**互助帮社区系统 V1.0**

* **使用说明**

|  |  |
| --- | --- |
| 版本 | 1.0 |
| 文档状态: | 编辑 |
| 作者: |  |
| 负责人: |  |
| 创建日期: | 2021年3月26日 |
| 更新日期: | 2021年3月30日 |

目录

[1 简介 4](#_Toc5205_WPSOffice_Level1)

[1.1 编写目的 4](#_Toc22212_WPSOffice_Level2)

[1.2 使用对象 4](#_Toc17897_WPSOffice_Level2)

[1.3 产品范围 4](#_Toc10608_WPSOffice_Level2)

[2 产品概述 5](#_Toc22212_WPSOffice_Level1)

[2.1 总体框架 5](#_Toc27431_WPSOffice_Level2)

[2.2 系统架构 5](#_Toc9097_WPSOffice_Level2)

[2.3 模块描述 6](#_Toc31361_WPSOffice_Level2)

[3 使用说明 7](#_Toc17897_WPSOffice_Level1)

[3.1 业务流程说明 7](#_Toc19250_WPSOffice_Level2)

[3.1.1 整体业务流程 7](#_Toc478_WPSOffice_Level3)

[3.1.2 消息处理流程 7](#_Toc20420_WPSOffice_Level3)

[3.1.3 数据访问流程 8](#_Toc6215_WPSOffice_Level3)

[3.2 功能模块说明 8](#_Toc5761_WPSOffice_Level2)

[3.2.1 交互处理模块 8](#_Toc5205_WPSOffice_Level3)

[3.2.2 连接数据模块 8](#_Toc13850_WPSOffice_Level3)

[3.2.3 存放模块 9](#_Toc2801_WPSOffice_Level3)

[3.2.4 通用服务模块 9](#_Toc10796_WPSOffice_Level3)

# 简介

现如今，随着经济的快节奏发展，人们的生活需求日益增多。在这些需求日益增多的同时，总会出现很多问题：当我们家里某些小电器出了些小故障时，当我们要为家人烹饪某道菜肴时，又或者当我们为了一时的需求而不得已购买某种工具时，却又发现用完一次，就永远搁置在一边了......这时，我们推出了“互助帮”，旨在在邻里之间搭建一个互帮互助的桥梁。

## 编写目的

本文档为使用说明文档，为产品的使用与维护提供信息基础。

## 使用对象

本文档的使用对象主要为产品测试与使用的业主。

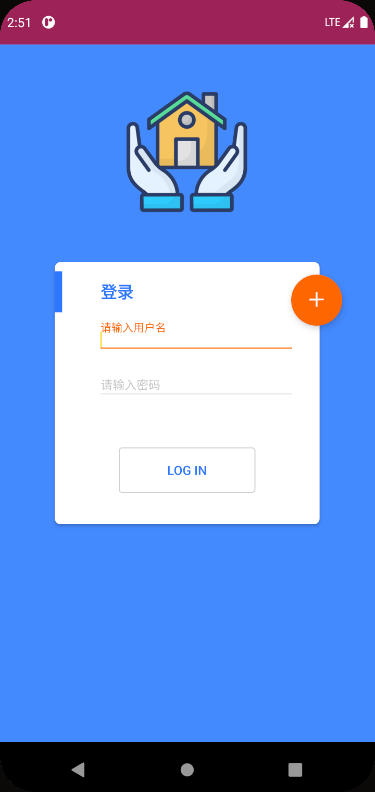
## 产品范围

互助帮，一个小区业主互助的安卓APP，通过但不限于请求帮忙，借物品，分享信息等，来增进社区中业主间的关系与认同感，便于居委会物业管理社区，业主熟悉彼此。主要包括如下模块，

