**Mobile Devices Project**

**Members:**

* Radescu Roxana
* Atomulesei Geanina
* Bokor Bianca

**Group:** 1231E

**Login Application**

We will create an application with one home screen activity, in which we will use 2 buttons (one is used for ‘Sign In’ option and other is for ‘Sign Up’ option). Then we will show Login and ‘New Registration’ screen on the onClick option of the respective buttons. All required Login and Registration user-data will be save in application’s own Android SQLite database. The database will be stored in the application’s context, so that no other android applications will access the data.

Steps Required to Create Android Login Registration Application

1. Create a Home Screen Activity , Which will hold ‘Sign In‘ and ‘Sign Up‘ options.

2. Create layouts for home screen and ‘Sign In‘ and ‘Sign Up‘ Screens.

3. Create a SQLite Database in the application’s context, so that we can save all required user data (‘Sign In‘ and ‘Sign Up‘ details).

4. Code Logic in Application’s Java files.

5. Run Android Login Registration app on Device/Emulator.

**Android Layouts for Login Registration Application**

**1. Default layout (main.xml)**

This is used for home screen in this application, which holds 2 buttons for Sign In and Sign Up options.

**2. login.xml**

This xml is used for Log In (Sign In) screen in this application.

**3. signup.xml**

This xml is used for first time user, so for new registration we will use this xml file.

**Java Codes**

**1. HomeActivity.java**

Up’ options. Also we will just create a reference to the instance of SQLite Database for storing and querying data in the database.

**2. SignUPActivity.Java**

Here we will have complete logic, after we will click ‘Sign Up’ option from main screen.

**3. DataBaseHelper.Java**

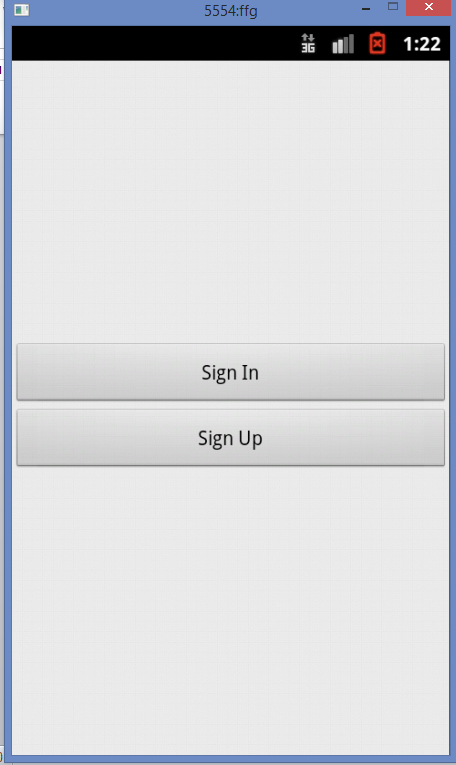
This file will be use to create a new DB, when no database exists in disk and the helper class will do the needful for us. Also this class will be help us to upgrade the version of the DB if required.

**4. LoginDataBaseAdapter.Java**

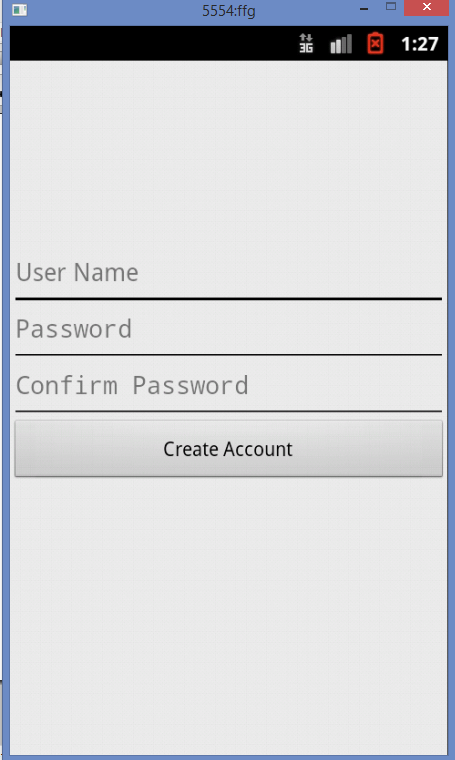
This file is required to handle all Database operations like (create DB, Insert record, update record, Delete record, Close DB, and Cursor related stuffs.

