

中山大学移动信息工程学院本科生实验报告

(2017 年秋季学期)

课程名称：移动应用开发

任课教师：郑贵锋

年级	15 级	专业 (方向)	移动互联网 M3
学号	15352458	姓名	庄博伟
电话	18922361239	Email	1026051192@qq.com
开始日期	2017.9.23	完成日期	2017.9.24

一、 实验题目

∞ 基本UI界面设计

熟悉 Android Studio 开发工具操作，熟悉 Android 基本 UI 开发，并进行 UI 基本设计

二、 实现内容

实现一个 Android 应用，界面呈现如下效果：

Lab1

中山大学学生信息系统



学号：

密码：

☒ 学生 ☐ 教职工

登录

注册

要求：

☞ 该界面为应用启动后的第一个界面；

☞ 各控件的要求如下：

要求只用一个 ConstraintLayout 实现整个布局；

标题字体大小 20sp，与顶部距离 20dp，居中；

图片与标题的间距为 20dp，居中；输入框整体距屏幕右边间距 20dp，上下两栏间距 20dp，内容（包括提示内容）如图所示，内容字体大小 18sp；

学号对应的 EditText 只能输入数字，密码对应的 EditText 输入方式为密码；

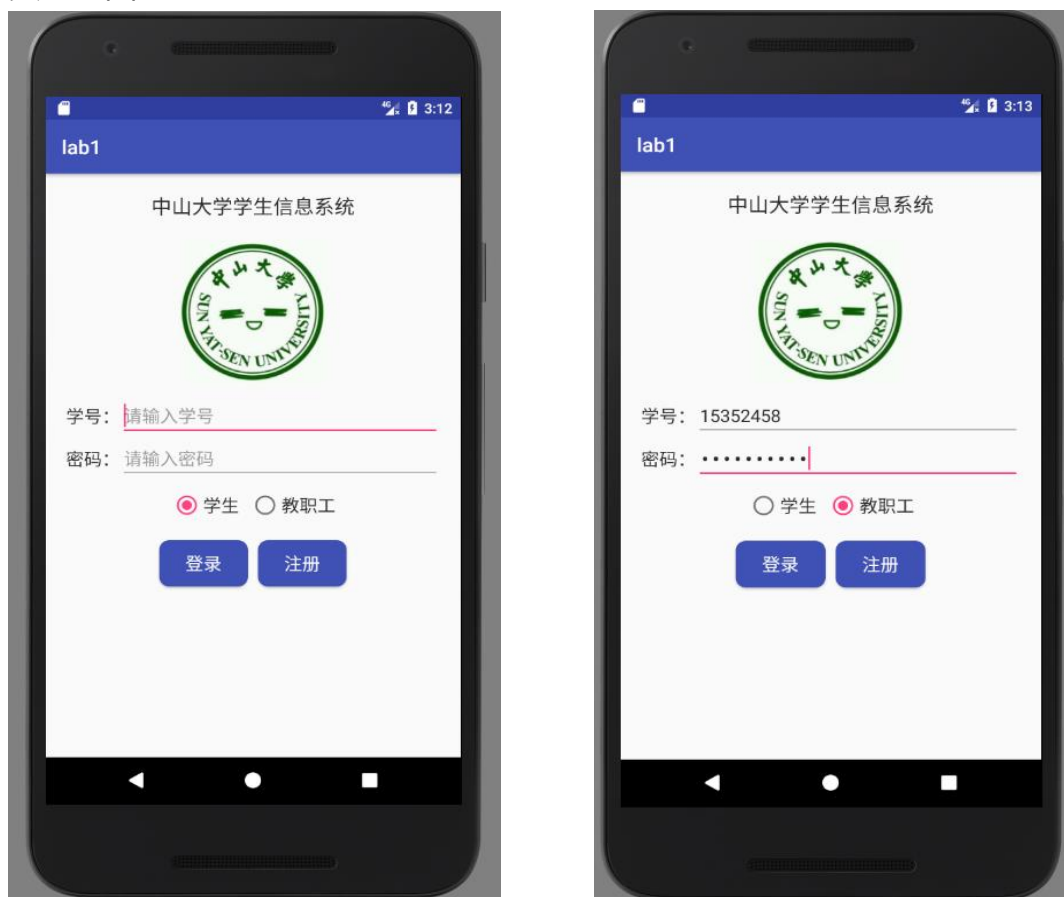
两个单选按钮整体居中，字体大小 18sp，间距 10dp，默认选中的按钮为第一个；

