



系统登录/注册模块(Android app)的开发

编写:	宋晨晨、余琴
组长:	李汉俊
组员:	宋晨晨、余琴
发布版本:	5.0
日期:	2018-12-18

变更记录

日期	版本	变更说明	作者
2018 年 12 月 18 日	1.0	初始版本	李汉俊、余琴、宋晨晨
2018 年 12 月 20 日	5.0	最终版本	李汉俊、余琴、宋晨晨

任务分工

模块	对应组员
后台处理	李汉俊
登录页面	余琴
注册页面	宋晨晨
欢迎界面	李汉俊
显示界面	李汉俊

目 录

1	引言.....	4
1.1	编写目的.....	4
1.2	背景.....	4
1.3	项目介绍.....	4
1.4	基线.....	5
1.5	范围.....	5
1.6	定义.....	5
1.7	参考资料.....	5
1.8	术语与缩写解释.....	6
2	模块命名规则.....	7
3	模块汇总.....	7
3.1	模块汇总表.....	7
3.2	模块关系图.....	7
4	子系统模块设计.....	8
4.1	模块 1.....	8
4.2	模块 2.....	11
4.3	模块 3.....	13
4.4	模块 4.....	14
4.5	模块 5.....	17
5	程序运行流程图.....	19
6	实验心得	22
7	附源代码.....	23

1 引言

1.1 实验目的

进一步理解 Java、SDK、ADT、Eclipse 的彼此依赖关系,并且能熟练使用 java 语言来编写 Android 工程。掌握 Android 应用开发环境的搭建方法以及虚拟机的配制方法,掌握 Android 工程创建方法,掌握基于虚拟机与真机的 Android 工程运行方法,了解 Activity 生命周期,理解 Activity 事件回调, onRestoreInstanceState()和 onSaveInstanceState 两个函数的使用。在此基础上,掌握 Android 客户端与服务器通信的原理并且运用到项目中,理解 Android 发送 http 网络请求,包括 GET 请求和 POST 请求,熟悉 Android 异步任务的处理方法,包括各种回调函数。能够运用 Servlet 和数据库来实现一个登录界面。最后用 json 实现 android 客户端与服务端的通讯。

1.2 实验背景

- a) 软件系统的名称: 系统登录/注册模块(Android app)的开发
- b) 项目负责人: 李汉俊
- c) 程序员: 李汉俊、余琴、宋晨晨
- d) 基线

Login.java Myinfo.java Mypage.java Sign.java BuildConfig.java
R.java activity_login.xml activity_my_info.xml activity_my_page.xml
activity_sign.xml

1.3 项目介绍

- a) app 名称: 知遇

b) 简介：现今社会年轻人的学习、生活压力非常大，很多人会因为失恋、工作不顺心等事导致心情低落、阴郁，所以我们小组在这次实验中决定开发“知遇”这款情感咨询 app，致力于为有情感或心理问题的人们提供一个平台，提供情感咨询等服务，帮助人们摆脱烦恼，收获快乐。

c) 实现：我们小组根据老师提供的实验四的实验文档，成功实现了“知遇”注册、登录的功能，我想后续我们还可以实现类似 QQ 里的聊天功能。

1.4 范围

系统包括的范围：安卓 2.2 以上的所有版本均可使用。

1.5 定义

本系统实现的功能系统登录/注册模块，系统包括的范围：安卓 2.2 以上的所有版本均可使用

1.6 参考资料

参考资料如下：

- a) <http://www.2cto.com/kf/201203/122729.html>
- b) http://blog.csdn.net/zh_qianwei/article/details/50717736
- c) <http://my.oschina.net/dlpinghailinfeng/blog/52564>

1.7 术语与缩写解释

缩写、术语	解 释
SPP	精简并行过程, Simplified Parallel Process
SD	系统设计, System Design
JDK	Java 语言的软件开发工具包, Java Development Kit
Eclipse	一个开放源代码的、基于 Java 的可扩展开发平台
SDK	软件开发工具包, Software Development Kit
ADT	安卓开发工具, Android Development Tools
HTTP	超文本传送协议 ,Hypertext transfer protocol
JSON	一种轻量级的数据交换格式, JavaScript Object Notation

2 模块命名规则

应用程序名：知遇

工程文件名：cureU

实验名：Android 系统登录/注册模块

	类名	函数名	
后台	SignIn	doGet	doPost
	SignUp	doGet	doPost
前端	activity_login	onCreate	onCreateOptionsMenu
		sign	doInBackground
		showerr	SignInSuccess
		cancel	
	activity_my_info	onCreate	onCreateOptionsMenu
		cancel	
	activity_my_page	onCreate	onCreateOptionsMenu
		showInfo	cancel
	activity_sign	onCreate	onCreateOptionsMenu
		sign	doInBackground
		onPostExecute	showerr
		SignInSuccess	cancel

