



**本科毕业设计**

|  |
| --- |
| **基于微服务架构的食品商城系统** |
|  |

**庄丰鸣**

**201525060230**

|  |  |
| --- | --- |
| 指导教师 | **冯健昭 讲师** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学院名称 |  | **数学与信息学院** | 专业名称 |  | **软件工程** |
| 论文提交日期 |  | 2019年4月15日 | 论文答辩日期 |  | 2019年5月11日 |

# 摘 要

随着IT行业的飞速发展,各公司采用的传统系统架构,已经不能满足日益增长的业务需求拓展,逐渐变得非常臃肿,急需进行升级改造。前些年是通过采用垂直系统架构进行改造,近几年又涌现出了微服务系统架构。通过向微服务系统架构进行改造之后的系统,拥有非常良好的可拓展性以及优秀的性能表现。

本食品商城系统的后台根据微服务架构思想进行搭建,主要采用目前在微服务架构实施上较为全面的框架SpringCloud进行开发。本系统主要有服务注册与服务发现中心、统一网关和业务服务三大部分，业务服务包含了基础服务、购物车服务、食品服务和交易服务四个单服务。基础服务中实现了用户操作功能、系统监控等功能；食品服务中实现了食品和食品类目相关功能；交易服务中实现了订单信息、交易支付相关功能。前端部分均采用Vue+NPM+webpack进行开发,都是当前前端开发的主流技术。

由于系统采用微服务架构搭建,服务数量急剧上升,为部署和运维带来了极大的挑战。系统一般会搭配Jenkins构建框架和Docker虚拟化技术,实现运维的智能化和一键化部署。

关键词：微服务 Vue SpringCloud

# Food mall based on micro-service architecture

Zhuang Fengming

(College of Mathematics and Information, South China Agricultural University, Guangzhou 510642, China)

**Abstract:** With the rapid development of the IT industry, the traditional system architecture adopted by various companies has been unable to meet the growing demand for business, and has gradually become very bloated and urgently needs to be upgraded. In the past few years, it has been transformed by adopting a vertical system architecture. In recent years, the micro-service system architecture has emerged. Through the transformation of the microservices system architecture, the system has very good scalability and excellent performance.

The background of the food mall system is built according to the micro-service architecture idea, mainly using the current SpringCloud framework, which is more comprehensive in the implementation of the micro-service architecture. The system mainly includes three parts: service registration and service discovery center, unified gateway and business service. The business service includes four basic services: basic service, shopping cart service, food service and transaction service. In the basic service, functions such as user operation function and system monitoring are realized; food service and food category related functions are realized in the food service; order information and transaction payment related functions are realized in the transaction service. The front-end part is developed using vue+npm+webpack, which is the mainstream technology for front-end development.

As the system is built with a micro-service architecture, the number of services has risen sharply, posing great challenges for deployment and operation and maintenance. It is generally used with Jenkins build framework and Docker virtualization technology to realize intelligent and one-click deployment of operation and maintenance.

**Key words:** micro-service Vue SpringCloud

# 目 录

[1 前言 1](#_Toc7078121)

[1.1课题的研究背景 1](#_Toc7078122)

[1.2国内外研究现状 1](#_Toc7078123)

[1.3 研究的目的意义 1](#_Toc7078124)

[1.4 论文的组织架构 2](#_Toc7078125)

[2 基于微服务架构的食品商城的技术基础 3](#_Toc7078126)

[2.1 gdpj后台系统 3](#_Toc7078127)

[2.1.1 eureka-service 服务注册中心 3](#_Toc7078128)

[2.1.1.1基于SpringCloud Eureka的服务注册与服务发现组件 3](#_Toc7078129)

[2.1.1.2 SpringBoot快速搭建服务统一服务注册中心 3](#_Toc7078130)

[2.1.2 api-gateway 统一网关服务 4](#_Toc7078131)

[2.1.2.1基于SpringCloud Zuul的统一网关组件 4](#_Toc7078132)

[2.1.2.2使用SpringBoot快速搭建统一api网关服务 4](#_Toc7078133)

[2.1.3 四个核心业务服务 4](#_Toc7078134)

[2.1.3.1 基于SpringCloud Eureka进行服务注册与发现 5](#_Toc7078135)

[2.1.3.2 基于SpringCloud Feign的声明式RPC服务调用 5](#_Toc7078136)

[2.1.3.3 单个服务采用SpringBoot+Mybatis+SpringMVC快速搭建 5](#_Toc7078137)

[2.2 gdpj-admin 食品管理前端系统 5](#_Toc7078138)

[2.2.1 渐进式框架Vue 6](#_Toc7078139)

[2.2.2 基于npm+webpack构建 6](#_Toc7078140)

[2.2.3 Element-UI桌面端组件库 6](#_Toc7078141)

[2.3 gdpj-web 商城前端系统 6](#_Toc7078142)

[2.3.1 Vuex程序开发状态管理 7](#_Toc7078145)

[2.3.2 iView UI组件库 7](#_Toc7078146)

[3 基于微服务架构的食品商城系统概要设计 8](#_Toc7078147)

[3.1 系统功能需求分析 8](#_Toc7078148)

[3.2 系统总体架构图设计 9](#_Toc7078149)

[3.3 数据库ER图设计 10](#_Toc7078150)

[3.4 系统开发平台和工具 10](#_Toc7078152)

[3.4.1 系统开发平台 10](#_Toc7078153)

[3.4.2 开发工具 10](#_Toc7078154)

[3.5 系统运行环境 10](#_Toc7078155)

[3.5.1 软件环境 10](#_Toc7078156)

[3.5.2 硬件环境 10](#_Toc7078157)

[4 基于微服务架构的食品商城系统详细设计 11](#_Toc7078158)

[4.1 功能结构分析 11](#_Toc7078159)

[4.2 数据库表结构设计 12](#_Toc7078160)

[4.3 系统平台技术架构 16](#_Toc7078161)

[4.4 相关主要工具包 17](#_Toc7078162)

[4.5 项目工程结构 17](#_Toc7078163)

[4.6 配置文件的定义与说明 20](#_Toc7078164)

[5 基于微服务架构的食品商城系统实现 21](#_Toc7078165)

[5.1 系统部分功能 21](#_Toc7078166)

[5.1.1 食品后台管理系统部分 21](#_Toc7078167)

[5.1.1.1 各服务运行状态监控展示 21](#_Toc7078168)

[5.1.1.2 食品管理模块展示 22](#_Toc7078169)

[5.1.1.3 食品类目管理模块展示 24](#_Toc7078170)

[5.1.1.4 订单模块展示 26](#_Toc7078171)

[5.1.1.5 交易模块展示 27](#_Toc7078172)

[5.1.1.6 系统运行日志模块展示 28](#_Toc7078173)

[5.1.2 商城部分 28](#_Toc7078174)

[5.1.2.1 商城首页展示 29](#_Toc7078175)

[5.1.2.2 用户注册登陆功能展示 30](#_Toc7078176)

[5.1.2.3 食品搜索功能展示 31](#_Toc7078177)

[5.1.2.4 食品详细信息页面展示效果 31](#_Toc7078178)

[5.1.2.5 购物车管理和下单流程效果展示 32](#_Toc7078179)

[5.1.2.6 模拟在线支付功能展示 33](#_Toc7078180)

[5.1.2.7 个人收货地址管理展示 34](#_Toc7078181)

[6 总结与展望 35](#_Toc7078182)

[6.1 总结 35](#_Toc7078183)

[6.2 展望 35](#_Toc7078184)

[参考文献 36](#_Toc7078185)

[致谢 37](#_Toc7078186)

# 1 前言

本系统主要目的是研究基于微服务架构的服务端后台搭建,服务端主要使用了Spring Cloud的微服务框架。近年来,随着IT行业高速发展,越来越多的互联网公司的系统出现了臃肿、性能瓶颈等问题。由于微服务架构的轻量、服务化、可拓展性、耦合度低等特点,许多公司决定对系统进行微服务化重构。开源社区中也涌现出许多优秀的微服务架构相关的组件,其中,Spring Cloud是一个微服务实施的综合性框架,它的优点在于集成了众多组件,诸如 Netflix Eureka、Netflix ribbon、Netflix feign、Netflix hystrix、Netflix zuul、Netflix config等等,使得开发者无需进行太多的配置信息就能轻而易举的使用这些组件,非常适合用来快速搭建一个微服务架构的服务端。同时,Spring Cloud完全支持Springboot,单个服务可以使用Springboot进行快速开发。

## 课题的研究背景

现今IT行业飞速发展,用户需求以及用户量激增,系统不断迭代升级,传统的单体应用系统架构只会变得越来越臃肿、性能差、难以升级,稍微一点改动就会影响整个系统,微服务架构的出现就是为了解决这些问题。近年来,社区中不断有新的技术贡献,使得微服务架构逐渐成熟,本人就是在这个背景之下,探索微服务架构的组成及其优点。

## 国内外研究现状

近年来,随着微服务架构技术的不断发展壮大,国内也涌现出一些比较优秀的技术。最火热的当属于Alibaba团队开源的Dubbo,它是一款服务治理核心框架,主要承担微服务架构生态中的服务注册与服务发现的角色, 目前广泛应用于阿里巴巴集团的各成员站点以及国内许多互联网公司中。是目前为止, 国内影响力比较大的一款框架。

在国外,在微服务架构相关技术的发展上,个人认为还是蛮领先的。比较著名的当属?Netflix开源的一整套微服务架构套件,同时它也是另外一款微服务框架SpringCloud的核心。这两款产品在国外社区中的热度是非常高的,同时开发的活跃度也是非常高,说明其前景是非常可观的。

## 1.3 研究的目的意义

对于微服务架构技术本身来讲,在微服务架构广泛应用的环境下,通过实践去探索它与传统系统架构的区别、微服务架构带来的好处、微服务架构系统的组成、微服务架构实施相关的技术、微服务架构系统开发流程等。 对于个人角度来讲,通过以上种种实践,对与系统开发过程中的架构设计、技术选型、编码等过程都会有较大的提升。在现今微服务架构发展如此迅速的情况下,也让自身能够跟上技术发展的潮流,去研究一些较为新颖的和高效的技术,拓宽自身的技术视野。

## 1.4 论文的组织架构

本文内容一共分为六章节，按照 背景--技术基础--概要设--详细设计--功能展示--个人总结的顺序展开。

其中第一章节讲述的是本食品商城所使用的微服务系统架构的使用背景、国内外发展趋势；

第二章节讲述的是整个食品商城系统技术基础，包含使用到的技术类目以及部分技术原理的分析；

第三章节讲述的是整个食品商城的概要设计部分，包含系统整体架构设计、ER图设计、系统需求分析、开发平台、开发工具等；

第四章节讲述的是食品商城的详细设计，包含了功能结构分析、数据库表设计、项目工程目录结构、系统平台技术架构、配置文件等；

第五章节讲述的食品商城的功能展示，包含商家管理系统gdpj-admin和商城系统gdpj-web两大部分，主要展示了系统监控、食品管理、订单管理、商城食品列表、商城食品详情、用户购物车等页面。

