需求规约

***图书管理系统***

东软XX科技有限公司

**(版权所有，翻版必究)**

**更改履历**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 版本 | 更改时间 | 更改人 | 更改章节 | 状态 | 更改描述 |
| 1 | 1.0 | 2023-8-23 | 全体成员 | 全部 | 新建 | 无 |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

状态：新建、增加、修改、删除。

**目录**

[第1章 绪论 4](#_Toc30804)

[1.1 研究背景 4](#_Toc30002)

[1.2 研究目的 4](#_Toc8325)

[第2章 系统分析 4](#_Toc13434)

[2.1 整体应用系统架构 4](#_Toc961)

[2.2 功能需求分析 4](#_Toc19886)

[2.3 非功能需求分析 4](#_Toc32014)

[第3章 系统设计 4](#_Toc27170)

[3.1 环境设计 4](#_Toc2359)

[3.2 用户界面设计 4](#_Toc12600)

[3.3 系统界面设计 4](#_Toc16616)

[3.4 应用程序设计 4](#_Toc29809)

[3.5 数据库设计 4](#_Toc1276)

[第4章 系统实现 4](#_Toc6419)

[4.1 系统模块功能实现 4](#_Toc30231)

[第5章 系统测试 5](#_Toc5091)

[5.1 测试用例 5](#_Toc23018)

[5.2 结果分析 5](#_Toc12227)

[第6章 结论与未来研究方向 5](#_Toc25569)

[6.1 结论 5](#_Toc24119)

[6.2 未来研究方向 5](#_Toc17426)

## 第1章 绪论

## 1.1 研究背景

随着互联网的普及和移动设备的普遍使用，以及人们生活习惯的改变，外卖行业在中国和全球范围内迅速发展。外卖服务已经成为人们生活中不可或缺的一部分，为用户提供了便捷、多样化的餐饮选择。尤其在特殊时期，如疫情期间，外卖服务更是成为了人们用餐的主要方式之一。

我们项目组也积极响应时代潮流，希望能够为人们提供更便捷的外卖订餐服务。因此，本项目“饿了吧外卖”应运而生，旨在为人们提供一个高效、便捷、多样化的外卖点餐平台，同时为食品商家提供一个方便的订单管理工具。

1.2 研究目的

本项目的主要研究目的包括：

1. 开发一个基于微信小程序的用户端，使用户能够浏览餐厅菜单、下单购买食品，并实现在线支付功能，提高用户点餐体验。
2. 开发一个网页管理端，供食品商家使用，实现订单管理、菜单管理、店铺信息管理等功能，帮助商家更好地管理外卖业务。
3. 利用现代Web开发技术，如Spring、Java、Vue等，构建一个高效、稳定、安全的系统，以确保系统的可维护性和扩展性。
4. 引入数据库技术（MySQL）和缓存技术（Redis）来存储和管理用户数据、订单数据等，以提高系统的数据处理效率和稳定性。
5. 通过本项目的实施，培养团队成员的软件开发技能，包括但不限于前端开发、后端开发、数据库设计、项目管理等方面的技能，提升团队协作和解决问题的能力。

通过实现上述目标，我们希望提供一个现代化的外卖订餐平台，提高餐饮服务的质量和效率，同时促进技术人才的培养和创新实践。

# 第2章 系统分析

2.1 整体应用系统架构

1. 系统架构概述

项目“饿了吧外卖”采用了分布式系统架构，包括用户端微信小程序、商家管理端网页以及后台服务。以下是系统的整体架构：

1. 用户端微信小程序：用户通过微信小程序进行菜单浏览、下单、支付等操作。小程序将用户请求发送至后台服务，并接收后台服务返回的数据。
2. 商家管理端网页：食品商家通过网页管理端进行订单处理、菜单管理、店铺信息管理等操作。网页管理端与后台服务进行通信，实时同步订单和菜单数据。
3. 后台服务：后台服务充当核心的业务逻辑处理中心，包括用户身份验证、订单处理、菜单管理、支付处理等功能。后台服务与数据库和缓存系统进行交互，确保数据的稳定性和高效性。
4. 数据库：使用MySQL数据库来存储用户信息、订单数据、菜单信息等重要数据。数据库用于持久性数据存储。
5. 缓存系统：使用Redis缓存系统来缓存热门数据，如菜单信息、用户会话信息等，以提高系统的响应速度。
6. 技术栈概述

项目的技术栈包括但不限于：

前端开发：HTML、CSS、JavaScript、Vue.js、ElementUI

后端开发：Spring Boot、SpringMVC、Java

数据库：MySQL、Mybatis

缓存：Redis

微信小程序开发

* 1. 功能需求分析

1. 用户端微信小程序功能需求

用户端微信小程序的功能需求包括但不限于：

1. 用户注册与登录：用户可以使用微信账号快捷登录。
2. 菜单浏览：用户可以浏览附近各家餐厅的菜单，查看菜品详情、价格等信息。
3. 购物车管理：用户可以将菜品添加到购物车，查看购物车内容，修改数量或删除菜品。
4. 订单下单与支付：用户可以选择菜品并下单，可以选择添加备注、餐具数量、发票。最后完成支付。
5. 地址管理：用户可以在订单页面选择收货地址，也可以新增收货地址。
6. 个人信息管理：用户可以编辑个人资料、修改密码、查看订单记录等。
7. 订单管理：用户可以查看使用历史订单信息
8. 商家管理端网页功能需求

