

## 一、软件名称

登录界面

## 二、完成人

陈玉淋

## 三、学号

20152100165

## 四、完成时间

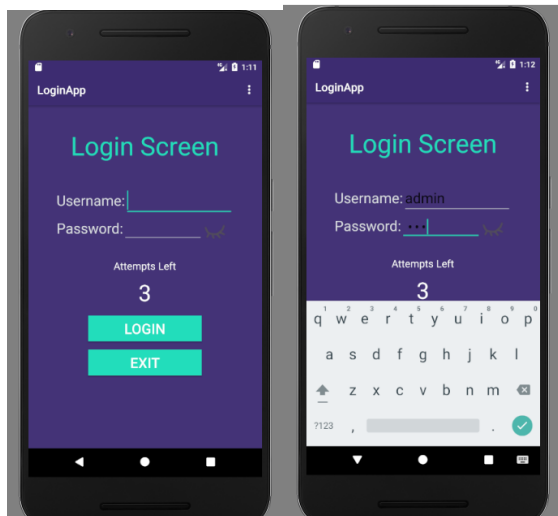
2017. 11. 05

## 五、软件内容简介

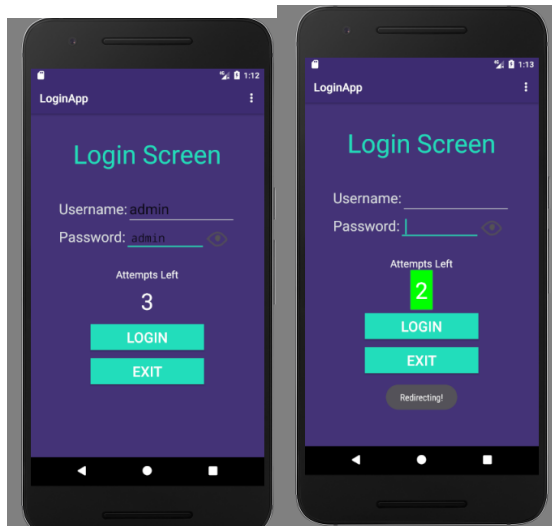
实现一个支持双语（中文和英文）的登录界面，管理安全性和错误尝试。可以对该 APP 进行扩展：设计更加美观的用户界面、实现更丰富的功能（选做）。

## 六、界面设计

（一）开始页面，点击眯眼图标图标变成睁眼可显示密码。

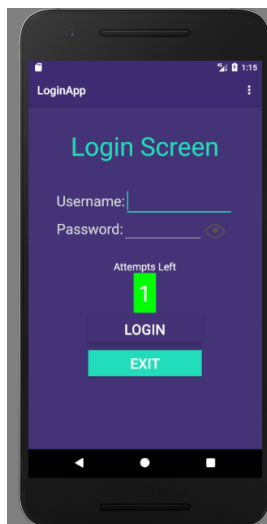


（二）输入密码，错误，显示剩余次数，并且对话框显示“密码错误”/成功显示“成功登陆”。3 次后登录按钮 不可点击。



(三) 按按钮时按钮颜色改变

(四) 可按右侧菜单栏退出



## 七、代码设计

(一) 详细设计过程及大概代码见登录页面设计教程

(二) Ui 测试代码（测试输入后，剩余次数变化）

```
package com.example.loginsurface;

import android.support.test.rule.ActivityTestRule;

import org.junit.Rule;
import org.junit.Test;

import static android.support.test.espresso.Espresso.onView;
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.ViewMatchers.withId;
import static
android.support.test.espresso.matcher.ViewMatchers.withText;

/**
 * Created by ChenYulin on 2017/11/5.
 */

public class MainActivityInstrumentationTest {
    private static final String STRING_TO_BE_TYPED = "admin";
    @Rule
    public ActivityTestRule<MainActivity> mActivityRule = new
ActivityTestRule<>(
        MainActivity.class);

    @Test
    public void sayHello() {

onView(withId(R.id.E_username)).perform(typeText(STRING_TO_BE_TYPED),
closeSoftKeyboard());

onView(withId(R.id.E_password)).perform(typeText(STRING_TO_BE_TYPED),
closeSoftKeyboard());
        onView(withId(R.id.Login)).perform(click());

        String expectedText = "3";

onView(withId(R.id.T_attempts)).check(matches(withText(expectedText))
);

onView(withId(R.id.E_username)).perform(typeText(STRING_TO_BE_TYPED),
closeSoftKeyboard());

onView(withId(R.id.E_password)).perform(typeText(STRING_TO_BE_TYPED),
closeSoftKeyboard());
        onView(withId(R.id.Login)).perform(click());

        String xpectedText = "2";

```

```

onView(withId(R.id. T_attempts)).check(matches(withText(xpectedText)))
;

onView(withId(R.id. E_username)).perform(typeText(STRING_TO_BE_TYPED),
closeSoftKeyboard());

onView(withId(R.id. E_password)).perform(typeText(STRING_TO_BE_TYPED),
closeSoftKeyboard());
        onView(withId(R.id. Login)).perform(click());

        String pectedText = "1";

onView(withId(R.id. T_attempts)).check(matches(withText(pectedText)));
    }
}

```

## 八、软件操作流程

参考六. 界面设计部分

## 九、难点（或遇到的问题）和解决方案

### （一）R 文件缺失

问题:R. id 中 R 为红色。

解决：只需要使用 Build -> Clean Project 和 Build -> Rebuild Project 重构下工程。

参考网站：<http://www.jb51.net/article/108275.htm>

后来重构也没用了，要在后面再写 R. java。

### （二）参见显示密码难点

### （三）参见使用菜单退出难点

## 十、不足之处和今后的设想

继续完成前面的闪屏界面。

附：

## 登录页面设计教程

一、完成一个支持双语（中文和英文）的登录界面，管理安全问题和错误尝试。

(一)修改 `res / values / string.xml` 添加必要的字符串，实现双语支持。由于作业中界面已经给出，所有需要字符串的控件以知且较少，所以我们先完成 app 的国际化。

