

本科毕业设计（论文）

**题目：基于SpringCoud+Uniapp+Vue的多商户独家小程序的星火外卖平台的设计与实现**

**Design and Implementation of Spark Takeout Platform, a Multi-Merchant Exclusive Mini Program Based on SpringCloud + Uniapp + Vue**

**学 生 姓 名： 庄杰**

**学 院： 计算机学院**

**专 业： 软件工程技术**

**学 号： 2022120325C130**

**班 级： 软件工程技术C1班**

**指 导 教 师： 刘兴建**

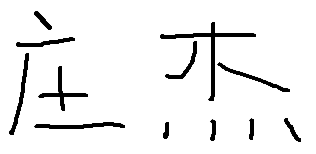
**二〇 年 月**

**教务处 制**

**毕业设计（论文）诚信声明**

本人郑重声明：所呈交的本科毕业设计（论文），是本人在指导教师的指导下，独立进行研究工作所取得的成果，成果不存在知识产权争议，除文中已经注明引用的内容外，本毕业设计（论文）不含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

特此声明！



学生签名： 日期： 年 月 日

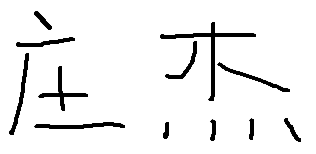
**毕业设计（论文）使用授权的说明**

本人完全了解广东工商职业技术大学关于收集、保存、使用学位论文的规定，即:

1.按照学校要求提交毕业设计（论文）的印刷本和电子版本；

2.学校有权保存毕业设计（论文）的印刷本和电子版，并提供目录检索与阅览服务，在校园网上提供服务；

3.学校可以采用影印、缩印、数字化或其它复制手段保存论文。本人同意上述规定。



学生签名： 日期： 年 月 日

# 

# 摘 要

在当今外卖市场竞争激烈的情况下，开发另一个外卖系统的动机主要源于现有平台存在的一系列问题。小商家在接入美团、饿了么等外卖平台时，面临高额的抽成率。尽管一些商家考虑自行开发小程序以提高利润，但这往往伴随着流量减少、高额前期投入以及上线成本等问题。

为解决这些挑战，我们着眼于一种基于本地社区的外卖系统。该项目将在一个有限的地域范围内进行推广，向附近的商家介绍本平台，并鼓励他们在平台上运营自己的店铺。店铺内将设置平台-店铺的二维码，方便用户扫码点餐。这一举措使得每个店铺都成为平台的一个广告位，而用户的主动参与将大大提高真实用户转化率。随着更多店铺入驻，平台的流量也有望显著增长。这种方式有效解决了商家面临的流量问题，同时免去了他们自行开发小程序的烦恼。

此外，本平台的抽成率相比传统外卖平台将大幅降低，甚至采取给合作店铺分红的方式，根据其销售情况进行分红。这一举措不仅使得店铺更愿意与平台合作，也增强了店铺对平台的归属感。

为此我们将以上述问题入手并作为基本的目标来开发我们的外卖平台系统。

关键词**：**多商户外卖平台；店铺入驻；流量增长

# Abstract

In today's fiercely competitive food delivery market, the motivation to develop another food delivery system mainly stems from a series of problems existing on existing platforms. Small businesses face high commission rates when accessing delivery platforms such as Meituan and Ele.me. Although some businesses consider developing their own mini programs to increase profits, this often comes with issues such as reduced traffic, high upfront investment, and launch costs.

To address these challenges, we focus on a local community based food delivery system. The project will be promoted within a limited geographical area, introducing the platform to nearby merchants and encouraging them to operate their own stores on the platform. A platform store QR code will be set up in the store to facilitate users to scan and order meals. This measure makes each store an advertising spot on the platform, and active user participation will greatly improve the conversion rate of real users. With more stores settling in, the platform's traffic is expected to significantly increase. This approach effectively solves the traffic problem faced by merchants, while eliminating the hassle of developing their own mini programs.

In addition, the commission rate of this platform will be significantly reduced compared to traditional food delivery platforms, and even dividends will be distributed to cooperative stores based on their sales situation. This measure not only makes stores more willing to cooperate with the platform, but also enhances their sense of belonging to the platform.

To this end, we will start with the above issues and use them as the basic goal to develop our food delivery platform system.

**Keywords：**Multiple merchant delivery platforms; store entry; and traffic growth

**目 录**

[1 绪论 1](#_Toc1667)

[1.1 研究背景及现状分析 1](#_Toc4952)

[1.2 研究的目的和意义 2](#_Toc19982)

[1.3 目前存在的问题 3](#_Toc11705)

[1.4 国内外研究现状 3](#_Toc1280)

[1.5 论文主要内容与结构 4](#_Toc27461)

[2 相关技术和工具 6](#_Toc18261)

[2.1 开发工具 6](#_Toc19705)

[2.2 前端技术 7](#_Toc23362)

[2.3 后端技术 8](#_Toc19640)

[2.4 中间件 10](#_Toc30106)

[2.5 本章小结 12](#_Toc19609)

[3 需求分析和系统数据库设计 13](#_Toc13915)

[3.1 系统可行性分析 13](#_Toc16835)

[3.2 功能需求分析 13](#_Toc18097)

[3.4 系统E-R图 15](#_Toc32489)

[3.5 数据库表设计 23](#_Toc3047)

[3.5.1 用户模块表设计 23](#_Toc25576)

[3.5.2 商家模块表设计 26](#_Toc16168)

[3.5.3 订单模块表设计 35](#_Toc8980)

[3.5.4 优惠券模块表设计 40](#_Toc30714)

[3.5.5 骑手模块表设计 44](#_Toc4664)

[3.6 本章小结 48](#_Toc26584)

[4 系统业务设计与实现 49](#_Toc27740)

[4.1 商家入驻实现 49](#_Toc14167)

[4.2 商家基本信息经营配置的修改实现 50](#_Toc21181)

[4.3 商家活动实现 51](#_Toc14778)

[4.4 用户整体主线的实现 52](#_Toc3751)

[4.5 商家接单提醒实现 53](#_Toc4579)

[4.6 骑手接单的实现 54](#_Toc25366)

[4.7 本章小结 56](#_Toc31740)

[5 系统测试与分析 57](#_Toc29755)

[5.1 功能测试 57](#_Toc18777)

[5.2 性能测试 61](#_Toc15457)

[5.3 本章小结 63](#_Toc18516)

[6 总结与展望 64](#_Toc198)

[6.1 总结 64](#_Toc24638)

[6.2 展望 64](#_Toc3678)

[参考文献 66](#_Toc10645)

[致 谢 68](#_Toc1441)

# 1 绪论

## 研究背景及现状分析

随着互联网和移动技术的飞速发展，外卖行业已经成为了人们生活中不可或缺的一部分。传统的外卖平台如美团、饿了么等在满足人们便利用餐需求的同时，也带来了一系列问题，尤其是对于小型商家而言，这些问题更加突出。

１．高额抽成问题：

外卖平台通常会从商家的订单中抽取一定比例的佣金作为平台服务费，这一比例往往较高，对于小商家而言是一个不小的负担。高额的抽成直接影响了小商家的利润空间，限制了它们的发展和成长。

２．小商家自主开发小程序的难题：

为了减少外卖平台的抽成，一些小商家可能会考虑自主开发自己的小程序进行外卖业务。然而，自主开发小程序需要前期投入高昂的费用，包括程序购买、定制开发、项目上线等成本，对于创业的新店来说是一个较大的负担。

３．流量稀缺和成本高昂：

即使一些小商家选择自主开发小程序，也面临着流量稀缺和成本高昂的问题。由于缺乏品牌知名度和推广力度，小商家的小程序往往无法吸引足够多的用户，导致流量稀缺，进而影响了销售和盈利能力。

针对以上问题，本研究关注的是如何通过新型外卖系统来解决这些难题。具体来说，本项目将在一个小的区域范围内推广新的外卖平台，吸引附近的小商家入驻，并通过店铺二维码推广，提升店铺的线上曝光和流量。用户可以通过扫码点餐的方式直接进入店铺点餐，实现店铺与平台的互动和合作。这一模式既解决了外卖平台高额抽成的问题，也减轻了小商家自主开发小程序的成本压力，同时提升了用户体验和店铺的归属感。

当前外卖行业竞争激烈，传统的外卖平台已经成熟，但也暴露出了一些不足之处。因此，探索新型的商业模式和解决方案对于推动外卖行业的创新和发展具有重要意义。通过本研究的实施，希望能够为小商家和消费者提供更加便捷、高效的外卖服务，促进外卖行业的可持续发展和进步。

## 研究的目的和意义

1. 帮助小商家减轻外卖平台高额抽成的负担：

通过设计一个新型的外卖系统，实施更低的抽成比例，从而直接减轻小商家在传统外卖平台上的财务压力，提高其在平台上的盈利水平。这一措施将增加小商家的利润空间，促进其持续发展和成长。

1. 提升小商家的线上曝光和客流量：

通过在小区域内推出新的外卖平台，为周边店铺提供一个专属的线上运营空间。这不仅能增强小商家的线上可见度和吸引力，还能通过店内二维码的推广，有效地提升店铺流量。这将有助于增加销售额，加强小商家与消费者之间的联系和互动。

1. 提高用户体验和满意度：

新外卖系统通过扫码点餐方式，优化了用户点餐流程，减少了等待时间。同时，通过二维码的直观展示，用户能更清晰地了解周边餐厅的选择，从而提升用户的选择性和整体用餐体验。这种改进将提高用户满意度和忠诚度，对系统的长期成功至关重要。

1. 探索新型外卖平台的商业模式：

该系统采用了与传统外卖平台不同的商业模式，例如降低抽成和向店铺提供分红。这种模式的研究不仅可以验证其对小商家的直接益处，还可以作为推动外卖行业商业模式创新的一个典范，为同行业其他平台的建立提供重要的参考。5. 推动本地经济发展和社区建设：

新的外卖系统专注于小区域内的商业合作和发展，这有助于推动本地经济的多元化发展和社区建设。通过支持小商家的成长和壮大，将带动就业增长、提升居民生活质量，进而实现经济社会的可持续发展。

## 1.3 目前存在的问题

推广外卖平台可能遇到商家抵触的问题，因为需要对商家进行抽成。为了解决这一问题，我们可以通过以下方式提高平台吸引力：

* 优化抽成策略：尽量降低每单的抽成比例，减轻商家负担，提高商家参与度。
* 加强推广：通过有效的推广活动提升平台知名度和用户量，增加订单量，从而弥补低抽成带来的收入减少。
* 优化用户体验：提升平台用户体验，包括简洁方便的下单流程、高效准时的配送服务等，提高用户满意度，增加复购率和订单量。
* 差异化服务：在商家管理方面做出差异化，如加强商家环境的检查，保证用户用餐安全，树立平台形象，吸引更多用户选择我们的平台。

## 1.4 国内外研究现状

外卖行业作为互联网时代的产物，受到了广泛的关注和研究。从国内外的研究现状来看，外卖行业的发展和变革呈现出多样化和复杂化的特点，主要集中在以下几个方面：

1. 外卖平台的商业模式研究：

国内外学者对外卖平台的商业模式进行了深入研究，探讨了其利润来源、市场定位、服务内容等方面的特点。研究表明，外卖平台的商业模式不断创新，从最初的佣金收费到后来的广告推广、会员服务等多种方式，不断丰富和完善，以满足不同用户和商家的需求。

1. 外卖平台对小商家的影响研究：

一些研究关注外卖平台对小商家的影响，特别是在抽成、品牌形象、市场竞争等方面的影响。研究发现，外卖平台的高额抽成对小商家的利润空间造成了一定的压力，但同时也为小商家带来了更多的曝光和订单，促进了其业务的发展。

1. 外卖平台的用户体验研究：

有学者对外卖平台的用户体验进行了调查和研究，主要关注用户对平台界面、下单流程、配送速度、食品质量等方面的满意度和反馈。研究结果表明，用户对外卖平台的要求越来越高，他们希望能够享受到更加便捷、快速、安全的外卖服务。

1. 外卖平台的社会影响研究：

一些研究关注外卖平台对社会的影响，包括对就业、交通、城市管理等方面的影响。研究发现，外卖行业的迅猛发展为就业提供了机会，但也带来了交通拥堵、环境污染等问题，需要政府和企业共同应对。

1. 外卖行业的监管研究：

国内外学者对外卖行业的监管进行了探讨和分析，主要包括食品安全、配送员管理、平台责任等方面的议题。研究认为，外卖行业的监管需要政府、企业和社会各方的共同努力，才能够实现行业的健康发展和可持续性。

## 1.5 论文主要内容与结构

1. 绪论，通过研究现外卖平台存在的问题，引出自研的平台的特点。

2. 相关技术与工具，主要说明前后使用了哪些技术框架，特别的是我们自研了自己的后端框架，用自研的技术解决了分布式系统下存在的问题。除此说明了使用系统使用了哪些工具，中间件。

3. 系统数据库设计，本文详细展示了数据库的UML图、数据库表的设计以及表的结构图。

4. 星火外卖平台的设计和实现，本文主要说了使用了系统使用了哪些技术与自研的框架来实现哪些需求，明了了系统是如何实现的。

5. 系统测试与分析，本文对系统进行了安全测试和功能测试，系统测试确保了系统功能能够正确进行。

# 2 相关技术和工具

本章简要阐释系统开发中应用的关键技术,包含开发系统所应用的语言技术、数据库技术等,以及系统开发所应用的关键技术环境。

## 2.1 开发工具

2.1.1 Visual Studio Code

作为前端编码工具。

Visual Studio Code（VS Code）是由微软开发的流行的开源代码编辑器。它为各种编程语言提供广泛的支持，包括调试功能、语法高亮显示、代码补全以及丰富的扩展库以增强生产力。借助其可定制的界面和强大的功能，VS Code被广泛用于Web开发、移动应用程序开发和其他软件项目。

2.1.2 IntelliJ IDEA

作为Java后端编码的重要工具。

IntelliJ IDEA是一款由JetBrains开发的强大集成开发环境（IDE），专为Java开发而设计。它提供高级的编码辅助、智能代码补全、内置版本控制系统以及各种软件开发工具。IntelliJ IDEA支持多种编程语言、框架和技术，因此成为全球Java开发人员的首选。

2.1.3 Navicat

作为数据库的管理的工具。

Navicat是一款强大的数据库管理工具，为管理和管理数据库提供了图形用户界面（GUI）。它支持各种数据库系统，如MySQL、PostgreSQL、Oracle、SQLite和SQL Server。Navicat提供数据操作、模式可视化、数据同步和数据建模等功能，是数据库开发人员和管理员的重要工具。

2.1.4 Another Redis Desktop Manager

作为Redis管理工具。

Another Redis Desktop Manager是一款跨平台的桌面应用程序，专为管理Redis数据库而设计。它为与Redis实例交互提供了用户友好的界面，允许用户查看和操作键、执行命令、监视服务器性能以及同时管理多个Redis连接。Another Redis Desktop Manager简化了Redis管理任务，提高了与Redis一起工作的开发人员的生产力。

