北京邮电大学高等教育自学考试

毕 业 设 计（论 文）

论文题目 学校图书管理信息系统的设计与实现

专　　业 计算机信息管理

指导教师

姓　　名 孟孝龙

准考证号 050118205882

自考前学历 专科

2023 年 9 月

北京邮电大学高等教育自学考试办公室制

毕业设计（论文）诚信承诺书

我承诺所呈交的毕业设计（论文）是按照北邮自考办的有关规定，独立研究完成的。本人在毕业设计（论文）写作过程中恪守学术道德和学术规范，设计（论文）中凡引用他人已经发表或未发表的成果、数据、观点等，均已注明并列出了有关文献的名称、作者、年份、刊物名称和出版文献的出版机构、出版地和版次等内容，除此之外均为本人的观点和研究成果。本人在毕业设计（论文）查重检测中所提交的查重报告内容真实。如存在毕业设计（论文）学术不端行为，本人愿接受处罚并承担责任。

本人手写签名：孟孝龙

___2023____年____9___月___19____日

查重报告

维普论文检测系统
VIP PAPER CHECK SYSTEM

简洁报告-大学生版

基础信息

论文标题：050118205882-孟孝龙-学校图书管理信息系统的设计与实现

作者姓名：孟孝龙　　所属单位：北京邮电大学自学考试

报告编号：165fd7e90ags8g2a　　检测时间：2023-09-15 11:47:16

检测字符数：17268

检测结果

全文总相似比：**18.21%**（全文总相似比= 复写率 + 他引率 + 自引率 + 专业术语）

复写率：14.21%　　他引率：4.0%　　自引率：0.0%

专业术语：0.0%　　自写率：81.79%

相似片段

总相似片段数：86

期刊：9　　博硕：58　　综合：0

外文：0　　自建库：0　　互联网：19

检测范围

中文科技期刊论文全文数据库	中文主要报纸全文数据库	中国专利特色数据库
博士/硕士学位论文全文数据库	中国主要会议论文特色数据库	港澳台文献资源
外文特色文献数据全库	维普优先出版论文全文数据库	互联网数据资源/互联网文档资源
高校自建资源库	图书资源	古籍文献资源
个人自建资源库	年鉴资源	IPUB原创作品

时间范围：　1989-01-01至 2023-09-15

指标说明：

复写率：相似或疑似重复内容占全文的比重

他引率：引用他人的部分占全文的比重，请正确标注引用

自引率：引用自己已发表部分占全文的比重，请正确标注引用

专业术语：公式定理、法律条文、行业用语等占全文的比重

关注公众号
唯一指定官网
vpcs.cqvip.com

北京邮电大学自学考试

北京邮电大学高等教育自学考试
毕业设计（论文）任务书

<table>
<tr><td>姓名</td><td>孟孝龙</td><td>准考证号</td><td>050118205882</td><td>专业</td><td colspan="2">计算机信息管理（独立本科段）</td></tr>
<tr><td>工作单位</td><td colspan="3">北京网医联盟科技有限公司</td><td>职务岗位</td><td colspan="2">Java 开发工程师</td></tr>
<tr><td>邮箱</td><td colspan="3">2933902772@qq.com</td><td>联系电话</td><td colspan="2">13552434594</td></tr>
<tr><td>自考本科前学历</td><td>专科</td><td>毕业院校</td><td colspan="2">开封文化艺术职业学院</td><td>所学专业</td><td>软件技术</td></tr>
<tr><td colspan="2">设计（或论文）题目</td><td colspan="5">学校图书管理信息系统的设计与实现</td></tr>
<tr><td colspan="7">本设计选题根据（欲解决的问题及其学术或实用价值）：
随着社会技术的不断进步，传统的图书管理系统暴露出了各种问题，其中包括管理效率低下、信息不及时更新、数据管理不易、记录存储不方便、容易出现数据丢失等。为了解决这些问题，开发一个高效、智能的学校图书管理信息系统变得非常必要。该系统适用于大多数学校，并且将能够提高图书管理效率，实现图书信息的全面管理、记录和存储，避免数据丢失和信息不准确的问题。同时，该系统还能够优化借阅流程，提升学生的学习体验和阅读兴趣。因此，研究和开发这样一个系统具有重要的现实意义和深远的发展前景。</td></tr>
<tr><td colspan="7">本设计的主要研究内容（设计要求和设计步骤要求详细到章节）：
一、绪论
1、图书管理系统的背景
2、图书管理系统的现状
3、图书管理信息系统的目的和意义
二、系统总体规划
1、需求的初步调查
2、新系统的目标
3、可行性分析
三、系统分析
1、详细调查
2、业务流程分析
3、数据流程分析
四、系统设计
1、系统模块设计
2、系统架构设计
3、数据库设计
4、系统代码设计
5、系统输入和输出设计
五、系统实现与测试
1、系统实现
2、系统测试
六、总结</td></tr>
</table>

七、附录
八、参考文献

本人在该设计中完成的具体工作：
本人独立完成该设计

主要参考文献、资料：

[1]王美杰.高校图书馆地方文献阅读推广创新路径研究——以怀化学院图书馆为例[J].怀化学院学报,2020,39(03):107-110.DOI:10.16074/j.cnki.cn43-1394/z.2020.03.023.
[2]何辉娟.基于大数据的智能图书管理系统的研究与设计[J].电子技术与软件工程,2023(06):222-225.
[3]徐晨,程红.政府会计制度改革对高校资产管理的影响研究[J].中国乡镇企业会计,2020(05):122-123.
[4]赵冰.计算机管理系统在校园图书馆中的应用[J].电子技术,2023,52(03):250-251.
[5]张俊明.计算机在图书馆资源管理中的应用[J].信息与电脑(理论版),2019(16):16-18.
[6]邓阳名,顾潇,梁爱媚.基于 Java 语言的数据库访问技术应用研究[J].中国新通信,2023,25(08):83-85.

审批意见	教师签字：

任务书下达时间（教师填写）　　年　　月　　日

注：本表格式可调整，填写内容不要超过 2 页，完成后 A4 纸双面打印，附在论文前。

摘要

本文利用信息技术实现了一套学校图书管理信息系统，以满足学校图书管理的需要。通过该系统，可以实现图书信息的共享和管理，提升读者的使用体验和满意度。

本文的整体组织结构包括绪论、系统总体规划、系统需求分析、系统架构设计、数据库设计、系统模块设计、系统实现、系统测试等方面。从而系统而全面的论述了学校图书管理信息系统的设计和实现。

在系统实现方面，系统被分为了前端和后端。前端主要使用了 HTML、CSS、JavaScript、Vue、Element 等前端技术。后端主要采用了 Java 语言、Maven 工具、MySQL 数据库，同时使用 Spring Boot 框架和 MyBatis 框架进行开发。系统在软件开发时，主要使用 IntelliJ IDEA 和 VSCode 作为系统软件开发工具。

在系统功能模块方面，主要包括系统管理、基础配置、图书管理、借阅管理这几个部分。

关键字：图书管理，信息共享，Java，Vue，MySQL

目录

摘要 I
目录 II
一 绪论 1
（一） 图书管理系统的背景 1
（二） 图书管理系统的现状 1
（三） 图书管理信息系统的目的和意义 2
二 系统总体规划 3
（一） 需求的初步调查 3
（二） 新系统的目标 3
1 数字化 3
2 网络化 3
3 系统化 3
4 智能化 4
（三） 可行性分析 4
1 经济可行性 4
2 技术可行性 4
三 系统分析 6
（一） 详细调查 6
1 业务流程的调查 6
2 物理结构的调查 6
3 相关问题的调查 7
（二） 业务流程分析 7
1 基础配置信息录入 7
2 图书信息管理 8
3 图书借阅管理 8
（三） 数据流程分析 9
（四） 功能需求分析 10
1 系统管理功能 11

2 基础配置功能 11
3 图书管理功能 12
4 借阅管理功能 12
（五） 非功能需求分析 13
1 兼容性 13
2 跨平台 13
3 可用性和健壮性 13
四 系统设计 14
（一） 系统模块设计 14
（二） 系统架构设计 14
（三） 数据库设计 16
1 数据库模型设计 16
2 数据库表结构设计 17
（1） 用户表 17
（2） 读者表 18
（3） 借阅表 19
（4） 图书表 19
（5） 层架表 20
（6） 书架表 20
（7） 图书室表 21
（8） 楼层表 21
（9） 图书馆表 22
（10） 校区表 22
（四） 系统编码设计 23
（五） 系统输入和输出设计 23
1 输出设计 23
2 输入设计 24
（六） 系统界面设计 24
五 系统实现与测试 25
（一） 系统实现 25

1 系统管理的实现 25
（1） 系统登录 25
（2） 读者管理 27
2 基础配置的实现 28
（1） 校区管理 28
（2） 图书馆管理 30
3 图书管理的实现 31
4 借阅管理的实现 33
（1） 借阅申请管理 33
（2） 借阅审核管理 33
（3） 失信人管理 34
（二） 系统测试 35
1 运行环境搭建 35
2 测试方法介绍 36
3 系统功能测试 37
（1） 系统管理 37
（2） 基础配置 37
（3） 图书管理 38
（4） 借阅管理 38
4 测试结果分析 39
六 总结与展望 40
（一） 工作总结 40
（二） 工作展望 40
七 参考文献 42
附录 43

学校图书管理信息系统的设计与实现

一 绪论

学校图书管理信息系统的功能是代替传统的人工方式来管理和检索图书，以提高管理效率、改善使用体验和加快检索速度。本章节将从图书管理系统的背景、图书管理系统的现状和图书管理信息系统的目的和意义，来阐述学校图书管理信息系统的重要性。

（一） 图书管理系统的背景

作为学校教学和科研的重要组成部分，学校图书馆是学校信息化和社会信息化的重要基地[1]，也是学术性机构之一。

图书管理系统不仅影响着我们的生活和学习，还影响着我们的思维方式，其水平更体现了学校总体办学水平。随着国内外信息技术的不断发展，各行各业都在向着数字化、自动化、网络化的方向转变。学校图书管理工作也逐渐向数字化、自动化、网络化方向转变[2]。在这一趋势下，图书管理信息系统作为一种新型的图书管理方式，已经成为学校图书管理的重要组成部分。

数字化的图书管理可以改变图书管理过程中的数据存储形式，减少数据存储空间，提高数据存储效率。自动化的图书管理系统方便读者借阅图书，提高读者的满意度。它能够实现图书的自动管理和定位。

网络化的图书管理系统可以提升读者的使用体验，通过网络连接，读者可以在任何时间、任何地点进行图书查询和借阅操作。这种系统可以为学校图书馆的信息化建设提供了强大的支持，使得图书馆能够更好地适应学校的发展和建设。

（二） 图书管理系统的现状

随着国内人民受教育的需求不断提高，学校生员不断增加，许多学校由分校组成，每个分校都有独立的图书馆，这也使得各地对图书管理的需求不断增加。然而，目前图书管理系统的发展水平不平衡，有些地区的图书管理系统网络化程度较高，可以在互联网上正常访问，但有些地区的图书管理系统只能在当前校区使用，各个校区的信

息无法共享， 形成了一个个封闭的信息资源“孤岛”[3]。

更有甚者，一些学校仍然采用传统的人工管理方式来管理图书，这种方式存在诸多问题，例如管理效率低下、信息不及时更新、数据管理不易、记录存储不方便、容易出现数据丢失等[4,5]。这些问题严重影响了图书管理的效率和准确性，也降低了读者的满意度。

因此，建立一套现代化的图书管理信息系统具有极其重要的必要性。

（三） 图书管理信息系统的目的和意义

本文旨在设计和实现一套高效、安全、可靠的学校图书管理信息系统，以满足学校图书管理的需求，进一步提高管理效率和准确性。

本文将采用分层架构设计，将系统分为客户端、服务器和数据库三层，实现系统的模块化和扩展性。在数据库设计方面，本文将采用关系型数据库，设计合理的数据表结构和索引，提高数据查询和管理的效率。在系统实现方面，本文将采用 Java 语言和 MySQL 数据库[6]，使用 Spring Boot 框架和 MyBatis 框架进行开发[7]，实现系统的功能和性能。在系统测试时，将进行系统功能测试，以确保系统的可用性和稳定性。

通过本文的研究，可以实现学校图书管理信息系统的设计和实现，提高图书管理的效率和准确性，方便读者借阅图书，提高读者的满意度，推动学校图书馆的发展和建设，为学校的教学和科研提供更加优质的服务和支持。

二 系统总体规划

总体规划是管理信息系统生命周期中的第一个阶段，也是系统实现过程中的第一步[11]，本章将从需求的初步调查、新系统的目标和可行性分析三个方面，对系统进行总体规划。

（一） 需求的初步调查

尽管学校图书管理系统已经存在很长时间，但不同学校的发展程度却不平衡。有些学校在不同校区之间的图书管理系统无法进行信息共享，形成了信息资源的“孤岛”。同时，一些学校仍在使用传统的手工方式管理图书，效率低下。因此，现在学校迫切需要一个高效的图书管理信息系统来解决信息资源孤岛问题，并代替传统的手工方式管理图书。

（二） 新系统的目标

1 数字化

学校图书管理信息系统中的信息资源，包括图书、人员、馆藏和书架等信息，将向数字化转移，用计算机技术进行存储，代替传统的手工记录和书面保存等相关形式，彻底解决资源易丢失和不易保存等问题。

2 网络化

学校图书管理信息系统中的信息资源将可以通过网络访问，实现信息的共享和交流，提高信息的传递和利用效率。

3 系统化

学校图书管理信息系统中的各个馆藏和校区的信息资源可以共享，形成一个完整的系统闭环，提高信息的整合和管理效率，为读者提供更加便捷的服务。

4　智能化

学校图书管理信息系统将代替传统的人工管理图书的方式，形成自动化、智能化的工作流程，并减少人工参与，提高管理效率和准确性，为图书馆的发展和建设提供更加有力的支持。同时，将注重优化用户体验，提高读者满意度。

