

# 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	大三	专业 ( 方向 )	移动互联网
学号	15352048	姓名	陈潇
电话	13727022947	Email	616131435@qq.com
开始日期	2017.10.10	完成日期	2017.10.16

### 一、 实验题目

1. 了解 Android 编程基础
2. 熟悉 ImageView、Button、RadioButton 等基本控件，能够处理这些控件的基本事件
3. 学会弹出基本的对话框，能够定制对话框中的内容，能对确定和取消按钮的事件做处理

### 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



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” 4/10

(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

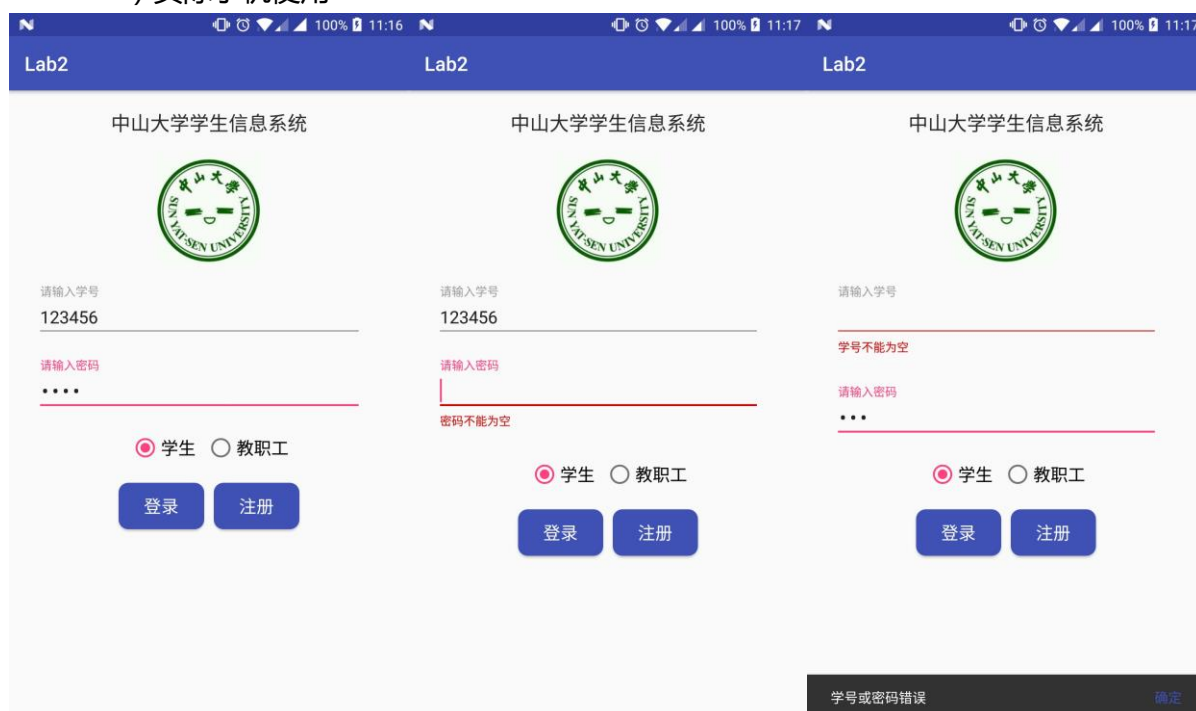
如果切换选项时，RadioButton选中的是“学生”，那么弹出Snackbar信息“学生注册功能尚未启用”，如果选中的是“教职工”，那么弹出Toast信息“教职工注册功能尚未启用”。

