Tutorial 5 Multi-Screen Application

We improved our project by adding main and query pages in last tutorial. In this tutorial, you will learn how to set up an Android application with multiple screens and how to navigate between these screens.

Contents

Background for the Exercise:		2	
		3	
I.	Update library project	3	
II.	Register QueryActivity in AndroidManifest.xml	6	
What to Submit		9	
Additional Examples (Optional)		11	

4 Background for the Exercise:

√ Event-handling

Via Graphical User Interface, we can see many *components* such as buttons and input boxes. These components enable us to interact with software. Each interaction is an *event* for a computer. For example, clicking a button is an event.

Typically, we create one activity for each screen in Android application. An activity takes care of showing a screen and responding to events, such as button clicks, on the screen.

To create a multi-screen android application, you have to understand:

- How to write event-handling routines.
- How to navigate from one screen to another.

✓ Borrowing from prior Tutorials

I chose to create a new Android project named Tutorial5yourname. Change yourname to your name for uniqueness.

- From Tutorial 4:
 - In the new project, you will need to to add query_yourname.xml under layout from Tutorial 4 as well as copy all of activity_main.xml and content_main.xml from Tutorial 4 under layout. Add arrays.xml under values and copy all of arrays.xml from Tutorial 4 as well as all of strings.xml to the corresponding xml files under values in your Tutorial 5.
- From Tutorial 3:
 - In the new project, you will need to create java packages command, view, and util and create a java classes named SQLCommand, DBConstant, DBOperator, DBOpenHelper, . After creating the java classes from Tutorial 3, copy the java code from each java class in Tutorial 3 to the corresponding java class in Tutorial 5.
 - *** Be sure to correct the package code line in each java class from Tutorial 3 to the correct directory listing (i.e., package) in Tutorial 5. ***

Exercise

I. Update library project

Modify the result-displaying screen (query_yourname.xml)

In the result-displaying screen, each entry you added in Spinner corresponds to a specific SQL query in the library database. For each query, add a new attribute in SQLCommand.

Notice: there are only two queries in this example, so you must complete it by adding other queries.

A sample SQLCommand class is shown below. The highlighted text is from Tutorial 3.

```
package com.example.tutorial5yourname.constant;
public abstract class SQLCommand
* SQL commands
* Including select/delete/update/insert
*/
  //query all students
 public static String QUERY_STUDENT = "select stid, stname from Student";
  //list all data in books table
  public static String QUERY_1 = "select lbcallnum, lbtitle from libbook";
  //List the call numbers of books with the title 'Database Management'
  public static String QUERY 2 = "select lbcallnum from libbook where lbtitle like
'%Database Management%'";
  public static String QUERY 3 = "";
  public static String QUERY_4 = "";
  public static String QUERY 5 = "";
  public static String QUERY 6 = "";
  public static String QUERY 7 = "";
```

2. Modify the code

✓ Create a new class named QueryActivity in package com.example.Tutorial5yourname. In this class, add event-handling code for 'Show Result' button. The following is the sample code of QueryActivity.

```
package com.example.tutorial5yourname;
import com.example.tutorial5yourname.constant.SQLCommand;
import com.example.tutorial5yourname.util.DBOperator;
import com.example.tutorial5yourname.view.TableView;
import android.app.Activity;
import android.content.Intent;
import android.database.Cursor;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.ScrollView;
import android.widget.Spinner;
import android.widget.Toast;
public class QueryActivity extends Activity implements OnClickListener {
Button backBtn,resultBtn;
Spinner querySpinner;
ScrollView scrollView;
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.query_yourname);
    //copy database file
    try{
        DBOperator.copyDB(getBaseContext());
    }catch(Exception e){
        e.printStackTrace();
    }
        backBtn=(Button)this.findViewById(R.id.goBack btn);
        backBtn.setOnClickListener(this);
        resultBtn=(Button)this.findViewById(R.id.showresult btn);
        resultBtn.setOnClickListener(this);
        querySpinner=(Spinner)this.findViewById(R.id.queryList_spinner);
scrollView=(ScrollView)this.findViewById(R.id.scrollview queryresults);
    public void onClick(View v)
{
       String sql="";
```

```
int id=v.getId();
       if (id==R.id.showresult btn){
              //show query result
              int pos=querySpinner.getSelectedItemPosition();
              if (pos==Spinner.INVALID_POSITION){
                     //User doesn't choose any query, show warning
                    Toast.makeText(this.getBaseContext(), "Please choose
a query!", Toast.LENGTH_SHORT).show();
                     return;
              scrollView.removeAllViews();
              if (pos==0){
                    //show all books
                     sql=SQLCommand.QUERY_1;
              }else if (pos==1){
                     //list the call numbers of books with the title
'Database Management'
                     sql=SQLCommand.QUERY_2;
              }else if (pos==2){
                     sql=SQLCommand.QUERY_3;
              }else if (pos==3){
                     sql=SQLCommand. QUERY 4;
              }else if (pos==4){
                    sql=SQLCommand. QUERY 5;
              }else if (pos==5){
                    sql=SQLCommand. QUERY 6;
              }else if (pos==6){
                    sql=SQLCommand. QUERY_7;
              Cursor cursor=DBOperator.getInstance().execQuery(sql);
              scrollView.addView(new
TableView(this.getBaseContext(),cursor));
       }else if (id==R.id.goBack_btn){
              //go back to main screen
              Intent intent = new Intent(this, YourNameActivity.class);
              this.startActivity(intent);
       }
}
```

