南京邮电大学

毕 业 论 文

(成人高等教育函授)

|  |  |
| --- | --- |
| 题 目 | 基于java的校园点餐系统设计与实现 |
| 专 业 | 计算机科学与技术 |
| 学生学号 | 23321435358 |
| 学生姓名 | 余浩 |
| 指导教师 | 徐璐 |
| 函授站 | 江苏社科 |

完成日期： 年 月 日

**学术诚信声明**

本人郑重声明：所提交的论文是本人的研究工作和取得的研究成果。本论文中除引文外，所有实验、数据和有关材料均是真实的。本论文中除引文和致谢的内容外，不包含其他人或其它机构已经发表或撰写过的研究成果。其他同志对本研究所做的贡献均已在论文中作了声明并表示了谢意。

声明人签名：

日 期：

**南京邮电大学继续教育学院高等函授**

**毕业论文（设计）任务书**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 题目 | 基于java的校园点餐系统设计与实现 | | | | | | | |
| 姓名 | 余浩 | 学号 | 23321435358 | | 专业 | 计算机科学与技术 | 函授站 | 江苏社科 |
| 联系  电话 | 13776577508 | 电子邮箱 | yuhaoscuchem@163.com | | | | | |
| 指导教师姓名 | 徐璐 | 职称 | 副教授 | | 工作  单位 | 江苏嘉华职业技术培训学校 | | |
| 设计内容与要求（选题要结合学生所学专业。要求写明本设计所涉及的分析方法或技术手段等；明确文章的章节架构，要求有独立见解，设计内容要详细。）  **1. 项目背景**  随着校园生活节奏的加快，传统的线下点餐方式逐渐无法满足学生的需求。校园点餐系统的开发不仅能够提高学生点餐的效率，优化餐厅的运营管理，还能提升学生的用餐体验。  **2. 技术手段**  前端技术：基于Vue.js的前端框架，构建响应式界面，提升用户体验。  后端技术：Java编程语言，Spring Boot框架进行服务端开发，利用Spring Security进行用户身份认证与权限管理。  数据库设计：使用MySQL进行数据库设计，确保数据的高效存储与访问，设计合理的表结构及索引。  缓存技术：使用Redis做缓存，减少对数据库的直接访问，进一步提高系统查询的性能。  支付接口：使用系统内的积分支付模拟实现支付功能。  API设计：采用RESTful API风格进行前后端数据交互，保证系统的可扩展性与可维护性。  安全性设计：对用户敏感信息进行加密存储，防止SQL注入、XSS等常见攻击，确保系统的安全性。  **3. 功能模块**  用户管理功能：学生用户可以注册、登录、修改个人信息，并通过个人账户查询历史订单、支付情况等。  菜单管理功能：管理员可对餐厅菜单进行增删改查，实时更新菜品信息，展示每日推荐和优惠活动。  点餐与支付功能：学生可以浏览菜品，选择所需菜品并完成点餐，系统支持积分支付。  订单管理功能：系统提供订单查询、订单修改、订单取消等功能，支持按时间、状态等多维度查询。  配送管理功能：根据订单信息，系统自动安排餐品制作与配送，学生可实时查看配送状态。  统计与报表功能：管理员可以查看销售数据、菜品热度、用户偏好等报表数据，为后期的运营决策提供支持。  **4. 论文结构**   1. 绪论 2. 开发技术介绍 3. 系统分析 4. 系统设计 5. 系统实现 6. 系统测试 | | | | | | | | |
| 主要参考文献(按照论文模板参考文献格式)   1. 钟育伙.基于SpringBoot+Vue的校园活动管理系统设计与实现[J].电子技术,2024,53(10):56-57. 2. 朴明,邱翠花,苗子.基于SpringBoot+小程序的信息采集系统设计与实现[J].电子技术,2024,53(10):47-49. 3. 殷志强,钟卫东.基于SpringBoot应用中的XSS攻击防御技术[J].工业控制计算机,2024,37(11):62-64. 4. 戴亚哲,李尤,赵利宏,等.基于SpringBoot+Vue的文旅平台设计与研究[J].无线互联科技,2024,21(21):70-72. 5. 乔凤权,李精伟,徐进,等.基于VUE的泵站运维系统的前端设计与实现[J].科技创新与生产力,2024,45(11):129-132. 6. 李晓薇.vue.js前端应用技术分析[J].网络安全技术与应用,2022,(04):44-45. 7. 罗光武,陈典灿,吴荷,等.应用Springboot+Vue框架的时间管理软件的设计与实现[J].工业控制计算机,2024,37(04):64-66. 8. 徐少军,李宗哲,梅杰,等.基于Springboot+Vue框架的质量检验监督管理系统研发[J].纺织标准与质量,2024,(01):11-14+21. 9. 昝国宁,王雨晴,刘娇龙,等.MySQL数据库自动化运维管理系统的设计与实现[J].铁路计算机应用,2024,33(09):39-43. 10. 庞敏.MySQL数据库的数据安全应用设计技术研究[J].数字通信世界,2024,(09):25-27. 11. 吴敏.基于Java Web的班级信息管理系统的设计与实现[J].中国新通信,2024,26(07):74-76. 12. 匡少华,丁昊,赵正平.基于Javaweb的大学生食堂外卖平台的设计与实现[J].信息与电脑(理论版),2020,32(01):91-94. 13. Liu Y .Design and Implementation of a Student Attendance Management System based on Springboot and Vue Technology[J].Frontiers in Computing and Intelligent Systems,2024,8(1):91-97. 14. Hejing W .Commerce Middle Office Management System Based on Springboot[J].International Journal of Advanced Network, Monitoring and Controls,2022,7(2):32-45. 15. Mario V ,Sanja K .DEVELOPMENT OF A LIBRARY APPLICATION USING VUE.JS AND FIREBASE TECHNOLOGIES[J].Polytechnic and design,2023,11(1):66-73. | | | | | | | | |
| 函授站（盖章） | | | | 论文指导教师审核意见  同意开题  论文指导教师（签名）徐璐 | | | | |
| 论文督导教师审核意见  论文督导教师（签名） | | | | | | | | |

备注：此表一式三份，学生留存一份，指导教师留存一份，督导教师留存一份。

年 月 日

**南京邮电大学继续教育学院高等函授**

**毕业论文（设计）督导教师论文中期检查表**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 姓名 | 余浩 | 学号 | 23321435358 | 专业 | 计算机科学与技术 | 函授站 | 江苏社科 |
| 题目 | 基于java的校园点餐系统设计与实现 | | | | | | |
| 毕业 论文 （设计） 任务书和工作计划 执行 情况 | 按时完成了相关文献的查阅，收集了国内外关于校园订餐系统和软件开发流程的资料，了解了当前研究现状和技术趋势。完成任务书的撰写，明确了研究目标、方法和技术路线，通过了导师的审核。  上网查询资料，整理分析了校园订餐系统的功能需求。完成需求规格说明书的编写，对系统的功能模块、业务流程、性能需求等进行了详细描述，为后续设计开发提供了准确依据。  基于需求分析结果，完成了系统架构设计，采用了前后端分离的架构模式，确定了前端使用 Vue.js 框架，后端使用 Spring Boot 框架。完成数据库设计，创建了用户表、菜单表、订单表等数据库表结构，设计了表之间的关联关系。同时完成了接口设计，定义了各功能模块的接口规范和数据交互格式。整体按计划完成了系统设计任务。  在开发过程中，遇到了前端页面布局兼容性问题和后端数据库连接不稳定的情况。通过查阅资料、，采用了 CSS 预处理器和适配方案解决了前端问题，调整了数据库连接配置和优化了服务器环境解决了后端问题。 | | | | | | |
| 毕业 论文 （设计） 修改 意见 |  | | | | | | |

