

一、系统分析

（一）需求分析

1.系统简述

图书管理系统是一个集成了现代信息技术、数据库技术、网络技术等多元素于一体的复杂系统,它主要服务于图书馆、书店等图书管理机构,旨在通过高效、准确的信息化手段,实现图书资源的查询、借阅、归还、补充等一系列管理活动。该系统不仅广泛应用于公共图书馆、学校图书馆、专业图书馆等不同类型的图书馆,还逐渐渗透到各类书店中,成为提升服务质量、提高工作效率的重要工具。通过图书管理系统,图书管理机构可以实现对图书资源的全面信息化管理,同时收集和分析用户信息,为用户提供更加个性化的服务体验。

2.系统开发的目标

图书管理系统开发的目标旨在通过技术手段提升图书馆或相关图书管理机构的运营效率和用户体验,同时确保资源的有效管理和利用。以下是图书管理系统开发的一些核心目标:

(1) 提升管理效率

- **自动化流程:** 通过自动化图书的录入、分类、检索、借阅、归还等流程,减少人工操作的繁琐和耗时,降低人为错误的发生概率,从而提高整体工作效率;
- **库存管理:** 实时跟踪图书的库存情况,包括在库、借出、预订等状态,帮助管理员及时了解图书资源的分布和利用情况,以便做出更加精准的采购或调配决策;
- **数据分析:** 通过收集和分析借阅数据,为图书馆提供关于读者偏好、图书需求趋势等有价值的信息,以指导未来的图书采购和服务优化方向。

(2) 优化用户体验

- **便捷查询:** 提供多种查询方式,如关键词搜索、分类浏览、高级检索等,使用户能够快速准确地找到所需图书。同时,系统还可以根据用户的查询历史和偏好,智能推荐相关书目,提高检索效率。;
- **在线服务:** 开发在线查询、借阅等功能,让用户无需亲自到图书馆即可享受图书服务。此外,系统还可以提供电子书阅读、音频书播放等在线阅读服务,进一步丰富用户的阅读体验;
- **个性化推荐:** 基于用户的借阅历史和偏好,系统可以生成个性化的图书推荐列表,帮助用户发现更多感兴趣的图书资源。

(3) 确保资源安全

- **数据保护:** 采用加密技术保护用户信息和图书数据的安全,防止数据泄露或被篡改;
- **账号管理:** 对不同账号用户设置不同的访问权限,确保只有授权人员才能访问敏感数据或执行关键操作;

- **备份与恢复**：定期备份数据，确保在发生系统故障或数据丢失时能够迅速恢复。

3.数据的汇总与分析

(1) 管理功能调查与分析

该图书管理系统可以划分出几个主要的管理功能，依次是用户管理、图书管理、借阅管理、预定管理、搜索板块以及提醒和通知板块。

A.用户管理

用户根据已注册的信息登录图书管理系统，根据账号的不同，不同用户被授予了不同的权限，能够借阅的书籍类别、书籍数目有所不同，能够对书籍信息进行操作的权限也有所不同。

B.图书管理

该图书管理系统能够对书籍的动向进行追踪。首先，有权限的账号能够对在库书籍进行查询、添加、编辑和删除等。其次，书籍能够分门别类地被追踪管理，例如按照作者、出版社或主题等进行分类。最后，该系统需要能够对图书的状态进行跟踪，如在库、借出、预定等。

C.借阅管理

该图书管理系统能够对书籍的借阅情况进行追踪，并完成借阅和归还操作。首先能够通过该系统完成书籍的借出和归还；其次能够通过该系统规定不同用户能拥有的借书时长，并且对逾期未还的书目进行汇总管理；最后该系统还能查阅每本书的借阅历史。

