



华南师范大学

登录界面实现和 UI 测试说明文档

课程名称：移动智能应用开发

项目名称：登录界面实现和 UI 测试

指导老师：曹阳

完成时间：2019 年 10 月 19 日

专 业：计算机科学与技术（软件方向）

姓 名：罗佳海

学 号：20172131144

华南师范大学教务处计算机学院

目录

一、	项目简介.....	3
(一)	项目名称.....	3
(二)	软件内容概述.....	3
二、	界面设计.....	3
(一)	主界面设计（中文）.....	3
(二)	主界面设计（英文）.....	5
(三)	功能设计.....	6
三、	代码设计.....	7
(一)	工程结构图.....	7
(二)	主界面布局预览.....	8
(三)	主要代码.....	8
四、	软件操作流程.....	8
(一)	流程图.....	8
五、	难点（或遇到的问题）与解决方案.....	9
六、	不足之处与未来设想.....	10
七、	主要代码.....	10
(一)	主 Activity.....	10
(二)	登录 Activity.....	12
(三)	UI 测试 BaseEspresso 类（通用方法）.....	16
(四)	具体 UI 测试（正确登录、错误登录）.....	22
(五)	布局文件.....	24
(六)	对话框弹出弹入动画.....	29
(七)	仿 Material Design 背景阴影.....	29
(八)	圆角矩形背景的 EditText.....	31
(八)	颜色设计.....	32
(九)	样式.....	32
(十)	字符串设定.....	34

一、 项目简介

（一）项目名称

登录界面实现和 UI 测试

（二）软件内容概述

该登录界面是基于 Android Studio 开发。

该登录界面的设计方面，采用扁平化风格，动效也尽可能参照 Material Design 风格，配色方面采用浅色调。与上一次不同的是，所有的按钮都用了自带的水波纹效果，而不用在 drawble 为按钮创建单独的 selector 来显示按钮按下的效果。

同时，为登录完成了双语界面、简单的输入的安全检查和错误尝试功能。另外还加了一个超链接跳转的功能。在登录按钮点击后，会有 ProgressBar 提示加载登录。

UI 测试方面，完成了对点击行为、输入行为、视图是否正确出现、按钮的可用状态等进行了测试。

二、 界面设计

（一）主界面设计（中文）

登录界面的实现

账号密码登录

用户名/邮箱

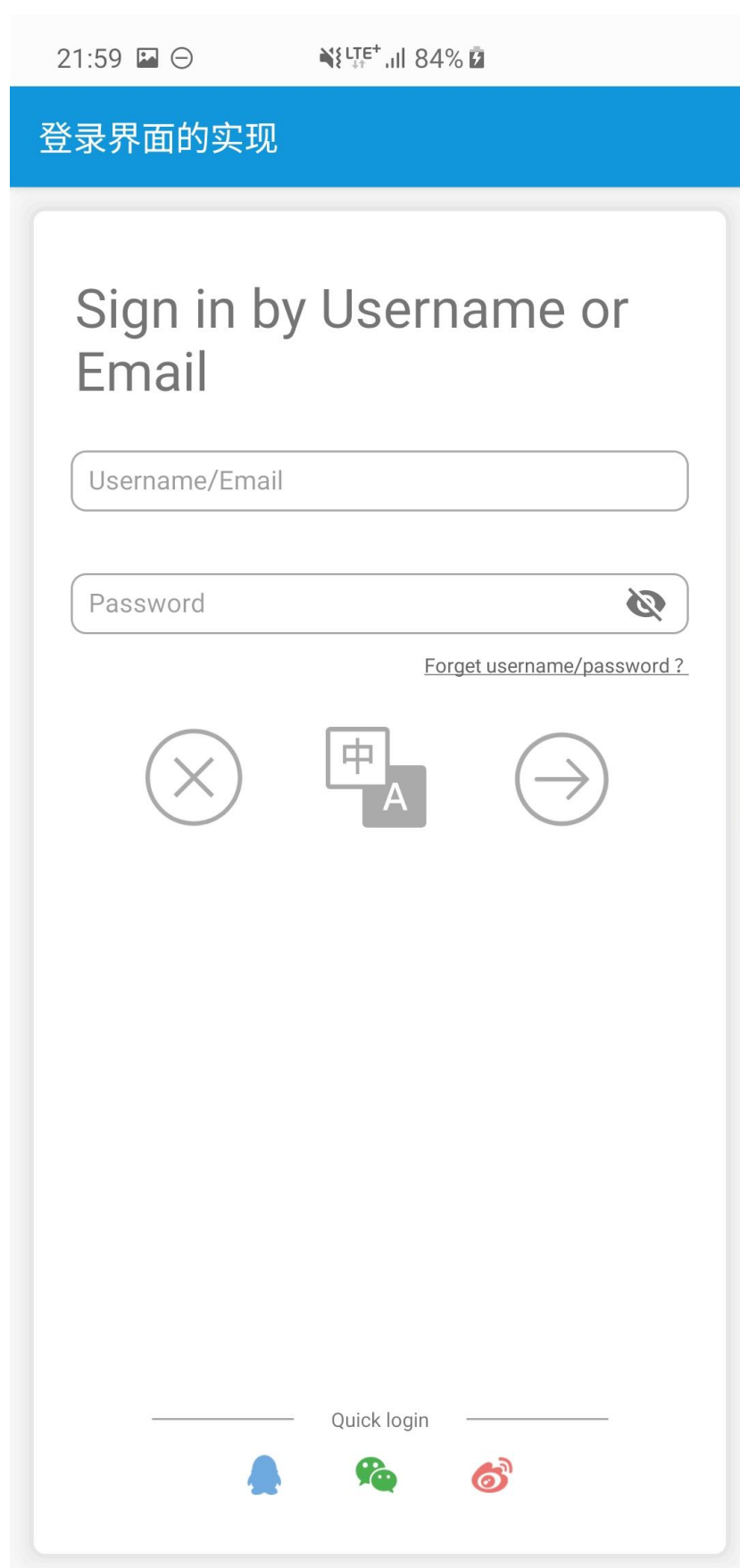
密码

[忘记用户名/密码？](#)