✓ Create a java class named YourNameActivity in package com.example.Tutoril5yourname and add event-handling code for the 'Do Query' button. For convenience, the code of this class is provided.

```
package com.exmaple.tutorial5yourname;
import com.example.tutorial5yourname.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.Button;
public class YourNameActivity extends Activity implements
OnClickListener{
Button checkoutBtn, queryBtn;
/** Called when the activity is first created. */
@Override
public void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       setContentView(R.layout.activity_main);
       checkoutBtn=(Button)this.findViewById(R.id.goCheckOut_btn);
       checkoutBtn.setOnClickListener(this);
       queryBtn=(Button)this.findViewById(R.id.goDoQuery_btn);
       queryBtn.setOnClickListener(this);
                          //copy database file
       try{
              DBOperator.copyDB(getBaseContext());
       }catch(Exception e){
              e.printStackTrace();
       }
}
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);
       }
}
```

II. Register QueryActivity in AndroidManifest.xml

AndroidManifest.xml is a configuration file in your Android project. It includes information of all activities in the project.

To register an activity:

a) Modify the code in AndroidManifest.xml.

✓ Double click AndroidManifest.xml

```
■ Project ▼
                                              🕀 🛨 💠 — 💿 YourNameActivity.java × 📠 AndroidManifest.xml ×
                                                                                                              © DBOperator.java × © TableView.java ×
                                                                      <?xml version="1.0" encoding="utf-8"?>
                 MainActivity
                                                                      <manifest xmlns:android="http://schemas.android.com/apk/res/android"</pre>

    QueryActivity

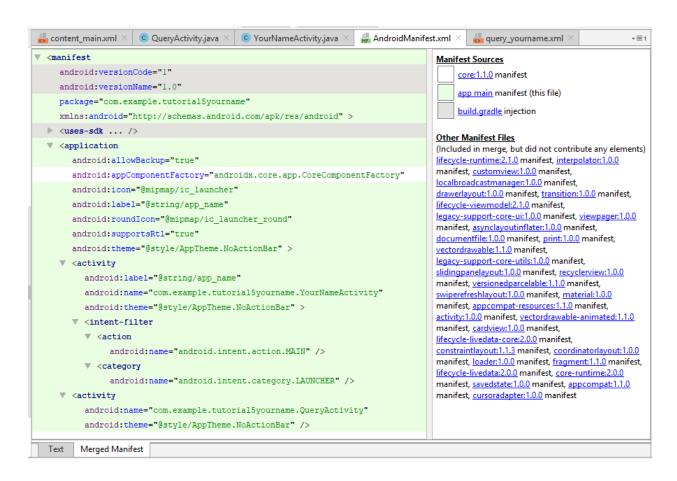
                                                                          package="com.example.tutorial5yourname">
                 YourNameActivity
         ▼ III res
                                                                           <application
           drawable
                                                                              android:allowBackup="true"
            ▶ 🖿 drawable-v24
                                                                              android:icon="@mipmap/ic launcher"
                                                                              android:label="Tutorial 5 yourname"
           ▶ layout
                                                                              android:roundIcon="@mipmap/ic launcher round"
            ▶ ■ menu
                                                                              android:supportsRtl="true"
            ▶ I mipmap-anydpi-v26
                                                                              android:theme="@style/AppTheme">
            mipmap-hdpi
                                                                              <activity
            mipmap-mdpi
                                                                                  android:name=".MainActivity"
            ▶ ☐ mipmap-xhdpi
                                                                                  android:label="Tutorial 5 yourname"
                                                                                  android:theme="@style/AppTheme.NoActionBar">
            ▶ 🖿 mipmap-xxhdpi
                                                                                  <intent-filter>
            ▶ ☐ mipmap-xxxhdpi
                                                                                      <action android:name="android.intent.action.MAIN" />
           ▶ ■ values
                                                                                      <category android:name="android.intent.category.LAUNCHER" />
         test
                                                                                  c/intent-filter
                                                                       manifest > application > activity > intent-filter > action
       gitignore.
      app.iml
                                                                Text Merged Manifest
```

