

基于 Java 的 xxx 的设计与实现

姓 名：_____
学 号：_____
指导教师：_____

2023 年 4 月

摘要

随着我国经济活力的不断提升和 xxx 的快速发展,信息的重要性正在显现出来。xxx 作为 xxx 的重要一环取得了突飞猛进的发展。由于具有高效便捷的优点,xxx 已经成为一种不可或缺的新型生活方式,近年来各大互联网企业 xxx, 获得了巨大成功。而对于这些平台来说,如何在保证交易不出错的前提下应对短时间内的大流量访问,确保交易系统的高性能成为了技术难点。基于当下 xxx 的需求,本系统采用轻量级框架 SpringBoot 进行开发,提供一个满足用户需求进行 xxx 的平台。本系统对 xxx 平台进行详细的需求调研之后,确定了平台所要实现的具体功能,依据面向对象的软件工程方法的流程进行开发。系统采用了当下流行的 SpringBoot 来作为后端开发框架,将数据库框架 MyBatis 作为数据层 ORM 框架,采用 B/S 架构进行开发,使用 Mysql 作为系统的存储模块来进行开发。系统设计包括管理员和普通用户两个角色用户,既为普通用户提供了 xxxx 以及 xxxx 的服务,也让管理员有用户维护、xxx 维护、xxx 维护的平台。在系统开发完成后,采用完善的测试用例对系统平台进行测试,确保该系统是一个的各项功能都能够正常运行、性能良好的 xxxx 平台。

关键词: SpringBoot; MySQL; 系统设计

Abstract

With the continuous enhancement of our country's economic vitality and the rapid development of xxx, the importance of information is appearing. As an important part of xxx, xxx has developed by leaps and bounds. Due to its advantages of high efficiency and convenience, xxx has become an indispensable new way of life. In recent years, various major Internet enterprises xxxx have achieved great success. xxxxxxxxxxxxxxxx

目 录

1 绪论	5
1.1 研究背景	5
1.2 研究现状	5
1.3 研究目标	5
2 相关技术介绍	5
2.1 JAVA 语言	5
2.2 SPRING 框架技术	6
2.2.1 Spring 框架介绍	6
2.2.2 SpringMVC 框架介绍	7
2.2.3 SpringBoot 框架介绍	8
2.3 VUE 前端技术	8
2.3.1 Vue 框架	8
2.3.2 Ant-Design-Vue 组件	10
2.4 NGINX 负载均衡技术	10
2.4.1 Nginx 应用场景	11
2.4.2 Nginx 优势	11
2.5 MYSQL 数据库	12
3 需求分析	12
3.1 系统功能需求分析	12
3.1.1 业务需求	12
3.1.2 数据需求	12
3.2 系统非功能性需求分析	12
3.2.1 系统处理能力需求	12
3.2.2 可靠性需求	12
3.2.3 可用性需求	13
3.2.4 维修性需求	13
3.2.5 环境适配需求	13
4 系统设计	14
4.1 设计原则	14
4.1.1 阶段开发原则	14
4.1.2 易用性原则	14
4.1.3 业务完整性原则	14
4.1.4 业务规范化原则	14
4.1.5 可扩展性原则	14
4.2 总体架构	14
4.3 功能设计	15
4.4 数据库设计	16
4.4.1 设计规则	16

4.4.2 表结构设计	16
5 系统实现.....	19
5.1 前台功能模块实现	19
5.1.1 注册登录模块.....	19
5.1.2 首页模块.....	20
5.1.3 详情页模块.....	21
5.1.4 搜索模块.....	21
5.1.5 用户中心模块.....	22
5.2 后台管理模块实现	22
5.2.1 用户管理.....	22
5.2.2 分类管理.....	23
5.2.3 标签管理.....	23
5.2.4 商品管理.....	23
5.2.5 订单管理.....	23
5.2.6 日志管理.....	24
5.2.7 运营管理.....	24
5.2.8 系统信息.....	24
5.2.9 总览模块.....	24
6 系统测试.....	25
6.1 测试策略	25
6.2 测试环境	28
6.3 测试用例及结果	28
6.3.1 后台功能测试用例.....	28
6.3.2 前台功能测试用例.....	29
6.4 测试总结	30
7 总结.....	31
参考文献.....	32