2.1.5 微信开发者工具

作为Vue3结合Uniapp开发编绎为微信小程序进行程序的开发辅助与调试的工具。

微信开发者工具是一款由微信官方提供的集成开发环境，专门用于微信小程序和微信公众号的开发和调试。它提供了代码编辑器、调试器、模拟器等功能，帮助开发者快速创建、测试和发布微信应用程序。微信开发者工具支持实时预览和调试，同时提供丰富的开发文档和工具，使开发者能够高效地构建和优化微信平台上的应用程序。

2.1.6 Docker环境部署

Docker能部署项目需要的服务，即下面中要说的中间件。

Docker 是一个开源的平台，用于开发、部署和运行应用程序。它使用容器技术，使开发者能够在容器中打包应用程序及其所有依赖关系，从而实现轻松部署应用程序的目的。Docker 容器是一种轻量级、可移植的软件打包方式，它将应用程序与其依赖项打包在一起，形成一个独立的可执行单元，可以在任何支持 Docker 的平台上运行。这种容器化的部署方式使得应用程序的开发、测试和部署过程更加高效、可靠，并且能够实现跨平台的应用程序运行。

## 2.2 前端技术

最基本的HTML+CSS+JavaScript技术外，还包括以下前端技术：

2.2.1 Vue.js

在项目将Vue作为前端开发核心框架。

Vue.js是一款轻量级的JavaScript前端框架，由Evan You于2014年创建。它具有简洁的API、响应式数据绑定、虚拟DOM等特性，使得构建交互式的Web界面更加简单和高效。Vue.js广泛应用于单页面应用（SPA）和动态网页的开发中。

2.2.2 ElementUI

在项目中将ElementUI构建商家的后台管理中心前端UI。

ElementUI是一套基于Vue.js的UI组件库，提供了丰富的UI组件和样式，包括按钮、表单、导航、布局等，帮助开发者快速构建美观、交互丰富的Web界面。ElementUI的组件设计符合现代Web开发的最佳实践，被广泛应用于Vue.js项目中。

2.2.3 Uni-app

在项目中将使用Uni-app结合Vue+TS构建用户端与骑手端的微信小程序。

Uni-app是一个使用Vue.js开发跨平台应用的解决方案，可以同时开发基于微信小程序、H5、App等多个平台的应用程序。它提供了一套统一的开发框架和组件库，使开发者能够通过编写一套代码，快速构建并发布到多个平台，提高了开发效率和跨平台应用的一致性。

2.2.4 其他JavaScript库

1、js-cookie

在项目中将使用该库完成商家端对cookie的操作，保存着商家的登录信息。

js-cookie是一个用于操作HTTP Cookies的JavaScript库，提供了简单的API，用于读取、设置和删除Cookies，方便开发者在客户端存储和管理用户信息和状态。

2、event-source-polyfill

在项目中将使用该库完成后端到商家端的SSE消息提示。

event-source-polyfill是一个用于解决浏览器兼容性问题的JavaScript库，用于支持使用EventSource接口进行服务器发送事件（Server-Sent Events）的通信。它提供了对不支持EventSource接口的浏览器的兼容性支持，使开发者能够在各种浏览器环境中使用服务器发送事件。

## 2.3 后端技术

2.3.1 Java

在项目中将使用Java作为后端开发的主语言。

Java是一种面向对象的编程语言，由Sun Microsystems（现在是Oracle Corporation）于1995年推出。它具有跨平台性、面向对象、健壮性和安全性等特性，被广泛应用于企业级应用、Web开发、移动应用和大数据处理等领域。

2.3.2 Spring

在项目中将使用Spring及全家桶作为后端开发的核心框架。

Spring是一个轻量级的Java开发框架，提供了全面的基础设施支持和丰富的功能，用于构建企业级Java应用程序。它包括Spring Core、Spring MVC、Spring Boot等模块，被广泛应用于后端开发领域。

2.3.3 MyBatisPlus

在项目中将使用该框架完成后端程序对数据库的操作，MyBatisPlus相较于MyBatis前者可以让我们少写非常多的sql，进而提高我们的开发效率。。

MyBatisPlus是MyBatis的增强版本，提供了更多的功能和便利性，如代码生成器、通用Mapper、分页插件等，简化了持久层的开发工作。

2.3.4 Spring Cloud

Spring Cloud是基于Spring Boot的微服务架构开发工具，提供了一系列的解决方案和组件，用于构建分布式系统中的各种模块和服务。它包括服务注册与发现、配置中心、负载均衡、断路器等功能，帮助开发者构建弹性、可靠的微服务架构。

2.3.5 Redisson

在项目中将使用Redisson+Redis作为分布锁的解决方案。

Redisson是一个基于Redis的Java驱动程序，提供了分布式、高可用、高性能的数据结构和服务，包括分布式锁、分布式集合、分布式对象等，用于简化Java应用程序中的分布式数据操作。

2.3.6 Sleuth

在项目中将使用Sleuth作为Zipkin进行链路追踪展示的数据来源，它一信息的收集器。

Sleuth是Spring Cloud中的一个分布式链路跟踪解决方案，用于跟踪分布式系统中的请求流程和调用链路，提供了可视化的监控和分析功能，帮助开发者诊断和优化分布式系统的性能和可靠性。

2.3.7 OpenFeign

在项目中将使用该技术完成服务间的调用。

OpenFeign是一个声明式的HTTP客户端，集成了Ribbon和Hystrix等功能，用于简化微服务架构中服务间的通信和调用，提供了可靠的服务调用和容错机制，提高了微服务架构的可维护性和可靠性。

2.3.8 自研的框架

* 自研的分布式安全框架：基于JWT+注解+反射开发的分布式安全框架，通过标注相关权限注解，在请求拦截时进行权限效验，框架还保证了内部接口的安全，使得有些接口只能进行项目的其它模块访问。
* 自研的分布式ID生成器: 基于雪花算法，使用Redis+Lua 互斥获取同一机房同一种模块的不同workerId, 使用类似于Redisson分布式锁的看门狗机制保持对workerId的持有，当出现较短时间回拨时使用阻塞方式，当为长时间回拨时通过切换为其它未被占用的workerId同时设置原有workerId的持有时间为回拨前的时间进行冻结，防止其它模块获取该workerId，避免出现产生的id重复。
* 自研的分布式Event基础框架：使用的是Redis 的Pub/Sub实现的分布式同一模块间的事件，使得同一模块间在于同一Pub/Sub内，分布式的SSE是基于该框架的，并商家端中使用了SSE，当商家有订单时，商家端会提醒有新订单，当业务需要向商家发送有新订单消息时，会看当前模块该商家的SSE，如果有直接发送，如果没有，会通过Event发送事件通知其它与其相同的模块来处理。

## 2.4 中间件

2.4.1 Nacos

Nacos是一个开源的服务发现和配置管理平台，用于实现微服务架构中的服务注册与发现、配置管理和动态路由等功能，提供了可靠的服务治理和服务管理能力，支持多种环境和部署方式。

2.4.2 MySQL

在本项目中将Mysql作为系统数据持久化的数据库。  
 MySQL是一个流行的开源关系型数据库管理系统，被广泛应用于Web应用程序和其他数据驱动的软件中，以其稳定性、性能和易用性而著称。

2.4.3 Redis

Redis在本项目存在众多的使用场景，如购物车、短信验证码、自研的框架等。

Redis是一个开源的内存数据库，用于存储键值对和数据结构，提供了高性能、高可用的数据存储和缓存服务，被广泛应用于Web开发、分布式系统和大数据处理等场景。

2.4.4 SEATA

SEATA在本项目中将用于作为保证分布式事务。

SEATA是一个分布式事务解决方案，提供了高性能、高可用的分布式事务管理服务，支持对分布式事务的原子性、一致性、隔离性和持久性保证，帮助开发者构建分布式系统中的事务处理功能。

2.4.5 RabbitMQ

在本项目中使用该中件间完成超时未支付订单自动取消的操作。

RabbitMQ是一个开源的消息队列软件，用于实现异步消息传递和事件驱动的架构，提供了可靠的消息传递和消息队列管理功能，被广泛应用于分布式系统、微服务架构和消息通信中。

2.4.6 XXL-JOB

在本项目中使用该中件间完成了对骑手/商家的到账。

XXL-JOB是一个分布式任务调度平台，用于实现定时任务和异步任务的调度和执行，提供了任务管理、任务调度、任务执行等功能，帮助开发者构建高效、可靠的任务调度系统。

## 2.5 本章小结

本单介绍了开发本项目主要使用的开发工具，然后介绍用到的前端与后端各技术及其在本项目的用处，特别地介绍了自研框架的实现原理。除此还介绍了其它服务的支持及用处，如Docker用来作为各中间件的部署工具。

# 3 需求分析和系统数据库设计

## 3.1 系统可行性分析

星火外卖平台-分布式外卖系统，通过不同模块提供的服务（用户、商家、订单、优惠券、骑手）通过网关结合Nacos获取到目标服务通过负载均衡将请求分发给对应模块服务处理，使得系统能够处理大量的客户端请求，下面是星火外卖平台-分布式外卖系统的可行性分析：

1. 技术可行性

分布式微服务经过多年的沉淀，分布式技术已经趋于成熟，各种中件间的出现使众多的分布式项目问题得于解决，由此在使用分布式项目来开发与外卖平台项目是可行与可靠的。

（2）经济可行性

开发分布式的外卖平台项目前期必要的主线功能开发是不需要更多资金的投入的，当项目完成上线时，在资金上将得到解决，后期再使平台的功能更加丰富与完善，因此在经济上是可行的。

（3）安全可行性

在分布式项目中通过加密技术与JWT的身份验证技术等手段来保证数据操作的安全性，通过后期部署时采取一系列灾难恢复措施来保证系统的可靠性和稳定性。所以在安全可行性方面，是可以得到保障的。

业务可行性：分布式的外卖平台系统需要支持各种不同的业务需求，例如订单管理、配送管理、商家管理、用户管理，因此在业务可行性方面我们需要调研类似产品如美团外卖、饿了么平台在功能上的分析，确保系统能够满足各种不同的业务需求。

综上所述，分布式的星火外卖平台是一个具有很高可行性的系统，但是在实际应用中，还需要进行充分的调研和分析，确保系统能够满足各种不同的业务需求，同时采取一系列安全措施来保证系统的安全性和稳定性。

## 3.2 功能需求分析

3.2.1 用户端

用户端必须要展示商家门店信息，这是用户进入商家店内下单的起点，通过点击交互进入店内进行下单的必要交互页面，还需要能查看订单信息，用户还需要能设置自己的收货地址，领取与查看优惠券等信息甚至还需要看到配送订单的配送进度。

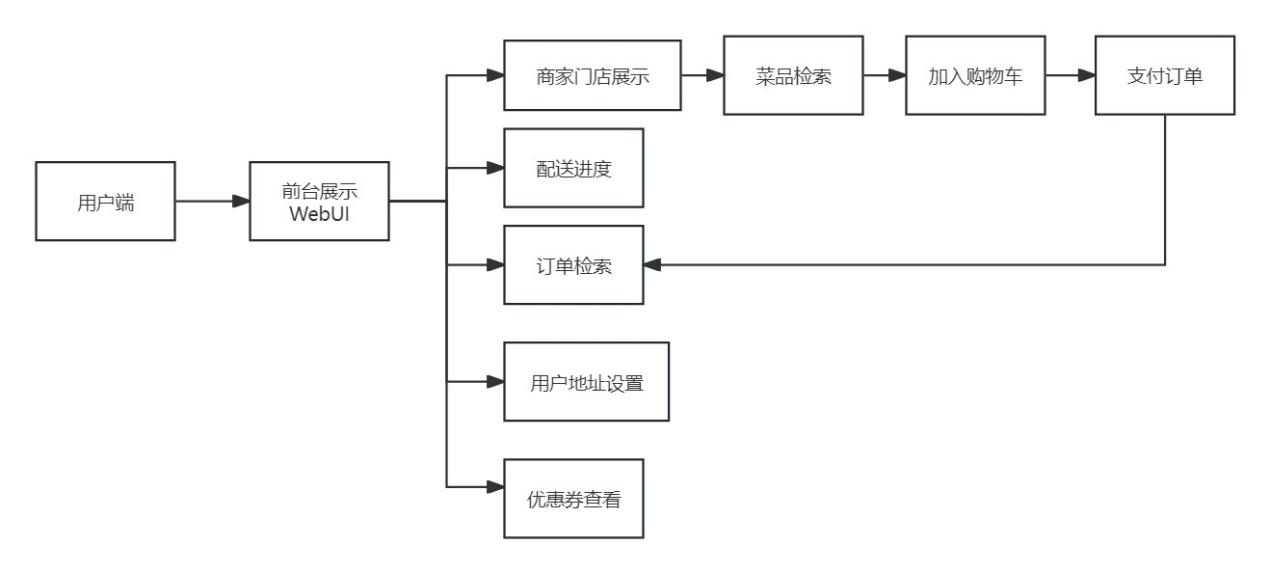


图1 用户端功能设计

3.2.2 商家端

商家端需要有对菜品分类与菜品的管理这是最为核心的、订单的管理，在订单管理中可以进行接单、拒单等操作，活动与优惠券的管理与数据统计。

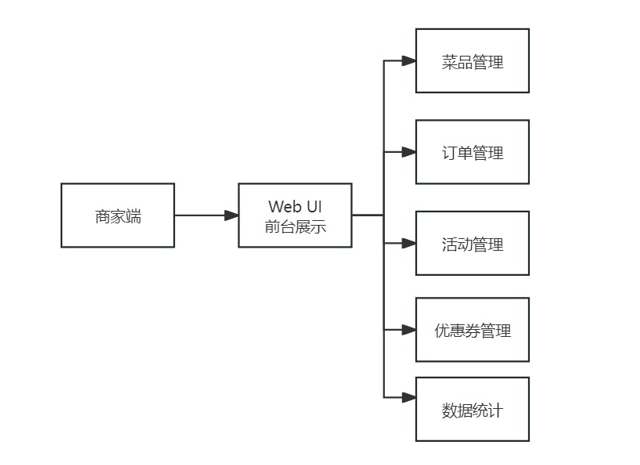


图2 商家端功能设计

3.2.3 骑手端

骑手端核心有以下几个功能设计：骑手接单中心，取餐、派送、路径规划、骑手个人中心。保证了骑手从接单到将餐配送到用户手里的整个流程。

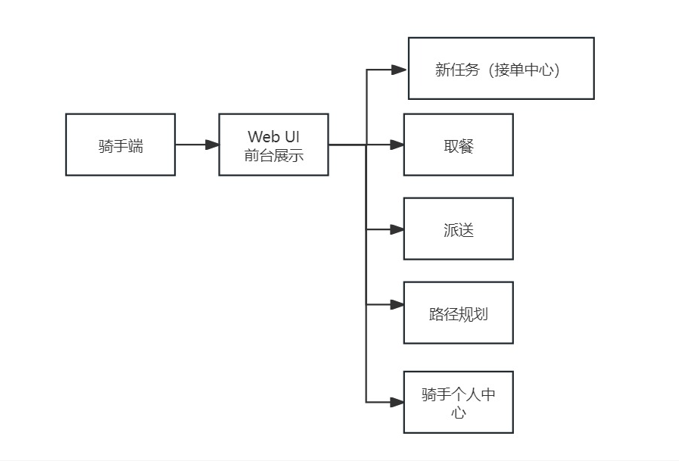


图3 骑手功能设计

数据库设计是系统设计相当重要的一个环节，一个系统的好坏有一大部分取决于数据库设计的好坏。本文的新生入校管理信息系统采用MYSQL进行设计。[[[1]](#endnote-0)]

