Introduction to android

Android is a Linux based operating system it is designed primarily for touch screen mobile devices such as smart phones and tablet computers. The android is a powerful operating system and it supports large number of applications in Smartphones. These applications are more comfortable and advanced for the users. The hardware that supports android software is based on ARM architecture platform. The android is an open source operating system. The android has got millions of apps available that can help you managing your life one or other way and it is available low cost in market at that reasons android is very popular. The android development supports with the full java programming language. Even other packages that are API and JSE are not supported. The first version 1.0 of android development kit (SDK) was released in 2008

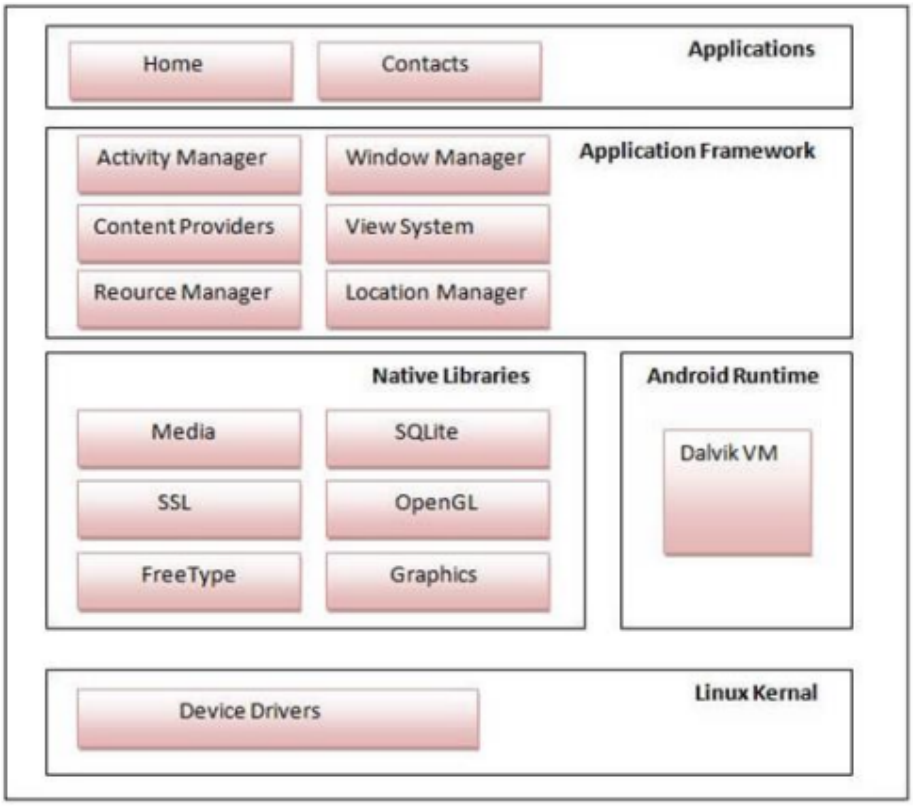
ANDROID ARCHITECTURE

The android is an operating system and is a stack of software components which is divided into five sections and four main layers that is

· Linux kernel
· Libraries
· Android runtime

Linux kernel

The android uses the powerful Linux kernel and it supports wide range of hardware drivers. The kernel is the heart of the operating system that manages input and output requests from software. This provides basic system functionalities like process management, memory management; device management like camera, keypad, and display etc., the kernel handles all the things. The Linux is really good at networking and it is not necessary to interface it to the peripheral hardware. The kernel itself does not interact directly with the user but rather interacts with the shell and other programs as well as with the hard ware devices on the system.



Libraries

On top of a Linux kennel there is a set of libraries including open source web browser such as webkit, library libc. These libraries are used to play and record audio and video. The SQLite is a data base which is useful for storage and sharing of application data. The SSL libraries are responsible for internet security etc.

Android runtime

The android runtime provides a key component called Dalvik Virtual Machine which is a kind of java virtual machine. It is specially designed and optimized for android. The Dalvik VM is the process virtual machine in the android operating system. It is software that runs apps on android devices.

The Dalvik VM makes use of Linux core features like memory management and multithreading which is in a java language. The Dalvik VM enables every android application to run its own process. The Dalvik VM executes the files in the .dex format.

Application frame work

The application frame work layer provides many higher level services to applications such as windows manager, view system, package manager, resource manager etc. The application developers are allowed to make use of these services in their application.

Program 1

Design a Login Form with username and password using LinearLayout and toast valid credentials.

Steps

1. Select **activity_login.xml**.

Source code

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android: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/tv_login"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_marginLeft="11dp"
        android:layout_marginStart="11dp"
        android:layout_marginTop="13dp"
        android:text="@string/Login"
        android:fontFamily="sans-serif-condensed"
        android:textSize="30sp"/>
    <TextView
        android:id="@+id/tv_username"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/tv_login"
        android:layout_alignStart="@+id/tv_login"
        android:layout_below="@+id/tv_login"
        android:layout_marginTop="80dp"
        android:fontFamily="monospace"
        android:text="@string/Username"
        android:textSize="25sp" />
    <EditText
        android:id="@+id/et_username"
        android:inputType="text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/tv_username"
        android:ems="17"
        android:layout_alignLeft="@+id/tv_username" />
    <TextView
        android:id="@+id/tv_password"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/et_username"
        android:layout_alignStart="@+id/et_username"
```

```
        android:layout_below="@+id/et_username"
        android:layout_marginTop="33dp"
        android:fontFamily="monospace"
        android:text="@string/Password"
        android:textSize="25sp" />

    <EditText
        android:id="@+id/et_password"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="textPassword"
        android:ems="17"
        android:layout_below="@+id/tv_password"
        android:layout_alignLeft="@+id/tv_password" />


    <Button
        android:id="@+id/btn_login"
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:layout_below="@id/et_password"
        android:layout_centerInParent="true"
        android:ems="12"
        android:layout_marginTop="30dp"
        android:text="@string/Login"/>

</LinearLayout>
```

Next is create another layout by right-clicking the layout folder namely **activity_user.xml**

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.razormist.simpleloginapplication.User">

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/Welcome"
        android:textSize="40sp"
        android:layout_marginTop="177dp"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
        android:textSize="30sp"
        android:layout_marginTop="30dp"
        android:layout_centerInParent="true"
        android:layout_below="@+id/textView1"
        android:text="@string/Administrator"/>

</RelativeLayout>
```

2.Android Manifest File-The Android Manifest file provides essential information about your app to the Android system in which the system must required before running the code.

Source code

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.login"
    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.login.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>
        <activity android:name="com.example.login.User"
            android:configChanges="orientation"
            android:screenOrientation="portrait">
            <intent-filter>
                <action android:name="android.intent.action.User" />
                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
    </application>


</manifest>
```

3. MainActivity.java

This code contains the main function of the application. This code will login the user when the username and password are entered correctly.

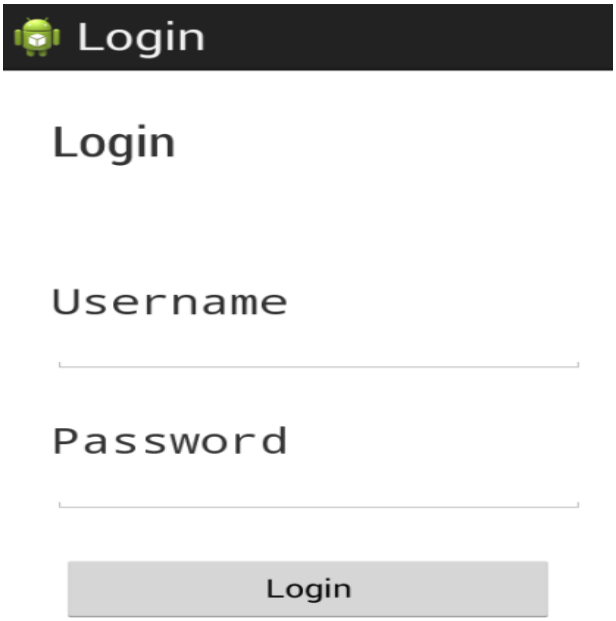
Source code

```
package com.example.login;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends Activity {
EditText et_username, et_password;
   Button btn_login;

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

      Login();
   }
   void Login(){
      et_username = (EditText)findViewById(R.id.et_username);
      et_password = (EditText)findViewById(R.id.et_password);
      btn_login = (Button)findViewById(R.id.btn_login);

      btn_login.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            if(et_username.getText().toString().equals("admin") &&
et_password.getText().toString().equals("admin")){
               Toast.makeText(MainActivity.this, "Username and Password is correct",
Toast.LENGTH_SHORT).show();
               Intent intent = new Intent(MainActivity.this,User.class);
               startActivity(intent);
            }else{
               Toast.makeText(MainActivity.this, "Username or Password is incorrect",
Toast.LENGTH_SHORT).show();
            }
         }
      });
   }
}
```

User.java-This code will render a new layout after the user successfully login. This is where the user is redirect after entering the correction information.

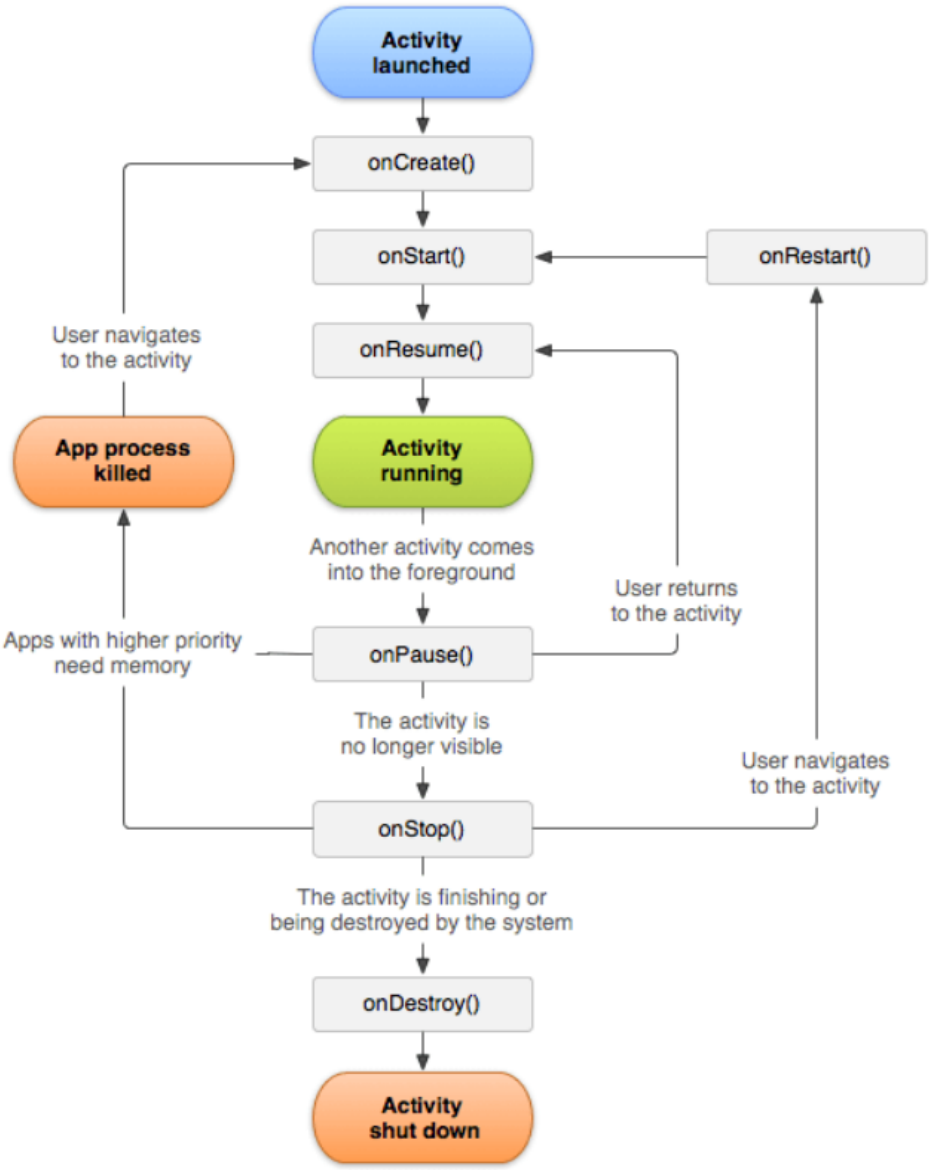
```
package com.example.login;
import android.app.Activity;
import android.os.Bundle;
public class User extends Activity  {
@Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_user);
   }
}
```

Sample Output



Program 2

Write a program that demonstrates Activity Lifecycle.



MainActivity.java

