



**JISHOU　UNIVERSITY**

**本科生毕业设计**

|  |  |  |  |
| --- | --- | --- | --- |
| **题 目：** | 基于推荐算法的网上图书销售系统的设计与实现 | | |
| **作 者：** | 宋志华 | | |
| **学 号：** | 2019403971 | | |
| **所属学院：** | 计算机科学与工程学院 | | |
| **专业年级：** | 计算机科学与技术，2019级 | | |
| **指导教师：** | 李宗寿 | **职 称：** | 副教授 |
| **完成时间：** | 2023年 12 月 14 日 | | |

**吉首大学教务处制**

目 录

[摘 要 I](#_Toc153988479)

[**Abstract** II](#_Toc153988480)

[第1章绪 论 1](#_Toc153988481)

[1.1课题现状分析 1](#_Toc153988482)

[1.2与其他研究的区别与优势 1](#_Toc153988483)

[1.3研究内容 2](#_Toc153988484)

[1.4本章小结 3](#_Toc153988485)

[第2章开发环境及技术介绍 4](#_Toc153988486)

[2.1开发技术介绍 4](#_Toc153988487)

[2.2 前后端分离框架 4](#_Toc153988488)

[2.3 Sapling Similarity Collaborative Filtering（SSCF） 5](#_Toc153988489)

[2.4数据库管理系统的选择 8](#_Toc153988490)

[2.5本章小结 8](#_Toc153988491)

[第3章系统需求分析 10](#_Toc153988492)

[3.1可行性分析 10](#_Toc153988493)

[3.2系统业务需求分析 10](#_Toc153988494)

[3.3系统功能划分 11](#_Toc153988495)

[3.4前端页面功能模块 13](#_Toc153988496)

[3.5管理员管理功能模块 13](#_Toc153988497)

[3.6系统UML用例图 15](#_Toc153988498)

[3.7本章小结 17](#_Toc153988499)

[第4章系统设计 18](#_Toc153988500)

[4.1系统设计思想 18](#_Toc153988501)

[4.2推荐算法模块设计 19](#_Toc153988502)

[4.3数据库设计 20](#_Toc153988503)

[4.4数据库实体图 21](#_Toc153988504)

[4.5系统E-R图 24](#_Toc153988505)

[4.6数据库表 25](#_Toc153988506)

[4.7本章小结 27](#_Toc153988507)

[第5章系统的实现 28](#_Toc153988508)

[5.1前端实现 28](#_Toc153988509)

[5.2后端实现 32](#_Toc153988510)

[5.3支付宝沙箱接口接入 34](#_Toc153988511)

[5.4系统管理部分实现 36](#_Toc153988512)

[5.5推荐系统集成 37](#_Toc153988513)

[5.6本章小结 39](#_Toc153988514)

[第6章系统测试 40](#_Toc153988515)

[6.1测试的目的 40](#_Toc153988516)

[6.2测试环境 40](#_Toc153988517)

[6.3测试方法 40](#_Toc153988518)

[6.4测试用例 41](#_Toc153988519)

[6.5测试结论 43](#_Toc153988520)

[6.6本章小结 43](#_Toc153988521)

[第7章总结 44](#_Toc153988522)

[参考文献 45](#_Toc153988523)

基于推荐算法的网上图书销售系统的设计与实现

摘 要

系统设计并实现了一个基于树苗相似度协同过滤推荐算法的网上图书销售系统。该系统旨在通过个性化推荐提高用户购书体验和销售效率。

前端开发中，主要采用了Vue.js作为框架，使用Axios对HTTP请求处理，前端开发技术ElementUI则提供丰富的UI组件库提供舒适华丽的视觉效果，并借助VueRouter管理前端路由，用Echarts提供数据可视化。这些技术的选择保证了系统的高效响应和用户友好的界面设计。后端开发中，选择了Spring Boot框架作为基础，结合Mybatis Plus简化数据库操作，利用Lombok减少样板代码，引入Hutool工具包增强Java编程能力，采用Java-JWT进行身份验证，以确保系统安全。在支付功能上，系统成功接入了支付宝沙盒环境Alipay，为用户提供便捷的在线支付服务。这一特性增强了系统的实用性和竞争力。推荐系统方面，在AmazonBook数据集上进行了测试，集成了最先进的Sapling Similarity Collaborative Filtering算法（SSCF）。与其他模型进行了比较，结果表明 SSCF 在标准数据集上的表现优于其他模型。

总的来说，这个基于推荐算法的网上图书销售系统不仅实现了基本的图书信息展示、订单管理等功能，而且通过高效的推荐算法提升了用户体验，具有较高的实用价值和市场潜力。

关键词：推荐算法；MVC；Java-JWT； SSCF；

**Design and Implementation of an Online Book Sales System Based on Recommendation Algorithms**

**Abstract**

The system has been designed and implemented as an online book sales system based on recommendation algorithms. The aim of this system is to enhance the user's book purchasing experience and sales efficiency through personalized recommendations.

In front-end development, Vue.js was primarily utilized as the framework, with Axios employed for handling HTTP requests. Element UI provided a rich library of UI components, while Vue Router was utilized to manage front-end routing. Echarts were used for data visualization. These technology selections ensured the system's efficient response and user-friendly interface design.In back-end development, Spring Boot framework was chosen as the foundation, combined with Mybatis Plus to simplify database operations. Lombok was utilized to reduce boilerplate code, and the Hutool tool package was introduced to enhance Java programming capabilities. Java-JWT was adopted for identity authentication, ensuring system security. In terms of payment functionality, the system successfully integrated the Alipay sandbox environment, offering users convenient online payment services. This feature enhances the system's practicality and competitiveness.Regarding the recommendation system, testing was conducted on the AmazonBook dataset, incorporating the cutting-edge Sapling Similarity Collaborative Filtering algorithm (SSCF). Comparisons with other models indicated that SSCF outperformed other models on standard datasets.

In summary, this online book sales system based on recommendation algorithms not only realizes basic functions such as book information display and order management but also enhances the user experience through efficient recommendation algorithms, making it have high practical value and market potential.

**Key words:** Recommendation Algorithm；MVC；Java-JWT； SSCF；

第1章 绪 论

1.1 课题现状分析

随着互联网技术的发展和人们阅读习惯的改变，网上图书销售系统用推荐系统的需求日益增加。因为，传统的图书销售方式存在一些问题，如信息不对称、用户需求难以满足等。因此，不少研究设计和实现了一个基于推荐算法的网上图书销售系统[1]，可以有效地解决这些问题。

传统的图书销售方式主要依赖于用户的主动搜索，这种方式往往不能满足用户的潜在需求。而推荐算法可以根据历史行为加上兴趣偏好模拟建模并预测图书，从而为用户推荐可能感兴趣的图书，降低用户的择书时间，提高用户的下单率和幸福值[2]。

此外，基于推荐算法的网上图书销售系统还可以通过数据建模，展现用户抉择模型，帮助卖家更好地理解需求，改善货物结构，提高销售额。同时，该系统也可以通过个性化推荐，提高了用户的黏性，增强了用户的忠诚度[3]。

最后，网上图书销售系统很多并没有用到最前沿的推荐算法。而且推荐系统一般需要基于公司级别的数据训练[4]，如何寻找合适的数据是个问题。之后则可以利用机器学习和数据挖掘方法来实现高质量的推荐。但机器学习可解释性较弱，性能要求较高，相反基于记忆的协同过滤简单直观，可解释性强[5]，如何选择合适的推荐算法，需要根据不同情况决定，如系统数据量等等。

综上所述，设计和实现一个基于推荐算法的网上图书销售系统具有重要的现实意义。本论文将对这一课题进行深入研究，探讨如何设计和实现这样一个系统，并对其性能进行评估和优化。

1.2 与其他研究的区别与优势

论文主题是基于推荐算法的网上图书销售系统的设计与实现，采用了前端Vue技术和后端SpringBoot框架，并集成了Sapling Similarity Collaborative Filtering推荐算法。与现有研究相比，论文有以下区别和优势：

技术栈组合：选择了前端Vue技术，配合Axios、ElementUI和Echart等工具进行开发，这些技术都是现代Web开发中的热门选择，能够提供良好的用户体验。而后端使用SpringBoot框架，搭配MybatisPlus、Lombok、Hutool和Java-JWT等工具，数据库用MySql，构建了一个高效稳定的后台服务。

推荐算法：当前最先进的协同过滤方法通常是基于模型的，使用诸如图卷积网络（GCN）和矩阵分解（MF）等技术，这些技术依赖于优化的超参数以最大化其性能。尽管如此，在如经济复杂性框架等实际应用中，由于其更高的简单性和可解释性，基于记忆的方法更受青睐于基于模型的方法，并被广泛用于预测和推荐目的[6]。本系统集成了Sapling Similarity Collaborative Filtering推荐算法[7]，并在AmazonBook数据集上进行了测试。这种算法基于决策树和记忆，能有效挖掘用户兴趣，提供个性化的推荐服务。并在多个数据集上取得了the state of art。并且图书推荐系统的数据采用Amazon-book数据集，这是一个企业级的数据。

支付接口：系统接入了支付宝沙盒，实现了在线支付功能，增强了系统的实用性。

数据可视化：利用Echart展示了销售数据，使得数据分析更加直观，有助于管理者更好地决策。

完整的系统设计与实现：论文涵盖了从需求分析到系统实现的全过程，不仅有理论探讨，也有实践应用，具有很高的实用价值。

综上所述，论文在技术选型、推荐算法、支付接口、数据可视化和系统完整性等方面都具有一定的创新性和优势。该研究还可以为其他领域的推荐系统设计提供参考和借鉴。推荐系统不仅在电子商务领域有着广泛的应用，在新闻推送、音乐推荐、电影推荐等领域也有着巨大的潜力。通过对网上图书销售系统的推荐算法进行深入研究，可以从中提炼出一些通用的设计原则和方法，为其他领域的推荐系统设计提供有价值的参考。

1.3 研究内容

基于推荐算法的网上图书销售系统的设计与实现的研究内容主要包括以下几个方面：

需求分析：首先，需要对网上图书销售系统的使用场景和用户需求进行深入调研。这包括了解用户的购买习惯、浏览行为、搜索偏好等信息，以确定系统的功能需求。此外，还需要考虑系统应具备的基本操作，如书籍搜索、浏览、购买、评价等。

