



中北大学  
NORTH UNIVERSITY OF CHINA

# 毕业设计说明书

## 基于 Spring Boot+HarmonyOS 的餐饮管理 与服务系统的设计与实现

学生姓名：冯佳和 学号：2113042621

学 院：软件学院

专 业：软件工程

指导教师：李杰 乞国钰

2025 年 06 月

# 基于 SpringBoot+HarmonyOS 的餐饮管理与服务系统的设计与实现

## 摘 要

由于国内外卖行业迅速发展,越来越多的外卖平台加入到竞争行列中来,对于许多中小型的餐饮企业来说,都受到了来自这些第三方外卖平台高昂佣金以及营销政策上的制约而面临严峻的问题:利润空间被不断压缩以至于出现了入不敷出的现象。因此为了改变这一现状,在本文的研究过程中提出了一种建立开放式外卖系统的想法并以此为基础展开了一系列相关的工作,希望通过这样的方式能够使得餐饮企业在一定程度上摆脱对外部平台的过度依赖,进而节约一部分经营的成本从而提高其收益能力。

本文的主要内容是利用 Java 语言编写了一个包含两个操作员角色即管理者和使用者的一个完整的开放式餐饮管理系统。其中管理员主要承担着员工的信息维护、菜单的分类管理、菜品种类管理、日常销售记录的数据查询统计和各种报表文件生成的任务;而用户则是可以直接访问网站查看商品信息、添加购买的商品到自己的购物车里边去、更新自己保存的常用收货地址或者个人信息等。当该项目完成了整个的需求分析阶段之后就进入了功能模块的设计过程,紧接着又进行了相应的数据库结构的设计工作。然后分别介绍了各个功能模块的具体实现方案。最后就是关于项目的整体运行环境介绍,包括了客户端部分的技术栈和服务端的部分技术栈的选择情况。

本文的前端部分采用了 HTML5+AJAX+Vue+Element-ui 的技术组合来进行开发,管理端的后台部分主要是由 Spring+SpringMVC+SpringBoot 和 Mybatis-plus 这三个主流开源组件共同组成而成的微服务架构体系。移动终端部分遵循的是适应华为鸿蒙操作系统 (HarmonyOS) 的原则,数据库方面使用的依然是 MySQL 数据库,同时因为考虑到不同场景下业务逻辑的不同特点所以选择 Redis 来做缓存,项目最终部署到了 Linux 系统上并通过 Tomcat 对其做了容器化处理。

**关键词:** 点餐流程, 餐厅经营, 效率提升, 顾客体验, 经营报表

# Design and Implementation of a Catering Management and Service System Based on SpringBoot and HarmonyOS

## ABSTRACT

With the booming development of China's food delivery industry, competition among third-party platforms has significantly impacted many catering enterprises. Numerous restaurants face challenges such as high commission fees and restricted marketing strategies imposed by third-party delivery platforms, leading to reduced profits or even operational losses. To address these issues, this paper proposes the development of an independent food delivery system that enables catering businesses to directly connect with customers, eliminating reliance on third-party platforms.

Through market research and analysis, this study focuses on small and medium-sized restaurants operating in urban areas. By integrating regional market characteristics and consumer behavior patterns, we aim to create an open food delivery platform specifically tailored for these establishments. This system will empower participating restaurants to reduce dependence on third-party platforms, lower operational costs, and improve profitability. Simultaneously, consumers will benefit from enhanced dining options, diversified menus, and personalized services.

This project is developed using Java, with system roles including administrators and users. Administrators can manage staff information, category classification, menu items, business data statistics, and operational report exports. Users can browse menus, manage shopping carts, update addresses, and modify personal information. After completing requirements analysis, functional module design, and database design, the implementation details of each functional module are thoroughly described. The frontend of this graduation project employs HTML5, AJAX, Vue, and Element-UI frameworks. The backend utilizes Spring, Spring MVC, Spring Boot, and MyBatis-Plus for the management system, while the mobile terminal targets compatibility with HarmonyOS. The database uses MySQL with Redis for caching, and the deployment environment is configured with Linux and Tomcat.

**KEY WORDS:** Ordering Process, Restaurant Operations, Efficiency Improvement, Customer Experience, Business Reports

## 目 录

1 绪论.....	1
1.1 研究背景和意义.....	1
1.2 国内外研究现状.....	2
1.2.1 国内研究现状.....	2
1.2.2 国外研究现状.....	3
1.3 研究目标和内容.....	4
1.4 论文组织框架.....	5
2 开发技术及开发环境.....	6
2.1 开发技术.....	6
2.1.1 Vue 简介 .....	6
2.1.2 Axios 简介 .....	6
2.1.3 Java 简介 .....	6
2.1.4 Spring Boot 简介 .....	7
2.1.5 MyBatis-Plus 简介 .....	7
2.1.6 MySQL 简介 .....	8
2.1.7 Maven 简介 .....	8
2.1.8 JDK 简介 .....	9
2.1.9 MD5 简介 .....	9
2.1.10 ArkTS 简介 .....	9
2.2 开发环境.....	10
3 系统分析.....	11
3.1 设计思想.....	11
3.2 设计原则.....	11
3.3 可行性分析.....	11

3.3.1 技术可行性.....	11
3.3.2 操作可行性.....	12
3.4 需求分析.....	12
3.4.1 功能性需求.....	12
3.4.2 非功能性需求.....	15
4 系统概要设计.....	16
4.1 系统用例图.....	16
4.2 系统功能模块设计.....	23
4.3 数据库设计.....	23
4.3.1 数据库概念结构设计.....	23
4.3.2 数据库表结构设计.....	30
4.4 安全性与可靠性设计.....	35
5 系统详细设计.....	37
5.1 系统功能模块设计.....	37
5.1.1 菜品管理模块.....	37
5.1.2 套餐管理模块.....	39
5.1.3 订单管理模块.....	40
5.1.4 鸿蒙移动端模块开发.....	41
6 系统测试.....	45
6.1 测试方法.....	45
6.2 功能测试.....	45
6.2.1 登录功能测试.....	45
6.2.2 员工管理测试.....	45
6.2.3 套餐管理测试.....	46
6.2.4 安全测试.....	47

---

7 结论与展望.....	49
参 考 文 献.....	50
致 谢.....	52

## 1 绪论

随着国家经济迅猛增长，民众消费能力提升，推动了餐饮业的兴旺。特别是学生与上班族群体，由于紧张的学习与工作节奏，他们更倾向于节省时间。目前，人们对食品种类及口味的需求日益提高，对于便捷订餐方式也提出了更高的期待。因此，外卖服务行业应声而起，旨在节省用餐时间<sup>[1]</sup>，提供高效便捷的服务体验。市面上涌现出众多第三方外卖平台，例如美团、饿了么、百度等。众多餐馆开始依靠这些平台，但也面临着新的挑战。比如，平台间的竞争影响了商家的盈利，平台高额的佣金要求商家压缩利润进行促销活动，同时营销策略受限等问题<sup>[2]</sup>。于是，许多大型及连锁餐饮企业选择自主开发外卖系统，直接服务消费者。然而，中小型餐馆因资金和技术限制，依旧维持传统的订餐模式<sup>[3]</sup>。调研显示，中小餐馆与网络结合的程度不高，因此提出打造一个区域性的餐饮网络营销平台，针对城区中小餐馆，结合市场特性和消费者购物习惯，开发开放的餐饮服务系统，这对于解决餐饮业问题、提升市民服务水平具有深远意义。

### 1.1 研究背景和意义

当前餐饮行业数字化转型加速，但传统系统存在显著痛点，如设备协同能力弱，跨端开发成本高，交互体验较为割裂。

本系统对于餐饮管理与服务有着非凡的意义，如以下几点：

（1）全流程自动化：整合预订、点餐、库存、结算模块，减少 30%人工操作。通过 MyBatis-Plus 持久层框架实现订单实时持久化，响应时间 $\leq 200\text{ms}$ 。

（2）优化数据驱动决策：基于销售数据的智能采购预测，降低食材损耗 15%以上，可视化报表助力精准营销（如套餐组合推荐）。

（3）多终端无缝体验：HarmonyOS 的“一次开发，多端部署”能力，支持手机扫码点餐、平板菜单更新、智能手表催单，分布式数据管理确保跨设备状态同步（如换桌后订单自动迁移）<sup>[4]</sup>。

（4）个性化服务提升：基于用户历史的智能推荐（如过敏源提示、口味偏好），提高复购率 25%。

（5）双技术栈融合示范：验证 SpringBoot 企业级后端与 HarmonyOS 物联网前端的兼容性，为跨平台系统提供新范式。

本研究通过融合 SpringBoot 的高效后端能力与 HarmonyOS 的分布式生态，构建了新一代餐饮管理系统，直击行业在效率、体验、成本方面的痛点。系统不仅推动餐饮业向智能化转型，更为多终端协同的物联网应用提供技术参考，兼具实践价值与创新意义。

### 1.2 国内外研究现状

虽然餐饮管理系统的相关研究已经比较完善了,但是国内外的研究基本都是基于单端或者弱协同的结构(比如: Springboot+Android/iOS)。国内有华为推出的 HarmonyOS 可以做到多终端的协同,不过主要是面向家庭场景的应用;国外也推出了 ReactNative、Flutter 等跨平台框架用于构建底层的基础跨端应用开发,但没有很好的解决设备之间的深层次的交互协作的问题<sup>[5]</sup>。而且现在还没有人提出过基于 Spring Boot 后端和鸿蒙分布式前端的一整套系统方案应用于餐饮行业的研究,更别提关于整个订餐流程中的所有环节都能进行实时协同的例子(从顾客下单到厨房接单再到送餐人员的位置跟踪),所以这个项目就是为了解决这个问题而产生的。

#### 1.2.1 国内研究现状

在国内餐饮管理系统研究方面形成了以 SpringBoot 技术栈为基础,以 HarmonyOS 分布式生态为突破方向的发展态势:一方面,由于 SpringBoot 具有快速开发、微服务架构等特点,在我国 90%以上的餐饮管理系统中被作为后端主流框架;另一方面,配合 MyBatis Plus 以及 Vuejs 实现高并发下的大额订单(例如二维火可支撑超过 5000 TPS)分发,并采用前后端分离模式,然而仍然存在着三个方面的痛点:首先,餐饮业普遍存在的信息孤岛现象依然严峻,POS 机、供应链管理、CRM 模块之间难以达到深层次融合的效果,使得部分供应商只能依靠经验进行进货量估算,而食材浪费率仍然保持在了惊人的 20%;其次,餐饮商家对于三方平台的依赖程度非常之深,中小商家若想接入美团、哗啦啦等 SaaS 服务平台则要支付高额的服务费,且自身的数据主权也受到了一定程度上的削弱;最后,餐饮行业的 AI 赋能尚处于浅层阶段,主要集中在商品结算环节上,诸如图像识别类的应用较为常见,而在销售预测分析、动销规划以及动态库存管理等方面鲜有涉及<sup>[6]</sup>。近几年来,随着 HarmonyOS 的到来,餐饮行业的整体发展开始逐渐步入快车道:其中欧盈万酒店平台已经证实了其可以完美地运行于 SpringBoot 后端之上,并借助分布式技术实现了手机下单、厨显屏同步更新、智能手环提醒等功能,做到了真正意义上的跨端协同工作(响应时延<100 ms);此外还能够利用 NFC 标签完成“碰一碰”



的鸿蒙服务卡片推送功能,并根据用户的过往订货记录为其提供个性化的精准营销策略,有效提升了客户的二次购买概率达至到 25%;并且原生鸿蒙系统所具有的上百个系统级入口极大地方便了消费者的点餐操作过程,进一步减少了中小餐饮门店对于 Android 或 iOS 移动终端的依赖度,从而加快了整个餐饮产业向自主可控方向发展的步伐。至于未来的市场发展趋势,则是将逐步转向云边端协同架构体系下(即 SpringBoot 负责云端的核心业务逻辑运算, HarmonyOS 负责边缘端的厨显设备、智能厨电等硬件的轻量化远程控制),并在此基础上衍生出全链路智能化的趋势(即将上游供应链的信息打通,用于下游厨房的备料计划制定与调整)。

### 1.2.2 国外研究现状

从全球来看,餐饮管理系统的整体发展趋势可以总结为:一是 AI 深度驱动;二是生态开放化。具体来说,在技术架构层面上,主要由轻量化的 SaaS 平台主导 (Saas+PaaS+SaaS),依托公有云服务进行部署,快速接入到企业内部,且大多使用闭源的技术栈,因此限制了个性化定制的需求。而在功能创新方向,则更加侧重于基于大数据的数据智能的应用<sup>[7]</sup>。比如 Toast 利用机器学习算法分析过往订单的历史记录,从而精准地预测出食材的损耗情况,并最终使得企业的食品浪费下降到了 12%(准确率达到 87%)。此外,物联生态也更加强调底层硬件之间的互联互通程度。例如 Square 平台就打通了所有相关设备的 API 接口,让其下属的灶头可以自动向后厨调度中心上传当前的温度等信息;德国 Lightspeed 则实现了与冷藏库的物联网连接,这样就能够根据后台存储的食物种类自动调节冰箱中的温度和湿度条件等等。最后也是最重要的一个特点是,各国政府都在强调对于信息安全和个人隐私保护的重视。其中最为典型的就欧盟 GDPR 法规的要求,该法规规定必须保证个人隐私安全并满足相应的规范,否则将会面临巨额罚款甚至牢狱之灾。所以欧洲国家的相关系统均需要严格遵守这一法律的规定,尤其是像 The Fork 这样的大型互联网公司更是如此。但是需要注意的一点是,虽然欧美很多大厂都已经推出了自己的系统,但是在对国内的操作系统方面的兼容性却几乎是空白状态。目前唯一的一个例外就是 Harmony OS 的发布,它是一个典型的“去中心化”的解决方案,即把整个终端体系分成多个节点,然后每个节点都可以作为一个独立的主体来进行运行。这种分布式的设计方式无疑给未来的软件开发带来了新的思路和发展空间,尤其是在不同终端之间实现高效的信息共享和资源调配方面具有很大的优势。不过遗憾的是,无论是谷歌还是苹果都没有推出类似的解决方案或者产品。也就是说,现在

我们看到的所有移动终端上的应用程序都是按照传统的集中式模式进行编写的。这就意味着要想真正打破这个局面就需要付出更多的努力才行。而且由于缺乏统一的标准和技术框架的支持,“跨域”访问成为了另一个难题。比如我们在使用手机时经常会遇到这样一个问题——当我们将一台电脑接到路由器之后再打开网页浏览器的时候就会发现之前收藏夹里面的内容已经不见了踪影!这是因为我们的 PC 和智能手机所使用的网络地址并不相同造成的。如果我们想要解决这个问题就必须找到一种方法能够让这两者之间建立起某种联系才有可能实现真正的无缝衔接。当然这里所说的并不是简单的 URL 转换关系而是指两者拥有共同的语言体系或者说是一套通用的通信协议才能够相互理解对方的意思进而达到目的。目前市面上比较流行的做法有两种:其一是 Websocket 协议;其二则是 HTTP 协议<sup>[8]</sup>。前者通常应用于聊天室类的网站当中后者主要用于在线购物或社交分享等领域之中。这两种通讯机制各有优缺点无法完全取代彼此只能算是各有所长而已并没有孰高孰低的说法。至于哪种更好还是要看具体情况而定不能一概而论。总而言之不管采取何种方式进行操作我们都希望能够尽量减少不必要的麻烦以及降低维护成本这样才能确保整个业务流程顺利进行下去才能更好地提升用户体验感。

### 1.3 研究目标和内容

本文针对目前餐饮业存在的各种弊端,在对当前餐饮行业现状及未来发展趋势进行了充分调研的基础上,提出并设计了一个基于 Springboot+HarmonyOS 的餐饮管理系统和服务系统。通过对餐饮行业的实际调查发现其存在以下问题:1. 人工点餐模式下,餐厅工作效率低;2. 客户用餐体验差;3. 消费者信息反馈不及时;4. 管理者缺乏有效管控手段<sup>[9]</sup>。因此,根据以上情况,结合餐饮管理的实际需要,本论文所要完成的主要功能有以下几个模块:员工管理、菜品分类管理、套餐管理、订单管理等功能。在系统设计部分主要运用了面向对象的思想来搭建整个系统的体系结构,并且绘制出相应的 UML 图,同时定义好各个类之间的关系以及属性、字段,最终形成数据库建模图,保证系统中各表之间能方便地建立关联。在系统的技术选型上选择 Java 语言来进行项目开发工作,其中 IDE 工具选用 IDEA 进行软件开发,前端框架采用 Vue 框架,后端框架采用 Spring Boot 框架,使用 Mybatis-Plus 做持久化层框架,与 MySQL 数据库进行交互连接,最后完成系统的全部业务流程。在整个系统开发的过程中都本着以人为本的原则,尽可能做到人性化,使该系统成为一个真正意义上的 B/S 结构系统,让广大使用者都能直观而准确地了解系统所能提供的各项功能,进而提高他们的

