

# 融聚邻里社区平台 软件分析设计文档

BUGKILLER<sup>1</sup>

---

<sup>1</sup>李冠达、孙世坤、张颖、刘畅



# Contents

<b>1</b>	<b>引言</b>	<b>1</b>
1.1	标识	1
1.2	系统概述	1
1.3	文档概述	1
<b>2</b>	<b>需求分析</b>	<b>1</b>
2.1	需求列表	1
2.1.1	功能需求	1
2.1.2	非功能需求	3
2.2	用例列表	4
2.2.1	管理员系统	4
2.2.2	商家系统	4
2.2.3	微信小程序系统	5
2.3	用例图	6
2.4	用例描述	8
2.4.1	Web-管理员端	8
2.4.2	Web-商家端	16
2.4.3	微信小程序端	26
<b>3</b>	<b>体系结构设计</b>	<b>35</b>
3.1	模块划分	35
3.2	系统行为描述	36
<b>4</b>	<b>模块间主要接口设计</b>	<b>41</b>
4.1	微信小程序部分的前后端交互	41
4.2	管理员 web 界面的前后端交互	49
4.3	商家 web 界面的前后端交互	58
4.4	商家端交易流程的前后端交互	67

<b>5</b>	<b>各模块的主要设计</b>	<b>71</b>
5.1	Web-管理员端 . . . . .	71
5.1.1	总体页面 App . . . . .	71
5.1.2	store . . . . .	71
5.1.3	router . . . . .	72
5.1.4	ManagerLogin . . . . .	72
5.1.5	ManagerIndex . . . . .	73
5.1.6	ResidentInfo . . . . .	73
5.1.7	ShopInfo . . . . .	74
5.1.8	ShopRegisterInfo . . . . .	75
5.1.9	OrderInfo . . . . .	75
5.1.10	ReceivablesInfo . . . . .	76
5.1.11	DiscountRule . . . . .	76
5.2	Web-商家端 . . . . .	77
5.2.1	总体页面 App . . . . .	77
5.2.2	store . . . . .	78
5.2.3	router . . . . .	78
5.2.4	ShopLogin . . . . .	78
5.2.5	ShopRegister . . . . .	79
5.2.6	ShopIndex . . . . .	80
5.2.7	MyInfo . . . . .	81
5.2.8	GoodsInfo . . . . .	82
5.2.9	StartTransaction . . . . .	84
5.2.10	TransactionHistory . . . . .	85
5.3	后端 . . . . .	86
5.3.1	总体架构 . . . . .	86
5.3.2	微信小程序后端 . . . . .	87
5.3.3	商家对应后端 . . . . .	88
5.3.4	管理员对应后端 . . . . .	90
5.4	微信小程序端 . . . . .	91
5.4.1	开发目录 . . . . .	91
5.4.2	全局模块 . . . . .	91
5.4.3	index . . . . .	93
5.4.4	start . . . . .	93
5.4.5	regist . . . . .	94
5.4.6	my . . . . .	95

5.4.7	shoplist . . . . .	96
5.4.8	shopdetail . . . . .	97
5.4.9	pay . . . . .	98
5.4.10	edit . . . . .	99
5.4.11	order . . . . .	100
5.4.12	score . . . . .	101
5.4.13	setting . . . . .	101
<b>6</b>	<b>数据库设计</b>	<b>103</b>
6.1	技术选型 . . . . .	103
6.2	模型描述 . . . . .	103
6.2.1	微信用户信息 . . . . .	104
6.2.2	订单信息 . . . . .	105
6.2.3	商家信息 . . . . .	106
6.2.4	管理员信息 . . . . .	106
6.2.5	商品信息 . . . . .	107
6.3	数据库连接 . . . . .	107
6.4	图片存放 . . . . .	107
<b>7</b>	<b>软件测试</b>	<b>109</b>
7.1	概述 . . . . .	109
7.2	后端测试环境 . . . . .	109
7.3	测试样例 . . . . .	109
7.3.1	工具类部分（主要是 https 请求的工具类） . . . . .	109
7.3.2	微信小程序部分 . . . . .	110
7.3.3	订单部分 . . . . .	110
7.3.4	商家端部分 . . . . .	111
7.3.5	管理员部分 . . . . .	112
7.4	测试结果 . . . . .	113
<b>8</b>	<b>鸣谢</b>	<b>115</b>



# 第 1 章. 引言

## 1.1 标识

本文档适用于 BUGKILLER 队开发的融聚邻里“社区商业数据联盟”1.0 版本。

## 1.2 系统概述

融聚邻里“社区商业数据联盟”是一个闭环的业务系统。通过设计和实现社区居民（会员）基础信息、会员等级及升级规则、获取/消费的积分规则以及商家结算规则，该系统可以实现前端会员消费、商家结算，后端对规则和数据可进行查看和管理。该系统由 BUGKILLER 队于 2019 年 10 月至 2020 年 1 月间开发和维护。北京融聚邻里教育咨询有限公司是该项目的需求方，清华大学计算机系软件工程课程为该项目提供支持。

## 1.3 文档概述

本文档是融聚邻里“社区商业数据联盟”系统的软件分析设计文档。本文档包含了以下内容：

- 软件开发过程中根据需求方说明进行的需求分析的结果
- 根据需求得到的系统结构设计方案
- 系统各个模块间接口的设计
- 系统各个模块的具体设计
- 数据库的具体设计

- 对系统进行的软件测试所使用的方法及测试的结果



## 第 2 章. 需求分析

### 2.1 需求列表

#### 2.1.1 功能需求

##### Web-管理员端

- 管理员的登录
- 平台注册用户（以下均简称“会员”）基础信息的管理
  - 查看与管理微信端会员基础信息。
- 会员等级及升级规则的创建与管理
  - 设定会员等级并设立其时效性规则，同时设定和管理会员等级的升降级规则
- 平台注册的商家信息的创建与管理
  - 审核申请注册合作的商家信息，查看并管理已合作商家的信息
- 积分获取规则的创建与管理
  - 创建、配置与管理会员在消费中获取积分的规则，同时管理其权限
- 积分消费规则的创建与管理
  - 创建、配置与管理会员在消费中使用积分的规则，同时管理其权限
- 会员积分积累信息的查看、管理
  - 查看每一位会员每一笔积分积累的记录以及进行相应管理
- 消费信息的查看、管理

- 查看会员每一笔积分消费的记录以及进行相应管理。
- 商家结算信息的查看、结算、管理
  - 查看和管理每一位合作商家在交易中获取的用户积分及其对应金额；定期审核并完成合作商家提出的申请结算。

## Web-商家端

- 未注册商家申请注册
  - 未注册商家可在填写必要的详细信息后申请注册并等候管理员审核结果
- 已注册商家（以下略）的登录
- 查看和维护基础信息
  - 查看并随时修改的基础信息（例如头像、店铺名称、本店获取积分比例等）
- 商品信息的维护
  - 对商店内商品信息的增、删、改、查，同时可使部分商品参与优惠活动，实行特价处理
- 结算信息查看
  - 查询本店内每一笔交易的详细信息，查看上一次平台结算至今商家获取的用户积分（根据交易发生时的优惠政策对应一定的结算金额），并可向平台发起结算申请
- 消费场景实现
  - 通过融聚邻里平台，与成为会员的消费者完成完整的交易流程——包括商品的挑选和录入、交易密钥的输入和验证、消费金额的计算、积分的使用和获取等，并记录下完成交易的订单

## 微信小程序端

- 会员绑定
- 会员登录
- 会员个人信息查看

- 会员消费信息的查看
  - 用户可查询到自己每一笔消费累计的积分详情。
- 会员权益查看
  - 用户可查询自己的会员权限，能优惠购买的商品，剩余能使用的积分等。
- 会员积分消费场景的实现
  - 用户到商家现场，购买商品或服务时，通过出示个人信息，获取商家折扣或是积分抵用消费金额。

### 2.1.2 非功能需求

#### 性能需求

##### 传输效率

- 客户要求：对正常负荷下的运行速度要求不影响用户正常用作，客户端运行时不会有呆滞的感觉。
- 前端做缓存
- 采用多次请求的方法

##### 安全性

- 基本原则：实现基本的验证
- 所有网络请求采用 https
- 通过微信的官方接口保证小程序的安全性
- web 端通过使用 token 技术验证来保证此部分的安全性

##### 鲁棒性

- 基本原则：系统具有一定的可靠性与稳定性，系统异常不可导致系统崩溃不能使用
- 服务器可以允许同时处理多个订单
- 服务器的所有实现考虑并行需求

- 服务器端尽可能地考虑输入的情况

## 2.2 用例列表

涉及的用例如下：

### 2.2.1 管理员系统

- 登录
- 管理会员信息
- 管理合作商家信息
- 审核合作商家提出的结算申请
- 审核未注册商家的注册申请
- 查看全平台交易记录
- 设定并管理全平台积分获取比例、消费比例的上限值
- 设定会员等级要求（消费额有效天数及各等级对应金额下限）和优惠规则（各等级对应的真实的积分获取比例）

### 2.2.2 商家系统

- 申请注册
- 登录
- 修改密码
- 维护店铺信息
- 维护商品信息
- 发起交易
- 查看交易记录

- 发起结算申请
- 注销账户

### 2.2.3 微信小程序系统

- 授权
- 注册
- 登录
- 修改信息
- 修改头像
- 浏览商家
- 浏览商品
- 发起支付
- 查看订单

## 2.3 用例图

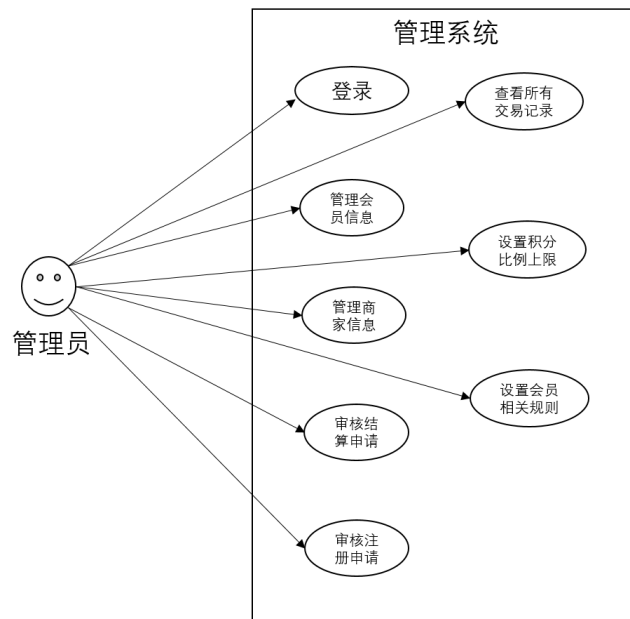


Figure 2.1: 管理系统用例图

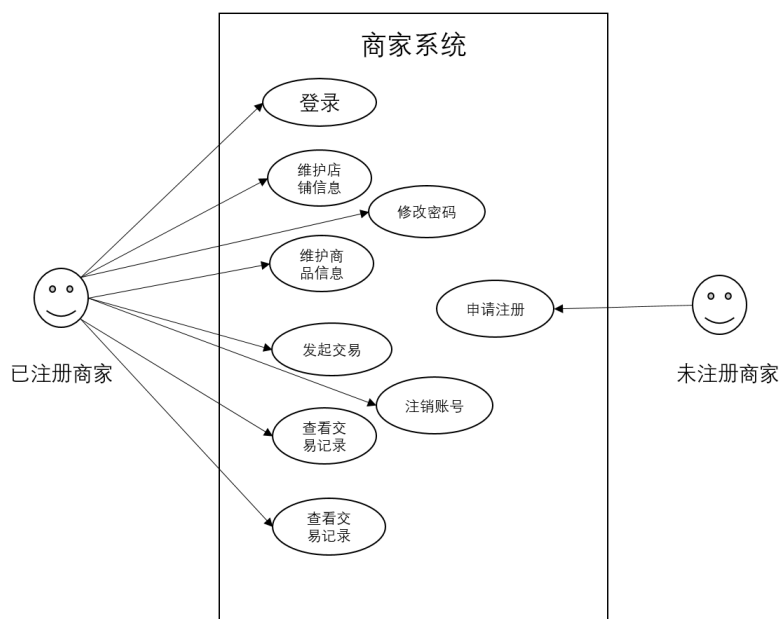


Figure 2.2: 商家系统用例图

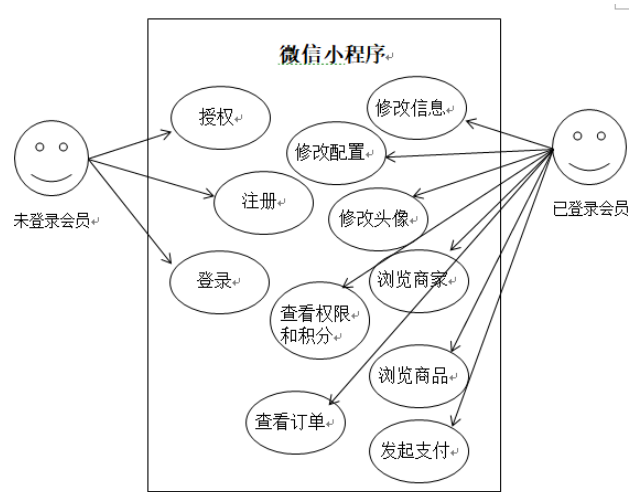
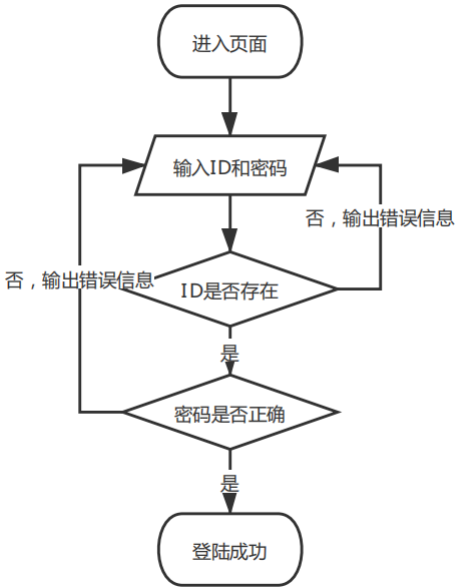