1. 在 “`res / values/`” 目录下创建两个用于国际化的文件 `strings.xml`。

(1) 如图 1.1 所示，右击 “`res / values / string.xml`” 中的 `string.xml`，选择 **【new】** -> **【Values resource file】**

(2) 出现图 1.2 所示对话框，输入文件名 `strings`，选择蓝色光标所在位置，点击右侧 **》》** 按钮，出现图 1.3 选择所需语言（就是找国旗，但是注意选择语言的准确性，比如中文可以有西藏语，别选错了），按 `ok` 即可。

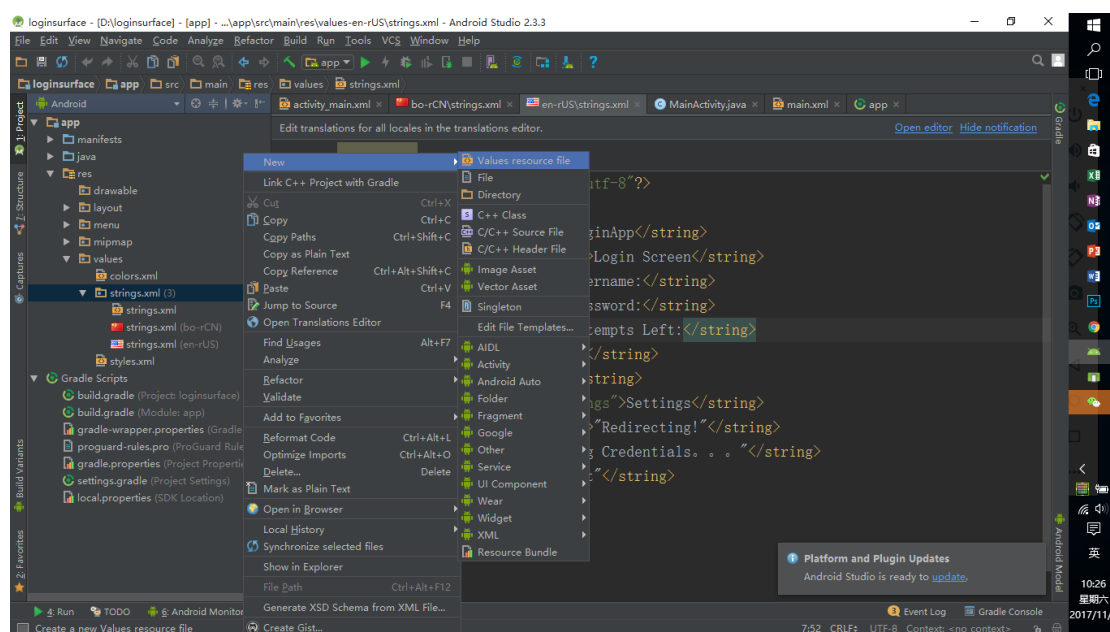


图 1.1

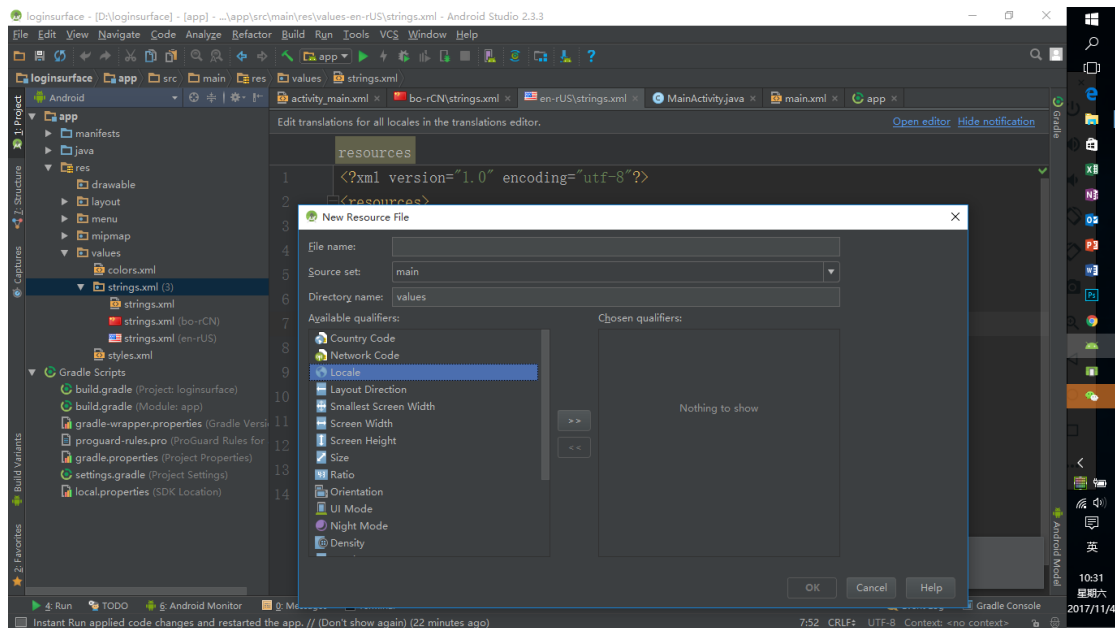


图 1.2

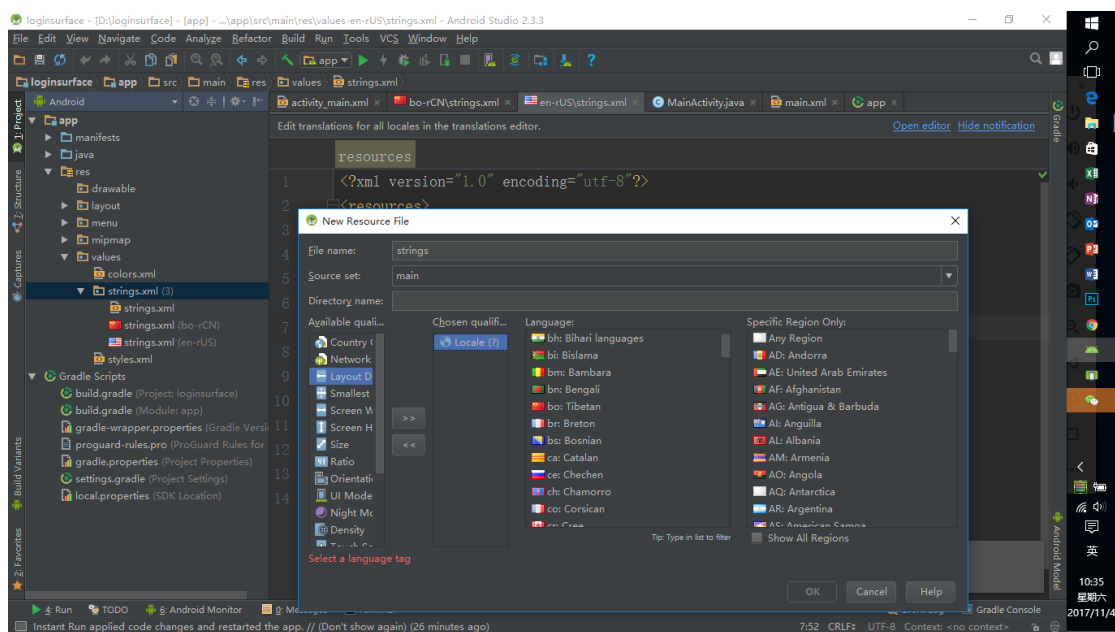


图 1.3(创建好的 strings 文件如图所示, 画着国旗的就是)

## 2. 接下来在创建好的 strings.xml 编写内容

(1) 点开中文的 strings.xml 文件，一开始出现图 2.1，比照图 2.2 输入内容。

```
<string name="username">用户名:</string>
```

其中

```
<string name="                ">    </string>
```

## 是固定格式

"username" 用户名:

可替换。

(2) Username 指在 res / layout/ activity\_main.xml 中定义的

```
android:text=
```

以用户名为例，代码如下其中，`android:text="@string/username"`

```
<TextView
    android:id="@+id/username"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:padding="10dp"
    android:text="@string/username" />
```