商家管理端网页的功能需求包括但不限于：

1. 登录与身份验证：商家需要登录管理端，进行身份验证。
2. 查看店铺营业情况：商家在工作台首页可以查看营业额、订单信息、菜品信息等等。也可以设置营业状态。
3. 菜单管理：商家可以添加、编辑、删除菜品，管理菜品分类。
4. 订单管理：商家可以查看新订单，接受或拒绝订单，标记订单状态（已接单、配送中、已完成等）。
5. 店铺信息管理：商家可以编辑店铺信息，包括店铺名称、地址、联系方式等。
6. 数据统计与报表：商家可以查看销售数据、订单量统计，生成报表。
7. 员工管理：管理员可以添加、修改及删除员工信息
   1. 非功能需求分析
8. 性能需求
9. 响应时间：系统应在用户提交请求后迅速响应，保持良好的用户体验。页面加载时间应控制在数秒内。
10. 并发处理能力：系统应能够处理大量并发用户请求，包括同时下单、浏览菜单等操作。
11. 可伸缩性：系统应具备良好的可伸缩性，以便在需要时扩展服务器资源，应对用户量的增加。
12. 安全性需求
13. 数据安全：用户敏感数据如密码和支付信息应进行加密存储，确保数据的机密性。
14. 身份验证：系统应使用安全的身份验证机制，防止未经授权的访问。
15. 支付安全：支付过程应采用安全的加密传输方式，保护用户的支付信息不被窃取。
16. 可用性需求
17. 系统稳定性：系统应保持高可用性，减少系统崩溃和故障的可能性。
18. 用户友好性：用户界面应设计友好，易于使用，提供清晰的操作指导。
19. 可维护性需求
20. 代码可维护性：项目代码应具备良好的可维护性，包括注释、文档、模块化等，以便后续维护和扩展。
21. 系统监控：系统应具备监控功能，能够及时发现和处理异常情况。

# 第3章 系统设计

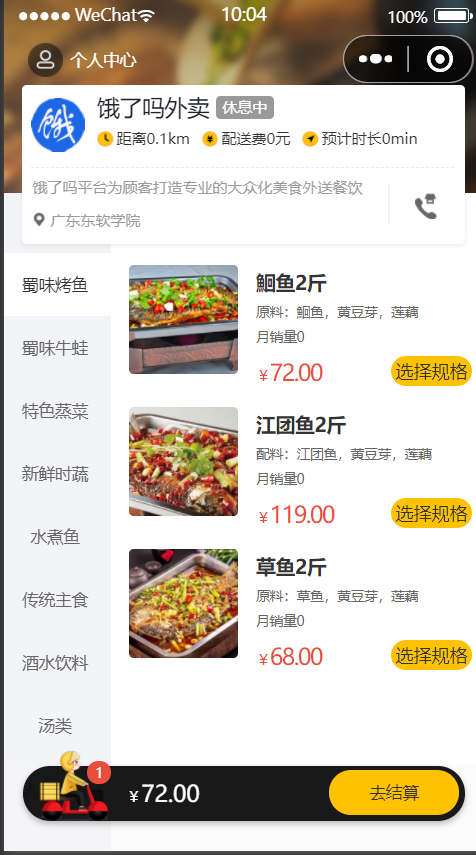
3.1 环境设计

1. 开发环境
2. IDE（集成开发环境）：我们将使用IDEA（IntelliJ IDEA）作为主要的Java开发IDE，以及VScode用于前端开发。
3. 前端技术栈：用户界面的开发将使用Vue.js作为前端框架，同时使用ElementUI库来快速构建用户界面。
4. 后端技术栈：后端将使用Spring Boot作为主要的Java开发框架，Spring MVC用于Web层，Spring Task用于任务调度，HttpClient用于处理HTTP请求，Spring Cache用于缓存管理，JWT（JSON Web Tokens）用于身份验证和安全性。
5. 微信小程序开发：为了实现用户端，我们将使用微信小程序开发工具。
6. 数据存储：我们将使用MySQL数据库来存储用户信息、订单数据、菜单信息等。同时，我们会使用Redis缓存来提高系统的响应速度。
7. 云存储：阿里云OSS（对象存储服务）用于存储用户上传的图片和文件。
8. 文档生成：我们将使用Swagger来生成API文档，以便开发人员和团队成员查阅和使用API。
9. 运行环境
10. 用户端微信小程序：用户可以在支持微信小程序的iOS和Android设备上运行。
11. 商家管理端网页：商家可以在支持现代Web浏览器的计算机上运行，如Chrome、Firefox、Safari等。
12. 后台服务：后台服务将部署在云服务器上，保证系统的高可用性和可伸缩性。
13. 数据库环境
14. 数据库服务器：使用MySQL数据库，将其部署在独立的数据库服务器上，确保数据的安全和稳定。
15. 缓存服务器：使用Redis缓存服务器，用于缓存热门数据，提高系统响应速度。
16. 1. 用户界面设计
17. 用户端微信小程序界面设计
18. 登陆：程序获取用户微信登陆信息。