# 2 基于微服务架构的食品商城的技术基础

## 2.1 gdpj后台系统

这部分是整个商城的整个后台,采用微服务架构划分,主要分为base基础信息服务、foodstuff食品信息服务、shoppingcart购物车信息服务、transaction交易信息服务、服务注册与服务发现中心、api-gateway统一网关服务这6个服务,服务之间采用SpringCloud Feign提供的声明式api进行互相调用。base基础信息服务主要提供用户注册、用户登陆、用户信息存储、用户信息查询、系统日志存储、系统监控信息存储等功能;foodstuff食品信息服务主要提供食品信息存储、食品信息查询、食品类目管理等功能;shoppingcart购物车信息服务主要提供用户购物车信息存储、购物车信息查询等功能、transaction交易信息服务主要提供用户订单信息存储、订单信息查询、交易记录存储、交易信息查询等功能;服务注册与服务发现中心主要提供各服务的信息注册、服务续租、服务发现等功能; api-gateway统一网关服务是整个后台系统的入口,主要提供url拦截过滤、请求映射、监控等功能。

### 2.1.1 eureka-service 服务注册中心

该服务模块是整个微服务系统架构中的服务注册与服务发现中心,其核心组件是Netflix Eureka ,一个云端服务发现，一个基于 REST 的服务，用于服务定位，以实现系统中间层服务发现和故障转移的组件。整个系统的服务都注册在该中心上,同时,它本身也是一个服务,可以通过水平拓展通过多个注册中心之间相互注册以提高系统的稳定性,当其中一个服务注册中心宕机时,还有其它的服务注册中心继续提供服务,不至于整个系统瘫痪。

2.1.1.1 基于SpringCloud Eureka的服务注册与服务发现组件

本食品商场系统采用SpringCloud Eureka来开发服务注册与服务发现中心。其核心组件是Netflix Eureka。整个中心主要的提供的功能有:服务注册、服务续约、服务发现、服务下线。其中服务续约主要是通过定时任务来实现,每隔一段时间向服务发送心跳,如果服务还在线,就进行续约,如果发现服务已经下线,就对其进行清理下线。

2.1.1.2 SpringBoot快速搭建服务统一服务注册中心

SpringCloud对SpringBoot非常的友好,通过SpringBoot,可以在短时间内快速开发出一个服务注册中心与服务发现中心。只要引入spring-cloud-starter-eureka-server这个模块,其次加上一些简单的配置信息,最后在项目的启动文件中,加上一行注解@EnableEurekaServer,即可完成一个服务注册与服务发现中心的开发。在开发其它服务的时候,只需要在配置文件中指定服务注册与服务发现中心的url,就可以实现服务注册与服务的发现功能。

### 2.1.2 api-gateway 统一网关服务

该服务模块是整个微服务系统的“门户”,是所有请求的入口。主要实现请求路由、请求过滤以及一些流量控制和安全校验功能。在微服务架构的系统中,各个服务之间独立开发,使用不同的端口甚至不同的域名。如果开发过程中,这些api接口不进行统一的处理,对使用者需要暴露各个服务的直接域名和端口,会使得web端或者移动端在调用的时候,会增加其复杂度同时对于各服务而言也有一定的危险性。为了解决这种情况,使用统一网关服务就发挥了作用,通过统一api网关,对外暴露统一的域名以及端口,外部进行调用的时候,只知道各服务的名字,而不知道也不需要知道各服务具体的地址和端口。在请求到达api网关时,再由api网关对服务名进行识别,将请求转发到对应的服务中去处理。同时,使用api网关还有另外一个好处,就是可以将一些统一的过滤和拦截逻辑放在其中例如用户是否登陆的判断,在网关这一层进行统一的判断和过滤处理,如果不满足条件则直接返回结果,就无需到各服务中进行判断和过滤。

2.1.2.1 基于SpringCloud Zuul的统一网关组件

本系统的api-gateway统一网关服务,是基于SpringCloud Zuul进行开发的,其核心是Netflix Zuul,是一个开源的api gateway组件。其主要承担整个微服务架构系统中统一网关的角色。通过网关,实现全请求的转发和过滤。用户可以自定义filter,在其中实现自己任何想要的逻辑。同时,网关统一接收请求,然后根据服务名进行分发到具体的服务中去处理。Zuul中还配备有断路器Hystrix以及负载均衡器ribbon,在一定程度上提升了整个微服务架构系统的可用性和稳定性。

2.1.2.2 使用SpringBoot快速搭建统一api网关服务

通过搭配使用SpringBoot可以快速开发出一个统一网关服务,在项目中引入Netflix Zuul的依赖,同时在配置文件中配置上服务注册与服务发现中心的地址,以便可以从注册中心拉取各服务的基本信息,供后续请求路由转发使用。在项目的启动文件上,加上@EnableZuulProxy注解,即可开启zuul的网关代理。过滤功能的实现,通过继承ZuulFilter,重写其中的run方法,在其中添加过滤逻辑即可实现一个统一的请求过滤器。

### 2.1.3四个核心业务服务

这几个模块是整个系统的业务服务,负责提供所有业务和系统相关的服务接口,有基础信息服务、食品信息服务、购物车服务、交易服务。其中基础信息服务包含:系统日志查询接口、系统监控信息查询接口、用户信息管理接口;食品信息服务包含:食品管理接口、食品类目管理接口;购物车服务包含:用户购物车管理接口;交易服务包含:订单管理接口、交易支付接口、交易记录管理接口。这几个服务统一注册到eureka-server这个服务注册中心上。服务之间的接口调用使用SpringCloud Feign声明式调用。食品服务部署了两个服务,通过Netflix Ribbon进行负载均衡。在各服务的稳定性和可用性上,使用了Netflix Hystrix熔断器,当请求超时或流量过大时候进行重试或者熔断,以保证系统的整体可用性。

2.1.3.1 基于SpringCloud Eureka进行服务注册与发现

这些业务模块,统一使用SpringCloud Eureka进行服务注册与服务发现,通过eureka。client。servieUrl。defaultZone配置参数指定服务注册与服务发现中心的地址,其次在项目的启动器中加上@EnableDiscoveryClient注解,开启服务注册与服务发现功能,即可成为微服务系统中的一份子。

2.1.3.2 基于SpringCloud Feign的声明式RPC服务调用

当各服务之间,需要进行接口之间的互相调用,这时候就需要使用的rpc工具。这里使用的Feign就是一个提供远程接口调用的工具,它封装了http请求,并且与SpringCloud Eureka紧密结合,通过配置注册中心的地址,就可以自动从Eureka服务中心拉去各个服务的地址列表。用户在使用的时候,只需要简单的进行method声明,即可以调用其它服务的接口了。声明过程,创建一个接口,通过@FeignClient(serverName)注解,声明远程接口的服务提供者,其次在接口方法中直接编写需要调用的方法,通过@GetMapping(restUrl)注解标示接口restApi地址,即可使用。

2.1.3.3 单个服务采用SpringBoot+Mybatis+SpringMVC快速搭建

单个业务服务,使用SpringBoot+Mybatis+SpringMVC三个框架集成快速开发。springboot是一款自动化配置程度很高同时集成web服务器tomcat的一款优秀的JAVA后端框架,Mybatis是一个轻量级的ORM框架,通过简单的一些配置,既可以实现数据库的映射关系,同时提供注解和xml两种方式,由用户自己选择sql的编写方式。SpringMVC是一款web mvc框架,用来开发web接口。通过者三个框架的集成,开发单个服务是非常高效率的。

## 2.2 gdpj-admin 食品管理前端系统

这是食品商城的后台管理系统的前端项目。供商家和管理员使用。项目使用目前比较主流的一个前端框架Vue进行编写,这是一套用于构建用户界面的渐进式框架;UI控件使用饿了么开源的Element-UI库, 一套基于 Vue 2.0 +的桌面端组件库,对于桌面端应用开发有较好的支持;可视化图表使用比较热门的ECharts, 一个使用 JavaScript 实现的开源可视化库，可以非常流畅的运行在移动端和PC端上，兼容了目前绝大多数的浏览器（如 IE8/9/10/11，Chrome，Firefox，Safari等），它底层依赖的是轻量级的矢量图形库 ZRender。gdpj-admin系统,主要提供服务监控、食品管理、食品类目管理、订单管理、支付记录、系统日志这6个页面。

### 2.2.1 渐进式框架Vue

Vue是一套用于构建前端用户界面的渐进式框架。与其它大型JavaScript框架不同的是，Vue 被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，不仅易于上手，同时还便于与第三方库或既有项目整合。（摘自vue官网）

### 2.2.2 基于npm+webpack构建

NPM全称nodejs package manage,是nodejs下的一个依赖包管理工具,与maven的作用极其相似。 npm 为团队打开了连接整个 JavaScript 世界的一扇大门（摘自npm官网）。包含超过 600000 个 包（package） （即javascript代码模块）。来自各大洲的开源软件开发者使用 npm 互相分享和借鉴。同时，包的结构使您能够轻松跟踪依赖项和版本。

Webpack是前端的一个项目构建工具,使用webpack,可以实现前端项目开发的模块化。本质上，webpack 是一个JavaScript 程序的静态模块打包工具。当 webpack 处理应用程序时，它会在内部构建一个依赖图，此依赖图会映射项目所需的每个模块，并生成一个或多个 bundle。

将npm+webpack搭配使用,使得开发一个前端项目变得十分方便,同时效率也更加高了。

### 2.2.3 Element-UI桌面端组件库

这是饿了么开源的一套ui组件库,它与Vue是天生一对,从一开始设计的初衷就是为vue提供ui支持。同时它是桌面端优先的一套ui组件,非常适合用来开发食品商城后台系统的前端系统。Element-UI是一套为开发者、产品经理和设计师准备的基于 Vue 2.0 +的桌面端UI组件库。

## 2.3 商城前端系统

这是商城的前端系统,供用户使用。项目同样使用了Vue框架进行开发。该项目主要包含了:商城主页、食品列表、食品详情展示、用户登陆、用户注册、用户信息、用户地址、用户购物车详情、用户订单列表、订单结、支付成功这11个页面。

### 2.3.1 Vuex程序开发状态管理

Vuex是一个专门为vue项目提供的开发状态管理工具,用于存储和管理一些项目中的公共状态和变量。Vuex 是一个专为 Vue.js 应用程序开发的状态管理模式。它对所有组件的状态变量信息进行集中存储和管理并进行应用。

Vuex主要有以下3个部分: state，驱动应用的数据源(可理解为程序中的变量); view，以声明方式将 state 映射到视图; actions，响应在 view 上的用户输入导致的状态变化（可以理解成一个监控方法）。

### 2.3.2 iView UI组件库

iView是与Vue配套的UI库中,组件最为丰富的一个;提供了Grid栅格、Layout布局、Font字体、Button按钮、Table表格、Icon图标、Form表单、Checkbox多选框、DataPicker日期选择器、Upload上传、Message消息、Modal对话框等组件。

# 3 基于微服务架构的食品商城系统概要设计

## 系统功能需求分析

## （1）基础服务(base-service)需求:

1. 用户注册功能,此需求有两部分,第一部分根据手机号码判断用户是否存在,如果用户存在,则返回错误信息提示,若是新用户,则返回验证码;第二部分是个人信息的填充,需要用户名、用户密码、用户类型、手机号码四个信息。
2. 用户登陆功能,需要手机号码与用户密码两个信息,通过手机号码和密码去做校验。其中密码使用MD5加密传输保证安全性。校验过程,首先通过手机号码判断用户是否存在,若不存在则返回提示信息,若存在则检验密码是否正确,通过MD5加密后的字符串进行判断。
3. 系统日志管理功能，此功能包含了日志收集、日志查看两个点，对外开放日志接口，其它服务只需要通过rpc进行日志的操作。
4. 系统监控功能，通过调用Springboot提供的原生接口，获取各个服务的系统运行信息，进行汇总，提供前端展示。