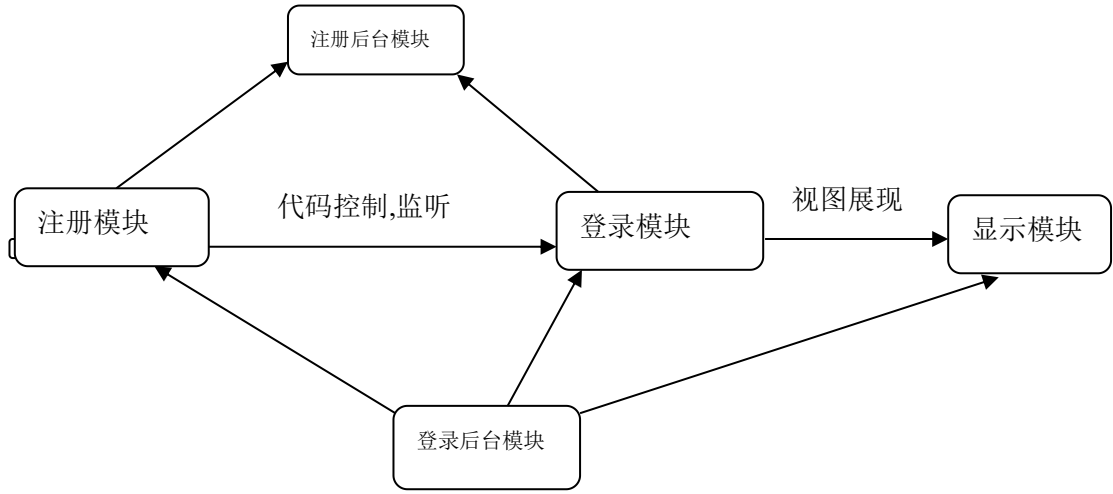
3 模块汇总

3.1 模块汇总表

模块名称	功能简述
登录模块	实现已注册用户登录，或为未注册用户转至注册界面
注册模块	实现未注册用户的账户注册服务
显示模块	显示登录\注册成功，且可查看用户信息
登录后台模块	连接后台数据库，对用户输入信息进行核实，判断能否成功登录
注册后台模块	判断用户输入信息是否满足注册条件，且将正确信息录入后台数据库

3.2 模块关系图

对于 Android http 通信,包含两个系统,分别是客户端系统和服务器系统，客户端系统包括视图模块和代码功能实现模块和模型模块，服务器系统包括控制模块。



4 子系统模块设计

4.1 模块 1

模块名称	登录模块
功能描述	实现已注册用户登录，或为未注册用户转至注册界面
接口与属性	所有用户界面元素都是由 View 和 ViewGroup 对象创建的。View 是一种可以在屏幕上绘制某种画面并且可以与用户互动的对象。ViewGroup 对象则是为了定义布局的接口而保存其他 View (和 ViewGroup)对象。Android 提供一个 View 和 ViewGroup 子类的集合，这个集合能提供相同的输入控制（例如按钮和文本框）和各种各样的布局模式（例如一个线性或者相对布局）。
数据结构与算法	XML 文件可以为布局提供一个可读结构，一个 View 的 XML 节点名称与它代表的 Android 类相对应。在布局中,使用了相对布局，线性布局,线性布局嵌套在相对布局中。

activity_login.xml

线性布局 LinearLayout 是按照垂直或者水平的顺序依次排列子元素(如果不指定, 默认为水平)。线性布局方向设为垂直, 让输入框和登录按钮垂直显示。Width 和 Height 均为 wrap_content, gravity 为 center。线性布局下, 嵌套 Table 布局, 设置 6 个 Table 行。第一个 Table 行显示 "LOGIN"; 第二行分两部分显示, 前半部分为 TextView, 显示 "username", 后半部分为 EditText, 需要编辑输入用户名; 第三行同第二行, 只是 TextView 显示 "password", EditText 需要编辑输入密码; 第四行设置按钮 Button, 显示信息为 "没有账号, 注册"; 第五行也设置按钮 Button, 显示信息为 "登录"; 第六行为 TextView, 在输入有误时才会显示该行信息 "用户名或密码为空/用户不存在/密码错误"。

用户名输入框中, 设置 id 为 login_username, layout_width 为 wrap_parent, layout_height 为 wrap_content, layout_marginTop 为 30dp, ems 为 10, text 为 username。对于密码输入框, 设置基本相同但是 id 为 login_password, text 为 password。

设置注册按钮, 设置 id 为 bt_cancel, layout_width 为 wrap_parent, layout_height 为 wrap_content, layout_gravity 为 center, layout_marginTop 为 5dp, background 为 color/transparent, text 为 "没有账号, 注册", textSize 为 15dp。

设置登录按钮, 设置 id 为 bt_sign, layout_width 为 wrap_parent, layout_height 为 wrap_content, layout_marginTop 为 30dp, layout_gravity 为 center 按钮上的文字为"登 录"。

	<p>对于 button 按键的样式, 在 bg_alibuybutton.xml 中设置, 为了凸显按钮按下的不同, 在 item 标签下 android:state_pressed="true" 和 android:state_focused="true" 来凸显不同, 对于未按下的按钮, 在 shape.xml 中进行描述, 设置其颜色和透明度。</p>
补充说明	<p>android:layout_below, 意思是该组件位于引用组件的下方, 而引用的组件就是这个属性值里面的内容 "@id/要引用的 id 名" ;</p> <p>android:layout_alignParentTop 是否对齐父组件的顶部;</p> <p>android:layout_weight 属性限定在水平布局时, 不同的控件占的宽度比率, 具体规则为: 如果水平布局有两个控件, 其 android:layout_weight 属性值分别为 n 和 m, 则属性值为 n 的控件所占的长度比例为总长的 $n/(n+m)$, 属性值为 m 的控件所占的长度比例为 $m/(n+m)$。属性值越大, 占的份额越多;</p> <p>fill_parent、wrap_content 和 match_parent 三个属性都用来适应视图的水平或垂直大小: fill_parent 设置一个顶部布局或控件为 fill_parent 将强制性让它布满整个屏幕。wrap_content 设置一个视图的尺寸为 wrap_content 将强制性地使视图扩展以显示全部内容。</p> <p>match_parent 在 Android2.2 中 match_parent 和 fill_parent 是一个意思;</p> <p>android:layout_alignParentLeft 贴紧父元素的左边缘</p> <p>android:layout_alignParentRight 贴紧父元素的右边缘</p> <p>android:layout_alignTop 本元素的上边缘和某元素的的上边缘对齐</p> <p>android:layout_alignLeft 本元素的左边缘和某元素的的</p>

	<p>左边缘对齐</p> <p>layout_gravity 是用来设置该 view 相对与起父 view 的位置</p> <p>在 Android 上一般不使用绝对布局。</p>
--	--