Figure 2.3: 微信小程序端用例图

## 2.4 用例描述

### 2.4.1 Web-管理员端

#### 登录

说明	管理员登陆系统
前提条件	管理员尚未登入系统
触发条件	打开管理员端网站即可
流程	 <pre>graph TD; A([进入页面]) --&gt; B[/输入ID和密码/]; B --&gt; C{ID是否存在}; C -- 否, 输出错误信息 --&gt; B; C -- 是 --&gt; D{密码是否正确}; D -- 否, 输出错误信息 --&gt; B; D -- 是 --&gt; E([登陆成功]);</pre>
后置条件	获取登录的 token



管理会员信息

说明	管理员管理平台上注册的会员信息
前提条件	管理员已经登入系统
触发条件	点击左侧导航栏中的“居民列表”
流程	<div><pre>graph TD; A([点击“居民列表” 进入页面]) --&gt; B[/查看会员信息/]; B --&gt; C([查看成功，结束]);</pre></div>
后置条件	无

管理商家信息

说明	管理员管理平台上注册的合作商家信息
前提条件	管理员已经登入系统
触发条件	点击左侧导航栏中的“商家列表”
流程	<div><pre>graph TD; A([点击“商家列表” 进入页面]) --&gt; B[/查看商家信息/]; B &lt;--&gt; C[/搜索商家/]; B --&gt; D([查看成功，结束]);</pre></div>
后置条件	无

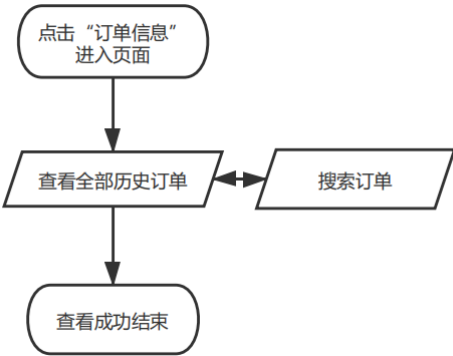
审核商家结算申请

说明	管理员审核并处理合作商家定期提出的结算申请
前提条件	管理员已经登入系统
触发条件	点击左侧导航栏中的“商家结算”
流程	<div><pre>graph TD; A([点击“商家结算” 进入页面]) --&gt; B[/查看申请结算信息/]; B --&gt; C[/同意或拒绝结算/]; C --&gt; B; C --&gt; D([结束]);</pre></div>
后置条件	同意或拒绝商家的结算申请, 并在数据库中做出相应改动

审核商家注册申请

说明	管理员审核并处理未注册商家的注册申请
前提条件	管理员已经登入系统
触发条件	点击左侧导航栏中的“商家申请”
流程	<div><pre>graph TD; A([点击“商家申请” 进入页面]) --&gt; B[/查看商家申请信息/]; B &lt;--&gt; C[/同意或拒绝申请/]; C --&gt; D([结束]);</pre></div>
后置条件	同意或拒绝商家的注册申请, 并在数据库中做出相应改动

查看全平台交易历史

说明	管理员查看全平台所有商家的交易历史
前提条件	管理员已经登入系统
触发条件	点击左侧导航栏中的“订单信息”
流程	<div><pre>graph TD; A([点击“订单信息” 进入页面]) --&gt; B[/查看全部历史订单/]; B &lt;--&gt; C[/搜索订单/]; B --&gt; D([查看成功结束])</pre></div>
后置条件	无

## 设置通用的积分获取比例和使用比例上限

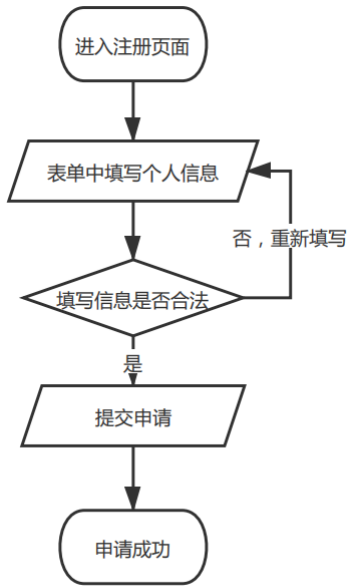
说明	管理员设置全平台通用的积分获取比例 (购买原价 1 元的商品所获得的积分数) 上限和使用比例 (购买原价商品时 1 积分可以代替的金额) 上限
前提条件	管理员已经登入系统
触发条件	点击左侧导航栏中的“优惠管理”，输入新的积分比例并点击”确认修改“
流程	<pre>graph TD; A([点击“优惠管理” 进入页面]) --&gt; B[/设置积分获取和兑换比例/]; B --&gt; C{确定是否修改}; C -- 是 --&gt; D([修改成功，结束]); C -- 否 --&gt; B;</pre>
后置条件	修改全平台的积分优惠规则

设置会员等级规则

说明	管理员设置会员等级规则及其对应的优惠政策。其中包括：消费额有效期、会员达到各个等级所需的有效期内的消费金额、各个等级对应的真实积分获取比例
前提条件	管理员已经登入系统
触发条件	点击左侧导航栏中的“优惠管理”，输入新的会员等级规则参数（共 11 个）并点击”确认修改“
流程	<div><pre>graph TD; A([点击“优惠管理” 进入页面]) --&gt; B[/设置会员等级规则 和优惠规则/]; B --&gt; C{确定是否修改}; C -- 否 --&gt; B; C -- 是 --&gt; D([修改成功，结束]);</pre></div>
后置条件	修改全平台的会员等级规则及优惠细则

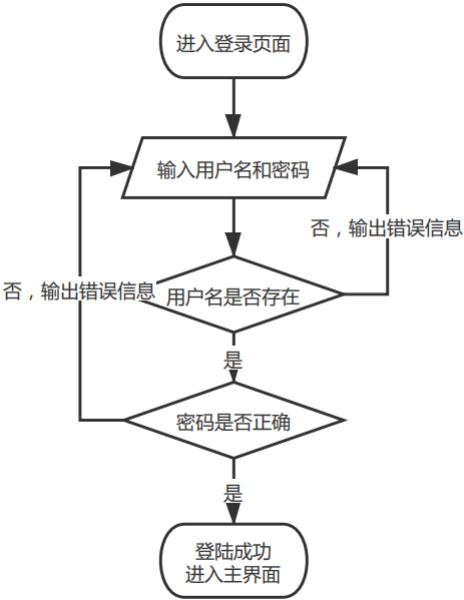
## 2.4.2 Web-商家端

### 申请注册

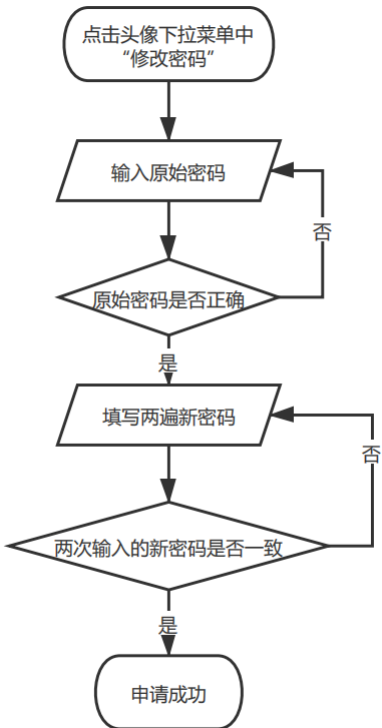
说明	未注册商家向平台申请注册，入驻平台
前提条件	商家未注册和登录
触发条件	在商家端登陆界面点击“申请注册”
流程	 <pre>graph TD; A([进入注册页面]) --&gt; B[/表单中填写个人信息/]; B --&gt; C{填写信息是否合法}; C -- "否, 重新填写" --&gt; B; C -- "是" --&gt; D[/提交申请/]; D --&gt; E([申请成功]);</pre>
后置条件	注册信息录入平台，等候管理员审核



登录

说明	商家登陆系统
前提条件	商家尚未登入系统，且商家已注册
触发条件	打开商家端网站即可
流程	<div><pre>graph TD; A([进入登录页面]) --&gt; B[/输入用户名和密码/]; B --&gt; C{用户名是否存在}; C -- 否, 输出错误信息 --&gt; B; C -- 是 --&gt; D{密码是否正确}; D -- 否, 输出错误信息 --&gt; B; D -- 是 --&gt; E([登陆成功 进入主界面]);</pre></div>
后置条件	获取登录的 token 及其他个人信息

## 修改密码

说明	商家修改自己在平台上账号的密码
前提条件	商家已经登入系统
触发条件	鼠标移至页面右上方头像，显示下拉菜单，点击其中的“修改密码”
流程	 <pre>graph TD; A([点击头像下拉菜单中 “修改密码”]) --&gt; B[/输入原始密码/]; B --&gt; C{原始密码是否正确}; C -- 否 --&gt; B; C -- 是 --&gt; D[/填写两遍新密码/]; D --&gt; E{两次输入的新密码是否一致}; E -- 否 --&gt; D; E -- 是 --&gt; F([申请成功]);</pre>
后置条件	无

维护商铺的个人信息

说明	商家查看并可修改自己店铺的详细信息，例如商铺名称、地址、店主姓名、联系方式等
前提条件	商家已经登入系统
触发条件	点击左侧导航栏中的“店铺信息”，或鼠标移至页面右上方头像，显示下拉菜单，点击其中的“个人信息”
流程	<div><pre>graph TD; A([点击“店铺信息” 进入页面]) --&gt; B[/查看个人信息/]; B &lt;--&gt; C[/修改个人信息 并提交/]; B --&gt; D([查看成功，结束]);</pre></div>
后置条件	无

维护商品信息

说明	商家查看并可维护自己的商品信息（增删改查）
前提条件	商家已经登入系统
触发条件	点击左侧导航栏中的“已有商品信息”
流程	<div><pre>graph TD; A([点击“已有商品信息”进入页面]) --&gt; B[/查看商品信息/]; B &lt;--&gt; C[/修改商品信息（增删改查）并提交/]; C --&gt; D([查看成功，结束]);</pre></div>
后置条件	无

发起交易

说明	在会员结账时，商家发起一笔交易
前提条件	商家已经登入系统
触发条件	点击左侧导航栏中的“发起交易”
流程	<pre>graph TD; A([点击“发起交易” 进入页面]) --&gt; B[/点选会员购买的商品 添加至购物车中/]; B --&gt; C[/输入会员提供 的交易密钥/]; C --&gt; D{密钥是否正确}; D -- 否 --&gt; C; D -- 是 --&gt; E([显示应支付金额 并等候会员完成支付]); E --&gt; F{会员是否完成了 实际支付}; F -- 否 --&gt; G[/点击“支付取消”/]; G --&gt; B; F -- 是 --&gt; H[/点击“支付完成”/]; H --&gt; I([显示订单详情 并结束]);</pre>
后置条件	输入选购商品的信息等一系列交易所需的操作

查看本商家的交易记录

说明	商家查看自己的交易历史记录
前提条件	管理员已经登入系统
触发条件	点击左侧导航栏中的“交易记录”
流程	<div><pre>graph TD; A([点击“交易记录” 进入页面]) --&gt; B[/查看交易记录/]; B --&gt; C([查看成功，结束]);</pre></div>
后置条件	无

发起结算申请

说明	由于交易历史中部分用户使用积分抵现，少收取的金额由商家和平台共同承担。商家定期内可向平台发起结算申请，申请平台将平台承担的金额部分转给商家
前提条件	商家已经登入系统，且尚没有未经审核的结算申请
触发条件	点击左侧导航栏中的“交易记录”，点击页面下方的“结算申请”
流程	<div><pre>graph TD; A([点击“交易记录” 进入页面]) --&gt; B[/点击“结算申请”/]; B --&gt; C{是否继续申请结算}; C -- 否 --&gt; B; C -- 是 --&gt; D{是否有尚未被管理员 处理的结算申请}; D -- 否 --&gt; E([申请成功]); D -- 是 --&gt; F([申请失败]);</pre></div>
后置条件	等候管理员审核结算申请