（2）食品服务（foodstuff-service）需求:

1. 食品添加、修改、删除功能，通过传入食品名字、类目id、库存、原产地、价格等参数，进行添加。
2. 食品查询功能，通过食品名字进行模糊查询。
3. 食品类目添加、修改、删除功能，通过传入类目名称、类目描述等参数，进行添加。

（3）购物车服务(shoppingCart-service)需求：

* 1. 用户添加商品进个人购物车功能，通过传入食品id、数量、购买时金额等参数，进行添加。
  2. 个人购物车详情查看，通过传入用户id，进行关联查询，返回用户购物车详情。

（4）交易服务(transaction-service)需求：

1. 订单新增功能，通过传入订单总金额、用户id、下单时间等参数，进行添加。
2. 用户订单详情查看功能，通过传入用户id，通过关联食品表和订单表，返回订单详情。

## 3.2 系统总体架构图设计

系统总体架构设计分为三大部分，服务注册与服务发现中心、业务服务区和统一网关。各业务服务统一注册到服务注册与服务发现中心，外部请求会首先进入到统一网关处，由网关系统进行请求过滤和解析，然后进行分发到具体的业务服务单元去处理。

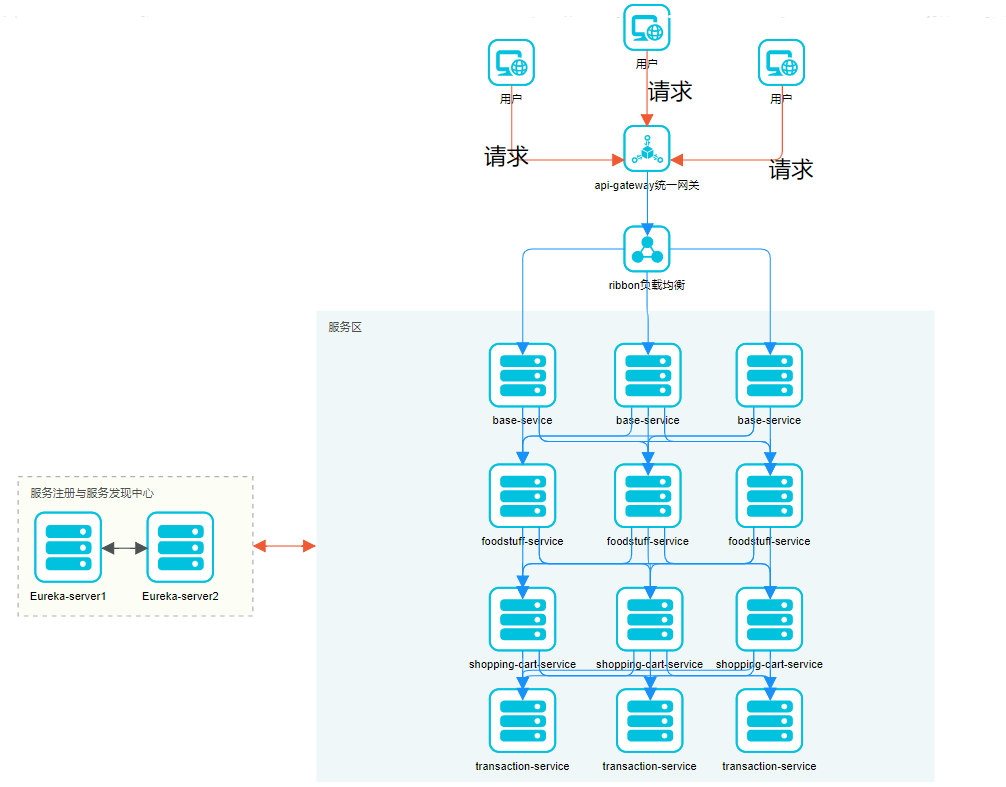


图1 系统总体架构图

## 3.3 数据库ER图设计

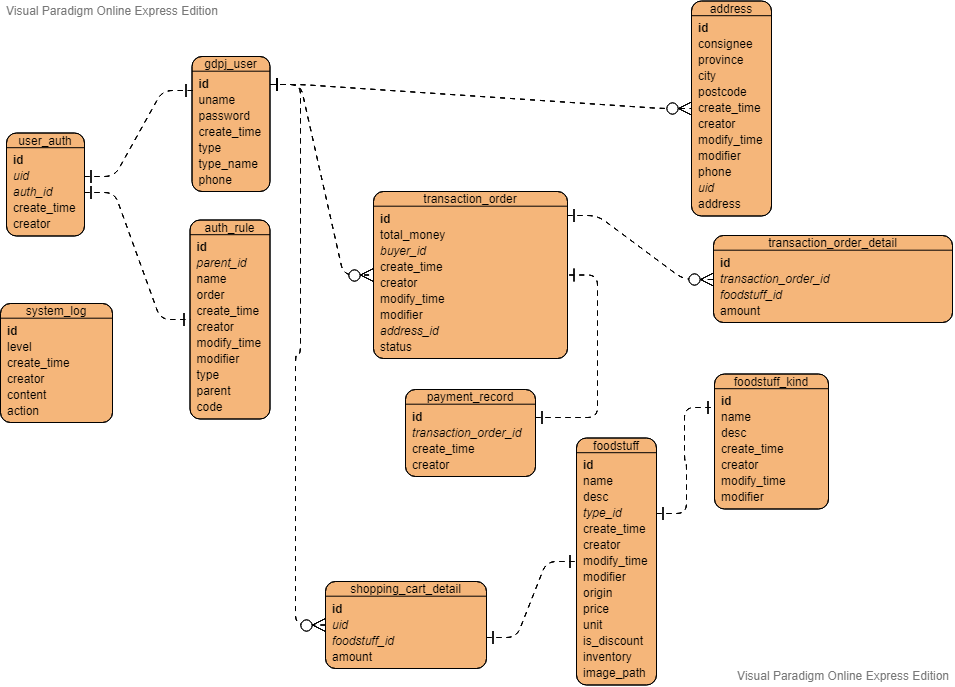


图2 数据库E-R图

## 3.4 系统开发平台和工具

### 3.4.1 系统开发平台

本食品商城系统在Windows10+ MacOS 10.14.3下开发,开发的主要语言使用JAVA 8、Javascript,数据库使用Mysql8.0。

### 3.4.2 开发工具

本食品商城系统的主要开发工具有IDEA2019、Maven、MysqlFont、Npm、Webpack、Tomcat。

## 3.5 系统运行环境

### 3.5.1 软件环境

Tomcat8、nodejs v10.15.1。

### 3.6.2 硬件环境

CPU:intel i5-7500、内存:8GB DDR4 、硬盘128SSD+1TBHDD。

# 4 基于微服务架构的食品商城系统详细设计

## 4.1 功能结构分析

系统使用微服务架构,功能点主要集中在几个业务服务中,base、foodstuff、shopping-cart、transaction这四个服务。

其中,base服务主要提供用户注册、用户登陆、用户地址添加、用户地址修改、系统日志添加、系统日志查询、服务运行信息收集等功能,都放在base服务中;foodstuff服务主要提供食品相关的功能,包括食品添加、食品修改、食品类目添加、食品类目修改、图片上传功能;shoppingCart服务主要提供购物车相关的功能,包括用户购物车详情、加入购物车、清空购物车功能;transaction服务主要提供与交易相关的功能,包括了订单生成、订单删除、订单详情查看、交易支付记录添加、交易支付详情查看。

图3 base服务功能图

图4 foodstuff服务功能图

图5 shoppingCart服务功能图

图6 transaction服务功能图

## 4.2 数据库表结构设计

数据库一共有11张表,分别是用户信息表gdpj\_user,用户收货地址信息表address,权限信息表auth\_rule,用户权限配置表user\_auth,系统日志信息表system\_log,食品信息表foodstuff,食品类目信息表foodstuff\_kind,购物车信息表shopping\_cart\_detail,订单记录表transaction\_order,订单详情表transaction\_order\_detail,支付流水记录表payment\_record。

图7 (数据库表结构图)

表1 gdpj\_user

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **是否为空** | **是否主键** | **默认值** | **属性** |
| id | int(11) | NO | 是 | 无 | 自增 |
| uname | varchar(30) | YES | 否 | 无 | 无 |
| password | varchar(30) | YES | 否 | 无 | 无 |
| create\_time | datetime | YES | 否 | 无 | 无 |
| type | int(11) | YES | 否 | 无 | 无 |
| type\_name | varchar(30) | YES | 否 | 无 | 无 |
| phone | varchar(20) | YES | 否 | 无 | 无 |

表2 address

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **是否为空** | **是否主键** | **默认值** | **属性** |
| id | int(11) unsigned | NO | 是 | 无 | 自增 |
| consignee | varchar(30) | YES | 否 | 无 | 无 |
| province | varchar(20) | YES | 否 | 无 | 无 |
| city | varchar(20) | YES | 否 | 无 | 无 |
| postcode | varchar(20) | YES | 否 | 无 | 无 |
| create\_time | datetime | YES | 否 | 无 | 无 |
| creator | varchar(30) | YES | 否 | 无 | 无 |
| modify\_time | datetime | YES | 否 | 无 | 无 |
| modifier | varchar(30) | YES | 否 | 无 | 无 |
| phone | varchar(30) | YES | 否 | 无 | 无 |
| uid | int(11) | YES | 否 | 无 | 无 |
| address | varchar(255) | YES | 否 | 无 | 无 |

表3 auth\_rule

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **是否为空** | **是否主键** | **默认值** | **属性** |
| id | int(11) | NO | 是 | 无 | 自增 |
| parent\_id | int(11) | YES | 否 | 无 | 无 |
| name | varchar(50) | YES | 否 | 无 | 无 |
| order | int(11) | YES | 否 | 无 | 无 |
| create\_time | datetime | YES | 否 | 无 | 无 |
| creator | varchar(30) | YES | 否 | 无 | 无 |
| modify\_time | datetime | YES | 否 | 无 | 无 |
| modifier | varchar(30) | YES | 否 | 无 | 无 |
| type | int(11) | YES | 否 | 无 | 无 |
| parent | varchar(50) | YES | 否 | 无 | 无 |
| code | varchar(30) | YES | 否 | 无 | 无 |

表4 user\_auth

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **是否为空** | **是否主键** | **默认值** | **属性** |
| id | int(11) | NO | 是 | 无 | 自增 |
| uid | int(11) | YES | 否 | 无 | 无 |
| auth\_id | int(11) | YES | 否 | 无 | 无 |
| create\_time | Datetime | YES | 否 | 无 | 无 |
| creator | varchar(30) | YES | 否 | 无 | 无 |

表5 system\_log

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **是否为空** | **是否主键** | **默认值** | **属性** |
| id | int(11) | NO | 是 | 无 | 自增 |
| level | varchar(10) | YES | 否 | 无 | 无 |
| content | text | YES | 否 | 无 | 无 |
| create\_time | datetime | YES | 否 | 无 | 无 |
| creator | varchar(30) | YES | 否 | 无 | 无 |
| action | varchar(30) | YES | 否 | 无 | 无 |

