

# 中山大学移动信息工程学院本科生实验报告

## ( 2016 年秋季学期 )

课程名称：移动应用开发

任课教师：郑贵锋

年级	14	专业 ( 方向 )	物联网
学号	14353259	姓名	商家煜
电话	13211143626	Email	shangjy3@mail2.sysu.edu.cn
开始日期	2016.9.22	完成日期	2016.9.22

### 一、 实验题目

#### 1) 基本 UI 界面设计

### 二、 实现内容

实现一个Android应用，界面呈现如下效果：



图：样例效果图

要求：

(1) 该界面为应用启动后看到的第一个界面

(2) 点击登录按钮：

如果用户名为空，弹出Toast信息“用户名不能为空”；

如果密码为空，弹出Toast信息“密码不能为空”；

如果用户名输入为“Android”，密码为“123456”，弹出如下对话框：



点击“确定”，弹出Toast信息——对话框“确定”按钮被点击。，点击“取消”，弹出Toast信息——对话框“取消”按钮被点击。

否则，弹出如下对话框：



确定和取消按钮点击后效果参见上一个对话框

点击注册按钮：

如果RadioButton选中的是“学生”，那么弹出Toast信息“学生身份注册功能尚未开启”，如果选中的是“教师”，那么弹出Toast信息“教师身份注册功能尚未开启”，以此类推。

RadioButton选择项切换：选择项切换之后，弹出Toast信息“XX身份被选中”，例如从学生切换到教师，弹出Toast信息“教师身份被选中”

### 三、 课堂实验结果

#### （1） 实验截图



图：效果图

## (2) 实验步骤以及关键代码

本次实验内容实在上一次实验的基础上完成的，下面将讲述本次实验的新增内容的步骤

### 2) 添加 Toast 功能

```
final RadioButton radiobutton1 = (RadioButton)findViewById(R.id.id1);
radiobutton1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Toast.makeText(getApplicationContext(), radiobutton1.getText().toString()+"身份被选中", Toast.LENGTH_LONG).show();
    }
});
```

radiobutton 是 TextView 的子类, 具有 TextView 的所有属性。用户可以通过单击 Button 来触发一系列事件, 然后为 Button 注册监听器来实现其监听事件。

通过 findViewById 函数来找到指定的控件, 然后通过强制装换, 变成 RadioButton 类。通过调用该类中的方法 setOnClickListener 监听器来实现监听当前 RadioButton 是否被点击, 并重写 onClick 函数才实现时间功能

其中, 在 RadioButton 被点击之后, 我们需要产生一个 toast 提醒消息, 方法为 makeText, 包含有三个参数, 分别为: context, text, duration。context 是一个上下文类型, 写上使用这个方法的 java 类名加上 this 即可, text 是 Toast 要显示的信息, duration 是 Toast 显示的时间长度, 有 Toast.LENGTH\_SHORT 和 Toast.LENGTH\_LONG 可选。最后再调用 show 来将该信息显示出来。

同理, 我们可以将实验要求的所有需要提示的步骤都通过这种方式进行代码的编写

### 3) 添加 Button 事件处理

```
Button zhuce = (Button) findViewById(R.id.zhuce);
zhuce.setOnClickListener((v) -> {
    if(radiobutton1.isChecked()){
        Toast.makeText(getApplicationContext(), radiobutton1.getText().toString()+"身份注册功能未开通", Toast.LENGTH_LONG).show();
    }
    if(radiobutton2.isChecked()){
        Toast.makeText(getApplicationContext(), radiobutton2.getText().toString()+"身份注册功能未开通", Toast.LENGTH_LONG).show();
    }
    if(radiobutton3.isChecked()){
        Toast.makeText(getApplicationContext(), radiobutton3.getText().toString()+"身份注册功能未开通", Toast.LENGTH_LONG).show();
    }
    if(radiobutton4.isChecked()){
        Toast.makeText(getApplicationContext(), radiobutton4.getText().toString()+"身份注册功能未开通", Toast.LENGTH_LONG).show();
    }
});
```

