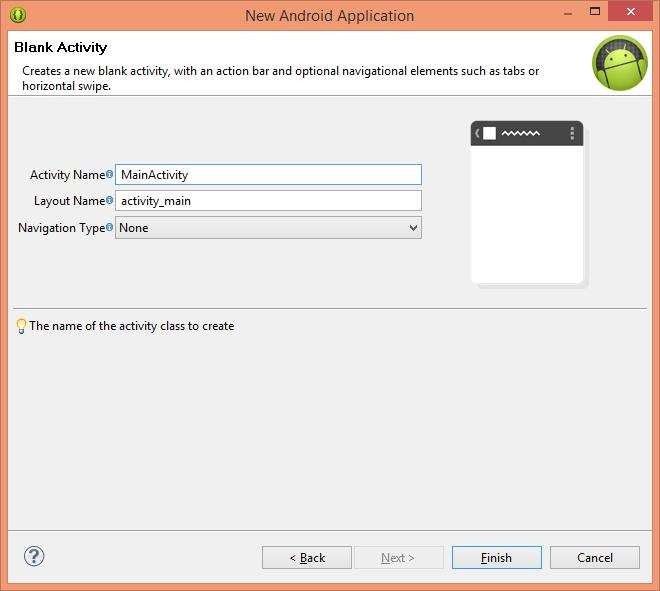
option is selected in order to create a new activity for your project, and click **Next**.



**Step 3:** On the next window click to select the **Blank Activity** and click **Next**.



**Step 4:** On the next window, name the specify **Activity Name, Layout Name and Navigation Type,** click **Next**.



**Step 5:** Drag and Drop All the needed Components from the Palette Window to the **Design View of** the **Activity main.xml.**

**Step 6:** Create a new activity named **Success.xml,** and add some content.

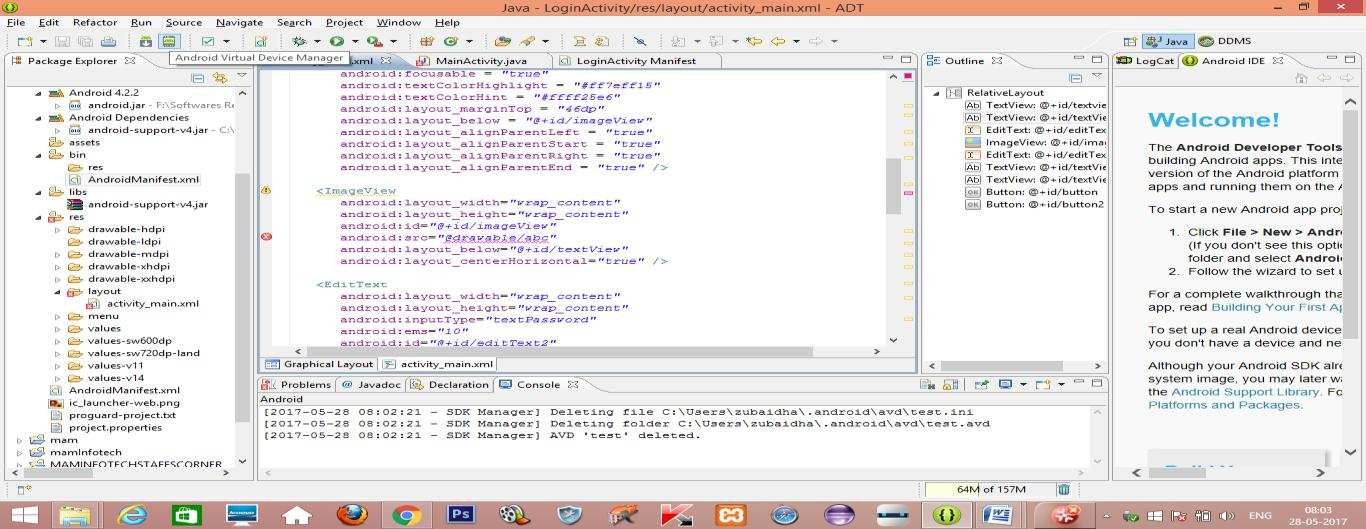
**Step 7:** Code the operations that are want to perform in the **MainActivity.java, of**

the preferred **.Xml** files.

