

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)	移动信息工程 (移动互联网)
学号	15352083	姓名	冯灏帆
电话	13712309295	Email	ares_fung@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”，如图：



<div>请输入学号</div> <div>1234567</div> <div>请输入密码</div> <div>.....</div> <div><input type="radio"/> 学生 <input checked="" type="radio"/> 教职工</div> <div><div>登录</div><div>注册</div></div>	<div>请输入学号</div> <div>123456</div> <div>请输入密码</div> <div>....</div> <div><input type="radio"/> 学生 <input checked="" type="radio"/> 教职工</div> <div><div>登录</div><div>注册</div></div>
学号或密码错误	登录成功
确定	确定

(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”。

☒ 学生 ☐ 教职工

登录

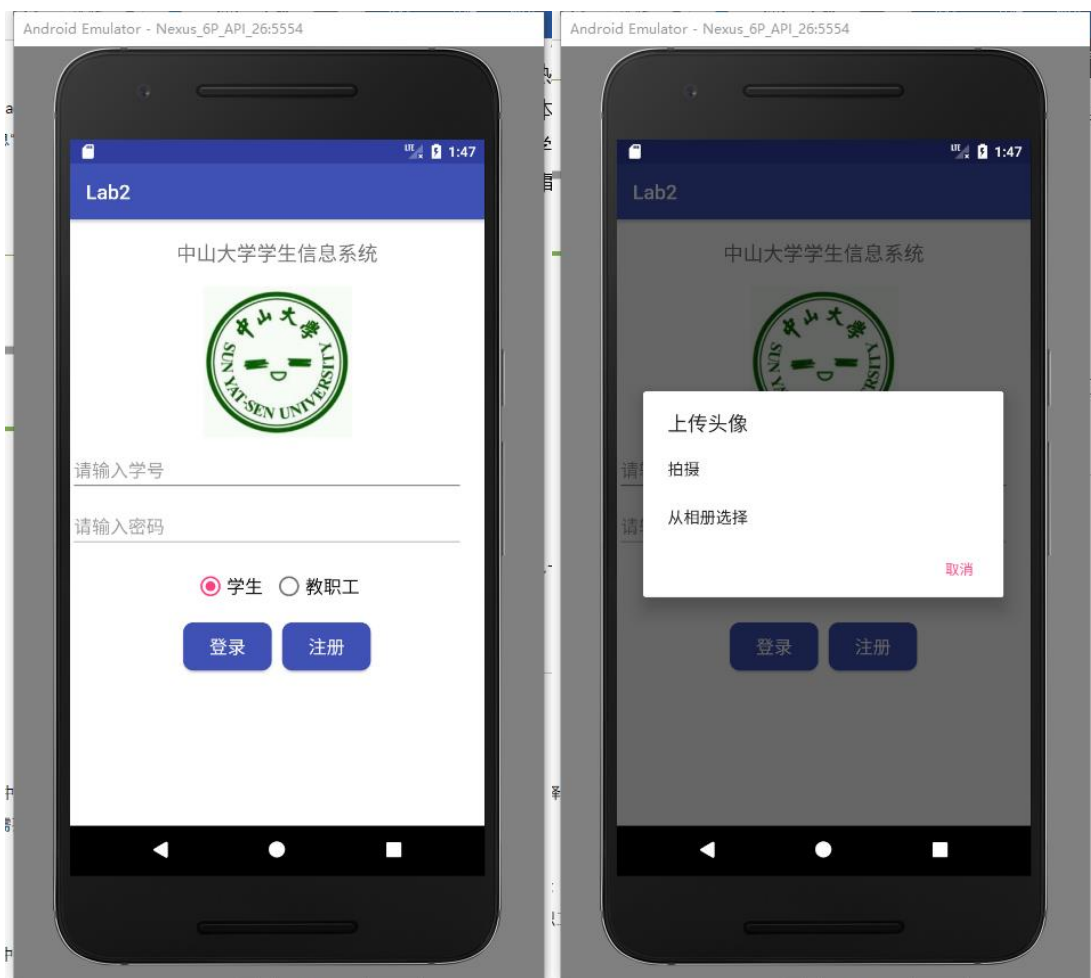
注册

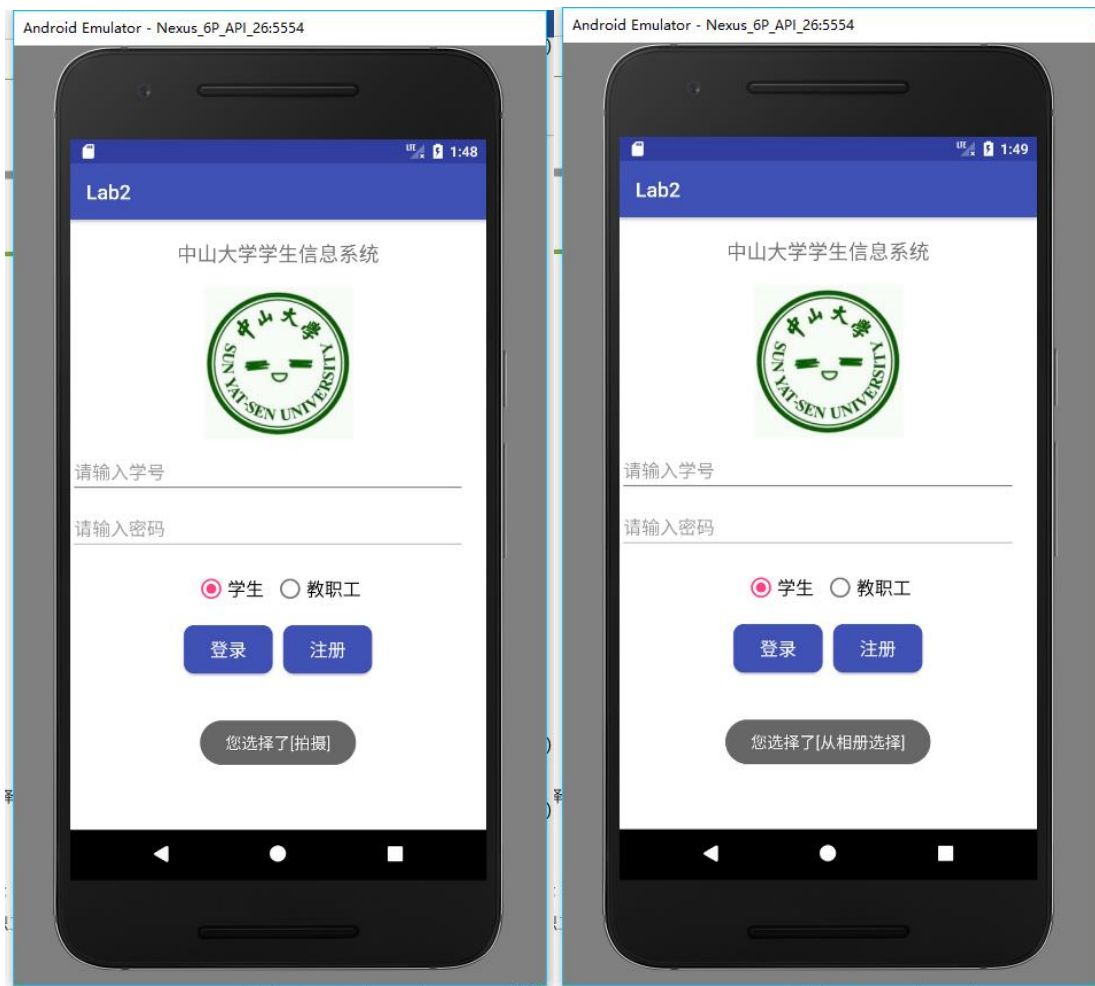
学生注册功能尚未启用