系统设计：在明确了用户需求后，进行系统的设计。这包括数据库设计（如书籍表、用户表、订单表等）、界面设计（如首页、商品详情页、购物车等）、以及推荐算法模块的设计。在设计过程中，要确保系统的易用性和可扩展性。

推荐算法设计：根据用户群体画像和明确的需求细节，选择合适的推荐算法或设计新的混合推荐策略。

系统实现：基于系统设计文档，进行软件开发。开发人员需要熟悉相关的开发语言与构建工具，以及良好的编码规范，以保证系统的鲁棒性和解耦合性。

推荐模型训练与评估：利用收集到的数据，训练推荐模型。同时，还需要不断调整和优化推荐算法，以提高推荐效果。

系统测试与优化：完成系统开发后，进行系统测试，如采用单元测试、集成测试、系统测试等，遵循黑盒测试与白盒测试的思想。在此过程中，发现的问题应及时修复，并对系统进行持续优化，提高用户体验。

用户反馈与迭代改进：系统上线后，需要收集用户反馈，对系统进行持续的迭代改进。这包括对推荐结果的满意度调查，以及对系统功能的完善和优化。

1.4 本章小结

本章首先介绍了图书推荐系统的背景，分析了目前图书推荐系统存在的诸多问题，其次介绍了本次基于树苗相似度推荐算法的研究意义以及与其他图书推荐系统研究的区别与优势，介绍了网页前后端开发的相关内容，以及网上图书销售系统的用途。最后，对设计流程做出简洁的小结。

第2章 开发环境及技术介绍

2.1 开发技术介绍

以下为基于推荐算法的网上图书销售系统的设计与实现所运用的开发技术介绍、使用的原因分析：

(1) 开发工具：选择了IntelliJ IDEA作为主要的开发工具。IDEA提供了丰富的插件和功能，最近推出大语言模型推荐助手，可以结合海量数据去很好的解答编程过程中遇到的任何问题。此外，它支持多种编程语言，包括本次使用的VUE，合技术Spring Boot管理，方便在一个统一的环境中进行多语言的开发[8]。

(2) 数据库：为了存储用户数据、书籍信息以及推荐结果等数据，选择了MySQL。MySQL是一个使用人数众多，发行时间久远，具有广泛的社区支持和大量的学习资源[9]。它的功能完善，易于操作和维护，适合项目需求。

(3) 前端技术：在前端开发中，采用了Vue.js框架，它具有数据绑定的特性可以实时更改页面显示的数据。，在http数据前后端交互方面，结合axios库。对于前端的页面元素的美观与可视化展示，使用Element UI组件库提供丰富的UI元素和样式。Vue.js以其轻量级、易上手的特点，能够快速构建出响应式的用户界面[10]。Element UI则提供了大量预定义的UI组件，简化了页面布局和样式设计的工作，最后Echarts提供数据可视化。

(4) 后端技术：在后端开发中，选择了Spring Boot框架，利用Mybatis Plus作为持久层框架，同时使用Lombok来减少样板代码。Spring Boot是Java领域最流行的微服务框架之一[11]，具有强大的自动化配置和便捷的开发特性。Mybatis Plus则是对Mybatis的增强，提供了许多实用的功能，比如通用CRUD操作、分页插件等[12]。Lombok通过注解的方式，在编程时非常方便，而在之后可以生成getter、setter等方法，节约时间。

综上所述，选择的这些技术和工具都是使用人数非常多、稳定且拥有巨量网上的帮助文档的，它们能够帮助高效地完成系统的开发工作，并确保系统的性能和稳定性。

2.2 前后端分离框架

在前后端分离的架构中，前后端各自负责不同的任务，并且可以独立开发和部署。具有很高的解耦合性，方便后续的维护合拓展。

前端主要负责用户界面的展示和交互，技术更新迭代非常快，它由Vue、以及部分CSS和JavaScript等Web技术构建。前端应用运行在用户的各平台浏览器上，直接与用户体验挂钩，提供视觉效果和操作反馈。随着现代前端框架如React、Vue.js和Angular的发展，前端应用的功能越来越强大，甚至可以处理一些原本需要在服务器端完成的任务。

后端则专注于业务控制。它主要包括服务器端的应用程序和数据库管理系统。后端使用各种编程语言（如Java、Python、Node.js等）和框架（如Spring Boot、Django、Express等）来实现功能。后端应用接收来自前端的axios请求，然后进行相应的业务处理并从Mysql中获取或更新数据，然后将结果通过接口返回给前端。

前后端分离的优势包括：分工明确：前端和后端开发者可以分别关注自己的领域，提高开发效率。可扩展性：由于前后端是解耦的，因此可以在不影响另一方的情况下独立扩展或升级。更好的用户体验：前端可以专注于优化用户界面和交互，提升用户体验。更高的安全性：后端可以集中处理安全问题，确保数据的安全传输和存储。易于维护和测试：每个部分都可以单独测试和调试，使得系统的维护更加容易[13]。

总的来说，前后端分离是一种现代Web开发的最佳实践，越来越多的人选择使用它来进行web构建精美的系统，它不仅提高了开发速度，也提升了系统的响应时间以及可用性和用户体验。

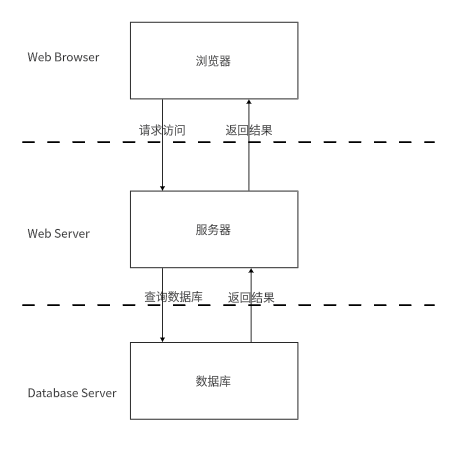


图1 WebServer前后端分离架构

2.3  Sapling Similarity Collaborative Filtering（SSCF）

Sapling Similarity Collaborative Filtering算法（SSCF）是适用于推荐系统中的记忆式协同过滤技术。Sapling Similarity受到决策树工作原理的启发，允许计算出正负两种相似值，而不仅仅是传统的基于共同邻居的正相似度。在传统的协同过滤中，有两种主要的推荐策略：用户基于和项目基于[14]。然而，现有的基于共同邻居的相似度计算方法存在一些局限性。首先，它们只允许正相似度，忽略了两个用户（或项目）可能非常不相似的情况[15]。其次，尽管它们具有较高的可解释性，但在推荐结果上不如基于模型（机器学习）的方法。

Sapling Similarity的关键思想是观察一个用户连接到一个项目的信息如何影响对另一个用户也连接到同一项目的先验概率估计：如果这个概率降低了，那么这两个用户的相似度就会是负的，否则就会是正的。这种方法能够捕捉到用户或项目之间的潜在不相似性或反关联性。在构建基于Sapling Similarity的记忆式协同过滤系统（SSCF）时，将用户基于和项目基于的方法结合起来，形成一种混合方法。实验结果显示，当用于构建推荐系统时，Sapling Similarity提供的推荐效果优于现有的相似度指标。并且，相比较于最先进的图卷积网络仅依赖于一个直观的超参数，其推荐准确性也与最先进的模型相当，在Amazon-Book数据集上甚至超越了所有其他模型，同时保留了记忆式方法的高度可解释性。因此，图书推荐算法将采用Sapling Similarity来计算用户和书籍之间的相似度，通过考虑正负相似性来提供更精确和个性化的图书推荐。这种算法不仅具有高性能的推荐能力，而且由于其基于记忆式协同过滤的方法，具有高度的可解释性，能够清晰地解释为什么某个书籍被推荐给特定的用户。

以下给出Sapling Similarity简略计算方法具体推导过程见参考文献[7]

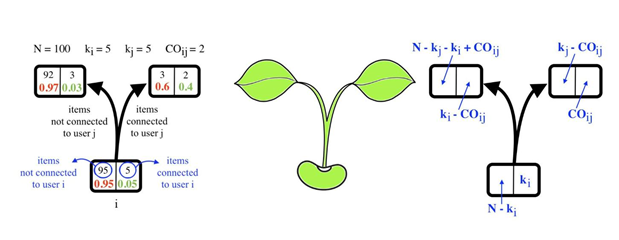


图2  Sapling Similarity（树苗相似性）

如上图所示，N为物品总数，为用户i喜欢的物品数量，为用户j喜欢的物品数量，表示用户i和用户j共同喜欢的数量。例如上图左边树苗根部豆子左侧表示对总数100物品中的95个表示不感兴趣或未评价，右侧表示对5个表示好感，左叶表示不感兴趣中的或未评价95个物品中的豆子，对于用户j有92个不感兴趣或未评价，3个表示好感，同理右侧表示对于用户i喜欢的5个物品中，用户j有三个不感兴趣或未评价，2个表示喜欢，也就是为2。计算用户i与用户j的Sapling Similarity（），我们使用了基尼不纯度（Gini Impurity，GI）。其中表示的感兴趣的占总的比例，表示不感兴趣或未评价占总的比例。

公式 1

如上图基尼不纯度在 p0 或 p1 为0时达到最小值0，在 p0 = p1 = 0.5 时达到最大值0.5。因此，GI 越低，箱子里的极化或确定性就越高。我们可以得出根部豆子基尼不纯度公式GI，左边叶子基尼不纯度公式以及右边叶子基尼不纯度公式为：

公式 2

公式 3

公式 4

结合以上三个公式，我们定义用户i与用户j的Sapling Similarity（）的绝对值为：

公式 5

公式 6

公式 7

其中，是用来配置权重，而GI用来归一化基尼变化量。值得注意的是的符号取决于以下公式。

公式 8

换句话说，如果i感兴趣的物品中，大多数的物品， j也非常感兴趣，以至于比例都大于i在总物品中感兴趣的物品所占的比例，则符号为正，反之则为负。在SSCF中，首先评估用户和项目之间的Sapling矩阵B(user)和B(item)。然后，构建用户基于亦或者项目基于的协同过滤。推荐项目α给用户i的置信值由以下等式给出：（ 表示在二部图中，用户l与项目α之间是否存在连接，是用户i和用户l之间的Sapling相似度）

