系统登录/注册模块（App）
详细开发文档

编写：李立天　　日期：2019-12-29
审核：　　日期：
批准：　　日期：
受控状态：　　是
发布版次：1.0　　日期：2019-12-29
编号：

变更记录

日期	版本	变更说明	作者
2019 年 12 月 29 日	1.0	初始版本	
2019 年 12 月 29 日	1.0	最终版本	

签字确认

系统模块	对应章节	对应部门	负责人签字

目 录

1 引言

1.1 编写目的

进一步理解 Java、SDK、Android Studio、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 背景

a) 软件系统的名称：Android http 通信服务

b) 项目负责人: 李立天

c) 程序员：李立天 索栗德 管泽伟

1.3 范围

系统包括的范围：
安卓端：Android 4.0 以上的所有版本均可使用。
网页端：主流浏览器均能使用

1.4 定义

一、 Android app 的开发

对于 Android app，包含如下界面：

（1） 登录界面：包含用户名、密码的文字标识以及相应的输入栏，记住用户名、密码勾选框，登录、注册以及忘记密码的按钮。当输入用户名以及密码后，点击登录按钮，则交数据提交至后台进行验证，如通过验证则跳转至欢迎界面，否则跳转回登录界面，并提示用户的验证错误原因；当用户点击注册按钮则跳转至注册界面；当用户点击忘记密码按钮则跳转至重置密码界面。

（2） 注册界面：包含用户名、密码以及确认密码的文字标识以及相应的输入栏，提交以及取消按钮。当输入相关信息后，首先验证输入的信息是否符合要求（用户名至少 5 位，最多 10 位，以英文字母开头，只允许包含英文字母、数字以及_，同时必须至少有一个大写英文字母；密码为 6-12 位，只允许包含英文字母、数字和_，同时要求确认密码必须与密码一致），如不符合要求则在界面内提示错误，只有符合要求才提交给后台进行注册操作。如注册成功则跳转至欢迎界面，否则跳转回注册界面并提示用户的注册错误原因；当用户点击取消按钮则返回登录界面。

（3） 重置密码界面：包括用户名输入框、电话号码输入框、验证码输入框、新密码输入框、新密码确认框以及显示密码勾选框。当用户点击发送验证码时将校验用户名和电话号码，并将验证码以手机短信的形式下发。点击确认重置则会校验验证码和用户名，重置成功则返回登录界面。最后是取消按钮，点击则返回登录界面。

（4） 欢迎界面：显示对用户的欢迎信息，其中必须包括用户的登录名。还包括修改密码按钮和编辑信息按钮。

（5） 修改密码界面：包括旧密码输入框、新密码输入框、新密码确认输入框以及显示密码勾选框。点击确认按钮则将旧密码和新密码提交至后台进行修改，点击取消则返回欢迎界面。

（6） 编辑信息界面：包括个人信息中的姓名、年龄、电话号码。将其显示在可编辑的 Text 框中。同样包括确认修改按钮和取消按钮。

1.5 参考资料

参考资料如下：

a) https://blog.csdn.net/qq_43901693/article/details/89741218

b) https://blog.csdn.net/liuyiming_/article/details/7704923

c) https://www.cnblogs.com/wuhuacong/p/10682451.html

1.6 术语与缩写解释

缩写、术语	解 释
Servlet	小服务程序或服务连接器
Tomcat	免费的开放源代码的 Web 应用服务器
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
SQL	结构化查询语言(Structured Query Language)

2 模块命名规则

应用程序名：HNU 登陆系统　　工程文件名：ParkingSystem

安卓前端：

HttpUtils　　（网络通信工具类）
FirstActivity.java　（临时界面）
LoginActivity　　（登录模块）
WelcomeActivity　（个人中心模块）
RegisterActivity　（注册模块）
ModifyActivity　　（修改密码模块）
ResetActivity　　（重置密码模块）
EditActivity　　（编辑信息模块）

网页前端：

login.jsp　register.jsp　reset.jsp　welcome.jsp　check.js

后端：

CheckCodeServlet　（核验验证码后台）
EditServlet　　（编辑信息后台）
ListServlet　　（获取个人信息后台）
LoginServlet　　（登录后台）
ModifyServlet　　（修改密码后台）
RegisterServlet　（注册后台）
SendCodeServlet　（发送验证码后台）