**Step 8:** On the **Mainfest.xml,** file the permission for accessing phone dialer has to obtained.

**Step 9:** Create a new activity named **Success.xml,** and add some content for mobility.

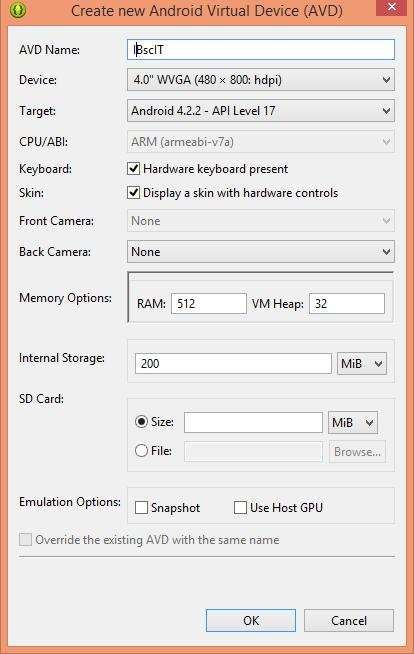
**Step 10:** Create an **AVD, by** Clicking the **Android Virtual Device Icon, on** the **Toolbar.**



**Step 11:** On the window, select **New.**

**Step 12:** On the next window, specify **AVD Name, Device, Target** and **Other Details**

and Click **OK.**



**Step 13:** Now a new **AVD** is created**.**

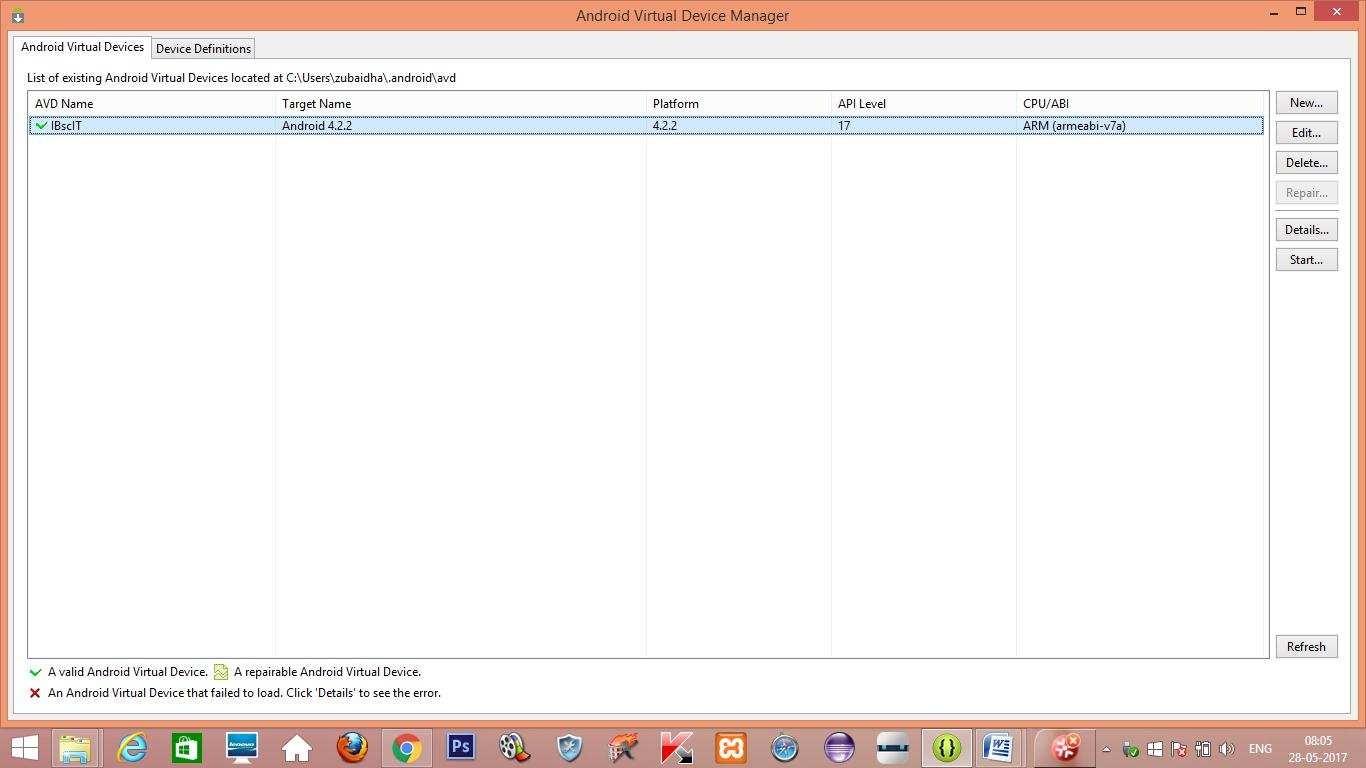
**Step 14:** On Completion of the **Project,** select **File ->Save All.**

