

实验一说明文档

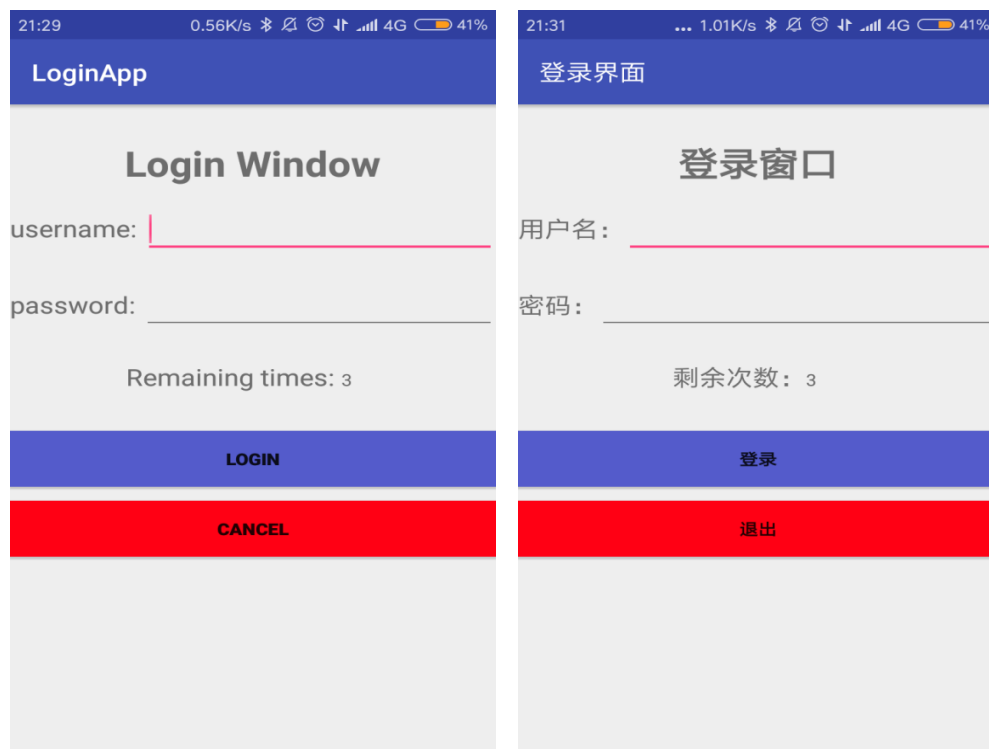
一、完成人信息

1. 姓名：钟勇辉
2. 学号：20152100083
3. 班级：计算机学院 15 级软件 5 班
4. 时间：2017 年 11 月 3 日

二、软件内容简介

实现一个支持双语（中文和英文）的登录界面，管理安全性和错误尝试。
同时实现 UI 测试。

三、界面设计



四、代码设计

1. activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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"
tools:context="com.example.rory_zhong.login.MainActivity">
```

```
<TextView
```

```
    android:id="@+id/titleName"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/title_name"
    android:layout_centerHorizontal="true"
    android:gravity="center_horizontal"
    android:layout_marginTop="30dp"
    android:textSize="30sp"
    android:textStyle="bold"/>
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="14dp"
    android:layout_below="@+id/titleName"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true">
```

```
<TextView
```

```
    android:id="@+id/userName"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/user_name"
    android:textSize="20sp"/>
```

```
<EditText
```

```
    android:id="@+id/myName"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="5dp"/>
```

```
</LinearLayout>
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="150dp">
```

```

<TextView
    android:id="@+id/userPassword"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/user_password"
    android:textSize="20sp"/>

<EditText
    android:id="@+id/myPassword"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="5dp"
    android:inputType="textPassword"/>
</LinearLayout>

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="220dp">

    <TextView
        android:id="@+id/messageName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/message_name"
        android:textSize="20sp"/>

    <TextView
        android:id="@+id/myTimes"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="3"
        android:layout_marginLeft="5dp"
        android:layout_marginRight="20dp"/>
</LinearLayout>

<Button
    android:id="@+id/loginButton"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="280dp"
    android:text="@string/login_button"
    android:textStyle="bold"

```

```

        android:background="#545bcb"
        android:onClick="onClick"/>

<Button
    android:id="@+id/cancelButton"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="340dp"
    android:text="@string/cancel_button"
    android:textStyle="bold"
    android:background="#ff0014"
    android:onClick="onClick"/>

