

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	15 级	专业 (方向)	软件工程(移动)
学号	15352133	姓名	黄少豪
电话	13727024545	Email	328730316@qq.com
开始日期	2017.10.13	完成日期	2017.10.14

一、 实验题目

事件处理

二、 实现内容

1. 实现一个 Android 应用，界面与下图一致:



2. 点击图片，弹出对话框



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]”

3. 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”

4. 点击登录按钮，若学号为空，则下图效果:

请输入学号

学号不能为空

请输入密码

☐ 学生 ☒ 教职工

若密码为空，则下图效果:

请输入学号

1234567

请输入密码

密码不能为空

☐ 学生 ☒ 教职工

若学号为 123456，密码为 6666，则下图效果:

请输入学号

123456

请输入密码

....|

☐ 学生 ☒ 教职工

登录成功 [确定](#)

其他情况:

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

5. 点击注册按钮，若选中的是学生，则有以下图效果:

☒ 学生 ☐ 教职工

登录 注册

学生注册功能尚未启用 确定

若选中的是教职工，那么弹出 Toast 信息“教职工注册功能尚未启用”。

三、 课堂实验结果

(1) 实验截图













(2) 实验步骤以及关键代码

1. 使用 TextInputLayout 控件输入学号与密码

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

    <EditText
```

修改 activity_main.xml 文件，使 EditText 包在一个 TextInputLayout 内部。

2. 点击图片，弹出对话框

修改 activity_main.xml 文件，在 ImageView 中添加属性 android:onClick=“imageClick”，表示点击图片时触发 imageClick 函数，下图是 imageClick 函数的实现：

```
public void imageClick(View view) {
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setTitle("上传头像");
    final String[] items = {"拍摄", "从相册选择"};
    builder.setItems(items, (dialog, which) -> {
        dialog.dismiss();
        Toast.makeText(MainActivity.this, "您选择了[" + items[which] + "]",
            Toast.LENGTH_SHORT).show();
    });
    builder.setNegativeButton("取消", (dialog, which) -> {
        dialog.dismiss();
        Toast.makeText(MainActivity.this, "您选择了[取消]",
            Toast.LENGTH_SHORT).show();
    });
    builder.create().show();
}
```

通过 AlertDialog.Builder 类的构造函数定义一个对话框创建对象 builder；通过 builder 的 setItems 函数在对话框内部创建一个列表；通过 builder 的 setNegativeButton 函数在对话框内部设置一个取消按钮；dialog.dismiss() 函数表示使对话框消失。

3. 切换 RadioButton 的选项，弹出 Snackbar 提示“您选择了 xx”

首先在 activity_main.xml 中定义一个 CoordinatorLayout 布局控件

```
<android.support.design.widget.CoordinatorLayout
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    app:layout_constraintBottom_toBottomOf="parent"/>
```

定义一个 showSnackbar 函数，message1 表示 Snackbar 显示的消息，message2 表示点击 Snackbar 的确定按钮后 Toast 的消息，boo 变量确定点击确定按钮后是否 Toast 消息。

```

public void showSnackbar(String message1, final String message2, final boolean boo) {
    coordinatorLayout = (CoordinatorLayout) findViewById(R.id.container);
    Snackbar.make(coordinatorLayout, message1, Snackbar.LENGTH_SHORT)
        .setAction("确定", (view) -> {
            if (boo == true)
                Toast.makeText(MainActivity.this, message2, Toast.LENGTH_SHORT).show();
        })
        .setActionTextColor(getResources().getColor(R.color.colorPrimary))
        .setDuration(5000)
        .show();
}

```

coordinatorLayout 用来确定 Snackbar 的布局；setAction 函数用来设置一个确定按钮，点击按钮时会根据 boo 值而决定是否弹出一个 Toast 消息；setActionTextColor 函数用来设置确定按钮的颜色；setDuration 用来设置 Snackbar 无操作时出现的持续时间。

接着定义一个 radioGroupCheck 函数，checkedId 参数表示被点击的 RadioButton 的 id。

```

public void radioGroupCheck() {
    mainRg = (RadioGroup) findViewById(R.id.radio_group);
    mainRg.setOnCheckedChangeListener((group, checkedId) -> {
        if (checkedId == R.id.student) {
            showSnackbar("您选择了学生", "Snackbar的确定按钮被点击了", true);
        }
        else{
            showSnackbar("您选择了教职工", "Snackbar的确定按钮被点击了", true);
        }
    });
}

```

4. 点击登陆或注册按钮，在 activity_main.xml 中给两个 Button 控件添加属性 android:onClick="buttonClick" 表示当点击按钮时触发

buttonClick 函数，以下是 buttonClick 函数的实现：

根据 id 获取界面中的输入框，默认设置输入框不显示错误消息；view 参数表示的是被点击的按钮。

```

public void buttonClick(View view) {
    username = (TextInputLayout) findViewById(R.id.username$rapper);
    password = (TextInputLayout) findViewById(R.id.password$rapper);
    username.setErrorEnabled(false);
    password.setErrorEnabled(false);
}

```

当被点击的 view 的 id 等于登录按钮的 id 时，先获取输入框中的学号与密码字符串，判断两个字符串的值。若学号或密码为空则输入框显示错误消息；否则若学号密码正确则用一个 Snackbar 显示登录成功，若不正确则用一个 Snackbar 显示学号或密码不正确，代码见下图：

