Tutorial 6 Manipulate Data

We learned how to query data from SQLite database in the last tutorial. In this tutorial, we will create a check out page and learn how to manipulate data using Android application.

Contents

Background for the Exercise:

• Parameterized SQL commands

Parameterized SQL is a SQL command that includes values that are obtained when the SQL is executed. For example, 'insert into student (stid, stname) values (?, ?)' is a parameterized SQL. The two question marks represent two parameters – student id and student name. These parameters are obtained in real time from the user.

Here is a sample code snippet showing how to use parameterized SQL in android project, based on previous tutorials.

```
  String sql = "insert into
checkout(stid,lbcallnum,coduedate,coreturned) values(?,?,?,'N')";
  String args[]=new String[3];
  //stid
  args[0]="100";
  //callnum
  args[1]="45";
  //duedate
  args[2]="2012-05-05";
  DBOperator.getInstance().execSQL(sql, args);
```

In the above example, we inserted a row in the checkout table indicating that a student (id 100) checked out a book (call number 45) on May 5, 2012.

Exercise

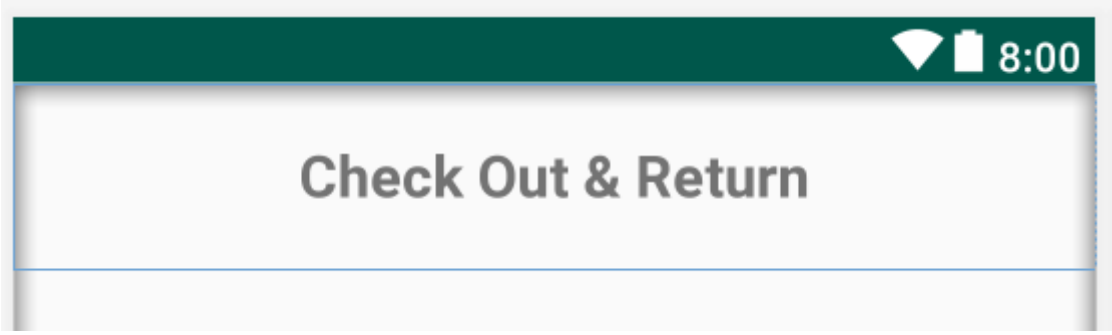
I. Set up Check Out page

1. **Create checkout_yourname.xml file**

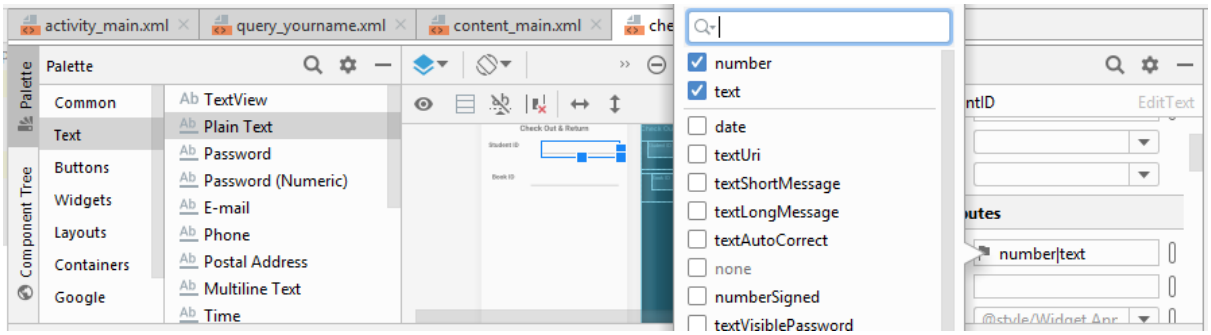
2. **Set up fundamental layout**
   Use **LinearLayout (Vertical)** as fundamental layout. Same format with previous files

3. **Add Title**
   - String: Check Out & Return
     R.string: title_checkout
   - Text size: 22sp
   - Text style: bold
   - Gravity: center
   - Layout margin top: 15dp
   - Layout width: wrap_content

   The UI should look like this:



4. **Set up Student ID section**
   - ✓ Expand **Layouts** and drag **LinearLayout (Horizontal)** to layout editor
     - Layout margin top: 15dp
     - Edit ID: layout_StudentID
   - ✓ Expand **Form Widgets** and drag **TextView** to this new layout
     - String: Student ID
       R.string: studentID
     - Text size: 18sp
     - Text style: bold
     - Layout margin left: 20dp
   - ✓ Expand **Text Fields** and drag **Plain Text** (the first one) to the same layout and put it on the right of "Student ID" TextView



Now, change the properties of the new Plain Text:

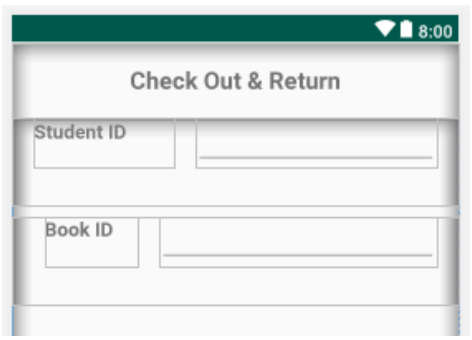
- ID: studentID_edittext
- Layout margin left: 5dp
- Layout margin right: 20dp

5. Do it yourself

Now that you have learned how to set up the Student ID section, it is your turn to set up Book ID section yourself.

