

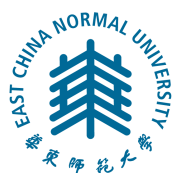
2020 届硕士专业学位论文

分 类 号: \_\_\_\_\_

学校代码: 10269 \_\_\_\_\_

密 级: \_\_\_\_\_

学 号: \_\_\_\_\_



# 華東師範大學

**East China Normal University**

**硕士专业学位论文**

**MASTER'S DISSERTATION**

**论文题目: 基于随机森林的个性化网络  
小说推荐系统的设计与实现**

院 系: 计算机科学与技术学院

专 业: 软件工程

研 究 方 向: 非定向

指 导 老 师: \_\_\_\_\_

学位申请人: \_\_\_\_\_

2020 年 10 月

# East China Normal University

**Title: Design and Implementation of  
Personalized Network Novel  
Recommendation System  
Based on Random Forest**

Department: School of Computer Science and Technology

Major: Software Engineering

Research Direction: Non-directional

Supervisor: \_\_\_\_\_

Candidate: \_\_\_\_\_

## 华东师范大学学位论文原创性声明

郑重声明：本人呈交的学位论文《基于随机森林的个性化网络小说推荐系统的设计与实现》，是在华东师范大学攻读硕士/博士（请勾选）学位期间，在导师的指导下进行的研究工作及取得的研究成果。除文中已经注明引用的内容外，本论文不包含其他个人已经发表或撰写过的研究成果。对本文的研究做出重要贡献的个人和集体，均已在文中作了明确说明并表示谢意。

作者签名：\_\_\_\_\_ 日 期： 年 月 日

## 华东师范大学学位论文著作权使用声明

《基于随机森林的个性化网络小说推荐系统的设计与实现》系本人在华东师范大学攻读学位期间在导师指导下完成的硕士/博士（请勾选）学位论文，本论文的研究成果归华东师范大学所有。本人同意华东师范大学根据相关规定保留和使用此学位论文，并向主管部门和相关机构如国家图书馆、中信所和“知网”送交学位论文的印刷版和电子版；允许学位论文进入华东师范大学图书馆及数据库被查阅、借阅；同意学校将学位论文加入全国博士、硕士学位论文共建单位数据库进行检索，将学位论文的标题和摘要汇编出版，采用影印、缩印或者其它方式合理复制学位论文。

本学位论文属于（请勾选）

- ( ) 1. 经华东师范大学相关部门审查核定的“内部”或“涉密”学位论文\*，于 年 月 日解密，解密后适用上述授权。
- ( ) 2. 不保密，适用上述授权。

导师签名：\_\_\_\_\_ 本人签名：\_\_\_\_\_

年 月 日

\* “涉密”学位论文应是已经华东师范大学学位评定委员会办公室或保密委员会审定过的学位论文（需附获批的《华东师范大学研究生申请学位论文“涉密”审批表》方为有效），未经上述部门审定的学位论文均为公开学位论文。此声明栏不填写的，默认为公开学位论文，均适用上述授权）。

## \_ 硕士学位论文答辩委员会成员名单

姓 名	职 称	单 位	备 注
XXX	教授	XXXXXX 大学数学系	主席
XXX	教授	XXXXXX 大学数学系	
XXX	教授	XXXXXX 大学数学系	

## 摘要

随着网络小说的兴起,数以千万计的网络小说数量让读者花费大量的时间在寻找符合个人喜好的小说上面。传统榜单推荐已经很难满足读者找寻小说的需求,读者希望能够很方便的找到自己喜欢看的小说,小说网站也迫切希望通过推荐读者喜欢的网络小说来增加读者对网站的粘性。因此实现网络小说个性化推荐具有重要的现实意义和社会价值。

本文利用 UML 统一建模技术,解决了系统功能性需求分析,非功能性需求分析等问题。采用聚类技术解决了小说类型划分问题。利用随机森林算法高准确率,高泛化性能的优势,实现网络小说个性化推荐功能,满足用户快速找寻网络小说的需求。在整体架构上,本论文采用基于 B/S 模式的三层架构设计方法,表示层实现用户界面,业务逻辑层解决系统的业务逻辑问题,数据访问层提供了对数据库的访问操作。使用 UML 统一建模语言,建立了系统的静态结构模型和动态过程模型。同时,根据对数据库的需求分析,设计了数据库的概念模型,并根据 E-R 图建立数据库的逻辑模型,在实现过程中,系统采用 ASP.NET 开发技术,使用母版页和 Web 服务器控件完成页面的布局。利用面向对象和事件驱动方法,编写相关控件的事件驱动代码。系统使用 MySQL 数据库存储小说特征信息,书架信息,用户信息等。在系统功能设计上,本文通过需求分析将系统功能模块划分为书架管理,书单管理,检索管理,个性化推荐,个人信息管理等五个功能模块。通过对这五个功能进行深入需求分析,功能设计,实现了系统的功能。经过功能测试和性能测试,表明这些模块能够正常完成其相应的功能,满足应用需求。

本系统具有操作简单,方便快捷,便于维护的特点,相对于传统的榜单推荐而言,本系统建立的个性化推荐模型从读者过往阅读记录中提取样本集和特征集,具有较高的准确率,贴合读者的实际需要,能够提供给读者更方便快捷的阅读服务。本系统是一次对网络小说个性化推荐的成功尝试,对推进小说网站全面展开网络小说个性化推荐具有非常大的意义,对将来的网络小说个性化推荐研发具有一定的借鉴意义。

**关键词:** 个性化推荐; 随机森林; OOB 误差估计; B/S 架构; MySQL 数据库



## Abstract

With the rise of online novels, tens of millions of online novels allow readers to spend a lot of time searching for novels that suit their personal preferences. Traditional list recommendation has been difficult to meet the needs of readers looking for novels. Readers hope to find novels they like easily. Novel websites are also eager to increase readers' stickiness to the website by recommending online novels that readers like. Therefore, the realization of personalized recommendation of online novels has important practical significance and social value.

This paper uses UML unified modeling technology to solve the problems of system functional requirement analysis and non-functional requirement analysis. The clustering technology is used to solve the problem of dividing the types of online novels. Using the advantages of the random forest algorithm's high accuracy and high generalization performance, the personalized recommendation function of online novels is realized to meet the needs of users to quickly find online novels. In the overall architecture, this paper adopts a three-tier architecture design method based on the B/S model. The presentation layer implements the user interface, the business logic layer solves the business logic problems of the system, and the data access layer provides access to the database. Using UML unified modeling language, the static structure model and dynamic process model of the system are established. At the same time, according to the analysis of the needs of the database, the conceptual model of the database is designed, and the logical model of the database is established according to the ER diagram. In the implementation process, the system adopts ASP.NET development technology, using the master page and Web server controls to complete the page layout. Use object-oriented and event-driven methods to write event-driven codes for related controls. The system uses MySql database to store novel feature information, bookshelf information, user information, etc. In terms of system function design, this paper divides the system function modules into five function modules: bookshelf management, book list management, retrieval management, personalized recommendation, and personal information management through demand analysis. Through in-depth demand analysis and functional design of these five functions, the functions of the system are realized. After functional testing and performance testing, it is shown that these modules can complete their corresponding functions normally and meet application requirements.

This system has the characteristics of simple operation, convenient and fast, and easy maintenance. Compared with traditional list recommendation, the personalized recommendation model established by this system extracts sample sets and feature sets from readers' past reading records, which has a higher accuracy rate, meets the actual needs of readers, and can provide readers with more Convenient and quick reading service. This system is a successful attempt to personalize the recommendation of online novels. It is of great significance to promote the comprehensive development of person-

alized recommendation of online novels on novel websites, and has certain reference significance for the development of personalized recommendation of online novels in the future.

**Keyword:** [Personalized recommendation] [Random forest] [OOB error estimation]  
[B/S architecture] [MySQL database]



## 目 录

摘要	i
Abstract	iii
目录	v
插图目录	viii
表格目录	ix
第一章 绪论	1
1.1 研究背景和意义	1
1.2 国内外研究现状	1
1.2.1 个性化推荐研究现状	1
1.2.2 网络小说个性化推荐研究现状	2
1.3 研究内容	2
1.4 论文结构	3
第二章 关键技术	4
2.1 随机森林算法	4
2.1.1 随机森林算法介绍	4
2.1.2 基于 OOB 误差的特征选择	6
2.2 面向对象与 UML 建模	6
2.2.1 面向对象	6
2.2.2 UML 建模	7
2.3 MySQL 数据库	7
2.4 B/S 架构	8
2.5 本章小节	9
第三章 系统需求分析	10
3.1 可行性分析	10
3.2 系统需求分析	10
3.2.1 系统功能性分析	10
3.2.2 非功能性需求	11
3.3 系统用例	11
3.3.1 用例模型	11
3.3.2 系统用例的建立	12
3.3.3 用例描述	14
3.4 基于随机森林分类的构建分析	18
第四章 系统设计	20
4.1 系统架构与设计原则	20
4.1.1 系统架构	20
4.1.2 设计原则	21
4.2 B/S 设计模式	21
4.3 系统功能模块设计	22
4.3.1 登录与注销	22
4.3.2 书单管理	23

4.3.3	小说加入书单顺序图	24
4.3.4	书架管理	24
4.3.5	小说加入书单顺序图	25
4.3.6	检索管理	25
4.3.7	个性化推荐	26
4.4	设计类图	28
4.5	顺序图	29
4.5.1	登录顺序图	29
4.5.2	小说加入书单顺序图	30
4.5.3	编辑书单顺序图	31
4.5.4	删除书单顺序图	32
4.6	数据库设计	33
4.6.1	概念设计	33
4.6.2	数据库表设计	34
<b>第五章</b>	<b>系统实现</b>	<b>38</b>
5.1	个性化推荐模块	38
5.1.1	数据集预处理	38
5.1.2	特征值重要性分析	44
5.1.3	BootStrapping 自取样	46
5.1.4	构建随机森林模型	47
5.1.5	个性化推荐	48
5.2	登录注销模块	49
5.3	书架管理模块	50
5.3.1	新建书架功能	50
5.3.2	修改书架功能	51
5.3.3	删除书架功能	52
5.4	书单管理模块	53
5.4.1	新建书单功能	53
5.4.2	修改书单功能	54
5.4.3	删除书单功能	55
5.5	本章小结	56
<b>第六章</b>	<b>系统测试</b>	<b>57</b>
6.1	测试目的	57
6.2	测试方法	57
6.3	功能测试	58
6.3.1	系统登录功能测试	58
6.3.2	书架管理功能测试	58
6.3.3	书单管理功能测试	61
6.3.4	检索管理功能测试	64
6.3.5	个人信息管理功能测试	65
6.3.6	个性化推荐功能测试	66
6.3.7	综合模块功能测试	69

6.4 性能测试 .....	69
6.5 本章小结 .....	70
<b>第七章 总结与展望</b>	<b>71</b>
7.1 总结 .....	71
7.2 展望 .....	72
<b>参考文献</b>	<b>73</b>
<b>致谢</b>	<b>76</b>

## 插图目录

2.1 随机森林生成流程图.....	5
2.2 B/S 架构图 .....	8
3.1 读者用例图 .....	13
3.2 书架管理用例图 .....	14
3.3 随机森林流程图 .....	19
4.1 系统软件架构图 .....	20
4.2 模块结构图 .....	22
4.3 登录顺序图 .....	23
4.4 书单管理流程图 .....	24
4.5 书架小说管理流程图.....	25
4.6 检索管理流程图 .....	26
4.7 个性化推荐流程图 .....	27
4.8 用户登录类图 .....	28
4.9 书单管理类图 .....	29
4.10 登录顺序图 .....	30
4.11 加入书单顺序图 .....	31
4.12 编辑书单内书本顺序图 .....	32
4.13 删除书单内书本顺序图 .....	33
4.14 系统部分 E-R 图 .....	34
5.1 Features GiniImportance .....	46
5.2 每日推荐界面 .....	48
5.3 登录界面 .....	49
5.4 管理主界面 .....	50
5.5 新建书架界面 .....	51
5.6 修改书架界面 .....	52
5.7 删除书架界面 .....	53
5.8 新建书单界面 .....	54
5.9 修改书单界面 .....	55
5.10 删除书单界面.....	56

## 表格目录

3.1 登录用例描述 .....	15
3.2 搜索小说用例描述 .....	15
3.3 加入书架用例描述 .....	16
3.4 小说评分用例描述 .....	17
3.5 加入书架用例描述 .....	18
4.1 MySql 中的用户登录表 .....	35
4.2 存储在 MySql 中的小说信息表 .....	35
4.3 存储在 MySql 中的书架信息表 .....	36
4.4 存储在 MySql 中的书单信息表 .....	36
4.5 存储在 MySql 中的小说样本表 .....	37
5.1 第一次处理后的存储的书本信息 .....	38
5.2 预处理后的存储的书本信息 .....	44
5.3 特征 Gini Importance .....	45
6.1 系统登录测试用例 .....	58
6.2 新建书架功能测试用例 .....	59
6.3 编辑书架功能测试用例 .....	60
6.4 删除书架功能测试用例 .....	61
6.5 书架书本管理测试用例 .....	61
6.6 新建书单功能测试用例 .....	62
6.7 编辑书单功能测试用例 .....	63
6.8 删除书单功能测试用例 .....	64
6.9 书单书本管理测试用例 .....	64
6.10 检索管理测试用例 .....	65
6.11 检索管理测试用例 .....	66
6.12 特征 Gini Importance .....	67
6.13 OOB 样本部分测试结果 .....	68
6.14 综合模块集成功能测试用例 .....	69
6.15 个性化推荐系统性能测试 .....	70

## 第一章 绪论

### 1.1 研究背景和意义

网络小说是指在线发布并供人网上阅读的小说，它们是网络环境中传统小说形式的发展和延伸，是传统小说基于媒体形式演变的新表达形式。

自 2020 年来，国内网络文学用户已达 4.55 亿，已经具有不可忽视的影响力<sup>[1]</sup>。在国外，网络文学也具有很大的影响力，根据艾瑞研究报告，中国网络文学在国外已拥有近千万的读者用户，另外在东南亚有超过 1.5 亿，欧洲超过 3 亿，美洲超过 4 亿的潜在粉丝用户等待挖掘<sup>[2]</sup>。网络文学继承了中国历史悠久的通俗小说和古典文化精华，与此同时也体现了时代性。网络性、文学性和时代性三性合一，让中国网文冲破不同文明阻隔，拥有了走向世界的力量，成为中国向世界推荐体现中国精神，兼具中国特色，蕴含东方底蕴的优秀文化的重要载体。

随着网络文学的兴起，数以千万计的小说数目导致了信息过载，读者花费大量的时间在浏览无用的消息上面，很难通过传统的搜索方式获取自己感兴趣的小说，继而导致读者的不断流失，小说网站必须提供给读者满意的作品，才能留住老读者并吸引新读者，而建立个性化推荐模型能够有效的提高读者的阅读体验。

本论文就是以此为出发点，通过网络信息抽取，个性化推荐技术等手段，构建网络小说个性化推荐模型，让读者可以轻松快捷获取自己感兴趣的小说。

### 1.2 国内外研究现状

#### 1.2.1 个性化推荐研究现状

国外很早就开始个性化推荐的研究，1994 年 David Goldberg 提出协同过滤的概念<sup>[3]</sup>，1995 年，Robert Armstrong 等人将个性化推荐应用在人工智能领域<sup>[4]</sup>，同年 8 月，麻省理工学院的 Henry Lieberman 在 IJCAI 上提出了个性化智能导航系统 Litizia<sup>[5]</sup>。Amazon 是美国最大的电子商务公司，它的个性化推荐主要是使用基于商品的协同过滤算法，通过对用户以前的购买和消费记录向用户推荐以

前购买过的或者浏览过但是没购买的商品,用户可以通过点击“我已经买了”或者“我不喜欢”来进行反馈优化推荐效果。Amazon 的销售方式非常成功,根据报告,Amazon 销售的 20%~30% 来自个性化推荐<sup>[6]</sup>。个性化推荐带来的巨大利润让国内外许多公司都推出自己的推荐系统,而在这些个性化推荐系统中,协同过滤算法是应用最广泛,最成功的技术<sup>[7]</sup>。协同过滤的主要思想是根据用户过去的兴趣偏好,找到相同品味或兴趣的邻居,根据邻居对其他商品的偏好来预测用户可能感兴趣的商品<sup>[8]</sup>。但随着互联网的迅速普及,用户和商品数目急剧上涨导致协同过滤的评分矩阵越来越大同时越来越稀疏,计算相似度的时间和复杂度急剧升高,矩阵的稀疏带来另一后果,评分矩阵绝大部分是未知的继而导致协同过滤算法经常预测错误<sup>[9]</sup>。因此研究人员试图通过集成学习来解决这个问题,通过统筹多个弱学习器来解决同一个问题,可以大大提高推荐系统的泛化性能<sup>[10-12]</sup>。Kearns 和 Valiant 指出,强学习算法为该算法基分类器具有很高的准确率,并且能在多项式的时间内收敛。找到准确率高于 0.5 的推荐算法非常容易,同时将它们集成又可以大大提高算法的性能<sup>[13]</sup>。与其他推荐算法相比,集成学习的优势在于性能相对稳定,泛化能力较强,而之前所述的个性化推荐算法准确率普遍不高,泛化性能较差,故可以借助集成学习的优势,把单一的推荐算法作为基学习器将两者结合起来以改善当前个性化推荐的不足<sup>[14]</sup>。

### 1.2.2 网络小说个性化推荐研究现状

在前面本文讲述了国内外个性化推荐研究的现状,国内外许多学者在这方面做了大量研究,并且进行了应用。然而,在网络小说的个性化推荐上面,国内研究仍然空缺,这主要是因为网络文学出现的时间短,缺少网络小说数据集的缘故,但随着网络文学的兴起,读者和小说数目的急速增加而带来的巨大利润让小说网站不得不重视网络小说的个性化推荐,而大量的读者和小说带来的信息过载问题,也让网络小说个性化推荐成为小说网站的重点研究方向。

## 1.3 研究内容

本文的工作内容主要是完成个性化网络小说推荐系统的开发过程。具体工作内容如下:

(1) 研究和分析了系统开发背景,阐述了个性化网络小说推荐系统对网站和读者的重要性,指出目前基于协同过滤算法的个性化推荐的不足和基于随机森林的

个性化推荐的优势。

(2) 对本文开发的个性化网络小说推荐系统所涉及的技术和算法进行研究，决定采用合适的开发方法和技术。

(3) 对个性化网络小说推荐系统进行需求分析，确定系统的用例模型，针对读者的阅读需求，设计了基于随机森林算法的分类预测模型。

(4) 对个性化网络小说推荐系统开展设计工作，确定了系统的功能结构，并对主要模块进行了详细的设计工作。

## 1.4 论文结构

本论文共分为七章，每章的具体内容简述为：

第一章，绪论，介绍了本文的研究背景和研究意义并且大略介绍了国内外个性化推荐和网络小说个性化推荐的研究现状，同时介绍了开发本系统的主要工作，意义和价值。

第二章，介绍相关背景技术，重点介绍了面向对象技术，随机森林算法，UML建模，和数据库技术。

第三章，对系统的功能需求和性能需求进行深入的分析并根据系统实际情况建立系统的用例模型，对系统主要用例进行了描述。

第四章，介绍了系统的架构设计和模块设计过程，从系统类图和顺序图的角度建立了系统的逻辑设计方案。

第五章，介绍了系统的实现过程，从界面的实现效果，业务逻辑功能实现和相关代码几方面进行了介绍。

第六章，介绍系统测试方法并针对主要功能做出了测试用例，描述了测试过程和结果。

第七章，总结了全文所做的工作，找出了不足并给出若干改进建议。



## 第二章 关键技术

### 2.1 随机森林算法

#### 2.1.1 随机森林算法介绍

随机森林算法是集成算法的集大成者<sup>[15-17]</sup>，随机森林的基学习器是决策树，决策树是通过判断分类对象多个特征的值继而预测分类对象类型的分类技术，决策树学习的目的是为了产生一颗泛化能力强，准确率高的决策树<sup>[18]</sup>，其基本流程如下所示：