公式 9

公式 10

SSCF被定义为项目基于和用户基于估计的加权平均值：

公式 11

其中γ是模型中的唯一参数，它调节在这种混合方案中给予用户基于和项目基于推荐的相对权重。通过这种方式，SSCF结合了用户和项目之间的相似性，以提供更准确的推荐。

总的来说Sapling Similarity（树苗相似性）是一种高效的推荐系统算法，它基于记忆式协同过滤，并能够处理非正相关的用户间关系。这种算法受到决策树启发，通过对用户连接到项目的先验概率影响进行分析，从而确定用户间的相似性。同时，该算法也具有较高的可解释性，可以帮助研究人员更好地理解用户间的关联性。

2.4 数据库管理系统的选择

在设计和实现一个基于推荐算法的网上图书销售系统的数据库管理系统时，考虑到我使用的数据集为Amazon-book 这个企业集的数据，系统需要处理大量数据和高并发访问，可能需要选择一种高性能的数据库管理系统。Amazon-book有有效用户数据52970条，有效书籍数据91599条，有效评估数据10467335条，部署服务器为阿里云两核4GB内存。所以选择MySql作为数据库管理系统。MySQL在处理大量数据和高并发请求时表现出色，通过高效的查询优化器和索引技术，能够提供快速的数据检索和更新。MySQL提供了丰富的数据类型、存储引擎、安全措施和管理工具，支持事务处理、视图、触发器、存储过程等功能。MySQL有一个相对简单的语法结构，对于初学者来说比较友好。同时，有大量的在线资源、文档和社区支持可以帮助用户解决问题和学习。

2.5 本章小结

在本章中，主要详细介绍了树苗相似度推荐算法（SSCF）的由来，和树苗相似度推荐算法（SSCF）是如何计算相似度的，提供了运算的数学公式。

本章首先，介绍了前后端分离。其中Element UI组件库提供丰富的样式和交互效果，销售情况由Echarts呈现。这样的技术选型使得前端界面美观且交互流畅，提高了用户体验。

后端则选择了Java作为编程语言，Spring Boot作为Web框架，Mybatis Plus作为持久层框架，Lombok用于简化Java对象的创建，Hutool提供了多种工具类方法，java-jwt用于用户身份验证，Maven作为项目管理工具。这些技术和工具的选择，保证了系统的稳定性和可维护性。

在推荐系统设计上，集成了Sapling Similarity Collaborative Filtering算法。实验结果表明，基于Sapling Similarity的协同过滤在精度和其他指标上均优于其他模型。

此外，在数据库方面使用了MySQL，主要原因是用户基数大，遇到开发难题可以很快地寻找到问题解决方案。

第3章 系统需求分析

3.1 可行性分析

在系统需求分析中，可行性分析是非常重要的一环。

首先，从技术角度来看，该系统采用了成熟且广泛应用于实际项目的开发技术，如HTML、CSS、JavaScript、Vue.js、axios、Element UI、vue-router、Spring Boot、Mybatis Plus、Lombok、Hutool、java-jwt、Maven等。这些技术具有很高的稳定性和可靠性，并且经过了大量实践验证，能够满足系统的需求。此外，Sapling Similarity Collaborative Filtering算法作为推荐系统的核心技术，也在多个数据集上得到了成功的应用，证明了其在推荐效果上的优越性。因此，从技术角度来说，该系统是完全可行的。

其次，从经济角度来看，虽然系统开发过程中需要投入一定的人力和物力资源，但是考虑到系统上线后可以带来的经济效益（如提高销售额、增强用户粘性等），这种投入是值得的。另外，由于选择了开源的技术和工具，大大降低了软件授权费用。

最后，从操作角度来看，系统设计时充分考虑了用户的使用习惯和体验，界面简洁易用，功能布局合理，用户可以进行书籍搜索、浏览、购买和评价等操作。同时，系统的后台管理功能也非常完善，管理员可以方便地进行各种管理等工作。

综上所述，基于推荐算法的网上图书销售系统在技术、经济和操作三个方面都具备可行性，值得进行进一步的设计和开发工作。

3.2 系统业务需求分析

在基于推荐算法的网上图书销售系统的需求分析中，业务需求是至关重要的一个环节。以下是系统的业务流程和功能需求：

用户注册与登录：新用户注册，同时系统验证有效性，并根据用户类型（如普通用户或管理员）跳转至相应的页面。

书籍搜索与浏览：用户可以使用关键词进行书籍搜索，系统应提供多种筛选条件以帮助用户快速找到所需书籍。同时，系统也应提供分类浏览功能，以便用户按照类别查看书籍。

书籍购买与评价：用户可以在系统中选择心仪书籍进行购买，系统需支持购物车功能以及多种支付方式。用户购买后还可以对书籍进行评价，为其他用户提供参考。

个性化推荐：系统集成树苗相似度（SSCF）推荐算法，根据用户的购买图书的历史，向用户推荐可能感兴趣的书籍。推荐结果应在首页或其他合适位置展示。

后台管理：管理员可以对书籍进行管理。例如，管理员可以加书、修改现有书籍的信息、处理用户的退款请求等。

数据统计与分析：系统应提供数据统计与分析功能，以便管理员了解销售情况、用户行为等信息。这些数据可以帮助管理员优化库存、调整营销策略等。

以上是基于推荐算法的网上图书销售系统的业务需求分析，涵盖了从用户注册到后台管理的各个环节，旨在提供全面、便捷的购书体验，同时满足管理员的运营管理需求。

3.3 系统功能划分

在基于推荐算法的网上图书销售系统的需求分析中，系统的功能划分是关键。以下是系统的功能模块划分：

用户前端页面：面向普通用户的前端页面主要包括书籍搜索、浏览、购买、评价等操作。用户可以使用关键词进行书籍搜索，并通过多种筛选条件找到所需书籍，用户添加到购物车并进行购买，对购买后的书籍进行评价。

