

# 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）	移动信息工程（移动互联网）
学号	15352049	姓名	陈新堉
电话	13670468594	Email	619156737@qq.com
开始日期	2017.09.27	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；

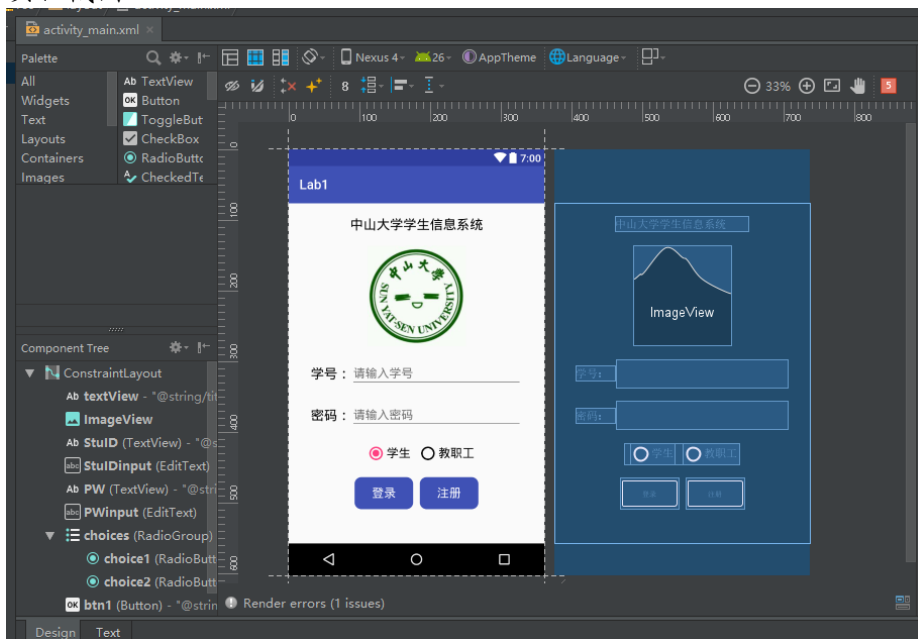
两个单选按钮整体居中，字体大小 `18sp`，间距 `10dp`，默认选中的按钮为第一个；

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

```
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.chan.lab1.MainActivity">
```

使用创建项目时就默认的 ConstraintLayout 布局。

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

使用 TextView 标签显示标题“中山大学学生信息”的标题。

其中 text 属性指出要显示的文字，textColor 属性指出字体颜色，textSize 指出字体大小，一般使用 sp 为单位。

layout\_weight, layout\_height 指出这个元素占据的范围，使用的值“wrap\_content”指这个元素的大小由其内容决定，够显示内容的大小就好。

layout\_constraintLeft\_toLeft 这些属性是用于指定元素受约束的对象，这里的“parent”是父元素也就是 ConstraintLayout 这个标签。

layout\_marginTop 指的是元素离在这个方向上的约束对象的距离，一般使用 dp 为单位。

```

<EditText
    android:id="@+id/StuIDinput"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintTop_toBottomOf="@+id/ImageView"
    app:layout_constraintLeft_toRightOf="@+id/StuID"
    app:layout_constraintRight_toRightOf="parent"
    android:layout_marginTop="20dp"
    android:layout_marginRight="20dp"
    android:ems="12"
    android:textSize="18sp"
    android:inputType="number"
    android:hint="@string/hintinput1"/>

```

使用 `EditText` 元素来显示输入框。

`ems` 属性是表示这个输入框预留了多大的空间来输入，属性值表示预留了可以输入这么多个字的空间。

`inputType` 属性表示输入框接受的信息类型，“number”指输入时弹出数字键盘。

`hint` 属性表示输入框在没有输入时灰色显示的提示信息。

```

<RadioGroup
    android:id="@+id/choices"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintTop_toBottomOf="@+id/PWininput"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    android:layout_marginTop="20dp">

```

单项选择的选项要包含在 `RadioGroup` 的标签里。

`orientation` 指定选项的排列是水平还是垂直的。

```

<RadioButton
    android:id="@+id/choice1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/choice1"
    android:textSize="18sp"
    app:layout_constraintRight_toLeftOf="@+id/choice2"
    android:layout_marginLeft="10dp"
    android:checked="true"/>

