移动应用开发实验报告

lab2：事件处理

姓名	学号	年级	专业	电话	邮箱	起止日期
王若曦	15352319	15级	软件工程	15354222552	1511330303@qq.com	2017.10.09-2017.10.10

实验题目

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，能对确定和取消的事件做处理

实验内容

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
> (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：

(4) 切换RadioButton的选项， 弹出Snackbar提示“您选择了xx”；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，情况如下：

(6)点击注册按钮

实验过程及结果

首先是对事件的处理，即点击对应控件时如何触发响应的事件。
为控件注册监听有两种方法：

1. 在布局文件中，为对应控件设置OnClick属性，然后在java代码中添加一个public void OnClick参数值{方法}

```
android:onClick="clicktp"
    (layout.xml)
public void clicktp(view v){
    method;
}
    (Activity.java)
```

1. 在java代码中绑定匿名监听器，并且重写onClick方法

```
Button btn = (Button) findViewById(R.id.button)
btn.setOnClickListener(new View.OnClickListener(){
  @Override
  public void onClick(View view){
    method;
  }
})
    (Activity.java)
```

这次实验我使用的都是第一种方法，感觉比较简单，代码写起来比较清晰。

对于ImageView的处理

点击图片后要求显示一个对话框AlertDialog，询问拍摄或者从相册选择，右下角有一个取消按钮。
首先，通过AlertDialog.Builder方法在原布局创建一个对话框对象builder。

```
AlertDialog.Builder builder = new AlertDialog.Builder(lab1.this);
```

调用setTitle()方法设置对话框标题。

```
builder.setTitle("上传头像");
```

这里要设置两个选项，分别为拍摄和从相册选择，所以要使用setItem()方法设置多个可点击的选项。这里选项上的文字通过外部定义的String数组变量设置。这里select数组里的值即各个选项。

```
final String[] select = {"拍摄", "从相册选择"};
        //    设置一个下拉的列表选择项
        builder.setItems(select, new DialogInterface.OnClickListener()
        {
            @Override
            public void onClick(DialogInterface dialog, int which)
            {
                Toast.makeText(lab1.this, "您选择了" + select[which], Toast.LENGTH_SHORT).show();
            }
        });
```

说一下我对这里final关键字的理解：因为在setItems方法中要使用string变量，而string定义在setItems方法外，所以为了使该内部类使用外部方法的变量，就要给外部方法的变量加上final来保证在该方法中只有一个select，包括它的内部类。

还有一个取消按钮通过setNegativeButton()方法来设置。

```
builder.setNegativeButton("取消", new DialogInterface.OnClickListener()
                {
                    @Override
                    public void onClick(DialogInterface dialog, int which)
                    {
                        Toast.makeText(lab1.this, "您选择了取消" , Toast.LENGTH_SHORT).show();
                    }
                });
```

这里要注意对话框AlertDialog中有三种button可供设置：
setPositiveButton() ; setNegativeButton() ; setNeutralButton().
但是每种button最多只能设置一个，也就是说一个对话框中最多有三个button。至于这三个button的区别，目前我看来是出现位置的不同，其中PositiveButton和NegativeButton并列位于右下角，NeutralButton单独位于左下角。
然后最简单的弹出信息是使用Toast.makeText()方法，三个参数分别为：上下文类型，即使用这个方法的java类名加上.this；要显示的信息；以及Toast显示时间长度，可选长或短。最后调用.show ()方法进行显示。

对于RadioButton的处理

点击RadioButton中的两个单选按钮时显示信息不是利用Toast，而是Snackbar。这是一种比Toast更高级的消息显示方法，比如它可以通过右滑来消除，而且还可以设置按钮如这里的“确定”。
但是高级是要付出代价的，那就是要配置多一个android.support:design，包括后面用到的TextInputLayout控件也是在这个库了的。在build.gradle(moudle)里添加

```
com.android.support    25.3.1
```

这里我用了25.3.1，没有按照实验文档上的26.0.0，只是为了和appcompat版本保持一致，具体问题在后面谈。（只贴出学生按钮的代码）

```
public void clickxs(View v){
        mod = 0; //登录模式为学生
        Snackbar.make(v,"您选择了学生",Snackbar.LENGTH_SHORT)
                .setAction("确定",new View.OnClickListener() {
                    @Override
                    public void onClick(View view) {
                        Toast.makeText(lab1.this, "Snackbar的确定按钮被点击了" , Toast.LENGTH_SHORT).show();
                    }
                })
                .show();
    }
```