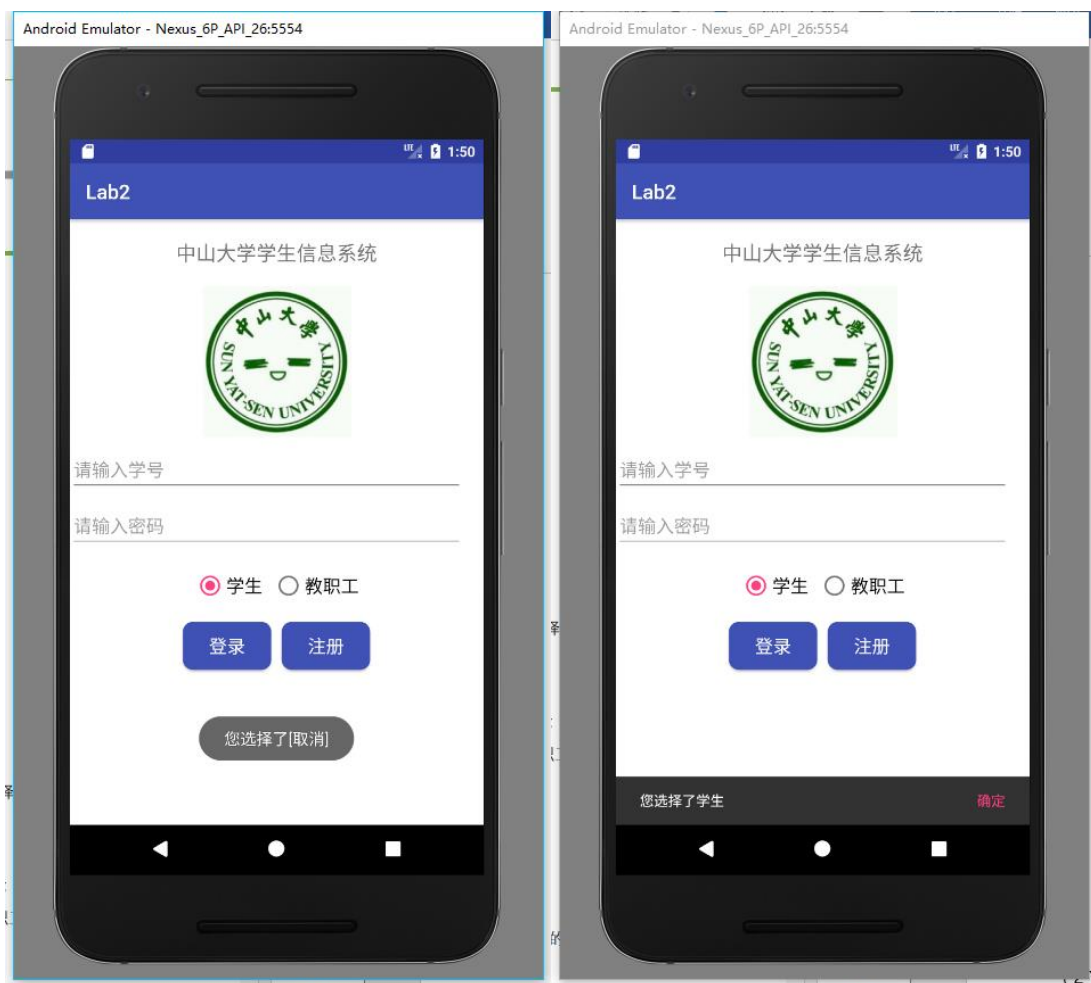
确定

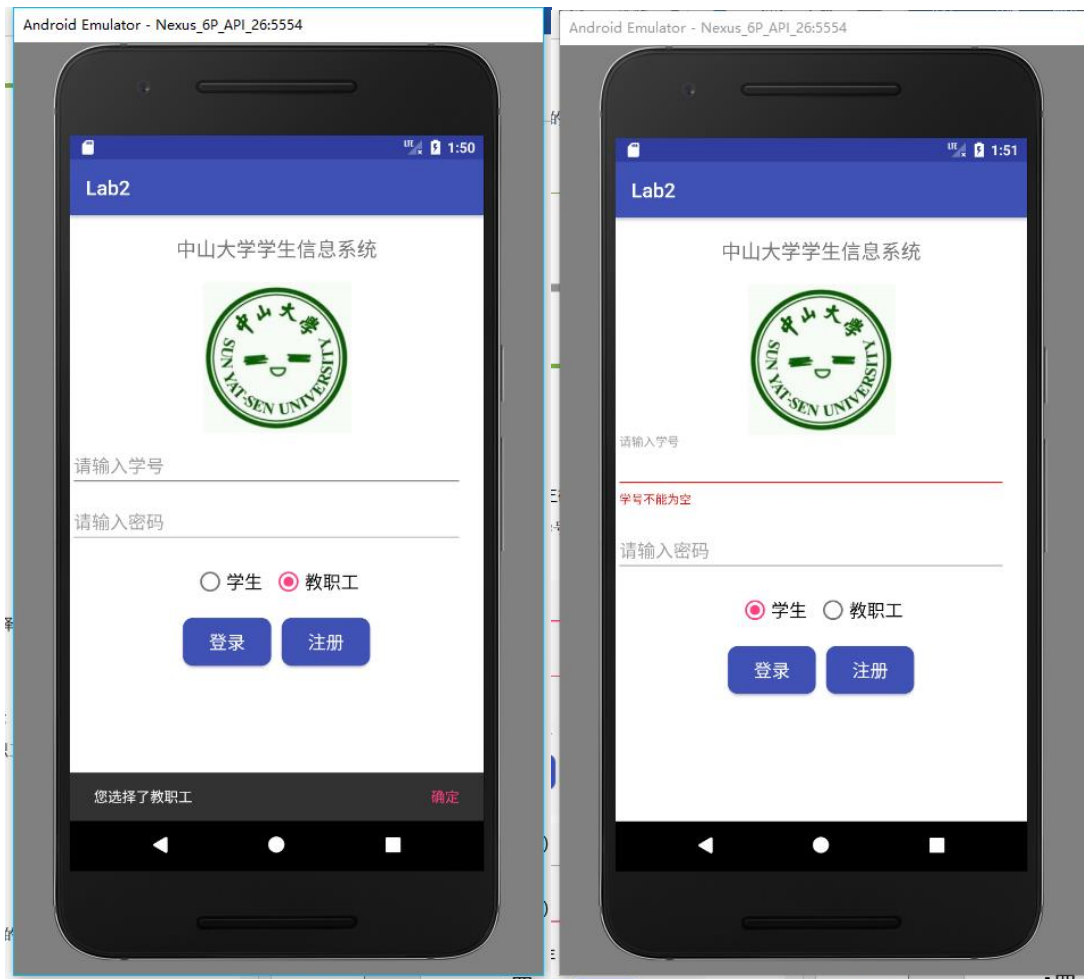
三、 课堂实验结果

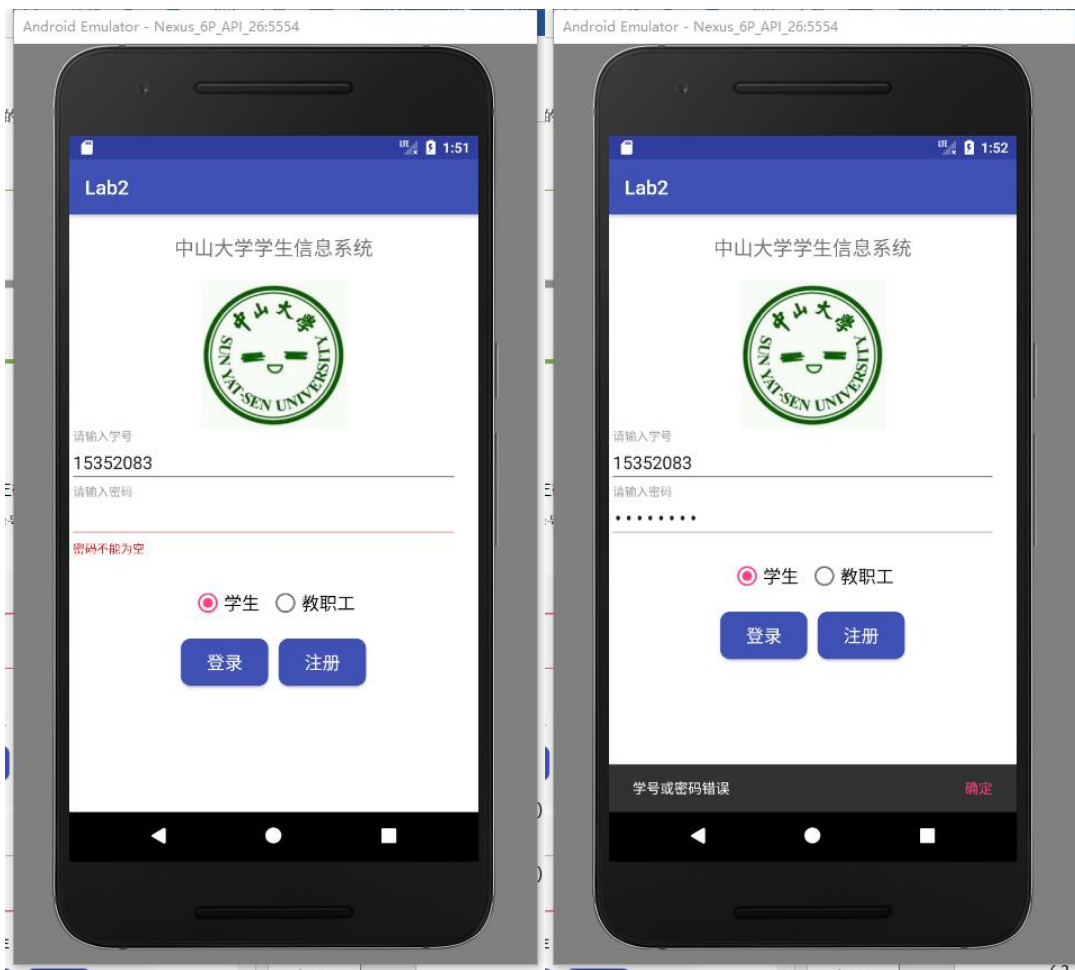
(1) 实验截图

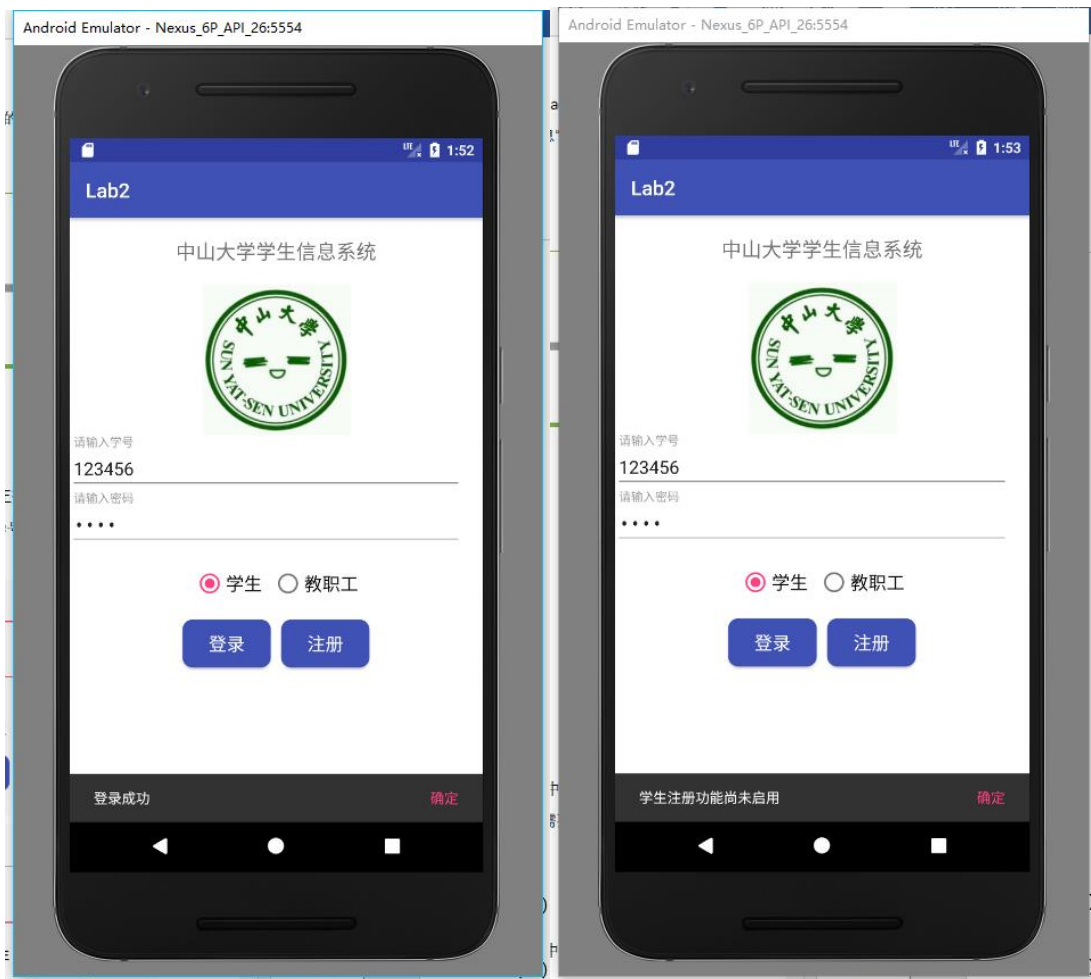














(2) 实验步骤以及关键代码

首先在 xml 布局文件中，将输入学号与输入密码的 EditText 控件放入 Textinputlayout 的控件当中，以便于实现后续的实验功能。

```
<android.support.design.widget.TextInputLayout
    android:id="@+id/textinputlayout"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    tools:layout_editor_absoluteY="0dp"
    tools:layout_editor_absoluteX="0dp"
    app:layout_constraintHorizontal_chainStyle="spread_inside"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/imageView">

    <EditText
        android:id="@+id/editText2"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginEnd="20dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="20dp"