督导教师签名： 年 月 日

此表一式二份，督导教师留存一份；学员一份作论文修改的依据，并附在所交论文指导教师评语表的后面。

摘 要

随着信息技术的发展，校园餐饮管理逐渐向数字化、智能化方向发展。传统的校园点餐模式存在排队时间长、信息不对称、管理效率低等问题。本论文设计并实现了一款基于Java技术的校园点餐系统，旨在提高校园餐饮管理的效率和用户体验。系统采用Java语言开发，结合Spring Boot框架，Redis缓存数据库和MySQL数据库进行后端开发，前端使用Vue.js技术实现页面交互。系统主要功能包括用户注册与登录、餐品浏览与点餐、订单管理、支付功能以及管理员的餐品管理与数据统计等。通过本系统，学生能够方便快捷地进行在线点餐，减少了排队时间；管理员能够实时查看订单情况，优化餐品管理。本论文通过详细分析系统需求、设计架构及实现过程，验证了该系统的可行性与有效性，为未来校园餐饮信息化建设提供了有益的参考。

关键词： Java；校园点餐系统；Spring Boot；Vue.js；MySQL；Redis

目 录

[第一章 绪论 1](#_Toc188222022)

[1.1 研究背景 1](#_Toc188222023)

[1.2 研究目的 1](#_Toc188222024)

[1.3 研究意义 1](#_Toc188222025)

[1.3.1 理论意义 1](#_Toc188222026)

[1.3.2 实践意义 2](#_Toc188222027)

[1.4 国内外研究现状 2](#_Toc188222028)

[1.4.1 国外研究现状 2](#_Toc188222029)

[1.4.2 国内研究现状 2](#_Toc188222030)

[1.5 研究方法与内容 2](#_Toc188222031)

[1.5.1 研究方法 2](#_Toc188222032)

[1.5.2 研究内容 3](#_Toc188222033)

[第二章 开发技术介绍 4](#_Toc188222034)

[2.1 Spring Boot介绍 4](#_Toc188222035)

[2.2 Vue.js介绍 4](#_Toc188222036)

[2.3 MySQL介绍 4](#_Toc188222037)

[2.4 Redis介绍 5](#_Toc188222038)

[第三章 系统需求分析 6](#_Toc188222039)

[3.1 可行性分析 6](#_Toc188222040)

[3.1.1 技术可行性 6](#_Toc188222041)

[3.1.2 经济可行性 6](#_Toc188222042)

[3.1.3 操作可行性 6](#_Toc188222043)

[3.2 需求分析 6](#_Toc188222044)

[3.2.1 用户需求 6](#_Toc188222045)

[3.2.2 功能需求 7](#_Toc188222046)

[3.2.3 性能需求 7](#_Toc188222047)

[3.2.4 安全需求 8](#_Toc188222048)

[第四章 系统设计 9](#_Toc188222049)

[4.1 系统总体架构设计 9](#_Toc188222050)

[4.2 功能流程设计 9](#_Toc188222051)

[4.2.1 系统开发流程 9](#_Toc188222052)

[4.2.2 登录功能流程 10](#_Toc188222053)

[4.2.3 系统操作流程 11](#_Toc188222054)

[4.2.4 添加数据流程 12](#_Toc188222055)

[4.2.5 修改数据流程 13](#_Toc188222056)

[4.2.6 删除数据流程 13](#_Toc188222057)

[4.3 数据库设计 14](#_Toc188222058)

[4.3.1 数据库概念设计（E-R模型） 14](#_Toc188222059)

[4.3.2 数据库逻辑设计（表结构设计） 19](#_Toc188222060)

[第五章 系统实现 24](#_Toc188222061)

[5.1 系统环境搭建 24](#_Toc188222062)

[5.2 用户功能实现 24](#_Toc188222063)

[5.2.1 登录注册 24](#_Toc188222064)

[5.2.2 菜单首页 24](#_Toc188222065)

[5.2.1 我的点单与我的订单 25](#_Toc188222066)

[5.2.2 我的收藏、我的评论与我的信息 25](#_Toc188222067)

[5.2.3 个人积分 25](#_Toc188222068)

[5.3 食堂管理员功能实现 25](#_Toc188222069)

[5.3.1 用户登录 25](#_Toc188222070)

[5.3.2 菜单管理 25](#_Toc188222071)

[5.3.3 配送管理 26](#_Toc188222072)

[5.3.4 评价管理 26](#_Toc188222073)

[5.3.5 积分管理 26](#_Toc188222074)

[5.3.6 数据统计 26](#_Toc188222075)

[5.3.7 点赞管理 26](#_Toc188222076)

[5.3.8 用户管理 26](#_Toc188222077)

[第六章 系统测试 28](#_Toc188222078)

[6.1 测试目的 28](#_Toc188222079)

[6.2 测试内容 28](#_Toc188222080)

[6.3 测试方法 36](#_Toc188222081)

[6.4 测试结论 36](#_Toc188222082)

[第七章 总结与展望 37](#_Toc188222083)

[7.1 研究总结 37](#_Toc188222084)

[7.2 研究不足与展望 37](#_Toc188222085)

[致 谢 39](#_Toc188222086)

[参考文献 40](#_Toc188222087)

# 绪论

## 研究背景

在当今数字化时代，校园生活也在不断地向智能化、便捷化方向发展[1]。随着高校规模的不断扩大，学生数量日益增加，传统的校园餐饮服务模式逐渐暴露出诸多问题 。以往学生在食堂就餐，常常面临高峰时段排队时间过长、菜品选择不够直观、信息沟通不及时等困扰，这些问题不仅影响了学生的就餐体验，也在一定程度上降低了食堂的运营效率。与此同时，互联网技术和移动应用的飞速发展为解决这些问题提供了新的思路和方法。越来越多的行业开始借助信息技术实现业务的转型升级，餐饮行业也不例外。在校园环境中，开发一个便捷、高效的点餐系统成为了满足师生就餐需求、提升校园服务质量的重要举措。

## 研究目的

本课题旨在设计并实现一个基于Java的校园点餐系统，通过该系统为师生提供更加便捷、高效的点餐服务。具体而言，系统需实现以下目标：

提升点餐便捷性：打破传统点餐受时间和空间限制的模式，让师生无论在教室、宿舍还是校园内其他场所，都能通过手机或电脑轻松完成点餐操作，节省排队等待时间。

优化食堂运营：帮助食堂工作人员更准确地掌握菜品预订情况，合理安排食材采购和菜品制作，减少食材浪费，提高运营效率。同时，通过系统的数据分析功能，为食堂调整菜品、优化服务提供数据支持。

增强用户体验：提供丰富的菜品展示、详细的菜品信息以及个性化的推荐功能，满足不同师生的口味需求。此外，还设置便捷的支付方式和完善的评价反馈机制，增强用户与食堂之间的互动。

## 研究意义

### 理论意义

丰富软件工程实践案例：本系统的开发涉及到Java语言、数据库技术、软件工程等多方面的知识和技术，通过实际项目的设计与实现，能够为相关领域的研究提供具体的实践案例，有助于进一步完善和丰富软件工程的理论与方法。

推动技术融合研究：在系统开发过程中，需要将前端开发技术、后端开发技术以及数据库管理技术等进行有机融合，探索不同技术之间的协同工作方式，为跨技术领域的研究提供一定的参考。

### 实践意义

改善校园餐饮服务质量：直接为师生提供便利的点餐服务，提升校园生活的满意度和幸福感。同时，优化后的食堂运营模式能够提高服务效率，减少资源浪费，有助于营造更加优质的校园环境。