管理页面：面向管理员的管理页面，可以处理订单状态，查看用户反馈等。同时，系统应提供数据统计与分析功能，帮助管理员了解销售情况、用户行为等信息。

用户注册与登录：其中如果忘记密码，可通过输入手机号找回

个性化推荐：系统应集成推荐算法。推荐结果应在首页展示。

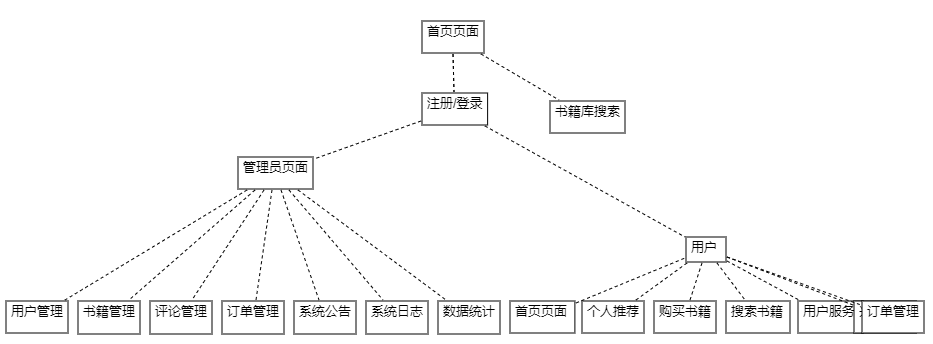
数据分析：对数据进行分析等，以帮助管理员更好地制定销售策略。

图3 系统功能划分图

以上是基于推荐算法的网上图书销售系统的功能模块划分。

1. 用户前端页面如下图4所示：

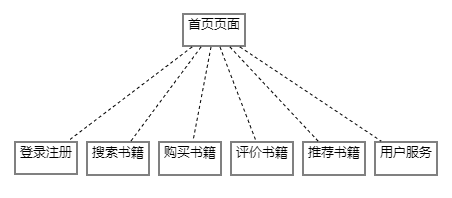


图4 前端页面功能模块图

（2）管理员页面如下图5所示：

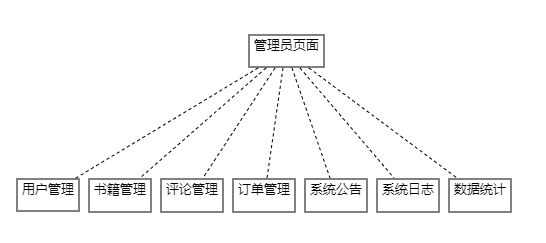


图5 后台管理功能模块图

3.4 前端页面功能模块

（1）用户欢迎页模块

用户首次默认访问欢迎页，可以在欢迎页进行书籍库的搜索。

模块图如下图6所示：

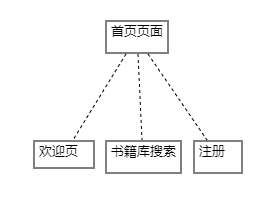


图6 用户欢迎页功能模块图

（2）图书商店的功能模块

用户注册登录后进入主页，购买和搜索，查看书籍推荐。

模块图如下图7所示：

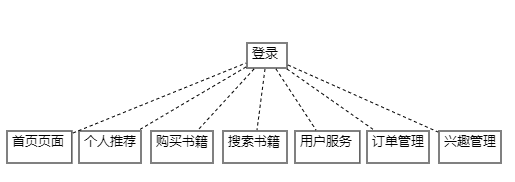


图7 图书商店的功能功能模块图

3.5 管理员管理功能模块

（1）用户的管理

管理员对普通用户进行相应的管理。

模块图如下图8所示：

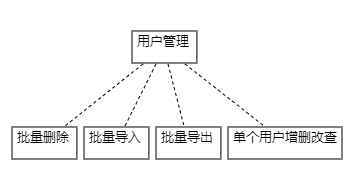


图8 用户的管理模块图

（2）订单管理

管理员可以订单进行管理的相关操作。

模块图如下9所示。

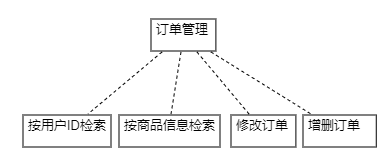


图9 订单管理模块图

（2）书籍管理

管理员可以对书籍进行管理的相关操作。

模块图如下10所示。

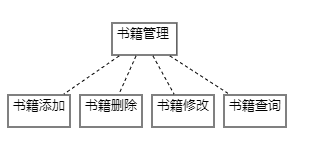


图10 书籍管理模块图

（2）评论管理

管理员可以对评论进行管理的相关操作。

模块图如下11所示。

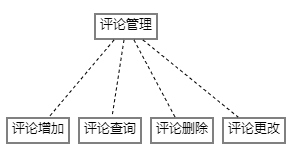


图11 评论管理模块图

（2）公告管理

管理员可以对公告进行管理的相关操作。

模块图如下12所示。

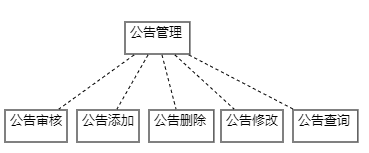


图12 公告管理模块图

3.6 系统UML用例图

以上所有的功能模块，相对应的用例图如下：

（1）订单管理模块用例图

描述：可以对订单进行操作，修改订单信息，但对于订单支付宝号等由第三方提供的管理员无法修改。

具体的UML用例图如下图13所示：

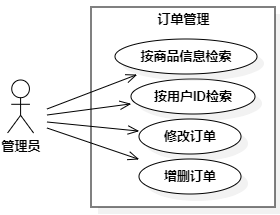


图13 订单管理模块UML用例图

（2）用户管理模块用例图

描述：用户信息操作。

具体的UML用例图如下图14所示：

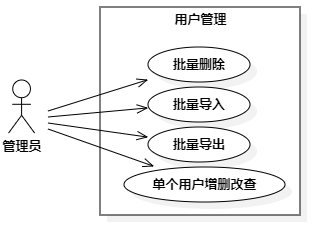


图14 用户管理模块用例图

（3）书籍管理模块用例图

描述：管理员对书籍进行基本管理。

具体的UML用例图如下图15所示：

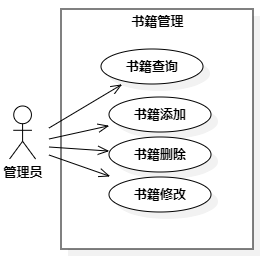


图15 书籍管理模块用例图

（4）公告模块用例图

描述：普通用户可以看到已开放公告的基本情况。管理员可以决定公告的开放。

具体的UML用例图如下图16所示：

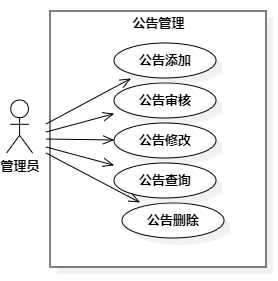


图16 公告管理模块实例图

（5）评论管理模块用例图

描述：管理员对评论进行管理。

具体的UML用例图如下图17所示：

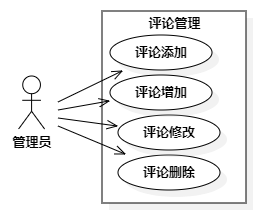


图17 评论管理模块实例图

3.7 本章小结

综上所述，本章节的系统需求分析为基于推荐算法的图书销售系统的开发提供了清晰的方向和框架。通过对用户需求的深入理解，为构建一个高效、易用的图书销售系统奠定了坚实的基础。接下来的章节将详细介绍系统的具体设计和实现过程。

第4章 系统设计

4.1 系统设计思想

在设计这个基于推荐算法的在线图书销售系统时，需要考虑以下几个关键因素：

用户体验：首先需要确保用户能够方便地找到他们想要购买或阅读的书籍。首页页面和搜索功能可以满足这一需求。

数据收集与分析：用户购买记录数据收集，找出用户会喜欢的物品。

推荐引擎：根据上述的数据分析结果，使用树苗相似度推荐算法（SSCF）进行计算，把推荐结果返回给前端。

管理员界面：管理员应能轻松管理网站的内容，如添加新书目、编辑评论以及处理订单等任务。同时，也需要有统计工具帮助管理者了解网站的整体运营情况。

移动端优化：考虑到越来越多的人选择用手机购物，所以平台必须支持移动设备并具有良好的用户体验。

综上所述，该系统的设计思路应该是围绕着“用户体验”为核心展开，结合先进的推荐算法和技术，打造一款既实用的在线图书销售平台。

具体如下图所示：

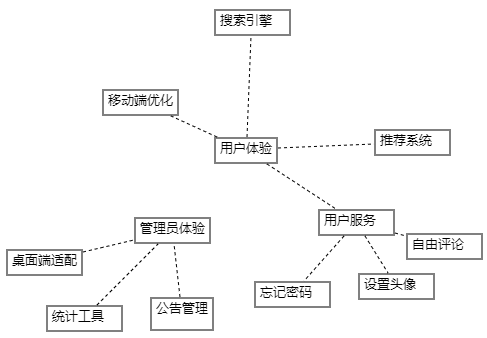


图18 系统设计思想图

4.2 推荐算法模块设计

（1）数据预处理：获取用户-图书交互数据，形成用户-图书二分网络。将数据转换为 unary（无权重）链接的形式，即如果用户购买了某本书，或者评分高于3，则用户-图书矩阵中的相应元素为1，否则为0。

（2）计算Sapling Similarity：对于项目基于的相似度计算，对每一对图书，使用Sapling Similarity方法计算它们的相似度。这涉及到分析一个图书被连接到另一个图书如何影响对第三个图书也被连接到第一个图书的概率的估计。对于用户基于的相似度计算，对每一对用户，同样使用Sapling Similarity方法计算他们的相似度，考虑一个用户连接到一本书如何影响对另一个用户也连接到同一本书的概率的估计。基于用户的设计思路如图19所示：

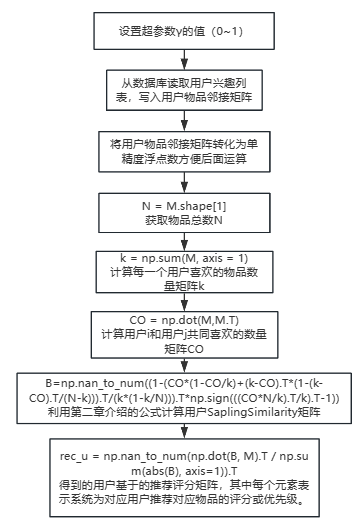


图19 基于用户推荐模块流程图

（3）构建SSCF（Sapling Similarity Collaborative Filtering）：结合用户基于和项目基于的Sapling Similarity结果，设计一个混合推荐策略。对于每个用户，找到与其最相似（无论正负相似度）的N个其他用户和M本书。根据这些相似用户和相似书籍的购买历史，计算并排序未被该用户购买过的书籍的推荐分数。

（4）生成推荐列表：对每个用户，根据其SSCF推荐分数，生成一个包含Top-K本书籍的推荐列表。

（4）反馈和优化： 用户对推荐结果进行反馈（如评论等行为），将这些反馈数据纳入系统以更新用户-图书二分网络和相应的Sapling Similarity计算定期评估和调整推荐算法的性能，如调整混合推荐策略中用户基于和项目基于的权重。

通过以上设计，基于Sapling Similarity的图书推荐算法能够充分利用Sapling Similarity的优势，提供具有更高精度和可解释性的图书推荐，同时考虑到用户和图书之间的正负相似性或反关联性。

4.3 数据库设计

数据库是在线销售系统的核心，对于图书推荐算法驱动的在线销售平台来说，有效地处理数据请求和响应是至关重要的。为了实现这一目标，整合了程序设计开发与先进的数据库技术，致力于构建一个适合各类用户的图书在线销售系统。该系统的设计旨在为图书销售者和购买者提供便利，通过精准的推荐算法最大化图书的销售潜力，同时充分利用信息化手段，有效解决传统图书销售模式中存在的问题[3]。

在设计图书在线销售系统的数据库时，通常会遵循以下几个阶段：首先，需求分析阶段，包括用户行为数据、图书信息以及推荐算法所需的各类参数等。其次，概念设计阶段，会构建一个抽象的、独立于具体实现细节的数据模型，定义数据实体和它们之间的关系。接下来是逻辑设计阶段，在此阶段，将细化数据模型，设计数据表结构、关联规则以及存储过程等，以支持推荐算法的运行和数据查询的效率。

最后，物理设计阶段，包括数据存储策略、索引优化、数据分区以及性能调优等实践问题。通过这些阶段的精心规划和实施，图书在线销售系统能够实现高效的数据库操作，提供准确的个性化推荐，从而提升用户满意度和销售业绩。

4.4 数据库实体图

数据库实体图（Entity Relationship Diagram，简称ERD）在数据库设计和管理中具有极其重要的作用。

（1）用户实体图

具体如图20示：

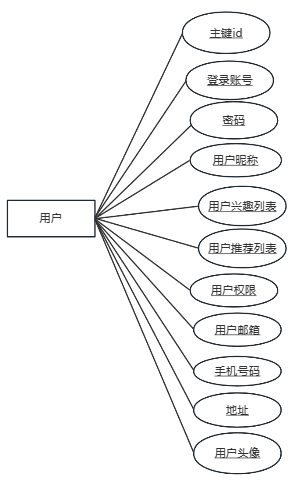


图20 用户实体图

（2）日志管理实体图

具体如图21示：

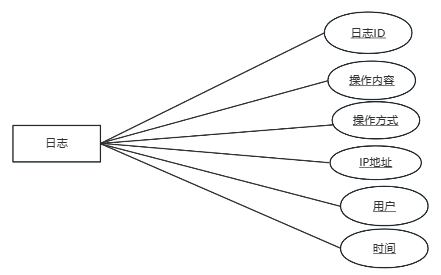


图21 日志管理实体图

（3）公告管理实体图

如具体图22示：

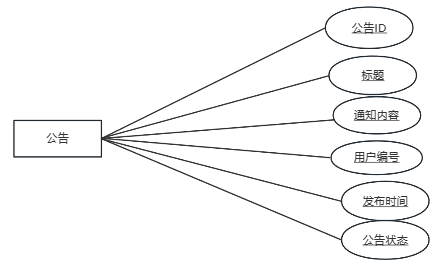


图22 公告管理实体图

（4）书籍实体图

如具体图23示：

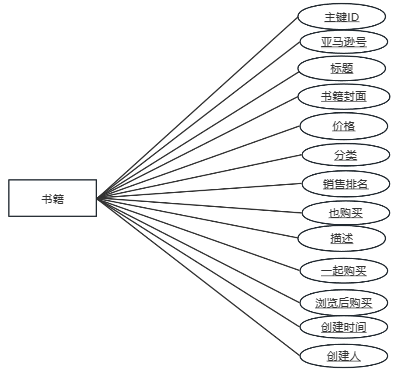


图23 书籍实体图

（5）评论实体图

如具体图24示：

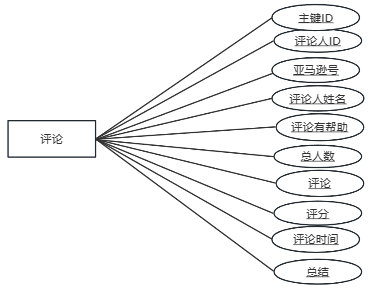


图24 评论实体图

（6）订单实体图

如具体图25示：

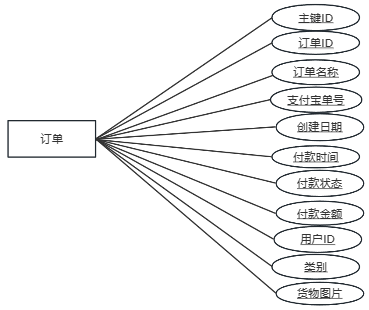


图25 订单实体图

4.5 系统E-R图

系统E-R图是一种可视化数据模型工具，它助力设计者和开发者深入理解数据模型及其之间的关联。通过这种方式，设计师能够标准化数据库模型，明确实体之间的联系，优化查询性能，并确保数据库的稳定性和完备性。此外，E-R图还能与实际业务需求相对应，这使得开发人员和业务人员能更有效地理解和对应系统的功能需求。简而言之，系统E-R图是连接数据结构和业务需求的一种强大桥梁，提升了系统设计和开发的效率与准确性。

