

中国大学生计算机设计大赛



软件开发类作品文档简要要求

作品编号：_____

作品名称：饿了没——基于 React 与 Gin 框架的外卖平台

作 者：刘遥，栾子骞，舒凯，雷玥，赵丛霖

版本编号：1.0

填写日期：2024 年 4 月 11 日

填写说明：

- 1、本文档适用于所有涉及软件应用与开发类的各个小类作品，包括：(1) Web 应用与开发 (2) 管理信息系统 (3) 移动应用开发（非游戏类）(4) 算法设计与应用 (5) 信创软件应用与开发 (6) 区块链应用与开发；
- 2、本文档为简要文档，不宜长篇大论，需简明扼要，建议设计二级目录，逻辑性强；
- 3、一级标题采用二号黑体，居中，二级标题采用三号黑体，靠左，根据需要可以设计三级标题，正文一律用五号宋体；
- 4、提交文档时，以 PDF 格式提交本文档；
- 5、本文档内容是正式参赛内容组成部分，务必真实填写。如不属实，将导致奖项等级降低甚至终止本作品参加比赛。

目 录

第一章 需求分析	4
1.1 开发背景.....	4
1.2 市场分析.....	4
1.2.1 目标用户.....	4
1.2.2 竞品分析.....	4
1.3 作品介绍.....	4
1.3.1 主要功能.....	5
1.3.2 优势和创新点.....	5
第二章 概要设计	5
2.1 总体设计.....	5
2.2 外卖点单流程.....	6
2.3 功能模块设计.....	6
2.3.1 用户管理模块.....	6
2.3.2 菜品管理模块.....	6
2.3.3 评论模块.....	6
2.3.4 订单管理模块.....	7
2.3.5 数据统计模块.....	7
第三章 详细设计	7
3.1 界面设计.....	7
3.1.1 顾客界面.....	7
3.1.2 员工界面.....	9
3.1.3 送餐员界面.....	9
3.2 数据库设计.....	10
3.2.1 ER 图设计.....	10
3.2.2 表设计与实体间关系.....	10
3.2.3 数据库访问.....	11
3.3 关键技术.....	12
3.3.1 基于 React 与 ElementUI 的前端设计	12

3.3.2 基于 Gin 与 Gorm 的后端架构.....	12
第四章 测试报告	12
4.1 各模块功能测试.....	12
4.2 性能测试.....	15
第五章 安装及使用	15
5.1 配置前端环境.....	15
5.2 配置后端环境.....	15
第六章 项目总结	16
参考文献	16

第一章 需求分析

1.1 开发背景

近年来，随着促进消费的政策落地，消费潜力不断释放，中国餐饮行业呈现新的发展态势。外卖行业作为餐饮行业子行业，依托电话、手机、网络的普及呈现高速发展的态势，其市场迅速扩大，外卖产业链逐步完善，餐饮外卖市场也逐步成熟。截至 2023 年 6 月，我国网上外卖用户规模达 5.35 亿人，较 2022 年 12 月增长 1372 万人，占网民整体的 49.6%。与此同时，外卖抽成居高不下，佣金费率普遍 18%-26% 之间，餐饮商家面临只有订单量却无利润的困境，打造商家专属的外卖系统迫在眉睫。2022 年 2 月国家发展改革委、财政部、人力资源社会保障部、等十四部门联合印发《关于促进服务业领域困难行业恢复发展的若干政策》，积极引导外卖等互联网平台企业进一步降低餐饮行业商户服务费标准，降低相关餐饮企业经营成本。基于此，本项目面向餐饮行业商户开发了订餐外卖系统，旨在提供方便快捷、无抽成的订餐服务。