        android:layout_marginTop="20dp"
        android:ems="30"
        android:hint="请输入学号"
        android:inputType="numberDecimal" />

</android.support.design.widget.TextInputLayout>
```

```
<android.support.design.widget.TextInputLayout
    android:id="@+id/textinputlayout2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    tools:layout_editor_absoluteY="0dp"
    tools:layout_editor_absoluteX="0dp"
    app:layout_constraintHorizontal_chainStyle="spread_inside"
    app:layout_constraintTop_toBottomOf="@+id/textinputlayout"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent">

    <EditText
        android:id="@+id/editText3"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginEnd="20dp"
        android:layout_marginRight="20dp"
        android:layout_marginTop="20dp"
        android:ems="30"
        android:hint="请输入密码"
        android:inputType="textPassword" />

</android.support.design.widget.TextInputLayout>
```

对于 ImageView 控件添加 OnClick 监听，弹出 AlertDialog 对话框。首先要创建一个 Builder，在利用这个 Builder 的 create 函数建立对话。根据实验的要求，对话框标题的修改可以

利用 setTitle 函数，设置了两个单选元素“拍摄”与“从相册选择”可以使用 setItems 函数，并对这些 Items 建立 OnClick 监听，触发时发送对应的 Toast 信息。

```
ImageView myimage = (ImageView) findViewById(R.id. imageView);
myimage.setOnClickListener(new ImageView.OnClickListener() {
    public void onClick(View v) {
        AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity. this);