两个按钮整体居中，与上方控件间距 20dp，按钮间的间距 10dp，文字大小 18sp。按钮背景框左右边框与文字间距 10dp，上下边框与文字间距 5dp，圆角半径 10dp，背景色为 #3F51B5

☞ 使用的组件：ConstraintLayout、TextView、EditText、Button、ImageView、RadioGroup、RadioButton

三、 课堂实验结果

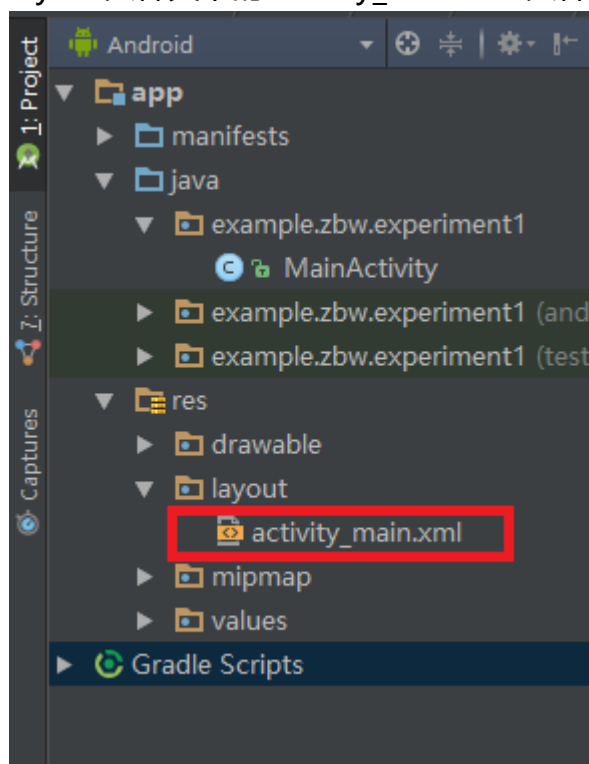
（1）实验截图



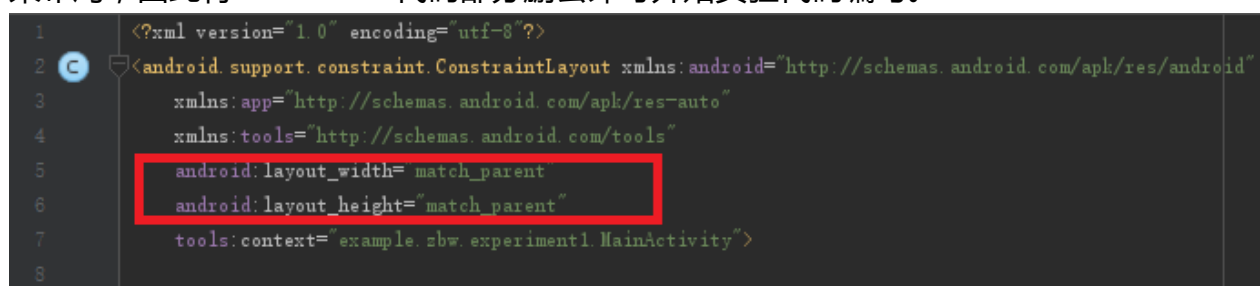
如上，左图为 app 启动后的第一个界面，右图为输入学号与密码后，学号只能输入数字，密码的输入方式为“点”（即隐藏），且对单选按钮进行操作的界面，符合实验题目要求。