图形用户界面, 应用程序

描述已自动生成

1. 主页：显示餐厅和菜单



1. 选择规格：展示菜品的规格



1. 购物车：显示用户已选菜品，支持增加、减少数量和删除操作。



1. 订单确认：用户确认订单

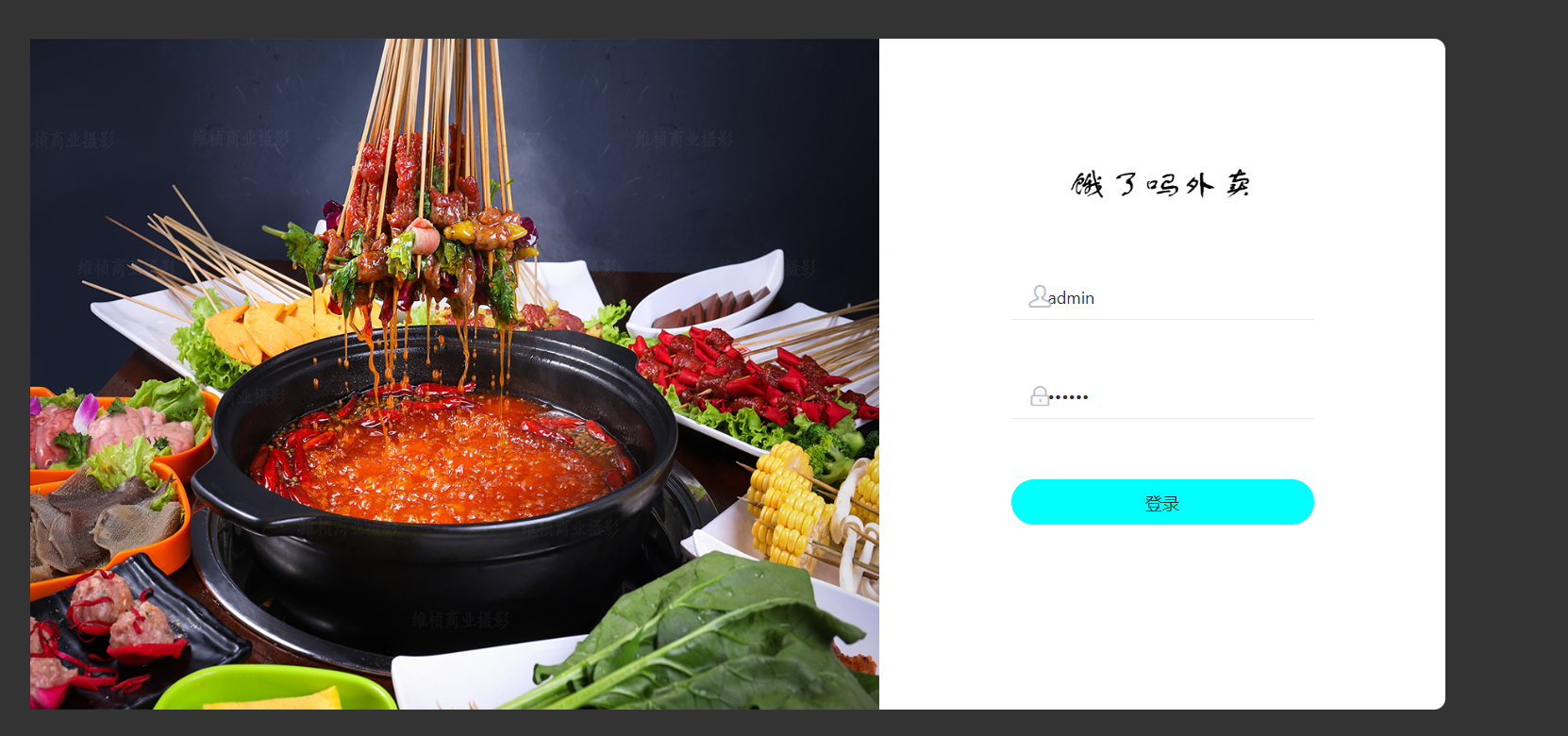
图形用户界面, 应用程序

描述已自动生成

1. 个人中心：用户可以查看个人信息、订单记录等。



1. 商家管理端网页界面设计
2. 登录页：商家输入用户名和密码进行登录。



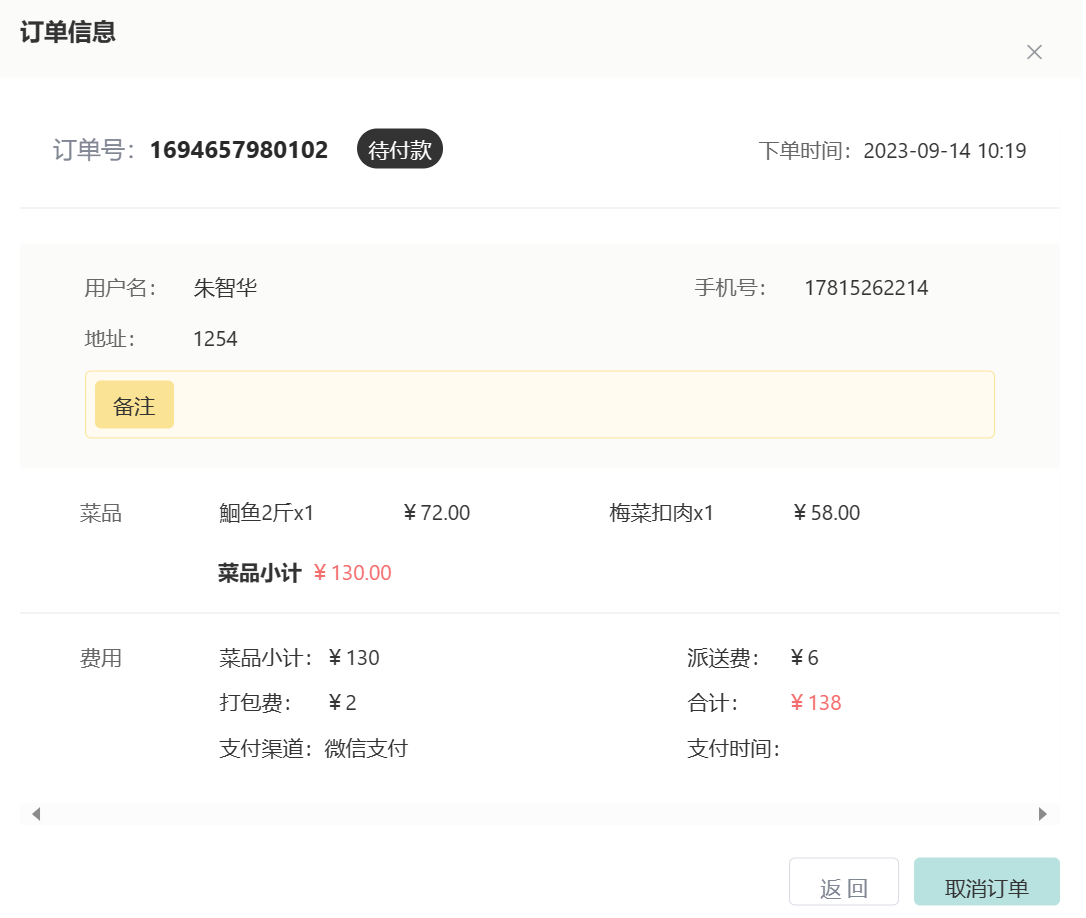
1. 菜单管理：展示菜品列表，支持添加、编辑、查询和删除操作。



1. 订单管理：显示订单列表，包括订单号、订单状态、用户名、手机号、地址和金额，支持接受（默认）、取消等操作。
2. 订单管理：显示订单列表（包括订单号、），支持接受、拒绝、查看订单信息等操作。

图形用户界面, 应用程序

描述已自动生成



1. 数据统计：统计一段时间（昨日、一周、一个月、当月）的营业额；统计订单



* 1. 应用程序设计

1. 用户端微信小程序应用程序设计
2. 前端开发：采用Vue.js框架进行前端开发，实现用户界面的交互和数据展示。
3. 微信小程序开发：使用微信小程序开发工具，实现微信登录、支付功能等。
4. 数据交互：通过RESTful API与后台服务进行数据交互，包括获取菜单信息、提交订单等。
5. 商家管理端网页应用程序设计
6. 前端开发：使用Vue.js框架进行前端开发，实现商家管理端界面的交互和数据展示。
7. 数据交互：通过RESTful API与后台服务进行数据交互，包括菜单管理、订单处理等。
   1. 数据库设计
8. 数据库表设计
9. 收货地址信息表：用于存储用户的收货信息