第三方登录



(二) 主界面设计 (英文)



(三) 功能设计

1. TextInputEditText 的 hint 与密码显示隐藏功能



Hint 能够在正确位置显示，没有与圆角矩形重合

2. 登录对话框弹出的动画

//dialog_pop_up_anim

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <!-- 弹出时动画 -->
    <translate
        android:duration="500"
        android:fromYDelta="100%p"
        android:toYDelta="0"
        android:interpolator="@android:anim/accelerate_interpolator"/>
</set>
```

//dialog_push_back_anim.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <!-- 弹入时动画 -->
    <translate
        android:duration="500"
        android:fromYDelta="0"
        android:toYDelta="100%p"
        android:interpolator="@android:anim/accelerate_interpolator"/>
</set>
```

原本做了对话框，发现 Activity 也可以用

3. ImageButton 水波纹

//style.xml

```
<style name="quick_login_btn_style">
    <item name="android:layout_width">wrap_content</item>
```

```

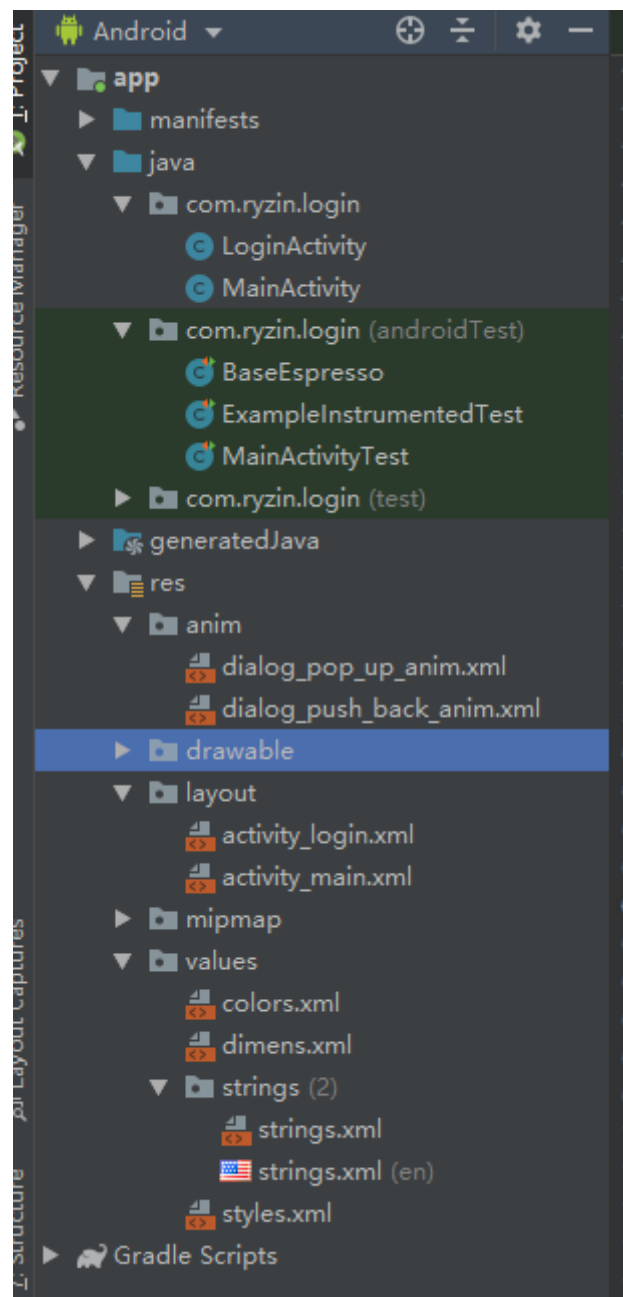
<item name="android:layout_height">wrap_content</item>
<item name="android:layout_marginHorizontal">20dp</item>
<item name="android:clickable">true</item>
<item name="android:focusable">true</item>
<item name="android:background">?android:attr/selectableItemBackgroundBorderless</item>
</style>