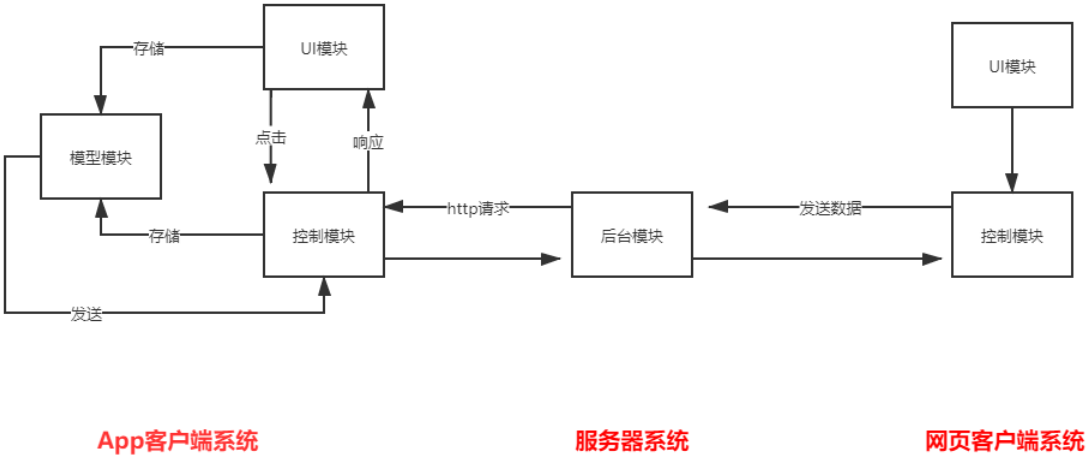
3 模块汇总

3.1 模块汇总表

子系统 A	
模块名称	功能简述
UI 模块	作为视图层展现给用户（xml 文件）
功能模块	处理用户的各种操作，将相关数据以 json 的格式发送至 servlet 后端进行处理，并接收处理后端的响应内容（Activity）
模型模块	存储数据和属性（R.java 和 user.xml）
子系统 B	
模块名称	功能简述
UI 模块	对网页进行的视觉优化文件，如 html，CSS 和 JS
功能模块	网页前端，以表单的形式向后台传输 json 数据，并响应后台的消息（如 jsp 文件）
子系统 C	
模块名称	功能简述
后台模块	实现的 servlet 功能，用于接收前端的消息并响应（Servlet）

3.2 模块关系图

对于 Android http 通信,包含三个系统,分别是 APP 客户端系统、网页客户端系统以及服务器系统,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	APP 的 UI 模块
功能描述	直接显示在屏幕上的视图,可以在其中放置多个控件并且设置其布局方式使其呈现在用户面前。
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。相对布局是一种通过设置相对位置进行的布局，如 android:layout_below 表示在目标组件之下。设置 id 的组件是为了控制模块（Activity）寻找方便。 **activity_login.xml**

	相对布局 RelativeLayout 可以通过指定子对象之间的关系或者子对象与父对象之间的关系来对空间或者文本进行布局。这种布局方式比较灵活，自由度大。Width 和 Height 均为 fill_parent,设置 background 可以将自己的图片保存入 drawable 中进行使用。 设置两个输入框，一个为用户名，设置 id 为 username。layout_width 为 fill_parent，layout_height 为 50dp，layout_paddingLeft 为 40dp，layout_marginTop 为 4dp，ems 为 10，hint 为“请输入用户名”。 对于密码输入框，设置基本相同但是 id 为 password，类型也显示为 password，这样输入密码时会显示为·而非密码，避免泄密。同时增加 singleLine=true 来保证输入的密码只能为一行，即不能使用回车键，hint 为“请输入密码”。密码框旁设置了一个 CheckBox，可以通过代码勾选是否显示密码。设置登录按钮，设置 id 为 log_button_1，layout_width 为 match_parent，layout_paddingLeft 为 40dp，background 为#000000（黑色），layout_gravity 为 center 按钮上的文字为"注册"，textColor 为"#FFFFFF"。 设置注册按钮，设置 id 为 log_button_2，layout_width 为 match_parent，layout_height 为 wrap_content，layout_maxHeight 为 50dp，layout_maxWidth 为 40dp，layout_marginTop 为 10dp，background 为#000000（黑色），layout_gravity 为 center 按钮上的文字为"登录"，textColor 为"#FFFFFF"。 对于 button 按键的样式，可以先在 drawable 文件夹中新建一个 xml 文件， 在文件里设置出想要的按键样式，使用时就将按键的 background 属性链接到新建的 xml 文件即可。设置提示信息，同 textview 文本框，将提示信息输入（比如密码不正确等），平时设置为不可见，只有点击登录键之后才会变成可见状态用于提示。 **其它 XML 文件的实现大致相同**
补充说明	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”; android:layout_alignParentTop 是否对齐父组件的顶部; 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; 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。match_parent 在 Android2.2 中 match_parent 和 fill_parent 是一个意思; android:layout_alignParentLe 贴紧父元素的左边缘 android:layout_alignParentRight 贴紧父元素的右边缘 android:layout_alignTop 本元素的上边缘和某元素的的上边缘对齐 android:layout_alignLeft 本元素的左边缘和某元素的的左边缘对齐 layout_gravity 是用来设置该 view 相对与起父 view 的位置

