

# 实验一

## 登陆与注册app设计与实现说明文档

作者：梁子宏 学号：20162180176

完成时间：2018.11.17

### 一、软件名称

具有登陆与注册功能的一个页面，并进行 UI 单元测试和支持中英双语功能。

### 二、软件内容简介

知识点使用：RelativeLayout 相对布局，SharedPreferences 文件存储，MVP 设计模式，渐变背景设计，双语设置，账号选择功能。

其实这只是一个简单的登陆与注册的界面设计，其实现不难，其中的双语功能和 UI 界面设计老师课堂上都有讲过，所以实现简单。

这款登陆与注册简单完成就非常简单。于是我做了一些特殊地设计，在这里，我运用了 MVP，model，view，presenter 的设计模式，将模型，界面和 presenter 分开。MVP 设计模式大体来说就是把 presenter 作为中介，view 所需要的东西都由 presenter 来进行沟通。打个比方，小明生病了，要去药房买药吃，MVP 设计模式就是由小明妈妈作为 presenter，让小明妈妈去买药，之后妈妈再喂小明吃药，这里就是 MVP 的思想。

但是为什么要用 MVP 的设计模式了，原因有两个。

一、安卓的特点就是代码多，通常 activity 会十分臃肿，一个普通功能的 activity 都要几百行代码。因为 activity 既作为 view，也负责数据的整理，所以，利用 MVP 模式可以将 activity 的代码最小化，仅仅完成控件初始化，设置响应事件的基本代码就可以。

二、当需要添加功能的时候，MVP 模式非常方便，需要添加某个功能，只需要在 presenter 的接口中添加函数，无需对 activity 做大量代码的添加，就像上个例子一样，小明只需要一句话就可以让小明妈妈干这样干那样，MVP 模式就是这样用的。

在布局方面，使用了一个渐变的颜色背景，这个不提，比较简单。

在 model 模型设计中，利用 SharedPreferences 进行用户名和对应密码的存储。还通过 getAll() 返回所以已注册的用户，实现下拉列表的提示。

