**基于Java的外卖点餐系统设计与实现**

姓 名： 周杰伦

学 号：

指导教师：

**xxxx年 xx月**

**摘要**

随着外卖行业的快速发展，外卖点餐系统成为餐饮行业的重要组成部分，为消费者提供了便捷的点餐服务，同时也提升了餐厅的运营效率。本系统采用了Java 编程语言，结合Spring Boot 框架作为后端开发技术，MySQL 数据库用于数据存储与管理。前端界面采用Vue.js 技术，提供了友好的用户交互体验。本文将详细介绍系统的功能设计，包括用户注册登录、菜单浏览、菜品管理、订单管理等核心功能。同时，系统还实现了后台管理界面，方便管理员管理菜单、订单和用户信息。通过本文研究与实践，展示了Spring Boot 和 Vue 技术在外卖点餐系统开发中的应用优势，提升了系统的性能和用户体验。未来，可以进一步完善系统功能，引入更多智能化和个性化的服务，以满足不断增长的外卖市场需求。

**关键词**：SpringBoot; MySQL; 系统设计; 点餐系统

**Abstract**

With the rapid development of the food delivery industry, the food delivery ordering system has become an important component of the catering industry, providing consumers with convenient ordering services and improving the operational efficiency of restaurants. This system adopts Java programming language, combined with Spring Boot framework as the backend development technology, and MySQL database is used for data storage and management. The front-end interface adopts Vue.js technology, providing a friendly user interaction experience. This article will provide a detailed introduction to the functional design of the system, including core functions such as user registration and login, menu browsing, dish management, and order management. At the same time, the system also implements a backend management interface, which facilitates administrators to manage menus, orders, and user information. Through research and practice in this article, the application advantages of Spring Boot and Vue technology in the development of takeout ordering systems have been demonstrated, improving system performance and user experience. In the future, the system functions can be further improved by introducing more intelligent and personalized services to meet the growing demand of the food delivery market.

目 录

[1 绪论 5](#_Toc163915683)

[1.1 研究背景 5](#_Toc163915684)

[1.2 研究现状 5](#_Toc163915685)

[1.3 研究目标 6](#_Toc163915686)

[2 相关技术介绍 6](#_Toc163915687)

[2.1 Java语言 6](#_Toc163915688)

[2.2 Spring框架技术 8](#_Toc163915689)

[2.2.1 Spring框架介绍 8](#_Toc163915690)

[2.2.2 SpringMVC框架介绍 9](#_Toc163915691)

[2.2.3 SpringBoot框架介绍 10](#_Toc163915692)

[2.3 Vue前端技术 10](#_Toc163915693)

[2.3.1 Vue框架 10](#_Toc163915694)

[2.3.2 Ant-Design-Vue组件 11](#_Toc163915695)

[2.4 Nginx负载均衡技术 12](#_Toc163915696)

[2.4.1 Nginx应用场景 12](#_Toc163915697)

[2.4.2 Nginx优势 13](#_Toc163915698)

[2.5 MySQL数据库 13](#_Toc163915699)

[3 需求分析 13](#_Toc163915700)

[3.1 系统功能需求分析 13](#_Toc163915701)

[3.1.1 业务需求分析 13](#_Toc163915702)

[3.1.2 数据需求分析 15](#_Toc163915703)

[3.2 系统非功能性需求分析 16](#_Toc163915704)

[3.2.1 系统处理能力需求 16](#_Toc163915705)

[3.2.2 可靠性需求 16](#_Toc163915706)

[3.2.3 可用性需求 17](#_Toc163915707)

[3.2.4 维修性需求 17](#_Toc163915708)

[3.2.5 环境适配需求 17](#_Toc163915709)

[4 系统设计 18](#_Toc163915710)

[4.1 设计原则 18](#_Toc163915711)

[4.1.1 阶段开发原则 18](#_Toc163915712)

[4.1.2 易用性原则 18](#_Toc163915713)

[4.1.3 业务完整性原则 18](#_Toc163915714)

[4.1.4 业务规范化原则 18](#_Toc163915715)

[4.1.5 可扩展性原则 18](#_Toc163915716)

[4.2 总体架构 18](#_Toc163915717)

[4.3 功能设计 19](#_Toc163915718)

[4.4 数据库设计 20](#_Toc163915719)

[4.4.1 设计规则 20](#_Toc163915720)