4.2 模块 2

模块名称	APP 功能模块
功能描述	这里是实现 Android 的 http 通信中客户端的功能。包括异步线程处理，访问 servlet，用 JSON 实现客户端和服务器端的通信等。
接口与属性	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。 Android 提供大量预定义的 view 来设计和组件你的布局。“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，例如线程布局、格子布局或相关性布局。可以子类化 View 和 ViewGroup 类（或者存在的子类）来创建自己的 widget 和而已并且应用它们到 activity 布局中。 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。这样可以单独维护用户接口设计，而与定义 activity 行为的代码无关。设置布局作为 UI 使用 setContentView()，传递资源布局的资源 ID。可是，你也可以创建新 Views 在你的 activity 代码，并且创建一个 view 层次通过插入新 Views 到 ViewGroup，然后使用那个布局通过传递到 ViewGroup 给 setContentView()。 HttpClient 中，接口 HttpParams 定义了这些参数的通用接口。
数据结构 与算法	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； onCreate 的方法是在 Activity 创建时被系统调用，是一个 Activity 生命周期的开始。在其中注册要用到的变量，比如 service，receiver，这些变量是无论 Activity 是在前台还是在后台都能够被响应到的，然后调用上面那个用来初始化的函数初始化布局信息。 activity 启动的时候为 onCreate ---> onStart ---> onResume：onStart 函数:注册一些变量。这些变量必须在 Android Activity 类在前台的时候才能够被响应。onResume 函数:调用一些刷新 UI 的函数，每当 Activity 调用到这里时就要刷新一下 UI 各控件的状态。在本项目中没有显示出来。 **HttpUtil1.java** 这里封装的是访问网络的 http 操作。这里主要是得到从服务器返回的 JSON 数组并且返回给 Activity.java 中，找到服务器的 IP 地址作为字符串，再设置编码格式。 使用 AsyncHttpClient 发送请求，用 POST 提交数据，所以创建一个 HttpPost 对象，用相应 servlet 的 URL 作为参数，转换编码格式为“UTF-8”后创建一个 HttpResponse 对象判断请求是否成功，如果请求成功则获取返回 json 报文，请求不成功则返回空。 在使用的过程中，我们只需要 HttpUtil.get(login_url, params, new JsonHttpResponseHandler())方法，将 url、json 数据传入，并自定义响应方法，就可以实现 Android 与 Servlet 之间的通信。 **FirstActivity.java** 这是一个临时的 activity，在这个 activity 当中只做了一件事情，就是判断我们在 sharedpreference 中保存的 login_state 是否为 1.如果为 1，表明用户上一次操作保持在登录状态，因此我们直接使用 intent 跳转至 WelcomeActivity。如果发现 login_state 为 0，则跳转到 login_state。

