

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：郑贵峰

年级	15 级	专业 (方向)	软件工程 (移动信息工程) 互联网方向
学号	15352211	姓名	林苗
电话	13763360840	Email	554562948@qq.com
开始日期	2017.10.14	完成日期	2017.10.17

一、 实验题目

实验二 事件处理

二、 实现内容

Lab2

中山大学学生信息系统



请输入学号

请输入密码

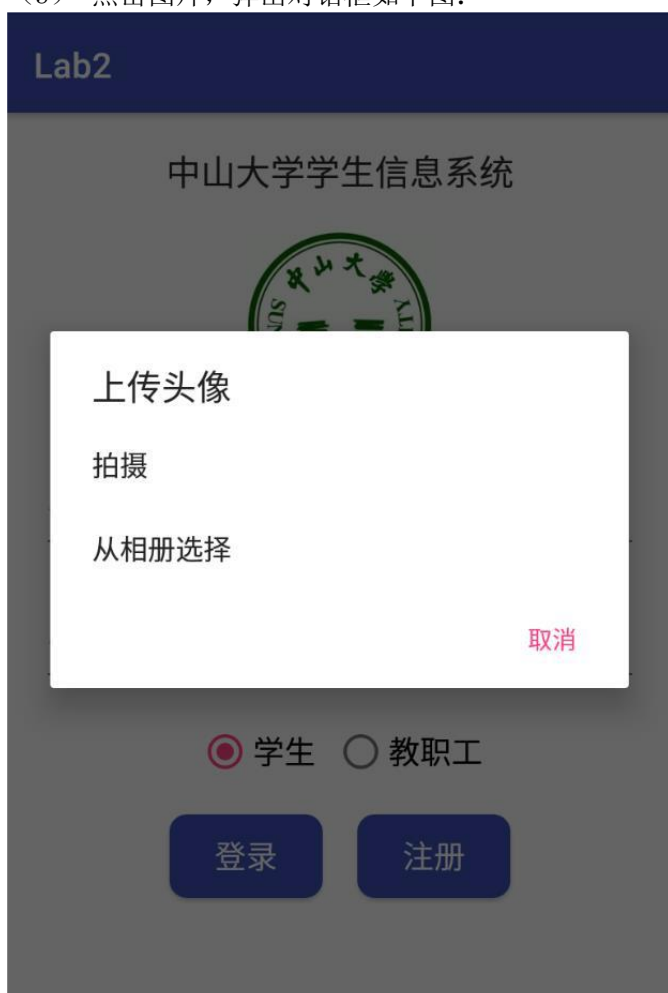
☒ 学生 ☐ 教职工

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：

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

(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

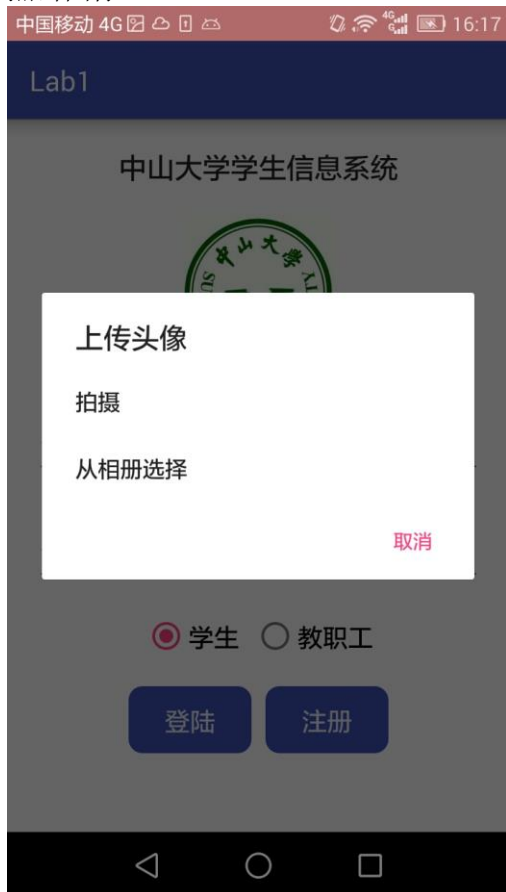
三、 课堂实验结果

(1) 实验截图

进入的第一个界面：



点击图标：



选择[拍摄]:



选择[从相册选择]:



选择[取消]:



切换到[教职工]:



切换到[学生]:



点击[确认]:



没有输入学号时点击登录:



没有输入密码时点击登录:



学号为“123456”且密码为“6666”，点击登录：

请输入学号

123456

请输入密码

.....

☒ 学生

☐ 教职工

登陆

注册

登陆成功

确认

学号或密码错误时：

请输入学号

123456

请输入密码

.....

☒ 学生

☐ 教职工

登陆

注册

学号或密码错误

确认

选择[学生]，点击[注册]：

☒ 学生

☐ 教职工

登陆

注册

学生注册功能尚未启用

确认

选择[教职工]，点击[注册]：



(2) 实验步骤以及关键代码

- 1) 修改 activity_main.xml 文件

将焦点定位在标题：

```
<TextView
    android:id="@+id/textView"
    android:focusable="true"
    android:focusableInTouchMode="true"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="中山大学学生信息系统"
```

使用 TextInputLayout 控件实现学号和密码输入，学号输入的输入类型使用“phone”，可以实现点击时弹出数字输入框：

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

```
<EditText
    android:id="@+id/editText2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="请输入学号"
    android:textColor="@color/colorPrimaryDark"
    android:textSize="18sp"
    android:digits="0123456789"
    android:inputType="phone"/>
```


2) 在 MainActivity.java 文件中:

定义一些内部成员变量:

```
private ImageView Img;
private TextInputLayout Num;
private TextInputLayout PW;
private EditText EditNum;
private EditText EditPW;
private RadioGroup RG;
private RadioButton RB1;
private RadioButton RB2;
private Button Login;
private Button Register;
```

在 protected void onCreate(Bundle savedInstanceState) 中, 初始化成员变量, 获取控件:

```
Img = (ImageView)findViewById(R.id.imageView);
Num = (TextInputLayout)findViewById(R.id.textinputlayout);
PW = (TextInputLayout)findViewById(R.id.textinputlayout1);
EditNum = (EditText)findViewById(R.id.editText2);
EditPW = (EditText)findViewById(R.id.editText);
RG = (RadioGroup)findViewById(R.id.radioGroup);
RB1 = (RadioButton)findViewById(R.id.button1);
RB2 = (RadioButton)findViewById(R.id.radioButton);
Login = (Button)findViewById(R.id.button2);
Register = (Button)findViewById(R.id.button);
```

设置对话框的标题 (setTitle), 对话框的内容 (setItems):

```
AlertDialog.Builder dialog = new AlertDialog.Builder(this);
dialog.setTitle("上传头像");
final String[] Array = new String[] {"拍摄", "从相册选择"};
final int selectedIndex[] = {0};
dialog.setItems(Array, new DialogInterface.OnClickListener ()
```

设置点击选项时对应的操作:

```
dialog.setItems(Array, new DialogInterface.OnClickListener () {

    @Override
    public void onClick(DialogInterface dialog, int which) {
        selectedIndex[0] = which;

        if(selectedIndex[0]==0) {
            Toast.makeText(MainActivity.this, "您选择了[拍摄]", Toast.LENGTH_SHORT).show();
        } else if(selectedIndex[0]==1) {
            Toast.makeText(MainActivity.this, "您选择了[从相册选择]", Toast.LENGTH_SHORT).show();
        }
    }
});
```

设置[取消]按钮以及点击时对应的操作:

```
dialog.setNegativeButton("取消",
    new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            Toast.makeText(MainActivity.this, "您选择了[取消]", Toast.LENGTH_SHORT).show();
        }
    });
```