---

#### 伪代码 2.1 决策树学习基本算法

---

**输入：**训练集  $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ ，属性集  $A = \{a_1, a_2, \dots, a_d\}$ .

**输出：**以 node 为根结点的一颗决策树

```

1: function TreeGenerate( $(D, A)$ )
2:   生成结点 node;
3:   if D 中样本全属于同一类别 C then
4:     将 node 标记为 C 类叶结点; return
5:   end if
6:   if  $A = \emptyset$  OR D 中的样本在 A 上取值相同 then
7:     将 node 标记为叶结点，其类别标记为 D 中样本数量最多的类; return
8:   end if
9:   从 A 中选择最优划分属性  $a_*$ ;
10:  for  $a_*$  中的每一个值  $a_*^v$  do
11:    为 node 生成一个分支；令  $D_v$  表示 D 中在  $a_*$  上取值为  $a_*^v$  的样本子集;
12:    if  $D_v$  为空 then
13:      将分枝结点标记为叶结点，其类别标记为 D 中样本最多的类; return
14:    else
15:      以 TreeGenerate( $D_v, A \setminus \{a_*\}$ ) 为分支结点;
16:    end if
17:  end for
18: end function

```

---

与一般集成学习算法不同的是，随机森林算法在特征集上也实现了随机，所

以具有更好的泛化性能。随机森林的每颗决策树通过自助法 (bootstrap) 进行有放回的等概率抽样来确定训练集，同时在结点划分时选择随机候选特征子集，最后采用所有基学习器中的多数类做为随机森林的最终预测结果。随机森林生成流程图如图2.1所示

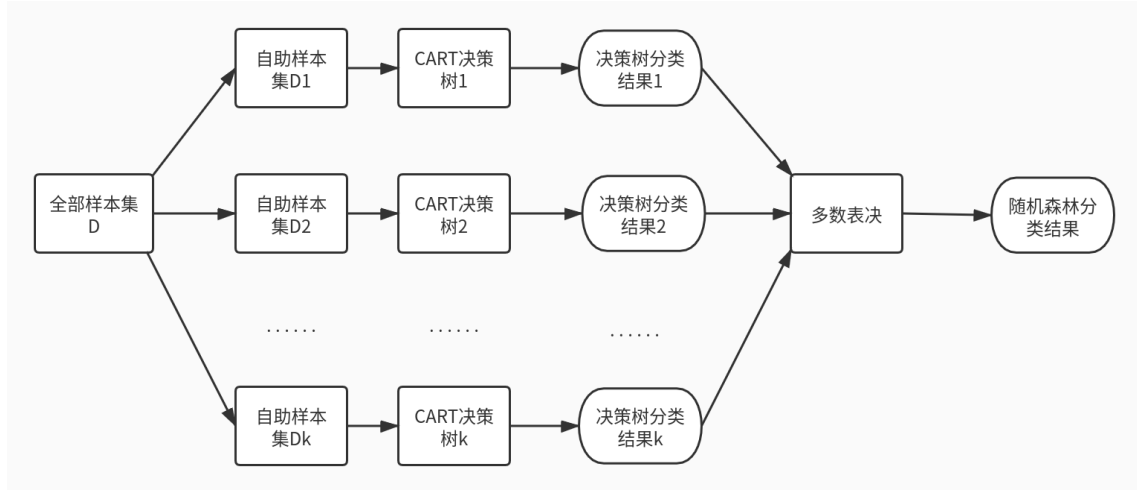


图 2.1 随机森林生成流程图

随机森林在生成过程中有以下特点：

(1) 单棵决策树的训练样本是通过从样本集中自助采样法得到，自助采样从样本容量为  $m$  的样本集  $D$  中采样  $m$  次，一个样本在  $m$  次采样不被采到的概率如下所示：

$$\lim_{m \rightarrow \infty} \left(1 - \frac{1}{m}\right)^m = \frac{1}{e} \approx 0.368 \quad (2.1)$$

由公式我们可知，训练样本集约占总样本集的 63.2%，剩下的 36.8% 可以在将来做为检测数据。

(2) 在决策树生成结点时，并不是从特征集所有特征中计算每个特征的基尼指数，然后选择最佳特征进行划分，而是从特征集中随机抽取  $K$  个候选特征，计算这  $K$  个特征的基尼指数，然后从里面选择最佳特征做为结点特征。在随机森林生成过程中， $K$  的值是固定的。

(3) 随机森林中每棵决策树不进行剪枝处理，随机森林本身两次随机策略（生成子树时随机抽取训练样本集，结点分裂时选择随机  $K$  个候选特征）能够很好的避免过拟合。

### 2.1.2 基于 OOB 误差的特征选择

在处理多维数据时, 如何从大量的特征中去除冗余或者无关的特征是提高预测模型泛化性能的关键。因为随机森林生成子树的训练集是通过自取样得到, 所以每颗子树都有大约 36.8% 的样本没有被使用到, 可以用这些样本来做 OOB 误差估计<sup>[19]</sup>, 该方法的优点就是可以直接获得模型的泛化误差, 而不需要通过交叉验证或额外的计算来检验模型的优劣, 且 OOB 误差估计属于无偏估计<sup>[20]</sup>。然后以 OOB 误差来对随机森林的特征集进行评估, 从而剔除不相关特征和冗余特征<sup>[21]</sup>。基于 OOB 误差估计的特征选择, 其基本原理是: 在随机森林模型生成过程中, 对特征  $X_j$  的取值随机置换, 假如置换该特征值后, 随机森林模型的预测准确率发生大的变化, 说明该特征比较重要<sup>[22]</sup>。可以通过依次置换所有特征的取值来估算每个特征的重要程度, 计算过程如下:

首先, 计算预测特征  $X_j$  在第  $t$  棵树中的重要性, 计算公式如下:

$$VI^{(t)}(X_j) = \frac{\sum_{i \in B(t)} I(y_i = y_i^{(t)})}{|B(t)|} - \frac{\sum_{i \in B(t)} I(y_i = y_{i,\pi_j}^{(t)})}{|B(t)|} \quad (2.2)$$

$B(t)$  是第  $t$  棵树的 OOB 数据,  $y_i^{(t)} = f^{(t)}(x_i)$  是预测特征  $X_j$  被替换前第  $i$  条记录的预测结果,  $y_{i,\pi_j}^{(t)} = f^{(t)}(x_i, \pi_j)$  是预测变量  $X_j$  被替换后第  $i$  条记录的预测记过, 由于节点的候选分裂属性是随机抽取的, 因此会出现第  $t$  棵树中没有预测特征  $X_j$  的情况, 此时定义  $VI^{(t)} = 0$ 。

变量  $X_j$  的重要性计算结果是取所有树中该变量重要性的平均值, 计算公式如下:

$$VI(X_j) = \frac{\sum_{t=1}^n VI^{(t)}(X_j)}{n} \quad (2.3)$$

计算出来的结果就是该特征对随机森林包外误差率的影响, 以此为依据来选择合适的特征子集。

## 2.2 面向对象与 UML 建模

### 2.2.1 面向对象

面向对象方法 (Object-Oriented Method) 通过类和对象对系统中的实体和行为进行描述, 它的优点体现在代码的安全性和复用性上<sup>[23]</sup>。对于代码的安全性问题, 因为对象的方法是进行封装的, 对于外部调用者而言, 除了那些共享的公开信息, 是无法看到对象的内部结构的。而在代码的复用问题上, 面向对象中父

类派生子类的对象会继承父类中对象的特点,这本身就是一种很好的代码复用,可以大大减少软件开发的代码量<sup>[24]</sup>。另外,面向对象开发还具有开发效率高,开发周期短的特点,而且也便于开展对系统进行功能扩充和维护工作。

### 2.2.2 UML 建模

UML 建模是软件开发必须要掌握的一项基本技能<sup>[25]</sup>,特别是在面向对象开发时代,良好的建模过程可以显著降低系统的开发难度和复杂性。在面向过程开发中,瀑布模型是最常用的,而在面向对象的开发过程中,主要使用 Rational 统一过程 (RUP) 方法,在使用 RUP 方法的过程中,一个非常重要的特点是需要 在软件开发的每一个阶段都建立适当的开发模型<sup>[26]</sup>。由于软件开发涉及许多过程,如需求分析,系统涉及,代码编写等,因此有必要使用 RUP 方法为其都建立模型。可以说,UML 在建模方面汲取了很多建模工具的有点,把建模工作的可视化和文档化做得近乎完美<sup>[27]</sup>。将 RUP 和 UML 建模结合起来的开发模式已经是当前主流的面向对象开发模式,UML 常用的建模图形有用例图、时序图、类图、活动图等<sup>[28]</sup>,在不同的开发阶段使用不同的建模图形描述,比如在系统需求分析阶段使用用例图和活动图来描述系统功能和业务活动流程<sup>[29]</sup>,在系统设计阶段在使用类图和顺序图来描述软件系统的静态结构和动态过程<sup>[30]</sup>,在系统开发阶段则用部署图和程序包图来展示软件和硬件之间的逻辑关系<sup>[31]</sup>。总而言之,UML 建模能够很好的展示系统开发过程中各个阶段所涉及的静态结构和动态过程。

## 2.3 MySql 数据库

MySql 是当前最流行的关系型数据库,功能强大,使用 Sql 语言进行增删改查<sup>[32]</sup>,本系统选择 MySql 数据库存储书本信息.书单信息等系统信息,主要出于以下考虑,首先,它可以很好地满足数据的移植性能,可以在不同的平台中运行,代码经过良好的测试,具有安全稳定的运行效率<sup>[33]</sup>,其次,它具有较好的适应性,在不同的操作系统中可以很好地运行,可以支持不同的用户语言的习惯,满足多语言的接口,而且还可以在源代码中实现数据对象的高效查询,满足极短时间范围内的数据对象的查询<sup>[34]</sup>。最后 MySql 数据库是开源的,完全免费,完全可以做到在不降低系统性能的条件下降低系统开发成本。综上所述,经过分析比较,MySql 数据库与其他的数据产品相比具有一定的特性和优势,可以满足用户

的数据存储和管理的要求

## 2.4 B/S 架构

B/S 架构中,B 代表的是 Browser,即浏览器;S 代表的是 Server,即服务器。B/S 架构,其实是一种特殊的 C/S 架构,只不过这个 C(Client)特指浏览器。对于 B/S 架构的系统,浏览器只负责显示用户页面和提交请求,所有的逻辑业务都集中在服务器中处理,系统进行升级更新时只要修改服务器的代码,极大地减轻了升级所需的工作量,降低了人工成本。因为这个架构是基于浏览器的,软件开发主要关注后台的业务处理逻辑,通过三层架构可以将数据与表现分离出来。前端是客户端,主要是用户的浏览器<sup>[35]</sup>。用户只需使用自己设备中的浏览器进行访问,不用安装部署任何软件,后期的使用更没有升级维护软件的麻烦。中间是服务器端,是进行业务处理的。后台是数据库,详细的架构具体如下图所示<sup>[36]</sup>:

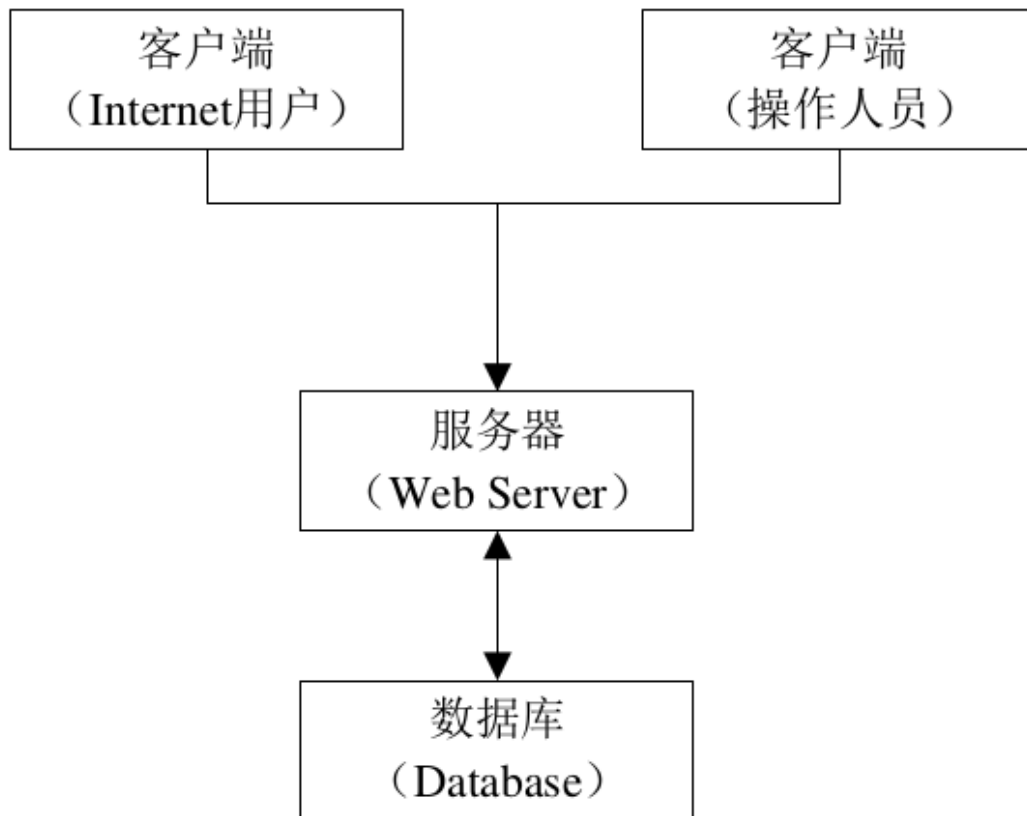


图 2.2 B/S 架构图

在上述的 B/S 架构中主要包括了三个部分,分别是客户端、服务器、数据库:

- (1) 客户端: 主要的工作是接收用户的数据信息。由用户端的浏览器工具来完成, 显示系统所处理结果。用户端在浏览器中通过 HTTP 请求将数据发送给服务器端, 对于符合要求的请求在服务器端进行处理, 将结果显示在用户端中。
- (2) 服务器端: 主要是进行业务的逻辑处理, 在三层架构中起到上下承启作用。
- (3) 数据库: 主要是进行中间业务的请求, 对各类请求的响应和处理, 并将处理结果的数据完成存储<sup>[37]</sup>。

## 2.5 本章小节

综上所述, 在本系统的开发中, 主要工作是中间业务的处理和数据库设计, 充分利用三层架构的优势, 建立业务与数据之间的连接。

## 第三章 系统需求分析

### 3.1 可行性分析

可行性分析是分析当前是否具备开发本系统的必要条件，以下从三个方面对系统进行可行性分析：

(1) 技术可行性。它是指目前使用的开发工具，模型和算法是否达到系统的设计要求。本系统采用的开发工具是.NET 平台下的 Visual Studio 集成式开发环境，使用 MySQL 数据库存储系统信息。这个开发组合完成可以满足 Web 的普通应用系统的开发需要。系统涉及的个性化推荐已经较为成熟，目前查阅到的文献资料较多，对系统的开发过程很有帮助。从算法实现角度看，本系统采用的随机森林算法，该算法在分类方面具有较高的准确率和泛化性能，因此完全可以满足网络小说个性化推荐系统中对小说个性化推荐的需求。

(2) 经济可行性是指在资源支持上开发本系统的可能性，资源包括人力资源，自然资源和资金条件等。任何软件的开发都离不开经济资源的支持，通常，用户系统开发的经济资源的占有率和系统目标的期望值成正比。对于小说网站来说，系统开发的目的是为了增加读者的粘性，增加写得出色的冷门小说曝光度，这本身对于小说网站的发展是必不可少的资金投入，而且这种投入也值得去尝试。

(3) 操作可行性，操作可行性主要指用户对系统的功能使用是否具有过大的障碍，本系统采用 B/S 模式开发，用户操作和上网一样简单。因此，从使用的角度看，用户不存在任何的应用障碍。

从系统开发本身带来的积极影响看，小说网站是支持本系统的开发的，完全可以将本系统做为小说网站的一个特色模块加入小说网站中去。

### 3.2 系统需求分析

#### 3.2.1 系统功能性分析

在分析功能需求之前，第一步是分析系统的参与者，明确参与者之后才能进一步分析系统应该提供的功能。经过对系统业务的初步了解后得知，系统的主要服务对象是读者，读者的功能性需求如下：

- (1) 书架管理功能包括新建书架，修改书架，删除书架等功能，以及对于书架中小说的管理，比如将小说从书架中删除，将小说更换书架等。
- (2) 书单管理功能包括新建书单，修改小说，删除书单等功能，以及对于书单中小说的管理，比如将小说从书单中删除，修改书单中某本小说的评分评论，将小说更换书单等。
- (3) 个性化推荐。读者服务信息，主要是每日推荐，通过基于读者书架上面的书本阅读信息进行个性化推荐。
- (4) 检索管理，主要包括小说和书单的检索，检索时输入小说和书单的关键字进行检索，检索成功则会列出相关的小说和书单，检索失败会给予提示。
- (5) 个人信息管理，主要有两个功能，一个是管理用户的个人信息，另一个是管理用户密码。
- (6) 登录与注销，读者通过登录才能使用系统功能, 通过注销退出系统。

### 3.2.2 非功能性需求

非功能性需求是指软件产品除了满足用户业务需求的功能外的一些软性需求，它包括：

- (1) 互操作性，互操作性是指用户与系统之间交互的性能，良好的互操作性意味着用户可以轻松便捷地使用系统功能，这就要求在界面的设计中必须提供良好的用户操作界面。
- (2) 安全性，安全性是系统价值的重要体现，如果一个系统不安全, 那对于用户而言, 会面临数据和信息丢失或泄密的风险。在网络小说个性化推荐系统开发过程中, 要保证系统保存的信息的安全以及用户访问系统过程的安全。
- (3) 可靠性，是指系统在指定时间内完成指定功能的能力，如果一个系统功能的完成需要用户等待很长时间，那么这个系统对用户而言也是不友好的。

## 3.3 系统用例

### 3.3.1 用例模型

用例模型由参与者、具体用例及其关系、用例描述以及系统边界组成，其中参与者是系统的使用者，是指系统之外的，在使用或与系统交互时所扮演的角



色，用例则是系统提供给参与者的功能，用例之间存在多种关系，如包含、扩展等，这些关系也需要反映在用例图中，系统边界指的是系统与外界的接口，参与者应该在系统边界之外。本系统的参与者主要是读者，系统提供的功能有书架管理、书单管理、个性化推荐、检索管理、个人信息管理等。

