

# 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	2015 级	专业 ( 方向 )	软件工程(移动信息工程)
学号	15352033	姓名	陈黄胤
电话	18588741286	Email	chenhy98@mail2.sysu.edu.cn
开始日期	2017.10.12	完成日期	2017.10.16

### 一、 实验题目

事件处理

### 二、 实现内容

实现一个 Android 应用，界面呈现如下效果：

Lab2

中山大学学生信息系统



请输入学号

请输入密码

☒ 学生 ☐ 教职工

登录

注册

要求：

实现一个 Android 应用，界面呈现与实验一基本一致，要求：

- (1) 该界面为应用启动后看到的第一个界面
- (2) 输入学号和密码的控件要求用 `TextInputLayout` 实现
- (3) 点击图片，弹出对话框如下图：



点击“拍摄”选项，弹出 `Toast` 信息“您选择了[拍摄]”；

点击“从相册选择”选项，弹出 `Toast` 信息“您选择了[从相册选择]”；

点击“取消”按钮，弹出 `Toast` 信息“您选择了[取消]”。

- (4) 切换 `RadioButton` 的选项，弹出 `Snackbar` 提示“您选择了 xx”；  
例如从选项“学生”切换到选项“教职工”，则提示“您选择了教职工”；



点击 `Snackbar` 上的“确定”按钮，则弹出 `Toast` 信息“`Snackbar` 的

确定按钮被点击了”

(5) 点击登录按钮

依次判断学号是否为空，密码是否为空，用户名和密码是否正确（正确的学号和密码分别为“123456”，“6666”）；不正确则给出错误信息，如学号和密码都正确则提示“登陆成功”。

(6) 点击注册按钮

如果切换选项时，RadioButton选中的是“学生”，那么弹出Snackbar信息“学生注册功能尚未启用”，如果选中的是“教职工”，那么弹出Toast信息“教职工注册功能尚未启用”。

### 三、 课堂实验结果

(1) 实验截图



(2) 实验步骤以及关键代码

### I.设计布局

```

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/title_name"
    android:textSize="20sp"
    android:textStyle="bold"

    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    android:layout_marginTop="20dp"

    android:id="@+id/textView" />

<ImageView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/s"
    android:contentDescription="@string/title_name"

    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView"

    android:layout_marginTop="18dp"
    android:id="@+id/image" />

```

首先与 lab1 中一样，创建一个 TextView，其内容为“中山大学学生信息系统”，并且按要求设置 text 的格式。然后设置控件的布局，首先与 parent 建立约束，设置该控件在界面的居中，并设置与 parent 的顶部距离 20dp。

再创建一个 ImageView 控件，设置好与 parent 的约束，并让控件的顶部与上面的 textView 控件底部对齐，并设置好图片路径。

```

<android.support.design.widget.TextInputLayout
    android:layout_width="330dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    app:layout_constraintTop_toBottomOf="@+id/image"
    android:layout_marginLeft="8dp"
    app:layout_constraintLeft_toLeftOf="parent"
    android:layout_marginRight="8dp"
    app:layout_constraintRight_toRightOf="parent"
    android:layout_marginStart="8dp"
    android:layout_marginEnd="8dp"
    android:id="@+id/ed1"
>

    <EditText
        android:id="@+id/editText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="number"
        android:hint="@string/Id_hint" />
</android.support.design.widget.TextInputLayout>

```

再创建一个两个 `TextInputLayout` , 里面摆放一个 `EditText` 分别用于学号和密码的输入 , 同时设置好其布局。

```

<RadioGroup
    android:id="@+id/id0"
    android:layout_height="wrap_content"
    android:layout_width="wrap_content"
    android:layout_marginTop="20dp"
    android:orientation="horizontal"
    android:checkedButton="@+id/id1"

    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@id/password">

    <RadioButton
        android:id="@+id/id1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/Student"
        android:textSize="18sp"
        android:layout_marginEnd="10dp"
    />

    <RadioButton
        android:id="@+id/id2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/Staff"
        android:textSize="18sp"
        android:layout_marginStart="10dp"
    />
</RadioGroup>

```

再创建单选控件。首先创建一个单选组控件 RadioGroup，设置为横向排列 horizontal，然后设置好对 parent 的约束，保证居中，以及和上面输入框的约束条件。然后在里面再建立两个 RadioButton 控件，设置好彼此之间的距离约束

```

<Button
    android:layout_width="88dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:layout_marginEnd="10dp"
    android:text="@string/Sign_in"
    android:textColor="@color/White"
    android:textSize="18sp"
    android:background="@drawable/my_button"
    android:id="@+id/sign_in"

    app:layout_constraintHorizontal_chainStyle="packed"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/id0"
    app:layout_constraintRight_toLeftOf="@+id/sign_up"/>

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:layout_marginStart="10dp"
    android:text="@string/Sign_up"
    android:textColor="@color/White"
    android:textSize="18sp"
    android:background="@drawable/my_button"
    android:id="@+id/sign_up"

    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/id0"
    app:layout_constraintLeft_toRightOf="@+id/sign_in"/>

```

最后是两个按钮控件的设置，使用 packed chain 来关联两个按钮，设置好居中以及两者之间的距离。按照 packed chain 的要求设置好同时对 background 进行设置：

```

<?xml version="1.0" encoding="utf-8"?>
<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="@color/Blue" />
</shape>