注销账号

说明	商家注销平台账号，终止与平台的合作
前提条件	商家已经登入系统
触发条件	点击左侧导航栏中的“商铺信息”，点击页面下方的“注销账号”
流程	<div><pre>graph TD; A([点击“店铺信息” 进入页面]) --&gt; B[/点击注销账号/]; B --&gt; C{确认注销账号?}; C -- 否 --&gt; A; C -- 是 --&gt; D([注销成功]);</pre><p>The flowchart illustrates the process of account cancellation. It begins with a rounded rectangle labeled '点击“店铺信息”进入页面' (Click 'Store Information' to enter the page). An arrow points down to a parallelogram labeled '点击注销账号' (Click 'Cancel Account'). Another arrow points down to a diamond-shaped decision box labeled '确认注销账号?' (Confirm account cancellation?). From the decision box, a path labeled '否' (No) loops back to the 'Click 'Store Information' to enter the page' step. A path labeled '是' (Yes) leads down to a final rounded rectangle labeled '注销成功' (Account cancellation successful).</p></div>
后置条件	无





### 2.4.3 微信小程序端

#### 授权、注册与登录

说明	会员第一次打开“融聚邻里”微信小程序时，先授权小程序获取自己的微信信息，然后注册成为线上的会员
前提条件	授权发生在使用者第一次打开小程序以及上次未同意授权时注册发生在使用者授权后且还未成为线上的会员时如果已经授权和注册，则直接完成登录
触发条件	点击“登录”按钮
流程	<pre>graph TD; A[进入登录界面] --&gt; B[点击“登录”按钮]; B --&gt; C{是否已授权}; C -- 否 --&gt; D[会员选择是否授权]; D -- 否 --&gt; A; D -- 是 --&gt; E{是否已注册}; E -- 是 --&gt; F[成功登录]; E -- 否 --&gt; G[会员自由选择头像]; G --&gt; H[会员输入昵称、邮箱、电话]; H --&gt; I{电话、邮箱格式是否正确}; I -- 否 --&gt; H; I -- 是 --&gt; F;</pre> <p>The flowchart illustrates the process for a user to log in or register in the WeChat Mini Program. It starts with the user entering the login interface and clicking the 'Login' button. A decision is made on whether the user is already authorized. If not, the user is prompted to authorize. If authorized, another decision is made on whether the user is already registered. If registered, the user proceeds to successful login. If not registered, the user is prompted to select a profile picture, enter their nickname, email, and phone number, and then a decision is made on whether the contact information format is correct. If correct, the user proceeds to successful login; otherwise, they are prompted to re-enter the information.</p>
后置条件	用户的数据信息和头像更新在数据库中，并实时传到本地缓存中

修改信息

说明	会员修改自己的昵称、邮箱和电话
前提条件	用户已登录
触发条件	用户切换到个人信息界面后，点击“修改信息”
流程	<div><pre>graph TD; A([进入信息修改界面]) --&gt; B[会员输入昵称、邮箱、电话]; B --&gt; C[会员点击“修改”或者“取消修改”按钮]; C --&gt; D{用户是否点击“取消修改”}; D -- 是 --&gt; E([回到用户信息界面]); D -- 否 --&gt; F{电话、邮箱格式是否正确}; F -- 是 --&gt; G([提示修改成功]); F -- 否 --&gt; B;</pre></div>
后置条件	会员新的昵称、电话、邮箱信息被存入数据库中，并在小程序端实时更新，存入本地缓存中。会员看到的是更新后的信息。

修改头像

说明	会员修改自己的头像
前提条件	会员已登录
触发条件	会员切换到个人信息界面后，点击自己头像旁边的“+”号
流程	<div></div>
后置条件	会员新的头像信息被存入数据库中，并在小程序端完成更新。更新后会员可以看到新的头像。

查看会员权限和积分

说明	会员查看当前的会员等级，以及当前等级下拥有哪些优惠
前提条件	会员已登录
触发条件	会员点击“会员信息”按钮
流程	<div><div>进入会员信息界面</div><div>↓</div><div>会员查看自己的等级、当前积分</div><div>↓</div><div>会员点击“查看权限”按钮</div><div>↓</div><div>会员点击“返回”按钮</div><div>↓</div><div>返回个人信息界面</div></div>
后置条件	无

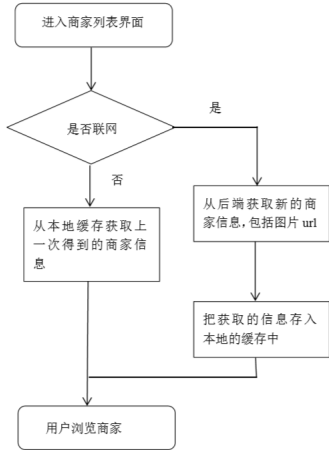
修改设置

说明	会员修改小程序设置，包括开启缓存、清空缓存、记录浏览历史等
前提条件	会员已登录
触发条件	会员点击“设置”按钮
流程	<div><div>进入设置界面</div><div>↓</div><div>会员通过一些按钮 和选择框完成设置</div><div>↓</div><div>会员点击“返回”按钮</div><div>↓</div><div>返回个人信息界面</div></div>
后置条件	无

查看订单信息

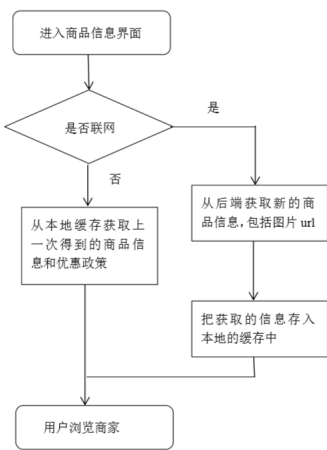
说明	会员查看历史订单记录
前提条件	会员已登录
触发条件	会员点击“交易历史”按钮
流程	<div><div>进入订单信息界面</div><div>↓</div><div>会员查看自己的交易历史信息</div><div>↓</div><div>会员点击“返回”按钮</div><div>↓</div><div>返回个人信息界面</div></div>
后置条件	无

## 浏览商家

说明	会员浏览商家列表
前提条件	会员已登录
触发条件	会员点击小程序底部的导航栏中的最左边按钮（商店的图标）
流程	 <pre>graph TD; A[进入商家列表界面] --&gt; B{是否联网}; B -- 是 --&gt; C[从后端获取新的商家信息, 包括图片 url]; B -- 否 --&gt; D[从本地缓存获取上一次得到的商家信息]; C --&gt; E[把获取的信息存入本地的缓存中]; E --&gt; F[用户浏览商家]; D --&gt; F;</pre>
后置条件	如果有联网，从后端更新的商家信息被存入本地缓存中



浏览商品

说明	会员浏览某个商店的商品列表，了解商品价格和本商店的优惠政策（折扣）
前提条件	会员已登录
触发条件	会员在商家列表中点击某个商家
流程	<div><pre>graph TD; A[进入商品信息界面] --&gt; B{是否联网}; B -- 是 --&gt; C[从后端获取新的商品信息, 包括图片 url]; B -- 否 --&gt; D[从本地缓存获取上一次得到的商品信息 and 优惠政策]; C --&gt; E[把获取的信息存入本地的缓存中]; E --&gt; F[用户浏览商家]; D --&gt; F;</pre></div>
后置条件	如果有联网，从后端更新的商品信息被存入本地缓存中

## 发起支付

说明	支付过程
前提条件	会员已登录
触发条件	会员点击小程序底部的导航栏中的中间按钮（支付的图标）
流程	<pre>graph TD; A[进入支付界面] --&gt; B[会员点击“发起支付”按钮]; B --&gt; C{是否联网}; C -- 否 --&gt; D[提示支付失败，原因是网络未连接]; C -- 是 --&gt; E[会员选择是否使用积分，接着获得后端发来的交易激活码，计时器打开]; E --&gt; F[会员在线下完成支付，此过程与小程序不相关]; F --&gt; G{是否超时}; G -- 是 --&gt; H[提示支付失败，原因是支付超时]; G -- 否 --&gt; I{商家是否确认交易已完成}; I -- 是 --&gt; J[提示交易成功，弹出本次订单信息]; I -- 否 --&gt; F;</pre>
后置条件	交易完成，数据库更新交易记录，把新的交易历史信息传给本地，存入本地缓存中

## 第 3 章. 体系结构设计

### 3.1 模块划分

我们将系统划分为以下模块:

- Web-管理员端

- 登录: 管理员登陆
- 会员信息: 查看所有注册的会员信息
- 商家信息: 查看所有合作商家的详细信息
- 历史订单: 查看平台所有历史订单信息
- 注册申请信息: 显示并审核商家的注册申请
- 结算申请信息: 显示并审核商家的结算申请
- 优惠及会员等级规则控制: 显示当前的优惠规则和会员等级规则, 同时可以修改规则

- Web-商家端

- 登陆注册: 登陆注册
- 个人信息: 查看、修改个人基础信息
- 商品信息: 查看、修改商家的商品信息
- 历史订单: 查看自己的历史订单, 并可发起结算申请
- 发起交易: 当有顾客结算时, 商家发起一笔交易申请并录入商品信息