D.搜索板块

用户可以根据书名、作者、ISBN 编号、出版社等基本信息对图书进行检索，同时也可以组合搜索条件来实现高级搜索。

E.提醒和通知板块

该图书管理系统能够向用户发生通知，帮助其更好地配合图书馆工作并且满足自身需求。例如可以发送借阅到期提醒、预订图书到货通知以及系统维护和更新通知等。

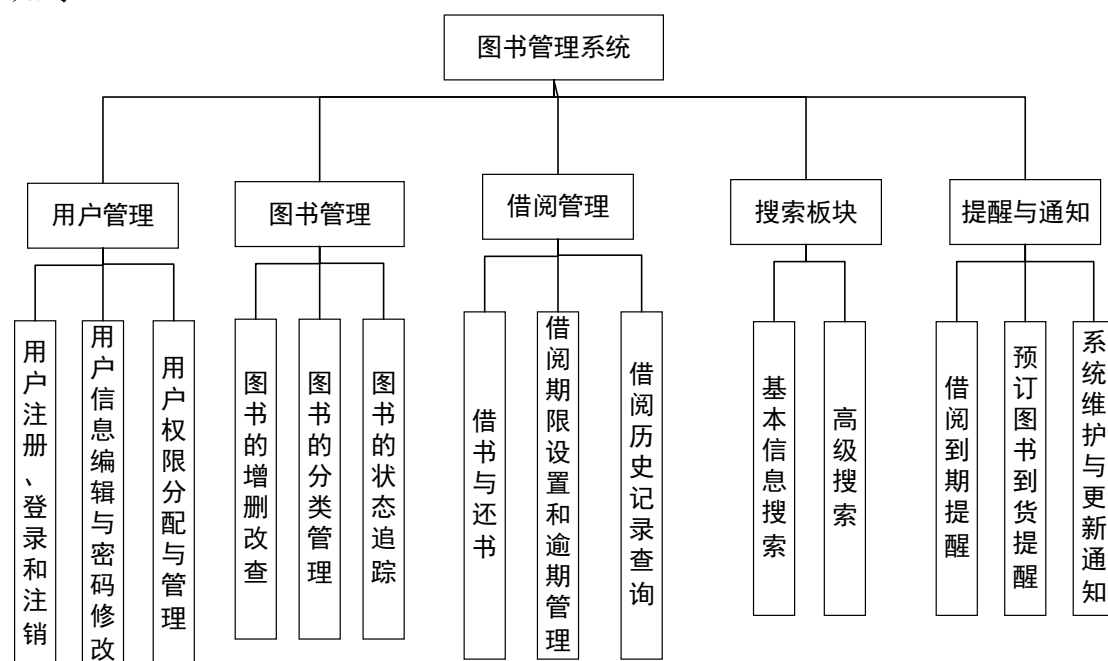


图 1 管理功能汇总

图书管理系统是一个能够进行管理信息的收集、传递、加工、保存、维护和使用的系统。它主要针对图书馆的功能范围及特点,制定相应的子系统互相连接,以达到高效运作的目的。这些子系统通常会涵盖图书馆的主要业务和工作,如图书的进购、编号、检索、统计和借阅情况等,全面实现这些业务的计算机管理。具体来说,图书管理系统具备以下主要功能:

- A.图书录入:** 包括图书的基本信息录入,如书名、作者、出版社、ISBN 号等,还可添加图书封面、图书简介等附加信息;
- B.图书查询:** 系统使用者可以根据关键词、作者、出版社等条件进行图书查询,并可查看图书详细信息,以便了解图书情况。系统还支持智能推荐相关书目,提高检索效率;
- C.图书借阅与归还:** 用户可以通过系统预定图书、自助借阅和归还图书。在借阅过程中,系统会自动更新图书的借阅状态,并提醒用户还书日期。归还时,系统也会自动完成相关归还操作;
- D.图书与用户管理:** 管理员可以对图书信息进行增加、删除、修改和查询等操作,方便管理图书馆的图书资源。同时,管理员还可以对用户信息进行管理,包括用户的基本信息和借阅记录等;
- E.信息记录:** 系统可以记录各类信息数据,包括用户账号登记、访问,图书的借阅与归还,图书数据库的更新等信息数据。

(2) 划分子系统

A.图书借预还子系统

功能	详情
图书查询	除了书名、具体编号搜索外,还可通过作者、出版社、出版时间、图书类型、图书主题等一系列关联词进行交叉检索来寻找所需的图书;
图书借阅	用户选择要借阅图书,输入账号信息并办理借阅手续后,实现图书的借阅,同时系统记录借阅信息;
图书归还	图书借阅期到时,用户输入账号信息和书籍信息后,完成归还图书的操作,系统更新借阅记录,检查图书状态。

B.图书更新与管理子系统

功能	详情
记录书目信息	记录所收纳图书的所有信息,以支持图书查询的系统功能;
图书数据的变动	通过图书管理员来更新已存在图书的信息,如价格变动、库存调整等;
图书的录入	图书管理员添加新书信息,包括书名、作者、ISBN、出版社、出版日期、分类、价格、库存量等;
图书删除	图书损毁、遗失或下架时,管理员按照名录移除不再需要的图书记录。

