西安邮电大学
毕业设计（论文）

题目：基于微信平台的酒庄在线销售系统

学院：自动化学院

专业：自动化

班级：自动 1403 班

学生姓名：贾兴武

学号：06141086

导师姓名：侯雪梅　职称：副教授

起止时间：2017 年 12 月 5 日 至 2018 年 6 月 10 日

毕业设计（论文）声明书

本人所提交的毕业论文《基于微信平台的酒庄在线销售系统》是本人在指导教师指导下独立研究、写作的成果，论文中所引用他人的文献、数据、图件、资料均已明确标注；对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式注明并表示感谢。

本人完全理解《西安邮电大学本科毕业设计（论文）管理办法》的各项规定并自愿遵守。

本人深知本声明书的法律责任，违规后果由本人承担。

论文作者签名：

日期：　　年　月　日

西安邮电大学本科毕业设计(论文)选题审批表

<table>
<tr><td>申 报 人</td><td>侯雪梅</td><td>职　　称</td><td>副教授</td><td>学　　院</td><td colspan="3">自动化</td></tr>
<tr><td>题目名称</td><td colspan="7">基于微信平台的酒庄在线销售系统</td></tr>
<tr><td>题目来源</td><td>科研</td><td colspan="2"></td><td>教学</td><td>是</td><td>其它</td><td></td></tr>
<tr><td>题目类型</td><td>硬件
设计</td><td></td><td>软件
设计</td><td>是</td><td>论文</td><td>艺术
作品</td><td></td></tr>
<tr><td>题目性质</td><td colspan="2">应用研究</td><td colspan="2">是</td><td colspan="2">理论研究</td><td></td></tr>
<tr><td>题目
简述</td><td colspan="7">随着人们生活水平的提高，越来越追求对生活的享受，葡萄酒作为一种生活的消遣品，也越来越受到民众欢迎。本题以 Java 编程为核心，实现了一个以微信为基础的线上售卖平台，旨在为民众提供便捷、实惠的购物平台。</td></tr>
<tr><td>对学
生知
识与
能力
要求</td><td colspan="7">1.熟练掌握 Java 知识。
2.熟练掌握 MySQL、Java Web 编程技术。
3.熟悉 Redis、JavaScript、HTML、CSS 的使用方法。
4.熟悉软件开发流程及较强的业务逻辑能力。</td></tr>
<tr><td>具体
任务
以及
预期
目标</td><td colspan="7">1.掌握 Java，MySQL 数据库的用法。
2.掌握 Web 项目开发流程。
3.微信酒庄在线销售系统开发成果形式：毕业论文、源程序清单。</td></tr>
<tr><td>时间
进度</td><td colspan="7">2017.12.16 - 2018.1.15 查阅相关文献资料；
2018.1.15 - 2018.3.1 完成开题报告；
2018.3.1 - 2018.5.1 完成总体构思和各模块代码编写；
2018.5.1 - 2018.5.31 完成毕业论文的撰写；
2018.6.1 - 2018.6.7 准备毕业论文答辩 PPT；
2018.6.8 - 2018.6.10 准备毕业论文答辩。</td></tr>
<tr><td>系（教研室）主任
签字</td><td colspan="2">2017 年 12 月 9 日</td><td>主管院长
签字</td><td colspan="3">2017 年 12 月 9 日</td></tr>
</table>

西安邮电大学本科毕业设计（论文）开题报告

学号	06141086	姓名	贾兴武	导师	张乐友/侯雪梅
题目	基于微信平台的酒庄在线销售系统				

选题目的

这是一款基于微信开发的，为葡萄酒电商而做的应用。

1）从本质上来讲，电商属于内容型产品，以葡萄酒作为专营的电商则属于内容单一的电商。在无法做到内容多的情况下，要把精准推荐作为做好内容独和优的前提。

2）从行业上考虑，葡萄酒属于较高端类消费品，高端葡萄酒大多存在于著名厂商和收藏家手里，因此要做到内容的独和优，要明确产品的定位是 B2C 直购&C2C 自营的混合营平台。

3）当前手机成为一种生活中必不可少的工具，而微信则是每天必要使用的应用。基于微信端开发可以最大限度的提高购买的便捷度。

前期基础

掌握技能： Java Web 编程，MySQL 数据库、Redis 的使用，微信公众号开发技术，JavaScript、CSS、HTML 等 Web 前端开发技术。

掌握工具： IDEA、SQLyogEnt、Web Storm、Postman、Git、Maven。

资料积累：《Effective Java 中文版》、《深入分析 Java Web 技术内幕》《MySQL5.6 从零开始学》

软件条件： Windows、JDK1.8、MySQL5.6、Redis4.0.6。

要研究和解决的问题

1. 高并发问题。 售卖平台，要考虑到高并发，多线程问题用于支持多用户同时访问以及一些“秒杀”活动的执行。

2. 数据库设计。该项目涉及较多数据，用户信息，商品信息，订单信息，排行榜信息，购物车信息等，如何巧妙的设计数据库的表间关系是遇到的第一个难点。

3. 微信公众号的开发。如何通过公众号获取到用户信息并将其加入到数据库中。

4. 前端页面的设计。如何设计出美观而又能显示完整数据、获得较好的用户体验度也是一个问题。

工作思路和方案

1. 将整个项目分为若干模块,以模块为单元，逐个模块进行开发，通过 Git 合并代码。
2. 先做后端代码，整合、测试完成后，再进行前端页面开发，最后进行前后端联调。

微信酒庄销售系统功能框架图如图所示：

微信酒庄销售系统是基于微信公众号所进行的二次开发。系统包括微信登录，下单购买，商品管理，权限控制，用户管理等模块。使用 Spring 进行事务管理以及管理员操作日志的持久化。视图层采用 SpringMVC 接收前端请求，持久层使用 MyBatis 封装 CURD 操作，数据层使用 MySQL 及 Redis 对数据进行存储。子模块之间使用层级 pom 来控制依赖关系，使用 Git 来进行代码托管。

本系统整体采用 SSM（Spring+SpringMVC+MyBatis）框架。Spring 是一种轻量级的开发框架，具有方便解耦、便捷开发等优点；SpringMVC 与 Spring 框架完美契合，且齐具有注解配置，极大地提高了易用性；MyBatis 采用接口和 xml 映射的方式，极大简化了 JDBC 操作以及对结果集的处理，简化了对数据库的 CURD 操作。

第 1 周—第 2 周： 查找相关资料，构思开发流程；

第 3 周—第 4 周： 拟定整体设计概要，设计数据库、展示页面；

第 5 周—第 7 周： 编写各个功能模块代码；

第 8 周—第 10 周：整合后端代码，开发前端页面；

第 11 周—第 12 周：前后端联调，对系统进行整体的测试评估；

第 13 周—第 14 周：撰写毕业论文，准备答辩。

指导教师意见

签字： 年 月 日

西安邮电大学毕业设计（论文)成绩评定表

<table>
<tr><td>学生姓名</td><td>贾兴武</td><td>性别</td><td>男</td><td>学号</td><td>06141086</td><td>专业班级</td><td>自动 1403</td></tr>
<tr><td>课题名称</td><td colspan="7">基于微信平台的酒庄在线销售系统</td></tr>
<tr><td>指导教师意见</td><td colspan="7">评分（百分制）：　　指导教师(签字)：_____　　年　月　日</td></tr>
<tr><td>评阅教师意见</td><td colspan="7">评分（百分制）：　　评阅教师(签字)：_____　　年　月　日</td></tr>
<tr><td>验收小组意见</td><td colspan="7">评分（百分制）：　　验收教师(签字)：_____　　年　月　日</td></tr>
<tr><td>答辩小组意见</td><td colspan="7">评分（百分制）：　　答辩小组组长(签字)：_____　　年　月　日</td></tr>
<tr><td>评分比例</td><td colspan="7">指导教师评分　（%）　评阅教师评分　（%）　验收小组评分　（%）　答辩小组评分　（%）</td></tr>
<tr><td>学生总评成绩</td><td colspan="2">百分制成绩</td><td></td><td colspan="2">等级制成绩</td><td colspan="2"></td></tr>
<tr><td>答辩委员会意见</td><td colspan="7">毕业论文(设计)最终成绩(等级)：

学院答辩委员会主任(签字、学院盖章)：　　年　月　日</td></tr>
</table>

目 录

摘 要

随着社会的进步以及人民生活水平的提高，葡萄酒这样一款之前较为“奢侈”的商品现在越来越平民化，人们在繁忙的都市生活之余，喝上一杯葡萄酒能让人放慢脚步，用闲暇时光慢慢品味人生，感受优雅、自如的高品质生活。然而葡萄酒的终端供应商的触角虽在一二线城市分布较为广泛，但渐渐发展的中等城市市场葡萄酒的供应却较为缺乏，从而使得人们想要在闲暇时间品上一杯红时酒却因为购买不便而扫兴。

针对上述这种葡萄酒售卖的困境，本课题设计出了一种基于微信平台的酒庄在线销售系统，它借助微信平台的高人流量推动整个销售系统的运作。采用微信公众号的形式，用户只需关注微信公众号并授权登录，即能开始浏览、购买商品，便捷的微信授权登录方式以及详尽的葡萄酒信息，能满足大部分人群对葡萄酒的需求，极大地方便了人们对葡萄酒商品的购买。

关键词：JavaScript；CSS；HTML；Spring；MyBatis；Redis；Git

ABSTRACT

With the progress of society and the improvement of people's living standard, the more " extravagant" goods are now becoming more and more popular with red wine. When people live in busy cities, a cup of red wine can slow you down. You take your leisure time to taste life slowly, and feel elegant and high quality life. However, although wine antenna terminal suppliers in a second tier city is widely distributed, but gradually the development of medium-sized city market supply of wine are scarce, which makes people want in their spare time on a cup of red wine is due to buy, do not buy the blanket.

In view of the difficulties of selling this kind of red wine, this topic designs an online marketing system based on WeChat platform, which promotes the operation of the whole sales system with the help of the adult flow of the WeChat platform. The WeChat public number form, users only need to pay attention to the public, and WeChat authorized login, can start browsing and buying goods. Convenient WeChat authorization login and detailed red wine information can meet the demand for red wine by most people, and it is a great place for people to buy red wine.

