

# 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 )	移动互联网
学号	15352055	姓名	陈泽伟
电话	13727023101	Email	1520414789@qq.com
开始日期	2017-10-10	完成日期	2017-10-16

### 一、 实验题目

事件处理

### 二、 实验内容



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”；

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”，如图：



#### (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

①按照上一次的实验内容，写出相同的代码，这里我直接 copy 了一份上一次的代码并将文件重命名为 lab2。

②对实验代码进行调整，主要包括：[1]去掉上一次实验的两个 TextView 控件，这两个 TextView 控件分别用来显示“学号”和“密码”，而本次实验不需要，所以直接去掉；[2]为上一次实验的两个 EditText 控件分别添加一个 TextInputLayout 控件，该控件用于接受用户输入，与 EditText 控件 配合使用有更好的用户体验，TextInputLayout 集合了输入提示，报错等功能，并且自带动画效果；[3]修改界面布局，使得每个控件按照正常的方式摆放。

关键代码如下：

```
<android.support.design.widget.TextInputLayout
    android:id="@+id/Input_SID"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:hint="@string/SID_hint"
    android:layout_marginTop="20dp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/image_SYSU" >
```

这里比较特殊的地方在于“android:hint= “@string/SID\_hint””，这里用了 TextInputLayout 控件的输入提醒的动画功能，即当输入框获得焦点时，提示信息会上漂到输入框上面继续显示。两个 TextInputLayout 控件代码基本一样，这里就不一一说明了。

③自定义对话框形式并编写 ImageView 控件的 java 代码，使其在被点击后能够弹出正确

形式的对话框。

关键代码如下：

```
final AlertDialog.Builder alertDialog = new AlertDialog.Builder(this);
alertDialog.setTitle(R.string.head_portrait);
alertDialog.setItems(R.array.get_photo,
    new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
            if(i == 0) {
                Toast.makeText(getApplicationContext(), R.string.take_photo_click,
                    Toast.LENGTH_SHORT).show();
            }
            else {
                Toast.makeText(getApplicationContext(), R.string.photo_album_click,
                    Toast.LENGTH_SHORT).show();
            }
        }
    });
alertDialog.setNegativeButton(R.string.cancel,
    new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
            Toast.makeText(getApplicationContext(), R.string.cancel_click,
                Toast.LENGTH_SHORT).show();
        }
    });
alertDialog.create();
```

在这里我们定义了一个对话框，调用 setTitle 函数对对话框标题进行设定，然后调用 setItem 函数，显示对话框的两个内容，其中每个内容被点击后会触发相应的功能，这里是弹出相应的信息，接着调用 setNegativeButton 函数为该对话框添加取消按钮并对其被点击后触发的功能进行设置，这里同样是弹出相应的信息，最后调用 create 函数创建该对话框。

```
ImageView ima = (ImageView) findViewById(R.id.image_SFSD);
if (ima != null) {
    ima.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) { alertDialog.show(); }
    });
}
```

这里编写 ImageView 控件的 java 代码，首先判断该 ImageView 控件是否被点击，如果被点击，则调用 alertDialog 控件的 show 函数，即将对话框的内容显示到屏幕上（注意：前面我们只是创建了该对话框，并没有让它显示出来）。

④为 RadioGroup 控件添加关联的 java 代码，使其能够实现切换 RadioButton 控件的选项时，弹出相应的 Snackbar 提示信息。

关键代码如下：

```

class myaffirm implements View.OnClickListener{
    @Override
    public void onClick(View v) {
        Toast.makeText(getApplicationContext(), R.string. affirm_click, Toast.LENGTH_SHORT).show();
    }
}

RadioGroup RG = (RadioGroup) findViewById(R.id. group_occupation);
RG.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(RadioGroup radioGroup, int i) {
        Snackbar snackbar;
        if(i == R.id. button_student) {
            snackbar = Snackbar.make(radioGroup, R.string. student_click, Snackbar.LENGTH_SHORT);
        }
        else{
            snackbar = Snackbar.make(radioGroup, R.string. teacher_click, Snackbar.LENGTH_SHORT);
        }
        snackbar.setAction(R.string. affirm, new myaffirm());
        snackbar.show();
    }
});

```

因为这里弹出的 Snackbar 提示信息中包含一个确定按钮，该确定按钮被点击时又会触发一个事件，这里同样是弹出一个信息，所以我首先编写了一个 myffirm 类，该类用于实现 Snackbar 提示信息的确按钮被点击这一事件，然后编写了一个 RadioGroup 监视器用于监视 RadioGroup 的状态，最后，根据当前 RadioGroup 的状态（哪一个选项被选中）进行相应的操作，这里都是弹出 Snackbar 提示信息，不过信息内容不一样。