```
package com.example.hello;
import android.os.Bundle;
public class MainActivity extends Activity {
 @Override
 protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_main);
```

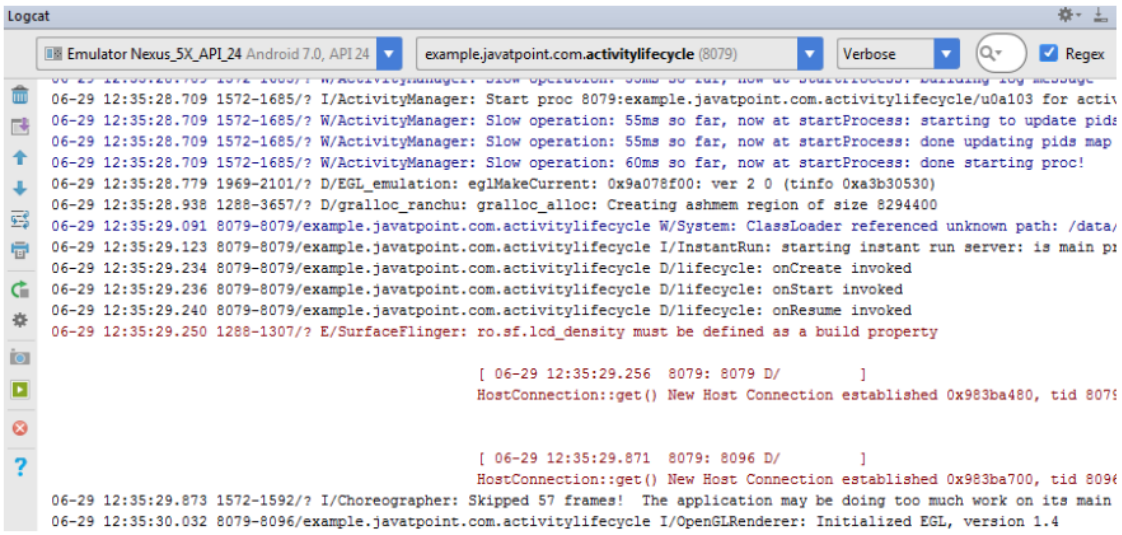
```
Log.d("lifecycle","onCreate invoked");
}
@Override
protected void onStart() {
super.onStart();
Log.d("lifecycle","onStart invoked");
}
@Override
protected void onResume() {
super.onResume();
Log.d("lifecycle","onResume invoked");
}
@Override
protected void onPause() {
super.onPause();
Log.d("lifecycle","onPause invoked");
}
@Override
protected void onStop() {
super.onStop();
Log.d("lifecycle","onStop invoked");
}
@Override
protected void onRestart() {
super.onRestart();
Log.d("lifecycle","onRestart invoked");
}
@Override
protected void onDestroy() {
super.onDestroy();
```

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

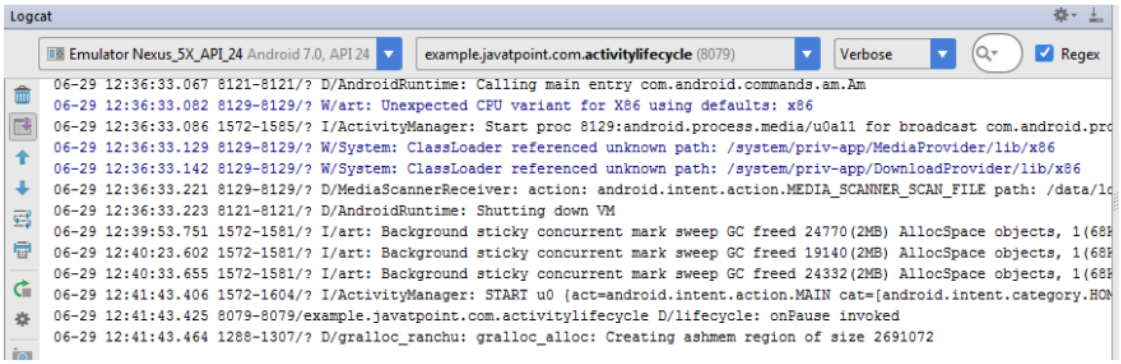
Output:

You will not see any output on the emulator or device. You need to open logcat.

Now see on the logcat: onCreate, onStart and onResume methods are invoked.

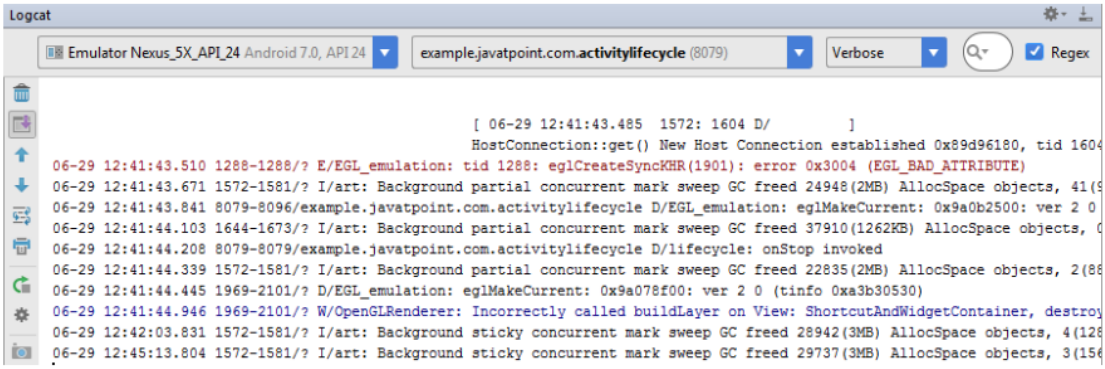


Now click on the HOME Button. You will see onPause method is invoked.



After a while, you will see onStop method is invoked.

After a while, you will see onStop method is invoked.



Program 3

Implementing basic arithmetic operations of a simple calculator

Activity main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/relative1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/edt1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/text" />

    <Button
        android:id="@+id/button1"
        style="?android:attr/buttonStyleSmall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignEnd="@+id/button4"
        android:layout_alignRight="@+id/button4"
        android:layout_below="@+id/edt1"
        android:layout_marginTop="94dp"
        android:text="@string/One" />

    <Button
        android:id="@+id/button2"
        style="?android:attr/buttonStyleSmall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignTop="@+id/button1"
        android:layout_toLeftOf="@+id/button3"
        android:layout_toStartOf="@+id/button3"
        android:text="@string/Two" />

    <Button
        android:id="@+id/button3"
        style="?android:attr/buttonStyleSmall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignTop="@+id/button2"
        android:layout_centerHorizontal="true"
```

```
    android:text="@string/three" />

<Button
    android:id="@+id/button4"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/button1"
    android:layout_toLeftOf="@+id/button2"
    android:text="@string/four" />

<Button
    android:id="@+id/button5"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBottom="@+id/button4"
    android:layout_alignLeft="@+id/button2"
    android:layout_alignStart="@+id/button2"
    android:text="@string/five" />

<Button
    android:id="@+id/button6"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/button3"
    android:layout_alignStart="@+id/button3"
    android:layout_below="@+id/button3"
    android:text="@string/six" />

<Button
    android:id="@+id/button7"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/button4"
    android:layout_toLeftOf="@+id/button2"
    android:text="@string/seven" />

<Button
    android:id="@+id/button8"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/button5"
    android:layout_alignStart="@+id/button5"
    android:layout_below="@+id/button5"
    android:text="@string/eight" />
```

```
<Button
    android:id="@+id/button9"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/button6"
    android:layout_alignStart="@+id/button6"
    android:layout_below="@+id/button6"
    android:text="@string/nine" />

<Button
    android:id="@+id/buttonadd"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignEnd="@+id/edt1"
    android:layout_alignRight="@+id/edt1"
    android:layout_alignTop="@+id/button3"
    android:layout_marginLeft="46dp"
    android:layout_marginStart="46dp"
    android:layout_toRightOf="@+id/button3"
    android:text="@string/Add" />

<Button
    android:id="@+id/buttonsub"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignEnd="@+id/buttonadd"
    android:layout_alignLeft="@+id/buttonadd"
    android:layout_alignRight="@+id/buttonadd"
    android:layout_alignStart="@+id/buttonadd"
    android:layout_below="@+id/buttonadd"
    android:text="@string/sub" />

<Button
    android:id="@+id/buttonmul"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/buttonsub"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_alignStart="@+id/buttonsub"
    android:layout_below="@+id/buttonsub"
    android:text="@string/mul" />

<Button
    android:id="@+id/button10"
    style="?android:attr/buttonStyleSmall"
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/button7"
    android:layout_toLeftOf="@+id/button2"
    android:text="@string/dot" />

<Button
    android:id="@+id/button0"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/button8"
    android:layout_alignStart="@+id/button8"
    android:layout_below="@+id/button8"
    android:text="@string/zero" />

<Button
    android:id="@+id/buttonC"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/button9"
    android:layout_alignStart="@+id/button9"
    android:layout_below="@+id/button9"
    android:text="@string/bracket" />

<Button
    android:id="@+id/buttondiv"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignEnd="@+id/buttonmul"
    android:layout_alignLeft="@+id/buttonmul"
    android:layout_alignRight="@+id/buttonmul"
    android:layout_alignStart="@+id/buttonmul"
    android:layout_below="@+id/buttonmul"
    android:text="@string/div" />

<Button
    android:id="@+id/buttoneql"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignEnd="@+id/buttondiv"
    android:layout_alignLeft="@+id/button10"
    android:layout_alignRight="@+id/buttondiv"
    android:layout_alignStart="@+id/button10"
    android:layout_below="@+id/button0"
    android:layout_marginTop="37dp"
    android:text="@string/equal" />
```