C.用户管理子系统

功能	详情
用户账号管理	管理用户账号注册、更改、注销等操作,同时记录用户信息,包括姓名、性别、年龄、联系方式、借阅证号等;
用户借阅记录	用户每次借阅图书时,系统将自动记录用户的借阅历史,包括

	借阅时间、归还时间、借阅图书等；
用户权限设置	系统可以按照用户身份设置不同用户类型的借阅权限，如借阅数量限制、借阅时长、可借阅的图书门类等。

D.通知与提醒子系统

功能	详情
借阅提醒	系统会在图书借阅即将到期时向用户发送信息，提醒读者归还；
系统公告	当系统要维护或升级而停用时，系统会提前发布系统维护、升级等通知消息告知用户。

（二）管理业务流程分析

1.总述

本图书管理系统的业务流程基本如下所示，通过该图书管理系统，主要能完成以下几个方面的工作：

首先，借阅者提供自己的借阅需求数目清单，通过自主查询或经由图书管理员，使用系统在图书总库中查看该书是否存在。如果存在，便将需求书目提供给借阅者，图书总库中该书登记为已借出。

然后，还书者将书籍在管理员处登记归还，图书总库中该书登记为在库中。

最后，书籍采购者根据需新增的书目清单采购图书，再将新购买的书目清单总结发送给图书管理员，图书管理员再结合需要清理的已破损、已丢失书目清单定期更新图书总库。

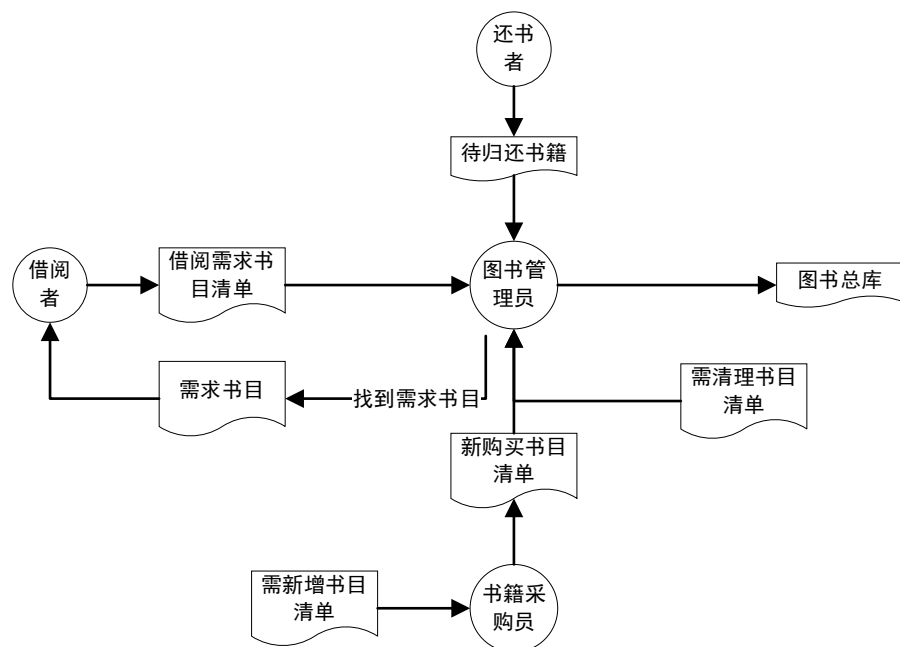


图2 图书管理系统总体业务流程图

2. 用户登录

用户输入账号和密码，系统再验证用户信息，如果信息正确则登陆成功，进入系统，信息错误则登录失败，流程结束。

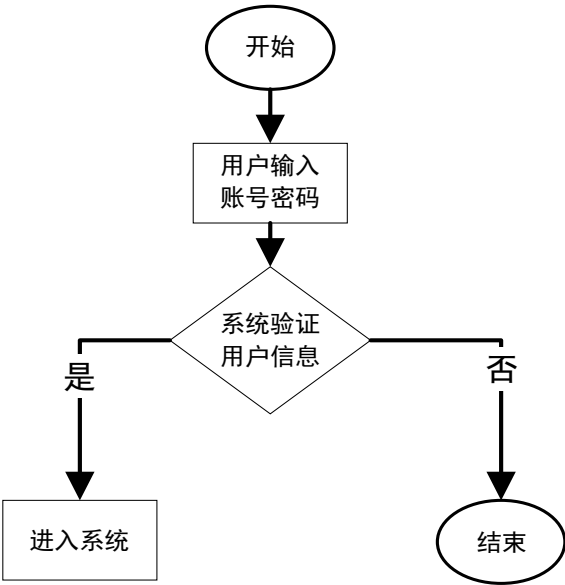


图 3 用户登录流程图