(3) 用户名即该 text 文本内容你将替换成什么。看图 2.3 找不同吧就能了解了。

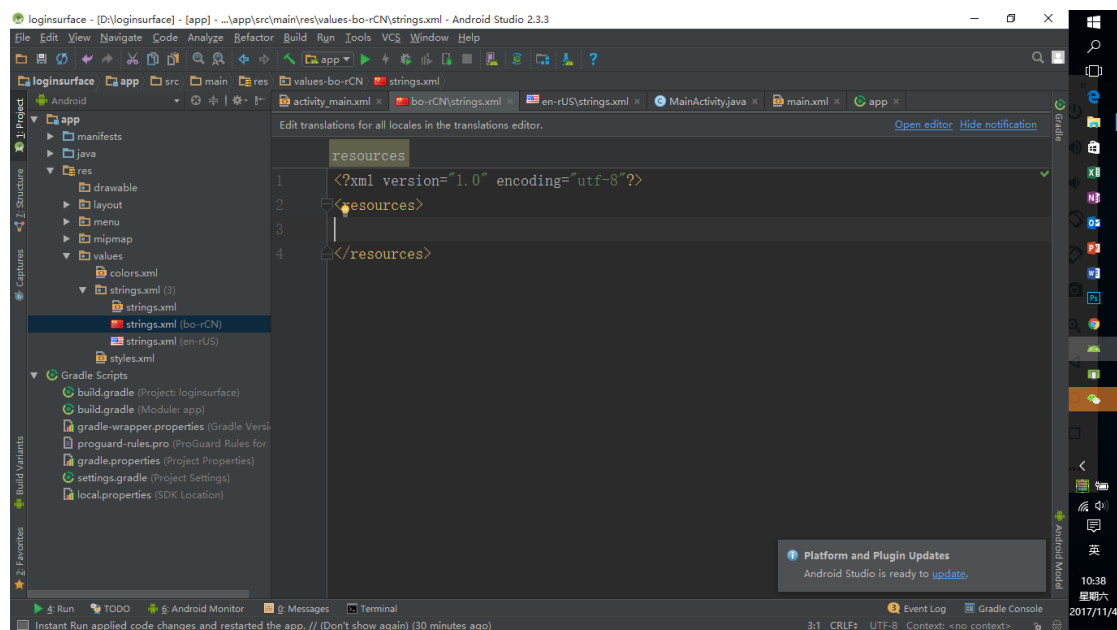


图 2.1

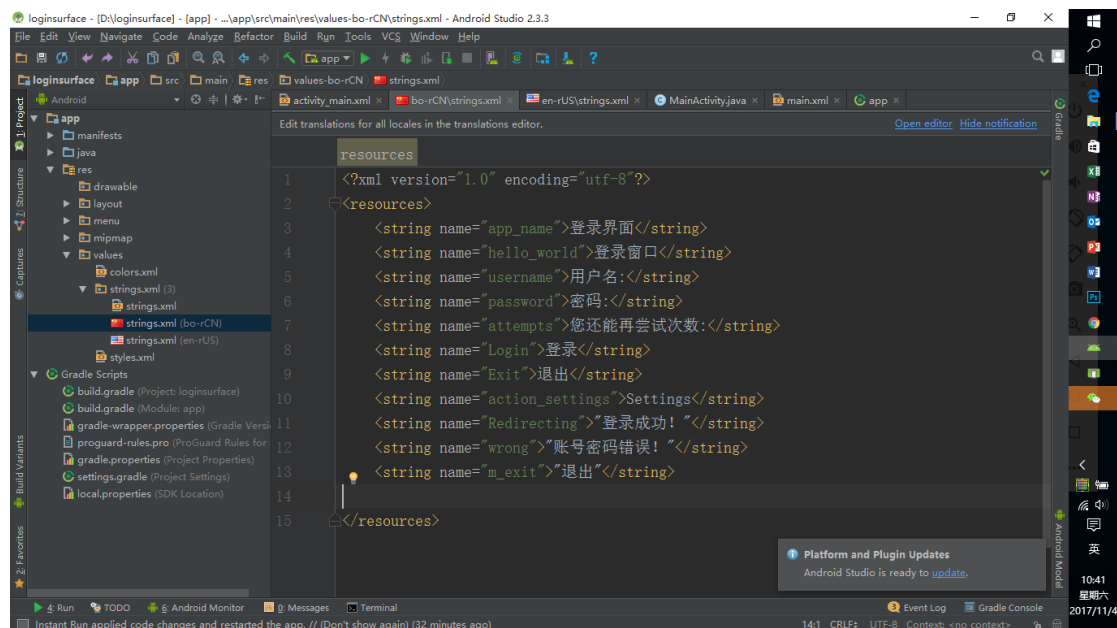


图 2.2(中文的 strings.xml)

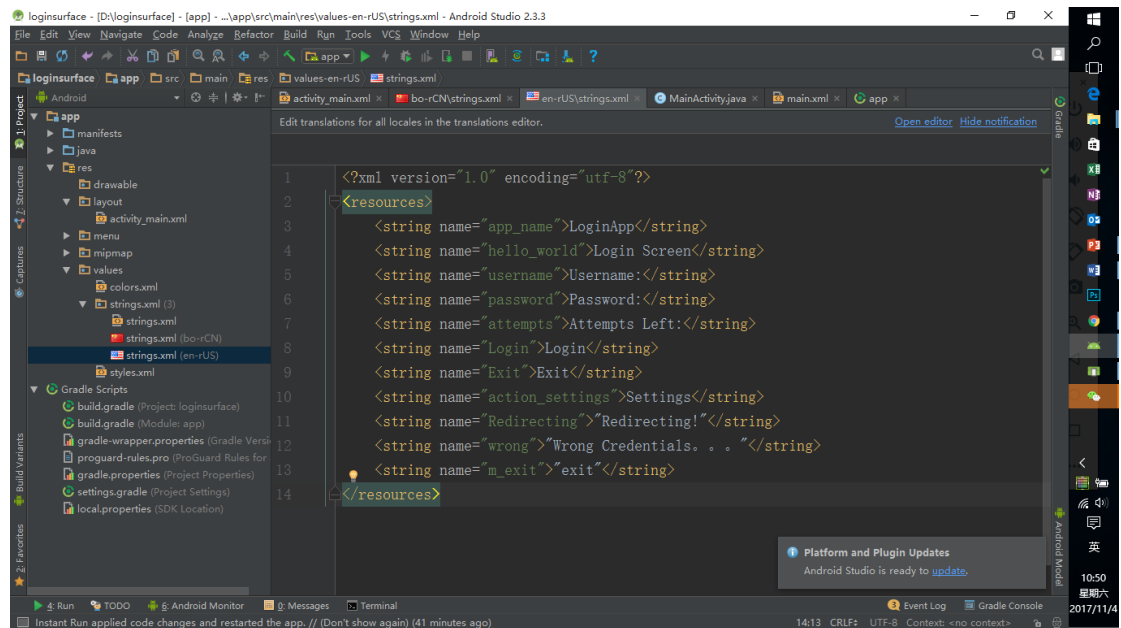


图 2.3 (英文的 strings.xml)