## 3.4 系统E-R图

用户表E-R图设计，用户端为小程序，登录使用了微信登录，需要在数据库中存储登录微信用户的open\_id，如下图3.4.1。

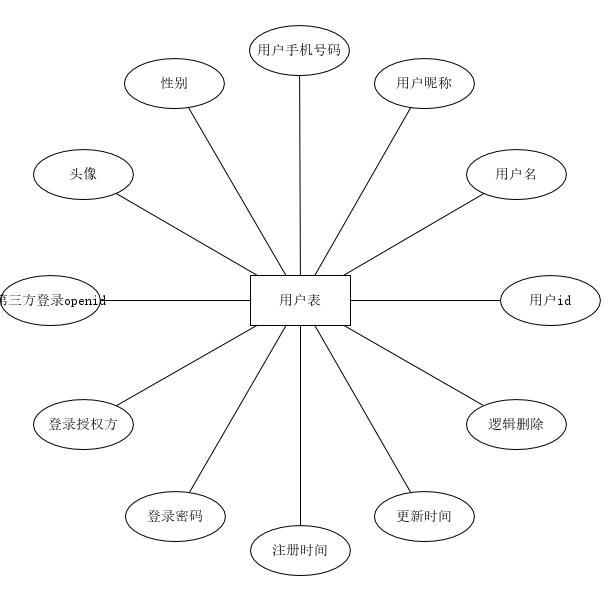


图4 用户表ER图

商家表E-R图设计，存储着用户的基本信息与经营信息、经营基本配置、经营状态除此还有店铺的位置信息。

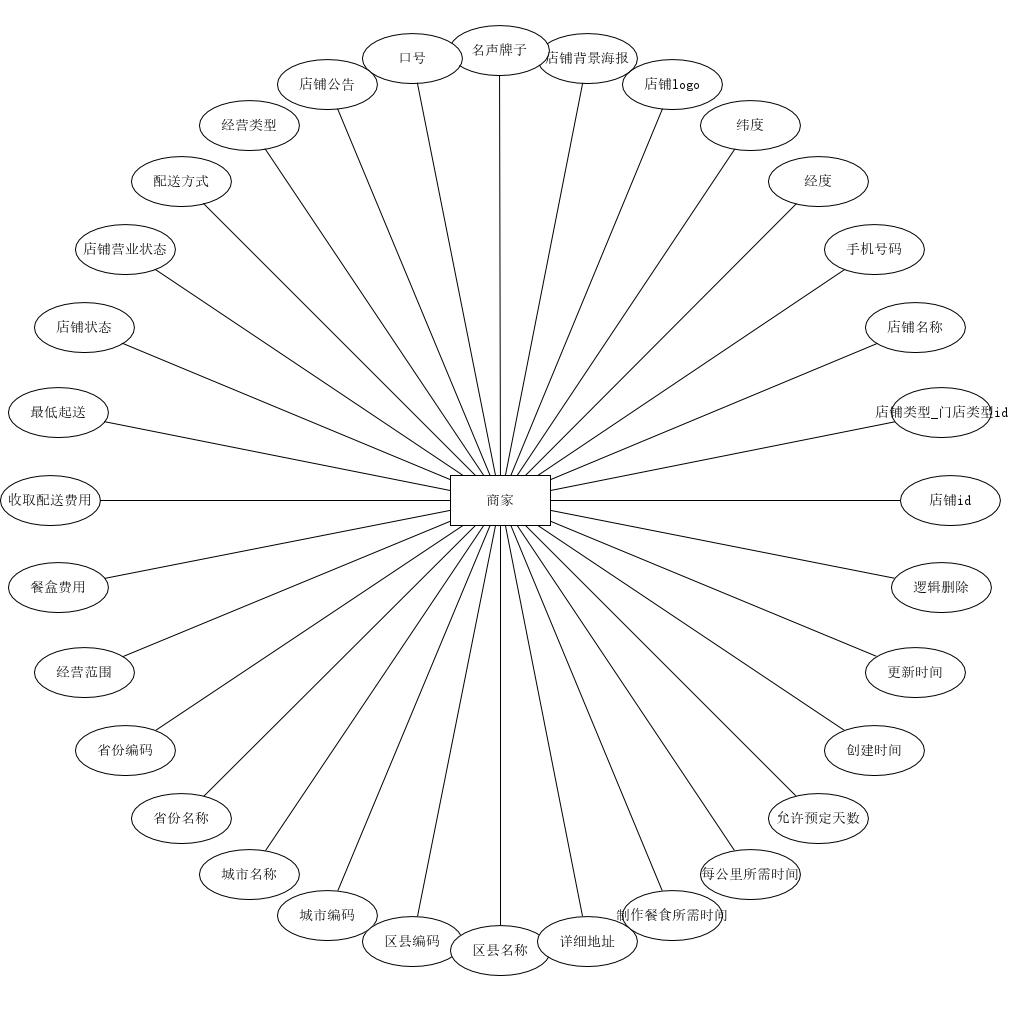


图5 店铺表ER图

配送订单表是在骑手表中，它是配送订单进行交于骑手配送的主表，保存是由哪个骑手进行配送，正常配送完后可所得的收益信息，还有保存着配送需要的其它信息。

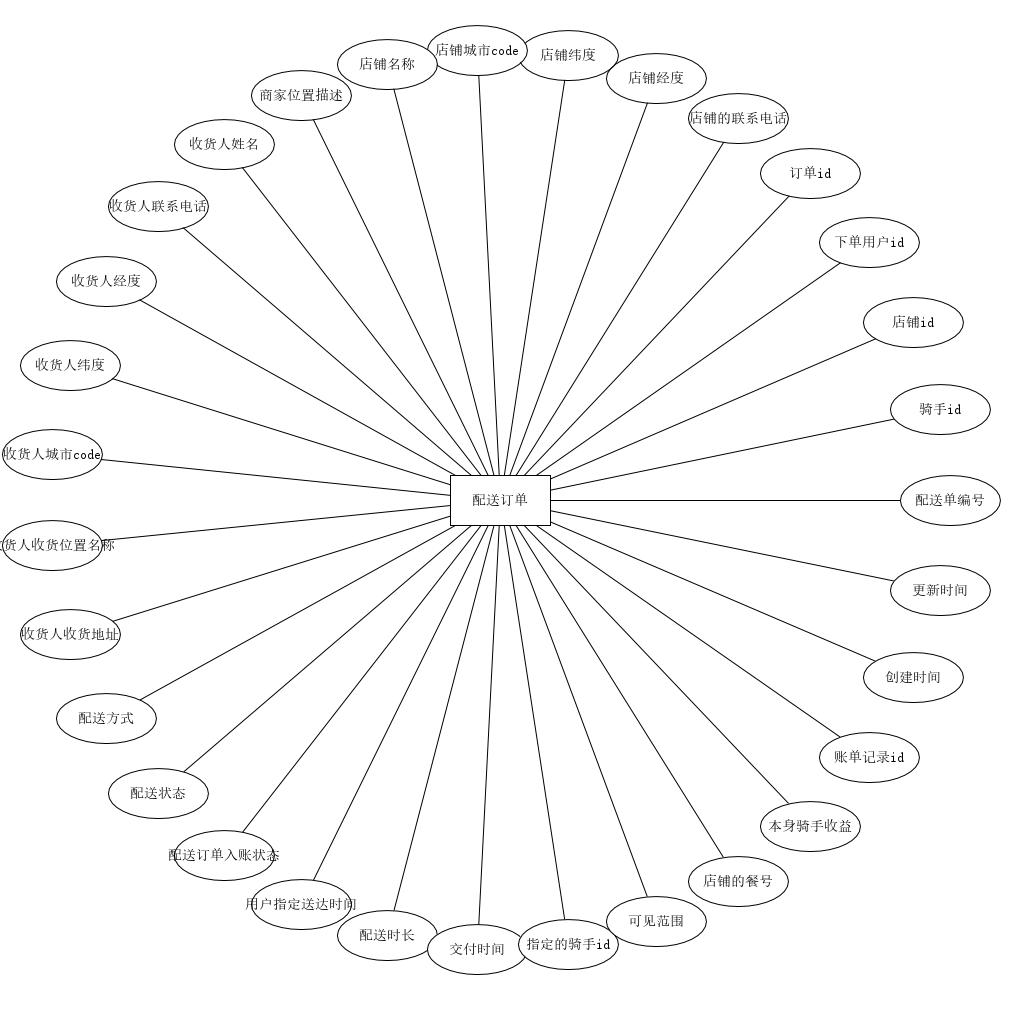


图6 配送订单表ER图

订单相关表E-R图，下面是订单表与订单明细表，一个订单有多个订单明细记录，所有的订单明细就是购物车信息，订单中存储中订单的取餐方式订单的code，如果是自提单，那就是取餐号，除此还保存着用户写的备注信息。而订单金额详细表，它保存着此订单所有与金额相关的运算参数值，如若订单为配送订单会保存有该订单配送费。

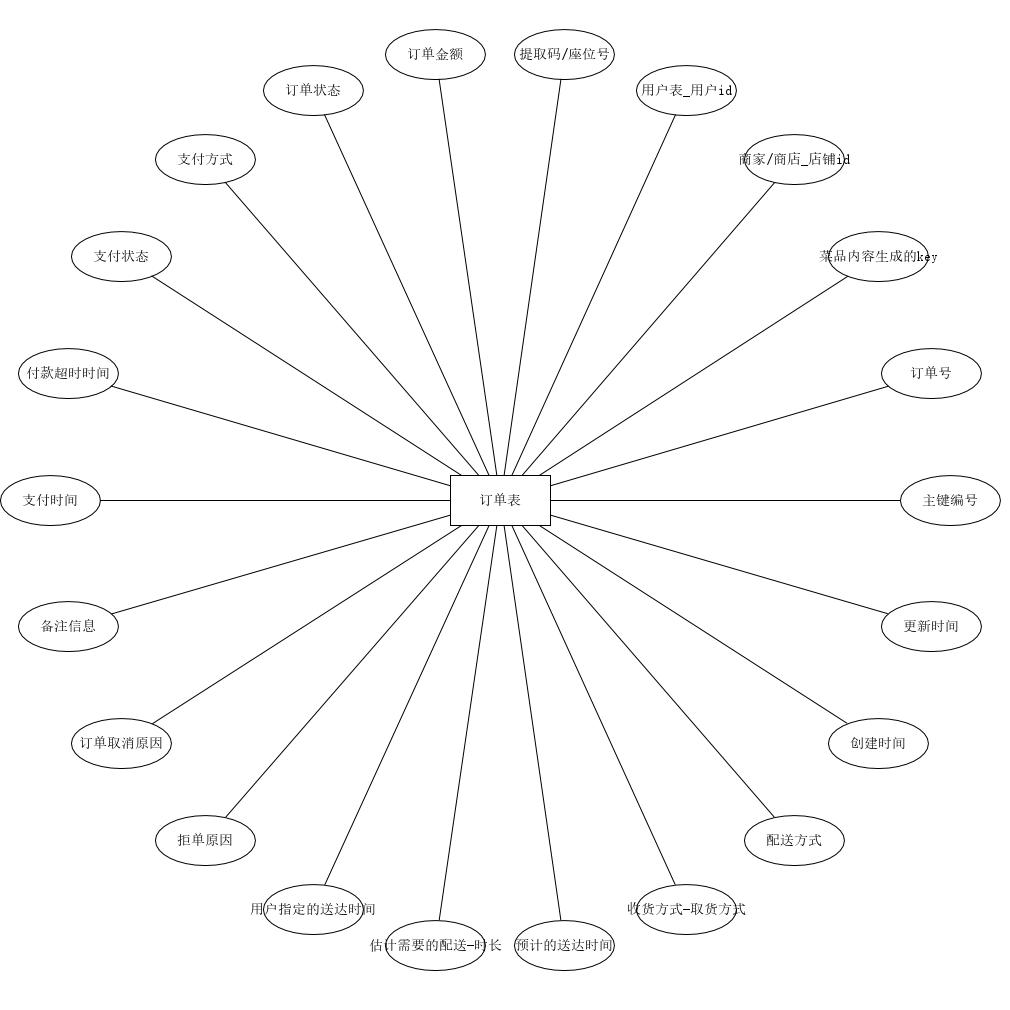


图7 订单表ER图

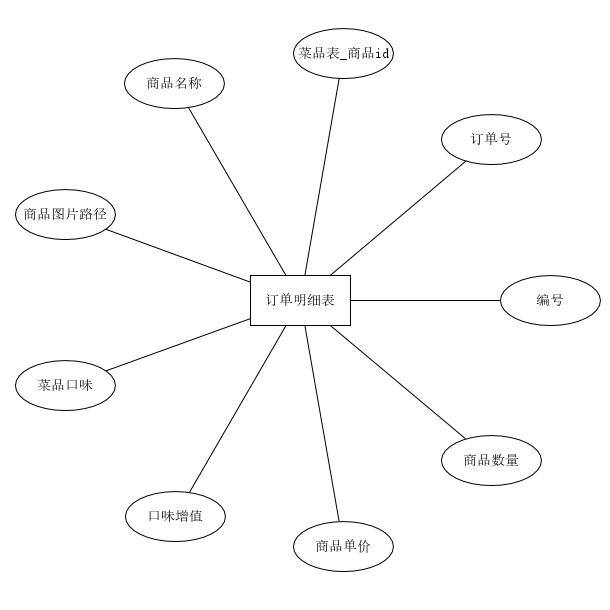


图8 订单明细表ER图

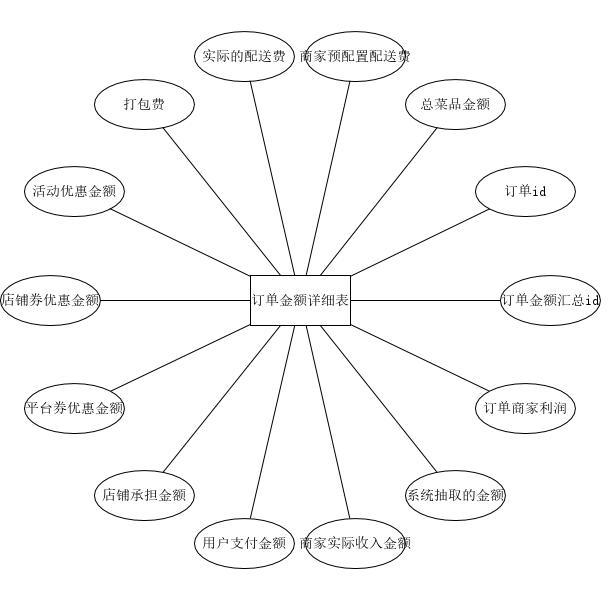


图9 订单金额详细表ER图

优惠券相关表如下，优惠券批次表，当商家创建优惠券时，操作的是这个表，即这个表保存着优惠券的核心信息，而优惠券实例表保存的是发放给用户的优惠券，只保存着实例信息如过期时间、膨胀金额等信息。