3. 查询书籍

用户输入查询条件，系统根据条件搜索书籍。如果找到书籍便显示书籍列表，流程结束；如未找到书籍便显示无结果信息，流程结束。

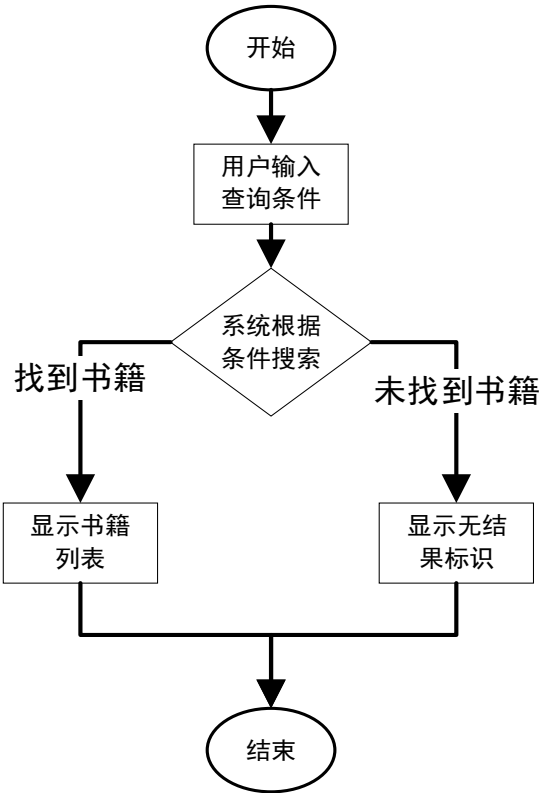


图 4 查询书籍流程图

4. 借阅书籍

用户选择书籍借阅，系统检查该书籍状态。若可借，用户则确认借阅，系统更新用户借阅信息，流程结束；若不可借，系统显示不可借信息，流程结束。

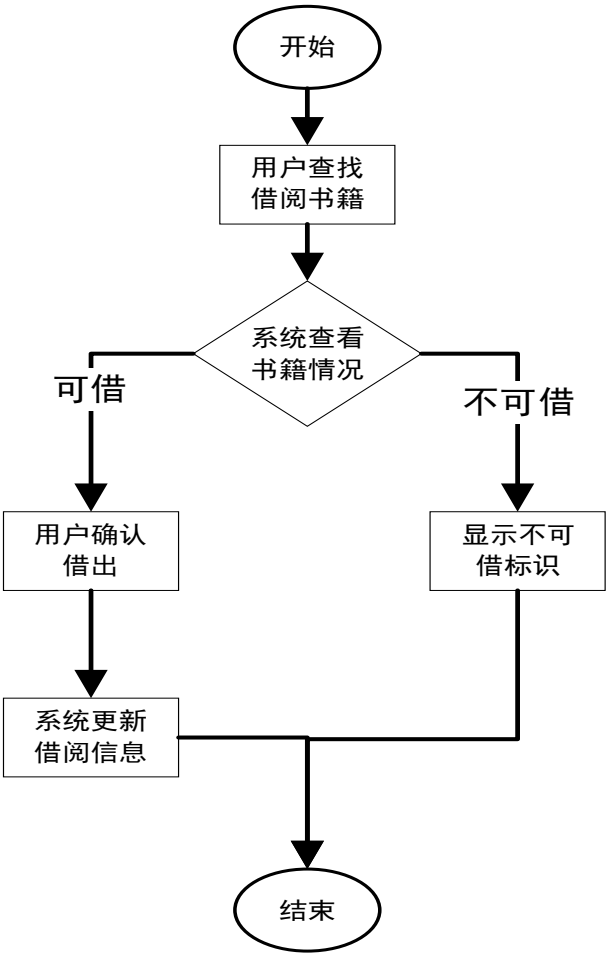


图 5 借阅书籍流程图

5. 归还书籍

用户输入归还书籍信息，系统检查归还书籍状态。若正常归还，系统则更新归还信息，结束；若逾期归还，系统则计算逾期费用，用户支付费用，再更新归还信息，结束。

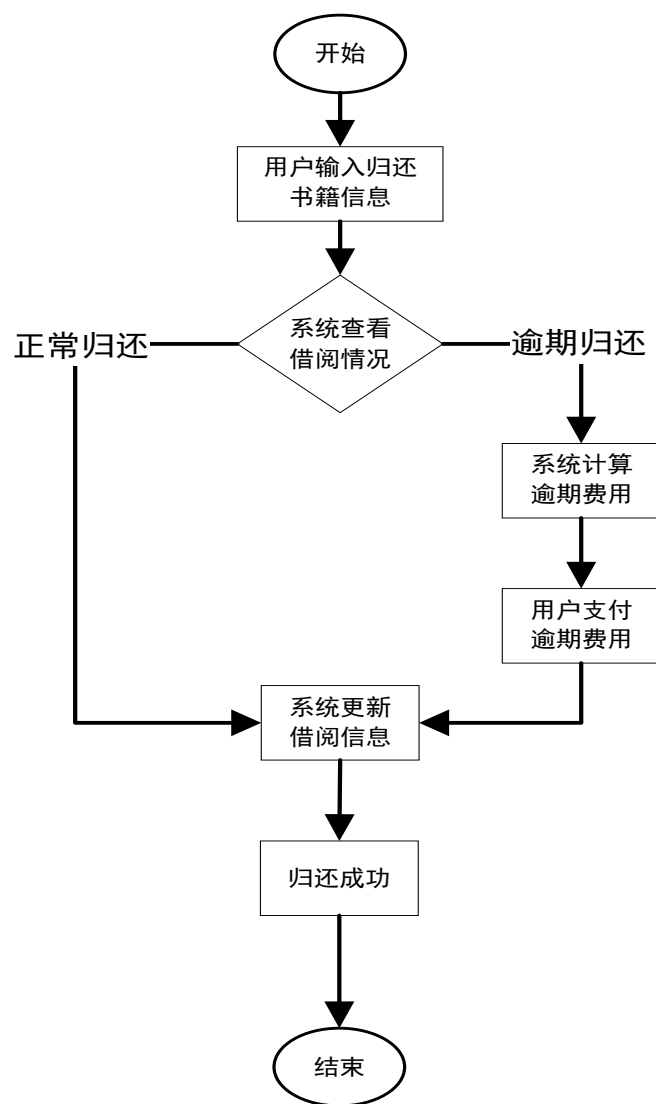


图 6 归还书籍流程

6. 管理员添加/删除书籍

管理员输入欲添加或删除的书籍有关信息，确认后更新图书总库。

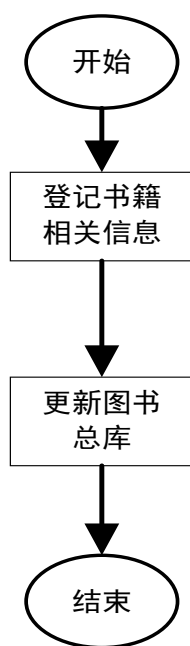


图 7 添加/删除书籍流程图

(二) 数据流程分析

本图书管理系统的数据流按照如下的流程在系统中有秩序地流动。

1. 读者登录

简介：读者通过输入账号和密码登录系统，系统验证读者登录信息表中的对应数据。

输入数据：账号（字符串）、密码（字符串）。

处理过程：系统验证读者登录信息表中的对应数据。

输出数据：登录成功（用户信息、权限）或登录失败（错误信息）。

2. 查询书籍

简介：读者可以查询书籍，系统会显示待查书籍名录。如果书籍在库，读者可以继续借阅流程。