        AlertDialog dialog = builder.create();
        final String [] Items={"拍摄", "从相册选择"};
        builder.setTitle("上传头像");
        builder.setItems(Items, new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface d, int i) {
                Toast.makeText(MainActivity. this, "您选择了["+Items[i]+"", Toast. LENGTH_SHORT).show();
            }
        });
        builder.setNegativeButton("取消", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface d, int i) {
                Toast.makeText(MainActivity. this, "您选择了[取消]", Toast. LENGTH_SHORT).show();
            }
        });
        builder.setCancelable(true);
        builder.show();
    }
});
```

同样地，对于 RadioGroup 中切换 RadioButton 的选项，添加 OnCheckedChangeListener 监听，触发时发出不同的 Snackbar 信息，并对 Snackbar 信息的确定按钮添加 OnClick 监听，若点击该确定按钮时将发送“Snackbar 的确定按钮被点击了”的 Toast 信息。

```
final RadioGroup myradiogroup=(RadioGroup) findViewById(R.id. radioGroup);
myradiogroup.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {
    public void onCheckedChanged(RadioGroup group, int checkedId) {
        RadioButton checkButton=(RadioButton) findViewById(checkedId);
        Snackbar snackbar = Snackbar.make(findViewById(R.id. main), "您选择了" + checkButton.getText().toString(), Toast. LENGTH_SHORT);
        snackbar.setAction("确定", new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Toast.makeText(MainActivity. this, "Snackbar的确定按钮被点击了", Toast. LENGTH_SHORT).show();
            }
        });
        snackbar.show();
    }
});
```