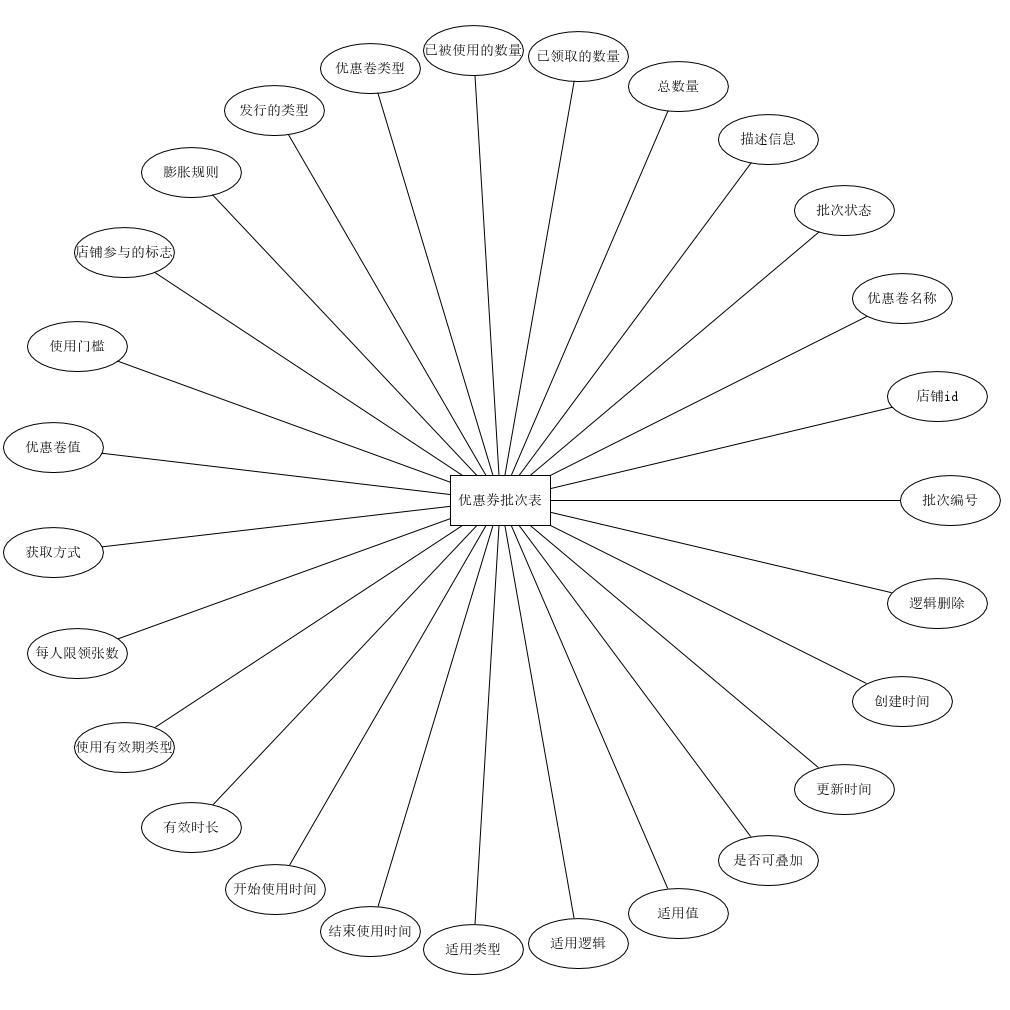


图10 优惠券批次表ER图

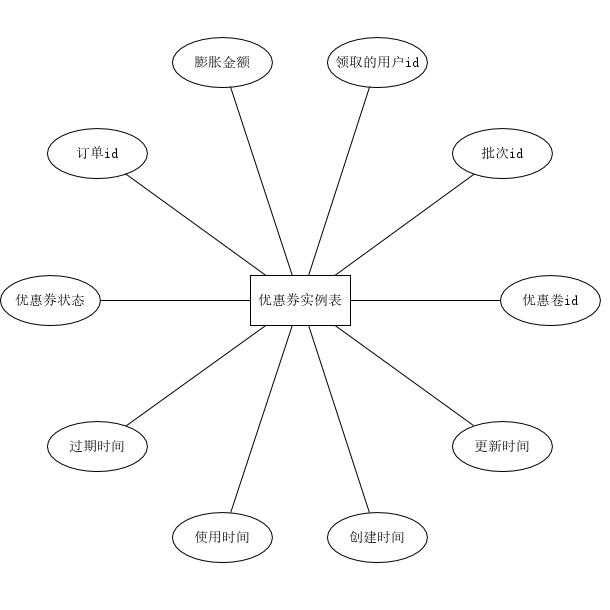


图11 优惠券实例表ER图

## 3.5 数据库表设计

### 3.5.1 用户模块表设计

表address\_book (收货地址表) ：包括经纬度及行政编码及用户收货地址信息如电话号码、联系人及地址信息，这些是表面的，最重要的是经纬度信息。

表1 用户收货地址表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | 地址id |
| 2 | user\_id | bigint | 20 | 用户表\_用户id |

续表1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 3 | consignee | varchar | 30 | 收货人 |
| 4 | sex | tinyint | 4 | 性别(0女，1男) |
| 5 | phone | varchar | 11 | 手机号码 |
| 6 | longitude | double | 23 | 经度 |
| 7 | latitude | double | 23 | 纬度 |
| 8 | province\_code | varchar | 12 | 省份编码 |
| 9 | province\_name | varchar | 32 | 省份名称 |
| 10 | city\_code | varchar | 12 | 城市编码 |
| 11 | city\_name | varchar | 32 | 城市名称 |
| 12 | district\_code | varchar | 12 | 区县编码 |
| 13 | district\_name | varchar | 32 | 区县名称 |
| 14 | detail\_address | varchar | 200 | 详细地址信息 |
| 15 | specific\_address | varchar | 128 | 具体地址（比如几楼几门牌号） |
| 16 | label | varchar | 32 | 标签 |
| 17 | last\_used\_time | datetime | 26 | 上次使用时间 |
| 18 | create\_time | datetime | 19 | 创建时间 |
| 19 | update\_time | datetime | 19 | 更新时间 |
| 20 | deleted | tinyint | 4 | 是否已经删除 |

表shop\_collect (店铺收藏表)：这个表是用户收藏店铺的信息表，包含了主键id，用户id，店铺id，以及添加收藏的时间。用户可以通过这个表收藏和查看他们喜欢的店铺。

表2 店铺收藏表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | 收藏编号 |
| 2 | user\_id | bigint | 20 | 用户id |
| 3 | shop\_info\_id | bigint | 20 | 店铺id |

续表2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 4 | add\_time | datetime | 19 | 收藏时间 |

表user (用户表)：用户表是存储用户信息的主要表，其中包含了用户的基本信息，如用户id，用户名，昵称，手机号，性别，头像，以及用户的第三方登录open\_id和平台。此外，用户的密码和帐号创建、更新时间，以及帐号是否被删除的信息也在此表中进行记录。每条用户信息都通过唯一的用户id进行标识和索引。

表3 用户表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | 用户id |
| 2 | username | varchar | 100 | 用户名 |
| 3 | nickname | varchar | 18 | 用户昵称 |
| 4 | phone | varchar | 11 | 用户手机号码 |
| 5 | sex | smallint | 6 | 性别(0表示女，1表示男) |
| 6 | avatar | varchar | 500 | 头像 |
| 7 | open\_id | varchar | 50 | 第三方登录openid |
| 8 | platform | varchar | 255 | 登录授权方（微信是1、... ） |
| 9 | password | varchar | 20 | 登录密码 |
| 10 | create\_time | datetime | 19 | 注册时间 |
| 11 | update\_time | datetime | 19 | 更新时间 |
| 12 | deleted | smallint | 6 | 逻辑删除（1已删除，0正常） |

表user\_asset (用户资产)：用户资产表主要用来记录用户的资产信息，包括用户的钱包金额和积分。每一条记录包含唯一的资产id，对应的用户id，钱包的金额，用户的积分，以及记录的创建和更新时间。这个表让用户能够查看和管理他们的资金和积分情况。

表4 用户资产表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | 用户资产id |

续表4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 2 | user\_id | bigint | 20 | 用户id |
| 3 | wallet\_amount | decimal | 11 | 钱包金额 |
| 4 | points | int | 10 | 积分 |
| 5 | create\_time | datetime | 19 | 创建时间 |
| 6 | update\_time | datetime | 19 | 更新时间 |

### 3.5.2 商家模块表设计

表comment (评价表)：主要记录了用户对商品的评价，包含评价星数、评价内容和评价时间等信息。

表5 评价表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | 评论编号 |
| 2 | user\_id | bigint | 20 | 用户表\_用户id |
| 3 | order\_number | bigint | 20 | 订单号 |
| 4 | rating\_type | smallint | 6 | 星数（1~5星） |
| 5 | comment | varchar | 512 | 评论文本 |
| 6 | create\_time | datetime | 19 | 评论时间 |
| 7 | update\_time | datetime | 19 | 更新时间 |
| 8 | deleted | tinyint | 4 | 逻辑删除 |

表dish (菜品表)：记录了菜品的基本信息，如名称、图片、价格，以及销售数量等。

表6 菜品表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | 商品id |
| 2 | dish\_category\_id | bigint | 20 | 商品分类\_分类id |
| 3 | shop\_id | bigint | 20 | 商家/商店\_店铺id |
| 4 | name | varchar | 50 | 菜品名称 |

续表6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 5 | image | longtext | 2147483647 | 商品样品图片 |
| 6 | labels | varchar | 100 | 标签项(多个标签用逗号隔开) |
| 7 | describe | varchar | 300 | 商品描述 |
| 8 | old\_price | decimal | 9 | 商品原价 |
| 9 | now\_price | decimal | 9 | 商品现价 |
| 10 | sell\_number | bigint | 20 | 卖出数量 |
| 11 | better\_display | tinyint | 4 | 是否优先展示（1优先展示，0普通展示） |
| 12 | status | smallint | 6 | 商品状态（0停售，1启售） |
| 13 | create\_time | datetime | 19 | 创建时间 |
| 14 | update\_time | datetime | 19 | 更新时间 |
| 15 | deleted | tinyint | 4 | 逻辑删除 |

表dish\_category (菜品分类表)：记录了商品的分类信息。

表7 菜品分类表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | 分类id |
| 2 | shop\_id | bigint | 20 | 商家/商店\_店铺id |
| 3 | name | varchar | 25 | 分类名 |
| 4 | sort | int | 10 | 排序字段 |
| 5 | status | int | 10 | 状态 |
| 6 | create\_time | datetime | 19 | 创建时间 |
| 7 | update\_time | datetime | 19 | 更新时间 |
| 8 | deleted | tinyint | 4 | 逻辑删除 |

表dish\_flavor (菜品口味表)：记录了菜品的口味，比如辣、酸、甜等。

表8 菜品口味表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | 口味项id |
| 2 | dish\_id | bigint | 20 | 菜品表\_商品id |
| 3 | name | varchar | 32 | 口味名称 |
| 4 | value | varchar | 255 | 口味值 |
| 5 | sort | tinyint | 4 | 显示顺序 |
| 6 | create\_time | datetime | 19 | 创建时间 |
| 7 | update\_time | datetime | 19 | 更新时间 |
| 8 | deleted | tinyint | 4 | 逻辑删除 |

表menu (菜单表)：记录了商家后台菜单menu的信息。

表9 菜单表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | 菜单id |
| 2 | parent\_id | bigint | 20 | 上级id |
| 3 | name | varchar | 64 | 对应前端文件名称 |
| 4 | label | varchar | 64 | 对应前端展示的名称 |
| 5 | route | varchar | 255 | 前端路由router路径 |
| 6 | component | varchar | 255 | 对应前端component |
| 7 | icon | varchar | 50 | 菜单图标 |
| 8 | perms | varchar | 255 | 授权(\*,admin,employee) |
| 9 | status | int | 10 | 状态：1启用，2禁用 |
| 10 | sort | int | 10 | 排序 |
| 11 | create\_time | datetime | 19 | 创建时间 |
| 12 | update\_time | datetime | 19 | 更新时间 |
| 13 | deleted | int | 10 | 逻辑删除 |

表merchant\_account (商家账号)：记录了商家的账号信息。

表10 商家账号表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | 编写 |
| 2 | shop\_id | bigint | 20 | 商家/商店\_店铺id |
| 3 | username | varchar | 16 | 用户名 |
| 4 | password | varchar | 255 | 密码 |
| 5 | name | varchar | 32 | 姓名 |
| 6 | phone | varchar | 11 | 手机号 |
| 7 | sex | smallint | 6 | 姓别（0女1男） |
| 8 | status | smallint | 6 | 状态(1正常 0锁定) |
| 9 | is\_admin | smallint | 6 | 是否为商家管理员账号 |
| 10 | create\_time | datetime | 19 | 创建时间 |
| 11 | update\_time | datetime | 19 | 最后修改时间 |
| 12 | deleted | tinyint | 4 | 逻辑删除 |

表shop (商家/商店)：记录了店铺的基本信息，比如名称、电话、地址。

表11 商家表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | 店铺id |
| 2 | shop\_category\_id | bigint | 20 | 店铺类型\_门店类型id |
| 3 | name | varchar | 1024 | 店铺名称 |
| 4 | phone | varchar | 11 | 手机号码 |
| 5 | longitude | double | 23 | 经度 |
| 6 | latitude | double | 23 | 纬度 |
| 7 | logo | varchar | 200 | 店铺logo |
| 8 | background\_poster | varchar | 200 | 店铺背景海报图片 |
| 9 | reputation | varchar | 128 | 名声牌子（1-品牌id,2-优选id） |

续表11

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 10 | flags | varchar | 128 | 口号（flags）,每个flag间逗号隔开 |
| 11 | notice | text | 65535 | 店铺公告 |
| 12 | business\_type | varchar | 6 | 经营类型(1支持配送、2支持店食、3、支持自提）可多选逗号隔开 |
| 13 | delivery\_method | tinyint | 4 | 配送方式（1平台配送，2商家配送） |
| 14 | business\_status | tinyint | 4 | 店铺营业状态（1营业中，0已打烊） |
| 15 | account\_status | smallint | 6 | 店铺状态（0待核实入驻、1已入驻、-1冻结中） |
| 16 | minimum\_delivery\_amount | decimal | 11 | 最低起送 |
| 17 | delivery\_fee | decimal | 11 | 商家保障配送费不超过的费用，超过的部分（从商家单中扣除给外卖员 |
| 18 | packing\_fee | decimal | 11 | 餐盒费用 |
| 19 | business\_scope | double | 23 | 经营范围（公里） |
| 20 | province\_code | varchar | 12 | 省份编码 |
| 21 | province\_name | varchar | 32 | 省份名称 |
| 22 | city\_name | varchar | 32 | 城市名称(地级市,...) |
| 23 | city\_code | varchar | 12 | 城市编码 |
| 24 | district\_code | varchar | 12 | 区县编码(县级市,...) |
| 25 | district\_name | varchar | 32 | 区县名称 |
| 26 | address\_describe | varchar | 200 | 详细地址/具体地址描述 |
| 27 | make\_time | int | 10 | 制作餐食所需时间（单位是分钟） |
| 28 | delivery\_time\_per\_km | int | 10 | 每公里所需的时间 |