```

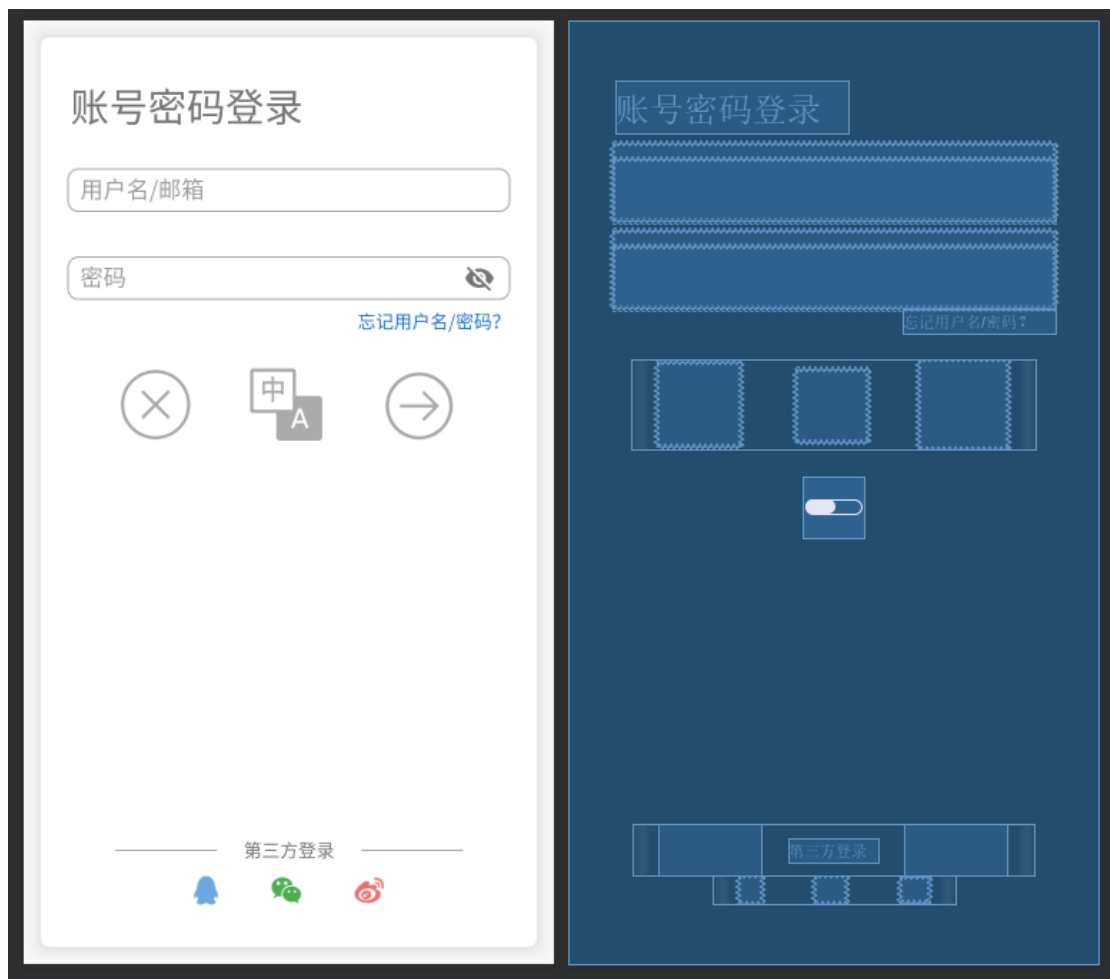
圆形水波纹

三、 代码设计

(一) 工程结构图



（二）主界面布局预览

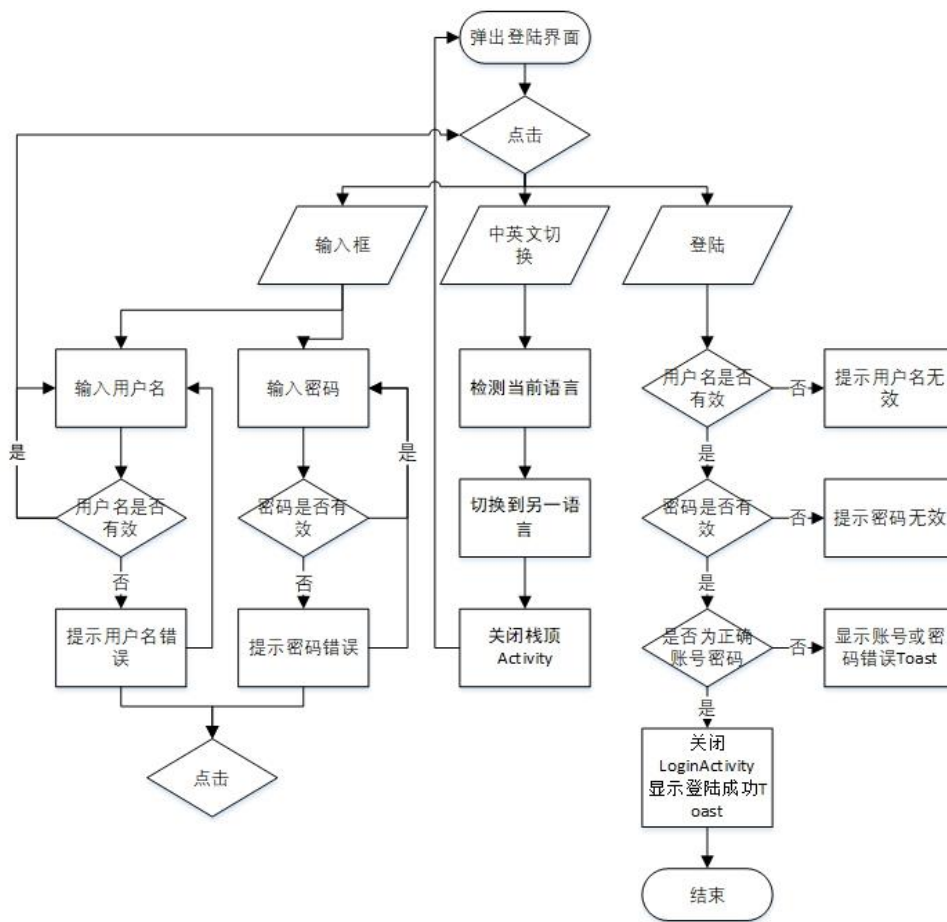


（三）主要代码

在文末展示。

四、 软件操作流程

（一）流程图



五、 难点（或遇到的问题）与解决方案

1. EditText 的 Hint 显示的位置：

一开始设想用圆角矩形作为 EditText 框的背景，没想到 Hint 的位置与圆角矩形重合了，有点难看，似乎也没有属性是调整 Hint 的显示位置的。只好用 Layer-list 在 EditText 上方模拟一个区域，隔开圆角矩形来正确显示 Hint。

2.水波纹：

自带的水波纹占用了 background 属性，所以之前占用 background 属性的地方要修改。

3.仿 Material Design 背景阴影的渐变：

如今有 Material 的依赖可以直接写一两行代码就可以有这个阴影，但是总有地方会没有，所以还是自己尝试一下。发现是渐变的感觉最难模仿，还是用 Layer-list 铺了很多层去模仿。当然也可以用其他软件画出背景透明的阴影图片来实现。

4.UI 测试中对 Toast 等没有 id 的控件的检测:

首先 Toast 显示得比较慢, 需要线程休眠一会才可以进行检测。网上说由于 Toast 的布局并不是属于我们当前应用的, 而是通过另一个 Service 控制的, 所以不能直接用普通的方法去检测 Toast 的内容是否是我们想要的。

此处通过 withText 找到对应的 Toast 的视图, 但这无法保证就是 Toast, 所以, 我们还需要保证我们找的 Toast 视图不在主界面的 DecorView 中, 这里使用了 withDecorView 和 not 两个 Matcher。最后, 通过 check 检查是否显示。

5.升级 AndroidX 依赖的写法:

网上的写法一般是旧版的, 如果想升级到 AndroidX, 可以查看文档。

<https://developer.android.google.cn/jetpack/androidx/migrate#migrate>

六、 不足之处与未来设想

1.登录功能的安全检测应更加全面, 现阶段只是对空、字符数等做了检测, 未来可以用正则表达式对其检查。

2.通过观察 Android Studio 自带的登录 Demo, 官方的在细节上、可扩展上考虑得更加全面。

3.对 Activity 跳转等进行更具体的测试。

本登录界面最初希望实现注册等的功能, 但实际上由于个人时间安排原因, 没有继续深入研究。对比现阶段的登陆, 功能上还有很大差距, 但基本上已经有了大概的样子。在大作业中我会花更多的精力, 考虑更加周全。

七、 主要代码

(一) 主 Activity

```
//java/com.ryzin.login/MainActivity.java
```

```
package com.ryzin.login;

import android.content.Intent;
import android.os.Bundle;
```

```

import androidx.appcompat.app.AppCompatActivity;
import android.view.KeyEvent;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    private Button loginBtn;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        loginBtn = findViewById(R.id.login_btn);

        loginBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this, LoginActivity.class);
                intent.addCategory("sd");
                startActivity(intent);
            }
        });
    }

    //（再按一次退出程序）两秒内双击返回键退出 app
    private long exitTime = 0;

    @Override
    public boolean onKeyDown(int keyCode, KeyEvent event) {
        if (keyCode == KeyEvent.KEYCODE_BACK
            && event.getAction() == KeyEvent.ACTION_DOWN) {
            if ((System.currentTimeMillis() - exitTime) > 2000) {
                Toast.makeText(getApplicationContext(), "再按一次退出程序",
                    Toast.LENGTH_SHORT).show();
                exitTime = System.currentTimeMillis();
            } else {
                finish();
                System.exit(0);
            }
        }
    }
}

```

```

        return true;
    }
    return super.onKeyDown(keyCode, event);
}
}

```

（二）登录 Activity

//java/com.ryzin.login/LoginActivity.java

```

package com.ryzin.login;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.content.res.Configuration;
import android.os.Bundle;
import android.os.Handler;
import android.text.Editable;
import android.text.TextWatcher;
import android.text.method.LinkMovementMethod;
import android.util.Patterns;
import android.view.KeyEvent;
import android.view.View;
import android.view.inputmethod.EditorInfo;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

import java.util.Locale;

public class LoginActivity extends AppCompatActivity {

    //登录错误计数器
    private int counter = 3;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
    }
}

```

```

final EditText usernameEditText = findViewById(R.id.username);
final EditText passwordEditText = findViewById(R.id.password);
final ImageButton loginButton = findViewById(R.id.login);
final ImageButton languageBtn = findViewById(R.id.language);
final ProgressBar loadingProgressBar = findViewById(R.id.loading);
final ImageButton loginBackBtn = findViewById(R.id.cancel);

//添加监听事件
TextWatcher afterTextChangedListener = new TextWatcher() {
    @Override
    public void beforeTextChanged(CharSequence s, int start, int count, int after) {
        // ignore
    }

    @Override
    public void onTextChanged(CharSequence s, int start, int before, int count) {
        // ignore
    }

    @Override
    public void afterTextChanged(Editable s) {
        if(!isPasswordValid(passwordEditText.getText().toString()))
passwordEditText.setError(getResources().getString(R.string.invalid_password));
        if(!isUserNameValid(usernameEditText.getText().toString()))
usernameEditText.setError(getResources().getString(R.string.invalid_username));
    }
};

usernameEditText.addTextChangedListener(afterTextChangedListener);
passwordEditText.addTextChangedListener(afterTextChangedListener);

loginButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        loadingProgressBar.setVisibility(View.VISIBLE); //加载圈转一会，缓解用户等待
情绪
        loginButton.setEnabled(false); //让按钮在加载过程中禁用

        new Handler().postDelayed(new Runnable() { //延时 1s 执行，让加载圈转一会
            @Override
            public void run() {

```

```

        //do something
        loadingProgressBar.setVisibility(View.GONE);

        //如果输入框没有变化就按了登录的情况
        if(!isUserNameValid(usernameEditText.getText().toString())) {
usernameEditText.setError(getResources().getString(R.string.invalid_username));
            loginButton.setEnabled(true);
            return;
        }
        if(!isPasswordValid(passwordEditText.getText().toString())) {
passwordEditText.setError(getResources().getString(R.string.invalid_password));
            loginButton.setEnabled(true);
            return;
        }

        //错误次数达到上限
        if(counter <= 0) {
            Toast.makeText(getApplicationContext(), R.string.lock_login,
Toast.LENGTH_LONG).show();
            return;
        }
        //成功登录
        if (usernameEditText.getText().toString().equals("admin") &&
passwordEditText.getText().toString().equals("admin")) {
            //提示
            Toast.makeText(getApplicationContext(), R.string.welcome,
Toast.LENGTH_LONG).show();
            finish(); //输入正确
        }
        else {
            counter--;
            //提示
            Toast.makeText(getApplicationContext(), R.string.login_failed,
Toast.LENGTH_LONG).show();
        }
        loginButton.setEnabled(true);
    }
}, 1000);

}

});