积极性和参与度。并且在此基础上还提供了相关的用户手册和培训方案,使得他们可以迅速学会如何正确使用此系统<sup>[10]</sup>。另外也给一些想要学习本系统的学生或公司提供了很好的平台。这对于企业来说不仅可以大大降低人力成本,而且对于鸿蒙端的消费者也可以轻松地实现在线点单或者查看自己的历史订餐记录。

### 1.4 论文组织框架

绪论本章主要介绍餐饮管理与服务系统研究的意义、国内外相关领域研究现状以及本文所采用的技术及开发环境;

需求分析:分别从系统业务流程图(即系统模块)和系统数据流两个方面来描述该系统所需要实现的具体功能,并说明每个具体功能所涉及的数据关系及其存储方式等;

系统设计:根据前面的需求分析结果进行系统的概要设计和详细设计,同时确定数据库的设计思路;

系统测试:通过实际案例模拟系统运行过程并对各功能模块逐一进行测试以达到预期效果;

最后包括研究成果及展望两部分内容,其中第一部分是对完成任务后所得出的结果或理论知识加以概括性的论述,得出具有普遍性规律的东西,并在此基础上提出自己的一些见解,从而为进一步开展相关的研究工作奠定基础。

## 2 开发技术及开发环境

### 2.1 开发技术

#### 2.1.1 Vue 简介

Vue 是基于 JavaScript 语言开发的一款用来构建用户界面的渐进式的框架, 具有简单灵活且高性能的特点, 并受到越来越多的人欢迎, 在使用 Vue 进行前端开发时可以极大的减少前段开发的难度, 使开发者能更加专注地完成一个富于交互以及有动态特效的页面设计<sup>[11]</sup>。Vue 拥有自己特有的响应式的数据绑定机制, 它可以实现数据与视图之间的双向绑定关系, 这种双向绑定的方式很大程度上减轻了开发者对 DOM 的手动操作的工作量, 同时也能提升应用程序的整体运行效率及用户的体验感。伴随着 Web 技术的发展, 以及前端应用场景越来越复杂化, Vue 也正在经历着自身的不断完善与发展来应对更多的挑战。除了不断地进行性能方面的优化之外, Vue 还会不断的加入新的功能点, 从而支持更多不同的业务需要; 除此之外, Vue 也会逐渐加强自身和其他前沿前端技术的结合程度, 以此达到更好的适配能力。

#### 2.1.2 Axios 简介

Axios 是一种 JavaScript 的 HTTP 客户端, 它遵循 Promise API, 适用于浏览器和 node.js。Axios 不是一种新技术, 而是对原生 XMLHttpRequest (XHR) 对象的封装, 并基于 Promise 实现, 使其更符合现代 JavaScript 的异步编程模式<sup>[12]</sup>。

#### 2.1.3 Java 简介

Java 是一种广泛使用的高级编程语言, 由 Sun Microsystems (现为 Oracle 公司的一部分) 于 1995 年首次发布。它被设计为具有尽可能少的实现依赖性, 这意味着 Java 编写的应用程序可以在任何支持 Java 虚拟机 (JVM) 的平台上运行, 这被称为 “一次编写, 到处运行” (Write Once, Run Anywhere, WORA) 的特性<sup>[13]</sup>。核其心特性如下:

(1) 面向对象:Java 与 C++一样 是一种面向对象的语言, 支持封装、继承和多态等基本面向对象编程的基本概念。可以让项目变得模块化和更容易维护。

(2) 平台无关性:Java 代码被编译成字节码, 而不是直接编译成机器码。字节码可以在任何安装了 JVM 的设备上运行, 从而实现了跨平台兼容性。

(3) 多线程:Java 内置了对多线程编程的支持, 使得编写并发程序更加容易。它提

供了线程管理、同步和通信等机制。

(4) 安全性:Java 提供了一个安全管理器, 允许应用程序定义其安全策略。它还通过沙箱 (Sandbox) 机制限制了代码对系统资源的访问, 增加了程序的安全性。

(5) 可移植性: Java 代码可以在任何支持 JVM 的平台上运行, 无需修改。

(6) 分布式: Java 支持网络通信和远程调用, 使得开发分布式应用程序变得容易。

#### 2.1.4 Spring Boot 简介

Spring Boot 是一个开源的 Java 基础框架。它简化了基于 Spring 的应用程序的初始搭建和开发过程, 使得开发者可以快速启动和运行 Spring 应用程序<sup>[14]</sup>。以下是 Spring Boot 的详细介绍: 其核心特性如下:

(1) 自动配置:Spring Boot 的核心特性之一是自动配置。例如, 如果类路径中存在 H2DataSource, Spring Boot 会自动配置内存数据库。

(2) 独立运行:Spring Boot 应用程序包含内嵌的 HTTP 服务器 (如 Tomcat、Jetty 或 Undertow), 使得应用程序可以打包成一个独立的 JAR 文件, 通过简单的 java -jar 命令运行。

(3) 微服务支持:Spring Boot 与 Spring Cloud 等微服务框架无缝集成, 支持微服务的快速开发和部署。

(4) 社区支持:Spring Boot 拥有一个庞大的社区和生态系统, 提供了大量的插件和扩展, 使得开发者可以轻松地扩展应用程序的功能。

(5) 测试支持:Spring Boot 提供了对 Spring Test 和 JUnit 的支持, 使得编写和运行单元测试和集成测试变得容易。

(6) 依赖管理:Spring Boot 使用 Maven 或 Gradle 作为构建工具, 并提供了 spring-boot-starter 依赖, 简化了依赖管理。

#### 2.1.5 MyBatis-Plus 简介

MyBatis-Plus 是一个开源的持久层框架, 是 Mybatis 的升级版, 在此基础上优化了优化了单表的 CRUD 操作, 并提供了类如代码自动生成器与分页插件内容, 用于简化数据库操作和简化数据库事务管理。它是一个半自动的持久层框架, 介于全自动化的 Hibernate 和完全手动编写 SQL 语句之间<sup>[15]</sup>。而 MyBatis 允许开发者编写 SQL 语句, 并将其与 Java 对象映射, 从而实现数据的持久化。其核心特性如下:

- (1) SQL 映射:MyBatis 允许开发者编写 SQL 语句,并将其映射到 Java 对象。这种方式使得 SQL 语句的编写和维护更加灵活和直观。
- (2) 性能:MyBatis 的性能通常优于全自动化的 ORM 框架,因为它允许开发者编写优化的 SQL 语句,并且减少了不必要的数据库交互。
- (3) 可扩展性:MyBatis 支持插件机制,允许开发者扩展其功能。例如,可以通过插件实现分页、缓存等高级功能。
- (4) 事务管理:MyBatis 支持声明式事务管理,可以通过注解来管理事务的回滚等行为<sup>[16]</sup>。
- (5) 动态 SQL:这使得处理复杂的查询条件变得更加简单。
- (6) 集成 Spring:MyBatis 可以与 Spring 框架无缝集成,利用 Spring 的依赖注入和事务管理功能,简化了 MyBatis 的配置和使用。
- (7) 支持多种数据库:MyBatis 支持多种关系型数据库,如 MySQL、Oracle、SQL Server 等,这可以让它可以在不同的数据库环境中使用。

#### 2.1.6 MySQL 简介

MySQL 提供了多种存储引擎,如 InnoDB 和 MyISAM,其中 InnoDB 支持事务处理和行级锁,适用于高并发读写操作的场景。MySQL 支持主从复制、分区、集群等技术,可以扩展到大规模数据存储和处理需求。它可以通过水平和垂直扩展满足不断增长的业务需求。MySQL 提供事务支持,包括 ACID。MySQL 提供了用户权限管理、数据加密、审计日志等安全特性,保护数据库系统和数据的安全性<sup>[17]</sup>。

#### 2.1.7 Maven 简介

Maven 是一个由 Apache 软件基金会维护的项目管理和构建自动化工具。Maven 的设计目标是使得构建、测试和部署软件项目的过程自动化,并且简化管理项目依赖的复杂性。

Maven 使用一个名为 POM (Project Object Model) 的 XML 文件来描述项目的构建过程、依赖关系和其他配置信息。这些依赖库可以是其他 Java 库,也可以是 Maven 插件,它们被存储在 Maven 的中央仓库中,Maven 会根据 POM 文件自动下载这些依赖。如图 2.1、2.2。



图 2.1 Maven 生命周期图

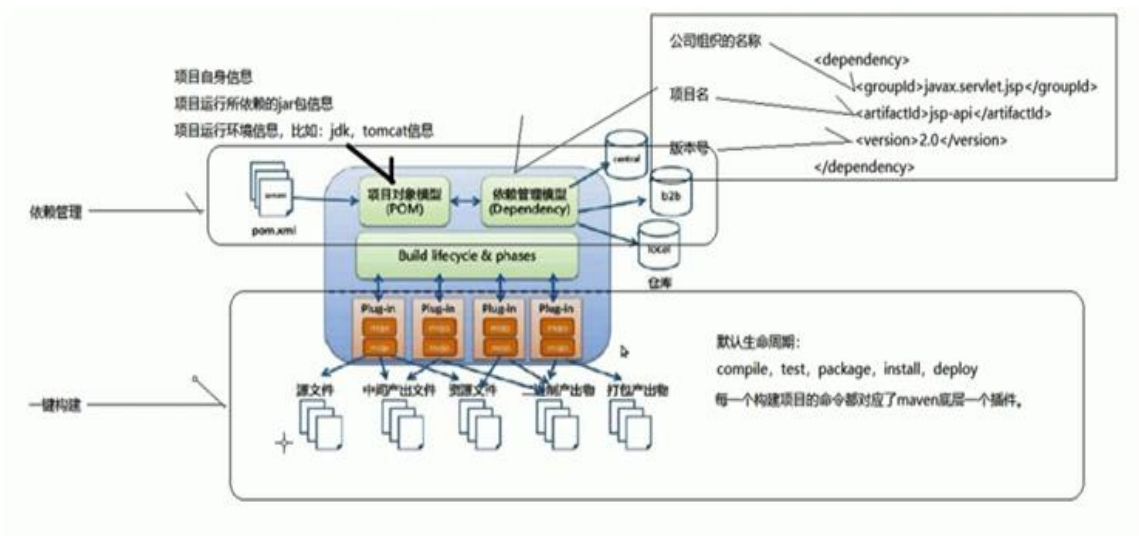


图 2.2 Maven 原理图

2.1.8 JDK 简介

JDK，全称为 Java Development Kit（Java 开发工具包），是由 Oracle 公司（之前是 Sun Microsystems）发布的用于开发 Java 应用程序的一套软件开发环境。它提供了 Java 语言的核心 API、编译工具、调试工具以及其他一些工具和库，使得开发者能够编写、编译和运行 Java 程序<sup>[18]</sup>。

2.1.9 MD5 简介

MD5 通过一系列复杂的操作，包括位运算、位移、条件逻辑等，将任意长度的输入数据转换成固定长度的输出。

2.1.10 ArkTS 简介

ArkTS 是鸿蒙生态的核心开发语言，基于 TypeScript 扩展，专为声明式 UI 设计，

其配套框架 ArkUI 提供组件化开发与响应式布局能力，通过 @State、@Link 等状态管理实现高效渲染。鸿蒙技术栈依托 方舟编译器 提升性能，支持分布式能力（跨设备协同）和原子化服务（免安装服务卡片）<sup>[19]</sup>。该技术栈为鸿蒙全场景设备（手机/车机/穿戴等）提供统一开发范式，兼顾效率与性能。

2.2 开发环境

- （1）网络环境：局域网。
- （2）硬件环境：笔记本电脑、运存 16G、硬盘 512G。
- （3）操作系统：Windows10，Harmony。
- （4）开发工具和集成开发环境：大学生健康档案管理系统的开发环境构建在一个稳定而高效的技术栈之上。系统采用 Windows 操作系统作为基础运行平台，移动端以适配鸿蒙操作系统为主，确保了开发过程的兼容性和稳定性。后端开发选择了 Java 语言，并以 JDK 17 版本为核心，利用其跨平台的特性和强大的社区支持。此外，系统的开发过程中还涉及到了 Maven 的使用，这是一个项目管理和构建自动化工具，负责处理项目的依赖管理和生命周期管理。为了进一步简化 Java 代码的编写，系统还集成了 Lombok 插件，它通过注解的方式减少了样板代码<sup>[20]</sup>。整个开发环境的配置旨在提供一个高效、稳定且易于维护的平台，以便开发团队能够专注于系统功能的开发和优化。见表 2.1。

表 2.1 开发环境表

开发环境组件	描述
操作系统	Windows, Harmony
JDK 版本	17
开发工具	IntelliJ IDEA
前端运行环境	node.js
数据库系统	MySQL
前端框架	Vue.js
后端框架	Spring Boot
版本控制系统	Git
项目管理工具	Maven
数据库操作框架	MyBatis-Plus
代码简化插件	Lombok



## 3 系统分析

### 3.1 设计思想

采用经典的分层架构：明确划分为视图层、业务逻辑处理层和数据持久化层，模块化设计可以提升整体开发效率。

后端框架：选择 Spring Boot 作为后端框架，它以其快速开发、易于部署和对微服务架构的生态支持而受到青睐。Spring Boot 加速了从原型到生产的过程。数据库：MySQL 被选为主要的关系型数据库，用于存储系统中的主要信息。前端框架：选择 Vue.js 等现代化的前端框架，实现用户友好的界面和良好的交互体验。

### 3.2 设计原则

（1）单一职责原则（Single Responsibility Principle, SRP）：

对于单一职责原则最基本的理解就是：一个类，应该只有一个可以导致其发生变化的原因。每个模块、类或方法应该只有一个单一的功能或责任。通过采用明确的职责划分和模块化设计，代码结构将变得更加清晰，各个组件之间的依赖关系得以简化。

（2）开放封闭原则（Open/Closed Principle, OCP）：

开放-封闭原则就是：软件实体（类、模块、函数等）应该是可以扩展的，但是不可修改。这种做法有助于减少对现有代码的影响，提升系统的稳定性和可扩展性。

（3）依赖倒置原则（Dependency Inversion Principle, DIP）：

在软件设计中，高层模块不应直接依赖于低层模块，而应共同依赖于抽象。

### 3.3 可行性分析

#### 3.3.1 技术可行性

餐饮管理与服务系统的技术可行性分析显示，该项目在技术层面具备实现的潜力和条件。系统采用 Windows 操作系统，利用其稳定性为开发和运行提供基础。开发工具选用了功能强大的 IntelliJ IDEA，支持高效的 Java 语言开发，同时前端技术采用 Vue.js 框架，这使得用户界面构建变得简洁且易于维护。后端则采用 Spring Boot 框架，简化了基于 Spring 应用的搭建和开发，同时集成了 MyBatis ORM 框架，便于数据库操作。

数据库方面，选择 MySQL 作为数据存储解决方案，其开源性和广泛的社区支持为数据的稳定性和安全性提供了保障。此外，系统的代码结构清晰，易于维护和扩展，技

术选型均考虑了当前流行趋势和社区支持，有利于系统的长期发展。

系统的功能模块经过了详细的测试，包括用户管理、后台管理、权限管理等，测试结果表明各模块功能正确且稳定。系统测试进一步验证了整个系统的集成和运行效果，确保了系统的可靠性。因此，从技术角度来看，大学生体检档案管理系统的开发不仅是可行的，而且具备了高效、安全、稳定运行的潜力。

### 3.3.2 操作可行性

餐饮管理与服务系统的操作可行性分析强调了用户界面的直观性、操作流程的简洁性以及个性化设置的可用性。无论是鸿蒙端用户查看订单信息，还是后台人员管理餐品信息，都设计为仅需简单几步即可完成。

系统的响应式设计确保了跨设备的一致体验，而强大的后端支持则保证了操作的快速响应。

安全性方面，系统通过细致的权限管理保护用户数据，使用户可以安全地进行操作。通过全面的用户测试，包括功能测试和可用性测试，系统显示出高度的用户操作可行性，能够满足不同用户群体的需求。

## 3.4 需求分析

### 3.4.1 功能性需求

该系统的用户角色有管理员和一般用户两种类型,其中管理员使用的是后台管理端来实现对各种资源的管理和处理;而一般用户则是使用前端来进行相关业务的操作.具体来说,管理员具有员工信息管理、分类管理、菜品信息管理、套餐信息管理、订单明细查询、订单管理、查看营业数据分析图、导出运营数据等功能模块;一般用户拥有菜品信息展示与详情显示、商品规格选择、购物车添加/删除商品、收货地址维护、订单列表浏览、订单提交付款等功能模块。

