

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	专业 (方向)	互联网
学号	15352006	姓名	蔡丽芝
电话	13538489980	Email	314749816@qq.com
开始日期	2017/10/13	完成日期	2017/10/14

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



实现功能如实验文档中所述，这里不加赘述

三、 课堂实验结果

（1）实验截图

打开应用后的第一个页面



点击图片



分别点击拍摄，从相册中选择，取消时



切换 RadioButton 的选项



点击登录按钮的四个状态

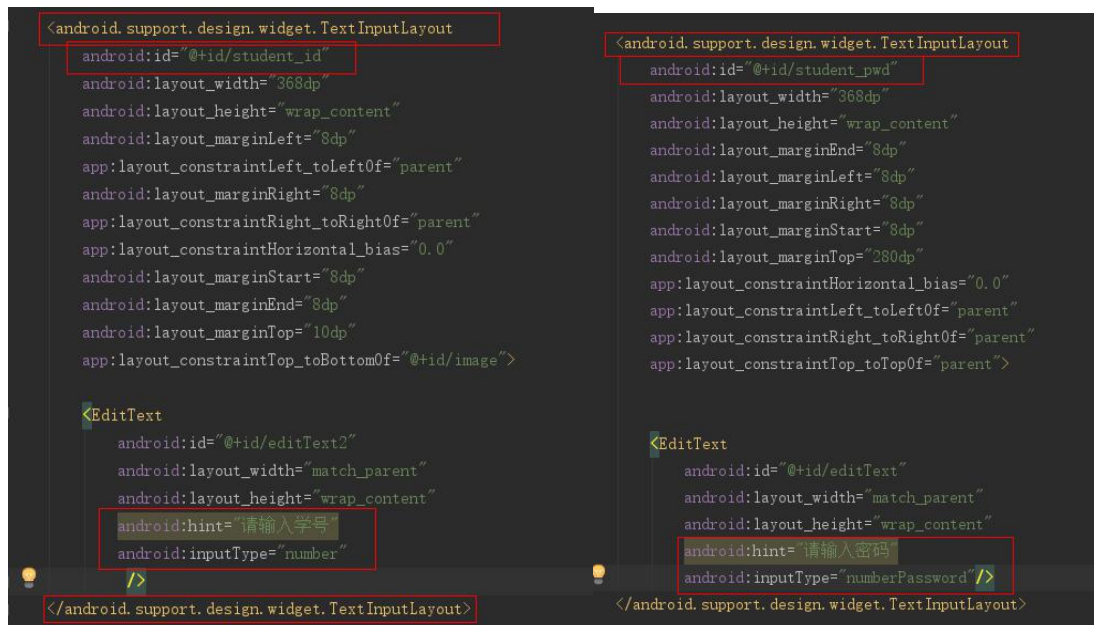


点击注册按钮



(2) 实验步骤以及关键代码

(1) 输入学号和密码的控件用 `TextInputLayout` 实现



对于学号输入框

```
android:id="@+id/student_id"
```

第一个 `TextInputLayout` 控件有一个唯一的 id: `student_id`

```
android:inputType="number"
```

`inputType="number"` 只能输入数字，并且在手机上会自动打开数字键盘

对于密码输入框

```
android:id="@+id/student_pwd"
```

第二个 `TextInputLayout` 控件同样有一个唯一的 id: `student_pwd`

```
android:inputType="numberPassword"
```

`inputType="numberPassword"` 只能输入数字，并且在手机上会自动打开数字键盘，同时输入的数字会被隐藏掉。

通过使用 `TextInputLayout` 布局，可以更方便地获取文本输入框中的内容和进行验证

(2) 点击图片，弹出对话框

a. 首先声明一个 `ImageView` 的变量，并进行赋值操作

```
private ImageView mImage;
```

```
mImage = (ImageView) findViewById(R.id.image);
```

b. 对相应的控件设置事件监听器

```
mImage.setOnClickListener(new View.OnClickListener() {  
    final String[] items = new String[] {"拍摄", "从相册中选择"};  
    @Override  
    public void onClick(View v) {  
        // 创建一个对话框对象  
        AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);  
        // 对对话框内容进行定义  
        builder.setTitle("上传头像").setItems(  
            items, (dialog, which) -> {  
                switch (which) {  
                    case 0:  
                        Toast.makeText(MainActivity.this, "您选择了[拍摄]", Toast.LENGTH_SHORT).show();  
                        break;  
                    case 1:  
                        Toast.makeText(MainActivity.this, "您选择了[从相册选择]", Toast.LENGTH_SHORT).show();  
                        break;  
                }  
            }  
        ).setNegativeButton(  
            "取消", (dialog, which) -> {  
                Toast.makeText(MainActivity.this, "您选择了[取消]", Toast.LENGTH_SHORT).show();  
            }).show();  
    }  
});
```

首先通过 AlertDialog.Builder 创建一个对话框对象 builder，然后分别通过 setTitle()函数和 setItem 函数和 setNegativeButton 函数设置相应的列表和事件绑定，当点击相应的文字事件发生的时候，调用 Toast.makeText()函数

(3) 切换 RadioButton 的选项

```
mRadioGroup = (RadioGroup) findViewById(R.id.radioGroup2);  
RB1 = (RadioButton) findViewById((R.id.student));  
RB2 = (RadioButton) findViewById((R.id.teacher));  
checked = (RadioButton) findViewById(mRadioGroup.getCheckedRadioButtonId());  
Rchecked = checked.getText().toString();
```