1 绪论

1.1 研究背景

随着 xxxxxxxxxxxxxxxx。

1.2 研究现状

Xxxxxxxxxxxxxxxxxxxxxx。

1.3 研究目标

xxxxxx。

2 相关技术介绍

2.1 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。

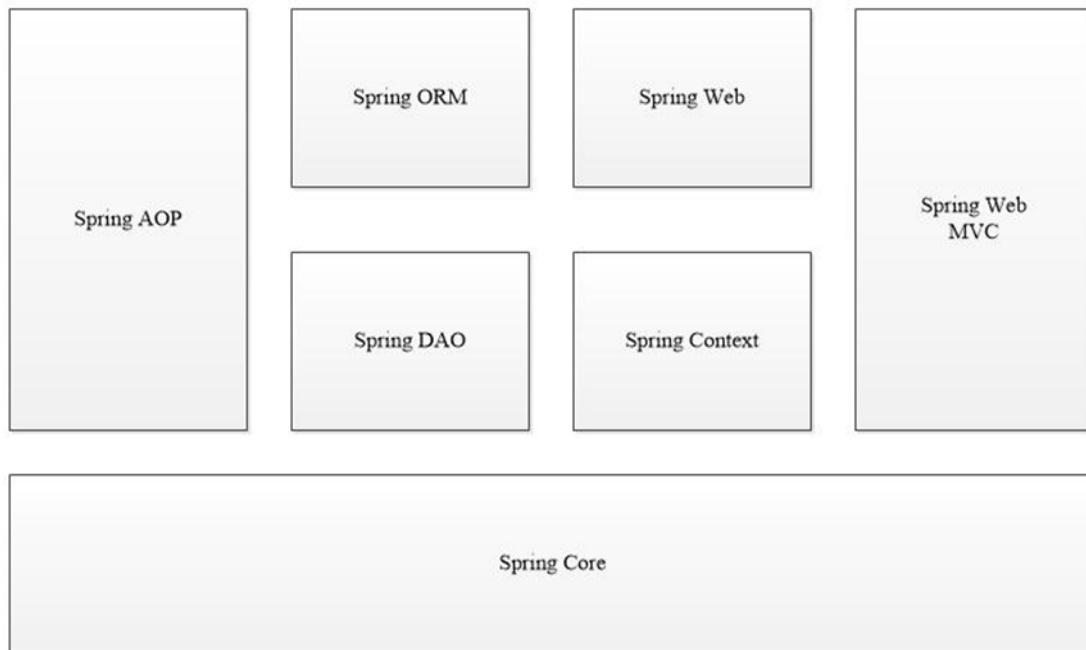
(8) 可移植

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。

2.2 Spring 框架技术

2.2.1 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有三种，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。

2.2.2 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。

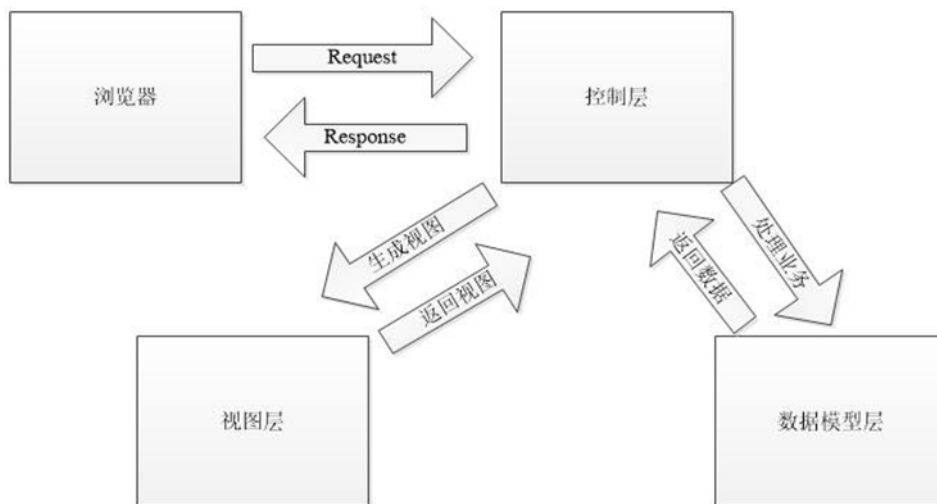
视图层：是用户与系统交互的主要场所，负责将从控制层调取需要的数据并把它们

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

2.2.3 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。

2.3 Vue 前端技术

2.3.1 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。

2.3.2 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：

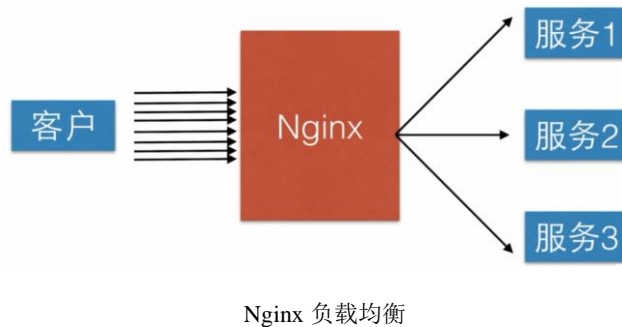
- (1) 共享 Ant Design of React 设计工具体系。
- (2) 支持环境现代浏览器和 IE9 及以上。
- (3) 支持服务端渲染。

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。

2.4 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。



2.4.1 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。

2.4.2 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。

2.5 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。

3 需求分析

3.1 系统功能需求分析

3.1.1 业务需求

.....

3.1.2 数据需求

.....