	LoginActivity.java 这里主要是实现登录操作。在这里定义两个 EditText 变量，三个 Button 变量，两个 Checkbox 变量，以及对应的后台处理 Servlet 的 URL。再在 onCreat 方法中初始化他们并为登录按钮添加监听事件。 在登录按钮的监听事件中，分别定义字符串得到两个输入内容的字符串。随后调用 login_check 函数对用户名和密码进行校验，如果用户名栏位空则输出提示“请填写用户名”，如果密码栏位空则输出提示“请填写密码”， 如果已经填写了用户名和密码，使用 login 执行登录操作。 执行登录操作时这里考虑使用我们刚刚定义的工具类 HttpUtils，将用户名和密码打包成一个 json 数据作为 parms，提交到我们指定的 URL。最后重定义一个响应方法，作为响应 Servlet 时的操作。这里需要重写 onSuccess 和 onFailure 方法。 OnSuccess 方法将在 servlet 响应成功时被自动调用，表明我们得到了网页的回应，将会接收到一个{state:?}的 json 字符串。 如果 state 为 1，表示我们登录成功，这里我们需要进行的操作有：1、将用户名和密码使用 SharedPreference 保存到 user.xml 当中，将来做保持登录功能的时候需要用到 2、使用 SharedPreference 保存勾选框的状态（是否记住用户名和密码）。3、获取用户的信息保存在 user.xml 当中。4、跳转到欢迎界面。5、设置 **login_state** 为 1. 如果 state 为 0，则表明用户名和密码错误，这时会弹出 toast 向用户进行提示。 当我们没有收到网页响应时，就会调用 onFailure 方法。这时我们只需要弹出 toast 向用户提示“网络错误”即可。 **WelcomeActivity.java** 进入到 WelcomeActivity 时，我们将首先调用 check_login()来核验用户名和密码是否正确。（因为有可能在其它设备更改了密码）。随后定义了 2 个按钮，分别对应“修改密码”和“编辑信息”。还有一个文本显示栏，设置为我们在 sharedpreference 当中保存的用户名。将会显示“欢迎你，亲爱的 XXX”字样。 最后，当我们按下”退出”按钮时，将返回到 loginActivity，同时将 login_state 置为 0 **RegisterActivity.java** 进入到 RegisterActivity，首先获取到许多文本框和按钮，分别对应了注册时需要填写的信息，当点击注册按钮时，首先会对数据的合法性进行校验（使用 Register_check 函数），随后当我们把注册信息提交给后台时，将会返回一个{state:?}。注册成功时，state 为 1，将会把用户信息保存在 user.xml（使用 sharedpreference），并跳转至欢迎界面，将 login_state 设置为 1. **ResetActivity.java** 进入到忘记密码功能，在这个 Activity 当中，需要用户输入用户名和绑定的手机号码，随后点击“发送验证码”，将用户名和手机号码发送至后台进行校验。如果发现是一致的，则后台随机生成一个六位数的验证码，向数据库中的 Codes 表中写入，并以短信的形式发送到手机。若用户名与手机号不匹配，则以 toast 的形式通知用户。 当用户接收到验证码时，将其填入验证码框，输入希望重置的新密码。最后点击提交按钮。将验证码提交给后台进行验证，如果验证成功，则在后台修改用户的密码，并将消息以 json 的形式返回给客户端。 **其它 Activity 的操作大致类似。**
补充说明	strings.xml 里定义了我们需要显示的字符串； 对 jsonObject 支持采用了 org.json 包； http 工具类是对 AndroidManifest.xm com.loopj.android.http.AsyncHttpClient 的封装操

	作； 文件中添加访问网络的权限：<uses-permission android:name="android.permission.INTERNET" />

4.3 模块 3

模块名称	APP 的模型模块
功能描述	R.java 文件自动生成，用来定义 Android 程序中所有各类型的资源的索引。（它是只读的，开发人员不对其修改）。
接口与属性	android 工程所有资源信息(组件、图片、字符等等)都是由 HashMap<Integer,Object>来存储的 key 值就是 R.java 中的静态变量值，value 就是相对应的各种对象信息(组件、图片、字符等等)。
数据结构 与算法	R.java 文件中默认有 attr、drawable、layout、string 等四个静态内部类，每个静态内部类分别对应着一种资源，如 layout 静态内部类对应 layout 中的界面文件，其中每个静态内部类中的静态常量分别定义一条资源标识符，如 public static final int main=0x7f030000;对应的是 layout 目录下的 main.xml 文件。
补充说明	R.java 及本地资源文件，使用“@+”声明的资源，系统会自动在 R.java 中创建。