利用输入学号与输入密码的两个 EditText 的父容器 Textinputlayout 发出实验要求的错误信息，错误信息的内容设置在 setError 属性当中，而 setErrorEnable 属性则控制该 Textinputlayout 控件是否允许显示错误信息。当学号密码正确或错误时，发出对应的 Snackbar 信息。

```

final EditText usernum = (EditText) findViewById(R.id. editText2);
final EditText password = (EditText) findViewById(R.id. editText3);
final TextInputLayout usernumlayout = (TextInputLayout) findViewById(R.id. textinputlayout);
final TextInputLayout passwordlayout = (TextInputLayout) findViewById(R.id. textinputlayout2);

Button btn1 = (Button) findViewById(R.id. button);
btn1.setOnClickListener(new Button.OnClickListener() {

    @Override
    public void onClick(View view) {
        boolean checkpassword = true;
        if (TextUtils.isEmpty(usernum.getText().toString())) {
            usernumlayout.setErrorEnabled(true);
            usernumlayout.setError("学号不能为空");
            passwordlayout.setError("");
            passwordlayout.setErrorEnabled(false);
            checkpassword = false;
        }
        else {
            usernumlayout.setError("");
            usernumlayout.setErrorEnabled(false);
            if (TextUtils.isEmpty(password.getText().toString())) {
                passwordlayout.setErrorEnabled(true);
                passwordlayout.setError("密码不能为空");
                checkpassword = false;
            }
            else {
                passwordlayout.setError("");
                passwordlayout.setErrorEnabled(false);
            }
        }
    }
}

```

```

if (checkpassword) {
    usernumlayout.setError("");
    usernumlayout.setErrorEnabled(false);
    passwordlayout.setError("");
    passwordlayout.setErrorEnabled(false);
    boolean correctness = false;
    if (usernum.getText().toString().equals("123456"))
        if (password.getText().toString().equals("6666"))
            correctness = true;
            if (!correctness) {
                Snackbar snackbar = Snackbar.make(findViewById(R.id.main), "学号或密码错误", Toast.LENGTH_SHORT);
                snackbar.setAction("确定", new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        Toast.makeText(MainActivity.this, "Snackbar的确定按钮被点击了", Toast.LENGTH_SHORT).show();
                    }
                });
                snackbar.show();
            }
            else {
                Snackbar snackbar = Snackbar.make(findViewById(R.id.main), "登录成功", Toast.LENGTH_SHORT);
                snackbar.setAction("确定", new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        Toast.makeText(MainActivity.this, "Snackbar的确定按钮被点击了", Toast.LENGTH_SHORT).show();
                    }
                });
                snackbar.show();
            }
        }
    }
}

```

最后对于两个单选 Button 的 OnClick 监听相对简单，利用之前 RadioGroup 调用成员函数 `getCheckedRadioButtonId()` 得到选中的 RadioButton 的 id，根据 id 判断选中的 RadioButton 发出不同的 Snackbar 信息。需要注意的是 java 语言中字符串的比较不能使用双等号，可以调用 string 库中的成员函数 `equals()` 比较，返回一个布尔值表示传入 `equals()` 函数的字符串是否与调用 `equals()` 函数的字符串相同。

```

Button btn2 = (Button) findViewById(R.id.button2);
btn2.setOnClickListener(new Button.OnClickListener() {
    @Override
    public void onClick(View view) {
        int checkedButtonid = myradiogroup.getCheckedRadioButtonId();
        RadioButton checkButton = (RadioButton) findViewById(checkedButtonid);
        final String words = "注册功能尚未启用";
        if (checkButton.getText().toString().equals("学生")) {
            Snackbar snackbar = Snackbar.make(findViewById(R.id.main), checkButton.getText().toString() + words, Toast.LENGTH_SHORT);
            snackbar.setAction("确定", new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    Toast.makeText(MainActivity.this, "Snackbar的确定按钮被点击了", Toast.LENGTH_SHORT).show();
                }
            });
            snackbar.show();
        }
        if (checkButton.getText().toString().equals("教职工")) {
            Toast.makeText(MainActivity.this, checkButton.getText().toString() + words, Toast.LENGTH_SHORT).show();
        }
    }
});
}

