

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	专业（方向）	移动互联网
学号	15352344	姓名	吴文标
电话	13112312320	Email	wwb.bill@qq.com
开始日期	2017.10.12	完成日期	2017.10.13

一、实验题目

事件处理

二、实现内容

实现一个 Android 应用，界面呈现如下效果：

Lab2

中山大学学生信息系统



请输入学号

请输入密码

☒ 学生 ☐ 教职工

登录

注册

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：

请输入学号

学号不能为空

请输入密码

☐ 学生 ☒ 教职工

请输入学号

1234567

请输入密码

密码不能为空

☐ 学生 ☒ 教职工

请输入学号

1234567

请输入密码

.....

☐ 学生 ☒ 教职工

学号或密码错误

请输入学号

123456

请输入密码

....

☐ 学生 ☒ 教职工

登录成功

(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

学生注册功能尚未启用

三、 课堂实验结果

(1) 实验截图

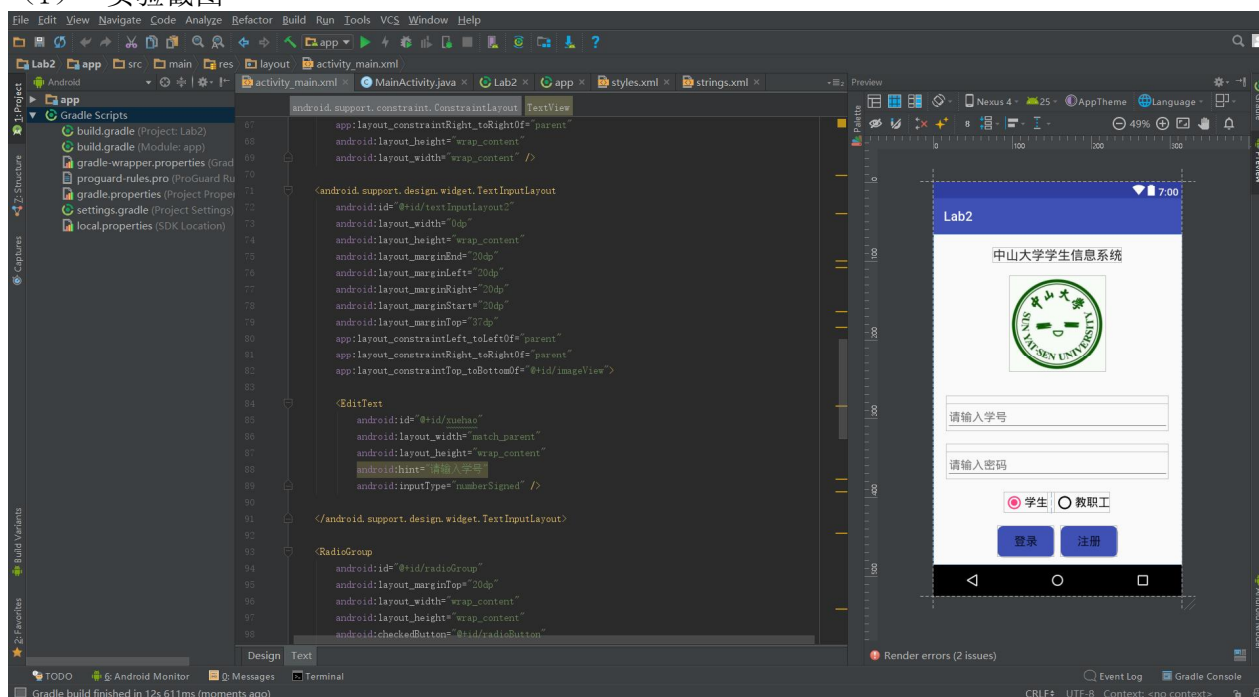


Figure 1 全览

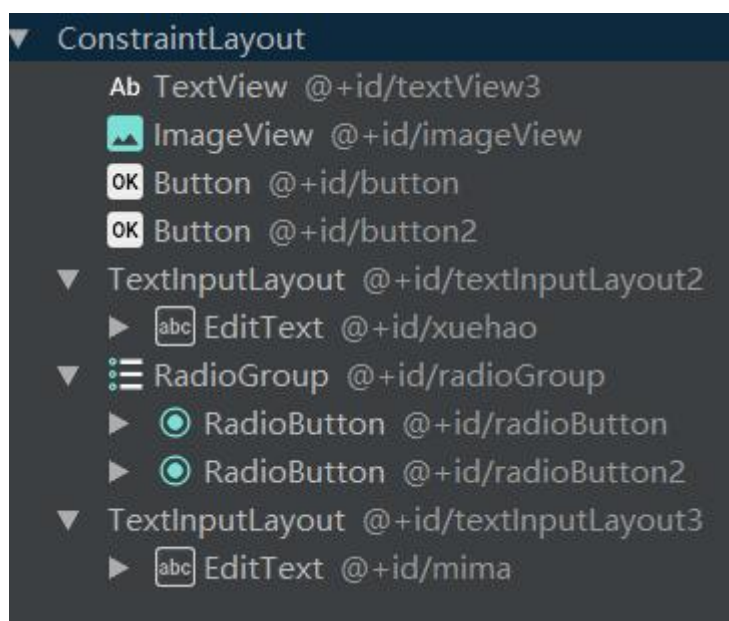


Figure 2 布局元素

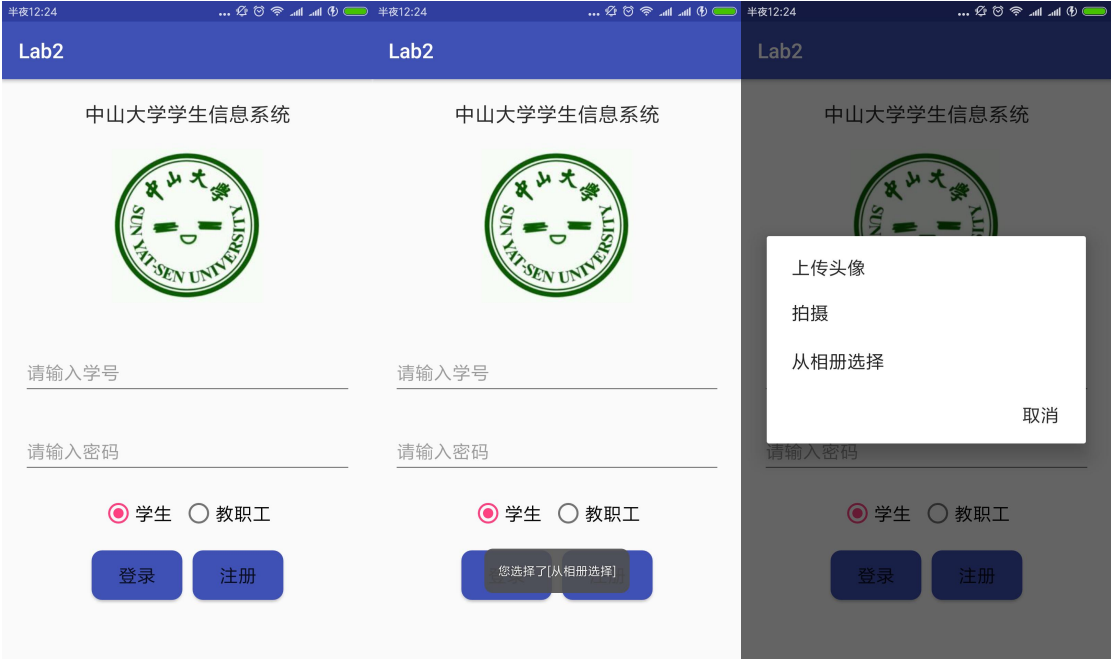




Figure 3 完成效果图 1-9

(2) 实验步骤以及关键代码

步骤 1: 新建 app 文件, 打开 activity_main.xml, design 窗格选择 ConstraintLayout 布局添加需要用到的控件, 按实验文档要求做好每个控件的样式和布局。其中 TextInputLayout 需要安装相应的库, 通过搜索添加该控件便会自动配置相关文件, 然后便可以使用了。

关键代码:

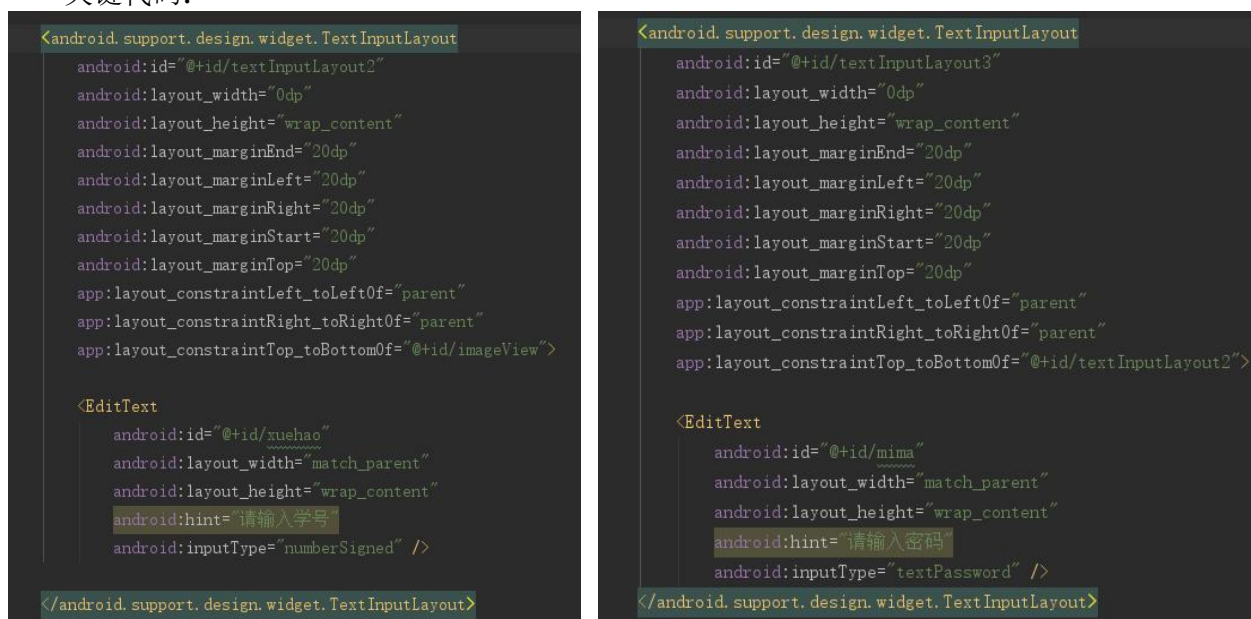


Figure 4 TextInputLayout

步骤 2: 点击图片, 弹出对话框及其相关功能实现。主要用到了 AlertDialog 和 Toast 控件, 由于 setMessage() 函数只能显示一条信息所以这里用上了 setItems(), 这里有一个疑问是程序在本人手机运行的对话框字体样式和 ppt 实例代码不同, 怀疑是版本问题。

关键代码:

```

mImage.setOnClickListener((view) -> {
    new AlertDialog.Builder(MainActivity.this)
        .setTitle("上传头像")
        .setItems(
            new String[] {"拍摄", "从相册选择"},
            (dialog, which) -> {
                switch (which) {
                    case 0:
                        Toast.makeText(getApplicationContext(), "您选择了[拍摄]", Toast.LENGTH_SHORT).show();
                        break;
                    case 1:
                        Toast.makeText(getApplicationContext(), "您选择了[从相册选择]", Toast.LENGTH_SHORT).show();
                        break;
                }
            }
        )
        .setNegativeButton("取消", (dialog, which) -> {
            Toast.makeText(getApplicationContext(), "您选择了[取消]", Toast.LENGTH_SHORT).show();
        })
        .show();
});

```

Figure 5 对话框以及 toast 消息功能

步骤 3: 切换 RadioButton 的选项，弹出 Snackbar 提示“您选择了 xx”及相关功能实现。主要用到 Snackbar 的 make 和 setAction 方法以及 Toast。

关键代码:

```

mGroup.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {
    public void onCheckedChanged(RadioGroup radioGroup, int checkID) {
        switch (checkID) {
            case R.id.radioButton:
                Snackbar.make(radioGroup, "您选择了学生", Snackbar.LENGTH_SHORT)
                    .setAction("确定", new View.OnClickListener() {
                        @Override
                        public void onClick(View view) {
                            Toast.makeText(getApplicationContext(), "Snackbar的确定按钮被点击了", Toast.LENGTH_SHORT).show();
                        }
                    })
                    .show();
                break;
            case R.id.radioButton2:
                Snackbar.make(radioGroup, "您选择了教职工", Snackbar.LENGTH_SHORT)
                    .setAction("确定", new View.OnClickListener() {
                        @Override
                        public void onClick(View view) {
                            Toast.makeText(getApplicationContext(), "Snackbar的确定按钮被点击了", Toast.LENGTH_SHORT).show();
                        }
                    })
                    .show();
                break;
        }
    }
});