```

```

loginBackBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        //waiting fix
        finish();
        //System.exit(0);
    }
});

//超链接点击跳转
TextView forget_password_textview = findViewById(R.id.forget_password_text_view);
forget_password_textview.setMovementMethod(LinkMovementMethod.getInstance()); //其实就这一句是关键

//切换语言按钮绑定响应事件
languageBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Locale locale = getResources().getConfiguration().locale; //获取当前语言
        String language = locale.getLanguage();
        shiftLanguage(language);
    }
});

}

// 检测账号有效性
private boolean isUserNameValid(String username) {
    if (username == null) {
        return false;
    }
    if (username.contains("@")) {
        return Patterns.EMAIL_ADDRESS.matcher(username).matches();
    } else {
        return !username.trim().isEmpty();
    }
}

// 检测密码有效性
private boolean isPasswordValid(String password) {
    return password != null && password.trim().length() > 1;
}

```

```

//切换语言
public void shiftLanguage(String sta) {
    Configuration config = getBaseContext().getResources().getConfiguration();
    if (sta.equals("zh")) {
        Locale.setDefault(Locale.ENGLISH);
        config.locale = Locale.ENGLISH;
    } else {
        Locale.setDefault(Locale.CHINESE);
        config.locale = Locale.CHINESE;
    }
    getBaseContext().getResources().updateConfiguration(config,
        getBaseContext().getResources().getDisplayMetrics());
    refreshSelf();
}

//重启顶部 Activity
public void refreshSelf() {
    Intent intent = new Intent(this, MainActivity.class);
    intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    startActivity(intent);
}
}

```

（三）UI 测试 BaseEspresso 类（通用方法）

//java/com.ryzin.login/BaseEspresso.java

```

//BaseEspresso 基类

package com.ryzin.login;

import android.app.Activity;
import android.content.Intent;

import androidx.test.espresso.intent.rule.IntentsTestRule;

import org.junit.Rule;
import org.junit.runner.RunWith;
import org.junit.runners.JUnit4;

import java.lang.reflect.ParameterizedType;

```



```

import static androidx.test.espresso.Espresso.onView;
import static androidx.test.espresso.action.ViewActions.clearText;
import static androidx.test.espresso.action.ViewActions.click;
import static androidx.test.espresso.action.ViewActions.closeSoftKeyboard;
import static androidx.test.espresso.action.ViewActions.replaceText;
import static androidx.test.espresso.action.ViewActions.scrollTo;
import static androidx.test.espresso.assertion.ViewAssertions.matches;
import static androidx.test.espresso.matcher.RootMatchers.withDecorView;
import static androidx.test.espresso.matcher.ViewMatchers.isDisplayed;
import static androidx.test.espresso.matcher.ViewMatchers.isEnabled;
import static androidx.test.espresso.matcher.ViewMatchers.withId;
import static androidx.test.espresso.matcher.ViewMatchers.withText;
import static org.hamcrest.CoreMatchers.not;
import static org.hamcrest.core.AllOf.allOf;

@RunWith(JUnit4.class)

public class BaseEspresso<T extends Activity> {

    @Rule
    public IntentsTestRule<T> mActivityRule = new IntentsTestRule<T>((Class<T>)
((ParameterizedType) getClass().getGenericSuperclass()).getActualTypeArguments()[0], true,
false);

    public T mActivity = null;

    /**
     * 初始化 Activity
     *
     * @return
     */
    public T getActivity() {
        return getActivity(null);
    }

    /**
     * 初始化 Activity
     *
     * @return mActivity
     */
    public T getActivity(Intent intent) {

```

```

        if (intent == null) {
            intent = new Intent();
        }

        if (mActivity == null) {
            mActivityRule.launchActivity(intent);
            mActivity = mActivityRule.getActivity();
        }

        return mActivity;
    }

    /**
     * 模拟用户的点击行为
     *
     * @param id
     */
    public void testClick(final int id) {
        onView(withId(id)).perform(closeSoftKeyboard(), click());
    }

    /**
     * 模拟用户的点击行为
     *
     * @param text
     */
    public void testClick(String text) {
        onView(withText(text)).perform(closeSoftKeyboard(), click());
    }

    /**
     * 模拟用户的点击行为
     *
     * @param id
     * @param text
     */
    public void testClick(final int id, String text) {
        onView(allOf(withId(id), withText(text))).perform(closeSoftKeyboard(), click());
    }

    /**
     * 模拟用户的点击行为
     *

```

```

    * @param id
    * @param scrollTo
    */
    public void testClick(final int id, boolean scrollTo) {
        if (scrollTo) {
            onView(allOf(withId(id))).perform(closeSoftKeyboard(), scrollTo(), click());
        } else {
            onView(allOf(withId(id))).perform(closeSoftKeyboard(), click());
        }
    }
}

/**
 * 模拟用户的输入文本行为
 *
 * @param id
 * @param text
 * @return
 */
public String testInputText(final int id, String text) {
    onView(withId(id)).perform(clearText(), replaceText(text), closeSoftKeyboard());
    return text;
}

/**
 * 检查 View 的文本变化是否正确
 *
 * @param id
 * @param text
 */
public void testTextEquals(final int id, String text) {
    onView(withId(id)).check(matches(withText(text)));
}

/**
 * 检查 View 是否可见
 *
 * @param id
 */
public void testViewVisible(final int id) {
    onView(withId(id))