(二)界面 UI，修改 RES /布局/ activity\_main 添加相应的 XML 组件。

1. 定义两个 TextView 询问用户的用户名和密码。

密码 TextView 必须 inputType 设置为 textPassword。

定义登录和退出的按钮

(1) 密码

```
<EditText android:inputType="textPassword" />
```

(2) 登录按钮（响应用 onclick 属性方法）

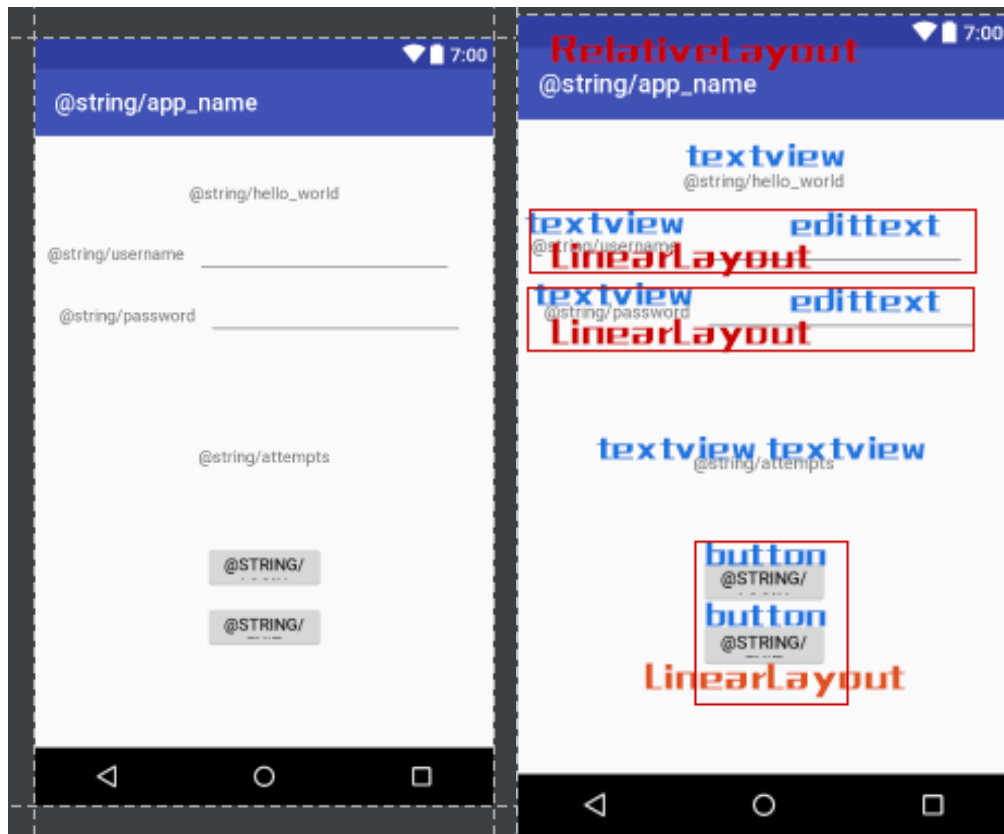
```
<Button android:id="@+id/Login"
        android:onClick="login"
```

(3) 退出按钮（响应用匿名内部类方法）

```
<Button android:id="@+id/Exit"
```

2. 基本界面布局





### 3. res/layout/activity\_main.xml 完整代码

```
<?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"
    android:orientation="vertical"
    tools:context="com.example.loginsurface.MainActivity">
    <TextView
        android:id="@+id/hello_world"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="40dp"
        android:text="@string/hello_world" />
    <LinearLayout
        android:id="@+id/L_username"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/hello_world"
        android:layout_centerHorizontal="true">
```

```
        android:layout_marginTop="20dp"
        android:orientation="horizontal">
        <TextView
            android:id="@+id/username"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:padding="10dp"
            android:text="@string/username" />
        <EditText
            android:id="@+id/E_username"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:padding="10dp"
            android:ems="10" />
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/L_username"
        android:layout_centerHorizontal="true"
        android:padding="10dp"
        android:orientation="horizontal">
        <TextView
            android:id="@+id/password"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:padding="10dp"
            android:text="@string/password" />
        <EditText
            android:id="@+id/E_password"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:padding="10dp"
            android:ems="10"
            android:inputType="textPassword" />
    </LinearLayout>
    <TextView
        android:id="@+id/attempts"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="260dp"
        android:text="@string/attempts" />
    <TextView
```

```

        android:id="@+id/T_attempts"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="260dp"
        android:layout_toRightOf="@id/attempt" />

```

<Button

```

        android:id="@+id/Login"
        android:layout_width="100dp"
        android:layout_height="40dp"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="340dp"
        android:onClick="login"
        android:text="@string/Login" />

```

<Button

```

        android:id="@+id/Exit"
        android:layout_width="100dp"
        android:layout_height="40dp"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="390dp"
        android:text="@string/Exit" />

```

</RelativeLayout>

(三) 修改 src / MainActivity.java 文件中添加必要的代码。

(四) 初稿完成测试

键入用户名和密码字段，然后按登录按钮。把 admin 作为用户名和 1234 为作为密码登陆。尝试失败（此次应用对话框，截图失败）。如图 4.1 所示，剩余次数 2:

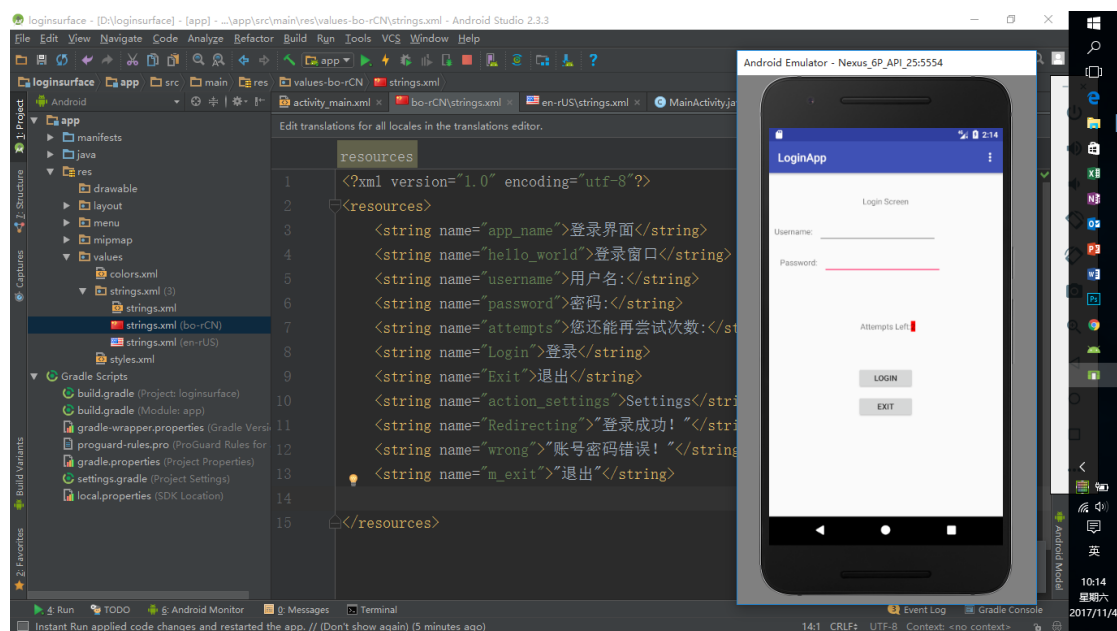


图 4.1

(2) 错误 3 次，会看到剩余次数 0，弹出对话框（截图成功），并且登录按钮被禁用。

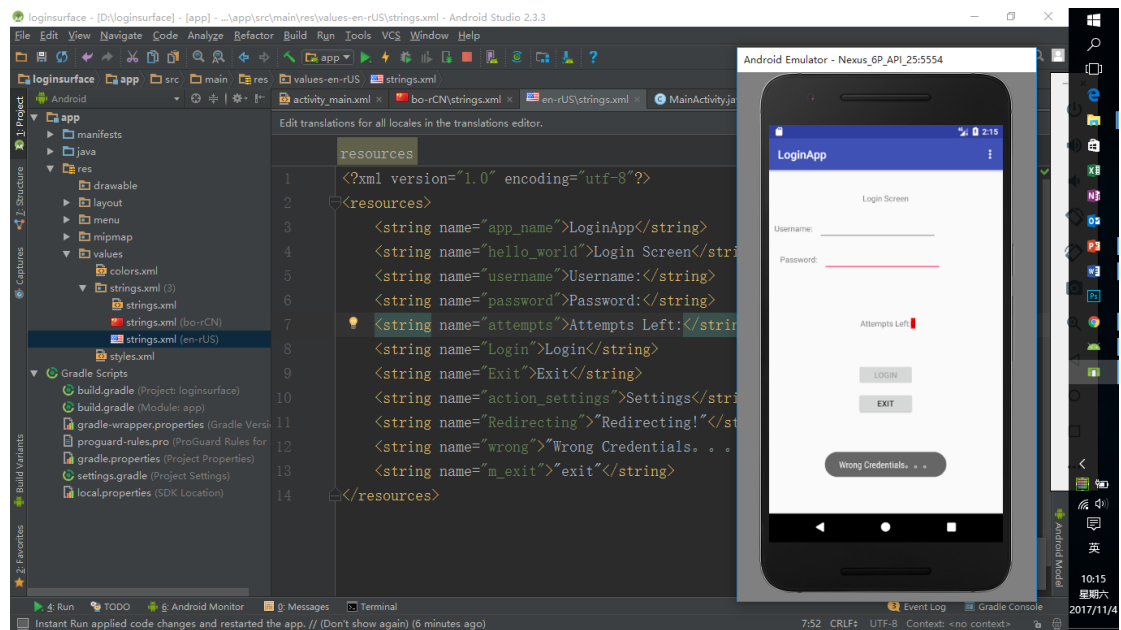


图 4.2 密码错误 3 次/弹出对话框/login 按钮失效

(3) 现在再次打开应用程序，这时候输入正确的用户名是 admin，密码为 admin 和点击登录。将成功登录。

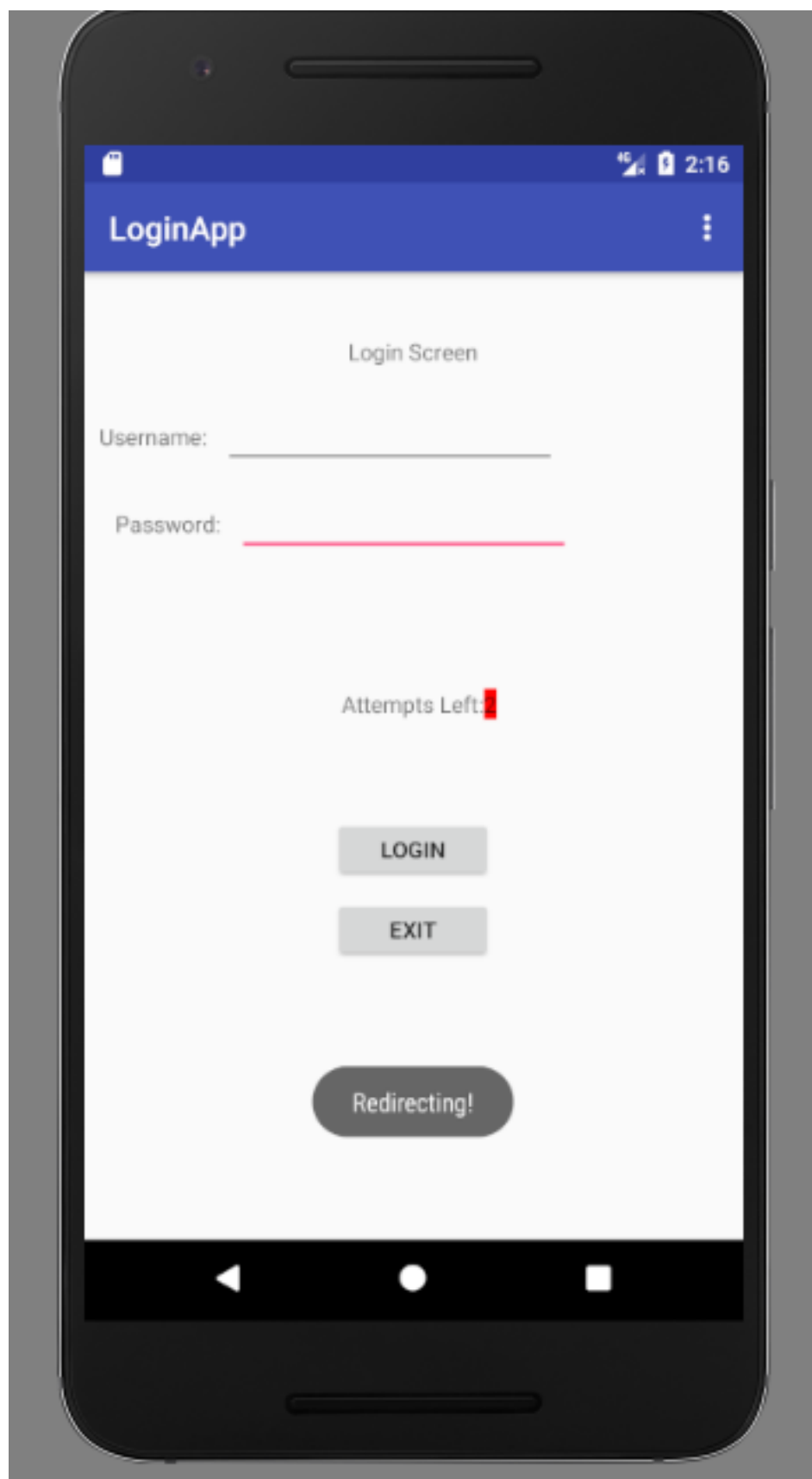


图 4.2 登陆成功

## 二、增加功能

## （一）显示密码可见性

### 1. 效果显示图

如图 1.1 开始密码不可见，点击眯眼图标改变密码可见性。



图 1.1

如图 1.2 点击眯眼变成睁眼，密码显示为可见。



图 1.2

### 2. 代码

#### （1）布局

在 res/layout/activity\_main.xml password 的线性布局中添加 bt\_pwd\_eye 按钮。设置眯眼图片。

```
<Button
    android:id="@+id/bt_pwd_eye"
    android:layout_width="40dp"
    android:layout_height="40dp"
    android:background="@drawable/password_close"/>
```

#### （2）mainactivity 代码

在 java/mainactivity 文件中添加以下代码

定义变量：

```
private Button bt_pwd_eye;
private boolean isOpen = false;
```

按钮匿名内部类

```
bt_pwd_eye=(Button)findViewById(R.id.bt_pwd_eye);
    bt_pwd_eye.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v) {
            if (isOpen) {
                isOpen = false;
            } else {
                isOpen = true;
            }
        }
    });
// 默认 isOpen 是 false, 密码不可见
    changePwdOpenOrClose(isOpen);

    exit=(Button)findViewById(R.id.Exit);
    exit.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v) {
```

```

        System.exit(0);
    }
});
}

```

changePwdOpenOrClose 函数

```

private void changePwdOpenOrClose(boolean flag) {
// 第一次过来是 false, 密码不可见