表名：address\_book

图表

描述已自动生成

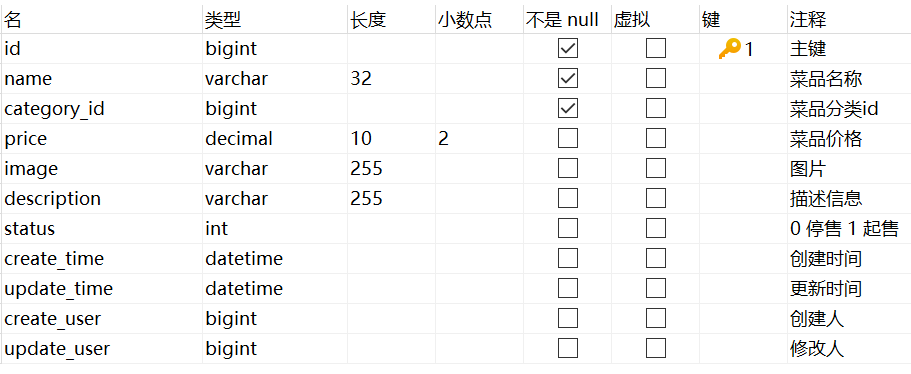
1. 菜品类别表：用于存储菜品的分类

表面：category



1. 菜品表：用于存储菜品信息

表名：dish



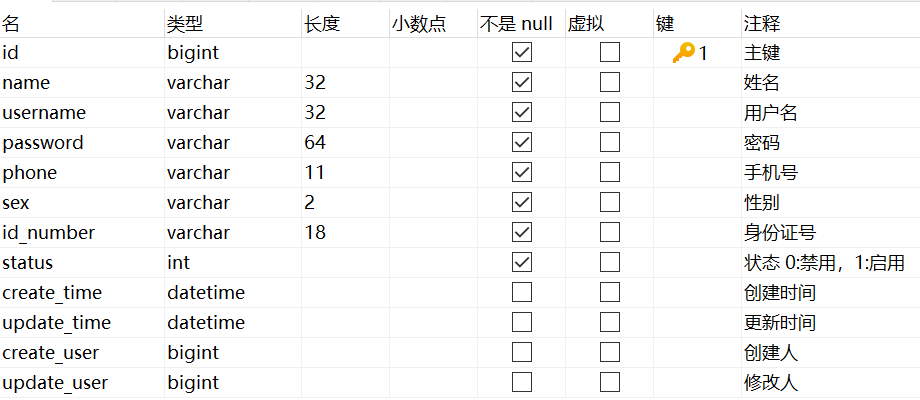
1. 菜品规格表：用于存储菜品的规格

表名：dish\_flavor



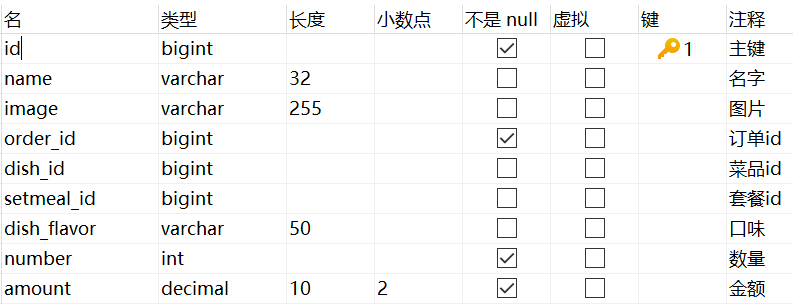
1. 管理员表：用于存储管理员信息

表名：employee



1. 订单详情表：用于存储用户下单的菜品及规格、数量等等

表名：order\_detail



1. 订单表：用于存储用户的订单

表名：orders



1. 套餐表：用于存储套餐信息

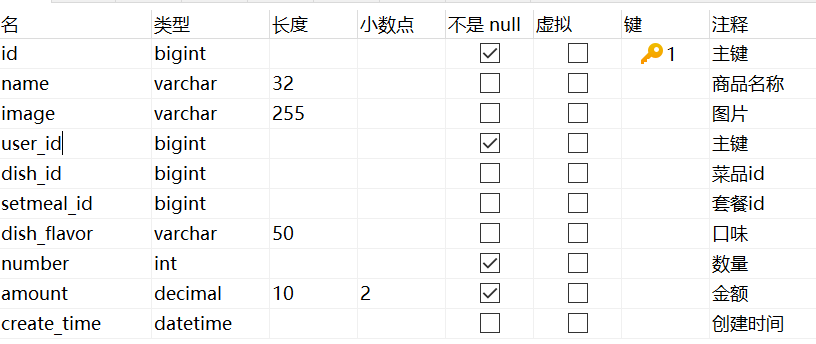
表名：setmeal

图形用户界面, 应用程序, 表格, Excel

描述已自动生成

1. 购物车表：用于存储用户下单的菜品信息

表名：shopping\_cart



1. 用户信息表：用于存储用户的信息

表名：user



# 第4章 系统实现

4.1 前端实现

1. 用户端微信小程序
2. 首页：用户可以浏览推荐餐厅、查看菜单，浏览菜品、价格。
3. 菜单浏览：用户可以按照菜品分类条件浏览菜单，查看详细信息和图片。
4. 购物车：用户可以将菜品添加到购物车，增加、减少数量，清空购物车等。
5. 订单确认：用户可以查看订单内容，选择支付方式，提交订单，获得订单号和预计送达时间。
6. 个人中心：用户可以查看个人信息，包括历史订单记录，管理收货地址。
7. 商家管理端网页
8. 登录：商家通过用户名和密码登录管理端。
9. 菜单管理：商家可以添加、编辑、删除菜品，包括菜品名称、价格、描述、图片、规格等信息。
10. 分类管理：商家可以给菜品设置分类消息，便于用户快速找到想要菜品
11. 订单管理：商家可以查看订单列表，接受或拒绝订单，标记订单状态，与用户进行实时沟通。
12. 数据统计与报表：商家可以查看销售数据统计报表，分析经营状况。
    1. **后端实现**
13. 用户认证和授权
14. 用户注册：用户可以通过提供必要信息完成注册，系统会为每个用户分配一个唯一的用户ID。
15. 用户登录：已注册的用户可以使用用户名和密码进行登录，系统验证用户身份并生成JWT令牌用于后续请求的身份验证。
16. JWT令牌：用户登录成功后，服务器会生成JWT令牌，包含用户ID和角色信息，用于验证用户的身份和授权访问。
17. 菜品管理
18. 菜品添加与编辑：商家可以添加新菜品并提供详细信息，包括菜品名称、价格、描述、图片、规格等。
19. 菜品删除：商家可以删除不再提供的菜品，保持菜单的更新。
20. 订单处理
21. 订单创建：用户下单后，系统创建订单，记录用户ID、餐厅ID、菜品ID、数量、状态等信息。
22. 订单状态更新：商家可以接受或拒绝订单，同时系统会实时更新订单状态，用户和商家可以实时查看订单状态。