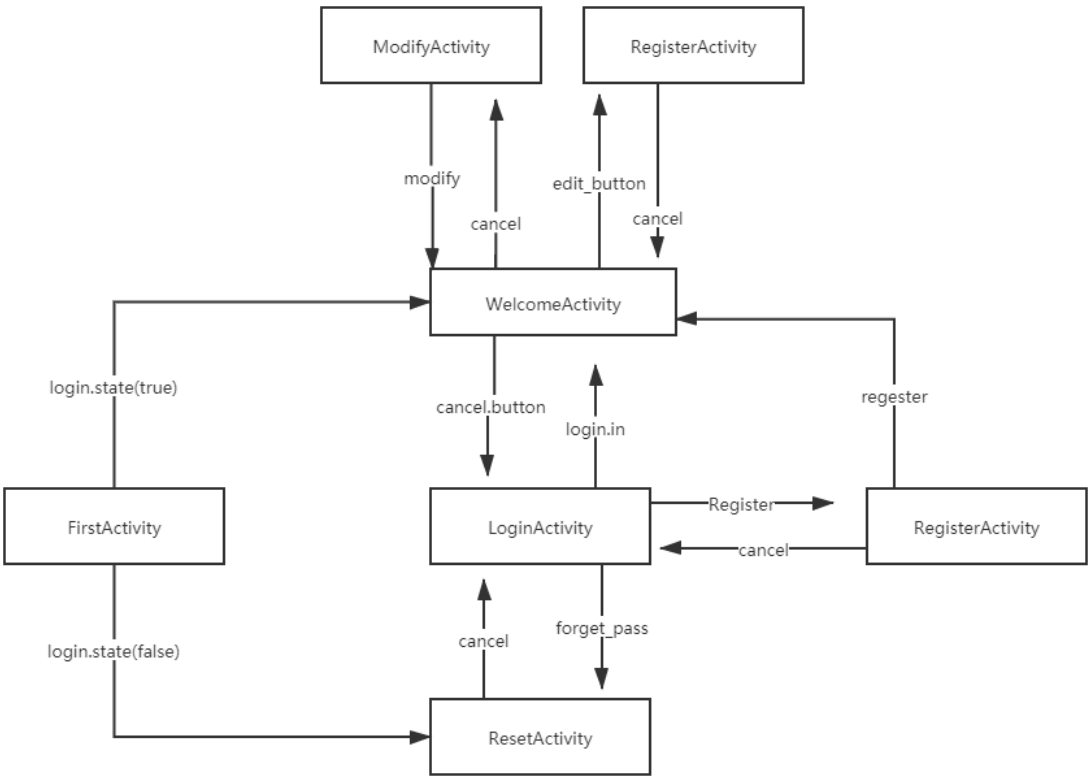
4.4 模块 4

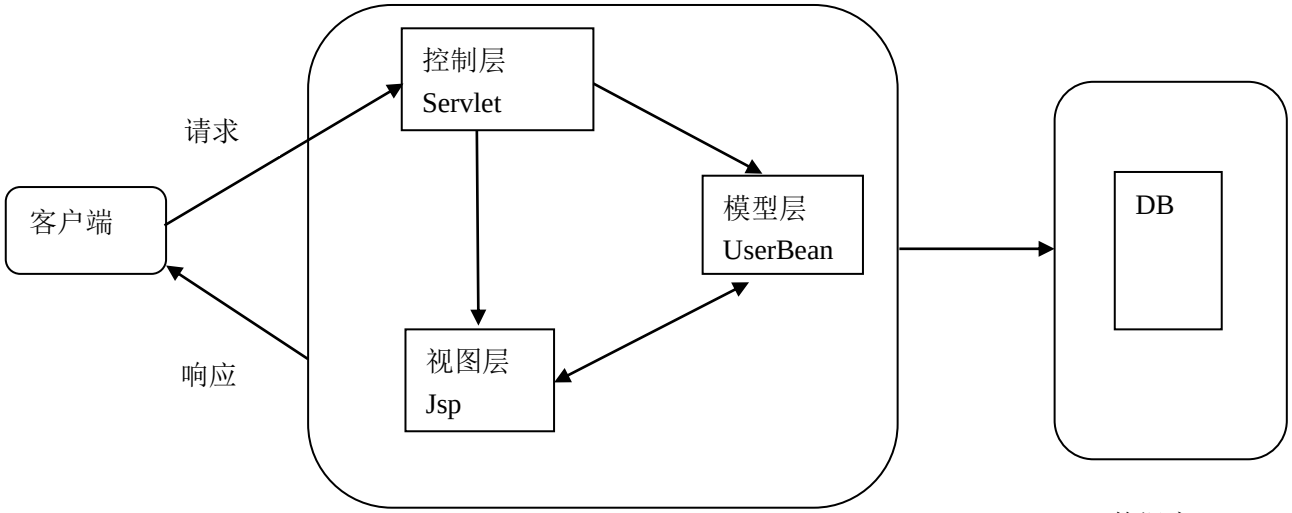
模块名称	后台模块
功能描述	这里主要是客户端的程序，具体体现为 LoginServlet.java，请求内容通过 servlet 返回给客户端，实现客户端和服务器的通信，注意这里需要访问数据库中的数据。
接口与属性	所有 Servlet 应用必须直接或者间接实现 Servlet 接口, Servlet 容器会将实现了 Servlet 接口的类加载至容器，以供访问。 ServletRequest 接口主要用于获取请求中的参数。HttpServletRequest 接口在 ServletRequest 接口的基础上增加了多种方法。 调用一个 Servlet 的 service()方法之前，Servlet 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。 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。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。java.sql.Statement 接口用来执行静态 SQL 语句，并返回执行结果。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。