### 三、界面设计

#### 1、主界面



## 2、注册成功提示



### 3、注册失败提示



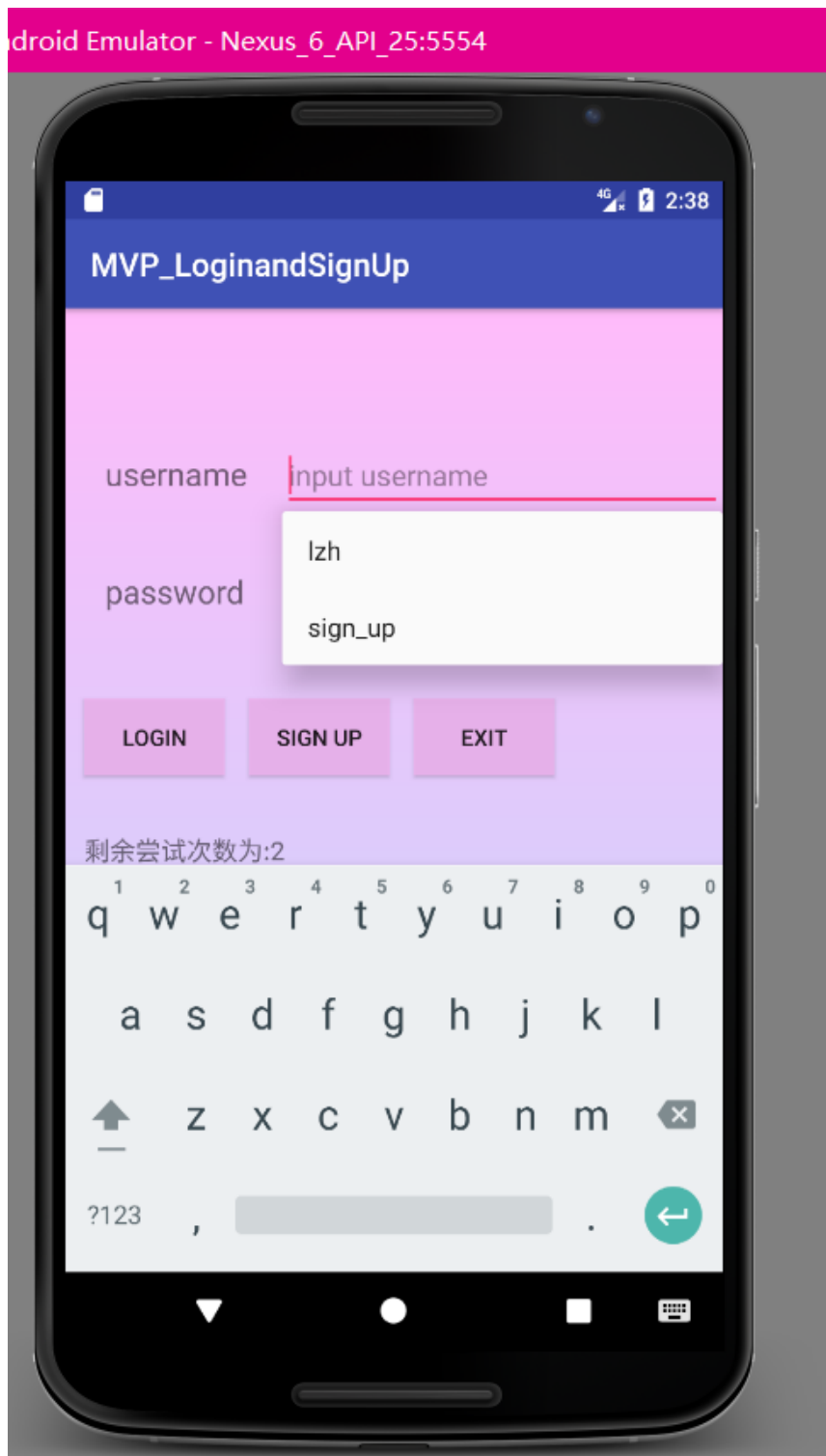
#### 4、登陆成功提示



## 5、登陆失败提示

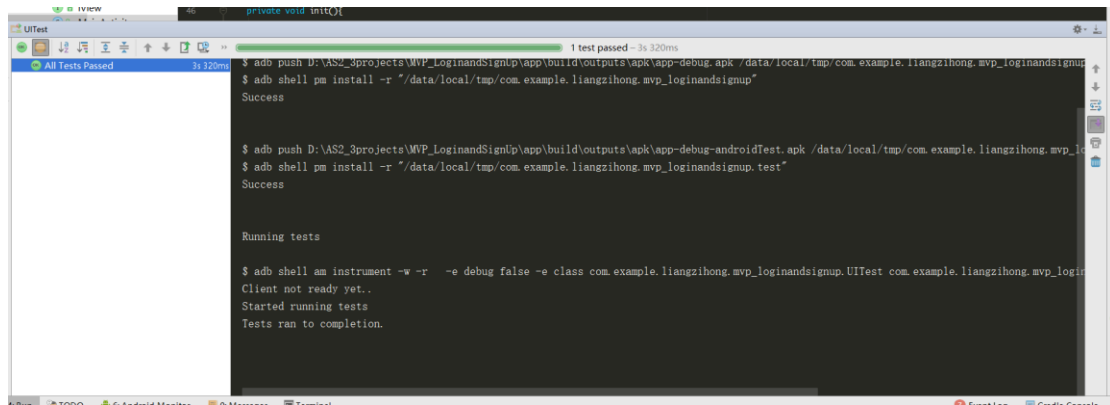


## 6、下拉用户提示





## 6、UI 单元测试成功提示



## 四、代码设计

主界面设计 cal\_activity.xml

```
package view;  
  
import android.content.Context;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.support.v7.widget.ListPopupWindow;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.AdapterView.OnItemClickListener;  
import android.widget.ArrayAdapter;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.example.liangzihong.mvp_loginandsignup.R;  
  
import org.w3c.dom.Text;  
  
import java.util.List;  
  
import presenter.ILoginPresenter;  
import presenter.ISignupPresenter;  
import presenter.LoginPresenter;  
import presenter.SignupPresenter;
```

```

public class MainActivity extends AppCompatActivity implements
    IView, View.OnClickListener, View.OnFocusChangeListener{

    private ILoginPresenter loginPresenter;
    private ISignupPresenter signupPresenter;

    private EditText inputname;
    private EditText inputpassword;
    private Button login_button;
    private Button signup_button;
    private Button exit_button;
    private TextView count_text;
    private int remain_count=3;

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

    private void init(){
        loginPresenter=new LoginPresenter(this);
        signupPresenter=new SignupPresenter(this);
        inputname=(EditText)findViewById(R.id.input_name);
        inputpassword=(EditText)findViewById(R.id.input_password);
        login_button=(Button)findViewById(R.id.login_button);
        signup_button=(Button)findViewById(R.id.signup_button);
        exit_button = (Button)findViewById(R.id.exit_button);
        count_text = (TextView)findViewById(R.id.remain_count_hint);

        login_button.setOnClickListener(this);
        signup_button.setOnClickListener(this);
        exit_button.setOnClickListener(this);
        inputname.setOnFocusChangeListener(this);

        count_text.setText("剩余尝试次数为:"+Integer.toString(remain_count));
    }

    // 设置下拉用户名出现的名字的响应函数
    private void showListPopulWindow() {
        List name_list = loginPresenter.getAllName();
    }

```

```

        if (name_list.size()==0) {
            return;
        }
        final String[] list = (String[]) name_list.toArray(new
String[name_list.size()]); //要填充的数据
        final ListPopupWindow listPopupWindow;
        listPopupWindow = new ListPopupWindow(this);
        listPopupWindow.setAdapter(new
