**中北大学软件学院毕业论文**



|  |  |
| --- | --- |
| **论文题目:** | **基于SpringBoot+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** |
| **所属院系：** | **软件学院** |
| **专 业：** | **软件工程** |
| **班 级：** | **21130416** |
| **指导老师:** | **卢鑫培** |
| **日 期：** | **2025年 05 月20日** |

声 明

本人郑重声明，本论文是在导师的指导下独立完成，除加注和致谢外，文中不包含他人所发表或撰写的成果。本人拥有自主知识产权，没有抄袭、剽窃他人成果，对于参考的文献已经加注并表示感谢。若有不实之处，本人愿意承担相关法律责任。

作者签名：

签字日期： 2025年05月20日

摘 要

随着国内外卖行业的蓬勃发展，外卖平台间的竞争影响了许多餐饮企业。许多餐饮企业面临着第三方外卖平台抽佣高、营销策略受限等问题，导致利润降低甚至亏本。针对这一问题，本论文提出了开发一个开放的外卖系统的想法，旨在使餐饮企业能够直接面向客户，摆脱对第三方平台的依赖。

通过调研和分析，本论文将重点关注中小型餐厅，以城区为服务对象，结合市场特点和消费者行为，开发一个适合这一需求的开放型餐饮管理与服务系统。通过该系统，中小型餐饮企业能够减少对第三方外卖平台的依赖，降低成本，提高利润。同时，消费者也能够享受到更多优质、多样化的餐饮选择和个性化的服务。

本项目选择Java语言进行开发，系统角色包括管理员和用户，管理员可以管理员工信息、分类管理、菜品管理、营业数据统计以及运营数据导出等。用户可以查看菜品信息、管理购物车、管理地址、修改个人信息等。项目经过需求分析、功能模块设计和数据库设计后，详细描述了各个功能模块的实现方式。本毕业设计前端使用HTML 5、AJAX、Vue、Element-UI框架实现，管理端后端使用Spring、Spring MVC、Spring Boot、MyBatis-Plus实现，移动端以适配HarmonyOS为目标，数据库采用MySQL，缓存使用Redis实现，项目部署使用Linux系统和Tomcat。

**关键词**：外卖订餐系统；HTML5；AJAX；Vue；Spring Boot；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:** Takeway ordering system; HTML 5; AJAX; Vue; Spring Boot; HarmonyOS

目 录

[1 绪论 1](#_Toc199767542)

[1.1 研究背景和意义 2](#_Toc199767543)

[1.2 国内外研究现状 3](#_Toc199767544)

[1.2.1 国内研究现状 3](#_Toc199767545)

[1.2.2 国外研究现状 4](#_Toc199767546)

[1.3 研究目标和内容 5](#_Toc199767547)

[1.4 论文组织框架(todo) 5](#_Toc199767548)

[2 开发技术及开发环境 7](#_Toc199767549)

[2.1 开发技术 7](#_Toc199767550)

[2.1.1 Vue简介 7](#_Toc199767551)

[2.1.2 Axios简介 7](#_Toc199767552)

[2.1.3 Java简介 7](#_Toc199767553)

[2.1.4 Spring Boot简介 8](#_Toc199767554)

[2.1.5 MyBatis-Plus简介 9](#_Toc199767555)

[2.1.6 MySQL简介 9](#_Toc199767556)

[2.1.7 Maven简介 10](#_Toc199767557)

[2.1.8 JDK简介 10](#_Toc199767558)

[2.1.9 MD5简介 11](#_Toc199767559)

[2.1.10 ArkTS简介 11](#_Toc199767560)

[2.2 开发环境 11](#_Toc199767561)

[3 系统分析 13](#_Toc199767562)

[3.1设计思想 13](#_Toc199767563)

[3.2 设计原则 13](#_Toc199767564)

[3.3 可行性分析 14](#_Toc199767565)

[3.3.1 技术可行性 14](#_Toc199767566)

[3.3.2 操作可行性 14](#_Toc199767567)

[3.4 需求分析 15](#_Toc199767568)

[3.4.1 功能性需求 15](#_Toc199767569)

[3.4.2 非功能性需求 18](#_Toc199767570)

[4 系统概要设计 20](#_Toc199767571)

[4.1 系统用例图 20](#_Toc199767572)

[4.2 系统功能模块设计 27](#_Toc199767573)

[4.3 数据库设计 28](#_Toc199767574)

[4.3.1 数据库概念结构设计 28](#_Toc199767575)

[4.3.2 数据库表结构设计 35](#_Toc199767576)

[4.4 安全性与可靠性设计 40](#_Toc199767577)

[5 系统详细设计 42](#_Toc199767578)

[5.1 系统功能模块设计 42](#_Toc199767579)

[5.1.1 菜品管理模块 42](#_Toc199767580)

[5.1.2 套餐管理模块 49](#_Toc199767581)

[5.1.3 订单管理模块 51](#_Toc199767582)

[5.1.2 移动端业务开发 53](#_Toc199767583)

[6 系统实现 56](#_Toc199767584)

[7 结论与展望 56](#_Toc199767585)

[8 参考文献 58](#_Toc199767586)

[9 致 谢 60](#_Toc199767587)

绪论

随着国民经济的快速发展，人民消费水平不断提高，促使了餐饮行业的繁荣。尤其是学生和上班族这些群体，他们忙碌的学习和工作使得他们更加注重节约时间。现在人们对食物的品种和味道的要求越来越高，同时也对如何方便快捷地订餐提出了更高的要求。因此，外卖行业应运而生，旨在节约用餐时间，提供高效的服务。市场上出现了很多第三方外卖平台，比如美团外卖、饿了么外卖、百度外卖等。许多餐饮店也开始依赖于这些外卖平台，但同时也带来了一些新问题。例如，平台之间的竞争会影响到商家的收益，平台收取的佣金较高，要求商家降低利润做活动，同时还存在营销策略受限的问题等。因此，许多大型和连锁餐饮企业选择不依赖第三方外卖平台，而是独立开发一套自己的餐饮外卖系统直接面向用户。然而，中小型餐厅由于资金和技术问题，仍然采用传统的订餐方式。通过调查发现，中小型餐厅将餐饮与网络应用结合的程度并不普遍，因此提出了区域性餐饮网络营销的想法，即以城区中小型餐厅为服务对象，结合市场特点和消费者购物行为，开发一个开放型餐饮服务平台，对于解决餐饮企业问题和为市民提供服务具有重要意义。

通过分析中小型餐饮痛点，决定以城区中小型餐厅为主体服务对象，结合城区周边住宅和学校园区的特点，开发出一套满足中小型餐饮需求的餐饮外卖系统。减轻餐饮店对第三方外卖平台的依赖和支出，实现更自由和更低成本的营销策略。减轻中小型餐厅的负担，提高收入。

本系统是面向所有人的餐饮外卖系统。系统用户角色可以分为管理员和顾客。管理员可以登录后台管理端进行后台工作的管理，顾客可以登录前台餐厅进行操作管理员可以管理员工信息、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、修改个人信息等。可以查看菜品信息、查看菜品详情、选择菜品规格、管理购物车、收货地址管理、修改个人信息、查看订单信息、下单支付等。

本项目选择使用Java语言进行开发，Java语言具有可跨平台和可移植性的特点，使项目能在不同的系统上运行，解决兼容性问题。此次开发的餐饮外卖系统是面向所有人的，所以在前台采取页面简洁的方法，前台使用Vue和Element-UI组件，使用户能快速、简便的浏览网页，使得用户体验更加良好。后台全部使用轻量级的Vue前端框架，操作简便，功能齐全，界面简单而优美使得用户体验以及功能都能得到保障。在后端上选择使用较为火热的Spring Boot框架，并整合MyBatis-Plus框架，在前后端交互上选择使用JQuery+AJAX技术，AJAX 是一种用于创建快速动态网页的技术。通过在后台与服务器进行少量数据交换，AJAX 可以使网页实现异步更新。 移动端以适配HarmonyOS为目标，总体来说，此次餐饮外卖系统的开发环境和运行环境需要的软件和工具比较复杂，但是它可以实现跨平台的应用程序开发和运行，具有很好的移植性和扩展性。

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。市场上出现了很多第三方外卖平台，比如美团外卖、饿了么外卖、百度外卖等。许多餐饮店也开始依赖于这些外卖平台，但同时也带来了一些新问题。例如，平台之间的竞争会影响到商家的收益，平台收取的佣金较高，要求商家降低利润做活动，同时还存在营销策略受限的问题等。因此，许多大型和连锁餐饮企业选择不依赖第三方外卖平台，而是独立开发一套自己的餐饮外卖系统直接面向用户。然而，中小型餐厅由于资金和技术问题，仍然采用传统的订餐方式。通过调查发现，中小型餐厅将餐饮与网络应用结合的程度并不普遍，因此提出了区域性餐饮网络营销的想法，即以城区中小型餐厅为服务对象，结合市场特点和消费者购物行为，开发一个开放型餐饮服务平台，对于解决餐饮企业问题和为市民提供服务具有重要意义。

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SpringBoot+MyBatis-Plus实现订单实时持久化，响应时间≤200ms。

（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）优化资源利用：电子化管理减少了对物理存储空间的需求，节约了大量的纸张和存储空间。同时，电子档案的云端存储和访问，使得资源利用更加灵活和高效。

（4）​​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）。

（5）​​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%。

（6）​​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

餐饮管理系统研究虽成熟，但国内外均依赖​​单端或弱协同架构​​（如SpringBoot+Android/iOS）。国内华为虽推出HarmonyOS支持跨设备协同，但行业应用集中于智能家居；国外则以React Native/Flutter实现基础跨端，无法解决设备深度互操作问题。目前，​​SpringBoot后端与鸿蒙分布式前端的全栈融合在餐饮领域尚属空白​​，尤其缺乏针对订单全链路实时协同（顾客点餐→厨房接单→配送追踪）的实证研究。本课题将填补该技术缺口。

### 1.2.1 国内研究现状

国内餐饮管理系统的研究已形成以​​SpringBoot技术栈为核心​​、​​HarmonyOS分布式生态为创新方向​​的格局。SpringBoot凭借其快速开发与微服务架构优势，成为90%国内餐饮系统的后端主流框架，结合MyBatis-Plus、Vue.js实现高并发订单处理（如二维火系统支持5000+TPS）和前后端分离，但存在三大瓶颈：

​​ 一是数据孤岛问题突出​​，POS系统、供应链管理、CRM模块间缺乏深度整合，导致采购决策滞后，食材损耗率仍高达20%；​​二是第三方平台依赖严重​​，中小商户使用美团、哗啦啦等SaaS平台需承担15%-20%手续费，且数据控制权弱化；​​三是智能化深度不足​​，AI应用多局限于图像识别（如菜品结算），缺乏基于大数据的销售预测与动态库存优化模型。

近年来，​​HarmonyOS的引入正推动行业变革​​：欧盈万酒店平台已验证其与SpringBoot后端的兼容性，通过分布式技术实现手机点餐、厨显屏实时同步、智能手表催单等跨端协同（响应延迟＜100ms）；同时依托“碰一碰”NFC标签触发鸿蒙服务卡片，结合用户历史订单数据实现个性化推荐，复购率提升25%；原生鸿蒙系统通过100余个系统级入口简化点餐流程，降低中小商户对Android/iOS的依赖，加速国产化替代。

未来趋势聚焦​​云边端协同架构​​（SpringBoot云端处理核心业务，HarmonyOS边缘端实现厨显设备、智能厨电的轻量化控制）及​​全链路智能化​​（整合供应链数据预测需求，联动后厨设备），但跨品牌设备适配需定制驱动，且中小餐饮企业自建系统成本仍较高（≥10万元），需进一步优化跨平台兼容性。

### 1.2.2 国外研究现状

国外餐饮管理系统的研究呈现​​AI深度驱动​​与​​生态开放化​​两大特征，技术路线与中国市场存在显著差异。在技术架构层面，以​​轻量化SaaS平台为主导​​（如美国Toast、英国Lightspeed），依赖公有云部署实现快速接入，但普遍采用闭源技术栈导致定制受限。功能创新上聚焦​​数据智能应用​​：Toast通过机器学习分析历史订单，实现食材损耗率降至12%（动态预测准确率达87%）；日本Recruit集团旗下AIR-RESTAURANT系统集成NLP技术，支持多语言复杂语音点餐（英语/日语混合指令识别准确率95%）。同时，​​物联生态整合更为深入​​：美国Square平台打通厨房设备API，实现灶具温度数据自动上报调度中心；德国Lightspeed与智能冷库系统互联，实时监控食材保存环境（温湿度波动±0.3℃）。值得注意的是，​​合规与隐私成为核心设计要素​​：欧盟系统强制符合GDPR要求，如荷兰TheFork采用差分隐私技术处理用户订单数据，匿名化率达100%。然而，国外研究对​​国产操作系统适配几乎空白​​，未出现类似HarmonyOS的分布式协同案例，其多端联动依赖安卓/iOS标准化协议（如蓝牙MESH），在跨设备低延迟交互（最低200ms）及国产芯片适配方面落后于中国方案。未来发展将强化​​生成式AI应用​​（如OpenAI合作Upserve开发菜单智能生成工具）和​​低碳化运维​​（英国Nutritics平台自动计算菜品碳足迹），但技术壁垒加剧垄断，中小企业需支付更高订阅费用（平均$150/月），与中国方案的普惠性形成对比。