为其他校园应用开发提供借鉴：本校园点餐系统的设计思路和实现方法可以为其他校园信息化应用的开发提供参考和借鉴，推动校园信息化建设的进一步发展。

## 国内外研究现状

### 国外研究现状

国外在餐饮信息化领域起步较早，相关技术和应用较为成熟。许多高校和餐饮企业已经广泛采用先进的点餐系统，如美国的一些高校食堂引入了智能化的自助点餐终端和移动点餐应用，学生可以通过手机应用提前预订餐食，到食堂后直接取餐，大大提高了就餐效率。同时，国外的点餐系统注重与其他校园管理系统的集成，如与学生账户系统、校园卡系统等无缝对接，实现了便捷的支付和身份验证功能。此外，在菜品推荐方面，一些系统利用大数据分析和人工智能算法，根据学生的历史点餐记录和偏好，为学生提供个性化的菜品推荐，提升了用户体验。

### 国内研究现状

近年来，随着国内高校信息化建设的快速推进，校园点餐系统也得到了广泛的关注和应用。国内的校园点餐系统在功能上不断完善，除了基本的点餐、支付功能外，还增加了菜品评价、营养分析、预订提醒等功能。同时，一些高校结合自身的特色和需求，开发了具有针对性的点餐系统，如部分农业院校的点餐系统突出了本地特色农产品在菜品中的应用，强调了菜品的绿色、健康理念。在技术实现方面，国内的校园点餐系统多采用B/S架构和移动应用相结合的方式，利用Java、Python等编程语言进行开发，数据库则选用MySQL、Oracle等主流数据库管理系统。

## 研究方法与内容

### 研究方法

文献研究法：通过查阅国内外相关文献资料，了解校园点餐系统的研究现状、发展趋势以及关键技术，为本课题的研究提供理论基础和技术参考。

需求分析法：通过问卷调查、访谈等方式，收集师生和食堂工作人员对校园点餐系统的功能需求和使用体验，明确系统的设计目标和功能模块。

系统设计法：运用软件工程的思想和方法，对校园点餐系统进行总体架构设计、功能模块设计、数据库设计等，确保系统的科学性、合理性和可扩展性。

实验测试法：在系统开发完成后，通过实验测试对系统的功能、性能、稳定性等进行全面检测，及时发现并解决系统中存在的问题，确保系统能够满足实际使用需求。

### 研究内容

系统需求分析：详细分析校园点餐系统的用户需求，包括师生的点餐需求、食堂工作人员的管理需求等，确定系统的功能模块和非功能需求。

系统设计：进行系统的总体架构设计，确定系统的技术选型和开发模式。完成功能模块设计，包括用户管理、菜品管理、点餐管理、订单管理、支付管理、评价管理等模块的设计。同时，进行数据库设计，确定数据库的表结构、字段类型以及表之间的关系。

系统实现：基于Java语言和相关开发框架，实现校园点餐系统的各个功能模块。完成前端界面的开发，确保界面友好、操作便捷。实现后端业务逻辑的处理和数据库的交互，保证系统的稳定性和高效性。

系统测试：制定系统测试计划，对系统进行功能测试、性能测试、兼容性测试等，确保系统满足设计要求和用户需求，能够稳定运行。

# 开发技术介绍

## Spring Boot介绍

Spring Boot是构建校园点餐系统后端的核心框架，它基于Spring框架，致力于简化Spring应用的初始搭建和开发过程。通过提供大量的默认配置，Spring Boot实现了快速开发，减少了样板代码。其内置的Tomcat服务器，让我们可以直接将应用打包成可执行的JAR文件，方便部署。在校园点餐系统中，Spring Boot负责处理HTTP请求，管理业务逻辑组件，以及与数据库交互。例如，在处理用户点餐请求时，Spring Boot的控制器（Controller）接收请求，调用服务层（Service）的业务逻辑进行订单处理，然后通过数据访问层（DAO）将订单数据持久化到数据库。同时，它对各种数据访问技术（如JDBC、MyBatis等）有良好的支持，方便我们根据需求选择合适的方式来操作数据库。

## Vue.js介绍

Vue.js是构建校园点餐系统前端界面的优秀框架。它采用组件化的开发模式，使得代码的可维护性和复用性大大提高。在点餐系统中，我们可以将界面拆分成多个独立的组件，如菜品展示组件、购物车组件、订单详情组件等。每个组件都有自己的模板、脚本和样式，彼此之间相互独立又协同工作。Vue.js的双向数据绑定机制，让数据的更新和视图的渲染变得简洁高效。例如，当用户在购物车中添加或删除菜品时，数据的变化会自动同步到视图上，无需手动操作DOM。同时，Vue.js拥有丰富的插件和生态系统，如Vue Router用于前端路由管理，Vuex用于状态管理，这些都为构建复杂的单页应用（SPA）提供了有力支持。

## MySQL介绍

MySQL是一款广泛使用的关系型数据库管理系统，用于存储校园点餐系统的各类数据。它具有开源、性能稳定、易于使用等特点。在系统中，MySQL数据库存储了用户信息（如用户名、密码、联系方式）、菜品信息（菜品名称、价格、图片、介绍）、订单信息（订单编号、用户ID、菜品ID、下单时间、订单状态）等。通过SQL语句，我们可以方便地进行数据的增、删、改、查操作。例如，使用INSERT语句向订单表中插入新的订单记录，使用SELECT语句查询用户的历史订单。MySQL的事务处理功能确保了数据操作的一致性和完整性，在涉及多个数据表的复杂操作（如创建订单时同时更新菜品库存和用户余额）时非常重要。

## Redis介绍

Redis是一个高性能的非关系型内存数据库，在校园点餐系统中主要用于缓存数据，提高系统的响应速度。由于Redis的数据存储在内存中，读写速度极快。我们可以将一些频繁访问的数据（如热门菜品信息、系统配置信息）缓存到Redis中。当用户请求这些数据时，首先从Redis中获取，如果缓存中没有再查询数据库，这样大大减少了数据库的负载和响应时间。此外，Redis还支持数据的过期策略，我们可以设置缓存数据的有效期，过期后数据自动从缓存中删除，保证数据的时效性。例如，对于一些实时性要求不高的菜品统计信息，我们可以将其缓存到Redis中，并设置较短的过期时间。

# 系统需求分析

## 可行性分析

### 技术可行性

本校园点餐系统基于Java语言开发，Java拥有丰富的类库和成熟的技术框架，如Spring Boot、Spring MVC等，能够为系统开发提供坚实的技术支持。前端采用Vue.js框架，其组件化开发模式和高效的数据绑定机制，可实现良好的用户交互界面。数据库选用MySQL，它是一款稳定且功能强大的关系型数据库管理系统，能满足数据存储和管理的需求。此外，Redis作为缓存数据库，可有效提升系统的响应速度。开发团队成员具备扎实的Java编程基础，熟悉相关框架和技术的使用，在技术层面具备完成系统开发的能力。

### 经济可行性

从开发成本来看，主要涉及开发人员的人力成本以及服务器租赁等费用。开发过程中使用的Java、Vue.js、MySQL、Redis等技术均为开源软件，无需额外支付软件授权费用。服务器方面，可根据系统初期的访问量选择相对经济实惠的云服务器，随着业务发展再进行升级。在运营成本上，系统投入使用后，可提高食堂运营效率，减少人工点餐的工作量，降低人力成本，同时通过优化食材采购，减少浪费，带来一定的经济效益。因此，从经济角度而言，该系统具有可行性。