表6 foodstuff

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **是否为空** | **是否主键** | **默认值** | **属性** |
| id | int(11) | NO | 是 | 无 | 自增 |
| name | varchar(99) | YES | 否 | 无 | 无 |
| desc | text | YES | 否 | 无 | 无 |
| typeid | int(11) | YES | 否 | 无 | 无 |
| create\_time | datetime | YES | 否 | 无 | 无 |
| creator | varchar(30) | YES | 否 | 无 | 无 |
| modify\_time | datetime | YES | 否 | 无 | 无 |
| modifier | varchar(30) | YES | 否 | 无 | 无 |
| origin | varchar(99) | YES | 否 | 无 | 无 |
| price | int(11) | YES | 否 | 无 | 无 |
| unit | varchar(20) | YES | 否 | 无 | 无 |
| is\_discount | int(11) | YES | 否 | 无 | 无 |
| inventory | int(11) | YES | 否 | 无 | 无 |
| image\_path | varchar(500) | YES | 否 | 无 | 无 |

表7 foodstuff\_kind

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **是否为空** | **是否主键** | **默认值** | **属性** |
| id | int(11) | NO | 是 | 无 | 自增 |
| name | varchar(99) | YES | 否 | 无 | 无 |
| desc | text | YES | 否 | 无 | 无 |
| create\_time | datetime | YES | 否 | 无 | 无 |
| creator | varchar(30) | YES | 否 | 无 | 无 |
| modify\_time | datetime | YES | 否 | 无 | 无 |
| modifier | varchar(30) | YES | 否 | 无 | 无 |

表8 shopping\_cart\_detail

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **是否为空** | **是否主键** | **默认值** | **属性** |
| id | int(11) | NO | 是 | 无 | 自增 |
| uid | int(11) | YES | 否 | 无 | 无 |
| foodstuff\_id | int(11) | YES | 否 | 无 | 无 |
| amount | int(11) | YES | 否 | 无 | 无 |

表9 transaction\_order

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **是否为空** | **是否主键** | **默认值** | **属性** |
| id | int(11) | NO | 是 | 无 | 自增 |
| total\_money | int(11) | YES | 否 | 无 | 无 |
| buyer\_id | int(11) | YES | 否 | 无 | 无 |
| create\_time | datetime | YES | 否 | 无 | 无 |
| creator | varchar(30) | YES | 否 | 无 | 无 |
| modify\_time | datetime | YES | 否 | 无 | 无 |
| modifier | varchar(30) | YES | 否 | 无 | 无 |
| status | int(11) | YES | 否 | 无 | 无 |
| address\_id | int(11) | YES | 否 | 无 | 无 |

表10 transaction\_order\_detail

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **是否为空** | **是否主键** | **默认值** | **属性** |
| id | int(11) | NO | 是 | 无 | 自增 |
| transaction\_order\_id | int(11) | YES | 否 | 无 | 无 |
| foodstuff\_id | int(11) | YES | 否 | 无 | 无 |
| amount | int(11) | YES | 否 | 无 | 无 |

表11 payment\_record

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **是否为空** | **是否主键** | **默认值** | **属性** |
| id | int(11) | NO | 是 | 无 | 自增 |
| transaction\_order\_id | int(11) | YES | 否 | 无 | 无 |
| create\_time | datetime | YES | 否 | 无 | 无 |
| creator | varchar(30) | YES | 否 | 无 | 无 |

## 4.3 系统平台技术架构

本食品商城系统是一个微服务架构的系统,主要使用以SpringCloud为核心的微服务框架。其中服务注册中心使用Netflix Eureka搭建,统一网关使用Zuul进行搭建,服务熔断器使用Netflix Hystrix,服务负载均衡使用Ribbon,其余各业务服务使用SpringBoot+Mybatis+SpringMVC集成开发。

## 4.4 相关主要工具包

SpringCloud-Dalston.SR1 springcloud的4.0版本主要依赖包、SpringBoot-1.5.4.RELEASE springboot主要依赖包、spring-cloud-starter-eureka-server Netflix Eureka Server的主要依赖包、spring-cloud-starter-eureka Netflix Eureka Client的主要依赖包、spring-boot-starter-actuator springboot监控信息依赖包、pagehelper-spring-boot-starter 分页插件依赖包、druid-spring-boot-starter-1.1.9 数据库连接池druid依赖包、spring-boot-starter-web springmvc整合包、mybatis-spring-boot-starter-1.3.0 mybatis整合包、mysql-connector-java-8.0.11 mysql8连接包、spring-cloud-starter-feign Netflix Feign依赖包。

## 4.5 项目工程结构

后台项目工程结构-主要分为服务注册与服务发现中心（eureka-server）、统一网关(api-gateway)、基础业务服务（base-service）、食品业务服务（foodstuff-service）、购物车业务服务（shopping-cart-service）、交易业务服务（transaction-service）六大模块。各服务独立开发和部署。

图8 后台项目工程结构图

前端后台管理系统工程结构-此部分为商家后台管理系统的前端项目部分，使用webpack进行vue的组件化。主要分为基础页面base、食品管理页面foodstuff、系统管理页面system、交易相关页面transaction四大板块。

图9 前端项目gdpj-admin工程结构

前端商城工程结构-此部分为商城端部分。使用webpack进行vue的组件化。主要分为公用组件（头部、底部、首页等）和业务组件（食品页面、购物车页面、结算页面）两大块。

图10 前端商城工程结构

## 4.6 配置文件的定义与说明

首先,各个服务的配置文件都有两份,一份dev开发环境配置,另一份是prod生产环境配置;

下面是一些公共的配置项

（1）Spring.profiles.active 指定食品商城后台配置文件(dev或prod)

（2）Management.security.enable 食品商城后台监控信息是否对外开放(用于收集监控信息)

（3）Server.port 食品商城后台服务启动http端口

（4）Spring.application.name 食品商城后台服务名称

（5）Eureka.client.serviceUrl.defaultZone 指定服务注册与服务发现中心的地址

（6）Zuul.add-host-header 食品商城后台网关是否开启屏蔽实际服务的地址

（7）Zuul.sensitive-headers 食品商城后台网关需要过滤那些http头部信息字段

（8）Zuul.host.connect-timeout-millis 食品商城后台网关连接超时时间(单位毫秒)

（9）Zuul.host.socket-timeout-millis 食品商城后台网关socket超时时间(单位毫秒)

（10）Ribbon.ReadTimeout ribbon读取超时时间(单位毫秒)

（11）Ribbon.SocketTimeout ribbon socket超时时间(单位毫秒)

（12）Spring.datasource.\*\*.\*\* 一级\*\*表示数据源名称,二级\*\*表示数据源配置信息(type、url、password等信息)