✓ In AndroidManifest.xml file, there are two tabs: Text and Merged Manifest.

Choose the text tab and adjust the code to the following:

```
✓ <?xml version="1.0" encoding="utf-8"?>
   <manifest xmlns:android="http://schemas.android.com/apk/res/android"</pre>
       xmlns:tools="http://schemas.android.com/tools"
       package="com.example.tutorial5yourname">
       <application
           android:allowBackup="true"
           android:icon="@mipmap/ic launcher"
           android:label="@string/app name"
           android:roundIcon="@mipmap/ic launcher round"
           android:supportsRtl="true"
           android: theme="@style/AppTheme.NoActionBar">
           <activity
               android: name=".YourNameActivity"
               android:label="@string/app name"
               android: theme="@style/AppTheme.NoActionBar">
                <intent-filter>
                    <action android:name="android.intent.action.MAIN" />
                    <category android:name="android.intent.category.LAUNCHER" />
               </intent-filter>
           </activity>
           <activity android:name=".QueryActivity"</pre>
                      android:theme="@style/AppTheme.NoActionBar"></activity>
       </application>
   </manifest>
```

The manifest tab will look like this:



What to Submit

1. Two Screenshots

Please capture the screenshots of your completed Query page and paste them into a Word document.

The screenshots should look like:





2. Partial Code

Please copy the code of **SQLCommand.java** file and paste it into the same Word document.

```
main.xml × 💿 QueryActivity.java × 💿 YourNameActivity.java × 👬 AndroidManifest.xml × 👼 query_yourname.xml ×
                                                                                                       © SQLCommand.java ×+
        package com.example.tutorial5yourname.constant;
         * SQL commands
 5
         * Including select/delete/update/insert
 6
 7
        public abstract class SQLCommand
 9
            //query all students
            public static String QUERY_STUDENT = "select stid, stname from Student";
            //list all data in books table
            public static String QUERY_1 = "select lbcallnum, lbtitle from Libbook";
13
            //List the call numbers of books with the title 'Database Management'
14
15
            public static String QUERY 2 = "select lbcallnum from libbook where lbtitle like '%Database Management%'";
            public static String QUERY_3 = "select * from Student";
16
            public static String QUERY_4 = "select * from Libbook";
18
            public static String QUERY 5 = "select * from Student";
            public static String QUERY_6 = "select * from Libbook";
19
            public static String QUERY 7 = "select * from Libbook";
21
24
         SQLCommand
```

All deliverables should be submitted via the Canvas assignment file upload as a <u>single</u> Word or PDF document by the due date.

Additional Examples (Optional)

If you are interested in learning how to set up **Radio Button** and **Check Button** with **conditional jumping** from one screen to another, here is more information you might find helpful.