**Step 15:** Select **Debug Icon** on the **Toolbar,** to debug the application.

**Step 16:** To run the application Select, **Run Icon** on the **Toolbar.**

**Step 17:** On the Android **Virtual Device Manager** window, select **AVD Name and**, Click

##### Start.



**Step 18:** On the Launch **Options** window, Check Wipe **User Data and**, Click **Launch.**

##### Program:

**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:paddingBottom="@dimen/activity\_vertical\_margin" android:paddingLeft="@dimen/activity\_horizontal\_margin" android:paddingRight="@dimen/activity\_horizontal\_margin" android:paddingTop="@dimen/activity\_vertical\_margin" tools:context=".MainActivity" >

<TextView

android:id="@+id/textview" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignParentTop="true" android:layout\_centerHorizontal="true" android:text="Phone Dialer" android:textSize="35dp" />

<TextView android:id="@+id/textView" android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" android:layout\_below="@+id/textview" android:layout\_centerHorizontal="true" android:text="II B.sc IT" android:textColor="#ff7aff24" android:textSize="35dp" />

<ImageView android:id="@+id/imageView" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_below="@+id/textView" android:layout\_centerHorizontal="true" android:src="@drawable/ic\_launcher" />

<Button

android:id="@+id/button1" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignLeft="@+id/imageView" android:layout\_alignParentBottom="true" android:layout\_marginBottom="104dp" android:text="Call" />

<EditText android:id="@+id/editText1"

android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"

android:layout\_alignRight="@+id/textview" android:layout\_centerVertical="true" android:ems="10"

android:hint="Please Enter 10 Numbers" android:inputType="number" >

<requestFocus />

</EditText>

</RelativeLayout>

##### MainActivity.java

package com.example.phonecaller;

import android.net.Uri; import android.os.Bundle; import android.app.Activity; import android.content.Intent; import android.view.Menu; import android.view.View;

import android.view.View.OnClickListener; import android.widget.Button;

import android.widget.EditText;

public class MainActivity extends Activity { EditText edittext1;

Button button1; @Override

protected void onCreate(Bundle savedInstanceState)

{ super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main);

//Getting the edittext and button instance edittext1=(EditText)findViewById(R.id.editText1); button1=(Button)findViewById(R.id.button1);

//Performing action on button click button1.setOnClickListener(new OnClickListener(){

@Override

public void onClick(View arg0) {

String number=edittext1.getText().toString();

Intent callIntent = new Intent(Intent.ACTION\_CALL); callIntent.setData(Uri.parse("tel:"+number)); startActivity(callIntent);

}

});

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present.

// getMenuInflater().inflate(R.menu.activity\_main, menu); return true;

}

##### Manifest.xml

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

<manifest xmlns:android="<http://schemas.android.com/apk/res/android>" package="com.example.phonecaller"

android:versionCode="1" android:versionName="1.0" >

<uses-sdk android:minSdkVersion="8" android:targetSdkVersion="17" />

<uses-permission android:name="android.permission.CALL\_PHONE" />

<application android:allowBackup="true"

android:icon="@drawable/ic\_launcher" android:label="@string/app\_name" android:theme="@style/AppTheme" >

<activity android:name="com.example.phonecaller.MainActivity" android:label="@string/app\_name" >

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER"