### 操作可行性

系统的目标用户为校园师生和食堂工作人员。对于师生来说，系统的前端界面设计将遵循简洁、直观的原则，操作流程符合日常使用习惯，易于上手。通过简单的注册、登录流程，即可方便地进行菜品浏览、点餐、支付等操作。对于食堂工作人员，系统的后台管理界面将提供清晰的数据展示和便捷的操作功能，如菜品管理、订单处理等，经过简单培训就能熟练使用。因此，在操作方面，系统能够满足用户需求。

## 需求分析

### 用户需求

学生：希望能够快速便捷地浏览食堂的菜品信息，包括菜品名称、图片、价格、口味、食材等。可根据个人喜好进行菜品搜索和筛选，如按菜系、价格区间筛选。在点餐时，能够方便地将菜品加入购物车，随时修改购物车中的菜品数量或删除菜品。支持多种支付方式，如积分支付、微信支付和支付宝支付等。下单后可实时查看订单状态，包括已支付、待配送、已完成等，还能对菜品和服务进行评价，为其他用户提供参考。

食堂管理人员：需要对菜品进行全面管理，包括菜品的添加、修改、删除操作。可设置菜品的库存、价格、每日推荐等信息。能够实时查看订单信息，包括订单详情、下单时间、用户信息等，并根据订单进行配餐和配送处理。同时，还希望系统能提供销售数据统计功能，方便了解菜品的销售情况，为食材采购和菜品调整提供依据。

### 功能需求

用户管理：实现用户的注册、登录、密码修改、个人信息管理功能。对用户进行身份验证，确保只有合法用户才能使用系统。

菜品管理：提供菜品的添加、编辑、删除功能。支持菜品分类管理，如早餐、午餐、晚餐分类，以及按菜系分类等。能够设置菜品的详细信息，如菜品名称、图片、描述、价格、库存、是否推荐等。

点餐管理：用户可浏览菜品列表，将菜品加入购物车，在购物车中进行菜品数量调整和删除操作。支持一键下单功能，下单时自动生成订单编号，并记录用户信息、点餐菜品及数量、下单时间等。

订单管理：食堂工作人员可查看所有订单信息，包括待处理订单、已处理订单、已完成订单等。对订单进行处理，如确认接单、配餐、标记配送等操作。用户可以查看自己的订单历史记录和订单详情。

支付管理：集成多种支付方式，如系统积分支付接口、微信支付、支付宝支付等。确保支付过程的安全、稳定，支付成功后自动更新订单状态。

评价管理：用户在订单完成后可对菜品口味、质量、服务态度等进行评价，可输入文字评价和打分。食堂工作人员可查看用户评价，以便改进服务和菜品质量。

数据统计分析：统计菜品的销售数据，如销售量、销售额、热门菜品排行等。生成报表，为食堂的食材采购、菜品调整提供数据支持。

### 性能需求

响应时间：系统应具备快速的响应能力，用户在进行菜品浏览、点餐、支付等操作时，页面加载和数据处理时间应控制在较短时间内，一般操作的响应时间不超过3秒，复杂操作（如大数据量的报表生成）不超过10秒，以提供流畅的用户体验。

吞吐量：能够满足校园内高峰时段的并发访问需求，保证在大量用户同时进行点餐、查询等操作时，系统仍能稳定运行，不出现卡顿、崩溃等现象。预计系统能够支持至少500个并发用户访问。

数据准确性：确保系统中数据的准确性，如菜品信息、订单信息、支付信息等。在数据的录入、存储、传输和处理过程中，要保证数据的完整性和一致性，避免数据丢失或错误。

### 安全需求

用户认证与授权：通过用户注册和登录机制，对用户进行身份认证，只有合法用户才能使用系统的功能。同时，根据用户角色（学生、教师、食堂工作人员等）进行授权，不同角色拥有不同的操作权限，如食堂工作人员才能进行菜品管理和订单处理，普通用户只能进行点餐和评价等操作。

数据加密：对于用户的敏感信息，如登录密码、支付密码等，在传输和存储过程中进行加密处理，防止信息泄露。采用SSL/TLS等加密协议保证数据在网络传输过程中的安全。

防止数据篡改：对系统中的数据进行完整性校验，防止数据被恶意篡改。例如，在订单数据处理过程中，采用数字签名等技术确保订单信息的真实性和完整性。

系统防护：部署防火墙、入侵检测系统等安全防护措施，防止外部非法攻击，如SQL注入攻击、XSS攻击等，保障系统的稳定运行和数据安全[3]。

# 系统设计

## 系统总体架构设计

校园点餐系统采用前后端分离的架构模式，这种架构能使前端和后端开发人员专注于各自领域，提升开发效率与系统可维护性。前端负责与用户交互，展示菜品信息、处理点餐操作等，通过Vue.js框架构建用户界面；后端采用Spring Boot框架，处理业务逻辑、与数据库交互等。前后端通过HTTP协议进行通信，数据格式采用JSON。

在系统架构中，还引入了Redis缓存数据库，用于存储频繁访问的数据，如热门菜品信息、用户会话数据等，减少数据库查询压力，提高系统响应速度。MySQL关系型数据库则用于持久化存储系统的核心数据，包括用户信息、菜品信息、订单信息等。



图 四‑1系统结构图

## 功能流程设计

### 系统开发流程

首先需要对系统进行需求分析时，了解订餐、管理菜品与订单等需求，接着是系统设计，确定架构，数据库和接口设计。开发阶段，前端用 Vue.js 搭建界面，后端基于 Spring Boot 处理业务逻辑，前后端通过 API 交互。完成后进入测试阶段，从功能、性能、安全多维度测试，修复问题。最后部署上线，将系统部署到服务器，上线后持续监控与优化，收集用户反馈，不断完善系统功能与性能 ，为用户提供更好的订餐体验。



图 四‑2系统开发流程图

### 登录功能流程

用户进入登录页面，按照页面的提示填写用户名和密码，页面判断填写的用户名密码格式是否正确，如果错误，页面直接提示错误，重新填写用户名和密码，格式正确则请求登录接口，接口返回成功，则成功登录系统，反之，重新输入用户名和密码。



图 四‑3系统登录流程图

### 系统操作流程

用户成功登录进入系统后，页面发起查询接口，后台验证权限，如果无权限直接结束接口调用，若有权限，则进入功能界面进行操作，操作过程中出现错误系统及时对客户进行反馈。



图 四‑4系统操作流程图

### 添加数据流程

用户向系统添加数据的时候，系统前端先校验数据的合法性，通过后向后台发起请求，后台再校验数据的合法性，通过两次校验后数据才能添加数据库。



图 四‑5修改数据流程图

### 修改数据流程

用户选择系统一条数据修改时，系统前端先校验修改后数据的合法性，通过后向后台发起请求，后台再校验数据的合法性，通过两次校验后数据才能添加数据库。



图 四‑6删除数据流程图

### 删除数据流程

用户选择一条或多条数据发起删除时，系统前端提示是否确认删除数据，确认后向后台发起请求，后台再校用户的权限，若无权限接口请求结束，若有则删除对应数据。



## 数据库设计

### 数据库概念设计（E-R模型）

用户实体：包含用户ID、用户名、密码、手机号、邮箱、头像、地址等属性。用户与订单之间存在“下单”关系，一个用户可以有多个订单，一个订单属于一个用户。



图 四‑7用户E-R图

菜品实体：有菜品ID、菜品名称、价格、库存、图片、描述、分类ID等属性。菜品分类与菜品是一对多关系，一个菜品分类下有多个菜品。



图 四‑8菜品E-R图

订单实体：包括订单ID、用户ID、订单时间、订单状态、配送地址、联系电话、总金额等属性。订单与菜品之间通过订单详情表建立多对多关系，一个订单包含多个菜品，一个菜品可被多个订单订购。