Key words: JavaScript; CSS; HTML; Spring; MyBatis; Redis; Git

第一章　绪论

1.1 课题背景

中华上下几千年的文化史，没有哪朝哪代离开过酒，酒文化已经是中华文明的一种传统，这虽然更多的指的是白酒，但是伴随着社会的进步与发展，人们对酒文化的理解与领悟也在悄然发生着改变，“不醉不归”的拼杀白酒几乎都是老一辈的人，而更多的80、90后则更加注重饮酒所带给的生活享受。将来几年，80后和90后会逐步成为社会的新生力量，这些年轻人对葡萄酒的锺爱远远超过白酒，因此将来葡萄酒很有可能会代替白酒的大部分市场。而且在近些年来，我国经济水平迅疾发展，居民收入水平不断上升，使得人民的消费能力得以提升，这也极大地推动了人们对 葡萄酒的消费欲望。

互联网时代的来临，微信、微博等新媒体发展势头日益飙升，她们所带给我们生活的改变相信中国老百姓们以及深切体会到了，新媒体不仅使我们传统的生活习惯发生了改变，而且催生了许多新的商业模式。当葡萄酒搭上互联网的“快车”，必将会在同行业掀起一阵波澜。因此，钻研基于微信平台的酒庄在线售卖系统是一件非常有意义的事情。酒庄在线销售系统能满足绝大部分人群对红酒的需求，她不仅能够为人们在购买红酒方面提供便捷，而且能够推动互联网行业的迅速发展。

1.2 课题研究现状

现阶段国内葡萄酒的销售主要有以下几种形式：

1.传统模式。这种销售模式是很多企业的选择，是各葡萄酒企业自营的品牌专卖店，只销售自家葡萄酒品牌，例如：张裕、华东、长城等。这种销售模式的优点在于葡萄酒的品质是有保障的，由于是自营店，不会存在买到“假酒”的情况，弊端在于现阶段大部分自营店只开设在一、二线城市，其他中下发展城市几乎没有，这样使得人们购买葡萄酒较为困难，并且专卖店的葡萄酒价格较高，普通老百姓的经济压力较大。

2.自媒体模式。这种模式即通过微博、贴吧、博客等方式向大多数人传递信息的一种方式。这种模式最大的优点在于，无论你在什么城市，只要接触到互

联网，有社交网络，就能得到葡萄酒的相关讯息，弊端在于现阶段自媒体鱼龙混杂，商品的真伪无法确认，人们有可能会买到假酒，这也是这种模式下葡萄酒价格一般较低的一个原因。

3.电商模式。这种模式即平台从商家拿货然后通过网站销售的方式，向用户售卖葡萄酒，例如酒美网、也买酒、京东等。这种模式优点在于，商品的真伪有了一定的保障，但无法保证一定会买到正品，价格梯度较为明显，人们可

以买自己能接受的商品，而弊端在于大部分电商模式仅支持 PC 端操作，不能做到随时随地访问。

1.3 课题任务

本次毕业设计中，将酒庄在线销售系统与微信平台相结合将是我研究的重点，实现以微信作为系统入口，为用户提供便捷的购物体验。本系统采用模块化开发设计，主要包括首页展示模块、购物车模块、个人中心模块、商品管理模块等。在前端页面的渲染、设计中主要涉及到Java Script、CSS、HTML等技术，用以提高用户体验度，给予用户良好的购物体验，其次用于接收用户发起的请求并将其传递到后端处理。在后端逻辑的处理中主要使用Spring、SpringMVC、MyBatis等技术，对业务逻辑的处理主要依赖于通过对MySQL数据库进行CRUD操作从而实现对整个系统数据的处理。

在本系统中，用户需要先通过关注绑定本系统的微信公众号，点击登入系统选项并授权登录销售系统，然后就可以浏览本平台中的各种商品，还可以执行商品加入购物车、下单购买、个人信息查询等操作，并且本系统具有库存报警功能，在商品库存不足时，系统将会自动报警，而且系统能根据用户的权限可以对商品属性进行修改。总的来说，本系统具有一般电商平台的基本功能，其微信授权登录是一亮点，依托微信平台为用户提供更加便捷的购物体验。

1.4 论文结构

本论文各章节内容安排如下：

第一章：绪论:主要描述本课题的研究背景、研究总体方向、研究任务、作品预期效果等。

第二章：系统需求分析:主要分析该系统的社会需求情况以及各功能模块逻辑要求、业务要求。

第三章：开发辅助工具及开发技术:主要讲述了在项目开发过程中涉及到的软件开发技术和使用到的软件开发工具。

第四章：软件系统设计:主要描述软件的架构、各个模块的功能以及设计思路。

第五章：系统软件测试:主要讲述通过一定的测试方法评估系统的稳定性和可用性。

第二章　系统需求分析

2.1 系统需求分析简述

酒庄在线销售系统的功能需求分析工作很大程度上决定了整个系统的开发方向，后期程序的开发均是建立在对需求的分析之上，如果分析错误，那么无论多么完美的程序代码编写和设计都会是徒劳。酒庄在线销售系统是依托微信平台而展开的一个购物商城的内容和流程，其研发目的在于使用户随时随地购买红酒成为可能。

酒庄在线销售系统的设计和实现主要需求包括以下几个方面：

1. 功能性：实现基本的业务逻辑要求，具有完整的流程以及完善的安全特性。
2. 可靠性：系统需具有完整的报警、报错体系以及对可能出现的异常做出快速的处理的能力。
3. 可用性：系统需包含美观、友好、可靠的交互页面，具有完整的文档、UI 支持。

2.2 系统需求分析详述

2.2.1 系统需求目标

酒庄在线销售系统是一个常见的商城售卖类平台，但本系统采用与微信平台相结合的形式，用微信公众号作为售卖平台入口，旨在为用户提供可以随时随地都能访问的服务。在设计该系统时，首先考虑到与微信登录以及登陆之后的跳转，其次是完善的购物流程以及各种错误信息的报警及处理，最后对用户权限的管理。

系统具体功能包括：

1. 个人中心：用户个人信息查询、订单信息查询、账户充值、收货地址管理等。
2. 购物车：以用户为单位保存用户加入到购物车中的商品。
3. 首页：包括包括个性推荐、严选品牌、大 V 用户、搜索四个模块，通过一定的推荐算法、筛选要求、积分规则、搜索条件向用户展示相对应商品榜或者用户积分榜。
4. 酒品管理：包括酒品库存管理，及商品属性管理。

2.2.2 系统模块需求分析

1.个人中心

由于本系统用户信息都是从微信平台获取，所以用户对于该信息仅有只读的权限而无法进行更改。在本模块中，用户可以查看到个人注册信息、历史消费记录、收货地址信息等。具体数据流图如图 2.1 所示。

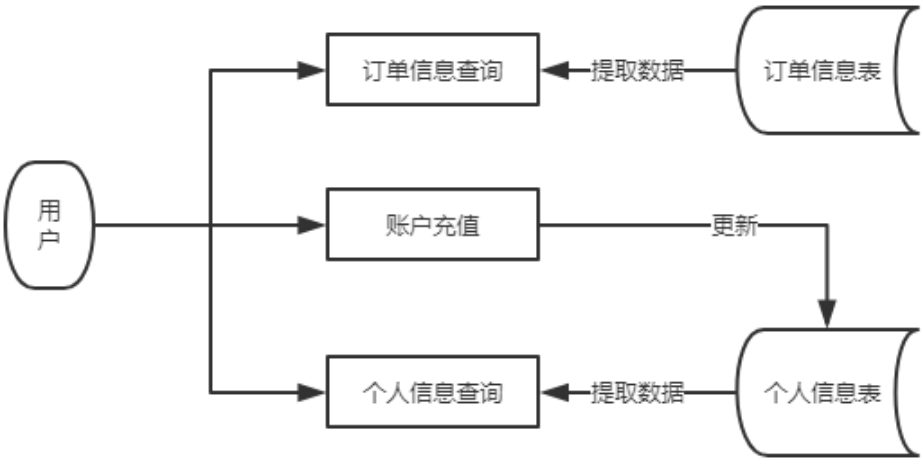

图 2.1 个人中心的数据流图

2.购物车

购物车功能与其他商城平台功能类似，即在挑选商品时，可以将比较符合心意的商品加入到购物车中，用户还可以浏览自己的购物车并对购物车中已有的商品做相应的操作， 是一个常见的较为便利的网上购物工具。该模块的具体数据流图如图 2.2 所示。

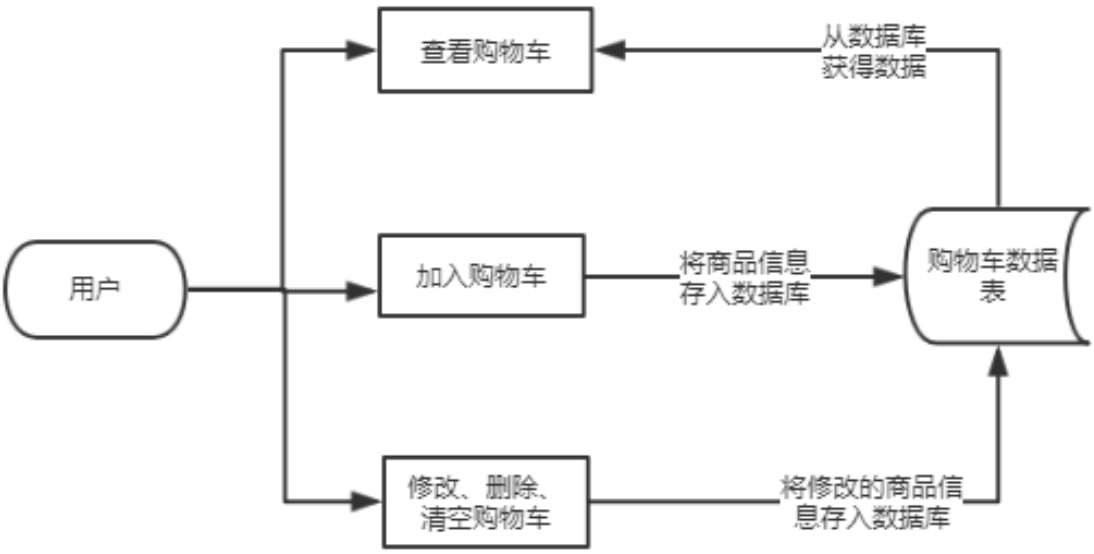

图 2.2 购物车的数据流图

3.首页

在微信授权登录成功后，用户即进入首页，首页又分为个性推荐、严选品牌、大 V 用户、搜索四个模块。

个性推荐模块，依据一定的推荐算法，为用户推荐量身定制的酒品，以达到方便用户挑选，提高购买量的目的；严选品牌模块，根据红白葡萄酒分类依

据商品销量向用户展示推荐商品，减少用户筛选数据的广度；大 V 用户模块，依据用户购买量、活跃度等信息计算积分，并以此生成排行榜；搜索模块，依据酒品名称可进行模糊搜索、精确查找然后展示与之相匹配的商品。该模块的具体数据流图如图 2.3 所示。