（三） 可行性分析

通过需求的初步调查，我们已经知道开发学校图书管理信息系统是非常有必要的。下面将从经济可行性、技术可行性两个方面进行可行性分析。

1　经济可行性

随着计算机技术的不断进步，计算机硬件成本逐渐降低，同时计算机软件开发技术变得越来越成熟，软件开发费用不断下降，因此开发图书管理信息系统的总成本较低。本系统采用开源免费的框架，前期只需要一个开发人员三个月左右的时间进行开发实现（包括需求、设计、实现），后期运行成本只需要一个运维人员进行运行维护，软件和人员成本都非常低。硬件方面可以使用华为云和阿里云，每年大概需要两三千的成本。因此，学校图书管理信息系统的整体成本非常低。

另一方面，随着人口老龄化的加重[8]，新生儿的出生人口不断下降[9]，适龄劳动人口不断减少，人力成本不断增加，一个人力成本大概在四千到五千的成本。学校图书管理信息系统在运行以后可以减少许多人力成本，根据不同学校会有所不同，一般可以减少 10 人左右的工作量，这将带来新系统的经济效益。同时，新系统的运行还会带来许多隐性效益，如减少时间成本等。

综上所述，新系统所带来的效益远远高于成本，因此在经济上是可行的。

2　技术可行性

在技术选型方面，本系统着重考虑技术的成熟性和稳定性。系统采用 MVC（Model-View-Controller）设计模式，将系统分为前端和后端。前端主要使用 HTML、CSS、js，并使用成熟的 vue 框架、elment-UI，提高开发效率。开发环境采用 nodejs，项目包管理工具和项目构建工具使用 npm 和 webpack。后端采用 Java 作为主要开发语言，使用 springboot、MyBatis 和 Redis 等成熟的技术框架，项目的构建和管理使用 Maven。数

据库方面，系统主要使用 MySQL 作为数据存储。

总体而言，本系统所使用的技术具有很好的成熟性和稳定性，因此在技术方面是可行的。

三 系统分析

系统分析是系统开发过程中的重要内容，系统分析最主要内容是弄清楚用户的想法和需求，主要明确系统未来“干什么”，并确定系统的逻辑模型[11]。本章就从下面几个方面进行系统分析。

（一） 详细调查

1 业务流程的调查

在学校图书馆中，图书管理流程包括购入图书、采编录入、上架、下架、浏览借阅、续借和归还等环节。首先，图书管理人员将从外部购入的图书进行采编并录入学校的藏书系统。接着，图书管理人员将录入的图书根据中图法上架到图书馆的书架上。根据相关规定或要求，图书管理人员会对图书进行下架或重新上架。学校的图书借阅人员可以在上架的图书中进行浏览和借阅。当图书借阅人员借阅图书时，图书管理人员负责处理和记录相关事项，借阅人员随后即可拿走图书。若借阅人员符合相关规定要求进行续借，图书管理人员会进行相应处理并记录，即完成续借。当图书被图书借阅人员归还时，图书管理人员会处理图书并记录相关信息，归还即完成。对于超过借阅期限未归还图书的借阅人员，学校有关人员将对其进行处理，并禁止其以后借书。其业务流程图如下图 3-1 所示。

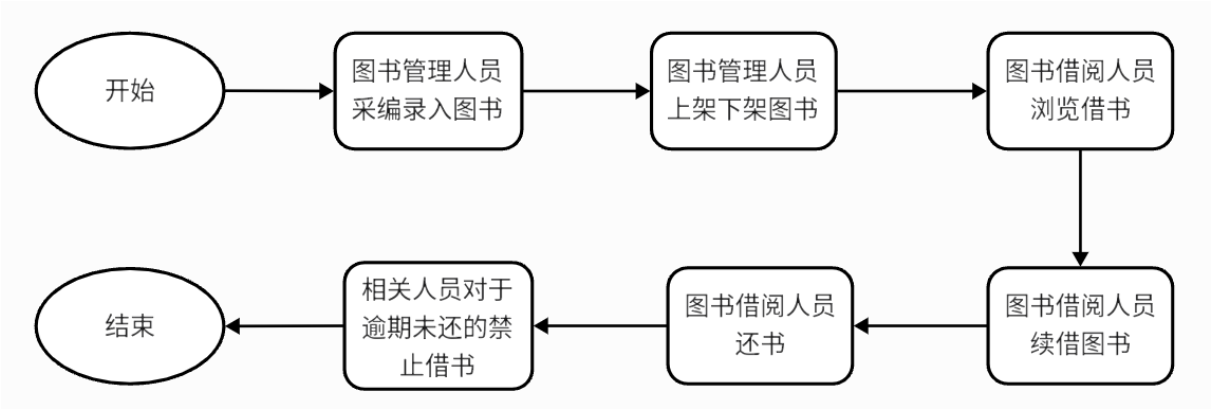

图 3-1 当前系统业务流程图

2 物理结构的调查

在物理结构方面现在的多数学校都拥有多个校区，每个校区都设有图书馆。图书

馆内部结构复杂，包括多个楼层、图书室、书架和层架，每个层架上存放着多本图书。

3　相关问题的调查

随着学校的发展和规模扩大，图书数量不断增加，学校也分成了多个校区。然而，这种发展也暴露出一些问题。例如，不同校区的图书数据无法共享，导致了数据查询、图书查找和借阅处理等方面的一系列问题。有些学校甚至仍采用传统的人工方式管理图书，这带来了更多的问题，如信息存储和检索效率等方面的挑战。

（二） 业务流程分析

经过对学校图书管理信息系统需求的详细调查，根据学校图书馆现存的问题对业务流程进行优化，将业务流程分为三个部分：基础配置信息录入、图书信息管理和图书借阅管理。

1　基础配置信息录入

基础配置信息录入流程包括以下步骤：图书管理人员登录系统，录入校区信息，然后录入图书馆信息，接着录入楼层信息，然后录入图书室信息，再录入书架信息，最后录入层架信息。基础配置信息录入的流程图如图 3-2 所示。

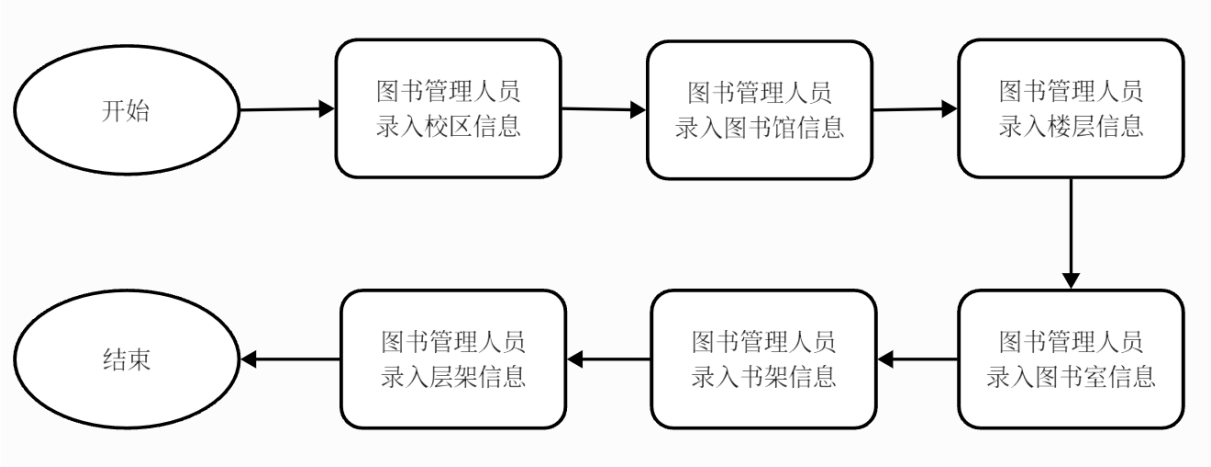

图 3-2 基础配置信息录入

2 图书信息管理

图书信息管理流程包括以下步骤：图书管理人员登录系统，先进行图书信息的录入（也称为新增），新录入的图书信息默认为上架状态，然后可以根据需要对图书进行下架操作，同时也可以对图书进行上架操作。图书信息管理的流程图如图 3-3 所示。

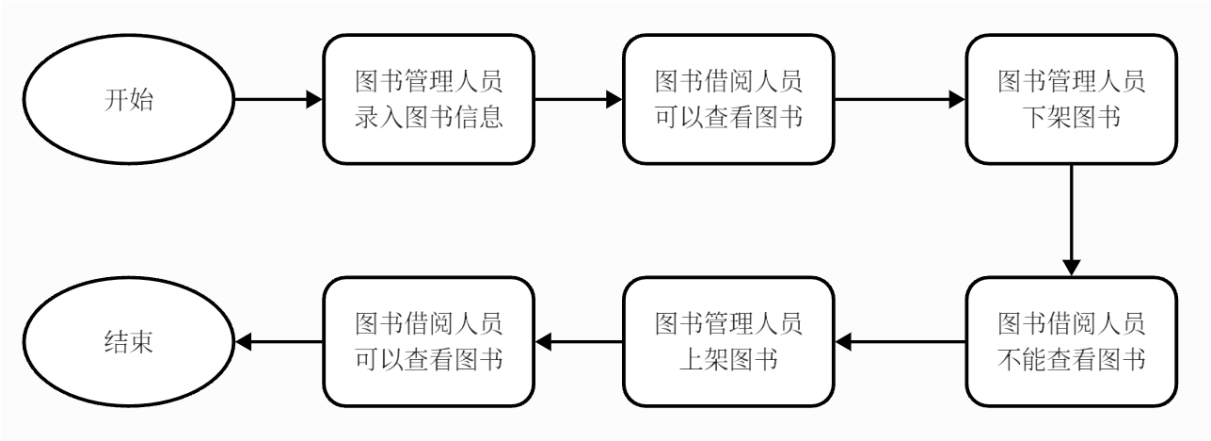

图 3-3 图书信息管理

3 图书借阅管理

图书借阅管理流程包括以下步骤：首先，图书管理人员和图书借阅人员都需要进行登录。图书借阅人员可以浏览图书并选择需要借阅的图书，然后提出借阅申请。图书管理人员会对申请进行审核，审核通过后借阅成功。在借阅期间，借阅人员如果需要续借图书，可以提出续借申请，经过图书管理人员的审核后续借成功。在归还图书时，借阅人员需要提出还书申请，图书管理人员审核通过后还书成功。对于逾期未归还图书的借阅人员，系统会将其列入失信人名单，失信人将无法借阅图书。图书管理人员可以根据具体情况将失信人移出失信人名单。图书借阅管理的流程图如图 3-4 所示。

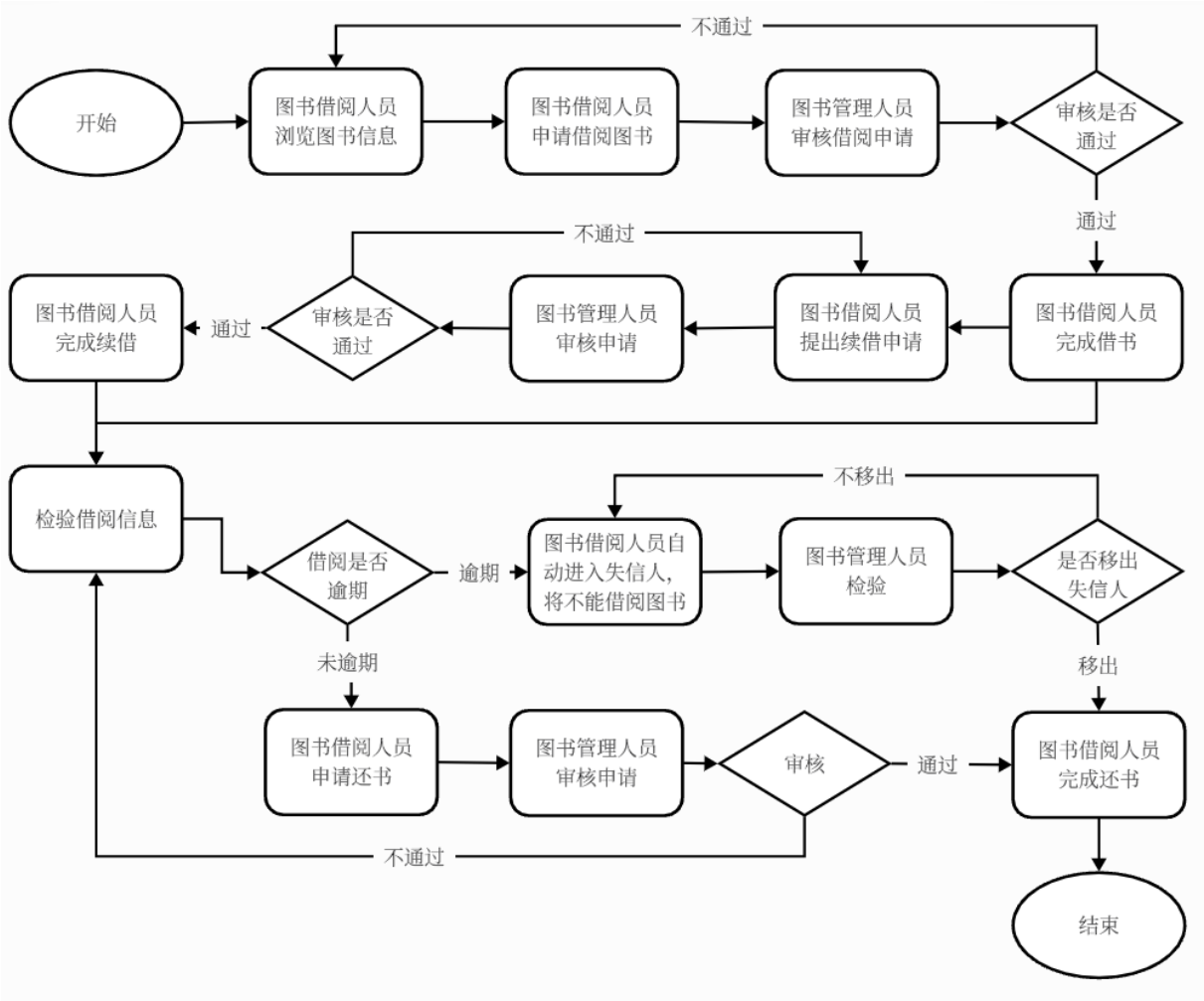

图 3-4 图书借阅管理

（三） 数据流程分析

数据流程分析是一种从数据的角度来理解系统的方法，其主要目的是形成新系统的逻辑模型。在数据流程分析中主要包括分层的数据流图、数据字典和补充说明等工具[11]。本文将通过分层的数据流图来详细说明学校图书管理信息系统的数据处理和流向。