/> </intent-filter>

</activity>

</application>

</manifest>

##### Strings.xml

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

<resources>

<string name="app\_name">PhoneCaller</string>

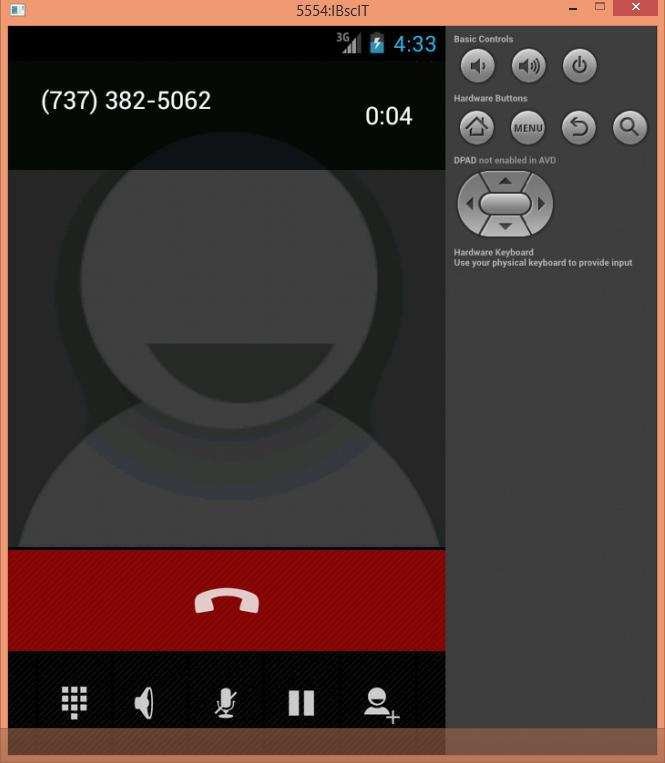
<string name="action\_settings">Settings</string>

<string name="hello\_world">Hello world!</string>

</resources

##### Output:





**Result:**

The above aim of the program has been achieved successfully

***Android Application Development 70***

#### DEVELOP APPLICATION THAT USES GPS LOCATION INFORMATION.

##### Aim:

**To create an android application to develop an application that uses GPS**

##### location information. Procedure:

**Step 1: Open eclipse or android studio and select new android project Step 2: Give project name and select next**

##### Step 3: Choose the android version. Choose the lowest android version (Android 2.2) and select next

**Step 4: Enter the package name. Package name must be two word separated by comma and click finish**

##### Step 5: Go to package explorer in the left hand side. Select our project.

**Step 6: Go to res folder and select layout. Double click the main.xml file. Add the code**

Source code:

<RelativeLayout xmlns:android="<http://schemas.android.com/apk/res/android>" android:id="@+id/relativeLayout1" android:layout\_width="match\_parent" android:layout\_height="match\_parent" >

<Button

android:id="@+id/show\_Location" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content android:text="Show\_Location" android:layout\_centerVertical="true" android:layout\_centerHorizontal="true"

/>

</RelativeLayout>

package gps.location;

//import android.R;

import android.app.Activity; import android.os.Bundle; import android.view.View; import android.widget.Button; import android.widget.Toast;

public class GPSlocationActivity extends Activity {

Button btnShowLocation; GPStrace gps;

@Override

public void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.main); btnShowLocation=(Button)findViewById(R.id.show\_Location); btnShowLocation.setOnClickListener(new View.OnClickListener() {

###### Android Application Development 71

@Override

public void onClick(View v) {

// TODO Auto-generated method stub gps=new GPStrace(GPSlocationActivity.this); if(gps.canGetLocation()){

double latitude=gps.getLatitude(); double longitude=gps.getLongtiude();

Toast.makeText(getApplicationContext(),"Your Location is

\nLat:"+latitude+"\nLong:"+longitude, Toast.LENGTH\_LONG).show();

}

else

{

gps.showSettingAlert();