4.2 模块 2

模块名称	注册模块
功能描述	实现未注册用户的账户注册功能
接口与属性	<p>所有用户界面元素都是由 View 和 ViewGroup 对象创建的。View 是一种可以在屏幕上绘制某种画面并且可以与用户互动的对象。ViewGroup 对象则是为了定义布局的接口而保存其他 View（和 ViewGroup)对象。</p> <p>一个 activity 的用户接口被一个层次化的视图提供 - - 继承于 View 类的对象。每个 View 控制 activity 窗口中的一个特定矩形区域并且能响应用户交互。例如，一个 view 可能是个 button，初始化动作当用户触摸它的时候。</p> <p>Android 提供大量预定义的 view 来设计和组件你的布局。</p> <p>“Widgets”是一种给屏幕提供可视化(并且交互)元素的 view，例如按钮、文件域、复选框或者仅仅是图像。“Layouts”是继承于 ViewGroup 的 View，提供特殊的布局模型为它的子 view。</p> <p>最普通的方法是定义一个布局使用 view 加上 XML 布局文件保存在程序资源里。</p>

<p>数据结构 与算法</p>	<p>XML 文件可以为布局提供一个可读结构, 一个 View 的 XML 节点名称与它代表的 Android 类相对应。在布局中,使用了相对布局, 线性布局, 线性布局嵌套在相对布局中。</p> <p>Activity 是一个应用中的组件, 它为用户提供一个可视的界面, 方便用户操作。在 MainActivity.java 的代码编辑界面, 添加一个类型为 EditText 的全局变量 et 和 static final int PICK_CONTACT_RESULT 的常量, 并在 onCreate 方法中添加变量 et 的初始化;</p> <p>activity_Sign.xml</p> <p>线性布局 LinearLayout 是按照垂直或者水平的顺序依次排列子元素(如果不指定, 默认为水平)。线性布局方向设为垂直, 让输入框和注册按钮垂直显示。Width 和 Height 均为 wrap_content, gravity 为 center。线性布局下,嵌套 Table 布局,设置 8 个 Table 行。第一个 Table 行显示“SIGN INFO”; 第二行分两部分显示, 前半部分为 TextView, 显示“用户名”, 后半部分为 EditText, 需要编辑输入用户名; 第三行分两部分显示, 前半部分为 TextView, 显示“姓名”, 后半部分为 EditText, 需要编辑输入姓名;</p> <p>第四行分两部分显示, 前半部分为 TextView, 显示“密码”, 后半部分为 EditText, 需要编辑输入密码; 第五行分两部分显示, 前半部分为 TextView, 显示“确认密码”, 后半部分为 EditText, 需要编辑输入确认密码; 第六行分两部分显示, 前半部分为 TextView, 显示“年龄”, 后半部分为 EditText, 需要编辑输入年龄; 第七行分两部分显示, 前半部分为 TextView, 显示“手机号码”, 后半部分为 EditText, 需要编辑</p>
---------------------	---

	<p>输入手机号码; 第八行设置两个按钮 Button, 显示信息为 “注册” 和 “取消” ;</p> <p>用户名输入框中, 设置 id 为 login_username, layout_width 为 wrap_parent, layout_height 为 wrap_content, layout_marginTop 为 30dp, ems 为 10, text 为 username。</p> <p>姓名、密码、确认密码、年龄、手机号码栏设置均和用户名输入框一致。</p> <p>设置注册按钮, 设置 id 为 bt_cancel, layout_width 为 wrap_parent, layout_height 为 wrap_content, layout_gravity 为 center, layout_marginTop 为 5dp, background 为 color/transparent, text 为 “注册” 和 “取消”, textSize 为 15dp。</p> <p>对于 button 按键的样式, 在 bg_alibuybutton.xml 中设置, 为了凸显按钮按下的不同, 在 item 标签下 android:state_pressed="true" 和 android:state_focused="true" 来凸显不同, 对于未按下的按钮, 在 shape.xml 中进行描述, 设置其颜色和透明度。</p>
--	--

4.3 模块 3

模块名称	显示模块
功能描述	显示登录\注册成功, 且可查看用户信息
接口与属性	<p>ViewGroup 对象则是为了定义布局的接口而保存其他 View (和 ViewGroup)对象。</p> <p>一个 activity 的用户接口被一个层次化的视图提供 - - 继承于 View 类的对象。每个 View 控制 activity 窗口中的一个特定矩形区域并且能响应</p>

	<p>用户交互。例如，一个 view 可能是个 button，初始化动作当用户触摸它的时候。</p> <p>Android 提供大量预定义的 view 来设计和组件你的布局。</p> <p>“Widgets” 是一种给屏幕提供可视化(并且交互)元素的 view，例如按钮、文件域、复选框或者仅仅是图像。“Layouts” 是继承于 ViewGroup 的 View，提供特殊的布局模型为它的子 view。</p>
数据结构与算法	<p>activity_my_page.xml</p> <p>线性布局 LinearLayout 是按照垂直或者水平的顺序依次排列子元素(如果不指定，默认为水平)。线性布局方向设为垂直，让输入框和注册按钮垂直显示。Width 和 Height 均为 wrap_content，gravity 为 center。线性布局下，嵌套 Table 布局，设置四个 Table 行。第一个 Table 行显示 “myinfo”；第二行显示 “welcome to here”，第三行显示 “hello”，第四行为 TextView，显示 “退出登录”，</p> <p>第一个显示框，设置 id 为 login_username, layout_width 为 wrap_parent，layout_height 为 wrap_content，layout_marginTop 为 30dp，ems 为 10，text 为 username。</p> <p>第二、三个显示框跟第一个设置完全一致</p>