续表11

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 29 | advance\_booking\_days | int | 10 | 允许提前预定的天数(-1，表示不允许预定，既然是今天，0是今天，1明天，..) |
| 30 | create\_time | datetime | 19 | 创建时间 |
| 31 | update\_time | datetime | 19 | 更新时间 |
| 32 | deleted | tinyint | 4 | 逻辑删除 |

表shopping\_cart (购物车项)：记录了用户的购物车信息。

表12 购物车项表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | 主键 |
| 2 | user\_id | bigint | 20 | 用户表\_用户id |
| 3 | shop\_id | bigint | 20 |  |
| 4 | name | varchar | 32 | 商品名称 |
| 5 | image | varchar | 255 | 商品图片路径 |
| 6 | dish\_flavor | varchar | 50 | 菜品口味 |
| 7 | count | int | 10 | 商品数量 |
| 8 | old\_price | decimal | 11 | 商品原价 |
| 9 | now\_price | decimal | 11 | 商品现价 |
| 10 | create\_time | datetime | 19 | 创建时间 |
| 11 | update\_time | datetime | 19 | 修改时间 |

表shop\_activity (商家活动)：该表决定了商家能进行怎样的活动。

表14 商家活动表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | 活动id（1:满减活动,2:新客立减 3减配送费 4神会员膨胀券-参与标志 ） |
| 2 | name | varchar | 128 | 活动名称 |

续表14

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 3 | describe | varchar | 255 | 描述信息 |
| 4 | flags | varchar | 255 | 作用flags(多个用逗号隔开) |
| 5 | striking\_icon | varchar | 1000 | 活动icon |
| 6 | striking\_char | char | 1 | 活动特征字符 |
| 7 | striking\_color | char | 7 | 活动特征颜色 |
| 8 | deleted | tinyint | 4 | 逻辑删除 |
| 9 | create\_time | datetime | 19 | 创建时间 |
| 10 | update\_time | datetime | 19 | 更新时间 |

表shop\_activity\_instance (商家活动实例)：记录的是商家创建的活动。

表15 商家活动实例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | 参与活动id |
| 2 | shop\_id | bigint | 20 | 店铺id |
| 3 | activity\_id | bigint | 20 | 活动id |
| 4 | title | varchar | 128 | 活动名（自动生成，如满12减2） |
| 5 | threshold | varchar | 8 | 门槛（各类型表示的值不同） |
| 6 | value\_type | tinyint | 4 | 优惠值类型（1:按金额 2:按折扣 3:活动特征值） |
| 7 | value | varchar | 8 | 优惠值 |
| 8 | customer\_type | smallint | 6 | 适用顾客类型（0:全部用户 1:门店新客 ） |
| 9 | start\_date | datetime | 19 | 活动开始时间 |
| 10 | end\_date | datetime | 19 | 活动结束时间 |
| 11 | weeks | varchar | 255 | 适用的周(1,2,3...,7) |
| 12 | time\_ranges | varchar | 255 | 适用的时间段(12:00-14:00,...) |

续表15

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 13 | create\_time | datetime | 19 | 创建时间 |
| 14 | update\_time | datetime | 19 | 更新时间 |
| 15 | deleted | tinyint | 4 | 逻辑删除 |

表shop\_asset (骑手资产)：记录了商家的资产信息，例如保证金和钱包金额。

表16 骑手资产表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | id |
| 2 | shop\_id | bigint | 20 | 店铺id |
| 3 | margin | decimal | 7 | 保证金 |
| 4 | wallet\_amount | decimal | 11 | 钱包金额 |
| 5 | create\_time | datetime | 19 | 创建时间 |
| 6 | update\_time | datetime | 19 | 更新时间 |

表shop\_business\_rules ：店铺的经营时间规则：记录了店铺的经营时间规则。

表17 店铺经营时间规则表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | id编号 |
| 2 | shop\_id | bigint | 20 | 店铺id |
| 3 | type | tinyint | 4 | 类型 1普通类型、2特殊类型 |
| 4 | weeks | varchar | 255 | 若是普通类型读此列（周） |
| 5 | start\_date | date | 10 | 若是特殊类型是读 |
| 6 | end\_date | date | 10 | 若是特殊类型是读 |
| 7 | times | varchar | 255 | 若干的时间段(12:30~14:30,17:30~18:00,...) |

表shop\_category (店铺类型)：记录了店铺的类型，比如快餐、烧烤等。

表18 店铺类型表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | 门店类型id |
| 2 | name | varchar | 64 | 门店类型名称 |
| 3 | sort | int | 10 | 排序优先级 |
| 4 | icon | varchar | 255 | 类型图标 |
| 5 | create\_time | datetime | 19 | 创建时间 |
| 6 | update\_time | datetime | 19 | 更新时间 |
| 7 | deleted | tinyint | 4 | 逻辑删除 |

表shop\_info (店铺资料表)：记录了店铺入驻的详细资料，例如营业执照和身份证照片。

表19 店铺资料表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | 店铺资料ID |
| 2 | shop\_id | bigint | 20 | 商家/商店\_店铺id |
| 3 | business\_license\_image | varchar | 200 | 营业执照 |
| 4 | permit\_image | varchar | 200 | 食品安全许可证 |
| 5 | id\_card\_image | varchar | 200 | 法定代表人手持身份证照片 |
| 6 | create\_time | datetime | 19 | 创建时间 |
| 7 | update\_time | datetime | 19 | 更新时间 |
| 8 | deleted | tinyint | 4 | 逻辑删除 |

表shop\_money\_record (收益进出账记录表)：记录了店铺的收入和支出。

表20 收益进出账记录表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | id |
| 2 | shop\_id | bigint | 20 | 店铺id |
| 3 | operation\_amount | decimal | 11 | 操作金额 |

续表20

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 4 | status | tinyint | 4 | 状态（0 等待入账 1已入账 2入账中断） |
| 5 | source\_describe | varchar | 128 | 说明(如：接单入账) |
| 6 | earliest\_arrival\_time | datetime | 19 | 最早到账时间 |
| 7 | create\_time | datetime | 19 | 创建时间 |
| 8 | update\_time | datetime | 19 | 更新时间 |

表shop\_order\_accounting\_record (店铺订单入账记录表)：记录了店铺的订单入账记录。

表21 店铺订单入账记录表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | 订单-入账记录id |
| 2 | order\_id | bigint | 20 | 订单id |
| 3 | money\_record\_id | bigint | 20 | 入账记录id |

表shop\_self\_pickup\_serial\_number (店铺自提自增)：记录了店铺自提/配送的next自增序号。

表22 店铺自提自境表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | 店铺自提自增状态编号 |
| 2 | shop\_id | bigint | 20 | 对应的自提订单id |
| 3 | last\_gen\_date | date | 10 | 店铺最近生成自提码的date |
| 4 | queue\_sequence\_number | int | 10 | 排队序号 |

### 3.5.3 订单模块表设计

表order (订单表)：该表主要记录了订单的详细信息，包括订单编号，订单金额，支付方式和状态，订单创建和更新时间等等。

表23 订单表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | 主键编号 |
| 2 | order\_no | varchar | 64 | 订单号 |
| 3 | order\_key | varchar | 128 | 菜品内容生成的key |
| 4 | shop\_id | bigint | 20 | 商家/商店\_店铺id |
| 5 | user\_id | bigint | 20 | 用户表\_用户id |
| 6 | code | varchar | 11 | 提取码/座位号 |
| 7 | amount | decimal | 11 | 订单金额 |
| 8 | state | smallint | 6 | 订单状态（1待支付 2待接单 3已接单 4派送中/制作完成 5已完成 6申请售后 7已取消） |
| 9 | pay\_method | smallint | 6 | 支付方式（1微信支付 2支付宝支付） |
| 10 | pay\_status | smallint | 6 | 支付状态（0未支付 1已支付 2退款） |
| 11 | pay\_timeout\_time | datetime | 19 | 付款超时时间 |
| 12 | pay\_time | datetime | 19 | 支付时间 |
| 13 | remark | varchar | 200 | 备注信息 |
| 14 | cancel\_reason | varchar | 255 | 订单取消原因 |
| 15 | rejection\_reason | varchar | 255 | 拒单原因 |
| 16 | designated\_delivery\_time | datetime | 19 | 用户指定的送达时间 |
| 17 | delivery\_duration | int | 10 | 估计需要的配送-时长 |
| 18 | estimated\_delivery\_time | datetime | 19 | 预计的送达时间 |
| 19 | receiving\_method | char | 1 | 收货方式-取货方式(1店食、2自提、3配送) |
| 20 | delivery\_method | tinyint | 4 | 配送方式（1平台配送，2商家自配） |
| 21 | create\_time | datetime | 19 | 创建时间 |
| 22 | update\_time | datetime | 19 | 更新时间 |

表delivery\_info 订单的配送信息即订单中用户的收货地址信息

表24 订单配送信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | 订单配送信息编号 |
| 2 | order\_id | bigint | 20 | 对应的订单id |
| 3 | consignee | varchar | 30 | 收货人 |
| 4 | sex | tinyint | 4 | 性别(0女，1男) |
| 5 | phone | varchar | 11 | 手机号码 |
| 6 | longitude | double | 23 | 经度 |
| 7 | latitude | double | 23 | 纬度 |
| 8 | province\_code | varchar | 12 | 省份编码 |
| 9 | province\_name | varchar | 32 | 省份名称 |
| 10 | city\_code | varchar | 12 | 城市编码 |
| 11 | city\_name | varchar | 32 | 城市名称 |
| 12 | district\_code | varchar | 12 | 区县编码 |
| 13 | district\_name | varchar | 32 | 区县名称 |
| 14 | detail\_address | varchar | 200 | 详细地址信息 |
| 15 | specific\_address | varchar | 128 | 具体地址（比如几楼几门牌号） |

表dish\_rating (订单商品评价表)：该表主要记录了对订单中商品的评价信息。

表25 订单商品评价表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | 编号 |
| 2 | order\_rating\_id | bigint | 20 | 评论编号 |
| 3 | dish\_id | bigint | 20 | 菜品表\_商品id |
| 4 | rating\_type | smallint | 6 | 表态（0踩，1赞） |

订单活动表 order\_activity ：用于保存的是订单适用的活动与活动减免的优惠。

表26 订单活动表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | 订单-活动id |
| 2 | order\_id | bigint | 20 | 订单id |
| 3 | activity\_id | bigint | 20 | 活动类型id |
| 4 | activity\_instance\_title | varchar | 128 | 活动标题 |
| 5 | activity\_instance\_id | bigint | 20 | 活动实例id |
| 6 | deduction\_amount | decimal | 9 | 优惠抵扣金额 |

表order\_amount (订单金额详细表)：该表主要记录了订单的金额详细信息，包括各种费用的金额和折扣的金额。

表27 订单金额详细表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | 订单金额汇总id |
| 2 | order\_id | bigint | 20 | 订单id |
| 3 | total\_dish\_amount | decimal | 11 | 总菜品金额 |
| 4 | preset\_delivery\_fee | decimal | 11 | 商家预配置配送费 |
| 5 | actual\_delivery\_fee | decimal | 11 | 实际的配送费 |
| 6 | packing\_fee | decimal | 11 | 打包费 |
| 7 | shop\_activity\_discount\_amount | decimal | 11 | 店铺活动优惠金额 |
| 8 | shop\_coupon\_discount\_amount | decimal | 11 | 店铺优惠券优惠金额 |
| 9 | platform\_coupon\_discount\_amount | decimal | 11 | 平台券优惠金额 |
| 10 | borne\_platform\_coupon\_amount | decimal | 11 | 店铺承担平台券金额 |
| 11 | user\_payment\_amount | decimal | 11 | 用户支付金额 |
| 12 | order\_shop\_income | decimal | 11 | 商家订单实际收入金额 |
| 13 | platform\_revenue | decimal | 11 | 系统抽取的金额 |
| 14 | order\_shop\_profit | decimal | 11 | 订单商家利润 |

订单优惠券表 order\_coupon ：是订单使用优惠券的订单表。

表28 订单优惠券表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | 订单-优惠券id |
| 2 | order\_id | bigint | 20 | 订单id |
| 3 | coupon\_name | varchar | 128 | 优惠券名称 |
| 4 | coupon\_id | bigint | 20 | 优惠券id |
| 5 | deduction\_amount | decimal | 9 | 优惠抵扣金额 |
| 6 | shop\_bearing\_amount | decimal | 9 | 店铺需要承担的金额 |

表order\_detail (订单明细表)：该表主要记录了订单中购买的所有商品的详细信息，包括商品的单价、数量等。

表29 订单明细表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | 编号 |
| 2 | order\_id | bigint | 20 | 订单号 |
| 3 | dish\_id | bigint | 20 | 菜品表\_商品id |
| 4 | name | varchar | 32 | 商品名称 |
| 5 | image | varchar | 255 | 商品图片路径 |
| 6 | dish\_flavor | varchar | 50 | 菜品口味 |
| 7 | appreciation | decimal | 9 | 口味增值 |
| 8 | spec\_price | decimal | 9 | 商品单价 |
| 9 | count | int | 10 | 商品数量 |

表order\_for\_search (用于搜索订单信息)：该表主要设计用于订单的搜索，包括订单的菜品枚举和店铺名称。

表30 用户搜索订单信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | id |

续表30

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 2 | order\_id | bigint | 20 | 订单id |
| 3 | dishes | varchar | 255 | 订单的菜品枚举(菜品1,菜品2,...) |
| 4 | shop\_name | varchar | 128 | 店铺名称 |

表rating (评价表)：该表主要记录了用户对订单的评论，包括星级、评论文本和评论时间中。

表31 评价表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | 评论编号 |
| 2 | user\_id | bigint | 20 | 用户表\_用户id |
| 3 | order\_number | bigint | 20 | 订单号 |
| 4 | star\_rating | smallint | 6 | 星数（1~5星） |
| 5 | text | varchar | 512 | 评论文本 |
| 6 | create\_time | datetime | 19 | 评论时间 |
| 7 | update\_time | datetime | 19 | 更新时间 |

### 3.5.4 优惠券模块表设计

coupon 优惠券：储存了一些关于优惠券的基本信息，如优惠券ID、批次ID、用户ID、扩展金额（指优惠券的金额膨胀），以及优惠券的状态、过期时间、使用时间、创建时间和更新时间等。

表32 优惠券表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | 优惠卷id |
| 2 | batch\_id | bigint | 20 | 批次id |
| 3 | user\_id | bigint | 20 | 领取的用户id |
| 4 | expansion\_amount | decimal | 9 | 膨胀的那部分优惠金额(n+m的m) |
| 5 | order\_id | bigint | 20 | 使用优惠卷的订单id |