ArrayAdapter<>(this, android.R.layout.simple_list_item_1, list)); //用 android 内置布
局，或设计自己的样式
        listPopupWindow.setAnchorView(inputname); //以哪个控件为基准，在该处以 mEditText
为基准
        listPopupWindow.setModal(true);

        listPopupWindow.setOnItemClickListener(new
AdapterView.OnItemClickListener() { //设置项点击监听
            @Override
            public void onItemClick(AdapterView<?> adapterView, View view, int i,
long l) {
                inputname.setText(list[i]); //把选择的选项内容展示在 EditText 上
                listPopupWindow.dismiss(); //如果已经选择了，隐藏起来
            }
        });
        listPopupWindow.show(); //把 ListPopWindow 展示出来
    }

    // 点击事件的 override
    @Override
    public void onClick(View v) {
        String name=inputname.getText()+" ";
        String password=inputpassword.getText()+" ";

        switch(v.getId()){
            case R.id.login_button:
                loginPresenter.startLogin(name,password);
                break;
            case R.id.signup_button:
                signupPresenter.startSignup(name,password);
                break;
            case R.id.exit_button:
                System.exit(0);
        }
    }

```

```
}

// editText 下拉的点击事件
@Override
public void onFocusChange(View v, boolean hasFocus) {
    if (hasFocus) {
        showListPopulWindow(); //调用显示 PopuWindow 函数
    }
}
```

//下面的方法 就是 实现 IView 的方法，是给 IPresenter 的实例运用的  
//所以上面的方法是真实 的 Activity 的代码，所以 Activity 是简洁的  
//而如果要再加功能， Activity 的代码其实也增加的不多，只是增加 Presenter 的代码而已。

```
@Override
public Context getContext() {
    return this;
}

@Override
public void successLogin() {
    Toast.makeText(this, "登录成功", Toast.LENGTH_SHORT).show();
    remain_count = 3;
    count_text.setText("剩余尝试次数为:"+Integer.toString(remain_count));
}

@Override
```

```

public void failLogin() {
    Toast.makeText(this, "登录失败，请核对账号密码", Toast.LENGTH_SHORT).show();
    remain_count --;
    Toast.makeText(this, "剩余"+Integer.toString(remain_count)+"次尝试次数",
Toast.LENGTH_SHORT).show();
    count_text.setText("剩余尝试次数为:"+Integer.toString(remain_count));
    if (remain_count == 0) {
        login_button.setClickable(false);
    }
}

@Override
public void successSignup() {
    Toast.makeText(this, "注册成功，请登录", Toast.LENGTH_SHORT).show();
    inputname.setText(null);
    inputpassword.setText(null);
}

@Override
public void failSignup(){
    Toast.makeText(this, "注册失败，账号已注册或账号密码为空",
Toast.LENGTH_SHORT).show();
}

}

```

## activity 的接口设计    IView.java

```

package view;

import android.content.Context;

/**
 * Created by Liang Zihong on 2018/3/2.