### **3.3.2 系统用例的建立**

用例模型包括业务用例模型和系统用例模型，这两种类型的用例模型都在系统分析期间使用，但这两种模型之间还是存在一定差异。两者之间的区别在于，业务用例模型带有一定抽象性，其描述的概念范围通常更广，具体和某个抽象的业务操作功能有关，而系统用例模型则是把业务用例模型经过细化后的具体用例，其描述的用例更详细具体，用例图最终都需要借助系统用例模型进行表达。对于读者而言，其用例图如图3.1所示。

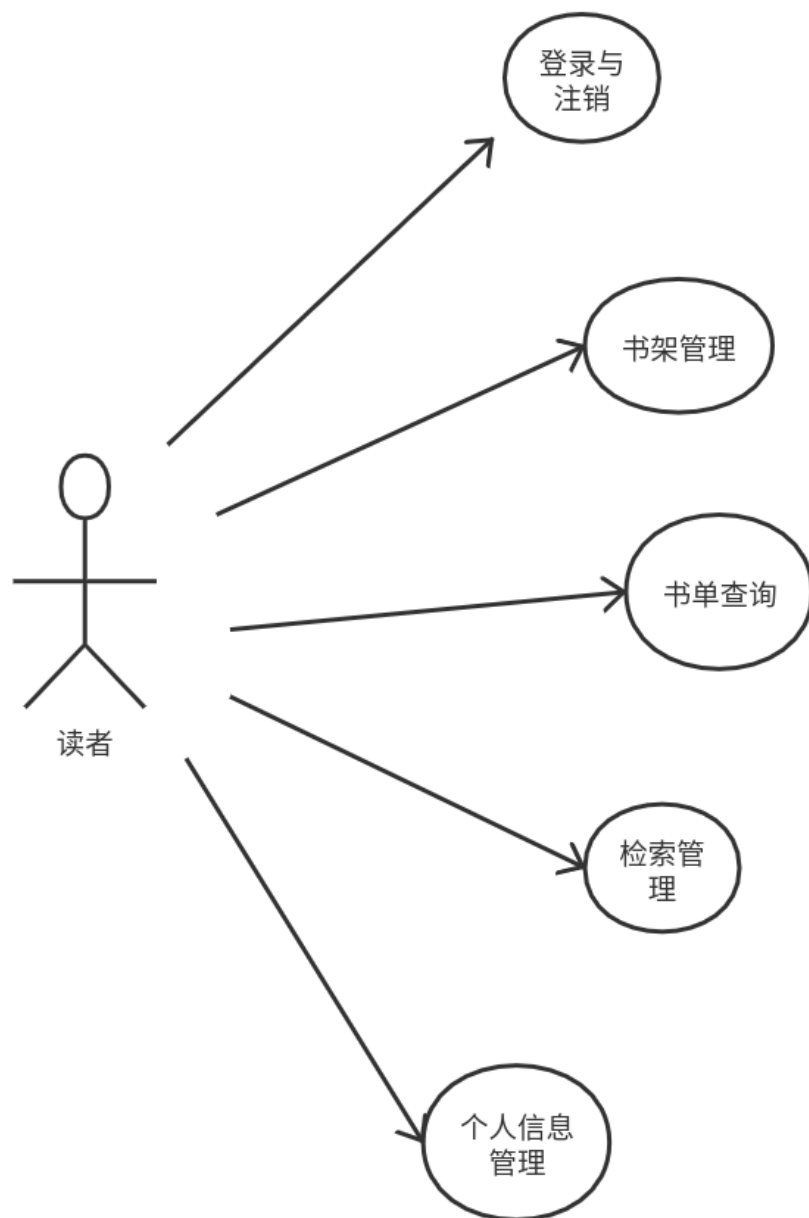


图 3.1 读者用例图

有些用例还可以再进行细化, 如书架管理用例可以细化为如图3.2所示的子用例

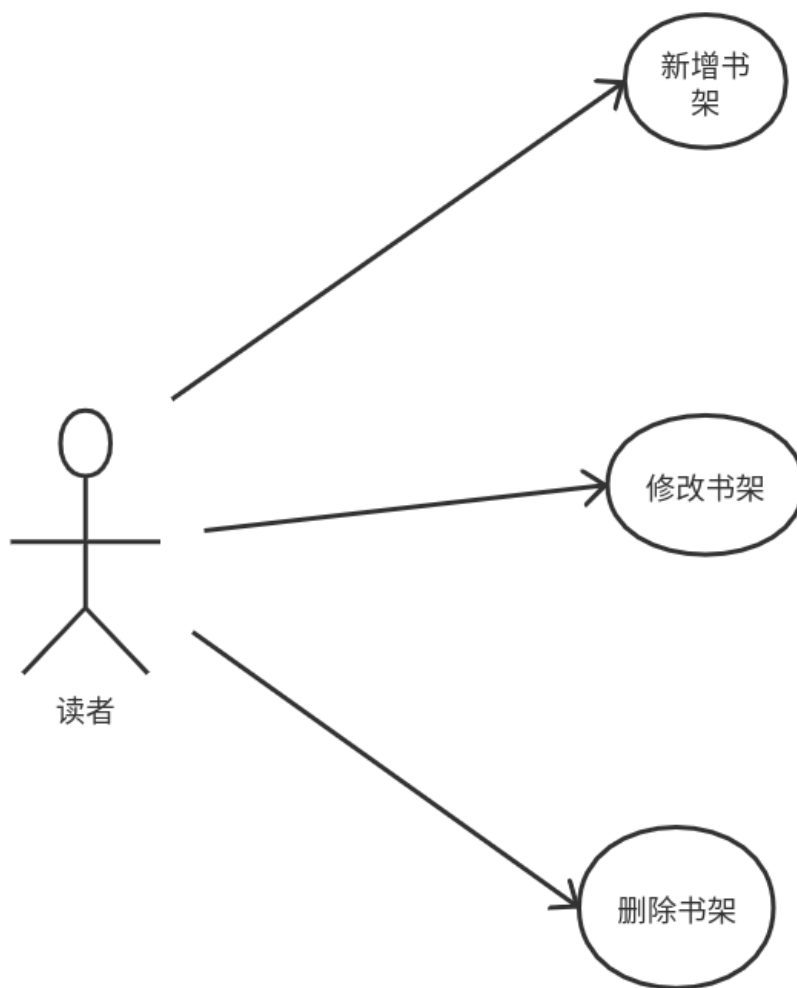


图 3.2 书架管理用例图

检索管理用例可以再次细化为检索小说，检索书单等用例。书单管理用例可以再次细化为新增书单，修改书单，删除书单等用例。

### 3.3.3 用例描述

用例图并没有详细描述用例的具体行为，为了更好的把握用例的执行细节，我们需要对用例进行更详细的描述。用例描述需要按照一定的格式进行，没有标准的格式，但至少需要包含参与者、简要说明、前置条件、基本事件流、其他事件流、后置条件等内容。描述的内容可以简单，也可以稍微复杂点，重点是要能够说明用例的具体执行步骤，下面介绍系统用例中的若干过程描述。

#### (1) 用例名称: 登录

登录用例描述如表3.1所示。

表 3.1 登录用例描述

参与者	读者用户
简要说明	读者通过自己的账号登录系统
前置条件	网络连接正常
基本事件流	①系统提示输入用户名和密码; ② 参与者输入用户名和密码; ③参与者点击” 登录” 按钮, 等待系统验证; ④用例结束。
其他事件流	在按” 登录” 按钮之前, 参与者随时可以取消本次登录。
异常事件流	①用户名和密码输入错误, 请参与者确认; ②返回登录界面;
后置条件	参与者成功登录系统, 并显示登录成功信息;

## (2) 用例名称: 搜索小说

搜索小说用例描述如表3.2所示。

表 3.2 搜索小说用例描述

参与者	读者用户
简要说明	参与者搜索小说
前置条件	参与者已经进入小说检索界面
基本事件流	①系统显示小说检索界面; ② 参与者输入小说名; ③参与者点击” 搜索” 按钮; ④用例结束。
其他事件流	在按” 搜索” 按钮之前, 参与者随时可以取消本次搜索。
异常事件流	异常事件流一: ①书名输入错误, 找不到书本; ②返回检索界面; 异常事件流二: ①书名为空, 无法进行检索; ② 返回检索界面;
后置条件	参与者完成对小说的搜索;

## (3) 用例名称: 加入书架

加入书架用例描述如表3.3所示。

表 3.3 加入书架用例描述

参与者	读者用户
简要说明	参与者将小说加入书架
前置条件	参与者已经搜到小说
基本事件流	①系统显示小说详情页； ② 参与者点击加入书架； ③参与者在下拉按钮选择书架名字； ④用例结束。
其他事件流	①书本已经加入书架，显示书架名字； ②点击书架名字，出现其他书架名字 ③点击新的书架名字，完成书本更换书架；
异常事件流	①书架已满，无法加入书架；
后置条件	参与者完成将小说加入书架；

#### (4) 用例名称: 小说评分

小说评分用例描述如表3.4所示。

表 3.4 小说评分用例描述

参与者	读者用户
简要说明	参与者对小说进行评分
前置条件	参与者进入小说详情页
基本事件流	①点击小说详情页”我的评分”按钮; ② 根据自己喜好评分为一星到五星; ④用例结束。
其他事件流	其他事件流一: ①点击发表评论; ②在输入框中输入对小说的评价; ③ 点击”发表评论”按钮, 评论保存并发表。 其他事件流二: ①浏览他人评论; ② 对他人评论点赞;
异常事件流	①评论字数超过限制, 无法发表评论; ②系统提示字数超过限制, 无法发表; ③修改评论, 直到可以符合要求;
后置条件	参与者完成对小说的评分, 评论;

## (5) 用例名称: 加入书单

读者将小说加入书单用例描述如表3.5所示。

表 3.5 加入书单用例描述

参与者	读者用户
简要说明	参与者将小说加入书单
前置条件	参与者已经搜到小说
基本事件流	①系统显示小说详情页; ② 参与者点击加入书单; ③系统出现方框, 要求选择加入书单名字, 对小说的评分, 对小说的评论; ④填写完成, 点击” 确定添加”。 ⑤用例完成。
其他事件流	①选择加入书单名字时, 选择创建新书单; ②系统出现创建书单界面; ③填写书单名字, 书单简介; ④点击创建书单; ⑤书单创建成功;
异常事件流	①书单已满, 无法加入书单;
后置条件	参与者完成将小说加入书单;

### 3.4 基于随机森林分类的构建分析

根据前面的随机森林算法流程, 在本系统中, 需要解决的实际问题是如何根据现有的书架书本数据对读者建立个性化模型, 然后根据模型给读者推荐合适的小说。在具体的处理过程中, 需要选择一定规模的小说做为样本集, 通过自取样划分为训练样本集和测试样本集, 然后使用训练样本集进行训练子树, 将多个子树的分类进行分类投票, 选择票数最高的类别。在这个训练过程中可以通过检测集对该模型进行 OOB 误差估计, 然后通过 OOB 误差的估计来进行特征的取舍, 从而优化模型。随机森林的流程图如图3.3:

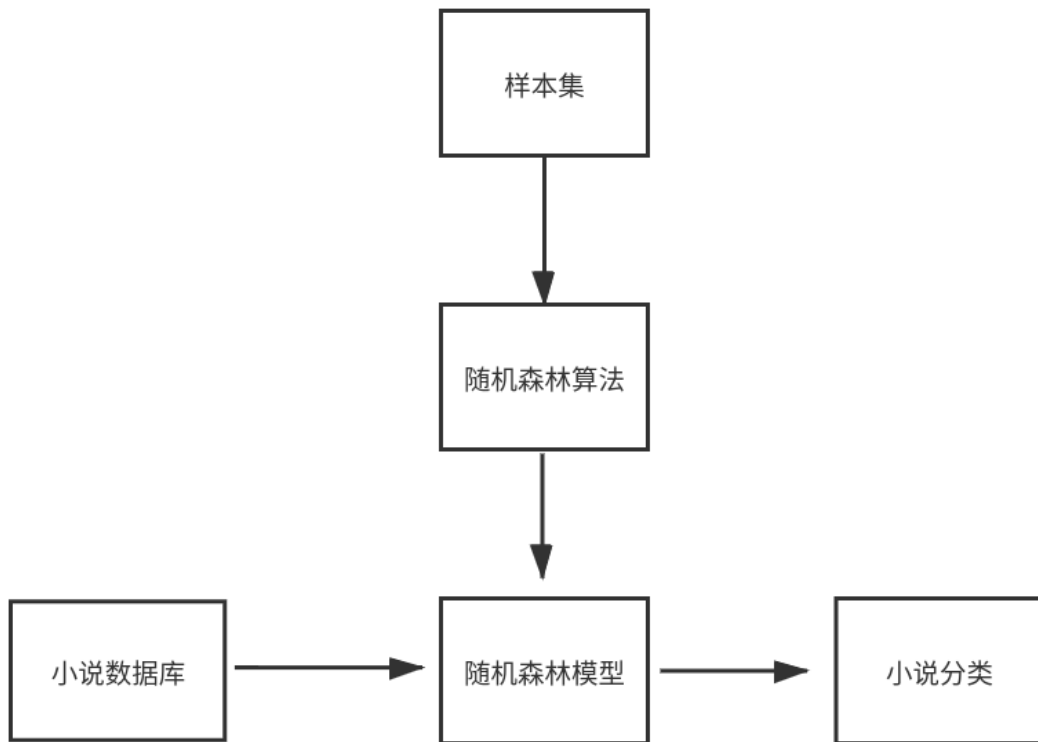


图 3.3 随机森林流程图



## 第四章 系统设计

系统设计是需求分析之后的阶段，主要根据系统需求对网络小说个性化推荐系统的功能模块，类图，数据库进行设计，系统设计的成果将会直接服务于系统的实施和代码编写过程。

### 4.1 系统架构与设计原则

#### 4.1.1 系统架构

系统架构包括硬件体系架构和软件设计架构两个方面。硬件架构采用客户端-应用服务器-数据库服务器模式。软件设计架构采用表示层、业务逻辑层和数据访问层三层架构形式。系统软件架构如图 4.1 所示。

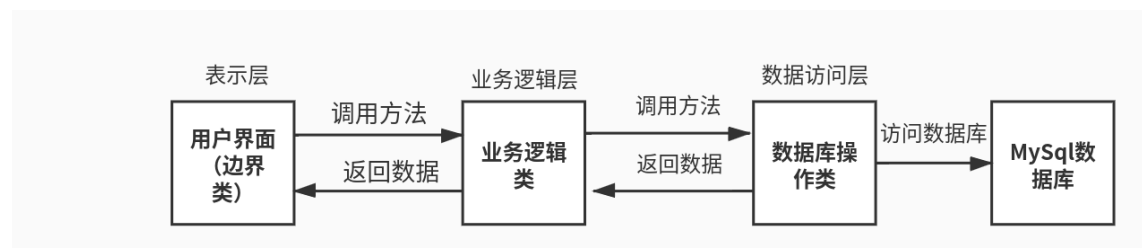


图 4.1 系统软件架构图

(1) 表示层: 即用户界面层, 它是由各种边界类组成。表示层的作用是为用户提供操作界面, 通过它可以输入数据或者显示系统的输出数据。在具体设计时, 表示层主要由各种 HTML 标记组成的网页文件实现, 但也会包含少量的用于客户端验证的 JavaScript 脚本代码。

(2) 业务逻辑层: 主要由各种处理业务逻辑功能的类组成, 用于负责处理网络小说个性化推荐系统管理过程中涉及的各功能模块的应用逻辑。它接收来自表示层中页面的请求, 完成相关的业务逻辑处理。如果涉及到数据库操作, 还需调用数据访问层中的数据操作类。

(3) 数据访问层: 是系统与数据库之间的操作界面, 主要用于解决对数据库的具体的访问过程。数据访问层由各种数据操作类组成, 往往会针对一张表定义一个数

据操作类,类中包含了对数据表中记录的增、删、改、查等功能的对应代码。

#### 4.1.2 设计原则

- (1) 坚持一切以用户为中心的原则。系统开发结果的好坏,最终的评论权在用户那里。如果开发的系统得不到用户的承认,那设计的系统就没有任何意义可言;
- (2) 坚持以实用性为导向的原则。任何一个系统的开发都是为了解决实际问题。对于本系统而言,其实用性主要体现在解决小说网站实际应用需求,帮助小说网站更好的开展对读者用户的服务;
- (3) 坚持以安全性为关键要素的原则。系统安全性体现在使用网络小说个性化推荐系统之前需要通过用户名和密码来验证是否属于合法用户,在保存用户信息时会对重要敏感信息进行加密保存;
- (4) 坚持以可扩展、可维护为重要指标的原则。为了便于对系统进行升级和维护,系统在设计时就应该要具有可扩展性和易维护性。对于本系统来说,软件设计采用的是基于 B/S 模式的三层架构设计理念,这可以满足可扩展性要求。同时,系统业务逻辑在服务器端实现,只需要对服务器端进行维护就能达到维护整个系统的目的。

## 4.2 B/S 设计模式

B/S 模式目前开发 Web 应用系统的主要模式。随着互联网时代的到来,浏览器已经成为用户访问服务器资源的主要工具,传统的 C/S 开发模式要求用户安装专门的软件做为前端,这对软件的升级和维护会带来一定的困难,尤其是客户端机器数量比较多且又分布在不同地理位置的情况下。采用 B/S 模式开发的应用系统的业务逻辑功能主要在服务器端实现,对系统功能的增删可以通过改变服务器中网页的数量就能完成,有利于对系统进行升级和维护。由于本系统用户在使用系统时的地理位置不常固定,采用 B/S 模式开发才符合用户的使用实际。这也是本系统为何采用 B/S 模式进行开发的原因。

### 4.3 系统功能模块设计

根据前面的需求分析结果, 可以把系统功能概括为以下几个模块, 系统模块结构如图 4-2 所示:

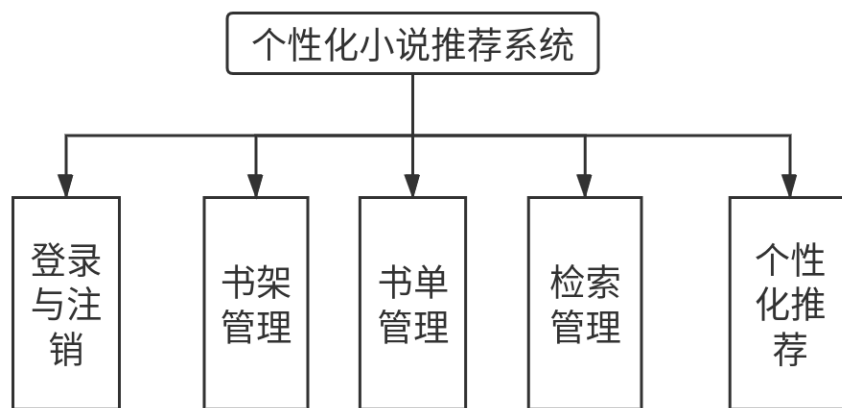


图 4.2 模块结构图

#### 4.3.1 登录与注销

在对登录功能进行设计时, 系统需要设计一个登录界面, 用于用户输入自己的登录信息。用户输完登录信息后向系统进行提交登录请求。系统会对用户进行身份验证, 验证通过后该用户才能进入系统访问。登录流程如图 4.3 所示。