1.2 市场分析

1.2.1 目标用户

主要客户为大中小型餐饮商户、订餐顾客、送餐员。

1.2.2 竞品分析

表 1 竞品分析

指标	我们	美团	饿了么
佣金抽成	无	18%-26%	15%-25%
个性化定制	有	无	无

与当前市场中的主流外卖软件进行对比，可以得出本项目具有较强的价格优势。

1.3 作品介绍

1.3.1 主要功能

系统将包含三个主要用户角色：顾客、餐厅员工和送餐员。顾客可以通过企业内联网使用系统，实现浏览菜单、下单订餐、支付、评价等功能。餐厅员工可以管理菜单、查看销售情况、接受订单并进行备餐、请求送餐等操作。送餐员可以确认送餐单并将餐品送达目的地。项目内容概要包括菜单管理、订单处理、送餐流程等功能的设计和实现。系统将确保用户能够方便快捷地浏览菜单、下单、支付并及时获知订单状态。

1.3.2 优势和创新点

- (1) 系统设计简洁大方、功能齐全。
- (2) 系统基于 Gin 框架与 React 开发，性能较高，部署方便
- (3) 促进外卖平台生态良好发展。

第二章 概要设计

2.1 总体设计

本系统采用 MVC 架构模式，通过身份校验从而实现针对用户、员工、外卖员不同身份的不同页面，实现从下单到接单到配送的外卖全流程服务。下图为该系统的架构示意图。

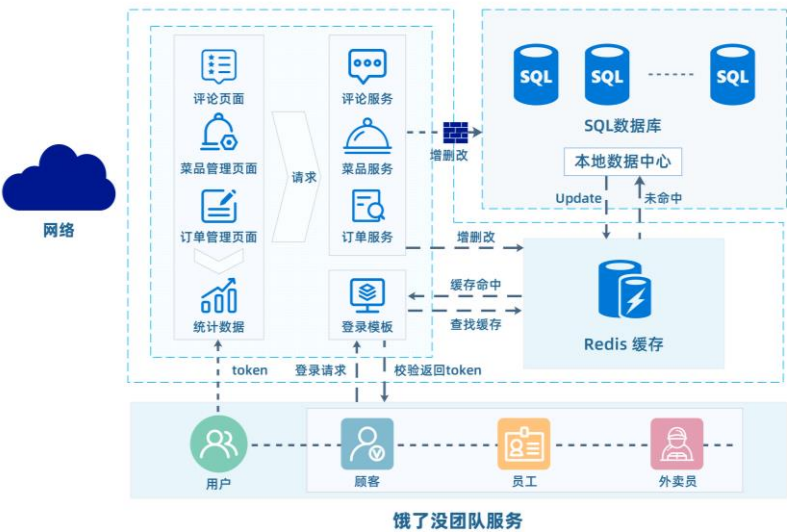


图 1 系统架构图

2.2 外卖点单流程



图 2 外卖点单流程

2.3 功能模块设计

2.3.1 用户管理模块

- 用户注册:** 允许用户注册系统账户, 提供必要的个人信息 (如用户名、密码、电话号码)。
- 用户登录:** 提供用户登录功能, 以验证用户身份, 进入系统, 并获取相关的权限和功能。
- 用户角色管理:** 管理用户角色, 如顾客、餐厅员工和送餐员, 并根据角色提供相应的权限。
- 用户信息编辑:** 允许用户编辑个人信息, 例如修改密码、更改电话号码等。

2.3.2 菜品管理模块

- 菜单管理:** 允许餐厅员工管理菜单, 包括添加新菜品、修改菜品信息、删除菜品等。
- 菜品查看:** 顾客可以查看餐厅的菜单, 浏览菜品的详细信息、价格和图片。