1.3 研究目标和内容

本研究旨在开发一个基于SprinBoot+HarmonyOS的餐饮管理与服务系统，以解决现有管理方式中存在的问题。该系统将采用JavaEE技术，结合IDEA开发平台和Spring Boot框架，实现电子化管理，提高效率和安全性。

本研究的内容涵盖了从系统需求分析到最终实现的全过程。首先，通过深入分析餐饮管理的现有问题和需求，确定了系统应具备的功能，包括员工管理、菜品分类管理、套餐管理、订单管理等。系统设计阶段，将采用面向对象的设计方法，构建系统的总体架构，并设计数据库模型，确保系统的数据存储和检索效率。在技术选型方面，本系统将使用Java语言作为开发语言，利用IDEA作为开发工具，前端采用Vue框架，后端采用Spring Boot框架，通过MyBatis-Plus与MySQL数据库进行交互，实现系统的业务逻辑。

系统开发过程中，将注重用户体验，设计一个用户友好的操作界面，满足不同用户（如管理员、员工、顾客等）的操作需求。同时，系统将实现套餐分类与种类的增删改查功能，以及灵活的权限分配，确保系统的安全性和可操作性。在系统测试阶段，将对各功能模块进行全面测试，确保系统功能的正确性和稳定性，及时发现并修复潜在的错误。最终，系统将部署到服务器上，并进行必要的维护和优化，确保系统在实际运行中的稳定性和可靠性。

此外，本研究还将提供用户培训和编写相关文档，帮助用户快速掌握系统使用方法，并为系统的长期运行提供支持。通过这些研究内容的实施，本项目将为高校提供一个高效、便捷的餐饮管理与服务系统，这不仅能够减轻管理人员的工作负担，也能为鸿蒙端用户提供便捷的下单和订单查询服务，具有重要的应用价值和广阔的应用前景。

1.4 论文组织框架(todo)

论文首先介绍了餐饮管理与服务系统的研究背景和意义，强调了在数字化时代，电子化管理对于提升餐饮平台管理效率和方便鸿蒙用户的重要性。接着，论文详细阐述了开发技术和开发环境，包括所采用的Vue框架、Java语言、Spring Boot框架、ArkTS语言、MyBatis-Plus以及MySQL关系型数据库等关键技术，并解释了这些技术在项目中的应用。

在系统分析阶段，论文进行了技术可行性、操作可行性的全面分析，确保了项目的实施基础。需求分析部分深入探讨了系统所需满足的功能和非功能需求，为系统设计提供了明确的指导。系统概要设计和详细设计部分则分别描述了系统的架构、功能、业务流程以及数据库设计，为实现阶段打下了坚实的基础。

实现部分通过界面截图和代码示例，具体展示了系统的主要功能模块，包括个人信息管理、健康档案管理、体检数据分析等。系统测试章节通过一系列测试用例，验证了系统的功能和性能，此外，论文还包括了参考文献的规范列举和对在研究过程中提供帮助的个人和集体的致谢。

最后，论文在结论部分总结了研究成果，并对未来的发展方向提出了建议和期望。

开发技术及开发环境

2.1 开发技术

### 2.1.1 Vue简介

Vue是一个用于构建用户界面的渐进式JavaScript框架。它以其易用性、灵活性和高效性而广受欢迎。Vue能够极大地简化前端开发的复杂性，使得开发者能够更专注于构建富有交互性和动态效果的用户界面[3]。Vue通过其独特的响应式数据绑定系统，让数据与视图之间保持自动同步，这一特性极大地减少了开发者在DOM操作上的负担，同时也提升了应用的性能和用户体验。随着Web技术的日益成熟和前端应用复杂度的增加，Vue也在不断演进，以适应更广泛的需求和挑战。除了持续的性能优化和新特性的引入外，Vue还注重与其他现代前端技术的集成，以满足不同场景下的开发需求。

### 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。

### 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。核其心特性如下：

（1）面向对象:Java 是一种面向对象的语言，支持封装、继承和多态等基本概念。这使得代码更加模块化、易于理解和维护。

（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。这使得 Java 成为开发跨平台应用程序的理想选择。

（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。它提供了RMI（远程方法调用）和CORBA（公共对象请求代理体系结构）等技术。

### 2.1.4 Spring Boot简介

Spring Boot是一个开源的Java基础框架，用于创建独立、生产级的基于Spring框架的应用程序。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[4]。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。它通过类路径中的类来自动配置 Spring 应用程序。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，具备Mybatis的功能，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。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[5]。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，可以通过配置文件或注解来管理事务的边界和行为。

（5）动态 SQL:MyBatis 支持动态 SQL，允许开发者根据条件动态生成 SQL 语句。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，使得它可以在不同的数据库环境中使用。

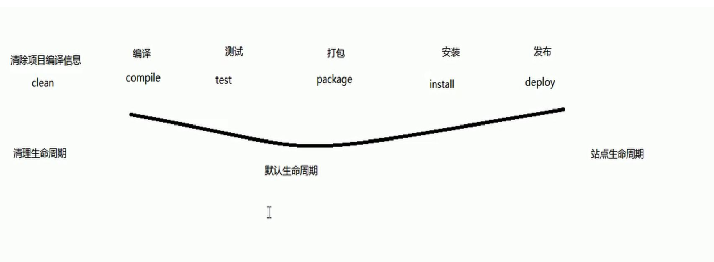
### 2.1.6 MySQL简介

MySQL是一个开源的关系型数据库管理系统，广泛应用于各种应用程序中。它以高性能、可靠性和易用性著称。MySQL使用表格形式存储数据，并支持SQL进行数据查询和管理。它适用于需要结构化数据存储和管理的场景。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，包括原子性、一致性、隔离性和持久性，保证数据操作的可靠性和一致性。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。

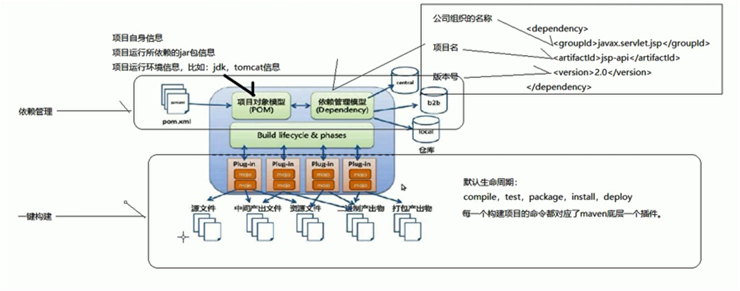
### 2.1.7 Maven简介

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，它主要用于 Java 项目。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。POM 文件中定义了项目的基本信息，如项目名称、版本、开发者信息等，同时也列出了项目所需的依赖库和插件。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。

### 2.1.9 MD5简介

MD5（Message-Digest Algorithm 5）是一种广泛使用的加密散列函数，它可以产生一个128位（16字节）的散列值（hash value），通常用32位的十六进制字符串表示。MD5由 Ronald Rivest 在1991年设计，最初被用来作为一种安全的密码散列算法。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）。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，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 1.8 版本为核心，利用其跨平台的特性和强大的社区支持。开发工具选择了 IntelliJ IDEA，这款集成开发环境以其智能代码补全、代码分析和版本控制功能著称，极大地提升了开发效率。前端技术栈则采用了 Vue.js，这是一个轻量级且易于上手的 JavaScript 框架，非常适合动态用户界面的构建。为了支持前端的运行，系统配置了 node.js 环境，这是一个基于 Chrome V8 引擎的 JavaScript 运行环境，能够提供高效的服务器端脚本执行能力。数据库方面，系统选择了 MySQL 作为数据存储解决方案，这是一个广泛使用的开源关系型数据库管理系统，以其高性能、可靠性和易用性著称。在后端框架的选择上，系统采用了 Spring Boot，这是一个简化 Spring 应用开发的工具，提供了自动配置、内嵌服务器等便利功能。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。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 |

系统分析

3.1设计思想

采用分层架构：本系统采用经典的三层架构模式，明确划分为表示层、业务逻辑层和数据访问层。

系统按照功能或业务需求进行模块化划分，每个模块具有明确的功能和职责。这种设计使得系统结构更加直观，便于开发、测试和维护。每个模块可以独立开发和测试，提高了代码的复用性和系统的可测试性。模块化设计还允许团队成员分工合作，各自负责特定模块的开发，从而提升整体开发效率。

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支持而受到青睐。Spring Boot简化了基于Spring的应用开发，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，用于存储用户信息、心理健康数据等。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**错误!未找到引用源。**。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[7]。

（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。抽象应该独立于具体实现细节，而具体实现细节则应依赖于抽象。

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。系统设计了详细的E-R模型，并构建了相应的数据库表，如用户表、分类表、订单表、日志表等，确保了数据的组织和存储满足需求。安全性方面，系统通过HTTPS协议确保数据传输和访问的安全性。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。系统设计考虑了用户易用性，提供了清晰的导航和逻辑流程，确保用户能够迅速掌握并有效使用各项功能。无论是鸿蒙端用户查看健康建议，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。此外，系统还提供了帮助和用户支持功能，以降低用户的学习难度。

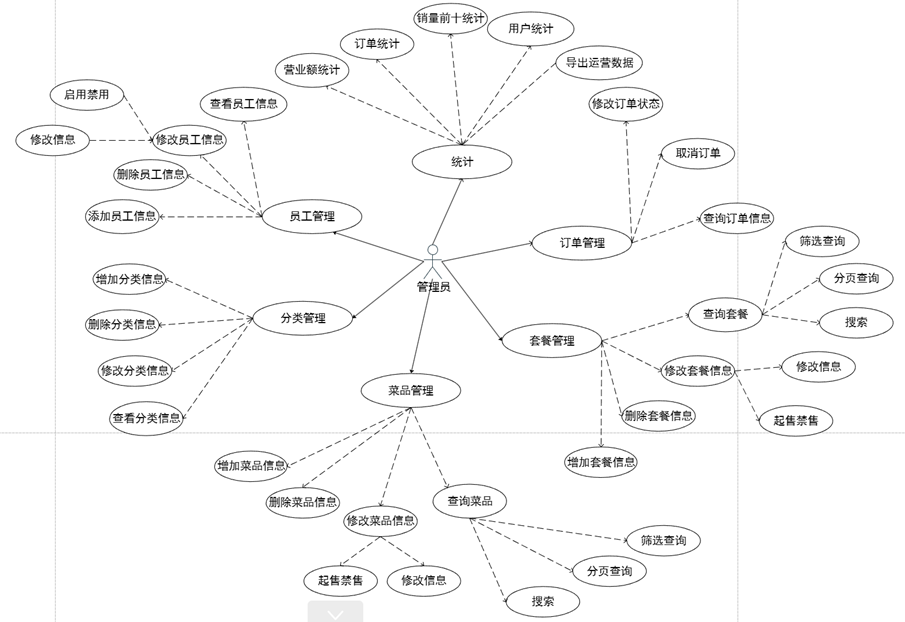
安全性方面，系统通过细致的权限管理保护用户数据，使用户可以安全地进行操作。通过全面的用户测试，包括功能测试和可用性测试，系统显示出高度的用户操作可行性，能够满足不同用户群体的需求，并确保了广泛的接受度和实用性。

3.4 需求分析

### 3.4.1 功能性需求

本系统是一个面向广大用户的外卖点餐系统，其用户角色主要划分为管理员和普通用户。管理员通过后台管理端执行各项管理任务，而用户则在前台进行各类点餐操作。管理员可以管理员工信息、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、查看营业数据统计图表、导出运营数据等。用户在前台则可以便捷地查看菜品信息、详情及规格，管理购物车和收货地址，查看订单信息，以及进行下单支付等操作。这些功能满足了用户从浏览菜品到完成点餐支付的全流程需求，提升了用户体验。

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，以及管理员三个角色，管理员用例图如图3-1 所示。



**图3-1 管理员用例图**

每个模块的主要的功能分别如下：

1. 员工管理

管理员可以在系统后台对员工信息进行管理，可以查询所有的用户的信息，也可以根据关键字进行模糊查询，也可以添加用户和修改用户信息，管理员还能禁用员工的权限，被管理员禁用的用户就不能在登录系统了。