23. 数据统计与报表
24. 销售数据统计：系统会定期统计销售数据，包括订单数量、销售额等，生成报表供商家查看。

**4.3 数据库实现**

1. 数据库表设计

数据库中包括了多个表，如用户信息表、菜品表、订单表、菜品规格表等，它们的设计和关系保证了数据的一致性和完整性。

1. 数据吃持久化

用户数据：用户信息、订单数据、菜单信息等数据被持久化存储在MySQL数据库中。

缓存管理：使用Redis缓存来提高热门数据的访问速度，减轻数据库负载。

# 第5章 系统测试

5.1 测试用例

1. 用户登录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例 | 输入 | 预想输出 | 实际输出 | 判定 结果 |
|  |
| **登录验证** |  |  |  |  |  |
| 正确的用户名和密码 | 输入正确的用户名和密码 | 成功登录，首次需修改密码 | **登录成功** | OK |  |
| 错误的用户名和密码 | 输入错误的账号或错误的密码 | 登录失败，显示错误提示 | **登录失败** | OK |  |

1. 订单管理

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例 | 输入 | 预想输出 | 实际输出 | 判定 结果 |
|  |
| **查看订单信息** |  |  |  |  |  |
| 点击“查看”查看订单完整信息 | 点击“查看” | 显示订单的详细信息 | **显示订单详细信息** | OK |  |
| **新增订单信息** |  |  |  |  |  |
| 订单地址格式错误 | 输入的地址格式不规范或为空 | 提示：地址格式错误 | **提示：地址格式错误** | OK |  |
| 联系人姓名为空 | 不输入联系人姓名 | 提示：姓名不能为空 | **提示：姓名不能为空** | OK |  |
| 联系电话格式错误 | 输入错误的电话格式或空值 | 提示：请输入正确的电话号码 | **提示：请输入正确的电话号码** | OK |  |
| 购物车未选购菜品 | 不选择菜品 | 无法选择“去支付” | **“去支付”按钮不可选** | OK |  |
| **查询订单信息** |  |  |  |  |  |
| 订单号不存在 | 输入不存在的订单号 | 提示：没有找到内容 | **提示：没有找到内容** | OK |  |
| 手机号不存在 | 输入不存在的手机号 | 提示：没有找到内容 | **提示：没有找到内容** | OK |  |
| 日期不正确 | 输入不存在的日期 | 提示：没有找到内容 | **提示：没有找到内容** | OK |  |
| 订单号存在 | 输入存在的订单号 | 显示该订单信息 | **显示该订单信息** | OK |  |
| 手机号存在 | 输入存在的手机号 | 显示该订单信息 | **显示该订单信息** | OK |  |
| 日期正确 | 输入正确的日期 | 显示日期内所有订单 | **显示日期内所有订单** | OK |  |

