

# 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	专业（方向）	移动互联网
学号	15352434	姓名	赵佐骑
电话	18686667712	Email	<u>1398772969@qq.com</u>
开始日期	2017/9/23	完成日期	2017/10/2

### 一、 实验题目

实验一：基本 UI 界面设计

### 二、 实现内容

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) 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；3/11

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`

(3) 使用的布局和控件：`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

首先要拉一条垂直居中的 `guideline` 作为参考线：

```
<android.support.constraint.Guideline
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/guideline"
    android:orientation="vertical"
    app:layout_constraintGuide_percent="0.5" />
```

最上面的 `TextView` 和 `ImageView` 都比较简单，不再赘述，接下来是中间的 `EditText` 部分：

首先把左边的 `TextView` 与 `EditText` 连成一个 `chain` 以方便调节，在确定高度方面，将 `TextView` 的顶部与图片的底部作为约束，同时将 `EditText` 的顶部与 `TextView` 的顶部的顶部作为约束，这样就能确保二者实在同一条直线上。这里需要注意的是要在 `TextView` 的样式中加入这样一行代码以确保居中：

```
<item name="android:gravity">center_vertical</item>
```

接下来的 `radio button` 记得放到一个 `RadioGroup` 中，这样才能实现单选的效果；下面两个 `Button` 需要新定义一个蓝色圆角矩形样式的 `.xml` 文件：

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <corners
        android:radius="10dp"/>
    <padding
        android:top="5dp"
        android:bottom="5dp"
        android:left="10dp"
        android:right="10dp"/>
    <solid
        android:color="@color/buttoncolor"/>
</shape>
```

然后在样式中这样引用即可：

```
<item name="android:background">@drawable/rectangle</item>
```

由于 Android 项目中不建议使用硬编码来使用资源，所以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 dims.xml 中。图片资源例外，是存放在 res 文件夹中的 mipmap 文件夹下或 drawable 文件夹下。

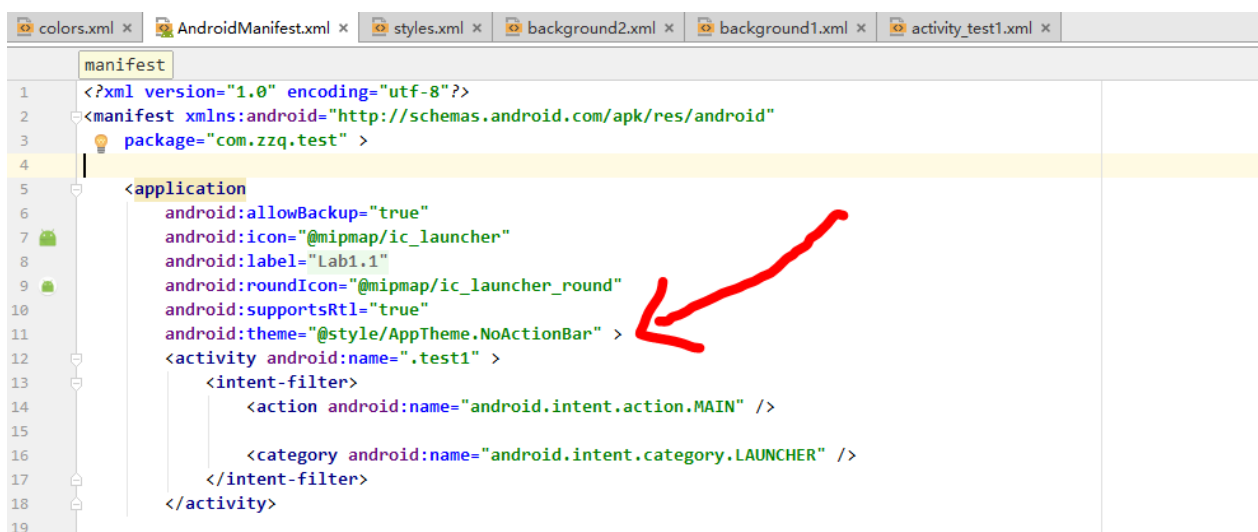
### （3）实验遇到困难以及解决思路

由于本次实验比较简单，遇到的一个问题就是 EditText 与左边的 TextView 高度不一致，加了一行设置 gravity 的代码就解决了。

## 四、课后实验结果

课后便想美化一下界面，首先觉得界面上方那个显示 APP 名称栏和状态栏有些丑，便想稍微做下改进，通过上网查找发现原来是在主界面中的 android:theme 中去掉的，并可以更改颜色，color 中的

<color name="colorPrimaryDark">#238d50</color>便是对应的最上方状态栏的颜色，为了迎合中大的风格，我选择了原谅绿色。同时 color 中的<color name="colorAccent">#27ae60</color>对应的是 radiobutton 之类的 view 被选中的颜色，我同样选择了绿色。



接着便找来中大的 logo，放在左上方，由于一般的界面都需要图标 icon 这样子，通过查找发现 Android Studio 有一个图标插件用来自动生成图标，叫 Material Icon Generator。它会自动生成不同分辨率的图标以适应不同机型。

接着便是怎么用图标的问题了，发现 `android:drawableRight="@drawable/..."`

语句可以在一个 View 的右边添加一个 icon，便在 `edittext` 里面分别加了以下代码：

```
android:drawableRight="@drawable/ic_account_circle_grey_300_24dp"
android:drawableRight="@drawable/ic_lock_black_24dp"
```

在查找更改 `edittext` 的样式的过程中，发现为了更好的用户体验，可以对选择编辑的部分进行样式改变，这里需要一个 `selector` 来控制，我在这里增加了一个描边的样式：

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:state_focused="true" android:drawable="@drawable/focus"></item>
    <item android:state_focused="false"
android:drawable="@drawable/edittextbg"></item>
</selector>
```

`focus.xml` 里面是这样定义的：

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <solid
        android:color="#FFFFFF"/>
    <corners
        android:radius="10dp"/>
    <stroke
        android:width="2dp"
        android:color="@color/sysugreen"/>
</shape>
```

总体下来结果如下：



## 五．实验思考及感想

总的来说本次实验相对简单，主要在于熟悉约束布局和软件的基本操作，同时感觉由设计稿转化成成品的难度还是蛮大的，很多都不像在 ps 里操作的那么自如...这就导致有的时候脑袋里想的画面实现不了，通过本次实验，基本熟悉了 Android Studio 的相关操作，并对 UI 设计有了一定的掌握。

作业要求：

1. 命名要求: 学号\_姓名\_实验编号，例如 15330000\_林 XX\_lab1。
2. 实验报告提交格式为 pdf。
3. 实验内容不允许抄袭，我们要进行代码相似度对比。如发现抄袭，按 0 分处理。