E-R图如下图26所示：

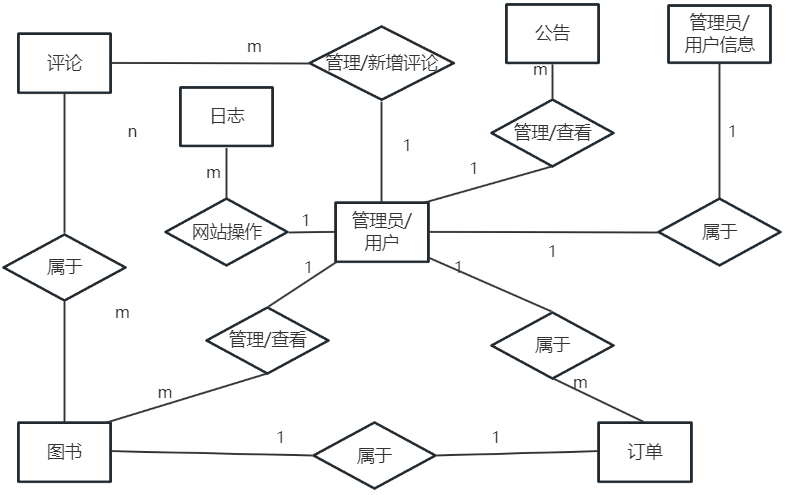


图26 系统E-R图

4.6 数据库表

在本系统中，数据表的设定是基于实际业务需求和流程进行的。选择了自增长整数类型来设定基本表。优秀的数据库设计需要全面考量多种因素：首先，数据表的配置必须精准对应业务需求；其次，字段设计应详尽且精炼，既要包含所有必要的信息，又要避免冗余；再者，主键设计应确保每个记录的唯一性，便于数据的查找和管理；接着，索引设计是为了加快数据检索的速度，提升系统响应效率；最后，数据库性能优化是持续的过程，旨在提高数据库的运行速度和稳定性。总的来说，一个高质量的数据库设计旨在提升数据库的性能和可靠性，减少信息处理的时间，从而大幅提升系统的整体运作效率。这需要在数据表设置、字段设计、主键设计、索引设计以及数据库性能优化等多个层面做出深思熟虑和精细调整。

（1）用户表（user）

表1 系统用户表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 长度 | 是否主键 | 是否可空 | 说明 |
| id | int | 11 | 是 | 否 | 主键id |
| username | varchar | 50 | 是 | 是 | 登录账号 |
| name | varchar | 50 | 否 | 是 | 用户昵称 |
| role | varchar | 30 | 否 | 是 | 用户权限 |
| email | varchar | 80 | 否 | 是 | 用户邮箱 |
| phone | varchar | 11 | 否 | 是 | 手机号码 |
| password | varchar | 50 | 否 | 是 | 密码 |
| address | varchar | 255 | 否 | 是 | 地址 |
| avatar | varchar | 500 | 否 | 是 | 用户头像 |
| recommendation | text |  | 否 | 是 | 用户推荐列表 |
| Interests | text |  | 否 | 是 | 用户兴趣列表 |

（2）日志（logs）

表2 日志表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 长度 | 是否主键 | 是否可空 | 说明 |
| id | int | 20 | 是 | 否 | 日志ID |
| operation | varchar | 30 | 否 | 是 | 操作内容 |
| type | varchar | 20 | 否 | 是 | 操作方式 |
| ip | varchar | 50 | 否 | 是 | IP地址 |
| user | varchar | 50 | 否 | 是 | 用户 |
| time | varchar | 50 | 否 | 是 | 时间 |

（3）公告表（notice）

表3 公告表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 长度 | 是否主键 | 是否可空 | 说明 |
| id | int | 8 | 是 | 否 | 公告ID |
| title | varchar | 50 | 否 | 是 | 标题 |
| content | text |  | 否 | 是 | 通知内容 |
| userid | int | 8 | 否 | 是 | 用户编号 |
| time | varchar | 25 | 否 | 是 | 发布时间 |
| open | bit | 3 | 否 | 是 | 公告状态 |

（4）书籍表（books）

表4书籍信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 长度 | 是否主键 | 是否可空 | 说明 |
| id | int | 11 | 是 | 否 | 主键ID |
| asin | varchar | 30 | 是 | 否 | 亚马逊号 |
| title | varchar | 100 | 是 | 否 | 标题 |
| imUrl | int | 4 | 是 | 否 | 书籍封面 |
| price | char | 1 | 否 | 是 | 价格 |
| categories | varchar | 64 | 否 | 是 | 分类 |
| description | date |  | 否 | 是 | 描述 |
| salesrank | char | 1 | 否 | 是 | 销售排名 |
| also\_bought | varchar | 500 | 否 | 是 | 也购买 |
| bought\_together | varchar | 64 | 否 | 是 | 一起购买 |
| buy\_after\_viewing | datetime |  | 否 | 是 | 浏览后购买 |
| authorid | varchar | 32 | 否 | 是 | 创建人 |
| time | varchar | 32 | 否 | 是 | 创建时间 |

（5）评论表（review）

表5评论信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 长度 | 是否主键 | 是否可空 | 说明 |
| id | bigint | 20 | 是 | 否 | 主键ID |
| reviewer\_id | bigint | 20 | 否 | 是 | 评论人ID |
| asin | varchar | 30 | 否 | 是 | 亚马逊号 |
| reviewer\_name | varchar | 30 | 否 | 是 | 评论人姓名 |
| helpful | varchar | 100 | 否 | 是 | 评论有帮助 |
| helpful\_all | int | 4 | 否 | 是 | 总人数 |
| review\_text | char | 1 | 否 | 是 | 评论 |
| overall | char | 1 | 否 | 是 | 评分 |
| summary | char | 1 | 否 | 是 | 总结 |
| review\_time | varchar | 64 | 否 | 是 | 评论时间 |

（6）订单信息表（order）

表6订单信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 长度 | 是否主键 | 是否可空 | 说明 |
| id | bigint | 20 | 是 | 否 | 主键ID |
| no | bigint | 20 | 否 | 是 | 订单ID |
| name | varchar | 30 | 否 | 是 | 订单名称 |
| alipay\_no | bigint | 20 | 否 | 是 | 支付宝单号 |
| date | varchar | 30 | 否 | 是 | 创建日期 |
| pay\_time | bigint | 20 | 否 | 是 | 付款时间 |
| state | varchar | 30 | 否 | 是 | 付款状态 |
| goods\_id | datetime |  | 否 | 是 | 物品ID |
| money | datetime |  | 否 | 是 | 付款金额 |
| userid | varchar | 64 | 否 | 是 | 用户ID |
| category | datetime |  | 否 | 是 | 类别 |
| goods\_imurl | varchar | 32 | 否 | 是 | 货物图片 |

4.7 本章小结

本章总结了系统的基本设计思路，详细阐述了功能模块和数据库的设计策略，旨在构建一个既稳定可靠、又易于维护和扩展的系统，以满足实际需求并适应未来的挑战。

第5章 系统的实现

在系统实现阶段，遵循模块化和分层设计的原则，将整个网上图书销售系统划分为前端实现、后端实现、支付宝接口接入、系统管理部分实现和推荐系统集成四个主要部分。

本系统通过阿里云的轻量级服务器部署，利用nginx进行托管，可以直接访问IP： **http://8.137.36.248/** 查看

5.1 前端实现

（1）Vue.js和Element UI的应用

Vue.js框架的核心特性是双向数据绑定，和组件化开发。可以利用Element UI进行组件化开发以提供用户友好的界面和丰富的交互功能。

首先，在全局安装Vue CLI。具体方法为在命令行使用npm包管理器安装。使用Vue CLI创建一个新的Vue.js项目。在命令行中，导航到希望存放项目的目录，然后运行以下命令：vue create project。创建项目后，Vue.js会生成一个标准的项目结构，主要文件：public/index.html：应用程序的入口HTML文件，包含了加载Vue应用的 <div id="app"> 标签。src/App.vue：主应用组件，通常包含其他组件和布局。src/main.js：应用程序的入口脚本，导入并启动Vue应用。 编写了很多vue组件页面，包括Login.vue用于登录注册，Manager.Vue展示用户主页，recommender.vue用于展示图书推荐列表等。Element UI是一款基于Vue.js的开源UI库，主要特点和优势有：高度可定制，组件丰富，易用性和文档完备，开发者友好等。使用的Element UI组件主要有表格、表单、按钮、提示信息等，使用了Element UI的动画和过渡效果，在展示图片时使用翻页，隐出等动态效果。