1. 分类管理

管理员将所有分类信息查询出来，分页展示到分类管理的主页面。另外管理员可以添加菜品和套餐的分类信息，也可以修改和删除菜品和套餐的分类信息。

1. 菜品管理

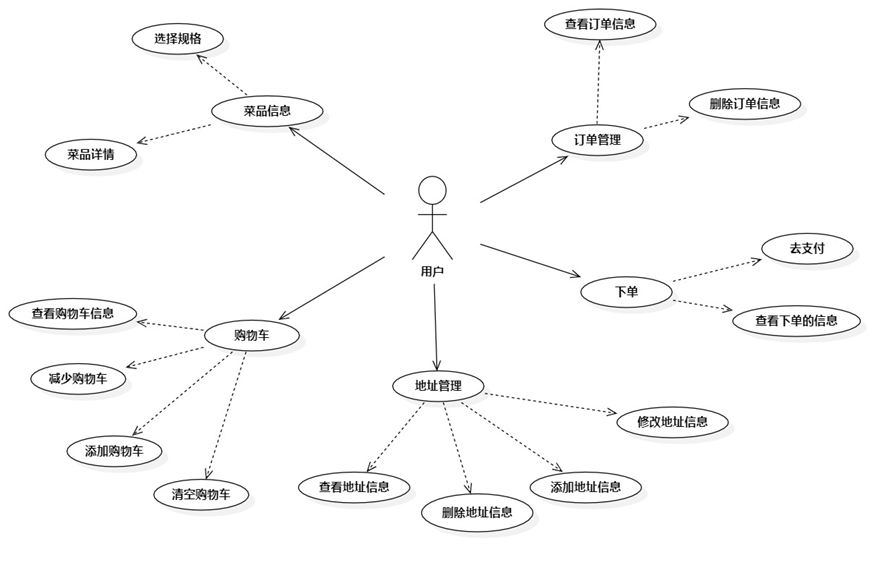
在菜品主页面中管理员会将所有的菜品信息查询出来并分页展示出来，管理员还可以对菜品信息进行增，删，改操作，除此之外还可以根据关键字对菜品进行模糊查询，方便管理员找到要操作的菜品信息。另外管理员还可以更改菜品的状态，控制该菜品是处于停售状态还是售卖状态。管理员还可以对菜品进行批量操作。

1. 套餐管理

在套餐管理主页面中管理员会将所有的套餐信息查询出来并分页展示出来，管理员还可以对套餐信息进行增，删，改操作，除此之外还可以根据关键字对套餐进行模糊查询，方便管理员找到要操作的套餐信息。另外管理员还可以更改套餐的状态，控制该套餐是处于停售状态还是售卖状态。管理员还可以对套餐进行批量操作。

1. 订单管理

管理员会将所有的订单信息查询出来并分页展示到订单管理页面，此外管理员还可以根据订单号，以及开始日期以及结束日期快速查询订单信息，另外管理员还可以查询订单的详细信息以及修改订单的状态。

用户端可以分为菜品信息，购物车，地址管理，下单，订单管理五大模块。其用例图如图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：

用户访问用户端能够查询到商家已上架并起售的菜品和套餐信息，并且可以选择口味，规格，点击菜品能查看到菜品的详细信息，点击套餐能看到套餐的所有菜品。

2．购物车：

用户选择菜品/套餐和口味规格之后能添加到购物车中，并且能够查看购物车中的菜品/套餐，以及对购物车中的菜品/套餐进行删除、修改，还可以一键清空购物车。

3．地址管理：

在地址管理页面，后台会将已有地址查询并显示出来，用户可以对已有地址进行修改和删除，还可以新增地址。点击勾选“设为默认地址”能够在之后点餐时默认使用这个地址。

4．下单

用户在菜品页面选择好商品之后点击提交订单，后台会统计商品价格以及配送费跳转到订单页面并显示出默认地址、合计价格等，在订单页面用户能够选择地址信息，备注信息，用户下单跳转到支付页面，支付成功后会跳转到这次订单信息页面。在订单信息页面用户还能够对订单进行退款申请以及再来一单的操作。

5．订单管理

用户进入订单管理界面，能够查询到所有历史订单，用户可以点击任意订单查看详细信息，并且能对订单进行申请退款、再来一单。

### 3.4.2 非功能性需求

餐饮管理与服务系统-管理员端，前端页面比较简单，安全性也比较高，能够流畅运用，适用于大多的操作系统。开发软件版本：Java语言，数据库运行工具、能够在后台存放大量的数据，且存放数据比较安全。基于过程管理，提高管理效果。并发度1000使得该平台在性能上都得到了大大提升。

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、

网络：100Mbps的无线网络

系统概要设计

基于Spring Boot + HarmonyOS的餐饮外卖系统共有后台管理员、后台普通员工和前台用户3个角色。

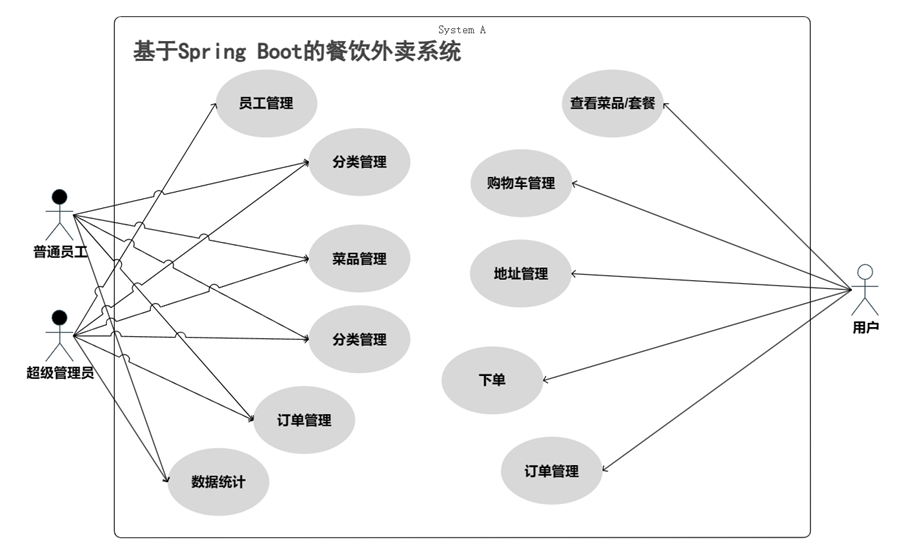
后台管理员功能模块：登录后台系统，拥有后台系统中的所有操作权限。

后台普通员工：登录后台管理系统，分类管理、对菜品、套餐、订单、数据统计等进行管理。

前台用户：登录前端页面，可以浏览菜品、添加购物车、设置地址、在线下单个人信息页面查看订单等操作。

* 1. 系统用例图

根据系统参与者的功能我们设计出系统用例图，其用例图如下图4-1所示：

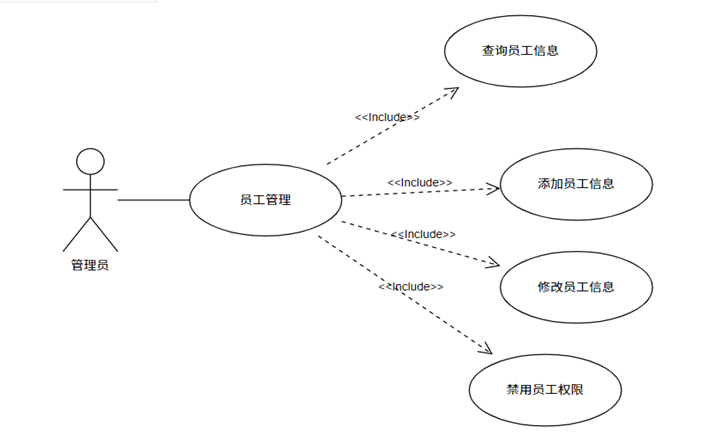


**图4-1 系统用例图**

每个模块主要功能分别如下：

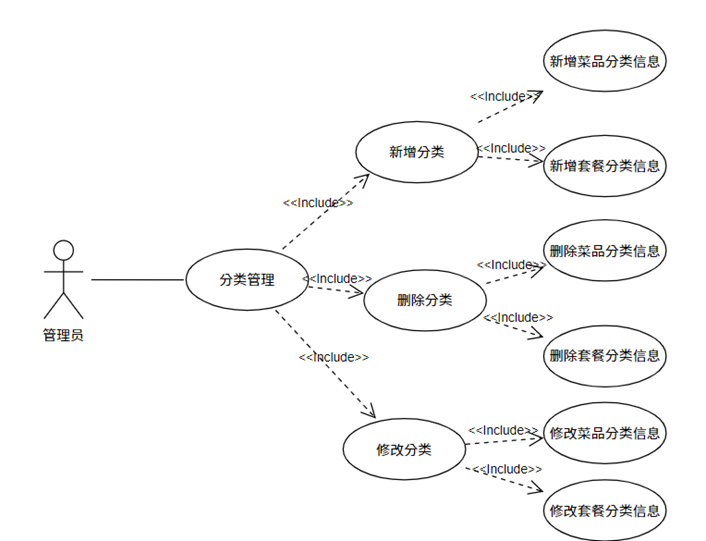
其中因超级管理员拥有后台系统所有权限，故后台管理端以其为例。

1. 员工管理：此功能模块包含了对普通员工的新增、修改、删除。当添加员工完成时可以选择保存或者保存并继续添加，当对某一员工修改信息时会回显数据库保存信息，并且可以对普通员工进行状态修改，可以选择是否禁用启用此用户，用例图如下图4-2所示。



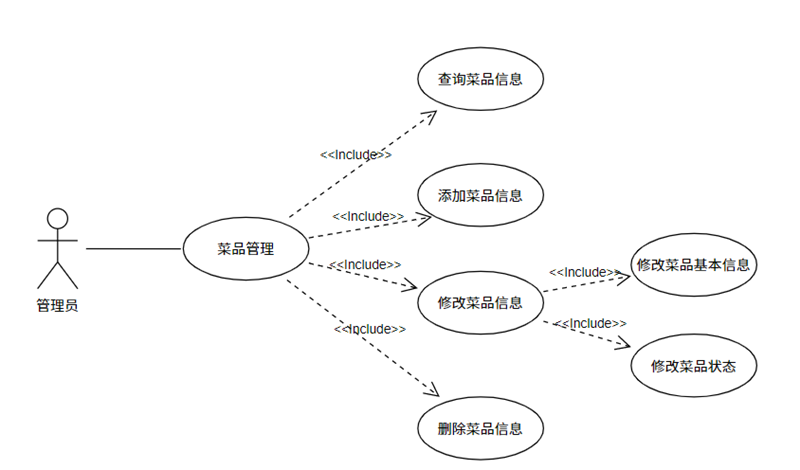
**图4-2 用户管理用例图**

2. 分类管理：在分类管理中，我们有两大分类，其一为菜品分类，二为套餐分类。而分类管理中的排序功能，为分类的种类进行排序，设置排序为升序排序，1最大。越是近1，则在移动端用户页面显示的时候，自动的分类就会越在上面。而1则为默认显示的首页，若多个分类为1，则以最后更新时间为准；在修改时会访问数据库，拿到保存的数据回显到页面上，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：菜品管理有着传统的CRUD，但是有所不同的是，在新增时会根据菜品分类进行在本分类下进行添加，分类不同菜品不同，同时在菜品中可以通过对此菜品的文件上传下载在本地，同时在页面中还有在移动端用户处显示；同时在进行菜品的新增修改时可以选择个性化口味；而批量删除则可以通过对菜品进行多选进行删除，而进行删除的菜品需要首先将菜品进行停售，也即“status=0”后才能删除；起售停售菜品就是改变菜品当前状态，用例图如下图4-4所示。



