Web论文

基于SpringBoot框架的晚上点餐系统

**专业：软件11809**

**学号：1804230920**

**姓名：郭 悦**

武汉纺织大学

计算机与人工智能学院

**2021** 年 7月

摘 要

在中国餐饮业的发展现状看，中国餐饮业当今的特点之一就是在线点餐服务发展迅速，随着互联网普及率的提高，以及互联网应用的深入，网上点餐已经成为我们生活中必不可少的一部分了。尤其是当代的大学生和上班族。种类繁多，价格便宜，送饭到手等是他们选择网上点餐的重要原因。

网上点餐系统是一个 O2O 商业模式的点餐网站系统，它通过在网上发布商品的信息让客户根据自己的需求在众多同类的产品中搜索筛选出自己要下单的商品。当用将自己心仪的商品加入购物车，付钱后，商家接单之后开始为客户服务，最后骑手把商品送到客户手里。

网上点餐系统是以java为后端开发语言，使用的SPRINGBOOT核心框架，以MYSQL为关系型数据库平台，以 vue.js为前端辅助框架，设计了网上点餐系统的 Web 端和管理后台两个平台。该系统客户点单老板做餐，管理员管理店铺等功能。

本文分析了客户在网上点餐过程中的筛选要求，购买商品，活态管理的则是所有的老板的店铺推成出新。

重点对系统的客户搜索商品，显示功能，订单信息进行了分析，并给出了系统主要 功能的实现方法，展示了最终的实现效果和测试结果。 目前该客户点餐的功能已基本完成，客户点餐的逻辑流程初步得到实现。

关键词：O2O，网上点餐，Java，pringboot框架，MySQL，vue.js

ABSTRACT

In the development situation of China's catering industry, one of the characteristics of China's catering industry today is the rapid development of online ordering service. With the improvement of Internet penetration rate and the depth of Internet application, online ordering has become an indispensable part of our life. Especially in the contemporary college students and office workers. Wide variety, cheap price, sending food to hand are the important reasons why they choose to order online.

The Online Ordering System is a O2O business model ordering website system that allows customers to search out many similar products to order according to their own needs. When you add your favorite goods to the cart, and pay, the merchant begins to serve the customer after receiving the order, and finally the rider sends the goods to the customer.

The online ordering system is the development language with java as the back-end, using the SPRINGBOOT core framework, MYSQL as the relational database platform and vue.js as the front-end auxiliary framework, and designing the Web-end and management background of the online ordering system. The system customers order owners to do meals, administrator management of the store and other functions.

This paper analyzes the screening requirements of customers in the process of online ordering. In the purchase of goods, all the stores of all owners are brought new.

Focus on the system customer search goods, display function, order information analysis, and gave the implementation method of the main functions of the system, showing the final realization effect and test results. At present, the customer ordering function has been basically completed, the logical process of customer ordering has been initially realized.

**Keywords**：O2O, Order your food online, Java, Springboot framework, MySQL ，vue.js

目录

[1 引言 1](#_Toc4406)

[1.1 网上点餐系统开发背景 1](#_Toc11489)

[1.3 O2O 商业模式 1](#_Toc1643)

[1.4 系统开发的目的与意义 3](#_Toc24606)

[2系统技术概括 4](#_Toc7232)

[2.1 系统前端技术概括 4](#_Toc22423)

[2.2系统后端技术概括 5](#_Toc13327)

[3 系统分析与设计 8](#_Toc6689)

[3.1 网上订餐系统概述 8](#_Toc16559)

[3.2 网上点餐系统需求分析 9](#_Toc8487)

[3.3 系统设计 10](#_Toc10251)

[3.3 点餐功能设计 10](#_Toc27755)

[4 系统实现 16](#_Toc24149)

[4.1 数据库实现 16](#_Toc11496)

[4.2 客户功能实现 18](#_Toc28646)

[4.2.2.商品搜索页面 21](#_Toc1977)

[4.2.3.支付管理 23](#_Toc5898)

[4.3商家功能实现 25](#_Toc14632)

[4.3.1商家注册 25](#_Toc9548)

[4.3.2添加商品管理 26](#_Toc24920)

[4.3.3.配送管理 27](#_Toc1020)

[4.4 管理员功能实现 28](#_Toc20265)

[5 总结与展望 30](#_Toc28164)

[参考文献 30](#_Toc834)

# 1 引言

## 1.1 网上点餐系统开发背景

随着现代生活节奏的加快，都市人越来越倾向简单生活。从传统的“以食为天”到今天的“没空做饭”，现在的上班族正经历着不小的转变。人们生活水平也逐渐提高，越来越多的人形成了一种消费的意识，愿意将钱花出去。尤其是对于一二线城市的人来说，平时上班就已经很忙，下班之后很少有时间去市集买菜做饭。

由于目前80、90慢慢成为主力消费群，网络消费更是成为必然，不愿意下厨、不会下厨更是成为现实，气候环境的恶劣影响，更是促进了外卖的发展。

气候反复无常，差距较大，在夏天，39度的高温让人无法走出空调房，人挤人的排队等餐，那是何等的煎熬，在冬天，北风呼啸，大家更是愿意宅在家里，等待美食上门。

在这种情况下，人们急需一个安全可靠的平台能够根据所在地不同，筛选出与之相对应的餐饮，并让客户有更多的选择。与此同时外卖将食品产业的运输环节独立出来，细化产业分工，使更多的饭店不再需要使店员兼职或单独培训外卖送餐员，得以集中在于食物的生产。同时餐饮行业通过和外卖公司的合作，扩展了线上客户量。所以外卖是餐饮行业进步、社会分工细化的利益最大选择。

论文所做的工作具有一定的社会意义，能够解决实际问题。

1.2 网上点餐行业发展状况

网上点餐已经成为成为一种人们生活的一部分。自2019年第三季度已经突破4亿用户，在使用外卖服务的用户性别方面，外卖并不是男人的专属，有49.7%的女性占据了中国外卖业务将近一半的用户量，男性用户在2019年第三季度的占比达到50.3%。根据不同用户的年龄段来看，25-29岁的年轻用户依然是外卖的主力，但30岁以上的用户占比也并不低，同时工作压力较大的二线城市是外卖服务的主要趋于，占比达到外卖用户总量的47.7%，交易额占比达到52.1%，同比增长7.6%。

2019年第三季度的外卖行业交易总额已经达到1790亿元，相较上一季度有360亿元的增长，一线城市用户24.3%的交易额占比虽然大约只有二线城市的一半，但同比增长幅度5.2%，已经超越了三线及三线以下的城市交易额占比。虽然午餐时段是外卖业务活跃度的峰值，但在早餐和晚餐的时间段中，用户使用外卖服务的活跃度也分别同比增加14.5%和2.9%，同时43.1%的用户已经加入任一外卖平台的会员服务，虽然这一比例并未达到整体用户的50%，但同比2018年第三季度已经有93.6%的提升，这意味着用户的忠诚度正在提升，跨平台订外卖的比重正在下降，小额红包的优惠力度已经很难从竞品平台中挖到新用户。

总体来说，由于生活节奏的加快，网上餐饮的种类丰富多样，网上点餐的已经发展成我们生活的一部分了。

## 1.3 O2O 商业模式

O2O 是 Online To Offline（线上到线下）的缩写，是指将线下的商务机会与互联网结合，让互联网成为线下交易的前台。O2O 的概念非常广泛，只要产业链中既可涉及到线上，又可涉及到线下，就可通称为 O2O。自O2O商业模式出现发展以来，其一共经历了三个阶段。

