

实验说明文档

软件名称: **LoginPageKSJ**

完成人: 姜淑晶

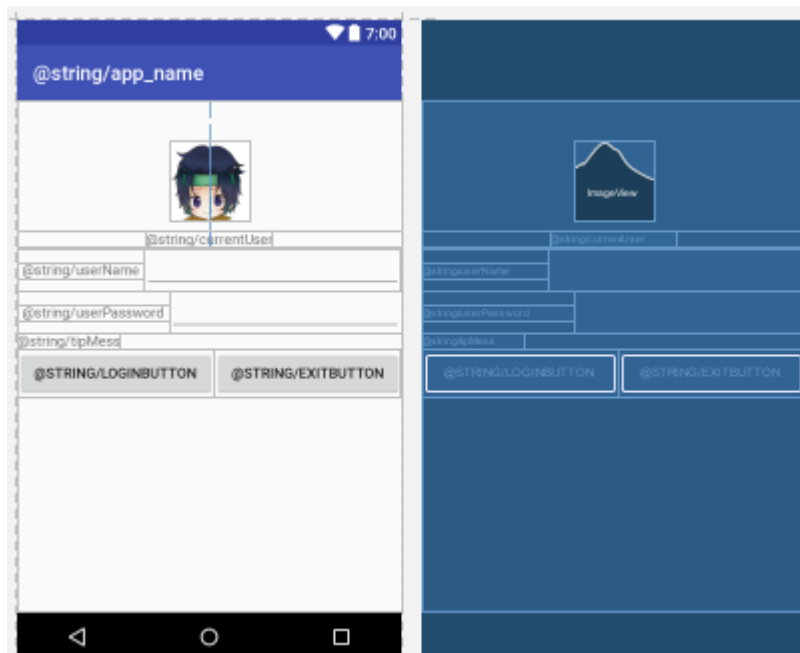
学号: 20152100061

完成时间: 2017-11-02

软件内容简介:

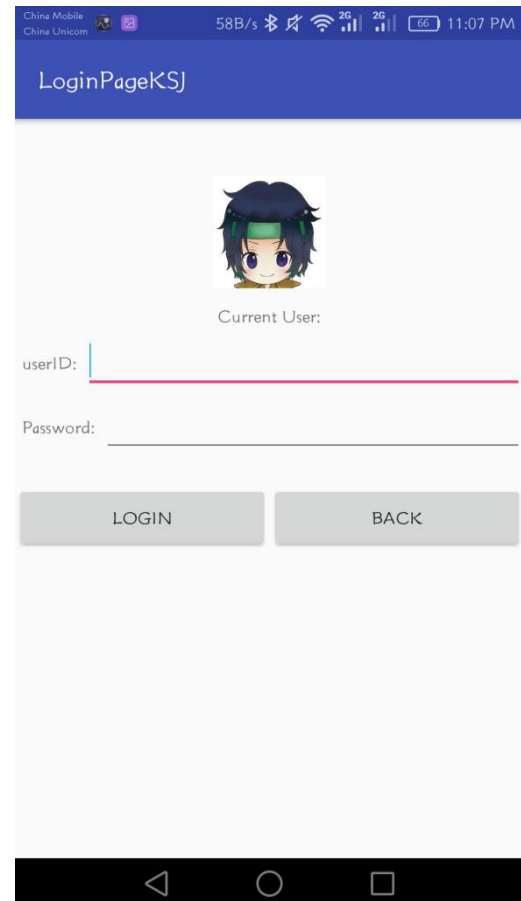
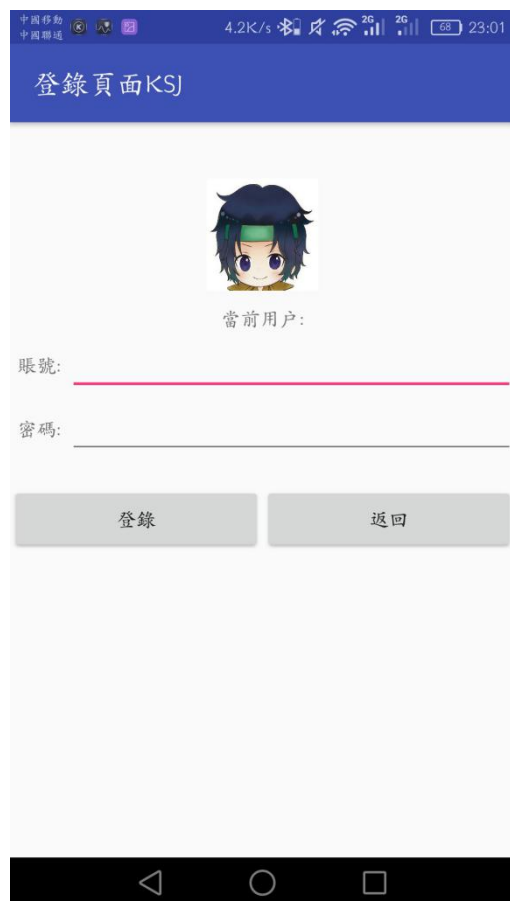
完成登录界面的开发, 要求实现一个支持双语(中文和英文)的登录界面, 并拥有管理安全问题和错误尝试功能, 实现自动化 UI 测试。

