

中山大学数据科学与计算机学院本科生实验报告

(2017 年秋季学期)

课程名称：手机应用平台开发

任课教师：刘宁

年级	15 级	专业 (方向)	软件工程 (计应)
学号	15331421	姓名	郑佳锋
电话	13719325472	Email	1773643139@qq.com
开始日期	10.14	完成日期	10.15

一、 实验题目

Lab4

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

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

- (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

密码不能为空

☐ 学生 ☒ 教职工

登录

注册

请输入学号

1234567

请输入密码

密码不能为空

☐ 学生 ☒ 教职工

登录

注册

请输入学号

1234567

请输入密码

.....

☐ 学生 ☒ 教职工

登录

注册

学号或密码错误

确定

请输入学号

123456

请输入密码

.....

☐ 学生 ☒ 教职工

登录

注册

登录成功

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（类似的功能仅演示一遍）



学生切换到教职工：

点击登录：

输入学号登录：



学号或密码错误：

学号密码都正确：



(2) 实验步骤以及关键代码

首先使用 `TextInputLayout` 实现输入学号和密码的控件，只需要在 `EditText` 的外面加一层 `TextInputLayout`，就可以出现动画提示和报错的效果。主要代码如下：

```
<android.support.design.widget.TextInputLayout
    android:id="@+id/input1"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    app:layout_constraintTop_toBottomOf="@+id/sysuiimage"
    android:layout_marginTop="20dp"
    app:layout_constraintRight_toLeftOf="@+id/guideline3"
    app:layout_constraintLeft_toLeftOf="@+id/guideline2"
>
<EditText
    android:id="@+id/inputgrid1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="18sp"
    android:inputType="number"
    android:focusableInTouchMode="true"
    android:hint="请输入学号"
/>
</android.support.design.widget.TextInputLayout>
```

接着，点击图片，弹出对话框。先通过 id 获取图片的对象，再设置 `setOnClickListener` 方法。里面实现 `onClick` 接口。利用 `AlertDialog.Builder` 建立对话框，`setTitle` 设置标题，`setItems` 设置列表选项，每个选项下设置 `Toast` 语句。用 `setNegativeButton` 设置取消按钮，定义取消事件 `click1`，即取消本界面。主要代码如下：

```
ImageView mImage = (ImageView) findViewById(R.id.sysuiimage);
mImage.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        AlertDialog.Builder dialogbuilder = new AlertDialog.Builder(MainActivity.this);
        dialogbuilder.setTitle("上传头像");
        dialogbuilder.setItems(new String[] {"拍摄", "从相册选择"}, (arg0, num) -> {
            switch (num) {
                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;
                default:
                    break;
            }
        });
        dialogbuilder.setNegativeButton("取消", click1);
        AlertDialog dialog = dialogbuilder.create();
        dialog.show();
    }
    private DialogInterface.OnClickListener click1 = (arg0, arg1) -> {
        arg0.cancel();
        Toast.makeText(MainActivity.this, "您选择了[取消]", Toast.LENGTH_SHORT).show();
    }
});
```

接下来，切换 `RadioButton` 的选项，弹出 `Snackbar` 提示。先获取 `RadioGroup` 对象 `mRadiogroup`，再设置 `setOnCheckedChangeListener` 进行监听。利用监听到的 `checkId` 得到哪个按钮被选中，再设置相应的 `Snackbar`。利用 `Snackbar` 的 `setAction` 事件设置“确定”按钮被点击后弹出 `Toast` 信息。

```

final RadioGroup mRadiogroup = (RadioGroup) findViewById(R.id.group);
mRadiogroup.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(RadioGroup group, @IdRes int checkedId) {
        switch(checkedId){
            case R.id.radioButton1:
                Snackbar.make(mRadiogroup, "您选择了学生", Snackbar.LENGTH_SHORT)
                    .setAction("确定", (view) -> {
                        Toast.makeText(MainActivity.this, "Snackbar的确定按钮被点击了", Toast.LENGTH_SHORT).show();
                    })
                    .setActionTextColor(getResources().getColor(R.color.colorPrimary))
                    .setDuration(5000)
                    .show();
                break;
            case R.id.radioButton2:
                Snackbar.make(mRadiogroup, "您选择了教职工", Snackbar.LENGTH_SHORT)
                    .setAction("确定", (view) -> {
                        Toast.makeText(MainActivity.this, "Snackbar的确定按钮被点击了", Toast.LENGTH_SHORT).show();
                    })
                    .setActionTextColor(getResources().getColor(R.color.colorPrimary))
                    .setDuration(5000)
                    .show();
                break;
            default:
                break;
        }
    }
});