早期第一阶段，O2O 线上线下初步对接，主要是利用线上推广的便捷性等把相关的用户集中起来，然后把线上的流量倒到线下，主要领域集中在以美团为代表的线上团购和促销等领域。在这个过程中，存在着主要是单向性、粘性较低等特点。平台和用户的互动较少，基本上以交易的完成为终结点。用户更多是受价格等因素驱动，购买和消费频率等也相对较低。

发展到第二阶段后，O2O 基本上已经具备了大家所理解的要素。这个阶段最主要的特色就是升级为了服务性电商模式：包括商品、下单、支付等流程， 把之前简单的电商模块，转移到更加高频和生活化场景中来。在新模式的推动和资本的催化下，出现了 O2O 的狂欢热潮，上门按摩、上门送餐、上门生鲜、上门化妆、[滴滴打车](https://baike.baidu.com/item/%E6%BB%B4%E6%BB%B4%E6%89%93%E8%BD%A6)等各种 O2O 模式开始层出不穷。而在这个阶段，移动终端、[微信支付](https://baike.baidu.com/item/%E5%BE%AE%E4%BF%A1%E6%94%AF%E4%BB%98)、数据算法等环节的成熟，加上资本的催化，用户数量激增，使用频率和依赖性开始上升，O2O 开始和用户的日常生活开始融合，成为生活中密不可分的一部分。

到了第三阶段，开始了明显的分化，一个是真正的垂直细分领域的一些公司开始凸现出来。比如专注于快递物流的速递易，专注于高端餐厅排位的[美味不用等](https://baike.baidu.com/item/%E7%BE%8E%E5%91%B3%E4%B8%8D%E7%94%A8%E7%AD%89)，专注于白领快速取餐的速位。另外一个就是垂直细分领域的平台化模式发展。由原来的细分领域的解决某个痛点的模式开始横向扩张，覆盖到整个行业。

O2O 其有四种线下线上组合模式。

第一类为“线上社区+线下消费/社区”。这是一种必须到线下进行消费的 O2O 模式，线上主要是在线交流互动，并开展一定的优惠或促销活动。这种模式适应于必须亲自到现场消费的宾馆、餐饮及其他领域。

第二类为“线上消费/社区+线下社区”。这是一种只在线上销售的 O2O 模式， 线上还有在线交流或开展促销作用，线下主要是面对面的交流互动或现场展示。这种模式适应于无线下门店仅有线上网店的纯线上电商领域。

第三类为“线上消费/社区+线下消费/社区”。这是一种线上与线下同时进行销售的 O2O 模式，线上、线下都有交流互动的需求，线上与线下还可以分别开展优惠或促销活动。这种模式适应于线上有网店且线下有门店的领域。

第四类为“线上社区+线下社区”。这是一种无销售业务的 O2O 模式，适用于同时需要线下及线下交流互动的社交平台[3]。

## 1.4 系统开发的目的与意义

作为本世纪最具颠覆性的消费革命。外卖时代的到来，实现了我们足不出户、尽享四方美食的梦想。方寸屏幕，指尖飞舞，只需须臾，不论是珍馐美食，还是家居用品，都可闪现手中。外卖的普及，让我们的生活幸福指数直线上升。

作为客户下楼，不用排队，手机点菜快，方便快捷，并且用一只手控制周围的餐馆，选择更广泛的美食，而且通过参与会员积分和全额折扣来订餐更具成本效益。更多的销售、评估、位置距离等筛选，更放心。

作为老板移动订购已经成为一种趋势，建立自己的订购和交付app平台，以扩大在线流量；而且不仅做3公里左右的业务，手机订购、外卖送货，把业务地图扩大到10公里左右；通过app，可以轻松举办各种营销活动，如优惠待遇、全额优惠、折扣、限时优惠、团体竞争等。建立会员积分制度。顾客消费越多，积分越高，享受的优惠就越多，从而大大增加了回头客的比例。新产品信息、特殊事件等。可以发送到用户的手机上进行准确的营销，而过去只能发送传单。

# 2系统技术概括

## 2.1 系统前端技术概括

2.11 VUE.js框架

Vue.js是一套构建用户界面的渐进式框架。与其他重量级框架不同的是，Vue采用自底向上增量开发的设计。Vue 的核心库只关注视图层，并且非常容易学习，非常容易与其它库或已有项目整合。另一方面，Vue完全有能力驱动采用单文件组件和Vue生态系统支持的库开发的复杂单页应用。  
 Vue.js的目标是通过尽可能简单的API实现响应的数据绑定和组合的视图组件。  
Vue.js自身不是一个全能框架——它只聚焦于视图层。因此它非常容易学习，非常容易与其它库或已有项目整合。另一方面，在与相关工具和支持库一起使用时，Vue.js也能驱动复杂的单页应用。Vue.js是一套构建用户界面的渐进式框架。与其他重量级框架不同的是，Vue采用自底向上增量开发的设计。Vue 的核心库只关注视图层，并且非常容易学习，非常容易与其它库或已有项目整合。另一方面，Vue完全有能力驱动采用单文件组件和Vue生态系统支持的库开发的复杂单页应用。  
 Vue.js的目标是通过尽可能简单的API实现响应的数据绑定和组合的视图组件。

Vue.js自身不是一个全能框架——它只聚焦于视图层。因此它非常容易学习，非常容易与其它库或已有项目整合。另一方面，在与相关工具和支持库一起使用时Vue.js也能驱动复杂的单页应用。

（1）创建 Ajax 无刷新网页

Ajax 是异步的 JavaScript 和 XML 的简称，可以协助开发出非常灵敏无刷新的网页。

Ajax 是 Jquery 框架的一部分功能简称，它在网页制作中应用广泛，其具体功能在上面叙述 Jquery 的特点中也有提到，即可以使我们发送请求时进行异步操作，在不刷新页面的情况下对后台的处理产生响应。不借助 Ajax 的话，我们在页面逻辑操作时，必须要进行从当前页面到控制视图层再返回当前页面的多次跳转，很影响用户的操作体验。而借助 Ajax，后端在处理请求时用户依然可以在前端页面继续进行浏览，直到响应返回前端用户再进行操作，整个过程更加连贯，用户的操作体验自然也会更好。

## 2.2系统后端技术概括

2.2.1MySQL 数据库

数据库分为关系型数据库和非关系型数据库。

关系型数据库指的是采用了关系模型来组织数据的数据库，拥有关系，元组， 属性，域，关键字，关系模式等概念。关系型数据库比较直观，更容易被理解， 使用起来较为方便，易于维护。但关系型数据库性能欠佳，在处理大量关系复杂的数据时，效率很低。我们常用的关系型数据库有 [Oracle](https://www.oracle.com/database/index.html)，[Microsoft SQL Server](https://www.microsoft.com/en-us/sql-server/)， MySQL，[PostgreSQL](https://www.postgresql.org/)，[DB2](https://www.ibm.com/analytics/us/en/db2/)，[Microsoft Access](https://products.office.com/zh-cn/access)，[SQLite](https://www.sqlite.org/) 等。

