

# 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 )	移动信息工程 ( 物联网 )
学号	15352084	姓名	冯镜蓉
电话	13265300250	Email	276965324@qq.com
开始日期	2017-10-3	完成日期	2017-10-4

### 一、 实验题目

#### 实验一 基本 UI 界面设计

### 二、 实验内容

实现一个 Android 应用，界面呈现如下效果：



要求：

- 1) 该界面为应用启动后看到的第一个界面；
- 2) 要求只用一个 ConstraintLayout 实现整个布局；
- 3) 标题：字体大小 20sp，与顶部距离 20dp，居中；
- 4) 图片：与标题的间距为 20dp，居中；
- 5) 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；
- 6) 单选按钮：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；
- 7) 按钮：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；

- 8) 使用的布局和控制件：ConstraintLayout、TextView、EditText、Button、ImageView、RadioGroup、RadioButton

### 三、 课堂实验结果

#### (1) 实验截图



#### (2) 实验步骤以及关键代码

A. 首先，在 src/res/layout 文件夹下写好布局文件 main\_layout.xml

- a) 首先在 src/main/res/values/colors 及 src/main /res/values/strings 中将布局文件中将会引用的颜色和字符串资源命名并定义完整，并将图片 sysu.png 加入 src/res/mipmap 文件夹。

```
<resources>
<string name="app_name">My Application_1</string>
<string name="Title">中山大学学生信息系统</string>
<string name="sid_label">学号: </string>
<string name="Edit_sid">请输入学号</string>
<string name="password_label">密码: </string>
<string name="Edit_password">请输入密码</string>
<string name="student">学生</string>
<string name="faculty">教职工</string>
<string name="login">登录</string>
<string name="signup">注册</string>
</resources>
```

- b) 标题—文本控件 TextView

```
<TextView
    android:id="@+id/Title"          定义资源 id (标识符)
    android:layout_width="wrap_content"  指定组件的高度、宽度均为恰好包含内容
    android:layout_height="wrap_content"
    android:text="@string/Title"  引用已定义的字符串资源 Title
    android:textSize="20sp"      指定文本字体大小与颜色
    android:textColor="@color/primary_black"
    android:layout_marginTop="20dp"
    app:layout_constraintTop_toTopOf="parent"  视图位置受父容器约束:
    app:layout_constraintLeft_toLeftOf="parent"  居中、距离顶部 20dp
    app:layout_constraintRight_toRightOf="parent" />
```

### c) 图片—图片控件 ImageView

```
<ImageView
    android:id="@+id/Image"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@mipmap/sysu"      引用 mipmap 资源文件夹中的图片 sysu
    app:layout_constraintTop_toBottomOf="@id/Title"  位置受到 Title 视图的约束:
    android:layout_marginTop="20dp"                顶部距离 Title 视图底部 20dp
    app:layout_constraintLeft_toLeftOf="parent"      位置受到父容器的约束: 居中
    app:layout_constraintRight_toRightOf="parent" />
```

### d) 输入框—文本控件 TextView 和输入框控件 EditText

```
<TextView
    android:id="@+id/sid_label"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/sid_label"
    android:textSize="18sp"
    android:textColor="@color/primary_black"
    app:layout_constraintLeft_toLeftOf="parent"
    android:layout_marginStart="20dp"
    app:layout_constraintTop_toBottomOf="@id/Image"
    android:layout_marginTop="20dp" />

<EditText
    android:id="@+id/Edit_sid"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:inputType="number"  指定输入框输入类型为数字
    android:hint="@string/Edit_sid"  输入框提示内容引用字符串资源
    android:textSize="18sp"
    app:layout_constraintRight_toRightOf="parent"  位置受 sid_label 基准线约束
    app:layout_constraintBaseline_toBaselineOf="@id/sid_label"
    app:layout_constraintLeft_toRightOf="@id/sid_label"
    android:layout_marginRight="20dp"  右受父容器约束: 右边间距 20dp
    android:layout_marginLeft="20dp" />  左对齐 sid_label 右部, 间距 20dp

<EditText
    android:id="@+id/Edit_password"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:inputType="textPassword"  指定输入框输入类型为密码文本
    android:hint="@string/Edit_password"  使输入的文字变为小点
    android:textSize="18sp"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintBaseline_toBaselineOf="@id/password_label"
    app:layout_constraintLeft_toRightOf="@id/password_label"
    android:layout_marginRight="20dp"
    android:layout_marginLeft="20dp" />
```