- ✓ Linearlayout (Horizontal)
  - Layout margin top: 10dp
  - Edit ID: layout_BookID
- ✓ Text View
  - String: Book ID
    R.string: bookID
  - Text size: 18sp
  - Text style: bold
  - Layout margin left: 30dp
- ✓ Plain Text
  - Layout margin left: 20dp
  - Layout margin right: 20dp
  - ID: bookID_edittext

Now, the UI should look like this:

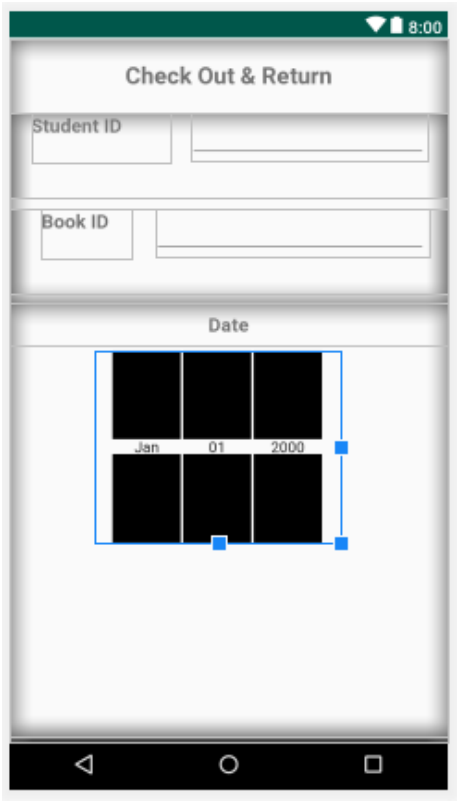


6. Set up date section

- ✓ Expand **Layouts** and drag **LinearLayout (Vertical)** to layout editor
  - Layout margin top: 15dp
  - Edit ID: layout_DatePicker
- ✓ Expand **Text** and drag **TextView** to this new layout
  - String: Date
    R.string: date
  - Text size: 18sp
  - Text style: bold
  - Layout margin left: 30dp
- ✓ Expand **Widgets** and drag **Calendar View** to the same layout and put it under "Date" TextView or copy the following script into the xml file as **DatePicker**. DatePicker takes less space than Calendar View.

```
<DatePicker
        android:id="@+id/datePicker1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="80dp"
        android:layout_marginTop="5dp"
        android:calendarViewShown="false"
        android:datePickerMode="spinner" />
```

The UI should look like this:



Now, change the property of **Calendar View** or **DatePicker**

- Layout gravity: center

7. Do it yourself

Set up other widgets and finish the page, make it like this picture on the right.

- ✓ **LinearLayout (Horizontal)**
  - Layout margin top: 25dp
  - ID: layout_btn
- ✓ **"Check Out" Button**
  - ID: checkout_btn
  - String: Check Out
    R.string: checkout_btn
  - Layout margin left: 40dp
  - Layout width: 100dp
- ✓ **"Return" Button**
  - ID: return_btn
  - String: Return
  - R.string: return_btn
  - Layout margin left: 40dp
  - Layout width: 100dp
- ✓ **"Summary" Page Button**
  - ID: summary_btn
  - String:Summary
  - R.string: summary_btn
  - Layout gravity: center
  - Layout width: 150dp
  - Layout margin top: 5dp
- ✓ **"Back to Main Page" Page Button**

We have created this button in query_yourname.xml page. Here you can just create a button and use the same string R.string: goBack_btn and id goBack_btn.

Tips: You can modify text size and width by yourself if it doesn't look good.

Now that you have implemented the Checkout & Return page, let`s learn how to obtain input values from this interface and store them into database.

II. Manipulate Data

1. Update SQLCommand

- In SQLCommand, add new attribute named CHECK_BOOK and RETURN_BOOK representing a parameterized insertion command into checkout table, as shown below.

```
 public static String RETURN_BOOK = "update checkout set coreturned=?
where stid=? and lbcallnum=?";

public static String CHECK_BOOK = "insert into
checkout(stid,lbcallnum,coduedate,coreturned) values(?,?,?,?)";
```

2. Create CheckoutActivity

- Create a new Class named CheckoutActivity in **library.yourname** package. This activity is associated with checkout/return screen. For convenience, the following is a sample code for CheckoutActivity.java file.

```
package library.yourname;

import java.text.SimpleDateFormat;
import java.util.Calendar;

import library.yourname.constant.SQLCommand;
import library.yourname.util.DBOperator;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.Toast;

public class CheckoutActivity extends Activity implements
OnClickListener {

 EditText stuIdEdit, bookIdEdit;
```

```
DatePicker datePicker;

public void onCreate(Bundle savedInstanceState) {
	super.onCreate(savedInstanceState);
	setContentView(R.layout.checkout_yourname);

stuIdEdit=(EditText)this.findViewById(R.id.studentID_edittext);
	bookIdEdit=(EditText)this.findViewById(R.id.bookID_edittext);
	datePicker=(DatePicker)this.findViewById(R.id.datePicker1);
}

public void onClick(View v)
{
	int id=v.getId();
	if (id==R.id.checkout_btn){
		//Check out a book
		DBOperator.getInstance().execSQL(SQLCommand.CHECK_BOOK,
this.getArgs(true));
		Toast.makeText(getBaseContext(), "Checkout
successfully", Toast.LENGTH_SHORT).show();
	}else if (id==R.id.return_btn){
		//Return a book

DBOperator.getInstance().execSQL(SQLCommand.RETURN_BOOK,
this.getArgs(false));
		Toast.makeText(getBaseContext(), "Return successfully",
Toast.LENGTH_SHORT).show();
	}else if (id==R.id.goBack_btn){
		//Go back to main screen
		Intent intent = new Intent(this,
YourNameActivity.class);
		this.startActivity(intent);
	}
}
/**
 * Get input data
 * including studentID, book callnum, date and returned state
 * @param isCheckout
 * @return
```

```
 */
private String[] getArgs(boolean isCheckout){
        String args[]=new String[4];
        //stid
        args[0] = stuIdEdit.getText().toString();
        //callnum
        args[1] = bookIdEdit.getText().toString();
        //date
        int year=datePicker.getYear();
        int month=datePicker.getMonth();
        int day=datePicker.getDayOfMonth();
        Calendar calendar = Calendar.getInstance();
        calendar.set(year, month, day);
        //format the date
        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-
dd");
        args[2] = dateFormat.format(calendar.getTime());
        if (isCheckout) args[3]="N";
        else args[3]="Y";
        return args;
 }
}
```

3. **Register CheckoutActivity in AnroidManifest.xml**

- Please do not forget to register CheckoutActivity in AndroidManifest.xml. If you do not remember how to do that, please review tutorial 5.