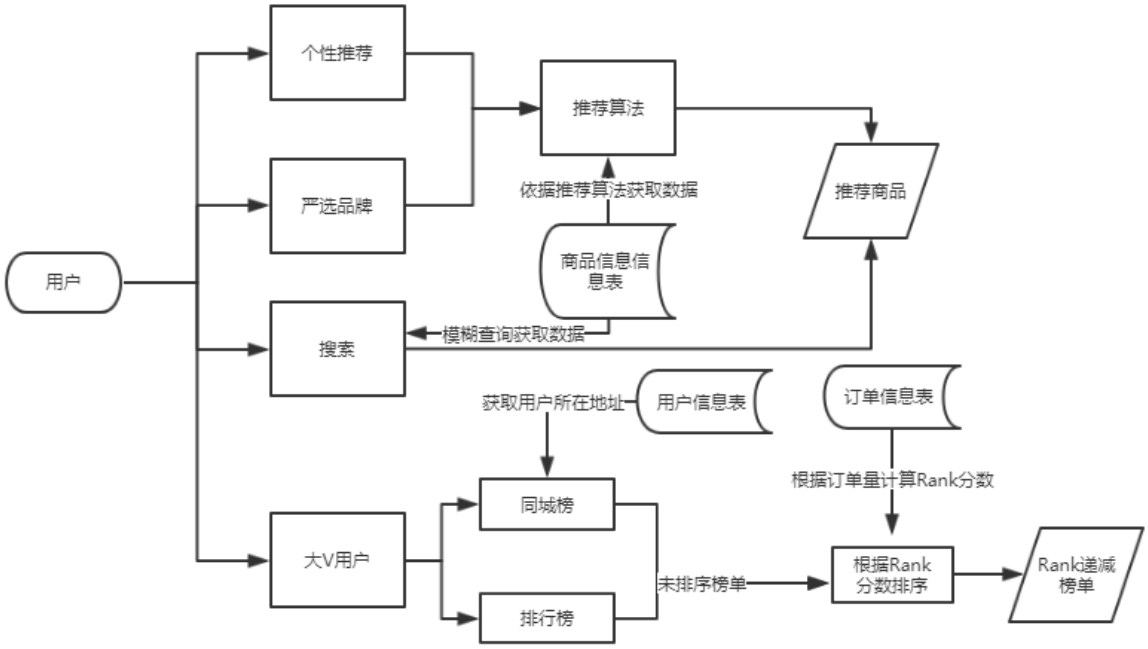

图 2.3 首页的数据流图

4.酒品管理

该功能是对管理员开通的一项服务，系统通过判断用户权限决定是否进入该系统。在该系统下，管理员可对葡萄酒的价格，葡萄酒的库存数量作出调整，也可以上架，或者下架葡萄酒，便于对葡萄酒进行布控，增加葡萄酒信息的灵活性。酒品库存和酒品评分都采用定时任务刷新，自动报警机制，简化商品管理，用于减轻系统管理员的工作负担。该模块的具体数据流图如图 2.4 所示。

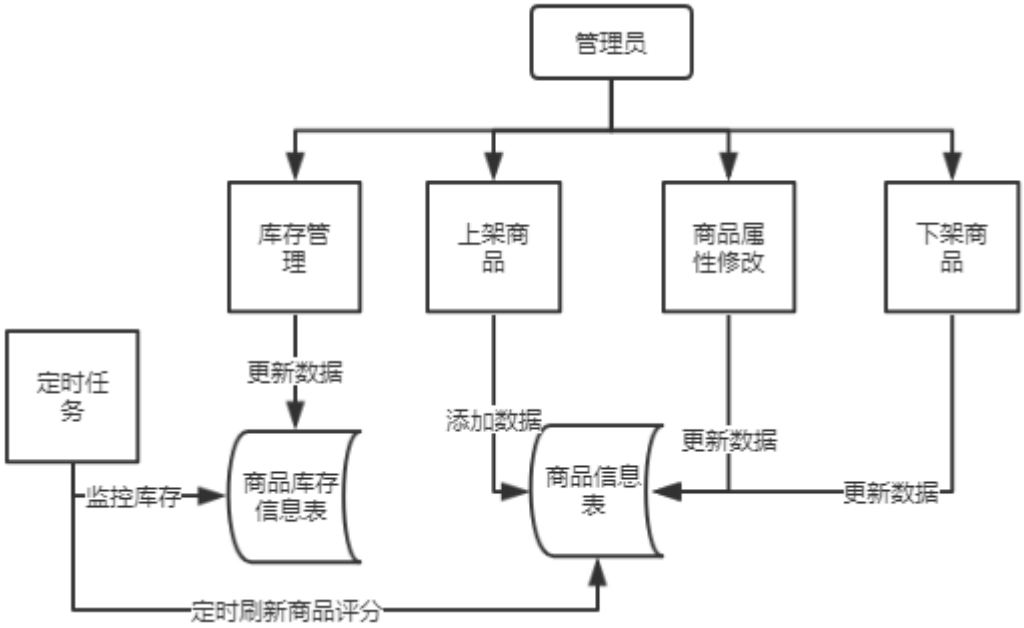

图 2.4 酒品管理的数据流图

5. 根据整个项目的需求分析，得出的软件处理流程概要图如图 2.5 所示：

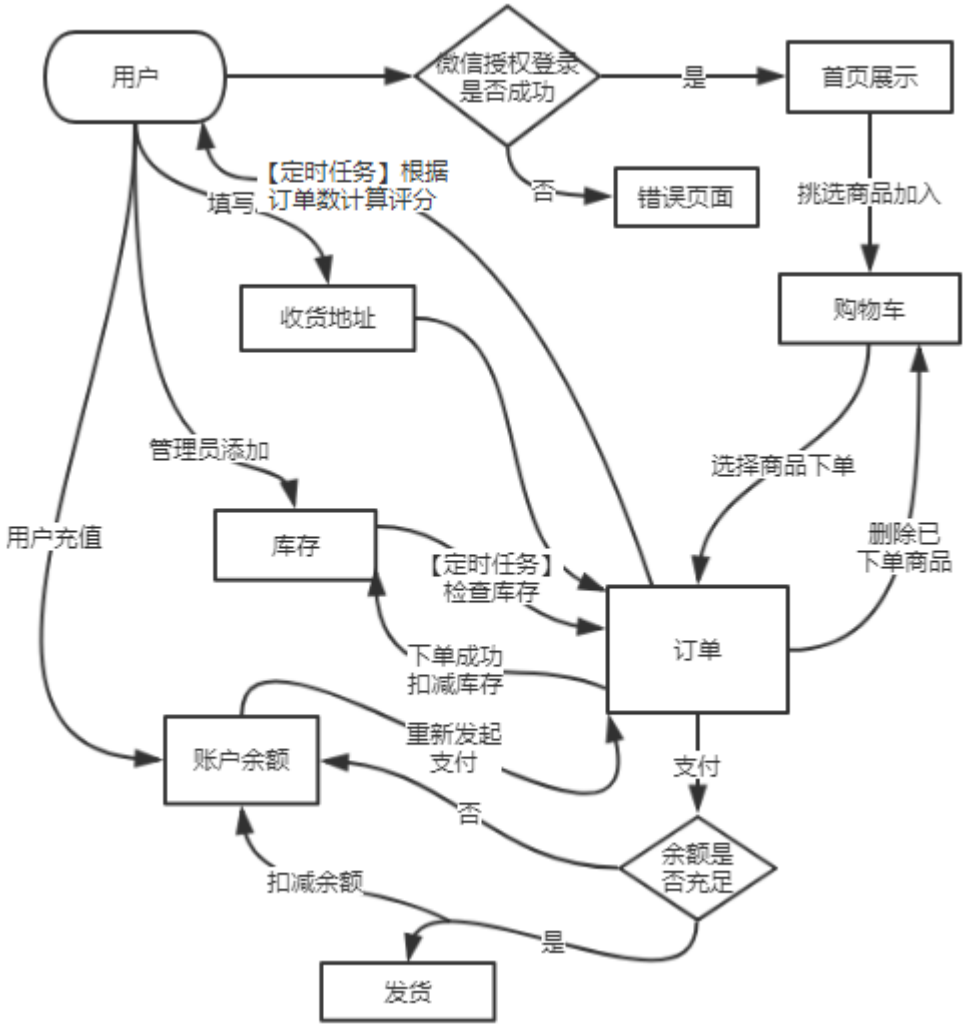

图 2.5 软件处理流程概要图

2.2.3 性能需求分析

本系统下业务交叉并不严重，故而程序接口调用数目并不是太多，但处理各个逻辑均需要和数据库进行交互，在大量数据环境下业务处理、数据库读写速率需要有所保障。

针对以上场景，在数据处理方面，系统采用缓存技术，提高对数据的读写效率；在数据库设计方面，合理使用各种索引，以减少查询时所需要的时间，提高系统处理业务逻辑的能力；程序编码方面，优化程序逻辑，做出重试、超时报错机制。考虑到成本原因，本系统并未采用分布式、数据库分库分表等依赖硬件支持的提高性能方法。

2.2.4 数据库需求分析

本系统的数据库主要是对用户操作过程中数据的记录以及对商品信息的记录功能。且本系统对数据的处理均依赖与数据库的交互，所以，建表时需严格依据数据库三范式并合理建立索引，让数据库性能达到最优。

因为本系统是以用户为最小消费单位，所以系统的实体之间均需要与用户完美衔接才可以。图 2.5 的 E-R 图详实的描写了该体系的数据库表间干系。

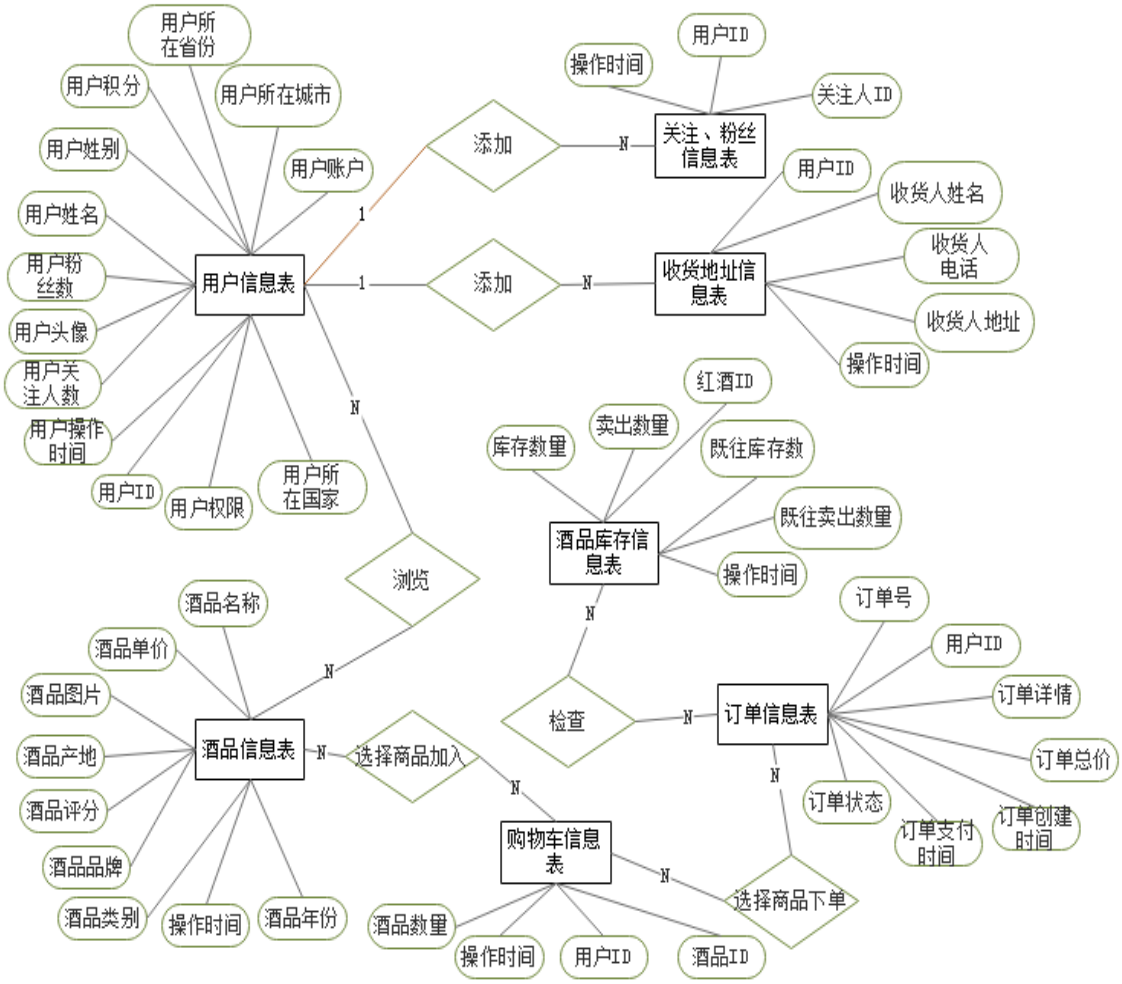

图 2.5 微信酒庄在线销售系统 E-R 图

第三章 开发工具及技术

3.1 开发工具 IDEA 介绍

IDEA，即 IntelliJ IDEA，是由捷克的软件开发公司 Jet Brains 开发的一款产品。它几乎集成了 Java 开发所需要的所有环境，在代码自动提示、J2EE 支持、代码审核、Git 集成等方面的能力可以说是最完备的，相较于 My Eclipse，它对各种插件以及开源框架的契合更加完美，是一款非常好用的 Java 语言开发工具。