**Testing**

Once we run the application will appear the home screen which contains the ‘Log in’ and ‘Sign up’ buttons.



In order to create a new account we should press the ‘Sign up’ button.



We introduce the data that is required and press ‘Create Account’ button.



If the two passwords match, a message saying ‘ Account successfully created ’ will appear.

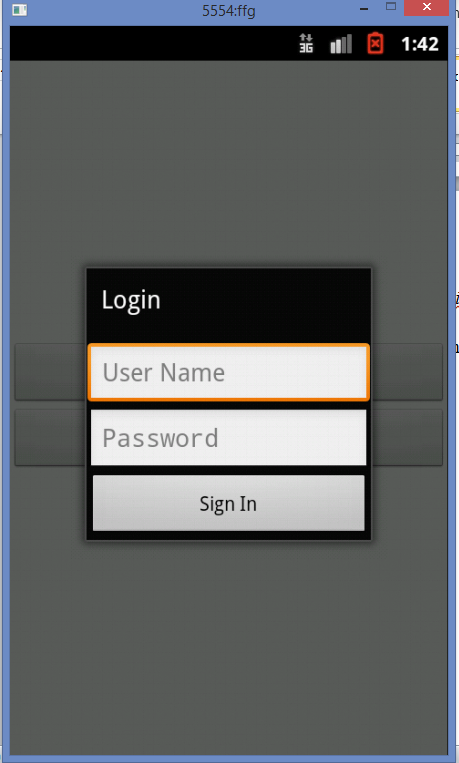


Otherwise there will appear an error message.

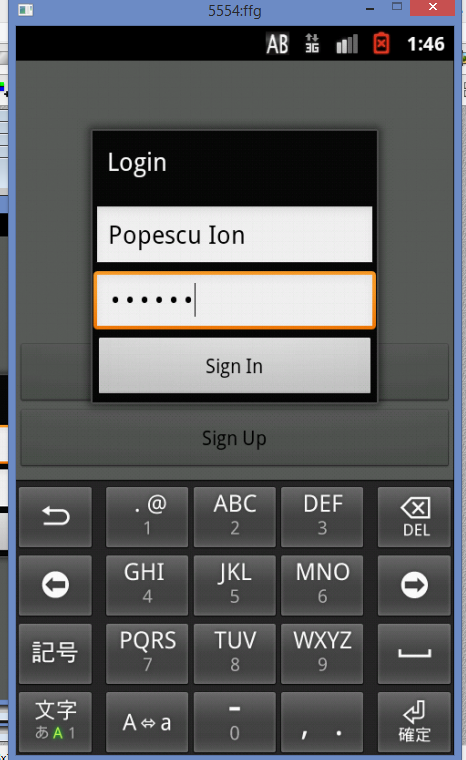


The new user will be introduced in  *login.db* file

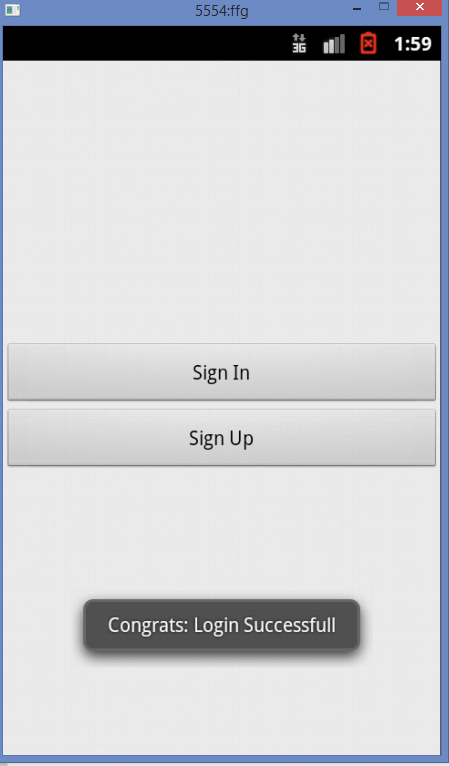
Now, we go back to the home screen and we can use the new user to Sign In.