学校图书管理信息系统的顶层数据流图主要涉及以下外部实体：系统管理人员、图书管理人员和图书借阅人员。系统管理人员主要负责提供管理信息（包括用户权限相关信息），同时从系统获取查询信息；图书管理人员主要负责输入基础信息（包括图书馆和图书的相关信息）和管理信息，同时从系统获取查询信息；图书借阅人员主要从系统获取查询信息，并提供申请信息。学校图书管理信息系统的顶层数据流图如图 3-5 所示。

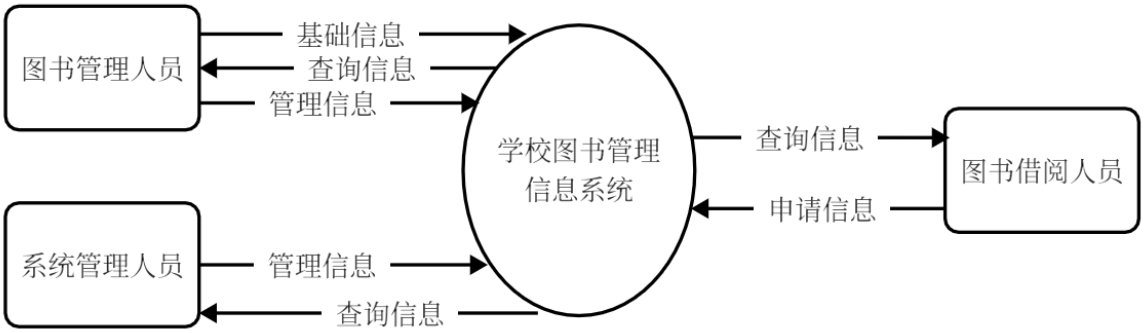

图 3-5 顶层数据流图

学校图书管理信息系统的零层数据流图涉及外部实体如图 3-6 所示。

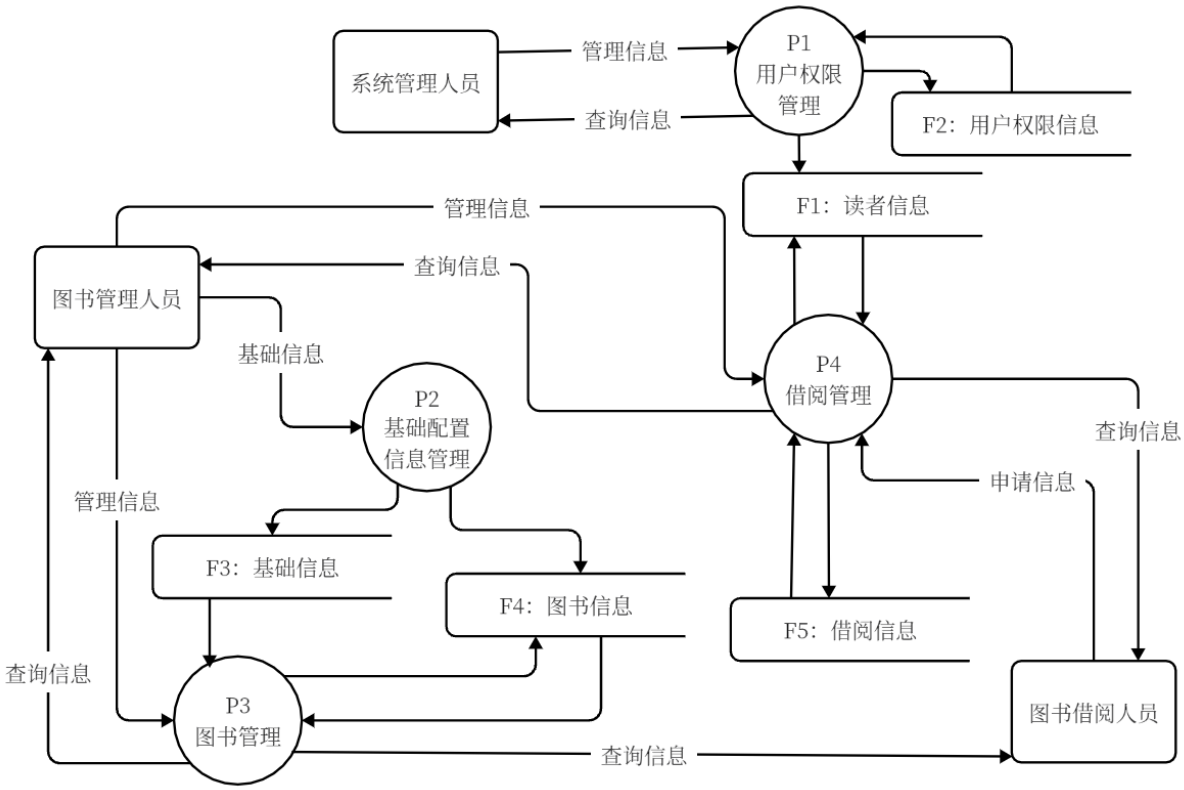

图 3-6 零层数据流图

通过数据流程分析，我们可以更好地理解学校图书管理信息系统的数据处理和流向，形成系统的逻辑模型。这将提高系统的管理效率，并为用户提供便捷的查询和申请服务。

（四） 功能需求分析

通过上面的分析，可以将学校图书管理信息系统大致分为四个功能模块，分别为：系统管理、基础配置、图书管理、借阅管理。

1 系统管理功能

系统管理主要是对系统整体进行管理，系统管理的功能需求主要有：首先，需要提供系统登录功能，以确保所有用户都能方便地登录系统。另外，系统还应提供用户角色权限管理功能，允许系统管理员对用户的角色和权限信息进行修改，以确保每个用户具备适当的权限。此外，还需要提供读者管理功能，使系统管理员能够分配图书借阅人员的读者信息，以便管理借阅记录和归还图书。最后，系统还应提供系统监控功能，使系统管理员能够实时监测系统的运行情况，包括系统的性能、错误日志和用户活动等信息。该功能用例图如图 3-7 所示。

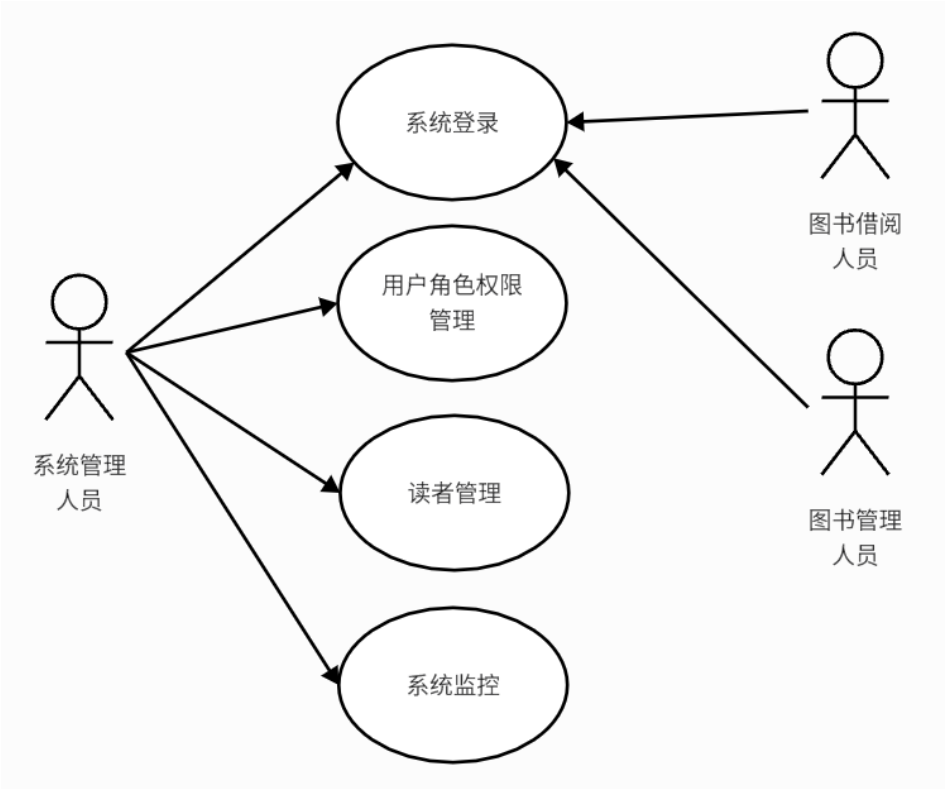

图 3-7 系统管理用例图

2 基础配置功能

基础配置的功能需求主要有：首先需要提供校区管理功能，确保图书管理人员可以录入校区信息。其次需要提供图书馆管理功能，确保图书管理人员可以录入图书馆信息。另外还需要提供图书室管理功能，确保图书管理人员可以录入图书室信息。另外还需要提供书架管理功能，确保图书管理人员可以录入书架信息。最后还需要提供层架管理功能，确保图书管理人员可以录入层架信息。该功能用例图如图 3-8 所示。

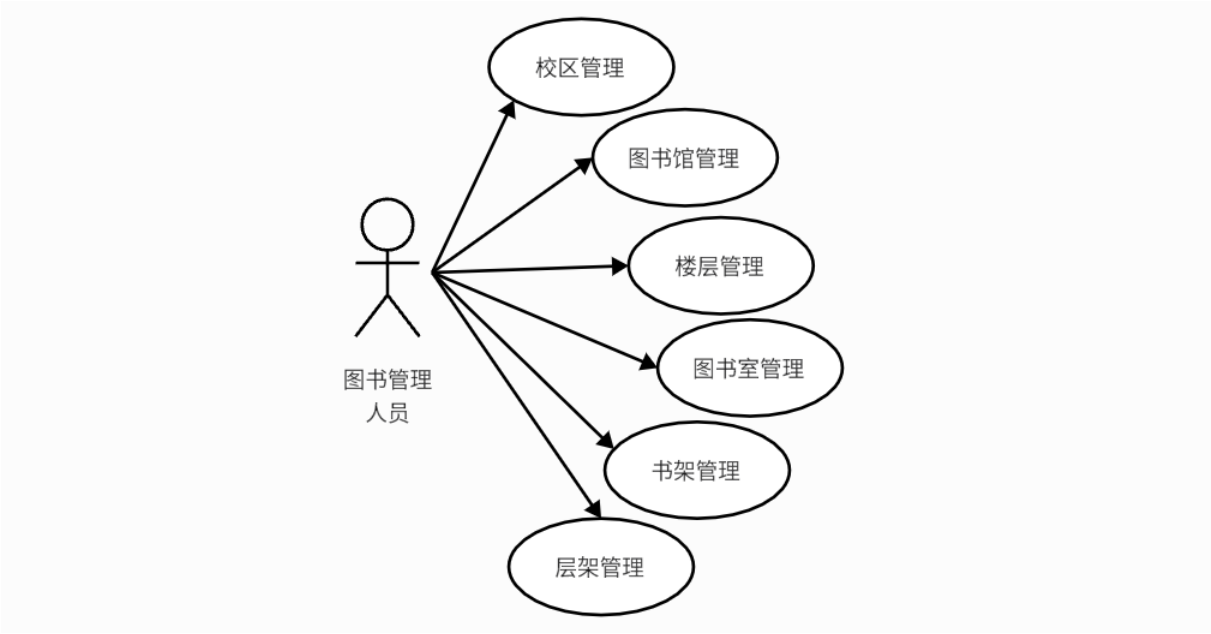

图 3-8 基础配置用例图

3 图书管理功能

图书管理功能主要由图书工作人员和图书借阅人员使用，其主要的功能需求有：图书管理人员需要拥有图书录入、图书下架、图书上架功能。图书借阅人员需要拥有图书查询功能和申请借阅功能。图书管理的用例图如图 3-9 所示。