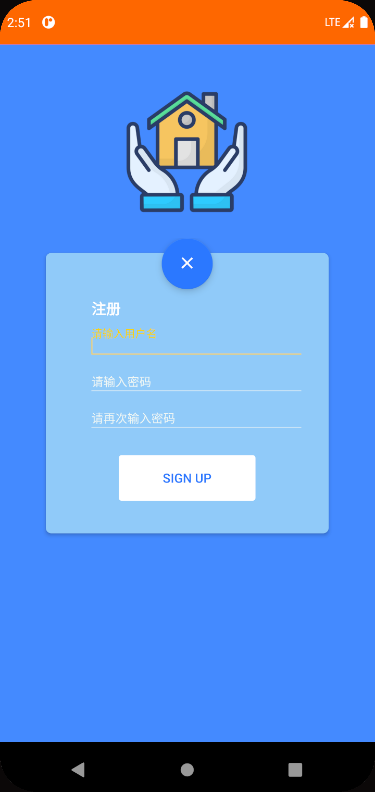
1. **登陆注册模块**：

此模块需要用户填写信息，如用户名，签名等等，

可以在聊天中观察到他人信息，并与其他模块相连；



登录界面



注册界面

**2) 浏览信息模块：**

可以阅读查看社区中的信息，

可以发表评论和留言，

可以查看信息被浏览的次数和历史记录；



热门界面



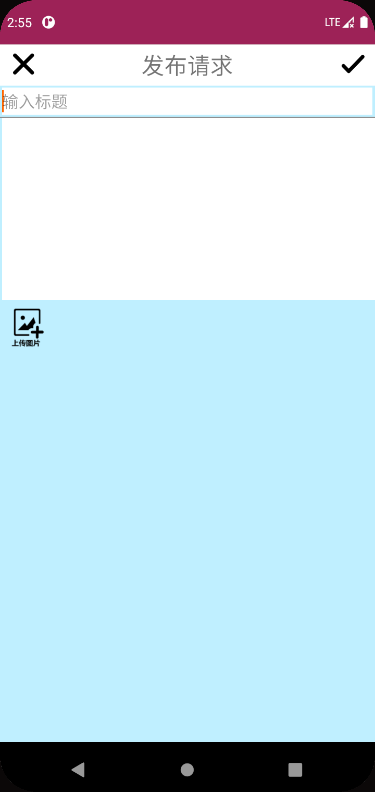
侧边栏的用户信息



评论界面

**3) 发布信息模块**

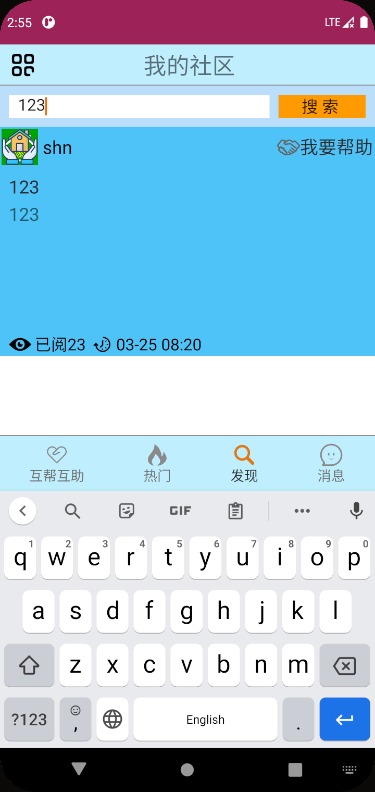
可以发表图片或者文字信息；



发布信息界面

**4) 搜索信息模块**

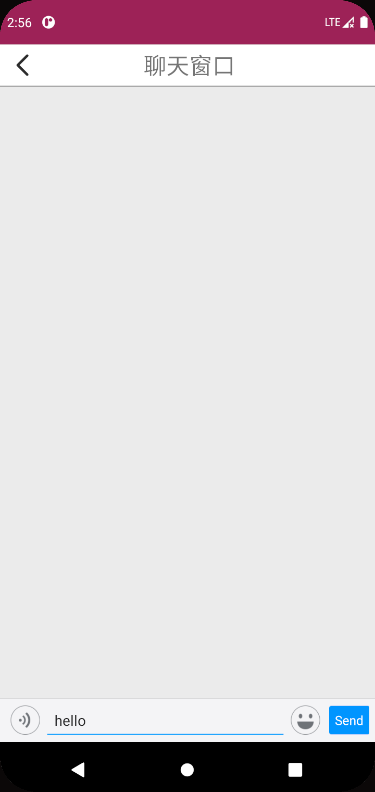
可以对信息进行检索；



搜索帖子的信息

**5) 通讯功能模块**

可以在信息界面点击发帖人头像可以与其进行聊天，在主界面有聊天列表；



私聊界面

# 产品概述

## 总体框架

Android界面作为表示层；在功能层接受用户传递的消息并返回相关信息，或者跳转到某些界面；数据层对数据管理存储。即使用C/S三层结构

## 系统架构

主要开发语言为Java和SQL语句，Android接受服务器传来的消息，处理点击事件等逻辑操作；tomcat服务器采用servlet与数据交互；数据的存储与使用采用sql server；聊天功能使用融云的Android框架进行通讯；

其中原生的网路请求方法为使用HttpURLConnection和HttpClient，使用起来较为复杂，第三方即是将复杂的部份封装起来，让我们更方便使用：采用NoHttp来实现Http标准协议框架，支持多种缓存模式，支持大文件、多文件上传、文件下载、自动维持Cookie、异步/同步请求；

Get请求透过网址获取数据，通过url即可；而post方法需要传入数据，其中数据的格式，采用JSON形式：将request/response的body主体转为JSONObject和JSONArray中；