```

if (view.getId() == R.id.login) {
    String user = username.getEditText().getText().toString();
    String pass = password.getEditText().getText().toString();
    if (user.isEmpty()) {
        username.setErrorEnabled(true);
        username.setError("学号不能为空");
        return;
    }
    if (pass.isEmpty()) {
        password.setErrorEnabled(true);
        password.setError("密码不能为空");
        return;
    }
    if (user.equals("123456") && pass.equals("6666")) {
        showSnackbar("登陆成功", "", false);
    }
    else{
        showSnackbar("学号或密码错误", "", false);
    }
}
}

```

当被点击的按钮是注册按钮时，若当前单选按钮选中的是学生，则用一个 Snackbar 显示学生注册功能尚未启用；否则就用一个 Toast 显示教职工注册功能尚未启用。代码如下：

```

else{
    mainSt = (RadioButton) findViewById(R.id.student);
    if (mainSt.isChecked()) {
        showSnackbar("学生注册功能尚未启用", "", false);
    }
    else{
        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. 在实现点击图片出现对话框时，点击很大片范围均会认为点中了图片，弹出对话框。解决方法是将 activity_main.xml 中 ImageView 的 layout_width 属性从 0dp 改成一个具体数值(使用 0dp 是为了使图片保持长宽比例)。
2. 在实现点击图片出现对话框时，使用 AlertDialog.Builder 的构造函数时，参数最好使用 MainActivity.this 代替 this，否则在某些情况会出现错误，比如在嵌套的内部类下使用 this。
3. 在使用 Snackbar 时，第一个参数推荐是使用 CoordinatorLayout 类对

象，所以需要先在 activity_main.xml 文件定义一个 CoordinatorLayout 类对象，然后在 java 程序中通过 findViewById 方法调用这个对象作为 Snackbar 的参数。

4. app 启动时默认第一个 EditText 是获得焦点的，为了解决这个问题可以通过在 ConstraintLayout 中加入两个属性 focusable 和 focusableInTouchMode 并将它们设置为 true。这里个人认为是使布局中所有位置均可获取焦点的意思从而在 app 启动时使焦点转移至某个非可编辑位置。

5. EditText 获取焦点后，还需要实现点击其他位置使 EditText 失去焦点并且输入键盘消失的功能。这里我实现了一个 bgClick 函数，表示点击背景后输入框失去焦点并且输入键盘消失的功能。代码如下：

```
public void bgClick() {
    bg = (ConstraintLayout) findViewById(R.id.bg);
    bg.setOnTouchListener(new View.OnTouchListener() {
        @Override
        public boolean onTouch(View v, MotionEvent event) {
            bg.requestFocus();
            InputMethodManager imm = (InputMethodManager) getSystemService(Context.INPUT_METHOD_SERVICE);
            imm.hideSoftInputFromWindow(getWindow().getDecorView().getWindowToken(), 0);
            return false;
        }
    });
}
```

6. EditText 输入为空时，点击登录按钮会出现错误信息提示，如果不加处理，错误信息提示会一直在那里不会消失，所以每次捕获按钮点击事件时都要重新设置 TextInputLayout 的 setErrorEnable 为 false 使其不显示错误信息提示，然后在需要显示错误信息提示时再设置为 true 即可。

四、课后实验结果

五、实验思考及感想

上次实验主要是处理布局相关，即主要是处理 activity_main.xml 文件，而本次实验则主要与 UI 交互有关，即 UI 的事件处理，主要处理 activity 的 java 程序。

本次实验遇到的问题不少，但基本都在上面提到总结过了，所以这里就讲讲技巧性的东西。

1. 仔细观察实验文档会发现需要用到多次 Snackbar，而每次使用

Snackbar 时若都重新写一次 Snackbar 的创建过程那将是很费功夫的，同时也是很影响代码的整体结构与美观程度的。所以这里可以将 Snackbar 的创建过程写到一个函数里面，然后需要用时直接调用函数并写入恰当的参数就可以适应各个需要 Snackbar 的场景了。

2. 如果可以，尽量将事件监听器整合到 xml 文件中，然后将监听到事件后需要调用的函数实现写入到 java 程序中，从而实现项目文件之间的解耦。比如 Button 组件的 click 监听事件的实现可以在 xml 文件中的 Button 组件中添加属性 `android:OnClick="函数名"` 代替在 java 程序中定义设置 `OnClickListener` 监听器。
3. Java 中 String 的比较是否相等有两种，一种是使用等于号判断，另一种是使用 `equals` 函数比较。前者不仅比较字符串的串值，还比较字符串对象的地址，即只有两个 String 对象引用的是同一个地址的字符串时才会判断相等。后者则只比较字符串值，符合我们的一般比较逻辑，所以本次实验需要使用字符串比较的地方都要使用 String 类提供的 `equals` 函数来判断。