```

```
*/  
  
public interface IView {  
    public void successLogin();  
    public void failLogin();  
  
    public void successSignup();  
    public void failSignup();  
  
    public Context getContext();  
}
```

## 模型的接口设计 IUser\_model.java

```
package model;  
  
import java.util.List;  
  
/**  
 * Created by Liang Zihong on 2018/3/2.  
 */  
  
public interface IUser_model {  
  
    public boolean isSignupSuccess(String name, String password);  
  
    public boolean isLoginSuccess(String name, String password);  
  
    public List getAllName();  
}
```

## model 设计      User\_model.java

```
package model;

import android.content.Context;
import android.content.SharedPreferences;
import android.content.SharedPreferences.Editor;

import java.util.ArrayList;
import java.util.List;
import java.util.Map;

import presenter.ILoginPresenter;
import presenter.ISignupPresenter;

/**
 * Created by Liang Zihong on 2018/3/2.
 */

public class User_model implements IUser_model {

    private SharedPreferences pref;
    private Editor editor;

    public User_model(Context context){
        pref=context.getSharedPreferences("StoreUser", Context.MODE_PRIVATE);
        editor=pref.edit();
    }

    /**
     * 注册成功返回 true, 不成功返回 false
     * @param name
     * @param password
     * @return 注册是否成功
     */
    private boolean startSignup(String name, String password) {
        if(name==null || password==null){
            return false;
        }
        else if(pref.getString(name,null)!=null){

```

```

        return false;
    }
    else{
        editor.putString(name,password);
        editor.commit();
        return true;
    }
}

/**
 * 登录成功返回 true, 不成功 返回 false
 * @param name
 * @param password
 * @return
 */
private boolean startLogin(String name, String password) {
    if(pref.getString(name,"").equals(password)){
        return true;
    }
    else{
        return false;
    }
}

/**
 * 遍历 文件之间存储的姓名, 然后返回出来
 * @return ArrayList<String>
 */
@Override
public List getAllName() {
    ArrayList<String> name_list = new ArrayList<>();
    Map<String, ?> all_Content = pref.getAll();
    for(Map.Entry<String, ?> entry : all_Content.entrySet()) {
        String name = entry.getKey();
        name_list.add(name);
    }
    return name_list;
}

@Override
public boolean isSignupSuccess(String name,String password) {
    return startSignup(name,password);
}

```



```
@Override
public boolean isLoginSuccess(String name,String password) {
    return startLogin(name,password);
}
}
```

登陆 presenter 接口代码

ILoginPresenter.java

```
package presenter;

import java.util.List;

/**
 * Created by Liang Zihong on 2018/3/2.
 */

public interface ILoginPresenter {
    public void startLogin(String name,String password);

    public List getAllName();
}
```

登陆 presenter 代码

LoginPresenter.java

```
package presenter;
```

```
import java.util.List;

import model.IUser_model;
import model.User_model;
import view.IView;

/**
 * Created by Liang Zihong on 2018/3/2.
 */

public class LoginPresenter implements ILoginPresenter {
    private IView view;
    private IUser_model user;

    //view 的操作都交给了 Presenter 来调用 。

    public LoginPresenter(IView view) {
        this.view=view;
        user=new User_model(view.getContext());
    }

    @Override
    public void startLogin(String name, String password) {
        if(user.isLoginSuccess(name,password))
            successLogin();
        else
            failLogin();
    }

    @Override
    public List getAllName() {
        return user.getAllName();
    }

    private void successLogin() {
        view.successLogin();
    }

    private void failLogin() {
        view.failLogin();
    }
}
```

```
}  
}
```