3.2 系统非功能性需求分析

3.2.1 系统处理能力需求

系统需要具备大规模、高并发的处理能力，在云服务部署的情况下，系统处理能力总需求为：

1. 支持并行同时开展不同类型的数据检索 100 次/秒，支持并行接收、查看检索结果 ≥ 100 ；
2. 并发量大于 100 次/秒时，系统响应时间 ≤ 1 秒。

3.2.2 可靠性需求

软件可靠性是指软件产品在规定条件下和规定时间内完成规定功能的能力。由于功能的持续能力具有概率统计特性，可靠性也可以定义为产品在规定条件

下和规定时间内完成规定功能的概率。为保证数据库检索工作的高效完成，系统软件和数据库需要有较高的可靠性：

- 1.后台服务系统持续性要求：7*24 小时不间断；
- 2.系统有效工作时间 $\geq 99\%$ ；
- 3.系统不间断运行 30 天，运行效率不会明显降低；
- 4.系统故障平均间隔时间 ≥ 300 天。

3.2.3 可用性需求

软件可用性是对软件系统易学、易用、用户满意等方面的评价。系统用户设计管理人员、技术人员、社会人员等，人员素质参差不齐，需要软件具有较高的可用性：

- 1.系统人机交互界面友好，便于使用；
- 2.对于 PC 端业务处理软件，具有相关专业知识的用户在接收 10 个工作日以内的培训后，90%的用户能够独立使用。

3.2.4 维修性需求

软件的维修性是指软件产品在规定的条件下和规定的时间内，按规定的程序和方法进行维修时，保持或恢复到规定状态的能力。在发生故障时，软件应具备较高的可维修性，保障普查工作顺利进行：

- 1.软件系统后台服务具备自动故障恢复功能，单个部署节点发生故障时，自动恢复时间 < 15 分钟，手工恢复时间 < 2 小时；
- 2.软件系统支撑各项功能的在线更新、迭代、升级和扩充，新功能开发测试完成后，更新集成时间 ≤ 5 个工作日，且升级过程中，用户日常使用不受版本更新和维护的影响。

3.2.5 环境适配需求

软件系统环境适配包括对于硬件环境的适配和软件环境的适配。考虑到参与本系统的人员有自主可控软硬件环境的情况，软件的环境适应性包括：

- 1.能够在自主可控的 CPU 硬件设备中部署和运行，性能能够达到设计要求；
- 2.支持各种主流数据库，可以在主流数据库中正常运行并保持性能；
- 3.B/S 架构软件具有良好的浏览器兼容性，支持 Windows 平台以及 Linux 操

作系统下的各类主流浏览器。

4 系统设计

4.1 设计原则

4.1.1 阶段开发原则

系统设计过程中，首先要对系统框架和数据结构全面设计，然后具体功能实现分阶段进行。

4.1.2 易用性原则

系统设计时需要充分考虑最终用户的特点和操作习惯，最大限度地减轻用户在使用软件时的负担，做到部分业务的自动化处理。

4.1.3 业务完整性原则

对于业务进行中的特殊情况能够做出及时、正确的响应，保证业务数据的完整性。

4.1.4 业务规范化原则

在系统设计的同时，也为将来的业务流程制定了较为完善的规范，具有较强的实际操作性。

4.1.5 可扩展性原则

系统设计要考虑到业务未来发展的需要，要尽可能设计得简明，各个功能模块间的耦合度小，便于系统的扩展，当然如果存在旧有的数据库系统，则需要充分考虑兼容性。

4.2 总体架构

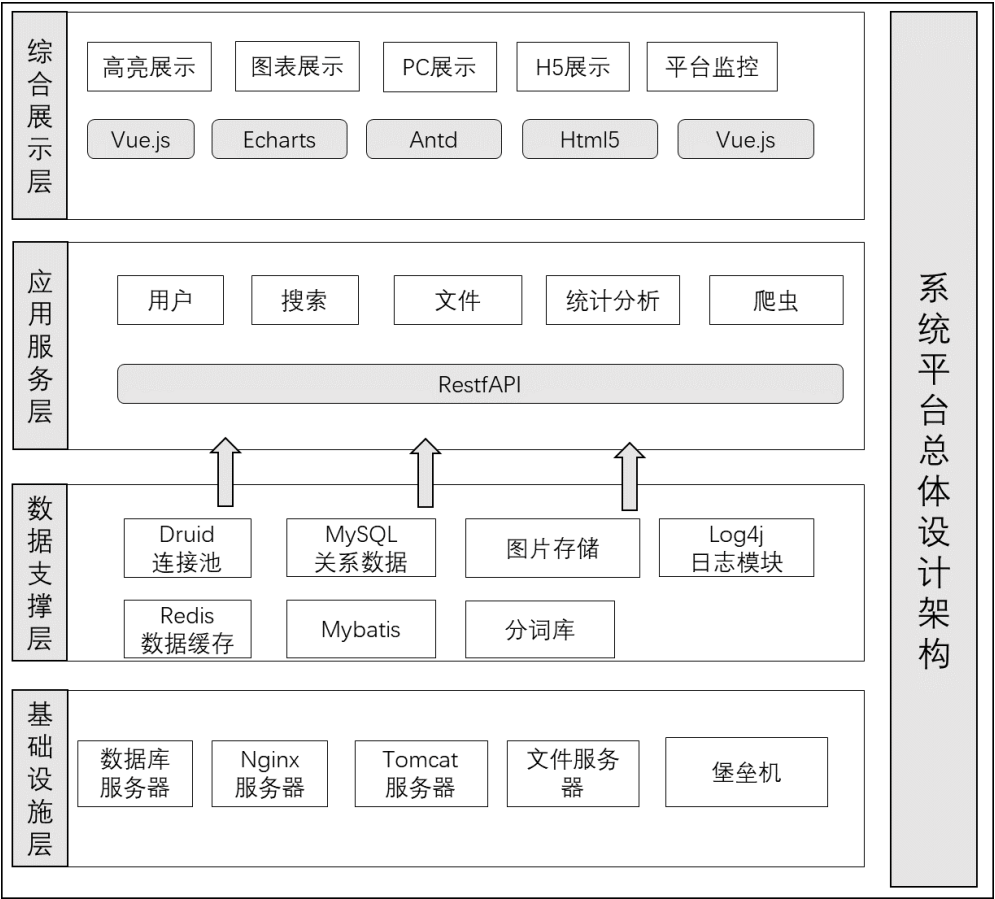
本平台采用了分层设计的思想，将平台所需提供的服务按照功能划分成不同的模块层次，每一模块层次只与上层或下层的模块层次进行交互（通过层次边界的接口），避免跨层的交互，这种设计的好处是：各功能模块的内部是高内聚的，而模块与模块之间是松耦合的。这种架构有利于实现平台的高可靠性，高扩展性以及易维护性。

