



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：郑贵峰

年级	2015	专业 (方向)	移动信息工程
学号	15352258	姓名	马东辉
电话	13727027316	Email	3262568848@qq.com
开始日期	2017.9.23	完成日期	2017.10.01

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) 各控件要求如下：

- ① 要求只用一个 ConstraintLayout 实现整个布局；
- ② 标题字体大小为 20sp，与顶部距离 20dp，居中；
- ③ 图片与标题的间距为 20dp，居中；输入框整体距屏幕右间距 20dp，内容（包括提示内容）如图所示，内容字体大小 18sp；
- ④ 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；
- ⑤ 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；
- ⑥ 两个按钮整体居中，与上方控件间距 20dp，按钮简单间距 10dp，文字大小 18sp。
按钮背景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) 实验步骤以及关键代码

a) 事先熟悉 UI 界面实现方法，先通过可视化视图完成一个小的界面，然后观察 XML 代码中发生的变化，进一步体会各个代码的作用；

b) 认真阅读给出的实验文档，查阅有关链接，先思考各个控件的实现方法，接着用代码实现；

c) 编写代码：

① 标题的实现：

中山大学学生信息系统

定义标题的 id 为 “title”，设置标题的高度和宽度为 wrap_content，即由内容确定，字体的设置通过 textsize 来实现，设置为 20sp；通过设置 top、left、right 三个方向的约束来实现标题的居中；标题与顶部距离 20dp，通过设置 Layout_marginTop 的值来实现。

具体代码如下：

```
<TextView
    android:id="@+id/title"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:text="中山大学学生信息系统"
    android:textColor="@android:color/black"
    android:textColorLink="@android:color/black"
    android:textSize="20sp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

设置高度和宽度

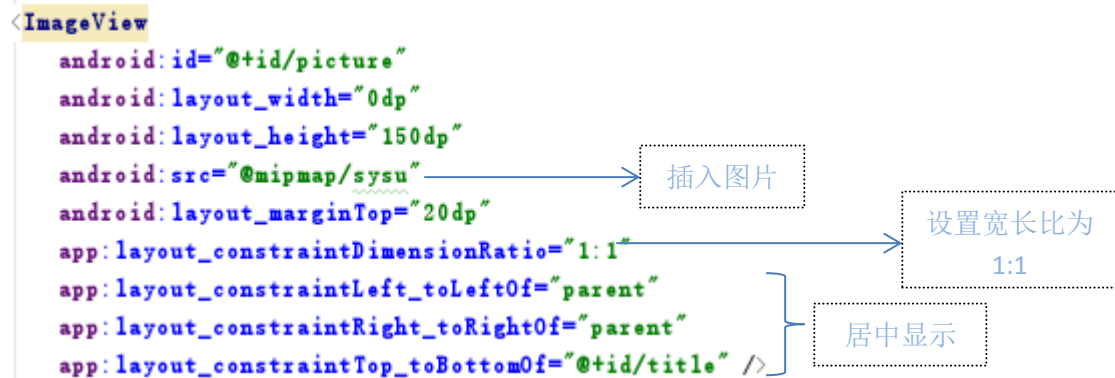
设置字体大小

通过设置约束使标题居中

② 插入图片：



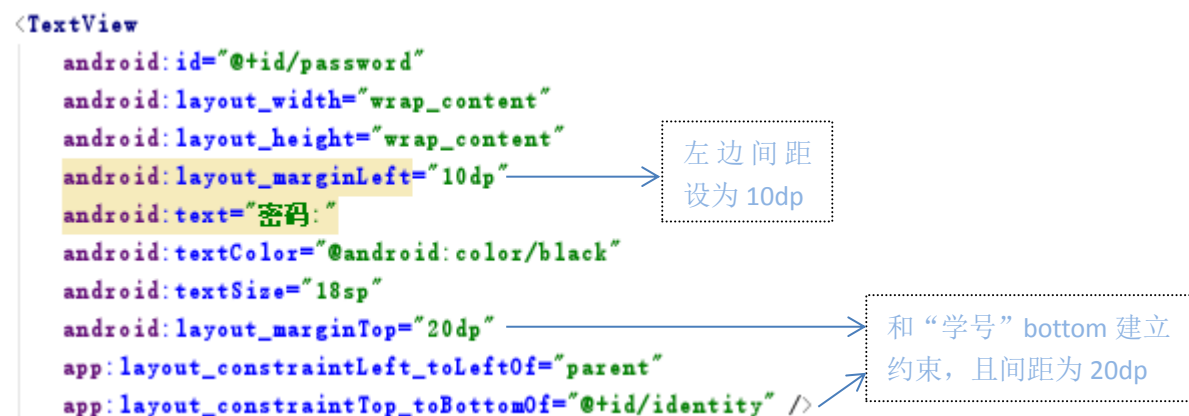
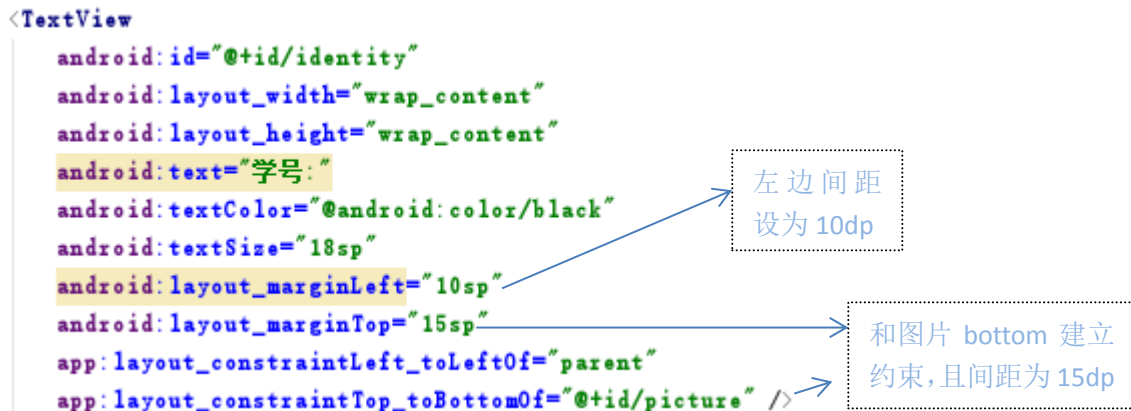
首先把要插入的图片放在 “app/res/mipmap ” 目录下，命名为 “sysu.png ”，然后通过 “android:src=” mipmap/sysu ” 插入图片；定义图片的 id 为 “picture”，由于图片太大，设置高度为 150dp 宽高比为 1：1，使其为合适的大小；图片为居中的，所以要在 top，left，right 方向设置约束，铜鼓 Layout_marginTop 设置与标题间距 20dp。具体代码如下：



③ “学号”、“密码” 的实现：



定义“学号”的id为“identity”，“密码”的id为“password”，两者左边都和父界面建立约束，“学号”的top和图片的bottom建立约束，“密码”的top和“学号”的bottom建立约束，两者间距为20dp，左边和父界面间距15dp，通过以上设置，就可以使其靠近父界面的左端位置，即上图所示。具体代码如下：



④ 学号、密码输入框的实现：



输入框为元素 `EditText` ,学号输入框的 id 为 "input1" ,密码输入框的 id 为 "input2" ;
提示文字由属性 `hint` 实现；由于学号只能输入数字，所以设置其输入属性为 `number` , 密码的输入属性为 `textPassword` , 其输入时会显示*；左边分别和 "学号：" 、 "密码：" 建立约束，学号输入框的 `top` 和图片的 `bottom` 建立约束，密码输入框的 `top` 和学号输入框的 `bottom` 建立约束，右边和右边框建立约束，且距离为 20dp。具体代码如下：