（2）实验步骤以及关键代码

- 新建项目工程，模版选择空模版，其他步骤基本默认。
- 进入项目编辑界面，在左侧project栏选择Android，首先双击res资源文件夹中layout文件夹下的activity_main.xml文件开始代码编写。



- 布局：布局就是让控件在里面按一定的次序排列好的一种组件，本身并不提供内容。实验要求只用一个ConstraintLayout实现整个布局，即约束布局（根据布局中其他元素或视图确定控件在屏幕中的位置）。双击进入activity_main文件后发现默认代码布局就是约束布局，因此将helloworld代码部分删去即可开始实验代码编写。

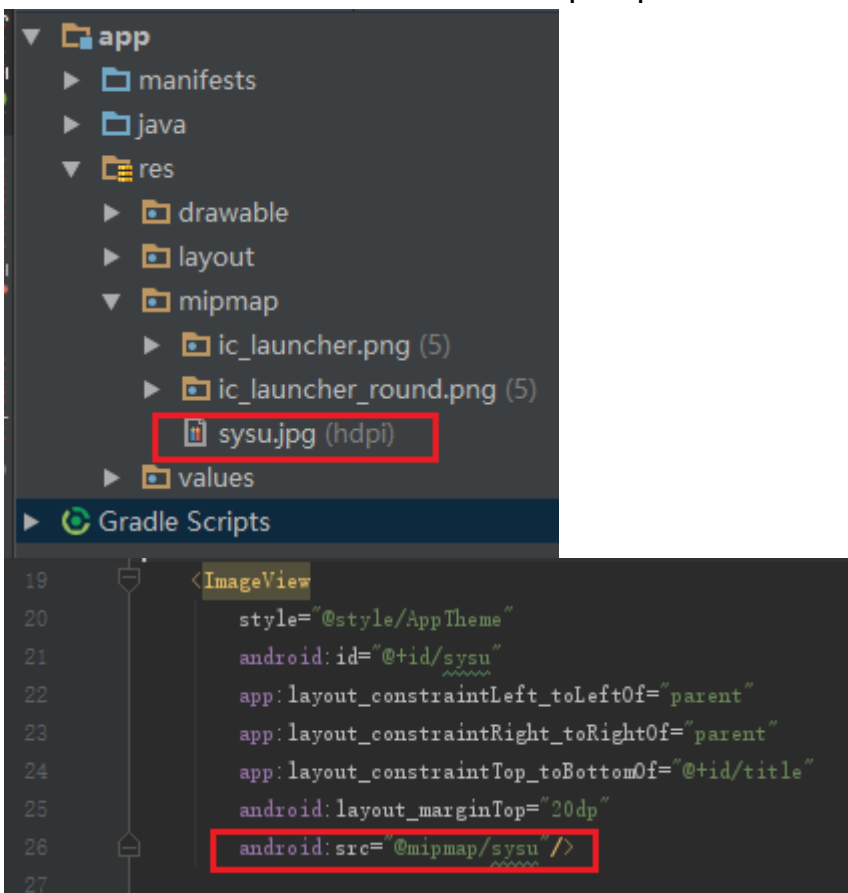


其中，match_parent 指的是组件的宽度或高度与父组件的宽度或高度一致，如果组件没有父组件，那么组件的宽度或高度与屏幕的宽度或高度一致。wrap_content 指组件的宽度或高度刚好可以包裹住组件内的子组件即可，在 ConstraintLayout 整个布局属性的设置里面使用 match_parent。

∞ 开始进行控件的设置。控件就是显示内容的组件。首先是标题使用textview控件，确定id名称后，layout_width与layout_height使用wrap_content，textColor textSize text 分别设置字体颜色，大小和文本内容，其中部分内容使用了引用资源style的方法，这个地方会在拓展部分详细说明。接着设置控件的位置，左右变节与父容器边界对齐，则为居中；上边界对齐父边界，再用marginTop使其与顶部距离20dp。

```
9      <TextView
10          style="@style/AppTheme"
11          android:id="@+id/title"
12          android:textSize="20sp"
13          android:text="中山大学学生信息系统"
14          app:layout_constraintLeft_toLeftOf="parent"
15          app:layout_constraintRight_toRightOf="parent"
16          app:layout_constraintTop_toTopOf="parent"
17          android:layout_marginTop="20dp"/>
18
```