Snackbar.make()方法中三个参数分别是根布局，显示的信息和显示时长。.setAction方法用来设置右侧按钮被点击时的事件，两个参数分别为按钮的文本和处理点击事件的逻辑。这里再嵌套一个Toast来显示“确定按钮被点击”的信息。
Snackbar还有其他方法，如setActionTextColor()方法设置右侧按钮颜色；setDuration()方法设置信息弹出时间，比上面的short或long更具体。
这里的全局变量mod是用来在后面登录的时候判断登录模式是学生还是教职工。

对于登录button的处理

题目要求学号和密码的输入框用TextInputLayout实现，这也是Android design support library中的一个控件，也是比EditText更高级，比如提供了输入提示，报错等功能。
首先在布局文件中，要用TextInputLayout控件把原来的EditText控件包起来，然后把原EditText控件的属性移植到包住它的

TextInputLayout控件中，主要是其相对于其他控件的位置布局，同时其他控件要相对于这个TextInputLayout控件进行布局。

```
<android.support.design.widget.TextInputLayout
        android:id="@+id/sno"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginRight="20dp"
        android:layout_marginTop="20dp"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView3"
        android:layout_marginEnd="20dp"
        >
    <EditText
        android:id="@+id/editText"
        android:layout_width="251dp"
        android:layout_height="wrap_content"
        android:layout_marginRight="20dp"
        android:layout_marginTop="20dp"
        android:hint="请输入学号"
        android:inputType="number"
        android:textSize="18sp"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView3"
        android:layout_marginEnd="20dp" />
    </android.support.design.widget.TextInputLayout>
```

有趣的是使用TextInputLayout控件后EditText中原来的text属性自动变为了hint。
java代码中，通过TextInputLayout获取输入框中输入的内容并转化为string类型，然后对其进行判断是否为空或是否正确。

```
TextInputLayout sno = (TextInputLayout) findViewById(R.id.sno);
TextInputLayout pwd = (TextInputLayout) findViewById(R.id.pwd);
String username =sno.getEditText().getText().toString();
String password =pwd.getEditText().getText().toString();
```

对输入内容的判断就是很简单的逻辑：先判断学号是否为空，else判断密码是否为空，else判断学号是否正确，若错误则弹出Snackbar信息，若正确则判断密码是否正确并弹出对应Snackbar信息。
要注意的是对获取的输入内容username和passw ord的判断不能之间写 "== null" 或 "== "123456"""，要通过对应方法。如判断是否为空：

```
if(TextUtils.isEmpty(username))
```

```

```

判断内容与某值是否相等(username == "123456")：

```
if(TextUtils.equals(username,"123456"))
```

```

```

还有一点是错误信息的设置通过setError()和setErrorEnabled()。其中setErrorEnabled用来设置是否显示错误信息(true为显示，false为不显示)。这里要记得输入正确后把setErrorEnabled设为false。

```
        if(TextUtils.isEmpty(username)){
            sno.setError("学号不能为空");
            sno.setErrorEnabled(true);
        }
        else{
            sno.setErrorEnabled(false);
            if(TextUtils.isEmpty(password)){
                pwd.setError("密码不能为空");
                pwd.setErrorEnabled(true);
            }
```