```

```

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

    /**
     * 检查 View 是否可见
     *
     * @param text
     */
    public void testViewVisible(String text) {
        onView(withText(text))
            .check(matches(isDisplayed()));
    }

    /**
     * 检查 View 是否可见 (Toast 等)
     *
     * @param id
     * @param uselessViable
     */
    public void testViewVisible(final int id, int uselessViable) {
        onView(withText(id))
            .inRoot(withDecorView(not(mActivityRule.getActivity().getWindow().getDecorVi
ew()))))
            .check(matches(isDisplayed()));
    }

    /**
     * 检查 View 是否可见
     *
     * @param id
     * @param text
     */
    public void testViewVisible(final int id, String text) {
        onView(allOf(withId(id), withText(text)))
            .check(matches(isDisplayed()));
    }

    /**
     * 检查 View 是否不可见
     *

```

```

    * @param id
    */
    public void testViewUnVisible(final int id) {
        onView(withId(id))
            .check(matches(not(isDisplayed())));
    }

    /**
     * 检查 View 是否可用
     *
     * @param id
     */
    public void testViewEnable(int id) {
        onView(withId(id))
            .check(matches(isEnabled()));
    }

    /**
     * 检查 View 是否不可用
     *
     * @param id
     */
    public void testViewUnEnable(int id) {
        onView(withId(id))
            .check(matches(not(isEnabled())));
    }

    /**
     * 休眠
     *
     * @param time
     */
    public void sleep(long time) {
        try {
            Thread.sleep(time);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}

```

```

/**
 * 获取字符串
 *
 * @param resId
 * @return
 */
public String getString(int resId) {
    if (mActivity == null) {
        getActivity();
    }
    return mActivity.getString(resId);
}
}

```

（四）具体 UI 测试（正确登录、错误登录）

// java/com.ryzin.calculator/Trans. java

```

package com.ryzin.login;

import android.content.Intent;

import org.junit.Assert;
import org.junit.Test;

import static org.junit.Assert.*;

public class MainActivityTest extends BaseEspresso<MainActivity>{

    @Test
    public void onCreate() {
        //启动 MainActivity
        getActivity();
        Assert.assertNotNull(mActivity);
        sleep(1000); //便于开发者查看情况

        //启动 LoginActivity
        testClick(R.id.login_btn);

        //测试正确登录////////////////////////////////////
        //测试 View 点击行为
        testClick(R.id.username);
        sleep(1000); //便于开发者查看情况
    }
}

```

```

//测试 View 输入与文本变化是否正确
testInputText(R.id.username, "a");
testTextEquals(R.id.username, "a");

//测试 hint 是否正确出现
testViewVisible(R.string.invalid_username, 1);
sleep(1000);

testInputText(R.id.username, "admin");
sleep(1000);

//测试 View 点击行为
testClick(R.id.password);
sleep(1000);

//测试 View 输入与文本变化是否正确
testInputText(R.id.password, "a");
testTextEquals(R.id.password, "a");

//测试 hint 是否正确出现
testViewVisible(R.string.invalid_password, 1);
sleep(1000);

testInputText(R.id.password, "admin");
sleep(1000);

//测试登录后, Toast 提示是否正确
testClick(R.id.login);
testViewVisible(R.id.loading);
sleep(2000); //等一会, Toast 出来得比较慢
testViewVisible(R.string.welcome, 1);
sleep(1000);

//测试错误登录////////////////////////////////////
//启动 LoginActivity
testClick(R.id.login_btn);
sleep(1000); //便于开发者查看情况

//测试 View 点击行为
testClick(R.id.username);
sleep(1000); //便于开发者查看情况

//测试 View 输入

```

```

testInputText(R.id.username, "asd");
testInputText(R.id.password, "dsad");

//测试登录后，第一次 Toast 提示是否正确
testViewEnable(R.id.login);
testClick(R.id.login);
testViewVisible(R.id.loading);
sleep(2000); //等一会，Toast 出来得比较慢
testViewVisible(R.string.login_failed, 1);
sleep(1000);

//测试登录后，第二次 Toast 提示是否正确
testViewEnable(R.id.login);
testClick(R.id.login);
testViewVisible(R.id.loading);
sleep(2000); //等一会，Toast 出来得比较慢
testViewVisible(R.string.login_failed, 1);
sleep(1000);

//测试登录后，第三次 Toast 提示是否正确
testViewEnable(R.id.login);
testClick(R.id.login);
testViewVisible(R.id.loading);
sleep(2000); //等一会，Toast 出来得比较慢
testViewVisible(R.string.login_failed, 1);
sleep(1000);

//测试登录后，第四次 Toast 提示是否正确
testViewEnable(R.id.login);
testClick(R.id.login);
testViewVisible(R.id.loading);
sleep(2000); //等一会，Toast 出来得比较慢
testViewVisible(R.string.lock_login, 1);
testViewUnEnable(R.id.login); //错误次数达到上限，登录按钮不可用
sleep(1000);
}
}

```

（五）布局文件

//res/layout/activity_login.xml（登录界面）

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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:id="@+id/login_dialog"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@drawable/dialog_login_background"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingBottom="@dimen/activity_vertical_margin">

<TextView
    android:id="@+id/sign_in_title"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/title_sign_in"
    android:textSize="30dp"
    android:layout_marginTop="30dp"
    android:layout_marginStart="20dp"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintStart_toStartOf="parent"
/>

<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/username_layout"
    tools:ignore="MissingConstraints"
    style="@style/TextInputLayout_login"
    app:layout_constraintTop_toBottomOf="@+id/sign_in_title">
    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/username"
        style="@style/EditText_login"
        android:hint="@string/prompt_email"
        android:inputType="textPersonName"/>
</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/password_layout"
    tools:ignore="MissingConstraints"
    style="@style/TextInputLayout_login"
    app:layout_constraintTop_toBottomOf="@+id/username_layout"
    app:passwordToggleEnabled="true">
    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/password"
        style="@style/EditText_login"

```

```

        android:hint="@string/prompt_password"
        android:imeActionLabel="@string/action_sign_in_short"
        android:imeOptions="actionDone"
        android:inputType="textPassword"
        app:layout_constraintTop_toBottomOf="@+id/username_layout" />
</com.google.android.material.textfield.TextInputLayout>