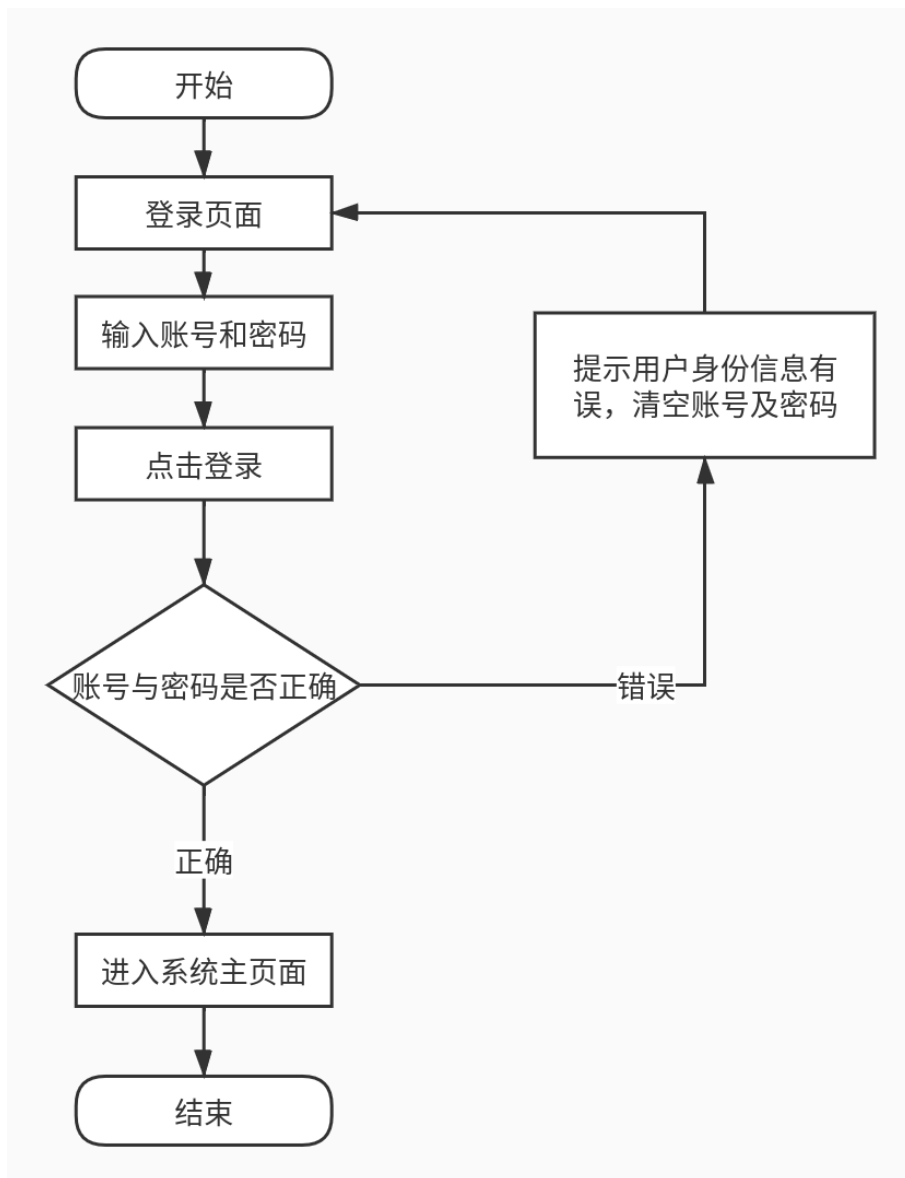


图 4.3 登录顺序图

### 4.3.2 书单管理

从模块的功能设计来看，书单管理主要是用于对书单中小说信息的添加，修改和删除等操作。添加新小说时，通过添加界面完成，最终把书单信息保存到数据库中该书单的信息表中。同样，对书单中小说信息的修改也是针对书单信息表进行，修改后也需要保存书单的修改信息。删除书单中小说信息时需要确认是否真的删除，因此一旦删除后读者对该小说的评论就不存在的，因此，可以给书单中的小说信息做个删除标志，先不进行物理删除，这样就有机会恢复该书单中的小说评论信息。书单管理流程如图 4.4 所示。

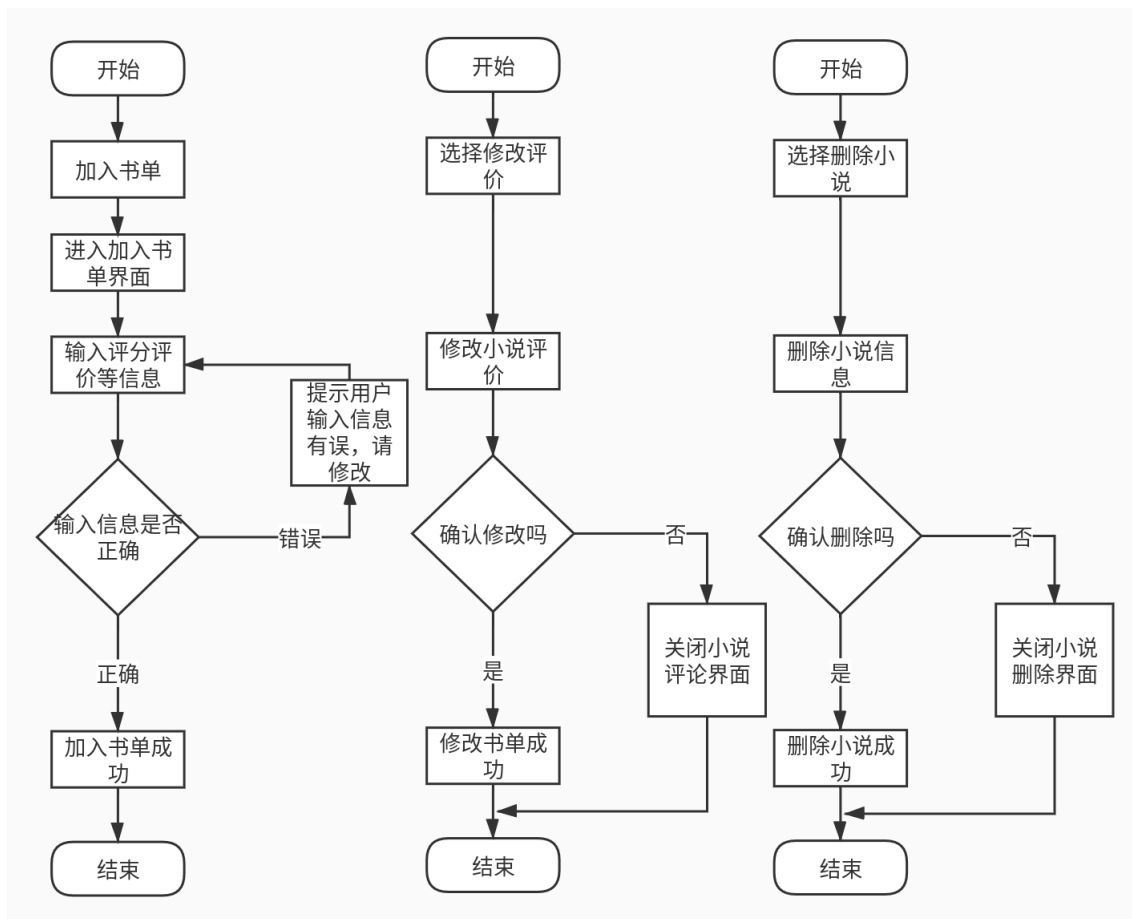


图 4.4 书单管理流程图

### 4.3.3 小说加入书单顺序图

### 4.3.4 书架管理

从模块的功能设计来看, 书架管理主要是用于对书架中小说信息的添加, 修改和删除等操作。添加新小说时, 通过添加界面完成, 最终把书单信息保存到数据库中该书架的信息表中。同样, 对书架中小说信息的修改也是针对书架信息表进行, 修改后也需要保存书架的修改信息。删除书架中小说信息时需要确认是否真的删除, 因此一旦删除后该书架中该小说就不存在的, 因此, 可以给书架中的小说信息做个删除标志, 先不进行物理删除, 这样就有机会恢复该书架中的小说评论信息。书架管理流程如图 4.5 所示。

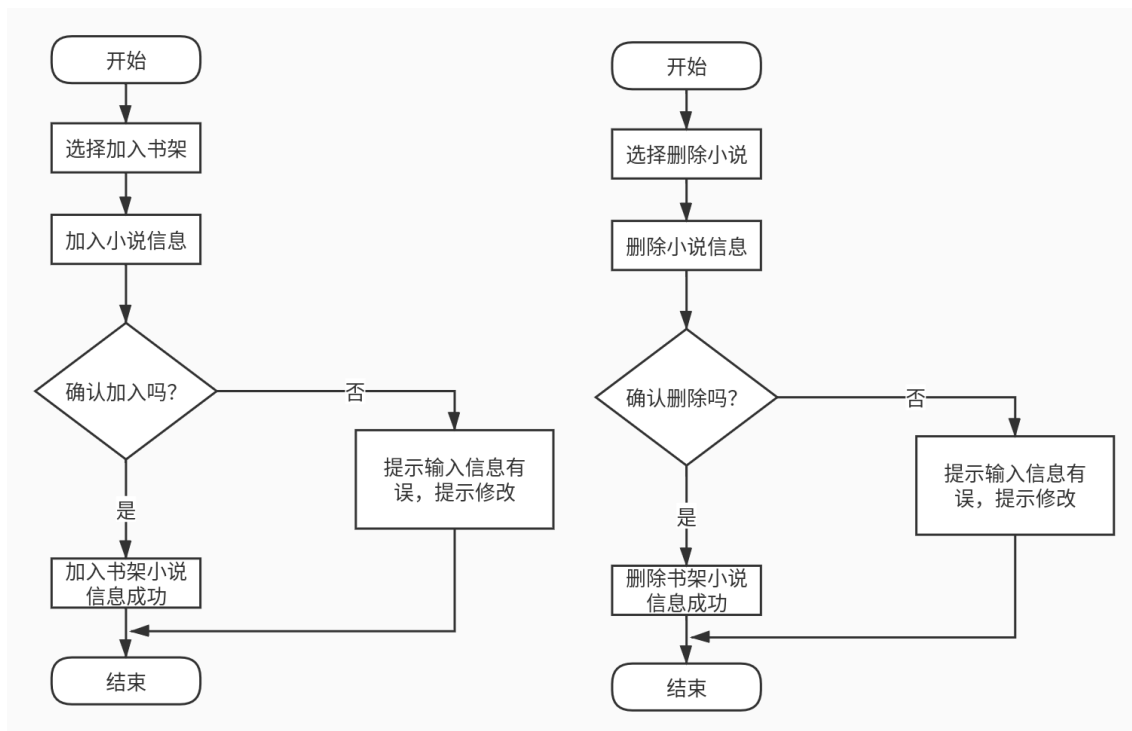


图 4.5 书架小说管理流程图

#### 4.3.5 小说加入书架顺序图

#### 4.3.6 检索管理

从模块的功能设计来看, 检索管理主要是用于对小说和书单的检索。检索管理流程如图4.6 所示。

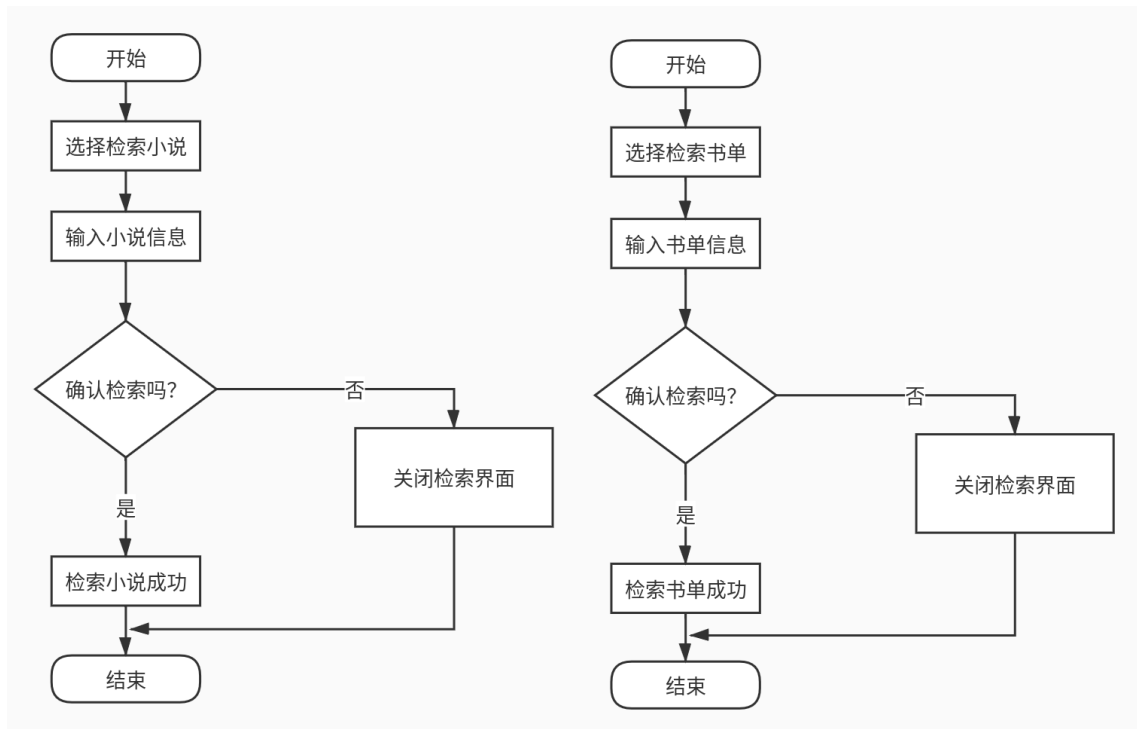


图 4.6 检索管理流程图

#### 4.3.7 个性化推荐

个性化推荐模块的功能是针对样本集训练出分类模型, 然后根据模型去预测读者对一个陌生小说是喜欢还是不喜欢。个性化推荐模块流程如图 4.7 所示

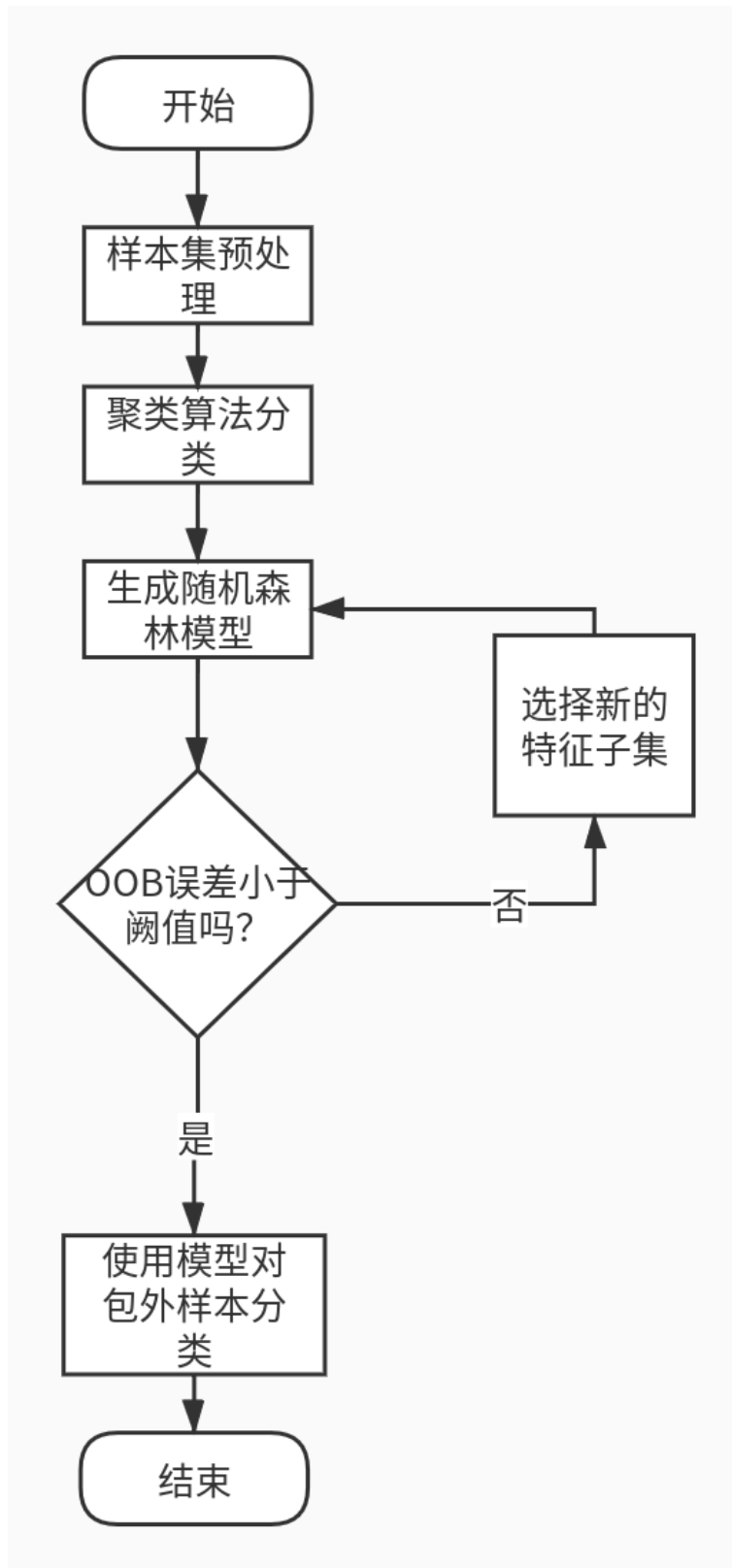


图 4.7 个性化推荐流程图



## 4.4 设计类图

我们一般把类划分为边界类、实体类和控制类。边界类对应用户界面; 实体类对应实体信息及其操作; 控制类对应于系统业务逻辑的管理, 在三层架构模型中, 边界类对应表示层中的类, 主要是对 Web 界面进行操作, 控制类对应于业务逻辑层中的类, 负责连接前端和后端, 而实体类对应数据访问层中的类, 负责对 MySQL 数据库中存储的信息进行操作。

### 1、登录类图

在登录类图中, Login.aspx 文件是属于表示层中的边界类对象, 登录管理类 LoginMange 是属于业务逻辑层的类, 登录操作类 loginop 则是属于数据访问层的类。用户登录类图, 如图 4.8 所示。

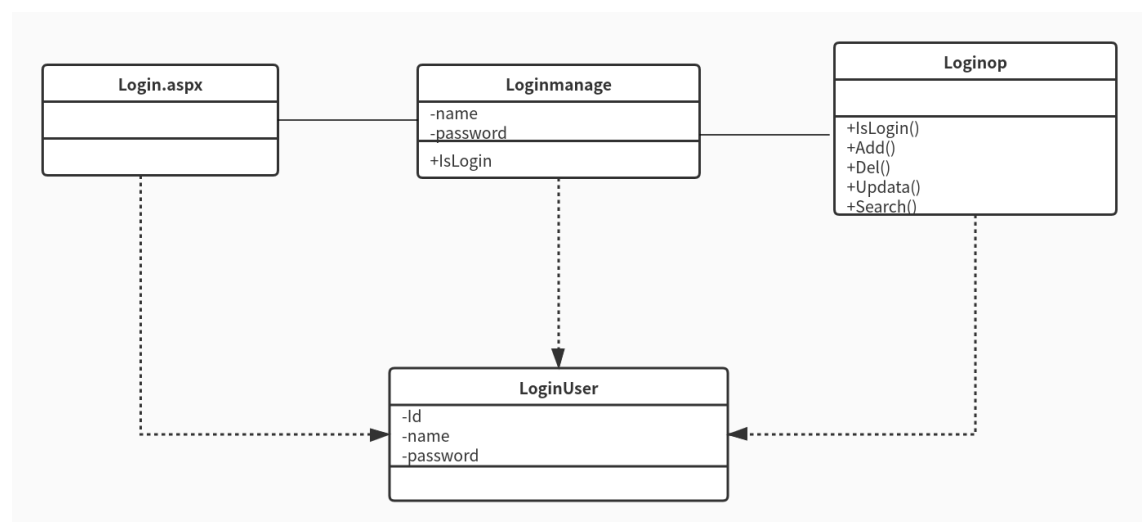


图 4.8 用户登录类图

说明如下:

- (1) Login.aspx 对应于表示层中的登录界面;
- (2) 控制类 LoginManage 对应于业务逻辑层, 类中方法 IsLogin() 用于返回用户登录成功与否的返回信息。IsLogin() 方法中使用了 name,password 两个参数。这两个参数由表示层中的边界类传递。

### 2、书单管理类图

书单管理类图如图 4.9 所示。

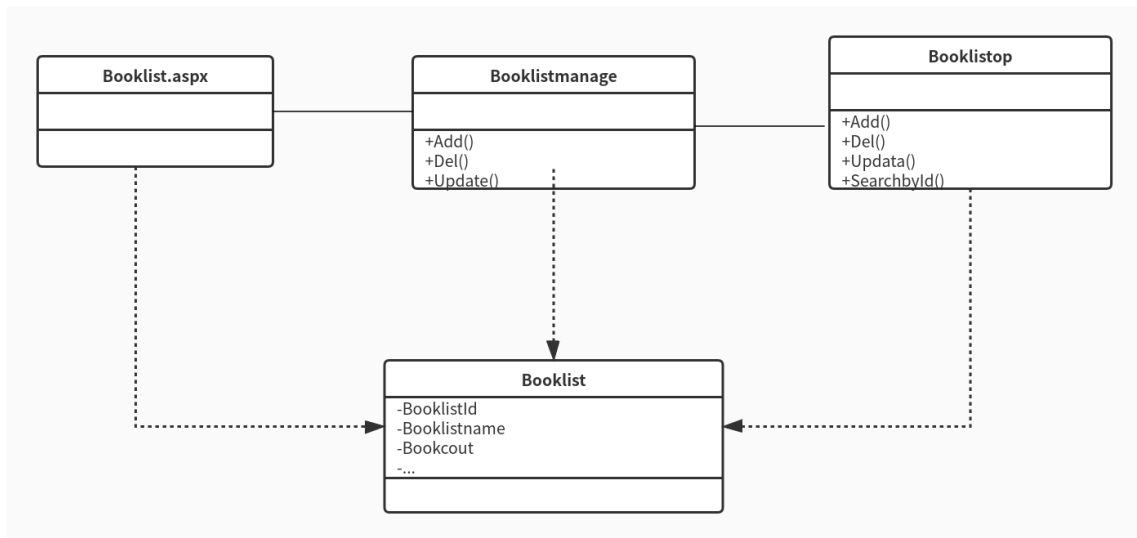


图 4.9 书单管理类图

说明如下：

- (1) 边界类 **Booklist.aspx** 对应于表示层, 用于提供客户编辑的界面;
- (2) 控制类 **Booklistmanage** 对应于业务逻辑层;
- (3) 实体操作类 **Booklistop** 对应于数据访问层;
- (4) 实体信息类 **Booklist** 的作用是保存书架信息, 方便其他类调用。

## 4.5 顺序图

### 4.5.1 登录顺序图

登录的过程就是一个从表示层中的登录界面开始的过程。用户向登录管理类发出登录请求这个事件, 登录管理对象则调用数据访问对象中的登录验证方法来验证本次登录请求是否成功。数据访问类中的登录验证方法是根据核对用户名和密码来完成。如果验证成功, 则从数据访问对象向登录管理对象返回登录验证通过的信息。登录管理对象同样把登录成功的信息向表示层界面返回。如果登录没有成功, 同样也会返回登录出错的信息, 登录顺序图如图4.10所示。

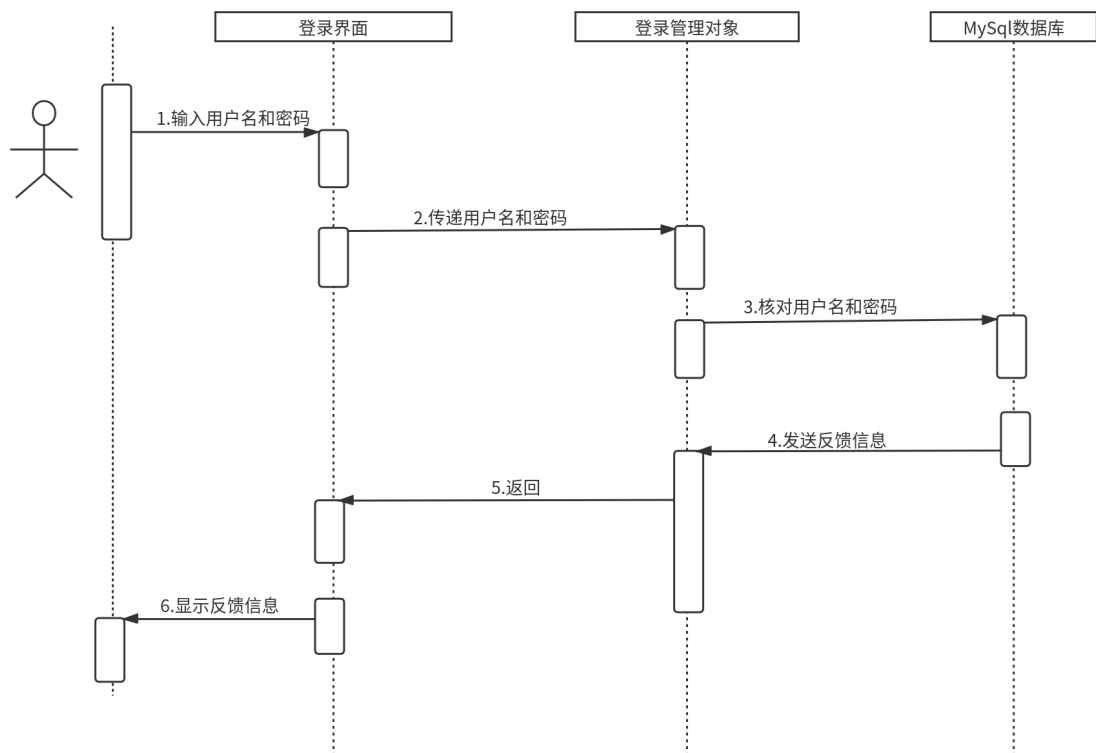


图 4.10 登录顺序图

#### 4.5.2 小说加入书单顺序图

在将小说加入书单时，首先向小说管理界面发出加入书单的请求，系统显示加入书单界面，等待用户输入对书本的评价，对书本的评分等信息。属于控制类的小说管理模块对用户提交的将小说加入书单的申请信息向属于小说管理类的书单管理对象传递。书单管理对象调用数据库模块中的书单管理类中的添加方法向用户书单信息中更新书单信息，将更新后的当前状态给用户，小说加入书单顺序图如图4.11所示。

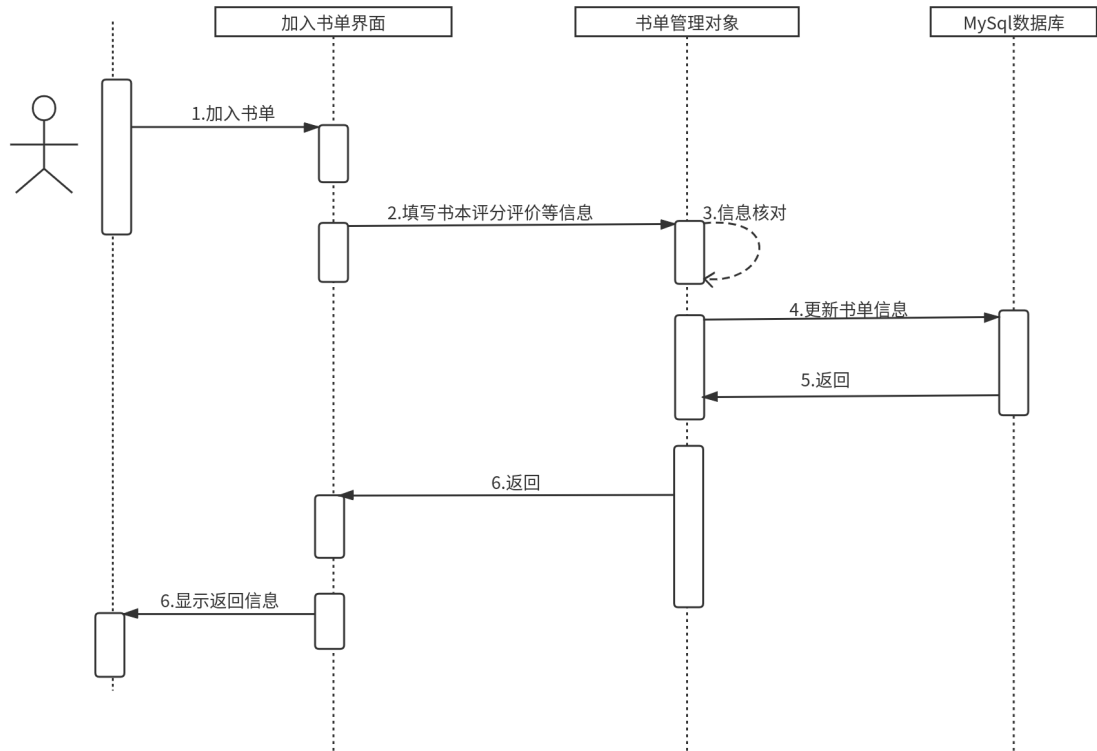


图 4.11 加入书单顺序图

### 4.5.3 编辑书单顺序图

在编辑书单中小说信息时，首先向书单管理界面发出编辑书单小说信息的请求，系统显示编辑书单小说信息界面，等待用户修改对书本的评价，对书本的评分等信息。属于控制类的小说管理模块对用户提交的编辑书单信息申请向属于小说操作类的书单操作对象传递。书单操作对象调用数据库模块中的书单管理类中的编辑方法对用户书单信息中更新书单信息，将更新后的当前状态给用户，编辑书单顺序图如图4.12所示。

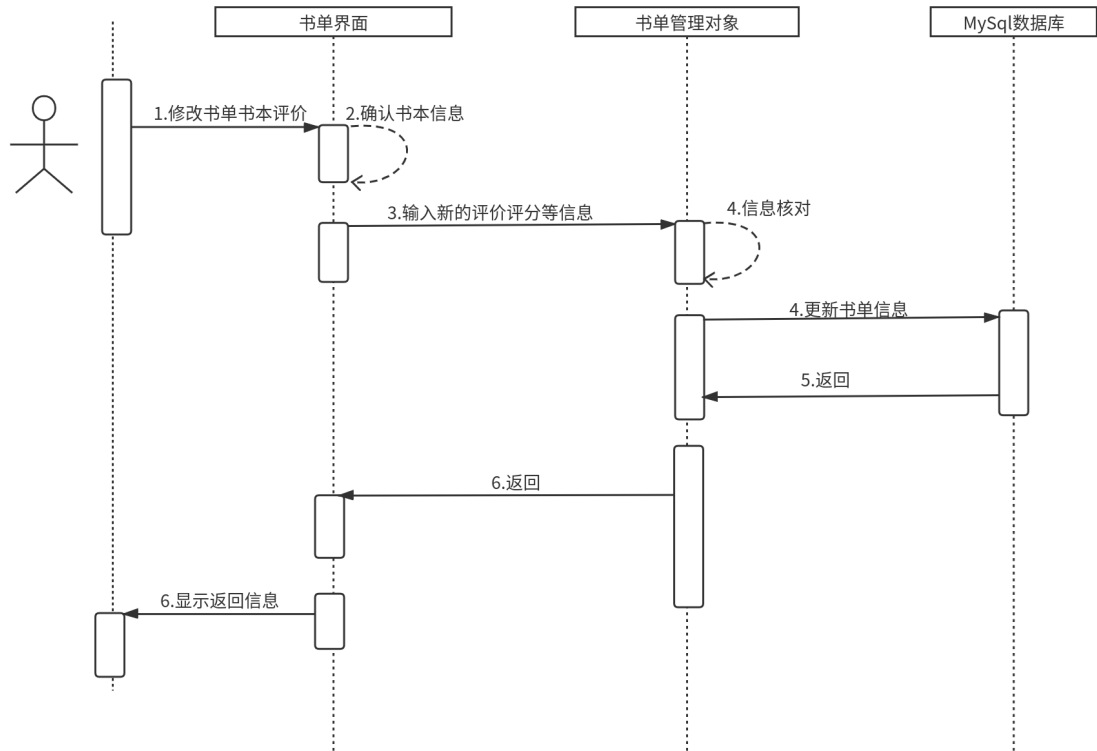


图 4.12 编辑书单内书本顺序图

#### 4.5.4 删除书单顺序图

在删除书单中小说信息时，首先向书单管理界面发出删除书单小说信息的请求，系统显示书单中所有小说信息列表，等待用户输入删除小说信息。属于控制类的小说管理模块对用户提交的删除书单小说信息申请向属于小说操作类的书单操作对象传递。书单操作对象调用数据库模块中的书单管理类中的删除方法对用户书单信息中更新书单信息，将更新后的当前状态给用户，编辑书单顺序图如图4.13所示。

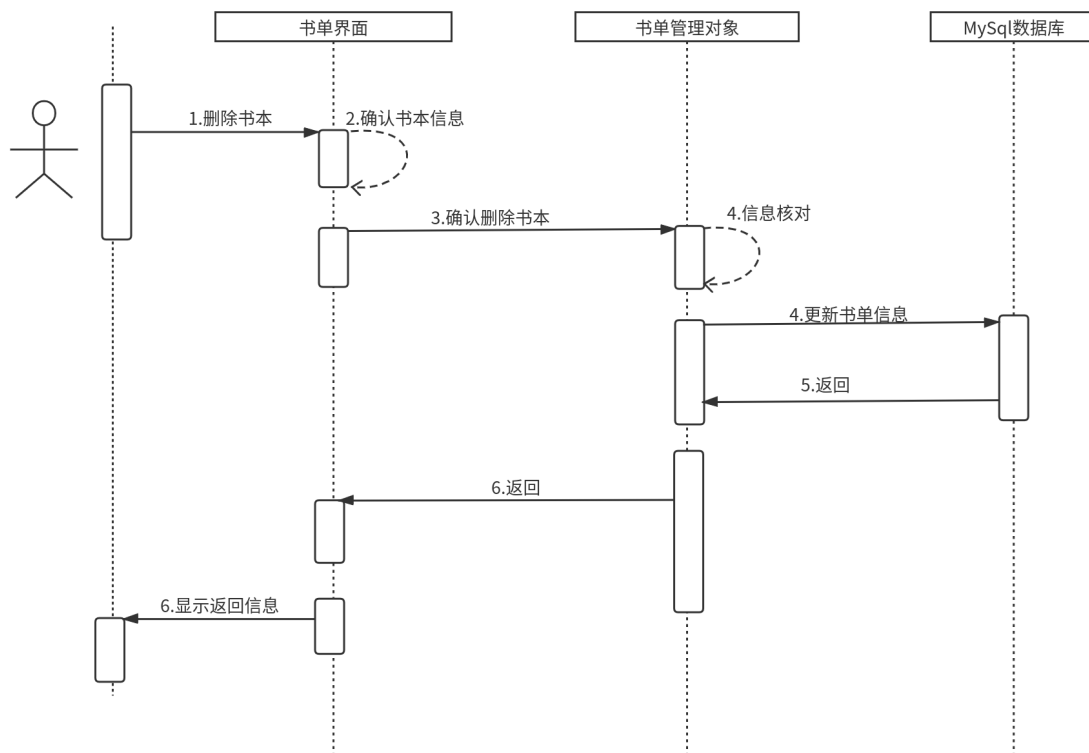


图 4.13 删除书单内书本顺序图

## 4.6 数据库设计

数据库在信息系统开发中的作用是无法替代的。与本地数据库相比,在采用 B/S 模式开发的系统中,使用的数据库被称为 Web 数据库。数据库设计的主要任务是如何从现实世界中建立起数据库的概念模型,以及数据库中应该包含哪些数据表。这就需要在数据库设计过程中及时发现存在于现实世界中的各种客观实体。然后确定这些实体之间是否存在某种直接或间接的联系。一旦确定实体后,就可以借助于 E-R 图来描述它们之间的关系。

### 4.6.1 概念设计

数据库的概念设计就是要建立 E-R 模型。概念设计的目的就是要找出系统中可能存在的实体及其直接的关系,以便建立合适的数据表结构。本系统部分 E-R 图如图 4.14 所示。

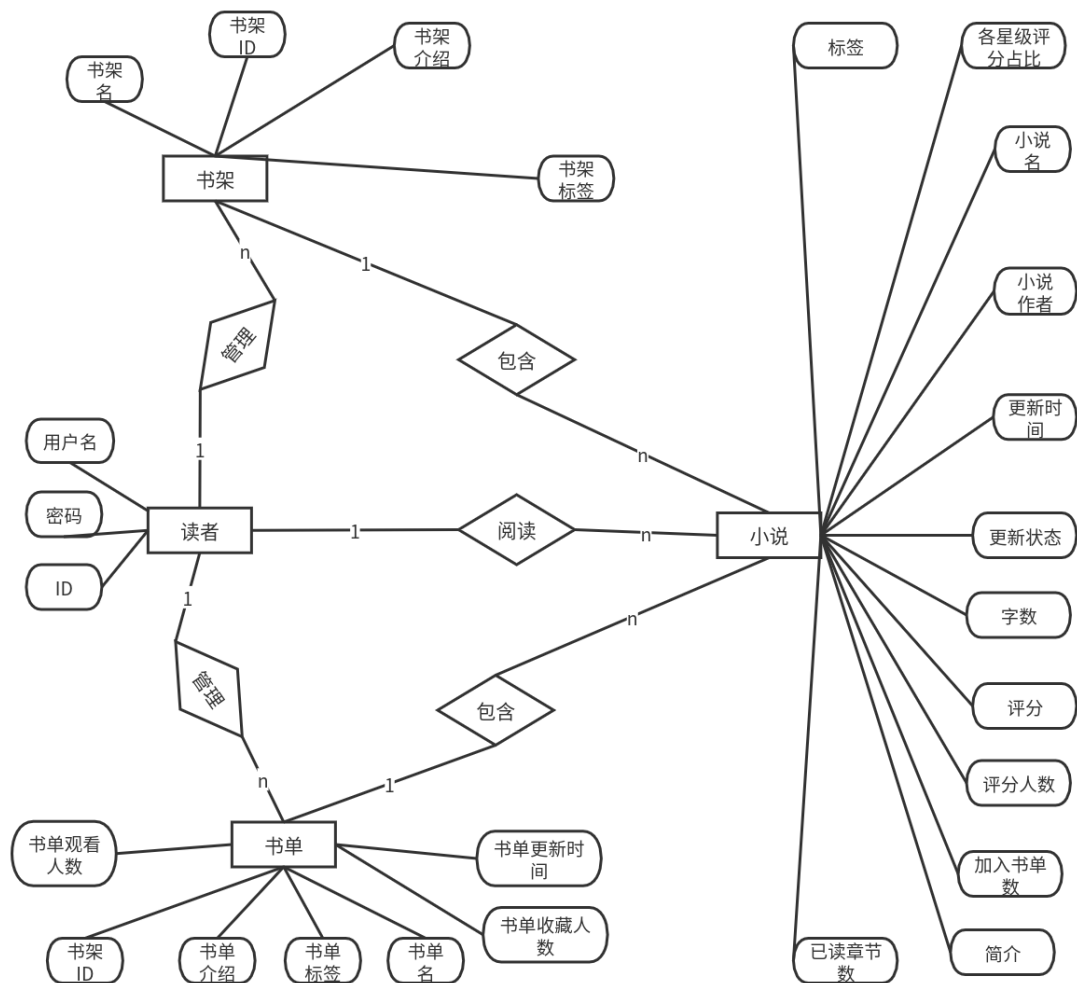


图 4.14 系统部分 E-R 图

#### 4.6.2 数据库表设计

根据对数据库进行需求分析, 可以得出系统中的主要数据表如下:

登录用户表, 用于保存登录用户信息包括用户密码, 用户名, 用户 ID 等信息。表结构如表 4-1 所示。

表 4.1 MySql 中的用户登录表

字段名	字段格式	中文说明
UserId	varchar(100) NOT NULL PRIMARY KEY	主键，用于唯一识别用户
Username	varchar(100) NOT NULL	用户名
Password	varchar(100) NOT NULL	密码
UserType	varchar(100) DEFAULT NULL	用户类型

1、小说信息表,用于保存小说的基本信息包括小说作者名,小说名,小说评分,更新状态,评分人数等信息。表结构如表4.2所示。

表 4.2 存储在 MySql 中的小说信息表

字段名	字段格式	中文说明
BookId	varchar(100) NOT NULL PRIMARY KEY	主键，用于唯一识别书本
BookName	varchar(100) NOT NULL	书本名字
BookAuthor	varchar(100) NOT NULL	书本作者
BookWordCount	varchar(100) DEFAULT NULL	书本字数
Bookstatus	varchar(40) DEFAULT NULL	书本状态
BookUpdata	varchar(100) DEFAULT NULL	书本距今最短更新时间
BookScore	float(3,1) DEFAULT NULL	书本评分
BookScoreCount	int(7) DEFAULT NULL	书本评分人数
OneStarRate	float(3,2) DEFAULT NULL	一星评分占比
TwoStarRate	float(3,2) DEFAULT NULL	二星评分占比
ThreeStarRate	float(3,2) DEFAULT NULL	三星评分占比
FourStarRate	float(3,2) DEFAULT NULL	四星评分占比
FiveStarRate	float(3,2) DEFAULT NULL	五星评分占比
BookClass	varchar(400) DEFAULT NULL	书本标签
AddListCount	int(7) DEFAULT NULL	加入书单数

2、书架信息表,用于保存书架的信息包括书架名,书架标签,书架介绍等信息。表结构如表4.3所示。



表 4.3 存储在 MySql 中的书架信息表

字段名	字段格式	中文说明
BookshelfId	varchar(100) NOT NULL PRIMARY KEY	主键，用于唯一识别书架
UserId	varchar(100) NOT NULL	用于唯一识别读者
BookshelfName	varchar(100) NOT NULL	书架名字
BookshelfIntro	varchar(10000) NOT NULL	书架介绍
BookshelfLabel	varchar(1000) NOT NULL	书架标签
BookId	varchar(10000) NOT NULL	书本 Id

3、书单信息表, 用于保存书单的信息, 如书单名字, 书单书本数目, 书单最近更新时间, 书单观看人数, 书单收藏人数等信息。表结构如表4.4所示。

表 4.4 存储在 MySql 中的书单信息表

字段名	字段格式	中文说明
BooklistId	varchar(100) NOT NULL PRIMARY KEY	主键，用于唯一识别书单
UserId	varchar(100) NOT NULL	用于唯一识别读者
BooklistName	varchar(100) NOT NULL	书单名字
BookCount	int(7) DEFAULT NULL	书本数目
BooklistUpdata	varchar(100) DEFAULT NULL	书单距今最短更新时间
BooklistviewCount	varchar(100) DEFAULT NULL	书单观看人数
BooklistCollCount	varchar(100) DEFAULT NULL	书单收藏人数
BooklistIntro	varchar(10000) NOT NULL	书单介绍
BooklistLabel	varchar(1000) NOT NULL	书单标签
BookId	varchar(10000) NOT NULL	书本 Id

4、小说样本集, 用于随机森林训练的小说样本的基本信息。包括小说字数, 小说评分, 评分人数, 加入书单数, 各星级评分占比等信息。表结构如表4.5所示

表 4.5 存储在 MySQL 中的小说样本表

存储名	存储格式	解释
BookId	varchar(100) NOT NULL PRIMARY KEY	主键，用于唯一识别书本
BookWordCount	float(3,2) DEFAULT NULL	书本字数
BookScore	float(3,1) DEFAULT NULL	书本评分
BookScoreCount	int(7) DEFAULT NULL	书本评分人数
AddListCount	int(7) DEFAULT NULL	加入书单数
OneStarRate	float(3,2) DEFAULT NULL	一星评分占比
TwoStarRate	float(3,2) DEFAULT NULL	二星评分占比
ThreeStarRate	float(3,2) DEFAULT NULL	三星评分占比
FourStarRate	float(3,2) DEFAULT NULL	四星评分占比
FiveStarRate	float(3,2) DEFAULT NULL	五星评分占比
Mode	int(7) DEFAULT NULL	各星级评分众数
Median	int(7) DEFAULT NULL	各星级评分中位数
StandardDeviation	float(3,2) DEFAULT NULL	各星级评分标准偏差
Quartile	float(3,2) DEFAULT NULL	各星级评分四分位数
DispersionCoefficient	float(3,2) DEFAULT NULL	各星级评分离散系数
Kurtosis	float(3,2) DEFAULT NULL	各星级评分峰度
Class	varchar(100) NOT NULL	书本类别

## 第五章 系统实现

### 5.1 个性化推荐模块

#### 5.1.1 数据集预处理

下载 MySQL 保存的小说数据集中最近加入书架的小说 180 本，对每本小说人为添加特征，小说章节数目，小说已看章节数目。移除部分明显对小说分类无影响的特征，将新的数据集重新加入 MySQL 数据库。

其表格如表5.1所示:

表 5.1 第一次处理后的存储的书本信息

存储名	存储格式	解释
BookId	varchar(100) NOT NULL PRIMARY KEY	主键，用于唯一识别书本
BookWordCount	varchar(100) DEFAULT NULL	书本字数
BookScore	float(3,1) DEFAULT NULL	书本评分
BookScoreCount	int(7) DEFAULT NULL	书本评分人数
OneStarRate	float(3,2) DEFAULT NULL	一星评分占比
TwoStarRate	float(3,2) DEFAULT NULL	二星评分占比
ThreeStarRate	float(3,2) DEFAULT NULL	三星评分占比
FourStarRate	float(3,2) DEFAULT NULL	四星评分占比
FiveStarRate	float(3,2) DEFAULT NULL	五星评分占比
AddListCount	int(7) DEFAULT NULL	加入书单数
BookChapterCount	int(7) DEFAULT NULL	书本章节总数
BookChapterHaveRead	int(7) DEFAULT NULL	书本已阅读章节数目

对于数据集进行预处理

#### 1、计算类型

小说喜好程度以通过章节总数与已阅读章节数目得出，其计算公式如下：

$$S = \frac{X_1}{X_2} \cdot \frac{2}{\pi} \cdot \arctan(0.04 \cdot X_2) \quad (5.1)$$

其中，X 是已读章节，Y 是章节总数，S 为个人根据阅读进度和书本大小对这本书的评分。对于类别评分，根据根据聚类技术以评分 0.5547 为分割点将小说划分为两类。将类别划分为两类，其代码为：

---

```
int Calculation_Class(int BookChapter,int BookChapterHaveRead)
{
    int flag = 0;
    double pai = 3.1415926;
    double x_tmp = (double)BookChapterHaveRead / (double)BookChapter;
    double y_tmp = 2/pai * atan(0.04* BookChapter);
    double z_tmp = x_tmp * y_tmp;
    if(z_tmp > 0.5547)
    {
        flag = 1;
    }
    return flag;
}
```

---

## 2、计算峰度

计算一星评分占比，二星评分占比,..., 五星评分占比的峰度，峰度衡量数据分布的平坦度，在本系统中它描述的是各星级评分人数占比的波动程度。

其公式如下：

$$K = \frac{\frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^4}{\left[ \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2 \right]^2} \quad (5.2)$$

其中 K 是峰度， $X_i$  表示第 i 个数值， $\bar{X}$  为平均值，n 为采样数量  
相应代码如下：

```
float Calculation_Kurtosis(float StarRate[])//峰度
{
    float K = 0;
    float molecular = 0;
    float Denominator = Calculation_StandardDeviation(StarRate);
    float Average_tmp = Calculation_Average(StarRate);
    for(int i = 0; i < 5; i++)
    {
        float Num_tmp = (StarRate[i] - Average_tmp);
        molecular += (Num_tmp * Num_tmp * Num_tmp * Num_tmp);
    }
    Denominator = (Denominator * Denominator * Denominator * Denominator);
    if(Average_tmp < 0.0001)
        K = 0;
    else
        K = molecular / Denominator;
    return K;
}
```

---

### 3、计算四分位数

计算一星评分占比，二星评分占比,...,五星评分占比的四分位数。

相应代码如下：

---

```

float Calculation_Quartile(float StarRate[])//四分位数
{
    float SumRate_tmp = 0;
    int Q1 = 2;
    int Q3 = 10;
    for(int i = 0; i < 5; i++)
    {
        float Score_tmp = (i + 1) * 2;
        if(SumRate_tmp < 0.25)
        {
            SumRate_tmp += StarRate[i];
            Q1 = Score_tmp;
        }
    }
    SumRate_tmp = 0;
    for(int i = 4; i >= 0; i--)
    {
        float Score_tmp = (i + 1) * 2;
        if(SumRate_tmp < 0.25)
        {
            SumRate_tmp += StarRate[i];
            Q3 = Score_tmp;
        }
    }
    return Q3 - Q1;
}

```

---

#### 4、计算标准偏差

计算一星评分占比,二星评分占比,...,五星评分占比的标准偏差,标准偏差反映一组测量数据离散程度,在本系统中它描述的是各星级评分人数占比的离散程度。

标准偏差公式如下:

$$\mu = \sqrt{\frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n}} \quad (5.3)$$

其中  $\mu$  是标准偏差,  $X_i$  指某个样本取值,  $\bar{X}$  指样本平均值。

相应代码如下:

---

```
float Calculation_StandardDeviation(float StarRate[])//标准偏差
{
    float Average_tmp = 0;
    float StandardDeviation = 0;
    for(int i = 0; i < 5; i++)
    {
        float Score_tmp = (i + 1) * 2;
        Average_tmp += (StarRate[i] * Score_tmp);
    }
    for(int i = 0; i < 5; i++)
    {
        float Score_tmp = (i + 1) * 2;
        StandardDeviation += ((Score_tmp - Average_tmp) * (Score_tmp - Average_tmp) *
                               StarRate[i]);
    }
    if(StandardDeviation <= 0.0001)
        StandardDeviation = 0;
    else
        StandardDeviation = sqrt(StandardDeviation);
    return StandardDeviation;
}
```

---

## 5、计算中位数

计算一星评分占比, 二星评分占比,..., 五星评分占比的中位数。

相应代码如下:

---

```
float Calculation_Median(float StarRate[])//计算中位数
{
float SumRate_tmp = 0;
int SumRateflag_tmp = 0;
for(int i = 0; i < 5; i++)
{
if(SumRate_tmp < 0.5)
{
SumRate_tmp += StarRate[i];
SumRateflag_tmp = i;
}
}
return (SumRateflag_tmp+1) * 2;
}
```

---

## 6、计算众数

计算一星评分占比, 二星评分占比,..., 五星评分占比的众数。

相应代码如下:

---

```
float Calculation_Mode(float StarRate[])
{
float MaxRate_tmp = 0;
int MaxRateflag_tmp = 0;
for(int i = 0; i < 5; i++)
{
if(StarRate[i] >= MaxRate_tmp)
{
MaxRate_tmp = StarRate[i];
MaxRateflag_tmp = i;
}
}
return (MaxRateflag_tmp+1) * 2;
}
```

---

## 6、计算离散系数

计算一星评分占比, 二星评分占比,..., 五星评分占比的离散系数。



离散系数公式:

$$c_v = \frac{\sigma}{\mu} \quad (5.4)$$

其中  $c_v$  为离散系数  $\sigma$  代表标准方差,  $\mu$  代表平均值。

最后将计算得到的特征信息存入 MySQL 数据库, 其内容如下表所示

表 5.2 预处理后的存储的书本信息

存储名	存储格式	解释
BookId	varchar(100) NOT NULL PRIMARY KEY	主键, 用于唯一识别书本
BookWordCount	float(3,2) DEFAULT NULL	书本字数
BookScore	float(3,1) DEFAULT NULL	书本评分
BookScoreCount	int(7) DEFAULT NULL	书本评分人数
AddListCount	int(7) DEFAULT NULL	加入书单数
OneStarRate	float(3,2) DEFAULT NULL	一星评分占比
TwoStarRate	float(3,2) DEFAULT NULL	二星评分占比
ThreeStarRate	float(3,2) DEFAULT NULL	三星评分占比
FourStarRate	float(3,2) DEFAULT NULL	四星评分占比
FiveStarRate	float(3,2) DEFAULT NULL	五星评分占比
Mode	int(7) DEFAULT NULL	各星级评分众数
Median	int(7) DEFAULT NULL	各星级评分中位数
StandardDeviation	float(3,2) DEFAULT NULL	各星级评分标准偏差
Quartile	float(3,2) DEFAULT NULL	各星级评分四分位数
DispersionCoefficient	float(3,2) DEFAULT NULL	各星级评分离散系数
Kurtosis	float(3,2) DEFAULT NULL	各星级评分峰度
Class	varchar(100) NOT NULL	书本类别

### 5.1.2 特征值重要性分析

在生成 CART 决策树时, 我们以基尼指数来衡量一个特征的划分能力, 从而选择出最佳划分特征做为结点进行分裂, 换言之, 其实在随机树生成的过程中就一直在对每个特征的划分能力进行排序, 所以在随机森林选择最优特征子

集时，我们可以先生成一颗完整的 CART 决策树，在生成过程中，记录每个特征划分时的基尼指数，进行加权线性累加，最后得到的每个特征的特征得分被称为 Gini Importance<sup>[38]</sup>。

根据 Gini Importance 对每个特征划分能力进行排序，权重为划分后的样本数量和总样本数量的比值。其每个特征的划分能力排序如表5.3所示：

表 5.3 特征 Gini Importance

特征名	Gini Importance	编码
BookScoreCount	0.333878	101
BookWordCount	0.303971	102
AddListCount	0.297159	103
Kurtosis	0.263036	104
StandardDeviation	0.261233	105
ThreeStarRate	0.259227	106
FiveStarRate	0.252293	107
FourStarRate	0.247871	108
BookScore	0.246668	109
DispersionCoefficient	0.227262	110
OneStarRate	0.218227	111
TwoStarRate	0.176709	112
Median	0.126521	113
Mode	0.124782	114
Quartile	0.0875916	115

其对应图形如图5.1所示：

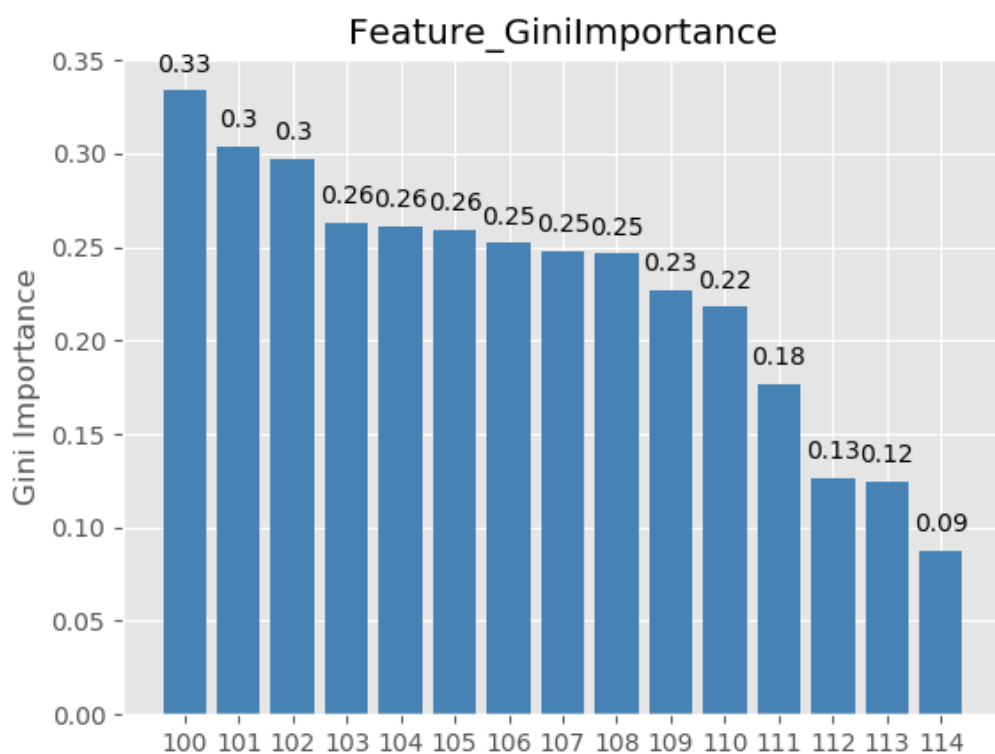


图 5.1 Features GiniImportance

由上图可以看出, BookScoreCount, BookWordCount, AddListCount 等特征具有较好的分类能力, Quartile 特征具有较差的分类能力。

### 5.1.3 BootStrapping 自取样

在生成每棵子树前, 对样本集进行自取样生成训练样本集和测试样本集, 其中测试样本集在之后可以做为包外数据对该子树进行 OOB 误差估计。

代码如下:

---

```

void Bootstrap_Sampling(vector<string> Atts_ChildData, map<string,
    vector<string>> TableData_Child, vector<int>& Numsign_TrainData, vector<int>&
    Numsign_TestData, map<string, vector<string>>& Train_TableData, map<string,
    vector<string>>& Test_TableData)
{
    int NumSample = (int)TableData_Child[Atts_ChildData[0]].size(); //样本数量
    map<int, int> Numsign_Data;
    //sleep(3);
    //srand((int)time(0)); //只需要一个种子就好了
    for(int i = 0; i < NumSample; i++) //自助取样
    {
        int Num_random = (rand() % NumSample);
        Numsign_Data[Num_random] ++;
    }
    for(int i = 0; i < NumSample; i++) //将Train标号和Test标号分别存入
    {
        if(Numsign_Data[i] > 0)
            Numsign_TrainData.push_back(i);
        else
            Numsign_TestData.push_back(i);
    }
    for(int i = 0; i < Numsign_TrainData.size(); i++) //存入Train样本集
    for(int k = 0; k < Atts_ChildData.size(); k++)
        Train_TableData[Atts_ChildData[k]].push_back(TableData_Child[Atts_ChildData[k]][Numsign_TrainData[i]]);
    for(int i = 0; i < Numsign_TestData.size(); i++) //存入Test样本集
    for(int k = 0; k < Atts_ChildData.size(); k++)
        Test_TableData[Atts_ChildData[k]].push_back(TableData_Child[Atts_ChildData[k]][Numsign_TestData[i]]);
    }

```

---

#### 5.1.4 构建随机森林模型

随机森林的参数只需要  $N$  和  $K$ ，其中  $N$  是子树的数目， $K$  是决策树在生成结点时从特征集中抽取的候选特征数目。

其关键代码如下所示：

```

int main()
{
    //输入
    string NameDataFile = "Treedata.txt";
    ReadSampleData(NameDataFile,AttsData_tmp,TableData_tmp);//读取样本文件
    srand((unsigned int)time(0));
    int N =799;
    int K = 3;
    Generate_ChildTreeSampleFile(N,FeatureSet_tmp2,DataSet_tmp,Numsign_RanForestTrain,data_TableRanForest);
    Generate_ChildTree(K,N,FeatureSet_tmp2,data_TableRanForest,RanForest);//创建子树生成随机森林模型
    double ErrorRate_OOB =
        Check_RandomForest(N,DataSet_tmp);//对随机森林模型进行OOB误差估计
    cout << "ErrorRate_OOB: " << ErrorRate_OOB << endl;
    return 0;
}

```

### 5.1.5 个性化推荐

#### 1、每日推荐界面

每日推荐界面如图所示。



图 5.2 每日推荐界面

#### 2、业务逻辑实现

每日推荐会调用建立好的随机森林模型对数据库中不在书架上的小说进行分类，一直到分类为好的小说数目达到 10，将小说信息提交到 web 界面显示出来。

## 5.2 登录注销模块

### 1、登录界面

登录界面是进入网络小说个性化推荐系统的第一道安全防护，用户在登录本系统时需要通过用户名和密码的校验，登录界面如图5.3所示，登录后的管理界面如图5.4所示

The diagram illustrates a login interface. At the top, the title "登录" (Login) is displayed. Below it, there are two input fields: the first is labeled "用户名" (Username) and contains the text "daiqiongfang"; the second is labeled "登录密码" (Login Password) and contains a series of dots. To the right of the password field is a red button labeled "CFht". Below these fields is a white button labeled "验证码" (Verification Code). At the bottom, there is a large orange button labeled "登录" (Login).