**图4-4 菜品管理用例图**

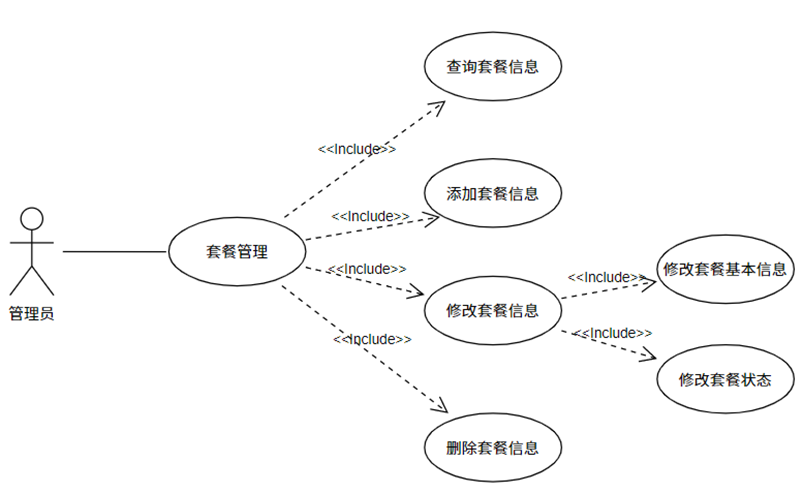
现以删除菜品为例模拟次功能模块的事件流，在删除菜品中有一个重要的关联条件为菜品是可以添加到套餐中的，即若想要删除菜品在某一正在售卖的套餐中，我们是无法对其进行删除的，否则会抛异常，套餐不完整也就没用，所以删除菜品用例规约表如下表4.1所示。

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： | 1. 管理员在本事件中可以选择首先停止销售此菜品，此时菜品状态被修改为“停售”，即“status=0” 2. 之后可以根据需求删除或批量删除想要删除的菜品。   3. 删除了菜品后，用户在移动端时是浏览不到此菜品的，当然在停售状态下的菜品也无法浏览。 |
| 扩展事件流： | 1. 若此菜品在某一套餐中，将无法被删除，只有当要删除的菜品关联的套餐没有在售卖，才可以被删除。 2. 如果认为此菜品在当前不可销售，却又不忍彻底删除则可以选择“停售”，当想要重新起售此菜品时可选择“起售”。 3. 可以批量删除不想要销售的菜品。 |
| 后置条件： | 删除菜品，将更新过的菜品数据或套餐数据保存到数据库中 |

4. 套餐管理：套餐管理与菜品管理相似，只是多了一个嵌套关系，一个套餐中

可以包含一个或多个菜品。在新增套餐中首先需对已有的套餐分类进行选择，之后在新增菜品时，可以通过已有菜品分类进行模糊查询搜索并可以批量添加；而进行删除的菜品需要首先将菜品进行停售，也即“status=0”后才能删除。在修改套餐中可以自主对套餐中的菜品进行删除，用例图如下图4-5所示。



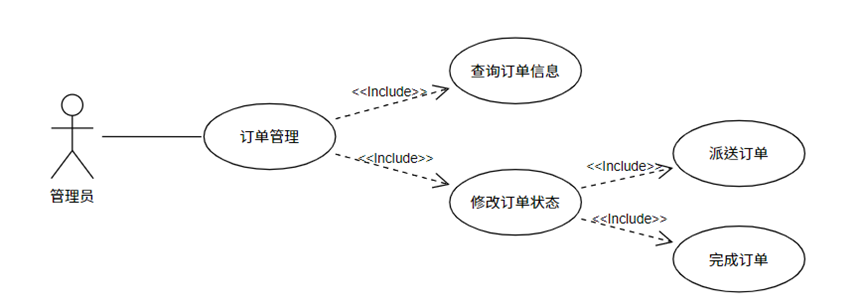
**图4-5 套餐管理用例图**

在添加套餐的时候，我们首先要选择已有的套餐分类，在此套餐分类下，我们添加一些我们各式菜品，当你想要添加菜品时，可以直接在搜索框中进行模糊查询进行添加，当添加后觉得不是很合理也可以直接进行删除菜品，当然觉得不够也可以点击“+”“-”进行个性化数量选择，之后上传文件下载到服务器显示，保存即可，所以新增套餐规约表如下表4.2所示。

表3.2 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： | 管理员角色已登录，添加的菜品起售中 |
| 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”，可以直接搜索栏搜索要添加的菜品，或者直接点击菜品分类去搜索，之后对菜品进行数量选择。 |
| 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，将更新过的套餐数据保存到数据库中 |

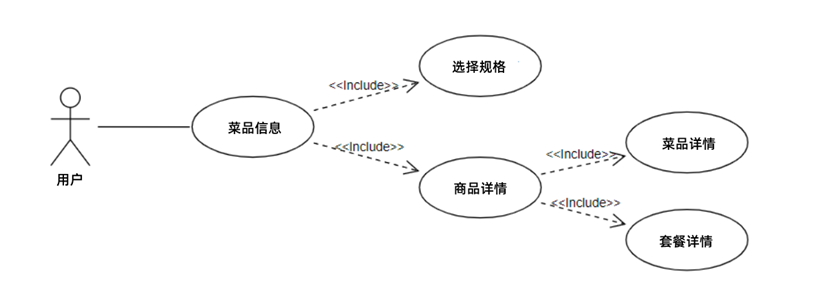
5. 订单管理：当用户选择菜品或套餐购入购物车后，在管理后端的订单管理中会显示当前的“订单明细”，管理员可以手动操作订单是否完成，订单状态会由“正在派送”更改为“已派送”，也可以查看订单的详细信息，用例图如下图4-6所示。



**图4-6 订单管理用例图**

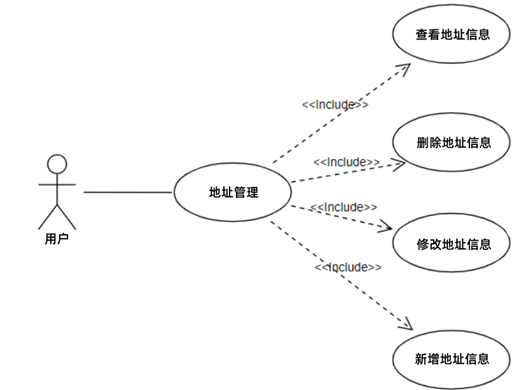
**用户端**

6．菜品信息：用户访问用户端能够查询到商家已上架并起售的菜品和套餐信息，并且可以选择口味，规格，点击菜品能查看到菜品的详细信息，点击套餐能看到套餐的所有菜品。用例图如图4-7所示。



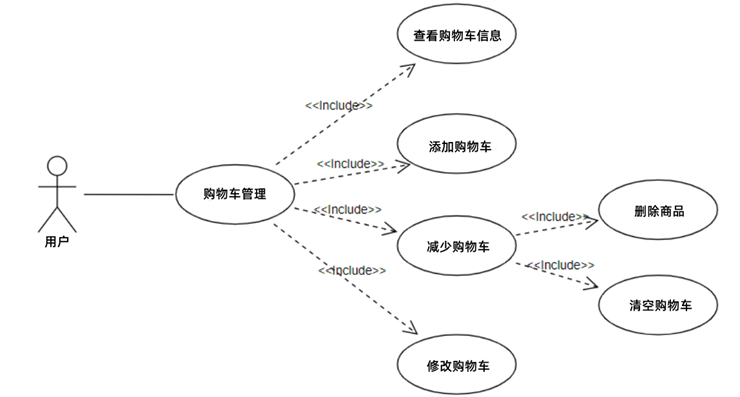
**图4-7 菜品信息用例图**

7．地址簿管理：开发移动端时，当用户购入菜品后下单要填写收货地址，添加收货地址并且可以给此地址打上一个标签，表示其属性，并且可以设置地址是否为默认地址，如果地址为默认地址则，此账户下一次下单后直接以此地址作为收货地址，同时可以去个人主页的地址管理可以查询指定用户的所有地址并可以对地址的CRUD，用例图如下图4-8所示。



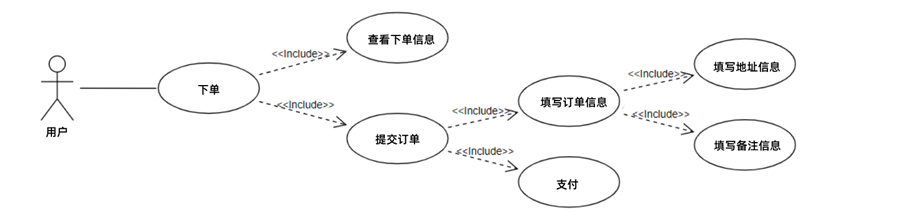
**图4-8 地址簿管理用例图**

8. 购物车管理：当用户想要选择菜品或者是套餐时，就可以点击“+”加入购物车，想要什么就添加什么，也可以点击购物车一栏查看购物车的选择，不想要的点击“—”，也可以直接选择清空购物车，用例图如下图4-9所示。



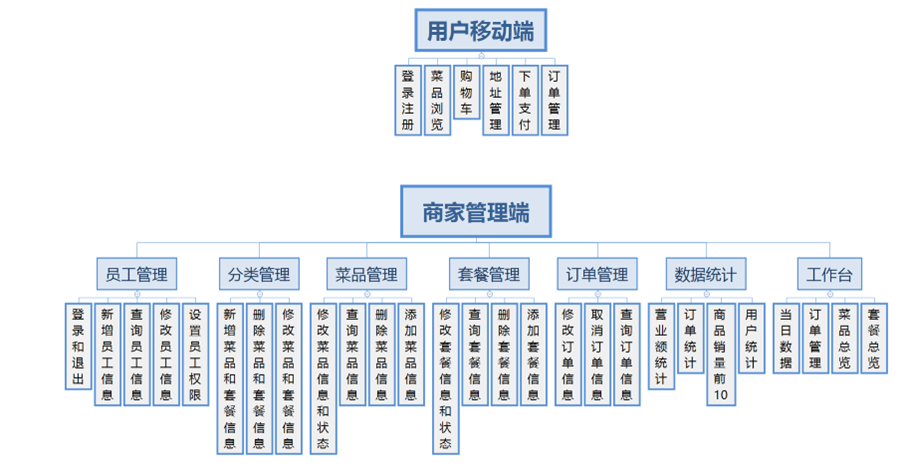
**图4-9 购物车管理用例图**

9. 下单：用户在菜品页面选择好商品之后点击提交订单，后台会统计商品价格以及配送费跳转到订单页面并显示出默认地址、合计价格等，在订单页面用户能够选择地址信息，备注信息，用户下单跳转到支付页面，支付成功后会跳转到这次订单信息页面。用例图如图4-10所示。