```
<EditText
    android:id="@+id/input1"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:hint="请输入学号"
    android:inputType="number"
    android:textColorHint="@android:color/darker_gray"
    android:textSize="18sp"
    android:layout_marginEnd="20dp"
    android:layout_marginLeft="25dp"
    android:layout_marginRight="20dp"
    android:layout_marginStart="25dp"
    app:layout_constraintLeft_toRightOf="@+id/identity"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/picture" />
```

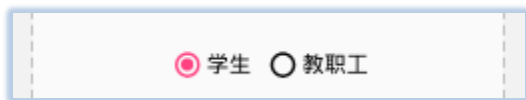
提示文字

输入文字的属

```
<EditText
    android:id="@+id/input2"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:hint="请输入密码"
    android:inputType="textPassword"
    android:textColorHint="@android:color/darker_gray"
    android:textSize="18sp"
    android:layout_marginEnd="20dp"
    android:layout_marginLeft="25dp"
    android:layout_marginRight="20dp"
    android:layout_marginStart="25dp"
    app:layout_constraintLeft_toRightOf="@+id/password"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/input1" />
```

和右边框的
距离 20dp

⑤ 两个单选按钮的实现：



两个单选按钮要放在元素 `RadioGroup` 里，`RadioGroup` 的 `id` 为 “`id0`”，单选按钮为元素 `RadioButton`，`id` 分别为 “`id1`” 和 “`id2`”；两个按钮是并排放置的，所以要将 `RadioGroup` 的 `orientation` 设为 “`horizontal`”；学生是默认的选项，通过 `checked=true` 实现；两个按钮的间距为 `10dp`，要先在 `id2` 里建立其 `left` 和 `id1 right` 的约束，然后设置 `layout_marginLeft` 的值为 `10dp` 即可。