界面设计:

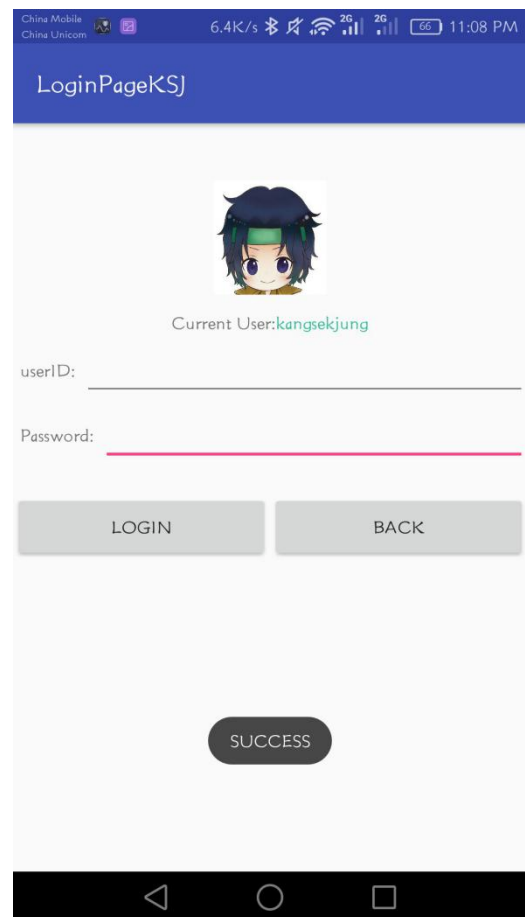
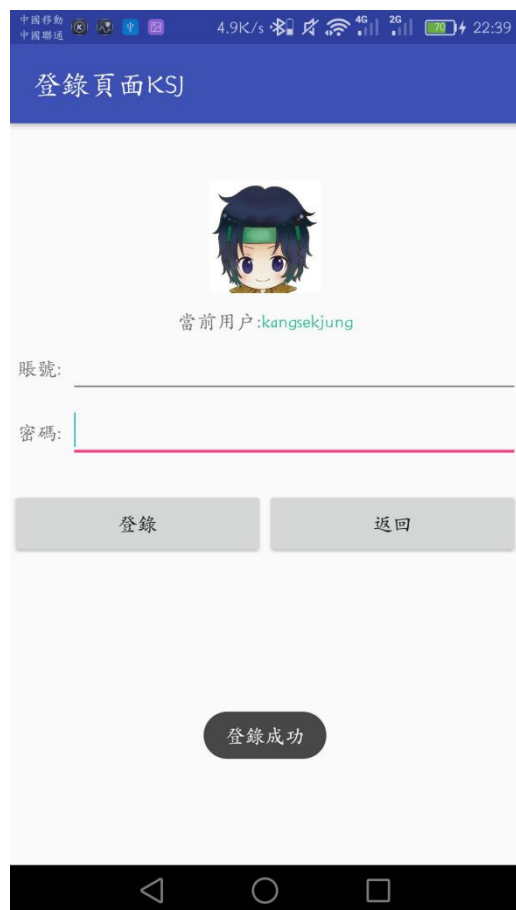