4.4 模块 4

模块名称	登录后台模块
功能描述	连接后台数据库，对用户输入信息进行核实，判断能否成功登录

接口与属性	<p>所有 Servlet 应用必须直接或者间接实现 Servlet 接口, Servlet 容器会将实现了 Servlet 接口的类加载至容器, 以供访问。</p> <p>ServletRequest 接口主要用于获取请求中的参数。HttpServletRequest 接口在 ServletRequest 接口的基础上增加了多种方法。</p> <p>调用一个 Servlet 的 service()方法之前, Servlet 会先创建一个 ServletRequest 与 ServletResponse, 并将它们作为参数传给 service 方法, 它们隐藏了将请求发给 servlet 以及响应发给浏览器的复杂性。HttpServletResponse 也增加了许多方法。</p> <p>每种数据库的驱动程序都应该提供一个实现 java.sql.Driver 接口的类, 简称 Driver 类。</p> <p>java.sql.DriverManager 类负责管理 JDBC 驱动程序的基本服务, 是 JDBC 的管理层, 作用于用户和驱动程序之间, 负责跟踪可用的驱动程序, 并在数据和驱动程序之间建立连接。java.sql.Connection 接口代表与特定数据库的连接, 在接连的上下文中可以执行 SQL 语句并返回结果, 还可以通过 getMetaData() 方法获得由数据库提供的相关信息。</p> <p>java.sql.Statement 接口用来执行静态 SQL 语句, 并返回执行结果。</p> <p>java.sql.PreparedStatement 接口继承并扩展了 Statement 接口, 用来执行动态的 SQL 语句, 即包含参数的 SQL 语句。java.sql.ResultSet 接口类似于一个数据表, 通过该接口的实例可以获得检索结果集, 以及对应数据表的相关信息, ResultSet 实例通过执行查询数据库的语句生成。</p>
数据结构与算法	<p>登录后台模块主要简单抽象出了数据访问层, 并实现数据库的操作。</p>

	<p>Login.java</p> <p>实现基础的连接 JDBC 数据库，确定数据库服务器 IP，数据库用户名，数据库密码等等。然后用 java.sql.DriverManager 的 getConnection 方法实现连接。执行 SQL 语句，从数据库读取数据，更新结果集。通过封装属性和方法成为具有某种功能或者处理某个业务的对象。</p> <p>在 doPost 方法中，设置数据的发送格式为 JSON，读入请求数据，在控制台打印请求内容（为 JSON 格式），然后用传过来的 JSON 格式的请求数据作为参数创建 JSON 数组。然后通过用户名创建对象。判断密码是否符合数据库中的内容，如果符合则将数据库中其他类型的数据封装进 JSON 数组中作为响应的内容发送回客户端。</p> <p>具体操作时，若输入的用户名或密码为空，则会显示“用户名或密码为空”。若不为空，则显示“正在登录”。此时访问后台数据库，通过访问已连接的数据库，获取后台结果，找到该用户名对应的密码。若与输入密码相同，则登录成功，否则登录失败。若该用户名不存在数据库中，则需要先注册才能登录，此时跳转至注册模块，它们连接同一个数据库，所以注册时输入的所有信息保存在数据库后，下次可直接访问。</p>
补充说明	<p>JSON 有两种表示结构，对象和数组。对象结构以“{”大括号开始，以“}”大括号结束。中间部分由 0 或多个以“,”分隔的“key(关键字)/value(值)”对构成，关键字和值之间以“:”分隔；数组结构以“[”开始，“]”结束。中间由 0 或多个以“,”分隔的值列表组成。</p> <p>JSONObject 可以看作是一个 json 对象,这是系统中有关 JSON 定义的基本单元，其包含一对儿(Key/Value)数值。JSONArray 它代表一组有序的</p>

