

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：

年级	15352272	专业（方向）	软件工程（移动信息工程）
学号	15352272	姓名	彭国栋
电话	15626066058	Email	578291308@qq.com
开始日期		完成日期	

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，弹出对话框如左图：

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”，如图



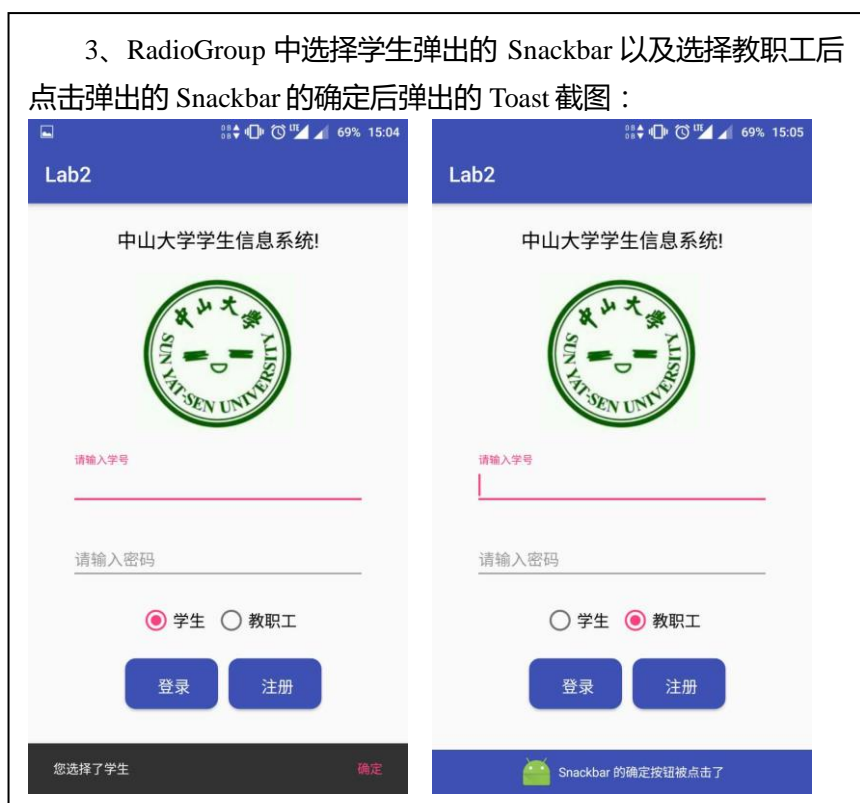
(6) 点击注册按钮

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”。

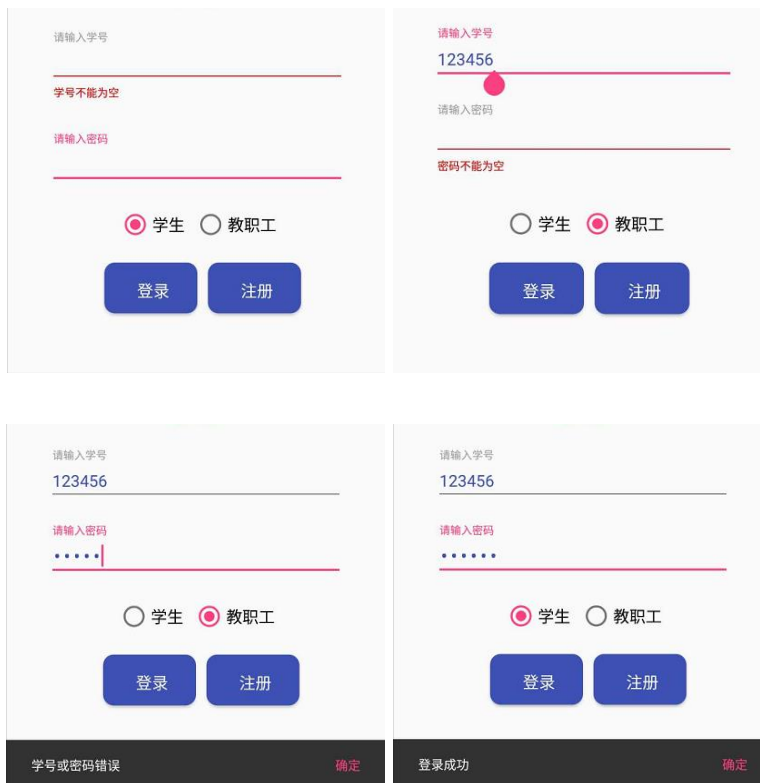
如果选中的是“教职工”，那么弹出 Toast 信息“教职工注册功能尚未启用”。

三、 课堂实验结果

(1) 实验截图



4、不输入学号、不输入密码、输入错误密码、输入正确密码的界面：



5、点击注册按钮后：



(2) 实验步骤以及关键代码

1、修改布局文件，由于仅仅只是在输入学号和密码的文本框变成了 `TextInputLayout`，而 `TextInputLayout` 是在 `android.support.design` 中，所以首先我们需要在 `build.gradle` 中加入：

```
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-v7:25.3.1'
    compile 'com.android.support.constraint:constraint-layout:1.0.2'
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:design:25.3.1'
}
```