```

一个选项用 `RadioButton` 标签。

`checked` 属性为“true”表示这个选项会被预先选中。

```

<Button
    android:id="@+id/btn1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/signin"
    android:textColor="@color/white"
    android:textSize="18sp"
    android:paddingLeft="10dp"
    android:paddingRight="10dp"
    android:paddingTop="5dp"
    android:paddingBottom="5dp"

```

```

        android:background="@drawable/shape_corner"
        app:layout_constraintTop_toBottomOf="@+id/choices"
        app:layout_constraintRight_toLeftOf="@+id/btn2"
        app:layout_constraintLeft_toLeftOf="parent"
        android:layout_marginTop="20dp"
        app:layout_constraintHorizontal_chainStyle="packed"/>

```

按钮使用 Button 标签实现。

layout\_constraintHorizontal\_chainStyle 属性指定互相约束的两个元素之间构成的链的形式。默认是平均分布的，因此没有规定为 packed 时，两个按钮之间总会有距离，这个距离不会因为设置 margin 而变化。

background 通过在另一个 xml 文件里设置的形状来为按钮设置边框。shape\_corner 的文件代码如下：

```

<?xml version="1.0" encoding="utf-8"?>
<shape
    xmlns:android="http://schemas.android.com/apk/res/android">
    <solid android:color="@color/colorPrimary" />
    <corners android:topLeftRadius="10dp"
        android:topRightRadius="10dp"
        android:bottomRightRadius="10dp"
        android:bottomLeftRadius="10dp"/>
    <stroke android:width="0dp" android:color="#000000" />
</shape>

```

solid 标签的 color 属性指定了这个形状的填充颜色。

corners 标签的 Radius 属性设置了边角的半径大小。

stroke 则是对边框设置。

### (3) 实验遇到困难以及解决思路

1. 一开始不懂如何在 constraintlayout 里面设置控件的位置，于是直接在 design 界面移动空间，发现代码出现了 marginLeft 的属性，于是猜测这个属性就是设置控件和约束兑现边缘的距离的。
2. 输入框和前面的文本的位置不知道怎样让他们对齐好看写。后来 TA 的提醒，让文本对其输入框的上下边缘，这样在垂直方向就相对输入框居中了，比较好看。

## 四、 课后实验结果

根据老师建议，从 github 的 awesome-android-UI 项目中找了一个 android-flat-button 的控件，把原来的按钮换成比较有立体感的 FButton。

使用 github 上的这个代码前，需要在 app 文件夹的 build.gradle 文件里添加新的依赖。

```

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.android.support:appcompat-v7:26.+'
    compile 'com.android.support.constraint:constraint-layout:1.0.2'
    compile 'info.hoang8f:fbutton:1.0.5' //新添加的依赖
    testCompile 'junit:junit:4.12'
}

```

还要在布局文件 activity\_main 中的 constraintLayout 的属性中添加新的 xmlns 的属性。

```

xmlns:fbutton="http://schemas.android.com/apk/res-auto"

```

之后再把原来代码的 Button 元素改为 github 上面提供的 FButton 元素

```

<info.hoang8f.widget.FButton
    android:id="@+id/btn1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/signin"
    android:textColor="@android:color/white"
    android:textSize="18sp"
    app:layout_constraintTop_toBottomOf="@+id/choices"
    app:layout_constraintRight_toLeftOf="@+id/btn2"
    app:layout_constraintLeft_toLeftOf="parent"
    android:layout_marginTop="20dp"
    app:layout_constraintHorizontal_chainStyle="packed"
    android:paddingLeft="10dp"
    android:paddingRight="10dp"
    android:paddingTop="5dp"
    android:paddingBottom="5dp"
    fbutton:buttonColor="@color/colorPrimary"
    fbutton:shadowEnabled="true"/>

```

其他约束信息保持一致。

实现效果如下：（gif 图片另存）



## 五、 实验思考及感想

第一次的 Android 应用开发实验，做的事情比较简单。

主要是进行对 xml 代码的熟悉和对约束布局的熟悉。

一般来说布局元素放在外层，里面包含了各个要在界面上显示布局的元素。一般一个元素对应一个控件。

在约束布局里，元素通过和其他元素，布局元素的相对关系来决定元素所在位置。元素之间可以形成链，有四种分布形式。

android studio 中也可以在 design 页面直接拉动控件来得到控件位置。

布局文件一般比较冗长，但复制粘贴的部分居多。

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。