**图4-10 下单用例图**

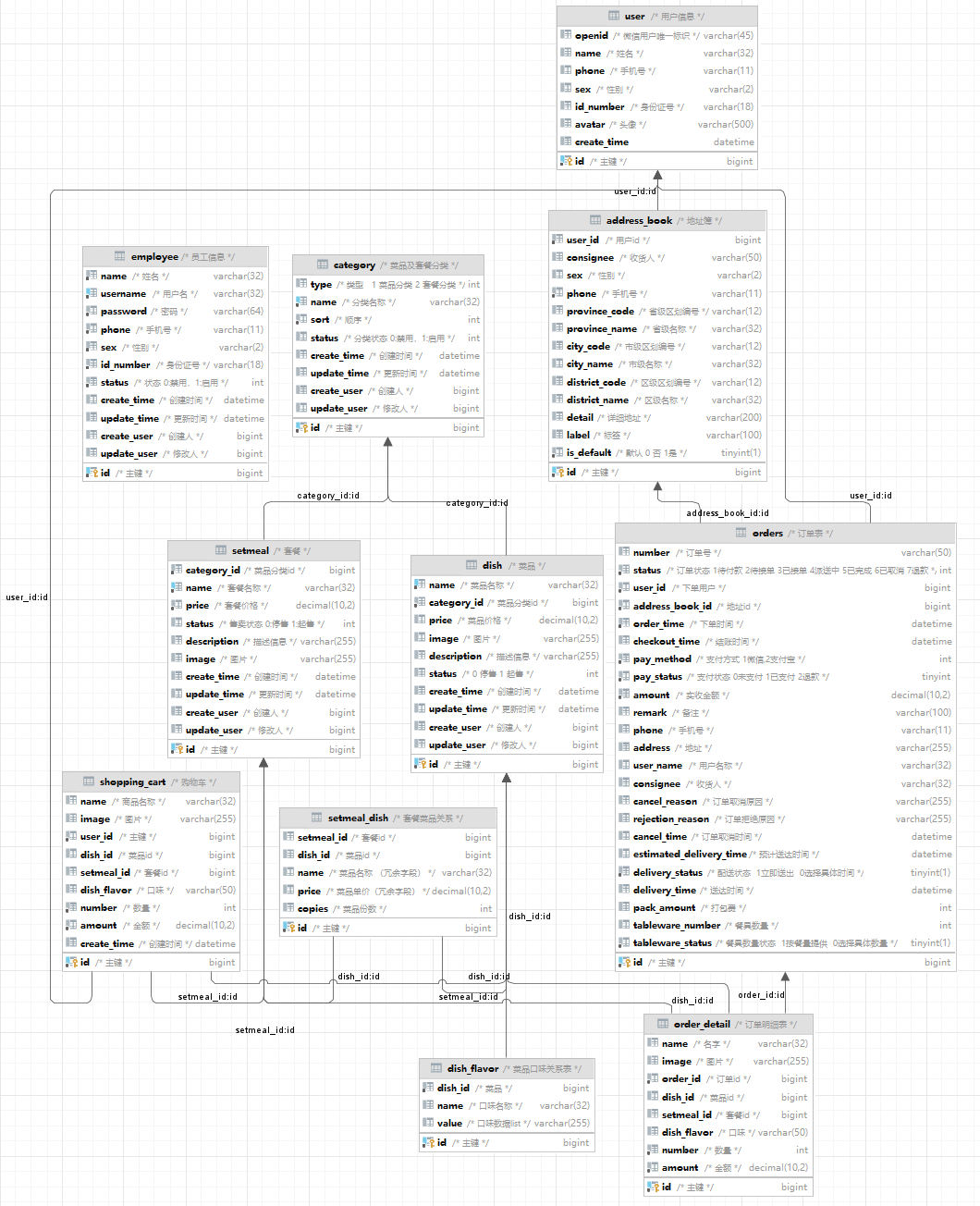
4.2 系统功能模块设计

餐饮管理与系统是由用户移动端、商家管理端两个模块相互结合而成。餐饮外卖系统功能图如图4-11所示。

**图4-11 餐饮管理服务系统后端功能图**

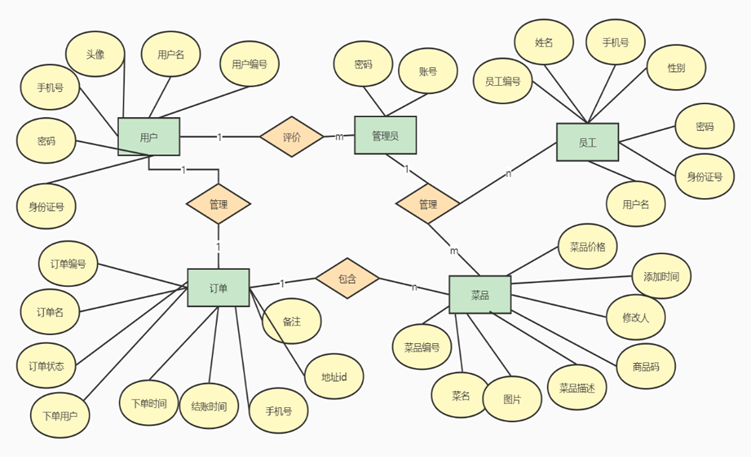
4.3 数据库设计

4.3.1 数据库概念结构设计

通过上述对基于餐饮管理与服务系统管理-管理员端的设计与实现，可以得出实现该系统需要使用的数据实体，从而得知该系统的类图如图4-12所示。

**图4-12 系统类图**

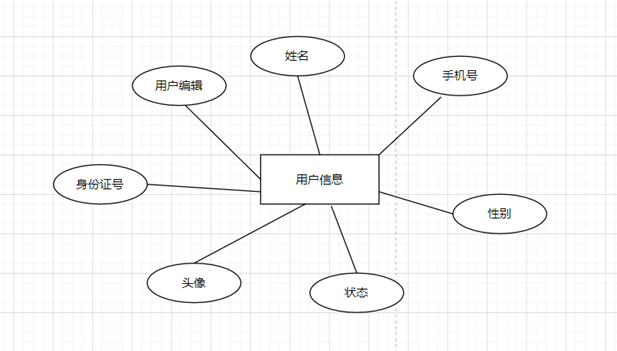
通过上述对基于餐饮管理与服务系统管理-管理员端的设计与实现，可以得出实现该系统需要使用的数据实体，从而得知各个实体属性图如下图4-13所示。



**图4-13 用户下单实体图**

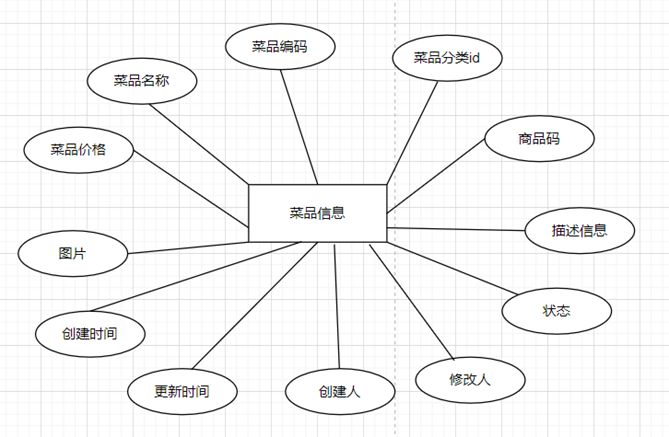
我抽象出系统的实体，并设计部分E-R图，如下。

用户信息（用户编号、姓名、手机号、性别、身份证号、头像、状态）如图4-14 所示。



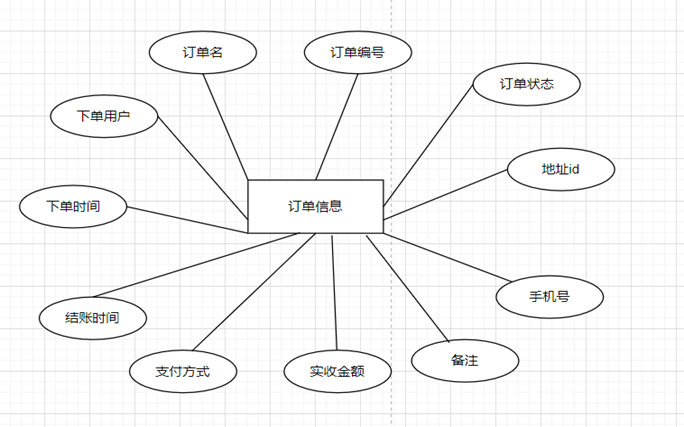
**图4-14 用户信息实体图**

菜品信息（菜品编号、菜品名称、菜品分类id、菜品价格、商品码、图片、描述信息、状态、顺序、创建时间、更新时间、创建人、修改人）如图4-15所示。



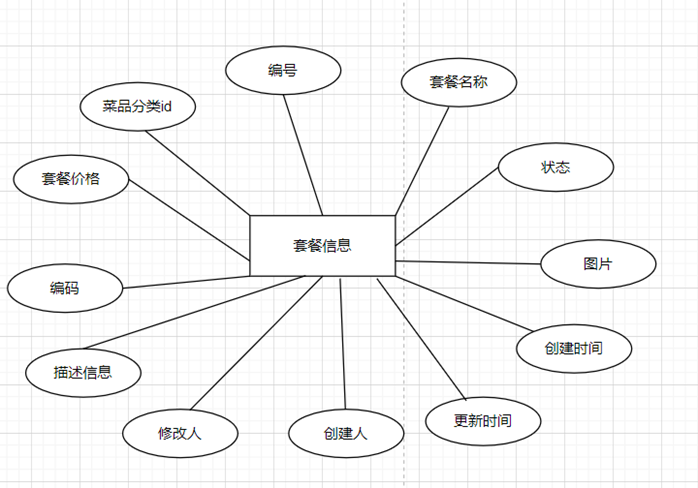
**图4-15 菜品信息实体图**

订单信息（订单编号、订单名、订单状态、下单用户、地址id、下单时间、结账时间、支付方式、实收金额、备注、手机号）如图4-16所示。

****

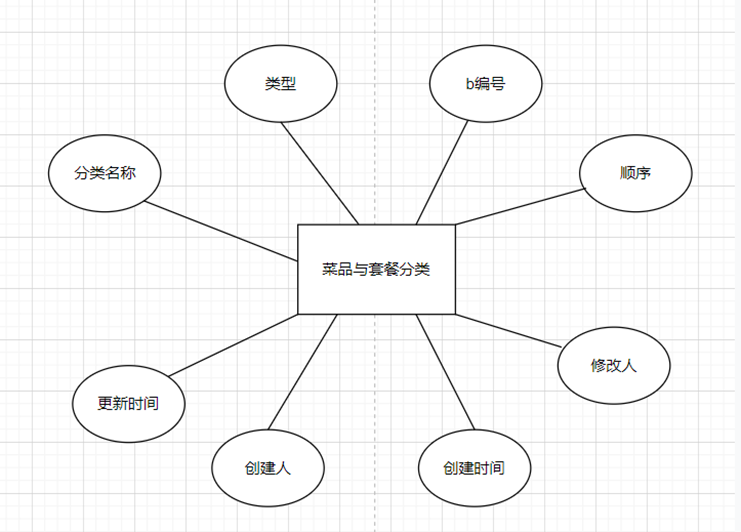
**图4-16 订单信息实体图**

套餐信息（编号、菜品分类id、套餐名称、套餐价格、状态、编码、描述信息、图片、创建时间、更新时间、创建人、修改人）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（编号、类型、分类名称、顺序、创建时间、更新时间、创建人、修改人）如图4-18所示。