```

实现了对按钮形状颜色的编辑



## II.设计 java 事件

首先设计图片上的事件，单击会出现对话选择框：

```
String[] LIST_VIEW={"拍摄","从相册选择"};
final AlertDialog.Builder alertDialog = new AlertDialog.Builder(MainActivity.this);
alertDialog.setTitle("上传头像").setItems(LIST_VIEW, new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        if (which == 0) Toast.makeText(getApplicationContext(), "你选择了[拍摄]", Toast.LENGTH_SHORT).show();
        else if (which == 1) Toast.makeText(getApplicationContext(), "你选择了[从相册选择]", Toast.LENGTH_SHORT).show();
    }
}).setNegativeButton("取消", new DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int which) {
        Toast.makeText(getApplicationContext(), "你选择了取消", Toast.LENGTH_SHORT).show();
    }
}).create();

ImageView Image = (ImageView) findViewById(R.id.image);
Image.setFocusable(true);
if (Image != null)
{
    Image.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View V) {
            alertDialog.show();
        }
    });
}
```

再设计选择单选后的 snackbar，每次选择都对 Selected 状态进行赋值

```

final Button Student = (Button)findViewById(R.id.id1);
final Button Teacher = (Button)findViewById(R.id.id2);

Student.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Student.setSelected(true);
        Teacher.setSelected(false);
        Snackbar.make(v, "你选择了学生", Toast.LENGTH_LONG)
            .setAction("确定", new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    Toast.makeText(MainActivity.this, "Snackbar的确定按钮被点击了", Toast.LENGTH_SHORT).show();
                }
            })
            .setActionTextColor(getResources().getColor(R.color.colorPrimary))
            .setDuration(5000)
            .show();
    }
});

Teacher.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Teacher.setSelected(true);
        Student.setSelected(false);
        Snackbar.make(v, "你选择了教职工", Toast.LENGTH_LONG)
            .setAction("确定", new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    Toast.makeText(MainActivity.this, "Snackbar的确定按钮被点击了", Toast.LENGTH_SHORT).show();
                }
            })
            .setActionTextColor(getResources().getColor(R.color.colorPrimary))
            .setDuration(5000)
            .show();
    }
});

```

再设计登录和注册的事件：

```

final TextInputLayout ID = (TextInputLayout)findViewById(R.id.ed1);
final EditText IDedt = ID.getText();
final TextInputLayout PASSWORD = (TextInputLayout)findViewById(R.id.password);
final EditText PASSedt = PASSWORD.getText();

Button Sign_in = (Button)findViewById(R.id.sign_in);
Button Sign_up = (Button)findViewById(R.id.sign_up);

Sign_in.setOnClickListener(new View.OnClickListener(){
    @Override
    public void onClick(View V){
        if(IDedt.getText().toString().length() == 0){
            ID.setErrorEnabled(true);
            ID.setError("学号不能为空");
            PASSWORD.setErrorEnabled(false);
        }
        else if(PASSedt.getText().toString().length() == 0){
            PASSWORD.setErrorEnabled(true);
            PASSWORD.setError("密码不能为空");
            ID.setErrorEnabled(false);
        }
        else{
            String Res;
            ID.setErrorEnabled(false);
            PASSWORD.setErrorEnabled(false);
            if(IDedt.getText().toString().equals("123456789") && PASSedt.getText().toString().equals("123456789")){
                Res = "登录成功";
            }
            else
                Res = "登录失败";
            Snackbar.make(V, Res, Toast.LENGTH_LONG)
                .setAction("确定", new View.OnClickListener(){
                    @Override
                    public void onClick(View view){
                        Toast.makeText(MainActivity.this, "Snackbar的确定按钮被点击了", Toast.LENGTH_SHORT).show();
                    }
                })
                .setActionTextColor(getResources().getColor(R.color.colorPrimary))
                .setDuration(5000)
                .show();
        }
    }
});

```

```

Sign_up.setOnClickListener(new View.OnClickListener(){
    @Override
    public void onClick(View v){
        String Res="";
        if(Teacher.isSelected()) Res="教职工";
        else if (Student.isSelected()) Res="学生";
        Snackbar.make(v, Res+"注册功能未启用", Toast.LENGTH_LONG)
            .setAction("确定", new View.OnClickListener(){
                @Override
                public void onClick(View view){
                    Toast.makeText(MainActivity.this, "Snackbar的确定按钮被点击了", Toast.LENGTH_SHORT).show();
                }
            })
            .setActionTextColor(getResources().getColor(R.color.colorPrimary))
            .setDuration(5000)
            .show();
    }
});

```

### (3) 实验遇到困难以及解决思路

实现 java 的时候遇到了很多困难，因为是第一次写，很多 android 的 java 接口都不熟悉，查了很多资料，修正过很多次才实现了实验目标。比如对单选按钮的状态进行赋值，最后的注册时要用到单选选了哪一个这个条件。最终利用 setSelected 这个属性来确定状态，从而实现了单选输出 snackbar 时的状态区分。

#### 四、 课后实验结果













## 五、 实验思考及感想

初次接触到 Android Studio 并初步接触到 APP 的布局设置，难免会遇到各种各样的问题。而遇到问题应该多查阅官方手册，因为很多问题都能在官方的使用手册中找到详细的解决方法。虽然 Google 为了降低 Android APP 开发的门槛，在 Android Studio 中加入了大量的可视化操作，但我觉得不仅要理解原理，更应该有读写相关代码的原理，不能完全依赖于这些可视化操作。对于 java 的事件处理，很多接口函数都不够熟悉，在初学阶段需要经常翻阅资料，设计的同时可以利用 Android Studio 的模拟器来测试代码的正确性和功能性。