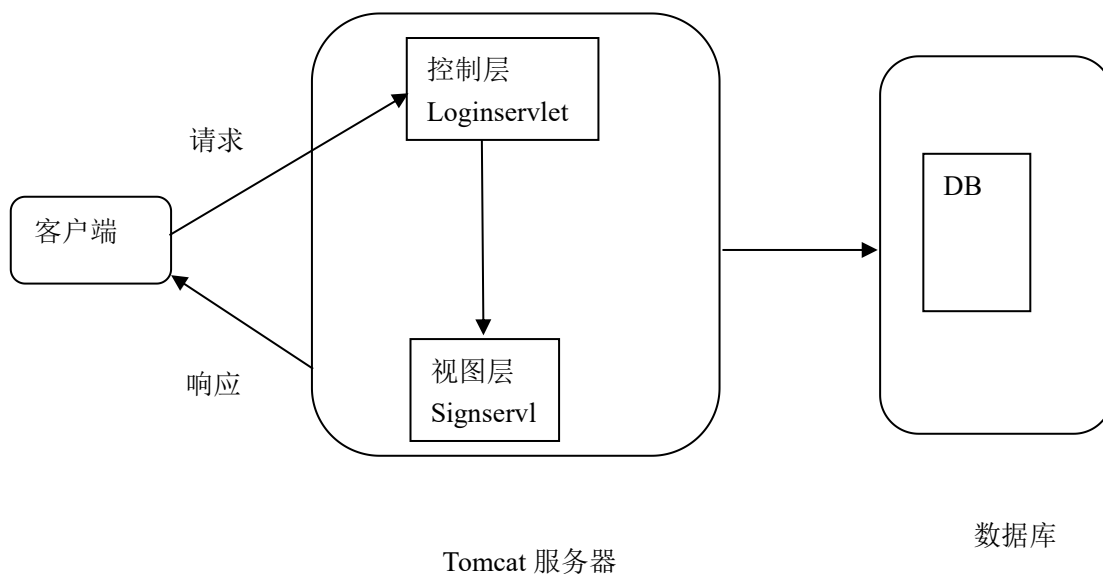
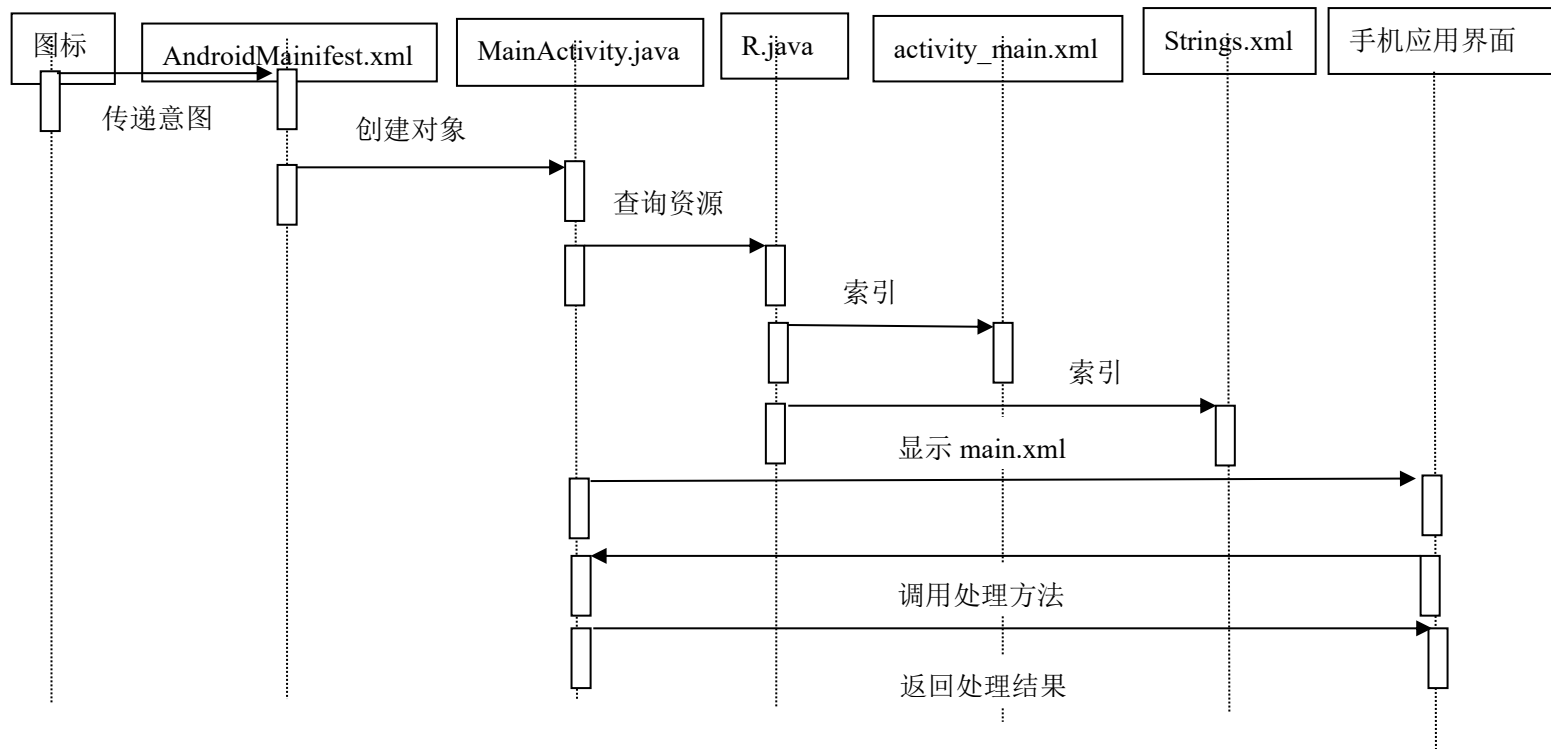
	<p>数值。将其转换为 String 输出(toString)所表现的形式是用方括号包裹, 数值以逗号“,” 分隔 (例如[value1,value2,value3])</p> <p>在 web.xml 中需要配置。</p>
--	--

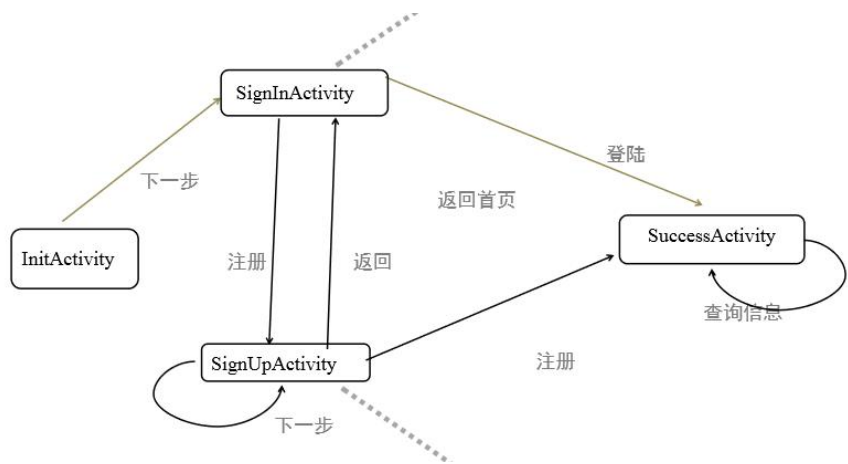
4.5 模块 5

模块名称	注册后台模块
功能描述	判断用户输入信息是否满足注册条件, 且将正确信息录入后台数据库
接口与属性	<p>每种数据库的驱动程序都应该提供一个实现 java.sql.Driver 接口的类, 简称 Driver 类。</p> <p>所有 Servlet 应用必须直接或者间接实现 Servlet 接口, Servlet 容器会将实现了 Servlet 接口的类加载至容器, 以供访问。</p> <p>ServletRequest 接口主要用于获取请求中的参数。HttpServletRequest 接口在 ServletRequest 接口的基础上增加了多种方法。</p> <p>调用一个 Servlet 的 service() 方法之前, Servlet 会先创建一个 ServletRequest 与 ServletResponse, 并将它们作为参数传给 service 方法, 它们隐藏了将请求发给 servlet 以及响应发给浏览器的复杂性。</p> <p>HttpServletResponse 也增加了许多方法。</p> <p>java.sql.ResultSet 接口类似于一个数据表, 通过该接口的实例可以获得检索结果集, 以及对应数据表的相关信息, ResultSet 实例通过执行查询数据库的语句生成。</p>
数据结构 与算法	注册后台模块主要简单抽象出了数据访问层, 并实现将注册信息导入数据库。