```

下一步，点击登录按钮。先获取两个 EditText 中的内容，再进行判断。若学号为空，提示“学号不能为空”；若学号不为空但密码为空，提示“密码不能为空”，若学号和密码都不空，但是任何一个错误，则提示“学号或密码错误”，以上情况均不出现，Snackbar 才提示“登录成功”，点击确定后弹出 Toast 信息。


```

Button logButton = (Button) findViewById(R.id.button1);
logButton.setOnClickListener((view) -> {
    TextInputLayout id = (TextInputLayout) findViewById(R.id.input1);
    TextInputLayout pass = (TextInputLayout) findViewById(R.id.input2);
    EditText idInput = id.getText();
    EditText passInput = pass.getText();
    if (idInput.getText().toString().equals("")){
        id.setErrorEnabled(true);
        id.setError("学号不能为空");
        pass.setErrorEnabled(false);
    }
    else if (!idInput.getText().toString().equals("") && passInput.getText().toString().equals("")){
        pass.setErrorEnabled(true);
        pass.setError("密码不能为空");
        id.setErrorEnabled(false);
    }
    else if (!idInput.getText().toString().equals("123456") || !passInput.getText().toString().equals("6666")){
        id.setErrorEnabled(false);
        pass.setErrorEnabled(false);
        Snackbar.make(mRadiogroup, "学号或密码错误", Snackbar.LENGTH_SHORT)
            .setAction("确定", (view) -> {
                Toast.makeText(MainActivity.this, "Snackbar的确定按钮被点击了", Toast.LENGTH_SHORT).show();
            })
            .setActionTextColor(getResources().getColor(R.color.colorPrimary))
            .setDuration(5000)
            .show();
    }
    else{
        id.setErrorEnabled(false);
        pass.setErrorEnabled(false);
        Snackbar.make(mRadiogroup, "登录成功", Snackbar.LENGTH_SHORT)
            .setAction("确定", (view) -> {
                Toast.makeText(MainActivity.this, "Snackbar的确定按钮被点击了", Toast.LENGTH_SHORT).show();
            })
            .setActionTextColor(getResources().getColor(R.color.colorPrimary))
            .setDuration(5000)
            .show();
    }
});

```

最后，点击注册按钮，利用 Radiogroup 的 `getCheckedRadioButtonId()` 来判断哪个按钮被选中，再弹出相应的 Snackbar。

```

Button regButton = (Button) findViewById(R.id.button2);
regButton.setOnClickListener((view) -> {
    if (mRadiogroup.getCheckedRadioButtonId() == R.id.radioButton1){
        Snackbar.make(mRadiogroup, "学生注册功能尚未启用", Snackbar.LENGTH_SHORT)
            .setAction("确定", (view) -> {
                Toast.makeText(MainActivity.this, "Snackbar的确定按钮被点击了", Toast.LENGTH_SHORT).show();
            })
            .setActionTextColor(getResources().getColor(R.color.colorPrimary))
            .setDuration(5000)
            .show();
    }
    else{
        Snackbar.make(mRadiogroup, "教职工注册功能尚未启用", Snackbar.LENGTH_SHORT)
            .setAction("确定", (view) -> {
                Toast.makeText(MainActivity.this, "Snackbar的确定按钮被点击了", Toast.LENGTH_SHORT).show();
            })
            .setActionTextColor(getResources().getColor(R.color.colorPrimary))
            .setDuration(5000)
            .show();
    }
});

```

(3) 实验遇到困难以及解决思路

首先遇到的困难是不知道对话框的列表选项应该使用什么方法，尝试了 `setMessage`，

setMultiChoiceItems 都不可行，后来翻资料才知道应该使用 setItems 方法。

遇到的下一个头疼的问题是，使用了 TextInputLayout 后，每次运行程序打开界面后，EditText 都会自动获取焦点，在输入框上方显示提示（“请输入学号”）。后来在网上插入两句代码 android:focusable="true" android:focusableInTouchMode="true"，放在 TextInputLayout 控件下依然不可行。几番尝试后，才知道这两句代码应该放在 TextInputLayout 的父控件上面，让父控件获取焦点，从而阻止 EditText 自动获取焦点。

四、 实验思考及感想

这次实验在上一次实验的基础上增加一些内容，包括对话框，Toast，Snackbar，TextInputLayout 和控件基本事件的处理，总体来说不算很难。主要是最开始还不知道 java 代码应该写在哪，后来弄清楚后写起来就不是很困难了。每个控件有许多方法可以用，不熟悉的时候只能根据名字一点一点试。通过这次实验，我明白了查看文档的重要性。盲目的尝试会做很多无用功，使用文档可以快速确定需要的函数，事半功倍！