首先实现的是比较简单的注册按钮的功能，同样通过 findViewById 的方式来选中指定 id 的控件，然后，因为注册功能是不需要弹出对话框的，只是简单的添加 Toast 功能即可，因此基本过程同上一个步骤一致

```
Button Deng = (Button) findViewById(R.id.dengl);
final EditText username = (EditText) findViewById(R.id.name);
final EditText password = (EditText) findViewById(R.id.key);
Deng.setOnClickListener((v) -> {
    if(TextUtils.isEmpty(username.getText().toString())){
        Toast.makeText(MainActivity.this, "用户名为空", Toast.LENGTH_SHORT).show();
    }
    else if(TextUtils.isEmpty(password.getText().toString())){
        Toast.makeText(MainActivity.this, "密码为空", Toast.LENGTH_SHORT).show();
    }

    else if(TextUtils.equals(username.getText().toString(), "Android")) {
        if (TextUtils.equals(password.getText().toString(), "123456"))
            alertDialog.show();
        else
            alertDialog1.show();
    }
    else
        alertDialog1.show();
});
```

在点击登录按钮的时候，根据题目要求我们需要判断用户的登录信息是否合法：首先我们判断用户名是否为空，若空则通过 Toast 弹出提示，然后再用同样的方式判断密码是否为空。如果前两个条件满足，则判断输入的信息是否与要求相同，如果相同则弹出登录成功的对话框，否则弹出登录失败的对话框

### 4) 添加简单对话框

```
final AlertDialog.Builder alertDialog = new AlertDialog.Builder(this);
alertDialog.setTitle("提示").setMessage("登录成功")
).setPositiveButton("确定",
    new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            Toast.makeText(getApplicationContext(), "对话框“确定”按钮被点击", Toast.LENGTH_SHORT).show();
        }
    }).setNegativeButton("取消",
    new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            Toast.makeText(getApplicationContext(), "对话框“取消”按钮被点击", Toast.LENGTH_SHORT).show();
        }
    }).create();
```

通过 AlertDialog.Builder 类来实现简单对话框的功能，其中包含以下功能：

setTitle()设置对话框标题;

setIcon()设置对话框图标;

setMessage()设置对话框内容;

setItems()设置要在对话框中显示的项目列表;

setView()设置自定义的对话框样式;  
setSingleChoiceItems()设置对话框显示一个单选框;  
setMultiChoiceItems()设置对话框显示一系列的复选框;  
setPositiveButton()设置对话框显示一个“确定”按钮;  
setNeutralButton()设置对话框显示一个“退出”按钮;  
setNegativeButton()设置对话框显示一个“取消”按钮;  
create()创建一个 AlertDialog 对话框;  
show()显示对话框;

此处我们将两个对话框分别设置为登录成功，并监听两个响应的确定和取消按钮是否被点击，若点击则弹出 Toast 提示消息即可。最后需要留意在编写完整整个对话框之后调用 show 函数来显示对话框

### (3) 实验遇到困难以及解决思路

本次实验课内容比较简单，基本上没有遇到什么困难。需要注意的是自己的命名习惯问题，有时候调用实例的时候可能会把两个不同的实例弄混，导致一些不必要的错误

## 四、 课后实验结果

### 1) 实验截图



## 2) 实验步骤

### ● 主界面

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.interference);  
  
    Button button1 = (Button) findViewById(R.id. oldtype);  
    button1.setOnClickListener((v) -> {  
        Intent intent = new Intent();  
        intent.setClass(Inteference.this, MainActivity.class);  
        startActivity(intent);  
    });  
  