创建对话框的对象 (.create()), 并在点击图标 (Img) 的时候显示对话框 (.show()):

```
final AlertDialog builder = dialog.create();
Img.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        builder.show();
    }
});
```

实现更改选项时的事件处理:

```
RG.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener {
```

在其中实现 onCheckedChanged 函数:

```
public void onCheckedChanged(RadioGroup group, int checkedID)
```

判断传进来的 checkedID 属于哪一个选项, 设置 Snackbar 的显示内容:

```
Snackbar.make(RG, "您选择了学生", Snackbar.LENGTH_SHORT)
```

当右侧的确认按钮被点击时所进行的操作:

```
.setAction("确认", 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();
    }
});
```

右侧按钮文本的颜色:

```
.setActionTextColor(getResources().getColor(R.color.colorPrimary))
```

Snackbar 弹出时间:

```
.setDuration(5000)
```

显示 Snackbar:

```
.show();
```

选择教职工的实现方法类似。

实现登陆按钮的监听

```
Login.setOnClickListener(new View.OnClickListener() {
```

在内部实现 onClick 函数:

```
public void onClick(View v) {
```

获得学号和密码两个文本的 string:

```
String number = EditNum.getText().toString();
```

```
String password = EditPW.getText().toString();
```

设置 `ErrorEnabled` 为 `false`，否则一次错误就算正确修改也不能消去错误提示：

```
Num.setErrorEnabled(false);
PW.setErrorEnabled(false);
```

依次判断学号、密码是否为空：

```
if (number.isEmpty()) {
    Num.setErrorEnabled(true);
    Num.setError("学号不能为空");
}

else if (password.isEmpty()) {
    PW.setErrorEnabled(true);
    PW.setError("密码不能为空");
}
```

判断学号、密码是否正确：

```
else if (number.equals("123456") && password.equals("6666")) {
    Snackbar.make(Login, "登陆成功", Snackbar.LENGTH_SHORT)
        .setAction("确认", 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();
            }
        })
        .setActionTextColor(getResources().getColor(R.color.colorPrimary))
        .setDuration(5000)
        .show();
}

else if (!number.isEmpty() && !password.isEmpty()) {
    Snackbar.make(Login, "学号或密码错误", Snackbar.LENGTH_SHORT)
        .setAction("确认", 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();
            }
        })
        .setActionTextColor(getResources().getColor(R.color.colorPrimary))
        .setDuration(5000)
        .show();
}
```

设置注册功能的监听事件处理：

```
Register.setOnClickListener(new View.OnClickListener() {
```

实现 `onClick` 函数：

```
public void onClick(View v)
```

如果选择的是[学生]，则弹出 Snackbar:

```
if (RG.getCheckedRadioButtonId() == RB1.getId()) {
    Snackbar.make(Register, "学生注册功能尚未启用", Snackbar.LENGTH_SHORT)
        .setAction("确认", 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()
            }
        })
        .setActionTextColor(getResources().getColor(R.color.colorPrimary))
        .setDuration(5000)
        .show();
}
```

否则弹出 toast 信息:

```
else if (RG.getCheckedRadioButtonId() == RB2.getId()) {
    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

- 1) 进入 app 的第一个界面，焦点集中在学号输入框。

解决思路：将焦点移到标题的文本框中。

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

- 2) 没有输入学号的时候点击登录，重新输入学号之后仍然有“学号不能为空”的提示。

解决思路：一旦检测到 number 为空，就会将 Num 的 ErrorEnabled 设置为 true，因此一直有错误提示。所以在每次遇到点击监听事件时，先把学号和密码的 ErrorEnabled 值赋为 false。

```
Num.setErrorEnabled(false);
PW.setErrorEnabled(false);
```

四、实验思考及感想

这次实验主要的难点是事件处理这一部分的写法。首先要声明一个私有的成员变量，写法如下：