该系统主要有三个角色，分别为鸿蒙端的顾客，后台管理端的员工，以及 admin 三个角色，每个模块的主要的功能分别如下：

#### 1. 员工管理：

管理员可以对系统的员工信息进行操作，在系统后台中可以查看所有用户的详细信息;并且支持按照关键字搜索功能，并且可以通过添加新用户、编辑已有用户以及删除原有用户的功能来完善系统中的用户数据等;同时还可以通过设置是否启用某条记录来实

现禁用某个用户的功能,如果该用户已经被管理员禁用了,则无法使用该账号登陆到本系统当中去。

## 2. 分类管理:

管理员把所有的分类信息查出并以列表形式显示在分类管理主页上。(2)还可以进行新增、修改或删除菜品/套餐分类的操作。

## 3. 菜品管理:

在菜品管理模块中,管理员可以将所有菜品的信息从数据库中读取,并且按照一定的规则将其分为若干个数据集,在网页上显示出来;同时,还能够完成添加、删除和修改的操作(当然需要用户输入相应的密码);并且可以根据用户的关键词来进行模糊搜索以找出相关的菜品信息等;此外,还能改变当前菜单项的状态,使它处于“已下架”或者“可销售”的状态等等。还有就是对于某些菜单项目而言,如果想对其做相同的处理,则可以用鼠标点击这些菜单项目的名称来实现批处理的功能。

## 4. 套餐管理:

在套餐管理主页上,管理员可以调出所有套餐的信息,并按一定规则(比如:时间)将其分成若干页来显示给用户;同时管理员也可以对套餐的相关数据项做增加、删除和修改等操作;此外,通过输入关键词,系统还支持模糊查询功能以帮助管理员快速定位到目标套餐信息;最后,针对某一类或某几类套餐,管理员可以对其进行暂停销售或恢复销售的操作。另外,为了提高系统的使用效率及安全性,本模块还提供了对所选套餐信息的批量处理功能。

## 5. 订单管理:

对于管理员来说,需要对所有订单进行浏览和查看;同时也可以通过输入订单编号、或者按照起始时间或终止时间来查找订单,并且可以查询出具体某笔订单的所有信息内容,还可以更改某个订单状态等操作功能实现。

鸿蒙用户端可以分为菜品信息管理,购物车增加与删除,个人送餐地址管理,下单,订单情况查询五大模块。用例图如图 3.2 所示:

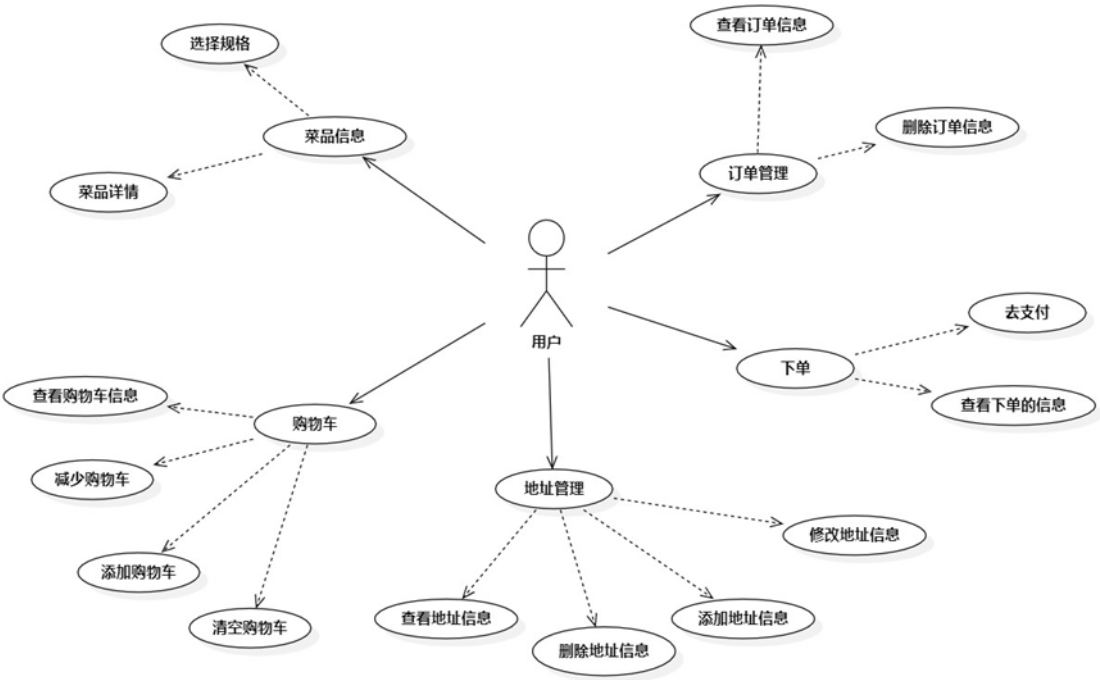


图 3.2 前台用户用例图

每个模块的主要的功能需求分别如下：

1. 菜品信息：用户通过登录用户端后可以浏览、搜索商家所发布上线销售的商品及套餐;并且可对商品进行口味选择以及规格的选择，在选中某个商品时会弹出该商品的具体详情页面，若选择了某一套餐则会显示此套餐下所有商品列表页
2. 购物车：当用户选择了想要的菜谱或者套餐及相应的口味规格后就可以将其加入购物车了,在此页面上可以查看当前购物车内的菜品或套餐信息,同时也可以根据自己的需求来对购物车内的菜品/套餐信息进行增加、减少、更改等操作,并可将购物车内所有商品一次全部清除掉。
3. 地址管理：后台地址管理界面设计:通过此模块,系统管理员能够查看所有已经存在的订单收货地址信息，并且可以根据需要对此类已有的地址进行增删操作;同时也可以在此处添加新的地址记录,在此功能中，如果想要设置某条地址作为默认的配送地址，则可以通过选择该地址前的复选框来实现。地址管理后端页截图当用户选择了某个订单中的商品提交订餐请求时，系统将会根据当前登录用户的个人资料获取其注册时填写的信息，包括姓名、电话号码等基本信息以及所绑定的所有地址信息，然后根据这些数据从数据库中检索出符合要求的相关内容以供用户确认或者重新输入相关信息。
4. 下单：当用户确定购买的菜品加入购物车，并且选中所有需要的商品后,点击“提

交”按钮将触发 ajax 请求,此时后台根据当前登录用户的 id 和菜品表中的数据计算出最终的价格及运费金额(此处假设系统提供免费配送服务),并将结果返回给前台;即为用户填写完相关信息后提交后的界面。其中红色框部分是订单页面的内容,包括了结算总价、优惠券抵扣情况等内容;绿色方块部分则是添加收货人地址的信息窗口。当用户确认无误后可以继续下单操作,若在此期间发现错误还可以回到上一步修改内容后再重新提交订单。同时为了方便客户后续查询自己曾经点过的餐品记录,也可以直接在这里查看之前的消费记录列表或再次下单。订单付款完成后的效果图在该页下方还有两个链接:一个是针对此次订单发起退订申请,另一个就是再下一次外卖时可以直接从这里进入点菜环节。

5. 订单管理:用户进入订单管理界面,可检索全部历史订单记录。用户可选取任一订单,浏览其详细资料,并有权申请退款或复购该订单。

### 3.4.2 非功能性需求

餐饮管理系统—管理员界面简便,安全级别高,运行流畅,兼容多数操作系统。采用 Java 开发,数据库工具高效,后台存储数据量大且稳固。以流程化管理优化管理成效,实现并发 1000,性能显著增强。

#### 1. 性能处理:

- (1) 并发度: 1000 (Tomcat)
- (2) 响应时间: 5s
- (3) 事务处理: 2 分钟
- (4) 可扩展性: 基于接口的 MVC 架构
- (5) 安全性: 基于角色的权限管理

#### 2. 硬件环境

处理器: Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz 2.11 GHz

内存容量: 16G

硬盘容量: 512G

#### 3. 软件环境

操作系统版本: Windows 10

系统软件: IntelliJ IDEA、DevEco Studio

数据库管理系统: MySQL、Redis

## 4 系统概要设计

依托 Spring Boot 与 HarmonyOS 技术构建的餐饮外卖平台，涉及三种用户角色：管理后台的全能管理员、负责日常管理的普通后台员工以及使用前台的消费者。全面负责后台系统的管理员，具备系统内的所有操作权限，能够登录进行管理。普通后台员工可登录管理系统，执行菜品、套餐、订单和数据的分类管理与统计分析。消费者通过前端页面登录，能够浏览菜品、加入购物车、设定收货地址、在线订餐，并在个人中心查看订单详情。

### 4.1 系统用例图

根据系统的各个参与者，设计出系统用例图，其用例图如下图 4.1 所示：

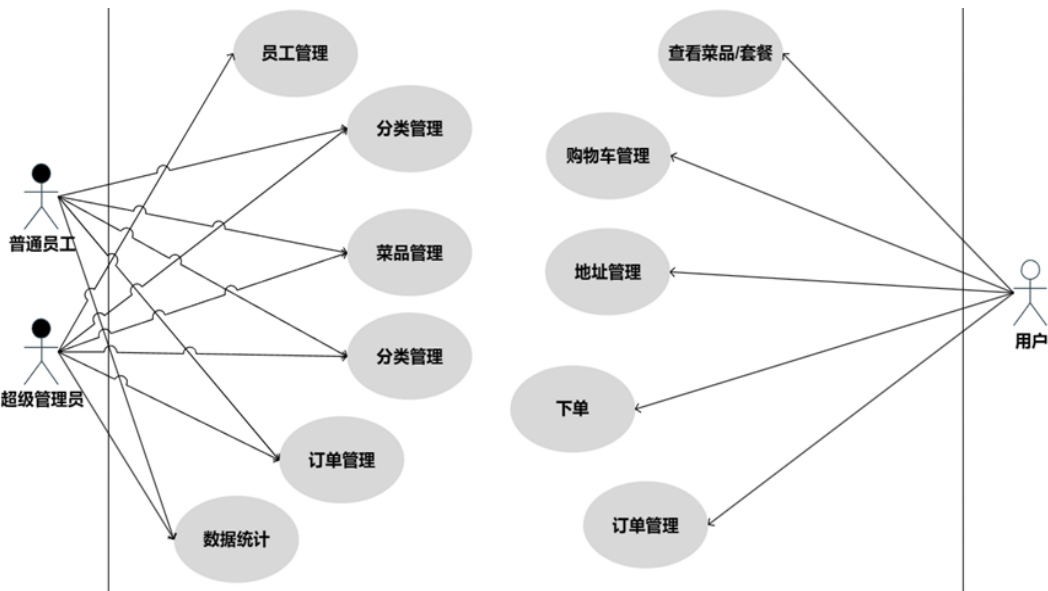


图 4.1 系统用例图

1. 员工信息管理：该功能模块涵盖了对普通员工的添加、调整及移除操作。在员工信息录入后，可选择仅保存或者保存后继续录入新员工信息。修改员工信息时，系统将显示数据库中保存的资料，同时允许对员工状态进行修改，包括启用或禁用用户，用例图如下图 4.2 所示。

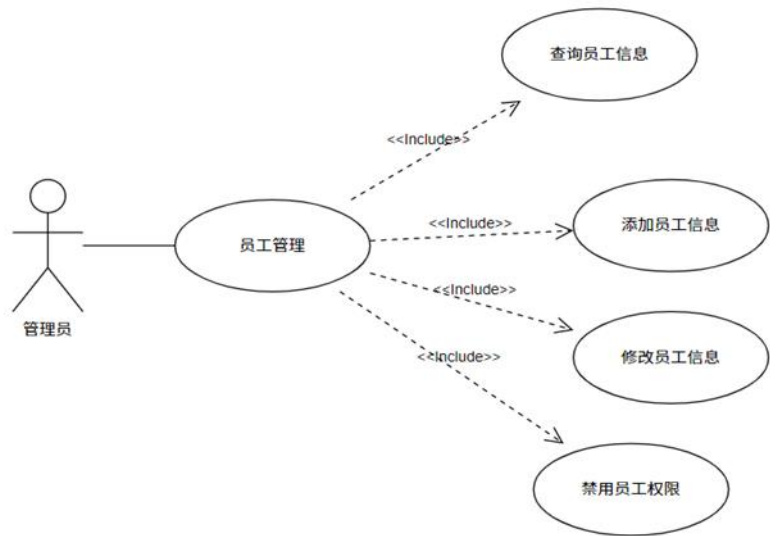


图 4.2 员工管理用例图

2. 分类管理：在分类管理范畴内，主要包括菜品与套餐两大类。该管理系统的排序功能旨在对类别进行升序排列，其中数字 1 代表最高优先级。数字越小，其在移动端用户界面中的展示位置越靠前。数字 1 作为默认首页显示，若多个类别均为 1，则以最近更新时间为排序依据。在编辑过程中，系统将访问数据库，提取保存的数据并在页面上进行展示。用例图如下图 4.3 所示。

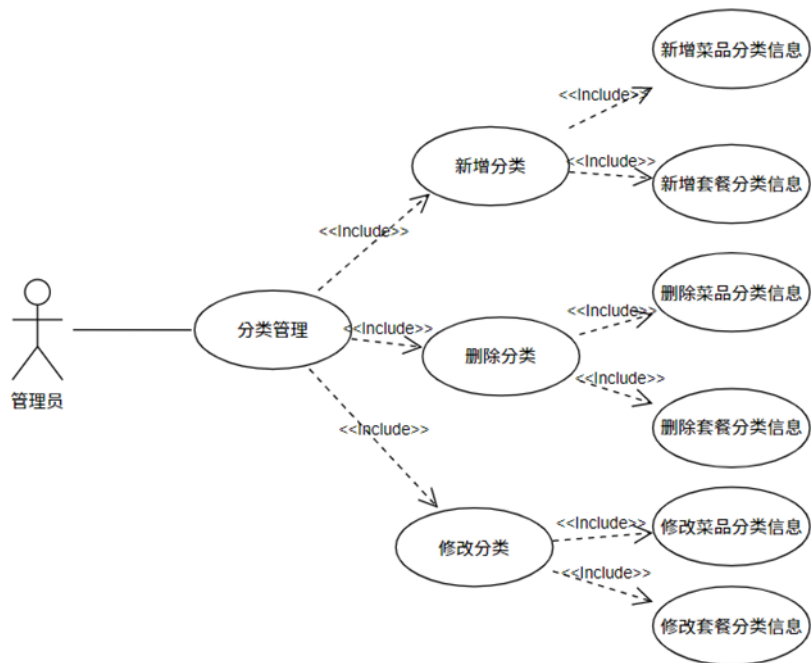


图 4.3 分类管理用例图

3. 菜品管理：菜品管理包含标准的创建、读取、更新和删除操作，其特色在于新增菜品时按分类进行，确保分类与菜品相对应。菜品可通过文件上传下载实现本地化处理，

并在移动端展示。在新增或修改菜品时，用户可选择个性化口味。批量删除菜品需先将其设为停售状态（即“status=0”），然后才能执行删除。调整菜品销售状态即改变其当前启用或停售状态。用例图如下图 4.4 所示。

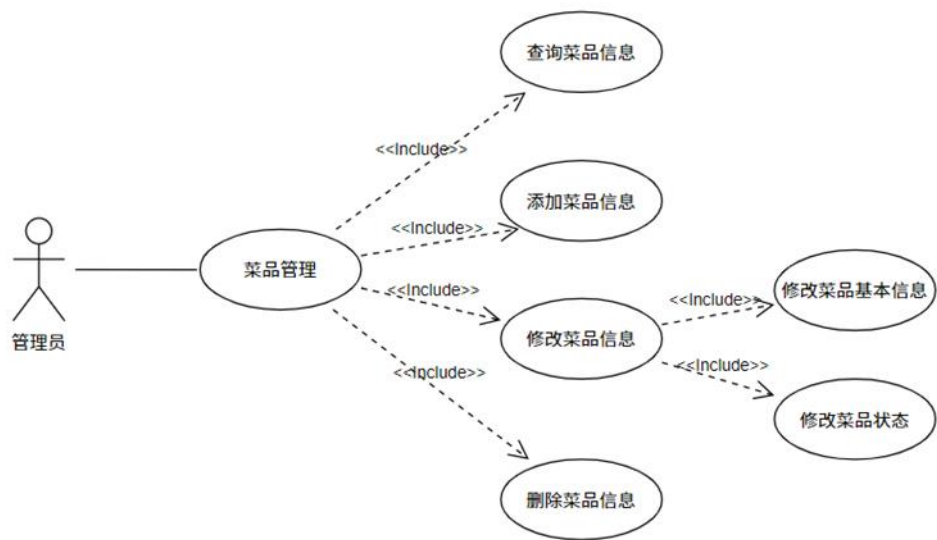


图 4.4 菜品管理用例图

以移除菜品为例，模拟该功能模块的事件流程。在执行菜品移除时，一个关键约束是菜品必须能够加入套餐中。换言之，若某菜品已被包含在正在销售的套餐内，则不允许对其进行删除操作，否则将引发异常。套餐若因删除菜品而不完整，便失去了其价值。因此，删除菜品的用例规范如下表 4.1 所示。