整个系统按其职能划分为四个模块层次，从下到上依次为：

- （1）基础设施层，保障整个平台的基础框架；

- (2) 数据支撑层，为应用层提供基础数据支撑；
- (3) 应用服务层，各类应用接口的 Restful 封装，包括搜索、用户、文件、统计分析、热门推荐、评论、xxx 等；
- (4) 数据展示层，为用户提供综合展示和使用。

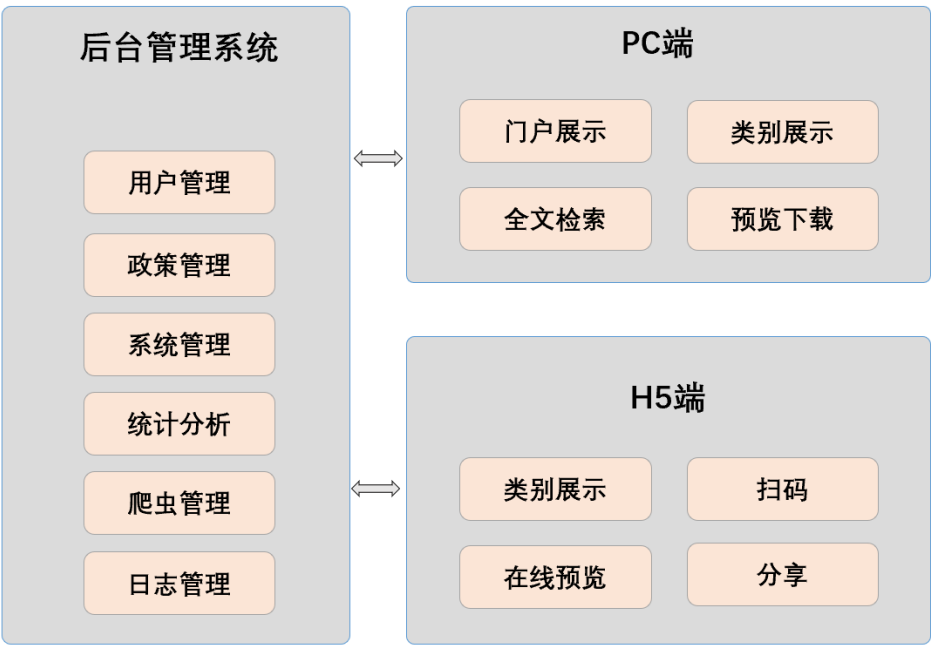
总体架构如下图所示：



总体架构图

4.3 功能设计

系统分为后台管理部分、PC 端部分。其中，后台管理部分包含有用户管理、政策管理、日志管理、统计分析、爬虫管理、系统管理模块；PC 端部分包含网站门户展示、类别展示、全文检索、预览下载功能。整体功能结构设计如图所示。



功能结构图

4. 4 数据库设计

4. 4. 1 设计规则

数据库表命名采用多段式命名，各段之间用下划线分隔，如 `b_xx`（一律小写）。其中：第一段表示模块或者主题，如图书模块为 `b`；第二段表示此表的实体类型，如文件表为 `file`。

字段名规则：列名采用多段式命名，各单词间用下划线分隔，如 `xx_xx`（一律小写）。应尽可能地采用简洁明了的列名以准确描述列的内容含义,根据需要可以一个单词或者多个单词进行命名。

4. 4. 2 表结构设计

本文分别设计了用户信息表、分类表、标签表、操作日志表、错误日志表、评论表、商品表、订单表。

用户信息表						
列名	字段类型	长度	是否为空	是否为主键	默认值	备注
id	bigint		NO	Y		
username	varchar	50	YES			用户名
password	varchar	50	YES			密码
role	varchar	2	YES			角色
status	varchar	1	NO			

nickname	varchar	20	YES			昵称
avatar	varchar	100	YES			
mobile	varchar	13	YES			
email	varchar	50	YES			
gender	varchar	1	YES			
description	longtext	4.29E+09	YES			
create_time	varchar	30	YES			
score	int		YES		0	
push_email	varchar	40	YES			推送邮箱
push_switch	tinyint		YES		0	推送开关
token	varchar	32	YES			
注 1: 用户角色 role 分为普通用户和管理员。						
注 2: 用户状态, 1-停用, 0-正常						