3.2 Tomcat 介绍

Tomcat 是一个开源的、免费的 Web 应用轻量级服务器。在并发要求不太高的中小企业被较多使用。对于 Java 应用来说，Tomcat 就是一个 Servlet 处理器，可以运行所有基于 Servlet 的 Java 程序。JSP，以及一些框架（Spring、SpringMVC、MyBatis）均是基于 Servlet 技术，所以都可以利用 Tomcat 运行起来。

3.3 MySQL 数据库介绍

MySQL 采用关联数据库的管理系统，要记录的数据被关联数据库存放在不同的数据表中，相比较于非关系型数据库将所有数据都存放在一个仓库中，MySQL 采用的这种方式极大地提高了数据的灵活性，便于读取。MySQL 因为其多线程、快速性、开源性、成本低、占用资源少等特点被多数中小型网站开发中所青睐。

3.4 Redis 介绍

Redis 现是由 Pivotal 支持开发，它是一个基于缓存的、Key-Value 型的非关系型数据库。由于 Redis 是将所有内容均存放在内存中，所以读写速度较于其它基于硬盘的数据库有很明显优势。由于 Redis 使用了单线程模式，并且在数据类型中采用 Key-Value 数据库，所以能满足极高并发读写的性能要求。但由于其所支持的数据类型限制，Redis 适合存储较简单数据类型，若有较复杂的数据类型还是需要与关系型数据库相结合使用。

3.5 SSM（Spring、SpringMVC、MyBatis）框架介绍

Spring 是由 Rod Johnson 创建，于 2003 年兴起的一个轻量级开源的设计层面的框架。Spring 主要是为了解决业务代码的耦合问题，在整个系统中将面向接口编程贯彻到底，并采用 IOC 技术使系统的耦合度降低。Spring 还具备面向切面编程的特征，该特征许可 Spring 将数据处理逻辑和体系级服务分离，从而使体系的内聚性更高。总的来说，Spring 为开发者提供了干净、便捷、易于管理的编码环境，极大提高了开发效率。

SpringMVC 是基于 Spring 开发的一款产物，是一个利用 Java 语言实现 Web MVC 设计模式的请求驱动类的轻量级开源框架。它主要采用控制器、处理器、解析器完成控制整个应用系统从请求到页面展示的过程，控制器拦截页面请求，处理器完成业务逻辑，解析器完成对页面的渲染。并且 SpringMVC 还提供注解功能，省去了大部分的配置文件，极大地提高了开发效率。

MyBatis 是 Apache 的一个开源项目，是一款基于 Java 语言的持久层框架，主要用来处理与数据库的交互问题。MyBatis 极大地简化了 JDBC 处理过程，消除了复杂的参数设置以及 JDBC 代码，使开发者只需要专注于 SQL 语句的编写，而且 MyBatis 的配置和 SQL 都写在 XML 中，便于统一的优化和管理，开发者只需要简单的 XML 配置即能使 Java 实体与数据库相对应，提高了代码的可维护性以及开发的效率。

3.6 HTML、CSS 、JavaScript 介绍

HTML 全称 Hyper Text Markup Language 即超文本标记语言，它被广泛用于标记页面中媒体、图片、网址等非文字类单元，从而使浏览器按顺序阅读到网页文件时，知道应该怎样展示文件中的内容。因为 HTML 是对浏览器服务的，所以浏览器类型不同，同一标记符所展示出的页面效果可能不同。

CSS 全称 Cascading Style Sheets 即层叠样式表，是一种用来衬托和衬着 XML、HTML 等文件的计算机语言。采用层叠技术，即对同一个单元进行渲染，浏览器只会显示最后一次样式。CSS 可以根据 HTML 所标记的单元进行渲染，定义该单元的显示方式，使页面看起来更加美观，提高用户体验度。

JavaScript 是一种应用于客户端 Web 开发的脚本语言，它可以在 HTML 页面中添加事件的驱动，如按钮、鼠标点击、弹框等，JavaScript 都可以对这些事件做出相对应的反应，从而使页面“动”起来，再结合 CSS 技术，使页面更加华丽。随着技术的发展 JavaScript 库的完善更多的技术涌现出来如 V8、Node.js 等，JavaScript 也慢慢的被用来进行一些服务端程序的编写。

3.7 Git 介绍

Git 是由美国计算机科学家 Torvalds 创建，是一个开源的免费的分布式版本控制工具。使用 Git 可以从远端 Git 仓库克隆完整项目代码，并且在本地根据不同开发目的，创建开发分支，编写代码，完成之后在提交到远端仓库，合并到主分支然后发布。Git 作为一个企业级版本控制工具，以其无单点故障、高内容完整性、分布式版本库等优点已被大多数企业所采用。

3.8 Maven 介绍

Maven 是一款功效十分壮大的项目管理工具。通常在一个项目很大时，借助 Maven 可以将其拆分多个模块，然后按模块开发，利于分工协作；同时借助 Maven 可以很规范的下载 Jar 包，便于对 Jar 的统一管理。

第四章　　系统详细设计

4.1 软件设计思路

该项目系统是基于 Java 平台下 B/S 体系的 MVC 结构。图 4.1 描述了详细描述了 MVC 模型的结构。采用分模块开发的方式，将各个功能点从整个系统中剥离，逐个功能进行开发，最后再将其整合到一起。图 4.2 描述了系统的完整功能框架。

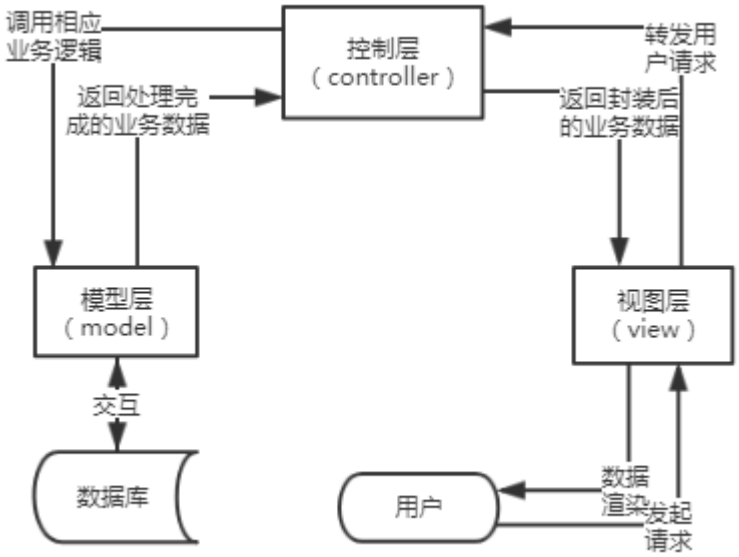

图 4.1 MVC 模型结构示意图

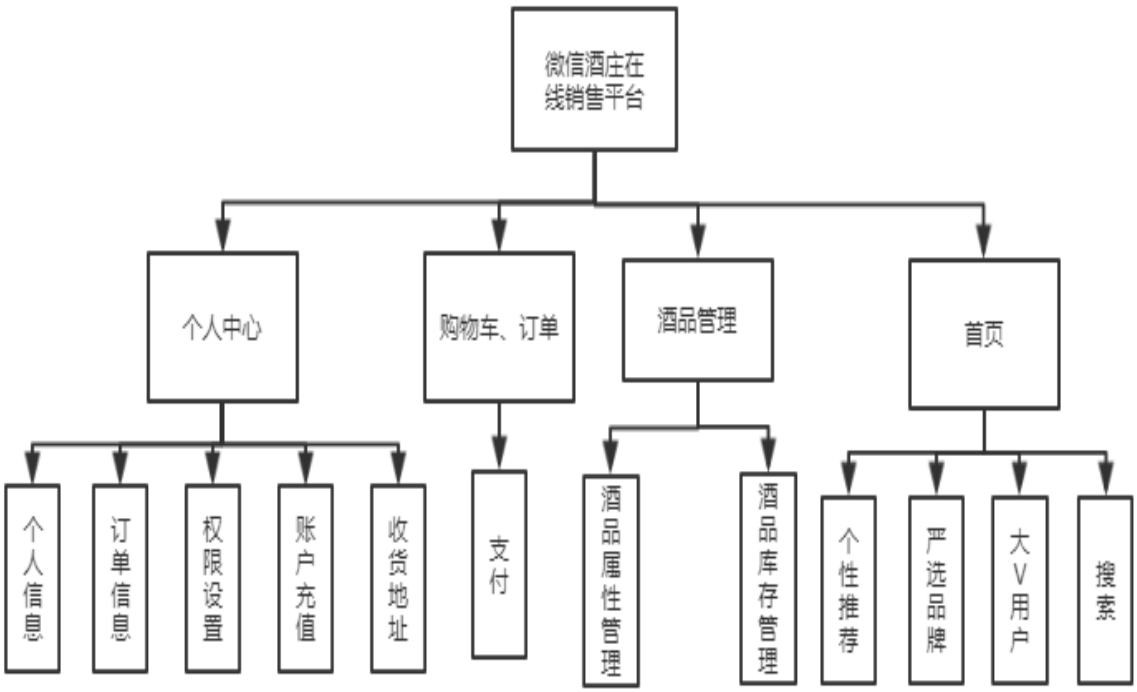

图 4.2 系统功能框架

4.2 数据库设计

由于本系统采用了 MVC 三层架构设计，模型层对数据的处理以及数据库的存储成为了整个系统的关键。为了提高数据访问层面的拓展性和可维护性，对数据库的操作都采用接口的方式，便于统一的管理和修改。