再在逻辑里嵌套Snackbar和Toast即可，这里不再贴完整代码。

对于注册button的处理

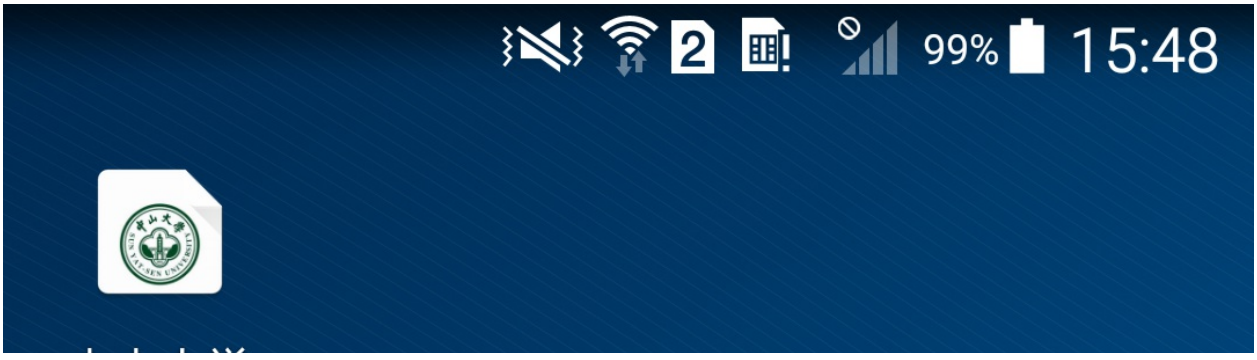
注册button主要是要区分单选按钮选中的是学生还是教职工。这里在外部定义了一个int类型变量mod用来保存选中的选项，即登录模式，0为学生，1为教职工。

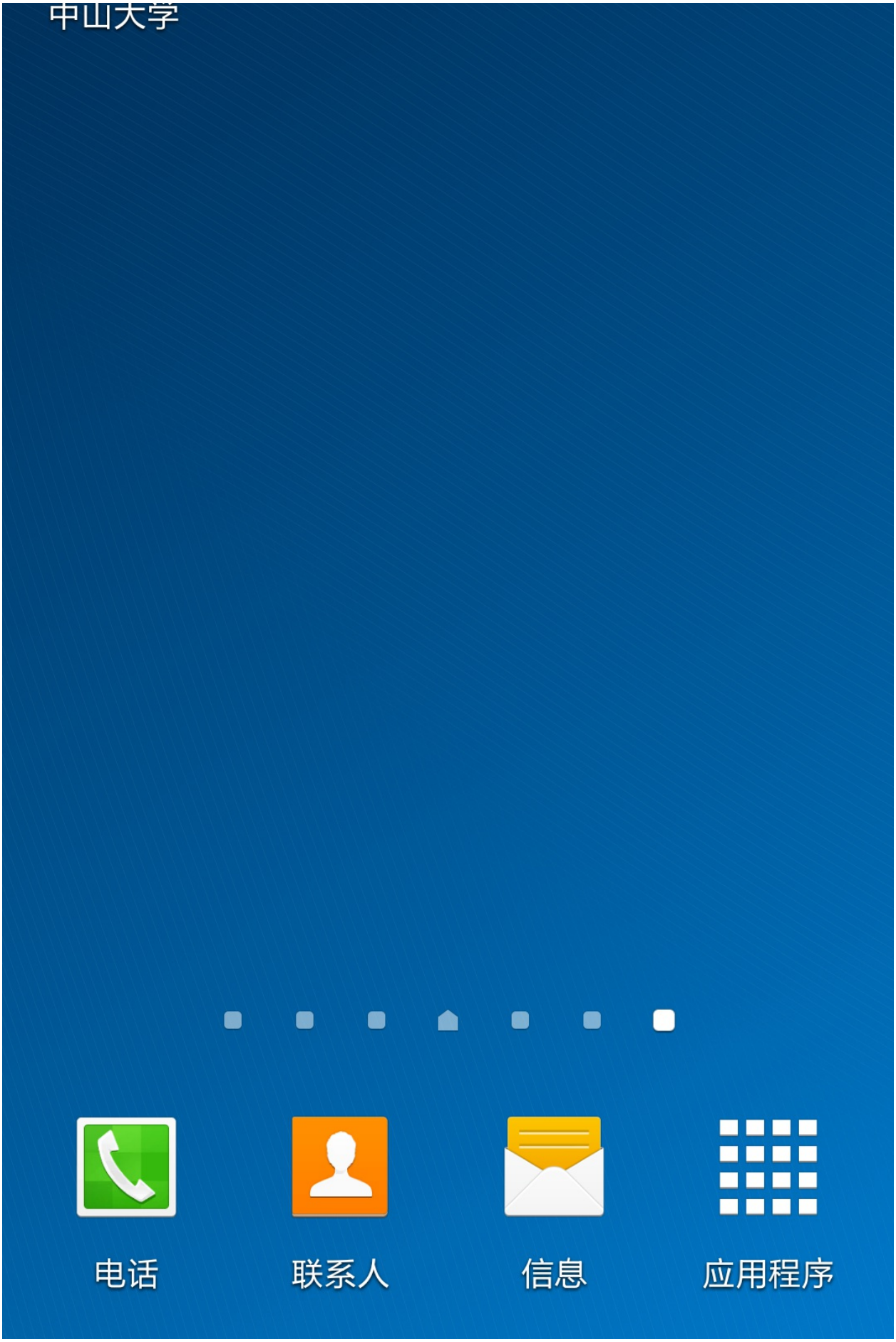
```
public void clickzc(View v){
        if(mod == 0){
            Snackbar.make(v,"学生注册功能尚未启用",Snackbar.LENGTH_SHORT)
                    .setAction("确定",new View.OnClickListener() {
                        @Override
                        public void onClick(View view) {
                            Toast.makeText(lab1.this, "Snackbar的确定按钮被点击了" , Toast.LENGTH_SHORT).show();
                        }
                    })
                    .show();
        }
        else if(mod == 1){
          Toast.makeText(lab1.this, "教职工的注册功能尚未启用" , Toast.LENGTH_SHORT).show();
            /*Snackbar.make(v,"教职工注册功能尚未启用",Snackbar.LENGTH_SHORT)
                    .setAction("确定",new View.OnClickListener() {
                        @Override
                        public void onClick(View view) {
                            Toast.makeText(lab1.this, "Snackbar的确定按钮被点击了" , Toast.LENGTH_SHORT).show();
                        }
                    })
                    .show();*/
        }
    }
```