图 四‑9订单E-R图

订单详情实体：包括订单ID、用户ID、订单时间、订单状态、配送地址、联系电话、总金额等属性。订单与菜品之间通过订单详情表建立多对多关系，一个订单包含多个菜品，一个菜品可被多个订单订购。



图 四‑10订单详情E-R图

评价实体：含有评价ID、用户ID、菜品ID、订单ID、评价内容、评分、评价时间等属性。一个用户可以对多个菜品进行评价，一个菜品也会收到多个用户的评价。



图 四‑11评价E-R图

配送实体：含有配送ID、用户ID、订单ID、配送地址、配送状态等属性。一个用户只能对一份订单进行一次配送。



图 四‑12配送E-R图

点赞实体：含有点赞ID、用户ID、菜品ID、点赞时间等属性。一个用户可以对多个菜品进行点赞，一个菜品也会收到多个用户的点赞。



图 四‑13点赞E-R图

点赞实体：含有收藏ID、用户ID、菜品ID、收藏时间等属性。一个用户可以对多个菜品进行收藏，一个菜品也会收到多个用户的收藏。



图 四‑14收藏E-R图

浏览记录实体：含有浏览ID、用户ID、菜品ID、浏览时间等属性。一个用户可以对多个菜品进行浏览，一个菜品也会收到多个用户的浏览。



图 四‑15浏览记录E-R图

积分实体：含有积分ID、用户ID、订单ID、浏览时间等属性。一个用户可以从多个订单获取积分，一个订单只能产生一条积分记录。



图 四‑16积分E-R图

### 数据库逻辑设计（表结构设计）

表 四‑1 用户表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段长度 | 是否为空 | 默认值 | 备注 |
| id | VARCHAR | 50 | 否 | 无 | 用户ID |
| student\_id | VARCHAR | 50 | 否 | 无 | 学号 |
| username | VARCHAR | 50 | 否 | 无 | 用户名 |
| password | VARCHAR | 255 | 否 | 无 | 密码 |
| name | VARCHAR | 100 | 否 | 无 | 真实姓名 |
| gender | VARCHAR | 2 | 否 | 无 | 性别 |
| phone | VARCHAR | 20 | 是 | 无 | 联系电话 |
| point | INT | 10 | 否 | 0 | 积分数量 |
| email | VARCHAR | 100 | 否 | 无 | 电子邮件地址 |
| avatar | VARCHAR | 255 | 是 | 无 | 用户头像 |
| role | VARCHAR | 2 | 否 | 1 | 系统角色 |
| reg\_date | VRACHAR | 20 | 是 | 无 | 注册系统日期 |
| last\_login | VARCAHR | 20 | 是 | 无 | 最后登录时间 |
| status | VARCHAR | 2 | 是 | 1 | 账号的状态 |

表 四‑2 菜品表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段长度 | 是否为空 | 默认值 | 备注 |
| id | VARCHAR | 50 | 否 | 无 | 菜品ID |
| name | VARCHAR | 100 | 否 | 无 | 菜品名称 |
| description | TEXT | 无 | 是 | 无 | 菜品描述 |
| price | FLOAT | 10 | 否 | 无 | 菜品价格 |
| category | VARCHAR | 2 | 否 | 无 | 菜品分类 |
| image\_url\_first | VARCHAR | 255 | 是 | 无 | 主图 |
| image\_url\_second | VARCHAR | 255 | 是 | 无 | 第二张图片 |
| image\_url\_third | VARCHAR | 255 | 是 | 无 | 第三张图片 |
| image\_url\_forth | VARCHAR | 255 | 是 | 无 | 第四张图片 |
| available | BOOLEAN | 1 | 是 | TRUE | 菜品是否可用 |
| create\_date | VARCHAR | 20 | 是 | 无 | 菜品创建日期 |
| update\_date | VARCHAR | 20 | 是 | 无 | 菜品更新日期 |

表 四‑3 订单表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段长度 | 是否为空 | 默认值 | 备注 |
| id | VARCHAR | 20 | 否 | 无 | 订单ID |
| user\_id | VARCHAR | 20 | 否 | 无 | 用户ID |
| menu\_id | VARCHAR | 20 | 否 | 无 | 菜品ID |
| quantity | INT | 10 | 否 | 无 | 菜品数量 |
| order\_time | TIMESTAMP | 无 | 是 | CURRENT\_TIMESTAMP | 订单生成时间 |
| total\_price | FLOAT | 10 | 否 | 无 | 订单的总价 |
| status | VARCHAR | 2 | 是 | 1 | 订单状态 |
| pay\_method | VARCHAR | 2 | 否 | 1 | 支付方式 |
| address | VARCHAR | 255 | 是 | 无 | 送餐地址 |
| del\_time | TIMESTAMP | 无 | 是 | 无 | 预计送餐时间 |
| del\_status | VARCHAR | 10 | 是 | 未配送 | 送餐状态 |

表 四‑4 订单详情表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段长度 | 是否为空 | 默认值 | 备注 |
| id | INT | 20 | 否 | 无 | 订单详情ID |
| order\_id | INT | 20 | 否 | 无 | 订单ID |
| menu\_id | INT | 20 | 否 | 无 | 菜品ID |
| quantity | INT | 10 | 否 | 无 | 数量 |
| unit\_price | FLOAT | 10 | 否 | 无 | 单价 |
| sub\_total | FLOAT | 10 | 否 | 无 | 合计金额 |

表 四‑5 评价表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段长度 | 是否为空 | 默认值 | 备注 |
| id | INT | 20 | 否 | 无 | 评价ID，主键 |
| menu\_id | INT | 20 | 否 | 无 | 菜品ID |
| user\_id | INT | 20 | 否 | 无 | 用户ID |
| rating | INT | 5 | 否 | 无 | 评分 |
| comment | TEXT | 无 | 是 | 无 | 评论内容 |
| eval\_time | TIMESTAMP | 无 | 是 | CURRENT\_TIMESTAMP | 评价时间 |

表 四‑6 配送表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段长度 | 是否为空 | 默认值 | 备注 |
| id | INT | 20 | 否 | 无 | 配送ID |
| order\_id | INT | 20 | 否 | 无 | 订单ID |
| person\_id | INT | 20 | 是 | 无 | 配送人员ID |
| start\_time | TIMESTAMP | 无 | 是 | 无 | 配送开始时间 |
| end\_time | TIMESTAMP | 无 | 是 | 无 | 配送结束时间 |
| status | VARCHAR | 2 | 是 | 1 | 配送状态 |
| address | VARCHAR | 255 | 是 | 无 | 实际送餐地址 |

表 四‑7 点赞表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段长度 | 是否为空 | 默认值 | 备注 |
| id | INT | 20 | 否 | 无 | 点赞ID |
| user\_id | INT | 20 | 否 | 无 | 用户ID |
| menu\_id | INT | 20 | 否 | 无 | 菜品ID |
| liked\_time | TIMESTAMP | 无 | 是 | CURRENT\_TIMESTAMP | 点赞时间 |

表 四‑8 收藏表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段长度 | 是否为空 | 默认值 | 备注 |
| id | INT | 20 | 否 | 无 | 收藏ID |
| user\_id | INT | 20 | 否 | 无 | 关联用户ID |
| menu\_id | INT | 20 | 否 | 无 | 菜品ID |
| favorite\_time | TIMESTAMP | 无 | 是 | CURRENT\_TIMESTAMP | 收藏时间 |

表 四‑9 浏览记录表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段长度 | 是否为空 | 默认值 | 备注 |
| id | INT | 20 | 否 | 无 | 浏览记录ID |
| user\_id | INT | 20 | 否 | 无 | 用户ID |
| menu\_id | INT | 20 | 否 | 无 | 菜品ID |
| browse\_time | TIMESTAMP | 无 | 是 | CURRENT\_TIMESTAMP | 浏览时间 |