[4.4.2 表结构设计 20](#_Toc163915721)

[4.4.3 ER图 24](#_Toc163915722)

[5 系统实现 24](#_Toc163915723)

[5.1 前台功能模块实现 25](#_Toc163915724)

[5.1.1 注册登录模块 25](#_Toc163915725)

[5.1.2 首页模块 26](#_Toc163915726)

[5.1.3 详情页模块 27](#_Toc163915727)

[5.1.4 搜索模块 28](#_Toc163915728)

[5.1.5 用户中心模块 29](#_Toc163915729)

[5.2 后台管理模块实现 31](#_Toc163915730)

[5.2.1 用户管理 31](#_Toc163915731)

[5.2.2 分类管理 31](#_Toc163915732)

[5.2.3 标签管理 32](#_Toc163915733)

[5.2.4 菜品管理 32](#_Toc163915734)

[5.2.5 订单管理 33](#_Toc163915735)

[5.2.6 日志管理 33](#_Toc163915736)

[5.2.7 系统信息 34](#_Toc163915737)

[5.2.8 总览模块 34](#_Toc163915738)

[6 系统测试 35](#_Toc163915739)

[6.1 测试策略 35](#_Toc163915740)

[6.2 测试环境 38](#_Toc163915741)

[6.3 测试用例及结果 39](#_Toc163915742)

[6.3.1 后台功能测试用例 39](#_Toc163915743)

[6.3.2 前台功能测试用例 39](#_Toc163915744)

[6.4 测试总结 40](#_Toc163915745)

[7 总结与展望 41](#_Toc163915746)

[参考文献 42](#_Toc163915747)

# 绪论

## 研究背景

随着互联网和移动技术的飞速发展，外卖行业在过去几年中蓬勃发展，成为人们生活中不可或缺的一部分。外卖点餐服务的兴起为消费者提供了便利，也为餐饮行业带来了新的商机和挑战。外卖点餐系统作为连接消费者和餐厅的桥梁，扮演着至关重要的角色。传统的电话订餐方式已经逐渐被线上点餐系统取代，消费者可以通过手机App或网站方便快捷地浏览菜单、下单支付，实现无缝的点餐体验。同时，对于餐厅来说，外卖点餐系统能够提升订单处理效率，降低人力成本，拓展销售渠道，提升竞争力。

在外卖市场竞争日益激烈的背景下，如何提升外卖点餐系统的性能、用户体验和管理效率成为了亟待解决的问题。基于Spring Boot 和 Vue 开发的外卖点餐管理系统，结合了后端技术和前端技术的优势，具有快速开发、高效运行、友好界面等特点，能够有效提升系统的整体质量和竞争力。

因此，本研究旨在探讨如何利用现代技术手段，设计并实现一套高效、稳定的外卖点餐管理系统，以满足不断增长的外卖市场需求，提升用户体验，促进餐饮行业的数字化转型与升级。

## 研究现状

外卖点餐管理系统作为餐饮行业的重要组成部分，受到了国内外研究者和企业的广泛关注与探讨。在国内，随着外卖市场的迅速发展，外卖点餐系统的研究和应用也日益增多。国内外卖点餐系统主要集中在功能完善、用户体验和系统稳定性等方面的研究。

国内外卖点餐系统常采用Java、.NET等技术进行开发，以实现系统的稳定性和高效性。国内研究重点主要集中在系统功能的完善和用户体验的提升，如订单管理、支付功能、评价系统等。一些知名外卖平台如饿了么、美团外卖等已建立起完善的外卖点餐系统，为消费者提供了便捷的点餐服务。

在国外，外卖点餐系统更加注重智能化和大数据分析的应用。一些国外知名企业如Uber Eats、DoorDash等通过大数据分析用户行为习惯，实现个性化推荐和优化配送路线，提升用户体验和系统效率。此外，一些知名菜品如迪士尼乐园也已建立数字化管理平台，实现智能化的访客管理和服务优化。

综合国内外研究现状可知，外卖点餐管理系统的研究重点逐渐向智能化、个性化和数字化方向发展。未来的发展趋势将是结合大数据分析、人工智能等先进技术，设计更加智能化、高效的外卖点餐系统，以提升用户体验和系统运营效率。通过借鉴国内外研究成果，可以为外卖点餐系统的设计与优化提供有益参考。

