Android应用软件设计

E 2 Intent and Intent Filter

学号: **SA18225112**
姓名: 何红飞

报告撰写时间:**2018/10/05**

1. 主题概述

1. 使用Intent的显性和隐性用法

使用Intent从一个 activity 切换到另一个activity中。这个activity可以是同一app下的活动，或者是两个app下的活动。考查了Intent的显性和隐性的用法。

2. 定义 IntentFilter 和 Permission 属性

在AndroidManifest文件中定义Activity的Intent-Filter和Application的Permission属性。

3. 设计布局并且根据情况隐藏或者显示

在通过使用 Intent 转换活动的时候，可以使用puextra（）函数传递附加值，MainScreen可通过判断传递进来的值确定是否隐藏或者显示布局或组件。

4. 用正则表达式验证登录

考察正则匹配在实际应用中的作用。

5. ProgessBar显示进度，2秒后消失

本此应用中采用progressDialog和Hlandler来实现

6. 新建一个工程，测试是否可以跳转到MainScreen

通过Intent的action和category设置来跳转应用，了解了Intent更高级的用法，并且也了解了permission的定义和使用

2. 假设

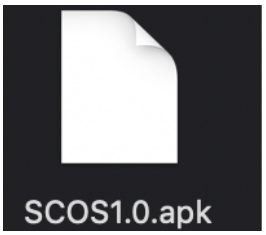
本应用是一款晚上订餐软件。这里假设用户要通过登录或者注册通过一定条件下方可正常进入点餐系统。因此系统要对页面的登录信息进行验证，并提醒用户输入是否有误。除此之外还要定义一些权限，使得其余软件跳转进来需要特定的声明。

3. 实现或证明

1. Github:

https://github.com/WokeFei/SCOS.git

2. APK：



3. 在MainScreen的屏幕上设计导航项并且在AndroidManifest.xml中对MainScreen定义其IntentFliter属性

首先，新建MainScreen活动，布局名称默认为layout_main_screen.xml，采用线性布局

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainScreen">
```

此布局调用用了了四个Button按钮控件， 并使他们横向排列列。

定义在AndroidManifest.xml 中注册好的活动MainScreen的IntentFilter属性:

```
<activity android:name=".MainScreen"
    android:permission="scos.permission.ACCESSSCOS"
    android:exported="true">
    <intent-filter>
        <action android:name="scos.intent.action.SCOSMAIN"></action>
        <category android:name="scos.intent.category.SCOSLAUNCHER"></category>
        <category android:name="android.intent.category.DEFAULT"></category>
    </intent-filter>
</activity>

<activity android:name=".LoginOrRegister">
```

4. 在 SCOSEntry 屏幕上水平向左滑动时，从 SCOSEntry 屏幕跳转到 MainScreen 屏 幕

为了了使得SCSOEntry活动能够监听活动上的触控事件、和手手势，SCOSEntry需要实现两个接口口:

```
public class MainScreen extends AppCompatActivity implements AdapterView.OnItemClickListener {
```

重写里里里面面的方方法onFling() 和 onTouch():如图所示:

```
    @Override
    public boolean onFling(MotionEvent e1, MotionEvent e2, float velocityX, float velocityY) {
        //设置最小的手势移动距离和手势移动速度
        final int FLING_MIN_DISTANCE=100;
        final int FLING_MIN_VELOCITY=200;

        //向左的手势
        if(e1.getX() - e2.getX() > FLING_MIN_DISTANCE && Math.abs(velocityX) > FLING_MIN_VELOCITY) {
            // 跳转到下一个Avtivity去，并且传递一个String值FromEntry
            Intent intent = new Intent( packageContext: SCOSEntry.this,MainScreen.class);
            intent.putExtra( name: "extra_data", value: "FromEntry");
            startActivity(intent);
        }
        return false;
    }

    @Override
    public boolean onTouch(View v, MotionEvent event) {
        return gd.onTouchEvent(event);
    }
```

其中为了了防止止误触，设置了了最小小的手手势移动距离和手手势移动速度，当手手势移动满足足条件时可，实现左划切换活动功能，与此同时，SCOSEntry向下个活动传递了“FromEntry”字符串串的值。

5. 隐藏导航项和展示导航项

在MainScreen中添加判断语句句，对上一一个活动传递过来的值就行行行验证，若值为“FromEntry”则展示导航项否则就隐藏导航项，实现过程如下，首首先取得上个活动传来 的值:

```
Intent intent = getIntent();
// 接收活动传递的值
String data = intent.getStringExtra( name: "extra_data");
// 判断是否和“FromEntry”相等;如果相等,则正常显示当前屏幕;如果不相等，则隐藏导航项
// 或者若登陆成功，也可正常显示主屏幕
```

再对值进行行行判定:

```
// 或者若登陆成功，也可正常显示主屏幕
if((data != null && data.equals("FromEntry")) || (data != null &&data.equals("LoginSuccess"))){
    setContentView(R.layout.activity_main_screen);
}
```

由于目目前MainScreen界面面只有导航栏，故可直接使整个布局设置是否可见见。

6. AndroidManifest.xml中定义Permission属性

在AndroidManifest.xml中添加如下字段:

```
        </activity>
        <activity android:name=".MainScreen"
            android:permission="scos.permission.ACCESSSCOS"
            android:exported="true">
            <intent-filter>
                <action android:name="scos.intent.action.SCOSMAIN"></action>
                <category android:name="scos.intent.category.SCOSLAUNCHER"></category>
                <category android:name="android.intent.category.DEFAULT"></category>
            </intent-filter>
        </activity>

        <activity android:name=".LoginOrRegister">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

    <permission android:name="scos.permission.ACCESSSCOS"
        android:protectionLevel="dangerous"></permission>
```

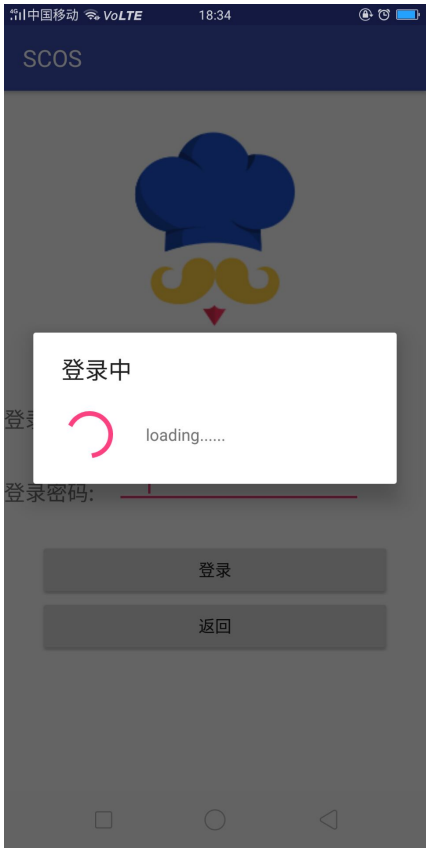
7. LoginOrResgister活动的登录、返回功能以及progressBar显示

需要对登录按钮进行如下实现：

当点击登录按钮时，使用 ProgressBar 显示登 录进度，2 秒钟后进度条消失;同时，使 用正则表达式验证“登录名”和“登 录密码”是否符合以下规则:不为空，只包含英文大小写 和数字。当输入内容 不符合规则时，则使用 setError 方法在当前输入框出提示错误信息 “输入内容 不符合规则”。