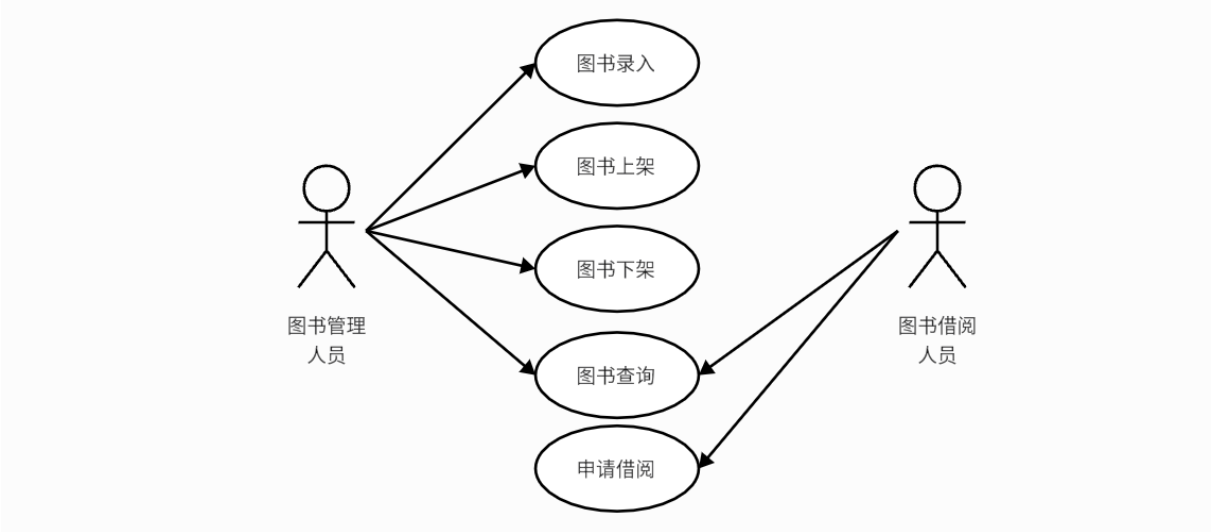

图 3-9 图书管理用例图

4 借阅管理功能

借阅管理功能主要由图书管理人员和图书借阅人员使用，其主要功能包括：借阅

申请功能、借阅审核功能、失信人管理功能。借阅管理的用例图如图 3-10 所示。

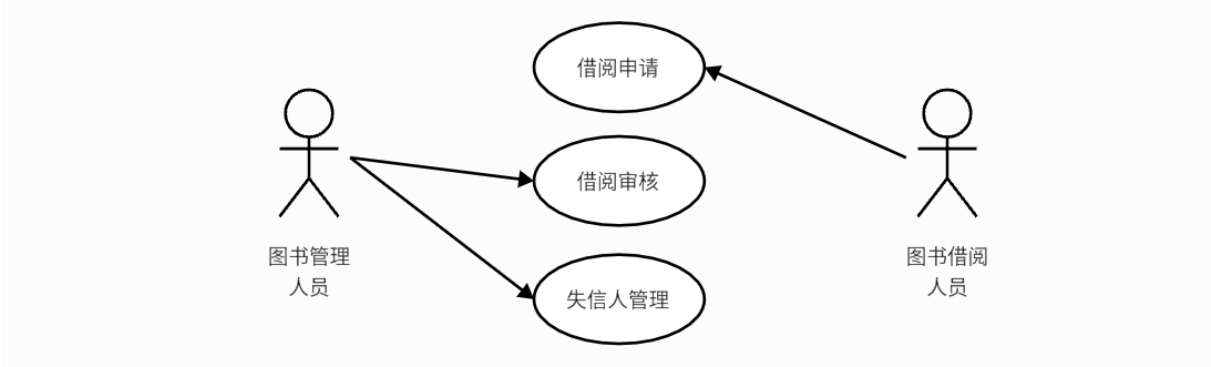

图 3-10 借阅管理用例图

（五） 非功能需求分析

1 兼容性

学校图书管理信息系统要具备良好的兼容性，能够在多种浏览器上运行，包括谷歌、IE10、Edge 等常见浏览器，为用户提供更加便捷的使用体验。

2 跨平台

学校图书管理信息系统需要具备跨平台的能力，为用户提供更加广泛的使用渠道。

3 可用性和健壮性

学校图书管理信息系统需要保持长期稳定运行，其可用率应在一年中保持在 95%以上，同时在用户输入错误时，系统应该给出相应的错误提示，提高系统的可用性和健壮性。系统还需要具备高度的安全性，确保用户信息和数据的安全性和保密性。

四 系统设计

系统设计是系统开发中的重要组成部分。前面我们在系统分析中明确了系统“干什么”的问题，这里我们通过系统设计明确系统“怎么干”的问题[11]。下面我们将从系统模块设计、系统架构设计、数据库设计、系统编码设计、系统输入和输出设计、系统界面设计这几个方面进行系统设计，在设计的同时参考简单性、完整性、适应性、可靠性、安全性和经济性等原则[11]。

（一） 系统模块设计

经过系统分析，为了满足高内聚低耦合的要求，学校图书管理信息系统被分为四个功能模块：系统管理模块、基础配置模块、图书管理模块和借阅管理模块。每个模块内部又包含了许多小模块，具体内容如模块结构图如图 4-1 所示。

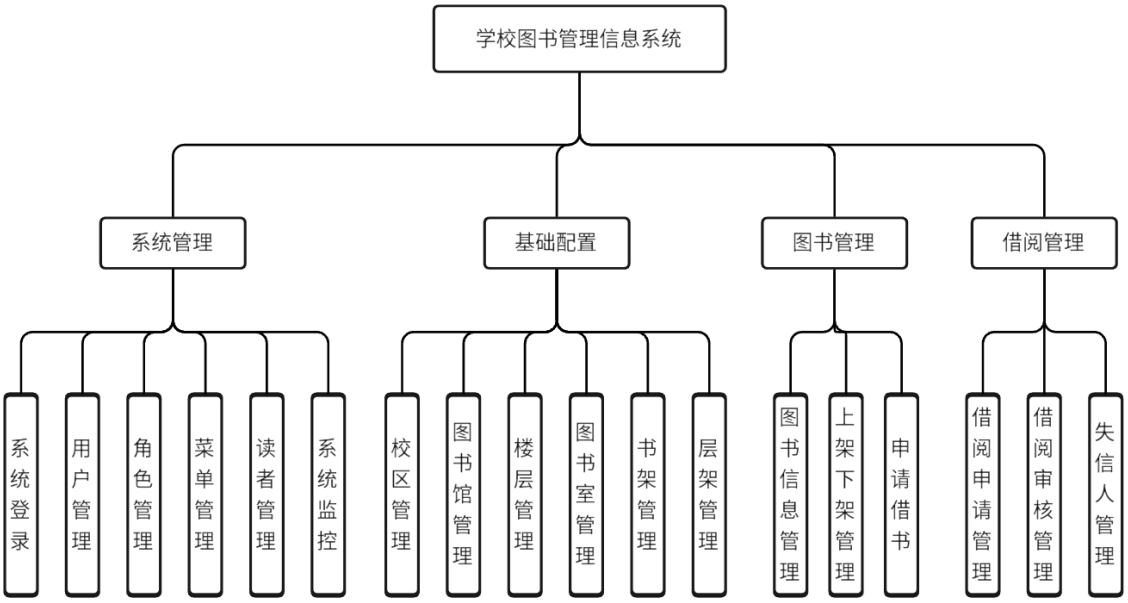

图 4-1 系统功能模块图

（二） 系统架构设计

系统使用 B/S 架构模式，同时后端使用单体架构模式，只需要浏览器就可以使用本系统的全部功能，正因为如此系统可以做到轻易的跨平台。系统架构图如图 4-2 所示。

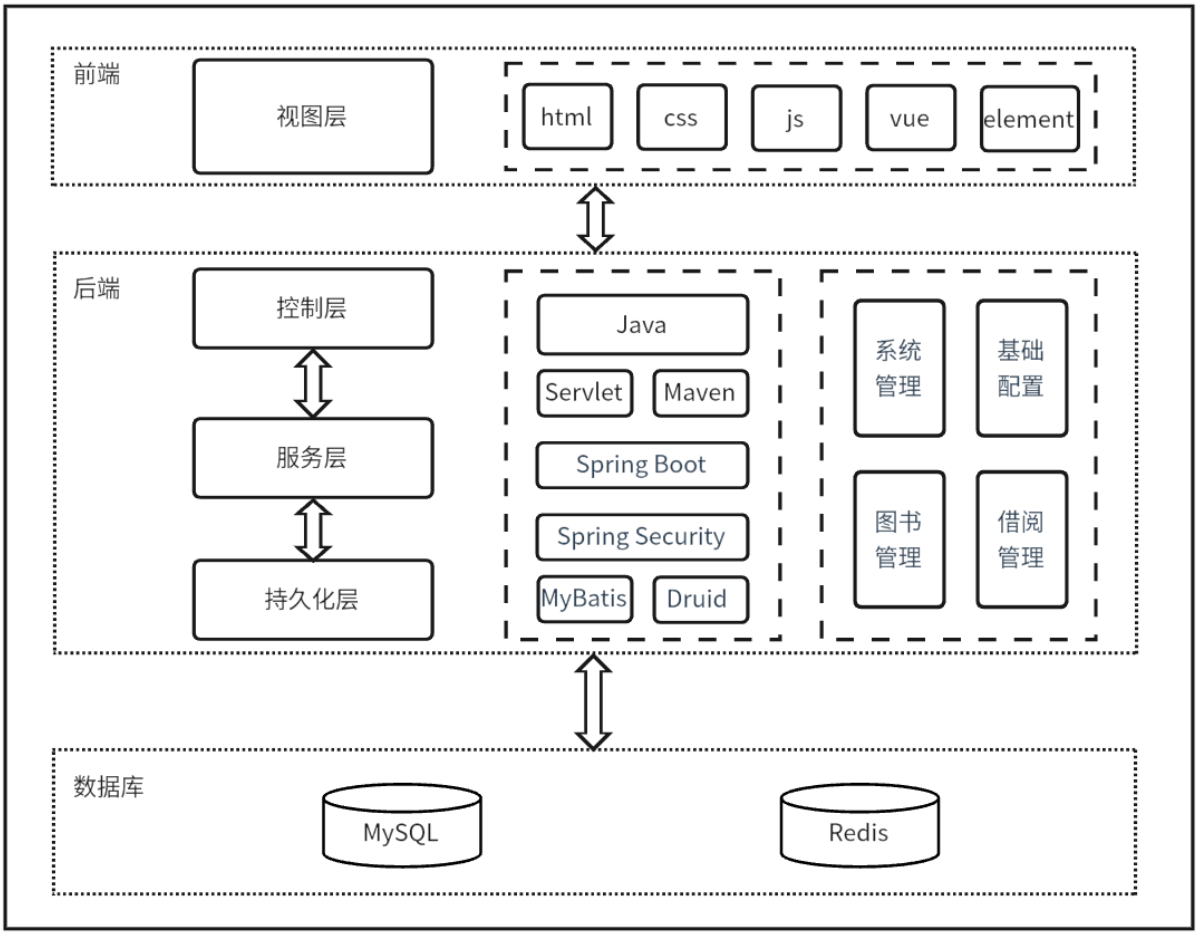

图 4-2 系统架构图

本系统从总的方面分为前端、后端、数据库三大模块，彼此之间进行数据双向通信。

前端负责展示界面的主要任务，使用了 html、css、js、vue、element 等前端技术。采用了 MVVM 设计模式，实现了数据的动态更新和界面的响应式变化。

后端主要包括控制层、服务层和持久化层。同时后端主要使用 Java 语言，使用 Maven 依赖第三方类库，Spring Boot 作为系统对象容器，Spring Security 确保系统安全，MyBatis 实现项目与数据库通信。系统功能包括系统管理、基础配置、图书管理和借阅管理。

数据库方面使用 MySQL 和 Redis，MySQL 用于主要数据存储，Redis 用于存储需要频繁查询且查询速度要求较高的数据。

系统序列图是一种能够清晰展示系统交互过程、时序关系和行为的模型。本系统序列图如图 4-3 所示。

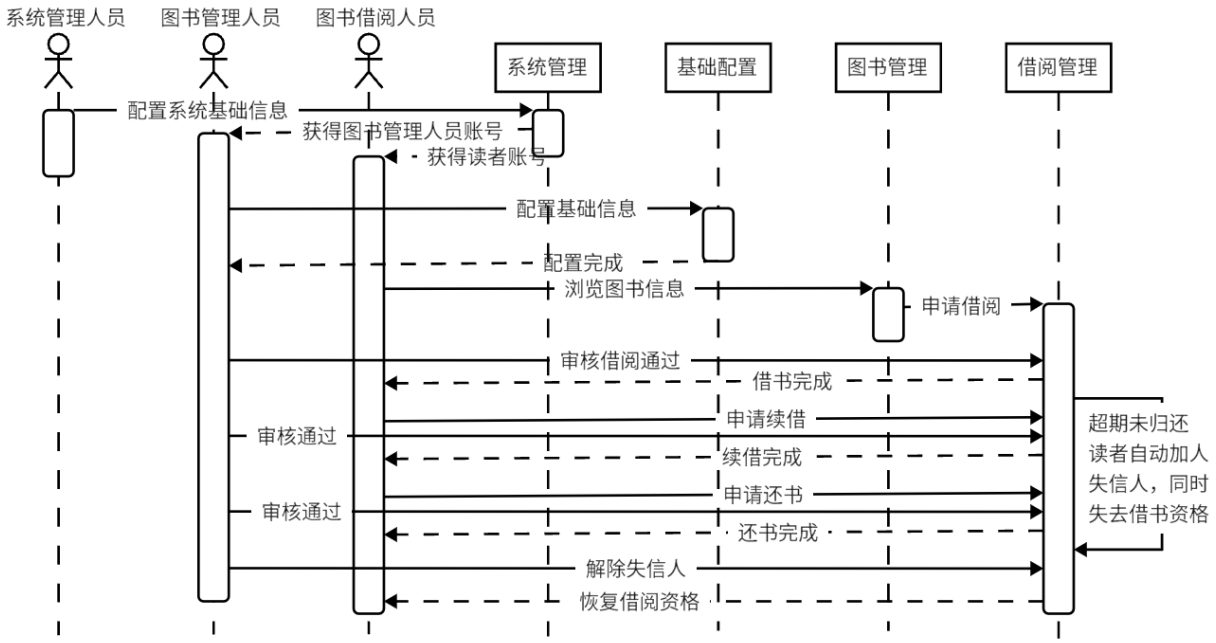

图 4-3 系统序列图

系统序列图主要包括了三个系统角色和四个系统业务模块，通过系统序列图，可以清晰地描述系统中各个参与者之间的交互流程，帮助读者更好地理解系统的工作原理和交互方式。

（三） 数据库设计

数据库设计在系统设计中居于核心地位，任何一个优秀的系统都离不开一个良好的数据库设计。一个良好的数据库设计对系统的稳定性、可用性、健壮性有着重要的作用。这里我们将严格按照数据库设计规范，建立符合数据库范式规范的数据库设计，并充分考虑系统在使用过程中的性能需求，在特殊情况下使用反范式的设计标准。