这里直接把教职工的信息也设置成了Snackbar，仔细看了实验文档后改成了Toast。

设置app图标

个人感觉默认图标有点丑，所以我就改了一个图标，虽然也是很丑。
添加图标图片的方法和添加其他图片不太一样，因为launch图标要适应不同的分辨率，所以要添加多种分辨率的launch图标。
右键 mipmap->new->image asset->选择launcher icon，选择图片，设置样式即可。
效果如图：




中山大学
电话
联系人
信息
应用程序

实验结果


99%
15:48
中山大学
中山大学学生信息系统
SUN YAT-SEN UNIVERSITY
请输入学号
请输入密码
学生
教职工
登录
注册

实验思考及感想

本次实验中遇到的问题：

1. 不知道响应代码应该写在哪里。因为第一次接触AS和java，上次单纯UI布局都是写在xml的布局文件中，甚至都是通过design拖动控件完成，所以这次既要写布局，又要写java代码处理事件，所以不知道代码要写在哪里。尤其是在java代码中不知道定义方法如public void click()要写在哪里。试着在onCreate函数后面并列写了一个button的click触发方法，成功运行后就知道要写在哪里了。
2. 对话框的布局。一开始看实验文档的时候以为对话框只能设置title，message和button，就拿那三种button来设了拍摄，从相册选择和取消，但是布局真是丑到爆，然后想是不是还要自己定义对话框的布局，但是实验文档没有提到要自定义，而且我自己写了自定义布局后LayoutInflater又无法resolve，不知道什么原因。然后想到是不是对话框还有其他属性可以用来设选项。查了资料后发现这种多选项按钮可以通过setItems方法来设置，其中要在方法外部用final关键字定义一个string类型数组变量，用来作为各选项的文本。这样就不用自定义对话框布局实现和实验文档一样的布局。
3. 配置android support design.为了使用高级方法Snackbar和TextInputLayout需要配置android support design来支持。实验文档中给出的是26.0.0版本，但是我全改成26后出现了好多问题，还要把API改了，而且我没下载android 8.0，所以干脆就接着用25版本的，感觉已经够用。这里要注意要和android.support:appcompat-v7保持一致，而且build.gradle上面的compileSdkVersion和buildToolsVersion等也要跟着改，否则会报错。
4. TextInputLayout布局。使用TextInputLayout进行布局时，要把它所包含的EditText的相对位置属性移植到TextInputLayout中，同时其他相对于原EditText布局的控件要将原EditText的id改为TextInputLayout的id，否则将会全挤在顶部。这里注意一个TextInputLayout只能包含一个EditText。

本次实验进行了简单的事件处理，同时对java也有了一定的了解，知道了代码怎么写。其实写完之后感觉和C++差不多，只是语法存在一些差异，也有可能是我还没有完整掌握，只是目前来看是这样。Android确实挺有趣的。