真机运行结果:

初始页面:

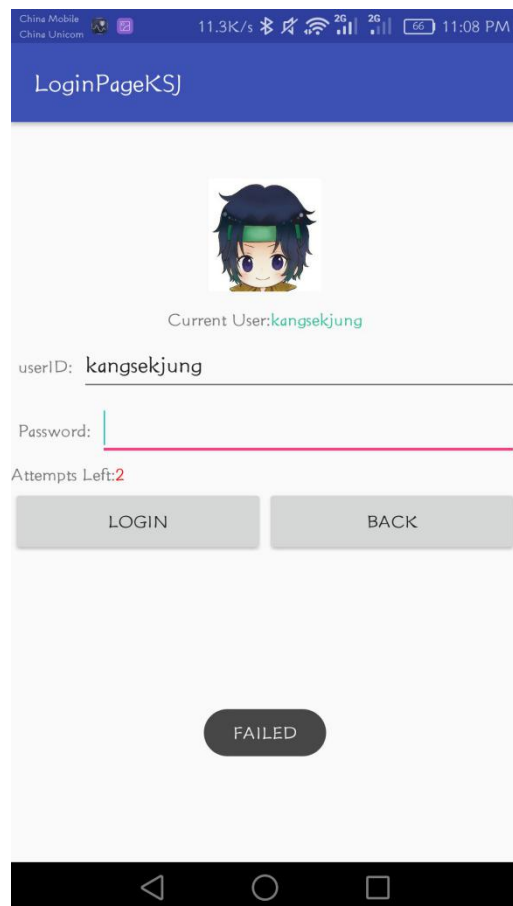
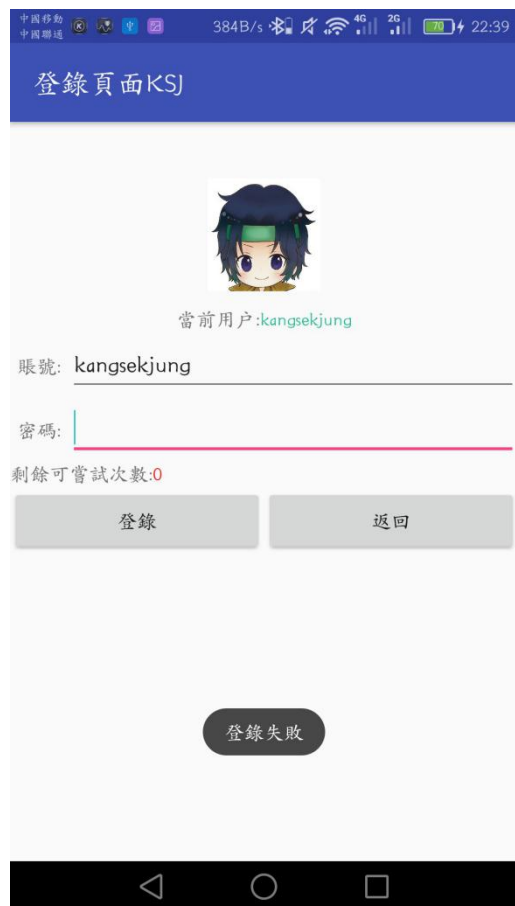


登陆成功界面:



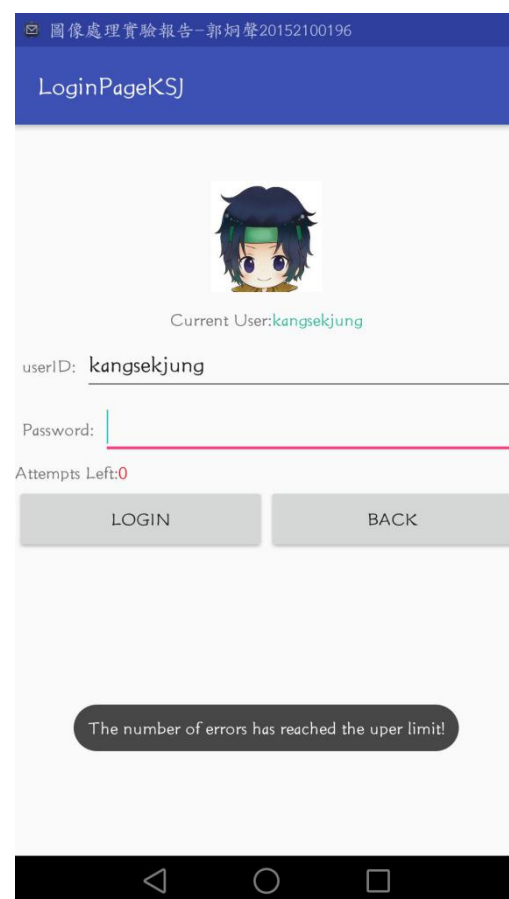
注：当用户成功登录后，会将用户名显示在“当前用户”栏上，直至下一个成功登录的用户出现才更新当前用户信息。

登录失败界面：



注：仅当用户第一次登录失败之后，才显示剩余可尝试次数的提示信息。

登录失败次数已达上限界面：



代码设计:

软件素材:



详细代码说明:

页面布局 xml 文件代码:

```
<!-- 采用线性布局做主布局，利用布局嵌套完成界面的设计 -->
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/loginPage"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    >
    <!-- 利用相对布局设置用户头像位于屏幕水平中线位置并定位垂直位置 -->
    <RelativeLayout
```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <ImageView
            android:id="@+id/logo"
            android:layout_width="80dp"
            android:layout_height="80dp"
            android:layout_centerHorizontal="true"
            android:layout_marginTop="40dp"
            android:layout_marginBottom="10dp"
            android:background="@drawable/logo1"
        />

```

```

    </RelativeLayout>

```

<!-- 利用相对布局设置“当前用户”显示窗口位于水平中间，嵌套线性布局完成“当前用户”显示窗口的横向布局 -->

```

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerHorizontal="true"
            android:orientation="horizontal">
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="@string/currentUser"/>
            <TextView
                android:id="@+id/curUser"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textColor="@color/ColorGreen"
                android:text="" />

        </LinearLayout>
    </RelativeLayout>

```

<!--利用线性布局完成账号密码的显示和输入框的位置布置，规定账号只能由大小写英文组成 -->

```

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
    >
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/userName"
        android:paddingLeft="5dp"
        android:paddingRight="5dp"
    />
    <EditText
        android:id="@+id/ID"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:digits="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
        android:maxLines="1"/>
    </LinearLayout>

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    >
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/userPassword"
        android:paddingLeft="5dp"
        android:paddingRight="5dp"
    />
    <EditText
        android:id="@+id/Password"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textPassword"
        android:maxLines="1"/>
</LinearLayout>

```

<!--利用线性布局实现“剩余次数”提示的布置-->

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:orientation="horizontal">

    <TextView
        android:id="@+id/attemptstip"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/tipMess"/>

    <TextView
        android:id="@+id/attempts"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="" />

</LinearLayout>

```

<!--利用线性布局完成登录与返回按钮的位置布置-->

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <Button
        android:id="@+id/login"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

        android:paddingLeft="15dp"
        android:paddingRight="15dp"
        android:layout_weight="1"
        android:text="@string/loginButton"/>

    <Button
        android:id="@+id/back"

```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:paddingLeft="15dp"
        android:paddingRight="15dp"
        android:text="@string/exitButton" />
    </LinearLayout>

</LinearLayout>

```

Strings.xml 文件代码:

中文版本:

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">登录页面</string>
    <string name="loginPageTitle">登录页面 KSJ</string>
    <string name="currentUser">当前用户: </string>
    <string name="userName">账号: </string>
    <string name="userPassword">密码: </string>
    <string name="tipMess">剩余可尝试次数: </string>
    <string name="loginButton">登录</string>
    <string name="exitButton">返回</string>
    <string name="errorMess">登录错误次数已达上限!</string>
    <string name="loginSuccess">登录成功</string>
    <string name="loginFailed">登录失败</string>
</resources>

```

英语版本:

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">LoginPage</string>
    <string name="loginPageTitle">LoginPageKSJ</string>
    <string name="currentUser">Current User: </string>
    <string name="userName">userID: </string>
    <string name="userPassword">Password: </string>
    <string name="tipMess">Attempts Left: </string>
    <string name="loginButton">LOGIN</string>
    <string name="exitButton">BACK</string>
    <string name="errorMess">The number of errors has reached the uper limit!</string>
    <string name="loginSuccess">SUCCESS</string>
    <string name="loginFailed">FAILED</string>
</resources>

```

软件功能实现 java 文件:

```

/**
 * Created by jiangjing on 2017/10/29.
 */

public class LoginPage extends AppCompatActivity {

    Button login;
    Button back;
    EditText id;

```



```

EditText pw;
TextView curUser;
TextView attempts;
TextView attemptstip;
int count;

/**
 * 获取控件 并做初始化
 */
private void Init() {
    login = (Button) findViewById(R.id. login);
    back = (Button) findViewById(R.id. back);
    id=(EditText)findViewById(R.id. ID);
    pw=(EditText)findViewById(R.id. Password);
    count=3;
    curUser=(TextView)findViewById(R.id. curUser);
    attempts=(TextView)findViewById(R.id. attempts);
    curUser.setText("");
    attempts.setText(""+count);
    attempts.setVisibility(View. INVISIBLE);
    attemptstip=(TextView)findViewById(R.id. attemptstip);
    attemptstip.setVisibility(View. INVISIBLE);
}

/**
 * 检测用户的账号密码是否正确对应
 * @param id 账号
 * @param pw 密码
 * @return
 */
public boolean isLoginValid(String id,String pw)
{
    if (id.equals(pw))
        return true;
    return false;
}

/**
 * 为需要的按钮控件设置监听事件
 */
private void setListener() {
    login.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (count==0) { //当剩余可尝试次数到达 0 时，显示上限提示，并清空密码输入框
                Toast.makeText(LoginPage. this,
                    getResources().getText(R.string. errorMess), Toast. LENGTH_SHORT).show();
                //id.setText("");
                pw.setText("");
            }
            else { //判断用户输入的账号与密码是否正确对应；正确对应则显示登录成功的信息、
                清空账号与密码输入框并更新“当前用户”的记录信息；
                if (isLoginValid(id.getText().toString(), pw.getText().toString()))
                {
                    Toast.makeText(LoginPage. this,
                        getResources().getText(R.string. loginSuccess), Toast. LENGTH_SHORT).show();
                    curUser.setText(id.getText().toString());
                    id.setText("");
                    pw.setText("");
                }
            }
        }
    });
}

```

```

    }
    //否则，显示登录失败的信息、清空密码输入框、减少用户可尝试的剩余次数并显示剩余次数提示框；
    else{
        Toast.makeText(LoginPage.this,
getResources().getText(R.string.loginFailed), Toast.LENGTH_SHORT).show();
        count--;
        attempts.setText(""+count);
        attempts.setTextColor(Color.RED); //错误次数标红
        //清空密码区
        pw.setText("");
        //设置提示可见
        attempts.setVisibility(View.VISIBLE);
        attemptstip.setVisibility(View.VISIBLE);
    }
}

});
back.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        finish();
    }
}); //点击返回按钮，则退出登录界面

}

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.loginpage);

    Init(); //初始化
    setListener(); //设置控件监听器
}
}