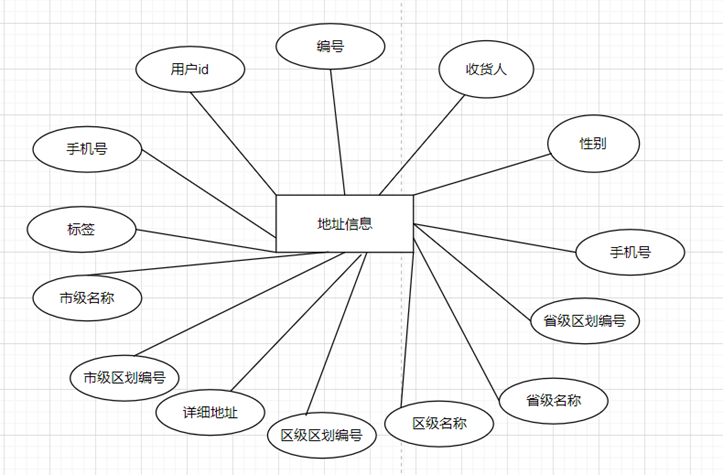
**图4-18 菜品与套餐分类信息实体图**

员工信息（员工编号、姓名、用户名、密码、手机号、性别、身份证号、状态、创建时间、更改时间、创建人、修改人）如图4-19所示。



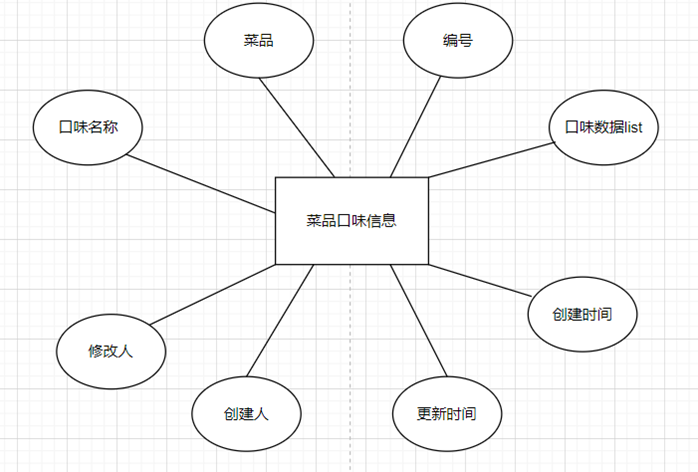
**图4-19 员工信息实体图**

地址信息（编号、用户id、收货人、性别、手机号、省级区划编号、省级名称、市级区划编号、市级名称、区级区划编号、区级名称、详细地址、标签）如图4-20。



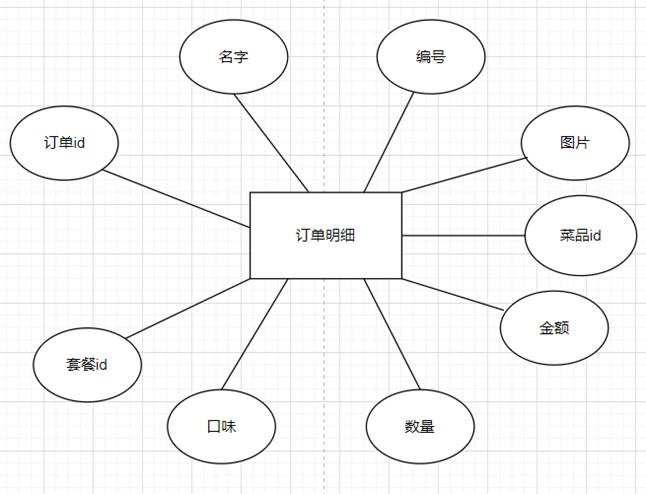
**图4-20 地址信息实体图**

菜品口味信息（编号、菜品、口味名称、口味数据list、创建时间、更新时间、创建人、修改人）如图4-21所示。



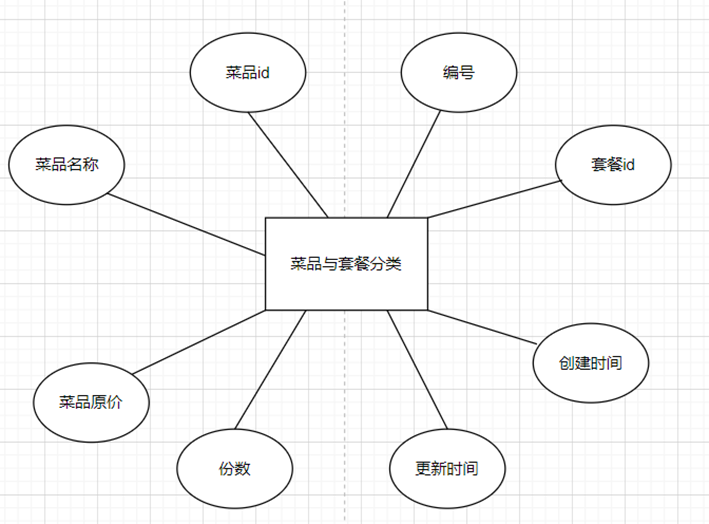
**图4-21 菜品口味信息实体图**

订单明细（编号、名字、图片、订单id、菜品id、套餐id、口味、数量、金额）如图4-22所示。



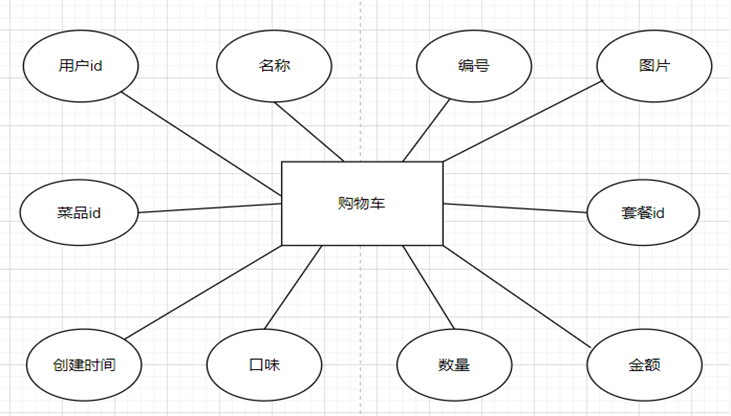
**图4-22 订单明细信息实体图**

套餐菜品关系信息（编号、套餐id、菜品id、菜品名称、菜品原价、份数、创建时间、修改时间）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（编号、名称、图片、用户id、菜品id、套餐id、口味、数量、金额、创建时间）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

根据数据库概念结构设计中的实体及属性描述、以及数据库的规范化理论，设计管理员信息表、分类信息表、菜品管理信息表、菜品口味关系表、员工信息表、订单信息表、订单明细表、套餐信息表、套餐菜品关系表、购物车信息表和用户信息表、表4.1至表4.11是各个数据库表的逻辑结构。

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，3已派送，4已完成，5已取消 | 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 套餐菜品关系表

|  |  |  |  |
| --- | --- | --- | --- |
| 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

以防止数据在传输过程中被窃取或篡改，系统中所有敏感信息，如用户密码、支付详情、商家经营数据等，都必须经过严格的加密处理。综合考虑选择使用唯一性较强的MD5加密算法，其特点是固定长度，不可逆。新增管理员时，将密码以MD5密文的形式存储到数据库中。登录时客户端输入密码会想将明文密码进行MD5加密，使其成为32位MD5密文，再取出数据库中存储的密文进行比对。流程如图4-25所示。



**图4-25 流程图**

2.负载均衡

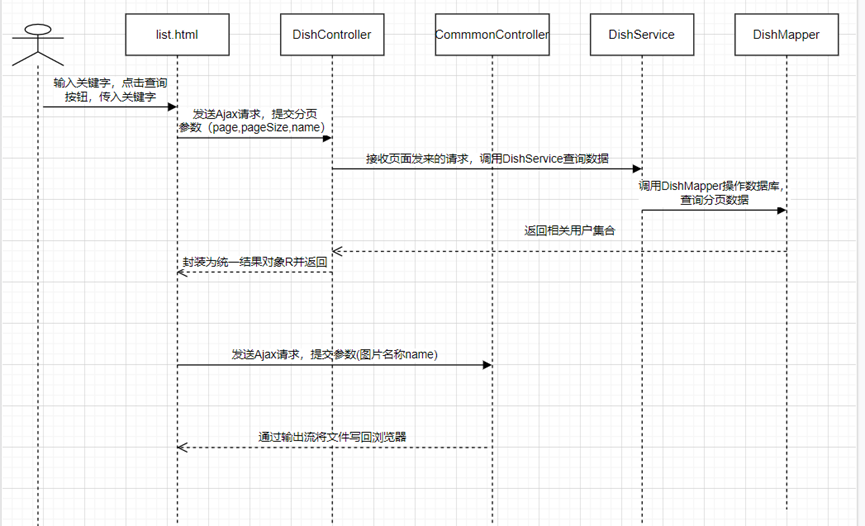
负载均衡服务器通过一定的调度算法将客户端的流量分发到不同的应用服务器上面，以实现性能的水平扩展及避免单点故障出现，提升系统的可靠性。这一点我们可以使用Nginx实现，修改Nginx配置文件使得前端的请求分配给多个服务器处理，同时修改负载均衡方式为轮询。

系统详细设计

5.1 系统功能模块设计

* + 1. 菜品管理模块
       1. 菜品管理主界面

菜品管理模块包括菜品的模糊查询、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。菜品的模糊以及分页查询的顺序图如图5-1所示。



**图5-1 菜品模糊查询顺序图**

（1）在list.html页面上，管理员通过点击输入框输入关键字，再点击查询按钮，就会出发点击事件发送Ajax请求，将关键字name，当前页数page以及每页的信息条数pageSize作为参数传入，调用DiahServlet中的page方法。

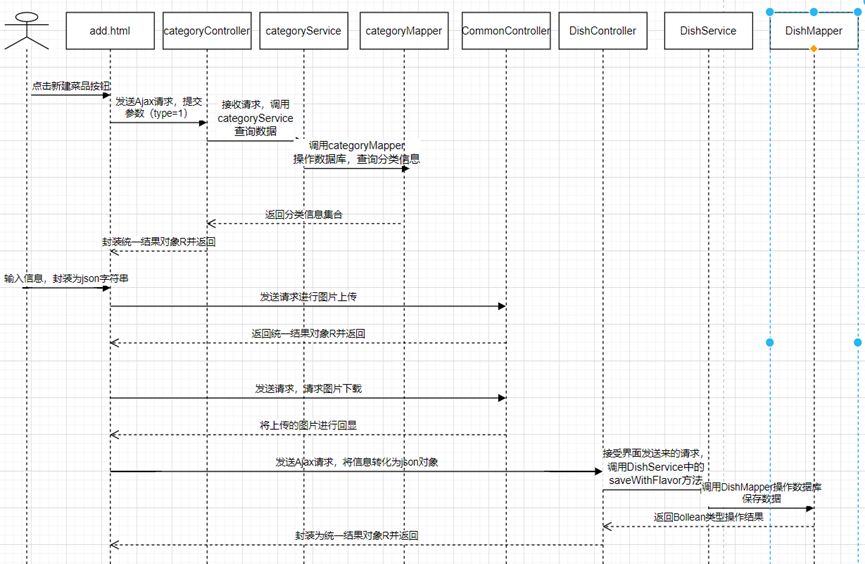
（2）DishServlet在得到参数后，会先以page，pageSize为参数来构建页面结构，再构建一个条件生成器 queryWrapper，再判断name是不是空，若不是，则把名称做为queryWrapper.like的参数来增加筛选条件，最后以 pageInfo、 queryWapper为参数来调用 DishService的page方法。在前端页面上进行菜的归属分类显示时，所需的是菜分类名 categoryName，而我们在查询菜单时，只能得到该菜的类别 id，而不是该类别名。因此，必须将每个菜 dish对象中的数据都包装在 dishDto对象中，并通过对 dishDto对象的 categoryName属性来实现在前端的正常显示。接下来，我们需要构建一个泛型为DishDto的分页构造器，命名为dishDtoPage。然后，从pageInfo对象中提取当前页的数据列表records。随后，我们将遍历records集合，并为集合中的每一个dish对象创建一个对应的DishDto对象，同时将dish对象中的数据赋值给DishDto对象，并特别设置DishDto对象的categoryName属性。在遍历records集合的过程中，我们将新创建的每一个dishDto对象添加到泛型为DishDto的List集合list中。这样，我们就得到了当前页需要显示的数据集合。最后，我们将当前页的数据集合list以及总条数封装到dishDtoPage对象中，并返回该对象，以供前端展示使用。

（3）DishService接收到参数之后，调用page方法将name拼接成数据库模糊查询所要求的字段格式（“%name%”）最后传到数据库查询Dao层的DishMapper中进行数据库的操作。

（4）Dao层接收到传过来的参数后。把参数放入SQL语句中进行数据库的查询操作，然后将查询结果封装为商品信息集合返回给控制器，最后组装成返回结果类R返回给前端视图。

5.1.1.2 新增菜品信息

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**

（1） 管理员点击新建菜品，页面会将类型type封装为Category对象，然后将Category对象作为参数，发送一个Ajax请求，请求调用CategoryController中的list方法将菜品分类显示到下拉框中。

（2） CategoryController接收到参数之后，首先取出type将type作为参数构造分页构造器，然后构造一个条件构造器queryWrapper，然后判断type是否为空，如果不为空，将type作为queryWrapper.eq的参数添加过滤条件，最后将queryWapper作为参数调用CategoryService的list方法。

（3） CategoryService接收到参数之后，调用list方法将type拼接按条件查询所要求的格式，最后传到数据库查询Dao层的CategoryMapper中进行数据库的操作。

（4） Dao层接收到传过来的参数后。把参数放入SQL语句中进行数据库的查询操作，然后将查询结果封装为商品信息集合返回给控制器，最后组装成返回结果类R返回给前端视图。

（5） 管理员输入菜品信息点击上传图片，页面会发送一个请求调用CommonController中的upload方法将图片保存。

（6） 图片保存之后，页面会在发一个请求调用CommonController中的download方法对图片进行下载并回显。

（7） 管理员输入完菜品信息之后点击保存按钮，就会触发点击事件，将填写的信息封装为json对象，并发送Ajax请求调用DishController中的SaveWithFlavor(obj)方法。

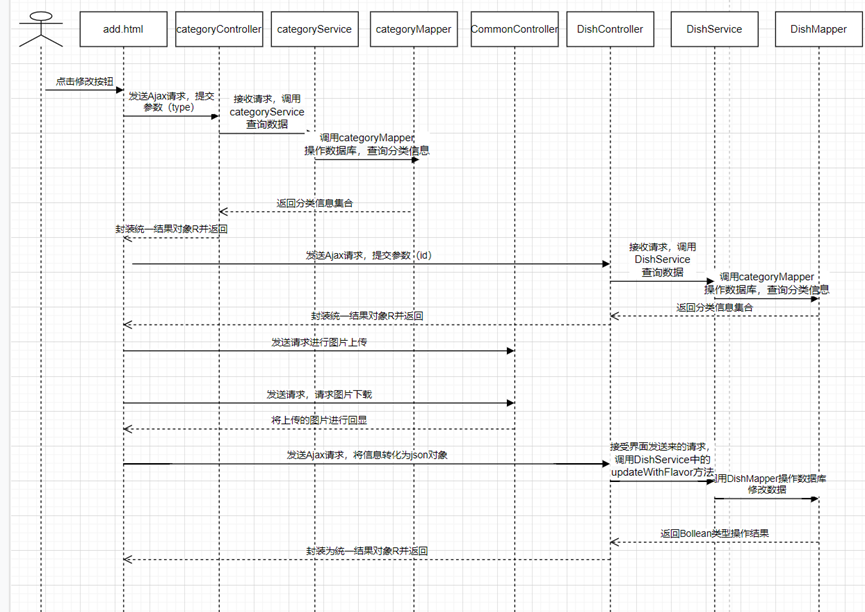
（8） DishController通过@RequestBody注解将json对象转换为java对象，调用DishController中的SaveWithFlavor(obj)方法，将该Java对象作为参数传入。

（9） DishService接收到参数之后，首先调用调用Dao层的DishMapper中的save(obj)方法将商品信息对象传入进行数据库的操作。然后从对象中获取菜品id和菜品口味，然后遍历菜品口味集合，给每个口味对象设置菜品id，最后调用dishFlavorService中的saveBatch()方法调用调用Dao层的dishFlavorMapper中的saveBatch()方法将商品信息对象传入进行数据库的操作。