表 四‑10 积分表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名称** | **数据类型** | **字段长度** | **是否为空** | **默认值** | **备注** |
| id | INT | 20 | 否 | 无 | 积分ID |
| user\_id | INT | 20 | 否 | 无 | 用户ID |
| amount | INT | 20 | 否 | 无 | 积分数量 |
| type | VARCHAR | 2 | 否 | 1 | 积分交易类型 |
| time | TIMESTAMP | 无 | 是 | CURRENT\_TIMESTAMP | 积分交易时间 |
| order\_id | INT | 20 | 是 | 无 | 订单ID |

# 系统实现

## 系统环境搭建

系统环境搭建配置如下表所示：

表 五‑1开发环境配置表

|  |  |
| --- | --- |
| 分类 | 详情 |
| 前端开发工具 | VS Code（代码编辑）、Chrome（调试） |
| 前端技术框架 | Vue.js（构建用户界面）、Element UI（组件库） |
| 前端依赖管理 | Node.js、npm，Vue CLI（搭建项目） |
| 后端开发工具 | IntelliJ IDEA |
| 后端技术框架 | Java、Spring Boot |
| 数据库 | MySQL（主数据库，存储持久化数据）、Navicat（数据库管理工具） |
| 缓存 | Redis（利用高速读写与缓存特性，缓存热门菜品数据、用户登录信息等），在 Spring Boot 项目中通过相关依赖配置使用 |
| 部署 | Tomcat 等服务器运行后端服务 |

## 用户功能实现

### 登录注册

采用Spring Security框架结合JWT（JSON Web Token）实现用户身份验证。用户在前端输入账号密码后，请求发送至后端。后端通过查询用户表验证账号密码正确性，若验证通过则生成JWT令牌返回给前端，前端将令牌存储在本地，后续请求携带令牌以验证用户身份。注册功能则是在用户提交注册信息后，后端对信息进行验证（如账号是否重复、密码强度等），无误后将用户信息插入用户表。

### 菜单首页

前端使用Vue.js构建页面，通过Axios向后端发送请求获取菜品信息。后端从数据库的菜单表中查询所有菜品数据，包括菜品名称、价格、图片、描述等，返回给前端。前端根据获取的数据进行页面渲染，展示菜品列表，并提供搜索、筛选功能，通过传递不同参数至后端实现按菜品分类、价格区间等筛选菜品。

### 我的点单与我的订单

用户在菜单首页选择菜品加入点单列表，前端将点单信息存储在本地缓存。用户确认点单后，前端将点单数据（包含菜品ID、数量、用户ID等）发送至后端。后端创建订单记录，插入订单表，并关联订单与菜品的关系到订单详情表。我的订单页面，前端向后端请求用户的订单列表，后端根据用户ID从订单表查询订单数据，包括订单状态（待支付、已支付、配送中、已完成等），返回给前端展示。对于不同状态的订单，前端展示不同的操作按钮，如待支付订单可进行支付操作，配送中订单可查看配送进度。

### 我的收藏、我的评论与我的信息

我的收藏功能实现时，前端在用户点击收藏按钮后，向后端发送包含用户ID和菜品ID的收藏请求。后端将收藏信息插入收藏表。在我的收藏页面，前端请求用户的收藏列表，后端查询收藏表并关联菜单表获取收藏菜品信息返回前端展示。我的评论功能中，用户提交评论时，前端将评论内容、用户ID、订单ID等信息发送至后端，后端插入评论表。我的评论页面，后端根据用户ID查询评论表并关联订单表和菜单表，返回用户的评论及相关菜品、订单信息给前端展示。我的信息页面则是前端向后端请求用户基本信息（如姓名、手机号、邮箱等），后端从用户表查询并返回，前端提供编辑功能，用户编辑后的数据发送至后端更新用户表。

### 个人积分

用户的积分变动通过积分表记录。下单成功、评价等操作触发积分增加时，后端根据积分规则计算积分，向积分表插入积分增加记录，关联订单ID。积分兑换商品或抵扣金额时，插入积分减少记录。个人积分页面，前端请求用户积分余额，后端查询积分表汇总用户积分返回前端展示。

## 食堂管理员功能实现

### 用户登录

与用户登录类似，使用Spring Security结合JWT验证身份，但验证逻辑关联管理员表，确保只有合法管理员能登录。

### 菜单管理

菜单管理功能通过前端的表单和表格组件实现与后端的数据交互。添加菜品时，管理员在前端填写菜品信息并上传图片，前端将数据发送至后端，后端验证数据合法性后插入菜单表。编辑菜品时，前端发送菜品ID及修改后的数据，后端根据ID更新菜单表数据。删除菜品则是前端发送菜品ID，后端删除菜单表中对应记录，并删除相关的订单详情关联记录。

### 配送管理

配送管理模块主要处理订单配送相关操作。后端从订单表获取待配送订单，分配给配送员，更新订单表中的配送员ID和订单状态为“配送中”。配送员完成配送后，后端更新订单状态为“已完成”。前端展示配送订单列表，可查看订单详情、配送员信息和配送进度。

### 评价管理

管理员在评价管理页面，前端向后端请求所有评论列表。后端查询评论表并关联用户表、订单表和菜单表，返回评论及相关信息。管理员可对评论进行审核（如标记违规评论），审核结果更新评论表中的审核状态字段。

### 积分管理

积分管理功能中，管理员可在前端设置积分规则（如下单多少金额获得多少积分），前端将规则数据发送至后端，后端更新积分规则配置表。管理员还能查看用户积分明细，前端请求用户积分记录，后端查询积分表并关联用户表和订单表返回数据展示。

### 数据统计

数据统计功能通过后端编写SQL查询语句实现数据聚合分析。统计菜品销量时，查询订单详情表统计各菜品的销售数量；统计用户活跃度，分析用户登录次数、下单频率等数据。后端将统计结果返回前端，前端使用Echarts等图表库将数据可视化展示，如柱状图展示菜品销量排名，折线图展示用户活跃度随时间变化。

### 点赞管理

在点赞管理页面，前端请求点赞记录列表，后端查询点赞表并关联用户表和菜单表，返回点赞用户信息和被点赞菜品信息。管理员可查看点赞数据，分析热门菜品。

### 用户管理

用户管理功能实现对用户信息的管理。管理员可在前端查询用户列表，后端从用户表查询所有用户信息返回。管理员还能对用户账号进行禁用、启用操作，前端发送用户ID和操作指令，后端更新用户表中的用户状态字段。

# 系统测试

## 测试目的

对校园订餐系统进行全面测试，验证系统是否满足设计要求和用户需求，确保系统功能的正确性、稳定性、性能以及安全性，及时发现并修复系统中存在的缺陷和问题，为系统的上线和后续稳定运行提供保障。

## 测试内容

功能测试：对用户和食堂管理员的各项功能进行逐一测试。如用户的登录注册、菜单浏览、下单、收藏、评论、查看个人信息和积分等功能；食堂管理员的登录、菜单管理（添加、编辑、删除菜品）、配送管理、评价管理、积分管理、数据统计、点赞管理、用户管理等功能。确保每个功能模块的操作流程顺畅，数据的输入、存储和展示正确无误。

性能测试：评估系统在不同负载情况下的性能表现，包括系统的响应时间、吞吐量、并发用户数等指标。模拟大量用户同时进行订餐、查询菜单等操作，测试系统在高并发场景下是否能够稳定运行，响应时间是否在可接受范围内，避免出现系统卡顿或崩溃的情况。