∞ 接下来是显示图片，采用ImageView控件，左右对齐父容器居中，顶部对齐标题的底部，再设置20dp距离。重要的src属性引用图片资源。在实验文件夹中将sysu.jpg图片文件复制到项目目录下res资源文件夹中mipmap文件夹下，然后通过src属性引用图片资源



```
19      <ImageView
20          style="@style/AppTheme"
21          android:id="@+id/sysu"
22          app:layout_constraintLeft_toLeftOf="parent"
23          app:layout_constraintRight_toRightOf="parent"
24          app:layout_constraintTop_toBottomOf="@+id/title"
25          android:layout_marginTop="20dp"
26          android:src="@mipmap/sysu"/>
27
```

接下来是2个输入框。拿其中一个输入框来说明，由一个TextView与一个EditText控件组成，TextView左对齐父容器，EditText右对齐父容器，TextView的右边界与EditText的左边界互相对齐形成chain。EditText的上下边界对齐TextView，宽度设置为0dp使输入框铺满剩下的整个屏幕。学号的输入框输入形式设置为数字，使用hint属性使用户未输入前提示“请输入学号”。

```
28 <TextView
29     style="@style/AppTheme"
30     android:id="@+id/xuehao"
31     android:text="学号:"
32     android:layout_marginTop="20dp"
33     android:layout_marginLeft="20dp"
34     app:layout_constraintLeft_toLeftOf="parent"
35     app:layout_constraintRight_toLeftOf="@+id/edit1"
36     app:layout_constraintTop_toBottomOf="@+id/sysu"/>
37
38 <EditText
39     style="@style/AppTheme"
40     android:layout_width="0dp"
41     android:id="@+id/edit1"
42     app:layout_constraintTop_toTopOf="@+id/xuehao"
43     app:layout_constraintBottom_toBottomOf="@+id/xuehao"
44     app:layout_constraintLeft_toRightOf="@+id/xuehao"
45     app:layout_constraintRight_toRightOf="parent"
46     android:layout_marginRight="20dp"
47     android:inputType="number"
48     android:hint="请输入学号"/>
49
```

接下来密码这一行也相似，唯一不同是输入框输入形式为textPassword使输入的文字变为小点，与输入提示为“请输入密码”。

```
50 <TextView
51     style="@style/AppTheme"
52     android:id="@+id/mima"
53     android:text="密码:"
54     android:layout_marginTop="20dp"
55     android:layout_marginLeft="20dp"
56     app:layout_constraintTop_toBottomOf="@+id/xuehao"
57     app:layout_constraintLeft_toLeftOf="parent"
58     app:layout_constraintRight_toLeftOf="@+id/edit2"/>
59
60 <EditText
61     style="@style/AppTheme"
62     android:layout_width="0dp"
63     android:id="@+id/edit2"
64     app:layout_constraintTop_toTopOf="@+id/mima"
65     app:layout_constraintBottom_toBottomOf="@+id/mima"
66     app:layout_constraintLeft_toRightOf="@+id/mima"
67     app:layout_constraintRight_toRightOf="parent"
68     android:layout_marginRight="20dp"
69     android:inputType="textPassword"
70     android:hint="请输入密码"/>
71
```

☞ 然后是两个单选按钮。使用RadioGroup控件，设置orientation属性为水平，表示按钮水平放置。再在控件中添加2个RadioButton，需要设置id才能正常工作。由于题目要求默认勾选第一个按钮，所以在第一个按钮添加属性checked设置值为true。

```
72 <RadioGroup
73     style="@style/AppTheme"
74     android:id="@+id/radio"
75     android:orientation="horizontal"
76     android:layout_marginTop="20dp"
77     app:layout_constraintTop_toBottomOf="@+id/mima"
78     app:layout_constraintLeft_toLeftOf="parent"
79     app:layout_constraintRight_toRightOf="parent">
80
81     <RadioButton
82         style="@style/AppTheme"
83         android:id="@+id/id1"
84         android:checked="true"
85         android:text="学生"/>
86
87     <RadioButton
88         style="@style/AppTheme"
89         android:id="@+id/id2"
90         android:layout_marginLeft="10dp"
91         android:text="教职工"/>
92 </RadioGroup>
93
```