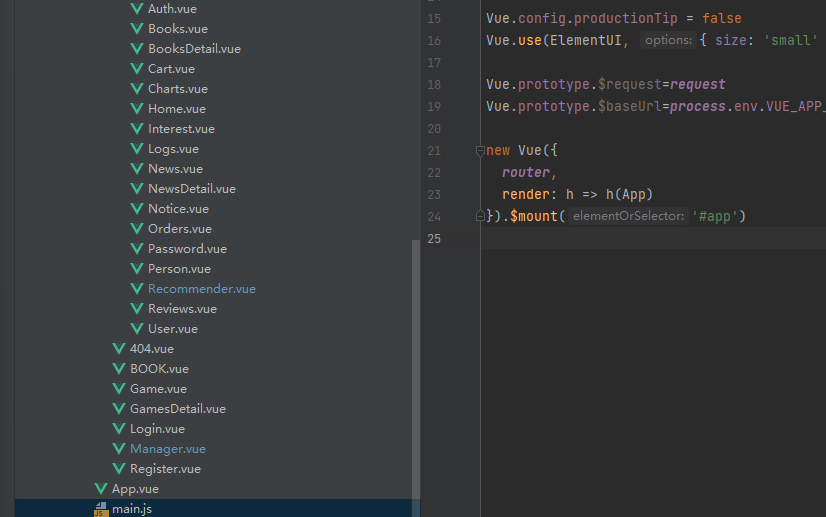


图27 主要vue文件

（2） Axios的使用  
 在Vue.js中使用Axios库进行前后端数据交互是一个常见且重要的任务。  
 在Vue项目中，打开命令行工具并运行以下命令来安装Axios：npm install axios。在打算使用Axios的Vue组件中，导入Axios库：import axios from 'axios';新建一个js文件用于配置请求。这里设置响应时间为60000毫秒，并统一设置后端接口地址。并在请求前和响应后，都先处理，如统一加token，对请求参数进行加密等。

|  |
| --- |
| const request = axios.create({  baseURL: process.env.VUE\_APP\_BASEURL,  timeout: 60000})  request.interceptors.request.use(config => {  config.headers['Content-Type'] = 'application/json;charset=utf-8';  let user = JSON.parse(localStorage.getItem("honey-user") || '{}')  config.headers['token'] = user.token  return config}, error => {  console.error('request error: ' + error)  return Promise.reject(error)});  request.interceptors.response.use( response => {  let res = response.data;  if (typeof res === 'string') {  res = res ? JSON.parse(res) : res}  if (res.code === '401') {  router.push('/login') }  return res;},error => {  console.error('response error: ' + error) // for debug  return Promise.reject(error)})  export default request |

（3）Vue-Router的管理

Vue Router是在页面路由和全局状态管理中扮演着关键角色。页面路由：在单页应用（SPA）中，Vue Router负责管理应用程序的多个视图或页面之间的导航。以下是在Vue Router中实现页面路由的主要步骤和功能。  
 安装和配置：首先，需要通过npm或yarn将Vue Router安装到项目中，并在Vue应用中进行配置。配置主要包括定义路由规则、设置路由组件以及挂载到Vue实例上。通过routes数组定义路由规则，每个路由规则包含一个路径（path）和一个对应的组件（component）。当用户在浏览器中导航到指定路径时，Vue Router会匹配相应的路由规则并渲染相应的组件。新建一个js文件保存路由配置如下：

|  |
| --- |
| { path: '403', name: 'Auth', meta: { name: '无权限' }, component: () => import('../views/manager/Auth') },  { path: 'home', name: 'Home', meta: { name: '系统首页' }, component: () => import('../views/manager/Home') },  { path: 'user', name: 'User', meta: { name: '用户信息' }, component: () => import('../views/manager/User') },  { path: 'person', name: 'Person', meta: { name: '个人信息' }, component: () => import('../views/manager/Person') },  { path: 'password', name: 'Password', meta: { name: '修改密码' }, component: () => import('../views/manager/Password') },  { path: 'news', name: 'News', meta: { name: '新闻信息' }, component: () => import('../views/manager/News') },  { path: 'newsDetail', name: 'NewsDetail', meta: { name: '新闻详情' }, component: () => import('../views/manager/NewsDetail') },  { path: 'booksDetail', name: 'BooksDetail', meta: { name: '书籍详情' }, component: () => import('../views/manager/BooksDetail') },  { path: 'notice', name: 'Notice', meta: { name: '系统公告' }, component: () => import('../views/manager/Notice') },  { path: 'logs', name: 'Logs', meta: { name: '系统日志' }, component: () => import('../views/manager/Logs') },  { path: 'charts', name: 'Charts', meta: { name: '数据统计' }, component: () => import('../views/manager/Charts') },  { path: 'orders', name: 'Orders', meta: { name: '订单管理' }, component: () => import('../views/manager/Orders') },  { path: 'books', name: 'Books', meta: { name: '书籍管理' }, component: () => import('../views/manager/Books') },  { path: 'recommender', name: 'Recommender', meta: { name: '推荐' }, component: () => import('../views/manager/Recommender') },  { path: 'reviews', name: 'Reviews', meta: { name: '评论' }, component: () => import('../views/manager/Reviews') },  { path: 'interest', name: 'Interest', meta: { name: '感兴趣' }, component: () => import('../views/manager/Interest') },  { path: 'cart', name: 'Cart', meta: { name: '购物车' }, component: () => import('../views/manager/Cart') },]}, |

（4） Echarts的数据展示

没有安装Echarts，可以通过npm在Vue项目中安装，在需要使用Echarts的Vue组件中，通过import语句引入Echarts库。在Vue组件的模板部分，创建一个具有固定宽度和高度的DOM元素作为Echarts的容器。在Vue组件的mounted生命周期钩子中，可以配置Echarts的图表类型、数据、样式等选项。

具体的功能页面如下图28所示。



图28 Echarts界面

代码实现：

|  |
| --- |
| mounted() {  let linetDom = document.getElementById('line');  let lineChart = echarts.init(linetDom);  let barDom = document.getElementById('bar');  let barChart = echarts.init(barDom);  barChart.setOption(option1)  let pieDom = document.getElementById('pie');  let pieChart = echarts.init(pieDom);  pieChart.setOption(option2)  this.$request.get('/charts').then(res => {  option.xAxis.data = res.data?.line?.map(v => v.date) || []  option.series[0].data = res.data?.line?.map(v => v.value) || []  lineChart.setOption(option)  option1.xAxis.data = res.data?.bar?.map(v => v.name) || []  option1.series[0].data = res.data?.bar?.map(v => v.value) || []  barChart.setOption(option1)  option2.series[0].data = res.data?.bar || []  pieChart.setOption(option2)  }) }, |

5.2 后端实现

（1） SpringBoot框架的选择和配置

为了编程过程中遇到的错误能够解决是用了2.0版本，更稳定，且容易排错。Spring Boot项目通常使用Maven或Gradle作为构建工具。Maven是Java项目的依赖管理工具，使用Spring Initializr（[https://start.spring.io/）可以快速生成一个新的Spring](https://start.spring.io/%EF%BC%89%E5%8F%AF%E4%BB%A5%E5%BF%AB%E9%80%9F%E7%94%9F%E6%88%90%E4%B8%80%E4%B8%AA%E6%96%B0%E7%9A%84Spring) Boot项目结构，包括基本的Maven或Gradle配置、所需的依赖和一个简单的启动类。在pom.xml或build.gradle文件中，可以添加所需的Spring Boot Starter依赖。最后，创建控制器、服务、数据访问层等组件，并在主启动类中使用@SpringBootApplication注解来标记这是一个Spring Boot应用。然后，通过运行主类中的main方法来启动应用。

配置如下所示。

|  |
| --- |
| server:port: 9090  spring:datasource:driver-class-name: com.mysql.cj.jdbc.Driver  url:jdbc:mysql://${ip}:3306/honey2024?useSSL=false&useUnicode=true&characterEncoding=UTF-8&serverTimezone=GMT%2b8  username: root  password: root  servlet:multipart:  max-file-size: 20MB  max-request-size: 20MB  ip: localhost |

（2） MyBatisPlus和Lombok的使用

MyBatisPlus和Lombok是两个在Java开发中常用的工具，它们分别在数据访问层和对象模型层提供便利。Lombok是一个Java库，它可以自动为类生成常见的getter/setter、构造函数、equals/hashCode/toString方法等。通过在类或字段上添加注解，Lombok可以在编译时自动生成所需的代码。在使用MyBatisPlus和Lombok时，通常的做法是：在实体类（Entity）中使用Lombok注解来减少样板代码，如@Data、@NoArgsConstructor和@AllArgsConstructor等。使用MyBatisPlus提供的各种API进行数据库操作，如插入、删除、更新和查询等，而无需手动编写SQL语句。如果需要进行复杂的查询，可以结合MyBatis的动态SQL特性以及Lombok的实体类来进行。

（3） Hutool工具包的引入  
 项目的pom.xml文件中，找到或添加 <dependencies> 标签，并在其中添加Hutool的依赖。在命令行中进入项目目录，然后运行以下命令来更新依赖：mvn clean install。一旦Hutool依赖被正确地添加和更新到项目中，就可以在Java代码中导入并使用Hutool提供的各种工具类和方法了。

（4） Java-JWT认证

当用户成功登录后，服务器会将用户的相关信息（如用户ID、角色、过期时间等）放入JWT的Payload部分。服务器将生成的JWT返回给客户端，客户端通常将其存储在本地（如浏览器的LocalStorage或Cookie中）。在后续的请求中，客户端需要在HTTP请求头（通常是Authorization header）中携带JWT。常见的格式是 Bearer <JWT>。服务器接收到请求后，从Authorization header中提取出JWT，并对其进行验证。验证过程包括检查Signature是否正确、Payload中的信息是否有效（如过期时间未到、Issuer与预期一致等）。可以使用诸如jjwt、Auth0的JavaJWT和Nimbus JOSE+JWT等第三方库来简化JWT的生成、解析和验证过程。

关键代码实现：

|  |
| --- |
| public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object handler) {  String token = request.getHeader("token");  if (StrUtil.isBlank(token)) {  token = request.getParameter("token");}  if (handler instanceof HandlerMethod) {AuthAccess annotation = ((HandlerMethod) handler).getMethodAnnotation(AuthAccess.class);  if (annotation != null) {  return true; }}  if (StrUtil.isBlank(token)) {  throw new ServiceException("401", "error1");}  String userId;  try {userId = JWT.decode(token).getAudience().get(0); }catch (JWTDecodeException j) {  throw new ServiceException("401", "error2");}  User user = userMapper.selectById(Integer.valueOf(userId));  if (user == null) {  throw new ServiceException("401", "error3"); }  JWTVerifier jwtVerifier = JWT.require(Algorithm.HMAC256(user.getPassword())).build();  try {jwtVerifier.verify(token);} catch (JWTVerificationException e) {  throw new ServiceException("401", "error4");}  return true;} |