而这个时候在 `xml` 布局文件中就能够使用 `TextInputLayout` 了：

```
<android.support.design.widget.TextInputLayout
    android:id="@+id/studentID_EditText_Wrapper"
    android:layout_width="280dp"
    android:layout_height="wrap_content"
    app:layout_constraintTop_toBottomOf="@+id/imagOfSysu"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    android:layout_marginTop="20dp"
    android:layout_marginRight="20dp"
    android:layout_marginLeft="20dp"
    >
    <EditText
        android:id="@+id/studentID_EditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textColor="@color/colorPrimary"
        android:textSize="18sp"
        android:inputType="number"
        android:hint="请输入学号"
    />
</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout
    android:id="@+id/studentPassword_EditText_Wrapper"
    android:layout_width="280dp"
    android:layout_height="wrap_content"
    app:layout_constraintTop_toBottomOf="@id/studentID_EditText_Wrapper"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    android:layout_marginTop="20dp"
    android:layout_marginRight="20dp"
    android:layout_marginLeft="20dp"
    >
    <EditText
        android:id="@+id/studentPassword_EditText"
        android:layout_width="280dp"
        android:layout_height="wrap_content"
        android:textColor="@color/colorPrimary"
        android:textSize="18sp"
        android:inputType="textPassword"
        android:hint="请输入密码"
    />
</android.support.design.widget.TextInputLayout>
```

2、根据对实验文档的分析，我们可以发现是对四个组件的操作：

- ①、图像 ImageView 的点击操作。
- ②、按钮组 RadioGroup 的点击操作。
- ③、登录按钮的点击操作。
- ④、注册按钮的点击操作。

那么在 Java 文件中，只需要分别对这几个部件进行相应的监听器绑定，就可以实现上述功能。首先看一下变量声明：

```
ImageView img;
RadioGroup Rg;
Boolean isStudent;
Button loginBt;
Button signinBt;
TextInputLayout textInputLayout_studentid;
TextInputLayout textInputLayout_studentpassword;
EditText studentid_Et;
EditText studentpassword_Et;
```

(变量定义)

以及上述变量分别绑定的按钮为：

```
img=(ImageView) findViewById(R.id.imagOfSysu); /*中山大学图片*/
Rg=(RadioGroup) findViewById(R.id.studentNature); /*RadioGroup组*/
isStudent=true; /*判定Rg选定的是否为学生*/
loginBt=(Button) findViewById(R.id.loginButton); /*登录按钮*/
signinBt=(Button) findViewById(R.id.signinButton); /*注册按钮*/
textInputLayout_studentid=(TextInputLayout) findViewById(R.id.studentID_EditText_Wrapper); /*TextInputLayout类型的学号框*/
textInputLayout_studentpassword=(TextInputLayout) findViewById(R.id.studentPassword_EditText_Wrapper); /*TextInputLayout类型的密码框*/
studentid_Et=(EditText) findViewById(R.id.studentID_EditText); /*学号文本输入框*/
studentpassword_Et=(EditText) findViewById(R.id.studentPassword_EditText); /*密码文本输入框*/
```

①、图像 ImageView 的点击操作。

```
final String content[]=new String[]{"拍摄","从相册中选取"};
img.setOnClickListener(new View.OnClickListener(){/*点击的监听器*/
    @Override
    public void onClick(View view)
    {
        AlertDialog.Builder bd=new AlertDialog.Builder(MainActivity.this);/*新建对话框*/
        bd.setTitle("上传头像");/*设置标题*/

        /*设置拍摄，从相册中选取以及取消按钮还有点击后弹出的Toast指示*/
        bd.setItems(content, new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface,int choice) {
                Toast.makeText(getApplicationContext(), "您选择了[" + content[choice] + "]", Toast.LENGTH_SHORT)
                    .show();
            }
        })
        .setNegativeButton("取消",new DialogInterface.OnClickListener()
        {
            @Override
            public void onClick(DialogInterface dialogInterface1,int check1){
                Toast.makeText(getApplicationContext(),"您选择了取消",Toast.LENGTH_SHORT)
                    .show();
            }
        });
        bd.create().show();/*创建对话框*/
    }
});
```