表 4.1 删除菜品规约表

用例名称:	删除菜品
用例 ID:	UC1001
角色:	后台系统管理员
前置条件:	管理员角色已登录、新增菜品同时此菜品状态为起售
基本事件流:	<div>1. 管理员有权优先将菜品撤下销售架，此时菜品状态将更新为”停售”，即标记为 “status=0”</div> <div>2. 管理员随后可以根据实际需要，对单个或多个菜品执行删除操作。</div> <div>3. 菜品一旦被删除，用户在移动端将无法检索到，同理，处于停售状态的菜品同样不对外显示。</div>
扩展事件流:	<div>1. 若某菜品包含于套餐内时，其不可被移除，仅在相关套餐停止销售时，该菜品才具备删除条件。</div> <div>2. 若菜品当前不宜销售，但又不宜彻底删除，可选择 “暂停销售” 以保留，待条件允许时，再选择 “恢复销售” 。</div> <div>3. 对于不再计划销售的菜品，可进行集中移除操作。</div>
后置条件:	对菜单进行剔除操作，确保最新的菜单或套餐信息得以数据库存储。

4. 套餐管理：套餐管理与菜品管理在本质上类似，区别在于套餐包含多个菜品的层级结构。在创建新套餐时，必须先从现有套餐类别中进行选择。新增菜品过程中，可以通过对现有菜品类别的模糊搜索，实现批量添加。删除菜品前，需先将菜品设置为停售状态，即标记为 “status=0”。在修改套餐时，可以自行删除其中的菜品。用例图如下图 4.5 所示。

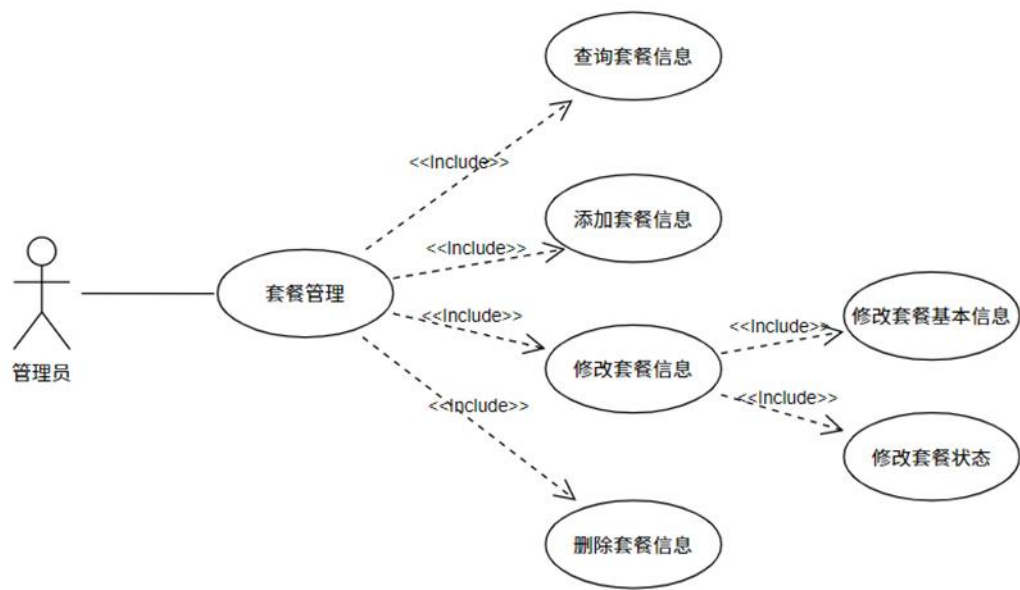


图 4.5 套餐管理用例图

在创建新套餐过程中，起初需选定既有套餐类别。在该类别之下，我们可以增添多样化的菜品。若需添加菜品，可通过搜索框实施模糊查找以便添加。若添加后感到不妥，可立即移除。此外，若数量不足或过多，可点击“+”“-”以个性化调整。完成菜品配置后，上传相关文件至服务器展示，并保存设置。据此，新增套餐规范可参照下表 4.2 所示。

表 4.2 新增套餐规约表

用例名称:	新增套餐
用例 ID:	UC1002
角色:	后台系统管理员
前置条件:	管理员已登录，添加的菜品须处于起售状态
基本事件流:	1. 管理员在本事件流中点击“新增套餐”，跳转添加套餐页面。 2. 选择已有套餐分类，编辑套餐名称价格。 3. 点击“添加菜品”，可以在搜索栏搜索，或者直接点击菜品分类去搜索。
扩展事件流:	1. 如果不想要添加某一菜品，也可以直接选择“删除菜品”。 2. 可以批量删除不想要添加的菜品。
后置条件:	新增套餐，并将最新套餐信息存入数据库

5. 订单管理：在顾客挑选并加入购物车菜品或套餐后，后台管理系统将展示“订单详情”，管理人员能够手动更新订单状态，从“配送中”变为“已送达”。同时，管理员可查阅订单的具体信息，相关用例图参见图 4.6。

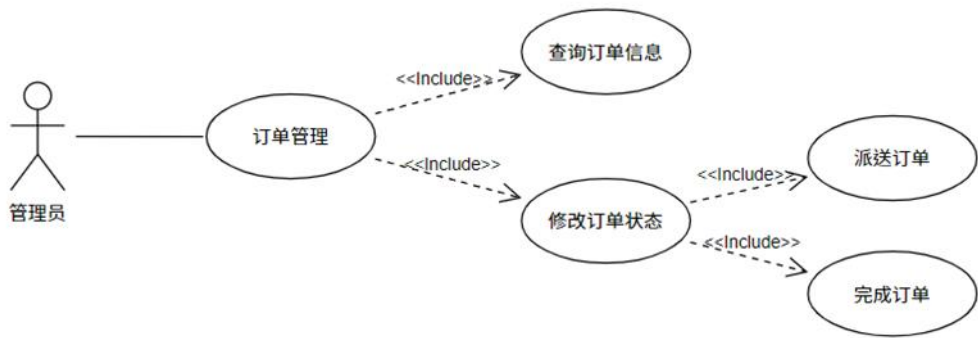


图 4.6 订单管理用例图

用户端模块：

6. 菜品信息：消费者可在客户端浏览商家发布的在售菜品及套餐详情，并可按口味、规格进行筛选。点击单一菜品可获取详细信息，点击套餐则展现其包含的全部菜品。相关用例图示于 4.7 图。

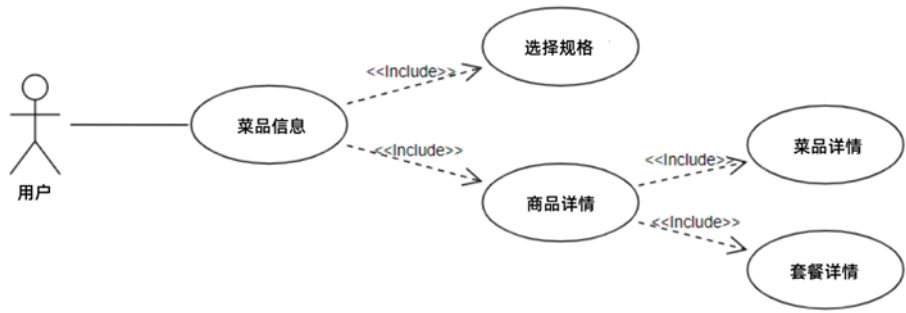


图 4.7 菜品信息用例图

7. 地址簿管理：在移动应用开发过程中，用户在购买菜品并下单时需填写收货地址。用户可新增收货地址，并为地址添加标签以标识其特性。用户还可设定某地址为默认地址，此后该账户再次下单将自动使用该地址。此外，用户可在个人主页的地址管理模块中查看、创建、更新或删除所有地址。相关用例图如图 4.8 所示。

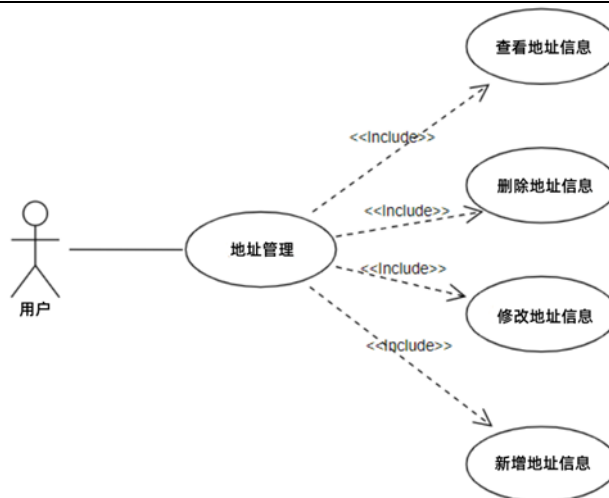


图 4.8 地址簿管理用例图

8. 购物车管理：当消费者意图挑选菜品或套餐时，可点击“+”按钮添加至购物车，随心所欲挑选所需物品。若需查看购物车内容，可点击相应栏位，对不满意的选项点击“—”移除，或直接选择清空购物车功能。相关操作流程如图 4.9 所示。

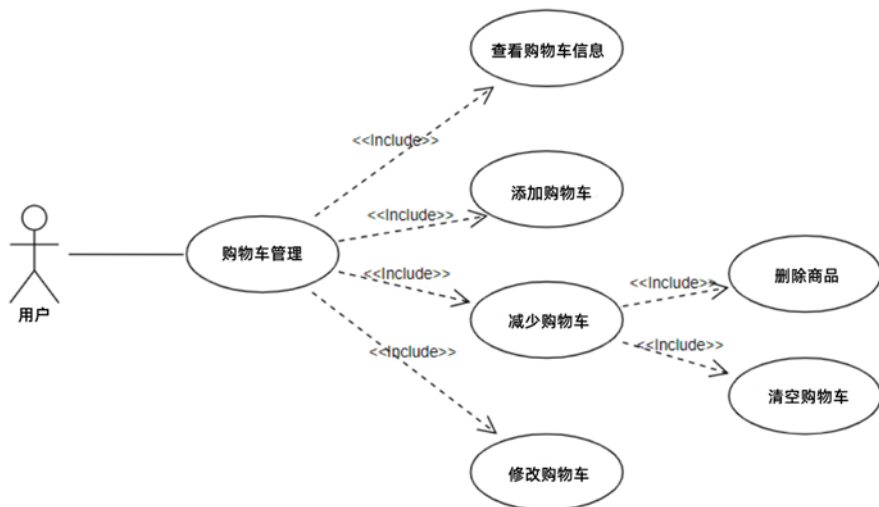


图 4.9 购物车管理用例图

9. 下单： 顾客在菜品界面选定商品并确认订单，系统将计算商品总额及运费后自动转至订单界面，展示预设地址和总价等信息。在此界面，顾客可更改地址和添加备注，完成下单后系统将引导至支付界面。支付完成后，页面将跳转至订单详情页。相关用例图如图 4.10 所示。