```
<RadioGroup
    android:id="@+id/id0"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginTop="20dp"
    app:layout_constraintTop_toBottomOf="@+id/input2"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent">
    <RadioButton
        android:id="@+id/id1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:checked="true"
        android:text="学生"
        android:textColor="@android:color/black"
        android:textSize="18sp" />
    <RadioButton
        android:id="@+id/id2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="10dp"
        android:text="教职工"
        android:textColor="@android:color/black"
        android:textSize="18sp"
        app:layout_constraintLeft_toRightOf="@+id/id1" />
</RadioGroup>
```

RadioGroup 的方向设为横向

默认选项为学生

设置两个按钮之间的距离

⑥ 两个按钮的实现：



按钮是通过元素 “Button” 实现的，登录、注册按钮的 id 分别为 “button1” 和 “button2”；主要有两点需要注意，要实现两个按钮之间的距离为 10dp 以及居中显示，需要将约束类型设为 “packed”，按钮背景的实现需要另外配置：

a. 先在 drawable 目录下新建一个 “Drawable resource file” 命名为 sean，然后将元素类型改为 shape；

b. 设置颜色，现在 values 目录下的 colors.xml 文件中定义需要的颜色命名为 “colorBottom”，即 `<color name="colorBottom">#3F51B5</color>`，然后在 sean 中调用此颜色即可；

c. 设置圆角半径，圆角半径需要在 corners 元素里设置；

d. 设置背景边界和文字之间的间距，在 padding 里设置 top，bottom，left，right 的值即可，具体实现如下：

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

button1 的实现代码如下，button2 的代码不在此详述，与 button1 几乎相同。

```
<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="@drawable/sean"
    android:text="登录"
    android:textColor="@android:color/white"
    android:textSize="18sp"
    android:layout_marginTop="20dp"
    app:layout_constraintTop_toBottomOf="@+id/id0"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toLeftOf="@+id/button2"
    app:layout_constraintHorizontal_chainStyle="packed" />
```

背景设为创建的背景

设置横向约束的属性为 packed

d) 新建虚拟机，在虚拟机上运行代码，具体截图见下面。

(2) 实验截图

a) 输入学号，测试得知只能输入数字。

b) 输入密码，显示.....



四、课后实验结果

1. 为界面添加一个背景：



2. 点击按钮时，按钮背景颜色变成橙色：

在 drawable 目录下新建 2 个 drawable source file，分别命名为 selector.xml 和 sean3.xml，sean3 为定义一个橙色的背景，其他要求和 sean.xml 一样，selector.xml 也是一个背景，只不过是根据是否点击按钮来选择不同的背景，不点击选择蓝色背景，点击之后选择橙色背景。可在 selector 元素中定义两个 item 元素，通过 state_pressed 的值来选择不同的 drawable source file，值为 false（未点击）选择蓝色，值为 true（点击）选择橙色。具体实现如下：



```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:drawable="@drawable/sean" android:state_pressed="false"/>
    <item android:drawable="@drawable/sean3" android:state_pressed="true"/>
</selector>
```

五、实验思考及感想

Android 应用是一个或者多个 activities 组成的，activities 就是一个容器，我们这次设计的简单的 UI 就在 activities 里装着，从写代码的文件名 main_activities.xml 也可以知道。通过上课老师的讲解以及实验中自己的摸索，对 UI 有了比较科学的理解，其实 UI-user interface，就是用户界面，是应用软件的操作逻辑、人机交互、界面的整体设计，好的 UI 设计可以方便的实现信息内部形势与人类可接受形式之间的转换，而且还会让软件变得更加有个性和创意，让软件的操作变得舒适自由。所以 UI 界面很重要，它相当于一个门面。

从 UI 的组成来说，UI 就是使用 View 和 ViewGroup 对象构建而成，View 对象用

于在屏幕上绘制可供用户交互的内容,ViewGroup 对象用于储存其他 View 和 ViewGroup 对象,以便定义界面的布局。

通过这次实验深刻体会到安卓开发不能开始就埋下头去写代码,一定要熟悉相关文档,比如这次的U I 界面设计,虽然是个很简单的界面,但如果不熟悉XML 语言的代码结构和方法函数以及一些元素的相关属性是怎么设置的话,就会无法下手,或者说只能用可视化编程,但这样对U I 的整体理解程度会大大降低。

实验中也遇到了一些问题,其中的一个问题就是不看说明文档直接下手去做,所以一头雾水,不知如何进行下去。其他比较重要的问题如下:

- ✧ 登录、注册按钮不能够居中,一开始我是把他们放在一个线性布局里,结果是可以解决,这里线性布局的作用应该相当于使他们成为一个整体,但是实验要求只能使用约束布局,所以就想其他的办法。然后我用了 guideline,也可以实现,看到说明文档中链的属性中 packed 属性可以把两个元素连接成一个整体,试了一下也可以,问题解决;
- ✧ 由于约束布局是近几年才成为主流的,所以一些元素的属性只有在最新版的 ConstraintLayout 版本里才有,只有更新到最新版本才能使用,比如设置约束的属性 layout_constraintHorizontal_chaintyle, 1.0.0 版本里是没有的,所以我就下载最新版本,但是还是没有这个属性,经过老师讲解,原来是因为没有同步,需要在 build.gradle(Module:app)中改成最新版本。问题得到解决。