数据结构 与算法	控制模块主要简单抽象出了数据访问层，并实现数据库的操作。同时将数据转化为 JSON 格式便于通信。 这里采用了分层的模型：Servlet-Service-Dao-Entity-Utils Utils 包含 JdbcUtils 数据库操作类，对执行 SQL 语句和 ResultSet 进行封装，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。 还包括封装的连接池操作 DataSourceUtils，以及发送验证码服务的 PhonceCode.java. Entity 层是实体数据层，包含了 users 和 person 的实体定义. 而 Dao 层是数据访问层，对存储在数据库当中的 Entity 进行操作。实际上是对 JdbcUtils 的调用，只需要传入相应的 SQL 语句，再处理结果即可。 Service 层是业务逻辑层，我们可以调用 userDao 中实现的方法，来适配我们自己的业务逻辑。例如登录、注册等具体的服务。 Servlet 层，就是后台处理代码，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。针对不同的请求，我们编写不同的 Servlet 做出响应。 由于在本次实验中我们设计了两个前端，对于这两个前端需要不同的响应措施。例如对于安卓端，我们需要在网页上调用 Writer 将 json 字符串打印出来让 Andtroid 获取；而网页前端需要我们设置一个 message 从后台传至前端。因此我们在发送 Json 数据的时候，对于安卓端添加一个项{Client：Android}，用于标识这是安卓端的请求。 **LoginServlet.java** 从前端获取用户名和密码，在 user 表中核验。若匹配，则返回{state:true}； **RegisterServlet.java** 从前端获取用户名、姓名、年龄、电话；检查用户名是否已经在 person 表中存在，若存在则返回{state:false}；若不存在则在 person 表中添加相应的条目，并返回 state 为 true **ListServlet.java** 当安卓登录成功和网页端登录、注册成功时将会进入的 servlet。对于安卓端，将 username 对应的 person 表信息以 json 的格式传回；对于网页端，直接显示 person 表的内容。 **SendCodeServlet.java** 前端提供用户名和手机号码，在 person 表中查找是否匹配。若匹配成功则生成一个六位数的随机码作为验证码。调用 phonecode.java 当中封装好的发送短信函数（使用阿里云短信服务），向手机号码发送验证码。并将验证码存储到数据库的 codes 表当中。最后返回一个{msg:”获取验证码成功”}。在失败时会返回{msg:”发送过于频繁”}或{msg:”网络错误”}。 这里不使用 session 的原因是：如果使用 session，可能会造成在多个浏览器中发送验证码而造成困扰的现象。 **CheckCodeServlet.java** 接收前端传来的用户名和验证码，在 codes 表中查找其是否匹配，如果匹配，则将用户提供的新密码写入 users 表中，并将验证码从数据库当中删除。如果不匹配，则返回{state:false}。 **其它的 servlet 思路大致与上面相同。**

补充说明	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。 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]）