</RelativeLayout>

MainActivity.java

```
package com.example.calc1;
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends Activity {

        Button button0, button1, button2, button3, button4, button5, button6,
    button7, button8, button9, buttonAdd, buttonSub, buttonDivision,
    buttonMul, button10, buttonC, buttonEqual;
        EditText result;
        float mValueOne, mValueTwo;
        boolean addition, mSubtract, multiplication, division;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        button0 = (Button) findViewById(R.id.button0);
        button1 = (Button) findViewById(R.id.button1);
        button2 = (Button) findViewById(R.id.button2);
        button3 = (Button) findViewById(R.id.button3);
        button4 = (Button) findViewById(R.id.button4);
        button5 = (Button) findViewById(R.id.button5);
        button6 = (Button) findViewById(R.id.button6);
        button7 = (Button) findViewById(R.id.button7);
```

```
button8 = (Button) findViewById(R.id.button8);
button9 = (Button) findViewById(R.id.button9);
button10 = (Button) findViewById(R.id.button10);
buttonAdd = (Button) findViewById(R.id.buttonadd);
buttonSub = (Button) findViewById(R.id.buttonsub);
buttonMul = (Button) findViewById(R.id.buttonmul);
buttonDivision = (Button) findViewById(R.id.buttondiv);
buttonC = (Button) findViewById(R.id.buttonC);
buttonEqual = (Button) findViewById(R.id.buttoneql);
result = (EditText) findViewById(R.id.edt1);
button1.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {
      result.setText(result.getText() + "1");
   }
});
button2.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {
            result.setText(result.getText() + "2");
   }
});
button3.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {
            result.setText(result.getText() + "3");
   }
});
button4.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {
            result.setText(result.getText() + "4");
```

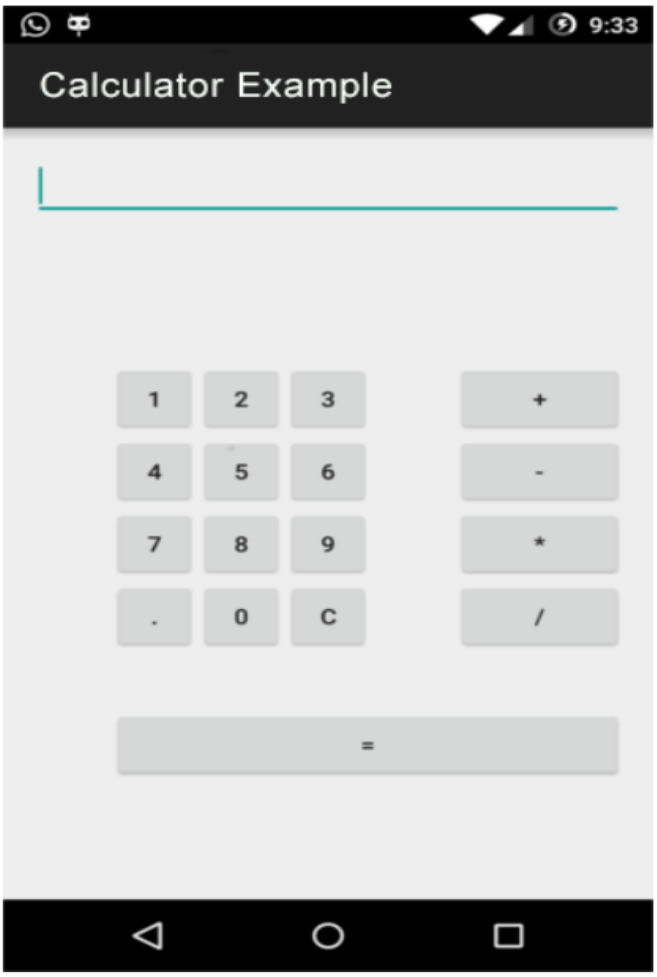
```
    }
});
button5.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
            result.setText(result.getText() + "5");
    }
});
button6.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
            result.setText(result.getText() + "6");
    }
});
button7.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
            result.setText(result.getText() + "7");
    }
});
button8.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
            result.setText(result.getText() + "8");
    }
});
button9.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
            result.setText(result.getText() + "9");
    }
});
```

```
button0.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
            result.setText(result.getText() + "0");
    }
});
buttonAdd.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        if (result == null) {
            result.setText("");
        } else {
            mValueOne = Float.parseFloat(result.getText() + "");
            addition = true;
            result.setText(null);
        }
    }
});
buttonSub.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        mValueOne = Float.parseFloat(result.getText() + "");
        mSubtract = true;
        result.setText(null);
    }
});
buttonMul.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        mValueOne = Float.parseFloat(result.getText() + "");
        multiplication = true;
```

```
        result.setText(null);
    }
});
buttonDivision.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        mValueOne = Float.parseFloat(result.getText() + "");
        division = true;
        result.setText(null);
    }
});
buttonEqual.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        mValueTwo = Float.parseFloat(result.getText() + "");

        if (addition == true) {
            result.setText(mValueOne + mValueTwo + "");
            addition = false;
        }
        if (mSubtract == true) {
            result.setText(mValueOne - mValueTwo + "");
            mSubtract = false;
        }
        if (multiplication == true) {
            result.setText(mValueOne * mValueTwo + "");
            multiplication = false;
        }

        if (division == true) {
            result.setText(mValueOne / mValueTwo + "");
            division = false;
```

```
            }
        }
    });
    buttonC.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
                result.setText("");
        }
    });
    button10.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
                result.setText(result.getText() + ".");
        }
    });
  }
  @Override
  public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
  }

}
```

Sample Output


9:33
Calculator Example
1
2
3
+
4
5
6
-
7
8
9
*
.
0
C
/
=

Program 4

Design a registration activity and store registration details in local memory of phone using Intents and SharedPreferences

Shared Preferences is the way in which one can store and retrieve small amounts of primitive data as key/value pairs to a file on the device storage such as String, int, float, Boolean that make up your preferences in an XML file inside the app on the device storage.

Step 1:- Make a new project named, SharedPreferenceSample.
Step 2:- As there will be two activity in the project, lets first define the xml for both.

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:gravity="fill_horizontal"
   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_Register"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignParentLeft="true"
      android:layout_marginLeft="80dp"
      android:text="Registeration"
      android:textAppearance="?android:attr/textAppearanceLarge" />
   <TextView
      android:id="@+id/textView1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignParentLeft="true"
      android:layout_below="@+id/textview_Register"
      android:layout_marginTop="25dp"
      android:text="Enter Name"
      android:textAppearance="?android:attr/textAppearanceMedium" />
   <EditText
      android:id="@+id/editText_name"
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:layout_alignLeft="@+id/textView1"
      android:layout_below="@+id/textView1"
      android:ems="10" >

   </EditText>
```

```
    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/editText_name"
        android:layout_below="@+id/editText_name"
        android:layout_marginTop="19dp"
        android:text="E-mail:"
        android:textAppearance="?android:attr/textAppearanceMedium" />
    <EditText
        android:id="@+id/editText_mail"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/textView2"
        android:layout_below="@+id/textView2"
        android:layout_marginTop="20dp"
        android:ems="10" >
    </EditText>
    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/editText_mail"
        android:layout_below="@+id/editText_mail"
        android:layout_marginTop="31dp"
        android:text="Password:"
        android:textAppearance="?android:attr/textAppearanceMedium" />
    <EditText
        android:id="@+id/editText_password"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/textView3"
        android:layout_below="@+id/textView3"
        android:layout_marginTop="24dp"
        android:ems="10" >
    </EditText>
    <Button
        android:id="@+id/button_register"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/editText_password"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="20dp"
        android:text="Register" />
</RelativeLayout>
```

Create another xml file and named it as second.xml

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

```
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="10dp" >
    <TextView
        android:id="@+id/second_text_name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:text="Username"
        android:textAppearance="?android:attr/textAppearanceMedium" />
    <TextView
        android:id="@+id/second_text_email"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:text="Email"
        android:textAppearance="?android:attr/textAppearanceMedium" />
    <TextView
        android:id="@+id/second_text_pass"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:text="Password"
        android:textAppearance="?android:attr/textAppearanceMedium" />
</LinearLayout>
```

Step 3:- MainActivity.java

```
package com.arpit.sharedpreferencesample;
import android.app.Activity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.SharedPreferences.Editor;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
public class MainActivity extends Activity {
    SharedPreferences pref;
    Editor editor;
    Button btn_register;
    EditText et_name, et_pass, et_email;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    et_name = (EditText) findViewById(R.id.editText_name);
    et_pass = (EditText) findViewById(R.id.editText_password);
    et_email = (EditText) findViewById(R.id.editText_mail);
```

```
    btn_register = (Button) findViewById(R.id.button_register);

    // creating an shared Preference file for the information to be stored
    // first argument is the name of file and second is the mode, 0 is private mode

    pref = getSharedPreferences("Registration", 0);
    // get editor to edit in file
    editor = pref.edit();

    // the tap of button we will fetch the information from three edittext
btn_register.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
        String name = et_name.getText().toString();
        String email = et_email.getText().toString();
        String pass = et_pass.getText().toString();

        if(et_name.getText().length()<=0){
           Toast.makeText(MainActivity.this, "Enter name", Toast.LENGTH_SHORT).show();
        }
        else if( et_email.getText().length()<=0){
           Toast.makeText(MainActivity.this, "Enter email", Toast.LENGTH_SHORT).show();
        }
        else if( et_pass.getText().length()<=0){
           Toast.makeText(MainActivity.this, "Enter password",
Toast.LENGTH_SHORT).show();
        }
        else{

        // as now we have information in string. Lets stored them with the help of editor
        editor.putString("Name", name);
        editor.putString("Email",email);
        editor.putString("Password",pass);
        editor.commit();   // commit the values

        // after saving the value open next activity
        Intent ob = new Intent(MainActivity.this, Second.class);
        startActivity(ob);

        }
        }
    });
   }
}
```

Step 4:- Create a second class file and named it as Second class

```
package com.arpit.sharedpreferencesample;
import android.app.Activity;
import android.content.SharedPreferences;
```

```
import android.os.Bundle;
import android.widget.TextView;
// in this activity we will fetch the value entered in main activity
public class Second extends Activity{
    TextView tv_name, tv_pass, tv_email;
    SharedPreferences pref;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        setContentView(R.layout.second);
    tv_name = (TextView) findViewById(R.id.second_text_name);
    tv_email = (TextView) findViewById(R.id.second_text_email);
    tv_pass = (TextView) findViewById(R.id.second_text_pass);
    pref = getSharedPreferences("Registration", 0);
   // retrieving value from Registration
    String name = pref.getString("Name", null);
    String email = pref.getString("Email", null);
    String password = pref.getString("Password", null);

    // Now set these value into textview of second activity
    tv_name.setText(name);
    tv_pass.setText(password);
    tv_email.setText(email);
     }
}
```

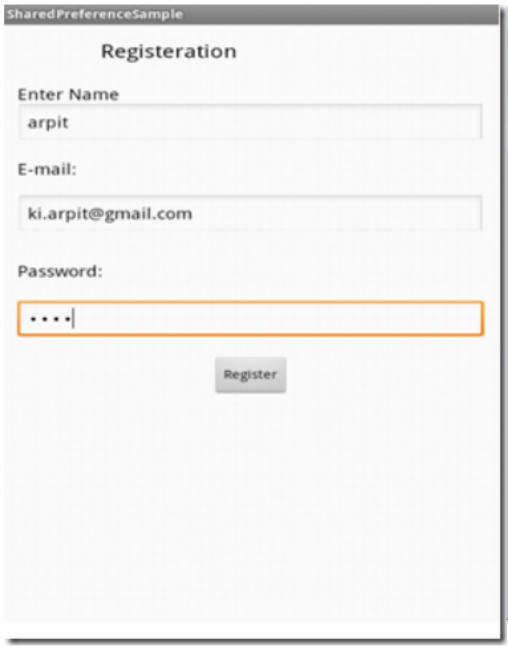
Step 5:- Edit Android Manifest file

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

</manifest>
```

Result:-

Entered information



Fetched value on second activity

Program 5

Develop an application using FrameLayout

In android, **Framelayout** is a **ViewGroup** subclass that is used to specify the position of **View** instances it contains on the top of each other to display only single **View** inside the FrameLayout. FrameLayout is designed to block out an area on the screen to display a single item. We can add multiple children to a FrameLayout and control their position by assigning gravity to each child, using the **android:layout_gravity** attribute.