②、按钮组 RadioGroup 的点击操作。

```
Rg.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {
    @Override /* 检查按钮 */
    public void onCheckedChanged(RadioGroup mgroup, @IdRes int checkedId) {
        String student_or_teacherMessage;
        if(checkedId==R.id.studentButton) /* 选择的为学生时 */ {student_or_teacherMessage="您选择了学生"; isStudent=true;}
        else {student_or_teacherMessage="您选择了教职工"; isStudent=false;}
        Snackbar.make(mgroup.getRootView(), student_or_teacherMessage, Snackbar.LENGTH_SHORT) /* 弹出snackbar */
            .setAction("确定", new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    Toast.makeText(getApplicationContext(), "Snackbar 的确定按钮被点击了", Toast.LENGTH_SHORT)
                        .show(); /* snackbar的确定被点击时候 */
                }
            })
            .show();
    }
});
```

③、登录按钮的点击操作。

```
loginBt.setOnClickListener(new View.OnClickListener() /* 登录按钮的点击监听器 */
{
    @Override
    public void onClick(View v)
    {
        String studentId=studentid_Et.getText().toString(); /* 获取学号 */
        String studentPassword=studentpassword_Et.getText().toString(); /* 获取密码 */
        if(studentId.length()==0) /* 学号长度为0即为空 */
        {
            textInputLayout_studentid.setError("学号不能为空");
            textInputLayout_studentid.setErrorEnabled(true);
        }
        else if(studentPassword.length()==0) /* 密码长度为0即为空 */
        {
            textInputLayout_studentpassword.setError("密码不能为空");
            textInputLayout_studentpassword.setErrorEnabled(true);
        }
        else if(studentId.equals("123456") && studentPassword.equals("666666")) /* 输入正确的账号密码 */
        {
            textInputLayout_studentid.setErrorEnabled(false);
            textInputLayout_studentpassword.setErrorEnabled(false);
            Snackbar.make(v.getRootView(), "登录成功", Snackbar.LENGTH_SHORT) /* 弹出Snackbar */
                .setAction("确定", new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        Toast.makeText(getApplicationContext(), "Snackbar 的确定按钮被点击了", Toast.LENGTH_SHORT).show();
                    }
                })
                .show();
        }
        else /* 密码错误的情况 */
        {
            textInputLayout_studentid.setErrorEnabled(false);
            textInputLayout_studentpassword.setErrorEnabled(false);
            Snackbar.make(v.getRootView(), "学号或密码错误", Snackbar.LENGTH_SHORT)
                .setAction("确定", new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        Toast.makeText(getApplicationContext(), "Snackbar 的确定按钮被点击了", Toast.LENGTH_SHORT).show();
                    }
                })
                .show();
        }
    }
});
```

④、注册按钮的点击操作。

```
signinBt.setOnClickListener(new View.OnClickListener() /* 注册按钮 */
{
    @Override
    public void onClick(View v) {
        if (isStudent == true) /* 如果RadioGroup选中的是学生，那么就弹出学生注册功能未启用的Snackbar */
            Snackbar.make(v.getRootView(), "学生注册功能尚未启用", Snackbar.LENGTH_SHORT)
                .setAction("确定", new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        Toast.makeText(getApplicationContext(), "Snackbar 的确定按钮被点击了", Toast.LENGTH_SHORT).show();
                    }
                })
                .show();
        else /* 否则弹出教职工注册功能未启用的Toast */
            Toast.makeText(getApplicationContext(), "教职工注册功能尚未启用", Toast.LENGTH_SHORT).show();
    }
});
```

(3) 实验遇到困难以及解决思路

不同于第一次实验，上一次实验是只存在一个空壳，而这次实验在原有的基础上，给一些按钮

图片设置了一些动作，以点击操作为主。在实验的过程中，最大的障碍应该就是对 java 语法的不熟悉和不清楚，导致在实验的过程中总是出现错误，例如在 `setOnClickListener` 方法调用的时候一开始以为是将其作为一个函数再写实现方法，`setOnClickListener() new View....{}`，而实际上应该将 `new View...{}` 的部分作为参数放在 `setOnClickListener` 的括号里面，即 `setOnClickListener(new View....{ })`，这样才是正确的方式。最主要的解决方式还是通过结合 PPT 和实验文档，如果仍然存在不理解的部分就去网上寻找一些类似的例子来提升自己对于语法的熟悉程度，最终才得以完成实验。

四、 课后实验结果

如实验结果所示。

五、 实验思考及感想

感觉只有通过实践才能更好的对新的知识进行掌握，对于 java 语言其实还是比较陌生的，而实现这次实验恰恰能够帮助我们对于 java 有一定的了解，感觉对比起 C++，java 更多的是借助系统的一些函数，从而方便了我们的编程，但感觉由于没有系统学习过 java，所以还需要再多一些练习才能更好地运用 java。