1 数据库模型设计

结合相关设计规范，对本系统中的实体和关系进行分析，得到系统的实体关系图如图 4-4 所示。由于系统监控涉及的内容比较多而且并非核心部分，所以这里不做设计和具体说明。

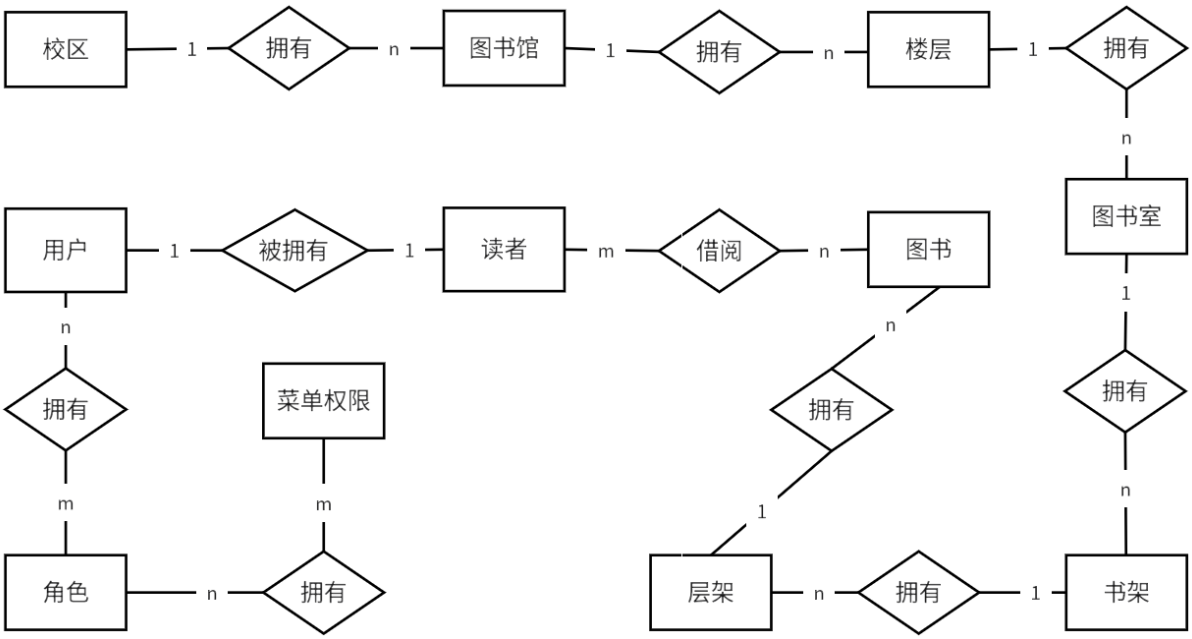

图 4-4 系统实体关系图

由图 4-4 可以看出，用户和角色是多对多的关系，角色和菜单权限是多对多的关系。读者和用户是 1 对 1 的关系。读者和图书是多对多的关系。

在图书馆管理方面，校区和图书馆是 1 对多的关系。图书馆和楼层是 1 对多的关系。楼层和图书室是 1 对多的关系。图书室和书架是 1 对多的关系。书架和层架是一对多的关系。层架和图书是 1 对多的关系。

通过以上关系的建立，系统能够实现用户和角色的灵活管理，读者和图书的借阅管理，以及图书馆和校区的组织管理。这样的关系模型能够提高系统的灵活性和可扩展性，为用户提供更好的图书馆管理体验。同时，这种关系模型也能够提供准确的数据关联，方便系统进行数据查询和统计分析，为图书馆管理者提供决策支持。

2 数据库表结构设计

下面将对数据库的表结构设计进行介绍。由于系统表结构设计较多，这里就省略了角色表和权限菜单表相关的表结构设计，主要对业务相关的表结构设计进行介绍。

（1） 用户表

用户表（sys_user），记录了用户的相关信息，如表 4-1 所示。

表 4-1 用户表

字段名称	字段类型	长度	是否主键	字段说明
user_id	bigint	20	是	用户 ID
dept_id	bigint	20	否	部门 ID
user_name	varchar	30	否	用户账号
nick_name	varchar	30	否	用户昵称
user_type	varchar	2	否	用户类型（00 系统用户）
email	varchar	50	否	用户邮箱
phonenumber	varchar	11	否	手机号码
sex	char	1	否	用户性别（0 男 1 女 2 未知）
avatar	varchar	100	否	头像地址
password	varchar	100	否	密码
status	char	1	否	帐号状态（0 正常 1 停用）
del_flag	char	1	否	删除标志（0 代表存在 2 代表删除）
login_ip	varchar	128	否	最后登录 IP
login_date	datetime		否	最后登录时间
create_by	varchar	64	否	创建者
create_time	datetime		否	创建时间
update_by	varchar	64	否	更新者
update_time	datetime		否	更新时间
remark	varchar	500	否	备注

（2） 读者表

读者表（t_readerinfo），记录了读者的相关信息，如表 4-2 所示。

表 4-2 读者表

字段名称	字段类型	长度	是否主键	字段说明
id	varchar	50	是	读者 id
userid	varchar	50	否	用户 id
name	varchar	50	否	读者姓名
idnum	varchar	50	否	身份证号
state	char	1	否	读者状态（0 正常，1 失信）
createtime	timestamp		否	创建时间
realtime	timestamp		否	更新时间

（3） 借阅表

借阅表（t_borrow），记录了读者借阅图书的相关信息，如表 4-3 所示

表 4-3 借阅表

字段名称	字段类型	长度	是否主键	字段说明
id	varchar	50	是	借阅 id
readerid	varchar	50	否	读者 id
bookid	varchar	50	否	图书 id
startdate	datetime		否	借阅开始日期
finishdate	datetime		否	借阅结束日期
state	char	1	否	借阅状态
createtime	timestamp		否	创建时间

（4） 图书表

图书表（t_bookinfo），记录了图书的相关信息，如表 4-4 所示。

表 4-4 图书表

字段名称	字段类型	长度	是否主键	字段说明
id	varchar	50	是	图书 id
layerid	varchar	50	否	层标 id
barcode	varchar	50	否	条码号
title	varchar	255	否	题名
author	varchar	255	否	作者
press	varchar	255	否	出版社
callnum	varchar	50	否	索书号
price	varchar	50	否	价格
borrowstate	char	1	否	借阅状态 0 在馆，1 在借
bookstatus	char	1	否	在架状态 0 上架，1 下架
operators	varchar	50	否	操作人员
createtime	timestamp		否	创建时间
realtime	timestamp		否	更新时间

（5） 层架表

层架表（t_layer），记录了书架中层架的相关信息，如表 4-5 所示。

表 4-5 层架表

字段名称	字段类型	长度	是否主键	字段说明
id	varchar	50	是	层架 id
shelfid	varchar	50	否	架标 uuid
code	varchar	50	否	层标代码
name	varchar	255	否	层标名称
orderid	int	11	否	排序
location	varchar	50	否	位置名称，格式 000.00.0
createtime	timestamp		否	创建时间
realtime	timestamp		否	更新时间

（6） 书架表

书架表（t_shelf），记录了书架的相关信息，如表 4-6 所示。

表 4-6 书架表

字段名称	字段类型	长度	是否主键	字段说明
id	varchar	50	是	id
roomid	varchar	50	否	图书室 id
code	varchar	50	否	架标代码
name	varchar	255	否	架标名称
orderid	int	11	否	排序
createtime	timestamp		否	创建时间
realtime	timestamp		否	更新时间

（7） 图书室表

图书室表（t_book_room），记录了图书室的相关信息，如表 4-7 所示。

表 4-7 图书室表

字段名称	字段类型	长度	是否主键	字段说明
id	varchar	50	是	id
floorid	varchar	50	否	楼层 id
code	varchar	50	否	代号
name	varchar	255	否	馆区名称
orderid	int	11	否	排序
createtime	timestamp		否	创建时间
realtime	timestamp		否	更新时间

（8） 楼层表

楼层表（t_floor），记录了楼层的相关信息，如表 4-8 所示。

表 4-8 楼层表

字段名称	字段类型	长度	是否主键	字段说明
id	varchar	50	是	id
libid	varchar	50	否	图书馆 id
code	varchar	255	否	代号
name	varchar	255	否	楼层名称
orderid	int	11	否	排序
createtime	timestamp		否	创建时间
realtime	timestamp		否	更新时间戳

（9） 图书馆表

图书馆表（t_library），记录了图书馆的相关信息，如表 4-9 所示。

表 4-9 图书馆表

字段名称	字段类型	长度	是否主键	字段说明
id	varchar	50	是	id
campusid	varchar	50	否	校区 id
code	varchar	255	否	代号
name	varchar	255	否	图书馆名称
createtime	timestamp		否	创建时间
realtime	timestamp		否	更新时间戳

（10） 校区表

校区表（t_campus），记录了校区的相关信息，如表 4-10 所示。

表 4-10 校区表

字段名称	字段类型	长度	是否主键	字段说明
id	varchar	50	是	id
code	varchar	255	否	代号
name	varchar	255	否	校区名称
createtime	timestamp		否	创建时间
realtime	timestamp		否	更新时间戳

（四） 系统编码设计

编码是数字或字符来代表事物或属性的符号[11]。在信息系统中，编码是人和机器共同的语言，是系统进行信息分类、鉴别、检索的关键[11]。本系统的基础配置、图书管理中都有使用编码。

在基础配置使用编码时，需使用“-”连接编码的多个含义部分，如图书馆编码“1-1-1”，表示“本学校的-第一个校区的-第一个图书馆”。在基础配置中的位置信息需使用“.”连接多个位置信息，如层架位置编码“002.01.1”，表示“测试间 1 架 A 面 .1 列 .1 层”。

在图书管理中用七位的数字编码表示本系统中的唯一一本书，如“1000762”表示“三国演义”这本书，该编码不允许重复。

（五） 系统输入和输出设计

系统的输入输出设计是先根据管理和用户的需要进行输出设计，然后反过来根据输出设计进行输入设计[11]。

1 输出设计

系统在输出方式方面主要采用屏幕输出的方式为主，同时在部分功能支持以文件输出的方式为辅。在输出内容方面图书信息中需要包含图书的详细位置信息，比如这本图书在哪个校区，哪个图书馆，哪个楼层，哪个图书室，哪个书架，哪个层架。

2　输入设计

系统在输入方式方面主要采用键盘输入的方式。在输入内容方面，因为输出时需要图书的详细位置信息，所以在输入时，需要输入学校的物理基础信息并与图书产生关联。

（六） 系统界面设计

系统界面是管理信息系统的门面。在界面设计时，充分考虑简单性、明了性、风格同一、色彩适中等相关原则[11]。将系统界面设计如图 4-5 所示。

图 4-5 系统界面图

在系统界面分为左右两个大部分。在左大部分的上面是系统 logo 和系统名称，下面是系统的菜单区域。在右大部分的上面是动态导航，会显示当前所在页面。在右大部分的下面显示主要的内容信息，会根据具体菜单的不同显示不同内容。

五 系统实现与测试

（一） 系统实现

本系统使用 IntelliJ IDEA 作为软件代码主要开发工具，同时使用 Adobe Photoshop 作为图片处理工具。下面将详细介绍各功能的实现。

1 系统管理的实现

系统管理功能主要是由系统管理人员使用。系统管理人员使用系统管理创建用户和读者相关信息，并给图书管理人员分配账号和给图书借阅人员分配读者。这里主要介绍系统登录和读者管理的实现。

（1） 系统登录

系统中的所有用户和读者都需要使用登录功能进入系统，然后才能使用系统中的其他功能。

系统登录功能的主要算法逻辑是，当用户登录时系统会判断用户输入的用户名和密码是否正确。如果正确将登录成功，生成登录令牌存入 Redis 中，同时返回给前端。如果不正确代码将抛出登录错误的异常，然后被系统捕获，同时返回登录失败的相关信息给前端。系统登录流程图如图 5-1 所示

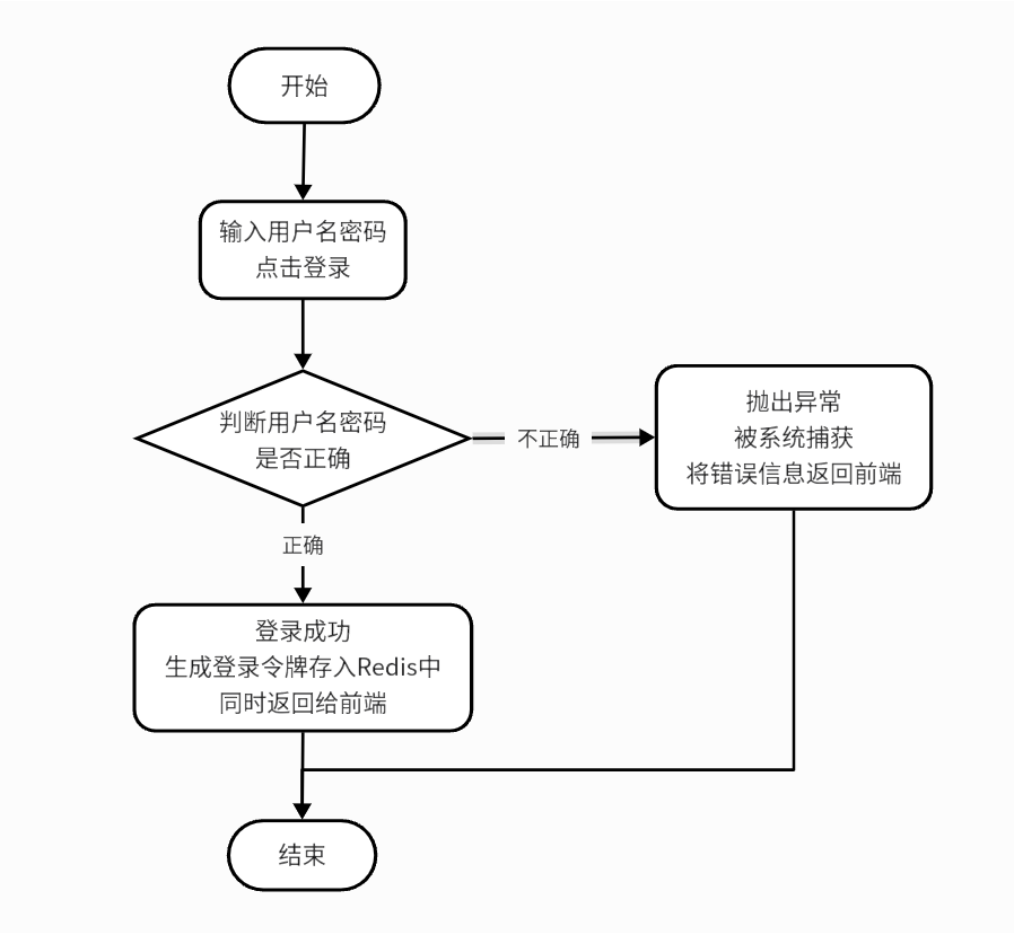

图 5-1 系统登录流程图

系统登录的关键代码如图 5-2 所示。

```
public AjaxResult login(@RequestBody LoginBody loginBody)
{
    AjaxResult ajax = AjaxResult.success();
    // 生成令牌
    String token = loginService.login(loginBody.getUsername(), loginBody.getPassword(), loginBody.getCode(),
            loginBody.getUuid());
    ajax.put(Constants.TOKEN, token);
    return ajax;
}
```