4. **Update YourNameActivity**

- Now we need to create linkage between the main page we created in tutorial 4 and check out page we just created. We want to jump to Check Out page after clicking the checkout button in main page.
- Go to **YourNameActivity.java** and find the code as following.

```
public void onClick(View v)
{
        int id=v.getId();
        if (id==R.id.goCheckOut_btn){

        }else if (id==R.id.goDoQuery_btn){
```

```
                Intent intent = new Intent(this, QueryActivity.class);
                this.startActivity(intent);
        }
}
```

- Insert the following code after "`if (id==R.id.goCheckOut_btn){ `".

```
Intent intent = new Intent(this, CheckoutActivity.class);
this.startActivity(intent);
```

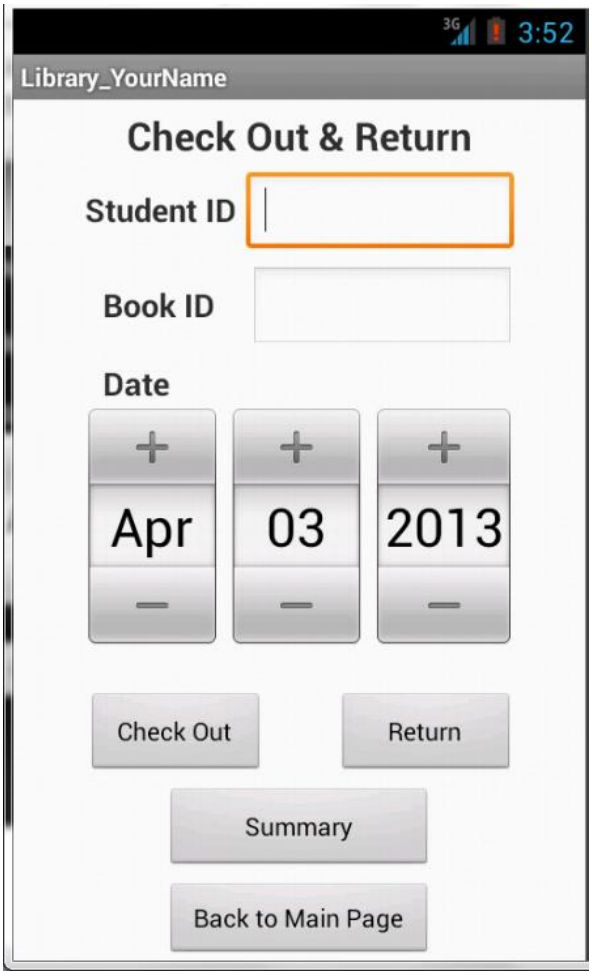
> ***Note:*** I want to point out that you should validate input data in checkout/return screen. For example, if the input student ID is '1001', it will be longer than the length of stid column in student table, so it is not a valid ID.

What to Submit

One Screenshots

Please capture the screenshots of your completed Check Out & Return page and paste them into a Word document.

All deliverables should be submitted via the Canvas assignment manager as a single Word or PDF document by the due date.



Additional Examples (Optional)

(These instructions have not been updated to Android Tutorial 3.5)

If you are interested in learning how to create **Drop Down Lists** using the **Spinner** component, here is more information you might find helpful.

What you can learn from this example:

- Setup Spinner component
- Get the dataset from the database and show the item list on the Spinner
- Insert items into or delete items out from the Spinner dynamically

Steps:

1. Create new project, name it : UIExample03

2. In res/values/strings.xml, input:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">UIexample03</string>
    <string name="action_settings">Settings</string>
    <string name="hello_world">Hello world!</string>
    <string name="add_item">Add</string>
    <string name="delete_item">Delete</string>
</resources>
```

3. In res/layout/activity_main.xml, input：

```
<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:orientation="vertical">

    <TextView
              android:id="@+id/spinner_selection"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:textIsSelectable="true"/>

    <Spinner
        android:id="@+id/spinner"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <EditText
        android:id="@+id/new_item"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```

```
            android:inputType="text"/>

        <Button
            android:id="@+id/add_button"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/add_item"
            android:onClick="addNewItem"/>

         <Button
            android:id="@+id/delete_button"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/delete_item"
            android:onClick="deleteItem"/>
    </LinearLayout>
```