Activity_main.xml

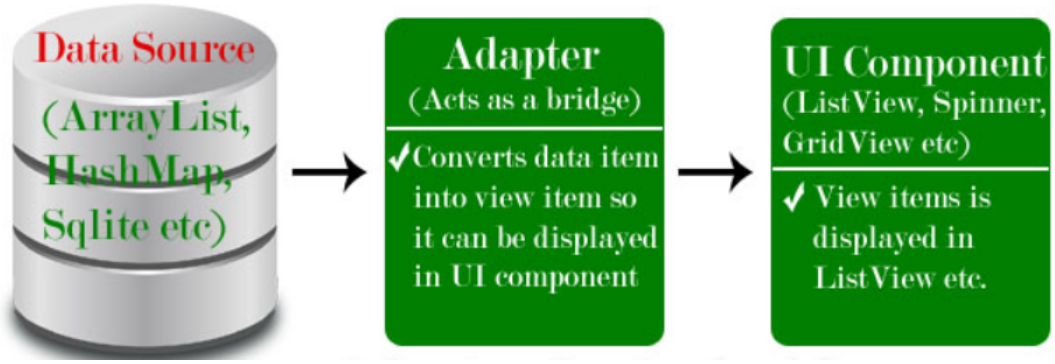
```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/ic_launcher"
    android:paddingBottom="@dimen/activity_vertical_margin"
```

```
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:visibility="visible"
    tools:context=".MainActivity" >

    <ToggleButton
        android:id="@+id/toggleButton1"
        android:layout_width="87dp"
        android:layout_height="68dp"
        android:background="@drawable/toggle"
        android:text="@string/ToggleButton"
        android:visibility="visible" />

</FrameLayout>
```

Sample output

6. Implement Adapters and perform exception handling.

In Android, Adapter is a bridge between UI component and data source that helps us to fill data in UI component. It holds the data and send the data to an Adapter view then view can takes the data from the adapter view and shows the data on different views like as ListView, GridView, Spinner etc. For more customization in Views ,the base adapter or custom adapters are used.



MainActivity.java

```
package com.example.arrayadapter;

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 {
        // Array of strings...
          String[] mobileArray = {"Android","IPhone","WindowsMobile","Blackberry",
            "WebOS","Ubuntu","Windows7","Max OS X"};
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
           R.layout.activity_listview, mobileArray);

        ListView listView = (ListView) findViewById(R.id.mobile_list);
        listView.setAdapter(adapter);
   }
   @Override
   public boolean onCreateOptionsMenu(Menu menu) {
      // Inflate the menu; this adds items to the action bar if it is present.
      getMenuInflater().inflate(R.menu.main, menu);
      return true;
   }
}
```

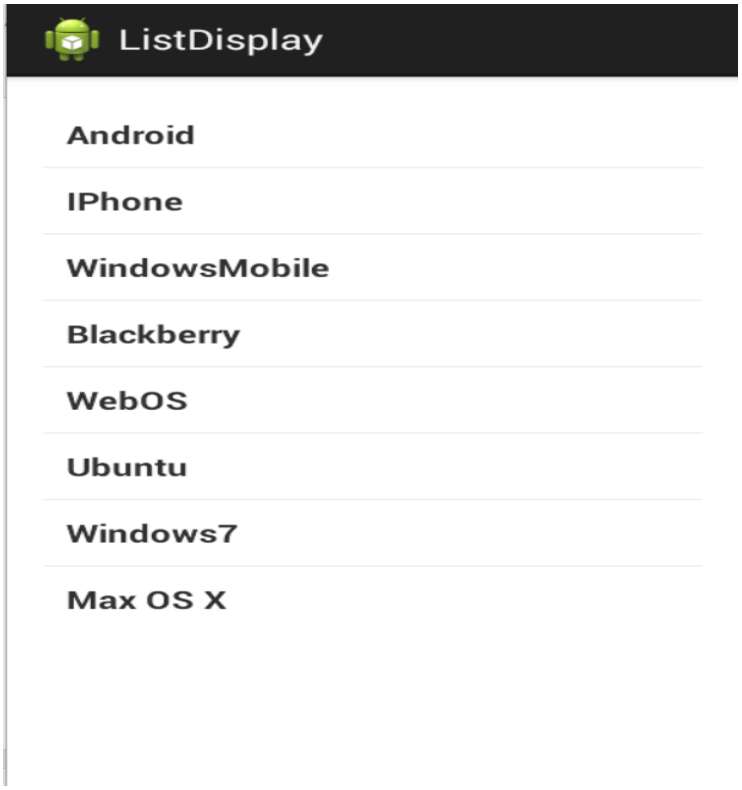
Activity_main.xml

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

   <ListView
     android:id="@+id/mobile_list"
     android:layout_width="match_parent"
     android:layout_height="wrap_content" >
```

```
  </ListView>
</LinearLayout>
```

Activity_list_view.xml

```
<?xml version="1.0" encoding="utf-8"?>
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
  android:id="@+id/label"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:padding="10dip"
  android:textSize="16sp"
  android:textStyle="bold" >
</TextView>
```



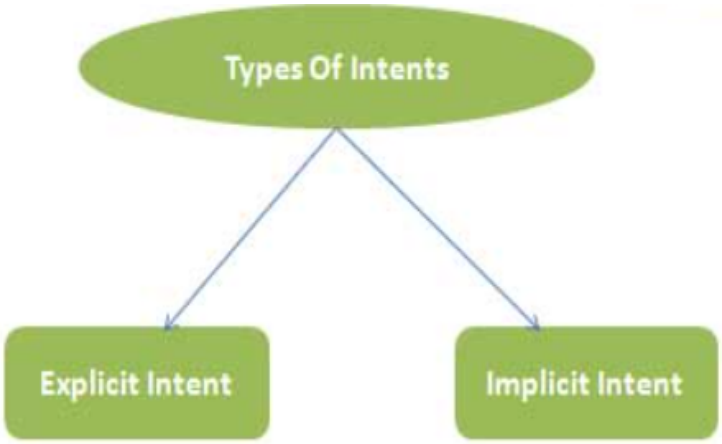
Program 6

Implement Intent to navigate between multiple activities

An *intent* is an object representing some action to be performed. The most common, but certainly not only, use for an intent is to launch an activity. There are two types of intents—**implicit** and **explicit**. An **explicit intent** is highly specific, where you know the exact activity to be launched, often a screen in your own app.

An **implicit intent** is a bit more abstract, where you tell the system the type of action, such as opening a link, composing an email, or making a phone call, and the system is

responsible for figuring out how to fulfill the request. You've probably seen both kinds of intents in action without knowing it. Generally, when showing an activity in your own app, you use an explicit intent.



Activity_Main.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android: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:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:textAppearance="?android:attr/textAppearanceMedium"
      android:text="@string/click"
      android:id="@+id/textView2"
      android:clickable="false"
      android:layout_alignParentTop="true"
      android:layout_alignParentStart="true"
      android:layout_marginTop="42dp"
      android:background="#3e7d02"
android:textColor="#ffffff" />
   <Button
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="@string/ExplicitIntentExample"
      android:id="@+id/explicit_Intent"
      android:layout_alignParentTop="true"
      android:layout_centerHorizontal="true"
      android:layout_marginTop="147dp" />
```

```
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/ImplicitIntentExample"
        android:id="@+id/implicit_Intent"
        android:layout_centerVertical="true"
        android:layout_centerHorizontal="true" />

</RelativeLayout>
```

Create another xml file named it as activity_second.xml

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

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:text="@string/NextPage"
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/textView"
        android:layout_alignRight="@+id/textView"
        android:layout_marginBottom="19dp"
        android:layout_marginRight="34dp"
        android:contentDescription="@string/back"
        android:text="@string/back" />

</RelativeLayout>
```

MainActivity.java

```
package com.example.intent;

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.widget.Button;
public class MainActivity extends Activity {
Button explicit_btn, implicit_btn;
 @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      explicit_btn = (Button)findViewById(R.id.explicit_Intent);
      implicit_btn = (Button) findViewById(R.id.implicit_Intent);
    //implement Onclick event for Explicit Intent
      explicit_btn.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
           Intent intent = new  Intent(getBaseContext(), SecondActivity.class);
            startActivity(intent);
   }
      });
    //implement onClick event for Implicit Intent
   implicit_btn.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
           Intent intent = new Intent(Intent.ACTION_VIEW);
            intent.setData(Uri.parse("https://www.google.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.main, menu);
        return true;
    }
}
```

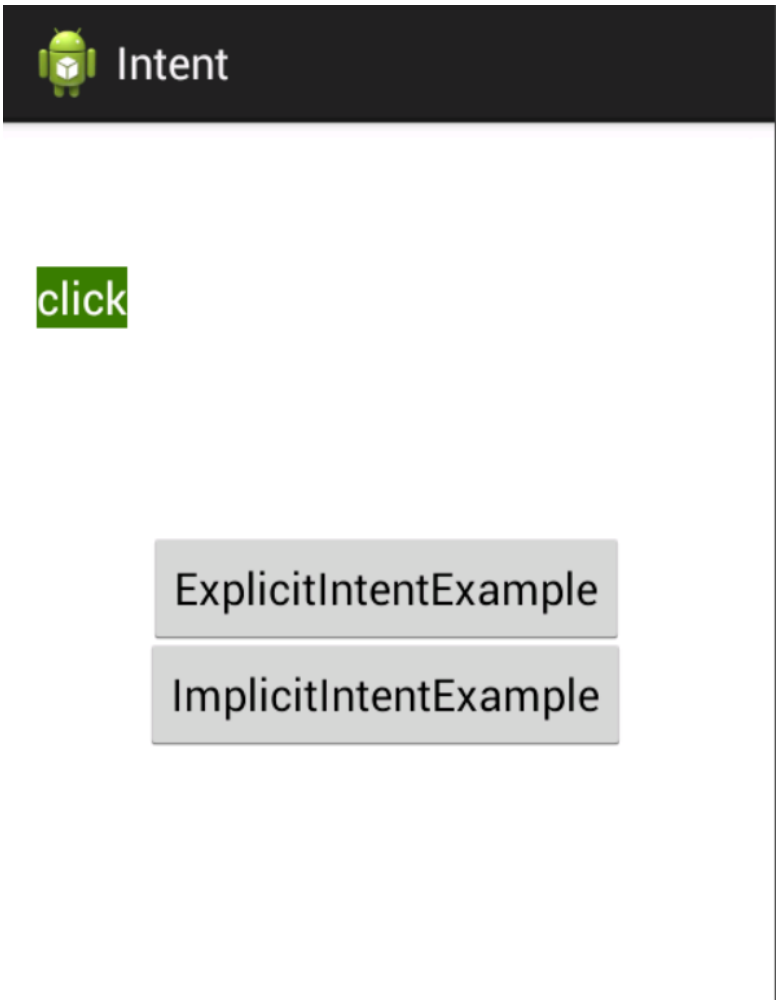
Create another java file and named it as SecondActivity.java