## 研究目标

本文旨在设计并实现一套高效、稳定的外卖点餐管理系统，以满足不断增长的外卖市场需求，提升用户体验，促进餐饮行业的数字化转型与升级。具体目标如下：

系统性能优化：通过优化系统架构和算法，提升系统的性能和响应速度，确保系统稳定运行并能够应对高并发情况。

用户体验提升：设计直观友好的用户界面，优化用户操作流程，提升用户体验，增加用户粘性和满意度。

智能化功能应用：引入智能化技术如推荐系统、个性化定制等，根据用户历史行为和偏好提供个性化推荐，提升用户满意度和订单转化率。

数据分析与决策支持：利用大数据分析技术对用户行为数据进行挖掘和分析，为餐厅提供决策支持，优化菜单设计、营销策略等。

系统安全性保障：加强系统的安全防护措施，保障用户数据和交易安全，防范各类网络攻击和数据泄露风险。

多终端适配：实现系统在不同终端（PC、平板）的适配，提供一致性的用户体验，满足用户在不同场景下的需求。

通过实现以上目标，本研究旨在为外卖点餐管理系统的设计与优化提供新的思路和方法，推动餐饮行业的数字化转型，提升系统的竞争力和市场影响力。

# 相关技术介绍

## Java语言

Java是一种通用的，基于类的，面向对象的编程语言。它是用于应用程序开发的计算平台。因此，Java是快速，安全和可靠的。它被广泛用于在笔记本电脑，数据中心，游戏机，科学超级计算机，手机等中开发Java应用程序。Java遵循WORA（Write Once, Run Anywhere。一次写入，到处运行）的原理，并且与平台无关。它也简单易学。

Java语言具有如下特点：

（1）一种面向对象的编程语言。

按照对象集合的方式组织程序是一种面向对象的编程思维。面向对象编程的四个主要概念是：抽象化，封装，继承，多态。

（2）一种与平台无关的语言。(它提供了在不同平台下的解释运行环境)

编译器将源代码转换为字节码，然后JVM执行编译器生成的字节码。该字节码可以在Windows，Linux和macOS的任何平台上运行，这意味着如果我们在Windows上编译程序，则可以在Linux上运行，反之亦然。每个操作系统具有不同的JVM，但是在字节码执行后，所有操作系统产生的输出都是相同的。这就是为什么我们称Java为平台无关的语言。

（3）简单

Java是简单的语言之一，因为它不具有指针，运算符重载，多重继承，显式内存分配等复杂功能。

（4）一种健壮的语言，吸收了C/C++语言的优点。

Java语言健壮，意味着可靠。它以这样的方式开发：尽力尽早检查错误，这就是Java编译器甚至能够检测到其他编程语言不易检测到的那些错误的原因。使Java健壮的Java主要功能是垃圾收集，异常处理和内存分配。

（5）具有较高的安全性。(自动回收垃圾，强制类型检查，取消指针)

在Java中，我们没有指针，因此我们无法访问越界数组，如果尝试这样做，它将显示ArrayIndexOutOfBoundsException。这就是为什么在Java中无法利用堆栈破坏或缓冲区溢出之类的一些安全漏洞的原因。

（6）分布式

我们可以使用Java编程语言创建分布式应用程序。远程方法调用和企业Java Bean用于在Java中创建分布式应用程序。Java程序可以轻松地分布在通过Internet连接相互连接的一个或多个系统上。

