

# 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：

年级	15	专业 ( 方向 )	移动信息工程
学号	15352063	姓名	成子晟
电话	13424169891	Email	929941797@qq.com
开始日期	9.23	完成日期	10.2

### 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； 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标题“中山大学学生信息系统”

```
<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textColor="@color/colorBlack"
    android:ems="10"
    android:textSize="20sp"
    android:text="中山大学学生信息系统"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"/>
```

使用 ConstraintLayout ，约束布局，规定其位置，这里将这个标题的左右以及上边与父容器的左右以及上边对其。最后实现的效果为对其顶部上边缘居中。此外还定义其字体颜色，在 colors.xml 中定义：

```
6 ■ <color name="colorBlack">#000</color>
```

## ②图片

```
<ImageView
    android:id="@+id/imageView"
    android:layout_width="124dp"
    android:layout_height="98dp"
    app:srcCompat="@mipmap/sysu"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView1"
    android:layout_marginTop="20dp"
    app:layout_constraintHorizontal_bias="0.502" />
```

将想要放入的图片文件先放置到该目录下：

lab\_3 > app > src > main > res > mipmap-hdpi > sysu.jpg

之后再使用 ImageView 空间直接在布局中添加该图片即可。

其中该图片的定位定位于全局左右的中间，并使用 layout\_constraintTop\_toBottomOf 将图片的上部对其标题“中山大学学生信息系统”的底部。

## ③学号输入

```
<TextView
    android:id="@+id/textView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textColor="@color/colorBlack"
    android:text="学号:"
    android:textSize="18sp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toLeftOf="@id/editText2"
    app:layout_constraintBottom_toBottomOf="@id/editText2"
    android:layout_marginBottom="0dp" />
```

```
<EditText
    android:id="@+id/editText2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@id/imageView"
    app:layout_constraintLeft_toRightOf="@+id/textView4"
    android:ems="10"
    android:layout_marginTop="20dp"
    android:layout_marginRight="20dp"
    android:textSize="18sp"
    android:inputType="number"
    android:hint="请输入学号"
/>
```

这里输入使用的控件可直接使用 Plain Text 直接得到，在这个输入的左侧有个小标题，将小标题的右侧与输入对其，下侧与输入下侧对其即可。在将该输入框上侧与 Image 下侧对其即可。此外这里使用了 margin，margin 代表了与某一侧相邻控件留出一定的位置。此外，输入框内的文字使用 hint，并定义输入框文字量小于等于 10。

#### ④密码输入

```
<TextView
    android:id="@+id/password"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textColor="@color/colorBlack"
    android:text="密码:"
    android:textSize="18sp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toLeftOf="@id/editText4"
    app:layout_constraintBottom_toBottomOf="@id/editText4"
/>

<EditText
    android:id="@+id/editText4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintLeft_toRightOf="@id/password"
    app:layout_constraintTop_toBottomOf="@id/editText2"
    android:layout_marginRight="20dp"
    android:ems="10"
    android:textColor="@color/colorGrey"
    android:inputType="textPassword"
    android:hint="请输入密码"
/>
```

密码输入模块和学号输入的实现几乎相同，定位等方面不多赘述。不同在与要将输入的模式，inputType 定义为 textPassword。

#### ⑤按钮

```
<RadioGroup
    android:id="@+id/radioGroup"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@id/editText4"
    android:layout_marginTop="20dp"
>
    <RadioButton
        android:id="@+id/radioButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginRight="10dp"
        android:text="学生"
        android:textSize="18sp"
        android:checked="true"
    />
    <RadioButton
        android:id="@+id/radioButton2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="教职工"
        android:textSize="18sp"
    />
</RadioGroup>
```

为了保证两个按钮对其且可统一移动，这里使用了 `radioGroup`，将两个 `RadioButton` 放置于一个 `Group` 中。之后只需要对这个 `Group` 进行定位等操作即可，方法与之前类似，不多赘述。

## ⑥登录与注册

```
<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginRight="5dp"
    android:layout_marginTop="20dp"
    android:background="@drawable/shapee"
    android:text="登录"

    android:textColor="@android:color/background_light"
    android:textColorHint="@color/colorWhite"
    android:textSize="18sp"
    app:layout_constraintRight_toLeftOf="@id/guideline"
    app:layout_constraintTop_toBottomOf="@id/radioGroup"
    tools:textColorHighlight="@android:color/white" />

<Button
    android:id="@+id/button3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="5dp"
    android:background="@drawable/shapee"
    android:layout_marginTop="20dp"

    android:text="注册"
    android:textColor="@android:color/background_light"
    android:textSize="18sp"
    app:layout_constraintLeft_toRightOf="@id/guideline"
    app:layout_constraintTop_toBottomOf="@id/radioGroup" />
```

这里使用了 `guideline` 的方法来保证两个按钮按照中心线对其。`Guideline` 的定义如下：

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

定义了一条名为 `guideline` 的，位置在屏幕中心，垂直的线。

我们的只需要将我们的登录按钮右侧对其该线的左边，注册按钮左侧对其该线的右侧就可以实现一个比较整齐的排版。

此外这里还需要实现这个登录按钮的圆角矩形化。这里要新定义一个背景框。在 `drawable` 中新建一个背景，如下：

```
shape
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <corners android:radius="10dp"></corners>
    <solid android:color="@color/colorBack"></solid>
</shape>
```

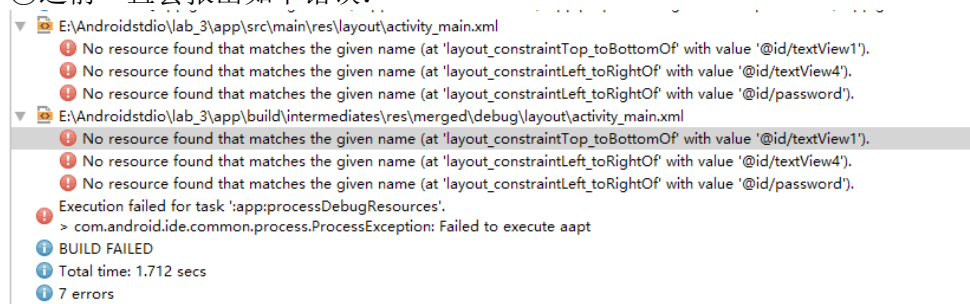
使用 `shape` 实现了圆角化。

### (3) 实验遇到困难以及解决思路

①开始写错了文件，本应该写在 activity\_main 的内容我写在了 conten\_main 里面了。最后选择重新建立了一个项目，解决了这个问题。

②在创建项目的时候选择错了安卓版本，兼容性只有 30%，导致控件无法正常的进行操作，最后找到了问题所在，修改了版本，兼容性到达了 98%以上，就没有问题了。

③之前一直会报出如下错误：



一直没有找到问题所在，后来在询问同学和 ta 之后才知道这些 id 名如果是第一次出现使用的话之前应该加上“+”号，好像这个问题 ta 在课上讲过而我没有好好听，真是不好意思。

## 四、 课后实验结果

## 五、 实验思考及感想

这次实验主要内容是创建一个 UI 界面，在界面的创建中主要使用了 TextView、EditText、Button、ImageView、RadioGroup、RadioButton 等等控件，布局主要是 ConstraintLayout。还有自定义颜色和背景等等。并不是十分难理解的东西，明白了怎么使用之后还是很容易实现的。希望之后的课程中还能学到更多。