非关系数据库指的是指与关系型数据库相对的非关系型的，分布式的数据库。它以键值对存储，且结构不固定，每一个元组可以有不一样的字段，每个元组可 以根据需要增加一些自己的键值对，不局限于固定的结构。非关系型的数据库结 构简单，处理数据时可以节省一点的时间和开销。但它仅限处理一些简单的没有 复杂关联关系的数据库，不适合持久存储海量数据。由于非关系型数据库都是针 对某些特定的应用需求出现的，因此，不同非关系数据库的特长也有所不同，主 要分为以下几类：面向高性能并发读写的 key-value 数据库，主流代表为 [Redis](https://redis.io/)， [Amazon DynamoDB](https://aws.amazon.com/cn/dynamodb/) ； 面向海量数据访问的面向文档数据库， 主流代表为[MongoDB](https://www.mongodb.com/)；面向搜索数据内容的搜索引擎，主流代表为 [Splunk](https://www.splunk.com/zh-hans_cn)，[Solr](https://lucene.apache.org/solr/)；面向可扩展性的分布式数据库，主流代表为 [Cassandra](https://cassandra.apache.org/)，[HBase](https://hbase.apache.org/)。

MySQL 是典型的关系型数据库，它由瑞典 MySQL AB 公司开发，在被收

购之后属于 [Oracle](https://baike.baidu.com/item/Oracle) 旗下产品。与其他关系型数据库相同，MySQL 将数据保存在不同的表中，确保效率并提高了灵活性。MySQL 使用 C 和 [C++](https://baike.baidu.com/item/C%2B%2B)编写，并使用了多种[编译器](https://baike.baidu.com/item/%E7%BC%96%E8%AF%91%E5%99%A8)进行测试，保证了[源代码](https://baike.baidu.com/item/%E6%BA%90%E4%BB%A3%E7%A0%81)的可移植性，使得它能够支持大多数的操作系统。

MySQL [所使用的操作语言 是用于访问数据库](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93/103728)的最常用标准化语言 SQL 语言。由于其体积小、速度快、总体拥有成本低，尤其是[开放源码](https://baike.baidu.com/item/%E5%BC%80%E6%94%BE%E6%BA%90%E7%A0%81/7176422)这一特点，一般中小型网站的开发都选择 MySQL 作为网站数据库。虽然 MySQL 与其他的大型数据库比有所不足，但对于个人开发者，MySQL 数据库包含了个人开发所需的全部功能，并且它是免费开放的，降低了学习成本。也正是出于上述原因，我在网上点餐系统中选择了 MySQL 数据库为项目的数据库。

2.2.2Tomcat 服务器

搭建 Web 项目，服务器是必不可少的。而 Tomcat 技术先进、性能稳定， 而且免费，是个人 JavaWeb 开发的首选。

Tomcat 服务器是一个免费的开放源代码的 Web 应用服务器，由 [Apache](https://baike.baidu.com/item/Apache/6265)、Sun 和其他一些公司及个人共同开发而成，属于轻量级应用[服务器](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8)。它更多在中小型系统和并发访问用户不是很多的场合下被普遍使用，是开发和调试 JSP 程序的首选。在 Tomcat 配置完成后，开发者就可以在服务器上运行 JSP 页面和 Servlet。它具有处理 HTML 页面请求的功能，同时还是一个 Servlet 和 JSP 容器。

我在项目中搭建的是 SPRINGBOOT内置的Tomcat 最新的版本，个人使用时经过一些简单的配置配合 Idea、Eclipse 等 Java 开发软件就能较为方便地使用。通常控制器负责从视图读取数据，控制用户输入，并向模型发送数据。

我在项目中使用的springboot内置了spring+springmvc，整合了mybatis-plus。springboot框架经常作为数据源较简单的 Web 项目的框架，在下面我也将对 springboot 框架进行一个大概的描述。

2.2.3 Spring 框架

Spring 框架用于简化软件开发而创建，是 JavaWeb 开发的常用框架。它的主要功能是使用 JavaBean 代替 EJB（Enterprise Java Beans 的技术简称，是之前的应用开发组件）。

在 Spring 框架之前，开发过程中编写的代码经常出现类似或者重复的情况，这导致开发过程复杂，开发效率和性能都不太理想。Spring 的出现就是为了解决早期这种应用开发的情况，它致力于提供一个以统一的、高效的方式构造整个应用，并且可以将单层框架以最佳的组合揉和在一起建立一个连贯的体系。

Spring 作为一个容器框架，其最大的两个特性是控制反转（IOC）和面向切面（AOP）。

控制反转：控制反转是 Spring 框架作为容器框架的主要特性。原本我们在Java 开发中，我们要使用对象时都需要自身创建对象再进行操作。而使用Spring 框架后，所需对象依赖的其它对象会通过被动的方式传递进来，而不是这个对象自己创建或者查找依赖对象，方便开发者使用对象的同时节省空间， 促进了松耦合。

[面向切面：面向切面是函数式编程](https://baike.baidu.com/item/%E5%87%BD%E6%95%B0%E5%BC%8F%E7%BC%96%E7%A8%8B/4035031)的一种衍生范型。利用面向切面可以对业务逻辑的各个部分进行隔离，从而使得业务逻辑各部分之间的[耦合度](https://baike.baidu.com/item/%E8%80%A6%E5%90%88%E5%BA%A6/2603938)降低， 提高程序的可重用性，同时提高了开发的效率。

在项目中使用 Spring 框架需要在项目中配置部分的配置文件，一般在项目中会配置文件搭配注解使用，简化配置并方便进行管理。

2.2.4 SpringMVC 框架

SpringMVC 是 Spring 框架的后续产品，它在 SSM 框架中的主要作用作为

Controller（控制器）部分处理 View（视图层）。SpringMVC 分离了控制器、模型对象、过滤器以及处理程序对象的角色，这种分离让它们更容易进行定制。

在实际的使用中，它的作用类似于 Servlet。配置文件配置完成后，创建对应的文件，并一一添加注解，通过这些注解的引用，前端的请求就能转到对应的方法进行处理。但不同的是，SpringMVC 使用注解，一个文件可以处理多个请求，在处理整个项目的请求时，更加轻松，管理起来也更加实用。

2.2.5 MyBatis -plus框架

MyBatis 框架是基于 Java 的持久层框架，用于定制化 sql、存储过程以及高级映射。

原本在 JavaWeb 开发中需要借助 JDBC 工具实现对数据库的 sql 语句操作， 但这样设置 sql 的参数并不方便。MyBatis 消除了几乎所有的 [JDBC](https://baike.baidu.com/item/JDBC) 代码和参数的手工设置以及[结果集](https://baike.baidu.com/item/%E7%BB%93%E6%9E%9C%E9%9B%86)的检索， 使用简单的 XML 或注解用于配置和原始映射， 将接口和 Java 的模型映射成数据库中的记录。

相比传统的持久层操作，MyBatis 框架下的持久层操作更为简单易学。只需要几个 jar 包和几个配置文件将可以成功搭建出所需的环境。搭建完环境后，开发者就可以开始 sql 映射文件的编写。由于 MyBatis 把 sql 语句统一在这些 sql 映射文件中，实际操作中的编写和管理起来也更为方便。

MyBatis 框架的框架架构流程如下：

(1)加载配置：在开启项目后，服务器会读取我们实现配置的文件以便在后面进行启用。

(2)sql 解析：当前台发送请求调用了事前设定的持久层对象的方法时，会根据方法名称在 sql 映射文件中找出 ID 与之相同的部分，之后传入参数，进行解析，得到最终要执行的 sql 语句和参数。