## 3.2 系统行为描述

主要通过 Web-管理员模块、Web-商家模块、微信小程序（会员）模块等各画一个 UML 序列图来描述（分别如图 3.1、3.2、3.4所示）。

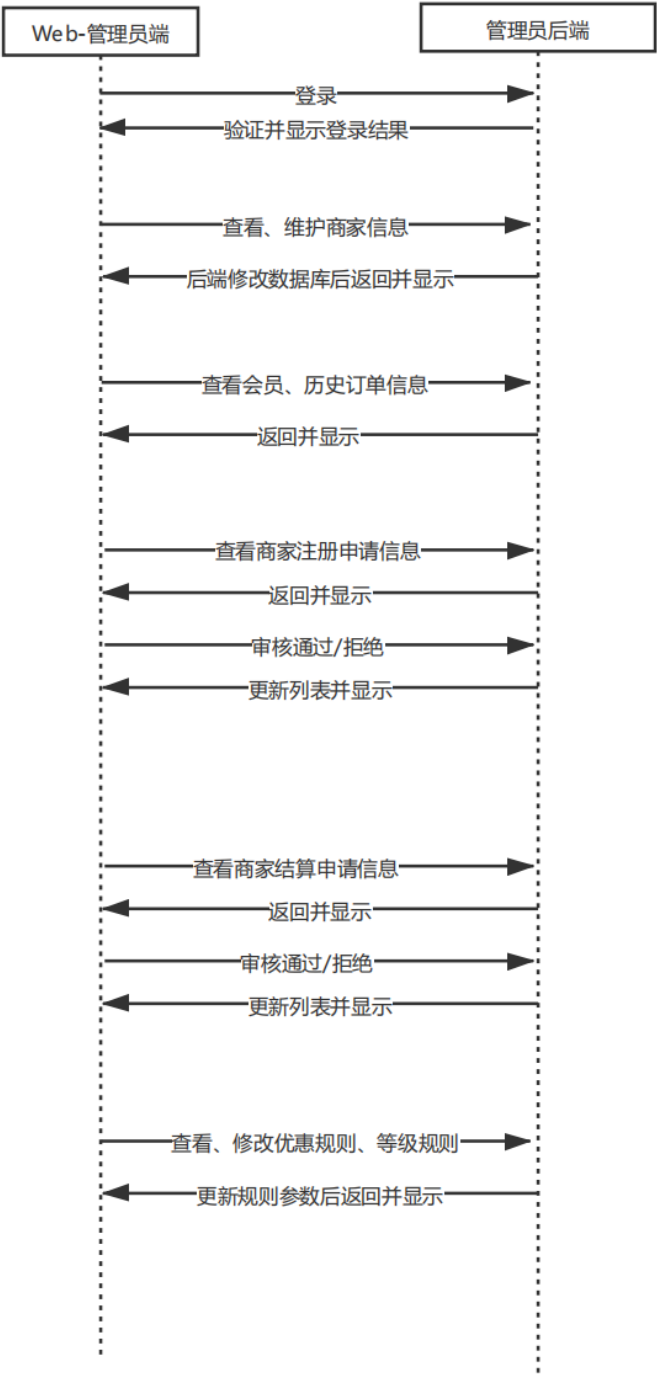


Figure 3.1: Web-管理员模块 UML 序列图

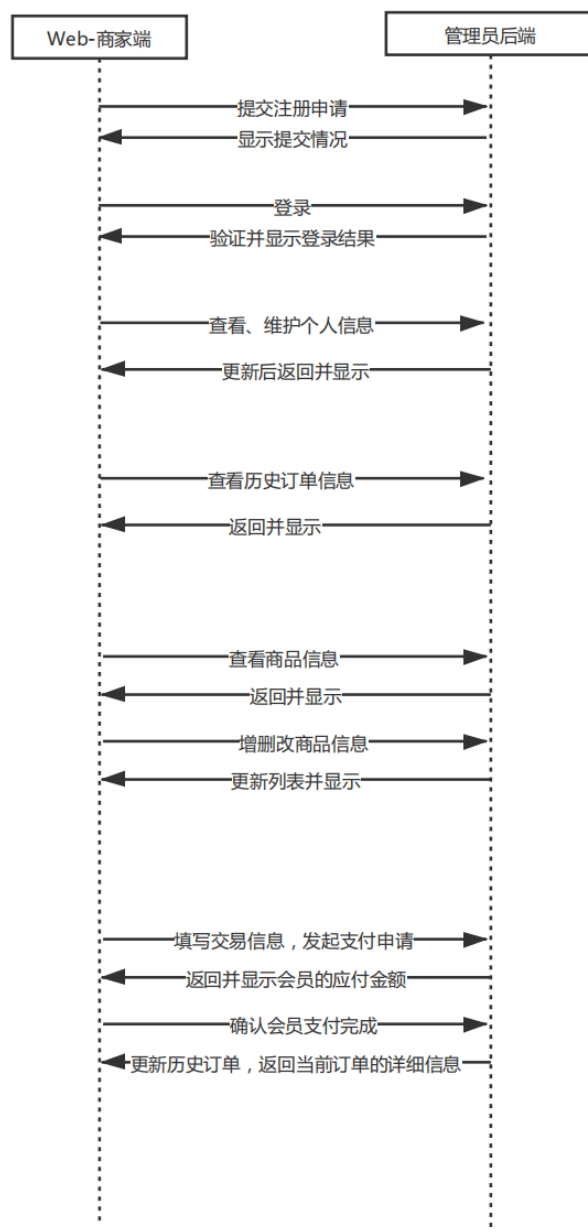


Figure 3.2: Web-商家模块序列图

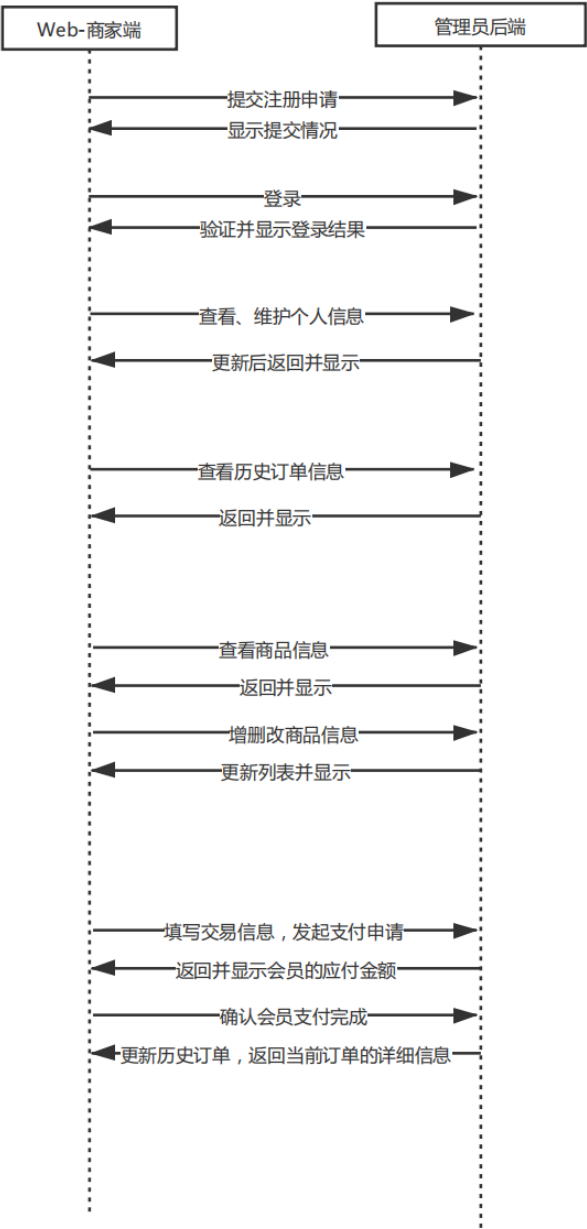


Figure 3.3: 微信小程序（会员）模块 UML 序列图

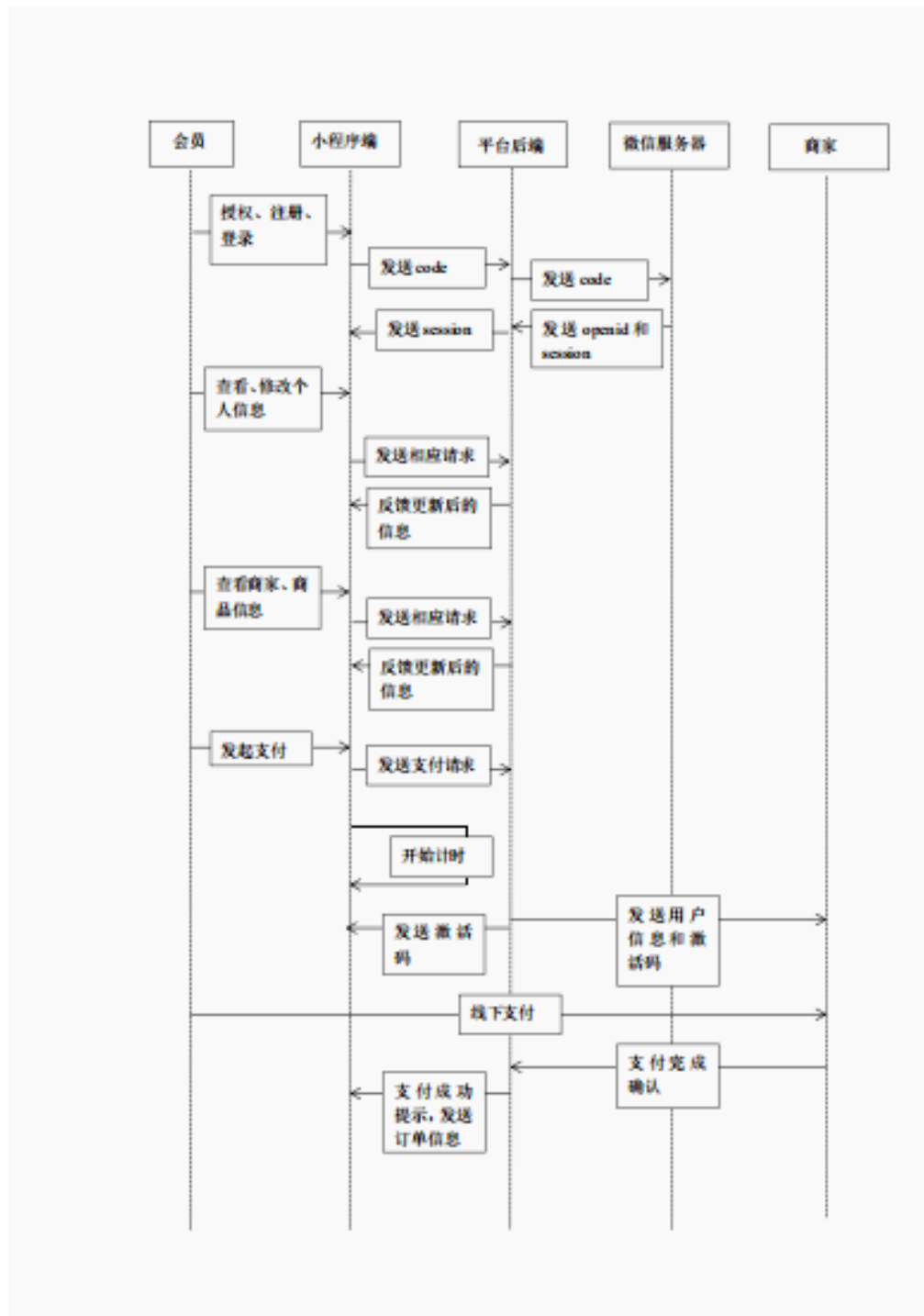


Figure 3.4: 微信小程序总体模块 UML 序列图



## 第 4 章. 模块间主要接口设计

### 4.1 微信小程序部分的前后端交互

微信小程序部分的 API 根 URL: /wx/。API 均通过 JSON 格式交换数据。传入及返回数据均为 JSON 对象。

此部分的 API 一览表如下:

URL	POST
/regist/	注册
/login/	登录
/getshop/	获取商家列表
/getgoodslist/	获取指定商家的商品列表
/wannashopping/	发起支付请求
/wannaorder/	获取订单信息
/uploadimage/	上传头像图片
/getmyinfo/	获取我的全部最新信息

**/regist/**

**POST** 注册 (第一次登录后将进行的完善信息的部分)

权限要求: 公开

参数:

名称	类型	要求	描述
session	字符串	微信官方的服务器生成	即时的登录凭证
nickname	字符串	中文/英文/符号，少于 12 个字符	昵称
email	字符串	标准的 email 格式	邮箱
phone	字符串	11 位数字	手机号码

返回（由于微信端只能接收和发送字符串，因此服务器将相关数据转化为字符串返回）：

名称	类型	描述
had	字符串	注册成功 =true，注册失败 =false
session	字符串	返回用户发送的 session 做验证
nickname	字符串	返回用户填写的 nickname
phone	字符串 11 位	返回用户填写的电话
email	字符串	返回用户的绑定 email
score	字符串	返回用户的初始积分 0
level	字符串	返回用户的初始等级 1（范围 1-5）
headimage	字符串	返回用户的头像的 url，如果用户未修改就是初始头像的 url
tag	字符串	返回根据用户最近的消费信息给用户打的标签

**/login/**

**POST 登录**

权限要求：公开

参数

名称	类型	要求	描述
code	字符串	小程序从微信官方服务器获取的认证码	认证码

返回：

名称	类型	描述
had	字符串	非第一次登录 =true，第一次登录 =false
session	字符串	返回用户发送的 code 对应的 session 作为用户的登录凭证
nickname	字符串	返回用户的 nickname
phone	字符串 11 位	返回的电话
email	字符串	返回用户的绑定 email
score	字符串	返回用户的积分
level	字符串	返回用户的初始等级 1（范围 1-5）
headimage	字符串	返回用户的头像的 url，如果用户未修改就是初始头像的 url
tag	字符串	返回根据用户最近的消费信息给用户打的标签

/getshop/

**POST** 获取商家列表

权限要求：公开

参数

名称	类型	要求	描述
session	字符串	用户登录成功时和用户绑定的认证码	认证码

返回：

名称	类型	描述
shoplist	字符串列表	商店名称的列表
shopidlist	字符串列表	商店 id 的列表
categorylist	字符串列表	商店所属类别的列表
betalist	字符串列表	商家对应的每积分可减免 RMB 的列表

名称	类型	描述
alphalist	字符串列表	商家对应的每 RMB 可以获得积分数的列表
shoplist	字符串列表	商店名称的列表

### /getgoodslist/

**POST** 获取指定商家的货物列表

权限要求：公开

参数

名称	类型	要求	描述
session	字符串	暂时的登录凭证需要有效	登录凭证
shopid	字符串		商家的唯一 id

返回：

名称	类型	描述
goodslist	一组元素的列表	认证不通过时返回 null

其中每组元素如下：

名称	类型	描述
goods_id	整型	商品 id
shop_id	整型	所在商店 id
goods_name	字符串	商品的名称
image	字符串	商品图片 url
price	双浮点型	商品价格
sale_price	双浮点型	商品的优惠价格

名称	类型	描述
is_sale	布尔型	是否特价

/getorderlist/

POST 获取该用户的所有订单列表

权限要求：公开

参数

名称	类型	要求	描述
session	字符串	暂时的登录凭证需要有效	登录凭证

返回：

名称	类型	描述
orderlist	一组元素的列表	认证不通过时返回 null

其中每组元素如下：

名称	类型	描述
skey	字符串	确定时间的唯一订单识别号
user_name	字符串	会员的昵称
start_time	字符串	订单开始时间
end_time	字符串	订单结束时间
shop_name	字符串	商家名称
shop_id	整型	商家的 id
delta	整型	用户在该订单的积分变化量
amount	字符串	此次的交易金额

名称	类型	描述
delta_receivables	双浮点数	商家应收的款项的变化量

### /wannashopping/

**POST** 客户端获取交易密钥 skey

权限要求：公开

参数

名称	类型	要求	描述
session	字符串	暂时的登陆凭证需有效	登录凭证
usescore	布尔型	到微信端自动转为字符串	是否使用积分

返回：

名称	类型	描述
skey	字符串	6 位的订单交易密钥
start_time	字符串	开始交易的时间

### /wannaorder/

**POST** 获取指定 skey 的订单

权限要求：公开

参数

名称	类型	要求	描述
session	字符串	暂时的登陆凭证需有效	登录凭证

名称	类型	要求	描述
skey	字符串	6 位字符串	订单在确定时间的唯一识别码

返回：

名称	类型	描述
skey	字符串	确定时间的唯一订单识别号
user_name	字符串	会员的昵称
start_time	字符串	订单开始时间
end_time	字符串	订单结束时间
shop_name	字符串	商家名称
shop_id	整型	商家的 id
delta	整型	用户在该订单的积分变化量
amount	字符串	此次的交易金额
delta_receivables	双浮点数	商家应收的款项的变化量

