# Lab 2 part 2

In this part, students will develop a mobile application to try different layouts. The layout included will be shown below:

* ConstraintLayout
* TableLayout
* ImageButton
* CheckBox
* RadioButton
* RatingBar

Screen captures of the application:

Graphical user interface

Description automatically generatedGraphical user interface, application

Description automatically generated

Graphical user interface

Description automatically generated with medium confidenceGraphical user interface, application

Description automatically generated

Step 1:

In Android Studio, create a new project named Lab 2 Part 2 with following project setting:

* Choose your project : **Empty Activity**
* Application Name : **Lab 2 Part 2**
* Project location : use the default setting
* Language : **Java**
* Minimum API level: **API 21: Android 5.0 (Lollipop)**
* Click **Finish**

Step 2:

Define string values in **res/values/strings.xml**

<resources>  
 <string name="app\_name">Lab 2 part 2</string>  
 <string name="cb\_mobile">Mobile</string>  
 <string name="cb\_web">Web</string>  
 <string name="select">You selected </string>  
 <string name="unselected">unselected</string>  
 <string name="fun">Fun</string>  
 <string name="boring">Boring</string>  
</resources>

Step 3:

Open the **res/layout/activity\_main.xml** file and insert the following:

*<?*xml version="1.0" encoding="utf-8"*?>*<androidx.constraintlayout.widget.ConstraintLayout 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=".MainActivity">  
  
 <ImageButton  
 android:id="@+id/button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="16dp"  
 android:padding="10dp"  
 android:src="@drawable/selector"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TableLayout  
 android:id="@+id/tableLayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="16dp"  
 android:layout\_marginEnd="16dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/button">

<TableRow>  
  
 <CheckBox  
 android:id="@+id/cb\_mobile"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:onClick="onCheckboxClicked"  
 android:text="@string/cb\_mobile" />  
  
 <CheckBox  
 android:id="@+id/cb\_web"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:onClick="onCheckboxClicked"  
 android:text="@string/cb\_web" />  
 </TableRow>  
 </TableLayout>  
  
 <RadioGroup  
 android:id="@+id/radiogroup"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="16dp"  
 android:orientation="vertical"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/tableLayout">  
  
 <RadioButton  
 android:id="@+id/radio\_fun"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:onClick="onRadioButtonClicked"  
 android:text="@string/fun" />  
  
 <RadioButton  
 android:id="@+id/radio\_boring"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:onClick="onRadioButtonClicked"  
 android:text="@string/boring" />  
 </RadioGroup>  
  
  
 <RatingBar  
 android:id="@+id/ratingBar"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="16dp"  
 android:numStars="5"  
 android:stepSize="1.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/radiogroup" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

Step 4:

Add pic1.png into drawable.

Step 5:

Create selector.xml in drawable file and insert the following code:

*<?*xml version="1.0" encoding="utf-8"*?>*<selector xmlns:android="http://schemas.android.com/apk/res/android">  
 <item android:drawable="@drawable/pic2"  
 android:state\_pressed="true" />  
 <item android:drawable="@drawable/pic1" />  
</selector>

Step 6:

Open **MainActivity.java** and insert the following code for the MainActivity class:

public class MainActivity extends AppCompatActivity {  
  
 private ImageButton button;  
 private RatingBar ratingBar;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 button = findViewById(R.id.*button*);

ratingBar =findViewById(R.id.*ratingBar*);  
  
 }

}

Step 7:

Add the following code into onCreate() method to build up the logic of image button. If we do not assign a method in the xml file, that will be another way to activate the button.

button.setOnClickListener(new View.OnClickListener() { @Override  
public void onClick(View v) { *// Perform action on clicks* Toast.*makeText*(MainActivity.this, "Clicked", Toast.*LENGTH\_SHORT*).show(); }  
});

Step 8:

Add onCheckboxClicked() outside the onCreate() method to build up the logic of check box. We need to use isChecked() function to check whether the check box is clicked.

public void onCheckboxClicked(View view) {  
 *// Is the button now checked?* boolean checked = ((CheckBox) view).isChecked();  
 *// Check which checkbox was clicked* switch(view.getId()) {  
 case R.id.*cb\_mobile*:  
 if (checked)  
 Toast.*makeText*(MainActivity.this, getResources().getString(R.string.*select*) + " " + getResources().getString(R.string.*cb\_mobile*), Toast.*LENGTH\_SHORT*).show();  
 else  
 Toast.*makeText*(MainActivity.this, getResources().getString(R.string.*cb\_mobile*) + " " + getResources().getString(R.string.*unselected*),  
 Toast.*LENGTH\_SHORT*).show();  
 break;  
 case R.id.*cb\_web*:  
 if (checked)  
 Toast.*makeText*(MainActivity.this, "You selected Web",  
 Toast.*LENGTH\_SHORT*).show();  
 else  
 Toast.*makeText*(MainActivity.this, "Web Unselected", Toast.*LENGTH\_SHORT*).show();  
 break;  
 }  
  
}

This time we will hard code some text component instead of using the element of strings.xml. What is the effect?

Step 8:

Step 9:

Add onRadioButtonClicked() outside the onCreate() method to build up the logic of radio button.

public void onRadioButtonClicked(View view) {  
 *// Is the button now checked?* boolean checked = ((RadioButton) view).isChecked();  
 *// Check which radio button was clicked* switch(view.getId()) {  
 case R.id.*radio\_fun*:  
 if (checked)  
 Toast.*makeText*(MainActivity.this, "Fun",  
 Toast.*LENGTH\_SHORT*).show();  
 break;  
 case R.id.*radio\_boring*:  
 if (checked)  
 Toast.*makeText*(MainActivity.this, "Boring",Toast.*LENGTH\_SHORT*).show();  
 break;

}  
}

Step 10:

Add the following code into onCreate() method to build up the logic of rating bar.

ratingBar.setOnRatingBarChangeListener(new RatingBar.OnRatingBarChangeListener() {  
 public void onRatingChanged(RatingBar ratingBar, float rating, boolean  
 fromUser) {  
 Toast.*makeText*(MainActivity.this, "New Rating: " + rating,  
 Toast.*LENGTH\_SHORT*).show();  
} });

Step 11 (extra):

In Step 6, we hardcoded some text, what will happen when we want to change the language? Let’s try~

Open the activity\_main.xml and click “Open Editor”.

Step 11 (extra):

Click the Earth icon, choose “HKSAR” as the language choice.

Icon

Description automatically generated with medium confidence

Then you will observe a column will be created. Type the Chinese meaning in that column and run the application. Then change the language of the emulator to see what’s worse.