### e) 单选按钮—控件 RadioButton 和 RadioGroup

RadioButton 是单选按钮，一组单选按钮需要包括在一个 RadioGroup 中，并且需要对 RadioGroup 和其包括的每一个 RadioButton 都设置 id，RadioButton 才能正常工作。

```
<RadioGroup
    android:id="@+id/id0"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"  指定朝向为水平
    app:layout_constraintTop_toBottomOf="@id/Edit_password"
    android:layout_marginTop="20dp"  距离输入框底部 20dp
    app:layout_constraintLeft_toLeftOf="parent"  居中
    app:layout_constraintRight_toRightOf="parent" />

    <RadioButton
        android:id="@+id/id1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintRight_toLeftOf="@+id/id2"
        android:layout_marginEnd="10dp"  与单选按钮 id2 左部距离 10dp
        android:textSize="18sp"
        android:text="@string/student"
        android:checked="true" />  默认选中单选按钮 id1

    <RadioButton
        android:id="@+id/id2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="18sp"
        android:text="@string/faculty" />
</RadioGroup>
```

#### f) 按钮—按钮控件 Button

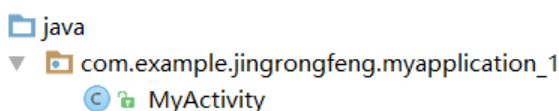
首先自定义背景边框：在 src/main/res/drawable 文件夹下新建一个 Drawable resource file，填写 file name 为 button\_bg，然后把自动生成的 selector 标签改为 shape，设置 shape 包含的 6 个属性：corners、gradient、padding、size、solid、stroke。

```
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <corners                控制边框四个角圆角的大小
        android:radius="10dp" />
    <padding                控制边框与内容的距离（内边距）
        android:top="5dp"
        android:bottom="5dp"
        android:left="10dp"
        android:right="10dp" />
    <solid                  控制背景边框颜色
        android:color="@color/colorPrimary" />
</shape>
```

接下来在布局文件中编写 Button 控件的代码，并引用上述背景边框资源：

```
<Button
    android:id="@+id/btn1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/login"
    android:textSize="18sp"
    android:textColor="@color/primary_white"
    android:background="@drawable/button_bg" 引用背景边框资源 button_bg
    app:layout_constraintRight_toLeftOf="@+id/btn2"
    android:layout_marginEnd="10dp" 与 btn2 间距 10dp
    app:layout_constraintTop_toBottomOf="@+id/id0"
    android:layout_marginTop="20dp" 与上方单选按钮控件间距 20dp
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintHorizontal_chainStyle="packed" /> 与 btn2 双向链接，组成 packed 链条
```

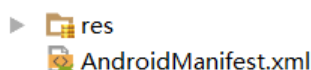
#### B. 然后创建一个 java 文件，在该文件中将布局引入



```
public class MyActivity extends AppCompatActivity {

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

#### C. 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



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

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

### (3) 实验遇到困难以及解决思路

由于刚开始不熟悉链条样式，最后的按钮控件总是无法达到理想效果；以及经常将不同布局方式和三类约束布局混淆。最后解决的方法是，先将实验文档重新认真阅读了一遍，明确了实验要求与所述知识点，并在遇到模糊的概念名词时上网查阅资料，看别人的博客与经验，例如 <http://www.cnblogs.com/ludashi/p/4883915.html> 等，进一步加深了对使用 AndroidStudio 进行 UI 界面设计的掌握，深入了解了布局与控件的知识，最后能够自己较为熟练地应用。

## 四、实验思考及感想

通过本次实验体会到了使用 AndroidStudio 进行 UI 界面设计，感受到 AndroidStudio 中已经包含了很漂亮丰富的 UI 库，使用方便，编程较为简单，能够满足很大部分的 Android App 的 UI 设计需求，并便于将项目文件统一化管理。

另一方面，在本次实验中初步接触 AndroidStudio，逐渐熟悉 Android Studio 开发工具的操作，尽管只是进行基本的 UI 设计，却为将来的进一步实验打下基础，初探到安卓应用开发的魅力。尽管不是大神，也能享受从零学起的经历，享受一步一个脚印的成就感。希望现在踩过的坑能换来将来走更少的弯路，现在一点一滴的学习都能在将来得到回报。