为了更进一步描述数据库的逻辑，图 4.3 详细介绍了数据库中各表的依赖关系。

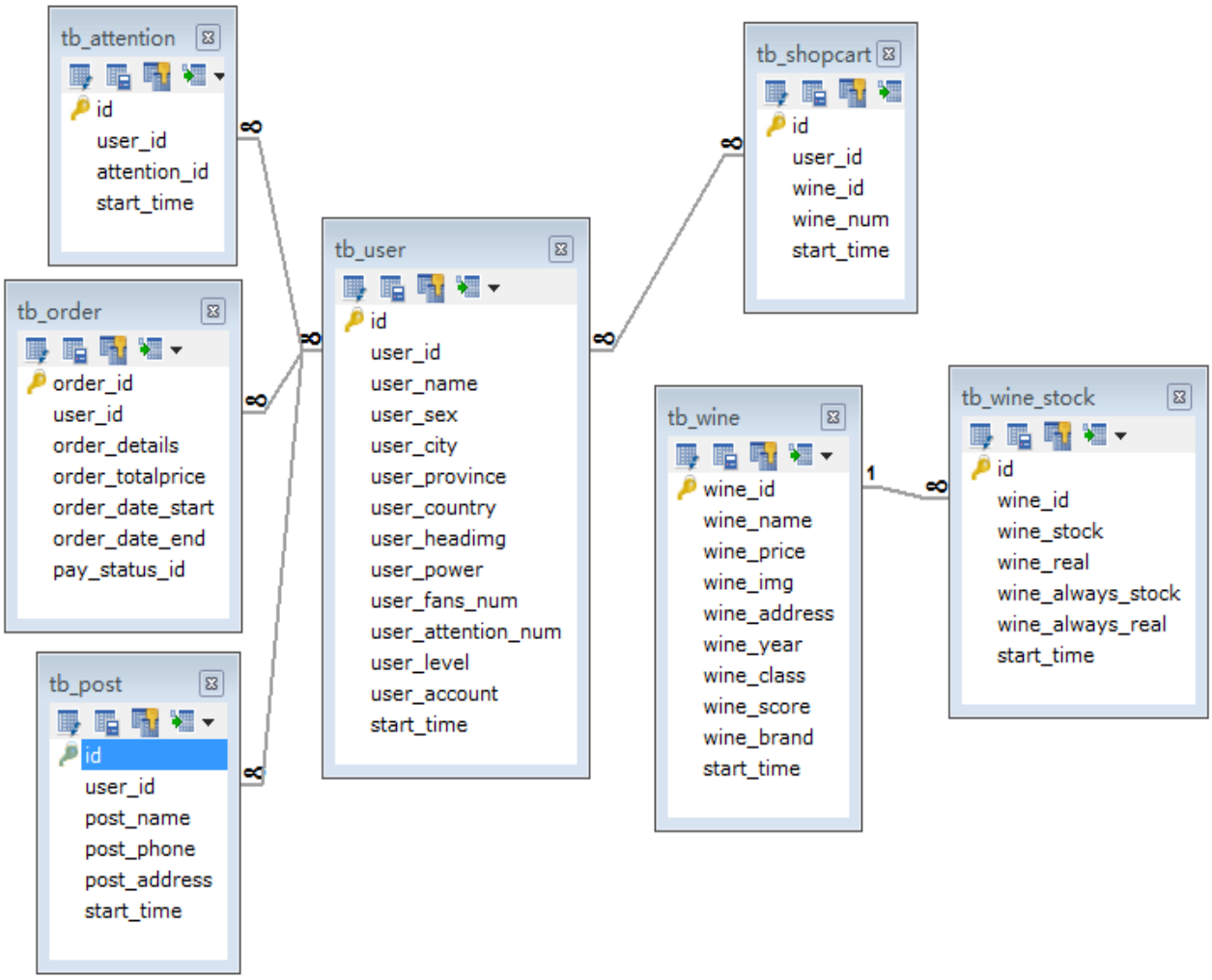

图 4.3 数据库表间关系

表 4.1~4.7 详细描述了图 4.3 中各表的详细信息：

表 4.1 tb_attentio 用户关注信息表

列名	数据类型	备注
id	bigint	关注表主键
user_id	varchar	用户 id
attention_id	varchar	关注人 id
start_time	timestamp	操作时间

表 4.2 tb_order 订单信息表

列名	数据类型	备注
order_id	bigint	订单号
user_id	varchar	用户 id
order_details	mediumtext	订单详情　Json 串
order_totalprice	double	订单总价
order_date_start	datetime	订单创建时间
order_date_end	datetime	订单支付时间
pay_status_id	tinyint	订单状态

表 4.3 tb_post 收货信息表

列名	数据类型	备注
id	int	主键
user_id	varchar	用户 ID
post_name	varchar	收货人姓名
post_phone	varchar	收货人电话
post_address	varchar	收货人地址
start_time	timestamp	操作时间

表 4.4 tb_shopcart 购物车信息表

列名	数据类型	备注
id	bigint	购物车主键 id
user_id	varchar	用户 ID
wine_id	bigint	红酒 id
wine_num	int	红酒数量
start_time	timestamp	操作时间

表 4.5 tb_user 用户信息表

列名	数据类型	备注
id	bigint	用户表主键
user_id	varchar	用户 ID
user_name	varchar	用户姓名
user_sex	varchar	用户性别
user_city	varchar	用户所在城市
user_province	varchar	用户所在省份

user_country	varchar	用户所在国家
user_headimg	text	用户头像 url
user_power	varchar	用户权限
user_fans_num	int	用户粉丝数
user_attention_num	int	用户关注人数
user_level	int	用户积分
user_account	double	用户账户
start_time	timestamp	操作时间

表 4.6 tb_wine 信息表

列名	数据类型	备注
wine_id	bigint	红酒 id
wine_name	varchar	红酒名称
wine_price	double	红酒单价
wine_img	varchar	红酒图片
wine_address	varchar	红酒产地
wine_year	varchar	红酒年份
wine_class	varchar	红酒类别
wine_score	double	红酒评分
wine_brand	varchar	红酒品牌
start_time	timestamp	操作时间

表 4.6 tb_wine_stock 库存信息表

列名	数据类型	备注
id	bigint	库存表 id
wine_id	bigint	红酒 id
wine_stock	bigint	库存数量
wine_real	bigint	卖出数量
wine_always_stock	bigint	既往库存数
wine_always_real	bigint	既往卖出数量
start_time	timestamp	操作时间

依据上述表格不难看出，tb_user 和 tb_wine 两张表存储了本系统的核心数据，各个业务的逻辑均建立在这两张表的基础之上。在设计数据表时，考虑到对这两张表操作的频率，采用了适当添加索引的方式来提高查询的效率。在

整体设计数据表时，也保证数据库三范式的基本要求下，尽量地降低了数据表的冗余度以削减数据库压力。

4.3 前端页面设计

前端页面的主要目的是用于接收后端传入的 Json 格式的数据，并将其解析、渲染，用较为直观、漂亮的方式向用户呈现，其次还要用于接收用户的请求，并将其传入后端进行处理。

前端页面的设计主要使用 div 布局、CSS 渲染、JavaScript 实现，主要有首页、购物车页、我的信息页、商品详情页、库存管理，订单信息等页面。

4.3.1 系统主页设计

系统主页其实是由一个架构组成，包含了个性推荐页、严选品牌页、大 V 用户页以及搜索页面。

1. 个性推荐页

个性推荐页面效果如图 4.4 所示：

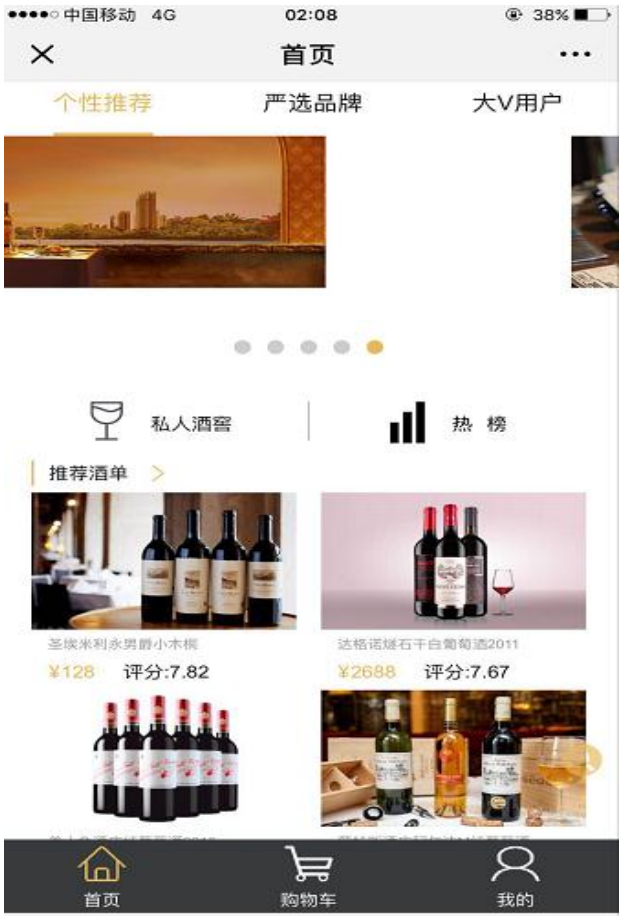

图 4.4 个性推荐页面

在个性推荐选项卡中，置于页面顶端的自动无限循环轮播图采用 jQuery 制作，当点击任一图片则会跳转到商品详情页。点击“私人酒窖”和“热榜”两

个选项卡，则会跳转至相应的推荐页面，在个性推荐页直接展示的为“智能推荐”商品。相同的点击图片则会跳转到相应的商品详情页。

2. 严选品牌页

严选品牌页面效果如图 4.5 所示：

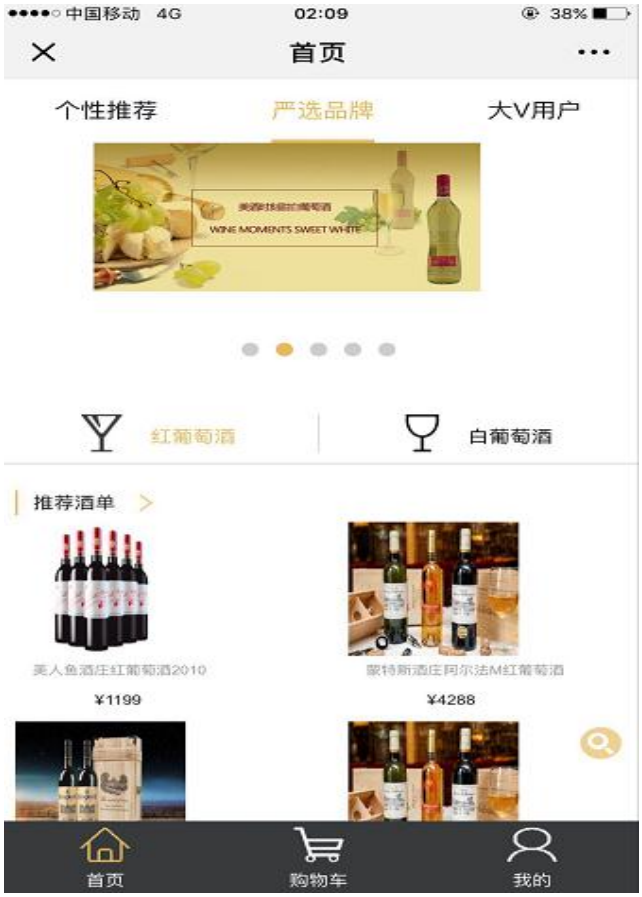

图 4.5 严选品牌页面

严选品牌中 banner 图与个性推荐一致。点击“红葡萄酒”、“白葡萄酒”更换到相应推荐页面，若点击推荐酒品的图片，则会跳转到相应的酒品详情页。所有的事件由 button 触发调用相应函数进行操作。