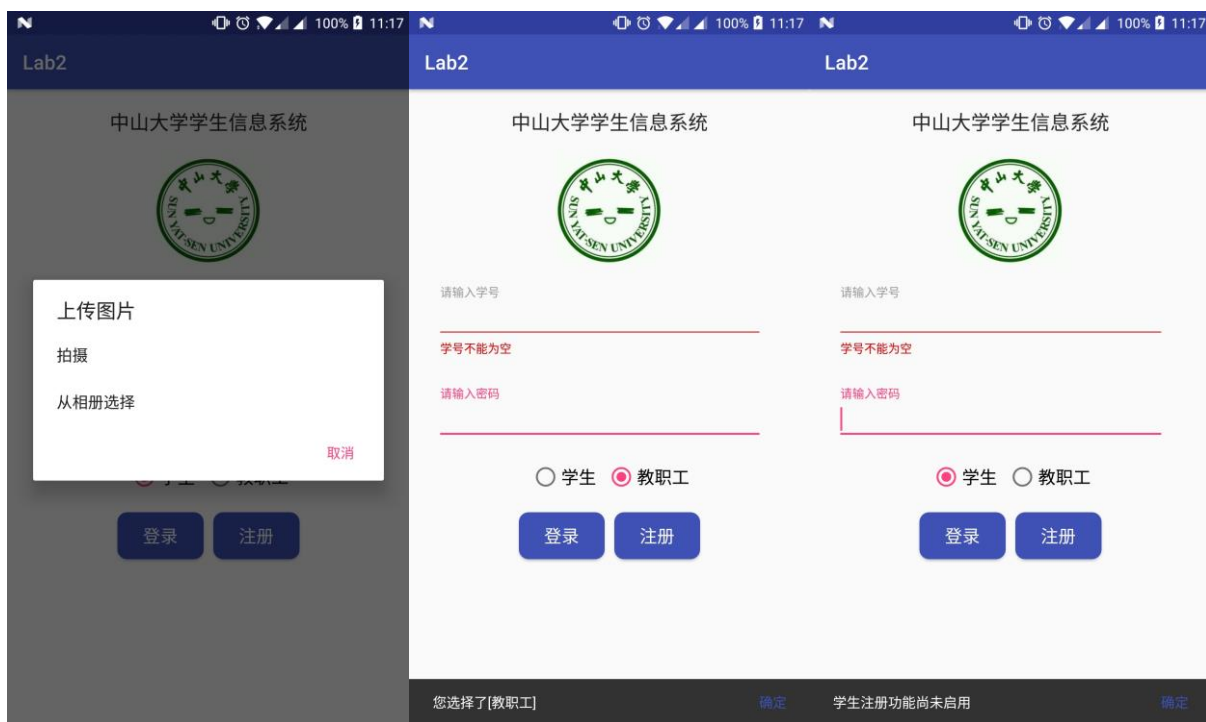
### 三、 课堂实验结果

(1) 实验截图

1) 使用 Android Studio 模拟的结果 略

2) 实际手机使用





## (2) 实验步骤以及关键代码

### 1) 修改界面布局 xml 文件。

```

39  <android.support.design.widget.TextInputLayout
40      android:id="@+id/editNum"
41      android:layout_width="wrap_content"
42      android:layout_height="wrap_content"
43      app:layout_constraintTop_toBottomOf="@+id/imageView"
44      app:layout_constraintRight_toRightOf="parent"
45      app:layout_constraintLeft_toLeftOf="parent"
46      android:layout_marginRight="20dp"
47      android:layout_marginTop="20dp">
48      <EditText
49          android:layout_width="wrap_content"
50          android:layout_height="wrap_content"
51          android:inputType="number"
52          android:hint="请输入学号"
53          android:textSize="18sp"
54          android:ems="16" />
55  </android.support.design.widget.TextInputLayout>

```

(3) 因为本次实验我们需要用到更新的 TextInputLayout 控件，该控件相当于是 EditText 的增强版，所以需要重新编写，大致步骤是将 TextInputLayout 完全包含在 EditText 外部，并且把 EditText 的原本约束写在 TextInputLayout 里包括 id。并且由于新的实验需要，我们要把输入文本框和原本的“学号”“密码”这两个文本的约束关系取消，并删除这两个文本控件。

2) 主要内容需要在 MainActivity 中编写。在对每一个需要监听的事件进行编写前，需要先实例化控件对象。通过 findViewById 函数实现。

ImageView 需要实现点击后弹出对话框，在对话框中进行选择操作并在 Snackbar 中弹出提示。

对话框通过实例化 AlertDialog.Builder 的实例实现，通过 setTitle 函数设置标题，再通过 setItems 函数设置事件监听，事件的内容使用 onClick 函数，并在其中调用 Toast.makeText 函数进行编写，即可。

最后通过 setNegativeButton 函数 Toast.makeText 函数显示取消按钮和显示内容即可。

最后还要在实例化的 ImageView 对象中设置监听 onClick 函数，调用对话框的实例化对象调用 show 函数实现。

```
30    ImageView mImage = (ImageView) findViewById(R.id. imageView);
31    final AlertDialog.Builder mDialog = new AlertDialog.Builder(this);
32    final String choseDialog[] = {"拍摄", "从相册选择"};
33    mDialog.setTitle("上传图片")
34    .setItems(choseDialog, (dialog, i) -> {
35        Toast.makeText(getApplicationContext(), "您选择了[" + choseDialog[i] + "]", Toast.LENGTH_LONG).show();
36    })
37    .setNegativeButton("取消", new DialogInterface.OnClickListener() {
38        @Override
39        public void onClick(DialogInterface dialog, int i) {
40            Toast.makeText(getApplicationContext(), "取消", Toast.LENGTH_LONG).show();
41        }
42    }).create();
43
44
45    mImage.setOnClickListener((view) -> { mDialog.show(); });
```