4. In src/com.example.uiexample03/MainActivity.java, input these:

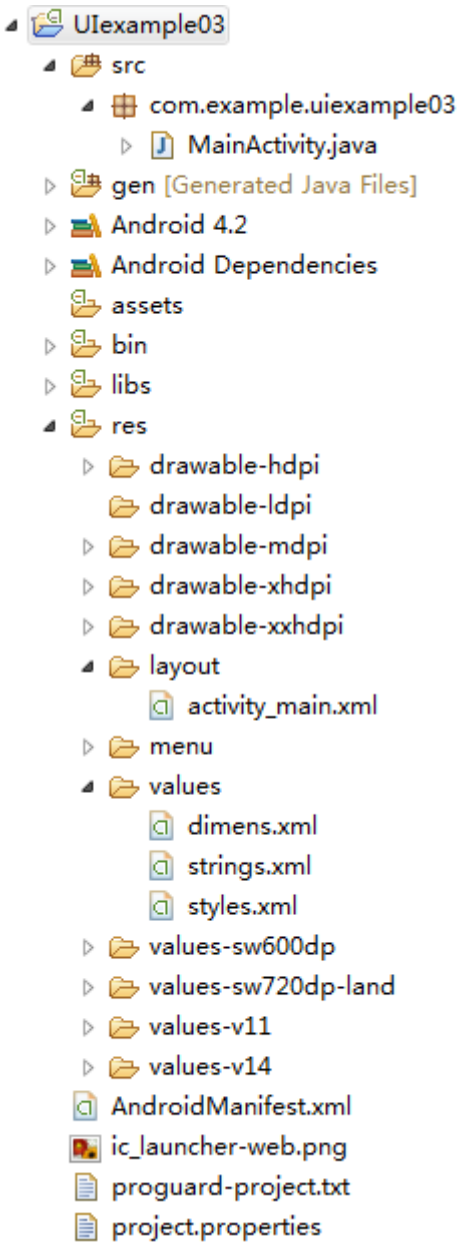
```
    package com.example.uiexample03;

    import java.util.ArrayList;
    import android.os.Bundle;
    import android.app.Activity;
    import android.app.AlertDialog;
    import android.app.AlertDialog.Builder;
    import android.content.DialogInterface;
    import android.view.View;
    import android.widget.AdapterView;
    import android.widget.AdapterView.OnItemSelectedListener;
    import android.widget.ArrayAdapter;
    import android.widget.EditText;
    import android.widget.Spinner;
    import android.widget.TextView;
    import android.widget.Toast;

    public class MainActivity extends Activity {
        private TextView selectedTextView;
        private EditText newItemEditText;
        private Spinner spinner;
        private ArrayAdapter<String> arrayAdapter;

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

            initialize();
        }

        private void initialize() {
            selectedTextView = (TextView)
    this.findViewById(R.id.spinner_selection);
            newItemEditText = (EditText) findViewById(R.id.new_item);
```

```
                spinner = (Spinner) this.findViewById(R.id.spinner);

                // set the items showed in Spinner, items should be put into
        List or
                // ArrayList
                ArrayList<String> systemList = new ArrayList<String>();
                systemList.add("Android");
                systemList.add("iOS");
                systemList.add("WindowsMobile");
                systemList.add("Palm");
                // set items list and textview style of items in ArrayAdapter
                arrayAdapter = new ArrayAdapter<String>(this,
                                android.R.layout.simple_spinner_item, systemList);
                // set the style of drop down views
                arrayAdapter

                .setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
                // set the ArrayAdapter into Spinner
                spinner.setAdapter(arrayAdapter);

                // set OnItemSelectedListner to Spinner
                spinner.setOnItemSelectedListener(new OnItemSelectedListener() {
                        @Override
                        public void onItemSelected(AdapterView<?> arg0, View
        arg1,
                                        int arg2, long arg3) {

                selectedTextView.setText(arrayAdapter.getItem(arg2));
                        }

                        @Override
                        public void onNothingSelected(AdapterView<?> arg0) {
                        }
                });
        }

        public void addNewItem(View view) {
                if (newItemEditText.getText().toString().equals("")) {
                        Toast.makeText(getApplicationContext(),
                                        "Please insert content into EditText!",
        Toast.LENGTH_LONG)
                                        .show();
                } else {
                        dialog("Are you sure to add a new item?", view);
                }
        }

        public void deleteItem(View view) {
                if (arrayAdapter.isEmpty()) {
                        Toast.makeText(getApplicationContext(),
                                        "There is no items to delete!",
        Toast.LENGTH_LONG).show();
                } else {
                        dialog("Are you sure to delete the selected item?",
        view);
                }
        }
```

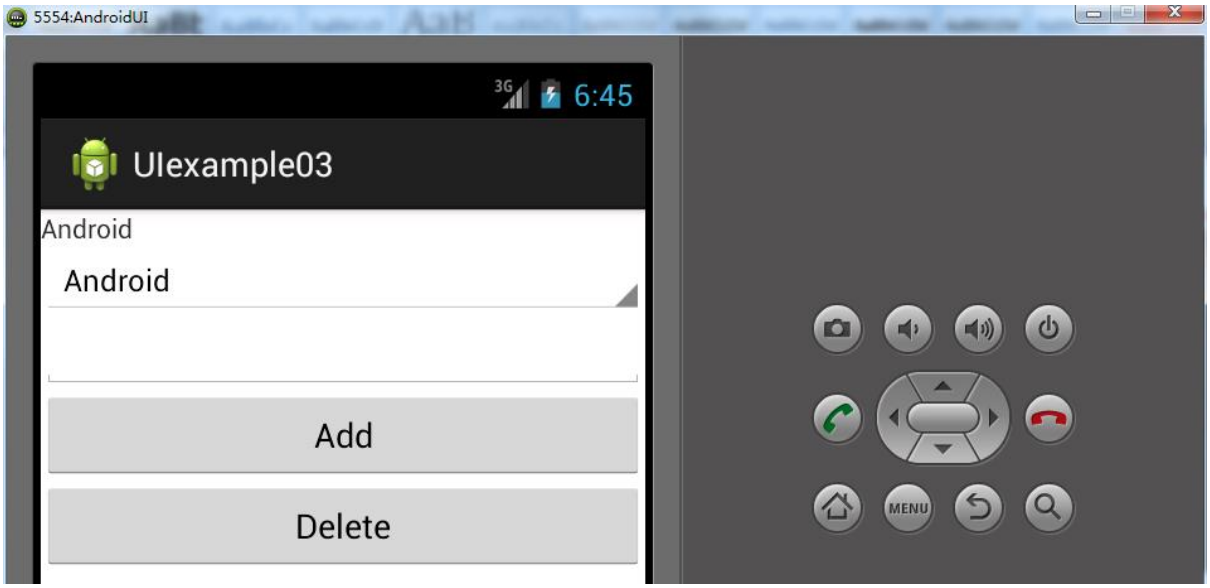
```
        // Confirmation operation
        public void dialog(String message, final View view) {
                AlertDialog.Builder builder = new Builder(MainActivity.this);
                builder.setMessage(message);
                builder.setTitle("Confirmation");
                builder.setPositiveButton("OK",
                                new
android.content.DialogInterface.OnClickListener() {
                                        public void onClick(DialogInterface
dialog, int which) {
                                                dialog.dismiss();
                                                switch (view.getId()) {
                                                case R.id.add_button:

                arrayAdapter.add(newItemEditText.getText()
                                                                        .toString());

                spinner.setSelection(arrayAdapter

                .getPosition(newItemEditText.getText()

                .toString()));
                                                        newItemEditText.setText("");

                Toast.makeText(getApplicationContext(),
                                                                        "A new item
has been added!",

                Toast.LENGTH_LONG).show();
                                                        break;
                                                case R.id.delete_button:

                arrayAdapter.remove(spinner.getSelectedItem()
                                                                        .toString());
                                                        if (!arrayAdapter.isEmpty())
{

                selectedTextView.setText("");
                                                        }

                Toast.makeText(getApplicationContext(),
                                                                        "The item has
been deleted!",

                Toast.LENGTH_LONG).show();
                                                        break;
                                                default:
                                                        break;
                                                }
                                        }
                                });
                builder.setNegativeButton("Cancel",
                                new
android.content.DialogInterface.OnClickListener() {
                                        public void onClick(DialogInterface
dialog, int which) {
```

```
                                                dialog.dismiss();
                                            }
                                        });
                        builder.create().show();
                }
        }
```