# 5 基于微服务架构的食品商城系统实现

## 5.1 系统部分功能

### 5.1.1 食品后台管理系统部分

这是商家和管理员使用的后台系统,主要有服务监控(主页)、食品管理、食品类别管理、订单管理、交易记录、系统日志这6个页面。下面会进行展示

5.1.1.1 各服务运行状态监控展示

（1）系统全局监控总览-通过收集各个服务的内存信息、堆栈使用率等参数进行汇总，在前端进行以图表形式进行展示。

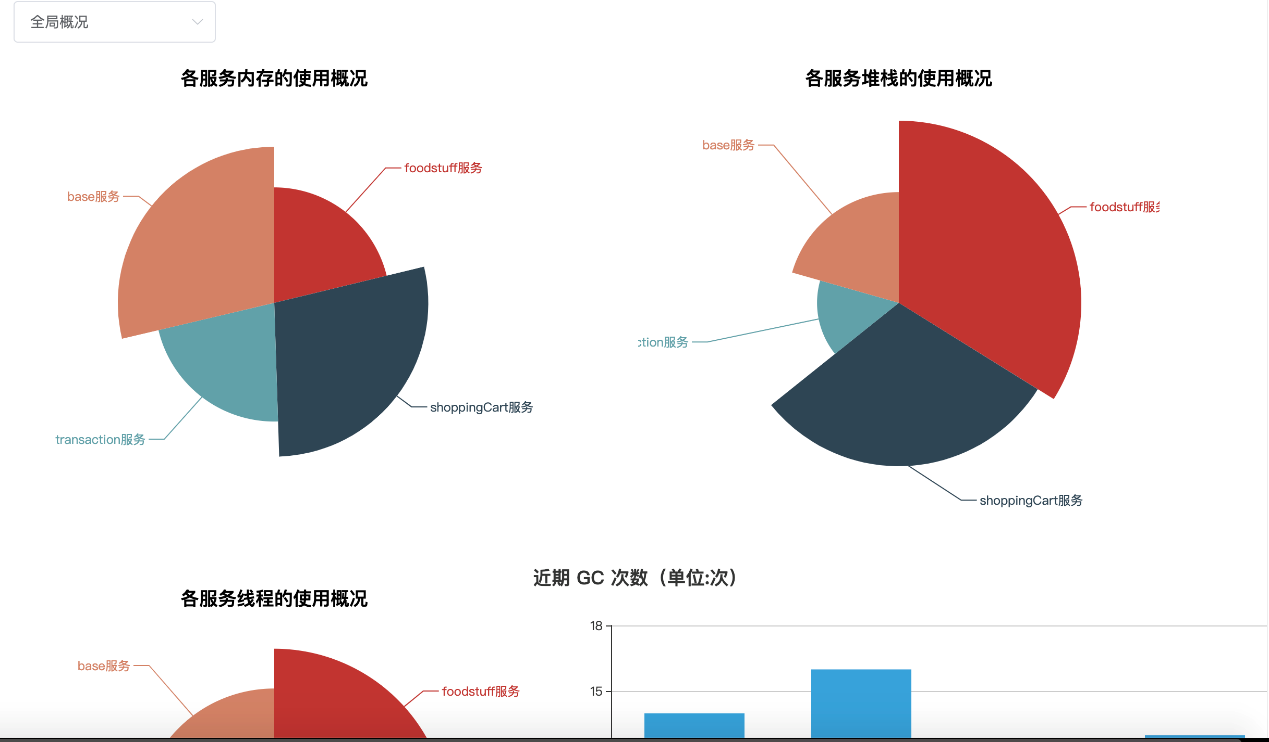


图11各服务运行状态监控展示

（2）单个服务的监控-具体每个服务的运行情况展示，主要展示内存使用情况和堆栈使用情况。

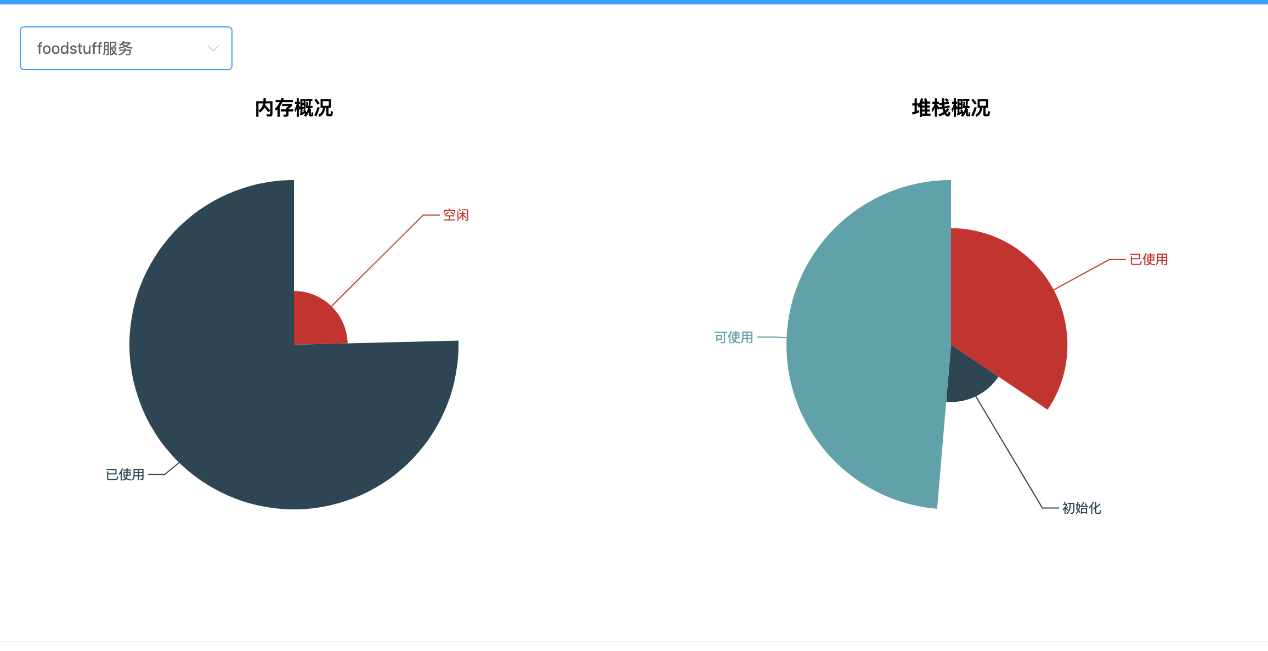


图12 单个服务的监控展示

5.1.1.2 食品管理模块展示

（1）食品列表-食品管理首页，目前所有的食品列表。

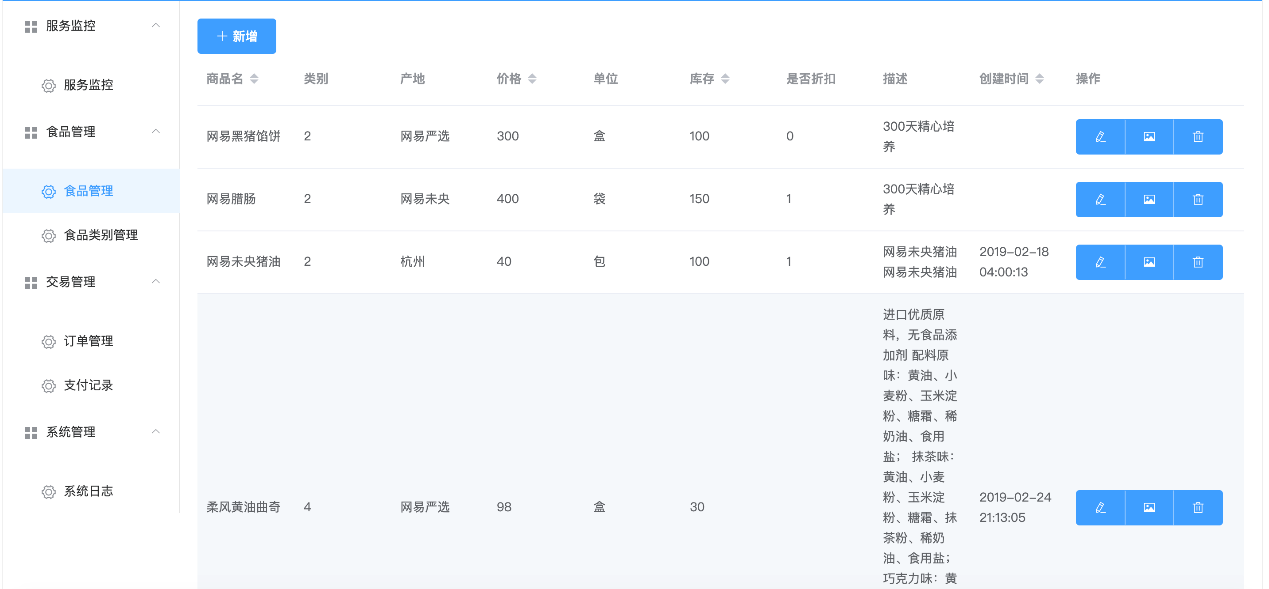


图13 现有食品列表

（2）食品新增-通过录入名称、价格、产地、描述、库存等信息，进行食品新增。

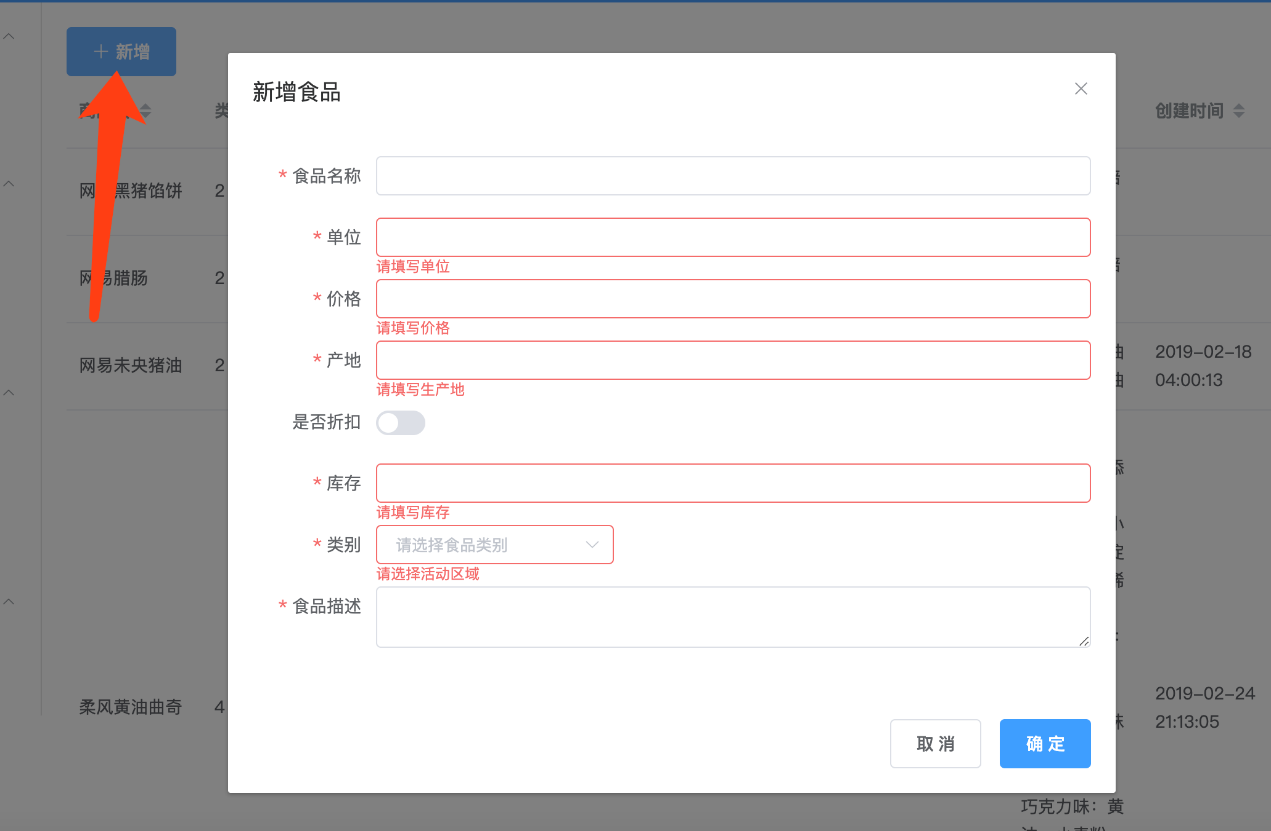


图14 新增食品

（3）食品编辑-通过自动填充，将原始数据填充到表单中，根据需要进行食品信息修改。



图15 修改食品

（4）食品图片上传和查看-以弹窗的形式，展示食品图片，同时可以进行添加。

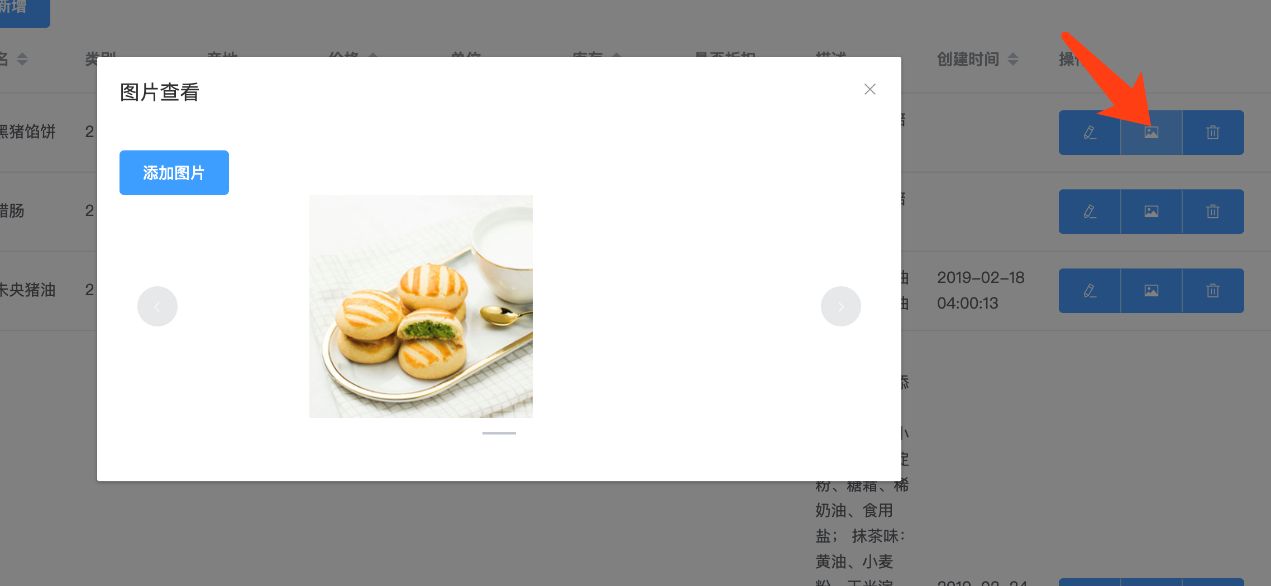


图16 食品图片上传与查看

5.1.1.3 食品类目管理模块展示

（1）食品类目列表-食品类目管理的首页，展示当前所有的食品类目列表。

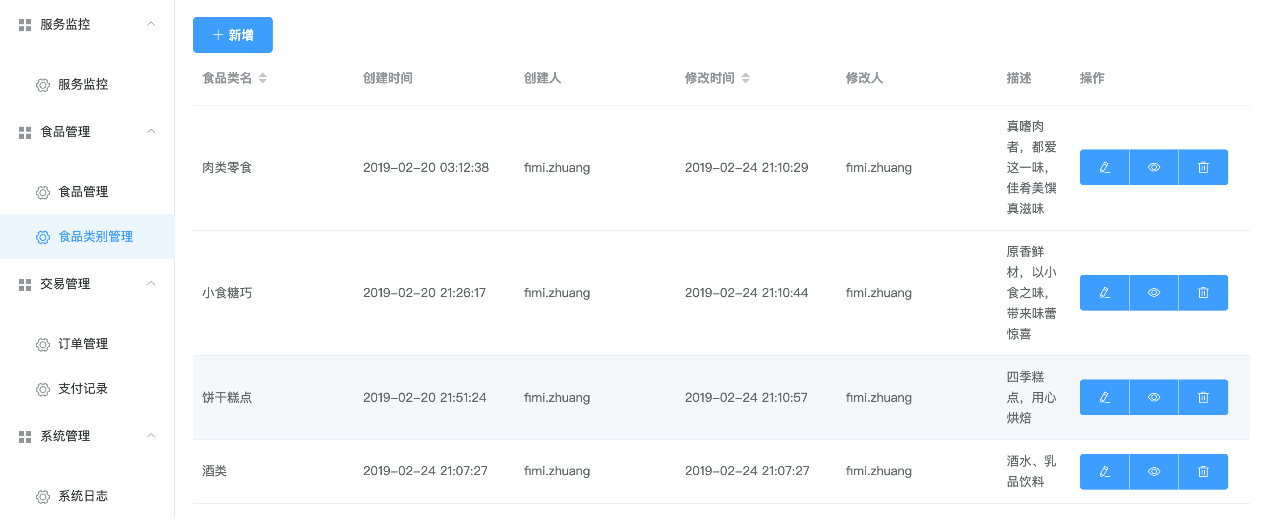