输入数据：查询条件（书名、作者、ISBN 等）。

处理过程：系统根据条件搜索书籍。

输出数据：待查书籍名录（书籍信息列表）或无结果信息。

3. 借阅书籍

简介：读者选择借阅书籍后，系统记录借阅信息，包括个人信息和借阅时间，这些借阅信息被记录在借阅名录中。

输入数据：选择的书籍 ID、用户 ID。

处理过程：系统记录借阅信息，包括个人信息和借阅时间。

输出数据：借阅确认（借阅记录）。

4. 归还书籍

简介：读者归还书籍后，系统更新归还信息，包括个人信息和归还时间，这些信

息会在借阅名录中更新。

输入数据：归还书籍 ID、用户 ID。

处理过程：系统更新归还信息，包括个人信息和归还时间。

输出数据：归还确认（更新后的借阅记录）或逾期费用信息。

5. 发送逾期提醒

简介：系统会定期检查借阅名录，对于逾期未还的书籍，发送逾期提醒给读者。

处理过程：系统定期检查借阅名录，对于逾期未还的书籍，发送逾期提醒给读者。

输出数据：逾期提醒信息。

6. 管理员相关操作

（1）管理员登录：管理员通过输入账号和密码登录系统，系统验证管理员登录信息表中的对应数据。

（2）新书入库：管理员可以添加新书信息，系统将新书编号及相关信息记录在图书总库中。

（3）注销图书：管理员可以注销不再需要的图书，系统会更新图书总库。

下图是对上述数据流程的直观阐释：

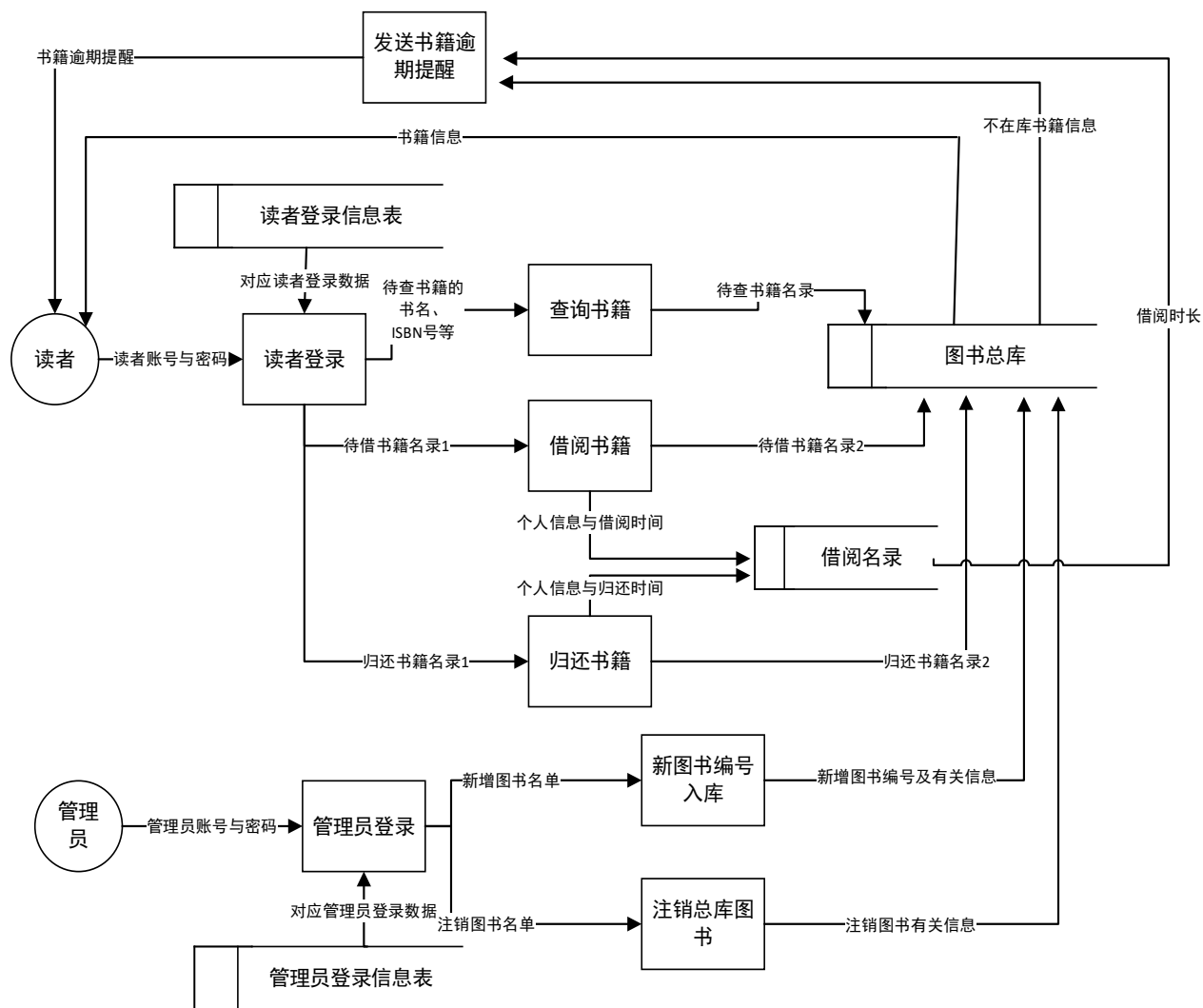


图 8 图书管理系统数据流程一览图

（四）数据字典的定义

1. 数据项的定义

数据项编号：A-01
 数据项名称：用户账号
 别名：账号
 简述：唯一标识每个用户的信息