```

自动化测试 UI 界面代码：

```

/**
 * Created by jiangjing on 2017/11/1.
 */

public class LoginPageInstrumentationTest {

    private static final String userId = "kang"; //用户名
    private static final String errorPW = "ksj"; //错误密码

    private LoginPage loginPage = new LoginPage(); //创建登录页面实现类
    @Rule
    public ActivityTestRule<LoginPage> mActivityRule = new ActivityTestRule<>(
        LoginPage.class);

    @Test
    public void LoginPageEnter() throws InterruptedException {
        //检测登录成功
        onView(withId(R.id.ID)).perform(typeText(userId), closeSoftKeyboard()); //输入账号
        onView(withId(R.id.Password)).perform(typeText(userId), closeSoftKeyboard()); //输入
    }
}

```

正确密码

```
    ///点击登录按钮
    onView(withId(R.id. login)).perform(click()); //点击登陆按钮
    //通过检测“当前用户”提示框目前的值是否为已成功登录 d 额用户名做成功依据
    onView(withId(R.id. curUser)).check(matches(withText(userId)));
    //检测登录失败
    onView(withId(R.id. ID)).perform(typeText(userId), closeSoftKeyboard());
    onView(withId(R.id. Password)).perform(typeText(errorPW), closeSoftKeyboard()); //输入
错误密码
    //按钮点击
    onView(withId(R.id. login)).perform(click());
    onView(withId(R.id. attempts)).check(matches(withText("2"))); //第一次登录错误，通过检测
    “剩余可尝试次数”提示框是否出现当前剩余次数“2”而判断；
    Thread.sleep(1000); //***toast 并非当前布局中的对象，无法直接检测；如果不加延时，可能
    Toast 会检测不到（猜想延时是为了预留 toast 对象形成显示的时间）
    //检测是否有显示相应内容的 toast 窗口
    onView(withText(R.string. loginFailed))
        .inRoot(withDecorView(not(mActivityRule.getActivity().getWindow().getDecorVi
ew()))))
        .check(matches(isDisplayed()));
    //检测连续三次的密码输入错误后第四次尝试登录时的信息提示
    onView(withId(R.id. Password)).perform(typeText(errorPW), closeSoftKeyboard());
    //按钮点击
    onView(withId(R.id. login)).perform(click());
    onView(withId(R.id. Password)).perform(typeText(errorPW), closeSoftKeyboard());
    //按钮点击
    onView(withId(R.id. login)).perform(click());
    Thread.sleep(3000); //等待之前的“登录失败”信息显示结束，否则第四次登录时检测 toast 的
    值仍未改变，造成误会
    onView(withId(R.id. Password)).perform(typeText(errorPW), closeSoftKeyboard());
    //按钮点击
    onView(withId(R.id. login)).perform(click());
    Thread.sleep(1000);
    onView(withText(R.string. errorMess))
        .inRoot(withDecorView(not(mActivityRule.getActivity().getWindow().getDecorVi
ew()))))
        .check(matches(isDisplayed()));

    }
}
```

Gradle 文件的 espresso 环境配置:

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 25
    buildToolsVersion "25.0.3"
    //解决测试版本用的库和最终用的库版本号不一致问题
    configurations.all {
        resolutionStrategy.force 'com.android.support:support-annotations:23.1.1'
    }
    defaultConfig {
        applicationId "com.kangsekjung.myapplication"
        minSdkVersion 15
        targetSdkVersion 25
        versionCode 1
        versionName "1.0"
    }
}
```

```

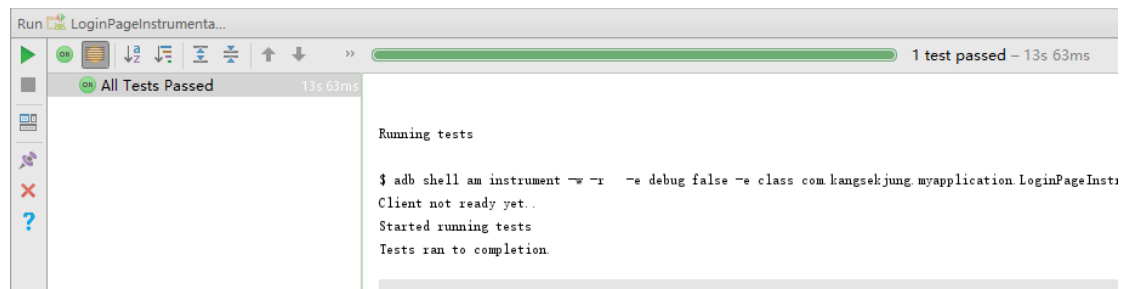
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"