```
package com.example.intent;

import android.app.Activity;
import android.os.Bundle;
import android.widget.Toast;

public class SecondActivity extends Activity{


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

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

    }
}
```

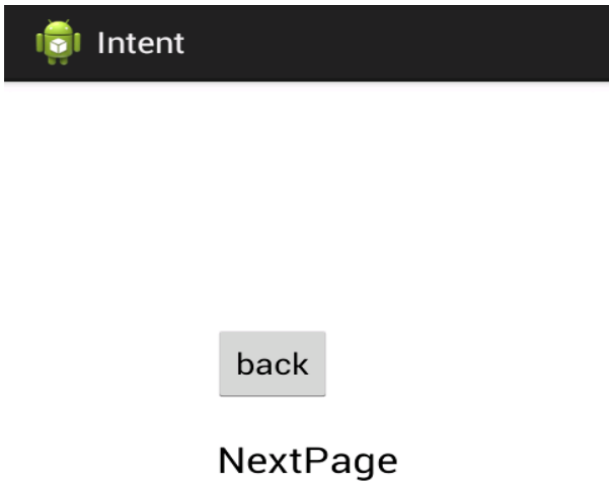
AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.intent"
    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.intent.MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
```

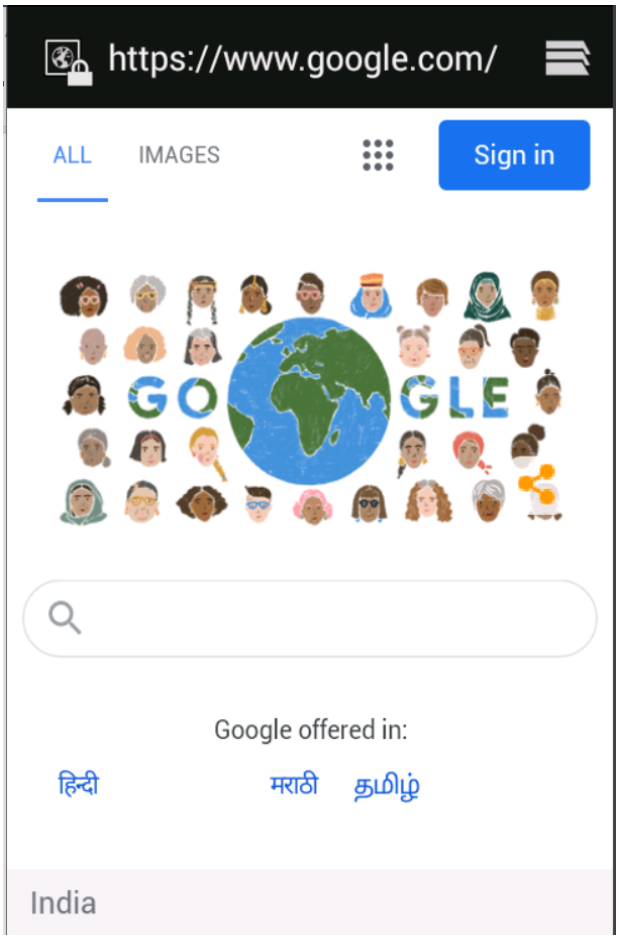
```
            <action android:name="android.intent.action.MAIN" />

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

    </activity>
  </application>

</manifest>
```



On clicking explicit Intent, move to second page.



On clicking implicit intent,move to corresponding url:


https://www.google.com/
ALL
IMAGES
Sign in
GO
GLE
Google offered in:
हिन्दी
मराठी
தமிழ்
India

Program 7

Implement Options Menu to navigate to activities

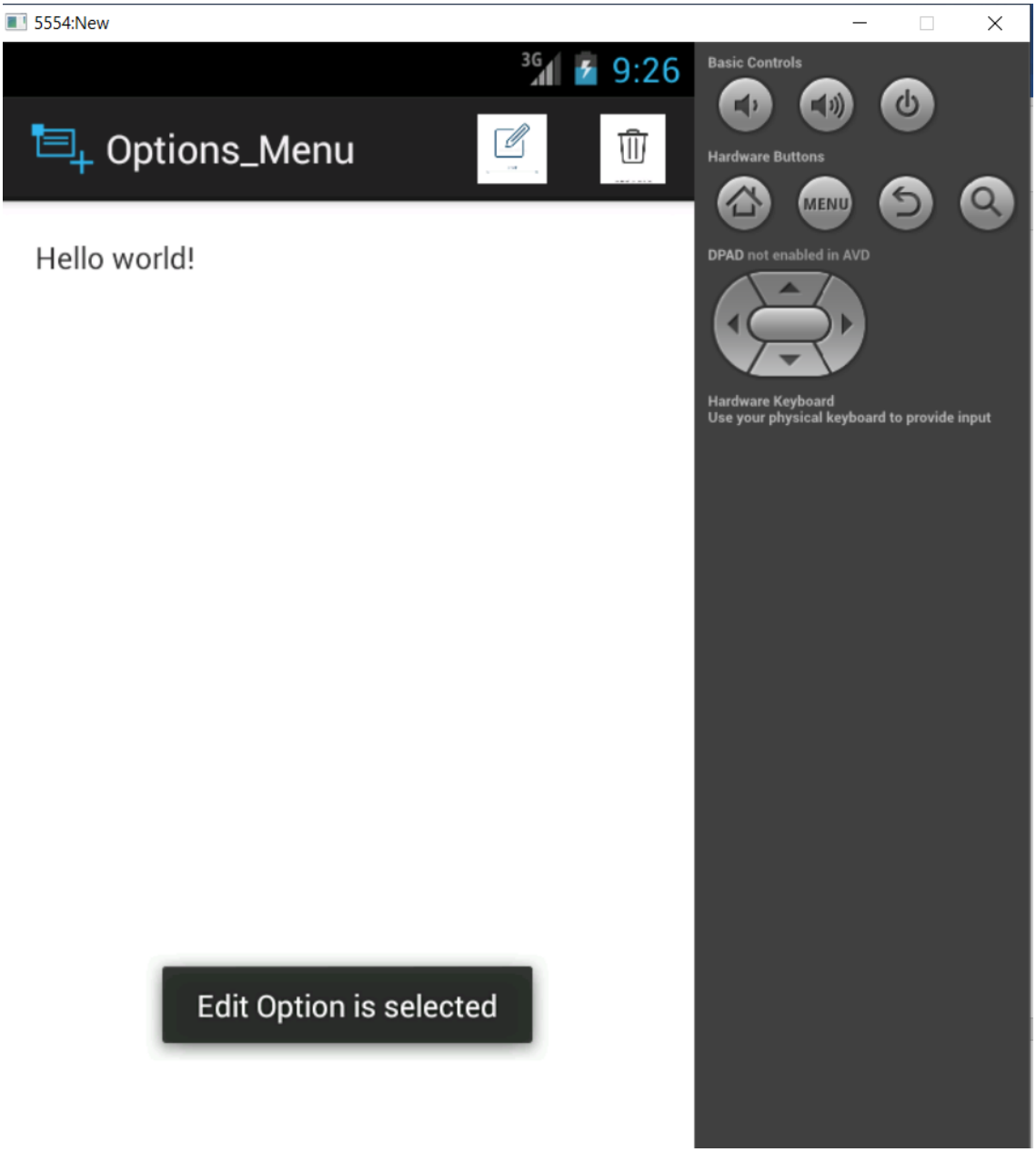
Create an xml file inside menu->mymenu.xml

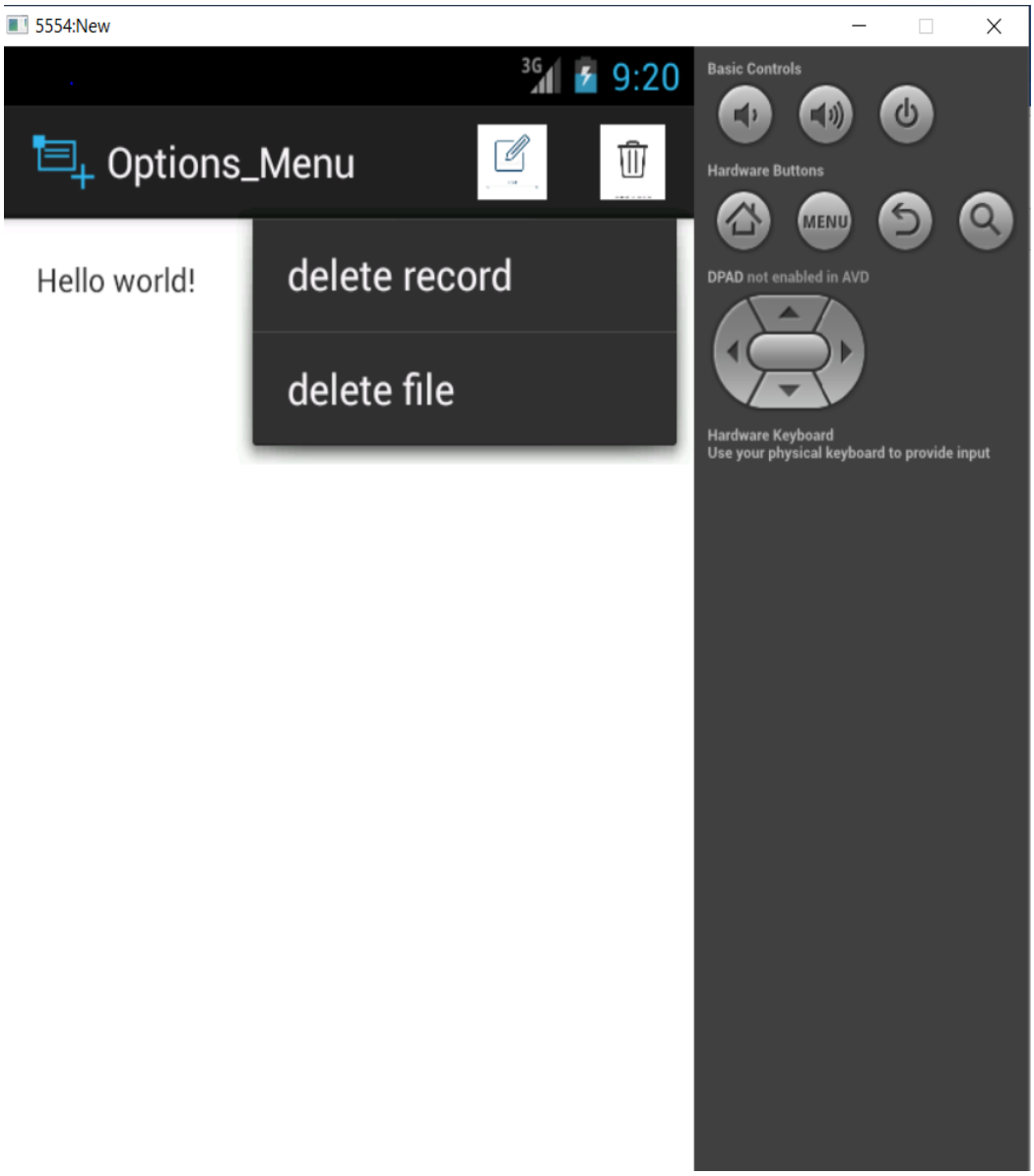
```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
 <item
    android:id="@+id/item_edit"
    android:title="@string/Edit"
    android:icon="@drawable/edit"
    android:showAsAction="ifRoom"/>
 <item
    android:id="@+id/item_delete"
    android:title="@string/delete"
    android:icon="@drawable/delete"
    android:showAsAction="ifRoom">
         <menu>
         <item
         android:id="@+id/item_delete_record"
         android:title="@string/deleterecord"/>
       <item
         android:id="@+id/item_delete_file"
         android:title="@string/deletefile"/>

         </menu>
 </item>
 <item
       android:id="@+id/item_file"
       android:title="@string/fileoptions"
       android:showAsAction="collapseActionView">
 <menu>

    <item
       android:id="@+id/item_file_save"
       android:title="@string/FileSave"/>
    <item
       android:id="@+id/item_file_open"
       android:title="@string/Fileopen"/>
 </menu>

 </item>
 </menu>
```