(3)sql 执行：将之前 sql 解析得到的参数整合到 sql 语句中，然后执行sql语句

(4)结果映射：将操作数据库的结果按照映射的配置进行转换，可以转换成 HashMap、JavaBean 或者基本数据类型，并将最终结果返回。

2.2.6 springboot框架

SpringBoot基于Spring4.0设计，不仅继承了Spring框架原有的优秀特性，而且还通过简化配置来进一步简化了Spring应用的整个搭建和开发过程。另外SpringBoot通过集成大量的框架使得依赖包的版本冲突，以及引用的不稳定性等问题得到了很好的解决。  
SpringBoot所具备的特征有：  
（1）可以创建独立的Spring应用程序，并且基于其Maven或Gradle插件，可以创建可执行的JARs和WARs；  
（2）内嵌Tomcat或Jetty等Servlet容器；  
（3）提供自动配置的“starter”项目对象模型（POMS）以简化Maven配置；  
（4）尽可能自动配置Spring容器；  
（5）提供准备好的特性，如指标、健康检查和外部化配置；  
（6）绝对没有代码生成，不需要XML配置。  
SpringBoot基于Spring4.0设计，不仅继承了Spring框架原有的优秀特性，而且还通过简化配置来进一步简化了Spring应用的整个搭建和开发过程。另外SpringBoot通过集成大量的框架使得依赖包的版本冲突，以及引用的不稳定性等问题得到了很好的解决。  
SpringBoot所具备的特征有：  
（1）可以创建独立的Spring应用程序，并且基于其Maven或Gradle插件，可以创建可执行的JARs和WARs；  
（2）内嵌Tomcat或Jetty等Servlet容器；  
（3）提供自动配置的“starter”项目对象模型（POMS）以简化Maven配置；  
（4）尽可能自动配置Spring容器；  
（5）提供准备好的特性，如指标、健康检查和外部化配置；  
（6）绝对没有代码生成，不需要XML配置。

# 3 系统分析与设计

## 3.1 网上订餐系统概述

网上订餐系统是基于 springboot框架，以 Java 为主要开发语言，以 MySQL

为数据库集合vue.js等前端框架而成的网站应用。该系统希望为时下年轻人提供了一个便利点餐的平台，很大程度上解决了现代年轻人的吃饭问题。在这里，客户可以直接搜索自己想要的食品，可以根据评分，近距离，价格等条件进行筛选，同时如果觉得这家商店的味道正宗也可以进行收藏，下次直接点击我的收藏直接购买。而商家也可以把食品发放到平台，供用户挑选。也可以给已经收藏本店的用户进行产品的推送和老客户的优惠活动。

## 3.2 网上点餐系统需求分析

3.2.1用户需求分析

满足用户的需求是网上订餐系统首要解决的问题。用户可能的需求如下：

* + - 1. 登录注册：基本上所有的网站在使用时都有登录注册功能，网上点餐系统也不例外。用户在查看餐品信息时可以不进行登录，以游客模式进行使 用，只有当需要点外卖的相关功能时在进行登录。
      2. 查看餐品：用户使用网上点餐系统首先需要的功能即是查看周边的美食，用户只有在浏览周边美食并进行对比后才能判断选择出最适合自己的餐品进行点餐。

餐品的筛选和排序：用户在浏览餐品时，餐品的数量过多会使得用户查看起来很不方便，所以该系统需要在查看餐品页面中添加筛选功能和排序功能。客户可以以价格，评分，优惠力度，是否参加满减等等属性为自己筛选餐品的标准，节省客户的时间。

* + - 1. 收藏商店：有时客户在点多这家商店之后觉得这家的饭菜很符合自己的口味可以将商家店铺收藏到自己的个人中心，以避免后续需要重新从众多商家中再去找之前感兴趣的餐品，导致浪费时间让客户产生不好的使用体验。

3.2.2 商家需求分析

商家需求同样是网上点餐系统需要解决的主要需求，商家可能的需求如下：

1. 客户实名：商家在上架餐品信息的同时也需要将自己的个人信息与餐品绑定以方便对餐品感兴趣的客户进行联系，所有要想使用商家的功能应该先进行实名操作，填写自己的真实信息后才能使用。
2. 上架餐品：上架餐品是网上点餐系统最基本的功能需求，点餐平台需要商家提供餐品及餐品的例如位置，价格，餐品介绍等基本信息，而商家则期望点餐平台展示该餐品并吸引客户达成让客户成功点餐的目的最终因此盈利。
3. 优惠卷管理：为了竞争，商家会在自己的店铺放面额不等的优惠卷来提高自己的成单量，但是也避免不了一些商家恶性竞争，放大额优惠卷。平台对此应该有所管理。
4. 积分管理：老客户是会员之后，每次点完餐就会有积分储存。这些积分可以直接在平台兑换。希望平台也给老用户更多实惠。

3.2.3 管理员需求分析

基本上所有的系统都需要管理员来对系统进行管理，对网上点餐系统同样如此。管理员的功能需求分析如下：

1. 个人信息管理：管理员需要查看当前系统中所有的注册的客户的信息，并以实名和未实名两种类型进行区分方便后续的管理，例如当注册客户出现了违规的个人信息时管理员应该可以警告该客户甚至删除该客户的账号信 息。
2. 商家注册请求：商家把请求发送到服务端，即后台管理员对用户的请求进行审批。有时后台管理员忘记查看信息，系统应该每十二小时或者一显示推送依次商家的注册请求。
3. 商家上架请求管理：商家想要上架餐品时不应该直接在客户输入了餐品的信息后就直接上架该餐品，而是后续交给管理员进行操作。当管理员审核了餐品上架的请求后再由管理员进行决策，确定最后该餐品是否上架。

## 3.3 系统设计

## 3.3 点餐功能设计

基于客户可能的需求，进行客户的功能设计。

客户在使用网上点餐系统的流程图如下：

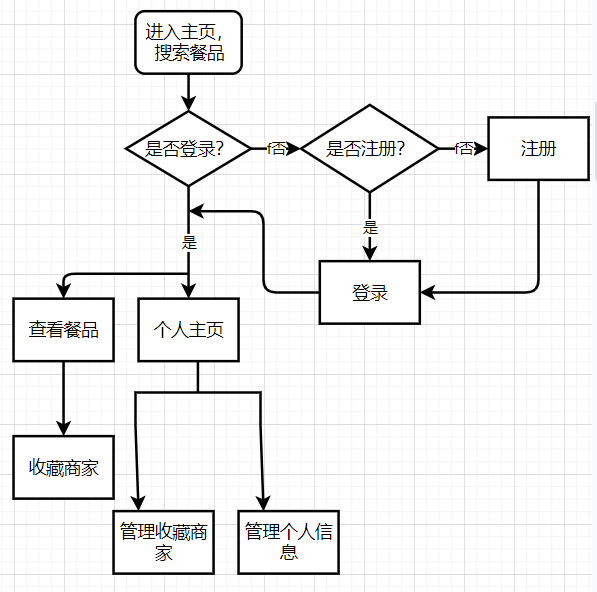


图 3.3.1 客户设计流程图

首先系统设定客户在未进行登录时仍然可以以游客模式在主页面查看餐品

信息，而当客户想要查看餐品的详情信息确定是否该餐品或想要使用个人主页的功能时则必须进行登录操作。

进行登录操作时，首先客户需要之前在网上点餐系统进行过注册操作。初次注册时需要客户输入自己的用户名和密码进行确认，进行注册后才可以使用登录操作。而客户在注册拥有账号之后，在登录页面输入电话用户名、预设的登录密码及验证码即可以进行登录操作。