通过servlet的接口规范，在servlet容器中（如tomcat）监听端口根据客户端的请求、URL，选择对应的servlet进行处理。通过service方法返回response对象给客户端

# 使用说明

## 业务流程说明

### **整体业务流程**

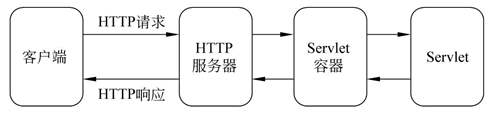


图 1  Servlet 应用程序的体系结构

用户在android设备上进行登录/注册等相关操作，此时会在客户端与服务端建立一个连接，通过ip地址和端口号确定，在向服务器发出一个get/post请求，通过层层传递，形成一段json格式数据发送到服务端

服务端通过——接受客户端的request请求，并通过解析得到json格式下正确的信息格式，并返回相关信息以及成功/失败信息

客户端解析服务端返回的信息，判断是否访问成功或者抛出相关的异常结束程序，并将得到的信息返回到用户所接触的界面上

服务端确定客户端接受的信息后，会将相关信息转存存储到数据库中，以长久的保存

### **消息处理流程**

具体的消息流程如下

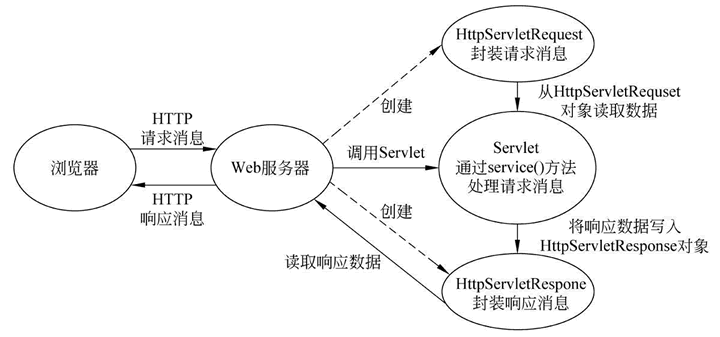


图 1  浏览器访问Servlet过程

首先浏览器向 Web 服务器发送了一个 HTTP 请求，Web 服务器根据收到的请求，会先创建一个 HttpServletRequest 和 HttpServletResponse 对象，然后再调用相应的 Servlet 程序。

在本项目中，servlet用来读取android客户端发送的隐式HTTP请求数据，访问数据库的各种表格和数据：在 Servlet 程序运行时，它首先会从 HttpServletRequest 对象中读取数据信息，然后通过 service() 方法封装请求信息、响应信息，并将处理后的响应数据写入到 HttpServletResponse 对象中。