5 程序运行流程图

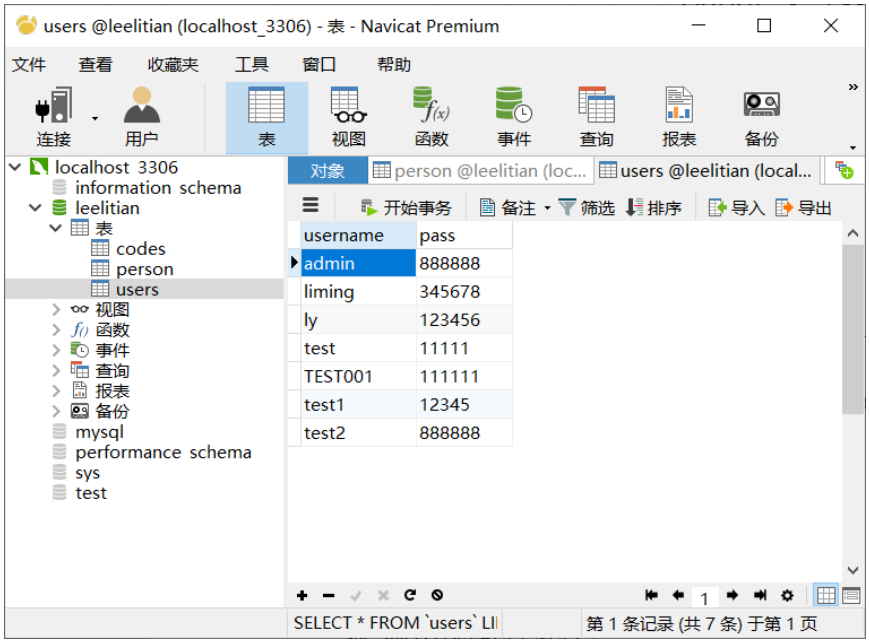




数据库建表语句：
包括 3 个表：User 表，Person 表，以及存储验证码的 Codes 表。

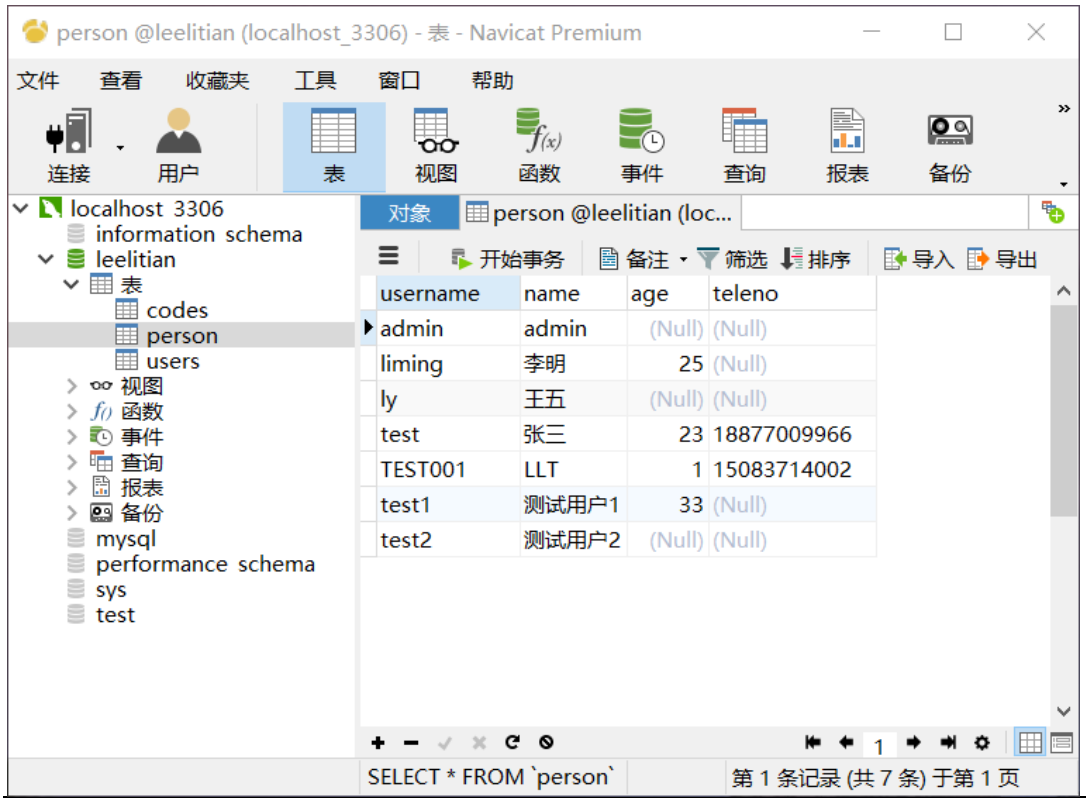
User 表：

```
CREATE TABLE `users` (
   `username` varchar(10) NOT NULL,
   `pass` varchar(12) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL,
   PRIMARY KEY (`username`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```



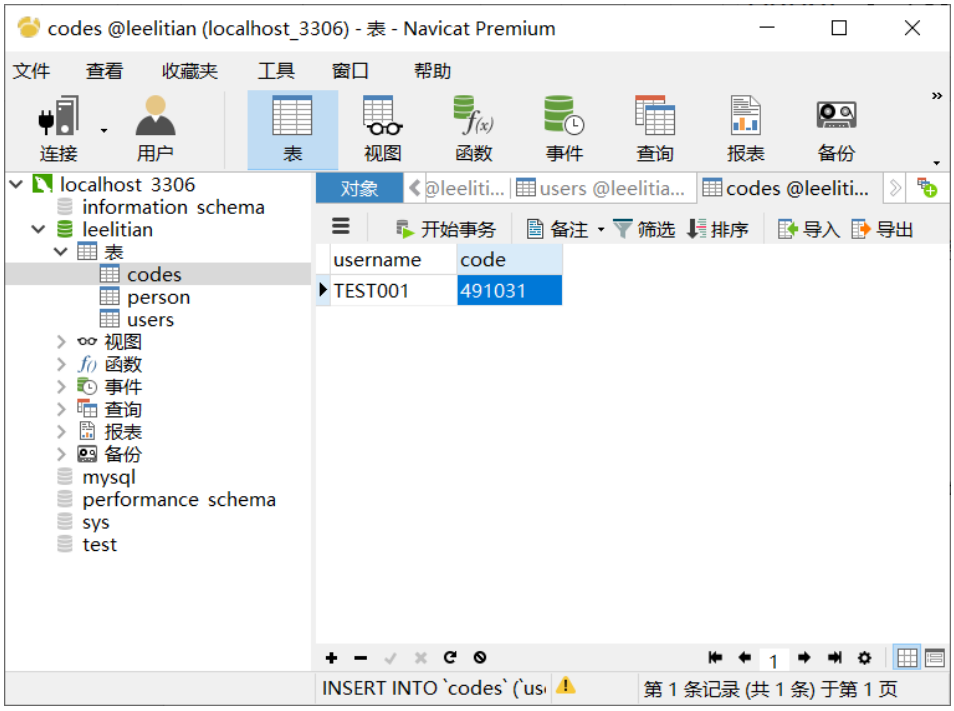
Person 表：