分类表

列名	字段类型	长度	是否为空	是否为主键	默认值	备注
id	bigint		NO	Y		
title	varchar	100	YES			标题
create_time	varchar	30	NO			

标签表

列名	字段类型	长度	是否为空	是否为主键	默认值	备注
id	bigint		NO	Y		
title	varchar	100	YES			

商品信息表

列名	字段类型	长度	是否为空	是否为主键	默认值	备注
id	bigint		NO	Y		
title	varchar	100	YES			名称
cover	varchar	100	YES			封面
description	longtext	4.29E+09	YES			描述
price	varchar	50	YES			价格
status	varchar	1	NO			
repertory	int		NO			库存
score	int		YES		0	分数
create_time	varchar	30	YES			
pv	int		YES		0	
recommend_count	int		YES		0	推荐数
wish_count	int		YES		0	

collect_count	int		YES		0	
classification_id	bigint		YES			

订单表

列名	字段类型	长度	是否为空	是否为主键	默认值	备注
id	bigint		NO	Y		
status	varchar	2	YES			
order_time	varchar	30	YES			
pay_time	varchar	30	YES			
thing_id	bigint		YES			
user_id	bigint		YES			
count	int		NO		0	
order_number	varchar	13	YES			订单编号
receiver_address	varchar	50	YES			
receiver_name	varchar	20	YES			
receiver_phone	varchar	20	YES			
remark	varchar	30	YES			

地址表

列名	字段类型	长度	是否为空	是否为主键	默认值	备注
id	bigint		NO	Y		
name	varchar	100	YES			姓名
mobile	varchar	30	YES			手机号
description	varchar	200	YES			地址
def	varchar	10	YES			默认
create_time	varchar	30	YES			
user_id	bigint		YES			用户 id

评论表

列名	字段类型	长度	是否为空	是否为主键	默认值	备注
id	bigint		NO	Y		
content	varchar	200	YES			评论内容
comment_time	varchar	30	YES			评论时间
like_count	int		NO		0	
user_id	bigint		YES			
thing_id	bigint		YES			

消息表

列名	字段类型	长度	是否为空	是否为主键	默认值	备注
----	------	----	------	-------	-----	----

id	bigint		NO	Y		
title	varchar	100	YES			
content	varchar	1000	YES			
create_time	varchar	30	YES			

日志表

列名	字段类型	长度	是否为空	是否为主键	默认值	备注
id	bigint		NO	Y		
re_ip	varchar	100	YES			
re_time	varchar	30	YES			
re_ua	varchar	255	YES			
re_url	varchar	200	YES			
re_method	varchar	10	YES			
re_content	varchar	200	YES			
access_time	varchar	30	YES			

5 系统实现

在结束了上一章系统设计部分之后,本章将对系统中核心功能模块的实现作出介绍,界面给出系统最终实现的功能效果图以及部分关键代码。本章将系统分为前台功能模块功能和后台管理模块两部分分别进行展示。

5.1 前台功能模块实现

5.1.1 注册登录模块

本系统的后端开发框架采用的是 SpringBoot, 所以使用 SpringBoot 来新建一个 Maven 项目, 应用 SpringMVC 的分层设计思想进行开发。用户需要通过注册和登录后的操作后, 方能进入 xxxx 系统, 首先用户在注册页面填入自己的基本信息, 包括用户名和密码等等。用户注册界面如下图所示:

Xxxx

用户注册成功之后, 可以通过登录界面输入用户名和密码进行登录, 便可进入到 xxxx 平台首页。登录界面如下图所示。用户进入登录界面输入账号和密码后, 登录的 url 的请求发给处理器后, 对用户输入的用户名和密码进行非空校

验,判断账号和密码是否已经填写,并从数据库中询是否有匹配的记录,如果查询成功则登录成功,如果未查询到则提示“用户名或密码错误”,系统将返回到登陆界面继续操作。

Xxxx

登录和注册部分代码如下所示:

5.1.2 首页模块

用户进入到首页后,即可看到 xxx 信息的列表,整个首页的布局分为三部分,顶部是搜索栏,左侧是分类和标签栏,右侧是列表栏。默认情况下,右侧列表栏会展示全部 xxx,当用户点击左侧分类栏的某一项时,右侧列表将会筛选出对应的 xxx。当用户点击左侧某标签时,右侧列表将会筛选出当前标签的所有 xxx。右侧列表栏顶部还有最新、最热、推荐的 tab 栏,当点击最新的时候,当前列表将按照时间倒序展示列表,当点击最热的时候,当前列表将按照 pv 数倒序展示列表,点击推荐的时候,列表将请求推荐接口,按照后端推荐逻辑获取列表。如果用户有搜索的需求,可以在顶部搜索栏输入关键词进行搜索。