（10）Dao层接收到传过来的参数后。把参数放入SQL语句中进行数据库的查询操作，然后将查询结果封装为商品信息集合返回给控制器，最后组装成返回结果类R返回给前端视图。

5.1.1.3 修改菜品信息

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**

（1）管理员点击修改菜品，页面会将类型type封装为Category对象，然后将Category对象作为参数，发送一个Ajax请求，请求调用CategoryController中的list方法将菜品分类显示到下拉框中。

（2）页面将菜品id作为参数，发送Ajax请求，请求调用DishController中的get()方法。

（3）DishControlle通过@PathVariable注解解析拿到菜品id，将菜品id作为参数调用DishService中的getById (id) 方法传到数据库查询的Dao层的DishMapper中调用selectById()方法进行数据库的操作来获取菜品基本信息。然后将获取到菜品基本信息封装到dish对象中，然后将dish对象中的信息拷贝到dishDto对象中，然后调用dishFlavorService中的list()方法，根据菜品id查询当前菜品所对应的口味信息，然后将口味信息设置在dishDto对象中，最后返回dishDto对象。

（4）Dao层接收到传过来的参数后将用户id取出，放入SQL语句中进行数据库的查询，返回对应的菜品信息到add.html中回显。

（5）界面还会发送一个请求，请求调用CommonController中的download方法进行图片的回显。

（6）所有信息回显之后，管理员可以输入要修改的信息，输入完毕之后点击保存按钮，之后就会页面就会将菜品信息数据封装为json对象，并发送Ajax请求，请求调用DishController中的update方法。

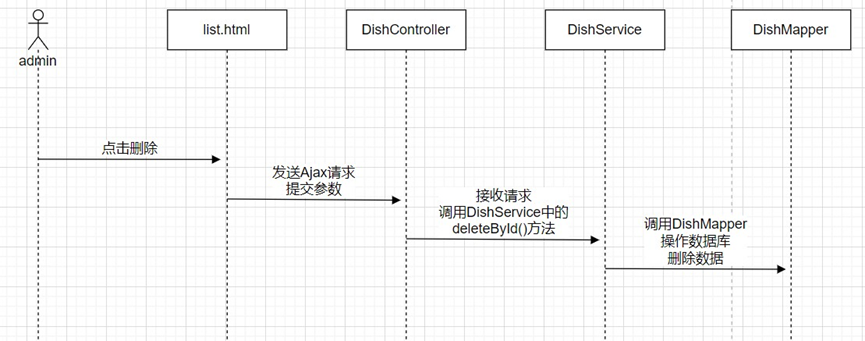
（7）DishController通过@RequestBody注解将接收到的json对象转为java对象，然后调用EmployeeService中的updateWithFlavor(obj)方法。

（8）DishService接收到Java对象的参数后，首先将参数传到数据库查询Dao层的DishMapper中调用updateById(obj)方法进行数据库的操作。然后调用dishFlavorService中的remove()方法清理当前菜品的所有口味信息。然后再调用dishFlavorService中的saveBatch()方法添加当前提交过来的口味数据。

（9）Dao层接收到传过来的参数后。把Java对象中的字段取出，放入SQL语句中进行数据库的修改，修改成功后返回Integer类型的值给控制器，如果为1则修改成功，否则将修改失败，最后控制器再将结果组装成R返回给前端视图。

5.1.1.4 删除菜品信息

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**

（1）在list.html上，管理员通过点击删除按钮，发送Ajax跨域请求提交菜品id，调用DishController中的deleteById()方法。

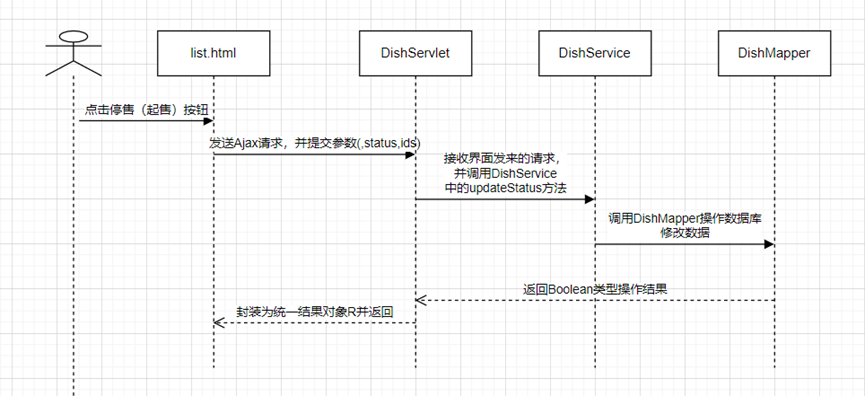
（2）DishController通过将接收的菜品id作为参数调用DishService中的deleteByIds()方法。

（3）DishService接收到参数后，首先判断参数中是否含有“，”，如果不含“，”就说明是删除单个菜品信息，就直接将参数传到数据库查询Dao层的DishMapper中调用removeById(id)方法;如果含有“，”，就说明删除的是多个菜品信息，就调用split方法根据“，”将各个id分割出来，然后创建一个list集合，将分割出来的id都添加到list集合中，最后将list传到数据库查询Dao层的DishMapper中调用removeByIds(list)方法

（4）Dao层接收到传过来的参数后。把参数取出，放入SQL语句中进行数据库的删除，插入成功后返回Integer类型的值给控制器，如果为1则删除成功，否则将删除失败，最后控制器再将结果组装成R返回给前端视图。

5.1.1.5 停售，启售菜品

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**

（1） 在list.html上，管理员通过点击删除按钮，发送Ajax跨域请求提交菜品id和菜品状态status，调用DishController中的updateStatus()方法。

（2） DishController通过将接收的菜品id和菜品状态status作为参数调用DishService中的updateStatus()方法。

（3） DishService接收到参数后，;就调用split方法根据“，”将各个id分割出来，然后创建一个list集合，将分割出来的id都添加到list集合中，最后将list传到数据库查询Dao层的DishMapper中调用updateStatus(list)方法

（4） Dao层接收到传过来的参数后。把参数取出，放入SQL语句中进行数据库的修改，修改成功后返回Integer类型的值给控制器，如果为1则修改成功，否则将修改失败，最后控制器再将结果组装成R返回给前端视图。

5.1.2 套餐管理模块

5.1.2.1 套餐管理主界面

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。

5.1.2.2 新增套餐信息

管理员可以通过点击“新建菜品”按钮来添加新的套餐。当点击此按钮后，add.html页面会首先发送一个Ajax请求到CategoryController的list方法，以获取套餐分类数据，并将这些数据展示在下拉选择框中供管理员选择。随后，页面会再次发起一个Ajax请求，这次请求同样会调用CategoryController的list方法，但目的是获取菜品分类数据。一旦获取到这些数据，它们将被展示在添加菜品的窗口中，供管理员在创建新套餐时选择和参考。接下来，页面会发起第三次Ajax请求。这次请求会调用DishController中的list方法，并传入之前选择的菜品分类作为参数，以查询对应分类下的菜品数据。查询到的菜品数据随后会被展示在添加菜品窗口中，以便管理员在创建套餐时能够选择具体的菜品。在管理员输入完套餐的相关信息并点击上传图片按钮后，页面会发送一个请求到CommonController中的upload方法。这个方法的目的是处理图片上传逻辑，将管理员选择的图片保存到服务器上。图片上传成功后，管理员可以继续完成套餐的创建流程。图片保存之后，页面会在发一个请求调用CommonController中的download方法对图片进行下载并回显。最后管理员点击保存按钮，页面发送一个Ajax请求，请求调用setmealController中的save方法保存数据。

5.1.2.3 修改套餐信息

管理员可以点击套餐管理主页面的右边操作栏编辑按钮修改套餐信息。

（1）管理员点击修改套餐，页面会将类型type封装为Category对象，然后将Category对象作为参数，发送一个Ajax请求，请求调用CategoryController中的list方法将菜品分类显示到下拉框中。

（2）页面将菜品id作为参数，发送Ajax请求，请求调用SetmealController中的get()方法。

（3）SetmealController通过@PathVariable注解解析拿到套餐id，将套餐id作为参数调用SetmealService中的getById (id) 方法传到数据库查询的Dao层的SetmealMapper中调用selectById()方法进行数据库的操作来获取菜品基本信息。然后将获取到菜品基本信息封装到Setmeal对象中，然后将dish对象中的信息拷贝到SetmealDto对象中，然后调用SetmealFlavorService中的list()方法，根据套餐id查询当前套餐所对应的口味所有菜品数据，然后将菜品信息设置在SetmealDto对象中，最后返回SetmealDto对象。

（4）Dao层接收到传过来的参数后将用户id取出，放入SQL语句中进行数据库的查询，返回对应的菜品信息到add.html中回显。

（5）界面还会发送一个请求，请求调用CommonController中的download方法进行图片的回显。

（6）所有信息回显之后，管理员可以输入要修改的信息，输入完毕之后点击保存按钮，之后就会页面就会将菜品信息数据封装为json对象，并发送Ajax请求，请求调用SetmealController中的update方法。

（7）SetmealController通过@RequestBody注解将接收到的json对象转为java对象，然后调用EmployeeService中的updateWithFlavor(obj)方法。

（8）SetmealService接收到Java对象的参数后，首先将参数传到数据库查询Dao层的SetmealMapper中调用updateById(obj)方法进行数据库的操作。然后调用SetmealFlavorService中的remove()方法清理当前套餐的所有菜品信息。然后再调用SetmealFlavorService中的insertBatch()方法添加当前提交过来的口味数据。

（9）Dao层接收到传过来的参数后。把Java对象中的字段取出，放入SQL语句中进行数据库的修改，修改成功后返回Integer类型的值给控制器，如果为1则修改成功，否则将修改失败，最后控制器再将结果组装成R返回给前端视图。

5.1.2.4 删除套餐信息

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。

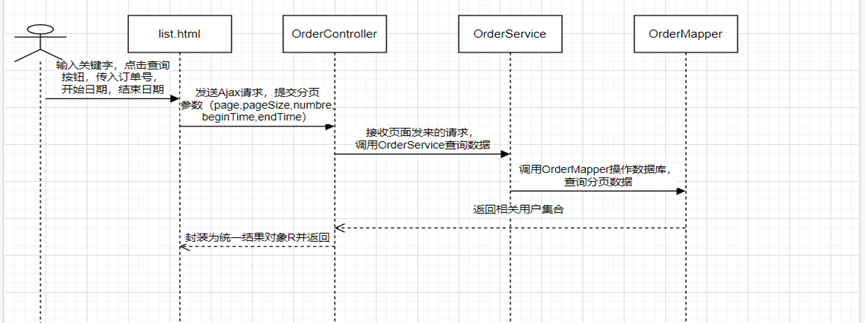
5.1.2.5 停售，启售套餐

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

5.1.3.1 订单管理主界面

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 订单模糊查询顺序图**

（1）在list.html页面上，管理员通过点击输入框输入订单号，开始日期，结束日期，再点击查询按钮，就会触发点击事件返送Ajax请求，将关键字number，开始日期beginTime，结束日期endTime，当前页数page以及每页的信息条数pageSize作为参数传入，调用OrederServlet中的getOrders方法。

（2）OrderServlet拿到参数之后，首先将page，pageSize作为参数构造分页构造器，然后构造一个条件构造器queryWrapper，然后分别判断number，beginTime，endTime是否为空，如果不为空，分别将number，beginTime，endTime作为queryWrapper.like，queryWrapper.ge， queryWrapper.eq的参数添加过滤条件，最后将pageInfo，queryWapper作为参数调用OrderService的page方法。

（3）OrderService接收到参数之后，调用page方法将name，beginTime，endTime拼接成数据库模糊查询所要求的字段格式最后传到数据库查询Dao层的OrderMapper中进行数据库的操作。

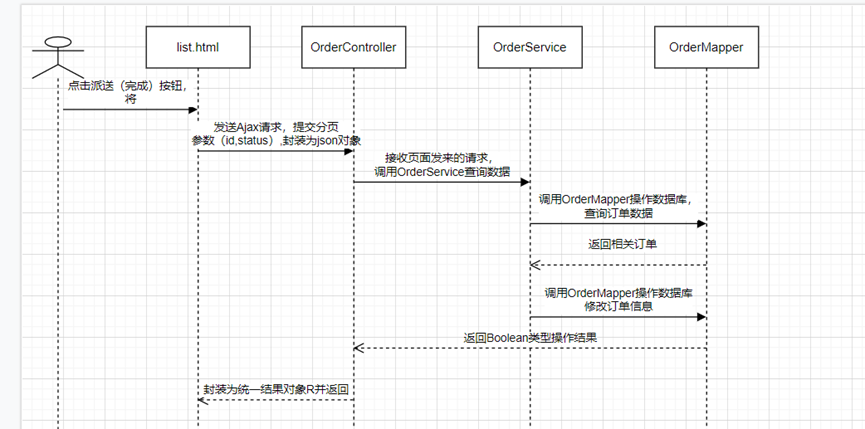
（4）Dao层接收到传过来的参数后。把参数放入SQL语句中进行数据库的查询操作，然后将查询结果封装为商品信息集合返回给控制器，最后组装成返回结果类R返回给前端视图。