图 5.3 登录界面



图 5.4 管理主界面

## 2、业务逻辑实现

当用户输入登录信息后,在客户端首先进行非空验证,具体由相应的 Javascript 客户端脚本代码实现。用户点击“确定”按钮,触发了该按钮的 Click 事件,从而由表示层对象调用业务逻辑层对象 LoginUserBLL 的验证登录信息方法 IsLogin() 函数的功能。同时也要判断当前登录用户角色是操作员还是管理员。

## 5.3 书架管理模块

### 5.3.1 新建书架功能

#### 1、界面的实现

新建书架界面如图 5.5 所示。

个性化网络小说推荐系统

欢迎 daiqf 今天是:2020年6月17日

菜单

- 书架管理
  - 新建书架
  - 修改书架
  - 删除书架
- 书单管理
  - 新建书单
  - 修改书单
  - 删除书单
- 检索管理
  - 搜索小说
  - 搜索书单
- 个性化推荐
  - 每日推荐
- 个人管理
  - 密码修改
  - 信息编辑
- 退出
  - 注销

新建书架界面 (带\*为必填项)

书架主人: daiqf

\* 书架标签: 请选择书架标签▼

\* 书架名:   
书架名最多可以输入20个字符;

\* 书架介绍: 

对书架进行简单介绍!

书架介绍最多可以输入200个汉字(含标点和空格);

确定

取消

图 5.5 新建书架界面

## 2、业务逻辑实现

新建书架功能由书架管理类 BookshelfMageBLL 中的添加方法 addbookshelf() 完成。在添加书架时, 需要对那些标注为必填项的区域中一定要输入相关信息, 否则会提示出错。对必填项的判断可以通过客户端验证实现, 这样可以减少服务器端的验证处理时间, 减轻服务器端的压力。填完后点击“确定”按钮向系统提交新的书架信息。系统后台通过执行 SQL 语句中的 INSERT 语句就可以保存新的书架信息。

### 5.3.2 修改书架功能

#### 1、界面实现

修改书架界面如图 5.6 所示。

51



菜单

▪ 书架管理

- 新建书架
- 修改书架
- 删除书架

▪ 书单管理

- 新建书单
- 修改书单
- 删除书单

▪ 检索管理

- 搜索小说
- 搜索书单

▪ 个性化推荐

- 每日推荐

▪ 个人管理

- 密码修改
- 信息编辑

▪ 退出

- 注销

个性化网络小说推荐系统

欢迎 daiqf 今天是:2020年6月17日

修改书架界面 (带\*为必填项)

书架主人: daiqf

\*旧书架名: 

请选择修改书架

\*书架标签: 

请选择书架标签

\*新书架名: 

书架名最多可以输入20个字符;

\*书架介绍: 

书架介绍最多可以输入200个汉字(含标点和空格);

对书架进行简单介绍!

确定

取消

图 5.6 修改书架界面

## 2、业务逻辑实现

在进入修改书架界面时,选择要修改的书架,然后输入新的书架名,介绍,标签。修改书架功能由书架管理模块 BookshelfManager 中的修改方法 updatebookshelf() 完成。在用户修改书架信息时,需要对那些标注为必填项的区域中一定要输入相关信息,否则会提示出错。对必填项的判断可以通过客户端验证实现,这样可以减少服务器端的验证处理时间,减轻服务器端的压力。填完后点击“确定”按钮向系统提交新的书架信息。系统后台通过执行 SQL 语句中的 UPDATE 语句就可以保存新的书架信息。

### 5.3.3 删除书架功能

#### 1、界面实现

删除书架界面如图 5.7 所示。



图 5.7 删除书架界面

## 2、业务逻辑实现

在进入删除书架界面时，选择要删除的书架，然后点击确定。删除书架功能由书架管理模块 BookshelfImageBLL 中的修改方法 deletebookshelf() 完成。填完后点击“确定”按钮向系统提交删除书架申请。系统后台通过执行 SQL 语句中的 DELETE 语句就可以删除书架信息。

## 5.4 书单管理模块

### 5.4.1 新建书单功能

#### 1、界面的实现

添加书单界面如图 5.8 所示。

个性化网络小说推荐系统

欢迎 daiqf 今天是:2020年6月17日

菜单

- 书架管理
  - 新建书架
  - 修改书架
  - 删除书架
- 书单管理
  - 新建书单
  - 修改书单
  - 删除书单
- 检索管理
  - 搜索小说
  - 搜索书单
- 个性化推荐
  - 每日推荐
- 个人管理
  - 密码修改
  - 信息编辑
- 退出
  - 注销

新建书单界面 (带\*为必填项)

书单主人: daiqf

\* 书架标签: 

请选择书单标签

\* 书架名: 

书单名最多可以输入20个字符;

\* 书架介绍: 

对书单进行简单介绍!

书单介绍最多可以输入200个汉字(含标点和空格);

确定

取消

图 5.8 新建书单界面

## 2、业务逻辑实现

添加书单功能由书单管理类 BooklistMageBLL 中的添加方法 addbooklist() 完成。在添加书单时, 需要对那些标注为必填项的区域中一定要输入相关信息, 否则会提示出错。对必填项的判断可以通过客户端验证实现, 这样可以减少服务器端的验证处理时间, 减轻服务器端的压力。填完后点击“确定”按钮向系统提交新的书单信息。系统后台通过执行 SQL 语句中的 INSERT 语句就可以保存新的书架信息。

### 5.4.2 修改书单功能

#### 1、界面实现

修改书单界面如图 5.9 所示。

54

菜单

▪ 书架管理

▪ 新建书架

▪ 修改书架

▪ 删除书架

▪ 书单管理

▪ 新建书单

▪ 修改书单

▪ 删除书单

▪ 检索管理

▪ 搜索小说

▪ 搜索书单

▪ 个性化推荐

▪ 每日推荐

▪ 个人管理

▪ 密码修改

▪ 信息编辑

▪ 退出

▪ 注销

修改书单界面 (带\*为必填项)

书单主人: daiqf

\*旧书名: 请选择修改书单

\*书单标签: 请选择书单标签

\*新书名:

书单名最多可以输入20个字符;

\*书单介绍:

对书单进行简单介绍!

书单介绍最多可以输入200个汉字(含标点和空格);

确定 取消

个性化网络小说推荐系统

欢迎 daiqf 今天是:2020年6月17日

图 5.9 修改书单界面

## 2、业务逻辑实现

在进入修改书单界面时,选择要修改的书单,然后输入新的书单名,介绍,标签。修改书单功能由书架管理模块 BooklistMageBLL 中的修改方法 updatebooklist() 完成。在用户修改书单信息时,需要对那些标注为必填项的区域中一定要输入相关信息,否则会提示出错。对必填项的判断可以通过客户端验证实现,这样可以减少服务器端的验证处理时间,减轻服务器端的压力。填完后点击“确定”按钮向系统提交新的书单信息。系统后台通过执行 SQL 语句中的 UPDATE 语句就可以保存新的书单信息。

### 5.4.3 删除书单功能

#### 1、界面实现

删除书单界面如图 5.10 所示。



图 5.10 删除书单界面

## 2、业务逻辑实现

在进入删除书单界面时,选择要删除的书架,然后点击确定。删除书单功能由书架管理模块 BooklistMageBLL 中的修改方法 deletebooklist() 完成。填完后点击“确定”按钮向系统提交删除书单申请。系统后台通过执行 SQL 语句中的 DELETE 语句就可以删除书单信息。

## 5.5 本章小结

本章完成了系统主要模块的功能实现,具体对界面实现、业务逻辑实现和关键代码等方面进行了阐述。

## 第六章 系统测试

### 6.1 测试目的

软件系统开发结束后必须要经过严格的系统测试。测试目的简单来说就是要找出各种各样的问题, 毕竟“人无完人, 孰能无过”。开发人员进行前期开发过程中, 一定会存在因设计上的缺陷或逻辑上的不严谨而导致的各种问题。而有些问题并不是在一开始就能发现并加以排除的。软件开发人员在项目的开发过程中, 除了在设计、分析上会有考虑不周的情况之外, 在编程时也会因为或这或那的原因犯错。虽然说错误在所难免, 但真正达标的系统是不允许出现影响用户使用的。因此, 要让开发的系统能很好的为客户服务, 就需要在软件系统交付之前进行各种纠错活动, 避免出现不好的后果。测试人员承担了项目软件开发的最后一道关口, 其责任重大, 是实现对用户负责的重要保证。

### 6.2 测试方法

系统测试方法主要有黑盒测试和白盒测试。黑盒测试也叫做功能测试, 主要对测试系统各个功能的实现情况; 白盒测试也被称为代码测试, 主要是程序设计人员对系统设计的结构逻辑进行测试。这两种测试方法还是有着很大的区别。黑盒测试过程相对简单, 对测试人员的要求也不高。测试人员只需要知道被测试模块的输入输出接口就可以进行功能测试。或者说, 只需要知道输入什么样的数据, 系统会产生什么样的输出结果。因此, 功能测试很直观也很形象, 测试的结果往往一目了然。白盒测试则有所不同, 它要求测试人员对系统结构具有深入的了解。因为系统的有些错误可能是因为软件的逻辑问题引起的。而对软件的逻辑问题进行排查需要花费较多的精力和时间。尤其对于那些拥有复杂结构的大型化软件系统其白盒测试过程会比较漫长。除了针对功能和代码方面的测试外, 本论文还针对系统的各个功能进行性能测试。虽然性能方面的测试不是重点, 但作为软件系统整体而言, 功能需求和性能需求同样是需要关注的问题, 两者之间缺一不可。

## 6.3 功能测试

### 6.3.1 系统登录功能测试

1、系统登录测试主要检测系统用户登录时出现的各种状况，以及对于异常事件流的处理。系统登录测试结果如表6.1所示：

表 6.1 系统登录测试用例

用例编号	UT-01		
功能描述	用户登录系统		
用例目的	测试用户登录功能		
前置条件	用户拥有此系统登录权限		
输入动作	数据	期望的输出	实际输出
输入正确的用户名和密码	daiqf,123456	正确登录系统	正确登录系统
输入错误的用户名	daiqff	不能登录	提示：用户名或密码错误
输入错误的密码	121212	不能登录	提示：用户名或密码错误
用户名为空		系统提示输入用户名	提示：输入用户名
密码为空		系统提示输入密码	提示：输入密码
用户名和密码都为空		系统提示输入用户名和密码	提示：输入用户名和密码

### 6.3.2 书架管理功能测试

书架管理功能测试，主要测试是否可以按照用户要求进行加入书架，删除书本，新建书架，修改书架，删除书架等功能。以及对于一些异常事件的处理，比如，新建书架未输入书架名，书架名输入字数超过限制等。

1、新建书架功能测试结果如表6.2所示：

表 6.2 新建书架功能测试用例

用例编号	UT-02		
功能描述	新建书架		
用例目的	测试是否能够正确新建书架		
前置条件	用户登录系统，进入新建书架界面		
输入动作	数据	期望的输出	实际输出
输入正确的书架名	粮草集	可以接受	正常
输入书架名为空		系统提示输入书架名	提示：书架名不能为空
输入书架名超过 20 个汉字	输入超过 20 个汉字的书架名	系统提示书架名太长	提示：书架名不能超过 20 个字
选择正确标签	幕后黑手	可以接受	正常
不选择标签		系统提示选择标签	提示：请选择书架标签
在选择标签下划栏目选择新建标签输入新标签名	灵气复苏	可以接受	正常
输入正确的书架介绍	输入书架介绍	可以接受	正常
输入的书架介绍为空		系统提示输入书架介绍	提示：书架介绍不能为空
输入书架介绍超过 200 个汉字	输入超过 200 汉字的书架介绍	系统提示书架介绍太长	提示：书架介绍不能超过 200 个字

2、编辑书架功能测试结果如表6.3所示：



表 6.3 编辑书架功能测试用例

用例编号	UT-03		
功能描述	编辑书架信息		
用例目的	测试是否能够正确编辑书架信息		
前置条件	用户登录系统，进入修改书架界面		
输入动作	数据	期望的输出	实际输出
在下划列表选择正确的书架名	粮草集	可以接受	正常
不选择书架名		系统提示选择书架名	提示：请选择要修改的书架
输入新的正确的书架名	粮草集	可以接受	正常
输入新的书架名为空		系统提示输入书架名	提示：书架名不能为空
输入新的书架名超过 20 个汉字	输入超过 20 个汉字的书架名	系统提示书架名太长	提示：书架名不能超过 20 个字
选择正确标签	幕后黑手	可以接受	正常
不选择标签		系统提示选择标签	提示：请选择书架标签
在选择标签下划栏目选择新建标签输入新标签名	灵气复苏	可以接受	正常
输入正确的书架介绍	输入书架介绍	可以接受	正常
输入的书架介绍为空		系统提示输入书架介绍	提示：书架介绍不能为空
输入书架介绍超过 200 个汉字	输入超过 200 汉字的书架介绍	系统提示书架介绍太长	提示：书架介绍不能超过 200 个字

3、删除书架功能测试结果如表6.4所示:

表 6.4 删除书架功能测试用例

用例编号	UT-04		
功能描述	删除书架信息		
用例目的	测试是否能够正确删除书架信息		
前置条件	用户登录系统，进入删除书架界面		
输入动作	数据	期望的输出	实际输出
在下划列表选择正确的书架名	粮草集	删除成功	正常
不选择书架名		系统提示选择书架名	提示：请选择要删除的书架

4、书架书本管理测试结果如表6.5所示：

表 6.5 书架书本管理测试用例

用例编号	UT-05		
功能描述	书架内书本管理		
用例目的	测试是否能正确管理书架书本		
前置条件	用户登录系统，进入任一书架界面		
输入动作	数据	期望的输出	实际输出
在书本侧栏选择删除书本		删除成功	正常
在书本侧栏选择更换书架中选择新的书架	在下划栏中选择毒草书架	更换书架成功	正常
在书本信息界面选择加入书架	在下划栏中选择粮草集	加入书架成功	正常
用户没有书架，在书本信息界面选择加入书架	在下划栏中选择新建书架	转入新建书架界面	正常

### 6.3.3 书单管理功能测试

书单管理功能测试，主要测试是否可以按照用户要求进行加入书单，删除书本，新建书单，修改书单，删除书单等功能。以及对于一些异常事件的处理，比

如，新建书单未输入书单名，书单名输入字数超过限制等。

1、新建书单功能测试结果如表6.6所示：

表 6.6 新建书单功能测试用例

用例编号	UT-06		
功能描述	新建书单		
用例目的	测试是否能够正确新建书单		
前置条件	用户登录系统，进入新建书单界面		
输入动作	数据	期望的输出	实际输出
输入正确的书单名	粮草集	可以接受	正常
输入书单名为空		系统提示输入书单名	提示：书单名不能为空
输入书单名超过 20 个汉字	输入超过 20 个汉字的书单名	系统提示书单名太长	提示：书单名不能超过 20 个字
选择正确标签	幕后黑手	可以接受	正常
不选择标签		系统提示选择标签	提示：请选择书单标签
在选择标签下划栏目选择新建标签输入新标签名	灵气复苏	可以接受	正常
输入正确的书单介绍	输入书单介绍	可以接受	正常
输入的书单介绍为空		系统提示输入书单介绍	提示：书单介绍不能为空
输入书单介绍超过 200 个汉字	输入超过 200 汉字的书单介绍	系统提示书单介绍太长	提示：书单介绍不能超过 200 个字

2、编辑书单功能测试结果如表6.7所示：

表 6.7 编辑书单功能测试用例

用例编号	UT-07		
功能描述	编辑书单信息		
用例目的	测试是否能够正确编辑书单信息		
前置条件	用户登录系统，进入修改书单界面		
输入动作	数据	期望的输出	实际输出
在下划列表选择正确的书单名	粮草集	可以接受	正常
不选择书单名		系统提示选择书单名	提示：请选择要修改的书单
输入新的正确的书单名	粮草集	可以接受	正常
输入新的书单名为空		系统提示输入书单名	提示：书单名不能为空
输入新的书单名超过 20 个汉字	输入超过 20 个汉字的书单名	系统提示书单名太长	提示：书单名不能超过 20 个字
选择正确标签	幕后黑手	可以接受	正常
不选择标签		系统提示选择标签	提示：请选择书单标签
在选择标签下划栏目选择新建标签输入新标签名	灵气复苏	可以接受	正常
输入正确的书单介绍	输入书单介绍	可以接受	正常
输入的书单介绍为空		系统提示输入书单介绍	提示：书单介绍不能为空
输入书单介绍超过 200 个汉字	输入超过 200 汉字的书单介绍	系统提示书单介绍太长	提示：书单介绍不能超过 200 个字

3、删除书单功能测试结果如表6.8所示:

表 6.8 删除书单功能测试用例

用例编号	UT-08		
功能描述	删除书单信息		
用例目的	测试是否能够正确删除书单信息		
前置条件	用户登录系统，进入删除书单界面		
输入动作	数据	期望的输出	实际输出
在下划列表选择正确的书单名	粮草集	删除成功	正常
不选择书单名		系统提示选择书单名	提示：请选择要删除的书单

4、书单书本管理测试结果如表6.9所示：

表 6.9 书单书本管理测试用例

用例编号	UT-09		
功能描述	书单内书本管理		
用例目的	测试是否能正确管理书单书本		
前置条件	用户登录系统，进入任一书单界面		
输入动作	数据	期望的输出	实际输出
在书本侧栏选择删除书本		删除成功	正常
在书本侧栏选择更换书单中选择新的书单	在下划栏中选择毒草书单	更换书单成功	正常
在书本信息界面选择加入书单	在下划栏中选择粮草集	加入书单成功	正常
用户没有书单，在书本信息界面选择加入书单	在下划栏中选择新建书单	转入新建书单界面	正常

#### 6.3.4 检索管理功能测试

检索管理功能测试，主要测试是否可以按照用户要求执行小说检索，书单检索等功能。以及对于一些异常事件的处理，比如，检索小说未输入小说名，小说

名输入错误等。

1、检索管理测试用例, 如表6.10所示:

表 6.10 检索管理测试用例

用例编号	UT-10		
功能描述	检索管理		
用例目的	测试是否能正确检索书本书单		
前置条件	用户登录系统, 进入检索界面		
输入动作	数据	期望的输出	实际输出
检索小说输入正确的小说名	明尊	检索成功	正常
检索小说输入错误的小说名	明明明	提示没有找到该小说	提示: 没有检索到该小说
检索小说不输入小说名		提示输入小说名	提示: 检索栏不能为空
检索书单输入正确的书单名	粮草集	检索成功	正常
检索书单输入错误的书单名	粮草集合合合	提示没有找到该书单	提示: 没有检索到该书单
检索书单不输入书单名		提示输入书单名	提示: 检索栏不能为空

### 6.3.5 个人信息管理功能测试

个人信息管理功能测试, 主要测试是否可以按照用户要求执行修改用户信息, 修改密码等功能。以及对于一些异常事件的处理, 比如, 修改用户信息时, 新的用户名为空, 修改密码时, 与旧密码相同等。

个人信息管理测试结果如表6.11所示:

表 6.11 检索管理测试用例

用例编号	UT-11		
功能描述	个人信息管理		
用例目的	测试是否能正确执行个人信息管理功能		
前置条件	用户登录系统，进入个人信息管理界面		
输入动作	数据	期望的输出	实际输出
修改用户名，输入新的用户名	liaoya	修改成功	正常
修改用户名，不输入用户名		提示用户名不能为空	提示：用户名不能为空
修改邮箱	test@foxmail.com	修改成功	正常
修改邮箱，不输入邮箱		非必填项目，允许为空	正常
修改密码，输入新的密码	12345678	修改成功	正常
修改密码，不输入密码		提示输入密码	提示：密码不能为空
修改密码，输入旧密码	123456	提示新密码不能和旧密码相同	提示：新密码不能和旧密码相同

### 6.3.6 个性化推荐功能测试

对于随机森林而言，样本集中的每一个样本都可以做为随机森林模型中大约 1/3 子树的袋外数据检测样本，检测该样本做为袋外数据时，该个性化推荐模型的预测结果与真实结果的差异，最后进行统计模型预测正确数目与样本数目的比值做为袋外误差估计，由于自取样的特性，这种误差估计是无偏差估计。个性化推荐模型最终选择的特征子集如表6.12所示。

表 6.12 特征 Gini Importance

特征名	中文说明	编码
BookWordCount	小说字数 (万)	101
BookScore	评分	102
BookScoreCount	评分人数	103
AddListCount	加入书单人数	104
FiveStarRate	五星评分占比	105
Mode	众数	106
Median	中位数	107
DispersionCoefficient	星级评分离散系数	108
Kurtosis	星级评分峰度	109
NumOOB	该样本 OOB 子树数目	110
Numgood	预测为 good 子树数目	111
Numbad	预测为 bad 子树数目	112
ResultP	随机森林预测类型	113
ResultT	样本真实类型	114

对于该个性化推荐模型的 OOB 误差估计，部分检测结果如表6.13所示：



表 6.13 OOB 样本部分测试结果

101	102	103	104	105	106	107	108	109	110	111	112	113	114
289.90	7.2	967	492	0.25	8.00	8.00	0.32	406.22	291	200	91	good	good
48.30	6.4	47	12	0.15	6.00	6.00	0.36	267.71	287	33	254	bad	bad
52.30	6.4	129	38	0.18	8.00	6.00	0.41	155.27	316	97	219	bad	bad
139.00	5.8	78	22	0.10	6.00	6.00	0.44	113.07	271	68	203	bad	bad
158.50	5.6	439	122	0.12	6.00	6.00	0.46	93.52	276	166	110	good	good
59.60	6.0	187	42	0.23	2.00	6.00	0.49	75.15	291	118	173	bad	good
38.90	5.8	27	8	0.11	6.00	6.00	0.39	181.37	290	42	248	bad	bad
143.50	6.5	177	53	0.21	6.00	6.00	0.39	196.73	322	99	223	bad	bad
21.30	5.4	51	14	0.02	8.00	6.00	0.44	110.48	291	66	225	bad	bad
82.50	5.8	78	25	0.09	8.00	6.00	0.42	143.84	296	37	259	bad	bad
13.40	6.9	51	15	0.16	6.00	6.00	0.31	491.68	306	111	195	bad	bad
21.80	6.0	25	12	0.20	6.00	6.00	0.44	114.51	310	111	199	bad	bad
27.50	5.5	48	9	0.09	6.00	6.00	0.41	156.17	281	36	245	bad	bad
123.80	5.9	454	136	0.20	2.00	6.00	0.50	68.44	299	154	145	good	good
257.20	6.6	658	204	0.23	6.00	6.00	0.39	188.48	289	190	99	good	bad
135.80	5.7	119	25	0.14	6.00	6.00	0.47	87.54	279	98	181	bad	good
33.60	5.6	36	10	0.12	8.00	6.00	0.48	80.31	290	106	184	bad	bad
24.10	6.5	23	7	0.28	10.00	6.00	0.42	137.29	291	63	228	bad	bad
122.40	6.7	250	70	0.28	10.00	6.00	0.40	181.41	301	187	114	good	bad
82.30	5.3	112	17	0.13	2.00	4.00	0.55	48.29	278	214	64	good	good
74.10	4.6	44	12	0.04	4.00	4.00	0.50	66.70	297	120	177	bad	bad
77.70	6.5	159	61	0.11	6.00	6.00	0.31	468.11	286	59	227	bad	bad
122.20	4.7	116	32	0.06	4.00	4.00	0.48	78.49	294	34	260	bad	good
74.80	5.5	70	23	0.11	6.00	6.00	0.44	114.10	278	33	245	bad	bad
35.90	4.8	11	2	0.20	2.00	4.00	0.62	28.01	276	74	202	bad	bad
62.70	6.5	100	32	0.19	6.00	6.00	0.37	249.04	313	40	273	bad	bad
147.20	4.5	150	31	0.05	2.00	4.00	0.57	40.03	278	189	89	good	good
53.00	7.2	263	73	0.27	8.00	8.00	0.32	418.19	302	134	168	bad	bad
233.90	7.1	290	68	0.36	10.00	8.00	0.39	194.89	270	161	109	good	good
150.70	5.4	29	9	0.11	6.00	6.00	0.48	80.73	293	102	191	bad	bad
78.00	5.8	44	6	0.09	6.00	6.00	0.38	211.54	308	22	286	bad	good
51.10	6.8	292	111	0.13	8.00	8.00	0.32	404.81	294	124	170	bad	bad

经过检测, 总计 154 个样本, 预测结果与真实结果不一致的有 31 个, 袋外误差估计为 20.12%, 预测为 good 的数目有 59 个, 其中真实结果为 good 的有 44, 查准率为 74.57%, 综上所述, 该个性化推荐模型基本符合个性化网络小说推荐的需要。

### 6.3.7 综合模块功能测试

综合模块集成测试, 主要功能点以及功能描述, 以及测试结果, 测试表如表 6.14 所示。

表 6.14 综合模块集成功能测试用例

用例编号	UT-12	
功能描述	所有功能集成测试	
用例目的	测试系统集成后的功能是否实现	
前置条件	用户登录系统	
模块名称	测试内容	测试结果
登录模块	用户名、密码正确和错误输入	正常
书架管理模块	书架的新建、修改、删除和书架小说的管理	正常
书单管理模块	书单的新建、修改、删除和书单小说的管理	正常
检索管理模块	检索小说和书单时的正确输入和错误输入	正常
个性化推荐模块	个性化小说推荐的袋外错误率是否符合预期	正常
个人信息管理模块	个人信息的编辑和密码的修改	正常

## 6.4 性能测试

性能测试也是系统测试中必不可少的一个重要环节, 它直接决定了系统的可用性。影响用户体验的最关键因素是系统的响应速度, 这也是本性能测试的着重点, 表 6.15 所示为系统性能测试用例的平均响应时间, 涵盖了该系统中的各项功能操作。

表 6.15 个性化推荐系统性能测试

用例编号	UT-13	
测试目的	测试系统各功能响应时间	
前置条件	用户具有该系统登录权限	
测试项目	期望性能 (平均值)	实际性能 (平均值)
登录成功进入主页面	响应时间 < 2s	响应时间 < 1s
打开各功能显示页面	响应时间 < 2s	响应时间 < 1s
刷新显示页面	响应时间 < 2s	响应时间 < 1s
打开添加书架页面	响应时间 < 2s	响应时间 < 1s
确定新建书架刷新书架主界面	响应时间 < 3s	响应时间 < 2s
打开任一书架主界面	响应时间 < 2s	响应时间 < 1s
修改书架信息然后刷新书架主界面	响应时间 < 3s	响应时间 < 2s
打开小说检索, 检索小说	响应时间 < 3s	响应时间 < 2s
删除书架然后刷新书架主界面	响应时间 < 3s	响应时间 < 2s
删除书单然后刷新书单主界面	响应时间 < 3s	响应时间 < 2s
执行每日推荐刷新每日推荐界面	响应时间 < 3s	响应时间 < 2s

## 6.5 本章小结

本次系统测试列举的用例基本覆盖了各个功能模块, 从测试结果来看, 该系统各大功能总体上达到了预期的设想, 在运行的过程中并没有出现明显的漏洞, 基本能够满足用户的使用需求。但是, 该系统的功能相对单一, 有待后续添增功能。

## 第七章 总结与展望

### 7.1 总结

本文通过对国内外个性化推荐以及网络小说个性化推荐的研究情况进行分析,论证了本系统开发的重要性,然后对当今主流技术进行了简要的阐述,如 UML 建模语言、B/S 架构、随机森林算法、MySQL 数据库等。之后正式进入软件工程的开发基本规范,从分析到设计、再到实现、最后到测试的整套流程进行开发。该系统的开发主要通过登录注销、书架管理、书单管理、检索管理、个性化推荐这 5 个管理模块展开,在系统开发过程中使用了 UML 建模技术。

在需求分析阶段,通过建立系统的用例模型来帮助软件开发人员更好的理解了系统的功能需求。在系统设计阶段,使用模块化设计的思想,将系统分成几个模块依次对每个模块建立模型,用类图描述了模块的具体软件结构。在对数据库的设计中,充分考虑了网络小说个性化推荐系统对数据存储的要求,使用了基于关系模式的 MySQL 数据库存储网络小说个性化系统中小说信息,书架信息等系统信息。在系统实现阶段,使用 ASP.NET 动态网页设计技术设计 Web 页面,使用 C++ 编程语言进行服务器端的操作。充分利用了 VS2012 开发工具的优点,灵活使用各种控件用于网页的布局、数据的输入输出以及实现数据绑定功能。经测试,系统在总体上实现了系统目标。

系统特色有以下几点:

- (1) 本系统选择 UML 做为建模工具,增强了系统开发逻辑,同时提高而系统开发达到效率和准确性;
- (2) 软件架构采用基于 B/S 模型的三层架构,每个层功能清晰,只能单向调动,保证了软件架构不被打乱;
- (3) 在内容创新上,本文设计使用随机森林算法做为网络小说个性化推荐的核心算法,充分利用了随机森林算法的高准确率和高泛化性能。

## 7.2 展望

由于笔者能力和水平有限,对软件工程及计算机应用领域中的相关技术、方法以及算法的运用还不够成熟。在本文中体现在:

- (1) 可以进一步优化界面效果。界面效果的好坏会直接影响到用户使用系统的第一感觉,后续的工作是多学点美工知识,进一步美化和完善目前的界面设计效果;
- (2) 系统的分类模型质量有待进一步提高。可以进一步增加随机森林中的特征集,从更多维度来分类,更准确的对读者进行推荐。比如说统计最新一章 24h 内的本章说数目和点赞数做为反映该小说读者活跃度的一个维度加入特征集。
- (3) 系统缺乏进行自主学习功能。本系统只考虑到固有书架的模型建立,如何根据读者之后的阅读取向,动态的对个性化推荐模型进行优化是一个很好的研究方向。

## 参考文献

- [1] 中国互联网络信息中心.《中国互联网络发展状况统计报告》.中国网信网,2020-04-28,[http://www.cac.gov.cn/2020-04/27/c\\_1589535470378587.htm](http://www.cac.gov.cn/2020-04/27/c_1589535470378587.htm).
- [2] 艾瑞咨询.2019 年中国网络文学“出海”研究报告.艾瑞网,2019-06-21,<http://report.iresearch.cn/report/201906/3389.shtml>.
- [3] 冷亚军, 陆青, 梁昌勇. 协同过滤推荐技术综述 [J]. 模式识别与人工智能,2014,27(08):720-734.
- [4] R Armstrong , D Freitag , T Joachims , et. al.WebWatcher: A Learning Apprentice for the World Wide Web[J].School of Computer Scienc,1995.
- [5] H Lieberman.Letizia: An Agent That Assists Web Browsing[C]. International Joint Conference on Artificial Intelligence -IJCAI, 1995,924-929.
- [6] G. Linden, B. Smith, J. York. Amazon. com recommendations: item-to-item collaborative filtering[J].IEEE Internet Computing, 2003, 7(1):76-80.
- [7] 项亮. 推荐系统实战 [M]. 北京: 人民邮电出版社,2012.
- [8] 吴颜, 沈洁, 顾天竺, 等. 协同过滤推荐系统中数据稀疏问题的解决 [J]. 计算机应用研究,2007.
- [9] Han J W, Micheline K, Pei J. 数据挖掘: 概念与技术 [M]. 北京: 机械工业出版社, 2015.
- [10] T G Dietterich.Machine Learning Research:Four Current Directions[J].AI Magazine,1997, 18(4):97-136.
- [11] M Sewell.Ensemble Learning[M].UCL Department of ComputerScience,2011,1-10.
- [12] 方育柯. 集成学习理论研究及其在个性化推荐中的应用 [D]. 成都: 电子科技大学,2011.
- [13] R Polikar, A Topalis, D Parikh, et al.An ensemble based data fusion approach for early diagnosis of Alzheimer’ s disease[J].Information Fusion,2008,9(1):83-95.
- [14] 陈高洁. 基于集成学习的个性化推荐算法研究 [D]. 成都: 电子科技大学,2013.

- [15] Breiman L. Random Forest [J]. Machine Learning, 2001, 45: 5-32.
- [16] Dietterich T G. Ensemble Methods in Machine Learning [J]. Proc International Workshgp on Multiple Classifier Systems, 2000, 1857(1): 1-15.
- [17] Dietterich T G. An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization [J]. Machine Learning, 2000,40(2): 139-57.
- [18] 周志华. 机器学习 [M]. 北京: 清华大学出版社, 2016: 73-92
- [19] Breiman,L.(1996a).Bagging Predictors. Machine Learning, 24(2):123-140.
- [20] Breiman L I, Friedman J H, Olshen R A, Stone C J. Classification and Regression Trees(CART) [J]. Encyclopedia of Ecology, 1998, 40(3): 582-8.
- [21] Verikas A, Gelzinis A, Bacauskiene M. Mining data with random forest: A survey and results of new tests[J]. Pattern Recognition, 2011, 44: 330-349.
- [22] Carolin S, Anne L, Thomas K, et al. Conditional variable importance for random forests[J]. BMC Bioinformatics, 2008, 9: 307.
- [23] Kadam S T, Bajpai S. Development of Web Annotation Technique for Search Result Records Using Web Database[C]// International Conference on Computing Communication Control and Automation. IEEE, 2015.
- [24] 揣锦华, 袁琪. 面向对象程序设计与 VC++ 实践 [M]. 北京: 清华大学出版社, 2016.
- [25] Dennis A, Wixom B H, Tegarden D. Systems Analysis and Design: An Object-Oriented Approach with UML[M]. Wiley Publishing, 2015.
- [26] 罗昊. 基于 UML 建模的驾校学员培训管理信息系统的概要设计 [J]. 中国管理信息化, 2016, 19(6):77-77.
- [27] 方明杨. UML 在证券投资管理系统建模中的应用 [D]. 厦门: 厦门大学,2007.
- [28] 阎琦, 韩建群, 夏颖. 基于 UML 的高校教学日常管理系统的建模与实现 [J]. 计算机技术与发展,2014,24(04):196-199.
- [29] 王菁.UML 建模、设计与分析标准教程 (2013-2015 版)[M]. 北京: 清华大学

- 出版社,2015.
- [30] 王萍, 和海莲. 基于 RUP 的 UML 建模技术的研究及应用 [J]. 信息与电脑: 理论版, 2016(4).
- [31] Cunha A, Garis A, Riesco D. Translating between Alloy specifications and UML class diagrams annotated with OCL[M]. Springer-Verlag New York, Inc. 2015.
- [32] 数据库基本原理及应用开发教程 [M]. 南京: 南京大学出版社, 俞海, 2017
- [33] 李华, 张井玲, 刘婷婷. 大数据时代下数据挖掘技术的应用研究 [J]. 现代信息科技, 2019(13):132-133.
- [34] 吕建驰. 机器学习算法在数据挖掘中的应用 [J]. 电子世界, 2019, (13):62-63.
- [35] 刘志鹏, 卫晨. Spring Data 在 JavaEE 系统中的应用与研究 [J]. 电子世界, 2014, (16):278.
- [36] 黄文博, 燕杨. C/S 结构与 B/S 结构的分析与比较 [J]. 长春师范学院学报, 2006(08):56-58.
- [37] 刘肖楠. 网络环境下数据库的应用分析 [J]. 科技传播, 2015, 7(03):123-124.
- [38] 王全才. 随机森林特征选择 [D]. 大连: 大连理工大学, 2011.



## 致谢

“日月忽其不淹兮，春与秋其代序”，丽娃河畔，共青路边，梧桐落木，一如既往。三年前，我乘着开往这座城市的列车，来到华师大，努力想要看清这周围的每一个角落。千来个日夜，莘莘研学之路，终以这论文的结束而告一段落。

如今，这两年多的时光还可算明了。人生路漫漫，但难免有所遗憾。正如这篇毕业论文，撰写接近尾声之际，仍然还有许多尚待修补的地方，限于当下能力有限，只好尽力而为。也许期年之后，回顾此文，自惭之至，但也可看清来时之路。

于笔记本电脑前

2020年9月30日