首页模块界面如下:

Xxx

首页列表展示的关键代码如下:

5.1.3 详情页模块

用户点击首页列表某个 xxx 时,系统将会跳转到对应 xxx 的详情页,详情页展示的是 xxx 的所有基本信息,包括名称、价格、xxx 状态、库存、封面、简介、收藏数、推荐列表等信息。用户通过点击立即购买按钮,可以购买当前 xxx,然后进入到确认页。页面右侧是交互栏,包括心愿单、收藏、分享等功能。页面下面是简介和评论 tab 页面,用户可以通过点击简介和评论标签来实现切换。如果用户想要评论当前 xxx 时,可以输入关键词,然后点击发送按钮,系统会调用发送评论的接口,将评论内容发送到后端,发送成功后,系统会给出发送成功的提示。除此之外,页面右侧还有热门推荐列表,该列表运用了后端推荐接口将数据获取到前端展示。

详情页模块界面如下:

Xxx

详情页关键代码如下:

Xxx

5.1.4 搜索模块

搜索模块实现了搜索功能,在页面顶端的搜索栏,用户输入关键词后,系统会自动跳转到搜索结果页面,在搜索结果页面,系统会调用后端 search 接口,将关键词内容传给后端,后端通过查询数据库,将列表信息返回给前端,前端页面拿到数据后,在将数据渲染到搜索结果页面。界面如下所示:

Xxx

搜索接口关键代码如下:

5.1.5 用户中心模块

点击顶部用户头像后，系统会跳转到用户中心模块。用户中心模块，集成实现了和用户相关的所有信息的展示，包括订单中心、地址管理、收藏、心愿单、个人资料等信息展示。其中，收藏和心愿单实现了收藏列表和心愿单列表的展示，我的订单实现了我的所有订单的展示，包括订单状态、订单物流信息、支付状态的显示。我的评论模块实现了用户对于所有 xxx 的评论列表。地址管理模块实现了当前用户的所有地址列表。编辑资料模块实现了用户基本信息显示，包括头像、昵称、手机号、邮箱、个人简介等。账号安全模块实现了用户密码的修改功能。消息管理模块实现了后台推送消息的显示。整个界面预览如下：

Xxx

其中订单列表关键代码如下：

Xxx

编辑资料接口关键代码如下：

5.2 后台管理模块实现

当使用管理员账号登录后台管理系统之后，即可进入后台管理主页。管理员在后台有对各个模块的功能的详细管理信息、并拥有对数据的管理的权限，可以对数据进行输入、现有数据进行修改和输出、添加新数据、管理权限等等。

5.2.1 用户管理

在用户信息管理模块，管理员能够修改用户信息，并根据用户身份的不同，给予当前用户相应的角色权限，并且修改状态、邮箱、手机号等信息。如果想新增用户，可以通过点击新增按钮进行新增，如果要删除用户，点击列表右侧的删除按钮进行删除，如果想查询用户，需输入用户名关键词进行搜索。

界面如下所示：

5.2.2 分类管理

点击左侧的分类管理，右侧将展示当前系统的所有分类信息，如果管理员想对当前分类进行编辑，可以点击编辑按钮进行编辑，如果管理员想对当前分类进行删除，可以点击删除按钮就行删除。界面如下所示：

5.2.3 标签管理

点击左侧的标签管理，右侧将展示当前系统的所有标签信息，如果管理员想对当前标签进行编辑，可以点击编辑按钮进行编辑，如果管理员想对当前标签进行删除，可以点击删除按钮就行删除。界面如下所示：

5.2.4 商品管理

在商品信息管理模块，管理员能够修改商品信息，包括修改商品状态、商品名称、价格、库存、封面、分类、标签等信息。如果想新增商品，可以通过点击新增按钮进行新增，如果要删除商品，点击列表右侧的删除按钮进行删除，如果想查询商品，需输入商品关键词进行搜索。界面如下所示：

Xxx

5.2.5 订单管理

在订单信息管理模块，管理员能够修改订单状态，如果想取消订单，可以通过点击取消按钮进行取消，如果要删除订单，点击列表右侧的删除按钮进行删除。界面如下所示：

Xxx

5.2.6 日志管理

在日志信息管理模块，包括三部分：登录日志、操作日志、异常日志。管理员通过点击这三项，在右侧列表栏将会显示对应的日志列表，日志基本信息有请求方式、ip 地址、操作时间等信息。界面如下所示：

5.2.7 运营管理

在运营管理模块，包括两部分：广告管理和通知管理。其中广告管理实现了网站广告设置功能，通知管理实现了整个系统的通知的显示与删除。界面如下所示：

5.2.8 系统信息

在系统信息模块，页面通过请求后端系统信息接口，后端通过系统工具类来实现系统名称、版本信息、操作系统、系统平台、CPU 核数、处理器、CPU 负载、系统内存、内存使用、内存利用率、系统语言、系统时区等信息，并返回给页面进行渲染显示。界面如下所示：