 类型：数值型
 长度：10 位
 取值范围：0-500000000

数据项编号：A-02
 数据项名称：用户密码
 别名：密码
 简述：用户登录的凭证
 类型：字符串
 长度：8-20 个字符
 取值范围：数字+大小写英文字母+符号

数据项编号：A-03 数据项名称：图书名字 别名：书名 简述：每本书的名字 类型：字符串 长度：1-100 个字符 取值范围：中英文字符	类型：数值型 长度：10 位或 13 位数字 取值范围：前缀+组区号+出版者号+出版序号+校验位
数据项编号：A-04 数据项名称：图书编号 别名：ISBN 号 简述：唯一标识每本图书的编号	

2. 数据结构的定义

数据结构编号：D-01 数据结构名称：用户信息 简述：用户的身份信息 数据结构组成：用户账号+密码	简述：书籍的基本信息 数据结构组成：图书名字+图书编号
数据结构编号：D-02 数据结构名称：书籍信息	

3. 数据流的定义

数据流编号：E-01 数据流名称：读者/管理员账号密码 简述：读者/管理员输入账号密码 数据流来源：读者/管理员 数据流去向：图书管理系统 数据流组成：用户信息 数据流量：1000 次/小时 高峰流量：10000 次/小时	数据流去向：图书总库 数据流组成：书名+编号 数据流量：500 次/小时 高峰流量：5000 次/小时
数据流编号：E-02 数据流名称：待查书籍名录 简述：读者查询书籍 数据流来源：读者	
数据流编号：E-03 数据流名称：待借书籍名录	简述：读者借阅书籍 数据流来源：读者

数据流去向：图书总库
数据流组成：书名+编号
数据流量：500 次/小时
高峰流量：5000 次/小时

数据流去向：图书总库