## UI 单元测试代码

## UITest.java

```
package com.example.liangzihong.mvp_loginandsignup;  
  
/**  
 * Created by Liang Zihong on 2018/11/17.  
 */  
  
import android.support.test.InstrumentationRegistry;  
import android.support.test.espresso.Espresso;  
import android.support.test.espresso.action.ViewActions;  
import android.support.test.rule.ActivityTestRule;  
import android.support.test.runner.AndroidJUnit4;  
import android.test.ActivityInstrumentationTestCase2;  
import android.test.suitebuilder.annotation.LargeTest;  
import android.view.View;  
import android.widget.TextView;  
  
import org.junit.After;  
import org.junit.Before;  
import org.junit.Rule;  
import org.junit.Test;  
import org.junit.runner.RunWith;  
  
import view.MainActivity;  
  
import static android.support.test.espresso.Espresso.onData;  
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.action.ViewActions.pressBack;
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;
import com.example.liangzihong.mvp_loginandsignup.R;

@RunWith(AndroidJUnit4.class)
@LargeTest
public class UITest {

    private static final String NAME = "lzh";
    private static final String PASSWORD = "0";

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

    @Test
    public void sayHello(){
        pressBack();

        onView(withId(R.id.input_name)).perform(click(),typeText(NAME),closeSoftKeyboard(
        )); //line 1 输入用户名

        onView(withId(R.id.input_password)).perform(click(),typeText(PASSWORD),closeSoftK
        eyboard()); //line 2 输入密码
        onView(withId(R.id.login_button)).perform(click()); //line3 点击登陆按钮

        String expectedText = "剩余尝试次数为:2";

        onView(withId(R.id.remain_count_hint)).check(matches(withText(expectedText)));
        //line 3
    }
}

```

界面设计 xml 代码

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

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/background_color">

    <TextView
        android:id="@+id/name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="25dp"
        android:layout_marginStart="25dp"
        android:layout_marginTop="90dp"
        android:text="@string/user_text"
        android:textSize="20sp"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <TextView
        android:id="@+id/password"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/name"
        android:layout_alignStart="@+id/name"
        android:layout_below="@+id/name"
        android:layout_marginTop="47dp"
        android:text="@string/password_text"
        android:textSize="20sp"
        android:layout_alignRight="@+id/name"
        android:layout_alignEnd="@+id/name" />

    <EditText
        android:id="@+id/input_name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPersonName"
        android:hint="@string/input_name"
        android:layout_alignBaseline="@+id/name"
        android:layout_alignBottom="@+id/name"
        android:layout_toRightOf="@+id/name"
        android:layout_marginLeft="22dp"
```

```
    android:layout_marginStart="22dp"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true" />
```

#### <EditText

```
    android:id="@+id/input_password"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/input_name"
    android:layout_alignStart="@+id/input_name"
    android:layout_alignTop="@+id/password"
    android:ems="10"
    android:textSize="20sp"
    android:inputType="textPassword"
    android:hint="@string/input_password"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true" />
```

#### <Button

```
    android:id="@+id/login_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="54dp"
    android:background="#caf0b2ef"
    android:text="@string/login"
    android:layout_below="@+id/password"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_marginLeft="12dp"
    android:layout_marginStart="12dp" />
```

#### <Button

```
    android:id="@+id/signup_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="#caf0b2ef"
    android:text="@string/signup"
    android:layout_marginLeft="15dp"
    android:layout_marginStart="15dp"
    android:layout_alignBaseline="@+id/login_button"
    android:layout_alignBottom="@+id/login_button"
    android:layout_toRightOf="@+id/login_button"
    android:layout_toEndOf="@+id/login_button" />
```