图 5-2 系统登录关键代码

系统登录的界面如图 5-3 所示。

图 5-3 系统登录界面

（2） 读者管理

在读者管理中可以新增、删除、修改读者，最主要的功能是给用户绑定读者信息。只有给用户绑定了读者信息，该账号才是完整的读者账号。

读者管理中给用户绑定读者信息的实现逻辑是，在打开读者新增或修改界面时，系统前端先去后端查询可以绑定的用户信息，然后系统管理人员选择一个用户和读者进行绑定，最后提交保存。新增读者的关键代码如图 5-4 所示。

```
public AjaxResult add(@RequestBody Readerinfo readerinfo)
{
    // 新增读者时，信用状态为0,即正常
    readerinfo.setState("0");
    return toAjax(readerinfoService.insertReaderinfo(readerinfo));
}
```

图 5-4 新增读者关键代码

新增读者的界面如图 5-5 所示。

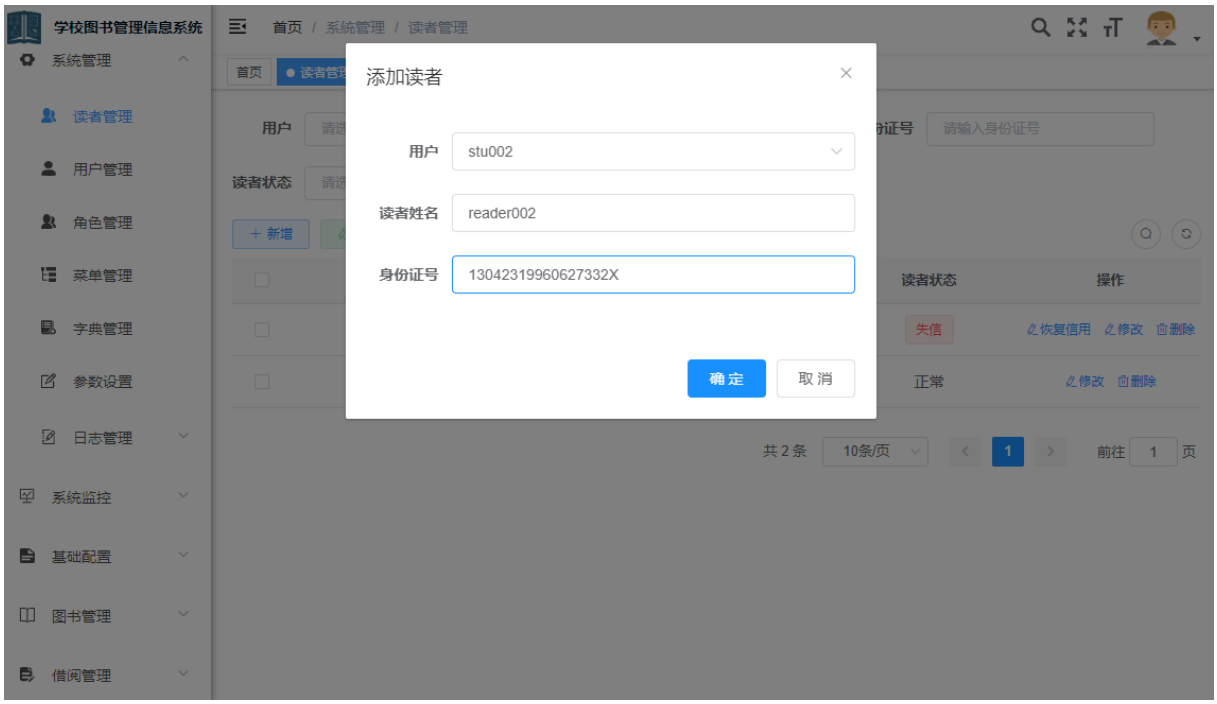

图 5-5 新增读者界面

2 基础配置的实现

基础配置主要是配置学校的基本物理信息，包括校区、图书馆、楼层等，主要是为了和图书产生关联，在查询寻图书时展现图书的精确物理位置。下面将介绍校区管理和图书馆管理的功能实现。

（1） 校区管理

该功能是记录学校校区的相关信息。该功能需要图书管理人员对信息进行维护和管理。新增校区的流程图如图 5-6 所示。

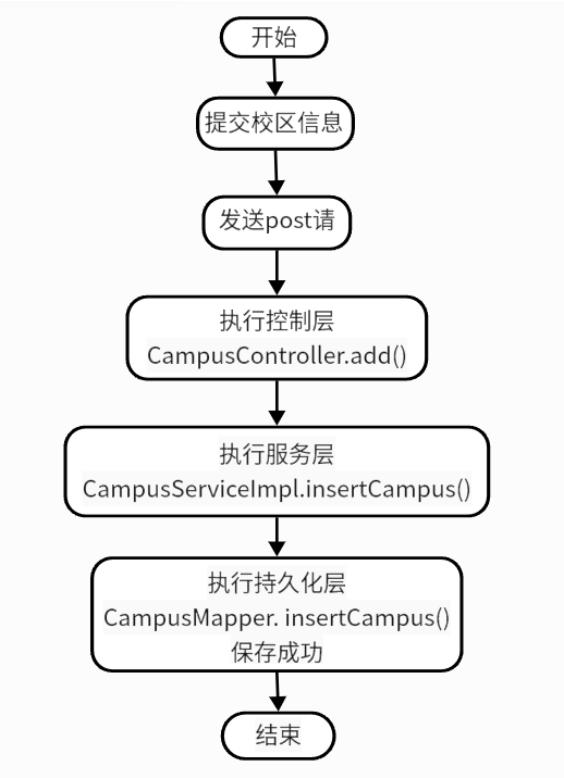

图 5-6 新增校区流程图

如图 5-6 所示。校区管理新增的主要实现逻辑是，图书管理人员在新增页面输入校区信息，然后点击提交按钮后，先执行前端代码发送 http 请求到后端控制层，然后后端系统执行控制层的 CampusController.add()方法，之后后端系统执行服务层的 CampusServiceImpl.insertCampus()方法，最后执行持久化层的 CampusMapper.insertCampus()方法完成数据的保存。服务层的新增关键代码如图 5-7 所示。

```
/**
 * 新增校区
 *
 * @param campus 校区
 * @return 结果
 */
1 usage
@Override
public int insertCampus(Campus campus)
{
    campus.setId(UUID.fastUUID().toString( isSimple: true));
    return campusMapper.insertCampus(campus);
}
```

图 5-7 新增校区关键代码

校区管理的列表界面如图 5-8 所示。

图 5-8 校区管理列表界面

（2） 图书馆管理

该功能是记录图书馆的相关信息，包括图书馆信息的增加、修改、删除和查询。在录入图书馆信息时需要依赖录入的校区的信息。该功能主要由图书管理人员使用。

图书馆新增功能的实现逻辑和校区管理的差不多，唯一的不同是在新增或修改时，需要选择该图书馆是属于那一个校区的。

图书馆管理的列表界面如图 5-9 所示。

图 5-9 图书馆管理界面

3 图书管理的实现

图书管理功能由图书管理人员和图书借阅人员使用。不同的角色在使用图书管理时所具有的功能是不同的。图书管理人员主要有图书的录入、删除、上架、下架等功能。图书借阅人员主要有图书的浏览和借阅功能。

图书管理人员和图书借阅人员所具有的功能不同，其实现方式主要是通过前端和后端两部分实现的。前端通过控制角色的权限来控制功能的不同，后端通过控制权限来控制功能的不同，也通过查找该用户是否拥有读者信息来控制是否可以使用该功能。

这里主要介绍一下借阅功能，图书管理借阅流程图如图 5-10 所示。

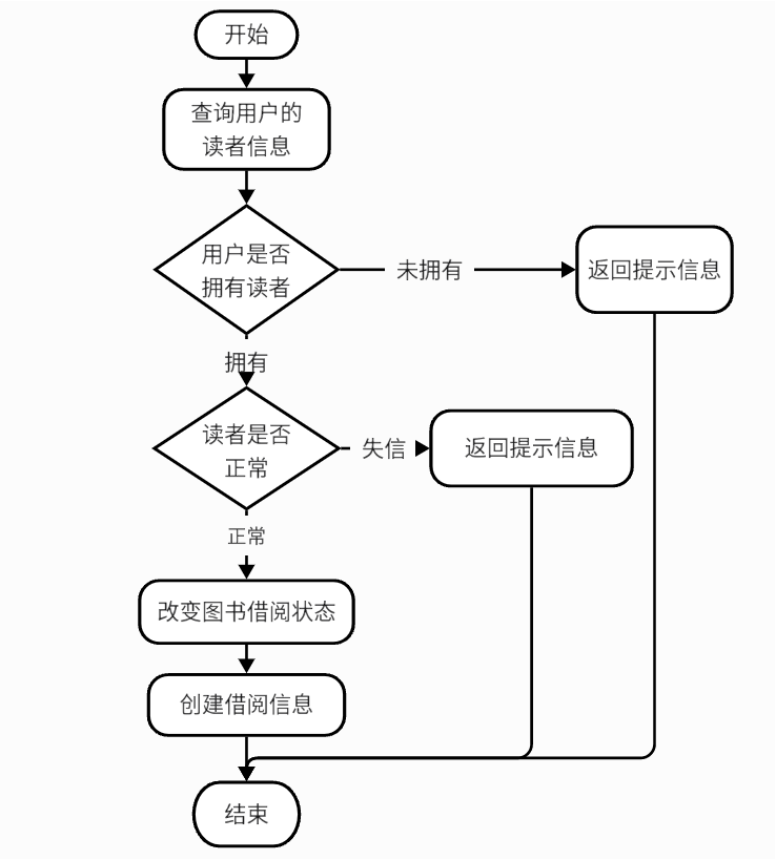

图 5-10 图书管理借阅流程图

图书借阅人员申请借阅时，系统首先会查询当前用户是否有读者信息，如果没有会直接返回，并提示用户没有读者信息。系统接着会判断当前读书是否处于失信状态，如果处于，会直接返回，并提示给用户相应信息。通过判断后，系统会将图书借阅状态变为在借，同时创建借阅记录。图书管理中借阅的核心代码如图 5-11 所示。

```java
@PreAuthorize("@ss.hasPermi('work:bookinfo:borrow')")
@Log(title = "图书", businessType = BusinessType.UPDATE)
@PutMapping("/borrow")
@Transactional
public AjaxResult borrow(@RequestBody Borrow borrow)
{
    // 判断当前用户是否有读者信息
    Readerinfo readerinfo = readerinfoService.selectReaderinfoByUserid(getUserId().toString());

    if(StringUtils.isNull(readerinfo)||StringUtils.isEmpty(readerinfo.getId())) {
        return AjaxResult.error( msg: "当前用户没有读者信息，请先完善读者信息");
    }else if ("1".equals(readerinfo.getState())){
        return AjaxResult.error( msg: "当前用户已被禁用，请联系管理员");
    }
    // 更新图书借阅状态
    Bookinfo bookinfo = new Bookinfo();
    bookinfo.setId(borrow.getBookid());
    bookinfo.setBorrowstate("1");
    bookinfoService.updateBookinfo(bookinfo);
    // 创建借阅记录
    borrow.setReaderid(readerinfo.getId());
    // 生成当前日期
    borrow.setStartdate(new Date());
    borrow.setState("0");
    AjaxResult success = AjaxResult.success(borrowService.insertBorrow(borrow));
    return success;
}
```