```
CREATE TABLE `person` (
  `username` varchar(10) NOT NULL,
  `name` varchar(20) NOT NULL,
  `age` int(11) DEFAULT NULL,
  `teleno` char(11) DEFAULT NULL,
  PRIMARY KEY (`username`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```



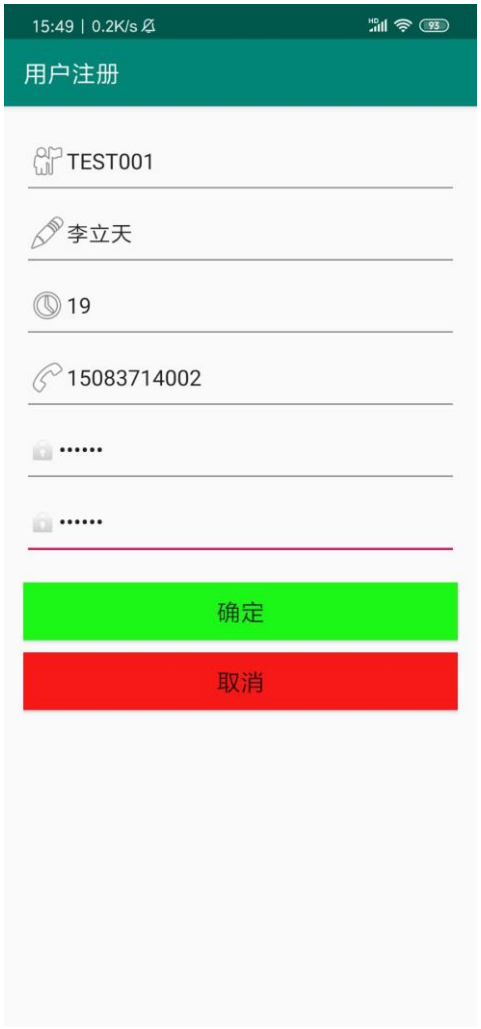
Codes 表：

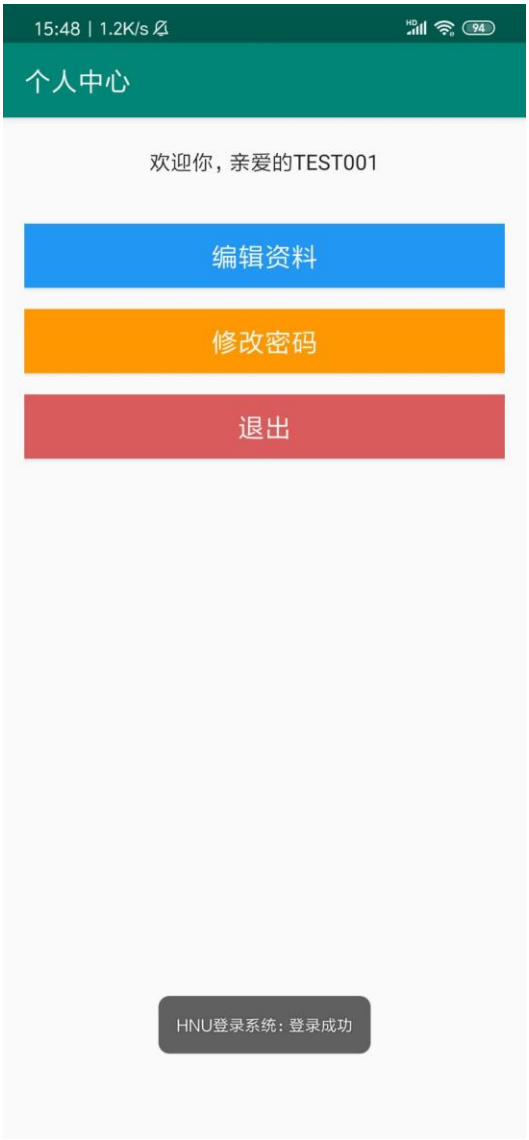
```
CREATE TABLE `codes` (
  `username` varchar(255) NOT NULL,
  `code` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```



程序在安卓手机上的运行截图如下：


15:48 | 0.1K/s
登录账号
TEST001
记住用户
记住密码
登录
注册
忘记密码


15:49 | 0.2K/s
用户注册
TEST001
李立天
19
15083714002
确定
取消


15:48 | 1.2K/s
个人中心
欢迎你，亲爱的TEST001
编辑资料
修改密码
退出
HNU登录系统：登录成功


15:49 | 0.0K/s
修改密码
111111
123456
123456
显示密码
确认修改
取消


15:50 | 1.3K/s
1069259986173558624
...您的验证码：328262，您正进行身份..
复制
TEST001
15083714002
填写获得的验证码
显示密码
发送验证码
确认重置
取消