```

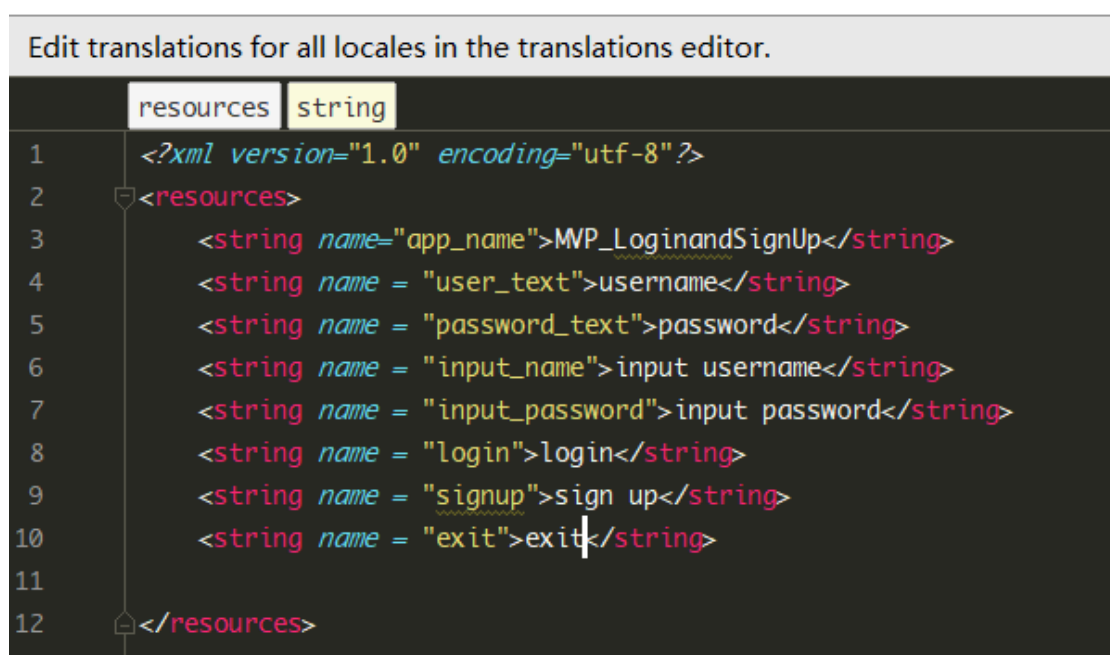
<TextView
    android:id="@+id/remain_count_hint"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/login_button"
    android:layout_alignStart="@+id/login_button"
    android:layout_below="@+id/login_button"
    android:layout_marginTop="36dp"
    android:textSize="16sp" />

<Button
    android:id="@+id/exit_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="#caf0b2ef"
    android:text="@string/exit"
    android:layout_marginLeft="15dp"
    android:layout_marginStart="15dp"
    android:layout_alignBaseline="@+id/signup_button"
    android:layout_alignBottom="@+id/signup_button"
    android:layout_toRightOf="@+id/signup_button"
    android:layout_toEndOf="@+id/signup_button" />

</RelativeLayout>

```

## 全球化文字设计



Edit translations for all locales in the translations editor.

```
resources string
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3     <string name="app_name">MVP_LoginandSignUp</string>
4     <string name="user_text">用户名</string>
5     <string name="password_text">密码</string>
6     <string name="input_name">请输入用户名</string>
7     <string name="input_password">请输入密码</string>
8     <string name="login">登陆</string>
9     <string name="signup">注册</string>
10    <string name="exit">退出</string>
11
12 </resources>
```

注册 Presenter 接口代码 ISignupPresenter.java

```
package presenter;

/**
 * Created by Liang Zihong on 2018/3/2.
 */

public interface ISignupPresenter {
    public void startSignup(String name, String password);
}
```

注册 Presenter 代码 ISignupPresenter.java

```
package presenter;

import android.content.SharedPreferences;
import android.content.SharedPreferences.Editor;

import model.IUser_model;
```



```
import model.User_model;
import view.IView;

/**
 * Created by Liang Zihong on 2018/3/2.
 */

public class SignupPresenter implements ISignupPresenter {
    private IView view;
    private IUser_model user;