**/uploadimage/**

**POST** 上传头像的图片

权限要求：公开

参数

名称	类型	要求	描述
session	字符串	在 header 中的数据	登录凭证
file	MultipartFile	上传的文件	上传的图片文件，将会映射到某个 url

返回相关信息：

信息	描述
you do not login!	登录认证不成功
file can't be empty!	上传图片不存在
upload successfully!	上传图片成功

**/getmyinfo/**

**POST** 获取用户的最新信息，在更新头像/完成交易等场景下应当调用此接口来更新用户信息

权限要求：公开

参数

名称	类型	要求	描述
session	字符串	暂时的登录凭证	登录凭证

返回：

名称	类型	描述
had	布尔型	此为无效数据
session	字符串	返回用户发送的 code 对应的 session 作为用户的登录凭证
nickname	字符串	返回用户的 nickname
phone	字符串 11 位	返回的电话
email	字符串	返回用户的绑定 email
score	字符串	返回用户的积分
level	字符串	返回用户的初始等级 1（范围 1-5）
headimage	字符串	返回用户的头像的 url，如果用户未修改就是初始头像的 url
tag	字符串	返回根据用户最近的消费信息给用户打的标签



## 4.2 管理员 web 界面的前后端交互

API 根 URL: /managers/。API 均通过 JSON 格式交换数据。传入及返回数据均为 JSON 对象。

此部分的 API 一览表如下:

URL	POST
/login/	登录
/accept_register/	管理员同意新商家的注册申请
/reject_register/	管理员拒绝新商家的注册申请
/accept_receivables/	管理员同意商家的结算申请
/reject_receivables/	管理员拒绝商家的结算申请
/set_alpha/	管理员设置比例 1(用户获得积分)
/set_beta/	管理员设置比例 2(用户使用积分)
/set_rankrule/	管理员设置用户等级比例
/get_type/	管理员查看用户标签
/delete_shop/	管理员删除商家

URL	GET
/show_shops/	显示商家列表
/show_users/	显示会员列表
/show_registers/	显示待处理的注册申请商家列表
/show_orders/	显示订单列表
/show_receivables/	显示待处理的结算申请商家列表

/login/

POST 登录

权限要求：公开

参数：

名称	类型	描述
id	正整数	管理员的唯一标识
password	字符串	登录密码

返回：

名称	类型	描述
manager	字符串	登录成功则返回字符串形式的 ManagerInfo 对象。否则返回错误信息
token	字符串	返回用户的 token，浏览器缓存用作验证

/accept\_register/

**POST** 管理员同意新商家的注册申请

权限要求：token 验证

参数：

名称	类型	描述
shopName	字符串	同意申请的商家名

返回：

名称	类型	描述
shopId	正整数	商家的唯一标识
shopName	字符串	商家名字
password	字符串	密码
ownerName	字符串	商店店主名字

名称	类型	描述
phone	字符串	店主联系电话
address	字符串	商家地址
category	正整数	商家类型
rate	字符串	商家自定义的优惠比例
receivables	字符串	商家未结算的积分

**/reject\_register/**

**POST** 管理员拒绝新商家申请

权限要求: token 验证

参数:

名称	类型	描述
shopName	字符串	拒绝申请的商家名称

返回: null

**/accept\_receivables/**

**POST** 管理员同意商家的结算申请

权限要求: token 验证

参数:

名称	类型	描述
shopId	正整数	同意申请的商家 id

返回:

名称	类型	描述
status	字符串	"accept"

**/reject\_receivables/**

**POST** 管理员拒绝商家的结算申请

权限要求: token 验证

参数:

名称	类型	描述
shopId	正整数	拒绝申请的商家 id

返回:

名称	类型	描述
status	字符串	"reject"

**/set\_alpha/**

**POST** 管理员设置比例 1(用户获得积分的比例)

权限要求: token 验证

参数:

名称	类型	描述
id	正整数	管理员的唯一标识
alpha	正浮点数	会员获取积分比例的上限

返回：

名称	类型	描述
status	字符串	若为"alpha" 则设置成功；否则失败，数据不变

`/set_beta/`

**POST** 管理员设置比例 2（用户兑换积分的比例）

权限要求：token 验证

参数：

名称	类型	描述
id	正整数	管理员的唯一凭证
beta	正浮点数	兑换积分比例的上限

返回：

名称	类型	描述
status	字符串	若为"beta" 则设置成功；否则失败，数据不变

`/set_rankrule/`

**POST** 管理员设置用户等级比例

权限要求：token 验证

参数：

名称	类型	描述
id	正整数	管理员的唯一凭证
effectiveTime	正整数	用户积分有效时间

名称	类型	描述
money{1-4}	正浮点数	升入各等级所需消费金额
alpha{0-4}	正浮点数	各等级可获取积分的比例

返回：

名称	类型	描述
status	字符串	若为"rankrule" 则设置成功；否则失败，数据不变

/get\_type/

**POST** 管理员查看会员的标签

权限要求：token 验证

参数：

名称	类型	描述
openid	字符串	会员的临时凭证

返回：

名称	类型	描述
type	字符串	会员的类型标签

/delete\_shop/

**POST** 管理员删除商家

权限要求：token 验证

参数:

名称	类型	描述
shopId	正整数	删除的商家 id

返回:

名称	类型	描述
status	字符串	"delete"

**/show\_shops/**

**GET** 显示商家列表

权限要求: token 验证

返回:

名称	类型	描述
shopList	数组	每个元素为 ShopInfo 对象的字符串形式

**/show\_users/**

**GET** 显示会员列表

权限要求: token 验证

返回:

名称	类型	描述
shopList	数组	每个元素为 wxusrInfo 对象的字符串形式

/show\_orders/

**POST** 获取订单列表

权限要求：token 验证

返回：

名称	类型	描述
orderList	列表	认证不通过时返回 null

其中每组元素如下：

名称	类型	描述
skey	字符串	确定时间的唯一订单识别号
userName	字符串	会员的昵称
startTime	字符串	订单开始时间
endTime	字符串	订单结束时间
shopName	字符串	商家名称
shopId	整型	商家的 id
delta	整型	用户在该订单的积分变化量
amount	字符串	此次的交易金额
deltaReceivables	双浮点数	商家应收的款项的变化量

/show\_registers/

**POST** 获取待处理的注册申请商家列表

权限要求：token 验证

返回：



名称	类型	描述
registerList	列表	认证不通过时返回 null

其中每组元素如下:

名称	类型	描述
shopName	字符串	商家名字
password	字符串	密码
ownerName	字符串	商店店主名字
phone	字符串	店主联系电话
address	字符串	商家地址
category	正整数	商家类型

**/show\_receivables/**

**POST** 获取待处理的结算申请商家列表

权限要求: token 验证

返回:

名称	类型	描述
receivablesList	列表	认证不通过时返回 null

其中每组元素如下:

名称	类型	描述
shopName	字符串	商家名字
shopId	正整数	商家唯一标识
receivables	正浮点数	商家应收款项

## 4.3 商家 web 界面的前后端交互

API 根 URL: /shop/。API 均通过 JSON 格式交换数据。传入及返回数据均为 JSON 对象。

此部分的 API 一览表如下:

URL	POST
/login/	登录
/register/	注册
/receivables/	商家提交结算申请
/add_goods/	添加商品
/delete_goods/	删除商品
/update_goods/	修改商品属性
/update_goods_image/	修改商品图片
/upload_image/	修改商家头像
/change_password/	修改商家密码
/update/	修改商家其他信息
/delete/	商家注销账号
/show_goods/	显示该商家的商品列表
/show_orders/	显示该商家的订单列表

/login/

**POST** 登录

权限要求: 公开

参数:

名称	类型	描述
shopName	字符串	商家名字, 具有唯一性

名称	类型	描述
password	字符串	登录密码

返回：

名称	类型	描述
user	字符串	登录成功则返回字符串形式的 ShopInfo 对象。否则返回错误信息
token	字符串	返回用户的 token，浏览器缓存用作验证

### /register/

**POST** 商家提交注册申请

权限要求：公开

参数：

名称	类型	描述
shopName	字符串	商家名字
password	字符串	密码
ownerName	字符串	商店店主名字
phone	字符串	店主联系电话
address	字符串	商家地址
category	正整数	商家类型

返回：

名称	类型	描述
status	字符串	”wrong” 表示用户名已存在，”submit” 表示成功提交申请。

/receivables/

**POST** 商家提交结算申请

权限要求：token 验证

参数：

名称	类型	描述
shopId	正整数	商家 id

返回：

名称	类型	描述
status	字符串	"submit" 则修改成功，"wrong" 表示用户不存在，"exist" 表示已提交过且未处理

/add\_goods/

**POST** 商家新增商品

权限要求：token 验证

参数：

名称	类型	描述
shopId	正整数	商家 id
goodsName	字符串	商品名称
isSale	Boolean	是否为特价商品
price	正浮点数	商品原价
salePrice	正浮点数	商品售价

返回：

名称	类型	描述
status	字符串	"add_goods" 则添加成功, "repetition" 为重复添加某商品

### /delete\_goods/

**POST** 商家删除商品

权限要求: token 验证

参数:

名称	类型	描述
goodsId	字符串	商品 id

返回:

名称	类型	描述
status	字符串	"delete_goods" 则删除成功; "none" 则找不到商品

### /update\_goods\_image/

**POST** 修改商品图片

权限要求: token 认证

参数

名称	类型	要求	描述
token	字符串	在 Headers 中的数据	登录凭证
goodsId	字符串	在 Headers 中的数据	商品 id
file	MultipartFile	上传的文件	上传的图片文件, 将会映射到某个 url

返回相关信息：

信息	描述
you do not login!	登录认证不成功
the id of goods doesn't exit!	Headers 中找不到商品 id
the id of goods is wrong !	Headers 中的商品 id 在服务器中不存在
file can't be empty!	上传图片不存在
upload successfully!	上传图片成功

**/upload\_image/**

**POST** 修改商家头像

权限要求：token 认证

参数

名称	类型	要求	描述
token	字符串	在 Headers 中的数据	登录凭证
file	MultipartFile	上传的文件	上传的图片文件，将会映射到某个 url

返回相关信息：

信息	描述
you do not login!	登录认证不成功
file can't be empty!	上传图片不存在
upload successfully!	上传图片成功

**/update\_goods/**

**POST** 商家更改已有商品的属性

权限要求：token 验证

参数：

名称	类型	描述
shopId	正整数	商家 id
goodsId	正整数	商品 id，不可更改
goodsName	字符串	更改后的商品名称
isSale	Boolean	更改后是否为特价商品
price	正浮点数	更改后商品原价
salePrice	正浮点数	更改后商品售价

返回：

名称	类型	描述
status	字符串	为"update_goods" 则修改成功；否则失败，数据库不变

**/change\_password/**

**POST** 商家更改登录密码

权限要求：token 验证

参数：

名称	类型	描述
shopId	正整数	商家 id
password	字符串	验证原来的密码
newPassword	字符串	更改后的密码

返回：

名称	类型	描述
status	字符串	"update" 则修改成功, "wrong" 表示用户不存在或密码错误

### /update/

#### POST 商家修改商家其他信息

权限要求: token 验证

参数:

名称	类型	描述
shopId	正整数	商家 id, 不可修改
shopName	字符串	商家名字
password	字符串	密码, 需与原密码一致
ownerName	字符串	商店店主名字
phone	字符串	店主联系电话
address	字符串	商家地址
category	正整数	商家类型
rate	正浮点数	自定义的积分换现比例与上限 beta 的比例, 默认为 1

返回:

名称	类型	描述
status	字符串	"update" 则修改成功, "wrong" 表示用户不存在或密码错误

### /delete/

#### POST 商家注销账户

权限要求: token 验证



参数:

名称	类型	描述
shopId	正整数	商家 id

返回:

名称	类型	描述
status	字符串	"delete" 则注销成功, "wrong" 表示用户不存在

/show\_goods/

**POST** 显示该商家的商品列表

权限要求: token 验证

参数:

名称	类型	描述
shopId	正整数	商家 id

返回:

名称	类型	描述
goodsList	列表	若 id 不存在则返回 null

其中每组元素如下:

名称	类型	描述
shopId	正整数	商家 id
goodsId	正整数	商品 id

名称	类型	描述
goodsName	字符串	商品名称
isSale	Boolean	是否为特价商品
price	正浮点数	商品原价
salePrice	正浮点数	商品售价

**/show\_orders/**

**POST** 获取该商家的订单列表

权限要求：token 验证

参数：

名称	类型	描述
shopId	正整数	商家 id

返回：

名称	类型	描述
orderList	列表	id 不存在时返回 null

其中每组元素如下：

名称	类型	描述
skey	字符串	确定时间的唯一订单识别号
userName	字符串	会员的昵称
startTime	字符串	订单开始时间
endTime	字符串	订单结束时间
shopName	字符串	商家名称