（7）多线程

Java支持多线程。它是Java的一项功能，它允许并发执行程序的两个或更多部分，以最大程度地利用CPU。

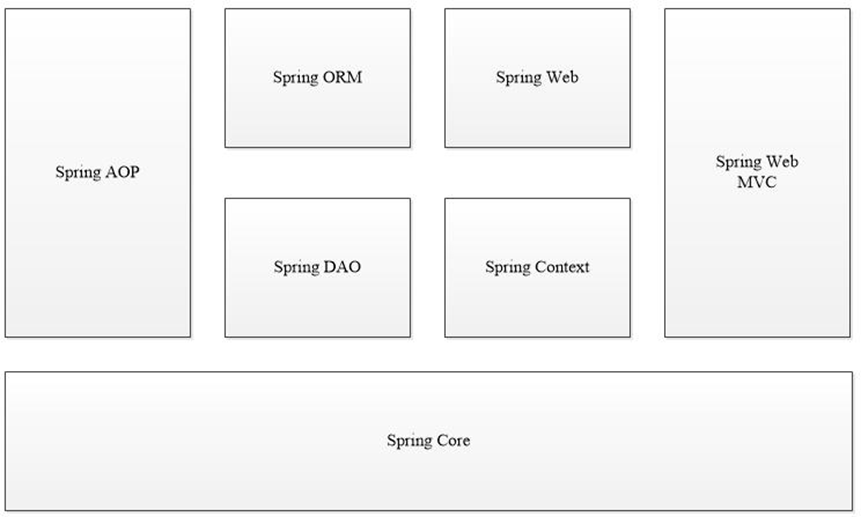
（8）可移植

众所周知，在一台机器上编写的Java代码可以在另一台机器上运行。Java的平台无关功能，其中可以将其平台无关字节码带到任何平台上执行，从而使Java可移植。

## Spring框架技术

### Spring框架介绍

Spring 是一款轻量级的 Java 框架，Spring 框架小巧而且灵活，是为了简化 Java 企业开发而诞生的。它使用一个特殊的类 JavaBean 来管理对象，将 Java 开发从繁杂的配置工作中解放出来。现在 Spring 已经是服务端开发不可获取的开发工具。Spring 框架包含了 7 个模块如图 所示：



Spring 框架模块

Spring有两个重要的特性,控制反转和面向切面。

IOC:Inversion of Control，又称控制反转，是面向对象编程思想中的中的一种设计方法，其思想为将对象的管理权交给 IOC 容器，开发者只关注业务逻辑本身的实现即可。在启动 Spring 时，会由 IOC 容器帮助对象找到相对应的依赖对象并注入，而不是由对象主动去找，这一过程称为依赖注入。依赖注入常用的方法由三种， constructor 构造方法注入、属性注入以及接口注入。本文主要使用的是构造方法注入。

AOP:Aspect Oriented Programming，又称面向切面编程。AOP 事实上并非一种编程技术，而是一种编程思想。即将与业务逻辑本身契合度不高但是又必不可少的代码块从项目中“切”出来，需要的时候可以单独对这个切面进行功能修改而不影响到项目主干内容，然后再将该切面引入原工程即可。同时该思想还能减少代码量，使得代码更工整、易懂，可读性更强。

### SpringMVC框架介绍

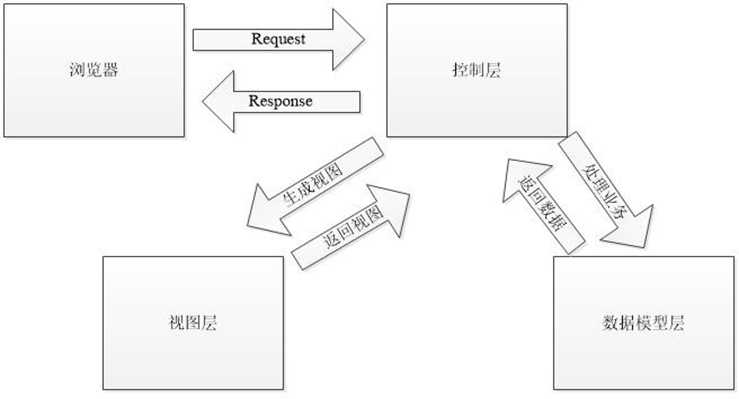
SpringMVC 是 Spring 提供的一个基于 MVC 设计模式的轻量级 Web 开发框架。MVC 设计模式，M（Model）指数据模型层，V（View）指视图层，C（Controller） 指控制层，每一层分别负责不同的功能。

视图层：是用户与系统交互的主要场所，负责将从控制层调取需要的数据并把它们呈现给用户，主要是一些使用 HTML、Flash 或者 XHTML 技术制作的界面。

控制层：控制层本身并不作出任何处理，主要负责接收视图层传入的请求并决定调用特定的模块去对处理请求，然后再指定视图并将结果发送给浏览器。

数据模型层：该层是整个 MVC 架构中最核心的部分，它主要负责封装数据并对数据进行操作。使用 MVC 结构的目的是为了将不同的业务分配给不同的层，实现业务的解耦。