5.1.3.2 查看订单信息

管理员点击订单右边操作栏中的查看按钮，可以查看订单详细信息。管理员点击订单右边操作栏中的查看按钮，前端会将订单id发送给后端OrderController，OrderController根据映射找到”/details/{id}”对应的方法getOrder()。getOrder方法获取到id信息后调用orderService.getDetailById(id)方法，交由orderService处理.orderService调用Dao层获取到对应id的订单信息返回到给前端显示。

5.1.3.3 派送/完成订单

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**

（1）在list.html上，管理员通过点击派送（完成）按钮将员工id以及员工状态status封装为json对象向Ajax发送跨域请求，调用OrderController中的updatestatus()方法。

（2）OrderController接收页面发送过来的参数并封装为Map对象，然后将Map对象作为参数调用OrderService中的updateStatus()方法。

（3）OrderService接收到Map对象的参数后，传到数据库查询Dao层的OrderMapper中调用getById(id)方法进行数据库的操作来获取订单的相关信息封装在orders对象中，然后给该对象的status设置为页面穿过来的status的值，然后传到数据库查询Dao层的OrderMapper中调用updateById(id)方法进行数据库的操作

（4）Dao层接收到传过来的参数后。把Java对象中的字段取出，放入SQL语句中进行数据库的修改，修改成功后返回Integer类型的值给控制器，如果为1则修改成功，否则将修改失败，最后控制器再将结果组装成R返回给前端视图。

5.1.2 移动端业务开发

5.1.2.1 添加购物车数据

移动端用户能够便捷地将菜品或套餐加入购物车。对于菜品，若设置了口味信息，用户需要先选择相应的规格后，方可将其加入购物车。而对于套餐，用户则可以直接点击“+”按钮，将当前选中的套餐快速添加到购物车中。

在购物车界面，用户可以轻松修改已选菜品和套餐的数量，以满足自己的需求。同时，用户还具备清空购物车的功能，以便在需要时快速清理购物车内的所有内容。这些设计旨在提升用户的购物体验，使其能够更加方便地管理和调整自己的订单。

其实添加菜品套餐到购物车中与之前的添加功能模块都相似，所以就不再详细说明，在这里描述一下开发过程吧。

（1）当我们点击“加入购物车”或者“+”按钮时，页面会发送ajax请求，请求Controller服务端，Controller层首先设置用户id，指定当前哪一个用户的购物车数据。并且获取当前购物车中的菜品id，套餐id。

（2）之后调用ShoppingCartService层的getOne方法查询当前菜品或套餐是否在购物车中，进行判断，如果存在，number+1，Service层调用updateById更新购物车中的菜品数据；如果不存在就添加到购物车中，默认设置number=1，然后调用Service层的save方法保存数据到购物车字符串中。

（3）返回R类型，菜品套餐成功购入购物车。

5.1.2.2 减少或清空购物车数据

（1）减少购物车数据

当移动端用户将菜品或者套餐添加到购物车，发现有些菜品又突然不是很想要了，或者觉得物低所值，那我们就可以通过购物车来减少其中的数据。在购物车中可以修改菜品和套餐的数量。这里与上述增加数据基本一致，不同的有几点。第一点：当查看购物车发现菜品或套餐的数量number=1时，我们想要减少数据时，则直接调用ShoppingCartService层的remove方法，清空该数据；如果该数据number>1，则用户每点击“.”按钮一次，number.1，之后Service层调用updateById更新购物车中的菜品数据，最终返回R类型，更新购物车中数据。

（2）清空购物车数据

当我们点进购物车后，右上角有一个“清空”，当我们点击“清空”后，页面发送请求，服务端接收Json数据，然后调用ShoppingCartService层的remove方法，直接将购物车清空，返回R类型，成功执行。

5.1.2.3 新建地址

地址簿，指的是移动端消费者用户的地址信息，用户登录成功后可以维护自己的地址信息。同一个用户可以有多个地址信息，但是只能有一个默认地址。

新建地址尤为简单，简要说一下。首先从session种拿取当前userId，将其赋给当前地址用户Id，输入完信息，封装成Json字符串，发送ajax请求给服务端，然后服务端调用AddressBookService的save方法，保存当前地址信息，返回R类型。

用户点击“地址管理”后，前端发送ajax请求，后端接收请求后查询该用户的地址信息，首先取出用户id，构造一个查询条件构造器，将用户id作为条件查询，并用修改时间作为排序条件，查询到数据后返回给前端页面。

用户点击“添加收货地址”，跳转到输入信息页面，提交后发送请求，后端接收到请求，处理数据将信息写入address\_book表中。

点击书写按钮，跳转到编辑地址页面，页面发送ajax请求，后端接收到请求后查询到该地址信息并进行回显，点击“保存地址”按钮，页面发送ajax请求，将数据以json的形式提交给后端，后端接收到数据并进行处理后响应给页面，点击“删除地址”按钮，页面发送ajax请求，将id作为参数提价给后端，后端接收到数据并进行处理后响应给页面。

5.1.2.4 设置默认地址

首先是获取当前CurrentId与AddressBook的UserId比较相等，之后设置AddressBook的默认条件getIsDefault为0，调用AddressBookService的update方法构造更新条件。更新后设置AddressBook的IsDefault为1，之后调用AddressBookService的updateById方法更新当前地址为默认地址，返回R类型。

5.1.2.5 下单

移动端用户将菜品或者套餐加入购物车后，就可以点击购物车中的“去结算”按钮，当我们点击结算后就会出现下图，而点击“去支付”按钮后就完成了下单的操作。

其实下单的代码逻辑相对会复杂一些，因为涉及到多个不同的订单类之间的转换，但主要从步骤来看就还算比较清晰的。主要步骤包括验证传入的订单数据是否完整和可行，比如从用户提交的订单数据中提取地址簿ID，查询并验证地址信息是否有效；以及根据用户ID查询购物车列表，确保购物车不为空。第二步就是创建订单对象，设置相关属性（如订单时间、状态、用户信息等），然后保存到数据库中。第三步，遍历购物车中的商品，为每个商品创建订单明细对象，设置相关属性（如订单ID、商品信息等），并批量保存到数据库中。第四步，清空用户的购物车。最后将创建的订单信息封装成返回对象，并返回给调用者。

5.1.2.6 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。

系统实现

结论与展望

在本次研究中，我成功开发了一个基于JavaEE技术的餐饮管理与服务系统，该系统采用Spring Boot框架和Vue.js前端框架，实现了餐品信息的电子化管理。管理系统的主要功能包括员工管理、分类管理、菜品管理、套餐管理、订单管理，全面覆盖了后台管理的各个环节。实际测试表明，系统运行稳定，功能完善，显著提升了后台管理的效率和鸿蒙用户使用的体验。系统的实施，不仅减轻了管理人员的工作负担，也为鸿蒙端用户提供了便捷的点餐和订单查询服务，具有重要的应用价值和广阔的应用前景。

系统的技术实现与创新体现在采用了面向对象的设计方法，结合了MyBatis持久层与MySQL数据库的交互，实现了数据的高效管理和操作。开发过程中遵循了模块化和组件化的原则，提高了代码的可维护性和可扩展性。在性能方面，系统在关键性能指标上均达到了预期目标，如系统启动迅速，首页加载和登录响应时间均在用户可接受的范围内，下单操作的响应时间迅速，保证了用户在进行点餐行为时的流畅体验。鸿蒙用户体验方面，系统界面友好，操作简便，用户可以轻松地进行个人信息的查看和编辑，以及健康档案的管理和查询。系统还提供了个性化的健康建议和健康知识阅读，增强了用户的参与感和满意度。

尽管本系统在功能和性能上均表现良好，但仍有进一步提升和完善的空间。未来可以考虑增加更多的个性化服务，如健康风险评估、疾病预防建议等，以提供更全面的健康管理服务。随着用户数量的增加和数据量的增长，系统的性能可能会面临新的挑战，因此，持续的性能优化和资源管理将是未来工作的重点。在数据安全和隐私保护方面，系统需要进一步加强，可以考虑引入更先进的加密技术和访问控制机制，确保用户数据的安全。此外，建立一个有效的用户反馈机制，及时收集用户的意见和建议，不断优化系统功能和用户体验，也是未来工作的重要方向。

随着新技术的不断涌现，如人工智能和大数据分析，未来的研究可以考虑将这些技术应用于健康管理系统中，提高系统的智能化水平和处理能力[16]。系统未来可以考虑在其他高校或医疗机构中推广应用，以实现更广泛的社会价值。通过不断的努力和创新，大学生健康管理系统有望成为健康管理领域的佼佼者，为用户提供更高质量的服务。同时，本研究的成果也为其他领域的档案电子化管理提供了参考和借鉴。

参考文献

1. 庄彦,未培.基于云数据库的大学生健康档案管理系统的设计与实现[J].辽宁科技学院学报,2016,18(01):17-19.
2. 陈淦涛.AI时代人工智能在健康教育领域的应用探索与思考[C]//广州市卫生健康宣传教育中心.广州市第十四届健康教育与健康促进学术交流活动稿集.广州市黄埔区南岗街社区卫生服务中心;,2024:2.DOI:10.26914/c.cnkihy.2024.014211.
3. Maya S ,Raymond C .Frontend Development Projects with Vue.js 3:Learn the fundamentals of building scalable web applications and dynamic user interfaces with Vue.js[M].Packt Publishing Limited:2023-03-17.DOI:10.0000/9781803236315.
4. Liu Y .Campus Second-Hand Textbook Trading Platform based on Vue 3 and Spring Boot[J].International Core Journal of Engineering,2024,10(6):
5. 罗辉琼,贺超波,李姣姣,等.基于SpringMVC及MyBatis的高校学术成果库设计与实现[J].现代信息科技,2024,8(04):25-29+35.DOI:10.19850/j.cnki.2096-4706.2024.04.006.
6. 滕明星,张恒,李超.ECharts多端图表引擎在变形监测数据可视化中的应用[J].北京测绘,2024,38(03):318-324.DOI:10.19580/j.cnki.1007-3000.2024.03.008.
7. 李大伟.基于人工智能的计算机应用软件开发技术分析[J].软件,2023,44(09):124-126.
8. 马晓光,封桂荣,孙睿,等.自主可控软件性能需求与关键技术研究[J].工业控制计算机,2022,35(03):1-3.
9. 孔文秀,慈航,李京,等.基于MVC架构的枣庄市重点企业防雷减灾平台[J].山西电子技术,2024,(02):91-93.
10. 柯妍,孙佳留,朱士飞,等.基于MySQL的煤质信息数据库设计[J].资源信息与工程,2024,39(03):117-121.DOI:10.19534/j.cnki.zyxxygc.2024.03.021.
11. 马瑀浓,刘滢,孟丹.大学生健康管理系统的设计与实现[J].数码世界,2019,(06):96.
12. 邵媛.大学生体质健康促进模型的构建研究[J].成都体育学院学报,2024,50(04):124-131.DOI:10.15942/j.jcsu.2024.04.015.
13. 刘慧玲,谭定英,陈平平.基于SpringBoot和Vue.js的大学生团队管理系统的设计[J].电脑编程技巧与维护,2024,(03):120-122.DOI:10.16184/j.cnki.comprg.2024.03.039.
14. ]张耀文.软件测试对于信息技术发展的影响[J].大众标准化,2024,(01):164-166.
15. ]Mehmood N M ,Jürgen B ,Kai P .Checklists to support decision-making in regression testing[J].The Journal of Systems & Software,2023,202
16. 俞成功.基于大数据分析的医院病案质量自动核查系统[J].自动化技术与应用,2024,43(07):184-188.DOI:10.20033/j.1003-7241.(2024)07-0184-05.

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。

此外，我还要特别感谢我的组员们。在共同的项目开发过程中，你们的智慧、协作和不懈努力，是我们团队能够成功的关键。每一个讨论、每一次头脑风暴、每一个共同工作到深夜的时刻，都是我宝贵的记忆。我们之间的相互支持和团队精神，是完成这项工作不可或缺的部分。

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。