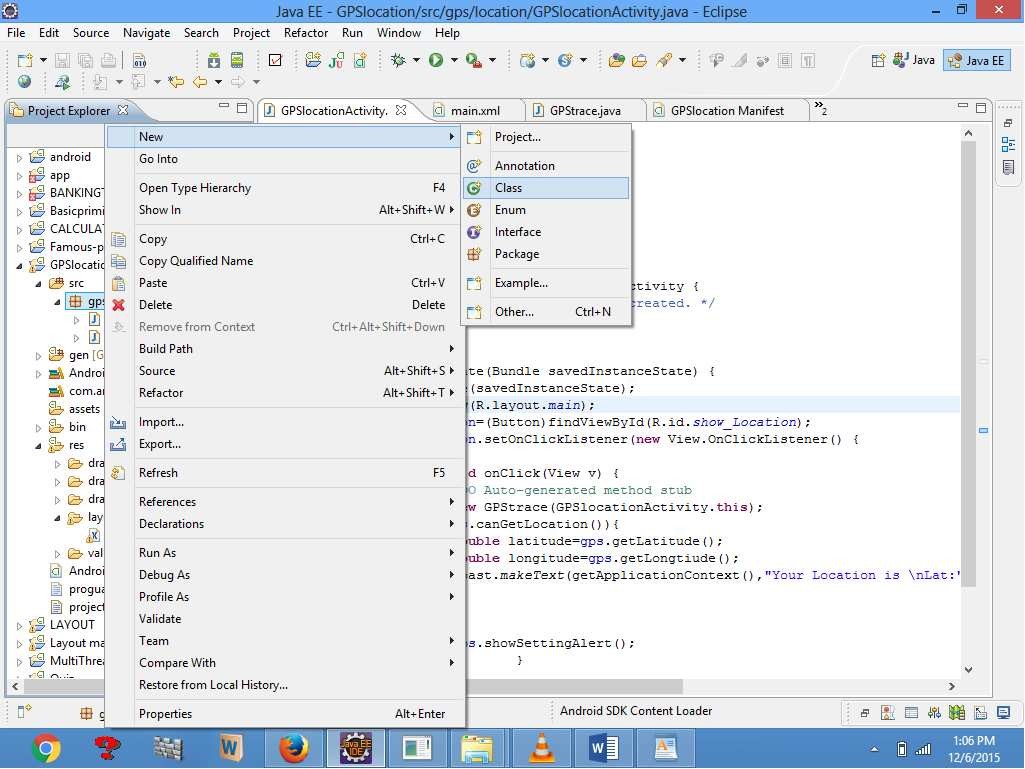
}

} }); } }

### )

##### Go to src folder and Right Click on your package folder and choose new class and give the class nams as GPStrace

Rathinam College of Arts & Science(Autonomous) 75



##### Step 9: Select the GPStrace.java file and paste the following code.

package gps.location;

import android.app.AlertDialog; import android.app.Service; import android.content.Context;

import android.content.DialogInterface;

###### Android Application Development 72

import android.content.Intent;

import android.location.Location;

import android.location.LocationListener; import android.location.LocationManager; import android.os.Bundle;

import android.os.IBinder; import android.provider.Settings;

public class GPStrace extends Service implements LocationListener{ private final Context context;

boolean isGPSEnabled=false; boolean canGetLocation=false; boolean isNetworkEnabled=false; Location location;

Rathinam College of Arts & Science(Autonomous) 76

double latitude; double longtitude;

private static final long MIN\_DISTANCE\_CHANGE\_FOR\_UPDATES=10; private static final long MIN\_TIME\_BW\_UPDATES=1000\*60\*1;

protected LocationManager locationManager; public GPStrace(Context context)

{

this.context=context; getLocation();

}

public Location getLocation()

{

try{

locationManager=(LocationManager) context.getSystemService(LOCATION\_SERVICE); isGPSEnabled=locationManager.isProviderEnabled(LocationManager.GPS\_PROVIDER); isNetworkEnabled=locationManager.isProviderEnabled(LocationManager.NETWORK\_P ROVIDER);

if(!isGPSEnabled && !isNetworkEnabled){

}else{ this.canGetLocation=true; if(isNetworkEnabled){

locationManager.requestLocationUpdates( LocationManager.NETWORK\_PROVIDER, MIN\_TIME\_BW\_UPDATES, MIN\_DISTANCE\_CHANGE\_FOR\_UPDATES,this);

}

if(locationManager!=null){ location=locationManager.getLastKnownLocation(LocationManager.NETWORK\_PROVI DER)