3) 实例化 RadioGroup 对象，使用 setOnCheckedChangeListener 函数监听，判断当前的 checkId，寻找对应的 RadioBox 的对象的 id 名，然后再通过 Snackbar 来发送消息到屏幕上。

```
52    final RadioGroup mGroup = (RadioGroup) findViewById(R.id. radioGroup);
53    mGroup.setOnCheckedChangeListener((group, checkedId) -> {
54        if (checkedId == R.id. radioButton_Student)
55        {
56            Snackbar.make(group, R.string. radioButton_Student, Snackbar.LENGTH_LONG)
57                .setAction("确定", (view) -> {
58                    Toast.makeText(getApplicationContext(), "Snackbar的确定按钮被点击了", Toast.LENGTH_LONG).show();
59                })
60                .setActionTextColor(getResources().getColor(R.color. colorPrimary))
61                .setDuration(5000)
62                .show();
63        }
64        else
65        {
66            // (else 部分类似。)
67        }
68    });
```

(else 部分类似。)

4) EditText 控件由于已经转换成了 TextInputLayout 控件，所以在实例化的时候要注意一下。先用 TextInputLayout 实例化一个对象 mNumber，再对 mNumber 调用 getEditText 函数。经过查询，TextInputLayout 需要使用 addTextChangedListener 函数，通过 alt+Enter 自动补全出了三个子函数，将需要改动的地方 setErrorEnabled 函数写在 onTextChanged 里，当传入 false 时，不会显示错误的信息。

```
85    mNumber = (TextInputLayout) findViewById(R.id. editNum);
86    mPassword = (TextInputLayout) findViewById(R.id. editPassword);
87    NumberEdit = mNumber.getEditText();
88    PasswordEdit = mPassword.getEditText();
89    NumberEdit.addTextChangedListener(new TextWatcher() {
90
91        public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {
92            if (charSequence.length() != 0) mNumber.setErrorEnabled(false);
93        }
94    });
```

以及

```
95    PasswordEdit.addTextChangedListener(new TextWatcher() {
96
97        public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {
98            if (charSequence.length() != 0) mPassword.setErrorEnabled(false);
99        }
100    });
```

5) 最后时 button 的实例化，由于 button login 触发的事件与 Text 文本有关，所以这里还需要与 TextInputLayout 的对象关联起来。Login 按键同样采用函数 setOnClickListener 监听事件。特别注意使用【EditText】而不是【TextInputLayout】的对象进行 getText()，要区分 getEditText，最后转化为 String 就可以得到输入的学号和密码的 string。

```
120 Button mLogin = (Button)findViewById(R.id.button_login);
121 mLogin.setOnClickListener(new OnClickListener() {
124     int msg;
125     String getNumber = mNumber.getEditText().getText().toString();
126     String getPassword = mPassword.getEditText().getText().toString();
```

第一步：学号和密码都正确，则登陆成功，判断条件为真时，修改 msg 为提示的字符串“登陆成功”，通过之后的 Snackbar 向屏幕发送消息。

第二步：学号和密码有一个不正确，则登陆不成功，执行与第一步类似的操作。

```
127     if (getNumber.equals("123456") && getPassword.equals("6666")) {
128         msg = R.string.loginSuccess;
129     }
130     else {
131         msg = R.string.loginFail;
132     }
```

第三步：Snackbar 向屏幕发送消息。

第四、五步：判断学号、密码是否为空，若为空的情况下，需要向对应的 TextInputLayout 调用函数 setErrorEnabled 函数，传入 True 以提示学号/密码不能为空。

```
144     if (getNumber.length() == 0)
145     {
146         mNumber.setErrorEnabled(true);
147         mNumber.setError("学号不能为空");
148     }
```

#### (4) 实验遇到困难以及解决思路

困难 1：虚拟机无法使用，提示 Instant Run applied code changes and restarted the app，虚拟机提示 Keep Stopping。

反复尝试无效，在充分判断代码、SDK、APT、布局文件都没有问题的情况下，采用 clean 整个 Project 再 Rebuild 就可以进入了。

困难 2：使用安卓手机时闪退。

定义出错，先用虚拟机发现也闪退，然后对代码逐条注释，发现是在实例化的时候出错。当布局文件中不存在某个控件，而编写时依然在 java 中对其实例化，就根本不会报错直接闪退。以后千万注意。

困难 2：当点击【登陆】按钮时按键没有正确触发。

首先先要搞清楚每一步的执行逻辑，其次才是去详细写每一步的代码。代码需要建立在逻辑之上，可以通过画图解决。

## 四、课后实验结果

运行正常，无其余扩展。

## 五、 实验思考及感想

本次实验是建立在上一次实验基础上的 UI 界面设计实验，根据我上一次的实验思考，我认为本次的实验依旧是体现“通用性”的一次实验。私以为通用性在安卓的操作中比较重要，因为安卓的代码比较繁琐，虽然我们可以通过活用自动补全的功能，但是从原理上这也是保证代码通用性的一种表现。比如某个实例化的对象，报错提示“不能存在于 innerclass 需要将其变成 final”。实例化的对象就相当于一个已经保存好的内容，只要调用相应的函数就可以实现在界面上的相关操作。

剩下的一点扩展思考是关于“事件监听”的，如果将一个 MainActivity 看作一个线程，那么该界面始终执行的是单线程的操作。那么之后的实验中应该会考虑到多线程的操作以及多核（多进程）的处理。

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。