图 5-11 借阅核心代码

图书管理的列表界面如图 5-12 所示，该页面是读者视角的界面，在页面表格操作列中拥有借阅功能。

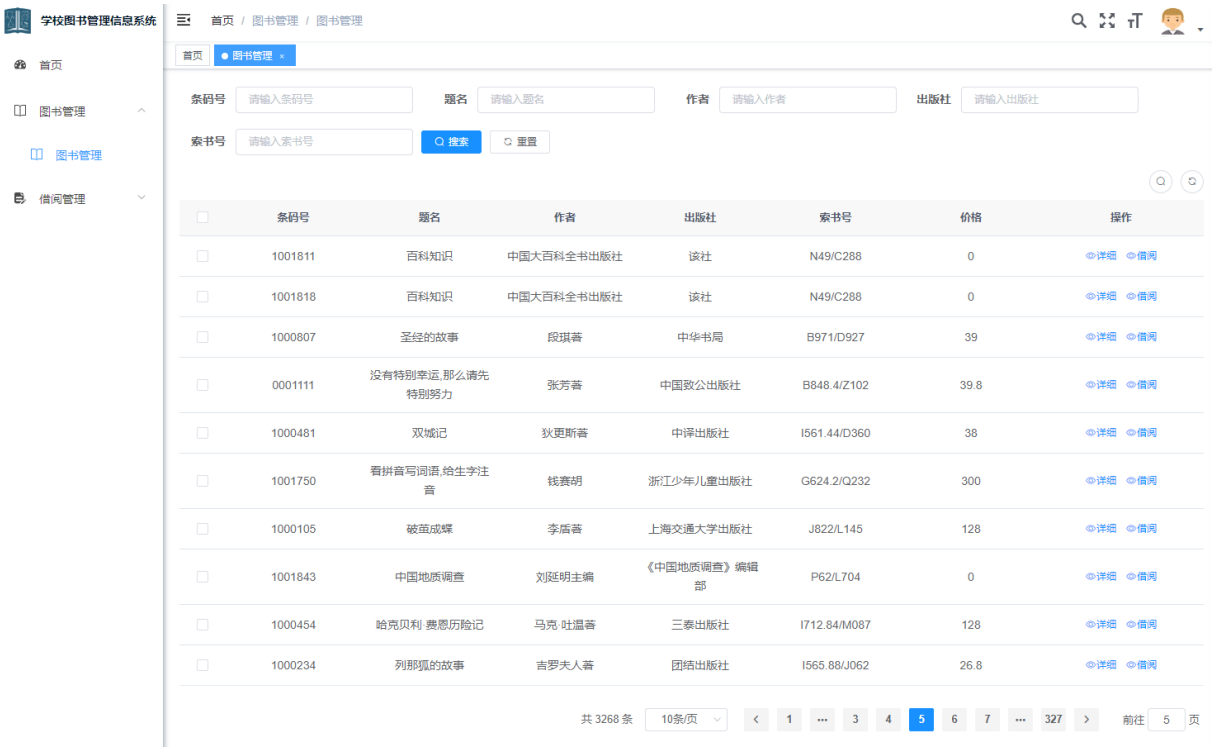

图 5-12 图书管理列表界面

4 借阅管理的实现

借阅管理中主要包含借阅申请管理、借阅审核管理、失信人管理。借阅申请管理主要由图书借阅人员使用，借阅审核管理和失信人管理主要由图书管理人员使用。下面将对这些功能的实现进行介绍。

（1） 借阅申请管理

图书借阅人员使用该功能可以提出续借申请、还书申请，同时也可以查看历史借阅记录。借阅申请管理列表界面如图 5-13 所示。

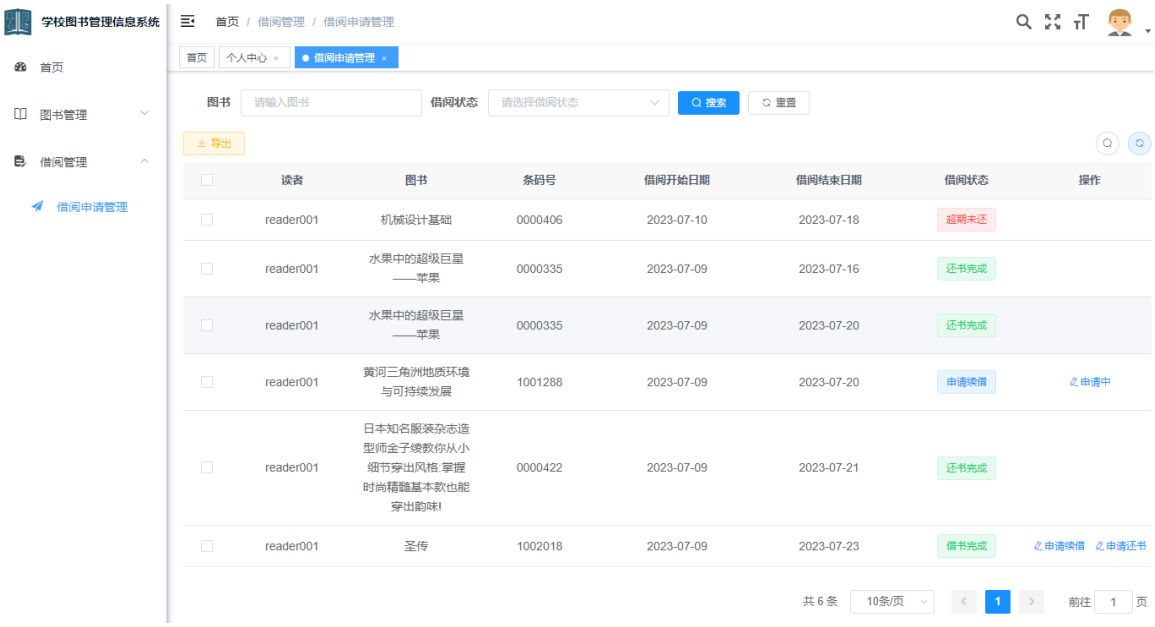

图 5-13 借阅申请管理列表界面

（2） 借阅审核管理

图书管理人员使用该功能审核通过或驳回借书申请、续借申请、还书申请，同时也可以查看历史借阅记录。审核通过还书申请的关键代码如图 5-14 所示。

```
    }else if ("4".equals(borrow.getState())) {
        // 还书审核通过
        borrow.setState("5");
        borrowService.updateBorrow(borrow);
        // 还书后修改图书状态
        Bookinfo bookinfo = new Bookinfo();
        bookinfo.setId(bookid);
        bookinfo.setBorrowstate("0");
        return toAjax(bookinfoService.updateBookinfo(bookinfo));
    }
    return success();
}
```

图 5-14 审核通过还书申请关键代码

借阅审核管理列表界面如图 5-15 所示。

图 5-15 借阅审核管理列表界面

（3） 失信人管理

超过借阅结束时间还未归还图书的读者会被系统自动移入失信人，处于失信状态的读者将不能借阅图书。如果图书借阅人员想要恢复信用，则需要图书管理人员在失信人管理中将失信的读者恢复信用。

失信人管理列表界面如图 5-16 所示。

图 5-16 失信人管理列表界面

图书借阅人员查看个人信息和读者信息界面如图 5-17 所示。

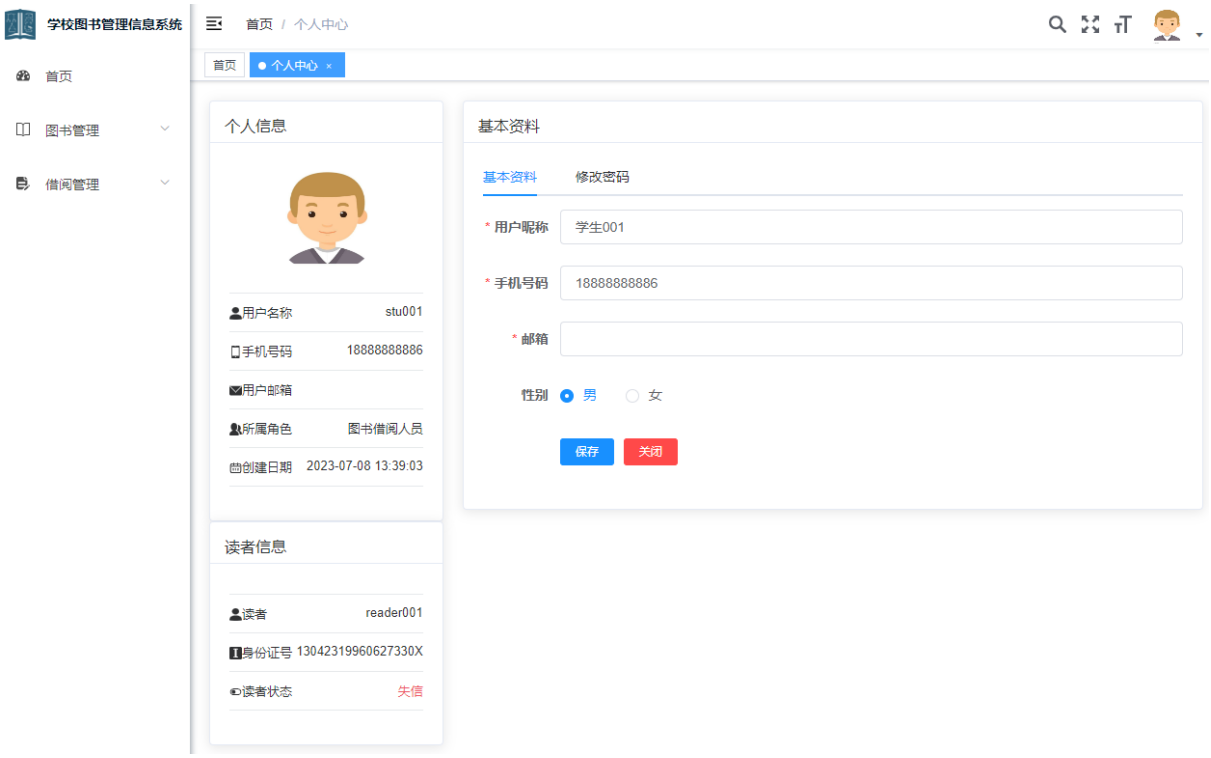

图 5-17 个人信息和读者信息界面

（二） 系统测试

系统测试是系统开发过程中不可缺少的一部分，通过测试可以检验系统是否达到了相关要求。下面将从以下几个方面介绍系统测试。

1 运行环境搭建

本系统采用 B/S 架构，拥有很好的兼容性，支持多种浏览器，如谷歌、edge 等，同时系统拥有很好的跨平台性。下面详细介绍系统的运行环境。

在运行环境方面需要 Java、MySQL、Redis、Maven、Node、Nginx，因为后端项目由 Maven 工具构建，打成 jar 包。后端项目通过 jar 包中内置的 Tomcat 运行，所以不需要额外的 Tomcat。具体运行环境列表如表 5-1 所示。

表 5-1 运行环境软件参照表

软件名称	版本	用途
Java	1.8 及以上	后端系统开发及运行环境
MySQL	5.7 及以上	数据存储
Redis	3.0 及以上	缓存数据存储
Maven	3.0 及以上	后端系统构建工具
Node	12 及以上	前端系统开发环境
Nginx	1.16.1 及以上	前端部署环境

运行过程如下：

后端：

（1）执行 bookManager-backend\sql\ book_manage.sql 脚本，创建 MySQL 数据库。

（2）执行后端项目的 Spring Boot 启动类。

前端：

（1）执行 cd bookManager-frontend，进入前端根目录。

（2）执行 npm install，加载项目依赖。

（3）执行 npm run dev，启动项目。

部署过程如下：

后端：

（1）执行 bookManager-backend\sql\ book_manage.sql 脚本，创建 MySQL 数据库。

（2）执行 Maven 工具的 package 命令将项目构建成 jar 包。

（3）使用 java -jar 命令执行打好的 jar 包程序。

前端：

（1）执行 cd bookManager-frontend，进入前端根目录。

（2）执行 npm run build:prod，将前端项目构建到 dist 文件夹

（3）将 dist 文件夹发布到 Nginx 静态服务器即可。

2 测试方法介绍

系统测试可以分为动态测试和静态测试。动态测试是指对计算机中实际运行的程

序进行测试，一般的测试方法有黑盒测试和白盒测试[11]。黑盒测试是功能性测试，是指不了解软件代码结构，根据系统业务功能对系统进行测试[11]。白盒测试是结构性测试，是指明确代码流程，根据代码逻辑设计用例，进行用例覆盖[11]。静态测试是指被测试的程序不在计算机上运行，采用人工检测和计算机辅助静态分析的手段对程序进行检测。常用的方式有桌前检查、代码审查、代码走查的方式。

3 系统功能测试

本文将采用黑盒测试对系统功能进行测试，检验系统功能是否正常。下面将从系统管理、基础配置、图书管理、借阅管理进行系统功能测试。

（1） 系统管理

系统管理模块的测试结果如表 5-2 所示。

表 5-2 系统管理测试用例

测试用例	操作步骤	期望结果	测试结果
系统登录	输入用户名、密码，点击登录	用户名、密码正确时，登录成功，不正确时，登录失败，提示相关信息	通过
用户角色权限管理	验证菜单权限功能是否正常；验证角色功能是否正常；验证用户角色功能是否正常	用户角色权限的增加、删除、修改、查询、给角色分配权限、给用户分配角色，这些功能正常	通过
读者管理	点击新增验证新增读者和给用户分配读者功能；点击修改验证修改读者和给用户分配读者功能；点击删除验证删除读者功能；点击搜索验证查询读者功能；	读者新增功能正常；读者修改功能正常；读者删除功能正常；读者查询功能正常	通过

（2） 基础配置

由于本模块的功能较多，这里主要对图书馆管理的测试结果进行介绍。图书馆管

理的测试结果如表 5-3 所示。

表 5-3 图书馆管理测试用例

测试用例	操作步骤	期望结果	测试结果
新增图书馆	点击新增按钮，选择图书馆所属校区，输入相关信息，点击保存	新增图书馆成功	通过
修改图书馆	点击修改按钮，修改相关信息，点击保存	修改图书馆成功	通过
查询图书馆	输入检索条件，点击搜索按钮	查询图书馆成功，展示出相关检索条件的图书	通过
删除图书馆	点击删除按钮，点击确定按钮	删除图书馆成功	通过

（3） 图书管理

图书管理模块的测试结果如表 5-4 所示。

表 5-4 图书管理测试用例

测试用例	操作步骤	期望结果	测试结果
图书录入	点击新增按钮，输入图书数据，点击提交按钮	图书新增成功	通过
图书上架	点击上架按钮	图书上架成功	通过
图书下架	点击下架按钮	图书下架成功	通过
图书查询	在列表上方查询区域输入查询信息，点击搜索按钮	图书列表展现出相关信息。	通过
申请借阅	点击借阅按钮，输入借阅结束时间，点击提交	借书申请成功，并且在借阅管理中国有相关记录	通过