;

if(location !=null){ latitude=location.getLatitude(); longtitude=location.getLongitude();

}

}

}

if(isGPSEnabled){ if(location==null){

locationManager.requestLocationUpdates(LocationManager.GPS\_PROVIDER,MIN\_TIM E\_B

W\_UPDATES, MIN\_DISTANCE\_CHANGE\_FOR\_UPDATES, this);

if(locationManager!=null){ location=locationManager.getLastKnownLocation(LocationManager.GPS\_PROVIDER);

[Android Application Development 73](#_TOC_250001)

if(location!=null){

latitude=location.getLatitude(); longtitude=location.getLongitude();

[Rathinam College of Arts & Science(Autonomous) 77](#_TOC_250000)

}

}

}

}

}

catch(Exception e)

{

e.printStackTrace();

}

return location;

}

public void stopUsingGPS(){ if(locationManager!=null){ locationManager.removeUpdates(GPStrace.this);

}

}

public double getLatitude(){ if(location!=null){ latitude=location.getLatitude();

}

return latitude;

}

public double getLongtiude(){ if(location!=null){ longtitude=location.getLatitude();

}

return longtitude;

}

public boolean canGetLocation(){ return this.canGetLocation;

}

public void showSettingAlert(){

AlertDialog.Builder alertDialog=new AlertDialog.Builder(context); alertDialog.setTitle("GPS is settings");

alertDialog.setMessage("GPS is not enabled.Do you want to go to setting menu?"); alertDialog.setPositiveButton("settings", new DialogInterface.OnClickListener() { @Override

public void onClick(DialogInterface dialog,int which){

Intent intent=new Intent(Settings.ACTION\_LOCATION\_SOURCE\_SETTINGS); context.startActivity(intent);

}

});

alertDialog.setNegativeButton("cancel", new DialogInterface.OnClickListener() { @Override

public void onClick(DialogInterface dialog, int which) {

// TODO Auto-generated method stub dialog.cancel();

}

Rathinam College of Arts & Science(Autonomous) 78

});

alertDialog.show();

}

@Override

public void onLocationChanged(Location location) {

// TODO Auto-generated method stub

}

@Override

public void onProviderDisabled(String provider) {

// TODO Auto-generated method stub

}

Rathinam College of Arts & Science(Autonomous) 79

@Override

public void onProviderEnabled(String provider) {

// TODO Auto-generated method stub

}

@Override

public void onStatusChanged(String provider, int status, Bundle extras) {

// TODO Auto-generated method stub

}

@Override

public IBinder onBind(Intent intent) {

// TODO Auto-generated method stub return null;

}

}

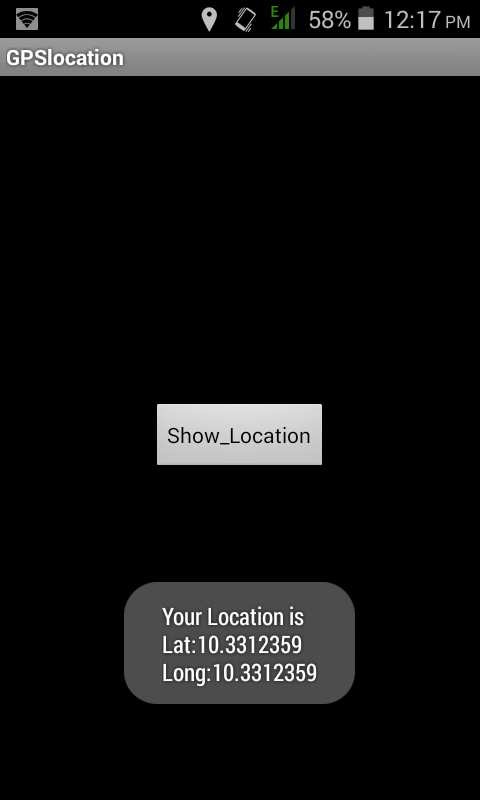
##### Step 10: Go to manifest.xml file and add the code below

<uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION"/>

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

##### Now go to main.xml and right click .select run as option and select run configuration.

* **Android output is present in the android emulator as shown in below**.



##### Result:

The above aim of the program has been achieved successfully.

Rathinam College of Arts & Science(Autonomous) 80