5.3 支付宝沙箱接口接入

接入支付宝沙箱接口是为了在开发和测试阶段模拟真实的支付环境，以下是接入步骤。

在支付宝开放平台中，创建一个新的应用，并选择相应的服务类型，如“电脑网站支付”。创建应用后，会获得一个appid，这是应用在支付宝中的唯一标识。同时，需要生成一对RSA密钥，其中公钥上传到支付宝，私钥自己保存，用于签名交易数据。在代码中，根据支付宝提供的API文档配置接口参数，包括appid、商户私钥、请求URL（通常是沙箱环境的URL）、业务参数（如订单金额、商品描述等）。之后，需要在服务器端编写代码来接收并处理支付结果通知，验证签名并更新订单状态。



图29 支付宝页面

关键代码实现：

|  |
| --- |
| public void pay(@RequestParam String id, HttpServletResponse Response) throws AlipayApiException, IOException {  Orders order = ordersService.getById(id);  if (order == null){ return ;}  AlipayClient alipayClient = new DefaultAlipayClient(gatewayUrl, "9021000132691178", aliPayConfig.getPrivateKey(), format, charset, aliPayConfig.getPublicKey(), signType);  AlipayTradePagePayRequest request = new AlipayTradePagePayRequest();  request.setNotifyUrl(aliPayConfig.getNotifyUrl());  JSONObject bizContent = new JSONObject();  bizContent.set("out\_trade\_no", order.getNo());  bizContent.set("total\_amount", order.getMoney());  bizContent.set("subject", order.getName());  bizContent.set("product\_code", "FAST\_INSTANT\_TRADE\_PAY");  request.setBizContent(bizContent.toString());  request.setReturnUrl(aliPayConfig.getReturnUrl());  AlipayTradePagePayResponse response = alipayClient.pageExecute(request);  String form = "";  try{form = response.getBody();} catch (Exception e){ e.printStackTrace(); }  Response.setContentType("text/html;charset=" + charset);  Response.getWriter().write(form);  Response.getWriter().flush();  Response.getWriter().close();} |

5.4 系统管理部分实现

管理员在管理界面进入后，根据实际需求转到相对应的模块，然后进行相应的变更管理，确保业务进行的有效完整。

具体的功能页面如下图30所示。

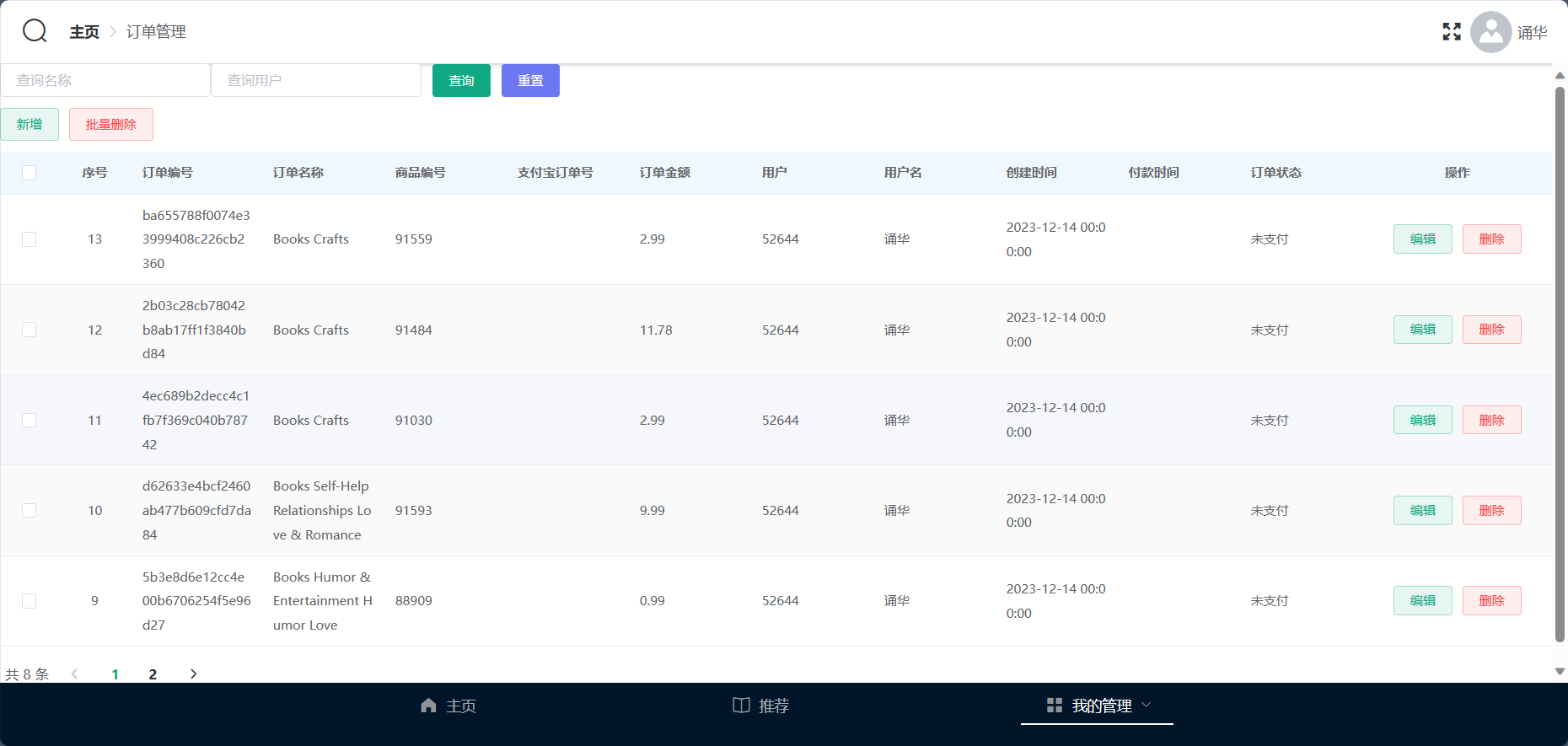


图30 系统管理页面

代码实现：

|  |
| --- |
| @HoneyLogs(operation = "订单", type = LogType.ADD)  @PostMapping("/add")  public Result add(@RequestBody Orders orders) {  User currentUser = TokenUtils.getCurrentUser();  orders.setUserid(currentUser.getId());  orders.setDate(DateUtil.today()); // 2023-10-08  orders.setNo(IdUtil.fastSimpleUUID());  ordersService.save(orders);  return Result.success();}  @HoneyLogs(operation = "订单", type = LogType.UPDATE)  @PutMapping("/update")  public Result update(@RequestBody Orders orders) {  ordersService.updateById(orders);  return Result.success();}  @HoneyLogs(operation = "订单", type = LogType.DELETE)  @DeleteMapping("/delete/{id}")  public Result delete(@PathVariable Integer id) {  ordersService.removeById(id);  return Result.success();}  @HoneyLogs(operation = "订单", type = LogType.BATCH\_DELETE)  @DeleteMapping("/delete/batch")  public Result batchDelete(@RequestBody List<Integer> ids) {  ordersService.removeBatchByIds(ids);  return Result.success();} |

5.5 推荐系统集成

在测试集上评估推荐系统的性能。评估指标NDCG（Normalized Discounted Cumulative Gain）具体如下图31所示。

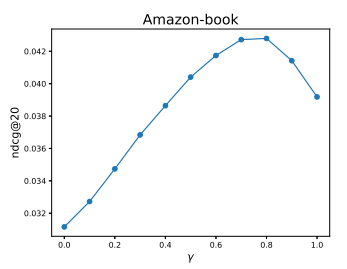


图31 在Amazon-book数据上的最佳γ值

值得注意的是，SSCF是一个基于记忆的协同过滤方法，它不需要训练阶段，且其唯一的超参数并未在测试集上进行优化。这意味着SSCF在计算效率和解释性方面可能具有优势，但在实际应用中，其性能可能会受到不进行超参数优化的影响。

尽管如此，在AmazonBook数据集上，SSCF的表现优于所有现有的模型，这是一个显著的结果，特别是考虑到它与经过优化的机器学习方法进行了比较。这表明SSCF在某些情况下可能提供了一种有效且易于解释的推荐系统构建方法。

代码实现：

|  |
| --- |
| t1 = time.time()  args = parse\_args()  g = args.gamma  data = args.dataset  model = args.similarity  test\_data = args.test  if test\_data == "validation":  N\_users, N\_items, M, train, test, = utils.read\_data\_validation(data)  else:  N\_users, N\_items, M, train, test = utils.read\_data(data)  M = M.astype(np.float32)  B = sml.similarity(M, model, 0)  rec\_u = np.nan\_to\_num(np.dot(B, M).T / np.sum(abs(B), axis=1)).T  B = sml.similarity(M, model, 1)  rec\_i = np.nan\_to\_num(np.dot(M, B) / np.sum(abs(B), axis=0))  rec = (1 - g) \* rec\_u + g \* rec\_i  scores\_user = utils.scores(train, test, rec\_u, N\_users, N\_items, K=20)  scores\_item = utils.scores(train, test, rec\_i, N\_users, N\_items, K=20)  scores = utils.scores(train, test, rec, N\_users, N\_items, K=20)  print("\nRESULTS\n")  print("user-based")  print("precision@20:\t{}\nrecall@20:\t{}\nndcg@20:\t{}".format(scores\_user[0], scores\_user[1], scores\_user[2]))  print("item-based")  print("precision@20:\t{}\nrecall@20:\t{}\nndcg@20:\t{}".format(scores\_item[0], scores\_item[1], scores\_item[2]))  print("final model")  print("precision@20:\t{}\nrecall@20:\t{}\nndcg@20:\t{}".format(scores[0], scores[1], scores[2]))  t2 = time.time()  print(t2 - t1, "seconds") |

5.6 本章小结

本章完成了具体的系统设计，展示了具体的浏览器展示页面，结合实际来具体分析，对核心的代码块进行系统性的记录，验证各种子模块的功能实现，确定了后续的测试环节。

 第6章 系统测试

6.1 测试的目的

通过系统测试，可以确保系统能够在实际环境中满足用户的需求和期望。同时，系统测试也是对软件开发过程的一种验证，有助于识别开发过程中可能存在的问题和改进点。

6.2 测试环境

表7 测试环境表

|  |  |  |  |
| --- | --- | --- | --- |
|  | 客户端服务器 | 数据库服务器 | 应用服务器 |
| 硬件配置 | CPU: AMD | CPU: 2核  内存：4.00GB  硬盘：40GB | CPU: 2核  内存：4.00GB  硬盘：40GB |
| 软件配置 | Google Chrome：120.0.6099.109 | mySql: 5.7.42 | Jdk: 8  Nginx: 1.24.0 |