安全测试：检查系统的安全性，防范潜在的安全风险。例如，测试用户登录的密码加密机制，防止密码明文传输；检测系统是否存在 SQL 注入、XSS 攻击等漏洞，确保用户数据和系统的安全性；验证不同用户角色的权限控制，确保食堂管理员和用户只能进行其权限范围内的操作。

兼容性测试：测试系统在不同浏览器（如 Chrome、Firefox、Edge 等）和不同设备（如电脑、平板）上的兼容性，确保系统在各种环境下都能正常显示和使用，用户体验不受影响。

表 六‑1 测试用例表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试用例编号 | 测试功能 | 测试步骤 | 预期结果 | 测试数据 | 测试类型 |
| 1 | 用户注册 | 1. 打开注册页面 | 注册成功，跳转到登录页面，且可使用新注册的账号登录 | 用户名: "test\_user", 密码: "test\_password", 邮箱: "test@example.com" | 功能测试 |
| 2. 输入合法的用户名、密码、邮箱 |
| 3. 点击注册按钮 |
| 2 | 用户注册 | 1. 打开注册页面 | 提示用户名已存在，不允许注册 | 用户名: "existing\_user", 密码: "test\_password", 邮箱: "test@example.com" | 功能测试 |
| 2. 输入已存在的用户名 |
| 3. 点击注册按钮 |
| 3 | 用户登录 | 1. 打开登录页面 | 登录成功，跳转到系统主页面 | 用户名: "test\_user", 密码: "test\_password" | 功能测试 |
| 2. 输入正确的用户名和密码 |
| 3. 点击登录按钮 |
| 4 | 用户登录 | 1. 打开登录页面 | 提示用户名或密码错误，登录失败 | 用户名: "wrong\_user", 密码: "wrong\_password" | 功能测试 |
| 2. 输入错误的用户名或密码 |
| 3. 点击登录按钮 |
| 5 | 菜单浏览 | 1. 登录系统 | 正常显示菜单列表，包含菜品名称、价格、图片等信息 | 无 | 功能测试 |
| 2. 进入菜单页面 |
| 6 | 菜单浏览 | 1. 登录系统 | 仅显示该分类下的菜品信息 | 筛选条件: "素菜" | 功能测试 |
| 2. 进入菜单页面 |
| 3. 点击筛选按钮，选择特定菜品分类 |
| 7 | 菜单添加 | 1. 以商家身份登录 | 新菜品信息成功添加到菜单列表 | 菜品名称: "新菜品", 价格: 10.00, 分类: "荤菜", 库存: 10 | 功能测试 |
| 2. 进入菜品管理页面 |
| 3. 点击添加菜品按钮 |
| 4. 输入新菜品信息，点击保存 |
| 8 | 菜单添加 | 1. 以商家身份登录 | 提示信息不完整，不允许添加 | 菜品名称: "不完整菜品", 分类: "素菜" | 功能测试 |
| 2. 进入菜品管理页面 |
| 3. 点击添加菜品按钮 |
| 4. 输入不完整的菜品信息（如缺少价格），点击保存 |
| 9 | 订单创建 | 1. 用户登录 | 订单创建成功，订单状态为 "待支付" | 菜品: "宫保鸡丁", 数量: 1, 地址: "学校宿舍 101", 支付方式: "微信支付" | 功能测试 |
| 2. 选择菜品添加到购物车 |
| 3. 点击去结算 |
| 4. 选择地址和支付方式，点击提交订单 |
| 10 | 订单创建 | 1. 用户登录 | 提示用户必须选择地址，订单不创建 | 菜品: "宫保鸡丁", 数量: 1, 支付方式: "微信支付" | 功能测试 |
| 2. 选择菜品添加到购物车 |
| 3. 点击去结算 |
| 4. 不选择地址，点击提交订单 |
| 11 | 订单支付 | 1. 用户登录 | 跳转至支付平台，完成支付后订单状态更新为 "已支付" | 无 | 功能测试 |
| 2. 进入待支付订单页面 |
| 3. 选择订单，点击支付按钮 |
| 12 | 订单支付 | 1. 用户登录 | 订单状态仍为 "待支付" | 无 | 功能测试 |
| 2. 进入待支付订单页面 |
| 3. 选择订单，点击支付按钮 |
| 4. 支付过程中取消支付 |
| 13 | 订单取消 | 1. 用户登录 | 订单状态更新为 "已取消" | 无 | 功能测试 |
| 2. 进入待支付订单页面 |
| 3. 选择订单，点击取消订单按钮 |
| 14 | 订单取消 | 1. 用户登录 | 提示不允许取消已支付订单 | 无 | 功能测试 |
| 2. 进入已支付订单页面 |
| 3. 选择订单，点击取消订单按钮 |
| 15 | 订单查看 | 1. 用户登录 | 显示订单详细信息，包括订单状态、菜品列表、地址、支付信息等 | 无 | 功能测试 |
| 2. 进入订单页面 |
| 3. 点击查看订单详情 |
| 16 | 订单查看 | 1. 用户登录 | 仅显示该订单编号的订单信息 | 订单编号: "123456" | 功能测试 |
| 2. 进入订单页面 |
| 3. 搜索特定订单编号 |
| 17 | 评价添加 | 1. 用户登录 | 评价成功添加，可在订单评价列表中查看 | 评分: 4, 评论: "味道不错" | 功能测试 |
| 2. 进入已完成订单页面 |
| 3. 对订单进行评价，输入评分和评论，点击提交 |
| 18 | 评价添加 | 1. 用户登录 | 提示请输入评分，评价不添加 | 评论: "味道不错" | 功能测试 |
| 2. 进入已完成订单页面 |
| 3. 对订单进行评价，不输入评分，点击提交 |
| 19 | 配送分配 | 1. 商家登录 | 订单状态更新为 "已分配" | 无 | 功能测试 |
| 2. 进入订单管理页面 |
| 3. 选择订单，点击分配配送 |
| 20 | 配送分配 | 1. 商家登录 | 提示订单已分配，不允许重复分配 | 无 | 功能测试 |
| 2. 进入订单管理页面 |
| 3. 选择已分配订单，点击分配配送 |
| 21 | 配送状态更新 | 1. 配送员登录 | 订单状态依次更新为 "已取餐"、"配送中"、"已送达" | 无 | 功能测试 |
| 2. 进入待配送订单页面 |
| 3. 选择订单，点击取餐 |
| 4. 点击开始配送 |
| 5. 点击送达 |
| 22 | 配送状态更新 | 1. 配送员登录 | 提示订单已送达，不允许操作 | 无 | 功能测试 |
| 2. 进入已送达订单页面 |
| 3. 选择订单，点击取餐 |
| 23 | 系统性能测试（并发下单） | 1. 使用多个模拟用户同时进行下单操作 | 系统能正常处理并发订单，响应时间在可接受范围内 | 无 | 性能测试 |
| 24 | 系统性能测试（大量菜单浏览） | 1. 多个用户同时浏览菜单页面 | 菜单页面加载时间在可接受范围内 | 无 | 性能测试 |
| 25 | 安全测试（SQL 注入） | 1. 在登录页面输入 SQL 注入语句作为用户名或密码 | 系统拒绝登录，不执行 SQL 语句 | 用户名: "' OR 1=1 --", 密码: "test\_password" | 安全测试 |
| 26 | 安全测试（XSS 攻击） | 1. 在评价输入框输入 XSS 脚本 | 系统对输入的脚本进行转义或过滤，不执行脚本 | 评论: "<script>alert('XSS');</script>" | 安全测试 |
| 2. 提交评价 |

