

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	15352408	姓名	张稼伟
电话	13531810182	Email	709075442@qq.com
开始日期	2017.9.25	完成日期	2017.9.26

一、 实验题目

事件处理

二、 实验目的

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

三、 实验内容

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

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：





(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”。



四、课堂实验结果

(1) 实验截图

入学密码的控





(2) 实验步骤以及关键代码

1.XML 布局部分：输入学号和密码的控件改用 `TextInputLayout` 实现

```

<android.support.design.widget.TextInputLayout
    android:id="@+id/textInputLayout"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginEnd="20dp"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:layout_marginStart="20dp"
    android:layout_marginTop="20dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/imageView"
    app:layout_constraintVertical_bias="0.0"
    >

    <EditText
        android:id="@+id/textid"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/userid_hit"
        android:inputType="number" />

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

```

```

<android.support.design.widget.TextInputLayout
    android:id="@+id/textInputLayout1"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textInputLayout"
    app:layout_constraintVertical_bias="0.0"
    android:layout_marginRight="20dp"
    app:layout_constraintRight_toRightOf="parent"
    android:layout_marginLeft="20dp"
    app:layout_constraintLeft_toLeftOf="parent"
    >

    <EditText
        android:id="@+id/textpassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/password_hit"
        android:inputType="textPassword" />

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

```

2. Java 部分事件处理

- a. 首先声明需用到的用来绑定控件的各种变量:

```

String tmp="";
ImageView mImage=null;
AlertDialog.Builder builder = null;
RadioGroup mRadioGroup=null;
RadioButton RadioButton_student=null;
RadioButton RadioButton_teacher=null;
Button Button_login=null;
Button Button_register=null;
TextInputLayout TextInputLayout_id=null;
TextInputLayout
TextInputLayout_password=null;
EditText EditText_id=null;
EditText EditText_password=null;

```

- b. 在初始化方法中为它们绑定控件

```

public void Init() { //初始化
    mImage = (ImageView) findViewById(R.id.imageView);
    builder = new AlertDialog.Builder(this);
    mRadioGroup = (RadioGroup) findViewById(R.id.radioGroup);
    RadioButton_student = (RadioButton) findViewById(R.id.radioButton1);
    RadioButton_teacher = (RadioButton) findViewById(R.id.radioButton2);
    Button_login = (Button) findViewById(R.id.button);
    Button_register = (Button) findViewById(R.id.button2);
    EditText_id = (EditText) findViewById(R.id.textid);
    EditText_password = (EditText) findViewById(R.id.textpassword);
    TextInputLayout_id = (TextInputLayout) findViewById(R.id.textInputLayout);
    TextInputLayout_password = (TextInputLayout) findViewById(R.id.textInputLayout1);
    AlertDialogInit();
    listenerInit();
}

```

- c. AlertDialog 模块的初始化，包括设置标题为“上传头像”，设置对话列表，列表元素分别是“拍摄”、“从相册选择”并且通过 onClick 方法设置点击它们的事件为抛出 Toast 信息。最后设置取消按钮以及按下它的事件。

```

public void AlertDialogInit() {
    builder.setTitle("上传头像");
    final String[] Choices={"拍摄","从相册选择"};
    builder.setItems(Choices, new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
            Toast.makeText(getApplicationContext(), "您选择了[" +
                Choices[i]+"]", Toast.LENGTH_SHORT).show();
        }
    });
    builder.setNegativeButton("取消", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
            Toast.makeText(getApplicationContext(), "您选择了[取消]",
                Toast.LENGTH_SHORT).show();
        }
    });
    builder.setCancelable(true);
}

```

d. 设置图片监听，点击的时候创建一个 AlertDialog 对话框。

```

public void ImageListenerInit() {
    //图片按钮监听
    mImage.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            AlertDialog dialog=builder.create();
            dialog.show();
        }
    });
}

```

e. 设置 RadioButton 的监听，首先将通过 `getId` 方法判断当前选中的是哪个按钮，然后将它的文本内容保存在字符串 `tmp` 中。调用 `Snackbar` 控件来抛出信息，其 `make` 函数第二个参数消息的内容就是“您选择了”跟 `tmp` 拼接之后的内容。

```

public void RadioButtonListenerInit() {
    //radio 监听
    mRadioGroup.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {
        @Override
        public void onCheckedChanged(RadioGroup group, int checkedId) {
            if (RadioButton_student.getId() == checkedId) { //保存当前选择
                tmp = RadioButton_student.getText().toString();
            }
            if (RadioButton_teacher.getId() == checkedId) {
                tmp = RadioButton_teacher.getText().toString();
            }
            Snackbar.make(mRadioGroup, "您选择了"+tmp, 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();
                    }
                })
                .setActionTextColor(getResources().getColor(R.color.colorPrimary))
                .show();
        }
    });
}