2.3.3 评论模块

评价订单：顾客可以为已送达的订单写评价，包括评分和文字评论。

查看评价：餐厅员工和送餐员可以查看已收到的评价，以了解顾客的反馈。

2.3.4 订单管理模块

订单下单：顾客可以选择菜品、指定送餐时间和地点，然后下单并支付。

订单接受：餐厅员工可以接受新订单，开始备餐，并更改订单状态为“备餐中”。

请求送餐：餐厅员工可以请求送餐员接单，将订单状态更新为“送餐中”。

订单状态追踪：顾客可以查看订单状态，包括“确认中”“备餐中”“送餐中”“已送达”。

订单历史：顾客、餐厅员工和送餐员可以查看以前的订单历史，包括订单的详细信息、时间和付款信息。

2.3.5 数据统计模块

销售概览：提供一段时间内的销售额、订单数量等基本统计信息。

销售排行：展示不同菜品的销售量和销售额排行。

客户活跃度：分析不同客户的订单数量，用于识别常客和新客户。

时段销售分析：分析一天中各个时段的销售情况，帮助调整备餐和人员安排。

第三章 详细设计

3.1 界面设计

3.1.1 顾客界面

顾客页面提供一个直观的界面以便顾客能够快速浏览菜单、下单订餐并跟踪订单状态。主要功能包括以下内容。

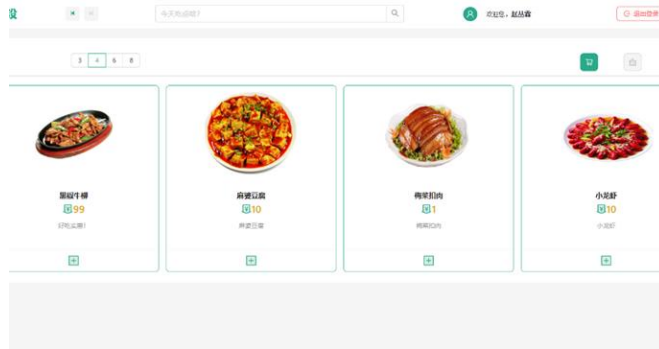


图 3 菜单页面



图 4 订单页面

菜单浏览：顾客进入系统后首先看到的是餐厅的菜单，每道菜品都配有名称、图片和价格等信息，以便顾客可以清晰地了解选择。

订餐流程：顾客可以通过简单的界面操作选择一个或多个菜品，并将它们添加到订单中。在订单页面，顾客可以确认所选菜品，指定送餐时间和地点，并选择支付方式完成订单。

订单状态追踪：顾客可以在页面上实时查看订单状态，包括订单的当前状态，如“确认中”、“备餐中”、“送餐中”以及“已送达”，以便顾客了解订单处理进度。



图 5 订单详情与评价

订单评价：一旦订单被标记为“已送达”，顾客将有权限为订单进行评价，提供反馈和建议，以便改善服务质量。

3.1.2 员工界面

员工页面提供一个高效管理菜单和订单的界面，使员工能够轻松地管理餐厅的运营。主要功能包括以下内容。

菜单管理：员工可以添加、删除、修改和查询菜单项的功能，确保菜单信息的及时更新和准确性，包括菜名、图片和价格等。

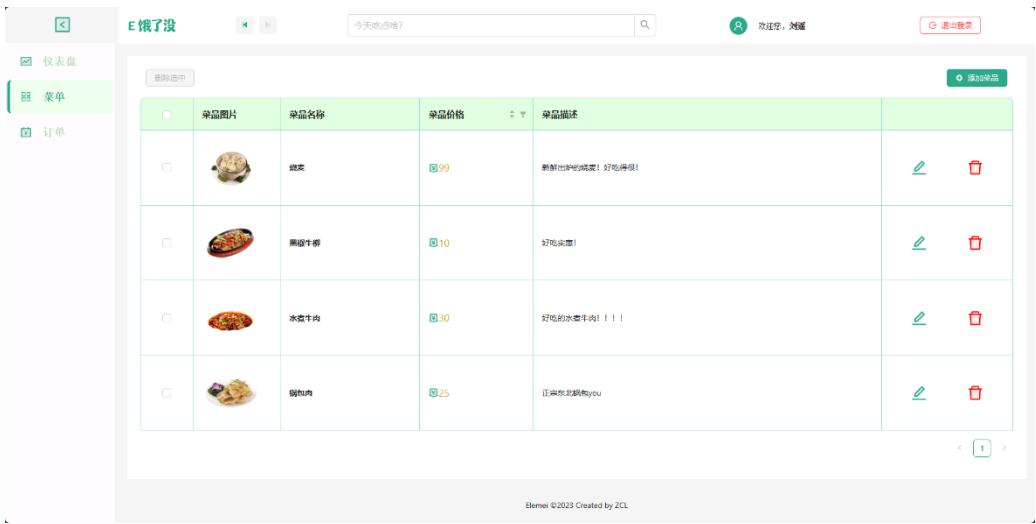


图 6 员工菜单管理页面

销售情况查看：员工可以查看之前的销售情况，包括订单的详细信息、时间和付款信息，以便进行销售数据的分析和统计。