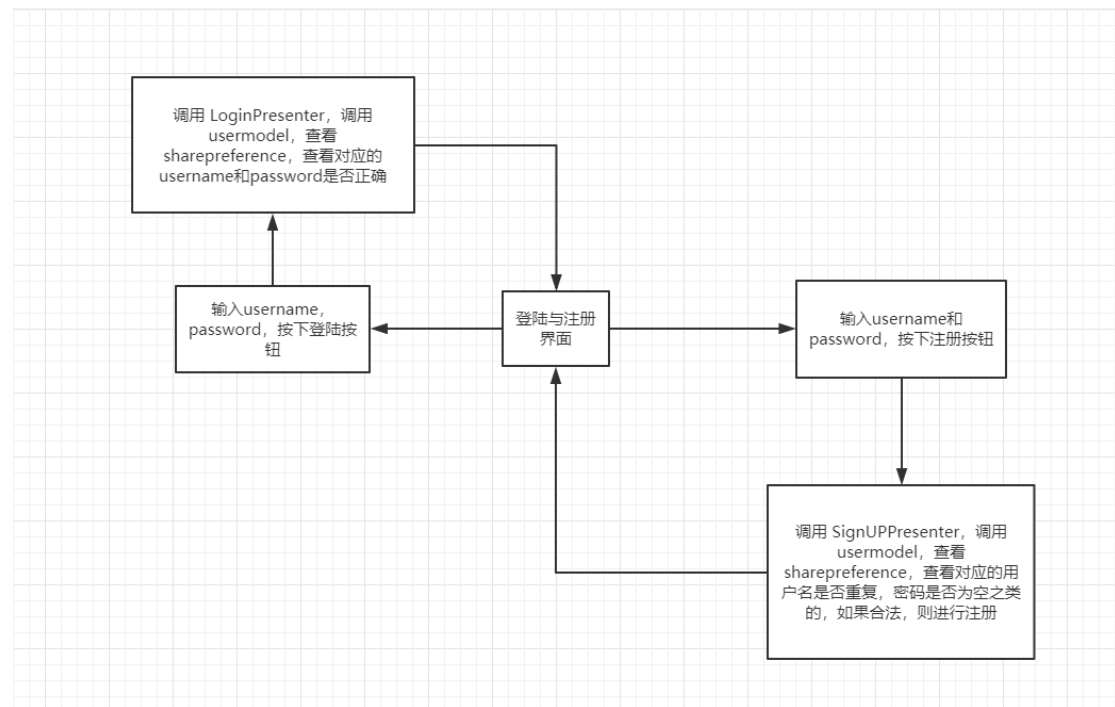
    public SignupPresenter(IView view){
        this.view=view;
        user=new User_model(view.getContext());
    }

    @Override
    public void startSignup(String name, String password) {
        if(user.isSignupSuccess(name,password)) {
            successSignup();
        }
        else{
            failSignup();
        }
    }

    private void successSignup() {
        view.successSignup();
    }

    private void failSignup() {
        view.failSignup();
    }
}
```

## 五、计算器流程图



## 六、程序的不足之处

1、没有考虑到记住密码功能，因为觉得其实也并不难实现，但是由于老师限时两小时，于是没有进一步扩充

2、在进行 UI 单元测试的时候，发现一个难以解决的 bug，当程序中出现类似下拉列表 `spinner`，`listview`，`adaperview` 之类的动态出现的控件时，UI 单元测试容易出现 `error`，并且难以解决，所以我解

决的方法仅仅是把 要出现的 `listpopwindow` 暂时注释掉来保证 UI 单元测试的顺利完成。

3、没有设置一个 `activity` 来实现跳转，显示登陆成功

4、利用 MVP 模式尽管会使代码变得繁琐，但是却把代码功能化，当以后进行添加和扩充功能时，只需添加 `presenter` 即可，并且保留了 `activity` 的简洁，使代码更容易理解。

## 七、难点和解决办法

1、这个程序难点在于 UI 单元测试的时候，动态出现的控件貌似解决不了 bug，所以我只能在进行测试时把下拉这个功能暂时注释掉。

2、这次程序尝试不用 `linearlayout` 来完成，转而在用 `relativelayout` 来完成，其实熟练了 `linearlayout`，`relativelayout` 效果反而不如 `linearlayout`，每个控件都受许多约束，难以操控。

## 八、今后对安卓开发的设想

我们编写代码，不仅是为了眼前的功能或者眼前的代码，我们需要为将来要添加的功能去做铺设，也为了使以后的自己能看懂代码，所以用到一定的设计模式十分有必要，不仅要把功能做出来，更要把代码写得好看，扩展性强，易于添加代码，易于维护，才是写代码更

高的目标。

最后，再次证实了 cs 的真理：纸上得来终觉浅，绝知此事要躬行。