mainActivity.java

```
package com.example.options_menu;
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Toast;
```

```
public class MainActivity extends Activity {

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
   }
@Override
   public boolean onCreateOptionsMenu(Menu menu) {
      // Inflate the menu; this adds items to the action bar if it is present.
      getMenuInflater().inflate(R.menu.mymenu, menu);
      return true;
   }


        @Override
        public boolean onOptionsItemSelected(MenuItem item) {
                // TODO Auto-generated method stub
                if(item.getItemId()==R.id.item_edit)
                {
                        Toast.makeText(getApplicationContext(),"Edit Option is selected",
Toast.LENGTH_LONG).show();
                }
                if(item.getItemId()==R.id.item_delete_file)
                {
                        Toast.makeText(getApplicationContext(),"delete file Option is
selected", Toast.LENGTH_LONG).show();
                }
                if(item.getItemId()==R.id.item_delete_record)
                {
                        Toast.makeText(getApplicationContext(),"delete record Option is
selected", Toast.LENGTH_LONG).show();
                }
                if(item.getItemId()==R.id.item_file_open)
                {
                        Toast.makeText(getApplicationContext(),"File open is selected",
Toast.LENGTH_LONG).show();
                }
                if(item.getItemId()==R.id.item_file_save)
                {
                        Toast.makeText(getApplicationContext(),"file save option is selected",
Toast.LENGTH_LONG).show();
                }
                return super.onOptionsItemSelected(item);
        }

}
```


5554:New
3G
9:26
Options_Menu
Hello world!
Edit Option is selected
Basic Controls
Hardware Buttons
MENU
DPAD not enabled in AVD
Hardware Keyboard
Use your physical keyboard to provide input



Program 8

Develop an application that uses ArrayAdapter with ListView.

```
package com.example.arrayadapter;

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 {
        // Array of strings...
          String[] mobileArray = {"Android","IPhone","WindowsMobile","Blackberry",
            "WebOS","Ubuntu","Windows7","Max OS X"};
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
           R.layout.activity_listview, mobileArray);

        ListView listView = (ListView) findViewById(R.id.mobile_list);
        listView.setAdapter(adapter);
   }
   @Override
   public boolean onCreateOptionsMenu(Menu menu) {
      // Inflate the menu; this adds items to the action bar if it is present.
      getMenuInflater().inflate(R.menu.main, menu);
      return true;
   }
}
```

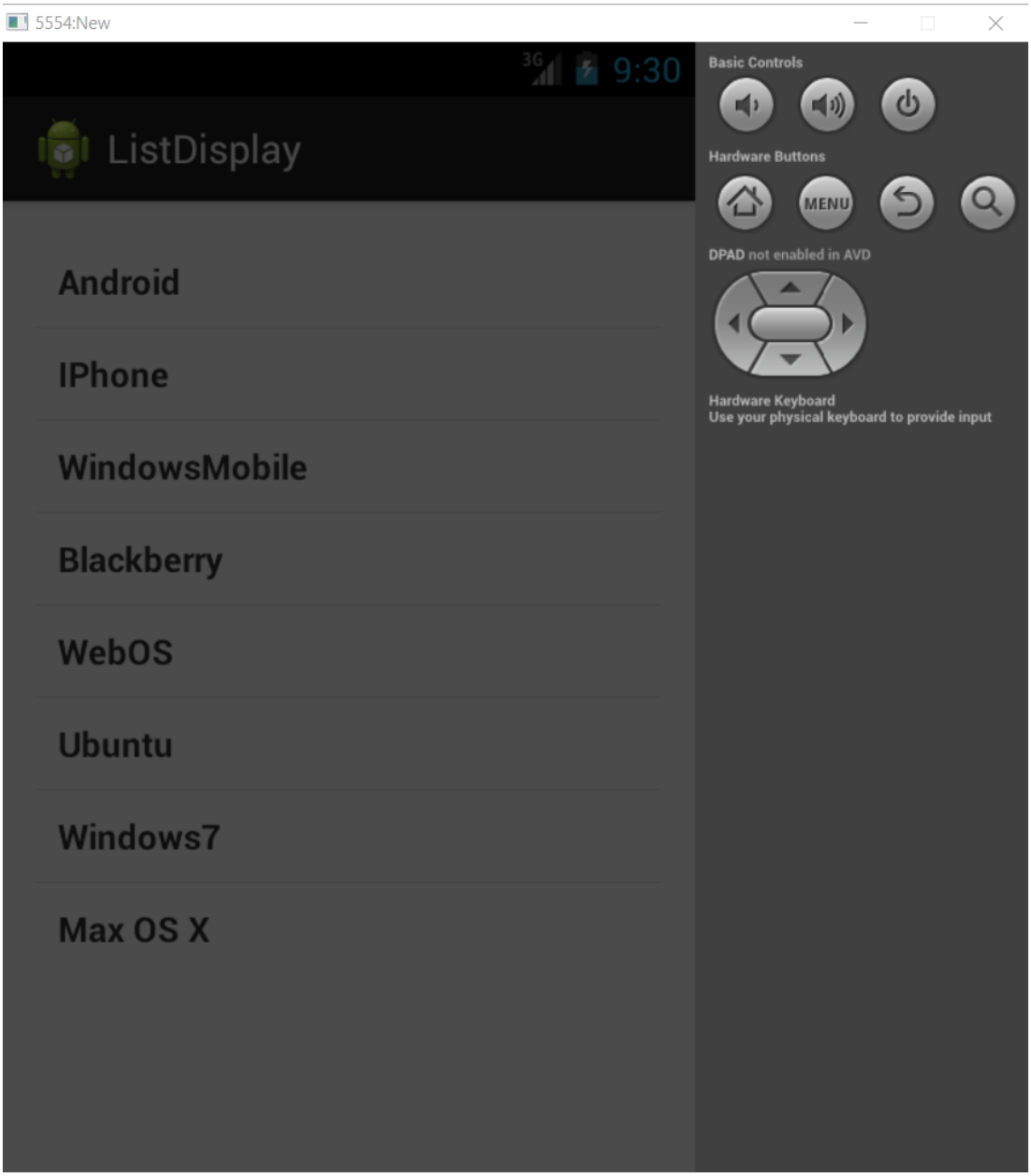
Activity_main.xml

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

   <ListView
     android:id="@+id/mobile_list"
     android:layout_width="match_parent"
     android:layout_height="wrap_content" >
   </ListView>
```

```
</LinearLayout>
```

Activity_list_view.xml

```
<?xml version="1.0" encoding="utf-8"?>
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
   android:id="@+id/label"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent"
   android:padding="10dip"
   android:textSize="16sp"
   android:textStyle="bold" >
</TextView>
```



Program 9

Develop an application that use GridView with images and display Alert box on selection

Activity_main.xml

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

<GridView
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:numColumns="2"
   android:stretchMode="columnWidth"
   android:columnWidth="72dp"
   android:id="@+id/dialog_gv"
   android:horizontalSpacing="2dp"
   android:verticalSpacing="2dp">

</GridView>

</LinearLayout>
```

Another xml file grid_adapter.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="96dp"
android:orientation="vertical">


<ImageView
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:src="@drawable/ic_launcher" />

<button
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:layout_alignParentRight="true"
   android:layout_alignParentTop="true" />
</RelativeLayout>
```

MainActivity.java

```
package com.example.gridviewimages;

import android.os.Bundle;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
```

```
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.View;
import android.widget.Toast;

public class MainActivity extends Activity {

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.grid_adapter);
      AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);
      //Yes Button
      builder.setPositiveButton("Yes", new DialogInterface.OnClickListener() {
         @Override
         public void onClick(DialogInterface dialog, int which) {
            Toast.makeText(getApplicationContext(), "Yes button Clicked",
Toast.LENGTH_LONG).show();
            dialog.dismiss();
         }
      });
      LayoutInflater inflater = getLayoutInflater();
      View dialoglayout = inflater.inflate(R.layout.grid_adapter, null);
      builder.setView(dialoglayout);
      builder.show();

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


5554:New
3G
9:39
Grid_view
Info
Privacy Policy
Basic Controls
Hardware Buttons
MENU
DPAD not enabled in AVD
Hardware Keyboard
Use your physical keyboard to provide input
Privacy Policy

Program 10

10. Develop an application that implements Spinner component and perform event handling

activity_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical"
  android:padding="10dip"
  android:layout_width="fill_parent"
  android:layout_height="wrap_content">

  <TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dip"
    android:text="@string/Category:"
    android:layout_marginBottom="5dp"/>

  <Spinner
    android:id="@+id/spinner"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:prompt="@string/spinnertitle"/>
  <Button
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="@string/NEXT"
      android:id="@+id/button"
      android:layout_alignParentBottom="true"
      android:layout_centerHorizontal="true"
      android:layout_marginBottom="137dp" />

</LinearLayout>
```

Create another file second.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical" >
   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:textAppearance="?android:attr/textAppearanceMedium"
      android:text="@string/Empty"
      android:id="@+id/txt_bundle"
```

```
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="103dp" />


</LinearLayout>
```