⑤对登录按钮进行 java 代码关联，使其能够在被点击后根据 TextInputLayout 控件内容进行相应操作。

关键代码如下：

```

Button sign_in = (Button) findViewById(R.id. button_sign_in);
final TextInputLayout input_SID = (TextInputLayout) findViewById(R.id. Input_SID);
final TextInputLayout input_password = (TextInputLayout) findViewById(R.id. Input_password);

```

首先我先创建了一个 Button 控件监视器和两个 TextInputLayout 控件监视器



```

sign_in.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        final String student_ID = input_SID.getText().toString();
        final String password = input_password.getText().toString();
        if (TextUtils.isEmpty(student_ID)) {
            input_SID.setEnabled(true);
            input_SID.setError("学号不能为空");
        }
        else if (TextUtils.isEmpty(password)) {
            input_SID.setError("");
            input_SID.setEnabled(false);
            input_password.setEnabled(true);
            input_password.setError("密码不能为空");
        }
        else {
            input_SID.setEnabled(false);
            input_password.setEnabled(false);
            if (student_ID.equals("123456") && password.equals("6666")) {
                Snackbar.make(view, R.string.success, Snackbar.LENGTH_SHORT).setAction(R.string.affirm, new MyAffirm()).show();
            }
            else {
                Snackbar.make(view, R.string.error, Snackbar.LENGTH_SHORT).setAction(R.string.affirm, new MyAffirm()).show();
            }
        }
    }
});

```

然后创建两个 string 用于存放 TextInputLayout 控件的输入内容，最后根据输入内容作出相应的应答，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为注册按钮编写 java 程序代码，使其能够根据 RadioButton 控件选中的内容给出相应的提示。

关键代码如下：

```

Button sign_up = (Button) findViewById(R.id.button_sign_up);
sign_up.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        RadioButton Rbs = (RadioButton) findViewById(R.id.button_student);
        if (Rbs.isChecked()) {
            Snackbar.make(view, R.string.student_sign_up, Snackbar.LENGTH_SHORT).show();
        }
        else {
            Snackbar.make(view, R.string.teacher_sign_up, Snackbar.LENGTH_SHORT).show();
        }
    }
});

```

写到这里代码就很容易理解了，直接写两个监视器用于监视注册按钮和 RadioGroup 中的其中一个按钮，然后调用 isChecked 函数判断所监视的 RadioGroup 按钮有没有被选中，最后根据不同结果做出响应就可以了。

到了这里，整个实验也就基本完成了，撒花撒花。。。

### (3) 实验遇到困难以及解决思路

①第一个难点是自定义对话框。由于系统所给的对话框中只能显示一个内容，而本次实验有两个内容需要显示，所以实现起来有点难，我也是查了很多资料，才最终知道要调用 `setItem` 函数来显示 array 内容的。

②第二个难点是关于 `RadioGroup` 中哪个选项被选择的问题，由于对 Android 开发不是很了解，所以我一开始并不知道可以直接使用 `if` 条件语句判断，也是通过查了大量资料才最终解决的问题。

③第三个难点是 java 字符串的问题，虽然 java 中字符串的函数用法大部分和 C++ 的相同，但是 java 在声明字符串时是用 `String`，这里的 S 是大写，而在 C++ 中是小写。在本次实验中，我被这个坑了。

## 四、 课后实验结果

暂时没有扩展功能。

## 五、 实验思考及感想

本次实验总体来说还是存在一定难度的，首先对于我来说，我之前没有学过 java 语言，没有任何的语言基础，很多在 C++ 里通用的函数到了这里都不能用，语法结构也有一定差别，这些都一定程度地增加了实验的难度。其次，本次实验用到了很多新函数，这些都是之前没有接触过的，有时为了查询一个功能该用哪个函数实现就需要查看大量的资料。

经过本次实验，我也意识到要成为一个优秀的 Android 开发者，耐心和细心尤为重要。在本次实验中，我很多次因为不够细心而犯了许多低级错误，由于 Android Studio 没有 Debug 功能，所以我便使用 `Toast` 来显示一些信息，这样就相当于 Debug 了。我始终相信，只有勇于坚持、小心谨慎，才能在 Android 开发中少走弯路并最终实现目标。