We introduce the corresponding User Name and Password and press ‘Sign In’ button

.

If the data are correct will appear the following message:



If the data are not correct there will appear an error message.



Listview Application

**In this android application we are creating a simple listview to display an array values.** Android ListView is a view which groups several items and display them in vertical scrollable list. The list items are automatically inserted to the list using an Adapter that pulls content from a source such as an array or database.

An adapter actually bridges between UI components and the data source that fill data into UI Component. Adapter can be used to supply the data to like spinner, list view etc.

The ListView is a subclass of AdapterView and it can be populated by binding them to an Adapter, which retrieves data from an external source and creates a View that represents each data entry. Android provides several subclasses of Adapter that are useful for retrieving different kinds of data and building views for an AdapterView ( ie. ListView). The most common adapter is *ArrayAdapter*.

Steps:

1. **activity\_list\_view\_android\_example.xml:**

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

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    android:orientation="vertical" >

</LinearLayout>

We have to Define ListView in xml file, so the complete .xml file will look like this:

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

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    android:orientation="vertical" >

**<ListView**

**android:id="@+id/list"**

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

**android:layout\_width="match\_parent">**

**</ListView>**

</LinearLayout>

**2.ListViewAndroidExample.java File**

public class ListViewAndroidExample extends Activity{

ListView **listView**;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.**activity\_list\_view\_android\_example**);

3.We have to **Get ListView object from xml:**

**listView**= (ListView) findViewById(R.id.list);

4. **We have to defined Array values to show in ListView:**

String[] **values**= new String[] { "Android List View",

                                             "Adapter implementation",

                                             "Simple List View In Android",

                                             "Create List View Android",

                                             "Android Example",

                                             "List View Source Code",

                                             "List View Array Adapter",

                                             "Android Example List View"

                                            };

**5.We have to Define a new Adapter:**

**ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,**

**android.R.layout.simple\_list\_item\_1, android.R.id.text1, values);**

**6.** **Assign adapter to ListView:**

**listView**.setAdapter(**adapter**);

7. **We have to make the ListView Item Click Listener:**

  listView.**setOnItemClickListener**(new OnItemClickListener() {

                  @Override

                  public void **onItemClick**(AdapterView<?> parent, View view,

                     int position, long id) {

**8.We have to make  the ListView Clicked item index and the ListView Clicked item value:**

int **itemPosition**= position;

String  **itemValue**= (String) listView.getItemAtPosition(position);

Finally,  **the ListViewAndroidExample.java file will look like this:**

 public class ListViewAndroidExample extends Activity {

        ListView **listView**;

        @Override

        protected void onCreate(Bundle savedInstanceState) {

            super.onCreate(savedInstanceState);

            setContentView(R.layout.**activity\_list\_view\_android\_example**);

**listView**= (ListView) findViewById(R.id.list);

            String[] **values**= new String[] { "Android List View",

                                             "Adapter implementation",

                                             "Simple List View In Android",

                                             "Create List View Android",

                                             "Android Example",

                                             "List View Source Code",

                                             "List View Array Adapter",

                                             "Android Example List View"

                                            };

**ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,**

**android.R.layout.simple\_list\_item\_1, android.R.id.text1, values);**

**listView**.setAdapter(**adapter**);

            listView.**setOnItemClickListener**(new OnItemClickListener() {

                  @Override

                  public void **onItemClick**(AdapterView<?> parent, View view,

                     int position, long id) {

                   int **itemPosition**= position;

                   String  **itemValue**= (String) listView.getItemAtPosition(position);

                    Toast.makeText(getApplicationContext(),

                      "Position :"+**itemPosition**+"  ListItem : " +**itemValue**, Toast.LENGTH\_LONG)

                      .show();

                  }

             });

        }

    }

**Testing**