客户在登录之后，可以继续查看商品信息，但此时客户可以查看到餐品的详细信息特别是餐品商家的信息，并可以进行收藏餐品操作。客户还可以使用个人主页功能，对自己的个人信息进行管理或管理自己的收藏餐品。个人主页的个人信息页面会显示客户注册时所输入的信息，对于实名客户还会额外显示客户的真实信息及性别。租客可以在自己的个人信息页面更改自己的个人信息，或是更改自己的登录密码。另外前面提到客户可以在餐品的详情页面收藏该餐品，而这些收藏的餐品会显示在客户个人主页的网上点单信息页面，客可以继续由该页面查看这些餐品的详情信息或是取消对部分餐品的收藏。

3.3.2商家功能设计

商家实际上也是先作为客户进行注册，在个人主页进行实名操作后才能成为商家并使用商家可用的相关功能。

商家在使用该网上点餐管理系统的流程图如下：

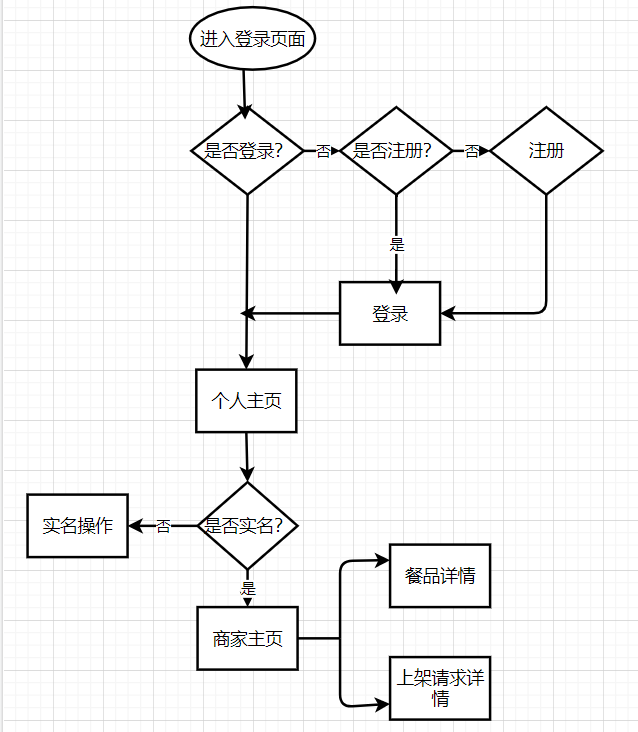


图 3.3.2 商家功能设计流程图

商家可使用的功能必须优先进行登录操作后才能使用，但与客户不同的是

商家在进行注册登录一系列操作后还有在个人主页进行实名操作才能真正使用商家的特有功能。

商家实名需要商家补充输入自己的真实姓名信息、性别信息及年龄信息， 客户实名后可以使用个人主页的商家主页的查看餐品详情功能、查看上架请求的详细功能。

3.3.3管理员功能设计

管理员的功能设计目标分为三类，一类是对客户的个人信息进行管理，一类是对餐品的信息和用户进行管理，最后一类是对商家的上架下架请求进行管理，总体的功能设计图如下图：

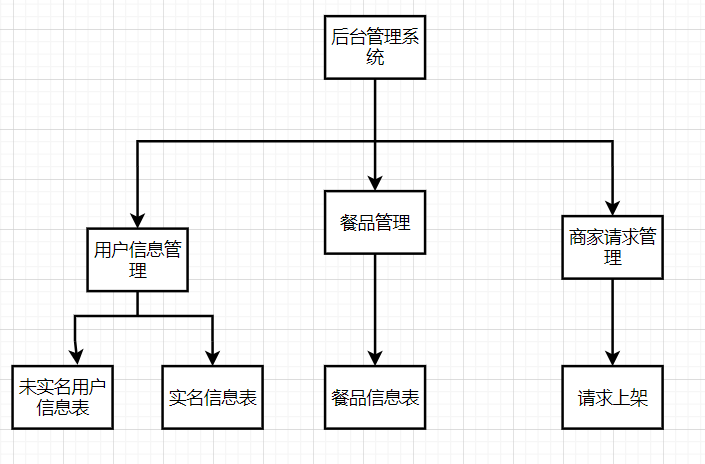


图 3.3.3 管理员功能设计流程图

对于管理员而言，客户信息管理和餐品信息管理更多是查看这些数据及排除其中不合理的数据，最主要的是处理商家的请求。管理员一一查看这些请求并决定是点击批准按钮批准这些请求或是拒绝这些请求。

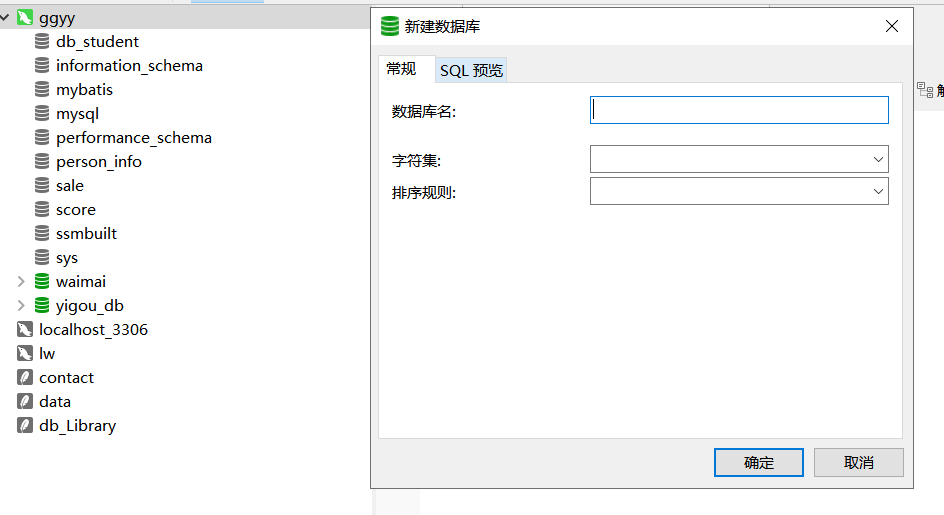
# 4 系统实现

## 4.1 数据库实现

在确定了数据库的设计方案后，可以实际开始创建数据库。数据库的创建有两种方式，一种是不借助数据库工具直接调用控制台输入 sql 语句创建数据库， 一种是借助数据库工具使用可视化页面创建数据库。这里为了后续方便操作的是 Navicat 数据库工具。

具体的操作图如下：

图 4.1.1 创建数据库操作图



在输入了数据库的名称、字符集、排序规则后数据库即可创建完成， 也可以使用 sql 语句‘create database 数据库名’创建数据库。 数据库创建完成后，接下来就是创建数据表的操作。创建数据表继续可以用数据库工具 Navicat 来实现。