名称	类型	描述
shopId	整型	商家的 id
delta	整型	用户在该订单的积分变化量
amount	字符串	此次的交易金额
deltaReceivables	双浮点数	商家应收的款项的变化量

## 4.4 商家端交易流程的前后端交互

API 根 URL: `/order/`。API 均通过 JSON 格式交换数据。传入及返回数据均为 JSON 对象。

此部分的 API 一览表如下:

URL	POST
<code>/calculate/</code>	第一次交互, 计算金额及积分
<code>/close/</code>	商家端取消订单
<code>/receivables/</code>	商家端结束订单, 线下支付完成

`/calculate/`

**POST** 根据所选商品计算应付金额及积分变化量

权限要求: token 验证

参数:

名称	类型	描述
skey	字符串	订单信息的唯一标识
shopId	正整数	商家 ID
originSum	正浮点数	原价商品金额, 用于积分互换

名称	类型	描述
saleSum	正浮点数	特价商品金额

返回：

名称	类型	描述
total	正浮点数	会员应付金额
delta	整数	积分变化量
status	字符串	”calculate” 表示成功，”none “表示不存在此订单

**/close/**

**POST** 商家端取消订单

权限要求：token 验证

参数：

名称	类型	描述
skey	字符串	订单信息的唯一标识

返回：

名称	类型	描述
status	字符串	”close” 表示取消成功，”none “表示不存在此订单

**/finish/**

**POST** 商家端结束订单，交易完成

权限要求：token 验证

参数:

名称	类型	描述
skey	字符串	订单信息的唯一标识

返回:

名称	类型	描述
skey	字符串	确定时间的唯一订单识别号
userName	字符串	会员的昵称
startTime	字符串	订单开始时间
endTime	字符串	订单结束时间
shopName	字符串	商家名称
shopId	整型	商家的 id
delta	整型	用户在该订单的积分变化量
amount	字符串	此次的交易金额
deltaReceivables	双浮点数	商家应收的款项的变化量
status	字符串	”finish” 表示成功, ”none “表示不存在此订单



## 第 5 章. 各模块的主要设计

### 5.1 Web-管理员端

#### 5.1.1 总体页面 App

前端总体页面类 App 是一个 Vue 组件，负责架构起整个前端页面，激活到各个子组件的路由跳转，并提供了各个组件通用的 reload 方法，便于实现各个页面的刷新操作。App 组件为整个前端的最顶层框架，由 main.js 文件创建并调用，同时在 main.js 中实现了验证 token 所需的对后端发出请求的拦截器，和判断交互状态所需的接收后端响应的拦截器。

#### Methods

方法名	参数	功能
reload	无	实现各个页面的刷新操作

#### 5.1.2 store

store 用来在运行过程中在浏览器中存储信息并供其他部分使用，该模块基于 Vuex store 实现。在这里我们只使用了 state 成员用于存储具体的变量，其他部分引用和修改时，直接使用即可。其内容如下：

state

属性名称	描述
baseurl	管理员端维护信息时与后端交互的后端主接口
info	管理员的基本信息，包含 ID、积分比例政策和用户等级规则
token	登录后随机分配的令牌，用于判断登陆状态

### 5.1.3 router

router 用于记录前端所能够到达的路径及其对应的 vue 组件，实现了路由跳转。同时在 router 实现了页面刷新时将 store 内容保存至浏览器 sessionStorage 再取出并应用于新页面的操作，防止页面刷新时 store 清空、数据丢失的情况。

### 5.1.4 ManagerLogin

ManagerLogin 为一个 vue 组件，是管理员端显示的第一个页面。该组件实现的功能为管理员的登录。

#### data

属性名称	描述
manager	登录框内表单的输入信息，包含 id 和 password 两个属性
now_bg_index	当前背景图片的序号，用于背景图片的轮播更换
main_imgs	背景图片组的路径信息

#### Methods

方法名	参数	功能
timer	无	计时器，用于背景图片的定时轮播更换



方法名	参数	功能
manager_login	无	向/login/发送登录信息，并根据后端响应结果页面跳转（登录成功）或显示错误信息（登陆失败）

### 5.1.5 ManagerIndex

ManagerIndex 为一个 vue 组件，是管理员端登陆后的主页面，其内部显示其他 vue 组件渲染出的页面，具有导航性质。

#### data

属性名称	描述
isCollapse	左侧导航栏是否展开

#### Methods

方法名	参数	功能
collapseOpen	无	左侧导航栏展开
collapseClose	无	左侧导航栏折叠
handleSelect	点选的导航栏中的键及对应子组件的路径	
handleDropdownCommand	无	对点选头像的下拉菜单中某一项的处理

### 5.1.6 ResidentInfo

ResidentInfo 是一个 vue 组件，用于显示和管理平台上的注册用户（即“会员”）的基本信息。

## data

属性名称	描述
tableData	会员的基本信息列表

### 5.1.7 ShopInfo

ShopInfo 是一个 vue 组件，用于显示和管理平台上的注册商家的基本信息。

## data

属性名称	描述
tableData	合作商家的基本信息列表
getSearchTable	关键词搜索后的商家信息列表
searchInfo	搜索栏中的关键词

## Methods

方法名	参数	功能
deleteRow	选中的行和行序号	删除某商家

## computed

方法名	参数	功能
searchTableData	无	根据搜索栏中的关键词计算应该显示的商家列表

### 5.1.8 ShopRegisterInfo

ShopRegisterInfo 是一个 vue 组件，用于审核和处理未注册商家的注册申请。

#### data

属性名称	描述
tableData	商家注册申请时填写的基本信息列表

#### Methods

方法名	参数	功能
accept	选中的申请列表中的行序号、申请列表	同意该商家的注册申请
reject	选中的申请列表中的行序号、申请列表	拒绝该商家的注册申请

### 5.1.9 OrderInfo

OrderInfo 是一个 vue 组件，用于显示和管理平台上发生的所有交易记录。

#### data

属性名称	描述
tableData	交易记录的详细信息列表
getSearchTable	关键词搜索后的交易记录列表
searchInfo	搜索栏中的关键词

## computed

方法名	参数	功能
searchTableData	无	根据搜索栏中的关键词计算应该显示的交易记录列表

### 5.1.10 ReceivablesInfo

ReceivablesInfo 是一个 vue 组件，用于审核和处理合作商家的结算申请。

## data

属性名称	描述
tableData	商家结算申请信息列表

## Methods

方法名	参数	功能
accept	选中的申请列表中的行序号、申请列表	同意该商家的结算申请
reject	选中的申请列表中的行序号、申请列表	拒绝该商家的结算申请

### 5.1.11 DiscountRule

DiscountRule 是一个 vue 组件，用于管理和设定平台优惠政策。例如适用于全平台的积分获取比例上限，积分消费比例上限，以及会员等级规则（包括各级对应的积分获取比例、到达各级所需的平台消费额、消费额有效期等）。注意，此处的“积分获取比例”是指：购买原价商品 1 元所得积分，不同会员等级下该比例不同，一般与会员

的等级成正相关；“积分消费比例上限”是指：购买原价商品时使用 1 积分的抵消金额，一般不同商家下该比例不同，是由商家自己制定的。

#### data

属性名称	描述
creditRuleForm	存储管理员输入的积分获取和消费比例上限的表单
rankRuleForm	存储管理员输入的会员等级规则参数的表单

#### Methods

方法名	参数	功能
changeAlpha	无	修改全平台的会员积分获取比例上限值
changeBeta	无	修改全平台的会员积分消费比例上限值
changeRankRule	无	修改用户等级规则及优惠规则

## 5.2 Web-商家端

### 5.2.1 总体页面 App

前端总体页面类 App 是一个 Vue 组件，负责架构起整个前端页面，激活到各个子组件的路由跳转，并提供了各个组件通用的 reload 方法，便于实现各个页面的刷新操作。App 组件为整个前端的最顶层框架，由 main.js 文件创建并调用，同时在 main.js 中实现了验证 token 所需的对后端发出请求的拦截器，和判断交互状态所需的接收后端响应的拦截器。

#### Methods

方法名	参数	功能
reload	无	实现各个页面的刷新操作

### 5.2.2 store

store 用来在运行过程中在浏览器中存储信息并供其他部分使用，该模块基于 Vuex store 实现。在这里我们只使用了 state 成员用于存储具体的变量，其他部分引用和修改时，直接使用即可。其内容如下：

#### state

属性名称	描述
baseurl	商家端维护信息时与后端交互的后端主接口
tradeurl	商家端做交易时与后端交互的后端主接口
info	商家的基本信息，包含店名、店主、联系电话、头像等
token	登录后随机分配的令牌，用于判断登陆状态

### 5.2.3 router

router 用于记录前端所能够到达的路径及其对应的 vue 组件，实现了路由跳转。同时在 router 实现了页面刷新时将 store 内容保存至浏览器 sessionStorage 再取出并应用于新页面的操作，防止页面刷新时 store 清空、数据丢失的情况。

### 5.2.4 ShopLogin

ShopLogin 为一个 vue 组件，是商家端实质上能够看到的第一个页面。该组件实现的功能有商家登录和注册页面的跳转。

#### data

属性名称	描述
shop	登录框内表单的输入信息，包含 shopName 和 password 两个属性
now_bg_index	当前背景图片的序号，用于背景图片的轮播更换
main_imgs	背景图片组的路径信息

## Methods

方法名	参数	功能
timer	无	计时器，用于背景图片的定时轮播更换
shop_login	无	向/login/发送登录信息，并根据后端响应结果页面跳转（登录成功）或显示错误信息（登陆失败）
shop_register	无	跳转到注册界面

### 5.2.5 ShopRegister

ShopRegister 为一个 vue 组件，用于未注册商家填写基本信息并申请注册，也可跳回至登录界面。

## data

属性名称	描述
form	注册框内表单的输入信息，包含商店名称、商店类别、店主姓名和电话、店铺地址、密码六个属性
password2	注册时第二次输入的密码，用于密码一致性检验
cateOptions	商店类别列表，将汉字与数组序号对应起来
now_bg_index	当前背景图片的序号，用于背景图片的轮播更换

属性名称	描述
main_imgs	背景图片组的路径信息

## Methods

方法名	参数	功能
timer	无	计时器，用于背景图片的定时轮播更换
submitRegister	无	检查表单信息合法性后，向/register/发送登录信息，并根据后端响应结果提示注册成功或注册失败
toLogin	无	跳转到登录界面

### 5.2.6 ShopIndex

ShopIndex 为一个 vue 组件，是商家端登陆后的主页面，其内部显示其他 vue 组件渲染出的页面，具有导航性质。

## data

属性名称	描述
isCollapse	左侧导航栏是否展开
isChangingPw	是否正在修改密码
changePwForm	修改密码的表单，用于核验原密码后修改新密码

## Methods



方法名	参数	功能
collapseOpen	无	左侧导航栏展开
collapseClose	无	左侧导航栏折叠
handleSelect	点选的导航栏中的键及对应子组件的路径	
handleDropdownCommand	无	对点选头像的下拉菜单中某一项的处理
changePw_cancel	无	取消修改密码
changePw_confirm	无	确认修改密码

### 5.2.7 MyInfo

MyInfo 为一个 Vue 组件，可以查看和修改维护商家个人信息。

#### data

属性名称	描述
myInfo	商家个人信息，包括 ID，店名等 9 个属性
editVisible	修改个人信息的表单是否可见
editForm	存储输入的修改信息的表单内容

#### Methods

方法名	参数	功能
editMyInfo	无	显示修改个人信息的表单
edit_info_confirm	无	向商家确认修改操作并向/update/发送修改的表单信息
edit_info_cancel	无	取消修改操作
deleteAccount	无	注销账号

方法名	参数	功能
handleAvatarSuccess	上传头像的后端返回	判断头像上传是否成功并更新显示的头像

#### computed

方法名	参数	功能
uploadHeader	无	在上传头像的请求头中添加 token

### 5.2.8 GoodsInfo

GoodsInfo 是一个 Vue 组件，用于商家查看并维护商品信息，可对商品进行增删改查操作。

#### data

属性名	功能
add_checksaleprice	表单合法性检查的规则——添加商品时折扣价格应低于原价
edit_checksaleprice	表单合法性检查的规则——修改商品信息时折扣价格应低于原价
tableData	完整的商品信息列表
getSearchTable	关键词搜索后的商品信息列表
addFormVisible	添加商品的表单是否可见
addForm	添加商品的表单
addRules	对上文表单合法性检查的应用
editFormVisible	修改商品的表单是否可见
editForm	修改商品的表单
editRules	对上文表单合法性检查的应用
editIndex	修改的商品在原完整商品列表中的序号

属性名	功能
searchInfo	搜索栏中的关键词
isManaging	当前是否处于编辑（管理）状态
currentPage	当前页号（用于分页显示）
pageSize	每一页显示商品的条数（用于分页显示）