视图层、控制层和数据模型层的关系如图所示：



SpringMVC 三层架构

### SpringBoot框架介绍

SpringBoot 本质上是Spring框架的扩展，在传统的 Spring 开发中，有大量的 XML 配置文件存在于项目中，使得项目变得臃肿，繁琐的配置导致了开发和部署效率的降低，于是 SpringBoot 应运而生。SpringBoot 是 Spring4.0 的升级版，它继承了 Spring 的所有优点，保留了 IOC 以及 AOP 的功能，也同样支持 SpringMVC 的功能；同时消除了设置 Spring 应用程序所需的 XML 配置，简化了配置和部署过程，只需要在pom 文件中添加需要的依赖即可，再加上注解的引入，使得开发过程变得更为高效，便捷。

## Vue前端技术

### Vue框架

Vue是一套用于构建用户界面的渐进式框架。与其它大型框架不同的是，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，还便于与第三方库或既有项目整合。另一方面，当与现代化的工具链以及各种支持类库结合使用时，Vue 也完全能够为复杂的单页应用提供驱动。

组件化开发即是对某些可以进行复用的功能进行封装的标准化工作。组件一般会内含他的内部UI元素、样式和JS逻辑代码，它可以很方便的在应用的任何地方进行快速的嵌入。组件内部可以使用其他组件来构成更复杂的组件。

组件化特点：

（1）组件是对逻辑的封装，不限于图形元素

即我们可以把if做成组件，按一个倒计时做成组件，把一段动画做成组件，把数据架构做成组件，而这些并不能成为控件，这些组件在不同的系统中可以做完美的嵌入。

（2）高内聚、低耦合

前端的组件化开发，可以很大程度上降低系统各个功能的耦合性，数据相互独立，并且提高了功能内部的聚合性。这对前端工程化及降低代码的维护来说，是有很大的好处的，内部结构密封，不与全局或其他组件产生影响，特别是针对逻辑复杂的功能能够进行拆分，更好排查问题。

（3）具备单个可移植性

组件可以提供很好的提升代码的可重用性（而非可复制性），如果有其他项目需要该组件可以直接引入使用，而不是拷贝代码，拷贝资源等。

（4）极大提高开发效率

组件划分后，组件的开发不受其他业务影响，可以多个组件并行开发，加快开发进度。在多人团队里，每个人只负责自己的业务模块，他对业务功能的增删改查，都只限定在自己的这个业务模块里，不会影响其他人的业务，他代码质量的好坏也只会影响到自己的业务模块；如果有新人的加入，可以直接分配组件进行开发，而非需要熟悉整个项目，可以从一个组件的开发使新进人员比较快速熟悉项目、了解到开发规范；对测试来说，也十分方便，大部分情况下，我们只需要着重测试修改过的业务组件即可，而不用老是进行全部回归测试。

（5）功能重用的基石

业务组件类似一个个积木一样，我们可以用积木搭建出不同的房子，同理我们也可以创建多个不同的APP。我们只需要维护好每个组件，需要用到该组件的功能时，一建引用集成就可以了。

### Ant-Design-Vue组件

Ant Design Vue是蚂蚁金服Ant Design官方推荐的Vue 版UI组件库，它其实是 Ant Design 的 Vue实现，组件的风格与 Ant Design 保持同步，组件的html结构和css样式也保持一致。 用下来发现它的确称得上为数不多的完整的 VUE 组件库与开发方案集成项目。Ant Design Vue 是使用Vue实现的遵循 Ant Design 设计规范的高质量UI组件库，用于开发和服务于企业级中后台产品。特性提炼自企业级中后台产品的交互语言和视觉风格。