    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'),
'proguard-rules.pro'
        }
    }
    //避免冲突
    packagingOptions {
        exclude 'LICENSE.txt'
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.android.support:appcompat-v7:25.3.1'
    compile 'com.android.support.constraint:constraint-layout:1.0.2'
    testCompile 'junit:junit:4.12'
    //espresso 的引用配置
    androidTestCompile 'com.android.support.test:runner:0.2'
    androidTestCompile 'com.android.support.test:rules:0.2'
    // ///androidTestCompile 'com.android.support.test.espresso:espresso-core:2.1'
}

```

UI 自动测试结果为：正确



软件操作流程：



难点（或遇到的问题）：

- 1.单元测试函数时报错，但真机运行正常；
- 2.进行自动化测试时，利用 `typeText(uesrID)` 函数于账号输入框与密码输入框输入测试用例（`userID='kangsekjung'`）时，真机实际对账号输入框输入的是“kang'se'k'jun'g”，而密码输入框则是“kangsekjung”，导致两者输入数据不一致，无法正确测试；
- 3.上一点问题解决后，出现另一个新问题：当测试用例 `userID="kangsekjung"` 时，真机账号输入框的实际的输入会发生乱序或自动添加某些字符，使得实际输入结果与代码设定的测试用例值不符；

4.进行 toast 内容是否出现的测试时，出现错误：Attempt to invoke virtual method 'android.view.View android.view.Window.getDecorView()' on a null object reference

解决方案：

1.尚未解决。

2.该问题产生的原因是手机输入法的默认输入语言为“中文拼音”，字符串“kangsekjung”在中文拼音输入的前提下，会自动在不符合拼音格式的字符间产生分隔符“'”；采用对账号输入框的输入格式进行限制的方式解决此问题，将账号输入框设置成只能输入大小写字母

（`android:digits="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"`）从而解决此测试问题。

3.目前仍未清楚真正的原因，但经过测试发现，减少测试用例的字符串个数（eg.改为“kang”）后，真机实际输入恢复正常，推测该问题是由于测试用例的字符串过长，产生某些细节问题。（注：第 2.3 点问题仅发生于账号输入框，密码输入框均正常输入）

4.仍为找出实际发生错误的原因，后来通过另一个参考网站找到一个能正确实现检测 Toast 是否出现的方法，即在原来的代码前添加 `Thread.sleep()` 函数。因此猜想出现该错误的原因是：由于 Toast 提示框是通过线程发生的，程序可能会在线程并未正式生成前已经进行检测，捕获对应的对象，因此使得捕获的对象为空，因此出现该空指针的错误。（参考网址：

<http://blog.csdn.net/yjp19871013/article/details/70570760>）

不足之处：

1.用户的账号密码的检测方式为简单的检测用户名与密码的值是否一致，而不是正常的账号密码映射检测；

2.错误管理中对错误登录的可尝试次数只针对软件的某一次使用，而不是针对每一个账号，也就是说在用户某一次使用该软件时，输入密码错误的次数总数为 3；

3.用户退出当前页面后再重新进入页面后，又能对同一个账号进行登录测试，并不能准确的实现安全管理功能；

今后的设想：

1.完成账号与密码的映射关系，实现更符合日常实际的账号登录功能；

2.实现每个账号都有三次登录错误的机会，每个账号的错误次数互不影响，独立记录；

3.实现对每个账号的登录错误次数的每日清零，使得每个账号每天都有三次错误机会；

参考资料：

利用 Espresso 进行 UI 自动测试的函数用法参考：<http://www.jianshu.com/p/22a09ae7b995>