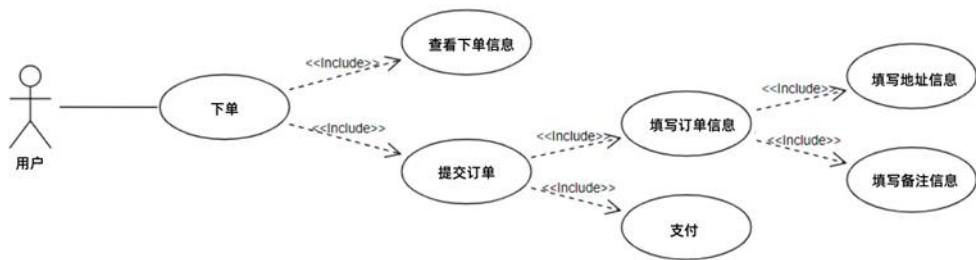


图 4.10 下单用例图

4.2 系统功能模块设计

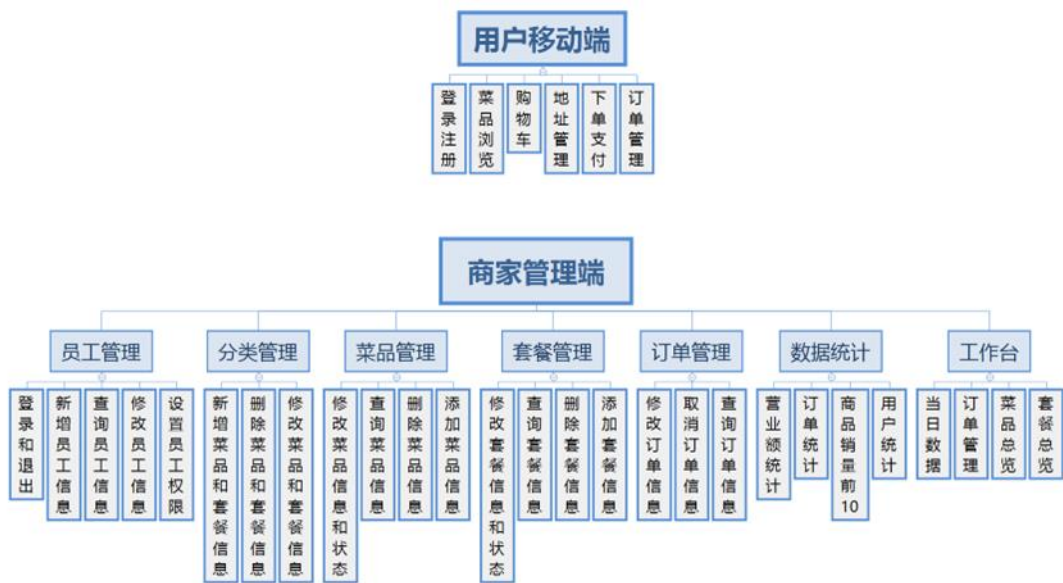


图 4.11 模块设计图

4.3 数据库设计

4.3.1 数据库概念结构设计

## 中北大学 2025 届毕业设计说明书

通过对餐饮管理与服务系统设计与实现，可得 UML 类图如图 4.12。

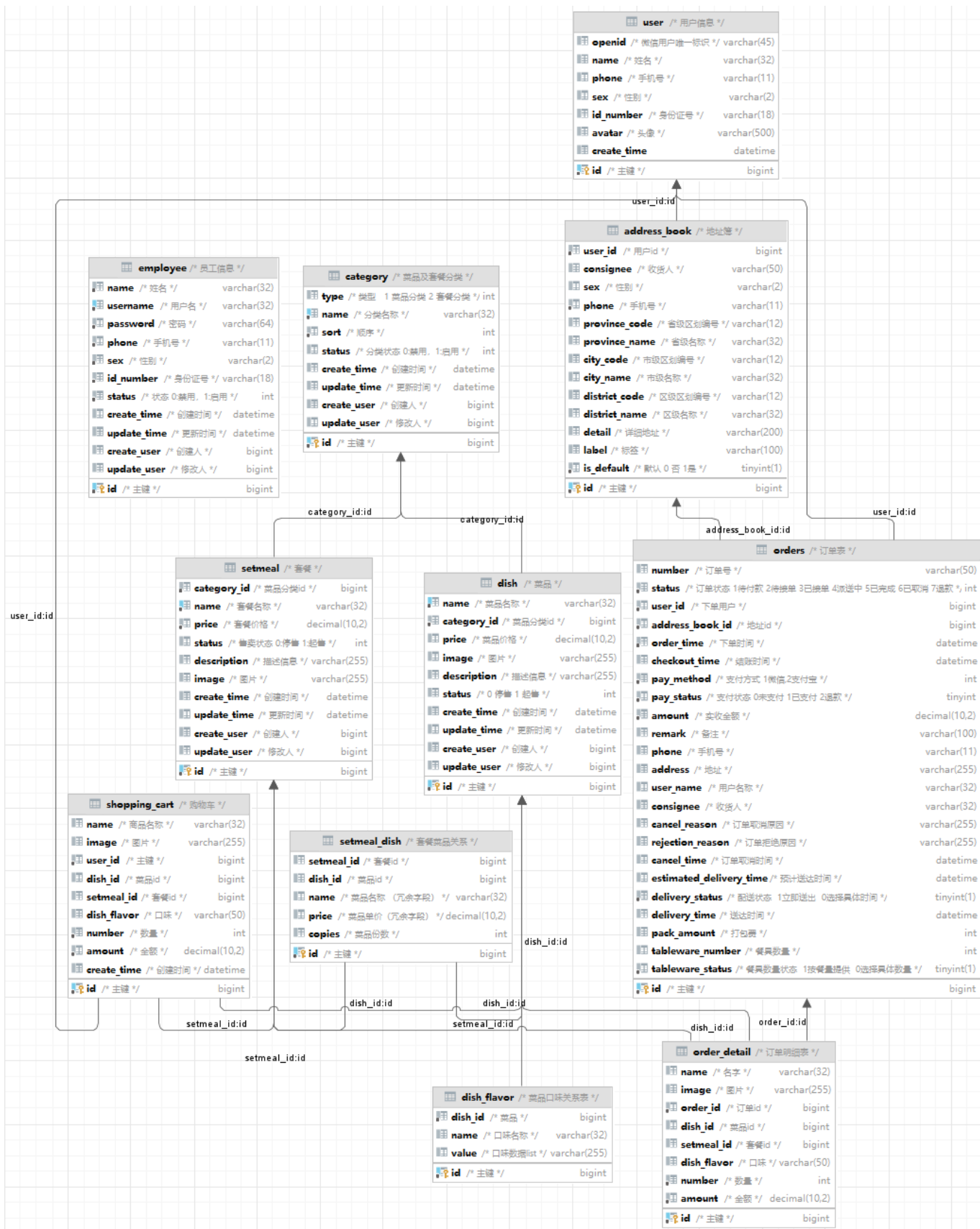


图 4.12 UML 类图

经过对餐饮管理与服务系统管理员端的设计与实施分析，可知构建系统所需数据要素。进一步了解各要素属性，如图 4.13 所示。

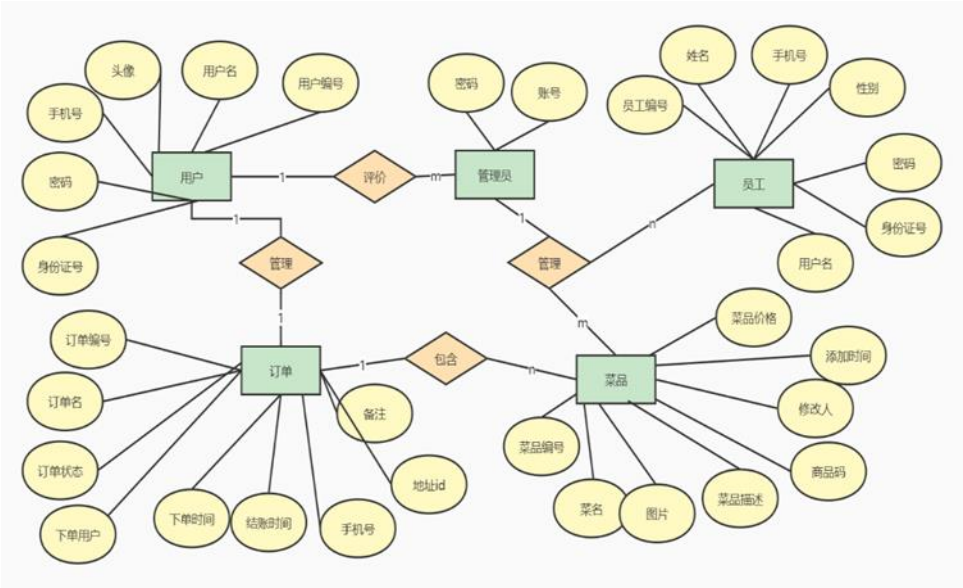


图 4.13 用户下单实体图

我抽象出系统的实体，并设计部分 E-R 图，如下。

用户信息 如图 4.14 所示。

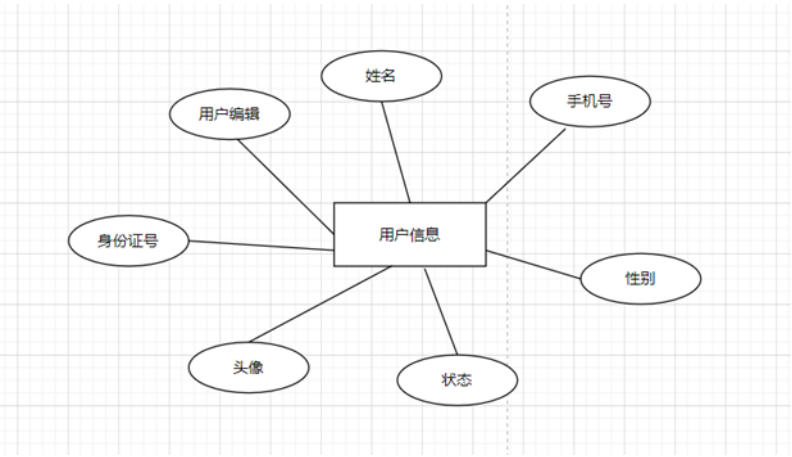


图 4.14 用户信息实体图

菜品信息 如图 4.15 所示。



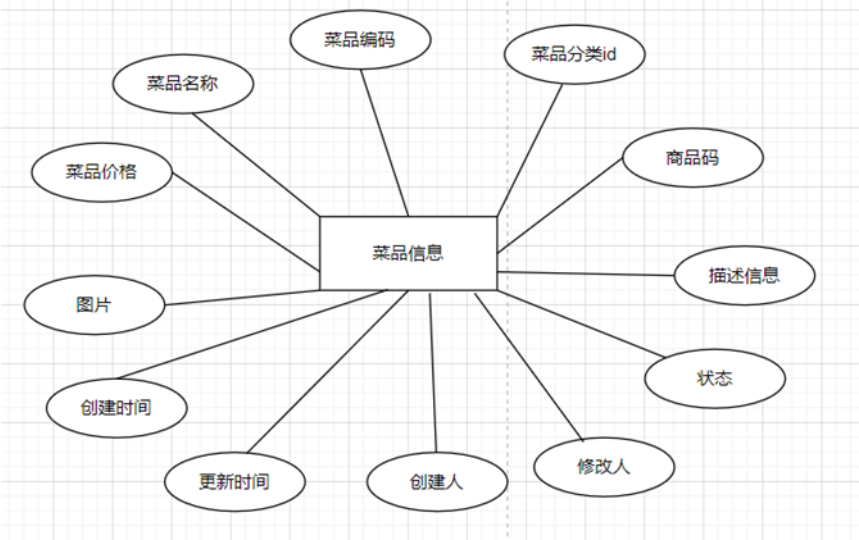


图 4.15 菜品信息实体图

订单信息 如图 4.16 所示。

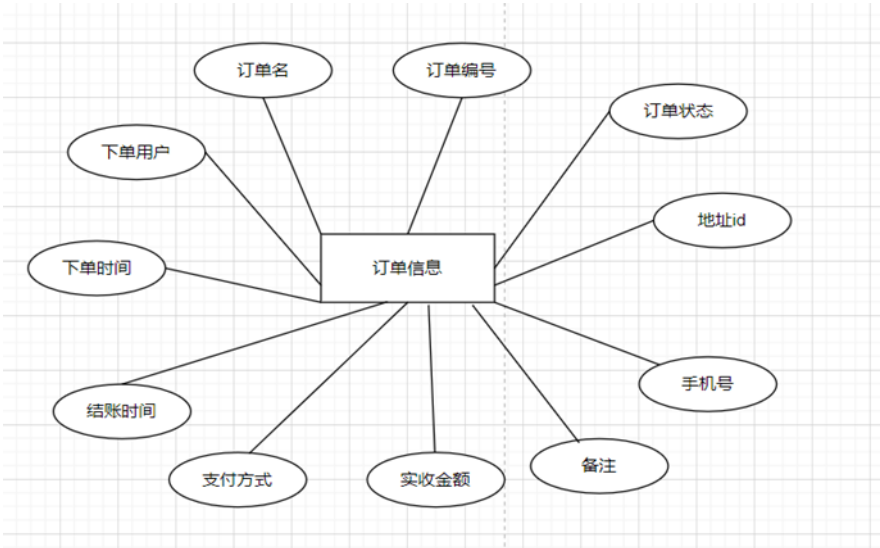


图 4.16 订单信息实体图

套餐信息 如图 4.17 所示。



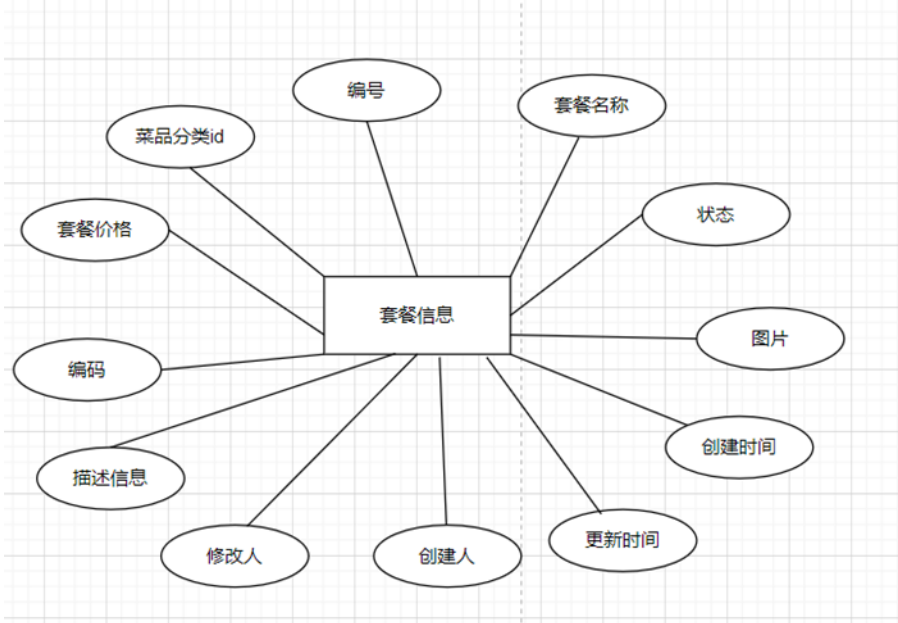


图 4.17 套餐信息实体图

菜品与套餐分类信息 如图 4.18 所示。

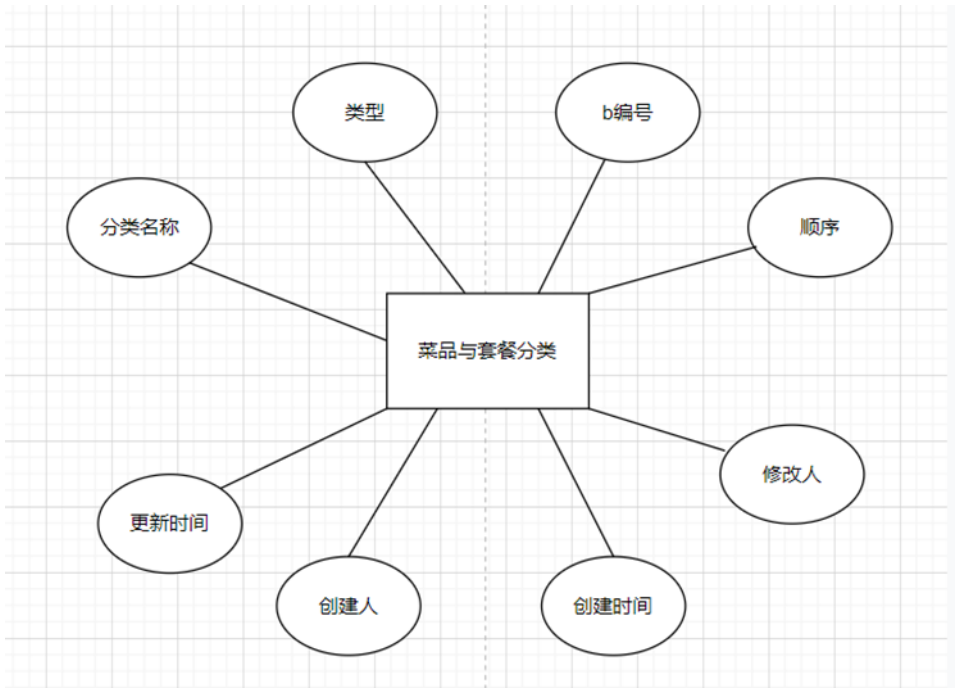


图 4.18 菜品与套餐分类信息实体图

员工信息 如图 4.19 所示。



图 4.19 员工信息实体图

地址信息 如图 4.20。

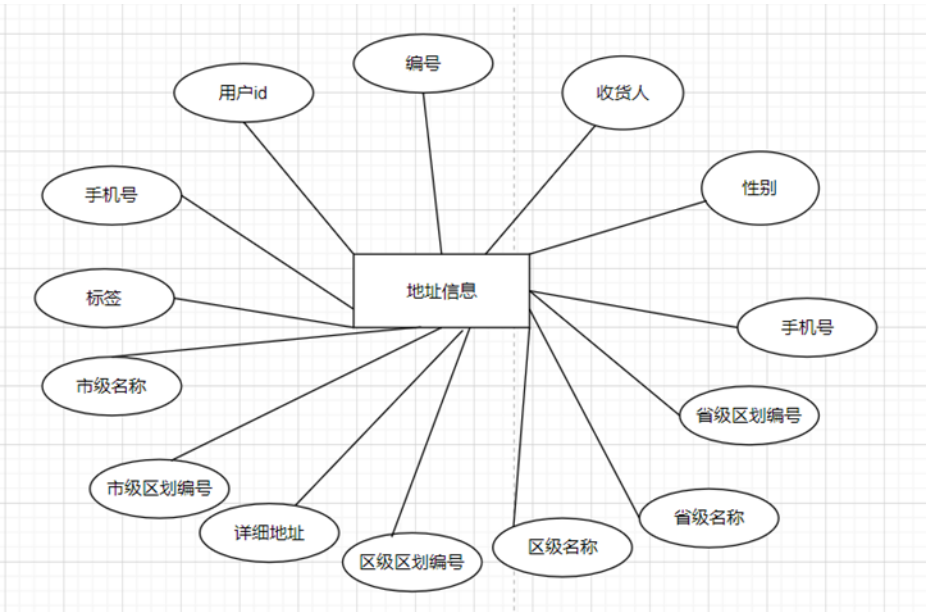


图 4.20 地址信息实体图

菜品口味信息 如图 4.21 所示。

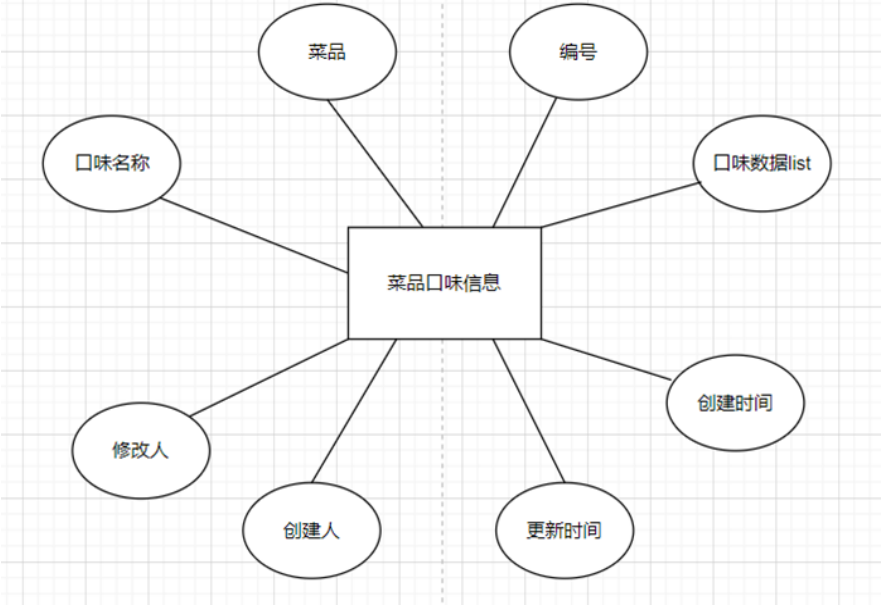


图 4.21 菜品口味信息实体图

订单明细 如图 4.22 所示。

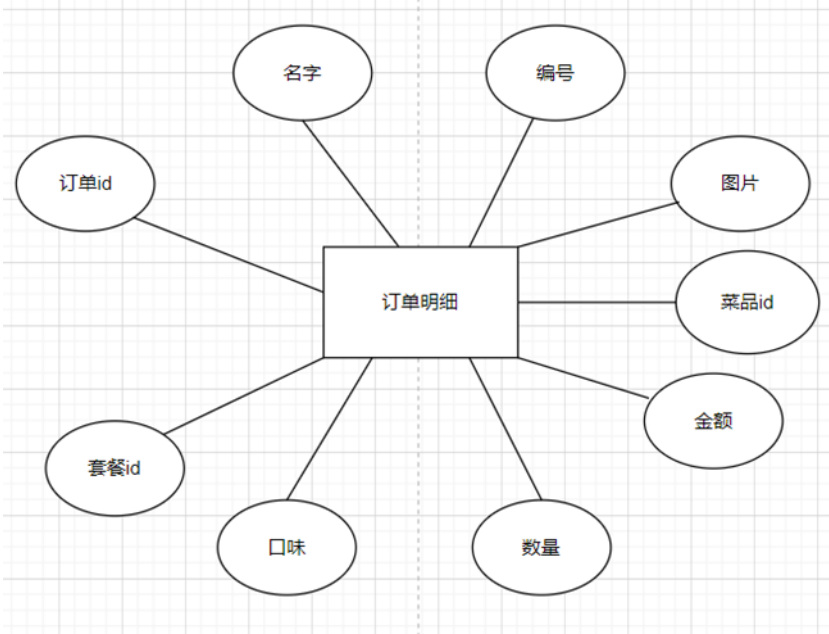


图 4.22 订单明细信息实体图

套餐菜品关系信息 如图 4.23 所示。

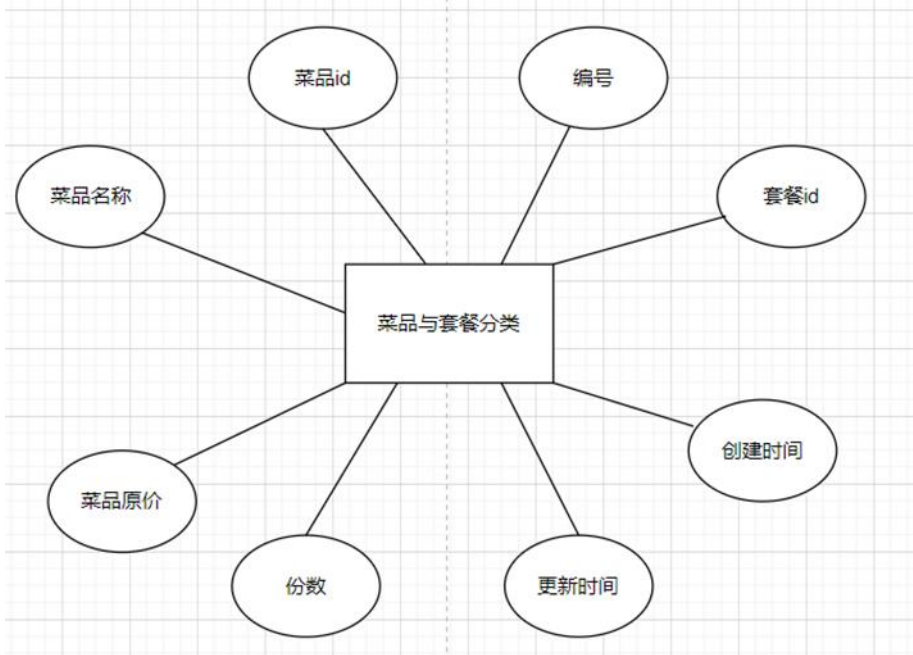


图 4.23 菜品与套餐分类信息实体图

购物车信息 如图 4.24 所示。

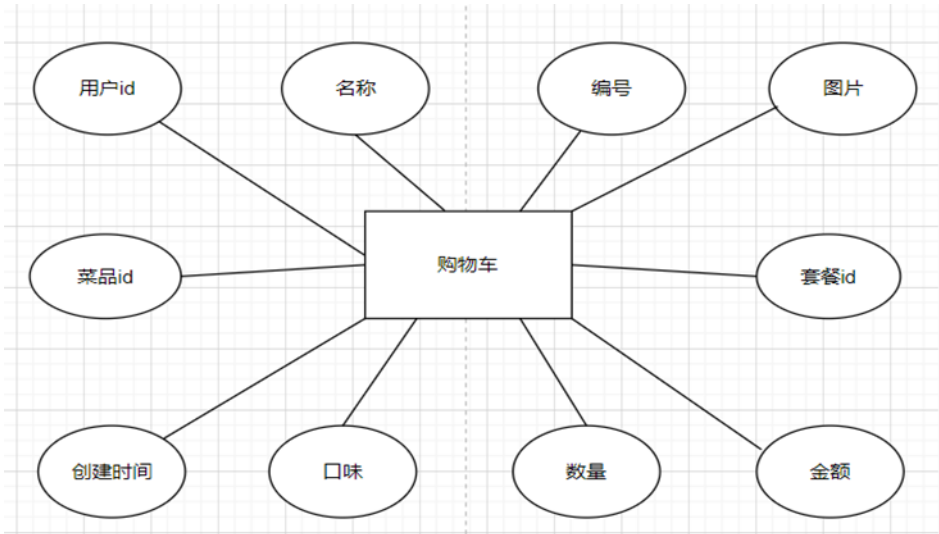


图 4.24 购物车信息实体图

4.3.2 数据库表结构设计

依据数据库概念设计中的实体属性定义及规范化理论，构建管理员、分类、菜品管理、口味关联、员工、订单及其明细、套餐及其菜品关联、购物车与用户信息等多个数据表。表 4.1 至表 4.11 展示了这些数据表的逻辑结构。

表 4.1 用户信息表

字段名	列类型与长度	描述	备注
id	bigint(20)	编号	主键
name	varchar(50)	姓名	非空
phone	varchar(100)	手机号	非空
sex	varchar(2)	性别	非空
id_number	varchar(18)	身份证号	非空
avatar	varchar(500)	头像	非空
status	int(11)	状态 0: 禁用 1: 正常	非空

表 4.2 管理员信息表

字段名	列类型与长度	描述	备注
num	varchar (255)	工号	主键
name	varchar (32)	姓名	非空
username	varchar (32)	姓名	非空
password	varchar (64)	密码	非空

表 4.3 分类信息表

字段名	列类型与长度	描述	备注
id	bigint(20)	编号	主键
type	int(11)	类型	非空
name	varchar(64)	分类名称	非空
sotr	int(11)	顺序	非空
Create_time	datetime	创建时间	非空
Update_time	datetime	更新时间	非空
Create_user	bigint(20)	创建人	非空
Update_user	bigint(20)	修改人	非空

表 4.4 菜品管理信息表

字段名	列类型与长度	描述	备注
id	bigint(20)	编号	主键
name	varchar(64)	菜品名称	非空
category_id	bigint(20)	菜品分类 id	非空
price	decimal(10, 2)	菜品价格	非空
code	varchar(64)	商品码	非空
image	varchar(200)	图片	非空
description	varchar(400)	描述信息	非空
status	int(11)	0 停售 1 起售	非空
sort	int(11)	顺序	非空
Create_time	datetime	创建时间	非空
Update_time	datetime	更新时间	非空
Create_user	bigint(20)	创建人	非空
Update_user	bigint(20)	修改人	非空
is_deleted	int(11)	是否删除	非空

表 4.5 菜品口味关系表

字段名	列类型与长度	描述	备注
id	bigint(20)	编号	主键
dish_id	bigint(20)	菜品	非空
name	varchar(64)	口味名称	非空
value	varchar(500)	口味数据 list	非空
Create_time	datetime	创建时间	非空
Update_time	datetime	更新时间	非空
Create_user	bigint(20)	创建人	非空
Update_user	bigint(20)	修改人	非空
is_deleted	int(11)	是否删除	非空