5.2.9 总览模块

在总览模块，系统实现了系统数据的统计分析功能，分别统计了商品数、订单数、网站访问量、热门商品排名、热门分类排名等数据。界面如下所示：

6 系统测试

系统测试是关注系统的外部特性。它的主要对象是进行系统测试的测试工程师。该系统所包含的测试用例范围包括随机抽查信息平台的所有功能测试用例、环境测试用例、性能测试用例以及 UI 测试用例等。

6.1 测试策略

(1) 集成测试

按照模块上下集关系,进行从上到下或者从下到上的集成测试方法进行集成测试,单元测试与集成测试主要考虑功能性测试。同时也对各个模块或者集成模块进行非功能性的抽样测试。

(2) 系统测试

对整合系统进行整合测试,主要测试系统的整体功能和全部非功能性的需求。

(3) 验收测试

验收测试首先进行正规性的测试,由技术人员模拟各用户环境,以用户的身份进行应用测试工作。然后进行非正规测试 alpha 测试和 bate 测试。

Alpha 测试

由开发人员模拟用户进行测试,允许对需求做修改工作。

Bate 测试

alpha 测试后将产品提交给特定用户进行测试,冻结系统需求。

(4) 功能测试

功能测试是保障测试软件各个功能模块正确、逻辑正确。对测试对象的功能测试应侧重于所有可直接追踪到用例或业务功能和业务规则的测试需求。测试的目标是核实数据的接受、处理和检索是否正确,以及业务规则的实施是否恰当。功能测试的主要参考为类似于功能说明书之类的文档。

(5) UI 测试

UI 测试指测试用户界面的风格是否满足客户要求,文字是否正确,页面美工是否好看,文字、图片组合是否完美,背景是否美观,操作是否友好等。

用户界面 (UI) 测试用于核实用户与软件之间的交互。UI 测试的目标是确保用户界面会通过测试对象的功能来为用户提供相应的访问或浏览功能。另外,UI 测试还可确保 UI 中的对象按照预期的方式运行,并符合行业的标准。包括用

户友好性，人性化，易操作性测试。

（6）性能测试

性能测试主要测试软件的性能，包括负载测试，强度测试，容量测试等。

（7）安全性测试

安全性主要体现在以下两个方面：

应用程序级别的安全性，包括对数据或业务功能的访问。

系统级别的安全性，包括对系统的登录或远程访问。

应用程序级别的安全性，可确保在预期的安全性情况下，用户只能访问特定的功能或用例，或者只能访问有限的数据库。

系统级别的安全性

可确保只有具备系统访问权限的用户才能访问应用程序，而且只能通过相应的网关来访问。比如输入管理员账户，检查其密码是否容易猜取，或者可以从数据库中获取。

（8）故障转移和恢复测试

故障转移和恢复测试指当主机软硬件发生灾难时候，备份机器是否能够正常启动，使系统是否可以正常运行。

故障转移和恢复测试可确保测试对象能成功完成故障转移，并能从导致意外数据损失或数据完整性破坏的各种硬件、软件或网络故障中恢复。如突然主机程序发生死机，备份机器是否能够启动，确保系统能够正常运行，从而不影响用户使用。

（9）安装测试

安装测试的目的就是确保该软件在运行环境下进行安装、升级等都能顺利进行。

（10）测试管理

软件测试的主要目的在于发现软件存在的错误 (bug)，对于如何处理测试中发现的错误，将直接影响到测试的效果。在实际测试过程中，对于每个错误，都需要进行测试、确认、修复、验证等管理过程：

（1）错误跟踪管理

为了正确跟踪每个功能项错误的处理过程，将软件测试中发现的每个错误作为一条条记录输入错误跟踪管理系统。正确设计每个错误的包含信息的字段内容

和记录错误的处理信息的全部内容。字段内容应该包括测试软件名称、测试版本号、测试人名称、测试事件、测试软件和硬件配置环境、发现软件错误的类型、错误的严重等级、详细步骤、必要的附图、测试注释等。处理信息包括处理者姓名、处理时间、处理步骤、错误记录的当前状态。正确的数据库权限管理是错误跟踪管理系统的重要考虑要素，一般要保证对于添加的错误不能从数据库中删除。

（2）软件错误的状态

新信息(New)：测试中新报告的软件缺陷；

打开 (Open)：被确认并分配给相关开发人员处理；

修正(Fixed)：开发人员已完成修正，等待测试人员验证；

拒绝(Declined)：拒绝修改缺陷；

延期(Deferred)：不在当前版本修复的错误，下一版修复；

关闭(Closed)：错误已被修复；

未处理(Reopen)：错误未处理；

（3）Bug 管理的一般流程

测试人员提交新的 Bug 入库，错误状态为 New。高级测试人员验证错误，如果确认是错误，分配给相应的开发人员，设置状态为 Open。如果不是错误，则拒绝，设置为 Declined 状态。开发人员查询状态为 Open 的 Bug，如果不是错误，则置状态为 Declined；如果是 Bug 则修复并置状态为 Fixed。不能解决的 Bug，要留下文字说明及保持 Bug 为 Open 状态。对于不能解决和延期解决的 Bug，不能由开发人员自己决定，一般要通过某种会议（评审会）通过才能认可。测试人员查询状态为 Fixed 的 Bug，然后验证 Bug 是否已解决，如解决置 Bug 的状态为 Closed，如没有解决置状态为 Reopen。