</RelativeLayout>

```

2. MainActivity.java

```

package com.example.rory_zhong.login;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    int counter=3;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    public void onClick(View view)
    {
        EditText username = (EditText)findViewById(R.id.myName);
        EditText password = (EditText)findViewById(R.id.myPassword);
        Button login=(Button)findViewById(R.id.loginButton);
        TextView times = (TextView)findViewById(R.id.myTimes);
        switch (view.getId())
        {
            case R.id.loginButton: {
                if (username.getText().toString().equals("admin") &&

```


4. strings.xml (zh)

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">登录界面</string>
    <string name="title_name">登录窗口</string>
    <string name="user_name">用户名: </string>
    <string name="user_password">密码: </string>
    <string name="message_name">剩余次数: </string>
    <string name="login_button">登录</string>
    <string name="cancel_button">退出</string>
    <string name="message_true">正在登录&#8230;</string>
    <string name="message_wrong">密码错误! </string>
</resources>
```

5. mainTest.java

```
package com.example.rory_zhong.login;

import android.support.test.filters.LargeTest;
import android.support.test.rule.ActivityTestRule;
import android.support.test.runner.AndroidJUnit4;

import org.junit.Rule;
import org.junit.Test;
import org.junit.runner.RunWith;

import static android.support.test.espresso.Espresso.onView;
import static android.support.test.espresso.action.ViewActions.clearText;
import static android.support.test.espresso.action.ViewActions.click;
import static android.support.test.espresso.action.ViewActions.closeSoftKeyboard;
import static android.support.test.espresso.action.ViewActions.typeText;
import static android.support.test.espresso.assertion.ViewAssertions.matches;
import static android.support.test.espresso.matcher.RootMatchers.withDecorView;
import static android.support.test.espresso.matcher.ViewMatchers.isDisplayed;
import static android.support.test.espresso.matcher.ViewMatchers.withId;
import static android.support.test.espresso.matcher.ViewMatchers.withText;
import static org.hamcrest.core.IsNot.not;

/**
 * Created by Rory_Zhong on 2017/11/1.
 */
@RunWith(AndroidJUnit4.class)
```

@LargeTest

```
public class mainTest {
```

```
    private static final String myNameTest="admin";
    private static final String myPasswordRight="admin";
    private static final String myPasswordWrong="admin123";
```

@Rule

```
public ActivityTestRule<MainActivity> mActivityRule = new ActivityTestRule<>(
    MainActivity.class);
```

@Test

```
public void right() {
```

```
    onView(withId(R.id.myName)).perform(typeText(myNameTest));
    onView(withId(R.id.myPassword)).perform(typeText(myPasswordRight));    //输入
```

正确密码时，测试是否正常

```
    onView(withId(R.id.myPassword)).perform(closeSoftKeyboard());
    onView(withId(R.id.loginButton)).perform(click());
    String expectedResult0="3";
    onView(withId(R.id.myTimes)).check(matches(withText(expectedResult0)));
    onView(withText(R.string.message_true))    //
```

检测 toast 是否显示

```
        .inRoot(withDecorView(not(mActivityRule.getActivity().getWindow().getDecorView()))
        .check(matches(isDisplayed()));
```

```
}
```

@Test

```
public void wrong() {
```

```
    onView(withId(R.id.myName)).perform(clearText());
    onView(withId(R.id.myPassword)).perform(clearText());
    onView(withId(R.id.myName)).perform(typeText(myNameTest));
    onView(withId(R.id.myPassword)).perform(typeText(myPasswordWrong));    //输入错
```

误密码时，测试是否正常

```
    onView(withId(R.id.myPassword)).perform(closeSoftKeyboard());
    onView(withId(R.id.loginButton)).perform(click());
    String expectedResult1="2";
    onView(withId(R.id.myTimes)).check(matches(withText(expectedResult1)));
    onView(withId(R.id.myPassword)).check(matches(withText("")));
    onView(withText(R.string.message_wrong))    //
```