订单处理：员工可以接受顾客支付的订单，并开始备餐。一旦备餐准备完毕，员工可以请求送餐，将订单分配给送餐员。订单状态将更新为“送餐中”，并显示送餐员的姓名。

3.1.3 送餐员界面

送餐员页面旨在提供一个简单直观的界面，以便送餐员能够快速确认送餐单并将餐品送达目的地。主要功能包括以下内容。

选择确认送餐单：送餐员登录后能够查看待送达的订单列表，并根据送餐时间和地点选择确认送餐单。确认后，送餐员开始配送餐品。

确认已送达：送餐员在将餐品送达给顾客后，可以在页面上确认订单已送达，以便更新订单状态和通知餐厅和顾客。



图 7 送餐员查看订单界面

3.2 数据库设计

3.2.1 ER 图设计

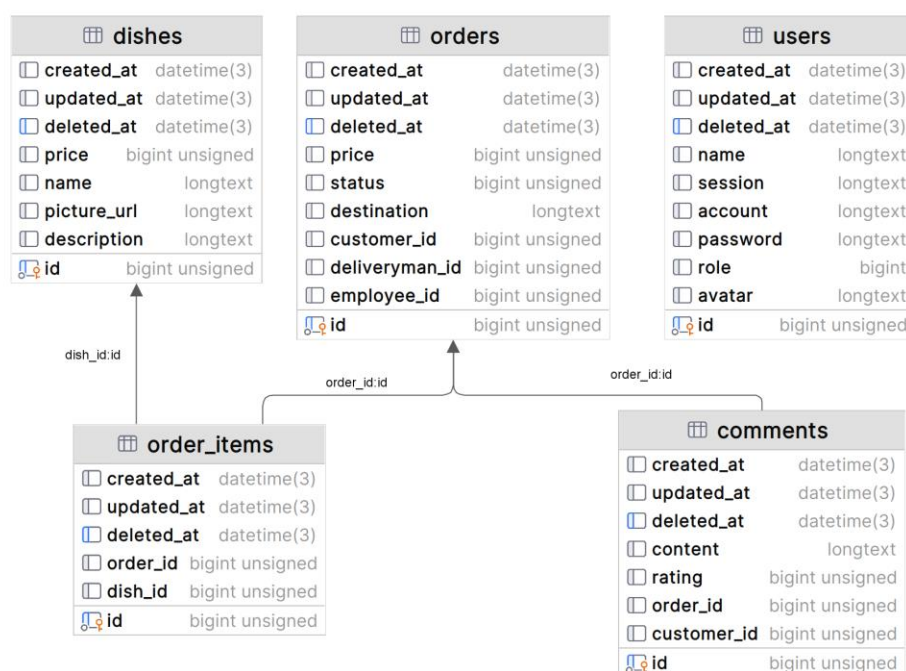


图 8 数据库 ER 图

3.2.2 表设计与实体间关系

(1) 表设计

customers: 存储顾客信息的表，包括顾客的 ID、姓名、登录会话、账户信息、密码等。还包括一个指示角色的字段。

deliverymen: 存储送餐员信息的表，包括送餐员的 ID、姓名、登录会话、账户信息、密码等。同样也有一个指示角色的字段。

dishes: 存储菜品信息的表，包括菜品的 ID、名称和价格等。

comments: 存储顾客对菜品的评论信息的表，包括评论的 ID、内容、评分等。还有外键指向菜品和顾客表，以关联评论和相关菜品、顾客。

orders: 存储订单信息的表，包括订单的 ID、名称、价格、状态、送达时间等。外键关联了顾客、送餐员和菜品表，以表示订单的相关信息。

stuffs: 存储餐厅员工信息的表，包括员工的 ID、姓名、登录会话、账户信息、密码等。同样也有一个指示角色的字段。

users: 存储用户信息的表，包括用户的 ID、姓名、登录会话、账户信息、密码等。这个表可能是一个通用的用户表，供多个角色使用。

(2) 实体间关系

1) 一对多关系

customers 和 orders: 一个顾客可以有多个订单，但一个订单只能属于一个顾客。

deliverymen 和 orders: 一个送餐员可以配送多个订单，但一个订单只能由一个送餐员配送。

dishes 和 comments: 一个菜品可以被多个评论，但一个评论只能属于一个菜品。

customers 和 comments: 一个顾客可以发表多个评论，但一个评论只能由一个顾客发表。