表 4.6 员工信息表

字段名	列类型与长度	描述	备注
id	bigint(20)	编号	主键
name	varchar(32)	姓名	非空
username	varchar(32)	姓名	非空
password	varchar(64)	密码	非空
phone	varchar(11)	手机号	非空
sex	varchar(2)	性别	非空
id_number	varchar(18)	身份证号	非空
status	int(11)	状态 0:禁用 1:正常	非空
Create_time	datetime	创建时间	非空
Update_time	datetime	更新时间	非空
Create_user	bigint(20)	创建人	非空
Update_user	bigint(20)	修改人	非空

表 4.7 订单信息表

字段名	列类型与长度	描述	备注
id	bigint(20)	编号	主键
number	varchar(50)	订单号	非空
status	int(11)	订单状态 1 待付款, 2 待派送	非空
user_id	bigint(20)	下单用户	非空
address_book_id	bigint(20)	地址 id	非空
order_time	datetime	下单时间	非空
checkout_time	datetime	结账时间	非空
pay_method	int(11)	支付方式 1 微信, 2 支付宝	非空
amount	varchar(255)	实收金额	非空
remark	varchar(255)	备注	非空
phone	varchar(11)	电话	非空
address	varchar(255)	地址	非空
user_name	varchar(32)	用户名	非空
consignee	varchar(50)	收货人	非空

表 4.8 订单明细表

字段名	列类型与长度	描述	备注
id	bigint(20)	编号	主键
name	varchar(50)	名字	非空
image	varchar(100)	图片	非空
order_id	bigint(20)	订单 id	非空
dish_id	bigint(20)	菜品 id	非空
setmeal_id	bigint(20)	套餐 id	非空
dish_flavor	varchar(50)	口味	非空
number	int(11)	数量	非空
amount	decimal(10, 2)	金额	非空

表 4.9 套餐信息表

字段名	列类型与长度	描述	备注
id	bigint(20)	编号	主键
category_id	bigint(20)	菜品分类 id	非空
name	varchar(64)	套餐名称	非空
price	decimal(10, 2)	套餐价格	非空
status	int(11)	状态 0 停售 1 起售	非空
code	varchar(32)	编码	非空
description	varchar(512)	描述信息	非空
image	varchar(255)	图片	非空
Create_time	datetime	创建时间	非空
Update_time	datetime	更新时间	非空
Create_user	bigint(20)	创建人	非空
Update_user	bigint(20)	修改人	非空
is_deleted	int(11)	是否删除	非空



表 4.10 套餐与菜品的关系表

字段名	列类型与长度	描述	备注
id	bigint(20)	编号	主键
setmeal_id	varchar(32)	'套餐 id	非空
dish_id	varchar(32)	菜品 id	非空
name	varchar(32)	菜品名称	非空
price	decimal(10, 2)	菜品原价	非空
copies	int(11)	份数	非空
sort	int(11)	排序	非空
Create_time	datetime	创建时间	非空
Update_time	datetime	更新时间	非空
Create_user	bigint(20)	创建人	非空
Update_user	bigint(20)	修改人	非空
is_deleted	int(11)	是否删除	非空

表 4.11 购物车信息表

字段名	列类型与长度	描述	备注
id	bigint(20)	编号	主键
name	varchar(50)	名称	非空
image	varchar(100)	图片	非空
user_id	bigint(20)	主键	非空
dish_id	bigint(20)	菜品 id	非空
setmeal_id	bigint(20)	套餐 id	非空
dish_flavor	varchar(50)	口味	非空
number	int(11)	数量	非空
amount	decimal(10, 2)	金额	非空
create_time	datetime	创建时间	非空

4.4 安全性与可靠性设计

1. 数据加密: 为保障数据传输安全, 避免敏感信息如用户密码、支付信息、商家数据在传输过程中遭受窃取或篡改, 系统需对这类信息实施严格加密。在多种加密算法中, MD5 因其具有唯一性且加密后长度固定、不可逆的特点而被选用。在新增管理员环节,

系统会将密码以 MD5 加密后的形式存入数据库。用户登录时,输入的密码会先经过 MD5 加密处理,生成 32 位的加密密文,再与数据库中存储的密文进行匹配。相关流程详见图 4.25。



图 4.25 流程图

2. 负载均衡: 负载均衡服务器通过一定的调度算法将客户端的流量分发到不同的应用服务器上面, 以实现性能的水平扩展及避免单点故障出现, 提升系统的可靠性。这一点我们可以使用 Nginx 实现, 修改 Nginx 配置文件使得前端请求分配给多个服务器处理, 同时修改负载均衡方式为轮询。

5 系统详细设计

5.1 系统功能模块设计

5.1.1 菜品管理模块

菜品管理功能涵盖了对菜品进行模糊搜索、新增、调整、移除、暂停销售及恢复销售等操作。管理员能够执行批量处理，如一次性删除或调整多条菜品状态。菜品模糊搜索及分页显示的流程图如图 5.1 所示。