```

```
});
}
```

f.最后是 Button 的事件监听。Button 按钮分登陆按钮 和 注册按钮。其中

注册按钮：设置点击监听器，当点下后通过 Snackbar 抛出 字符串 tmp+ “注册功能尚未启用” 的信息。

登陆按钮：设置点击监听器。

第一步先判断学号密码是不是正确的，若是，则通过 Snackbar 抛出 “登陆成功的信息”；

否则，先将两个 TextInputLayout 控件的错误信息通过 setError 方法设置好。然后判断学号或者密码是否为空，如果为空则将对应的 TextInputLayout 控件的错误信息显示出来，方法是 setErrorEnabled(true)，否则通过 setErrorEnabled(false)隐藏错误信息最后通过 Snackbar 抛出 “学号或密码错误” 的信息。

```
public void ButtonListenerInit() {
    //注册按钮监听
    Button_register.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Snackbar.make(Button_register, tmp + "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();
                    }
                })
                .setActionTextColor(getResources().getColor(R.color.colorPrimary))
                .show();
        }
    });

    //登录按钮监听
    Button_login.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {

            if(EditText_id.getText().toString().equals("123456")&&EditText_password.getText().toStr
ing().equals("6666")) {
                Snackbar.make(Button_login, "登录成功", 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();
                        }
                    })
                    .setActionTextColor(getResources().getColor(R.color.colorPrimary
))
                    .show();
            }
            else{
                TextInputLayout_id.setError("学号不能为空");
                TextInputLayout_password.setError("密码不能为空");
            }
        }
    });
}
```

```

        if(EditText_id.getText().toString().length()==0)
            TextInputLayout_id.setErrorEnabled(true);
        else TextInputLayout_id.setErrorEnabled(false);
        if(EditText_password.getText().toString().length()==0)
            TextInputLayout_password.setErrorEnabled(true);
        else TextInputLayout_password.setErrorEnabled(false);
        Snackbar.make(Button_login, "学号或密码错误", 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();
                }
            })
            .setActionTextColor(getResources().getColor(R.color.colorPrimary))
            .show();
    }
});
}
}

```

(3) 实验遇到困难以及解决思路

这次实验在编程逻辑上并不困难，唯一的难点在于我们不熟悉 java 里如何进行安卓的事件处理编程，经常想到了要做什么事情，但是不知道要调用什么方法去做，这个时候除浏览了老师上课的课件以及实验 PDF 中给出的方法提示，我还自己百度看了一些例子，一个一个方法去查意思，然后再通过尝试得到了想要的效果。

五、课后实验结果

在查 Snackbar 的时候，官方推荐根布局使用 CoordinatorLayout，好处是：

- 1.用户可以滑动（右滑）消除掉 snackbar。
- 2.当 snackbar 出现的时候，布局会移动一些 UI 元素，比如右下角的悬浮按钮会自动上移。

于是我在 XML 文件中加入了下列代码：

```

<android.support.design.widget.CoordinatorLayout
    android:id="@+id/main_content"
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:layout_constraintBottom_toBottomOf="parent">
</android.support.design.widget.CoordinatorLayout>

```

在 java 代码中，声明变量绑定：

```

CoordinatorLayout contain;
contain = (CoordinatorLayout) findViewById(R.id.main_content);
然后将所有 Snackbar.make 的第一个 view 参数改为 contain，如：
Snackbar.make(contain, "学号或密码错误", Snackbar.LENGTH_SHORT)

```

然后运行查看效果发现可以右滑动消除 snackbar 了。



六、 实验思考及感想

这次实验我对 **Android** 中基本的事件编程有了一定的熟悉和了解。对于一个新手来讲，最大的障碍就是不知道用什么 **API**，需要自己一个一个去查，去学习。这个过程虽然会花费比较长的时间，但是最后看到效果之后，一切辛苦都是值得的。未来这些控件还有很多功能等待我去发掘。