stuffs 和 orders: 一个餐厅员工可以处理多个订单，但一个订单只能由一个餐厅员工处理。

2) 多对多关系

orders 和 dishes: 一个订单可以包含多个菜品，一个菜品也可以包含在多个订单中。

3.2.3 数据库访问

Repository 层是项目的数据库访问层。这一层通过为每个数据库表创建对应的 repository 文件（如 user.go, order.go, dish.go 等）来组织代码，确保了数据操作的逻辑清晰和高度模块化。使用 GORM 库作为 ORM 工具，提高了开发效率，简化了数据库操作的复杂性。此外，Repository 层不仅实现了基本的 CRUD（创建、读取、更新、删除）操作，还提供了更为复杂的查询逻辑（如根据用户 ID 查询订单），以支持业务中各种多样化和定制化的数据需求。

其中，order_items 表建立了订单与菜品之间的多对多关系，使每个订单可以包含多个菜品，同时每种菜品也可能出现在多个订单中，并为灵活实现菜品和订单处理逻辑提供了支持。

3.3 关键技术

3.3.1 基于 React 与 ElementUI 的前端设计

使用 React 和 ElementUI 进行开发具有显著的优势。React 的声明式编程模型、组件化架构和虚拟 DOM 技术，不仅提高了应用的性能和开发效率，还确保了代码的可读性和可维护性。React 强大的生态系统则为开发提供了丰富的支持。另一方面，ElementUI 通过提供一系列设计优雅、响应式的预制组件，极大地加速了 UI 开发过程，保证了界面的一致性和专业性。基于 Gin 与 Gorm 的后端架构

3.3.2 基于 Gin 与 Gorm 的后端架构

结合使用 Gin 和 Gorm，能够快速构建出结构清晰、易于维护且性能卓越的后端应用。Gin 是一个高性能 HTTP Web 框架，具有轻量级且易于部署的特性，通过简化路由和中间件的设置，加速了 RESTful API 的开发和部署过程。Gorm 是一个富有特色的 ORM 库，它为数据库操作提供了一种更加直观和高级的抽象，支持广泛的数据库系统，并且能够自动处理繁琐的数据库交互和映射工作。

第四章 测试报告

4.1 各模块功能测试

对用户模块、菜品模块、评论模块、订单模块的接口进行测试。逐一测试：用户注册、用户登录，创建菜品、删除菜品、获取菜品、更新菜品，创建评论、获取评论，创建订单、更新订单、获取订单。通过响应结果可见各接口均能正常工作。