6.3 测试方法

以下为主要的测试方法：

功能测试：这包括但不限于图书搜索功能、用户账户管理、购物车功能、订单处理流程以及核心的推荐功能。

性能测试：性能测试确保其在高流量和复杂操作场景下的稳定性和响应能力。具体指标可能包括页面加载速度、搜索响应时间、推荐算法的计算效率、系统吞吐量（如每秒处理的请求数量）、并发用户数下的系统稳定性等。这些测试有助于发现并优化可能存在的性能瓶颈，提高用户体验。

用户界面测试：界面的易用性、一致性、美观度以及兼容性都是重要的测试内容。测试人员需要验证界面布局是否合理，按钮和链接是否正常工作，页面间的导航是否流畅，同时还要考虑不同设备和浏览器的兼容性问题。

数据库测试：由于网上图书销售系统依赖于数据库存储大量的图书信息、用户数据和交易记录，因此对数据库的测试至关重要。测试内容包括数据的正确性、完整性、一致性以及安全性。例如，检查数据是否能够正确插入、更新和删除，查询结果是否准确，是否存在数据冗余或冲突，以及数据加密和备份恢复功能是否有效。

推荐准确性与多样性测试：对于推荐系统的核心功能，需要专门进行推荐准确性与多样性测试。这包括评估推荐结果的相关性、新颖性和覆盖率，以及对比不同推荐算法的效果。可以通过A/B测试、离线评估和在线反馈等方式收集数据，分析推荐结果的质量和用户满意度。

兼容性测试：测试是否可以在不同设备上使用。

通过以上全面的测试方法，可以有效地发现并修复基于推荐算法的网上图书销售系统中的缺陷和问题，确保系统的稳定运行和高质量用户体验，从而提升销售效率和用户满意度。

6.4 测试用例

（1）登录注册页面测试

表8 登录注册页面测试表

|  |  |  |  |
| --- | --- | --- | --- |
| 用例名称 | 登录注册页面 | 测试时间 | 2023-12-14 |
| 对应需求 | 登录注册页面 | 测试人员 | 宋志华 |
| 功能描述 | 验证登录注册，以及找回密码 | | |
| 测试目的 | 功能测试 | | |
| 前置条件 | 数据库中存在用户（账号：admin 密码：admin） | | |
| 测试步骤 | 登录：1．在登录界面中输入账号和密码，以及验证码  2．单击“登录”按钮  注册：1.在注册页面输入账号和密码注册 2. 单击“注册”按钮  找回密码：输入账号在找回密码页面填写用户名和手机号，点击找回 | | |
| 序号 | 输入数据/动作 | 预期结果 | 实际结果 |
| 1 | 输入用户名和密码登录 | 通过服务器数据库验证 | 和预期结果一致 |
| 2 | 输入非法用户登录 | 提示用户不存在/密码错误 | 和预期结果一致 |
| 3 | 输入用户名和密码注册 | 注册成功 | 和预期结果一致 |
| 4 | 找回密码 | 通过服务器数据库验证 | 和预期结果一致 |

（2）界面功能测试

表9 界面功能测试表

|  |  |  |  |
| --- | --- | --- | --- |
| 用例名称 | 界面功能 | 测试时间 | 2023-12-14 |
| 对应需求 | 信息管理 | 测试人员 | 宋志华 |
| 功能描述 | 浏览页面，验证图书管理，用户管理，订单管理，评论管理，推荐系统 | | |
| 测试目的 | 页面功能完善可行 | | |
| 前置条件 | 可以使用管理员账号登录（账号：admin 密码：admin） | | |
| 测试步骤 | 1．登录系统  2．单击“登录”按钮  3.浏览与信息管理 | | |
| 序号 | 输入数据/动作 | 预期结果 | 实际结果 |
| 1 | 用户增删改查 | 成功 | 和预期结果一致 |
| 2 | 图书增删改查 | 成功 | 和预期结果一致 |
| 3 | 订单增删改查 | 成功 | 和预期结果一致 |
| 4 | 评论增删改查 | 成功 | 和预期结果一致 |
| 5 | 推荐系统显示 | 成功 | 和预期结果一致 |

（3）用户购物车支付宝购买测试

用户可以在页面进行添加到购物车购买，系统会给出相应提示。只要使用沙盒账号支付，能正常支付。测试数据如下表11所示：

表10 管理员支付宝购买测试表

|  |  |  |  |
| --- | --- | --- | --- |
| 用例名称 | 支付宝付款 | 测试时间 | 2023-12-14 |
| 对应需求 | 购买图书 | 测试人员 | 宋志华 |
| 功能描述 | 对购物车图书进行购买 | | |
| 测试目的 | 图书购买是否成功 | | |
| 前置条件 | 数据库中存在用户（账号：user，密码：user） | | |
| 测试步骤 | 1．在登录界面中输入账号和密码  2．对购物车图书点击购买 | | |
| 序号 | 输入数据/动作 | 预期结果 | 实际结果 |
| 1 | 购买图书 | 去服务器请求支付宝接口 | 跳转支付，支付成功和预期结果一致 |
| 2 | 删除不想要订单 | 成功 | 和预期结果一致 |

6.5 测试结论

基于推荐算法的网上图书销售系统在设计与实现方面达到了预期的目标。系统功能齐全、性能稳定、安全性高、兼容性强，并且推荐算法在提高用户体验和个性化推荐方面表现出色。通过本次测试，确认该系统具备上线运营的条件，并建议在实际应用中持续收集用户反馈和进行迭代优化，以不断提升系统质量和用户体验。

6.6 本章小结

本章主要介绍了系统测试的全过程，例如推荐模块、购买模块、还有用户最基本的登录注册等等。

第7章 总结

本论文详细探讨了基于推荐算法的网上图书销售系统的设计与实现。以下是对整个研究工作的总结：

技术选型与框架集成：选择了前端Vue.js技术，结合Axios、Element UI和Vue Router等工具进行开发，以提供用户友好的界面和高效的响应。后端采用Spring Boot框架，搭配Lombok、Hutool、Mybatis Plus和Java-JWT等工具，构建了一个稳定、高效的后台服务。推荐算法集成与优化：集成了Sapling Similarity Collaborative Filtering推荐算法，并在AmazonBook数据集上进行了测试和优化。该算法基于决策树和记忆机制，能够有效挖掘用户兴趣，提供个性化的图书推荐服务。测试结果表明，该算法在多个数据集上的性能优于其他模型，实现了state of the art的推荐效果。系统提供了便捷的图书搜索、浏览、购买和评价等功能，满足用户购书需求。利用Echarts对销售数据进行可视化展示，帮助管理者进行数据分析和决策。在线支付接口集成：系统成功接入了支付宝沙盒环境，实现了安全、便捷的在线支付功能，增强了系统的实用性和竞争力。系统测试与评估：测试结果证实了系统的稳定运行、高效性能、良好安全性以及优秀的用户体验。实际应用价值与未来展望：该系统不仅实现了基本的图书销售功能，而且通过高效的推荐算法提升了用户体验，具有较高的实用价值和市场潜力。基于推荐算法的网上图书销售系统的研究不仅可以应用于电子商务领域，还为新闻推送、音乐推荐、电影推荐等领域提供了参考和借鉴。未来的研究可以进一步探索更先进的推荐算法、优化系统性能、增强数据安全性以及提升用户体验。

总的来说，本论文通过对基于推荐算法的网上图书销售系统的设计与实现的研究，展示了如何利用现代Web技术和推荐算法来提升网上图书销售的效率和用户体验。这一研究成果不仅对图书销售行业具有实际应用价值，也为相关领域的推荐系统设计提供了有价值的参考和启示。

参考文献

1. 王连喜.一种面向高校图书馆的个性化图书推荐系统[J].现代情报, 2015, 35(12):6.DOI:10.3969/j.issn.1008-0821.2015.12.007.
2. 李梓嘉.基于图书市场营销的战略研究[J].前卫, 2023(5):0173-0175.
3. 于娜.电商平台智能推荐对消费者购物行为的影响分析[J].西部皮革, 2021(5):110-111.
4. Albora G, Zaccaria A. Machine learning to assess relatedness: the advantage of using firm-level data[J]. Complexity. 2022;2022
5. Ghazarian S, Nematbakhsh MA. Enhancing memory-based collaborative filtering for group recommender systems[J]. Expert systems with applications. 2015;42(7):3801–3812.
6. Zaccaria A, Cristelli M, Tacchella A, Pietronero L. How the taxonomy of products drives the economic development of countries[J]. PloS one.2014;9(12):e113770.
7. Giambattista Albora , Lavinia Rossi Mori, Andrea Zaccaria. Sapling Similarity: a performing and interpretable memory-based tool for recommendation [J]. arXiv preprint arXiv: 2210.07039v2, 6 May 2023.
8. 乔国辉.IntelliJ IDEA软件开发与应用[M].清华大学出版社,2021.
9. 黄龙泉,王磊,林程华主编.MYSQL数据库原理及应用 数据库[M].中国铁道出版社有限公司,2023.
10. 霍春阳.Vue.js设计与实现 软硬件技术[M].人民邮电出版社,2022.
11. 张峰.应用SpringBoot改变web应用开发模式[J].科技创新与应用, 2017(23):2.
12. 叶刚,王立河,王英明,等.基于Mybatis Plus的动态生成代码设计与实现[J].电脑编程技巧与维护, 2019(7):2.DOI:10.3969/j.issn.1006-4052.2019.07.002.
13. 李宇,刘彬.前后端分离框架在软件设计中的应用[J].无线互联科技, 2018.DOI:CNKI:SUN:WXHK.0.2018-17-018.
14. 朱扬勇孙婧.推荐系统研究进展[J].计算机科学与探索, 2015, 000(005):513-525.DOI:10.3778/j.issn.1673-9418.1412023. 14(9):8.DOI:CNKI:SUN:RJXB.0.2003-09-017.
15. Xie F, Chen Z, Shang J, Feng X, Li J. A link prediction approach for item recommendation with complex number[J]. Knowledge-Based Systems. 2015;81:148–158.