What you can learn from this example:

- Remove title bar and status bar of the application
- Setup radio and check button
- Conditional jumping from one screen to another

Steps:

***** These screens have not been updated to Android Studio 3.5 ***********

- 1. Create a new project, name it: UIExample02
- 2. Add the following codes into res/values/strings.xml:

```
<resources>
   <string name="app name">UIexample02</string>
   <string name="hello_world">Hello world!</string>
   <string name="menu_settings">Settings</string>
   <string name="title_activity_main">MainActivity</string>
<!-- first layout -->
   <string name="genber_title">Gender:</string>
   <string name="genber_female">Female</string>
   <string name="genber male">Male</string>
   <string name="interest_title">Interest:</string>
   <string name="interest_movie">Movie</string>
   <string name="interest_basketball">Basketball</string>
   <string name="interest music">Music</string>
   <string name="interest it">IT</string>
   <string name="button_submit">Submit</string>
   <!-- second layout -->
   <string name="layout2_label">Second Layout</string>
   <string name="gender title2">Your Gender:</string>
   <string name="interest_title2">Your Interest:</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"</pre>
```

```
android:layout_height="match_parent"
android:orientation="vertical" >
<TextView
    android:layout width="wrap content"
    android:layout height="wrap content"
    android:text="@string/genber title" />
<RadioGroup
    android:id="@+id/radiogroup_gender"
    android:layout width="wrap content"
    android:layout height="wrap content" >
    <RadioButton
        android:id="@+id/radio_female"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/genber female" />
    <RadioButton
        android:id="@+id/radio_male"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/genber male" />
</RadioGroup>
<TextView
    android:layout width="wrap content"
    android:layout height="wrap content"
    android:text="@string/interest_title" />
<CheckBox
    android:id="@+id/check movie"
    android:layout width="wrap content"
    android:layout_height="wrap_content"
    android:text="@string/interest_movie" />
<CheckBox
    android:id="@+id/check basketball"
    android:layout width="wrap content"
    android:layout_height="wrap_content"
    android:text="@string/interest_basketball" />
<CheckBox
    android:id="@+id/check music"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/interest movie" />
<CheckBox
    android:id="@+id/check_it"
    android:layout width="wrap content"
    android:layout_height="wrap_content"
    android:text="@string/interest it" />
<Button
    android:id="@+id/button_submit"
    android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
android:layout_gravity="center"
android:onClick="submit"
android:text="@string/button_submit" />
</LinearLayout>
```

4. in res/layout/second layout.xml, input:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"</pre>
    android:layout width="match parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout height="wrap content"
        android:text="@string/gender_title2" />
    <TextView
        android:id="@+id/textview_gender"
        android:layout width="wrap content"
        android:layout_height="wrap_content"/>
    <TextView
        android:id="@+id/textView4"
        android:layout width="wrap content"
        android:layout_height="wrap_content"
        android:text="@string/interest_title2" />
    <TextView
        android:id="@+id/textview_interest"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
       </LinearLayout>
```

5. in src/com.example.uiexample02/MainActivity.java, input these:

```
package com.example.uiexample02;