数据流组成：书名+编号
数据流量：500 次/小时
高峰流量：5000 次/小时

数据流编号：E-04
数据流名称：归还书籍名录
简述：读者归还书籍
数据流来源：读者

数据流编号：E-05
数据流名称：书籍逾期提醒
简述：通知读者书籍逾期
数据流来源：图书总库，借阅名录
数据流去向：读者
数据流组成：书籍信息+读者信息
数据流量：500 次/小时
高峰流量：5000 次/小时

数据流编号：E-06
数据流名称：新增图书名单
简述：管理员新增书籍
数据流来源：管理员
数据流去向：图书总库
数据流组成：书名+编号
数据流量：500 次/小时
高峰流量：5000 次/小时

数据流编号：E-07
数据流名称：注销图书名单
简述：管理员书籍
数据流来源：管理员
数据流去向：图书总库
数据流组成：书名+编号
数据流量：500 次/小时
高峰流量：5000 次/小时

4. 处理逻辑的定义

处理逻辑编号：F-01
处理逻辑名称：读者登录
简述：验证读者身份并授权访问
输入的数据流：读者账号密码
处理：根据数据流读者信息，检索读者信息，授权登录
输出的数据流：登录成功/登录失败
处理频率：每个读者每次登录处理一次

处理逻辑编号：F-03
处理逻辑名称：查询书籍
简述：读者查询书籍
输入的数据流：书籍信息
处理：根据书籍信息，检索图书总库，输出书籍信息
输出的数据流：待查书籍名录
处理频率：每个读者每次查询处理一次

处理逻辑编号：F-04
处理逻辑名称：借阅书籍
简述：读者借阅书籍
输入的数据流：待借书籍名录 1
处理：根据书籍信息，检索借阅名录，输出待借书籍名录
输出的数据流：待借书籍名录 2
处理频率：每个读者每次借阅处理一次