首先通过 findViewById(R.id.radioGroup2)函数找到 RadioGroup 并赋值给 mRadioGroup

通过 `findViewById` 函数同理将 RB1(第一个学生的单选按钮), RB2 (第二个教职工的单选按钮), `checked` (被选中的按钮) 进行赋值, 最后, 调用 `getText()` 函数将 `checked` 中的文字选中赋值给 `Rchecked`, 便于后面的处理和判断

```
mRadioGroup.setOnCheckedChangeListener((group, checkedId) -> {
    if(checkedId == RB1.getId()) {
        checked = (RadioButton) findViewById(mRadioGroup.getCheckedRadioButtonId());
        Rchecked = checked.getText().toString();
        Snackbar.make(group, "您选择了学生", Snackbar.LENGTH_INDEFINITE)
            .setAction("确定", (v) -> {
                Toast.makeText(MainActivity.this, "Snackbar 的确定按钮被点击了", Toast.LENGTH_SHORT).show();
            }).setActionTextColor(getResources().getColor(R.color.colorPrimary)).show();
    }
    else if(checkedId == RB2.getId()) {
        checked = (RadioButton) findViewById(mRadioGroup.getCheckedRadioButtonId());
        Rchecked = checked.getText().toString();
        Snackbar.make(group, "您选择了教职工", Snackbar.LENGTH_INDEFINITE)
            .setAction("确定", (v) -> {
                Toast.makeText(MainActivity.this, "Snackbar 的确定按钮被点击了", Toast.LENGTH_SHORT).show();
            }).setActionTextColor(getResources().getColor(R.color.colorPrimary)).show();
    }
});
```

`mRadioGroup` 设置事件监听器, 当被选中的 `radio` 对象发生改变的时候, 会触发事件, 通过判断 `checkedId` 和 `RB1.getId()`, 来决定是哪个按钮被选择, 按钮发生变化的时候, `checked` 和 `Rchecked` 也必须进行相应的更新。使用 `Snackbar` 控件进行相应的提示

(5) 点击登录按钮

```
mIdText = (TextInputLayout) findViewById(R.id.student_id);
mPassWord = (TextInputLayout) findViewById(R.id.student_pwd);
login = (Button) findViewById(R.id.login);
regist = (Button) findViewById(R.id.register);
rootView = getWindow().getDecorView().findViewById(android.R.id.content);
```

分别通过 `findViewById` 和每个控件的 `id`, 获得相应的控件

```
login.setOnClickListener((v) -> { // 将 login 绑定点击事件
    String id = mIdText.getText().toString();
    String pwd = mPassWord.getText().toString();
```

首先调用 `getEditText()` 函数获得控件 `EditText`，然后调用 `getText()` 函数获得控件里的文本内容

```
if(id.length() == 0 || pwd.length() == 0) {
    if(id.length()==0) {
        mIdText.setErrorEnabled(true);
        mIdText.setError("学号不能为空");
    }
    if (pwd.length()==0) {
        mPassWord.setErrorEnabled(true);
        mPassWord.setError(("密码不能为空"));
    }
    if(id.length()!=0) {
        mIdText.setErrorEnabled(false);
    }
    if(pwd.length()!=0) {
        mPassWord.setErrorEnabled(false);
    }
}
else {
    if (id.equals("123456") && pwd.equals("6666")) {
        Snackbar.make(rootView, "登录成功", Snackbar.LENGTH_INDEFINITE)
            .setAction("确定", (v) -> {
                Toast.makeText(MainActivity.this, "登录成功", Toast.LENGTH_SHORT).show();
            }).setActionTextColor(getResources().getColor(R.color.colorPrimary)).show();
    }
    if (!id.equals("123456") || !pwd.equals("6666")) {
        Snackbar.make(rootView, "学号或密码错误", Snackbar.LENGTH_INDEFINITE)
            .setAction("确定", (v) -> {
                Toast.makeText(MainActivity.this, "登录失败", Toast.LENGTH_SHORT).show();
            }).setActionTextColor(getResources().getColor(R.color.colorPrimary)).show();
    }
}
```

如果出现一个 `EditText` 的文本内容为空，就调用 `setError` 进行错误提示

通过比较文本内容和要求的账号密码是否相匹配，然后使用 `Snackbar` 控件进行相应的提示

(6) 点击注册按钮


```

regist.setOnClickListener((v) -> {
    if (Rchecked.equals("学生")) {
        Toast.makeText(MainActivity.this, "学生注册功能尚未启用", Toast.LENGTH_SHORT).show();
    } else if (Rchecked.equals("教职工")) {
        Toast.makeText(MainActivity.this, "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

答：实验一开始遇到的困难是依赖问题，由于我的 sdk 版本是 25.0，不能直接将 `compile 'com.android.support:design:26.0.0-alpha1'` 放到 `dependencies` 中，后来虽然我将其改为对应的版本，但是我的 AS 找不到对应的安装包，所以一开始弄了很久，最后通过升级 SDK 和 SDK platform 到 26.0 才解决了问题，其次就是对事件监听器和一些功能如对话框，Toast，Snackbar 不熟悉，不知道怎么使用，也不知道怎么在对话框中进行对于的事件操作，最后查询 Android 的官方学习文档和网上的博客，终于学会了如何使用上述控件和功能框。

四、课后实验结果

当两个输入框都没有填内容的时候，能同时显示错误提示

晚上10:44

login2.3

中山大学学生信息系统



请输入学号

学号不能为空

请输入密码

密码不能为空

☒ 学生 ☐ 教职工

注册 登录

五、 实验思考及感想

通过这次的实验，对于 `ImageView`，`Button`，`RadioButton` 这些基本的控件更加地熟悉和了解，基本上能处理一些简单的事件，一些常见的比如说对话框，也是基本能掌握，但是一开始接触入手的时候，很困难，不知道怎么学习，也不知道如何 `debug`，导致我这次实验花了很多时间才完成，希望自己下次实验能吸取这次实验的教训，高效地完成。