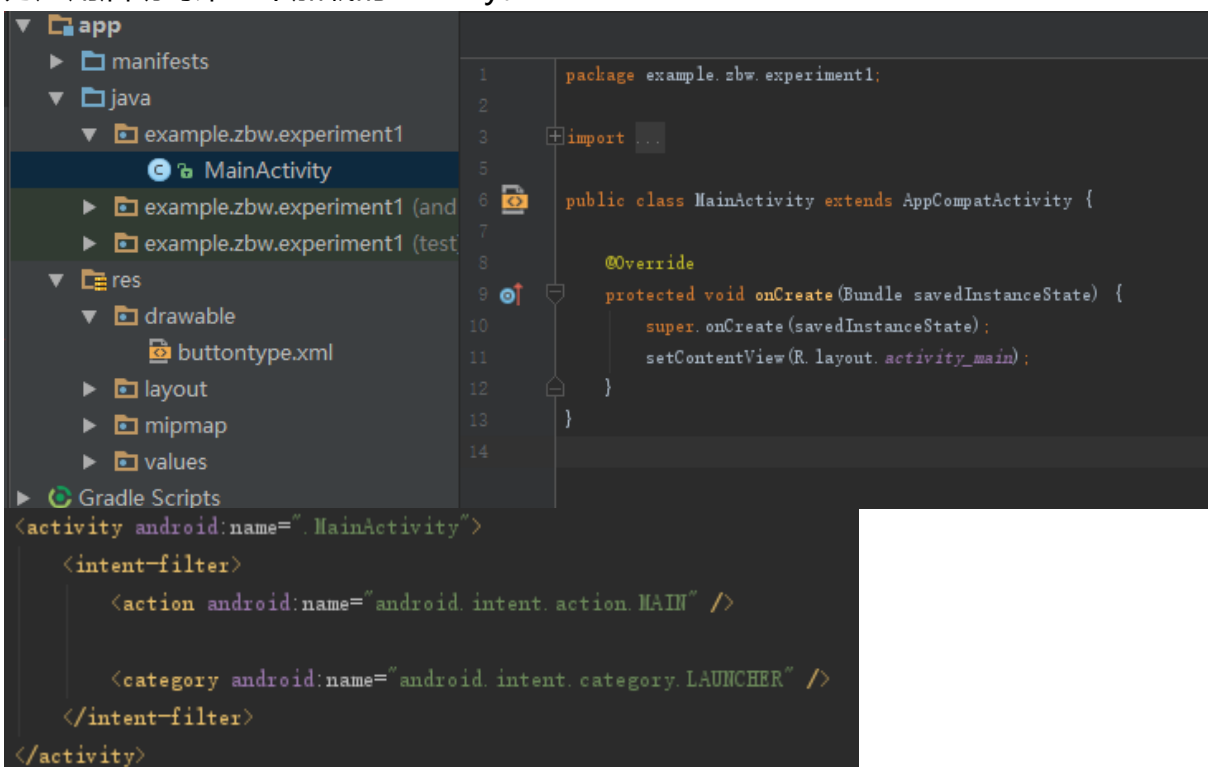
☞ 最后是登录和注册两个按钮。与输入框处约束相似，将两个按钮连成链，再通过app:layout_constraintHorizontal_chainStyle改变水平链样式为packed使其紧密连接。注意要设置题目要求的按钮间隔10dp，且登录与注册两字颜色为白色。

```
94 <Button
95     style="@style/AppTheme"
96     android:id="@+id/button1"
97     android:layout_marginTop="20dp"
98     app:layout_constraintTop_toBottomOf="@+id/radio"
99     app:layout_constraintLeft_toLeftOf="parent"
100     app:layout_constraintRight_toLeftOf="@+id/button2"
101     app:layout_constraintHorizontal_chainStyle="packed"
102     android:background="@drawable/button_type"
103     android:text="登录"
104     android:textColor="@color/Primary_white"/>
105
106 <Button
107     style="@style/AppTheme"
108     android:id="@+id/button2"
109     android:layout_marginTop="20dp"
110     app:layout_constraintTop_toBottomOf="@+id/radio"
111     app:layout_constraintRight_toRightOf="parent"
112     app:layout_constraintLeft_toRightOf="@+id/button1"
113     android:layout_marginLeft="10dp"
114     android:background="@drawable/button_type"
115     android:text="注册"
116     android:textColor="@color/Primary_white"/>
117
```