```

Figure 6 切换单选按钮

步骤 4: 登录按钮相关功能实现。在这里遇到的 bug 可谓是非常多的了，后面心得详述。

关键代码:


```

mBtnLogin.setOnClickListener(new View.OnClickListener() {
    final TextInputLayout layout1 = (TextInputLayout) findViewById(R.id.textInputLayout2);
    final TextInputLayout layout2 = (TextInputLayout) findViewById(R.id.textInputLayout3);

    public void onClick(View view) {
        mNumberEdit = layout1.getEditText();
        mPasswordEdit = layout2.getEditText();
        String number = mNumberEdit.getText().toString();
        String password = mPasswordEdit.getText().toString();
        if (TextUtils.isEmpty(number)) {
            layout1.setErrorEnabled(true);
            layout1.setError("学号不能为空");
        } else {
            layout1.setErrorEnabled(false);
            if (TextUtils.isEmpty(password)) {
                layout2.setErrorEnabled(true);
                layout2.setError("密码不能为空");
            } else {
                layout2.setErrorEnabled(false);
                if (number.equals("123456") && password.equals("6666")) { //equals & == 的区别
                    Snackbar.make(view, "登录成功", Snackbar.LENGTH_SHORT)
                        .setAction("确定", (view) -> {
                            Toast.makeText(getApplicationContext(), "Snackbar的确定按钮被点击了", Toast.LENGTH_SHORT).show();
                        })
                        .show();
                }
            }
        }
    }
});

```

Figure 7 登录按钮功能

步骤 5：点击注册按钮相关功能。难点在于获取 ID。

关键代码：

```

mBtnRegister.setOnClickListener(new View.OnClickListener() {

    public void onClick(View view){
        int selected = mGroup.getCheckedRadioButtonId();
        switch (selected){
            case R.id.radioButton:
                Snackbar.make(view, "学生注册功能尚未启用", Snackbar.LENGTH_SHORT)
                    .setAction("确定", new View.OnClickListener() {
                        @Override
                        public void onClick(View view) {
                            Toast.makeText(getApplicationContext(), "Snackbar的确定按钮被点击了", Toast.LENGTH_SHORT).show();
                        }
                    })
                    .show();
                break;
            case R.id.radioButton2:
                Toast.makeText(getApplicationContext(), "教职工注册功能尚未启用", Toast.LENGTH_SHORT).show();
                break;
            case -1:
                Toast.makeText(getApplicationContext(), "bug", Toast.LENGTH_SHORT).show();
                break;
        }
    }
});

```

Figure 8 注册按钮功能

(3) 实验遇到困难以及解决思路

- 1) 一开始做实验时并不知道 java 的书写格式,通过资料查询用了不少方法实现了一些功能,比如说在 xml 设置 android:onclick,然后就可以在 java 中直接使用相关的点击功能,后来觉得这样有点不合理,而后就查询一下各大博主的文章看了一下书写规范,再加上实验指导的一些内容才最终写成了现在这个风格的代码。
- 2) EditText 自动获取焦点的取消,一直比较不喜欢这个自动获取焦点,后来在 QQ 群看到同学讨论后,去进行一番搜索找到了解决方法,现在的焦点自动落在 Textview。代码如下:

```
android:focusable="true"  
id:focusableInTouchMode="true"
```

- 3) 点击登录按钮时 TextInputLayout 事件响应失败,在查询了一下一些博客文章后知道还需要加上一个 import android.support.design.widget.TextInputLayout;才能正常调用相关函数方法。
- 4) EditText 用一般的方法选择是不正确的,后来在实验指导 ppt 上看到,这里的 EditText 是包含在 TextInputLayout 里面的所以需要在 TextInputLayout 中 getEditText() 才能选中。
- 5) 一开始不知道 TextInputLayout 自带了 error 提示样式,自己设计了好久的样式还不适配,之后也是在各大 blog 平台看到相关的用法再自己查看了一下 API 发现了这个用法,完美解决。代码: `setErrorEnabled(true); setError("xxxx");`
- 6) 在判断学号密码正确时,初始使用的是“==”来进行判断,一直判断不来,后来了解到“==”比较的是 2 个对象的地址,而“equals”比较的是 2 个对象的内容,显然这里的两个对象地址不同,所以改为 equal 就好了。
- 7) 最后一个卡了我一小时的 bug 是注册按钮,代码无误,查了好久都找不到问题,最后自己检测代码时发现是,注册按钮的变量还没有获取到 ID,也就是少了这一句 `mBtnRegister = (Button) findViewById(R.id.button2);`这里 Androidstudio 是不会报错的,还是要自己注意。

四、 课后实验结果

优化代码,代码风格一致化。

五、 实验思考及感想

相比第一次 Android 实验,本次实验比较注重编程能力,有一段时间没使用 java, C, C++这一类语言了,很多代码写着都不太顺畅,好在 Android Studio 的提示非常友好,自动补全也很强,所以在了解 API 以及基本的写法后还是可以比较顺畅的完成实验,遇到的问题也主要的一些比较细节的部分,而这些在网上都有很多前辈有解决办法。总之这一次的实验让我对安卓应用开发的兴趣提高了很多,同时也意识到了自己在编程方面的不足,希望之后能学到更多。

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。