3. 搜索页

搜索页面效果如图 4.6 所示：

图 4.6 搜索推荐页面

在点击搜索框时会自动呼出输入键盘，对搜索结果的显示也是采用列表的形式，若点击任一结果酒品的图片，则会跳转到相应的酒品详情页。

4. 大 V 用户页

大 V 用户页面效果如图 4.7 所示：

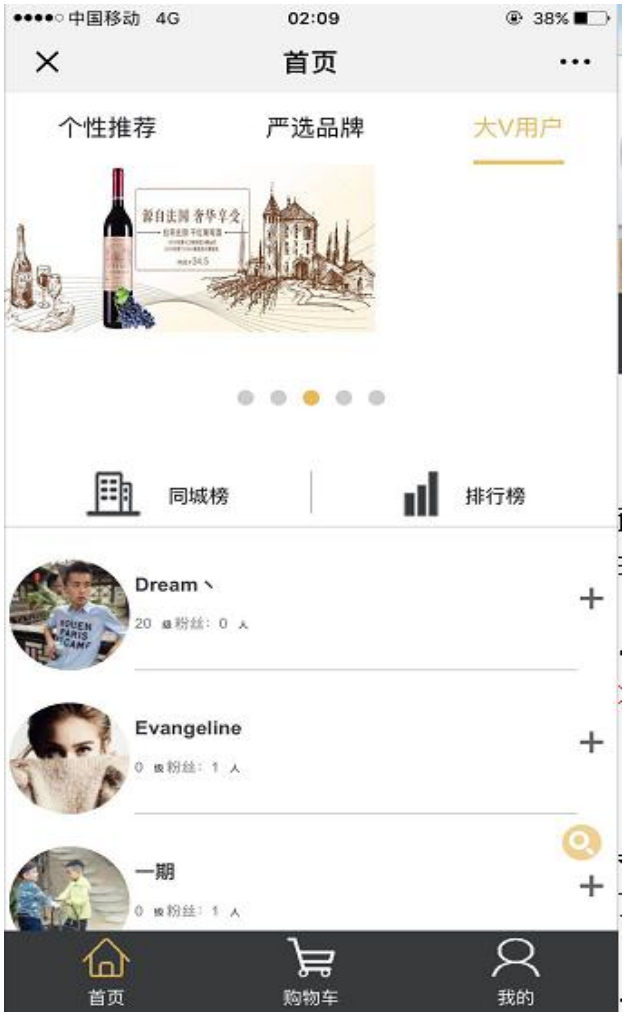

图 4.7 大 V 用户页面

大 V 用户界面中 banner 图与个性推荐一致。进入大 V 用户选项卡，则默认显示“同城榜”，点击排行榜中用户头像时则会跳转到用户详情页。在加载方式上采用了懒加载的形式替代了分页加载，使用户浏览信息更加方便。

4.3.2 购物车页设计

购物车页面效果如图 4.8 所示：

图 4.8 购物车页面

购物车页面中，点击“+”、“-”、“全选”时页面对应的数量和总价要相应变化。该效果主要采用 Ajax 异步刷新的方式，对局部数据进行更新较于加载整个网页节省了很多的时间。

在点击去结算之后，页面将会弹出确认框，点击“确定”将会跳转到订单详情页，且订单详情页信息为购物车页面勾选的商品，而且在订单支付成功之后，购物车中相关商品将会被删除。

4.3.3 个人主页设计

个人主页效果如图 4.9 所示：

图 4.9 个人主页面

个人主页包含了“全部订单”、“账户充值”、“待支付”、“商品属性管理”，“权限设置”等选项卡，点击选项跳转到相对应的页面。“商品属性管理”、“权限设置”选项卡只是针对于管理员开放，普通用户访问页面没有该选项。在任意用户主页点击“+”则会关注该用户，自己将会成为该用户的粉丝。

点击“全部订单”选项，将会显示该用户已支付的所有订单，并在点击查看详情之后，会显示出该笔订单的详细内容。点击“充值”选项，将会弹出充值框，在用户输入完要充值的金额后点击确定将会充值成功，且个人信息中的余额数也会被刷新。

4.3.4 商品详情页设计

商品详情页效果如图 4.10 所示：

图 4.10 商品详情页面

商品详情页面着重展现了商品的图片、价格、名称、库存等信息。在用户调整商品数量时，总价会相应的发生变化，用户选择商品数量后，点击“加入购物车”即完成加入购物车操作，完成该操作后页面不会跳转，还是停留在商品详情页。

4.3.5 订单查询页设计

订单查询页效果如图 4.11 所示：

图 4.11 订单查询页面

订单查询页采用列表的形式显示当前用户所有下单记录，订单的状态（已支付、超时未支付，待支付）也会进行标注。为了确切地保证订单数据的完整性以及安全性，订单数据对于用户是无法进行修改。

4.3.6 下单支付页、账户充值页、收货信息页设计

下单支付页面、账户充值页面、收货信息页面效果如图 4.12 所示：

图 4.12 下单支付页面

本系统当前仅支持账户余额支付，在支付时会显示当笔消费详情，若余额充足则支付完成，若余额不足，则会提示跳转账户充值页面，提示该笔订单将会在消费记录中显示。同时在下单时也可进行收货信息的选择，若没有收货信息可选或者需要新添加收货信息，则会引导进入收货信息页面。

在用户输入正确的充值金额，点击“立即充值”则充值成功，同时会返回个人中心页，便于用户之后的操作。对用户输入的数据，前端使用了数据的校验，制止了脏数据的传入。

在用户输入收货信息后，点击“添加”则会添加成功，并且会返回到订单支付页面，便于用户继续操作。在用户输入数据时，前端的数据会进行“判空”、“错误字符”等校验，对于不良字符会及时给予提示。

4.4 后端模块设计

4.4.1 首页展示模块设计

首页展示模块，主要用于在用户登录成功后依据不同的筛选条件向用户展示各类商品，以激发用户的购买欲，提高销售额。根据不同的推荐算法条件，该模块又分为个性推荐、严选品牌、大 V 用户、搜索四个功能块。

1. 个性推荐模块

个性推荐，即通过用户的购买记录，喜好等方面为用户推荐合适的商品，减少用户挑选商品的时间同时也增加了用户浏览后的购买的可能性。

在私人酒窖选项卡中，主要是在用户购买的记录中选取购买次数最多的 6 项酒品推荐给用户，若用户无购买记录或者购买记录中酒品少于 6 项，则将最热销的商品拼接在推荐表之后使其数目达到 6 项然后推荐给用户。

在智能推荐选项卡中，是以用户购买较为频繁酒品的类别、产地、品牌为筛选条件，为用户推荐同类型的商品。同样的若推荐商品数目少于 6 或者用户无购买记录，则以最热销的商品填补。

热销榜，则是以商品的销售量、库存量为衡量依据，计算公式为：

商品评分=该商品既往销售量/该商品既往库存

同时为了数据的可观性，设置该分数下限为 3.0，上限为 10.0，且商品评分更改时间在每天早上 6 点，由定时任务触发。定时任务触发效果如图 4.13 所示：

```
c.j.design.task.UpdateWineScoreTask - 【酒品评分】第14个酒品评分更新成功,评分数:4.52
c.j.design.task.UpdateWineScoreTask - 【酒品评分】开始更新第15个酒品评分,评分数:3.0
org.mybatis.spring.SqlSessionUtils - Creating a new SqlSession
org.mybatis.spring.SqlSessionUtils - SqlSession [org.apache.ibatis.session.defaults.DefaultSqlSession@16
o.m.s.t.SpringManagedTransaction - JDBC Connection [com.alibaba.druid.proxy.jdbc.ConnectionProxyImpl@15c
c.j.d.d.m.WineDao.updateWineScore - ==>  Preparing: UPDATE tb_wine SET wine_score = ? WHERE wine_id = ?
c.j.d.d.m.WineDao.updateWineScore - ==> Parameters: 3.0(Double), 15(Long)
c.j.d.d.m.WineDao.updateWineScore - <==    Updates: 1
org.mybatis.spring.SqlSessionUtils - Closing non transactional SqlSession [org.apache.ibatis.session.def
c.j.design.task.UpdateWineScoreTask - 【酒品评分】第15个酒品评分更新成功,评分数:3.0
c.j.design.task.UpdateWineScoreTask - 【酒品评分】开始更新第16个酒品评分,评分数:3.0
```

图 4.13　定时任务触发效果图

2. 严选品牌

严选品牌则是根据商品的某种属性进行分类根据一定的排序规则展示给用户，它与个性推荐的最大区别在于，个性推荐是根据用户信息进行推荐，而严选品牌只是根据商品属性进行推荐。严选品牌有红葡萄酒和白葡萄酒两个选项卡，都是根据葡萄酒的种类进行区分，且根据酒品评分由高到低排序展示给用户。

3. 大 V 用户

大 V 用户，是为了让用户能更好的了解到其他用户的购买趋势、购买能力，而根据用户的购买行为，积分排榜的一种形式。用户每购买一件商品则能

积一分，用户每次访问排行榜时更新自己的积分并刷新排行榜，以此保证用户看到的是最新的榜单。且大 V 用户中的同城榜选项卡，可以根据用户注册所在地生成同城榜，让用户能更好的体验到周围其他用户的购买力。

4. 搜索

搜索模块，即根据用户输入的商品名称，在整个商品库中进行筛选，挑选出与之相匹配的商品，并以列表的方式展示给用户。其主要使用到 MySQL 的模糊查询关键字“LIKE”，简单的通过一句 SQL 完成该功能如图 4.14 所示。

```
<select id="searchWine" resultMap="BaseResultMap">
    select
    <include refid="Base_Column_List"/>
    from tb_wine
    WHERE
    wine_name LIKE concat(concat("%",#{content}),"%")
    LIMIT 10
</select>
```

图 4.14 模糊查询功能 SQL 语句

4.4.2 个人中心模块设计

个人中心模块，旨在为用户处理个人信息提供服务，使用户完成对个人基本信息、购买信息的查询与修改。该模块主要由个人信息（微信平台拉取）、订单信息、账户充值、收货地址管理四个模块组成。

1. 个人信息模块

由于本系统借助微信平台授权登录，所以用户的所有个人信息均来源于微信平台用户注册的信息，用户对于个人信息只有只读权限。微信授权登录是在用户进入授权登录页面之后获得到 code，然后以 code 为凭据来换取网页的授权 access_token，再根据唯一标识用户的 openid 获取到用户的基本信息。图 4.15 详尽说明了微信授权登录流程。