续表32

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 6 | state | tinyint | 4 | 优惠券状态（0未使用，1 锁定中(订单未完成) 2已使用 -1已失效（券使用退货后券失效场景）） |
| 7 | expiration\_time | datetime | 19 | 过期时间 |
| 8 | usage\_time | datetime | 19 | 使用时间 |
| 9 | create\_time | datetime | 19 | 创建时间 |
| 10 | update\_time | datetime | 19 | 更新时间 |

表coupon\_batch 优惠券批次：储存了优惠券批次的元数据，如批次编号、店铺ID、优惠券名称、优惠券类型和发行类型，以及一些其他的特性，如膨胀规则、适用类型和逻辑、是否可以叠加使用等。

表33 优惠券批次表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | 批次编号 |
| 2 | shop\_id | bigint | 20 | 店铺id（仅店铺卷，平台卷null） |
| 3 | name | varchar | 128 | 优惠卷名称 |
| 4 | status | tinyint | 4 | 批次状态（0不可领取 1可领取） |
| 5 | describe | varchar | 255 | 描述信息（如规则描述） |
| 6 | total | int | 10 | 总数量 |
| 7 | issued\_count | int | 10 | 已领取的数量 |
| 8 | used\_count | int | 10 | 已被使用的数量 |
| 9 | type | tinyint | 4 | 优惠卷类型(0满减/无门槛卷/代金券，1折扣卷) |
| 10 | issue\_type | tinyint | 4 | 发行的类型（0平台卷、1店铺卷） |
| 11 | expansion\_rule | varchar | 255 | 膨胀规则-为空不可膨胀{"膨胀后金额":概率值,...}（issue\_type=0） |

续表33

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 12 | shop\_participation\_flag | varchar | 128 | 店铺参与的标志/店铺参与的平台活动标志（使用平台膨胀卷时需要看店铺是否有这个授权标志） |
| 13 | threshold\_amount | decimal | 9 | 最低消费多少金额可用优惠券（门槛） |
| 14 | par\_value | decimal | 9 | 优惠卷值(根据type：抵扣金额 | 折扣率) |
| 15 | obtain\_way | tinyint | 4 | 获取方式（ -1:兑换码 , 1:店铺公开展示-手动领取. 2:非公开群内手动领取 ） |
| 16 | per\_person\_limit | int | 10 | 每人限领张数 |
| 17 | usage\_validity\_period\_type | tinyint | 4 | 使用有效期类型（1:固定日期 2:相对有效期） |
| 18 | usage\_period\_days | int | 10 | 自领取之日起有效天数（单位：天） |
| 19 | usage\_start\_time | datetime | 19 | 开始使用时间 |
| 20 | usage\_end\_time | datetime | 19 | 结束使用时间 |
| 21 | applicable\_type | tinyint | 4 | 适用类型（0:无适用限制 1:适用类别 2:适用商品） |
| 22 | applicable\_logic | tinyint | 4 | 适用逻辑（1:适用选择 2:不适用选择的） |
| 23 | applicable\_values | varchar | 255 | 适用值（选择的类目ids或商品ids） |
| 24 | is\_stackable | bit | 1 | 是否可以叠加使用(0 不可叠加 1可叠加) |
| 25 | update\_time | datetime | 19 | 更新时间 |
| 26 | create\_time | datetime | 19 | 创建时间 |
| 27 | deleted | tinyint | 4 | 逻辑删除 |

表coupon\_order 省钱包订单：则是记录了优惠券订单的信息，包括订单ID、用户ID、优惠券产品ID、支付金额、以及订单状态、创建时间和更新时间等。

表34 省钱包订单表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | 优惠卷订单id |
| 2 | user\_id | bigint | 20 | 购买的用户id |
| 3 | coupon\_product\_id | bigint | 20 | 优惠卷产品id |
| 4 | payment\_amount | decimal | 11 | 支付的金额 |
| 5 | state | tinyint | 4 | 状态（0待支付，1已购买 -1未使用退购 -2已取消） |
| 6 | create\_time | datetime | 19 | 创建时间 |
| 7 | update\_time | datetime | 19 | 更新时间 |

表coupon\_product 省钱包产品

优惠券产品表（coupon\_product）和优惠券产品批次表（coupon\_product\_batch）主要是关于优惠券产品和批次的定义。在优惠券产品表中，记录了产品ID、价格、优惠金额、状态，以及产品名称、描述、创建时间和更新时间等。而在优惠券产品批次表中，则主要保存了与产品批次相关的内容，包括产品ID、批次ID以及数量

表35 省钱包产品表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | 卷商品id |
| 2 | price | decimal | 11 | 价格 |
| 3 | preferential | decimal | 11 | 优惠（价格上减去优惠价格） |
| 4 | status | tinyint | 4 | 状态（0 不可买 1可买） |
| 5 | name | varchar | 128 | 名称 |
| 6 | describe | varchar | 255 | 描述信息 |
| 7 | create\_time | datetime | 19 | 创建时间 |
| 8 | update\_time | datetime | 19 | 更新时间 |

表coupon\_product\_batch 优惠券批次表

表36 优惠券批次表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | id |
| 2 | product\_id | bigint | 20 | 卷商品id |
| 3 | batch\_id | bigint | 20 | 卷批id |
| 4 | number | int | 10 | 数量 |

### 3.5.5 骑手模块表设计

表delivery\_order 配送订单表

表37 配送订单表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | 配送单编号 |
| 2 | rider\_id | bigint | 20 | 骑手id |
| 3 | shop\_id | bigint | 20 | 店铺id |
| 4 | user\_id | bigint | 20 | 下单用户id |
| 5 | order\_id | bigint | 20 | 订单id |
| 6 | shop\_phone | varchar | 16 | 店铺的联系电话 |
| 7 | shop\_latitude | double | 23 | 店铺位置geo-经度 |
| 8 | shop\_longitude | double | 23 | 店铺位置geo-纬度 |
| 9 | shop\_city\_code | varchar | 12 | 店铺所在城市code |
| 10 | shop\_name | varchar | 64 | 比如XXX店 |
| 11 | shop\_address\_describe | varchar | 255 | 商家描述的具体位置 |
| 12 | consignee\_fullname | varchar | 32 | 收货人姓名 |
| 13 | consignee\_phone | varchar | 16 | 收货人的联系电话 |
| 14 | consignee\_latitude | double | 23 | 收货人地址geo-经度 |
| 15 | consignee\_longitude | double | 23 | 收货人收货地址geo-纬度 |

续37

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 16 | consignee\_city\_code | varchar | 12 | 收货人所在城市code |
| 17 | consignee\_address\_name | varchar | 255 | 收货人收货位置名称（如XX公寓） |
| 18 | consignee\_address\_describe | varchar | 255 | 收货人收货地址描述的具体位置 |
| 19 | delivery\_method | tinyint | 4 | 配送方式（1平台配送，2商家配送） |
| 20 | delivery\_state | tinyint | 4 | 1 待骑手接单 2 骑手已接单 3骑手已到店 4 骑手已取单 5 已交付给用户 |
| 21 | income\_state | tinyint | 4 | 1 待完成 2 待入账 3 已入账 |
| 22 | designated\_delivery\_time | datetime | 19 | 用户指定的送达时间 |
| 23 | delivery\_duration | int | 10 | 配送时长（分钟） |
| 24 | delivery\_time | datetime | 19 | 交付时间（骑手接单后设置） |
| 25 | assign\_rider\_id | bigint | 20 | 指定的骑手id（null时表示可自由抢单） |
| 26 | visible\_range | double | 23 | 可见范围（动态扩大） |
| 27 | code | int | 10 | 店铺的餐号 |
| 28 | income | decimal | 7 | 完成后骑手所得收益（并非实际收益，超时后开始扣款，当超时达到配送时间时收益为-50%） |
| 29 | transfer\_accounts\_id | bigint | 20 | 账单记录id |
| 30 | create\_time | datetime | 19 | 创建时间 |
| 31 | update\_time | datetime | 19 | 更新时间 |

骑手表 rider ：包括骑手的登录信息

表38 骑手表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | 骑手id |
| 2 | stationmaster\_id | bigint | 20 | 所属站长id |
| 3 | phone\_area\_code | int | 10 | 手机号码地区区号前缀 |
| 4 | phone\_number | varchar | 11 | 手机号/登录用户名 |
| 5 | password | varchar | 255 | 密码 |
| 6 | full\_name | varchar | 32 | 骑手姓名 |
| 7 | type | smallint | 6 | 骑手类型（ 1专送 2众包） |
| 8 | state | smallint | 6 | -1已冻结 0待审核、1正常 |
| 9 | id\_card | varchar | 18 | 身份证 |
| 10 | work\_status | smallint | 6 | 0 已下线 1 已上线 |
| 11 | create\_time | datetime | 19 | 创建时间 |
| 12 | update\_time | datetime | 19 | 更新时间 |

表rider\_asset (骑手资产)：保存着骑手的金额相关信息，如钱包金额。

表39 骑手资产表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | id |
| 2 | rider\_id | bigint | 20 | 骑手id |
| 3 | margin | decimal | 7 | 保证金 |
| 4 | wallet\_amount | decimal | 11 | 钱包金额 |
| 5 | create\_time | datetime | 19 | 创建时间 |
| 6 | update\_time | datetime | 19 | 更新时间 |

表rider\_money\_record (收益进出账记录表)

表40 用户收货地址表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | id |
| 2 | rider\_id | bigint | 20 | 骑手id |
| 3 | operation\_amount | decimal | 11 | 操作金额 |
| 4 | status | tinyint | 4 | 状态（0 等待入账 1已入账 2入账中断） |
| 5 | source\_describe | varchar | 128 | 说明(如：接单入账) |
| 6 | earliest\_arrival\_time | datetime | 19 | 最早到账时间 |
| 7 | create\_time | datetime | 19 | 创建时间 |
| 8 | update\_time | datetime | 19 | 更新时间 |

表rider\_order\_accounting\_record (骑手配送订单入账记录表)

表41 骑手配送订单入账记录表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | 订单-入账记录id |
| 2 | order\_id | bigint | 20 | 订单id |
| 3 | money\_record\_id | bigint | 20 | 入账记录id |

表rider\_settings：保存的是用户的接单配置信息，如量下开启系统派单

表42 骑手配置表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | settings编号 |
| 2 | rider\_id | bigint | 20 | 骑手id |
| 3 | system\_dispatch | tinyint | 4 | 是否系统自动派单 1开启，0关闭 |
| 4 | max\_taking\_order | int | 10 | 单次最大接单数 |
| 5 | create\_time | datetime | 19 | 创建时间 |
| 6 | update\_time | datetime | 19 | 更新时间 |

表rider\_work\_region：保存的是骑手的工作位置信息。

表43 骑手工作位置表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 说明 |
| 1 | id | bigint | 20 | id |
| 2 | rider\_id | bigint | 20 | 所属rider编号 |
| 3 | longitude | double | 23 | 经度 |
| 4 | latitude | double | 23 | 纬度 |
| 5 | province\_code | varchar | 12 | 省份编码 |
| 6 | province\_name | varchar | 32 | 省份名称 |
| 7 | city\_code | varchar | 12 | 城市编码 |
| 8 | city\_name | varchar | 32 | 城市名称 |
| 9 | district\_code | varchar | 12 | 区县编码 |
| 10 | district\_name | varchar | 32 | 区县名称 |
| 11 | detail\_address | varchar | 200 | 详细地址信息 |
| 12 | create\_time | datetime | 19 | 创建时间 |
| 13 | update\_time | datetime | 19 | 更新时间 |

## 3.6 本章小结

本章主要是对系统的需求分析和可行性分析评估以及可能性研究，着重对系统的前台和后台需求进行了深度探索，并构建了合适的数据表结构来更优秀地完成业务逻辑实现。

# 4 系统业务设计与实现

## 4.1 商家入驻实现

商家入驻界面，将填写店铺必要的信息，店铺的基本信息、地址信息、营业相关信息如营业执照、食品安全许可证。提交注册信息后，申核通过后，会通过绑定的手机号发送账号信息以短信的方式发送给商家登录。

如图12所示：



图12 商家入驻页面

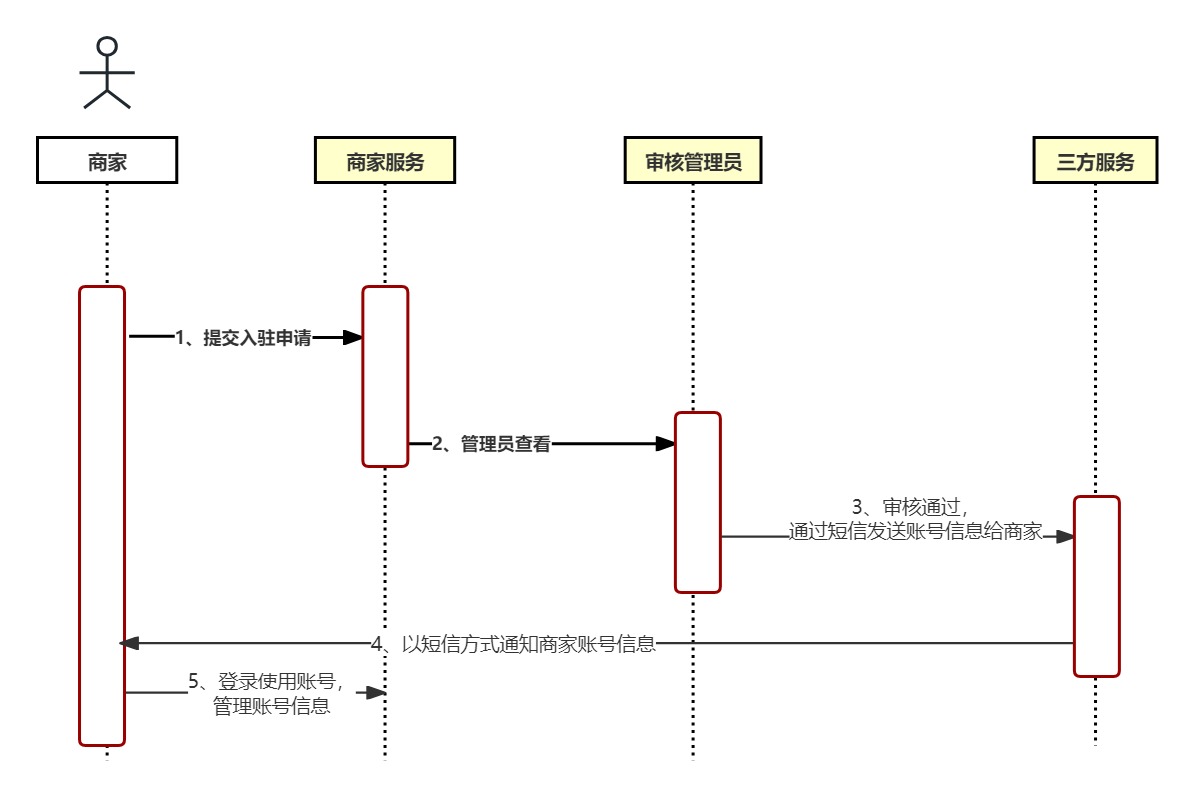


图13 商家入驻时序图

## 4.2 商家基本信息经营配置的修改实现

商家使用账号进行登录后，即可进行店铺信息信息的修改。



图14 商家经营配置信息



图15 商家营业时间信息

## 4.3 商家活动实现

商家活动系统设置了以下几种常见的活动：

1. 满减活动：当适用的商品总和满足指定的金额时，优惠券将为可用状态，在订单确认时可以进行选择使用。
2. 新客立减：新客立减如果是初次或在一段时间内没有在店内购买时，会认为是新客，在商家创建了新客立减活动时，满足条件的用户下单金额够时，会直接立减。
3. 立减配送费：当是配送订单时，商家设置了配送费，在满足立减配送费时，立减配送费有两种，第一种是统一立减，第二种是按距离减，当满足金额时，会进行减免。
4. 神会员膨胀券：商家可以选择参加神会员活动，当商家参与了，就相当于给了商家授权，允许用户使用膨胀券，用户可以对神券进行膨胀，膨胀部分的金额由商家承担。基本的5元或平台券的基本面额由平台承担，如一张5元券，用户膨胀为8元，那平台还是承担5元而膨胀的部分3元由商家承担。神膨胀活动其实是给用户让利的方式。