```
public void onClick(View v) {
    switch (v.getId()){
        case R.id.button_login:
            // 显示ProgressDialog
            progressDialog = new ProgressDialog( context: this);
            progressDialog.setTitle("登录中");
            progressDialog.setMessage("loading......");
            progressDialog.setCancelable(true);
            progressDialog.show();
            // 通过Handler发送延时消息
            Handler handler = new Handler();
            handler.postDelayed(new Runnable() {
                @Override
                public void run() {
                    // 正则表达式，只匹配英文大小写字母和数字
                    String regex = "^[a-z0-9A-Z]+$";
                    // 判断输入框中的内容是否符合要求
                    if(edit_user.getText().toString().matches(regex) && edit_password.getText().toString().matches(regex)){
                        // 屏幕跳转至 MainScreen，并向 MainScreen 类传 递一个数据 String 值为“LoginSuccess”
                        Intent intent = new Intent( packageContext: LoginOrRegister.this,MainScreen.class);
                        intent.putExtra( name: "extra_data", value: "LoginSuccess");
                        startActivity(intent);
                    }
                    if (!edit_user.getText().toString().matches(regex)){
                        edit_user.setError("输入内容不符合规则");
                    }
                    if(!edit_password.getText().toString().matches(regex)) {
                        edit_password.setError("输入内容不符合规则");
                    }
                    // ProgressDialog不会自动消失，记得在让其在2秒后消失
                    progressDialog.dismiss();
                }
            }, delayMillis: 2000);    //延迟两秒
            break;
```

结果如图所示：



8. 从另一个app中打开MainScreen活动

采用用隐形Intent的方方式启动MainScreen:

```
@Override
public void onClick(View v) {
    switch(v.getId()){
        case R.id.button:
            Intent intent = new Intent( action: "scos.intent.action.SCOSMAIN");
            intent.addCategory("scos.intent.category.SCOSLAUNCHER");
            startActivity(intent);
            break;
        default:
            break;
    }
}
```

同时在TestSCOS的AndroidManifest.xml 文文件中请求权限:

```
<uses-permission android:name="scos.permission.ACCESSSCOS"></uses-permission>
```

4. 结论

本次作业加深了我们对活动穿梭的理解，同时对InetntFilter属性和Permiss权限属性有 了一定的锻炼。也让我们了解了Intent的显性和隐性用法。同时本节还有手势、触控监听 的功能;对登录功能做了一个开端，使用正则匹配符合条件的输入 。

Question1: 请查阅资料总结 android:protectionlevel 的不同类型，并说明如何使用:
Android:protectionlevel 有如下类型:

(1) Normal 权限被声明为Normal级别，任何应用都可以申请，在安装应用时，不会直接提示给用户，点击全部才会展示。

(2) Dangerous 权限被声明为Dangerous级别，任何应用都可以申请，在安装应用时，会直接提示给用户。

(3) Signature 权限被声明为Signature级别，只有和该apk定义了了这个权限的apk用相同的私钥签名的应用才可以申请该权限。 frameworks/base/core/res/AndroidManifest.xml声明的权限为Signature级别，那么只有Android官方使用相同私钥签名的应用才可以申请该权限。

(4) SignatureOrSystem
权限被声明为SignatureOrSystem级别，有两种应用可以申请该权限。
1) 和该apk(定义了了这个权限的apk)用相同的私钥签名的应用
2) 在/system/app目录下的应用

Question2: IntentFilter 如何测试 Action、Category、Data，并说明 IntentFilter用法

对于Action:
Intent中的Action必须能够和Activity过滤规则中的Action匹配.(这里的匹配是完全等). 一个过滤规则中有多个action,那么只要Intent中的action能够和Activity 过滤规则中的任何一个action相同即可匹配成功。

对于**Category:**
如果Intent中的存在category那么所有的category都必须和Activity过 滤规则中的category相同。才能和这个Activity匹配。Intent中的category数量量可能少于 Activity中配置的category数量量，但是Intent中的这category必须和Activity中配置的 category相同才能匹配。

对于**Data**:
类似于action匹配，但data有更复杂的结构 IntentFilter用法:Intent Filter就是 用来注册 Activity 、Service 和 Broadcast Receiver 具有能在某种数据上执行一个动作的能力。
使用 Intent Filter ，应用程序组件告诉 Android ，它们能为其它程序的组件的动作请求提供服务，包括同一个程序的组件、本地的或第三方的应用程序。
为了注册一个应用程序组件为 Intent 处理理者，在组件的 manifest 节点添加一个 intent-filter 标签。

5. 参考文献

1. 左右滑动切换活动 https://blog.csdn.net/l_do_can/article/details/50411600
2. IntentFilter详解 https://blog.csdn.net/wuwenxiang91322/article/details/7671593
3. Android权限管理 https://blog.csdn.net/jltxgcy/article/details/48288467
4. Handler深入浅出 https://www.cnblogs.com/JohnTsai/p/5259869.html