图17 食品类目列表

（2）食品类目新增-以弹窗的形式，录入食品类目名称、描述等信息，进行添加。

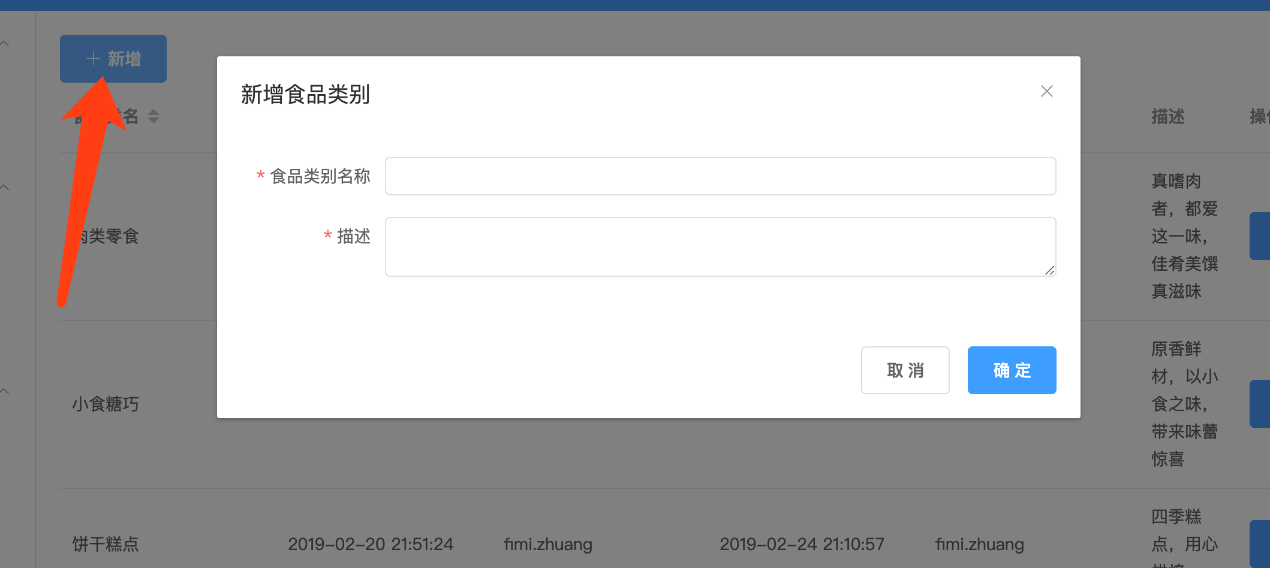


图18 食品类目新增

（3）食品类目编辑-通过自动填充，讲原信息填充到表单，后根据需要进行编辑。

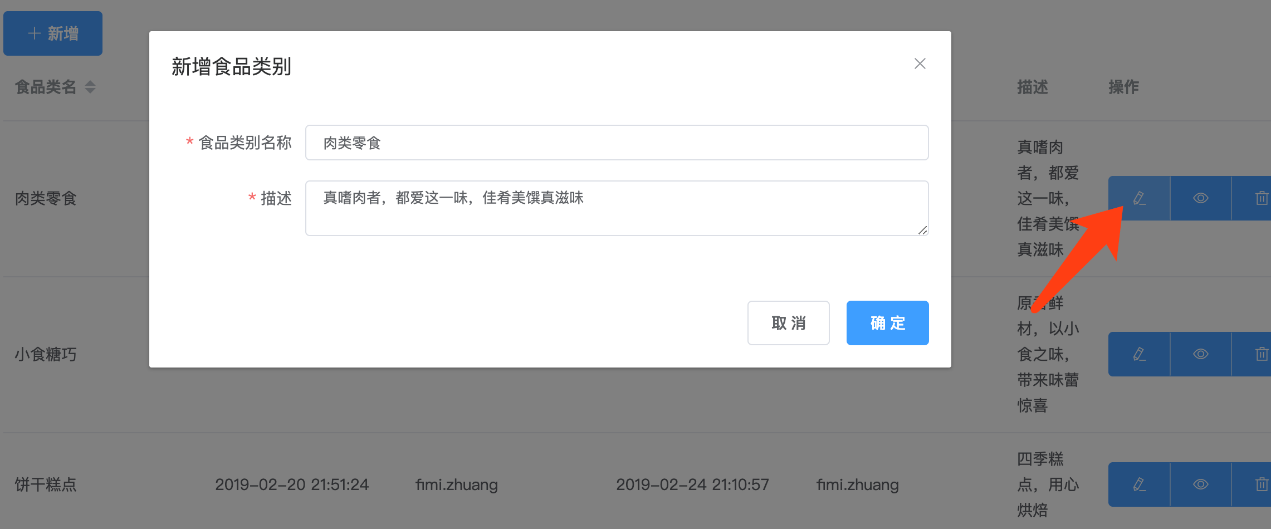


图19 食品类目编辑

5.1.1.4 订单模块展示

（1）所有订单列表-订单管理页面首页，展示当前商城所有的订单列表。



图20 订单列表

（2）订单详情查看-以弹窗的形式，展示该订单的详情信息（食品名称、数量）。

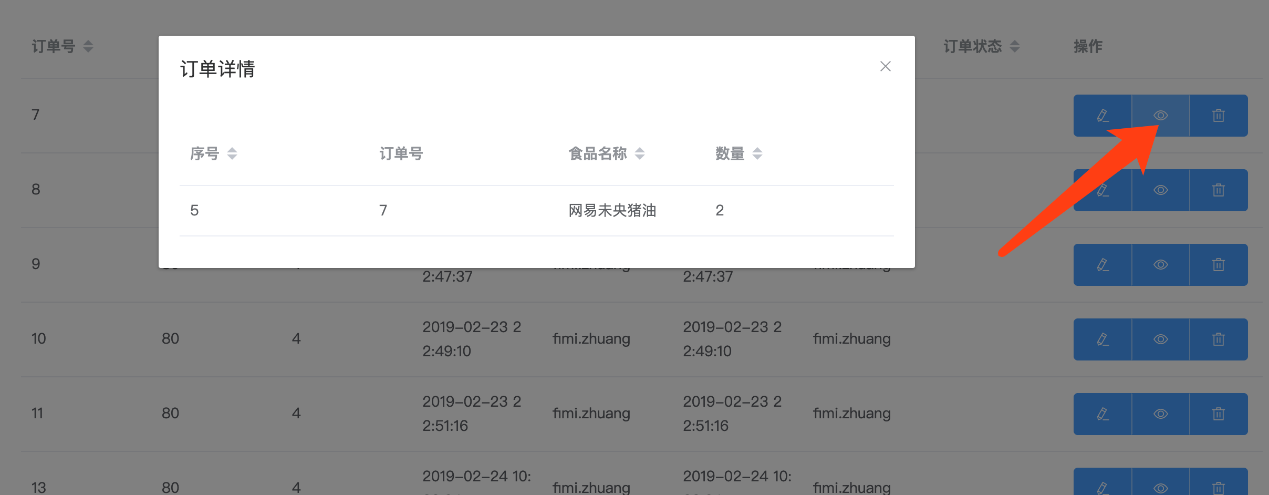


图21 订单详情

5.1.1.5 交易模块展示

（1）所有交易记录查看-交易记录首页，展示当前商城所有支付流水。



图22 交易记录列表

（2）交易关联的订单详情查看-以弹窗的形式，展示该条流水对应的订单详情（食品名称、总金额）。



图23 交易记录详情

5.1.1.6 系统运行日志模块展示

（1）系统所有日志查看-日志管理首页，展示当前系统产生的所有日志列表。

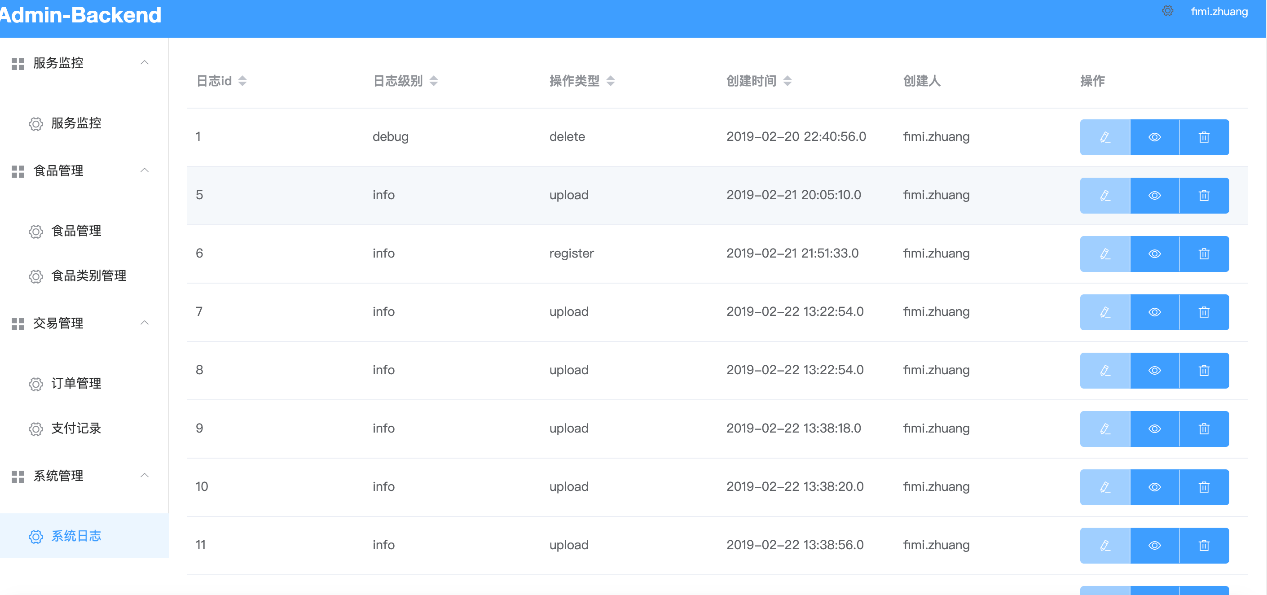


图24 系统日志列表

（2）单个日志详情查看-以弹窗的形式，展示该条日志的详细信息。

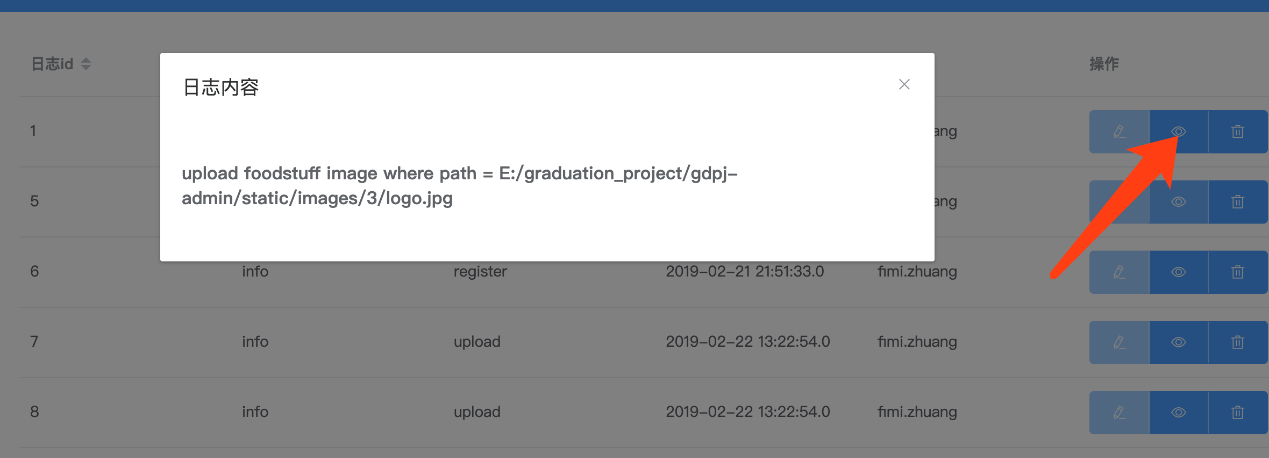


图25 日志详情

### 5.1.2 商城部分

商城部分,主要供买家用户使用。主要有商城主页、所有食品列表、单个食品详情、用户购物车、用户个人信息、用户地址管理、订单结算、交易成功、用户登陆、用户注册、用户反馈这11个页面