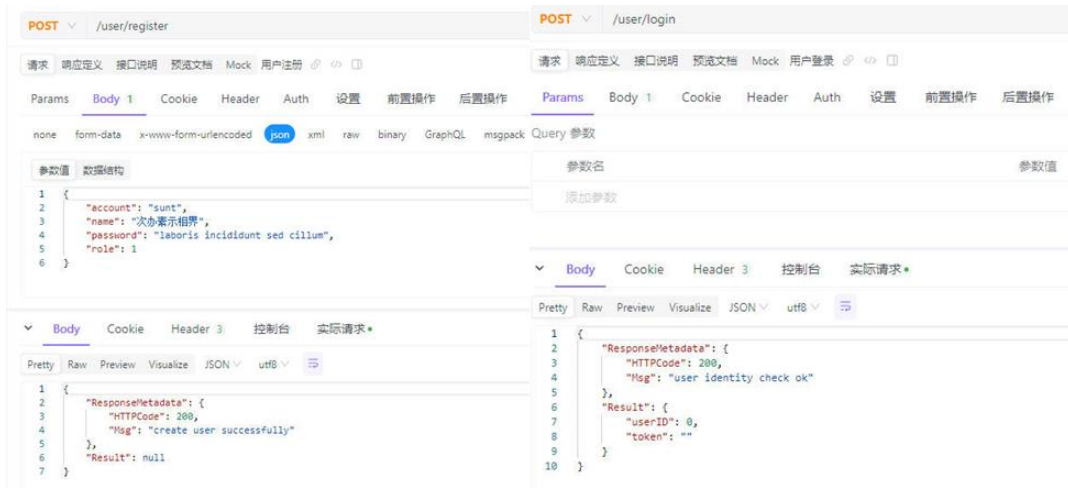


图 9 用户注册、用户登录测试结果

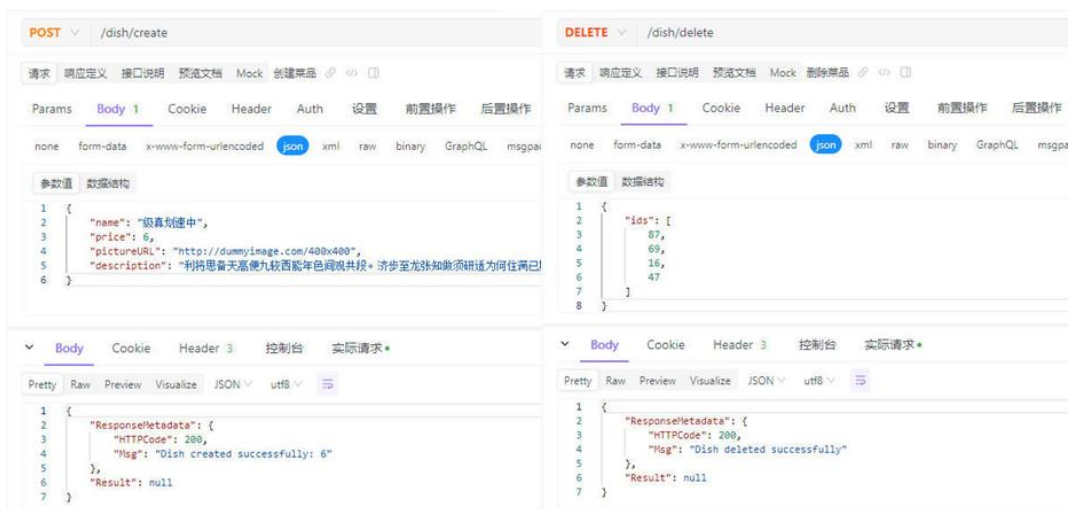


图 10 创建菜品、删除菜品测试结果

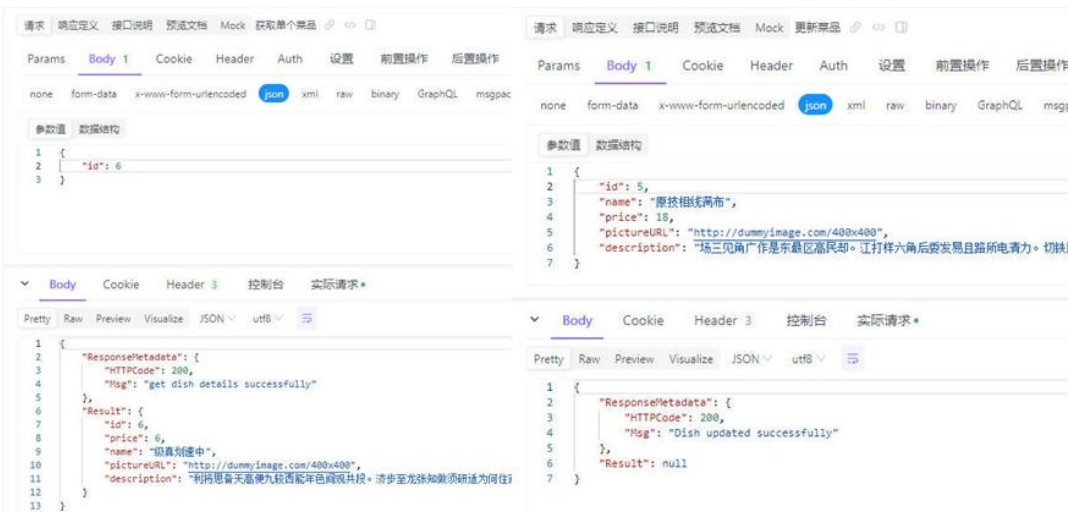


图 11 获取菜品、更新菜品测试结果

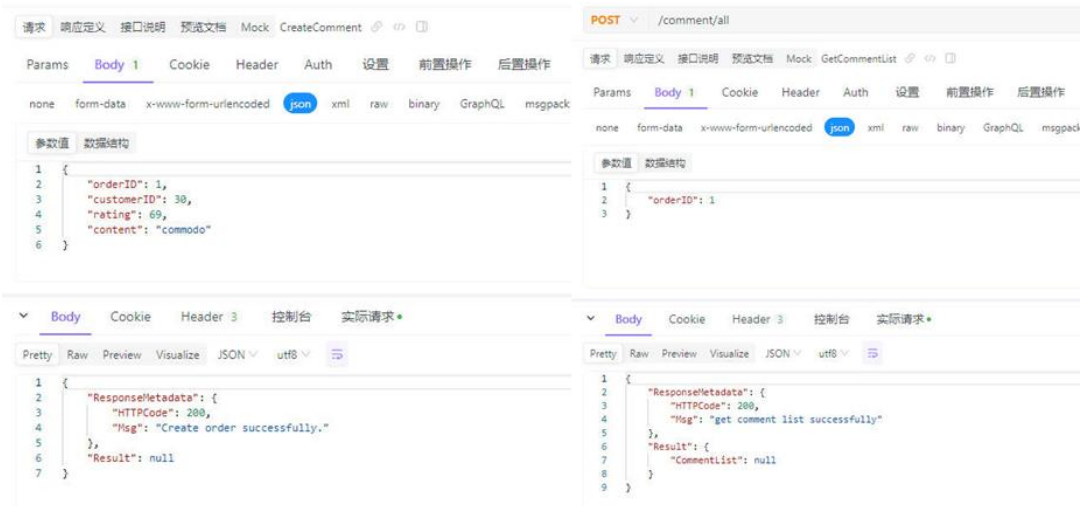


图 12 创建评论、获取评论测试结果

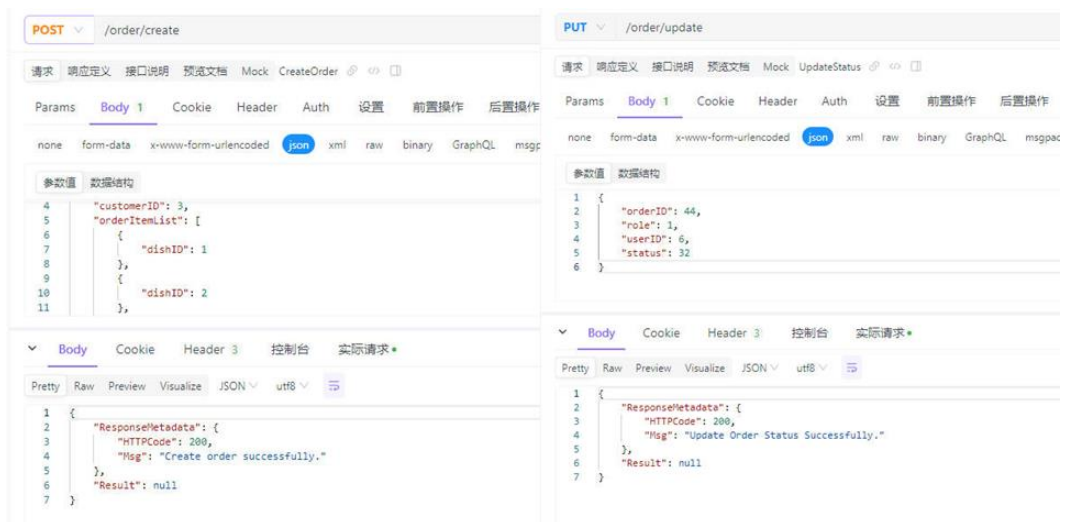


图 13 创建订单、更新订单测试结果

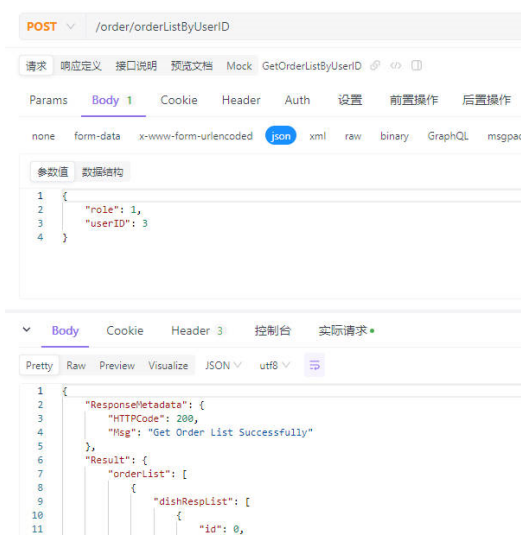


图 14 获取订单列表测试结果

4.2 性能测试

为了测试系统在高并发场景下的性能，我们以创建菜品接口为例，测试在多线程情况下的平均耗时。结果如下：