## Methods

方法名	参数	功能
delete_goods	无	删除商品
manage_goods	无	在是否管理（编辑）商品状态之间切换
add_goods	无	弹出添加商品的表单，并添加商品
add_goods_cancel	无	取消添加商品
add_goods_confirm	无	确认添加商品
edit_goods	无	弹出修改商品的表单，并修改某件商品
edit_goods_cancel	无	取消修改商品
edit_goods_confirm	无	确认修改该商品
updateImageSuccess	上传商品图片的后端返回	判断商品图片上传是否成功并更新显示的图片
isSaleFormatter	无	将商品是否折扣的布尔变量做映射，以字符串形式显示在表格中

## computed

方法名	参数	功能
updateHeader	无	在上传商品图片的请求头中添加 token 和对应商品的 ID
searchTableData	无	计算搜索后得到的直接用于显示的表格

### 5.2.9 StartTransaction

StartTransaction 是一个 Vue 组件，用于商家发起一笔交易，其中具体需要商家点选购买的商品、输入交易密钥、确认收款。

#### data

属性名	功能
tableData	完整的待选商品信息列表
baskets	购物车
originSum	所购商品按原价售卖的总价
saleSum	所购商品按折扣价格的总价
skey	交易密钥
needKey	当前是否需要显示密钥输入框
basketVisible	购物车表单是否可见
orderVisible	显示订单信息的对话框是否可见
order	订单详情
finalVisible	交易结束后的订单信息对话框是否可见
final	交易结束后的完整订单信息
currentPage	当前页号（用于分页显示）
pageSize	每一页显示商品的条数（用于分页显示）

#### Methods

方法名	参数	功能
ListChange	改变数量的商品在商品列表中的行	商品列表中改变商品数量
BasketChange	改变数量的商品在购物车列表中的行	购物车中改变商品数量

方法名	参数	功能
seebaskets	无	查看购物车
noseebaskets	无	关闭购物车显示
submit	无	提交订单
cancel	无	取消提交订单，并返回购物车界面
confirm	无	商品信息填写完毕，需要显示密钥输入框
emptybaskets	无	清空购物车
completeOrder	无	用户支付完成，商家确认订单成功完成
cancelOrder	无	用户取消支付，订单中止
saleFormatter	无	将商品是否折扣的布尔变量做映射，以字符串形式显示在表格中

### computed

方法名	参数	功能
originsum	无	得出非折扣商品总价
salesum	无	得出折扣商品总价
sum	无	得出商品总价

### 5.2.10 TransactionHistory

TransactionHistory 为一个 vue 组件，用于查看该商家的历史订单，并可向平台申请结算。

### data

属性名称	描述
tableData	历史订单列表

Methods

方法名	参数	功能
settlement	无	向平台发出结算申请，等候管理员审批

5.3 后端

5.3.1 总体架构

后端的总体架构如图 5.3 所示：

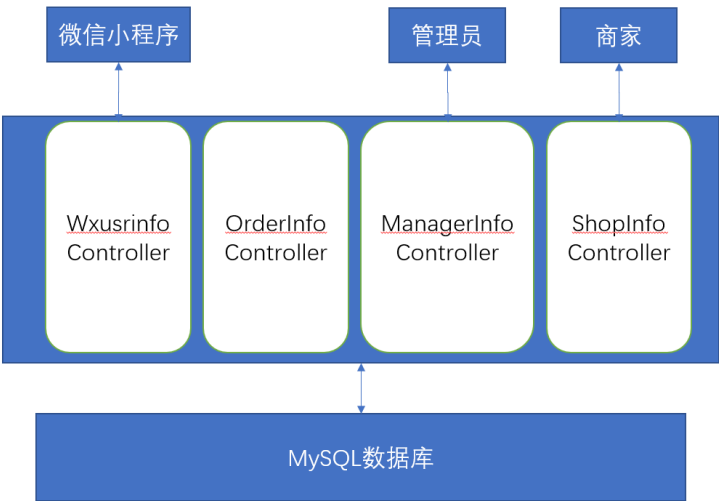


Figure 5.1: 后端总体架构

首先，后端分为 MySQL 数据库服务器、微信小程序/管理员/商家后端逻辑服务器和图片库服务器（后两者是同一个实例），中间通过 Nginx 进行 https 加密。客户端可以通过直接的请求和数据库服务器、三端后端服务器和图片库服务器进行直接的交互。

其中三端后端逻辑服务器将会直接和 MySQL 数据库服务器通过 Springboot 和 Mybatis plus 的映射进行通信，实现信息的储存。主要逻辑在 controller 类中实现。

### 5.3.2 微信小程序后端

微信小程序后端搭建于 linux+jdk1.8 + Tomcat7 + mysql 5.6 + spring boot2 + mybatis plus 环境，并依赖于如表 5.38 所示的包。

**Table 5.38:** 微信小程序后端依赖的主要包（另外两个后端与之相同）

包名	版本	说明
java-jwt	3.4.0	Token 验证
maven-compiler-plugin	1.8	maven 框架
spring-boot-starter-thymeleaf	自动选择	模板引擎
mysql-connector-java	自动选择	mysql 驱动
mybatis-plus-boot-starter	3.2.0	mybatis plus 依赖
mybatis-plus-generator	3.2.0	mybatis-plus 的框架生成器
jacoco-maven-plugin	0.8.1	代码覆盖率相关
lombok	1.18.0	通过注解自动填写 set 和 get 方法
spring-boot-starter-test	自动选择	单元测试
spring-boot-starter-web-services	自动选择	springboot 单元测试环境
junit	4.12	单元测试

#### WxusrinfoController

方法名	参数	功能
userLogin	LoginRequest	通过和微信服务器交互，实现登录，返回用户信息和庸碌登录凭证
userRegist	RegistRequest	通过 session 认证，如若未注册补充相关信息，返回用户信息
getShop	UniversalRequest	通过 session 认证，获取商家信息
getGoodsList	OrderListRequest	通过 session 认证并指定商家，获取商家的商品信息
getOrderList	UniversalRequest	通过 session 认证，获取该会员的所有订单信息

方法名	参数	功能
wannaShopping	WannaShoppingRequest	通过 session 认证并且是否使用积分，建立订单
wannaOrder	WannaOrderRequest	通过 session 认证，凭借唯一 skey 获得刚刚成交的订单信息
uploadImage	MultipartFile	通过 Headers 里的 session 认证，上传图片到服务器
getMyInfo	UniversalRequest	通过 session 认证，获取会员的最新信息

这是本部分在/wx/的前缀下的请求地址和相关映射的函数，通过 @PostMapping 注解实现本部分为用户请求和对应的路由函数。

请求地址	映射函数
login	WxusrinfoController.userLogin
regist	WxusrinfoController.userRegist
getshop	WxusrinfoController.getShop
getgoodslist	WxusrinfoController.getGoodsList
getorderList	WxusrinfoController.getOrderList
wannashopping	WxusrinfoController.wannaShopping
wannaorder	WxusrinfoController.wannaOrder
uploadimage	WxusrinfoController.uploadImage
getmyinfo	WxusrinfoController.getMyInfo

### 5.3.3 商家对应后端

本部分为商家 web 端操作对应的后端逻辑架构。

#### ShopInfoController



方法名	参数	功能
userLogin	shopName, password	商家端登录, 获取 token
userRegister	shopInfo	新商家填写信息, 提交注册申请, 加入数据库
receivables	shopId	商家提交结算申请, 已有未处理申请的商家不可重复提交
changePassword	shopName, password, newPassword	商家修改密码, 需验证原密码
update	shopInfo	商家修改其他信息
showGoods	shopId	显示该商家商品列表
showOrders	shopId	显示该商家订单列表
updateGoods	shopId	更改数据库里的商品信息
deleteGoods	shopId	删除数据库里的商品
addGoods	shopId	向数据库添加商品
delete	shopId	删除数据库里的商家信息

## OrderInfoController

方法名	参数	功能
calculate	skey, shopId, originSum, saleSum	根据商家输入的信息和订单信息, 计算应付金额和积分变化量, 完善订单信息
close	skey	商家关闭订单, 从数据库中删除
finish	skey	线下支付完成, 修改数据库中订单状态

## Config

类名	功能
CorsConfig	解决跨域访问的问题

类名	功能
InterceptorConfig	拦截器，验证 token

### 5.3.4 管理员对应后端

本部分为管理员 web 端操作对应的后端逻辑架构。

#### ManagerInfoController

方法名	参数	功能
login	id, password	管理员登录，获取 token
acceptRegister	shopInfo	管理员同意注册商家，修改数据库
rejectRegister	shopInfo	管理员拒绝注册商家，修改数据库
acceptReceivables	shopId	同意结算申请，商家积分变化
rejectReceivables	shopId	拒绝结算申请
deleteShop	shopId	从数据库中删除商家信息
showShops	null	显示数据库中所有商家
showOrders	null	显示数据库中所有订单
showReceivables	null	显示数据库中所有待处理的结算申请
showusers	null	显示数据库中所有会员信息
showRegisters	null	显示数据库中所有待处理的注册申请
setAlpha	id, alpha	更改获取积分比例
setBeta	id, beta	更改兑换积分比例
setRankRule	id, beta	更改会员等级等规则
getType	openid	获取指定会员的标签

## 5.4 微信小程序端

### 5.4.1 开发目录

为了简化开发，微信小程序的框架已经为每个页面分好了一个模块，每个模块中含有配置文件、布局文件、样式文件和脚本文件（基于 JavaScript）。此外，有一个用于设置全局信息的全局模块，以及一些外部导入的辅助模块。本系统的开发目录如下图所示：

### 5.4.2 全局模块

全局模块用于存储小程序的全局变量，以及在打开小程序时异步进行的一些操作。

data

变量	功能
userinfo	Object 类型（类似字典），存储会员的微信数据，包括微信昵称、微信的头像路径等
userdata	Object 类型，存储会员在“融聚邻里”平台中的个人信息，包括头像 url、昵称、电话、邮箱、积分、会员等级等
hasAuth	会员是否已经授权，可以通过微信提供的 API 获得相关信息，授权在一段时间后会过期，需要重新授权
enableStore	会员是否开启本地缓存

app.js	全局的 js 文件，存储全局变量，和启动时的加载项
app.json	全局的配置文件，包括页面的索引、外部模块的导入、底部导航栏的设置等
app.wxss	全局的样式文件，包括每个页面的与边框的距离、默认的字大小等
pages	所有的页面
index	启动时界面，包括欢迎信息和“融聚邻里” logo
index.wxml	启动界面的布局
index.js	启动界面的脚本，设置了启动页面的存在时间
index.json	启动界面的配置，包括导航栏标题、不使用外部模块等
index.wxss	启动界面的样式表，包括 logo 大小、文字样式等
start	登录界面，包含登录按钮
start.wxml	
start.js	
start.json	
start.wxss	
regist	注册界面
regist.wxml	
regist.js	
regist.json	
regist.wxss	
my	个人信息界面
my.wxml	
my.js	
my.json	
my.wxss	
shoplist	商家列表界面，含有分类导航栏
shoplist.wxml	
shoplist.js	
shoplist.json	
shoplist.wxss	
shopdetail	商品详情界面，展示商品价格、优惠政策等
shopdetail.wxml	
shopdetail.js	
shopdetail.json	
shopdetail.wxss	
pay	支付界面，包含支付按钮和支付指导
pay.wxml	
pay.js	

Figure 5.2: 开发目录一

pay.json	
pay.wxss	
edit	个人信息修改界面
edit.wxml	
edit.js	
edit.json	
edit.wxss	
order	历史订单界面
order.wxml	
order.js	
order.json	
order.wxss	
score	积分信息和会员权限界面
score.wxml	
score.js	
score.json	
score.wxss	
setting	设置界面
setting.wxml	
setting.js	
setting.json	
setting.wxss	
components	外部导入的模块
loadmore	下拉加载时动画
navigator	顶部导航栏
images	所有内嵌的图片，主要是一些 logo 和 icon
utils	辅助函数，包括日期与时间的格式化函数等

Figure 5.3: 开发目录二（接上图）

#### method

方法名	功能
onLaunch	完成一些默认设置，比如打开缓存、异步获取用户的微信信息

### 5.4.3 index

小程序的首页，是小程序开启时显示的欢迎界面。

#### method

方法名	功能
onLaunch	检测 session 的时效性，失效时需要异步发送用户信息给服务器换取新的 session；开启计时器，两秒后跳转
changePage	选择切换到登录界面还是个人信息界面

### 5.4.4 start

登录界面，含有登录按钮。

#### data

变量	功能
shouldAppearAuthorizationButton	按钮是否开启认证询问

变量	功能
canIUse	是否支持小程序的 getUserInfo 功能

## method

方法名	功能
onLaunch	通过小程序的 API 获取认证信息和会员的微信信息
login	通过小程序的 API 登录到微信服务器，获得服务器提供的 code；用 code 与后端通信，判断是否注册
changePage	已注册直接进入商家列表页；未注册进入注册界面
getUserInfo	当前版本不支持小程序 getUserInfo 功能时，进行手动的信息获取