1. 菜品管理

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例 | 输入 | 预想输出 | 实际输出 | 判定 结果 |
|  |
| **新增菜品信息** |  |  |  |  |  |
| 菜品名称为空 | 不输入菜品名称，点击保存 | 提示：请输入菜品名称 | **提示：请输入菜品名称** | OK |  |
| 菜品分类为空 | 不选择菜品分类，点击保存 | 提示：请选择菜品分类 | **提示：请选择菜品分类** | OK |  |
| 菜品图片为空 | 不添加菜品图片，点击保存 | 提示：菜品图片不能为空 | **提示：菜品图片不能为空** | OK |  |
| 菜品价格为空 | 不输入价格，点击保存 | 提示：菜品价格格式有误，请输入大于零且最多保留两位小数的金额 | **提示：菜品价格格式有误，请输入大于零且最多保留两位小数的金额** | OK |  |
| 菜品做法配置为空 | 不输入菜品做法配置 | 保存成功 | **保存成功** | OK |  |
| 菜品备注为空 | 不输入菜品备注 | 保存成功 | **保存成功** | OK |  |
| **修改菜品信息** |  |  |  |  |  |
| 菜品名称为空 | 不输入菜品名称，点击保存 | 提示：请输入菜品名称 | **提示：请输入菜品名称** | OK |  |
| 菜品分类为空 | 不选择菜品分类，点击保存 | 提示：请选择菜品分类 | **提示：请选择菜品分类** | OK |  |
| 菜品图片为空 | 不添加菜品图片，点击保存 | 提示：菜品图片不能为空 | **提示：菜品图片不能为空** | OK |  |
| 菜品价格为空 | 不输入价格，点击保存 | 提示：菜品价格格式有误，请输入大于零且最多保留两位小数的金额 | **提示：菜品价格格式有误，请输入大于零且最多保留两位小数的金额** | OK |  |
| 菜品做法配置为空 | 不输入菜品做法配置 | 保存成功 | **保存成功** | OK |  |
| 菜品备注为空 | 不输入菜品备注 | 保存成功 | **保存成功** | OK |  |
| 菜品名称正确 | 输入正确的菜品名称，点击保存 | 保存成功 | **保存成功** | OK |  |
| 菜品分类正确 | 选择菜品分类 | 选择菜品分类 | **保存成功** | OK |  |
| 菜品图片格式正确 | 添加符合要求的图片 | 保存成功 | **保存成功** | OK |  |
| 菜品价格正确 | 输入正确价格，点击保存 | 保存成功 | **保存成功** | OK |  |
| **删除菜品信息** |  |  |  |  |  |
| 删除菜品信息 | 点击“删除”按钮 | 提示：无法删除起售中菜品 | **提示：无法删除起售中菜品** | OK |  |
| 删除菜品信息 | 点击“停售”，再点击“删除”按钮 | 提示：删除成功 | **提示：删除成功** | OK |  |
| **查询菜品信息** |  |  |  |  |  |
| 查询不存在的菜品 | 输入不存在的菜品名称查询 | 不显示内容 | **不显示内容** | OK |  |
| 选择菜品分类查询 | 选择某一分类点击“查询” | 显示该分类下的菜品 | **显示该分类下的菜品** | OK |  |
| 按售卖状态查询 | 选择售卖状态 | 显示不同售卖状态的菜品 | **显示不同售卖状态的菜品** | OK |  |
| 查询存在的菜品 | 输入存在的菜品名称查询 | 显示菜品内容 | **显示菜品内容** | OK |  |

1. 员工管理

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例 | 输入 | 预想输出 | 实际输出 | 判定 结果 |
|  |
| **查看员工信息** |  |  |  |  |  |
| 查看员工信息 | 点击员工管理 | 显示员工信息 | **显示员工信息** | OK |  |
| **添加员工信息** |  |  |  |  |  |
| 员工姓名为空 | 不输入员工姓名，点击保存 | 提示：请输入员工姓名 | **提示：请输入员工姓名** | OK |  |
| 员工账号为空 | 不输入员工账号，点击保存 | 提示：请输入员工账号 | **提示：请输入员工账号** | OK |  |
| 员工手机号码为空 | 不输入员工手机号码，点击保存 | 提示：请输入员工手机号码 | **提示：请输入员工手机号码** | OK |  |
| 身份证号码为空 | 不输入身份证号码，点击保存 | 提示：请输入身份证号码 | **提示：请输入身份证号码** | OK |  |
| 员工姓名格式正确 | 输入正确的员工姓名，点击保存 | 保存成功 | **保存成功** | OK |  |
| 员工账号格式正确 | 输入正确的员工账号 | 保存成功 | **保存成功** | OK |  |
| 员工手机号码格式正确 | 输入正确的员工手机号码，点击保存 | 保存成功 | **保存成功** | OK |  |
| 身份证号码格式正确 | 输入正确的身份证号码，点击保存 | 保存成功 | **保存成功** | OK |  |
| **修改员工信息** |  |  |  |  |  |
| 员工姓名为空 | 不输入员工姓名，点击保存 | 提示：请输入员工姓名 | **提示：请输入员工姓名** | OK |  |
| 员工账号为空 | 不输入员工账号，点击保存 | 提示：请输入员工账号 | **提示：请输入员工账号** | OK |  |
| 员工手机号码为空 | 不输入员工手机号码，点击保存 | 提示：请输入员工手机号码 | **提示：请输入员工手机号码** | OK |  |
| 身份证号码为空 | 不输入身份证号码，点击保存 | 提示：请输入身份证号码 | **提示：请输入身份证号码** | OK |  |
| 员工姓名格式正确 | 输入正确的员工姓名，点击保存 | 保存成功 | **保存成功** | OK |  |
| 员工账号格式正确 | 输入正确的员工账号 | 保存成功 | **保存成功** | OK |  |
| 员工手机号码格式正确 | 输入正确的员工手机号码，点击保存 | 保存成功 | **保存成功** | OK |  |
| 身份证号码格式正确 | 输入正确的身份证号码，点击保存 | 保存成功 | **保存成功** | OK |  |
| **禁用员工信息** |  |  |  |  |  |
| 禁用员工信息 | 选中员工点击“禁用” | 账号状态更改成功！ | **账号状态更改成功！** | OK |  |
| **查询员工信息** |  |  |  |  |  |
| 员工姓名为空 | 不输入员工姓名点击查询 | 页面不更新 | **页面不更新** | OK |  |
| 查询存在的员工 | 输入存在的员工姓名 | 显示该员工的信息 | **显示该员工的信息** | OK |  |