其中，按钮要使用自定义背景边框。在res资源文件夹drawable文件夹下新建 buttontype.xml文件，然后把自动生成的selector标签改为shape，corner用于控制边框四个角圆角的大小radius值按要求设置为10dp，solid用于控制背景颜色，将其的color属性按照要求设置为#3F51B5（此处使用了资源调用color），padding用于控制背景边框与背景中的内容的距离，也就是内边距。一共包含四个属性：bottom、top、left、right，分别用于控制上、下、左、右的内边距，这里按照需求分别设置为5dp,5dp,10dp,10dp,设置完成之后就可以在布局中使用了。

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <shape xmlns:android="http://schemas.android.com/apk/res/android">
3      <corners
4          android:radius="10dp"/>
5      <padding
6          android:top="5dp"
7          android:bottom="5dp"
8          android:left="10dp"
9          android:right="10dp"/>
10     <solid
11         android:color="@color/button"/>
12 </shape>
```

∞ 这样布局就完成了。而要在应用中显示布局的话，在创建项目的时候已经完成了必要步骤。即在MainActivity.java中将布局引入，然后在注册文件中注册，将该Activity设置为应用启动时第一个加载的Activity。



```
1 package example.zbw.experiment1;
2
3 import ...
4
5
6 public class MainActivity extends AppCompatActivity {
7
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12     }
13 }
14
```

```
<activity android:name=".MainActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

然后就可以运行了。

(3) 实验遇到困难以及解决思路

实验问题	解决思路
添加图片时发现图片相对较大	mipmap 有不同图片尺寸的文件夹，将图片放在最小的-hdpi 文件夹再调用。
输入框与左侧的文字不在同一水平线	将输入框上下边界对齐 TextView
输入框无法铺满整个剩余屏幕	将宽度设置为零 (0dp) 可解决问题并且提高布局性能，注意约束要和 0dp 的方向一致。否则无效。
登录与注册两个按钮之间的间距无法达到要求	设置链式结构，水平链样式设为 packed 之后再设置间隔距离

四、 课后实验结果

∞ 实验文档中有涉及到拓展知识：关于资源的引用。Android项目中不建议使用硬编码来使用资源，建议将各类资源定义在res文件夹中的values文件夹下，字符串资源定义在strings.xml中，颜色资源定义在colors.xml中，距离，字体大小资源定义在dimens.xml中。图片资源例外，是存放在res文件夹中的mipmap文件夹下或者drawable文件夹下。而自己在完成基础实验后也在试着修改代码风格，尽量使用更多的资源引用来简化代码。

1) string定义即将使用的text内容

```
1  <resources>
2      <string name="app_name">lab1</string>
3      <string name="title">中山大学学生信息系统</string>
4      <string name="a">学号: </string>
5      <string name="b">请输入学号</string>
6      <string name="c">密码: </string>
7      <string name="d">请输入密码</string>
8      <string name="e">学生</string>
9      <string name="f">教职工</string>
10     <string name="g">登录</string>
11     <string name="h">注册</string>
12 </resources>
```