	<div>Sign.java</div> <div>这里主要是实现注册操作。在这里定义六个 EditText 变量，两个 Button 变量，在注册按钮的监听事件中，分别定义字符串得到六个输入内容的字符串。如果用户名栏为空或未按输入要求输入信息则输出提示“用户名长度需5-10 位用户名需要以英文字母开头，由英文字母数字和_组成 用户名必须包含至少一个大写字母 ”，如果姓名栏为空，则输出“姓名不能为空”，如果姓名栏为非法字符，则输出“姓名不能有非法字符”，如果密码栏为空，则输出“密码长度应为 6-12 位”，如果密码栏有非法字符，则输出“密码应由英文字母数字和_组成”，如果确认密码栏为空或与密码栏输入信息不一致，则输出“输入不一致”，如果年龄栏为空，则输出“年龄为空”，如果手机号码栏为空，则输出“电话为空”，如果手机号码栏未按要求输入，则输出“请输入正确的电话号码”使用 executeSign 执行注册操作。</div> <div>执行注册操作时这里考虑使用 AsyncTask，AsyncTask 允许执行的一个异步的任务在后台。考虑将耗时的操作放在异步任务当中来执行，并随时将任务执行的结果返回给 UI 线程来更新 UI 控件。</div>
--	---

5 程序运行流程图



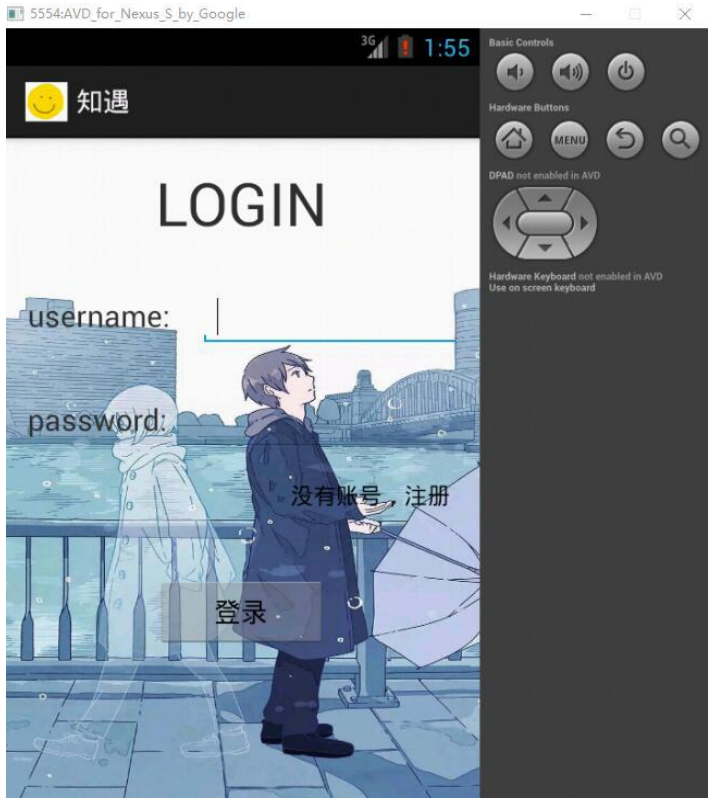


数据库中数据:

	username	name	age	phone
	Hanjun	呀刚发的	28	15200823422
	Hanjun1	李汉俊哈...	20	15200823422
	Littlejun2	你好呀	18	15200823422
▶	TX1998	宋茶茶	99	15616265159
	Yl_1610404	知南茶温暖	20	15199609258
*	NULL	NULL	NULL	NULL

	username	password
	Hanjun	123456
	Hanjun1	123456
	Littlejun2	123456
	TX1998	888888
▶	Yl_1610404	yl05257...
*	NULL	NULL

程序在虚拟机上运行如下:



控制台输出数据:

```
请求内容:[{"password":"19960121","username":"HollyLussie"}]  
返回内容:[{"userHobby":"看剧","userWeight":45,"userColor":"白色","userHeight":162,"userConstellation":"水瓶座","userId":1}]
```

6、实验心得

本次实验，悲喜交织，真的无从说起，本来前端还好，因为只是拖动组件，构建好一个布局，一般采用 linearlayout 布局，加上 tablelayout 就可以设计好一般的画面了。一开始不会用来错误的方式创建了每个页面，导致报错不停，最后问了老师才知道，是用 creat activity 来创建的。这样就不会漏掉一些配置。然后，在里面想用一些图片，作为按钮，需要透明的背景，找同学用 ps 弄了好久才弄出来。前端其实还好，主要是后端，一开始一直连不上后台服务，后来发现要设置一个网络的允许问题才行。但是，问题还没有结束，我们发现，中文处理不了，然后又找各种方法，最后才弄好，这个是真的痛苦，通过定位在前端，或者后端，一步步找到问题。当然，还有一个问题就是，我发现，如果用本地服务器的话，当你电脑换了一个网络之后，那么这个 ip 又变了，你又得修改，尤其是，如果你下到手机上使用的时候，你手机和电脑还要用的是同一个网络才可以使用。不然，手机连接不上后台。为了解决这个问题，然后找了个云服务器，把后端代码部署到服务器上面，这还有很多工作，比如要在服务器上面装 tomcat，还有 java 的环境，mysql 等等。最后把后端代码部署到服务器，这样我们的手机用普通网络就可以得到后台的服务。所以，就可以给远地的小伙伴玩了。

