

中山大学移动信息工程学院本科生实验报告

课程名称：移动应用开发

任课教师：郑贵锋

一、实验题目

基本 UI 界面设计

二、实验内容

实现一个 Android 应用，界面如下图，要求如下：



1. 该界面为应用启动后看到的第一个该界面
2. 各控件的要求如下：
 - 1) 要求只用一个 `ConstraintLayout` 实现整个布局。
 - 2) 标题字体大小 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。
 - 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，按钮间的间距 10dp，文字大 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

(一) 实验截图：

1. 界面效果如下：



2. 虚拟机上效果如下:

Android Emulator - Pixel_API_26:5554



(二) 关键步骤:

1. 进入 design 界面将需要的 ConstraintLayout、TextView、EditText、Button、ImageView、RadioGroup、RadioButton 等控件拖入。
2. 按照要求间距对这些控件进行排布。
3. 按照需要的间隔对控件进行链接居中处理。

4. 修改部分控件属性，例如输入框，修改文字，更改颜色，修改 RadioButton 默认选项。
5. 增加一条辅助线使得按钮能够准确的居中。
6. 在 drawable 中增加一个 buttonstyle.xml 用来修改按钮的圆角和颜色。

(三) 实验中遇到的困难和解决思路：

1. Button 无法对齐，在调整界面大小之后会位置会移动，如果将其连到边界上又会移动，最后增加了一条辅助线位于 50% 固定，然后按钮链接到线上。
2. 当一个控件没有上 or 下, 左 or 右中的两条链住的时候控件在虚拟机上运行不能处于正常位置，解决方式全部暴力连上去然后调整即可
3. 实验中 RadioButton 上下排布，增加 android:orientation="horizontal" 解决

(四) 代码及界面：

1. 标题字体大小 20sp，与顶部距离 20dp，居中，实现方式如下：

```
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="0dp"
    android:layout_marginRight="0dp"
    android:layout_marginTop="20dp"
    android:text="@string/titles"
    android:textAlignment="center"
    android:textColor="@android:color/black"
    android:textSize="20sp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

2. 图片与标题的间距为 20dp，居中，图片设置为 src。background 会根据 ImageView 组件给定的长宽进行拉伸，而 src 就存放的是原图的大小，不会进行拉伸，这样图片不会变形。

```
<ImageView
    android:id="@+id/imageView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="0dp"
    android:layout_marginLeft="0dp"
    android:layout_marginRight="0dp"
    android:layout_marginTop="20dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView"
    app:layout_constraintVertical_bias="0.0"
    app:srcCompat="@mipmap/sysu"
    app:layout_constraintHorizontal_bias="0.5" />
```

3. 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。此处我们可以偷懒直接拖出控件中已经设置好的，这样就不需要额外添加属性，输入框前提示直接使用 TextView 并链到 EditText 上。

```
<EditText
    android:id="@+id/editText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="0dp"
    android:layout_marginLeft="0dp"
    android:layout_marginRight="20dp"
    android:layout_marginTop="20dp"
    android:ems="12"
    android:hint="@string/user_name_hit"
    android:inputType="number"
    android:selectAllOnFocus="false"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/imageView"
    app:layout_constraintVertical_bias="0.0" />
<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="0dp"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="0dp"
    android:layout_marginRight="0dp"
    android:text="@string/user_name"
    android:textColor="@android:color/black"
    android:textSize="18sp"
    app:layout_constraintBottom_toBottomOf="@+id/editText"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toTopOf="@+id/editText"
    app:layout_constraintVertical_bias="0.5"
    tools:layout_constraintBottom_creator="1"
    tools:layout_constraintTop_creator="1"
    app:layout_constraintRight_toLeftOf="@+id/editText"
    app:layout_constraintHorizontal_bias="0.0" />
```

4. 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。此处我们使用 RadioGroup 然后拖入两个 RadioButton。

```
<RadioGroup
    android:id="@+id/radioGroup"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```

        android:layout_marginBottom="0dp"
        android:layout_marginTop="20dp"
        android:checkedButton="@+id/radioButton1"
        android:orientation="horizontal"
        android:visibility="visible"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editText2"
        app:layout_constraintVertical_bias="0.0"
        tools:layout_constraintLeft_creator="1"
        tools:layout_constraintRight_creator="1">

        <RadioButton
            android:id="@+id/radioButton1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="10dp"
            android:layout_weight="1"
            android:text="@string/radio_check1" />

```

5. 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 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，按钮样式交由定义的 buttonstyle.xml 搞定，居中问题使用辅助线。

```

<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:layout_marginRight="5dp"
    android:layout_marginBottom="8dp"
    android:layout_marginLeft="0dp"
    android:background="@drawable/buttonstyle"
    android:text="@string/button_text1"
    android:textAlignment="center"
    android:textColor="@android:color/white"
    android:textSize="18sp"
    app:layout_constraintTop_toBottomOf="@+id/radioGroup"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintVertical_bias="0.0"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toLeftOf="@+id/guideline"
    app:layout_constraintHorizontal_bias="1.0" />

<android.support.constraint.Guideline
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/guideline"

```

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

四、课后实验结果

1. 上述实验中的 buttonstyle.xml 在课后实现，根据实验文档给出的教程我们很容易写出如下代码：

```
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <corners android:radius="10dp"/>
    <padding android:top="5dp"
        android:bottom="5dp"
        android:left="10dp"
        android:right="10dp"/>
    <solid
        android:color="#3f51b5"/>
</shape>
```

2. 之后我们需要把按钮的背景设为 buttonstyle.xml 即可
3. 将所有文字加入 string.xml，使用 @string/ 获得文字

```
<resources>
    <string name="app_name">Lab1</string>
    <string name="titles">中山大学学生信息系统</string>
    <string name="user_name_hit">请输入学号</string>
    <string name="password_hit">请输入密码</string>
    <string name="user_name">学号: </string>
    <string name="password">密码: </string>
    <string name="radio_check1">学生</string>
    <string name="radio_check2">教职工</string>
    <string name="button_text1">登录</string>
    <string name="button_text2">注册</string>
</resources>
```

4. 实验中的显示布局已经在项目创建时自动生成无需修改。

五、实验思考及感想

这次实验比较简单，只需要把各种控件拿出来组 UI 就可以了，主要就是注意下要求，没有什么过于复杂的地方，最多就是对齐比较麻烦，其他的东西实验文档已经很清楚了，照着实验文档进行实验即可，直接在可视界面对控件属性进行修改速度比使用代码修改会方便很多，同时在可视界面直接组控件，比去找代码也会方便，当然有的属性要翻一下的时候还是用代码比较快比如那个 radiogroup 的水平排布。