（4） 借阅管理

借阅管理模块的测试结果如表 5-5 所示。

表 5-5 借阅管理测试用例

测试用例	操作步骤	期望结果	测试结果
失信人管理	点击失信人菜单，进入失信人列表页面，点击恢复信用	列表页面展示读书信息，失信读者后方恢复信用按钮，点击按钮读者恢复到正常状态	通过
借阅申请管理中申请续借	点击借阅申请管理菜单，进入列表界面，点击申请续借按钮，输入完成日期，点击提交	申请续借成功，同时借阅审核管理中借阅记录状态改变。	通过
借阅申请管理中申请还书	点击借阅申请管理菜单，进入列表界面，点击申请还书按钮	申请还书成功，同时借阅审核管理中借阅记录状态改变	通过
借阅审核管理中审核通过	点击借阅审核管理菜单，进入列表界面，点击审核通过按钮	审核成功，借阅记录的状态改变，当借阅状态变为还书完成时，图书借阅状态变为在馆	通过
借阅审核管理中驳回申请	点击借阅审核管理菜单，进入列表界面，点击驳回按钮	驳回成功，借阅记录状态变为申请钱的状态	通过

4 测试结果分析

经过对系统进行全面的测试，得出了以下测试结果：

系统的功能表现出色，所有设计的功能都得到了有效实现。用户可以轻松地完成图书借阅、归还、续借等操作，管理员可以进行图书管理、用户管理等操作，系统的各项功能都能正常运行。

系统经过长时间的运行和多次测试，表现出了很好的稳定性。没有出现系统崩溃、数据丢失等问题。系统能够持续稳定地运行，为用户提供可靠的服务。

综上所述，通过对系统的全面测试，我们可以得出结论，系统在功能和稳定性方面表现出色。系统已经准备好进行实际的上线和部署，为用户提供高质量的服务。

六 总结与展望

（一） 工作总结

在信息革命的背景下，各行各业都在向着信息化、网络化的方向转变。处在这样一个时代，学校图书管理向着信息化的方向转变也是历史发展下的必然结果。本文设计实现的学校图书管理信息系统，也是顺应历史发展的趋势，它将带动学校图书管理从人工手动管理转变为信息化管理，同时也将带动信息孤岛的信息化转变为整体共享的信息化。

本设计在实现时，使用了当前最为成熟的信息技术。从而极大的提升了系统的扩展性、稳定性、健壮性。

在本次系统设计过程中，翻阅了大量相关资料，阅读了中外文献，学习了业内先进技术。通过互联网上的技术文档，不断修正设计和技术方案，最终形成了更合理的信息系统。

作为一个程序开发人员，自己对自己的本职工作比较熟悉，也就只限于编码和实现。对于软件工程的其他方面相对比较陌生，多数概念大多都是停留在书本上，没有经过具体的实践。

通过这次实践，有了将理论和实践相结合的机会。从软件开发的系统分析、系统设计到软件开发的系统实现、系统测试，将所有的工作紧密连接，从而有了更加整体化，更加系统化，看待整个软件工程的机会。从而对整个软件工程的系统化开发有了更加深刻的理解和感悟。

（二） 工作展望

虽然本设计实现对学校的图书管理系统做了很大程度上的信息化改造，如从手工到信息化，从信息孤岛到信息共享，但是由于时间、资源、能力等有限，本系统还存在着许多地方的不足：

（1）引入物联网技术，比如 RFID 射频识别技术。现在系统主要还是使用人工的方式实现对信息的识别，加入 RFID 射频识别技术以后，只需要将 RFID 信息和图书信息进行绑定就可以实现对图书的自动化识别，从而极大的提高信息化程度。

（2）引入当前最火的人工智能和大数据技术。系统在现实的使用过程中会产生大

量的数据，如图书搜索的信息，图书浏览的信息，借阅图书的信息等等。将这些大量的、低价值密度的数据[10]交给人工智能，通过选择合适的算法，让人工智能从中挖掘出有价值的信息。如通过历史图书的搜索量、浏览量、借阅量，预测出未来人类浏览、借阅图书的趋势，从而更好的为人类服务。

相信在新技术不断引入，物联网、大数据、人工智能、机器人等技术不断融合的形式下，在可以预见的未来中，人类信息化一定会得到卓越的发展，图书管理工作也将取得全新的变化，这将协助人类迈向更加文明的阶梯。

七 参考文献

[1] 王美杰.高校图书馆地方文献阅读推广创新路径研究——以怀化学院图书馆为例[J].怀化学院学报,2020,39(03):107-110.DOI:10.16074/j.cnki.cn43-1394/z.2020.03.023.

[2] 何辉娟.基于大数据的智能图书管理系统的研究与设计[J].电子技术与软件工程,2023(06):222-225.

[3] 徐晨,程红.政府会计制度改革对高校资产管理的影响研究[J].中国乡镇企业会计,2020(05):122-123.

[4] 赵冰.计算机管理系统在校园图书馆中的应用[J].电子技术,2023,52(03):250-251.

[5] 张俊明.计算机在图书馆资源管理中的应用[J].信息与电脑(理论版),2019(16):16-18.

[6] 邓阳名,顾潇,梁爱媚.基于 Java 语言的数据库访问技术应用研究[J].中国新通信,2023,25(08):83-85.

[7] 张旭刚,张昕,高若寒.基于 Spring Boot 与 MyBatis 框架构建动态读写分离模型[J].微型电脑应用,2021,37(02):84-86+98.

[8] 张欣悦.我国人口老龄化的现状特点和发展趋势及其对策研究[J].中国管理信息化, 2020, 23(05): 195-199.

[9] 于也雯, 龚六堂. 生育政策、生育率与家庭养老[J]. 中国工业经济, 2021(05): 38-56. DOI: 10.19581/j.cnki.ciejournal.2021.05.002.

[10] 黄靖. 数据库系统原理[M]. 北京: 机械工业出版社, 2018.

[11] 刘世峰. 信息系统开发与管理[M]. 北京: 机械工业出版社, 2011.

附录

计算机主要程序清单：

```
/**
 * 图书 Controller
 *
 * @author dream
 * @date 2023-07-05
 */
@RestController
@RequestMapping("/work/bookinfo")
public class BookinfoController extends BaseController
{
    @Autowired
    private IBookinfoService bookinfoService;

    @Autowired
    private IReaderinfoService readerinfoService;

    @Autowired
    private IBorrowService borrowService;

    /**
     * 查询图书列表
     */
    @PreAuthorize("@ss.hasPermi('work:bookinfo:list')")
    @GetMapping("/list")
    public Object list(Bookinfo bookinfo)
    {

        Readerinfo readerinfo = readerinfoService.selectReaderinfoByUserid(getUserId().toString());
```

```
    startPage();
    List<Bookinfo> list = bookinfoService.selectBookinfoList(bookinfo);
    return this.getDataTableAjaxResult(list).put("readerinfo",readerinfo);
}

/**
 * 导出图书列表
 */
@PreAuthorize("@ss.hasPermi('work:bookinfo:export')")
@Log(title = "图书", businessType = BusinessType.EXPORT)
@PostMapping("/export")
public void export(HttpServletResponse response, Bookinfo bookinfo)
{
    List<Bookinfo> list = bookinfoService.selectBookinfoList(bookinfo);
    ExcelUtil<Bookinfo> util = new ExcelUtil<Bookinfo>(Bookinfo.class);
    util.exportExcel(response, list, "图书数据");
}

/**
 * 获取图书详细信息
 */
@PreAuthorize("@ss.hasPermi('work:bookinfo:query')")
@GetMapping(value = "/{id}")
public AjaxResult getInfo(@PathVariable("id") String id)
{
    return success(bookinfoService.selectBookinfoById(id));
}

/**
 * 新增图书
 */
@PreAuthorize("@ss.hasPermi('work:bookinfo:add')")
@Log(title = "图书", businessType = BusinessType.INSERT)
@PostMapping
```

```
    public AjaxResult add(@RequestBody Bookinfo bookinfo)
    {
        bookinfo.setBorrowstate("0");
        bookinfo.setBookstatus("0");
        return toAjax(bookinfoService.insertBookinfo(bookinfo));
    }

    /**
     * 修改图书
     */
    @PreAuthorize("@ss.hasPermi('work:bookinfo:edit')")
    @Log(title = "图书", businessType = BusinessType.UPDATE)
    @PutMapping
    public AjaxResult edit(@RequestBody Bookinfo bookinfo)
    {
        return toAjax(bookinfoService.updateBookinfo(bookinfo));
    }

    /**
     * 修改图书状态
     */

@PreAuthorize("@ss.hasAnyPermi('work:bookinfo:up,work:bookinfo:out,work:bookinfo:bor
row')")
    @Log(title = "图书", businessType = BusinessType.UPDATE)
    @PutMapping("/changeStatus")
    public AjaxResult changeStatus(@RequestBody Bookinfo bookinfo)
    {
        return toAjax(bookinfoService.updateBookinfoStatus(bookinfo));
    }

    /**
     * 借阅图书
     */
```

```
    @PreAuthorize("@ss.hasPermi('work:bookinfo:borrow')")
    @Log(title = "图书", businessType = BusinessType.UPDATE)
    @PutMapping("/borrow")
    @Transactional
    public AjaxResult borrow(@RequestBody Borrow borrow)
    {
        // 判断当前用户是否有读者信息
        Readerinfo                              readerinfo                              =
readerinfoService.selectReaderinfoByUserid(getUserId().toString());

        if(StringUtils.isNull(readerinfo)||StringUtils.isEmpty(readerinfo.getId()))
{
            return AjaxResult.error("当前用户没有读者信息，请先完善读者信息");
        }else if ("1".equals(readerinfo.getState())){
            return AjaxResult.error("当前用户已被禁用，请联系管理员");
        }
        // 更新图书借阅状态
        Bookinfo bookinfo = new Bookinfo();
        bookinfo.setId(borrow.getBookid());
        bookinfo.setBorrowstate("1");
        bookinfoService.updateBookinfo(bookinfo);
        // 创建借阅记录
        borrow.setReaderid(readerinfo.getId());
        // 生成当前日期
        borrow.setStartdate(new Date());
        borrow.setState("0");
        AjaxResult success = AjaxResult.success(borrowService.insertBorrow(borrow));
        return success;
    }

    /**
     * 删除图书
     */
    @PreAuthorize("@ss.hasPermi('work:bookinfo:remove')")
```

```
    @Log(title = "图书", businessType = BusinessType.DELETE)
    @DeleteMapping("/{ids}")
    public AjaxResult remove(@PathVariable String[] ids)
    {
        return toAjax(bookinfoService.deleteBookinfoByIds(ids));
    }
}

/**
 * 图书Service接口
 *
 * @author dream
 * @date 2023-07-05
 */
public interface IBookinfoService
{
    /**
     * 查询图书
     *
     * @param id 图书主键
     * @return 图书
     */
    public Bookinfo selectBookinfoById(String id);

    /**
     * 查询图书列表
     *
     * @param bookinfo 图书
     * @return 图书集合
     */
    public List<Bookinfo> selectBookinfoList(Bookinfo bookinfo);

    /**
     * 新增图书
```

```
     *
     * @param bookinfo 图书
     * @return 结果
     */
    public int insertBookinfo(Bookinfo bookinfo);

    /**
     * 修改图书
     *
     * @param bookinfo 图书
     * @return 结果
     */
    public int updateBookinfo(Bookinfo bookinfo);

    int updateBookinfoStatus(Bookinfo bookinfo);

    /**
     * 批量删除图书
     *
     * @param ids 需要删除的图书主键集合
     * @return 结果
     */
    public int deleteBookinfoByIds(String[] ids);

    /**
     * 删除图书信息
     *
     * @param id 图书主键
     * @return 结果
     */
    public int deleteBookinfoById(String id);
}

/**
```

```
 * 图书Service业务层处理
 *
 * @author dream
 * @date 2023-07-05
 */
@Service
public class BookinfoServiceImpl implements IBookinfoService
{
    @Autowired
    private BookinfoMapper bookinfoMapper;

    /**
     * 查询图书
     *
     * @param id 图书主键
     * @return 图书
     */
    @Override
    public Bookinfo selectBookinfoById(String id)
    {
        return bookinfoMapper.selectBookinfoById(id);
    }

    /**
     * 查询图书列表
     *
     * @param bookinfo 图书
     * @return 图书
     */
    @Override
    public List<Bookinfo> selectBookinfoList(Bookinfo bookinfo)
    {
        return bookinfoMapper.selectBookinfoList(bookinfo);
    }
```

```
/**
 * 新增图书
 *
 * @param bookinfo 图书
 * @return 结果
 */
@Override
public int insertBookinfo(Bookinfo bookinfo)
{
    bookinfo.setId(UUID.fastUUID().toString(true));
    return bookinfoMapper.insertBookinfo(bookinfo);
}

/**
 * 修改图书
 *
 * @param bookinfo 图书
 * @return 结果
 */
@Override
public int updateBookinfo(Bookinfo bookinfo)
{
    return bookinfoMapper.updateBookinfo(bookinfo);
}

/**
 * 批量修改图书状态
 *
 * @param bookinfo 图书
 * @return 结果
 */
@Override
public int updateBookinfoStatus(Bookinfo bookinfo)
```

```
    {
        return bookinfoMapper.updateBookinfoStatus(bookinfo);
    }

    /**
     * 批量删除图书
     *
     * @param ids 需要删除的图书主键
     * @return 结果
     */
    @Override
    public int deleteBookinfoByIds(String[] ids)
    {
        return bookinfoMapper.deleteBookinfoByIds(ids);
    }

    /**
     * 删除图书信息
     *
     * @param id 图书主键
     * @return 结果
     */
    @Override
    public int deleteBookinfoById(String id)
    {
        return bookinfoMapper.deleteBookinfoById(id);
    }
}
```