总的来说，虽然做了很久，但是通过这个实验，能够综合前几个实验所学到的知识，运用到实际上，确实增长了不少知识。

7、附程序源代码

1:Login.java

```
package com.example.mystore;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;

import org.json.JSONObject;

import android.os.AsyncTask;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class Login extends Activity {

    private EditText username;
    private EditText password;
    private TextView err;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        username = (EditText) this.findViewById(R.id.login_username);
        password = (EditText) this.findViewById(R.id.login_password);
        err = (TextView) this.findViewById(R.id.err);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
```

```

        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.login, menu);
        return true;
    }

    public void sign(View v){
        Intent intent = new Intent(this, Sign.class);
        startActivity(intent);
    }

    public void login(View v) throws IOException{
        String user = username.getText().toString();
        String pass = password.getText().toString();
        if(user.equals("")||pass.equals("")){
            System.out.println("user or pass is null");
            err.setText("用户名或密码为空");
            err.setVisibility(View.VISIBLE);
        }
        else{
            err.setText(null);
            err.setVisibility(View.INVISIBLE);
            Toast.makeText(this,"正在登录",Toast.LENGTH_SHORT).show();
            new SignInProcess().execute(user,pass);
        }
    }

    // Intent intent = new Intent(Login.this,Sign.class);
    // startActivity(intent);
}
//一定要开一个新的线程
private class SignInProcess extends AsyncTask<String, String, String> {
    @Override
    protected String doInBackground(String... params) {

        String username = params[0];
        String password = params[1];
        String result = "";

        //String s_url = "http://localhost:8080/background/NewLogin?username="+username+"&password="+password;
        // String s_url = "http://222.240.54.27:8080/background/NewLogin?username="+username+"&password="+password;
        String s_url = "http://10.63.28.235:8080/background/NewLogin?username="+username+"&password="+password;

        try {
            URL url = new URL(s_url);
            HttpURLConnection conn =
                (HttpURLConnection) url.openConnection();

```



```
        conn.setRequestMethod("GET");
        conn.connect();
        InputStream is=conn.getInputStream();
        InputStreamReader reader = new InputStreamReader(is, "UTF-8");
        int temp;
        while((temp=reader.read()) != -1) {
            result += (char)temp;
        }
    } catch(Exception e) {
        e.printStackTrace();
    }
    System.out.println("成功获得后台结果");
    return result;
}

@Override
protected void onPostExecute(String result) {
    try {
        JSONObject result_json = new JSONObject(result);
        //判断后台是否返回错误
        if(result_json.has("error")) {
            String error_code;
            error_code = result_json.getString("error");
            err.setText(error_code);
            err.setVisibility(View.VISIBLE);
            password.setText(null);
        } else {

            SignInSuccess(result_json);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

//成功验证，设置跳转页面
private void SignInSuccess(JSONObject info) {
    Intent intent = new Intent(this,MyPage.class);
    Toast.makeText(this,"登录成功",Toast.LENGTH_SHORT).show();
    try {
        intent.putExtra("username", info.getString("username"));
        intent.putExtra("name", info.getString("name"));
        intent.putExtra("age", info.getInt("age"));
    }
```

```
        intent.putExtra("phone", info.getString("phone"));
    } catch (Exception e) {
        e.printStackTrace();
    }
    startActivity(intent);
}

}
```

2. Myinfo.java

```
package com.example.mystore;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.Menu;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
public class MyInfo extends Activity {
    private Intent intent;
    private TextView username;
    private TextView name;
    private TextView age;
    private TextView phone;
    private String str_username;
    private String str_name;
    private String str_phone;
    private int int_age;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_my_info);
        username = (TextView) this.findViewById(R.id.username_info);
        name = (TextView) this.findViewById(R.id.name_info);
        age = (TextView) this.findViewById(R.id.age_info);
        phone = (TextView) this.findViewById(R.id.phone_info);

        intent= getIntent();
        str_username = intent.getStringExtra("username");
        str_name = intent.getStringExtra("name");
        int_age = intent.getIntExtra("age", 0);
        str_phone = intent.getStringExtra("phone");
    }
}
```

```
        username.setText(str_username);
        name.setText(str_name);
        age.setText(""+int_age);
        phone.setText(str_phone);
    }

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

    public void cancel(View v){
        Intent intent1 = new Intent(this,MyPage.class);
        try {
            intent1.putExtra("username", str_username);
            intent1.putExtra("name", str_name);
            intent1.putExtra("age", int_age);
            intent1.putExtra("phone", str_phone);
        } catch (Exception e) {
            e.printStackTrace();
        }
        startActivity(intent1);
    }
}
```

3. Mypage.java

```
package com.example.mystore;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.Menu;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;

public class MyPage extends Activity {
    private Intent intent;
    private TextView username;
    private String str_username;
    private String str_name;
    private String str_phone;
```

```
private int int_age;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_my_page);
    username = (TextView) this.findViewById(R.id.mypage_username);
    intent= getIntent();
    str_username = intent.getStringExtra("username");
    str_name = intent.getStringExtra("name");
    int_age = intent.getIntExtra("age", 0);
    str_phone = intent.getStringExtra("phone");
    username.setText(str_username);
}

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

public void showInfo(View v){
    Intent intent1 = new Intent(this,MyInfo.class);
    try {
        intent1.putExtra("username", str_username);
        intent1.putExtra("name", str_name);
        intent1.putExtra("age", int_age);
        intent1.putExtra("phone", str_phone);
    } catch (Exception e) {
        e.printStackTrace();
    }
    startActivity(intent1);
}

public void cancel(View v){
    Intent intent = new Intent(this,Login.class);
    startActivity(intent);
}

}
```

4. Sign.java

```
package com.example.mystore;
import java.io.InputStream;
import java.io.InputStreamReader;
```

```
import java.io.UnsupportedEncodingException;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.regex.Pattern;
import org.json.JSONObject;
import android.os.AsyncTask;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
public class Sign extends Activity {
    private EditText username;
    private EditText name;
    private EditText password;
    private EditText confirm;
    private EditText age;
    private EditText phone;
    private Button sign;
    private Button cancel;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sign);
        username = (EditText) this.findViewById(R.id.sign_username);
        password = (EditText) this.findViewById(R.id.sign_password);
        confirm = (EditText) findViewById(R.id.sign_confirm);
        name = (EditText) findViewById(R.id.sign_name);
        age = (EditText) findViewById(R.id.sign_age);
        phone = (EditText) findViewById(R.id.sign_phone);
        sign = (Button) findViewById(R.id.bt_sign);
        cancel = (Button) findViewById(R.id.bt_cancel);

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