图 4.1.2 创建数据库操作图

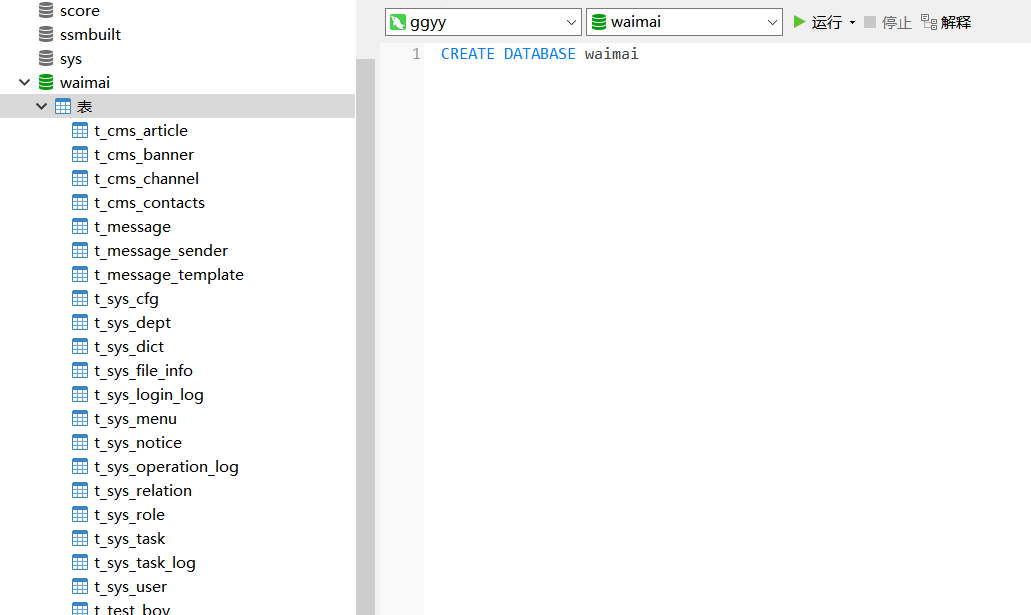
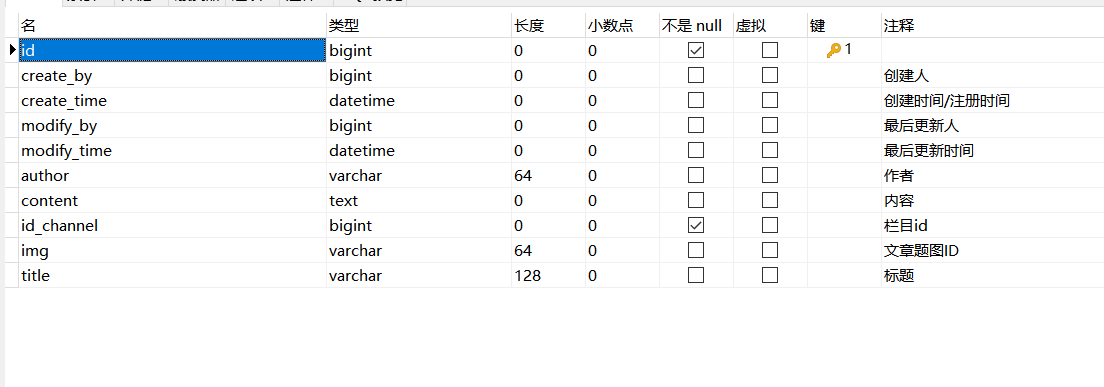
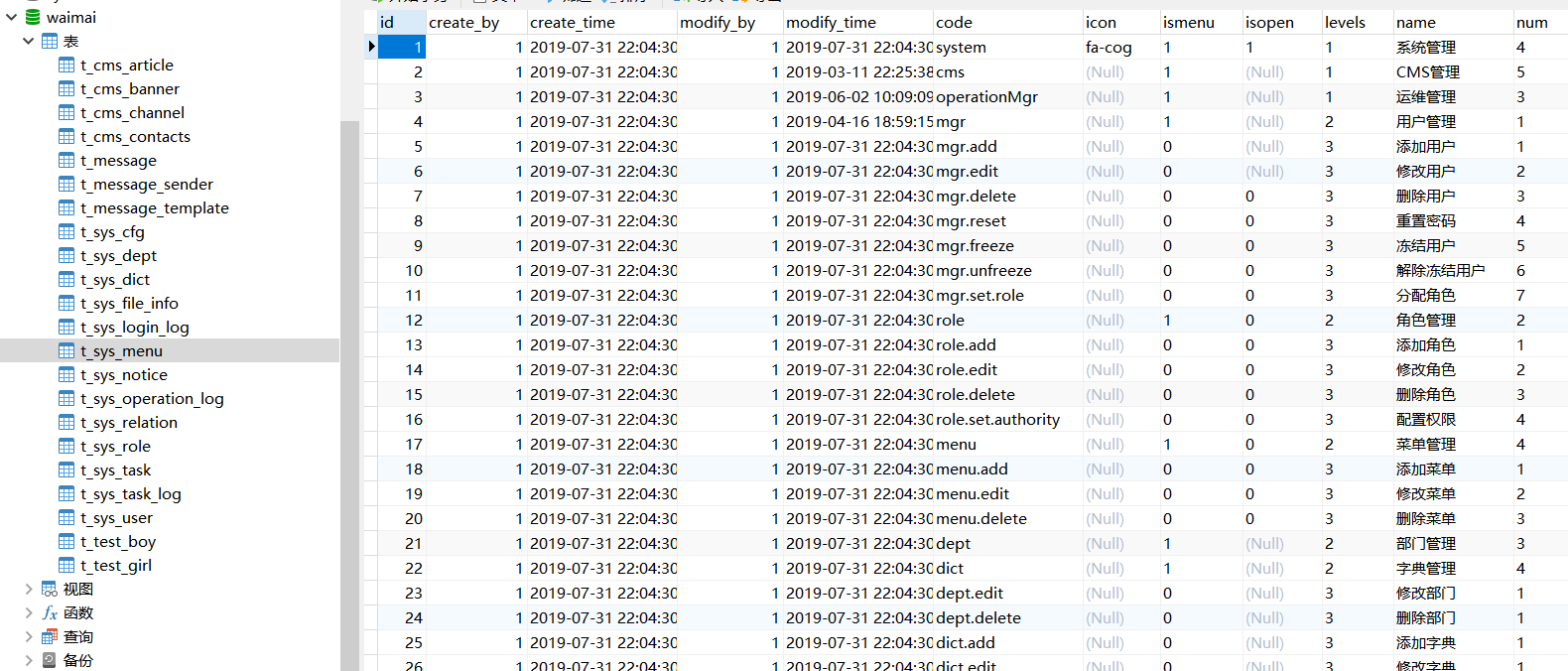


图 4.1.3 创建数据表操作图



点击新建表按钮，补充完成字段后即可完成数据表的创建

图 4.1.4 系统数据表



## 4.2 客户功能实现

客户、商家、管理员的功能实现采用了 SOA设计模式，都是在 springboot+vue框架的帮助下完成这些功能。

大体的功能实现思路按照以下步骤：

1. 创建maven项目，导入相关的jar包依赖。
2. 数据建库连接。
3. 创建实体类，创建Applicatiaon.java 启动类
4. 创建 Controllerr 层，处理前端传过来的请求。