开箱即用的高质量 Vue 组件。具有如下优势：

（1）共享Ant Design of React设计工具体系。

（2）支持环境现代浏览器和 IE9 及以上。

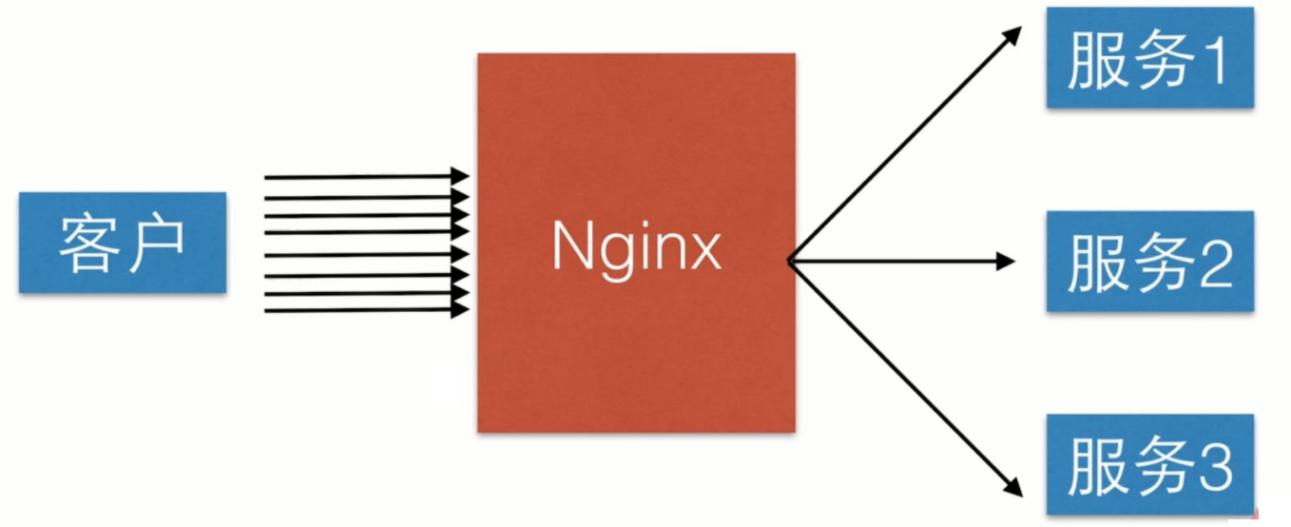
（3）支持服务端渲染。

众所周知，Ant Design 作为一门设计语言面世，经历过多年的迭代和积累，它对 UI 的设计思想已经成为一套事实标准，受到众多前端开发者及企业的追捧和喜爱，也是React开发者手中的神兵利器。Ant-design-vue 能够让 Vue 开发者也享受到 Ant Design 的优秀设计。

Ant-design-vue 是 Ant Design 的 Vue 实现，组件的风格与 Ant Design 保持同步，组件的 html 结构和 css 样式也保持一致，真正做到了样式 0 修改，组件 API 也尽量保持了一致。

## Nginx负载均衡技术

Nginx是一款自由的、开源的、高性能的HTTP服务器和反向代理服务器；同时也是一个IMAP、POP3、SMTP代理服务器；nginx可以作为一个HTTP服务器进行网站的发布处理，另外nginx可以作为反向代理进行负载均衡的实现。



Nginx负载均衡

### Nginx应用场景

**场景一：应用于高访问量的业务**

如果您的应用访问量很高，您可以通过配置监听规则将流量分发到不同的服务器上。

**场景二：横向扩张系统**

可以根据业务发展的需要，通过随时添加和移除服务器，来扩展应用系统的服务能力，适用于各种Web服务器和App服务器。

**场景三：消除单点故障**

当其中一部分服务器发生故障后，负载均衡会自动屏蔽故障的服务器，将请求分发给正常运行的服务器，保证应用系统仍能正常工作。

**场景四：同城容灾 （多可用区容灾）**

为了提供更加稳定可靠的负载均衡服务，当主可用区出现机房故障或不可用时，负载均衡仍然有能力在非常短的时间内切换到另外一个备可用区恢复服务能力；当主可用区恢复时，负载均衡同样会自动切换到主可用区提供服务，保证服务依然正常运行。

### Nginx优势

1. 解决并发压力，提高应用处理性能（增加吞吐量，加强网络处理能力）；
2. 提供故障转移，实现高可用；
3. 通过添加或减少服务器数量，提供网站伸缩性（扩展性）；
4. 安全防护：负载均衡设备上做一些过滤，黑白名单等处理。
5. Nginx支持页面热部署，不用重启服务器，前端升级更无缝。
6. Nginx很好的匹配了本项目中子系统多、服务器部署量大、并发量大、运行效率要求高等特点，因此选用Nginx承担系统部署中的负载均衡工作。

## MySQL数据库

MySQL是一个关系型数据库管理系统，由瑞典MySQL AB 公司开发，属于 Oracle 旗下产品。MySQL 是最流行的关系型数据库管理系统之一，在 WEB 应用方面，MySQL是最好的 RDBMS (Relational Database Management System，关系数据库管理系统) 应用软件之一。