调用时候的格式：

```
android:text="@string/title"
```


2) color 设置颜色

```
1      <?xml version="1.0" encoding="utf-8"?>
2      <resources>
3          <color name="colorPrimary">#3F51B5</color>
4          <color name="colorPrimaryDark">#303F9F</color>
5          <color name="colorAccent">#FF4081</color>
6          <color name="Primary_black">#D5000000</color>
7          <color name="Primary_white">#FFFFFFFF</color>
8          <color name="button">#3F51B5</color>
9      </resources>
10
```

调用时候的格式：

```
android:textColor="@color/Primary_white"
```

3) dimen 设置字体大小

```
1      <?xml version="1.0" encoding="utf-8"?>
2      <resources>
3          <dimen name="title_text">20sp</dimen>
4          <dimen name="normal_text">18sp</dimen>
5      </resources>
```

调用时候的格式：

```
android:textSize="@dimen/title_text"
```

4) style 的设置

```
1      <resources>
2
3      <!-- Base application theme. -->
4      <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
5          <!-- Customize your theme here. -->
6          <item name="colorPrimary">@color/colorPrimary</item>
7          <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
8          <item name="colorAccent">@color/colorAccent</item>
9          <item name="android:layout_width">wrap_content</item>
10         <item name="android:layout_height">wrap_content</item>
11         <item name="android:textColor">@color/Primary_black</item>
12         <item name="android:textSize">@dimen/normal_text</item>
13     </style>
14
15 </resources>
16
```

可以在控件中进行调用，调用时候的格式：

```
style="@style/AppTheme"
```

除此之外，由于上理论课的时候老师有讲到另外几个不同的布局，我也尝试用不同的布局来完成这一个实验。在自己的尝试下，我用了线性布局LinearLayout与表格布局TableLayout来完成拓展实验，同样也能达到实验所需要的结果（项目工程也会一同打包）。相比之下，用线性布局与表格布局需要层层嵌套，而约束布局则一个就能完成实验内容，后者显得更加简单快捷。

五、 实验思考及感想

第一次的安卓实验不算特别复杂，也有充足的时间去完成实验并进行拓展，目的是熟悉 Android 基本 UI 开发,并进行 UI 基本设计，我确实在这次实验中掌握了一些布局方法，以及一些控件的使用。

这次实验要求使用比较新的一种布局：ConstraintLayout 约束布局来完成。其中也用到了一些基础的控件。约束布局，根据布局中的其他元素或视图, 确定 View 在屏幕中的位置, 受到三类约束, 即其他视图, 父容器(parent), 基准线(Guideline)。而比较需要注意的是链样式，在设计控件之间的间距时非常有用。

而控件方面则运用了 TextView、EditText、Button、ImageView、RadioGroup、RadioButton 这几个。关于控件里面的属性设置，在实验中反反复复地调试也有了更深层次的理解。其中印象最深刻的是学会了让列表框适配整个屏幕的方法。在约束布局下是利用宽度为 0dp 来实现，在表格布局中则是通过属性 stretchColumns 来实现，都是非常有用且容易忽视的细节。

其中，实验文档中有一处提到了 ImageView 的 src 和 background 属性的区别，通过查阅资料也是弄懂了。background 会根据 ImageView 组件给定的长宽进行拉伸，而 src 就存放的是原图的大小，不会进行拉伸。src 是图片内容（前景），bg 是背景，可以同时使用。此外：scaleType 只对 src 起作用；background 可设置透明度，比如在 ImageButton 中就可以用 android:scaleType 控制图片的缩放方式。

在实验外的拓展中体会到了资源引用的好处，能够让自己的编码更加方便简洁，很大程度减少不必要的重复代码量。在以后的安卓设计中，会更多的使用资源引用的方法，对一些高频使用的重复代码段利用不同的资源进行引用会方便很多。如果需要修改，则直接在 res 文件夹中修改就 ok 了。而在尝试了不同的布局设计方法后，对几个基础常用的局部也有了深刻的理解。另外，理论课上涉及的剩余两种常用布局，RelativeLayout 相对布局与 FrameLayout 框架布局这次实验并没有使用到，相信在之后的实验中会接触的到。