```
public void sign(View view) throws UnsupportedOperationException {
    String str_username = username.getText().toString();
    String str_name = name.getText().toString();
    String str_age = age.getText().toString();
    String str_phone = phone.getText().toString();
    String str_password = password.getText().toString();
    String str_confirm = confirm.getText().toString();

    //检查 username
    String mess = "";
    if(str_username.length() < 5 || str_username.length() > 10)
        mess += " 用户名长度需 5-10 位 ";
    String pattern1 = "^[a-zA-Z][a-zA-Z\\d_]*";
    String pattern2 = ".*[A-Z].*";
    boolean match1 = Pattern.matches(pattern1, str_username);
    boolean match2 = Pattern.matches(pattern2, str_username);
    if(!match1) {
        mess += " 用户名需要以英文字母开头，由英文字母数字和_组成 ";
    }
    if(!match2) {
        mess += " 用户名必须包含至少一个大写字母 ";
    }

    if(!mess.isEmpty()) {
        Toast.makeText(this, mess, Toast.LENGTH_SHORT).show();
        return ;
    }

    //检查姓名
    if(str_name.isEmpty()){
        mess += "姓名不能为空";
        Toast.makeText(this, mess, Toast.LENGTH_SHORT).show();
        return ;
    }
    String pattern3 = "[a-zA-Z\\u4E00-\\u9FA5]*";
    boolean match3 = Pattern.matches(pattern3, str_name);
    if(!match3){
        mess += "姓名不能有非法字符";
        Toast.makeText(this, mess, Toast.LENGTH_SHORT).show();
        return ;
    }
    str_name = new String(str_name.getBytes("ISO-8859-1"), "UTF-8");
    //检查密码
    mess = "";
```

```
        if(str_password.length() < 6 || str_password.length() > 12
||str_confirm.length()>12||str_confirm.length()<6)
            mess += " 密码长度需 6-12 位 ";
        if(!str_password.equals(str_confirm)){
            mess += " 输入不一致 ";
        }
        String pattern6 = "[a-zA-Z\\d_]*";
        boolean match6 = Pattern.matches(pattern6, str_password);
        if(!match6) {
            mess += " 密码应由英文字母数字和_组成 ";
        }
        if(!mess.isEmpty()) {
            Toast.makeText(this,mess,Toast.LENGTH_SHORT).show();
            password.setText(null);
            confirm.setText(null);
            return ;
        }
        //检查 age
        if(str_age.isEmpty()){
            mess+=" 年龄为空 ";
            Toast.makeText(this,mess,Toast.LENGTH_SHORT).show();
            return ;
        }
        String pattern4 = "^[1-9]\\d{0,1}$";
        boolean match4 = Pattern.matches(pattern4, str_age);
        if(!match4){
            mess+=" 年龄只能为 0-99 的整数 ";
            Toast.makeText(this,mess,Toast.LENGTH_SHORT).show();
            return ;
        }

        //检查电话
        if(str_phone.isEmpty()){
            mess+=" 电话为空 ";
            Toast.makeText(this,mess,Toast.LENGTH_SHORT).show();
            return ;
        }
        String pattern5 = "^(13[0-9])|(15[^4,\\D])|(17[0-9])|(18[0,5-9])\\d{8}$";
        boolean match5 = Pattern.matches(pattern5, str_phone);
        if(!match5){
            mess+=" 输入正确的电话号码 ";
            Toast.makeText(this,mess,Toast.LENGTH_SHORT).show();
            return ;
        }
    }
```

```
        new SignUpProcess().execute(str_username, str_password, str_name, str_age,
str_phone);
    }
    private class SignUpProcess extends AsyncTask<String, String, String> {
        @Override
        protected String doInBackground(String... params) {
            String username = params[0];
            String password = params[1];
            String name = params[2];
            try {
                name = new String(name.getBytes("ISO-8859-1"), "UTF-8");
            } catch (UnsupportedEncodingException e1) {
                // TODO Auto-generated catch block
                e1.printStackTrace();
            }
            String age = params[3];
            String phone = params[4];
            String result = "";
            //String s_url = "http://222.240.54.27:8080/background/Sign?username="
            "+username+"&password="+password+"&name="+name+"&age="+age+"&phone="+phone;
            String s_url = "http://10.63.28.235:8080/background/Sign?username="
            "+username+"&password="+password+"&name="+name+"&age="+age+"&phone="+phone;
            try {
                URL url = new URL(s_url);
                HttpURLConnection conn = (HttpURLConnection) url.openConnection();
                conn.setRequestMethod("GET");
                conn.connect();
                InputStream is = conn.getInputStream();
                InputStreamReader reader = new InputStreamReader(is, "UTF-8");
                int temp;
                while((temp=reader.read()) != -1) {
                    result += (char)temp;
                }
            } catch (Exception e) {
                e.printStackTrace();
            }
            return result;
        }
        @Override
        protected void onPostExecute(String result) {
            try {
                JSONObject result_json = new JSONObject(result);
                if(result_json.has("error")) {
                    String error_code;
```



```
        error_code = result_json.getString("error");
        showerr(error_code);
        if(error_code.equals("用户名已被注册")){
            username.setText(null);
        }
        if(error_code.equals("姓名已被注册")){
            name.setText(null);
        }
    } else {
        SignInSuccess(result_json);
    }
}
catch (Exception e) {
    e.printStackTrace();
}
}

}

public void showerr(String error_code){
    Toast.makeText(this,error_code,Toast.LENGTH_SHORT).show();
}

private void SignInSuccess(JSONObject info) {
    Toast.makeText(this,"注册成功", Toast.LENGTH_SHORT).show();
    Intent intent = new Intent(this, MyPage.class);
    try {
        intent.putExtra("username", info.getString("username"));
        intent.putExtra("name", info.getString("name"));
        intent.putExtra("age", info.getInt("age"));
        intent.putExtra("phone", info.getString("phone"));
    } catch (Exception e) {
        e.printStackTrace();
    }
    startActivity(intent);
}

public void cancel(View v){
    Intent intent = new Intent(Sign.this,Login.class);
    startActivity(intent);
}
}
```