

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
学号	15352055	姓名	陈泽伟
电话	13727023101	Email	1520414789@qq.com
开始日期	2017-09-26	完成日期	2017-10-07

一、 实验题目

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；

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

①创建 Lab1 项目并选择 Empty Activity 作为 App 的启动活动。

②创建布局 ConstraintLayout，关键代码如下：

```
android:layout_width="match_parent"
android:layout_height="match_parent"
```

layout_width= " match_parent " 和 layout_height= " match_parent " 分别将活动界面这一构件的宽度和高度进行强制性扩展，以填充布局单元内尽可能多的空间。

③创建控件 TextView 以显示标题，关键代码如下：

```

<TextView
    android:id="@+id/text_title"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/title"
    android:textSize="20sp"
    android:layout_marginTop="20dp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

```

id 是对该控件的命名，用于识别该控件；

layout_width= " match_centent " 和 layout_height= " match_content " 是对控件的宽度和高度进行强制性视图扩展，以显示控件中的全部内容；

text= " @string/title " 是对 strings.xml 文件中 title 内容的引用；

textSize 用于设置该控件字体大小；

layout_marginTop 指定该属性所在控件距上部控件的最小值，其中该上部控件可以按下面的方法来确定；

layout_constraintLeft_toLeftOf= " parent " 表示该控件的左边与 parent 控件（即父控件）的左边之间有一个约束，其他相似代码可以类推得到。

④创建控件 ImageView 用于显示图片，关键代码如下：

```

app:srcCompat="@mipmap/sysu"

```

srcCompat= " @mipmap/sysu " 表示引用 mipmap 文件中的 sysu 图像，这里 mipmap 文件有 5 个文件夹，分别为 mipmap-hdpi、mipmap-mdpi、mipmap-xhdpi、mipmap-xxhdpi、mipmap-xxxhdpi，一开始我并不明白为什么有这么多各文件夹，也不知道它们都有什么用处，后来经过不断地测试我发现，原来同一图片放在不同文件夹下在 App 里面显示出来的大小是不一样的，于是我便根据实验界面要求，将图片放在了 mipmap-xxhdpi 文件夹下。

⑤创建 EditText 控件用于输入，关键代码如下：

```

android:inputType="number"
app:layout_constraintBaseline_toBaselineOf="@+id/text_student_ID"

```

inputType= " number " 限定输入格式只能为数字；

layout_constraintBaseline_toBaselineOf= " @+id/text_student_ID "，将小孩的基线约束到目标子项的基线，在这里表示此控件与 text_student_ID 控件水平对齐。

⑥创建 RadioGroup 控件用于约束两个 RadioButton 控件，关键代码如下：

```

android:checkedButton="@+id/button_student"
android:orientation="horizontal"

```

checkedButton= " @+id/button_student " 表示默认选中 button_student 按钮；

orientation= " horizontal " 表示此控件的布局方式是水平布局。

⑦创建 RadioButton 控件。

⑧创建 Guideline 控件，用于使 Button 控件水平居中，关键代码如下：

```
android:orientation="vertical"  
app:layout_constraintGuide_percent="0.5"
```

orientation= " vertical " 表示该引导线是在整个布局中是垂直布局的；

layout_constraintGuide_percent= " 0.5 " ，表示在整个布局中引导线距离左边框的百分比，这里表示距离左边框 50% 的位置。

⑨设置按钮背景框，在 drawable 文件夹下新建一个 Drawable resource file，填写 file name，然后把自动生成的 selector 标签改为 shape，shape 下有多个属性，padding，radius，solid 等等，关键代码如下：

```
<shape xmlns:android="http://schemas.android.com/apk/res/android"  
    android:shape="rectangle" >  
  
    <solid android:color="#3F51B5" />  
    <corners android:radius="10dp" />  
    <padding  
        android:top="5dp"  
        android:bottom="5dp"  
        android:left="10dp"  
        android:right="10dp" />  
</shape>
```

shape= " rectangle " 表示形状为圆角矩形；

color= " #3F51B5 " 表示颜色为蓝色；

radius= " 10dp " 表示圆角半径为 10dp；

padding 用于设置按钮背景框上下左右边框与文字间距。

⑩创建 Button 控件，关键代码如下：

```
android:background="@drawable/buttonstyle"
```

Background= " @drawable/buttonstyle " 表示使用 drawable 文件下的 buttonstyle 格式作为此控件的背景框。

(3) 实验遇到困难以及解决思路

①第一个难点是图片控件中图片大小的限定。一开始我直接把图片拷贝到 drawable 目录下，想利用控件属性来规定图片大小，但后来我发现我们并不知道图片的大小，也不知道怎么规定图片的大小。因此，我上网查了相关的资料，知道应该将图片导入到 mipmap 目录下，且我发现该目录下有多个与 mipmap 相关的文件夹，经过一遍遍的尝试，我终于找到了限制图片大小的方法，并最终将大小适中的图片放到了界面上。

②第二个难点是 EditText 控件如何与 Text 控件处于同一水平线以及 EditText 控件如何调整宽度使得刚好占满整行。这次我是直接在网上搜索相关资料，知道可以用 layout_constraintBaseline_toBaselineOf 约束来使两个控件处于同一水平线，而关于 EditText 控件的宽度是我在无意中试出来的，一开始我随意设置了几个参数，发现效果都不理想，所以我想着先把它设为零，等后面的写完了再来弄，结果就这样被我试出来了。

③第三个难点是 RadioGroup 控件如何实现水平显示。这一难点我也是通过上网查找相关资料并最终解决的。

④第四个难点是如何实现两个 Button 的水平居中。这个难点的解决也是相当偶然，这是我在网上查阅其他问题的解决方案是无意中发现的。短视频里通过添加了界面引导线实现了两个 Text 控件的水平居中，后来我便根据此方法实现了两个 Button 的水平居中。

⑤第五个难点是如何自定义 Button 的背景边框。对于这一难点，我是直接根据 PPT 上面的资料和网址，通过 PPT 上的步骤和网上的教程，一步步实现的，其实也不是很难，但对于一开始接触 Android 开发的小白来说，还是存在一定难度的，还好教程很详细，不然我还真不一定做得出来。

四、 课后实验结果

暂时没有扩展功能。

五、 实验思考及感想

本次实验总体来说不是很难，只是要求实现一个 App 的开始界面，但是由于我对于 Android 开发完全没有经验以及对 Android Studio 的使用还没有完全掌握，导致我几乎完全看不懂代码，也不知道如何在各个界面中切换以及如何安装和使用虚拟机，所以前前后后我走了很多弯路。后来我认真真地在百度上搜索每一段代码所代表的意思以及每个控件的属性如何控制，并把前几节实验课的 PPT 重新复习了一遍，然后尝试着重新装了一遍虚拟机，经过国庆整整两天努力，我终于完整地实现了本次实验的内容。

经过本次实验，我也意识到要成为一个优秀的 Android 开发者，耐心和细心尤为重要。在本次实验中，我很多次因为不够细心而犯了许多低级错误，好在 Android Studio 有自动检测错误的功能，使得我能够很容易地发现错误并给予改正，不然估计要花相当长的时间来 Debug。我始终相信，只有勇于坚持、小心谨慎，才能在 Android 开发中少走弯路并最终实现目标。