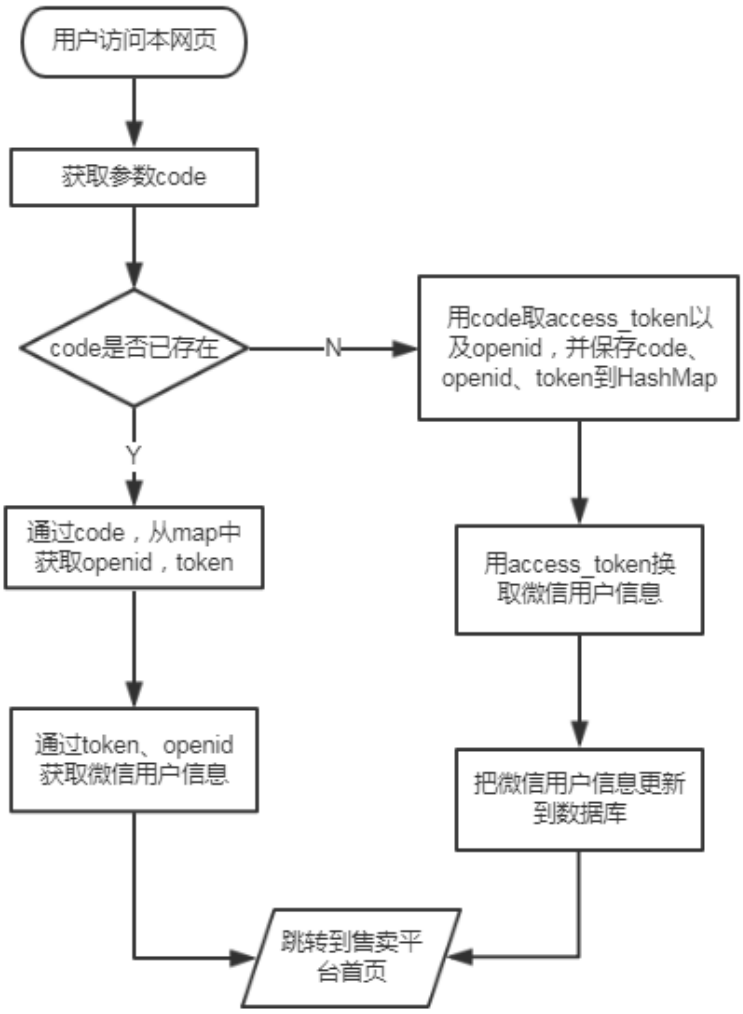

图 4.15 微信授权登录处理流程

执行过程中通过入口地址和回调地址完成与微信平台数据的交互，具体代码实现方式如图 4.16 微信授权登录入口地址、图 4.17 微信授权登录回调地址所示。

```
@RequestMapping(value = "login.htm")
public void login(HttpServletResponse resp) throws ServletException, IOException {
    try {
        logger.info("【微信授权】：用户请求微信授权登录");
        String callBackUrl = "http://dreamjxw.imwork.net:80/jxw/design/user/callback.htm";
        String codeURL = "https://open.weixin.qq.com/connect/oauth2/authorize?appid="
                + AppInfo.APPID.getAppInfo() +
                "&redirect_uri=" + URLEncoder.encode(callBackUrl) +
                "&response_type=code" +
                "&scope=snsapi_userinfo" +
                "&state=STATE#wechat_redirect";
        resp.sendRedirect(codeURL);
    } catch (Exception e) {
        logger.error("【微信授权】微信授权登录时出现异常", e);
    }
}
```

图 4.16 微信授权登录入口地址代码块

```
@RequestMapping(value = "callback.htm")
public void callBack(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    try {
        logger.info("【微信授权】：登录回调方法");
        String code = req.getParameter( s: "code");
        String openid;
        String token;
        String accessTokenUrl = "https://api.weixin.qq.com/sns/oauth2/access_token?appid=" + AppInfo.APPID
                .getAppInfo()
                + "&secret=" + AppInfo.APPSECRET.getAppInfo()
                + "&code=" + code
                + "&grant_type=authorization_code";
        JSONObject jsonObject = JsonUtils.doGetJson(accessTokenUrl);
        logger.info("【微信授权】：Token信息,:{}", new Gson().toJson(jsonObject));
        openid = jsonObject.getString( key: "openid");
        token = jsonObject.getString( key: "access_token");
        String infoURL = "https://api.weixin.qq.com/sns/userinfo?access_token=" + token + "&openid=" + openid +
                "&lang=zh_CN";
        JSONObject jsonObjects = JsonUtils.doGetJson(infoURL);
```

图 4.17 微信授权登录回调地址代码块

2. 订单信息模块

订单信息模块，主要记录用户下单的行为所产生的数据，无论是否支付完成，订单信息中都会有记录，且在订单超过 30 分钟未支付，则该笔订单将会无效。在某笔订单产生时，其所对应商品的库存将会被扣减，若订单中任一商品库存不足则会下单失败，同时在产生订单后购物车中所对应商品将会被删除，以提高用户购物体验。对订单的查询有两种方式：订单编号和用户 ID，以满足不同的业务场景需求。

3. 账户充值模块

由于本系统暂未接入正式的交易系统，所以期间的账户余额都是模拟的。用户可通过简单的操作为自己的账户充值金额。该模块主要是为了完善本系统的购物流程而设计，若要正式上线投入生产则需要与真实交易系统相连。

4. 收货地址管理模块

本系统主要是用于存储用户下单时的收货信息。在填写过一次之后，用户可在下次购物时，在之前已填的收货信息选项中选择一项，为用户的购物过程提供方便，提高用户体验度。

4.4.3 购物车模块设计

购物车作为电商平台的一项重要工具，它主要是用于记录和保存用户挑选完的商品，等待用户结算。用户可以对购物车中商品做增加、删除、修改、查询，且支持批量结算操作。本系统购物车基于数据库实现，将用户操作的数据均在数据库中做持久化操作，且以用户为单位绑定，便于用户的再次浏览。

4.4.4 商品管理模块设计

商品管理模块中集成了用户权限功能，只有管理员才可以对商品的属性做出修改。该模块是由商品属性管理和商品库存管理两个功能点构成。

1. 商品属性管理

拥有管理员权限的用户可对商品的价格做出相应的调整，并且还可以上架新商品或者下架商品，提高了商品信息的灵活性和平台的广度。

2. 商品库存管理

商品库存检测依据定时任务对商品库存数进行刷新以日志的形式输出商品的销售信息，若剩余库存数小于总库存 10%，则会触发报警，在系统日志中输出对应商品信息，以提醒商家增加库存。图 4.18 为系统对商品库存信息处理的输出样例。

```
INFO  com.jxw.design.task.CheckStockTask - 【检查库存】已获取所有酒品库存信息
INFO  com.jxw.design.task.CheckStockTask - 酒品1已卖出18件，库存100
INFO  com.jxw.design.task.CheckStockTask - 酒品2已卖出10件，库存100
INFO  com.jxw.design.task.CheckStockTask - 酒品3已卖出29件，库存300
INFO  com.jxw.design.task.CheckStockTask - 酒品4已卖出2件，库存100
INFO  com.jxw.design.task.CheckStockTask - 酒品5已卖出0件，库存100
INFO  com.jxw.design.task.CheckStockTask - 酒品6已卖出6件，库存100
INFO  com.jxw.design.task.CheckStockTask - 酒品7已卖出0件，库存100
INFO  com.jxw.design.task.CheckStockTask - 酒品8已卖出0件，库存100
INFO  com.jxw.design.task.CheckStockTask - 酒品9已卖出0件，库存100
INFO  com.jxw.design.task.CheckStockTask - 酒品10已卖出3件，库存100
INFO  com.jxw.design.task.CheckStockTask - 酒品11已卖出0件，库存100
INFO  com.jxw.design.task.CheckStockTask - 酒品12已卖出0件，库存100
INFO  com.jxw.design.task.CheckStockTask - 酒品13已卖出0件，库存100
```

图 4.18 系统对商品库存信息的输出样例

4.4.5 其他辅助设计

拥有优秀的工具类，能提高不少开发效率，Java 虽带有较多工具类，但在本系统的使用中并不是十分方便，所以采用自定义工具类的方式，设计了某些工具类如 Json 格式化工具类、资源文件读取工具类等。因为本系统采用前后端分离的开发方式，所以前后端数据接口必须统一，所以设计了后端数据传输模型，便于统一管理，协同开发。

1. 后台返回值模型

该模型主要是对返回数据进行包装，加入了一些特殊标识符以及特定的属性，用于规范返回值类型，使前后端数据一致，便于开发中对数据的渲染及修改。

返回类型参数说明详见表 4.7

表 4.7 后台返回值模型参数

参数名	说明
ret	请求成功或者失败标志 true or false
msg	返回报错信息（程序出错、异常信息）
errargs	错误对象
data	返回数据
ver	版本

若前端页面请求成功，且后台返回正确有效的数据，则会返回：

示例：

```
{
        "ret": true,
        "msg": null,
        "errargs": null,
        "data": "添加收货信息成功",
        "ver": "1.0"
}
```

“ret”是 true 即表明该次请求有效，返回数据成功。

若请求出错，或返回无效数据，则会返回：

示例：

```
{
        "ret": false,
        "msg": "您还没有关注的人，快去关注你的 Ta 吧~~",
        "errargs": null,
        "data": null,
        "ver": "1.0"
}
```

“ret”为 false，则表明该次请求返回数据无效，此时提示信息会在“msg”中，“data”仅当“ret”为 true 时有值，其他情况皆为 null。

2. 资源文件读取工具

资源文件的读取主要用于对系统中固定参数的设置，如微信登录时的 APPID、APPSECRET 和数据库连接参数。采用资源文件读取的方式，便于后期对参数的调整，增加系统的灵活性。图 4.19 描述了主要使用到的程序片段。

```java
/**
 * 获取指定key的属性值。
 * <p/>
 * <p>
 * 属性值未找到时，返回指定的默认值。
 * </p>
 *
 * @param key          属性名
 * @param defaultValue 属性默认值
 * @return 指定属性名的属性值
 */
public static String getProperty(String key, String defaultValue) {
    String result = props.get(key);
    if (result == null) {
        return defaultValue;
    }
    return result;
}
```

图 4.19　资源文件读取程序片段

其中各参数已在注释中写出。读取文件时还是采用键值对的方式，使属性值与名称一一对应，方便数据提取。

第五章 系统测试

软件质量的重要性对一个软件体系来讲是无可厚非的，而软件的质量就必须经由软件测试来保证。软件测试是指经由某些测试方法、测试用例找出软件系统中存在缺陷和漏洞的过程。软件的测试并不仅仅只是代码层面上的测试，还包括系统架构、需求分析、逻辑设计等系统开发中的各个阶段。