1. 密码管理

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例 | 输入 | 预想输出 | 实际输出 | 判定 结果 |
|  |
| **修改密码** |  |  |  |  |  |
| 原密码错误 | 输入错误的原密码 | 提示：请输入正确的原密码 | **提示：请输入正确的原密码** | OK |  |
| 新密码格式错误 | 输入格式不正确的新密码 | 提示：新密码格式错误 | **提示：新密码格式错误** | OK |  |
| 密码与确认密码不一致 | 输入不一样的新密码和确认密码 | 提示：新密码不一致 | **提示：新密码不一致** | OK |  |
| 原密码正确 | 输入正确的原密码 | 保存成功 | **保存成功** | OK |  |
| 新密码格式正确 | 输入格式正确的新密码 | 保存成功 | **保存成功** | OK |  |
| 密码与确认密码一致 | 输入一样的新密码和确认密码 | 保存成功 | **保存成功** | OK |  |

1. 营业状态管理

5.2 结果分析

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例 | 输入 | 预想输出 | 实际输出 | 判定 结果 |
|  | |
| **设置营业状态** |  |  |  |  | |
| 营业状态为营业中 | 更改营业状态为营业中 | 更改成功 | **更改成功** | OK | |
| 营业状态为打烊中 | 更改营业状态为打烊中 | 更改成功 | **更改成功** | OK | |

# 第6章 结论与未来研究方向

6.1 结论

在本项目中，我们成功地开发了名为“饿了吧外卖”的外卖平台，该平台提供了用户端微信小程序和商家管理端网页，允许用户浏览菜单、下单，并使商家能够管理订单和菜单。整个项目的开发和实现基于一系列技术栈和模块，包括Vue.js、Spring Boot、MySQL、Redis、微信小程序等。

项目的主要成果和结论如下：

1. 功能齐全的外卖平台：我们成功实现了一个功能齐全的外卖平台，包括用户认证、菜品管理、订单处理、数据统计与报表等核心功能。
2. 用户友好的界面：通过使用Vue.js和ElementUI，我们创建了用户友好的界面，使用户能够轻松浏览菜单、下单、支付和管理个人信息。
3. 实时通信和消息推送：通过WebSocket，我们实现了实时通信和订单状态的实时更新，提高了用户和商家之间的交互性。
4. 数据库和缓存优化：通过MySQL和Redis的合理使用，我们优化了数据存储和访问，提高了系统的性能和响应速度。
5. 性能和安全性测试：我们进行了性能测试和安全性测试，确保系统在高并发情况下的稳定性和安全性。
   1. 未来研究方向

尽管我们已经成功实现了“饿了吧外卖”项目，但仍有一些未来研究方向和改进的空间，以进一步提升平台的性能和用户体验：

1. 多平台支持：未来可以考虑扩展到更多的平台，如iOS和Android应用程序，以覆盖更广泛的用户群体。
2. 推荐系统：引入智能推荐系统，根据用户的历史订单和偏好，推荐适合他们口味的菜品。
3. 多语言支持：增加多语言支持，以满足不同地区和语言的用户需求。
4. 数据分析和机器学习：使用数据分析和机器学习技术，进一步优化菜单、价格和促销策略，提高用户留存率。
5. 社交媒体集成：将社交媒体集成到平台中，使用户能够与朋友分享菜单、订单和评论。
6. 支付方式扩展：支持更多的支付方式，以方便用户进行支付，如支付宝、Apple Pay等。
7. 持续性能优化：根据用户使用情况，持续优化数据库查询和响应速度，提高系统的性能。
8. 安全性增强：定期进行安全性评估，确保用户数据的安全，防止潜在的安全漏洞。
9. 用户反馈和改进：积极收集用户反馈，不断改进用户界面和体验，以满足用户需求。