#### 5.1.2.1 商城首页展示

（1）首页-商城前端首页，主要包含搜索框、顶部导航栏、侧边导航栏和中部轮播图这几个部分。



图26 商城端首页

（2）侧边导购栏-通过光标进入事件，自动展开该分类下的所有导购信息。



图27 侧边导购栏目

5.1.2.2 用户注册登陆功能展示

（1）用户登陆界面-通过手机号码和用户密码两个参数，进行登录操作。



图28 用户登录页面

（2）用户注册界面-包含手机号验证和密码设置两个步骤。



图29 用户注册页面

5.1.2.3 食品搜索功能展示

搜索词“网易”结果-通过在搜索框中输入“网易”关键字，展示出来的搜索结果。

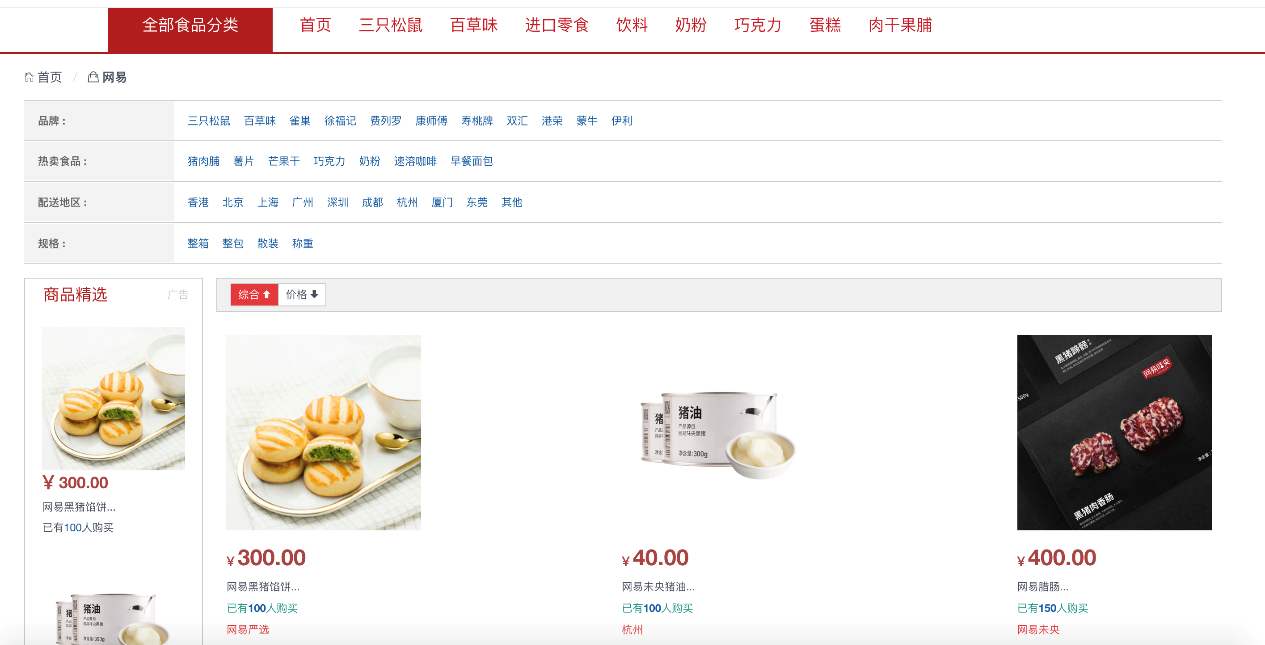


图30 食品搜索展示

5.1.2.4 食品详细信息页面展示效果

整体展示-单击某个食品图片，进入食品详情页面，主要展示价格、名称、食品描述和食品参数。

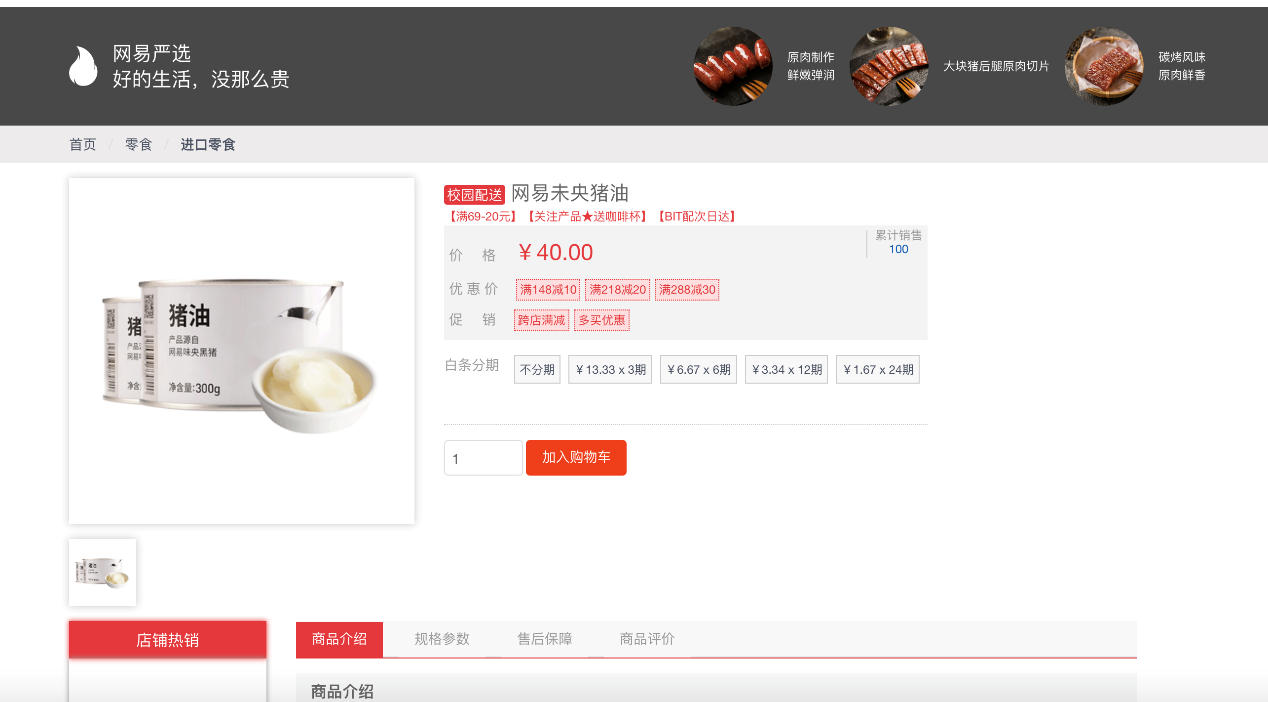


图31 食品详情展示

5.1.2.5 购物车管理和下单流程效果展示

（1）添加购物车-在食品详情页面，通过“加入购物车”按钮，可以将一定数量的食品加入到个人购物车中。



图32 加入购物车

（2）购物车详情-单击顶部导航栏中的“购物车”链接，可以链接到个人购物车详情页面，展示食品名称、图片缩略图、数量和单价。



图33 个人购物车详情

（3）订单结算-通过点击购物车中的结算按钮，进入订单结算页面，该页面会展示食品缩略图、食品名称、数量、单价、总额和收货地址。



图34 订单结算页面

5.1.2.6 模拟在线支付功能展示

订单支付-单击结算页面中的“支付订单”按钮，即可完成模拟在线支付。

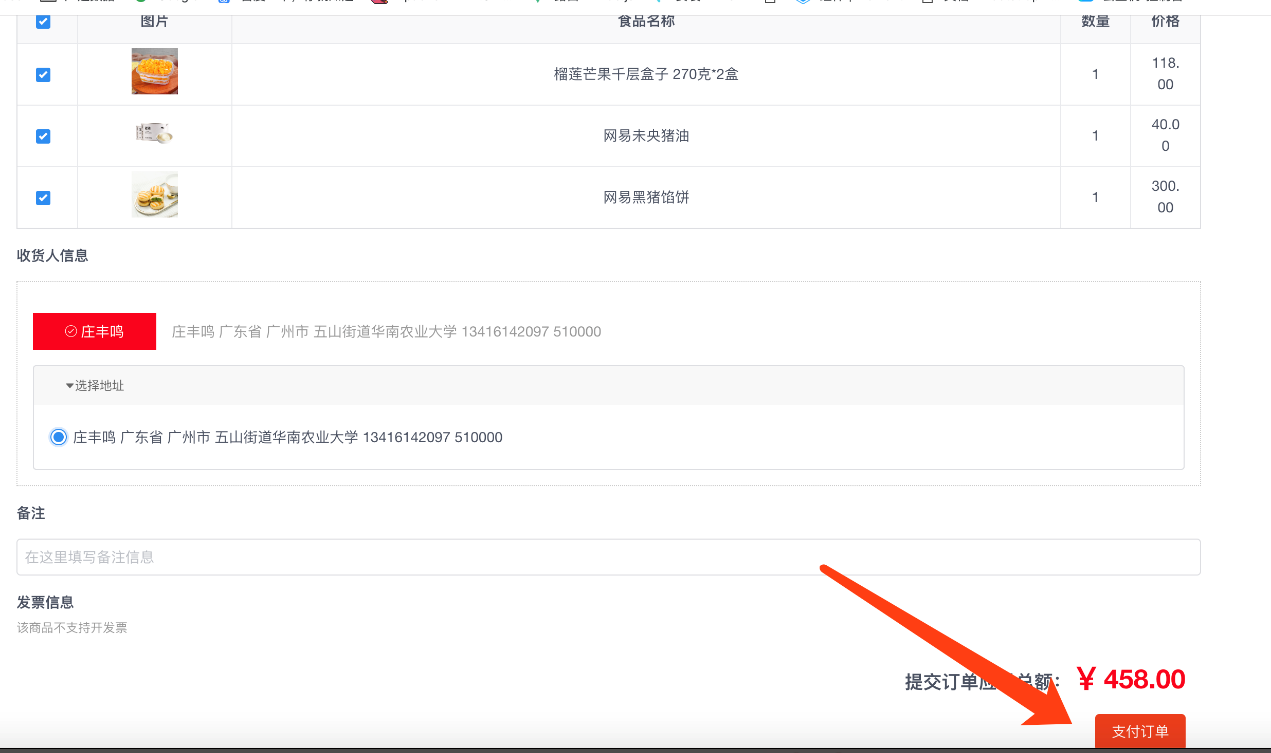


图35 订单结算

5.1.2.7 个人收货地址管理展示

（1）个人收货地址查看-点击顶部导航栏中的user标志，来到个人信息主页，首先展示的是个人收货地址列表。

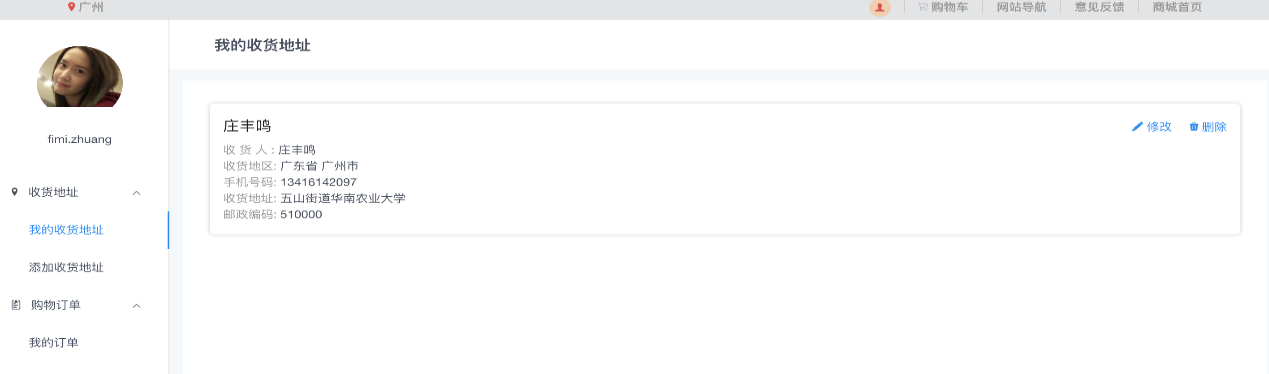


图36 个人收货地址

（2）个人收货地址添加-通过输入收件人、地区、详细地址等信息，添加个人收货地址。

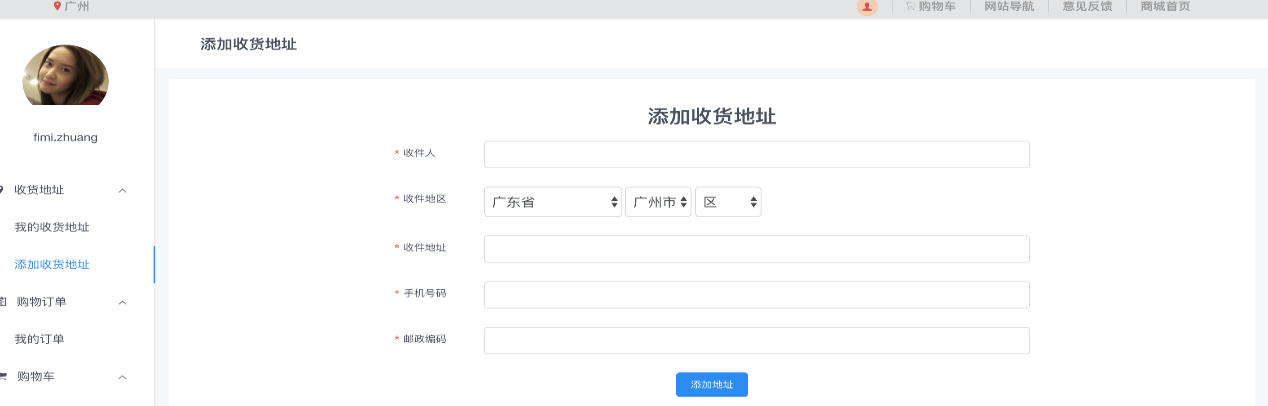


图37 添加个人收货地址

（3）个人收货地址修改-单击“修改”按钮，自动将原信息填充表单，用户修改所需要的信息即可。



图37 个人收货地址修改

# 6 总结与展望

## 6.1 总结

这次毕设,主要是探索和研究目前比较火的一个后端架构---微服务架构。并且结合SpringCloud这个为微服务架构实施提供全方位支持的框架,写了一个食品商城系统进行实践。过程中使用对技术都是比较新颖以及主流的,并且绝大部分技术都是一边学习一边运用,对于个人的技术面有比较大的拓展以及个人技术水平有一个比较大的提升。

后端方面基于SpringCloud搭建微服务架构服务端,搭配Springboot+Mybatis快速开发4个业务服务，同时加入统一网管、负载均衡、熔断器等技术，提升了后台系统的稳定性和可用性；两个前端项目均使用vue+npm+webpack搭建,通过使用vue框架，使得前端页面性能更优同时逻辑稳定，使用npm+webpack对项目进行管理，极大提升了开发效率同时规范了项目工程结构。

## 6.2 展望

微服务架构涉及的技术面非常广和深，涉及服务治理、一键部署、智能化运维等方面的知识。在本食品商城系统中，也仅仅是实现了一个大致的模样,在服务治理上是比较到位了,在服务的部署以及智能化运维的实现上还是有一定的欠缺。因为服务的拆分和独立部署,会使得服务数非常多,同时部署难度系数提升，将直接导致部署和运维成为一个难题。如果要进一步完善的话,可以先从自动化运维入手,使用Jenkins自动化构建框架然后配合Docker虚拟化技术,通过Jenkins拉取代码变更，将单个服务制作成docker镜像进行部署。这将使得服务水平拓展变得更加容易,同时服务的部署更加智能化。

# 参 考 文 献

翟永超.SpringCloud微服务实战[M].电子工业出版社,2017:1-300

吕宇琛.SpringBoot框架在web应用开发中的探讨[J].科技创报,2018,15(08):168-173.周立. Spring Cloud与Docker微服务架构实战[M].电子工业出版社,2018:56-78

徐郡明. MyBatis技术内幕[M].电子工业出版社,2017:115-154

疯狂软件. Spring+MyBatis企业应用实战[M].电子工业出版社,2018:161-166