    if (flag) {
// 密码可见

bt_pwd_eye.setBackgroundResource(R.drawable.password_open);
// 设置 EditText 的密码可见

password.setTransformationMethod(HideReturnsTransformationMethod
        .getInstance());
    } else {
// 密码不接见

bt_pwd_eye.setBackgroundResource(R.drawable.password_close);
// 设置 EditText 的密码隐藏

password.setTransformationMethod(PasswordTransformationMethod
        .getInstance());
    }
}

```

### 3. 难点

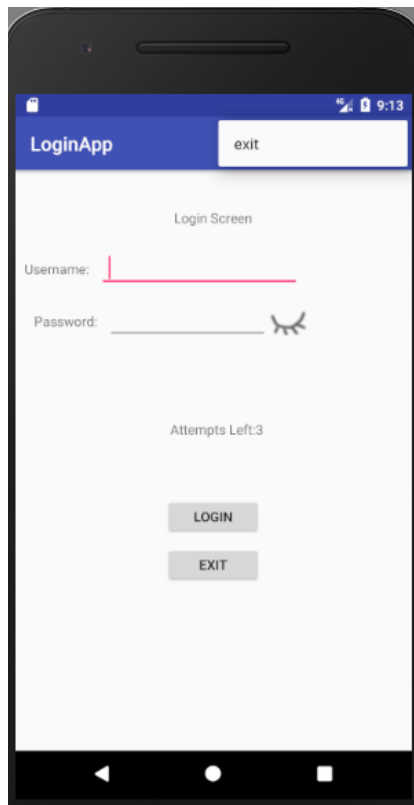
问题：按钮图标为自定义图标（自己画的很丑很可爱），使用以下代码  
 android:background="@drawable/password\_close”，此代码需要将图片无法导入 drawable 文件。

解决：直接复制粘贴到 drawable 文件。

（二）使用菜单退出

1. 效果展示

右上角菜单栏退出



## 2. 难点及代码

原本准备调用下面函数完成菜单退出，结果出错。

```
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is
    present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}
```

解决过程

A. 需要在 res 目录下创建 menu 文件夹【Res】右击【new】->【android resource directory】，并在 menu 文件夹中建立 xml 文件【new】->【menu resource files】并输入代码

```
<item
    android:id="@+id/m_exit"
    android:title="@string/m_exit"/>
```

B. 该函数仍无法使用，修改函数如下：

```
public boolean onCreateOptionsMenu(Menu menu)
{
    MenuInflater mf = new MenuInflater(this);
    mf.inflate(R.menu.main, menu); //这里的 R.menu.main 为 XML 文件
    的 ID
    return super.onCreateOptionsMenu(menu);
}

@Override
```



```

public boolean onOptionsItemSelected(MenuItem item)
{
    if(item.getItemId() == R.id.m_exit) finish();
    return super.onOptionsItemSelected(item);
}