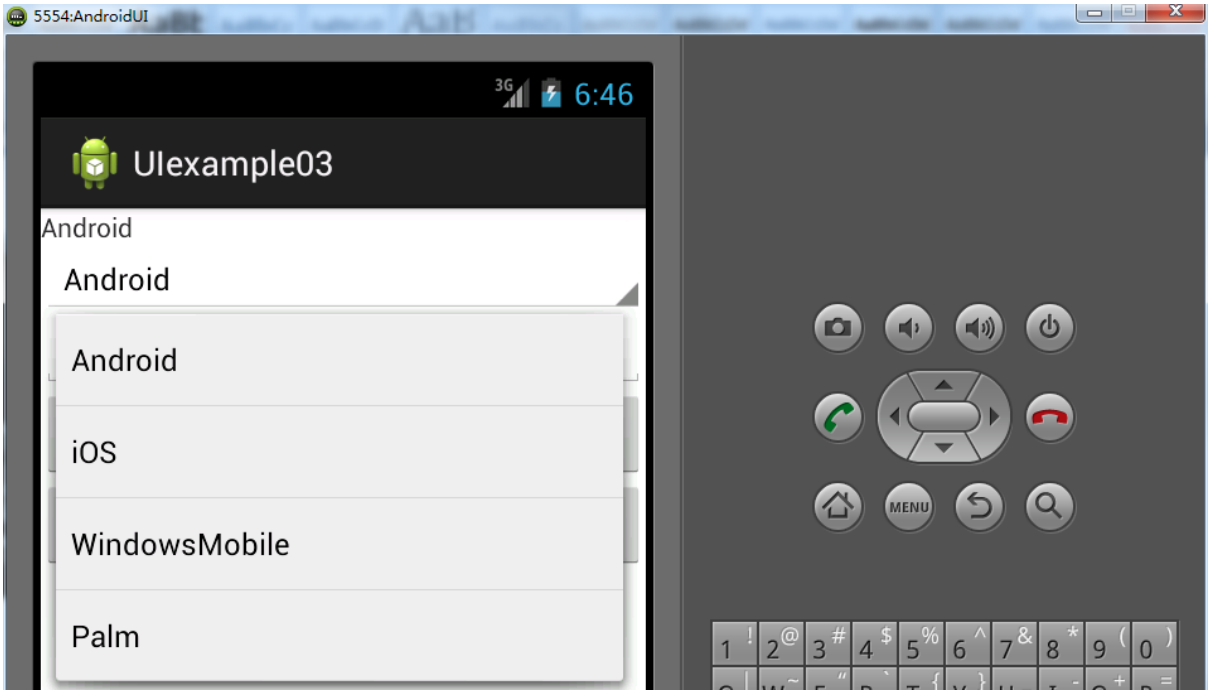
5. At the end of project, the index of the whole project should as follow:



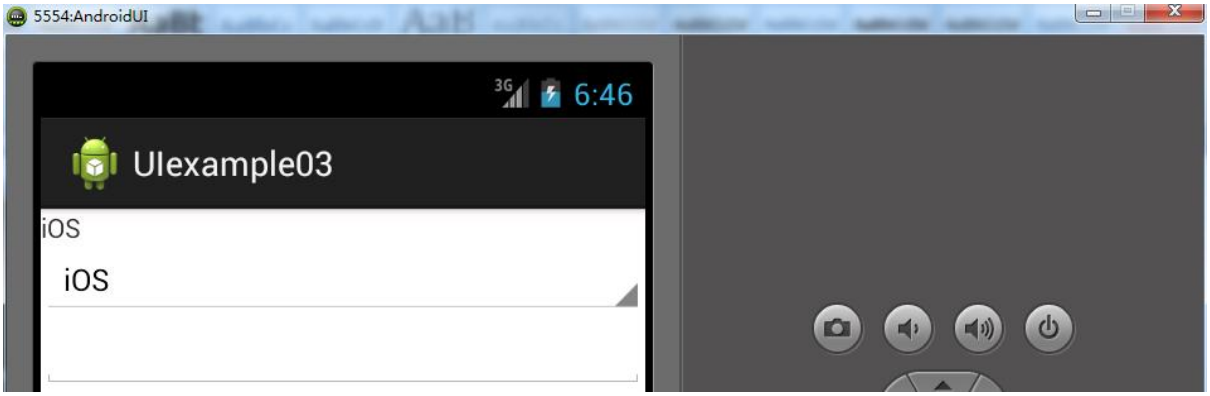
6. The running result will be：



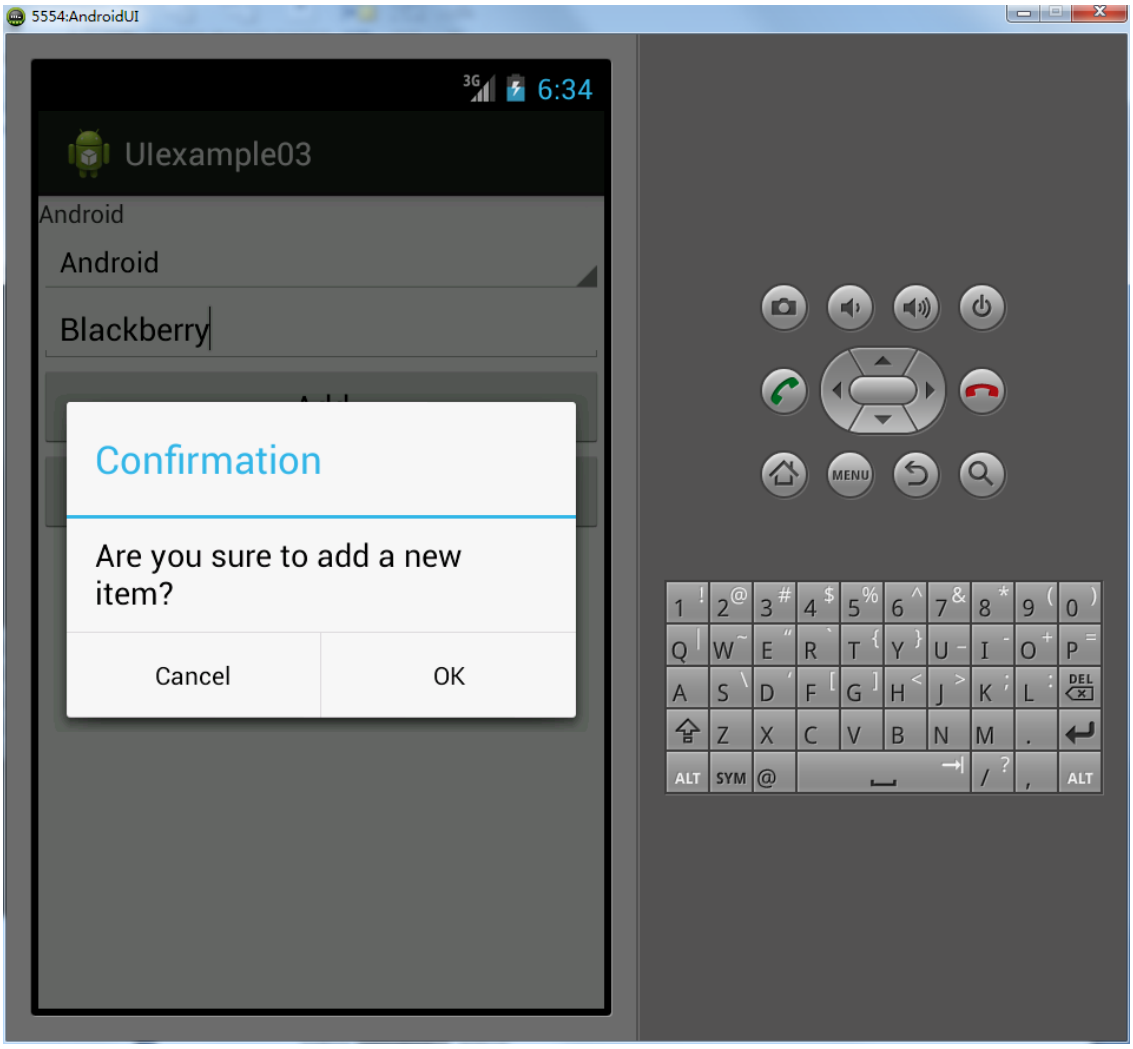
Click the Spinner to get the drop box and select one item (iOS):