```

（3）实验遇到困难以及解决思路

第一个困难时对于 Snackbar 的使用，不同于 Toast 的第一个参数为 content 上下文类型，Snackbar 的第一个参数是根布局，对于本次实验的要求，我们需要给约束布局加上 id，然后利用 findViewById () 方法指向约束布局，带入到 Snackbar 的创建调用当中。

另外一个困难在于对 Textinputlayout 的使用，本次实验要求将两个 EditText 控件放置到 Textinputlayout 当中。当触发对登录 Button 的 OnClick 监听时，必须依次检查学号是否为空、密码是否为空、学号或密码是否正确。刚刚实现该功能时，没有及时取消 Error 信息的显示，当输入学号的 Textinputlayout 给出“学号不能为空”的 Error 信息时，无论输入密码的 Textinputlayout 是否输入了密码，都不能显示 Error 信息。同样的，我们需要在每次点击登录按钮后触发的 OnClick 监听中重置两个 Textinputlayout 的 Error 信息与显示 Error 信息的权限。

```
boolean checkpassword = true;
if (TextUtils.isEmpty(usernum.getText().toString())) {
    usernumlayout.setErrorEnabled(true);
    usernumlayout.setError("学号不能为空");
    passwordlayout.setError("");
    passwordlayout.setErrorEnabled(false);
    checkpassword = false;
}
else {
    usernumlayout.setError("");
    usernumlayout.setErrorEnabled(false);
    if (TextUtils.isEmpty(password.getText().toString())) {
        passwordlayout.setErrorEnabled(true);
        passwordlayout.setError("密码不能为空");
        checkpassword = false;
    }
    else {
        passwordlayout.setError("");
        passwordlayout.setErrorEnabled(false);
    }
}
```

然后确保输入的学号密码文本非空后，与正确学号密码比较在根据正确与否，选择发出 Toast 信息或 Snackbar 信息。

四、课后实验结果

在对两个文本输入框进行测试的过程中，没有实现自动隐藏的键盘非常不便且严重影响用户体验。为了对此问题进行优化，在约束布局中先加入一个横宽填充父容器的线性布局作为一个辅助元素，并将该线性布局的 clickable、focusable 与 focusableInTouchMode 属性设置为 true。

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/traceroute_rootview"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#FFFFFF"
    android:clickable="true"
    android:gravity="center_horizontal"
    android:orientation="vertical"
    android:focusable="true"
    android:focusableInTouchMode="true">

</LinearLayout>

```

然后在 MainActivity.java 文件当中加入对此线性布局的 OnClick 监听。调用系统命令隐藏触摸键盘。

```

final LinearLayout myrootlayout = (LinearLayout)findViewById(R.id.traceroute_rootview);
myrootlayout.setOnClickListener(new LinearLayout.OnClickListener() {
    @Override
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.traceroute_rootview:
                InputMethodManager imm = (InputMethodManager)
                    getSystemService(Context.INPUT_METHOD_SERVICE);
                imm.hideSoftInputFromWindow(v.getWindowToken(), 0);
                break;
        }
    }
});

```

五、 实验思考及感想

有了 Lab1 对 Android 开发的熟悉后，对于功能实现上的改动与对于代码改进思考多了很多，也开始使用 OOP 的思想进行 Android 开发，同时尝试加入新功能进行优化与创新。对于简单的事件处理已经得心应手，课后尝试了除点击事件以外的监听，获益良多。进步最大的一点是对于 Android 开发的大概层次慢慢有了相对肤浅的了解，并且很多时候需要修改添加的定义或实现也知道在哪个文件当中，故在阅读更多他人的优秀 Android 项目源码与文档时能够更快地吸收理解，转换为自己在开发设计中运用的知识与技能。