张耀春. Vue.js权威指南[M].电子工业出版社,2016:21-33

[美]坎特伦. Node.js实战[M]. 人民邮电出版社,2014:13-33

吴浩麟. 深入浅出Webpack [M].电子工业出版社,2018:8-12

周俊鹏. 前端工程化：体系设计与实践[M].电子工业出版社,2018:12-18

郑天民. 微服务设计原理与架构[M]. 人民邮电出版社,2018:19-33

王磊. 微服务架构与实践[M].电子工业出版社,2015:5-15

# 致 谢

在毕业论文即将完成，亦意味着大学生活即将结束，毕业遥遥无期，转眼就要各奔东西了。

这四年期间，感谢维德科技和虎牙科技给了我实习的平台，感谢指导教师冯健昭老师，从毕业论文的开题报告开始，到论文的初稿，再到论文的定稿，一直到最后论文的完成，冯健昭老师多次组织毕业设计和毕业论文的讨论会，指导我如何把握毕业设计的方向和进度，并为我提出许多建议，热心解答我的问题。

感谢华南农业大学每一位传授学识给我的老师们与教授们，让我接触到不同知识领 域，使我懂得如何处理一个又一个棘手的问题，每一次严谨的教学都让我受益匪浅。

感谢在大学生活中与我风雨兼程的同学和朋友们，使我在大学四年里度过了愉快的 时光。

最后，特别感谢我的父母与兄弟，他们总是无怨言地鼓励与支持着我，使我总能够 没有压力地学习与工作。他们是我人生中最灿烂的宝藏。

**华南农业大学**

**本科生毕业论文成绩评定表**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 学号 | 201525060230 | | | | | 姓名 | | 庄丰鸣 | 专业 | 软件工程 | | |
| 毕业论文题目 | | | 基于微服务架构的食品商城 | | | | | | | | | |
| 指导教师评语  成绩（百分制）：　　　　　　　　　　　　　　　　　　　　指导教师签名：　　　　　　　　　　　　年　　月　　日 | | | | | | | | | | | | |
| 评  阅  人  评  语  及  成  绩  评  定 | | 成绩  评定  标准 | | **评分项目** | | | | | | | **分值** | **得分** |
| 选题  质量  20% | 1 | | 专业培养目标 | | | | 5 |  |
| 2 | | 课题难易度与工作量 | | | | 10 |  |
| 3 | | 理论意义或生产实践意义 | | | | 5 |  |
| 能力  水平40% | 4 | | 查阅文献资料与综合运用知识能力 | | | | 10 |  |
| 5 | | 研究方案的设计能力 | | | | 10 |  |
| 6 | | 研究方法和手段的运用能力 | | | | 10 |  |
| 7 | | 外文应用能力 | | | | 10 |  |
| 成果  质量40% | 8 | | 写作水平与写作规范 | | | | 20 |  |
| 9 | | 研究结果的理论或实际应用价值 | | | | 20 |  |
| 评阅人评语  成绩（百分制）：　　　　　　　　　　　　　评阅人签名：　　　　　　　　　　　　年　　月　　日 | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 答  辩小组  评语及  成  绩  评  定 | 评价项目 | 具体要求（A级标准） | 最高分 | 评分 | | | | |
| A | B | C | D | E |
| 论文质量 | 论文结构严谨，逻辑性强；有一定的学术价值或实用价值；文字表达准确流畅；论文格式规范；图表（或图纸）规范、符合要求。 | 60 | 55-60 | 49-54 | 43-48 | 37-42 | ≤36 |
|  |  |  |  |  |
| 论文报告、讲解 | 思路清晰；概念清楚，重点（创新点）突出；语言表达准确；报告时间、节奏掌握好。 | 20 | 19-20 | 17-18 | 15-16 | 13-14 | ≤12 |
|  |  |  |  |  |
| 答辩情况 | 答辩态度认真，能准确回答问题 | 20 | 19-20 | 17-18 | 15-16 | 13-14 | ≤12 |
|  |  |  |  |  |
| 答辩小组评语  是否同意通过论文答辩（打√）   1. 同意 2. 不同意   成绩（百分制）：　　　　　　　　 答辩小组成员（签名）：  　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　 年 月 日 | | | | | | | |
| 成绩总评 | 论文总评分数：  教学院长签名：  学院盖章：  年 月 日 | | | | | | | |

续上表：