测试时，基本的测试方法包括静、动态测试，代码复查和黑盒、白盒测试，依据不同的开发阶段，测试又可划分为单元测试、系统测试、确认测试。

由于本系统的适用情况，本系统主要使用单元测试和系统测试。

5.1 测试方法说明

1. 单元测试

单元测试是程序测试的最小单元，是以系统的功能模块为测试主体的测试方式。单元测试的目的即为发现系统中各个功能模块可能存在的问题，以及是否按照需求设计。

本系统的单元测试只要是采用 Postman 模拟请求方式进行测试，分析系统返回结果是否达到预期，并且对系统参数校验，边界值判定等作出测试。具体操作示例如图 5.1 所示。

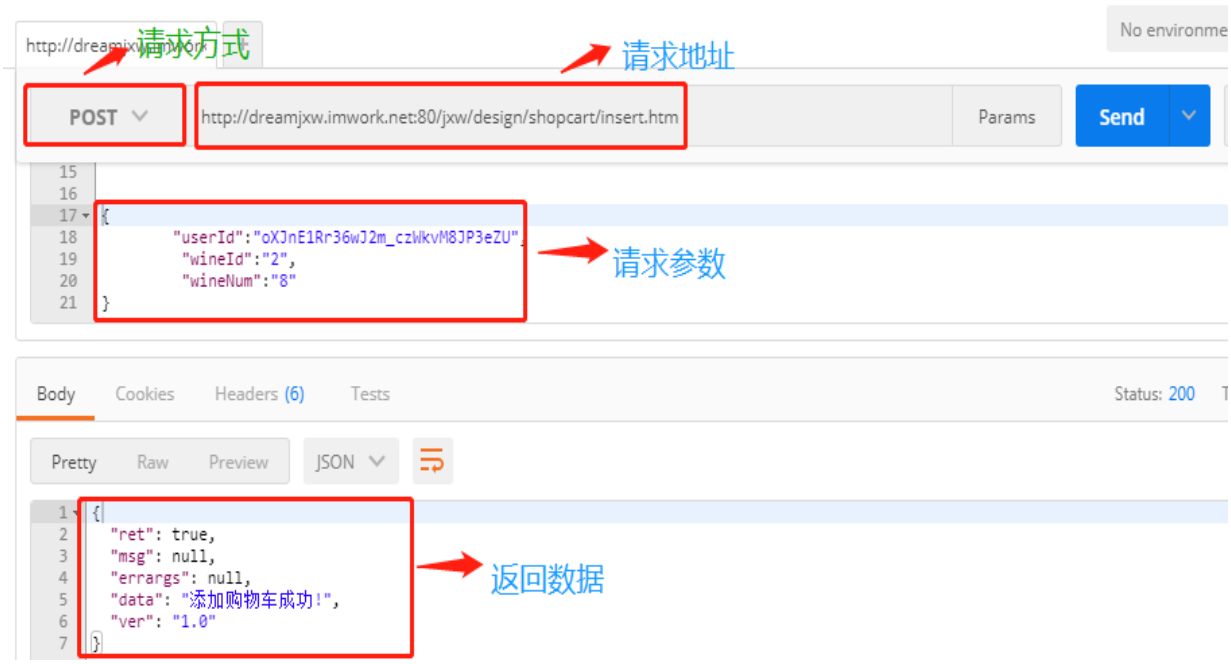

图 5.1 单元测试示例

2. 系统测试

系统测试是在完成各个模块的功能测试以后，对系统整体业务流程进行测试的过程。系统测试的目的是为了发现系统流程是否有缺陷，业务交叉是否影响到模块的功能，以及业务流程是否与需求相对应。

5.2 测试结果分析

在对该系统进行单元测试时，发现了模块功能上存在的一些缺陷，在将这些缺陷解决之后，又通过对整个系统进行后端联调以及前后端联调，测试结果表明，本系统现阶段能正常运转，业务数据处理准确，系统流程正确，满足了系统需求所规定的要求，软件设计成功。

结束语

本课题使用 Java 编程技术和 MySQL 数据库技术及 B/S 式实现了基于微信平台的酒庄在线销售系统。

在本课题开发过程中，我遇到了很多问题，但在解决这些问题的过程中也让我学习到了许多之前没有了解到的技术，并且在系统设计时，为了解决业务交叉，代码逻辑简化等问题也费了不少心力。但这一切在指导老师和同学们的帮助下都得以解决。

本系统最终将微信平台的酒庄在线售卖平台的功能都已基本实现，包括微信授权登录，商品智能推荐，下单结算等。从后台软件基本框架的搭建到最终前端页面渲染数据，期间或大或小都遇到了许多问题，主要是自己编程能力还不够强，导致业务实现与需求之间有一定的出入。

但在此次经历后我学习到了许多程序设计的技巧，知道了软件开发基本流程，掌握数据库应用程序开发的相关知识。虽然本系统还存在一些不太完善的地方，但就现阶段的开发现状来说，我取得的进步是非常大的。最重要的是，因为本次开发采用前后端分离的方式，所以通过这次设计，让我切身体会到前后端分离开发带来的便捷以及其中应该注意的地方，也让我慢慢的具备了一个软件开发工程师应该具备的基本素质，这对于我之后的工作是十分有利的。

由于本人领悟能力及技术能力有限，本系统中可能还有很多不合理或者疏忽的地方，希望读者能给予宝贵的意见和建议，在此本人表示衷心的感谢。

致　谢

在毕设指导老师耐心的教导和我不懈的努力下，我终于如期完成了毕业论文的创作。在此，特别要感谢我的论文指导老师---西安邮电大学自动化学院侯雪梅老师，本篇论文能够如期的完成离不开老师的悉心教导。无论是在我论文的立题、开题、创作还是毕设作品的设计方面侯雪梅老师均给出了许多指导性的意见和建议，费心费力的耐心指导我。不仅如此，我还要感谢西安邮电大学自动化学院自动化专业的各位授课老师们，是老师们的无私奉献、耐心教导，才使我在大学学习期间掌握更多知识，为撰写本论文打下基础。

此外，还要感谢我的同学们、朋友们在毕业论文创作过程中给予我的支持和帮助，在我遇到困难时你们总能在一旁鼓励我，给我加油鼓劲，为我提供无限动力。同时也要感谢各位参考文献中的作者，通过参考你们的研究成果以及论述过程，使我对研究课题有了更好的理解。

当然我的论文作品还不是很成熟，还有很多待完善的地方。但是通过这次做论文的经历我学习到了很多能让我终身受益的东西，它们在我以后的生活和工作中都是非常有价值的，我也会在之后的生活中再接再厉，往更好的方向发展。

参考文献

[1] 李运莉. web 数据库应用系统性能优化[M]. 北京：人民邮电出版社，2015.

[2] 库俊国. 基于 J2EE 技术的 Web 应用体系研究及实践[M]. 北京：人民邮电出版社，2014.

[3] 卜佳旭. 基于 MVC 模式企业信息管理系统设计与实现. 北京希望电脑公司，2014.

[4] 彭晓青. MVC 模式的应用架构系统的研究与实现[J]. 电子工业出版社，2014.

[5] 张桂珠、刘丽、陈爱国. Java 面向对象程序设计（第 2 版）北京邮电大学出版社.

[6] 毕广吉. Java 程序设计实例教程[M]. 北京：冶金工业出版社，2014 年.

[7] 徐茂. 浅析面向 SQL 数据库注入攻击的 Java Web 防御措施[J]. 网络安全技术与应用，2016,（10）：85-86.

[8] 李鹏博、于立婷、王天琪. Java web 软件框架技术探析[J]. 通讯世界，2016,（14）：288.

[9] 梁北海. 基于污点分析的 Java Web 程序脆弱性检测方法研究[D]. 华中科技大学，2015.

[10] 霍剑峰. 基于 JAVA WEB 的虚拟数字图书电子商务平台设计与实现[D]. 吉林大学，2015.

[11] Elliotle R. Java network programming O'Reilly[M]. 北京: 机械工业出版社, 2016.

[12] Ted Husted. Struts In Action[M]. 北京: 电子工业出版社, 2015.

[13] Richard M. Enterprise Javabeans[J]. 北京: 中国青年出版社, 2016.

[14] Bruce E. Thinking in Java[M]. 北京: 人民邮电出版社, 2015.

[15] Chuck Gavaness. Programming Jakarta Struts2nd Edition[J]. 北京: 电子工业出版社, 2014.

附录

部分程序代码（创建订单逻辑代码）：

```
@RequestMapping(value = "addOrder.htm", method = RequestMethod.POST)
  @ResponseBody
  public Result addOrder(@RequestBody OrderReq orderReq) {
    try {
      Preconditions.checkArgument(orderReq.getUserId() != null, "用户 ID 不可为
    空");
      Preconditions.checkArgument(orderReq.getOrderGoods() != null ||
    orderReq.getOrderGoods().size() != 0, "订单信息不可为空");
      logger.info("【订单系统】请求下订单，请求参数:{}", new
    Gson().toJson(orderReq));
      List<OrderGoods> orderGoods = orderReq.getOrderGoods();
      for (int i = 0; i < orderGoods.size(); i++) {
    Preconditions.checkArgument(orderReq.getOrderGoods().get(i).getWineId() !=
    null, "第" + i + "件商品 ID 不可为空");
    Preconditions.checkArgument(orderReq.getOrderGoods().get(i).getWineNum()
     != null, "第" + i +"件商品数量不可为空");
      }
      double wineTotalPrice = 0;
      for (OrderGoods orderGood : orderGoods) {
        boolean b = wineStockService.checkStock(orderGood.getWineId(),
    Long.valueOf(orderGood.getWineNum()));
        if (!b) {
          logger.error("【订单系统】添加订单失败,酒品" +
    orderGood.getWineId() + "号，库存不足!!!");
          return Result.buildFailedResult(-1, "添加订单失败,酒品" +
    orderGood.getWineId() + "号，库存不足!!!");
        }
        int i = wineStockService.updateStock(orderGood.getWineId(),
    Long.valueOf(orderGood.getWineNum()));
        if (i > 0) {
          Wine wine = wineService.selectWineByWineId(orderGood.getWineId());
          orderGood.setWineName(wine.getWineName());
          wineTotalPrice += orderGood.getWineNum() * wine.getWinePrice();
        } else {
          new UpdateFailException("第" + orderGood.getWineId() + "个酒品库存
    更新失败");
```

```
            }
        }
        Order order = new Order();
        Long orderId = DateTime.now().getMillis();
        order.setOrderId(orderId);
        order.setUserId(orderReq.getUserId());
        order.setWineTotalPrice(wineTotalPrice);
        order.setOrderGoods(orderGoods);
        order.setOrderDateStart(DateTime.now().toDate());
        int i = orderService.insertOrder(order);
        if (i > 0) {
            logger.info("【订单系统】添加订单成功");
            return Result.buildSuccessResult(order);
        }
        logger.info("【订单系统】添加订单失败");
        return Result.buildFailedResult(-1, "添加订单失败");
    } catch (IllegalArgumentException ie) {
        logger.error("【订单系统】非法参数异常", ie);
        return Result.buildFailedResult(-1, "非法参数异常");
    } catch (Exception e) {
        logger.error("【订单系统】添加订单时出现异常", e);
        return Result.buildFailedResult(-1, "服务器开小差了~~ 请稍后重试");
    }
}
```