（4）软件错误流程管理要点

为了保证错误的正确性，安排有丰富测试经验的测试人员验证发现的错误是否是真正的错误，书写的测试步骤是否准确，可以重复。每次对错误的处理都要保留处理信息，包括处理姓名、时间、处理方法、处理意见、Bug 状态。拒绝或延期错误不能由程序员单方面决定，应该由项目经理，测试经理和设计经理共同决定。错误修复后必须由报告错误的测试人员验证后，确认已经修复，才能关闭错误。加强测试人员与程序员的交流，对于某些不能重复的错误，可以请测试人员补充详细的测试步骤和方法，以及必要的测试用例。

6.2 测试环境

为了保证软件版本的控制，本文建立三个环境，分别是：开发环境，测试环境以及运行环境。



开发环境：软件产品开发工作所用的环境。

测试环境：软件测试工作所用的环境。

运行环境：软件运行的环境。

软件在各个环境中的迁移：

当软件经过开发完毕，将软件产品移植到测试环境进行测试，这样测试和开发工作可以相互独立，互不影响。

当软件测试完成发现错误，开发人员在开发环境中修改错误，修改好后，打成数据包，传输到测试环境进行回归测试。

当软件决定运行时，将软件从测试环境移植到运行环境，供用户使用。

6.3 测试用例及结果

6.3.1 后台功能测试用例

模块名称	测试内容	并发数	状态	平均耗时(秒)
登录首页	1) 管理员登录到后台首页	100	正常	0.218
	2) 前端并发刷新后台首页			
用户管理列表	1) 查看用户列表	100	正常	0.582
	2) 新增一个用户			
	3) 删除一个用户			
	4) 查询用户关键词			
日志管理	1) 刷新操作日志列表	100	正常	0.877
	2) 刷新错误日志列表			
	3) 刷新登录日志列表			

分类管理	1) 新增一个分类	100	正常	0.697
	2) 刷新分类列表			
	3) 删除一个分类			
	4) 更新一个分类			
标签管理	1) 新增一个标签	100	正常	0.233
	2) 刷新标签列表			
	3) 删除一个标签			
	4) 更新一个标签			
评论管理	1) 刷新评论列表	100	正常	0.605
	2) 删除一条评论			
总览	1) 刷新总览页面	100	正常	0.572
订单管理	1) 刷新订单列表	100	正常	0.339

6.3.2 前台功能测试用例

模块名称	测试内容	并发数	状态	平均耗时(秒)
登录	1) 用户登录到前台首页	100	正常	0.594
	2) 并发刷新前台首页			
注册	1) 填写信息注册账号	100	正常	0.382
搜索	1) 顶部输入内容并 Enter	100	正常	0.227
	2) 顶部留空并 Enter			
	3) 刷新搜索页面			
首页	1) 刷新首页	100	正常	0.677
	2) 点击一个分类查看结果			
	3) 点击一个标签查看结果			
	4) 点击一个单项跳转详情			
详情页	1) 多次刷新详情页	100	正常	0.653
	2) 点击收藏			
	3) 点击添加心愿单			
	4) 点击评论按钮并刷新			
	5) 发送一条评论			
	6) 评论点击最新或最热			
	7) 点击喜欢一条评论按钮			
订单确认页	1) 点击确认按钮	100	正常	0.601
	2) 返回上一页			
用户中心	1) 刷新订单信息	100	正常	0.587
	2) 刷新地址管理信息			
	3) 刷新我的收藏界面			
	4) 刷新我的心愿单界面			

	5) 刷新资料界面			
	6) 刷新安全界面			
	7) 刷新推送设置界面			
	8) 刷新消息界面			
通知中心	1) 点击通知中心 icon	100	正常	0.217

6.4 测试总结

从后台并发性能测试用例结果看,该系统可以支持到 100 并发需求,按在线用户和执行事务比例 10: 1 算,可满足 1000 人在线需求。后台管理功能的性能亦达到非功能性需求所定义的指标。从事务高并发时服务器资源占用的情况来看,资源瓶颈大部分在应用服务器的 CPU 时间及数据库服务器的磁盘 IO。说明在同等数据量及服务器配置的情况下,提高应用服务器的 CPU 配置,提高数据库服务器的磁盘 IO 能力,能够进一步提升系统性能。

7 总结

本文使用 SpringBoot 框架设计了基于分层架构的商城项目，完成了数据库和模型的设计，并且实现了注册、登陆、创建商品、评论、购买下单和收藏点赞等基础功能和后台管理功能。在系统中也着重阐述了使用 SpringBoot 作为开发框架的原因，相比于 Spring 和 J2EE 繁琐复杂的配置，SpringBoot 更加轻量化更加方便，同时又比较稳定，极大地减少了开发人员的工作量，并且减少了开发周期。然后通过测试用例对不同的场景和数据进行了测试，通过测试发现该系统具有较高的性能，满足了社会对于 xxxxx 的需要。

本文对于 xxxx 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 phantosjs 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 xxxx 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.