```

### 3. 参考网站

[https://zhidao.baidu.com/question/1639769489338481020.html?qbl=relate\\_question\\_0&word=android%20%B2%CB%B5%A5%CD%CB%B3%F6](https://zhidao.baidu.com/question/1639769489338481020.html?qbl=relate_question_0&word=android%20%B2%CB%B5%A5%CD%CB%B3%F6)

## 三、仿 qq 闪屏（未完成）

### （一）动作代码

```

package com.example.loginsurface;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;

/**
 * Created by ChenYulin on 2017/11/5.
 */
public class SplashActivity extends Activity {
    private Intent intent;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.splash);
        startMainAvtivity();
    }

    private void startMainAvtivity() {
        new Handler().postDelayed(new Runnable() {
            public void run() {
                intent=new
Intent(SplashActivity.this,MainActivity.class);
                startActivity(intent);
                SplashActivity.this.finish();//结束本 Activity
            }
        }, 3000);//设置执行时间
    }
}

```

### （二）布局代码

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout

```

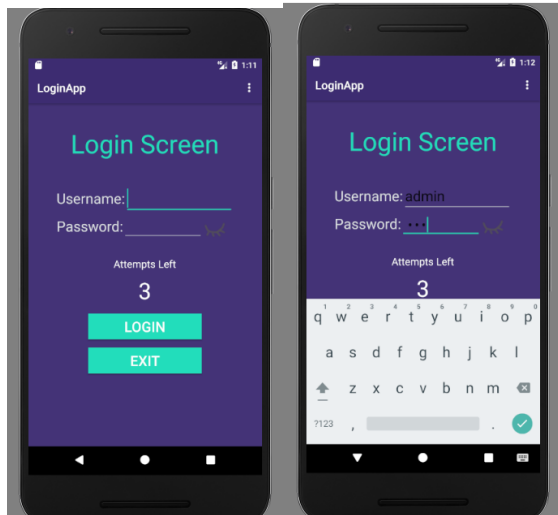
```
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:scaleType="matrix"
        android:src="@drawable/splash" />

</LinearLayout>
```

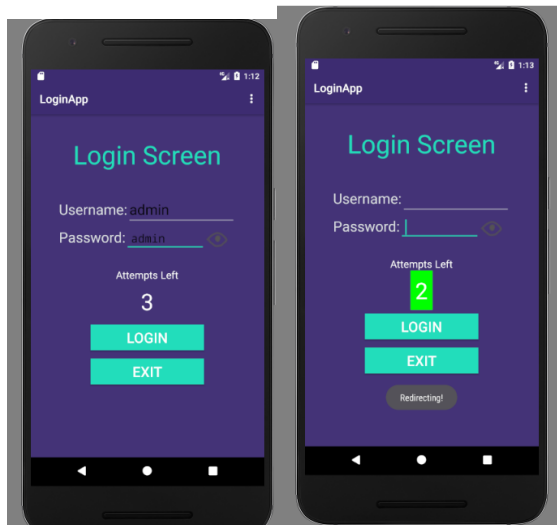
## 四、ui 调整

### (一)效果展示

开始页面，点击眯眼图标图标变成睁眼可显示密码。

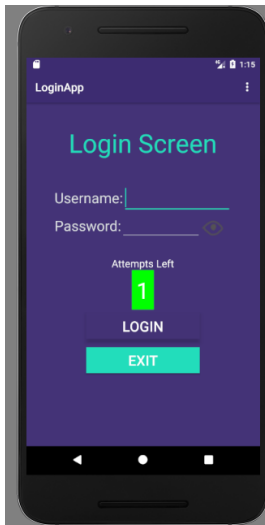


输入密码，错误，显示剩余次数，并且对话框显示“密码错误”/成功显示“成功登陆”。3次后登录按钮 不可点击。



按按钮时按钮颜色改变

可按右侧菜单栏退出



## (二)难点解决

Edittext 样式设置

代码:

Res/value/styles 文件

```
<item name="colorControlNormal">@color/colorwhite</item>
<item name="colorControlActivated">@color/colorAccent</item>
```

Res/value/color 文件

```
<color name="colorAccent">#22d4bb</color>
<color name="colorwhite">#dbdee2</color>
```

按钮变色，见上篇作业。

Over

附：

（一）一个带圆角且背景色为#55FFFFFF（淡蓝色）的 RelativeLayout 布局代码如下：

XML/HTML 代码

?

1	<?xml version="1.0" encoding="UTF-8"?>
2	<shape xmlns:android="http://schemas.android.com/apk/res/android">
3	<solid android:color="#55FFFFFF" />
4	<!-- 设置圆角
5	注意：bottomRightRadius 是左下角而不是右下角
6	bottomLeftRadius 右下角-->
7	<corners android:topLeftRadius="10dp"
8	android:topRightRadius="10dp"
	android:bottomRightRadius="10dp"
	android:bottomLeftRadius="10dp"/>
	</shape>

solid 表示填充色，这里填充的是淡蓝色。corners 是设置圆角。

dp (即 dip, device independent pixels)设备独立像素：这个和设备硬件有关，一般我们为了支持 WVGA、HVGA 和 QVGA ，不依赖像素。在 android 上开发的程序将会在不同分辨率的手机上运行。为了让程序外观不至于相差太大，所以引入了 dip 的概念。比如定义一个矩形 10 x 10dip. 在分辨率为 160dpi 的屏上，比如 G1，正好是 10 x 10 像素。而在 240 dpi 的屏，则是 15 x 15 像素。换算公式为  $\text{pixs} = \text{dips} * (\text{density}/160)$ 。density 就是屏的分辨率。

22dabb 绿色

443377 紫色

7788aa 灰色