import java.util.ArrayList;
import java.util.List;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.View;
import android.view.Window;
```

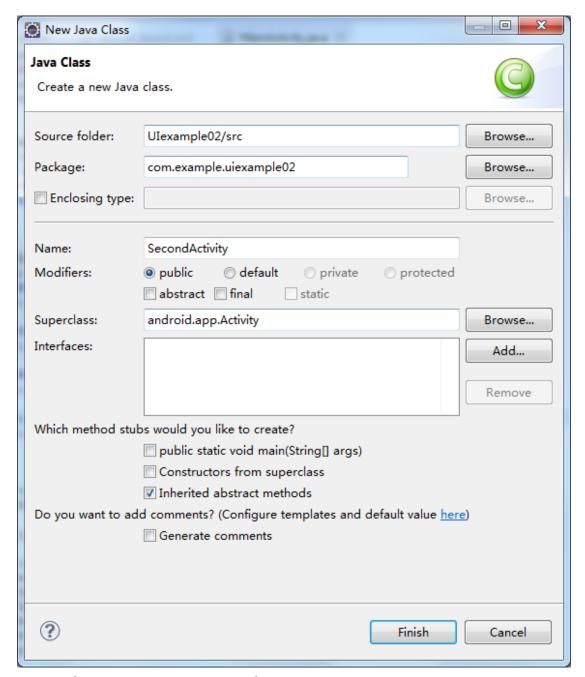
```
import android.view.WindowManager;
import android.widget.CheckBox;
import android.widget.RadioButton;
public class MainActivity extends Activity {
       private RadioButton femaleRadioButton;
       private RadioButton maleRadioButton;
       private CheckBox movieCheckBox;
       private CheckBox basketCheckBox;
       private CheckBox musicCheckBox;
       private CheckBox itCheckBox;
       private String gender;
       private List<String> interests;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //hide title
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        //hide status bar
        int flag=WindowManager.LayoutParams.FLAG FULLSCREEN;
        this.getWindow().setFlags(flag, flag);
        setContentView(R.layout.activity_main);
        initialize();
    }
    private void initialize(){
       femaleRadioButton=(RadioButton) findViewById(R.id.radio_female);
       maleRadioButton=(RadioButton) findViewById(R.id.radio male);
       movieCheckBox=(CheckBox) findViewById(R.id.check movie);
       basketCheckBox=(CheckBox) findViewById(R.id.check_basketball);
       musicCheckBox=(CheckBox) findViewById(R.id.check_music);
       itCheckBox=(CheckBox) findViewById(R.id.check_it);
       //default: female
       femaleRadioButton.setChecked(true);
    }
    private void setGender(){
       if(femaleRadioButton.isChecked()){
              gender="Female";
       }else if(maleRadioButton.isChecked()){
              gender="Male";
              }
    }
    private void setInterest(){
       interests=new ArrayList<String>();
       if(movieCheckBox.isChecked()){
              interests.add("Movie");
       if(basketCheckBox.isChecked()){
```

```
interests.add("Basketball");
       if(musicCheckBox.isChecked()){
               interests.add("Music");
       if(itCheckBox.isChecked()){
               interests.add("IT");
    }
    //dealing with the click of the submit button
    public void submit(View view){
       setGender();
       setInterest();
       //set the data which you want to transfer into next activity(layout)
       Bundle bundle=new Bundle();
       bundle.putString("gender", gender);
       bundle.putStringArrayList("interest", (ArrayList<String>) interests);
       Intent intent=new Intent(MainActivity.this,SecondActivity.class);
       intent.putExtras(bundle);
       startActivity(intent);
    }
}
```

6. Right click on package com.example.uiexample02 --> New --> Class

Name: SecondActivity;

Superclass: android.app.Activity;



7. In src/com.example.uiexample02/SecondActivity.java, input:

```
package com.example.uiexample02;
import java.util.List;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.widget.TextView;

public class SecondActivity extends Activity {
```

```
private TextView genderTextView;
private TextView interestTextView;
@Override
protected void onCreate(Bundle savedInstanceState) {
       // TODO Auto-generated method stub
       super.onCreate(savedInstanceState);
       setContentView(R.layout.second_Layout);
       initialize();
       setResult();
}
private void initialize(){
       genderTextView=(TextView) findViewById(R.id.textview_gender);
       interestTextView=(TextView) findViewById(R.id.textview_interest);
}
private void setResult(){
       //get the data comes from the first layout
       Intent intent=this.getIntent();
       Bundle bundle=intent.getExtras();
       String gender=(String) bundle.get("gender");
       List<String> interests=bundle.getStringArrayList("interest");
       String yourInterest="";
       for(String interest:interests){
              yourInterest=yourInterest+interest+"\n";
       //show results in the textview
       genderTextView.setText(gender);
       interestTextView.setText(yourInterest);
}
```

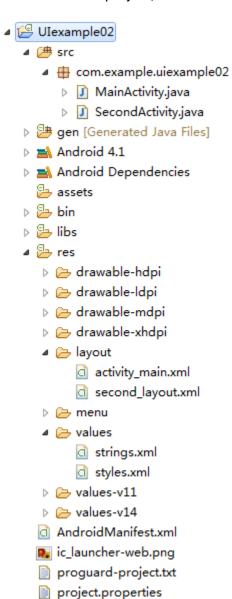
8. To register the new created activity (SecondActivity.java) in the AndroidMenifest.xml, input:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.uiexample02"
    android:versionCode="1"
    android:versionName="1.0" >

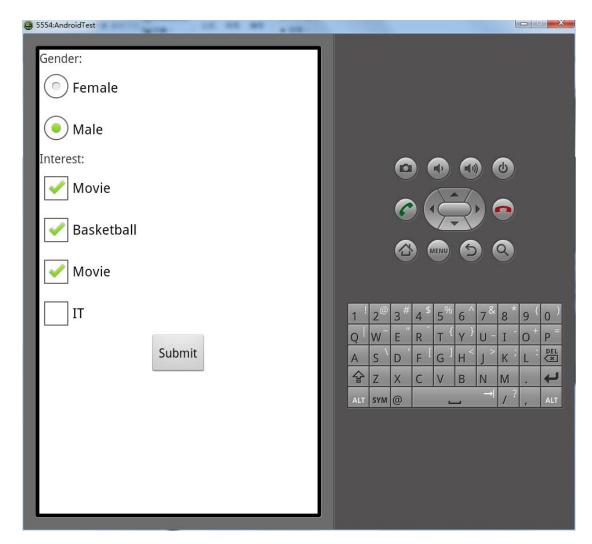
    <uses-sdk
        android:targetSdkVersion="8"
        android:targetSdkVersion="15" />

    <application
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
        android:name=".MainActivity"</pre>
```

