

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)	互联网
学号	15352438	姓名	郑培炜
电话	15989000640	Email	zhengpw3@mail2.sysu.edu.cn
开始日期	2017.9.25	完成日期	2017.9.26

一、 实验题目

了解和熟悉约束布局以及各元素的属性，按照要求设计一个 UI。

二、 实现内容

实现一个 Android 应用，界面呈现如下效果：

要求：

- 该界面为应用启动后看到的第一个界面
- 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**。

- 使用的布局和控件：**ConstraintLayout**、**TextView**、**EditText**、**Button**、**ImageView**、**RadioGroup**、**RadioButton**。



三、 实验结果

(1) 实验截图



(2) 实验步骤以及关键代码

- 标题的设置。

```
<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:text="@string/title"
    android:textAppearance="@android:style/TextAppearance.Material"
    android:textSize="20sp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

- Logo 图片：先将图片放入 **drawable** 文件夹。然后在 **ImageView** 里面引用。注意调整一下图片大小。

```
<ImageView
    android:id="@+id/imageView2"
    android:layout_width="146dp"
    android:layout_height="103dp"
    android:layout_marginTop="20dp"
    app:srcCompat="@drawable/sysu"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView2"
    android:contentDescription="@string/symbol" />
```

- 文本框，注意中心线和左边线的约束。

```
<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="20dp"
    android:text="@string/Student_id"
    android:textAppearance="@android:style/TextAppearance.Material"
    android:textSize="18sp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintBaseline_toBaselineOf="@+id/editText"
    app:layout_constraintBottom_toBottomOf="@+id/editText"
    app:layout_constraintTop_toTopOf="@+id/editText"
/>
```

- 文本输入框

```
<EditText
    android:id="@+id/editText"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:textSize="18sp"
    android:layout_marginEnd="20dp"
    android:layout_marginStart="30dp"
    android:layout_marginTop="20dp"
    android:ems="10"
    android:hint="@string/提示 1"
    android:inputType="number"
    app:layout_constraintLeft_toRightOf="@+id/textView5"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/imageView2" />
```

- RadioGroup 以及 RadioButton

```
<RadioGroup
    android:id="@+id/radioGroup2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:alwaysDrawnWithCache="false"
    android:checkedButton="@+id/radioButton4"
    android:orientation="horizontal"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/editText2"
    android:layout_marginTop="20dp" >

    <RadioButton
        android:id="@+id/radioButton4"
        android:layout_width="wrap_content"
```

```

        android:layout_height="wrap_content"
        android:layout_marginEnd="20dp"
        android:layout_weight="1"
        android:text="@string/student"
        android:textSize="18sp"
        app:layout_constraintRight_toLeftOf="@+id/radioButton3" />

```

(3) 实验遇到困难以及解决思路

- 文本框和文本输入框的水平距离无法控制。
一开始只是让文本输入框的左边对齐文本框的右边，并且设置了 **margin**，发现没用，两个框还是挤在一起了。
最后还是将文本输入框的 **layout_width** 属性设置为 **0dp**，正确设置 **margin** 后就可以正常显示。

- 文本框与文本输入框垂直方向没有中心对其。

解决方法 1：让文本框的 **baseline** 用文本输入框的 **baseline** 来约束，即

```
app:layout_constraintBaseline_toBaselineOf="@+id/editText"
```

解决方法 2：让文本输入框的上下边框用文本输入框的上下边框来约束，即

```
app:layout_constraintBottom_toBottomOf="@+id/editText"
app:layout_constraintTop_toTopOf="@+id/editText"
```

- 两个 **button** 无法水平居中且距离为 **10dp**。

解决方法 1：使用 **guideline**，将 **guideline** 竖直摆放且平分父容器。然后两个 **button** 分别用 **guideline** 来约束。

```

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

```

```

        android:layout_marginEnd="10dp"
        app:layout_constraintRight_toLeftOf="@+id/guideline"

```

解决方法 2：利用链条，将两个链条用 **packed** 的方式链接在一起，然后只需要让左边的 **button** 的左边对齐父容器，右边 **button** 的右边也对齐父容器，就可以水平居中。然后两个按钮的距离可以像上面的文本框控制距离一样来用。

```

app:layout_constraintHorizontal_chainStyle="packed"
app:layout_constraintRight_toLeftOf="@+id/button3"

```

四、 实验思考和感想

每种基本元素的属性虽然很多，但从本质上来说很多都是相似的，所以只需要记住某些特殊用途的功能，如链条等，其他功能都容易从 AS 的代码补全提示中找到答案的。不得不感叹 AS 的强大，虽然跑一次 AS 的 8G 内存就会被占用 7G 以上。

要做好一个优美的界面还是有一定难度的，之后要多练练其他类型的布局，早日做出好心仪的界面。

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。