## 5.4.5 regist

### 注册界面

## data

变量	功能
imageUrl	图片的本地链接
emailHolder	输入邮箱的提示信息
phoneHolder	输入电话的提示信息

变量	功能
nicknameHolder	输入昵称的提示信息
nicknameValue	昵称初始值，为会员的微信昵称
nickname	昵称
phone	电话
email	邮箱
registDisable	是否允许注册
phoneWarnInfo	电话格式出错提醒
emailWarnInfo	邮箱格式出错提醒
hasPhoneWrong	电话格式是否出错
hasEmailWrong	邮箱格式是否出错

#### method

方法名	功能
PhoneBlur	输入电话失去焦点
EmailBlur	输入邮箱失去焦点
NicknameInput	输入昵称时的响应事件
PhoneInput	输入电话时的响应事件
EmailInput	输入邮箱时的响应事件
chooseImage	选择图片上传
judgeBtnDisable	格式出错时不允许点击按钮
judgeCorrect	输入格式的正确性判断

#### 5.4.6 my

个人信息界面。

**data**

变量	功能
imgUrl	服务器提供的头像链接
nickname	昵称
tag	服务器提供的会员称号

**method**

方法名	功能
onLoad	获取头像、昵称、称号信息
onclick	点击头像时进入个人信息界面

**5.4.7 shoplist**

商家列表界面。

**data**

变量	功能
hasData	本分类中是否有商家
shopImageUrl	没有商家时的图片提示
shops	所有的商家列表
showshops	当前分类下的商家列表
currentTab	当前 tab 页
tlist	分类列表
page	当前 tab 页的页码数



变量	功能
slideWidth	滑块宽
slideLeft	滑块位置
totalLength	当前滚动列表总长

#### method

方法名	功能
onLoad	计算顶部导航栏比例
onPullDownRefresh	下拉刷新
onReachBottom	到达底部
getRatio	计算每个滑块的比例
swichNav	切换滑块（分类）
getList	异步获取商家列表
gotoshop	页面切换，进入商店

#### 5.4.8 shopdetail

商品详情界面。

#### data

变量	功能
id	商店 id
name	商店名
productList	商品列表
noData	是否没有商品
af	积分兑商品的折扣信息

变量	功能
bt	商品兑积分的折扣信息

#### method

方法名	功能
onLoad	页面初始化
onPullDownRefresh	下拉事件
onReachBottom	触底事件
getDataList	异步获取对应商店
id	的商品列表

#### 5.4.9 pay

支付界面。

#### data

变量	功能
timer	计时器
skey	服务器发来的激活码
hasKey	当前持有的激活码是否有效
useScore	是否使用积分
startTime	支付开始时间
hasOrder	支付是否成功
order	支付成功后的订单信息

## method

方法名	功能
payRequest	发起支付请求
sendPay	收到服务器的激活码，打开计时器
waitForEnd	等待商家完成结算
askForEnd	等待商家完成结算期间，每隔固定时间与服务器通信获取支付情况

## 5.4.10 edit

个人信息修改界面，与注册界面比较类似。

## data

变量	功能
imageUrl	图片的本地链接
emailHolder	输入邮箱的提示信息
phoneHolder	输入电话的提示信息
nicknameHolder	输入昵称的提示信息
nicknameValue	昵称初始值，为会员的当前昵称
nickname	昵称
phone	电话
email	邮箱
registDisable	是否允许注册
phoneWarnInfo	电话格式出错提醒
emailWarnInfo	邮箱格式出错提醒

变量	功能
hasPhoneWrong	电话格式是否出错
hasEmailWrong	邮箱格式是否出错

#### method

方法名	功能
PhoneBlur	输入电话失去焦点
EmailBlur	输入邮箱失去焦点
NicknameInput	输入昵称时的响应事件
PhoneInput	输入电话时的响应事件
EmailInput	输入邮箱时的响应事件
chooseImage	选择图片上传
judgeBtnDisable	格式出错时不允许点击按钮
judgeCorrect	输入格式的正确性判断

#### 5.4.11 order

交易历史。

#### data

变量	功能
orderList	历史订单列表

#### method

方法名	功能
onLoad	加载页面时异步加载历史订单信息
goto	点击某个订单可以进入对应的界面

#### 5.4.12 score

会员的积分信息和会员权限。

data

变量	功能
score	积分
level	等级
tag	称号

method

方法名	功能
onLaunch	异步从服务器获取相关信息

#### 5.4.13 setting

小程序的一些基本设置，比如是否使用缓存，是否记录浏览历史。

data

变量	功能
storage	是否使用缓存
limit	是否记录浏览历史

## method

方法名	功能
onLoad	从服务器获取用户上一次的设置
clearStorage	调用小程序的 API 清空本地缓存
switchChange	用户进行某个按钮的切换

## 第 6 章. 数据库设计

### 6.1 技术选型

数据库选用的 mySQL 数据库，版本采用的是指定要求的 MySQL 版本 5.6。

### 6.2 模型描述

数据库存储的数据类型有五类：

- 微信用户信息 (Wxusrinfo)：用于存储用户账户的个人信息，包括昵称、邮箱、openid 等等
- 订单信息 (OrderInfo)：用于存储用户和商家交易订单的信息，包括双方的名字，订单识别号等等
- 商家模型 (ShopInfo)：用于存储商家的信息，包括商店名称、密码、应收款项等等
- 管理员模型 (ManagerInfo)：用于存储管理员信息，包括密码、积分比例、用户升级规则等等
- 商品模型 (GoodsInfo)：用于存储商品信息，包括商品名称、价格、所在商店等等

### 6.2.1 微信用户信息

域	类型	用途
id	整型	主键
openid	字符串	用户的唯一认证 id
name	字符串	用户名
phone	字符串	用户的手机号码
mail	字符串	用户的邮箱信息
score	整型	用户的积分信息
level	整型	用户的等级
session	字符串	用户每次登录更新，通过微信服务器保证安全的登录认证
headimage	字符串	用户的头像 url
tag	字符串	根据用户的消费方式生成的标签



6.2.2 订单信息

域	类型	用途
use_score	布尔型	是否使用积分
openid	字符串	用户的唯一认证 id
user_name	字符串	用户名
score	整型	用户的交易前积分信息
level	整型	用户的交易前等级
start_time	字符串	订单的开始时间
end_time	字符串	订单的结束时间
state	字符串	订单的状态
shop_id	整型	商家的唯一识别号
shop_name	字符串	商家名称
delta	整型	用户积分变化量
amount	双浮点型	交易金额
delta_receivables	双浮点型	商家应收款项的变化量

### 6.2.3 商家信息

域	类型	用途
shopId	整型	主键
shopName	字符串	商店名称，具有唯一性
password	字符串	商户密码
category	整型	商家类型
receivables	浮点数	商家应收款项
rate	浮点数	商家自定义的换取积分的比例
phone	字符串	商家的手机号码
address	字符串	商店地址
ownerName	字符串	商户姓名
headImage	字符串	用户的头像 url

### 6.2.4 管理员信息

域	类型	用途
id	整型	主键
password	字符串	管理员密码
effectiveTime	整型	会员积分有效时长
alpha	浮点数	会员获取积分比例的上限
beta	浮点数	会员兑换积分比例的上限
money{1-4}	浮点数	会员到达各等级所需花费的金额数
alpha{0-4}	浮点数	各等级会员获取积分的比例

### 6.2.5 商品信息

域	类型	用途
goodsId	整型	主键
shopId	整型	所在的商店 id
goodsName	字符串	商品名称
price	浮点数	商品原价
salePrice	浮点数	商品特价
isSale	boolean	是否为特价商品
image	字符串	商品图片的 URL

## 6.3 数据库连接

在 Secoder 平台，我们构建了和仓库并列的 MySQL 数据库容器，并在代码中通过 spring 的相关配置来配置 url: jdbc:mysql://database.bugkiller.secoder.local:3306/serverdata?useUnicode=true&characterEncoding=UTF-8，注意配置对中文的支持以及数据库的连接驱动（driver-class-name: com.mysql.jdbc.Driver），同时通过 mybatis 的相关配置来配置映射文件（mapper-locations: classpath:mapper/\*.xml）以及相关的实体类（type-aliases-package: com.example.app.bean）。

## 6.4 图片存放

图片存放采用了直接存储在一个地址映射中，通过 spring 的配置把用户上传的文件通过随机长文件名（UUID）的方式存放在/opt/hello-maven/pic/目录中，同时也实现了通过 url 的方式来图片回显。



# 第 7 章. 软件测试

## 7.1 概述

在开发过程中，我们对软件进行了细致的测试，以保障其具有充分的可靠性。

## 7.2 后端测试环境

测试在项目 Dockerfile 描述的 Docker 镜像中进行。

后端使用了 junit 来进行单元测试的编写，用 mockMvc 的相关方法来模拟前端的请求，使用 Assert 的相关方法来进行断言。

## 7.3 测试样例

我们的单元测试中包含了以下测试样例：

### 7.3.1 工具类部分（主要是 https 请求的工具类）

- 使用正确的参数以 get 方法访问 www.baidu.com
- 使用错误的参数以 get 方法访问 www.baidu.com
- 使用正确的参数以 post 方法访问 www.baidu.com
- 使用错误的参数以 post 方法访问 www.baidu.com
- 使用正确的参数（json 格式）以 post 方法访问 www.baidu.com

### 7.3.2 微信小程序部分

- 登录部分由于需要和微信官方服务器进行双重验证因此无法进行单元测试
- 以及部分便于 debug 留下来的部分老接口的测试不再列出
- 以错误的 session 进行注册
- 以正确的 session 进行注册
- 以错误的 session 进行获取商家列表
- 以正确的 session 获取商家列表
- 以 5 种不同的等级的 session 分别尝试获取对应的商家列表
- 以正确的 session 获取历史订单
- 以错误的 session 获取历史订单
- 以正确的 session 发起支付请求
- 以错误的 session 发起支付请求
- 在正确的状态以正确的 session 申请获取刚刚支付的订单
- 在正确的状态用错误的 session 申请获取刚刚支付的订单
- 在错误的状态以正确的 session 申请获取刚刚支付的订单
- 用正确的 session 信息获取正确商家 id 的商品列表
- 用正确的 session 信息获取错误商家 id 的商品列表
- 用错误的 session 信息获取正确商家 id 的商品列表
- 用正确的 session 获取我的最新信息
- 用错误的 session 获取我的最新信息

### 7.3.3 订单部分

- 用错误的信息去计算商家应付金额和用户积分变化量
- 用正确的信息去计算商家应付金额和用户积分变化量

- 测试删除订单
- 以正确信息测试商家发起结束订单请求
- 以错误信息测试商家发起结束订单请求

#### 7.3.4 商家端部分

- 以不存在的商家名进行登录
- 以已存在的商家名及错误的密码进行登录
- 以已存在的商家名及匹配的密码进行登录
- 以已存在的商家名进行注册
- 以已注册但未处理的商家名进行注册
- 以可行的商家名进行注册
- 以正确的原密码修改密码
- 以错误的原密码修改密码
- 以正确的原密码修改其他信息
- 以错误的原密码修改其他信息
- 以不存在的用户修改其他信息
- 增加商品
- 删除商品
- 更新商品信息
- 以错误的 token 获取商品列表
- 以正确的 token 获取商品列表
- 以错误的 token 获取订单列表
- 以正确的 token 获取订单列表
- 以错误的 token 提交结算申请
- 以正确的 token 提交结算申请

- 以正确的 token 重复提交结算申请
- 以错误的 token 注销账户
- 以正确的 token 注销账户

### 7.3.5 管理员部分

- 以不存在的 id 进行登录
- 以存在的 id 及错误的密码进行登录
- 以存在的 id 及匹配的密码进行登录
- 以正确的原密码修改密码
- 以错误的原密码修改密码
- 同意新商家注册申请
- 同意不存在的商家注册申请
- 拒绝新商家注册申请
- 拒绝不存在的商家注册申请
- 同意结算申请
- 拒绝申请
- 以错误的 token 获取商家列表
- 以正确的 token 获取商家列表
- 以错误的 token 获取订单列表
- 以正确的 token 获取订单列表
- 以错误的 token 获取会员列表
- 以正确的 token 获取会员列表
- 以错误的 token 获取注册申请商家列表
- 以正确的 token 获取注册申请商家列表
- 以错误的 token 获取结算申请商家列表



- 以正确的 token 获取结算申请商家列表
- 设置 alpha 值
- 设置 beta 值
- 设置等级制度
- 删除商家

## 7.4 测试结果

我们项目通过了上述所有测试样例。最终后端测试的代码覆盖率为 80%。



## 第 8 章. 鸣谢

经过近两个月的奋斗和耕耘，本项目如期保质完成。在此，BUGKILLER 开发组向认真负责地帮助我们的老师和助教团队、耐心与我们交流的项目需求方、一同前行的同学们，以及项目中使用到的前端和后端基础框架的开发人员、互联网上提供了诸多技术支持的文章和博客的作者，表示由衷的感谢！