MainActivity.java

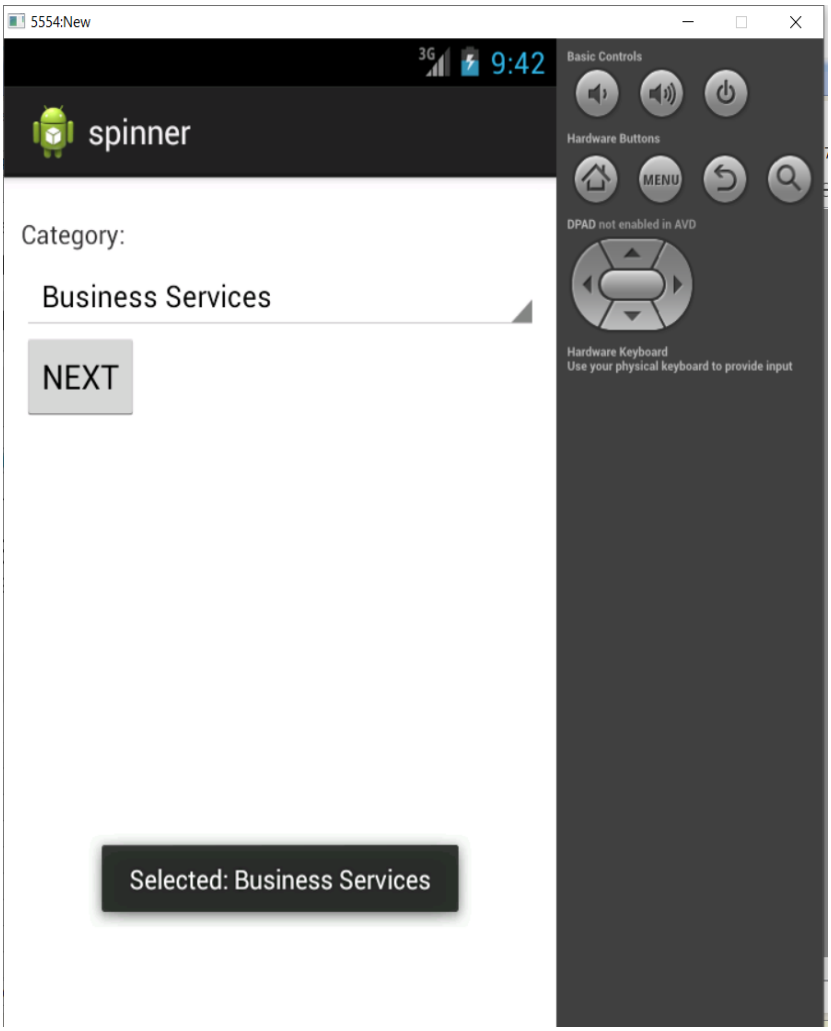
```
package com.example.spinner;

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

import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.Menu;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.Spinner;
import android.widget.Toast;

public class MainActivity extends Activity implements AdapterView.OnItemSelectedListener
{


   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
    // Spinner element
      final Spinner spinner = (Spinner) findViewById(R.id.spinner);
      Button button=(Button)findViewById(R.id.button);

      // Spinner click listener
      spinner.setOnItemSelectedListener(this);

      // Spinner Drop down elements
      List<String> categories = new ArrayList<String>();
      categories.add("Automobile");
      categories.add("Business Services");
      categories.add("Computers");
      categories.add("Education");
```

```
        categories.add("Personal");
        categories.add("Travel");

    // Creating adapter for spinner
        ArrayAdapter<String> dataAdapter = new ArrayAdapter<String>(this,
android.R.layout.simple_spinner_item, categories);

    // Drop down layout style - list view with radio button

dataAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);

        // attaching data adapter to spinner
        spinner.setAdapter(dataAdapter);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent= new Intent(MainActivity.this,SecondActivity.class);
                intent.putExtra("data",String.valueOf(spinner.getSelectedItem()));
                startActivity(intent);
            }
        });
    }
    public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
      // On selecting a spinner item
      String item = parent.getItemAtPosition(position).toString();

      // Showing selected spinner item
      Toast.makeText(parent.getContext(), "Selected: " + item,
Toast.LENGTH_LONG).show();
    }
    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.main, menu);
        return true;
    }

}
```

Create SecondActivity.java

```
package com.example.spinner;
```

```
import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;

public class SecondActivity extends Activity {
        @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.second);

      TextView  textView=(TextView) findViewById(R.id.txt_bundle);
      Bundle bundle=getIntent().getExtras();
      String data=bundle.get("data").toString();
      textView.setText(data);
   }

}
```



Program 11

Develop an application as a micro project which uses SQLite database as an assignment

SQLite is an Open Source database. SQLite supports standard relational database features like SQL syntax, transactions and prepared statements. The database requires limited memory at runtime (approx. 250 KByte) which makes it a good candidate from being embedded into other runtimes. SQLite supports the data types TEXT (similar to String in Java), INTEGER (similar to long in Java) and REAL (similar to double in Java). All other types must be converted into one of these fields before getting saved in the database. SQLite itself does not validate if the types written to the columns are actually of the defined type, e.g. you can write an integer into a string column and vice versa

SQLite in Android SQLite is embedded into every Android device. Using an SQLite database in Android does not require a setup procedure or administration of the database. You only have to define the SQL statements for creating and updating the database. Afterwards the database is automatically managed for you by the Android platform. Access to a SQLite

database involves accessing the file system. This can be slow. Therefore it is recommended to perform database operations asynchronously. I f your application creates a database, this database is by default saved in the directory DATA/data/APP_NAME/databases/FILENAME

The parts of the above directory are constructed based on the following rules. DATA is the path which theEnvironment.getDataDirectory() method returns. APP_NAME is your application name. FILENAME is the name you specify in your application code for the database.

SQLite architecture

Packages

The android.database package contains all necessary classes for working with databases. The android.database.sqlitepackage contains the SQLite specific classes.

Creating and updating database with SQLiteOpenHelper

To create and upgrade a database in your Android application you create a subclass of the SQLiteOpenHelper class. In the constructor of your subclass you call the super() method of SQLiteOpenHelper, specifying the database name and the current database version.

In this class you need to override the following methods to create and update your database.

onCreate() - is called by the framework, if the database is accessed but not yet created.

onUpgrade() - called, if the database version is increased in your application code. This method allows you to update an existing database schema or to drop the existing database and recreate it via the onCreate() method.

Both methods receive an SQLiteDatabase object as parameter which is the Java representation of the database.

The SQLiteOpenHelper class providest the getReadableDatabase() and getWriteableDatabase() methods to get access to anSQLiteDatabase object; either in read or write mode.

The database tables should use the identifier _id for the primary key of the table. Several Android functions rely on this standard.

SQLiteDatabase

SQLiteDatabase is the base class for working with a SQLite database in Android and provides methods to open, query, update and close the database.

More specifically SQLiteDatabase provides the insert(), update() and delete() methods.

In addition it provides the execSQL() method, which allows to execute an SQL statement directly.

The object ContentValues allows to define key/values. The key represents the table column identifier and the value represents the content for the table record in this column. ContentValues can be used for inserts and updates of database entries.

Queries can be created via the rawQuery() and query() methods or via the SQLiteQueryBuilder class .

rawQuery() directly accepts an SQL select statement as input.

query() provides a structured interface for specifying the SQL query.

SQLiteQueryBuilder is a convenience class that helps to build SQL queries.

The following gives an example of a query() call.

```
return database.query(DATABASE_TABLE, new String[] { KEY_ROWID,
KEY_CATEGORY, KEY_SUMMARY, KEY_DESCRIPTION },  null, null, null,
null, null);
```

The method query() has the following parameters.

Table 1. Parameters of the query() method

Parameter	Comment
String dbName	The table name to compile the query against.
String[] columnNames	A list of which table columns to return. Passing "null" will return all columns.
String whereClause	Where-clause, i.e. filter for the selection of data, null will select all data.
String[] selectionArgs	You may include ?s in the "whereClause"". These placeholders will get replaced by the values from the selectionArgs array.
String[] groupBy	A filter declaring how to group rows, null will cause the rows to not be grouped.
String[] having	Filter for the groups, null means no filter.
String[] orderBy	Table columns which will be used to order the data, null means no ordering.

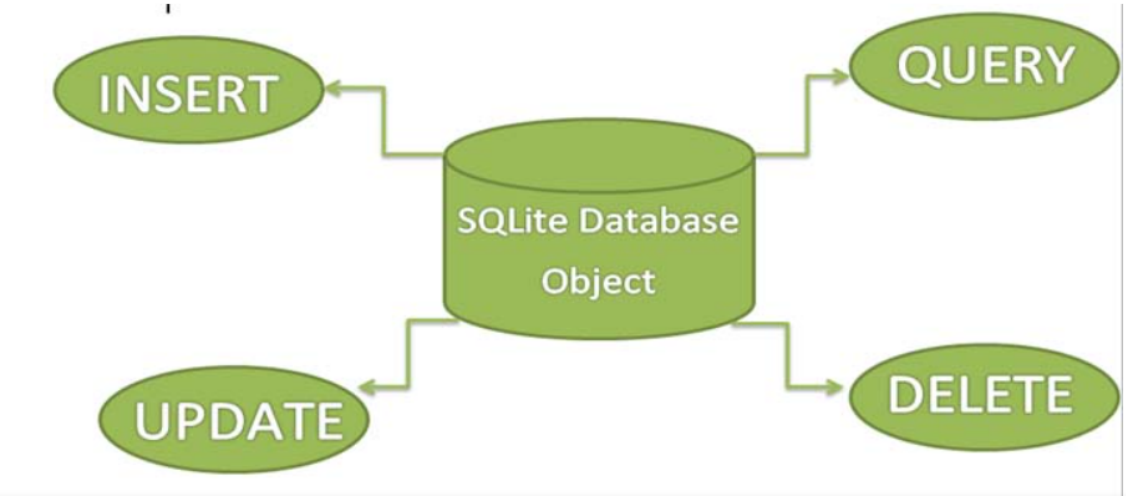
Cursor

A query returns a Cursor object. A Cursor represents the result of a query and basically points to one row of the query result. This way Android can buffer the query results efficiently; as it does not have to load all data into memory. To get the number of elements of the resulting

query use the getCount() method. To move between individual data rows, you can use the moveToFirst() and moveToNext() methods. The isAfterLast()method allows to check if the end of the query result has been reached. Cursor provides typed get*() methods, e.g. getLong(columnIndex), getString(columnIndex) to access the column data for the current position of the result. The "columnIndex" is the number of the column you are accessing. Cursor also provides the getColumnIndexOrThrow(String) method which allows to get the column index for a column name of the table. A Cursor needs to be closed with the close() method call.

ListViews, ListActivities and SimpleCursorAdapter

ListViews are Views which allow to display a list of elements. ListActivities are specialized activities which make the usage of ListViews easier. To work with databases and ListViews you can use the SimpleCursorAdapter. The SimpleCursorAdapter allows to set a layout for each row of the ListViews. You also define an array which contains the column names and another array which contains the IDs of Views which should be filled with the data. The SimpleCursorAdapter class will map the columns to the Viewsbased on the Cursor passed to it. To obtain the Cursor you should use the Loader class.