    Button button2 = (Button) findViewById(R.id. newtype);  
    button2.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View v) {  
            Intent intent = new Intent();  
            intent.setClass(Inteference.this, Second_activaty.class);  
            startActivity(intent);  
        }  
    });  
}
```

主界面中简单的设置两个按钮，在 java 编程中，通过设置监听器来判断是否按到按钮，然后通过调用别的界面的类来实现页面的转换。其中用到了 Intent 类，通过方法 setClass(当前界面，需要跳转的界面)来设定跳转方法。最后通过 startActivity 方法来进行跳转

### ● 添加界面

```
<activity android:name=".Inteference">  
    <intent-filter>  
        <action android:name="android.intent.action.MAIN"/>  
        <category android:name="android.intent.category.LAUNCHER"/>  
    </intent-filter>  
</activity>  
  
<activity android:name=".MainActivity">  
    <intent-filter>  
        <action android:name="com.litreily.MainActivity" />  
        <category android:name="android.intent.category.DEFAULT" />  
    </intent-filter>  
</activity>  
  
<activity android:name=".Second_activaty">  
    <intent-filter>  
        <action android:name="com.litreily.SecondActivity"/>  
        <category android:name="android.intent.category.DEFAULT"/>  
    </intent-filter>  
</activity>
```

通过在 AndroidManifest.xml 中添加页面，这里我们将 Inteference 设置为主界面，开启程序时默认第一个打开，然后设置两个界面分别为新的界面（通过 Snackbar 等新的界面来进行编辑）和旧界面（通过课程要求进行编辑）

## ● Snackbar

```
Button zhuce = (Button) findViewById(R.id.zhuce);
zhuce.setOnClickListener((v) -> {
    if(radiobutton1.isChecked()){
        Snackbar.make(v, radiobutton1.getText().toString()+"身份注册功能未开通", Snackbar.LENGTH_LONG).show()
    }
    if(radiobutton2.isChecked()){
        Snackbar.make(v, radiobutton2.getText().toString()+"身份注册功能未开通", Snackbar.LENGTH_LONG).show()
    }
    if(radiobutton3.isChecked()){
        Snackbar.make(v, radiobutton3.getText().toString()+"身份注册功能未开通", Snackbar.LENGTH_LONG).show()
    }
    if(radiobutton4.isChecked()){
        Snackbar.make(v, radiobutton4.getText().toString()+"身份注册功能未开通", Snackbar.LENGTH_LONG).show()
    }
});
```

类似于 Toast 的方法，我们通过 snackbar 的方法 make 来生成一个提示功能。参数包括显示的 view，需要显示的字符串以及显示的时间长度。最后通过 show 方法来实现展示。

## ● TextInputLayout

```
<android.support.design.widget.TextInputLayout
    android:id="@+id/more_username"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="10dp"
    android:gravity="center"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp">

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="请输入用户名"
        android:gravity="center"/>

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

TextInputLayout 也是 Android Design Support Library 中的一个控件，用于接受用户输入，在 xml 文件中我们先如图设置这个控件。

```
if(usernameText.getText().getText().toString().isEmpty()){
    usernameText.setErrorEnabled(true);
    usernameText.setError("用户名为空");
    Snackbar.make(getWindow().getDecorView(), "用户名为空", Snackbar.LENGTH_SHORT).show();
}
```

通过其内置的方法 getText() 来获取 EditText 中的内容，然后通过与之前 EditText 模块相同的方法来判断输入是否合法。同时我们也可以使用其自带的错误提示方法来判断当前输入的信息是否为空（即使用 setErrorEnabled 函数和 setError 函数来显示错误信息）

## 五、实验思考及感想

本次实验第一次使用了 java 编程的方法来实现一些简单的事件处理功能包括监听、消息提示、简单对话框等实现。通过听老师上课的讲解以及 ppt 上的介绍能够初步的掌握所需要使用到的基本语法和实现方式。然后课后有不懂的，例如如何新建页面等问题，通过百度都有了进一步的学习和了解。课后将继续进一步实现课程上讲解的如自动补充文字等内容，希望能将页面做的更美观更方便实用