检测 toast 是否显示

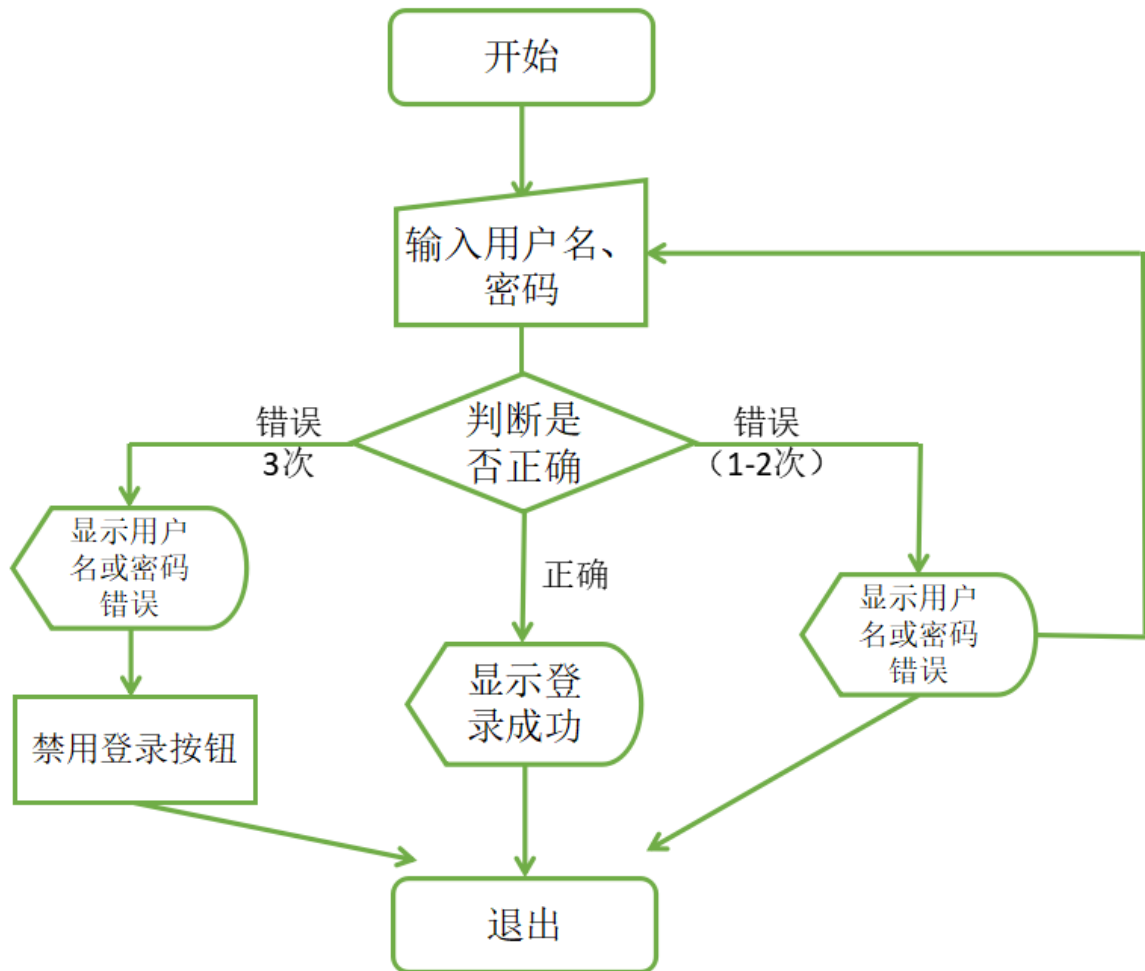
```
        .inRoot(withDecorView(not(mActivityRule.getActivity().getWindow().getDecorView()))
        .check(matches(isDisplayed()));
```

```

        .check(matches(isDisplayed()));
    }
}

```

五、软件操作流程



六、难点及解决方法

1. 支持双语的界面？

可以建两个 string.xml 文件，分别对应中文、英文，在使用时@string/name 就行。

2. 如何进行 UI 自动化测试？

可以采用 Android 自动化测试框架 Espresso, Espresso 是 Google 提供的 Android 自动化测试框架, 属于 Android 测试支持库, 主要用来进行 Android 界面(UI)相关的自动化测试。

(1) 在 EditText 中输入

```
onView(withId(R.id. myName)).perform(typeText(myNameTest));
```

(2) 点击 button

```
onView(withId(R.id. loginButton)).perform(click());
```

(3) 验证是否正确

```
onView(withId(R.id. myTimes)).check(matches(withText(expectResult0)));
```

(4) 验证 toast 显示

```
onView(withText(R.string. message_true))  
    .inRoot(withDecorView(not(mActivityRule.getActivity().getWindow().getDecorView())))  
    .check(matches(isDisplayed()));
```

七、不足之处及今后设想

界面不够完善, 还可以加上恰当的动画。