Open res -> layout -> activity_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:orientation="vertical" android:layout_width="match_parent"
   android:layout_height="match_parent">
   <TextView
      android:id="@+id/fstTxt"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginLeft="100dp"
      android:layout_marginTop="150dp"
      android:text="Name" />
   <EditText
      android:id="@+id/txtName"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
```

```
        android:layout_marginLeft="100dp"
        android:ems="10"/>
    <TextView
        android:id="@+id/secTxt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/Location"
        android:layout_marginLeft="100dp" />
    <EditText
        android:id="@+id/txtLocation"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="100dp"
        android:ems="10" />
    <TextView
        android:id="@+id/thirdTxt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Designation"
        android:layout_marginLeft="100dp" />
    <EditText
        android:id="@+id/txtDesignation"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="100dp"
        android:ems="10" />
    <Button
        android:id="@+id/btnSave"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="100dp"
        android:text="Save" />
</LinearLayout>
```

Open app -> java -> package -> MainActivity.java

```
package com.example.sql;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
public class MainActivity extends Activity {
    EditText Id,name, loc, desig;
    Button saveBtn;
    Intent intent;
    Button DeleteBtn;
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        name = (EditText)findViewById(R.id.txtName);
        loc = (EditText)findViewById(R.id.txtLocation);
        desig = (EditText)findViewById(R.id.txtDesignation);
        saveBtn = (Button)findViewById(R.id.btnSave);
        saveBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String username = name.getText().toString()+"\n";
                String location = loc.getText().toString();
                String designation = desig.getText().toString();
                DbHandler dbHandler = new DbHandler(MainActivity.this);
                dbHandler.insertUserDetails(username,location,designation);
                intent = new Intent(MainActivity.this,DetailsActivity.class);
                startActivity(intent);
                Toast.makeText(getApplicationContext(), "Details Inserted
Successfully",Toast.LENGTH_SHORT).show();
            }

        });

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

}
```

create a java class myDbAdapter. java.

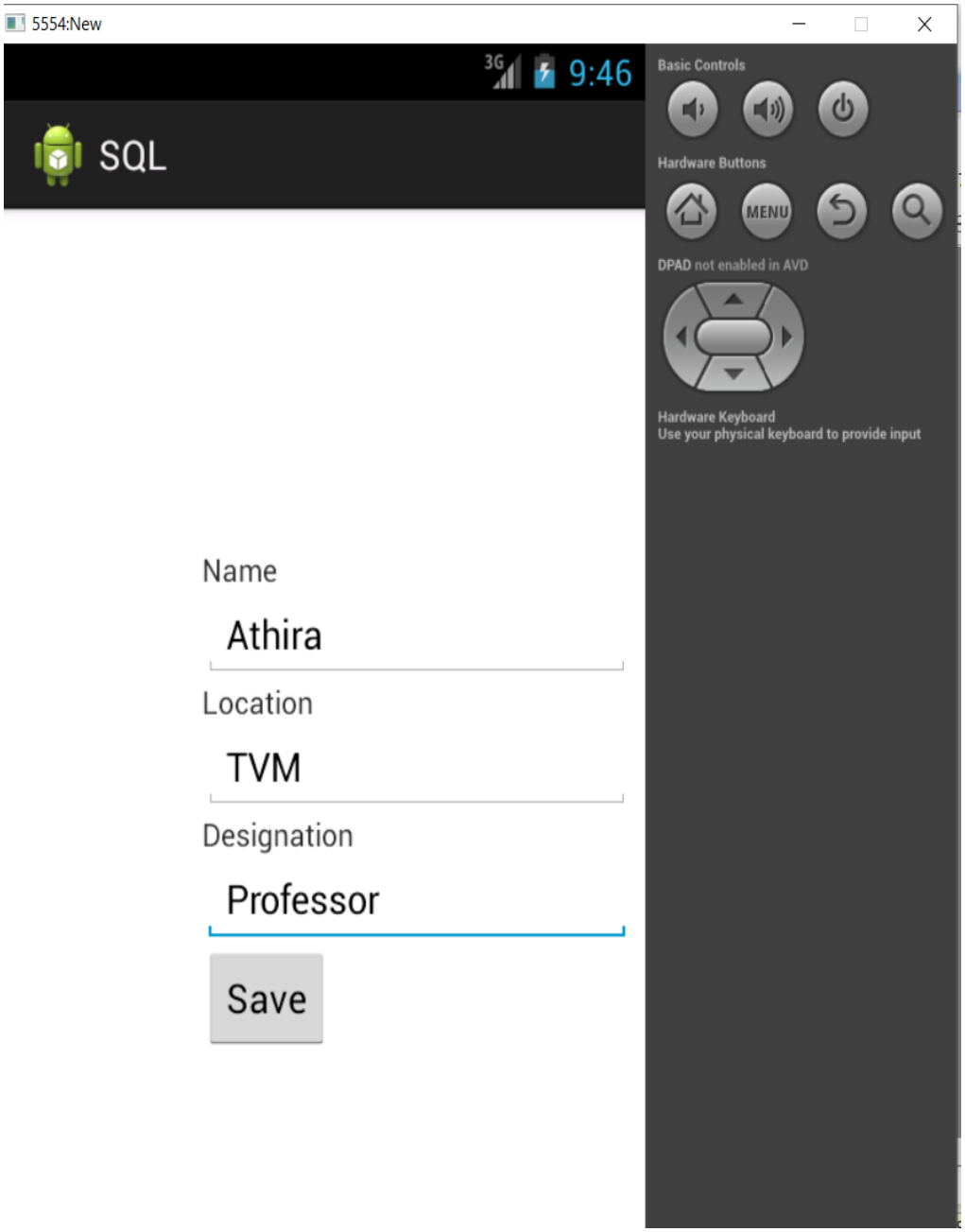
```
package com.example.sql;

import java.util.ArrayList;
import java.util.HashMap;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteDatabase.CursorFactory;
import android.database.sqlite.SQLiteOpenHelper;

public class DbHandler extends SQLiteOpenHelper {
        private static final int DB_VERSION = 1;
```

```
    private static final String DB_NAME = "usersdb";
    private static final String TABLE_Users = "userdetails";
    private static final String KEY_ID = "id";
    private static final String KEY_NAME = "name";
    private static final String KEY_LOC = "location";
    private static final String KEY_DESG = "designation";

        public DbHandler(Context context) {
                super(context,DB_NAME, null, DB_VERSION);
                // TODO Auto-generated constructor stub
        }

        @Override
        public void onCreate(SQLiteDatabase db) {
                String CREATE_TABLE = "CREATE TABLE " + TABLE_Users + "("
           + KEY_ID + " INTEGER PRIMARY KEY AUTOINCREMENT," + KEY_NAME
+ " TEXT,"
            + KEY_LOC + " TEXT,"
            + KEY_DESG + " TEXT"+ ")";
      db.execSQL(CREATE_TABLE);
                // TODO Auto-generated method stub

        }

        @Override
        public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
                // Drop older table if exist
      db.execSQL("DROP TABLE IF EXISTS " + TABLE_Users);
      // Create tables again
      onCreate(db);
                // TODO Auto-generated method stub

        }

        // Adding new User Details
    void insertUserDetails(String name, String location, String designation){
      //Get the Data Repository in write mode
      SQLiteDatabase db = this.getWritableDatabase();
      //Create a new map of values, where column names are the keys
      ContentValues cValues = new ContentValues();
      cValues.put(KEY_NAME, name);
      cValues.put(KEY_LOC, location);
      cValues.put(KEY_DESG, designation);
      // Insert the new row, returning the primary key value of the new row
      long newRowId = db.insert(TABLE_Users,null, cValues);
      db.close();
    }

 // Get User Details
    public ArrayList<HashMap<String, String>> GetUsers(){
```

```
        SQLiteDatabase db = this.getWritableDatabase();
        ArrayList<HashMap<String, String>> userList = new ArrayList<HashMap<String,
String>>();
        String query = "SELECT name, location, designation FROM "+ TABLE_Users;
        Cursor cursor = db.rawQuery(query,null);
        while (cursor.moveToNext()){
            HashMap<String,String> user = new HashMap<String, String>();
            user.put("name",cursor.getString(cursor.getColumnIndex(KEY_NAME)));
            user.put("designation",cursor.getString(cursor.getColumnIndex(KEY_DESG)));
            user.put("location",cursor.getString(cursor.getColumnIndex(KEY_LOC)));
            userList.add(user);
        }
        return  userList;
    }
 // Get User Details based on userid
    public ArrayList<HashMap<String, String>> GetUserByUserId(int userid){
        SQLiteDatabase db = this.getWritableDatabase();
        ArrayList<HashMap<String, String>> userList = new ArrayList<HashMap<String,
String>>();
        String query = "SELECT name, location, designation FROM "+ TABLE_Users;
        Cursor cursor = db.query(TABLE_Users, new String[]{KEY_NAME, KEY_LOC,
KEY_DESG}, KEY_ID+ "=?",new String[]{String.valueOf(userid)},null, null, null, null);
        if (cursor.moveToNext()){
            HashMap<String,String> user = new HashMap<String, String>();
            user.put("name",cursor.getString(cursor.getColumnIndex(KEY_NAME)));
            user.put("designation",cursor.getString(cursor.getColumnIndex(KEY_DESG)));
            user.put("location",cursor.getString(cursor.getColumnIndex(KEY_LOC)));
            userList.add(user);
        }
        return  userList;
    }
 // Delete User Details
    public void DeleteUser(int userid){
        SQLiteDatabase db = this.getWritableDatabase();
        db.delete(TABLE_Users, KEY_ID+" = ?",new String[] { Integer.toString(userid) });
        db.close();
    }
    // Update User Details
    public int UpdateUserDetails(String location, String designation, int id){
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues cVals = new ContentValues();
        cVals.put(KEY_LOC, location);
        cVals.put(KEY_DESG, designation);
        int count = db.update(TABLE_Users, cVals, KEY_ID+" = ?",new
String[]{String.valueOf(id)});
        return  count;
    }

}
```

In this step create another java class DetailsActivity.class

```
import java.util.ArrayList;
import java.util.HashMap;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ListAdapter;
import android.widget.ListView;
import android.widget.SimpleAdapter;

public class DetailsActivity extends Activity {
        Intent intent;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.details);
      DbHandler db = new DbHandler(this);
      ArrayList<HashMap<String, String>> userList = db.GetUsers();
      ListView lv = (ListView) findViewById(R.id.user_list);
      ListAdapter adapter = new SimpleAdapter(DetailsActivity.this, userList,
R.layout.list_row,new String[]{"name","designation","location"}, new int[]{R.id.name,
R.id.designation, R.id.location});
      lv.setAdapter(adapter);
      Button back = (Button)findViewById(R.id.btnBack);
      back.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            intent = new Intent(DetailsActivity.this,MainActivity.class);
            startActivity(intent);
         }
      });
   }

}
```


5554:New
9:46
SQL
Basic Controls
Hardware Buttons
MENU
DPAD not enabled in AVD
Hardware Keyboard
Use your physical keyboard to provide input
Name
Athira
Location
TVM
Designation
Professor
Save