图 15 接口性能测试

结果显示，在四线程批量创建菜品的情况下，平均每条线程耗时 16s，每次创建菜品耗时 4ms，且所有测试用例均通过。这说明在高并发场景下系统仍能稳定运行，且速度较为可观。

第五章 安装及使用

5.1 配置前端环境

npm 版本 9.8.1 js 包管理工具

在项目根目录下执行：

```
1. npm install -g serve
2. cd web
3. serve -s build
```

5.2 配置后端环境

服务器环境：Centos 8，提供了稳定的 Linux 操作系统环境。

Go：使用版本 1.20.7，Go 语言以其高性能和并发支持，成为后端开发的优选。

Gin：版本 v1.9.1，一个高效的 Go Web 框架，用于快速开发 HTTP API。

Gorm：版本 v1.5.1，是一个优秀的 Go ORM 库，简化数据库操作。

MySQL：版本 8.1，作为数据库管理系统，用于存储和管理数据。

在项目根目录下执行：

```
1. go mod tidy
2. cd cmd
3. go run main.go
```

第六章 项目总结

本项目中，用户界面层的构建实践了现代 Web 应用开发的理念，利用 React 前端框架与后端的 RESTful API 相结合。这一层作为用户与系统交互的直接接口，通过其高效的数据交互和界面呈现机制，提升了用户体验。React 的使用使得用户界面的开发变得模块化和组件化，每个 React 组件都可以独立地开发和测试，提高了代码的可复用性和维护性。此外，项目中 RESTful API 的设计为前端和后端的交互提供了一个清晰和标准化的通信协议，这种设计支持了异步数据加载和页面的即时更新，使得用户界面能够快速反应用户操作，同时确保了数据的实时性和准确性。在项目的开发过程中，前后端团队能够独立工作，仅通过定义好的接口进行交互，这种解耦合的结构不仅提高了开发效率，也简化了后期的维护工作。

参考文献

【请按照标准参考文件格式填写】