<TextView
    android:id="@+id/forget_password_text_view"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/state_forget_password"
    android:textColor="@color/colorHrefText"
    app:layout_constraintTop_toBottomOf="@+id/password_layout"
    app:layout_constraintEnd_toEndOf="@+id/password_layout"/>

<LinearLayout
    android:id="@+id/main_btn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@id/forget_password_text_view"
    android:layout_marginTop="20dp"
    android:gravity="center_vertical">

    <ImageButton
        android:id="@+id/cancel"
        style="@style/quick_login_btn_style"
        android:src="@drawable/button_cancel_icon"
        />

    <ImageButton
        android:id="@+id/language"
        style="@style/quick_login_btn_style"
        android:src="@drawable/button_language_icon"
        />

    <ImageButton
        android:id="@+id/login"
        style="@style/quick_login_btn_style"
        android:src="@drawable/button_sign_in_icon"
        />

```

```

</LinearLayout>

<ProgressBar
    android:id="@+id/loading"
    style="?android:attr/progressBarStyle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:visibility="invisible"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@id/main_btn"
    android:layout_marginTop="20dp"/>

<LinearLayout
    android:id="@+id/quick_login_title"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintBottom_toTopOf="@id/quick_login_btn">

    <View
        android:id="@+id/quick_login_line_left"
        android:layout_width="80dp"
        android:layout_height="40dp"
        android:layout_marginHorizontal="20dp"
        android:background="@drawable/quick_login_title">

    </View>

    <TextView
        android:id="@+id/quick_login_prompt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_vertical"
        android:text="@string/title_quick_login" />

    <View
        android:id="@+id/quick_login_line_right"
        android:layout_width="80dp"
        android:layout_height="40dp"
        android:layout_marginHorizontal="20dp"
        android:background="@drawable/quick_login_title">

    </View>
</LinearLayout>

```

```

<LinearLayout
    android:id="@+id/quick_login_btn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintBottom_toBottomOf="parent"
    android:layout_marginBottom="30dp"
    android:gravity="center_vertical">

    <ImageButton
        android:id="@+id/quick_login_qq"
        style="@style/quick_login_btn_style"
        android:src="@drawable/button_quick_login_qq_icon"
    />

    <ImageButton
        android:id="@+id/quick_login_wechat"
        style="@style/quick_login_btn_style"
        android:src="@drawable/button_quick_login_wechat_icon"
    />

    <ImageButton
        android:id="@+id/quick_login_sina_weibo"
        style="@style/quick_login_btn_style"
        android:src="@drawable/button_quick_login_sina_weibo_icon"
    />
</LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

```

//res/layout/activity_main.xml (主界面)

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toTopOf="parent">
    <Button
        android:id="@+id/login_btn"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:text="@string/action_sign_in_short">

    </Button>
</LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

```

（六）对话框弹出弹入动画

//res/anim/dialog_pop_up_anim.xml

```

<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <!-- 弹出时动画 -->
    <translate
        android:duration="500"
        android:fromYDelta="100%p"
        android:toYDelta="0"
        android:interpolator="@android:anim/accelerate_interpolator"/>
</set>

```

//res/anim/dialog_push_back_anim.xml

```

<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <!-- 弹入时动画 -->
    <translate
        android:duration="500"
        android:fromYDelta="0"
        android:toYDelta="100%p"
        android:interpolator="@android:anim/accelerate_interpolator"/>
</set>

```

（七）仿 Material Design 背景阴影

//res/drawable/dialog_login_background.xml

```
<?xml version="1.0" encoding="utf-8"?>
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">
    <item>
        <shape android:shape="rectangle">
            <padding
                android:bottom="3dp"
                android:left="3dp"
                android:right="3dp"
                android:top="3dp" />
            <solid android:color="#0DCCCC" />
            <corners android:radius="10dp" />
        </shape>
    </item>
    <item>
        <shape android:shape="rectangle">
            <padding
                android:bottom="3dp"
                android:left="3dp"
                android:right="3dp"
                android:top="3dp" />
            <solid android:color="#10CCCC" />
            <corners android:radius="10dp" />
        </shape>
    </item>
    <item>
        <shape android:shape="rectangle">
            <padding
                android:bottom="3dp"
                android:left="3dp"
                android:right="3dp"
                android:top="3dp" />
            <solid android:color="#15CCCC" />
            <corners android:radius="10dp" />
        </shape>
    </item>
    <item>
        <shape android:shape="rectangle">
            <padding
                android:bottom="3dp"
                android:left="3dp"
                android:right="3dp"
                android:top="3dp" />
            <solid android:color="#20CCCC" />
        </shape>
    </item>
```

```

        <corners android:radius="10dp" />
    </shape>
</item>
<item>
    <shape android:shape="rectangle">
        <padding
            android:bottom="3dp"
            android:left="3dp"
            android:right="3dp"
            android:top="3dp" />
        <solid android:color="#30CCCC" />
        <corners android:radius="10dp" />
    </shape>
</item>
<item>
    <shape>
        <solid android:color="#FFFFFF" />
        <corners android:radius="6dp" />
    </shape>
</item>
</layer-list>

```

（八）圆角矩形背景的 EditText

//res/drawable/edit_text_login.xml

```

<layer-list xmlns:android="http://schemas.android.com/apk/res/android">
    <!--模拟背景的 padding 区域 和背景不突兀 看不出的颜色-->
    <item>
        <shape>
            <!-- 实心 -->
            <solid android:color="@android:color/transparent"/>
        </shape>
    </item>
    <!--模拟背景的 padding 的 size 6dp, 为了 hintText 上移一点-->
    <item
        android:bottom="8dp"
        android:top="8dp">
        <!--原来的需求区域-->
        <shape xmlns:android="http://schemas.android.com/apk/res/android">
            <!--填充色-->
            <solid android:color="@android:color/white"/>
            <!--描边-->
            <stroke android:color="@android:color/darker_gray" android:width="1dp"/>
            <!--圆角角度-->