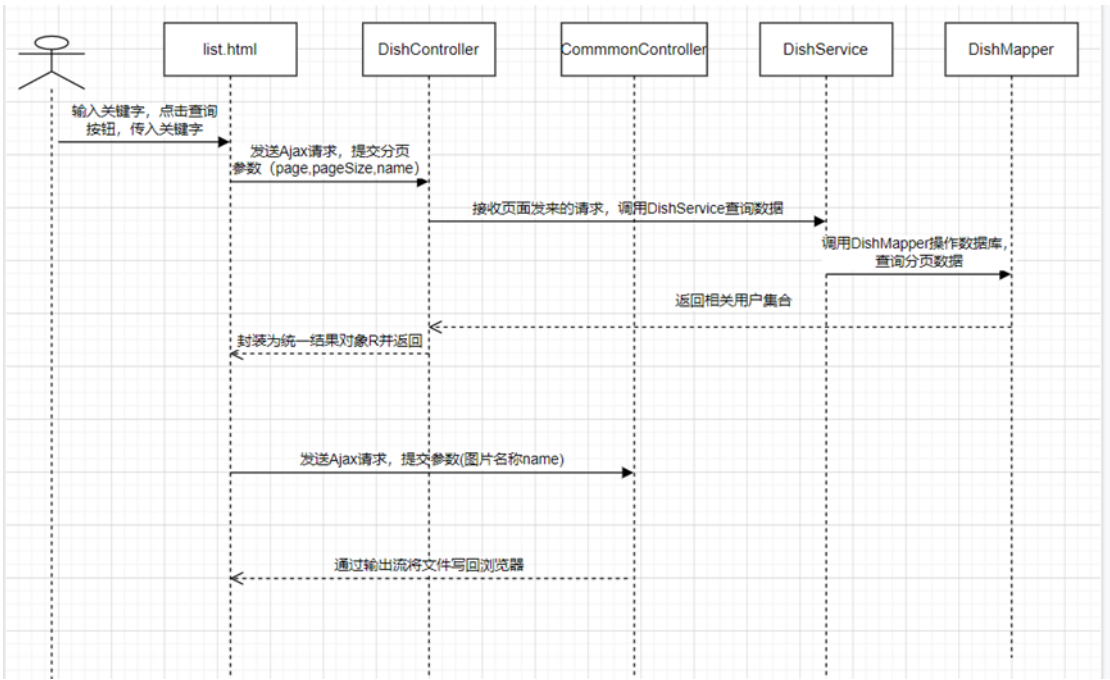


图 5.1 菜品模糊查询顺序图

管理员可以点击新建菜品按钮来新增菜品，顺序图如图 5.2 所示。

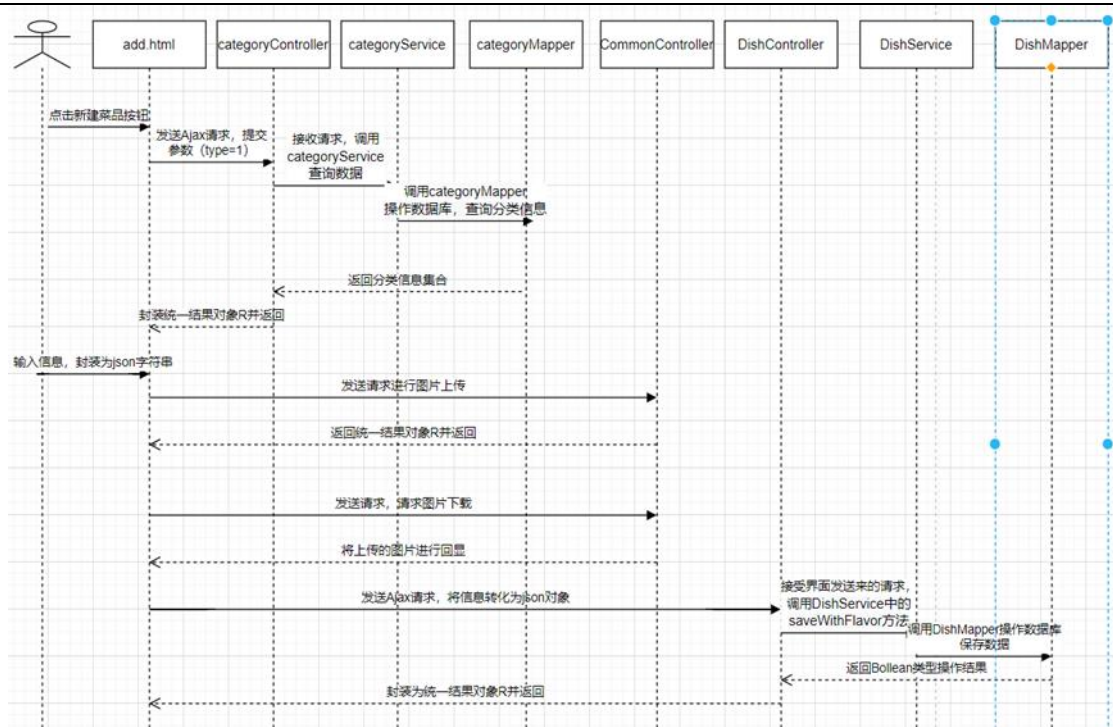


图 5.2 新增菜品顺序图

管理员可以点击菜品管理主页面的右边操作栏编辑按钮修改菜品信息，修改菜品执行顺序图如图 5.3 所示。

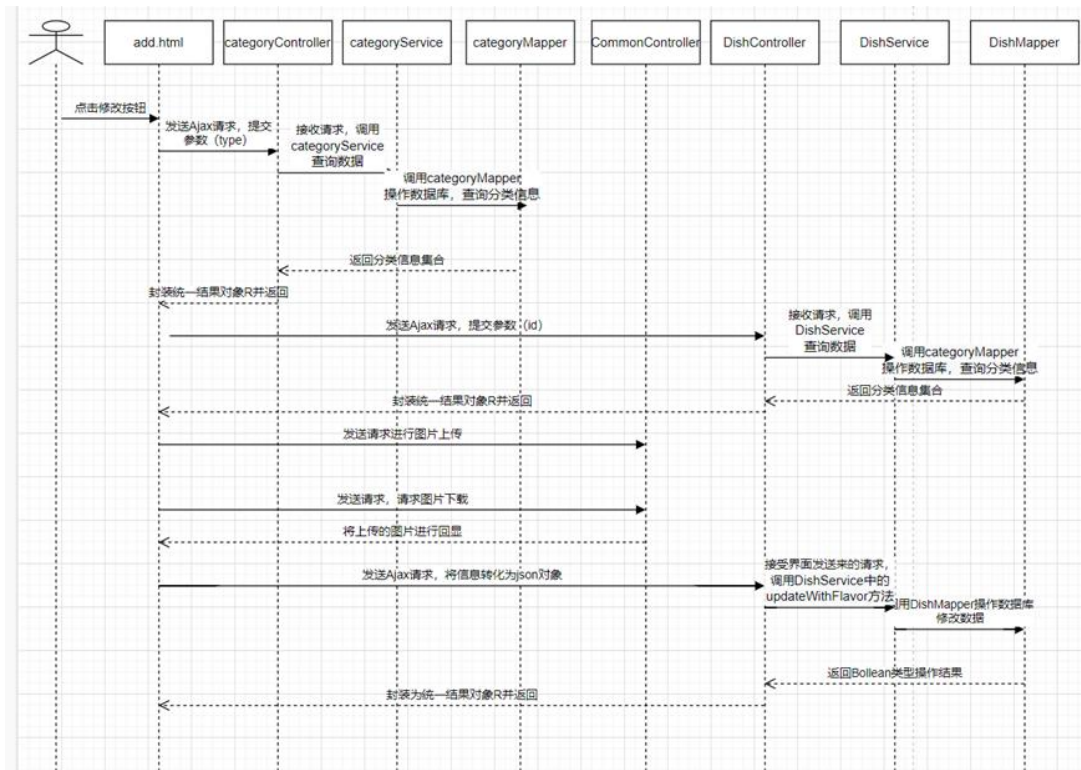


图 5.3 修改菜品顺序图

管理员可以点击菜品信息右侧操作中的删除按钮来删除单个菜品信息，也可以选中多个菜品，然后点击批量删除按钮来删除多个菜品信息。删除菜品信息顺序图如图 5.4 所示。

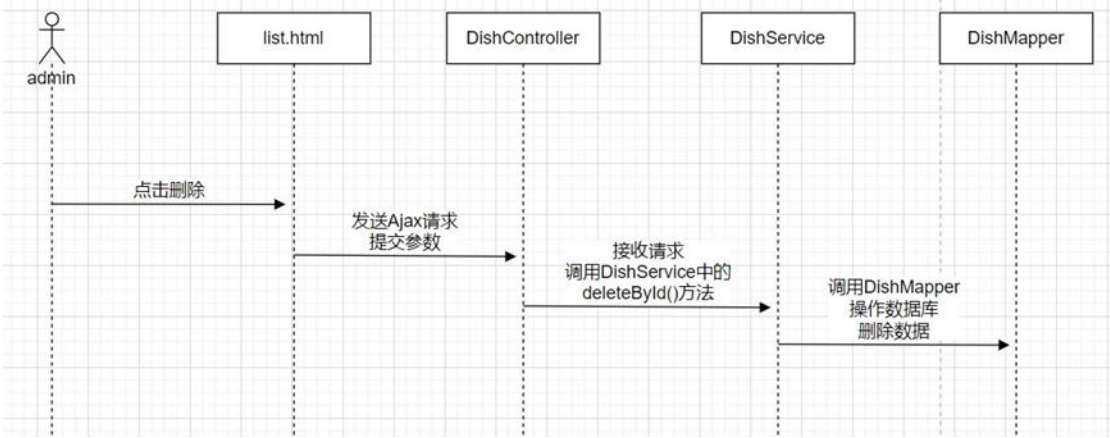


图 5.4 删除菜品顺序图

起售停售菜品：管理员可以点击菜品信息右侧操作中的停售（起售）按钮来修改单个菜品的状态，也可以选中多个菜品，然后点击批量停售（起售）按钮来修改多个菜品的状态。处于停售状态的菜品不会在展示在移动端前台上。停售（起售）菜品顺序图如图 5.5 所示。

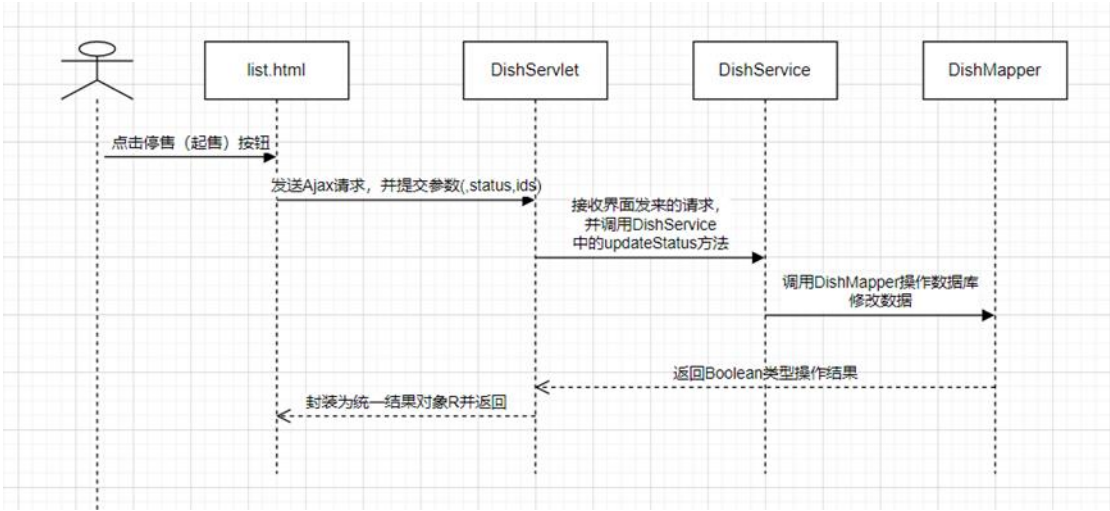


图 5.5 停售(起售)菜品顺序图

5.1.2 套餐管理模块

套餐管理模块包括套餐的模糊查询、添加、修改、删除、停售、起售等功能，管理员还可以进行批量操作，删除多条数据或者停售、起售多条数据。

管理人员可通过点击“新增套餐”按键来添加新的菜品组合。一旦触发该按键，

add.html 界面会首先通过 Ajax 方式向 CategoryController 的 list 功能发送请求，以收集套餐分类信息，并在下拉菜单中展示供管理人员挑选。此后，页面将再次通过 Ajax 方式发出请求，此次依然调用 CategoryController 的 list 功能，但旨在获取菜品分类的相关信息。这些信息随后在添加菜品窗口中展示，供管理人员在设立新套餐时作为参考。紧接着，页面会发起第三次 Ajax 请求，此次请求调用 DishController 的 list 功能，并将先前选择的菜品分类作为参数传递，用以检索该分类下的具体菜品数据。这些菜品数据随后在添加菜品窗口中展示，以便管理人员在构建套餐时选取特定菜品。管理人员完成套餐信息的输入并点击“上传图片”按钮后，页面将发送请求至 CommonController 的 upload 功能，以处理图片上传事宜，将所选图片储存至服务器。图片上传成功后，管理人员可以继续套餐创建过程。图片储存完毕，页面会再次发出请求，调用 CommonController 的 download 功能，以实现图片的下载和显示。最终，管理人员点击“保存”按钮，页面将通过 Ajax 请求调用 setmealController 的 save 功能，以保存所有数据。

管理员可以点击套餐信息右侧操作中的删除按钮来删除单个套餐信息，也可以选中多个套餐，然后点击批量删除按钮来删除多个套餐信息。如果套餐处于售卖状态（status 为 1），则不能删除。

管理员可以点击套餐信息右侧操作中的停售（起售）按钮来修改单个套餐的状态，也可以选中多个套餐，然后点击批量停售（起售）按钮来修改多个套餐的状态。处于停售状态（status 为 0）的菜品不会在展示在移动端前台上，处于售卖状态（status 为 1）的菜品不会被删除，支持批量启售与停售。