## 测试方法

黑盒测试：在功能测试中主要采用黑盒测试方法，不考虑系统内部的代码结构和实现细节，只关注系统的输入和输出。通过设计各种测试用例，输入不同的测试数据，检查系统的响应是否符合预期结果。例如，在测试用户登录功能时，输入正确和错误的用户名、密码组合，验证系统是否给出正确的提示信息。

性能测试工具：使用 JMeter 进行性能测试。通过创建测试计划，模拟不同数量的并发用户，设置请求的参数和执行次数，对系统的接口进行压力测试。收集并分析测试结果，得出系统在不同负载下的性能指标数据。

安全扫描工具：利用 Burp Suite 进行安全测试，通过拦截和分析 HTTP 请求，检测系统是否存在常见的安全漏洞。对发现的漏洞进行详细分析和验证，确定漏洞的严重程度和影响范围。

手动测试：在兼容性测试中，采用手动测试的方式，在不同的浏览器和设备上访问系统，检查页面布局是否错乱、功能是否可用，及时记录发现的兼容性问题。

## 测试结论

经过全面的系统测试，校园订餐系统在功能方面，大部分功能能够正常使用，满足设计需求，但仍存在少量功能缺陷，如在高并发下单时，偶尔出现订单信息重复提交的情况，已记录并安排修复。性能方面，系统在并发用户数达到 [X] 时，响应时间稍有延长，但仍在可接受范围内；当并发用户数超过 [X] 时，系统出现响应缓慢甚至短暂无响应的现象，需要对系统进行性能优化。安全方面，通过安全扫描发现了一些潜在的 SQL 注入风险和 XSS 攻击漏洞，已进行修复。兼容性测试结果显示，系统在主流浏览器和设备上均能正常运行，但在某些旧版本浏览器上存在页面样式显示异常的问题，需要进一步优化。总体而言，校园订餐系统基本满足上线要求，但还需要针对测试中发现的问题进行修复和优化，以确保系统的稳定性、性能和安全性。

# 总结与展望

## 研究总结

本研究围绕基于 Java 的校园点餐系统展开，通过详细的需求分析、系统设计以及开发实现，成功打造出一个功能较为全面的校园点餐系统。在用户管理功能上，实现了用户注册、登录、信息修改等操作，保障用户能安全便捷地使用系统；菜品管理功能为餐饮商家提供了菜品信息的增删改查服务，极大地方便了菜品更新与管理；点餐与支付功能涵盖了从点餐下单到多种支付方式选择的完整流程，确保交易安全、高效；订单管理功能可实时跟踪订单状态，方便用户和管理员了解订单详情；配送管理功能实现了配送员的任务分配与配送状态更新，提升配送效率；统计与报表功能则能对订单数据、销售数据等进行统计分析，为运营决策提供数据支撑。

从技术层面，基于 Java 语言的 Spring Boot 框架构建后端，利用其依赖注入和自动配置特性，提高开发效率与系统稳定性；前端采用 Vue.js 框架结合 Element UI 组件库，实现了美观且交互性强的用户界面；数据库选用 MySQL，满足系统数据存储与管理需求。经过多轮测试，系统各项功能运行稳定，达到预期设计目标，有效解决了校园点餐场景下的诸多痛点，提升了校园餐饮服务水平。

## 研究不足与展望

尽管本系统取得了一定成果，但仍存在一些不足之处。在功能方面，系统目前的个性化推荐功能较弱，无法精准地根据用户历史点餐数据和偏好推荐菜品。同时，在多校区或大规模校园场景下，配送管理的复杂度增加，当前配送路径规划和调度策略有待进一步优化。

在未来展望中，功能优化是重点方向之一。后续可引入深度学习算法，如协同过滤算法，对用户点餐数据进行深度挖掘，实现更加精准的个性化菜品推荐，提高用户点餐的满意度和效率。针对配送管理，可结合地理信息系统（GIS）技术，实现实时路况获取和智能配送路径规划，根据配送员位置、订单分布和交通状况，动态规划最优配送路线，提高配送效率，降低配送成本。

技术升级也是重要的发展方向。随着微服务架构的普及，可将现有系统进行微服务化改造，将各个功能模块拆分为独立的微服务，实现独立部署、独立扩展和灵活运维，提高系统的可维护性和可扩展性。在数据处理方面，引入大数据处理框架，如 Hadoop 和 Spark，应对未来可能出现的海量订单数据和用户行为数据，进行更高效的数据存储、处理和分析，挖掘更多有价值的信息，为系统运营和决策提供更强大的数据支持。

致 谢

在完成这篇论文的过程中，我深感自己收获的不仅仅是学术成果，更有满满的感动与感恩。在此，我要向所有给予我帮助的人们致以最诚挚的谢意。首先，我要衷心感谢我的论文指导老师，从论文的选题、资料收集到结构搭建和内容撰写，每一个环节都凝聚着您的心血。论文督导老师，始终以严谨认真的态度监督着我，对我的论文提出了许多宝贵的修改意见，使我不断改进，促使我对研究课题有了更深刻的认知。另外，我还要感谢为我提供帮助的同学们。在我为论文的某个问题而苦恼时，是你们的建议和帮助给了我灵感；在我感到压力和焦虑时，是你们的鼓励和陪伴给了我力量。最后，我会带着这份感恩之情继续前行，努力提升自己，不辜负大家的期望。再次向帮助过我的人表示深深的感谢！

参考文献

1. 钟育伙.基于SpringBoot+Vue的校园活动管理系统设计与实现[J].电子技术,2024,53(10):56-57.
2. 朴明,邱翠花,苗子.基于SpringBoot+小程序的信息采集系统设计与实现[J].电子技术,2024,53(10):47-49.
3. 殷志强,钟卫东.基于SpringBoot应用中的XSS攻击防御技术[J].工业控制计算机,2024,37(11):62-64.
4. 戴亚哲,李尤,赵利宏,等.基于SpringBoot+Vue的文旅平台设计与研究[J].无线互联科技,2024,21(21):70-72.
5. 乔凤权,李精伟,徐进,等.基于VUE的泵站运维系统的前端设计与实现[J].科技创新与生产力,2024,45(11):129-132.
6. 李晓薇.vue.js前端应用技术分析[J].网络安全技术与应用,2022,(04):44-45.
7. 罗光武,陈典灿,吴荷,等.应用Springboot+Vue框架的时间管理软件的设计与实现[J].工业控制计算机,2024,37(04):64-66.
8. 徐少军,李宗哲,梅杰,等.基于Springboot+Vue框架的质量检验监督管理系统研发[J].纺织标准与质量,2024,(01):11-14+21.
9. 昝国宁,王雨晴,刘娇龙,等.MySQL数据库自动化运维管理系统的设计与实现[J].铁路计算机应用,2024,33(09):39-43.
10. 庞敏.MySQL数据库的数据安全应用设计技术研究[J].数字通信世界,2024,(09):25-27.
11. 吴敏.基于Java Web的班级信息管理系统的设计与实现[J].中国新通信,2024,26(07):74-76.
12. 匡少华,丁昊,赵正平.基于Javaweb的大学生食堂外卖平台的设计与实现[J].信息与电脑(理论版),2020,32(01):91-94.
13. Liu Y .Design and Implementation of a Student Attendance Management System based on Springboot and Vue Technology[J].Frontiers in Computing and Intelligent Systems,2024,8(1):91-97.
14. Hejing W .Commerce Middle Office Management System Based on Springboot[J].International Journal of Advanced Network, Monitoring and Controls,2022,7(2):32-45.
15. Mario V ,Sanja K .DEVELOPMENT OF A LIBRARY APPLICATION USING VUE.JS AND FIREBASE TECHNOLOGIES[J].Polytechnic and design,2023,11(1):66-73.