系统按照点单系统设计图来一一实现客户功能，具体的实现图如下：

**4.2.1登录页面功能实现：**

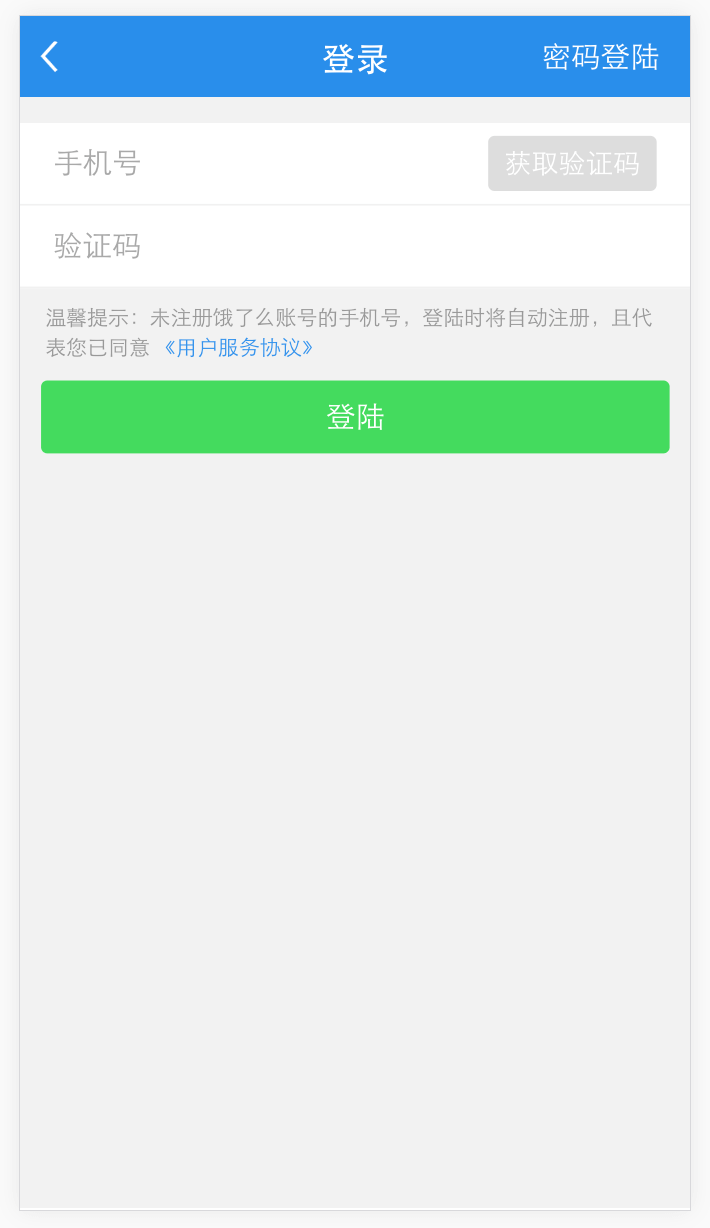


图 4.2.1 系统登录页面



图 4.2.2系统登录页面

已经注册的客户输入手机号码、密码、验证码后即可进行登录，未注册的客户可以点击上方的注册按钮，这样就会跳转到注册页面。

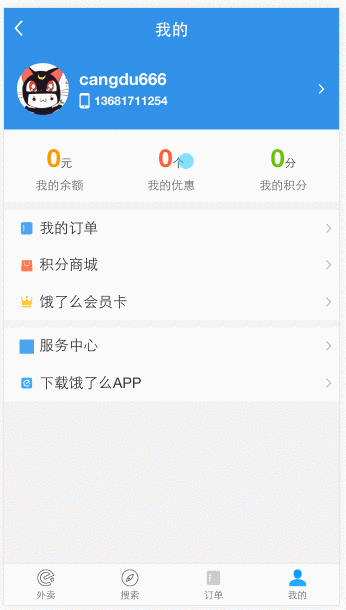


图 4.2.3登录我的页面

客户输入正确的客户名和密码就会我的页面。

**4.2.2.商品搜索页面**



图 4.2.4系统定位

点餐搜索之前，系统会让客户开启手机定位。定位是更精确的搜索到附近的商家返回给客户。

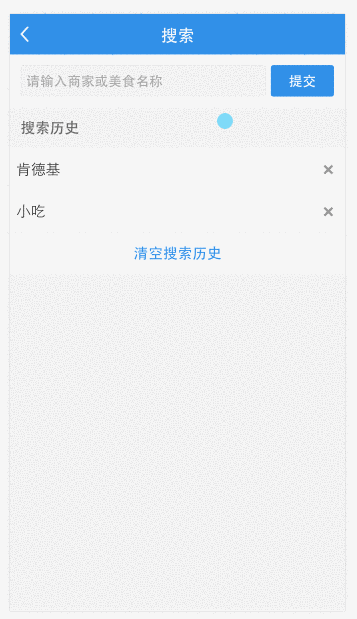


图 4.2.5系统搜索页面

客户点击搜索框进入搜索页面，在搜索框中输出想要的食品。在此页面还会有搜索过的历史和清空搜索历史的记录。



图 4.2.6系统搜索定位

系统通过客户精准的搜索，返回给附近满足客户要求的所有商家。

**4.2.3.支付管理**



图 4.2.7客户点餐

客户选定商家后，进入商家店铺，选择自己的目标产品加入购物车。



图 4.2.7客户下单

客户把商品加入购物车之后，可以选择店铺的优惠卷参加满减，然后在填写自己的姓名，地址和电话之后点击确认下单。支付成功之后即下单成功。

## 4.3商家功能实现

商家上传执照（注册）功能实现：

**4.3.1商家注册**

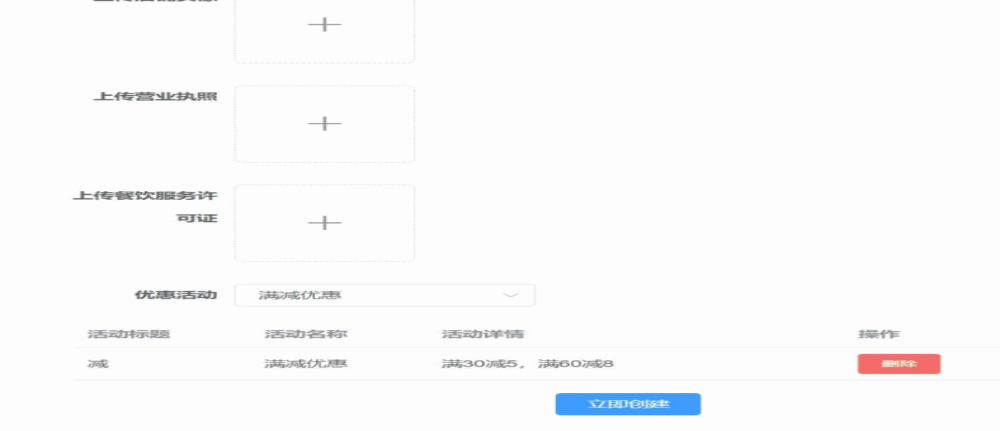


图 4.3.1商家上传营业执照

商家注册账号之后，上传营业执照和餐饮服务许可证等管理员审核通过之后即可在网站中上传食品。上传的同时也可以添加优惠满减的活动。

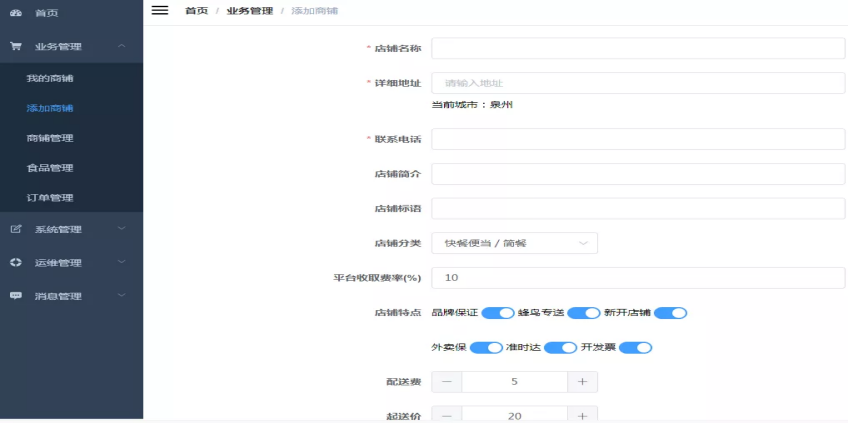


图 4.3.2商家设置店铺信息

注册成功之后即可上传店铺名称，详细地址，联系电话等基本信息。

**4.3.2添加商品管理**



图 4.3.3商家上传营业执照

商家添加食品的种类及其具体详细信息。



图 4.3.4食品管理

商家点击食品管理可以查看食品的具体信息包括食品的ID，评分以及月销售的情况，还有申请的审批状态。

**4.3.3.配送管理**

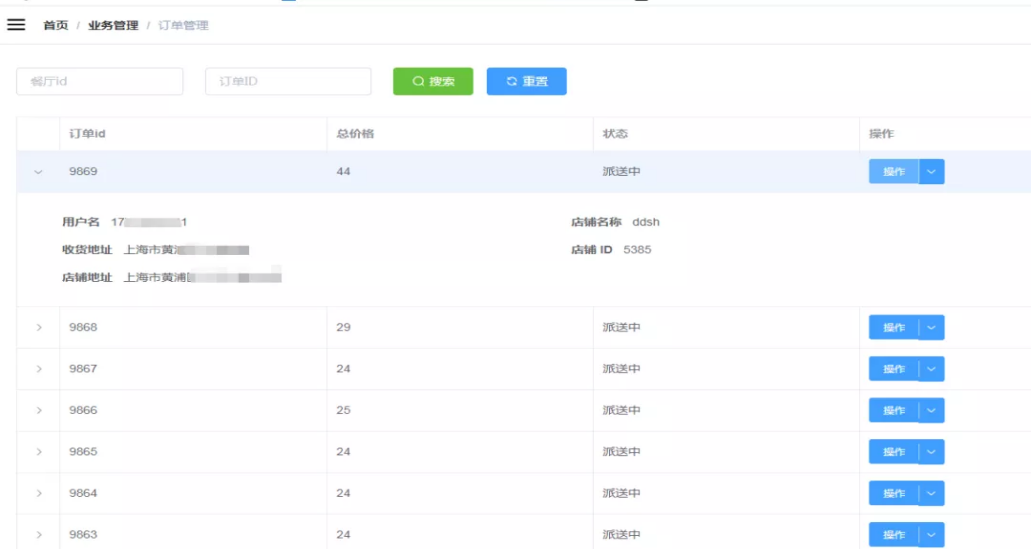