后端在结算时会使用责任链方式进行计算优惠金额。

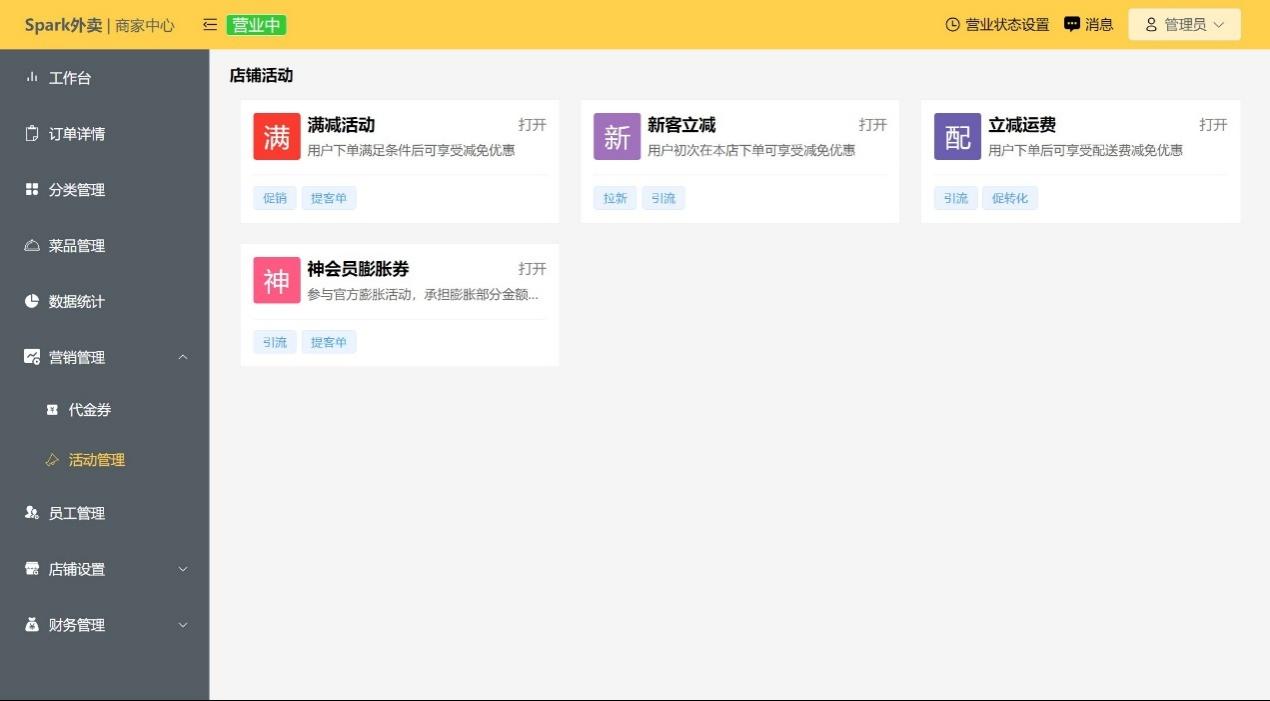


图16 商家活动管理

我们还可以进行优惠券的创建：优惠券分两种满减券（无门槛）、商品券，折扣券只能用于商品进行打折。



图17 商家优惠券更新

## 4.4 用户整体主线的实现

当商家开启营业后，用户就可以在页面发现商家了，请求时后端会sql方式来查询，为了查询快速，前端还需要传入省级行政编码来作为过滤条件查询这会使得查询地更快，速度是省级>市级>区级，项目使用的是省级来查询查询，因为范围越细，在边界处会出现商家“在附近但不可见”问题。 进入商家如果商家营业中，就可以进行点餐，点餐使用到购物车，购物车信息使用的是Redis来存在，不可永久存储，有指定的过期时间，当指定的时间外没有对购物车进行操作来更新过期时间，那过期将会过期。用户下单时还可以使用优惠券，使用的优惠券可以选择平台券、店铺券，进入选择平台券时后端查询的是用户拥有的平台券与店铺券不一样店铺券后台查询的不仅是用户拥有的还会查询店铺的店铺店内可用券，这里说的可用券是看券的可用条件来的，而至于是否能真正可用是前端根据当前可优惠金额来决定。



**图18 用户端主线功能**

## 4.5 商家接单提醒实现

下单后，商家通过自研的SSE就可以收到订单提示，原理是商家客户端与服务端建立连接后，在服务端保存了未断开的“会话”信息，后端是分布式的，同一个模块会有多个实例，当会话在B上，而如果在A会话上处理了接单时，在A上是找不到商家的会话的，会通过分布式的SSE，当当前模块没有会话信息时，会发送事件，通知与其相同的其它模块来查看自家是否有该商家的会话，如果有就会进行新订单消息的发送。



图19 商家来单提醒

## 4.6 骑手接单的实现

商家接单后，对于商家配送的配送订单，系统会创建配送单，并采用特定算法选择附近骑手进行系统派单。然而，系统派单存在一个问题，即每个骑手不能同时接受多个订单，否则可能导致某一骑手无法接受所有订单，同时也会造成附近骑手无法获得订单。为解决此问题，在派发订单给骑手时，会给骑手加一个“屏避”是使用Redis来实现的，这使得在一段时间内不会再向该骑手派发系统订单。

骑手可接受系统派单或获取附近的配送订单进行接单配送。项目中还使用了的算法进行路径规划,骑手可以通过查看路经规划辅助接单与配送[8]。

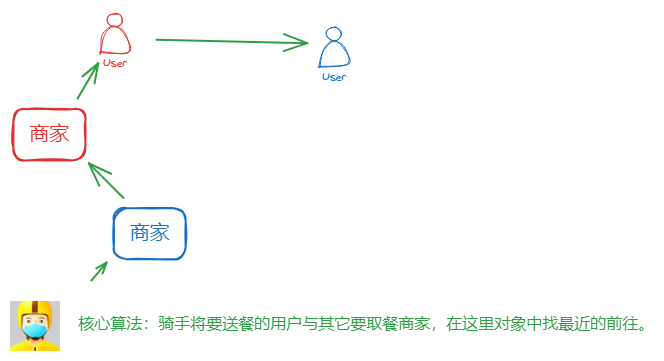


图20 路径规划进核心算法



图21 系统派单与配送单的路线规划

除了可以看单个订单的路径规划，我们还可以查看骑手接的所有配送订单的路径规划：



图22 多单派送路线规划

## 4.7 本章小结

本章主要下对整个系统的主线进行功能演示，从商家入驻用户看到商家进入店内下单，再到商家接单与骑手的接单配送。

通过本项目的学习，我收获了：

1. 通过开发这个项目，使我学会了Java框架的开发、与一些中件间的高级用法，如Redis GEO、Pub/Sub、Hash类型；通过解决一些问题收获了对一些框架的实现原理。
2. 学会了解决一些分布式项目问题的，比如如何在分布式项目中使用SSE、WebSocket、如何使用CompletableFuture提高接口访问效率，及如何在异步请求中保证SEATA的有效、分布式间的模块安全。
3. 熟悉了如购物车、优惠券 等的业务逻辑。

# 5 系统测试与分析

为了检测星火外卖平台的运行的稳定性及使用的正常性，对系统进行了相关的、有针对性的测试，使系统每个模块正确地完成相关的需求，需要对系统进行稳定性有效性测试 [[[2]](#endnote-1)]。

## 5.1 功能测试

5.1.1 测试内容

本文系统测试主要包括系统的安全测试和功能测试。

* 安全测试 :安全测试的主要目的是验证系统的身份验证机制，确保只有经过授权的系统用户能够成功登录系统，而未经授权的非系统用户无法登录系统。
* 功能测试 :功能测试主要是表单操作，如果输入错误的信息，如果数据并不是系统预期的，那应该相应的提示，使用户能够正确的输入信息，确保系统稳定有效地运行。

5.1.2 测试的方法

这次测试采用的是黑盒测试方法，也被称为功能测试。黑盒测试不关注程序的内部逻辑结构和内部特性，而是专注于检验每个功能是否能够按照需求规格书中所描述的方式正常使用。

5.1.3 单元测试用例

安全测试

安全测试用例如表44所示：

表44 商家登录测试描述表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试用例编号 | 测试内容 | 测试步骤 | 预期结果 | 实际结果 | 测试结论 |
| case001 | 输入正确的用户名和密码 | 进入系统用户登录界面，输入正确的用户名和密码，点击登录 | 登录系统成功，提示“登录成功”，进入到首页 | 登录系统成功，提示“登录成功”，进入到首页 | 通过 |
| case002 | 输入正确的用户名，不输入密码 | 进入系统用户登录界面，输入正确的用户名但不输入密码，点击登录 | 登录系统失败，提示“登录失败，密码不能为空，请重新输入正确的密码” | 登录系统失败，提示“登录失败，密码不能为空，请重新输入正确的密码” | 通过 |

续表44

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试用例编号 | 测试内容 | 测试步骤 | 预期结果 | 实际结果 | 测试结论 |
| case003 | 输入正确的用户名，输入非空错误的密码 | 进入系统用户登录界面，输入正确的用户名但输入错误的密码，点击登录 | 登录系统失败，提示“登录失败，密码错误，请重新输入正确的密码” | 登录系统失败，提示“登录失败，密码错误，请重新输入正确的密码” | 通过 |
| case004 | 输入非空错误的用户名，输入非空错误的密码 | 进入系统用户登录界面，输入非空错误的用户名和输入非空错误的密码，点击登录 | 登录系统失败，提示“登录失败，用户名错误和密码错误，请重新输入正确的用户名和密码” | 登录系统失败，提示“登录失败，用户名错误和密码错误，请重新输入正确的用户名和密码” | 通过 |
| case005 | 输入非空错误的用户名，输入正确的密码 | 进入系统用户登录界面，输入非空错误的用户名和输入正确的密码，点击登录 | 登录系统失败，提示“登录失败，用户名错误，请重新输入正确的用户名” | 登录系统失败，提示“登录失败，用户名错误，请重新输入正确的用户名” | 通过 |
| case006 | 不输入用户名，输入正确的密码 | 进入系统用户登录界面，不输入用户名和输入正确的密码，点击登录 | 登录系统失败，提示“登录失败，用户名不能为空，请重新输入正确的用户名” | 登录系统失败，提示“登录失败，用户名不能为空，请重新输入正确的用户名” | 通过 |
| case007 | 不输入用户名，输入非空错误的密码 | 进入系统用户登录界面，不输入用户名和输入非空错误的密码，点击登录 | 登录系统失败，提示“登录失败，用户名不能为空，请重新输入正确的用户名” | 登录系统失败，提示“登录失败，用户名不能为空，请重新输入正确的用户名” | 通过 |

测试用例case005结果如下图23所示：

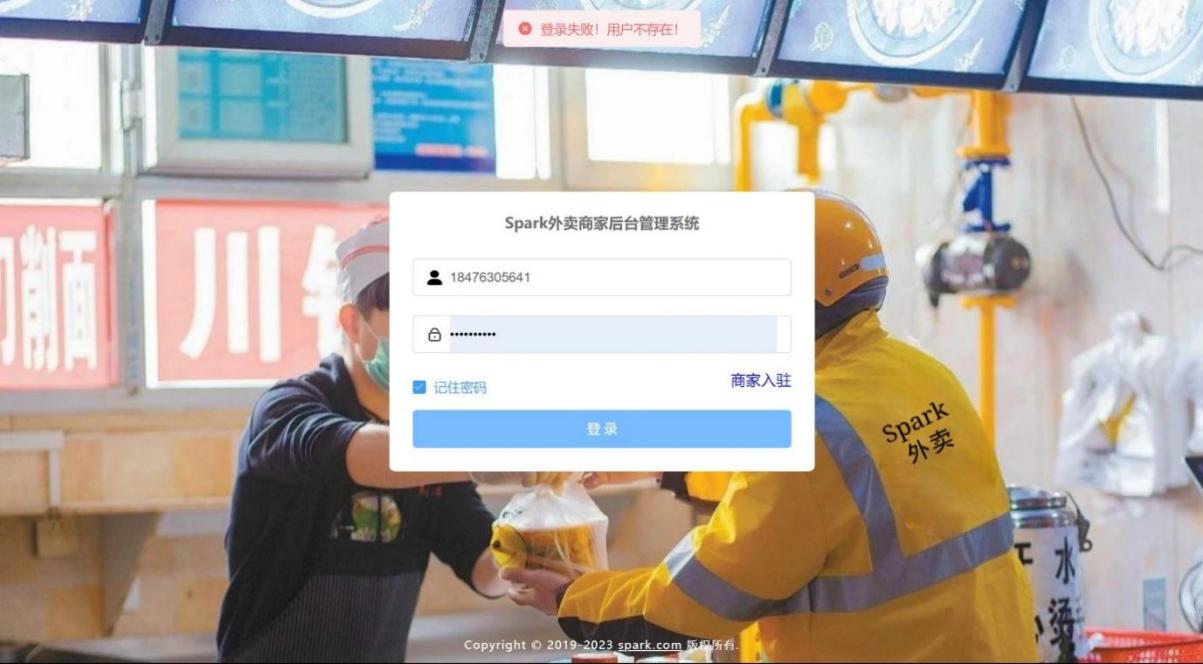


图23 case005测试效果图

5.3.2 功能测试

1. 商家测试用例如表45所示：

表45 商家模块测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试用例编号 | 测试内容 | 测试步骤 | 预期结果 | 实际结果 | 测试结论 |
| case001 | 商家打烊 | 商家点击“营业状态设置”，选择“已打烊” | 用户端无法看到该店铺 | 用户端无法看到该店铺 | 通过 |
| case002 | 商家接单 | 商家点击“订单详情”选择“待接单”进行接单 | 在骑手端可以看到配送单 | 在骑手端可以看到配送单 | 通过 |
| case003 | 商家拒单 | 商家点击“订单详情”选择“待接单”进行拒单 | 微信出现退款提醒，并且订单为已取消状态 | 微信出现退款提醒，并且订单为已取消状态 | 通过 |
| case004 | 搜索订单号 | 商家点击“订单详情”，输入搜索的订单号进行“查询” | 只有一条记录，并且订单号是搜索的” | 只有一条记录，并且订单号是搜索的” | 通过 |
| case005 | 搜索手机号 | 商家点击“订单详情”，输入搜索的手机号进行“查询” | 只有一条记录，并且记录的手机号是搜索的” | 只有一条记录，并且记录的手机号是搜索的” | 通过 |
| case006 | 选择下单的“开始时间”与“结束时间” | 商家点击“订单详情”，选择开始时间与结束时间进行“查询” | 结束的订单时间在选择内的 | 结束的订单时间在选择内的 | 通过 |