最后，Web 服务器会从 HttpServletResponse 对象中读取到响应数据，并发送给浏览器。

### **数据访问流程**

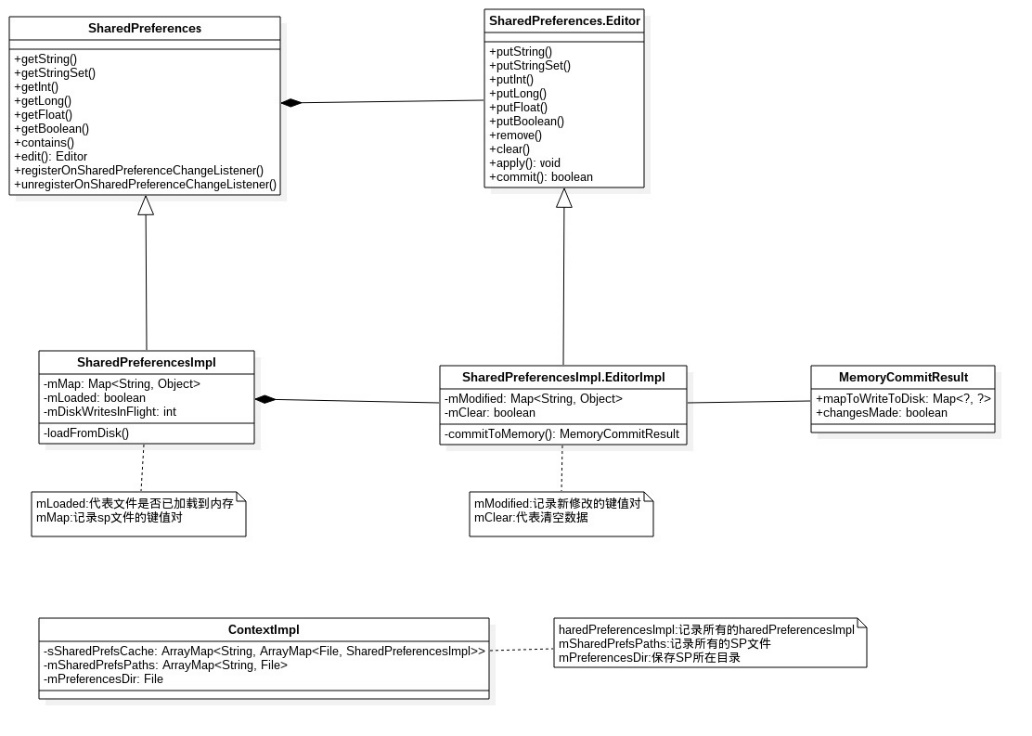
资源访问采用REST的定义，通过HTTP get/put/post/patch/delete等方法完成。例如：

1）资源服务层收到REST的get请求，根据请求内容，可以调用相应的服务bean获取资源，也可以将请求转到route进行后续处理获取资源，并返回资源。

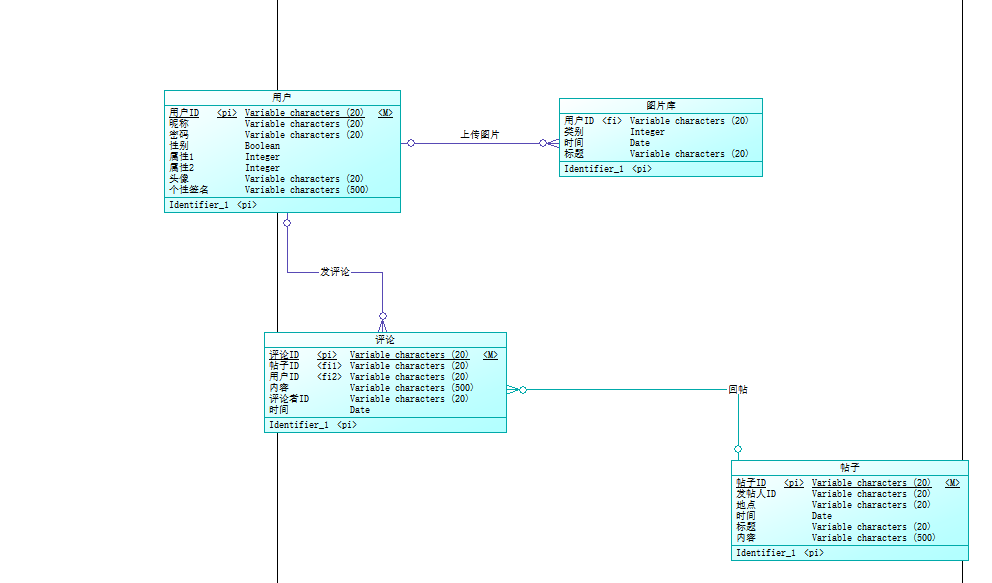
2）资源服务层收到REST的put或者patch请求，根据请求内容调用bean或者转到route进行资源更新。