```
private [类型] [名字];
```

如 `private Button Register;`

在 onCreate 函数中，绑定控件：

```
[名字] = ([类型]) findViewById(R.id.[该控件在 xml 中的 id]);
```

如： `Register = (Button) findViewById(R.id.button);`

设置监听函数：

```
[名字].setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        //点击了该控件之后进行的一系列操作
        .....
    }
})
```

}

如:

```
Register.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (RG.getCheckedRadioButtonId() == RB1.getId()) {
            Snackbar.make(Register, "学生注册功能尚未启用", Snackbar.LENGTH_SHORT)
                .setAction("确认", 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();
                    }
                })
                .setActionTextColor(getResources().getColor(R.color.colorPrimary))
                .setDuration(5000)
                .show();
        } else if (RG.getCheckedRadioButtonId() == RB2.getId()) {
            Toast.makeText(MainActivity.this, "教职工注册功能尚未启用", Toast.LENGTH_SHORT).show();
        }
    }
});
```

若是 RadioGroup 的设置更换选项的监听，则变为:

```
RG.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(RadioGroup group, int checkedID) {
```

其中 checkedID 为选择的序号。

弹出 toast 框的写法:

Toast.makeText(MainActivity.this, "[.....]", Toast.LENGTH_SHORT).show();

如: Toast.makeText(MainActivity.this, "您选择了[取消]", Toast.LENGTH_SHORT).show();

Snackbar 的写法:

```
Snackbar.make([控件名], ".....", Snackbar.LENGTH_SHORT)
    .setAction("...", new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            //Snackbar 右侧按钮被点击时的操作
        }
        //设置右侧按钮的颜色
    })
    .setActionTextColor(getResources().getColor(R.color.**))
    //设置 Snackbar 的持续时间，单位为 ms
    .setDuration()
    //显示
    .show();
}
```

如:

```
Snackbar.make(RG, "您选择了学生", Snackbar.LENGTH_SHORT)
    .setAction("确认", 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();
        }
    })
    .setActionTextColor(getResources().getColor(R.color.colorPrimary))
    .setDuration(5000)
    .show();
```

对话框的写法:

AlertDialog.Builder 对象名=new AlertDialog.Builder(this);

对象名.setTitle("标题");

//设置选项

```
.setItems(Array, new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface 对象名, int 选项值{//事件处理}
})
```

//设置取消按钮

```
.setNegativeButton("...", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface 对象名, int 选项值{//事件处理}
})
```

//创建对象

final AlertDialog 新对象名 = 对象名.create();

//调用

新对象.show();

如:

```
AlertDialog.Builder dialog = new AlertDialog.Builder(this);
dialog.setTitle("上传头像");
final String[] Array = new String[] {"拍摄", "从相册选择"};
final int selectedIndex[] = {0};
dialog.setItems(Array, new DialogInterface.OnClickListener() {

    @Override
    public void onClick(DialogInterface dialog, int which) {
        selectedIndex[0] = which;

        if(selectedIndex[0]==0) {
            Toast.makeText(MainActivity.this, "您选择了[拍摄]", Toast.LENGTH_SHORT).show();
        }else if(selectedIndex[0]==1) {
            Toast.makeText(MainActivity.this, "您选择了[从相册选择]", Toast.LENGTH_SHORT).show();
        }
    }
});
```

```

        dialog.setNegativeButton("取消",
            new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    Toast.makeText(MainActivity.this, "您选择了[取消]", Toast.LENGTH_SHORT).show();
                }
            });
        final AlertDialog builder = dialog.create();
        builder.show();
    }
}

```

实验感想

实验一开始还是不知从何入手，但通过文档以及一些参考网上的一些小例子，先在 AndroidStudio 上打一遍，试的多了也能依样画葫芦做出我们实验需要的东西。后来整个实验完成了个大概之后，再重新看回整份代码，对每个模块该怎么写的思路也渐渐清晰了。一些细节的点在和同学讨论之后也得以完美解决。看例子、与同学讨论还是很高效的学习方法。

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。