MySQL是一种关系型数据库管理系统，关系数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。

MySQL所使用的 SQL 语言是用于访问数据库的最常用标准化语言。MySQL 软件采用了双授权政策，分为社区版和商业版，由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，一般中小型和大型网站的开发都选择 MySQL 作为网站数据库。

**获取完整版，可联系vx：Lengqin1024**

# 总结与展望

本文采用了SpringBoot框架设计了一套基于分层架构的外卖点餐管理系统。系统完成了数据库和模型的设计，并成功实现了注册、登录、菜品创建、评论、下单购买、收藏点赞等基本功能以及后台管理功能。在系统开发过程中，重点阐述了选择SpringBoot作为开发框架的原因。相较于传统的Spring和J2EE框架需要繁琐复杂的配置，SpringBoot以其轻量化、便捷性和稳定性脱颖而出，极大地简化了开发人员的工作量，缩短了开发周期。通过充分的测试用例覆盖不同场景和数据，系统经过测试表现出较高的性能表现。这一结果符合社会对外卖点餐系统的需求，为用户提供了高效、稳定的点餐体验。本文对于外卖点餐系统的应用研究提供了一套行之有效的解决方案，但是随着研究的深入，了解到还有许多研究方向与优化方案在本文中没有涉及到，包括：

（1）用户界面的优化。本文对于前端界面的设计较为简单，用户界面还不够美观，未来可以尝试使用更多 CSS 样式对网页进行美化，优化用户体验

（2）使用 vite 全页面静态化技术，在服务端完成 html，css 的加载工作， 渲染成纯静态html文件，然后将这些文件直接以静态资源的方式部署到内容分发网络CDN上。这样的做法可以极大程度上提升响应系统速度，也更利于网站的稳定性与安全性。

（3）引入智能算法推荐技术。系统目前还不具有数据挖掘等算法的部分，在后续的开发中笔者会努力学习这部分知识，完善该系统，对旅游数据进行有效的利用。

参考文献

[1]何为,迟文恒. Java服务端研发知识图谱[M].机械工业出版社:信息科学与技术丛书, 201812.430.

[2]杨开振. 深入浅出Spring Boot 2.x[M].人民邮电出版社:, 201808.442.

[3]古凌岚,张婵,罗佳. Java系统化项目开发教程[M].人民邮电出版社:, 201802.313.

[4]肖睿,喻晓路,朱微,张超,戴立坤. Java Web应用设计及实战[M].人民邮电出版社:大数据开发实战系列, 201801.238.

[5]李宗花,朱林. 软件工程原理与实践[M].南京大学出版社:, 202008.252.

[6]陆惠恩,褚秋砚. 软件工程[M].人民邮电出版社:, 201712.284.

[7]肖睿,郭泰,王丁磊. SSH框架企业级应用实战[M].人民邮电出版社:大数据开发实战系列, 201801.344.

[8]付森,石亮,吴起立,刘冰. MySQL开发与实践[M].人民邮电出版社:, 201408.261.

[9]陈陆扬. Vue.js前端开发快速入门与专业应用[M].人民邮电出版社:, 201702.207.

[10]陈晓兵. Web前端开发技术[M].南京大学出版社:, 202008.252.

[11]杨洋,刘全. 软件系统分析与体系结构设计[M].南京东南大学出版社:, 201710.207.

[12]王昊,刘友华. 信息系统分析与设计[M].南京大学出版社:, 202109.323.

[13]张永奎. 数据库原理与设计[M].人民邮电出版社:, 201909.260.

[14]佟伟光,郭霏霏. 软件测试[M].人民邮电出版社:, 201501.281.

[15]郑钢,贺亚涛,尤胜涛. 大型网站服务器容量规划[M].人民邮电出版社:201608.204.

[16]姚庆涛.浅析计算机网络应用安全[J].网络安全技术与应用,2023(05):169-171.

[17]熊威.Java编程基础课程类和对象的教学设计[J/OL].中国教育技术装备:1-4

[18]齐研科,杨颖,贺喜,蔡雨耕,赵家骏,王一先. 软件开发系统及方法[P]. 重庆市：2023-05-02.

[19]朱若榕.人机交互软件界面设计的重要性[J].文化产业,2023(12):153-155.

[20]刘德山,金百东. Java设计模式深入研究[M].人民邮电出版社:, 201407.222.