图 4.3.5商品配送

商家点击业务管理，可以查看商品出售的情况，即配送的状态。

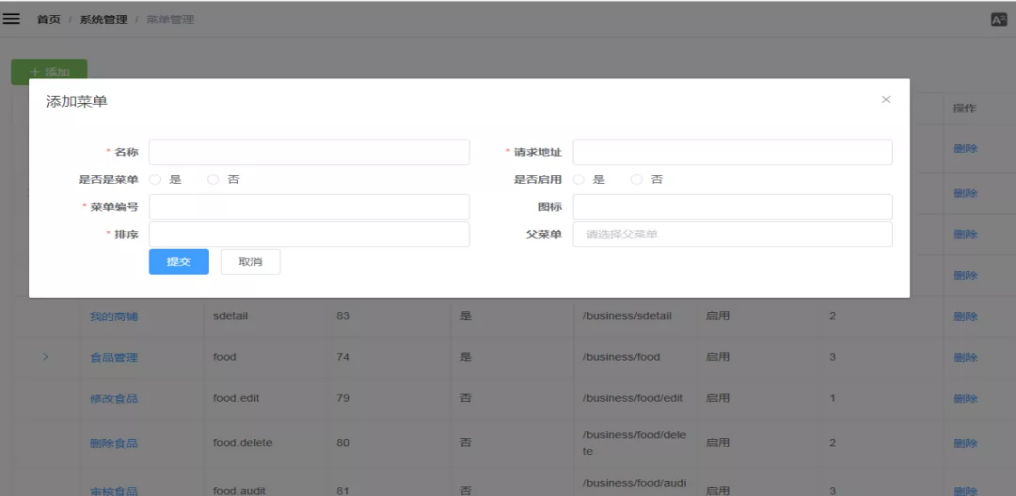


图 4.3.6菜单管理

商家点击添加菜单，可以添加新的菜品。具体包括名称，编号，排序等。

## 4.4 管理员功能实现

管理员的功能实现被设置在了另外的后台管理系统页面，管理员已实现的功能效果如下：

**4.4.1.管理员登录**



图 4.4.1管理员登录

管理员正确输入客户名和账号，点击登进入管理员页面。



图 4.4.2管理员登录

4.4.2.商家管理

管理员成功登录之后，显示新增的用户，未处理消息以及收入。收入包括邮件营销，联盟和视频广告，还包括直接访问和搜索引擎访问的收入。

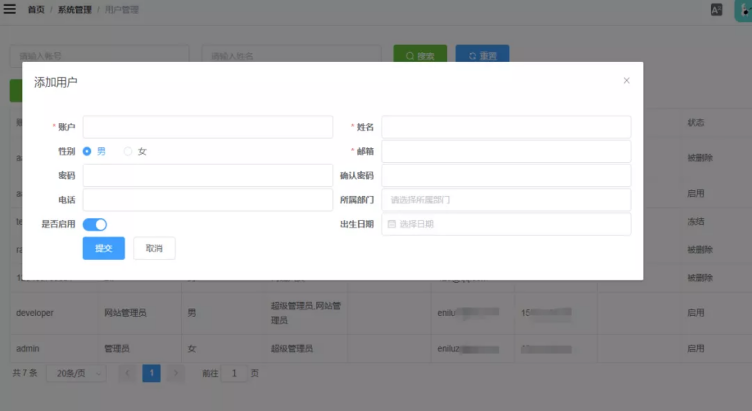


图 4.4.3管理员商家管理

管理员处理商家信息。审核商家的申请，审核通过之后把商家的信息添加到商家管理中。然后把审核的结果返回给商家。



图 4.4.4管理员店铺管理

管理员进入商品管理页面之后，就可以对商家店铺进行管理，即增删改查。

# **5 总结与展望**

参考文献

[1]中国报告大厅.网上点餐行业定义及分类,2015-12-08. <http://m.chinabgao.com/k/fangwuzulin/21100.html>

[2]百度百科.O2O 模式,2020-02-10.

<https://baike.baidu.com/item/O2O/8564117?fr=aladdin>

[3]Casillas J C. O2O Business Patterns and Developing Strategies [J]. International Business Review, 2015, 5(3): 465-479.

1. 李兴华.Java开发实战经典[M]. 北京：清华大学出版社，2009.

[5] 李钟尉，马文强，陈丹丹.Java从入门到精通[M].北京：清华大学出版社，2008.

[6]百度百科.bootstrap,2020-04-10.

https://baike.baidu.com/item/Bootstrap/8301528

[7]百度百科.jquery,2019-10-18.

https://baike.baidu.com/item/jQuery/5385065?fr=aladdin