```

```

        <corners android:radius="8dp"/>

    </shape>
</item>
</layer-list>

```

（八）颜色设计

//res/values/colors.xml

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#1296db</color>
    <color name="colorPrimaryDark">#00574B</color>
    <color name="colorAccent">#666666</color>
    <color name="colorHrefText">#1C71CF</color>
    <color name="colorLightGray">#666666</color>
    <color name="colorNavBarButton">#515151</color>
    <color name="colorLightWhite">#c3c3c3</color>
    <color name="colorWhite">#fcfcfc</color>
</resources>

```

（九）样式

//res/values/styles.xml

```

<resources>

    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
        <!-- Customize your theme here. -->
        <item name="android:windowNoTitle">true</item>
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorWhite</item>
        <item name="colorAccent">@color/colorAccent</item>
        <!--亮色状态栏，则会使用黑色图标-->
        <item name="android:windowLightStatusBar">true</item>
    </style>

    <style name="TextInputLayout_login">
        <item name="android:layout_width">0dp</item>
        <item name="android:layout_height">60dp</item>
        <item name="android:layout_marginEnd">18dp</item>
        <item name="android:layout_marginStart">18dp</item>
        <item name="android:layout_marginTop">8dp</item>
    </style>

```



```

<!--<item name="counterEnabled">true</item>-->
<!--<item name="counterMaxLength">20</item>-->
<item name="errorEnabled">true</item>
<item name="layout_constraintEnd_toEndOf">parent</item>
<item name="layout_constraintStart_toStartOf">parent</item>
</style>

<style name="EditText_login">
    <item name="android:layout_width">match_parent</item>
    <item name="android:layout_height">50dp</item>
    <item name="android:background">@drawable/edit_text_login</item>
    <item name="android:paddingHorizontal">10dp</item>
    <item name="android:selectAllOnFocus">true</item>
    <item name="android:textSize">18sp</item>
    <item name="hintTextAppearance">@style/hintText_login</item>
</style>

<style name="hintText_login" parent="TextAppearance.Design.Hint">
    <item name="android:textSize">20sp</item>
</style>

<style name="login_dialog_style">
    <item name="android:windowFrame">@null</item>
    <!--边框-->
    <item name="android:windowIsFloating">true</item>
    <!--是否浮现在 activity 之上-->
    <item name="android:windowIsTranslucent">false</item>
    <!--半透明-->
    <item name="android:windowNoTitle">true</item>
    <!--无标题-->
    <item name="android:windowBackground">@android:color/transparent</item>
    <!--背景透明-->
    <item name="android:backgroundDimEnabled">false</item>
    <!--模糊-->
</style>

<style name="login_dialog_pop_up_anim" parent="android:Animation">
    <item name="android:windowEnterAnimation">@anim/dialog_pop_up_anim</item>
    <!--对话框进入动画-->
    <item name="android:windowExitAnimation">@anim/dialog_push_back_anim</item>
</style>

<style name="quick_login_btn_style">
    <item name="android:layout_width">wrap_content</item>

```

```

        <item name="android:layout_height">wrap_content</item>
        <item name="android:layout_marginHorizontal">20dp</item>
        <item name="android:clickable">true</item>
        <item name="android:focusable">true</item>
        <item
name="android:background">?android:attr/selectableItemBackgroundBorderless</item>
    </style>

    <style name="AppTheme.NoActionBar">
        <item name="windowActionBar">false</item>
        <item name="windowNoTitle">true</item>
    </style>

    <style name="AppTheme.AppBarOverlay" parent="ThemeOverlay.AppCompat.Dark.ActionBar" />

    <style name="AppTheme.PopupOverlay" parent="ThemeOverlay.AppCompat.Light" />
</resources>

```

(十) 字符串设定

//res/values/strings.xml

```

<resources>
    <string name="app_name">登录界面的实现</string>
    <string name="title_activity_login">登录账号</string>
    <string name="title_sign_in">账号密码登录</string>
    <string name="title_register">手机号注册</string>
    <string name="prompt_email">用户名/邮箱</string>
    <string name="prompt_password">密码</string>
    <string name="action_sign_in">登录或注册</string>
    <string name="action_sign_in_short">登录</string>
    <string name="welcome">"欢迎!"</string>
    <string name="invalid_username">用户名无效</string>
    <string name="invalid_password">密码必须多于一个字符</string>
    <string name="login_failed">"账号或密码错误"</string>
    <string name="state_forget_password"><a href="https://www.bing.com/">忘记用户名/密码?
</a></string>
    <string name="title_quick_login">第三方登录</string>
    <string name="left_enter_password">剩余输入密码次数: </string>
    <string name="lock_login">错误次数已达上限</string>
</resources>

```

//res/values/strings.xml (en)

```
<resources>
  <string name="app_name">Login Implement</string>
  <string name="title_activity_login">Sign in</string>
  <string name="title_sign_in">Sign in by Username or Email</string>
  <string name="title_register">Phone register</string>
  <string name="prompt_email">Username/Email</string>
  <string name="prompt_password">Password</string>
  <string name="action_sign_in">Sign in or register</string>
  <string name="action_sign_in_short">Sign in</string>
  <string name="welcome">"Welcome !"</string>
  <string name="invalid_username">Not a valid username</string>
  <string name="invalid_password">Password must be >1 characters</string>
  <string name="login_failed">"Wrong username or password"</string>
  <string name="state_forget_password"><a href="https://www.bing.com/">Forget
username/password? </a></string>
  <string name="left_enter_password">Available times for entering password left:</string>
  <string name="title_quick_login">Quick login</string>
  <string name="lock_login">Wrong password out of limit</string>
</resources>
```