### **数据存储流程**

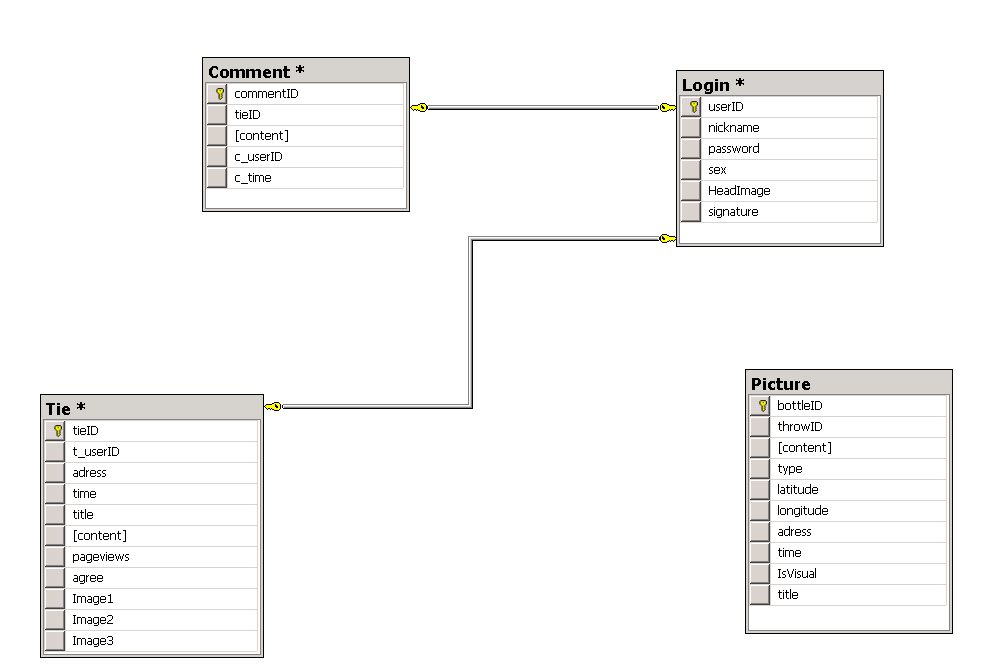
使用sharedpreference保存基本数据格式，可以永久存储数据，editor对数据可以进行一定的修改；



而对于用户自定义的类型，则利用JSON来存储和交换文本信息，将对象与文本进行转化，得到JSONObject文本格式并用String基本类型转换，之后利用JDBC通过sql语句存储到sql server中。



数据库的逻辑结构设计



数据库关系图

## 功能模块说明

### **登录注册模块**

登录注册通过使用sharedpreference保存key-value，可以不通过数据库，直接在程序的各个activity中保存，修改，提交各种基本数据类型的值，使得下一侧登陆时候能够自动保存用户名和密码实现自动登录；

同时，当程序退出后，用户信息将会通过网络传输到数据库中；

### **浏览信息模块**

帖子作为信息，有各种形容元素：如发帖人，内容，时间等等，这里是使用arraylist<collection>方法，将类作为动态数组的元素进行存放数据，再通过recycleView组件进行存放，显示在界面中；

### **发布信息模块**

分为标题、内容和插入图片，通过uri获取图片，放入压缩数据流中进行存储，一同其他的请求数据，用json的形式进行传输。

### **搜索信息模块**

从服务器中加载数据，同样也是通过servlet前去访问数据，并得到它的返回值，放入JSONArray中进行比较，然后通过网络请求获得存放在数据库中的数据。

### **通讯功能模块**

采用融云的即时通讯模块，有好友的信息列表以及私人聊天的界面，根据app key获得该key下的所有用户，而app secret用来生成sha1加密签名保证数据的安全性，同时每个用户获得一个token用来保存信息