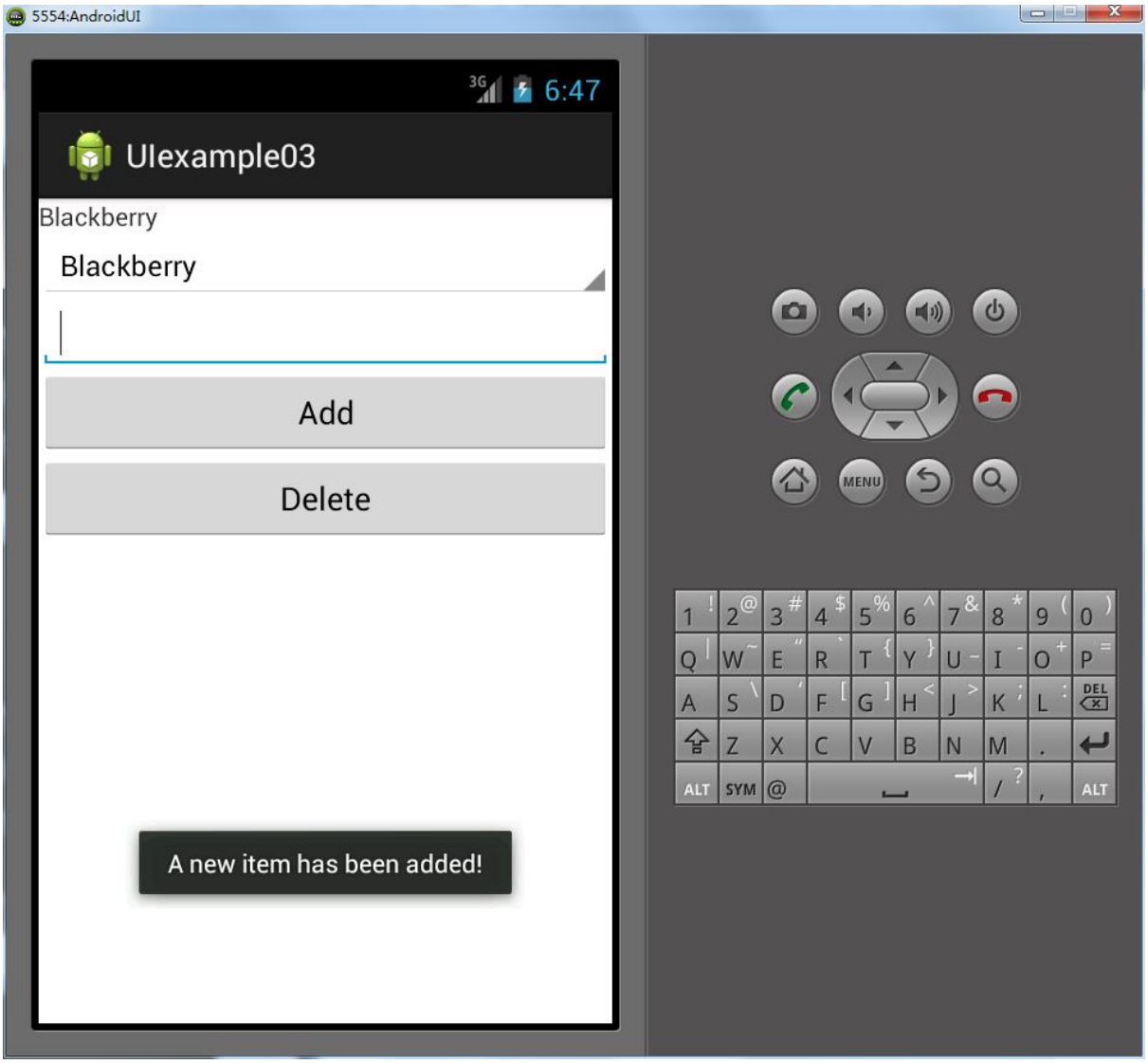
The item in the spinner and TextView is iOS:



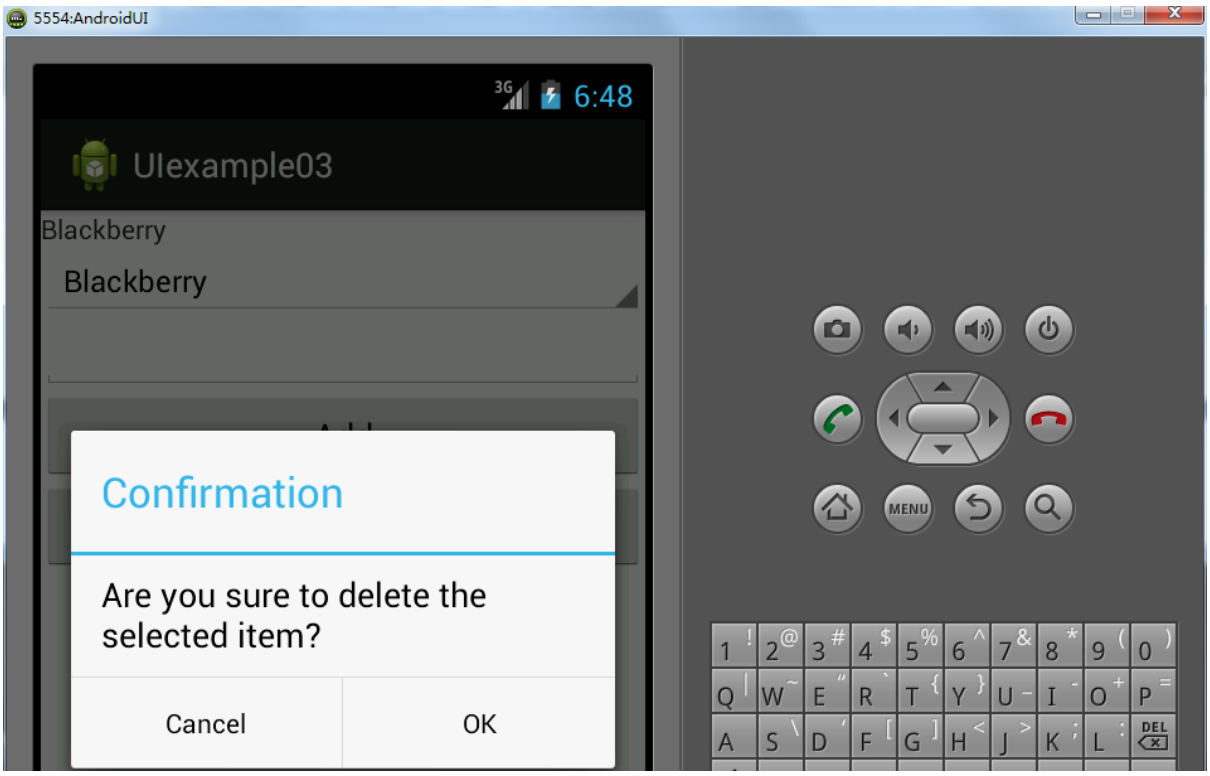
Input Blackberry in the TextView, then click the Add button:



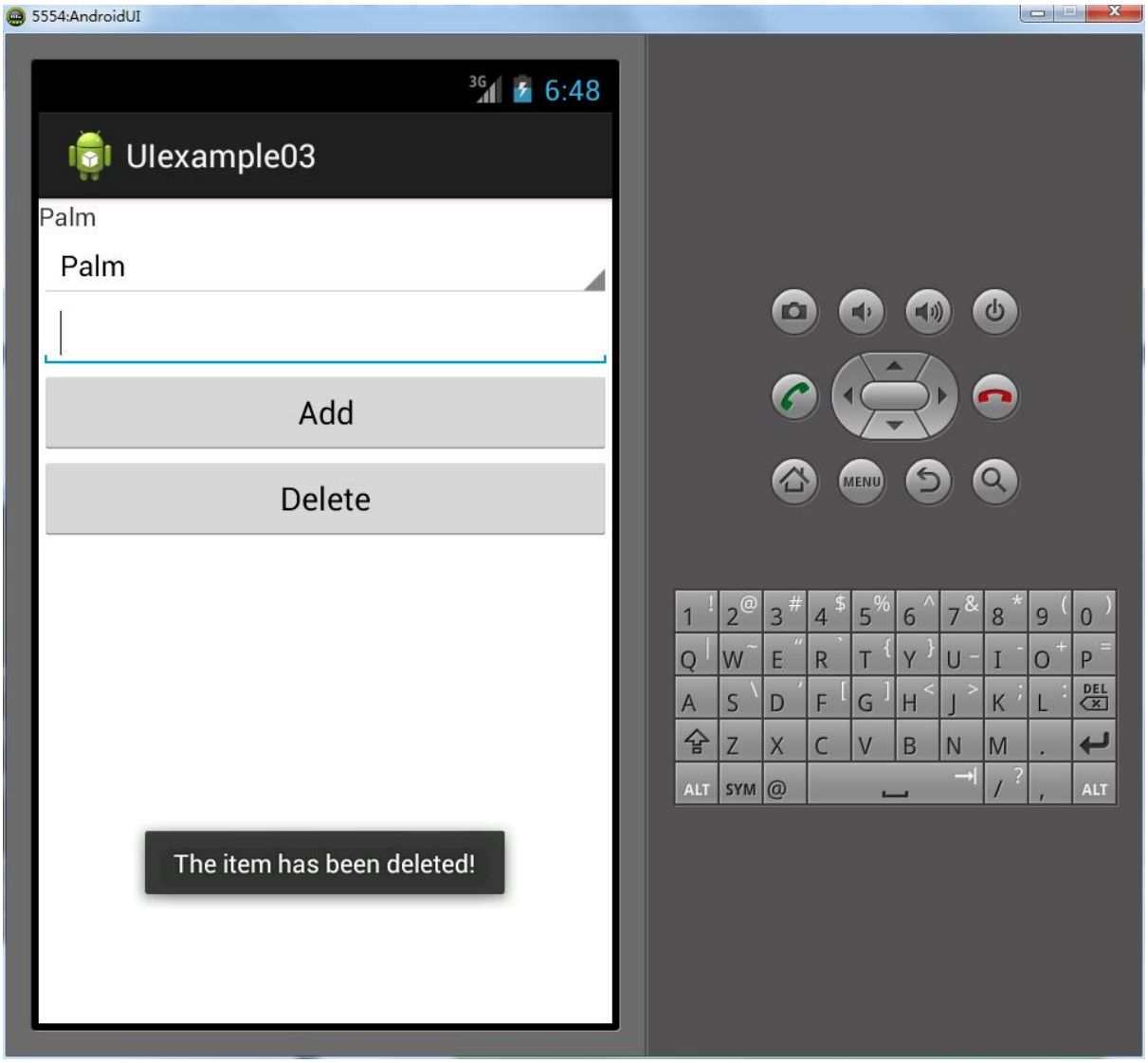
Then Click OK:



Then click Delete button:


5554:AndroidUI
6:48
UIexample03
Blackberry
Blackberry
Confirmation
Are you sure to delete the selected item?
Cancel
OK
MENU

Click OK:



Explanation:

The code to set the data (List or ArrayList) into ArrayAdapter, then put ArrayAdapter to Spinner:

```
        // set items list and textview style of items in ArrayAdapter
        ArrayAdapter<String> arrayAdapter = new ArrayAdapter<String>(this,
        android.R.layout.simple_spinner_item, systemList);
        // set the style of drop down views
        arrayAdapter.setDropDownViewResource(android.R.layout.simple_spinner_drop
        down_item);
        // set the ArrayAdapter into Spinner
        spinner.setAdapter(arrayAdapter);
```

The code to add new item into Spinner:

```
        arrayAdapter.add("your new items");
```

To remove an item from Spinner, Parameter here means the item content you want to remove:

```
arrayAdapter.remove(spinner.getSelectedItem().toString());
```

Show Alert Dialog:

```
AlertDialog.Builder builder = new Builder(MainActivity.this);
builder.setMessage("Are you sure you want to delete an item?");
builder.setTitle("Confirmation");
builder.setPositiveButton("OK",
   new android.content.DialogInterface.OnClickListener() {
              public void onClick(DialogInterface dialog, int which) {
                           dialog.dismiss();
                           //Your Code...
              }
       });
builder.setNegativeButton("Cancel",
       new android.content.DialogInterface.OnClickListener() {
              public void onClick(DialogInterface dialog, int which) {
                           dialog.dismiss();
                     //Your Code...
              }
       });
builder.create().show();
```

Set Listener on Spinner, when one item has been selected, to do something:

```
// set OnItemSelectedListner to Spinner
spinner.setOnItemSelectedListener(new OnItemSelectedListener() {
       @Override
       public void onItemSelected(AdapterView<?> arg0, View arg1,
                                  int arg2, long arg3) {
              //Your Code...
       }

       @Override
       public void onNothingSelected(AdapterView<?> arg0) {
         //Your Code...
       }
});
```

Use Toast to show information:

```
// parameters: context; message; Toast.LENGTH_LONG or
Toast.LENGTH_SHORT
Toast.makeText(getApplicationContext(),"Please insert content into
EditText!", Toast.LENGTH_LONG).show();
```