### 5.1.3 订单管理模块

在订单明细模块有模糊查询，查看订单，派送订单，完成订单等功能。顺序图如图 5.6 所示。

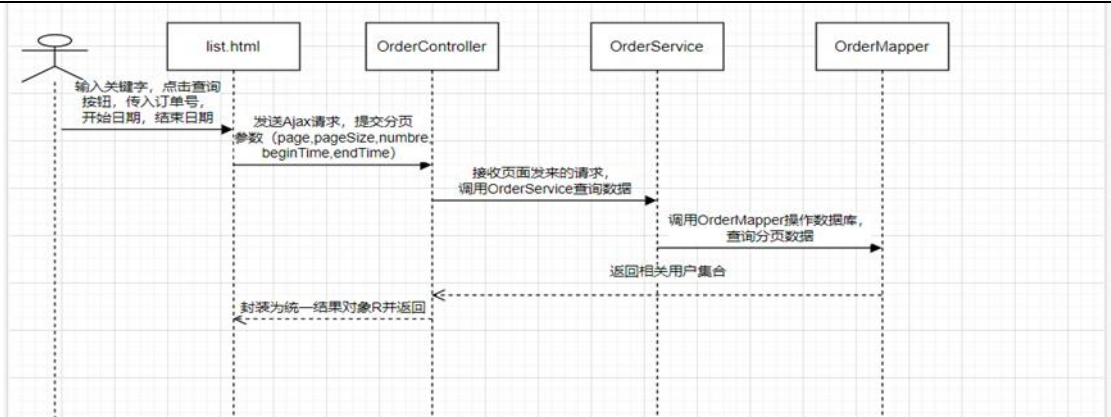


图 5.6 订单模糊查询图

管理员点击订单右边操作栏中的派送（完成）按钮，可以修改订单的状态，执行顺序图如图 5.7 所示。

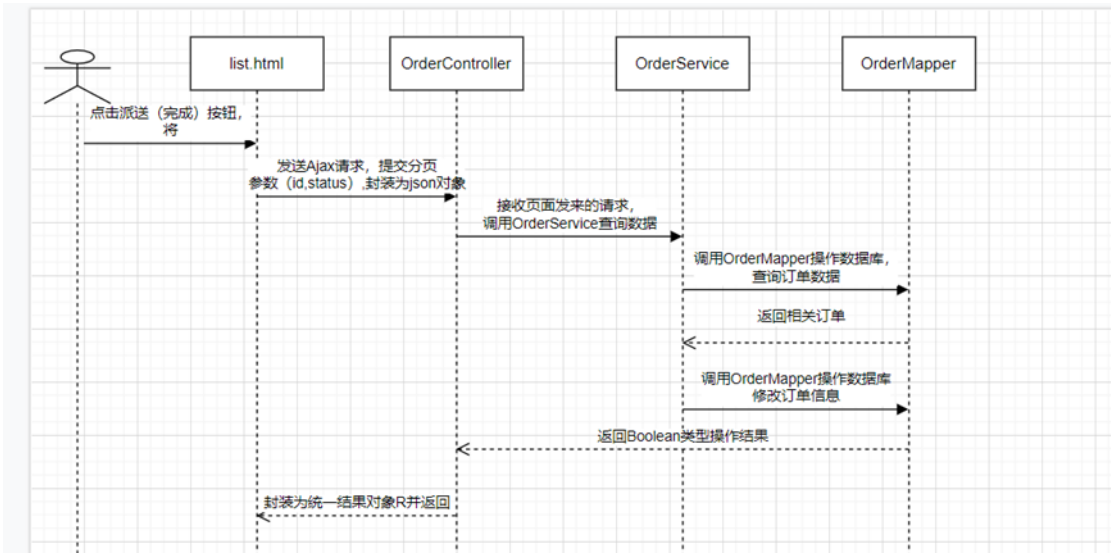


图 5.7 派送、完成顺序图

#### 5.1.4 鸿蒙移动端模块开发



中北大学 2025 届毕业设计说明书

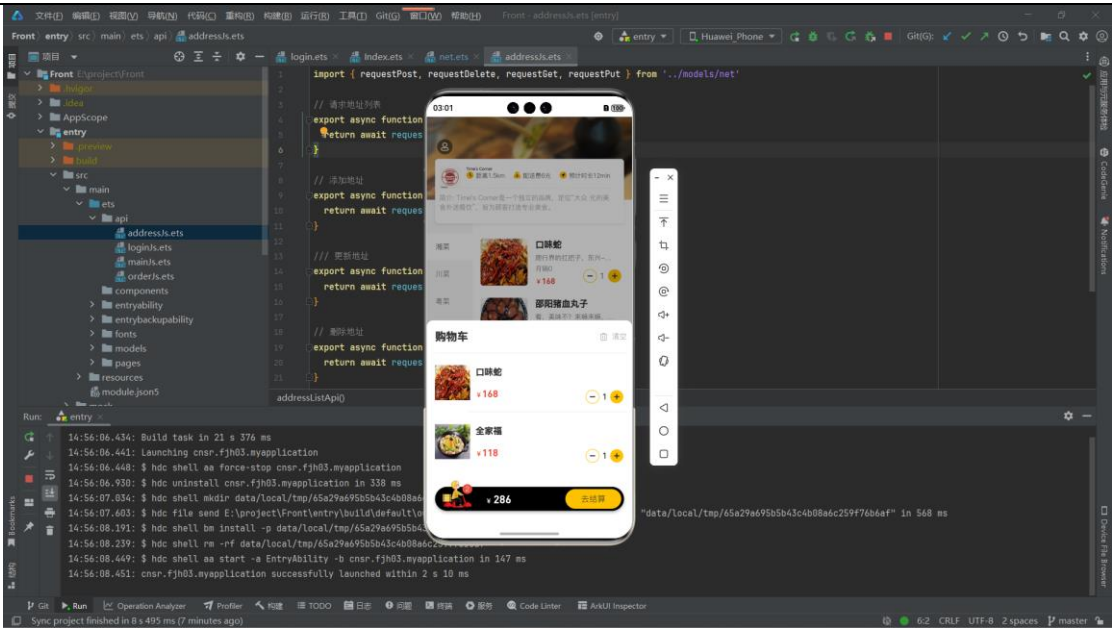


图 5.8 减少或清空购物车数据

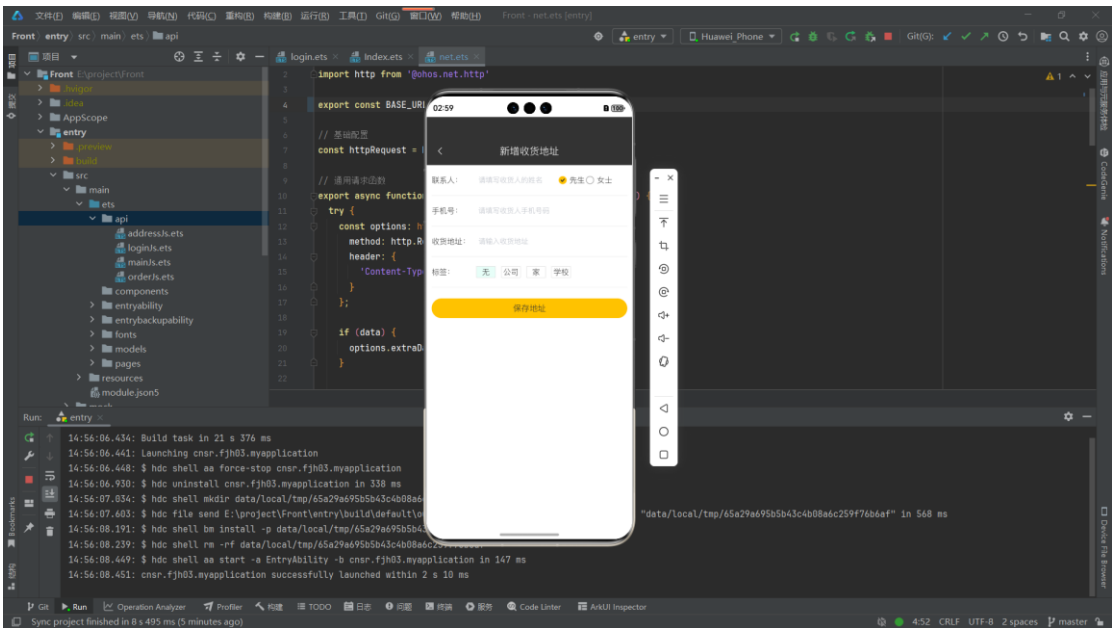


图 5.9 添加地址





图 5.10 设置默认地址

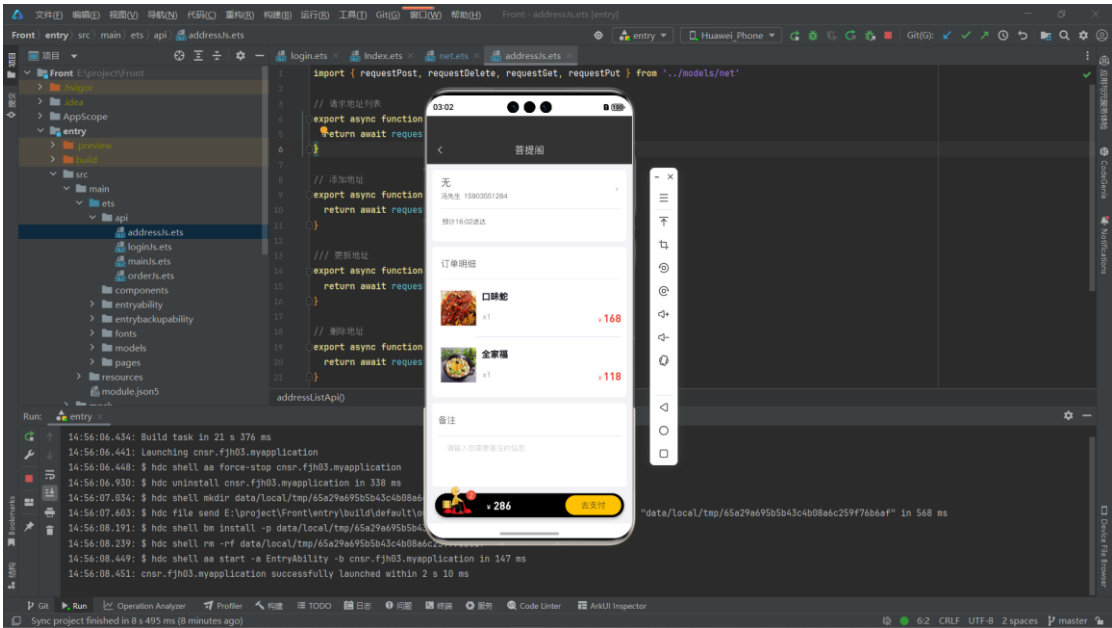


图 5.11 下单

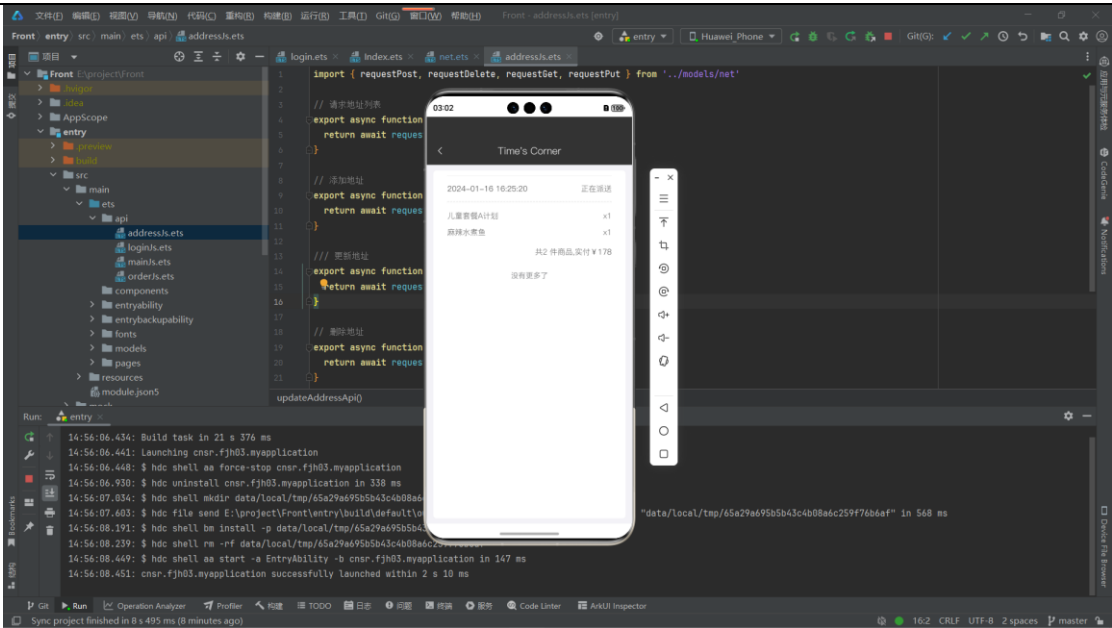


图 5.12 查看历史订单

用户点击“历史订单”可以查看用户的历史订单，前端发送 ajax 请求，后端接收到请求，构造一个查询条件构造器，将用户的 id 作为条件进行查询，并将下单时间为排序条件，将最新的订单排在前面。

6 系统测试

6.1 测试方法

在系统的整个开发过程中，测试是至关重要的一环。通过测试，我们可以及时发现并修正系统中的不足，从而不断完善和优化系统。测试不仅能够帮助我们找到代码编写错误或遗漏的问题，还能确保系统在使用过程中的稳定性和可靠性。从项目启动到完成的每一个阶段，测试都伴随着开发过程，形成一个“测试-调试-测试”的循环，确保最终交付的产品质量可靠、性能优良。因此，测试是系统开发不可或缺的一部分，对于保证系统的质量和用户体验至关重要。

6.2 功能测试

6.2.1 登录功能测试

测试流程：

以管理员（admin 用户）、身份从登录界面登录，测试系统的登录功能，以及管理员用户是否成功建立连接。

管理员登陆测试结果如表 6.1。

表 6.1 测试结果表一

测试用例 (用户名、密码)	预期结果	测试结果
( “ ” , “ ” )	用户名不能为空	提示：输入用户名
( “admin” , “ ” )	密码不能为空	提示：输入密码
( “admin”, “123456” )	密码错误	提示：输入正确密码
( “admin”, “123123” )	登录成功	登录成功，跳转到员工管理页面

6.2.2 员工管理测试

测试流程：

以管理员（admin 用户）从登录界面登录管理员，测试系统的员工管理功能，测试结果如表 6.2。

表 6.2 测试结果表二

测试用例 (名称、价格、库 存、描述，图标，分类、 状态)	预期结果	测试结果
添加用户 (zhangsan、张三、 16638980885 、 男 ， 412721200010213366)	添加成功，返回员 工列表显示员工的信息。	添加成功，返回员工列表显 示员工的信息
添加用户 ( “ ” 、 张 三 、 16638980885 、 男 ， 412721200010213366)	添加失败	提示框 “请输入账号”
禁用员工 (id:1591235383657 455618, status:0)	禁用成功，该用户 不能在登录系统	禁用成功，客户端中该用户 操作栏中 “禁用” 按钮变为 “启 用” 按钮
修改员工 ( “张三” 修改为 “李四” )	修改成功	员工姓名 “张三” 改为 “李 四”
修改员工 ( “4127212000102 13366” 修改为 “412721200” )	修改失败	提示框 “身份证号不正确”

6.2.3 套餐管理测试

测试流程：以超级管理员（admin 用户）从登录界面登录管理员，测试系统的套餐管理功能，测试结果如表 6.3。

表 6.3 测试结果表三

测试用例 (名称、价格、库 存、描述，图标，分类、 状态)	预期结果	测试结果
添加套餐 (“精品四人餐”、 “四人餐”、“254”，菜品 两道，图片一张，“好吃划 算”)	添加成功，返回套 餐列表显示套餐的信息。	添加成功，返回套餐列表显 示套餐的信息。
添加用户 (“”、“四人餐”、 “254”，菜品两道，图片一 张，“好吃划算”)	添加失败	提示框“请输入套餐名称”
删除套餐 (ids:1594972446119 768066	删除失败	删除失败，弹出提示框显示 “套餐正在售卖中，不能删除”
修改员工 (“精品四人餐” 修改为“超值四人餐”)	修改成功	套餐名称“精品四人餐”改 为“超值四人餐”

6.2.4 安全测试

测试流程：测试管理员，员工的登录安全性，再测试不是本系统用户是否能登录成功，测试系统的安全性。管理员，员工登录测试结果如表 6.4。

表 6.4 测试结果表四

测试用例 (学号/工号、用户名、密码)	测试结果
管理员登录 (admin、123456)	登录成功，跳转到员工管理主页
管理员登录 (admin、123456)	登录失败，提示“登录失败”
员工登录 (lisi, 123456)	登录失败，提示“员工账号已被禁用”
员工登录 (xiaoming, 123456)	登录成功，跳转到员工主页

## 7 结论与展望

中小型餐厅受限于资金和技术因素，往往不会选择搭建独立的餐饮服务平台，而是沿用传统的订餐模式。鉴于此，本论文旨在以城区内的中小型餐厅为服务对象，深入剖析市场特点与消费者购物行为，进而提出构建开放型餐饮服务平台的构想。此举不仅有助于解决餐饮企业在运营过程中遇到的问题，更能为市民提供更为便捷的服务，具有显著的实际意义。

本系统所设计并实现的外卖订餐平台，其核心功能在于实现网络订餐的便捷性。鸿蒙用户通过前台系统能够轻松浏览各类菜品，将心仪的菜品加入购物车，编辑并管理个人信息及收货地址，完成购物车的结算并下单支付。而管理员则能通过后台系统对菜品信息和套餐信息进行维护，确保前台用户能够选择到最新、最全的菜品进行下单。此外，后台系统还提供了订单查看与操作的功能，方便管理员对订单进行实时监控与管理。

本系统相对于其他餐饮管理系统，本系统还需完善以下三个方面：

系统流程简单，功能没有创新。后期设计在线秒杀活动、某节日享折扣活动、团购活动等。商家与用户交互不强，用户只能通过备注单向与商家联系，希望后期能够加入联系商家”以及评论功能。

系统的安全性，系统可能存在未知的安全隐患，系统一旦遭到黑客攻击，盗取用户隐私信息，对系统进行破坏以致无法使用。后期会执行全面的安全审计，确保数据安全加密，遵循正确的日志记录实践，定期漏洞扫描及更新，

高并发性的完善，系统开发所有用户，当同时访问量多的时候，系统容易瘫痪。后期会采用优化算法和代码、数据库优化和分布式技术解决高并发问题。

## 参考文献

- [1] 贺怡龙, 张永荟, 林霭堃等.基于 React 框架的校园点餐 APP 系统设计[J].产业与科技论坛, 2021, 20(13):44-46.
- [2] 黎蔼萍, 吴双庆. 基于 SpringBoot 和微信小程序的碳普惠商城小程序的设计与实现[J]. 数字化用户, 2021, 27(2):62-64.
- [3] 李章恒. 校园外卖系统设计与实现[D]. 山东:山东大学, 2022.
- [4] 马雪山, 张辉军, 陈辉等.前后端分离的 Web 平台技术研究与实现[J].电子技术与软件工程, 2022(08):70-73.
- [5] 田松涛, 段元梅.基于 SpringBoot 的线上商城平台设计[J].无线互联科技, 2022, 19(01):56-57.
- [6] 黄朝富, 姬小波, 吕明等.基于微信平台的校园点餐系统的设计与实现[J].工业控制计算机, 2020, 33(01):112-113+127.
- [7] 宁功林.基于微信小程序的大学校园点餐管理系统设计与实现[J].华东科技, 2022, (02):77-79.
- [8] 阿里云混合云备份 HBR[J]. 网络安全和信息化, 2023(1):38.
- [9] 高瞩, 张雨璇.基于服务设计的超市自助购物系统设计[J].时尚设计与工程, 2022, (02):1-7.
- [10]任建新, 王一鸣, 李鑫等.基于 Java Web 的智慧商城购物系统设计[J].信息技术与信息化, 2022, (07):23-27.
- [11]张雪敏. 浅议 MVC 设计模式在 JavaWeb 中的作用[J]. 科技风, 2018(22): 58-59.
- [12]刘震林, 喻春梅. 基于 MVC 模式的 JAVA Web 开发与实践应用研究[J]. 网络安全技术与应用, 2021(01): 57-58.
- [13]陆丽.MVC 设计模式在 JavaWeb 开发中的应用研究[J]. 信息通信,2020(04): 104-106.
- [14]梁艳玲. 基于 HTML5 的 Web 前端开发技术研究[J]. 电子技术与软件工程,2020(24): 44-45.
- [15]王影.软件系统测试过程管理及可追踪性方法研究[C].2015 航空试验测试技术学术交流会论文集,2015.
- [16] MARIADB.Maria DB Server[DB/OL].(2023-01-24).
- [17] SURYOTRISON GKO H,JAYANTO P,TJAHYANTO A.Design and development of



- backend application for public complaint systems using microservice spring boot[J].Procedia Computer Science,2017,24:736-743.
- [18] WESTMATTELMANN D, GROTENHERMEN J G,SPRENGER M, et al. The show must go on—Virtualisation of sport events during the COVID-19 pandemic[J].European Journal of Information Systems,2021,30(2):119-136.
- [19] Rippika N ,Golla R M ,Godbole S .Making effective carbon-box testing technique using prioritised-selected test cases driven combinatorial approach[J].International Journal of System Assurance Engineering and Management,2025,16(3): 1-26.
- [20] Silva M ,Paiva R C A ,Mendes A .GAMFLEW: serious game to teach white-box testing[J].Software Quality Journal,2024,33(1):5-5.

## 致 谢

在完成这份实习实训论文的过程中，我深刻感受到了来自不同方向的支持与帮助，对此我怀有深深的感激之情。

首先，我要向我的实习公司表达最诚挚的感谢。公司不仅提供了一个充满活力和挑战的工作环境，更让我有机会将所学知识应用于实践中。特别要感谢公司里的导师们，像卢老师这样认真且耐心的导师们，你们不仅在工作中以耐心和专业的态度指导我，而且在学术探索上也给予了我巨大的帮助。你们的经验分享和专业知识的教授，让我在专业领域的认识更加深入，对工作的理解更加透彻。

其次，我要感谢我的母校，感谢学校提供的实习机会和全方位的支持。软件工程专业的老师们，你们严谨的学术态度和深入浅出的教学方式，为我打下了坚实的理论基础。正是这些知识，让我在实习岗位上能够迅速适应，有效解决实际问题。

我也要感谢所有在这段旅程中给予我帮助的人，包括给予我灵感和启发的同学、提供资源和帮助的人员，以及所有在背后默默支持我的朋友和同事。请接受我最诚挚的谢意。这份项目和论文的完成，是我们共同努力的成果，我将继续携带着这份感激，走向未来的学术和职业生涯。