1. 测试用例case004结果如下图所示：

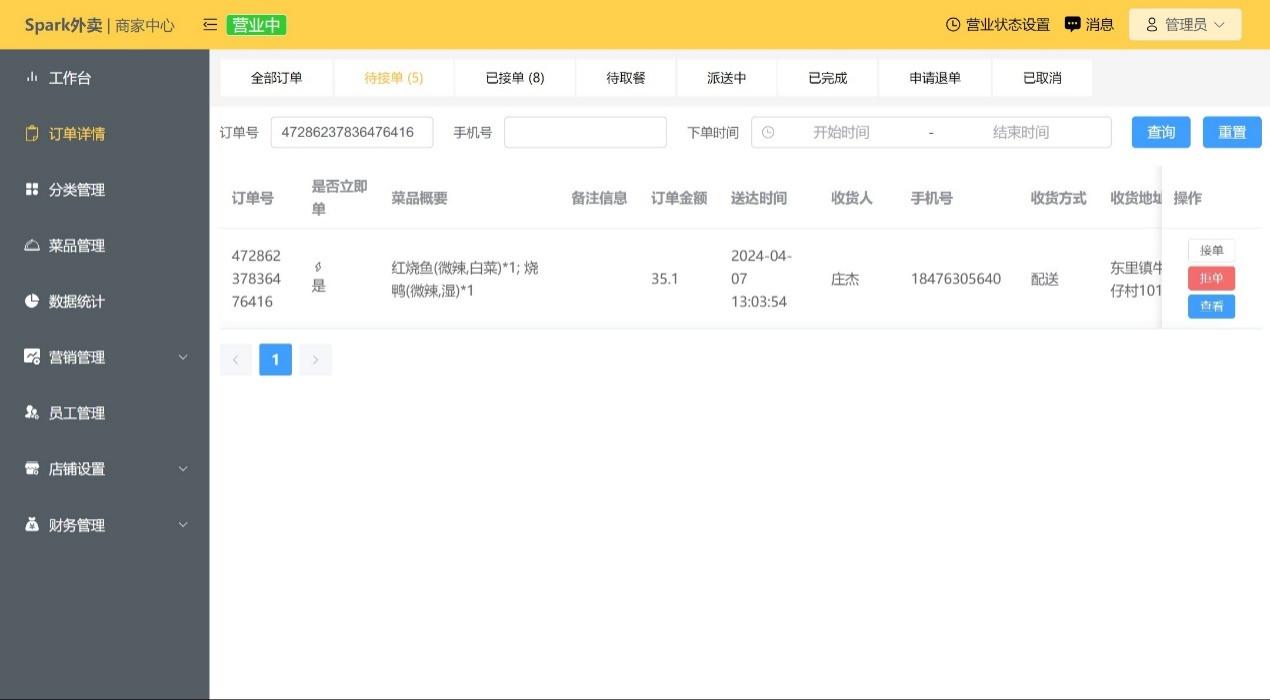
图24 测试用例case005结果图



图25 测试用例case005结果图

**5.4 测试结果分析**

测试团队成员作为项目小组的一部分，从项目需求开始深入理解系统业务，制定了周密的测试计划。在开发和实施过程中，持续跟踪和测试项目的各个阶段版本。通过对测试结果的分析，确认项目的设计和实施符合项目需求规格说明书的要求。

项目的开发和实施已满足当前业务的功能和性能需求，并采取了相应的系统安全和备份措施。然而，随着系统的运行和后期工程的投入，可能会对系统造成一定压力。因此，在后续工程的开发过程中，需要不断考虑对一期工程的性能优化和功能调整，并加大对系统安全和备份方面的资源投入，以确保系统能够持续稳定、可靠地运行，满足7x24小时的要求。

## 5.2 性能测试

我们选用了JMeter这个开源工具，为了对我们网站进行性能分析，以期优化。在这个过程中，我们采取了控制变量的手段，使得在每次压力测试中只有一个变量会发生改变。这样做的目的是为了更精确地了解每个变量对性能测试结果产生的影响。完成测试后，我们会参考测试结果进行优化，我们会采取一步一步的优化策略，对网页进行改进。经过持续的分析和测试，我们将逐步找出并解决问题，以此来提高我们网站主页的性能和用户体验。

**5.2.1 优惠券包性能测试**

我们将使用100个线程，每个线程发起30次请求，对后端API进行压力测试。我们将根据测试结果中的吞吐量、90%响应时间和99%响应时间来评估页面的性能。这种测试方法可以更全面地评估网站的性能，发现潜在问题并及时解决，以提升用户体验和满意度。

下面中在保持接口内部不变下，测试在不同层级的访问下的性能：

表46 API 压测结果表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 压测内容 | 压测线程数 | 线程请求数 | 吞吐量/s | 90%响应时间 | 99%响应时间 |
| 简单服务 | 100 | 30 | 1509 | 55 | 340 |
| Geteway  (加入网关层) | 100 | 30 | 900 | 128 | 239 |
| 内网穿透（网络模拟） | 100 | 30 | 74 | 1065 | 13283 |
| 生产环境  （warn日志级别、关掉链路追踪） | 100 | 30 | 77 | 785 | 11038 |

经过上面测试，得到以下结论：

1. 同时在加入一个层级Geteway或nginx层下也会导致访问性能的下降；
2. 在复杂的网络环境下会导致较大的网络延迟；
3. 在关闭一些日志、链路追踪也会得到一些性能的优化；

**5.2.2 API网关的性能测试**

在系统中，API网关充当了用户访问的主要入口和中转站，因此我们选择对API网关进行性能测试来评估整个系统的性能。我们利用JMeter工具进行压力测试，以收集系统的性能数据。以下是通过压力测试整理出的数据，如表47所示。

表47 API 压力测试表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试用例 | API压力测试 | | | | |
| 测试模块 | 微服务API网关 | | | | |
| 测试目标 | 通过测试系统网关的并发请求数来衡量系统的并发性能 | | | | |
| 步骤 | 并发访问量 | 系统最小响应  时间（ms） | 系统平均响应  时间（ms） | 系统最大响应  时间（ms） | 测试结果 |
| 1 | 5000 | 121 | 243 | 542 | 良好 |
| 2 | 10000 | 245 | 345 | 598 | 良好 |
| 3 | 15000 | 1302 | 2234 | 2801 | 良好 |
| 4 | 20000 | 1982 | 3201 | 3512 | 良好 |
| 5 | 25000 | 2401 | 4001 | 4929 | 良好 |

根据测试结果，我们可以得出结论：尽管系统面临不断增加的并发访问量，但系统依然能够保持良好的响应时间。这一发现非常重要，因为它表明系统在高并发情况下仍然具有稳定性和可靠性。实际上，这个结果证明了我们在系统架构和配置方面所做的优化措施已经取得了成效，从而提高了系统的可扩展性和可维护性。因此，在未来进行系统升级和优化时，我们可以有针对性地改进，进一步提升系统的性能和稳定性。

## 5.3 本章小结

经过多次测试，我们发现了一些之前未察觉到的错误或未完成的功能。经过修正后重新进行测试，最终测试结果与项目开发初期设想的预期结果一致。

# 6 总结与展望

## 6.1 总结

分布式的星火外卖平台是一个复杂的系统，需要综合考虑多个方面的设计和实现。本文将从以下几个方面进行论述：架构设计、数据管理、性能优化、安全性设计以及用户体验。

首先，架构设计是分布式的星火外卖平台设计的核心。在架构设计中，需要考虑系统的可扩展性、高可用性以及负载均衡。为了实现这些目标，可以采用微服务架构，将每个功能模块拆分成不同的服务，并通过 API 进行通信。容器技术如 Docker 和 Kubernetes 可用于部署这些微服务。

其次，数据管理也是分布式的星火外卖平台设计中的关键部分。为了确保数据的一致性和可靠性，可以使用分布式数据库，例如 MySQL Cluster、Cassandra 和 MongoDB等。此外，缓存技术如 Redis 和 Memcached 可以提高读取速度。

第三，性能优化对于确保系统稳定运行至关重要。为了提高系统的响应速度和吞吐量，可以采用 CDN 技术加速静态资源的传输，例如使用阿里云 CDN。负载均衡器可以分配请求到不同的服务器上，以避免单点故障。

第四，安全性设计是分布式的星火外卖平台不可或缺的功能。采用 SSL 证书加密数据传输和身份验证等措施可保护用户数据的安全性。实时监测系统中的异常操作和攻击行为也是必要的。

最后，用户体验至关重要。分布式的星火外卖平台需要提供友好的界面和流畅的操作体验，吸引更多用户。可使用 React、Vue.js 等前端框架构建用户界面，并采用响应式设计以适配不同终端设备。

尽管存在诸多挑战和技术难题，但只要采用正确的设计和实现策略，就能构建出高效、稳定和安全的分布式的星火外卖平台，为用户提供更好的购物体验。

## 6.2 展望

随着分布式的星火外卖平台市场的不断扩张和用户数量的增加，传统的单体化系统已经不能满足需求。为了提高系统的可靠性、可扩展性以及响应速度，分布式的星火外卖平台已经成为了当前的趋势。本文将展望分布式的星火外卖平台设计与实现的未来发展趋势。

首先，分布式的星火外卖平台将更加注重数据一致性和高可用性。在传统的单体化外卖系统中，由于存在单点故障，当系统崩溃时，整个系统将无法正常运行。而分布式的星火外卖平台通过将数据划分为多个分片存储在不同的节点上，可以避免单点故障的问题，并且提高了系统的可靠性和可用性。但是，在分布式系统中，由于网络延迟等原因，数据的一致性可能会出现问题。因此，未来的研究将集中于解决数据一致性和高可用性的问题，例如使用副本机制和多版本并发控制技术等来确保数据的一致性和可用性。

其次，分布式的星火外卖平台将更加注重负载均衡和性能优化。在大规模的外卖平台系统中，服务器集群的规模往往非常庞大，并且用户的访问量也非常高。因此，实现负载均衡和性能优化将成为分布式的星火外卖平台设计的重要问题之一。未来的研究将集中于设计更加高效的负载均衡算法和性能优化技术，例如使用缓存、异步处理和分布式任务调度等技术来提高系统的性能。

最后，分布式的星火外卖平台将更加注重安全性和隐私保护。在交易中，用户的个人信息和交易信息往往涉及到隐私和安全问题。因此，未来的研究将集中于设计更加安全和可靠的分布式的星火外卖平台，例如使用加密技术、访问控制机制和安全审计机制等技术来确保系统的安全性和隐私保护。

# 参考文献

[1] 王磊, 孟利民. 基于高并发的分布式系统幂等设计[J]. 计算机测量与控制, 2022, 30(3): 234～238.

[2] 孙波. 浅谈微服务架构、Docker 和 Kubernetes[J]. 现代电视技术, 2022(2): 100～103.

[3] 唐权, 周蓉, 张勇. RabbitMQ 消息中间件在 Spring Boot 教学中的应用[J]. 现代信息科技, 2020, 4(18): 125～127.

[4] 孙景玉, 孙浩, 高婷玉, 秦雯波, 陈虹云. 基于 Redis 的高并发秒杀系统的研究与实现[J]. 信息记录材料, 2022, 23(12): 45～47.

[5] 周喜平, 杜航勤, 梁俊礼. 经纬度计算与 Redis 的 GEO 算法对比[J]. 中国科技信息, 2024, (01): 86～88.

[6] 胡弦. Spring Cloud Alibaba 微服务架构实战派[M]. 电子工业出版社，2022,(1): 77～80.

[7] 方志宁, 谢华, 张金营等. 基于微服务架构的统一应用开发平台[J]. 仪器仪表用户, 2023, 30(03): 93～97.

[8] 唐传茵, 章明理, 李静红, 等. 基于改进蚁群算法的外卖配送路径规划研究[J]. 南京信息工程大学学报, 2024, 16(2): 145～154.

[9] 郭靖文. 基于遗传算法的外卖路径优化研究[D]. 辽宁工程技术大学, 2023,32(1): 23～40

[10] 秦少其. 基于配送效率的平台专送激励性定价策略[J]. 中国储运, 2024, (04): 53.

[11] 胡宇航. 外卖骑手劳动关系认定问题研究[D]. 辽宁科技大学, 2023,(5): 21.

[12] 季甜甜, 刘冬冬. 基于 Vue 前端性能的研究与分析[J]. 阜阳师范大学学报(自然科学版), 2024, 41(1): 15～22.

[13] Mohan A, Jayaraman S, Jayaraman B. A declarative approach to detecting design patterns from Java execution traces and source code[J]. Information and Software Technology, 2024, 41(01): 15～22.

[14] Fahmani L, Benhadou S. Optimizing 2D path planning for unmanned aerial vehicle inspection of electric transmission lines[J]. Scientific African, 2024, (1): 36～42.

[15] Zhao J, Mao H, Mao P, et al. Learning path planning methods based on learning path variability and ant colony optimization[J]. Systems and Soft Computing, 2024, (1): 24～31.

[16] Li X, Yu S, Gao Z X, et al. Path planning and obstacle avoidance control of UUV based on an enhanced A\* algorithm and MPC in dynamic environment[J]. Ocean Engineering, 2024, (1): 18～25.

[17] Zhang T, Fu T, Ni T, et al. Data-driven excavation trajectory planning for unmanned mining excavator[J]. Automation in Construction, 2024, (1): 30～37.

[18] Luo J, Zhuang J, Jin M, et al. An energy-efficient path planning method for unmanned surface vehicle in a time-variant maritime environment[J]. Ocean Engineering, 2024, (1): 22～29.

[19] Yang Y. Design and Implementation of Online Food Ordering System Based on Springcloud[J]. Information Systems and Economics, 2022, 3(4): 21～42.

[20] Baouya A, Hamid B, Gürgen L, et al. Rigorous Security Analysis of RabbitMQ Broker with Concurrent Stochastic Games[J]. Internet of Things, 2024, (1): 16-23.

# 致 谢

随着本论文即将完成，我内心的情感难以言喻。倘若没有我的指导老师刘兴建老师的强大支持和悉心开导，这篇论文是无法顺利完成的。从论文主题的挑选，到开题报告的编写，到系统设计的构思，到系统开发的过程，再到论文反复修改直到最后的定稿，刘老师都严谨审视每一个环节。若非刘兴建老师以耐心教诲，和多次审阅论文提出专业且深入的修改建议，论文的顺利完成也是难以为继的。我在这里向我尊敬的指导老师刘老师表达我最真挚的谢意。

1. [↑](#endnote-ref-0)
2. [↑](#endnote-ref-1)