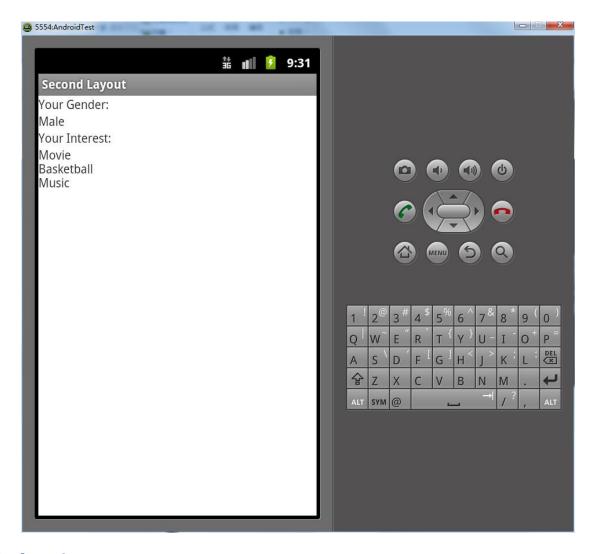
9. At the end of project, the index of the whole project should as follow:



10. The running result will be:



Select the gender and interests, click Submit button:



Explanation:

The code to hide title bar and status bar always exist in onCreate() method, and should be coded before setContent() method:

```
//hide title
requestWindowFeature(Window.FEATURE_NO_TITLE);
//hide status bar
int flag=WindowManager.LayoutParams.FLAG_FULLSCREEN;
this.getWindow().setFlags(flag, flag);
```

RadioButton is always used in a Group. Select one then others will lost focus. So, in the layout.xml file, RadioButtons should all be listed in the RadioGroup, as follow:

```
<RadioGroup
    android:id="@+id/radiogroup_gender"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" >
    <RadioButton</pre>
```

```
android:id="@+id/radio_female"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/genber_female" />
    <RadioButton
    android:id="@+id/radio_male"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/genber_male" />
    </RadioGroup>
```

Check whether the RadioButton/CheckButton is checked:

```
//if checked then return "true", else return "false"
femaleRadioButton.isChecked();
movieCheckBox.isChecked();
```

Jump from one layout to another layout:

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

Jump from one layout to another layout with dataset:

```
//put the data in the bundle with key and value,
bundle.putString(key,value)
Bundle bundle=new Bundle();
bundle.putString("gender", gender);
bundle.putStringArrayList("interest", (ArrayList<String>) interests);
//prepare the jump target
Intent intent=new Intent(MainActivity.this,SecondActivity.class);
//attach the dataset bundle to intent
intent.putExtras(bundle);
//jump
startActivity(intent);
```

Receive the data from former layout:

```
//get intent comes from former layout
Intent intent=this.getIntent();
//get the dataset bundle attached in the intent
Bundle bundle=intent.getExtras();
//get each data value from the bundle according to data key and transfer
data to its former type
String gender=(String) bundle.get("gender");
List<String> interests=bundle.getStringArrayList("interest");
```