处理逻辑编号：F-05
处理逻辑名称：归还书籍
简述：读者归还书籍
输入的数据流：归还书籍名录 1
处理：根据书籍信息，检索借阅名录，输出归还书籍名录 2
输出的数据流：归还书籍名录 2
处理频率：每个读者每次归还处理一次

处理逻辑编号：F-07
处理逻辑名称：新图书入库
简述：管理员入库新图书
输入的数据流：新增图书名单
处理：根据新增图书名单，检索信息，输出新增图书编号等信息
输出的数据流：新增图书编号
处理频率：每个管理员每次新增书籍处理一次

处理逻辑编号：F-02
处理逻辑名称：管理员登录
简述：验证管理员身份并授权访问
输入的数据流：管理员账号密码
处理：根据数据流管理员信息，检索管理员信息，授权登录
输出的数据流：登录成功/登录失败
处理频率：每个管理员每次登录处理一次

处理逻辑编号：F-06
处理逻辑名称：注销总库图书
简述：管理员注销总库图书
输入的数据流：注销图书名单
处理：根据注销名单，检索信息，输出注销图书有关信息
输出的数据流：注销图书信息
处理频率：每个管理员每次注销图书处理一次

5. 数据存储的定义

数据存储编号：G-01

数据存储名称：读者登录信息表

简述：存储读者的登录信息
数据存储组成：读者账号+密码
关键字：读者登录
相关联的处理：读者登录

数据存储组成：管理员账号+密码
关键字：管理员登录
相关联的处理：管理员登录

数据存储编号：G-02
数据存储名称：管理员登录信息表
简述：存储管理员的登录信息

数据存储编号：G-03
数据存储名称：借阅名录
简述：存储借阅信息
数据存储组成：读者信息+借阅时间/归还时间
关键字：读者借阅
相关联的处理：借阅书籍，归还书籍

数据存储编号：G-04
数据存储名称：图书总库
简述：存储所有书籍的信息
数据存储组成：书籍信息
关键字：书籍信息
相关联的处理：查询书籍，借阅书籍，归还书籍，新图书入库，注销图书

6. 外部实体的定义

外部实体编号：H-01
外部实体名称：读者
简述：查询、借阅、归还书籍
输入的数据流：书籍信息，待借书籍名录 1，归还书籍名录 1
输出的数据流：书籍信息，书籍逾期提醒，待借书籍名录 2，归还书籍名录 2

简述：入库新图书，注销图书
输入的数据流：新图书名单，注销图书名单
输出的数据流：新增图书有关信息，注销图书有关信息

外部实体编号：H-02
外部实体名称：管理员

二、系统设计

（一）MVC 架构

1.模型（Model）

描述：模型代表应用程序的数据结构和业务逻辑。本系统中，模型负责管理图书信息、用户信息、借阅记录等数据，并提供数据操作的方法。

主要实体：

用户模型：包含用户信息，如用户账号、密码、权限等。

图书模型：包含图书信息，如书名、作者、ISBN、分类、状态（在库、借出）等。

借阅记录模型：包含借阅信息，如记录 ID、用户 ID、图书 ID、借阅日期、归还日期、逾期状态等。

2.视图（View）

描述：视图负责展示数据（模型）到用户界面，并收集用户输入。在图书管理系统中，视图包括所有用户界面元素，如登录页面、读者端/管理端首页、图书列表页面、借阅/归还页面等等。

主要界面：

登录视图：显示登录表单，接收用户输入的账号和密码。

读者端视图：包括找书、借书、还书等板块，以及借阅记录的展示。

管理员端视图：包括图书管理（增添新书、删除旧书）等界面。

3.控制器（Controller）

描述：控制器负责接收用户输入，调用模型和视图去完成用户请求。它作为模型和视图之间的中介，从模型获取数据并将其传递给视图展示。

主要功能：

用户认证：处理用户登录和登出的请求。

图书管理：处理图书查询、借阅和归还的请求。

数据更新：处理图书信息的添加、编辑和删除请求。

通知管理：处理发送逾期提醒等请求。

（二）用户流程图

本图书管理系统前端设计的用户流程基本如下：

用户在登录界面根据不同账号进入读者界面或管理界面：读者界面可以进行找书、借书和还书，其中借书和还书会更新借阅记录。管理界面可以进行新增书籍和删除书籍，这两项操作都将更新书库。

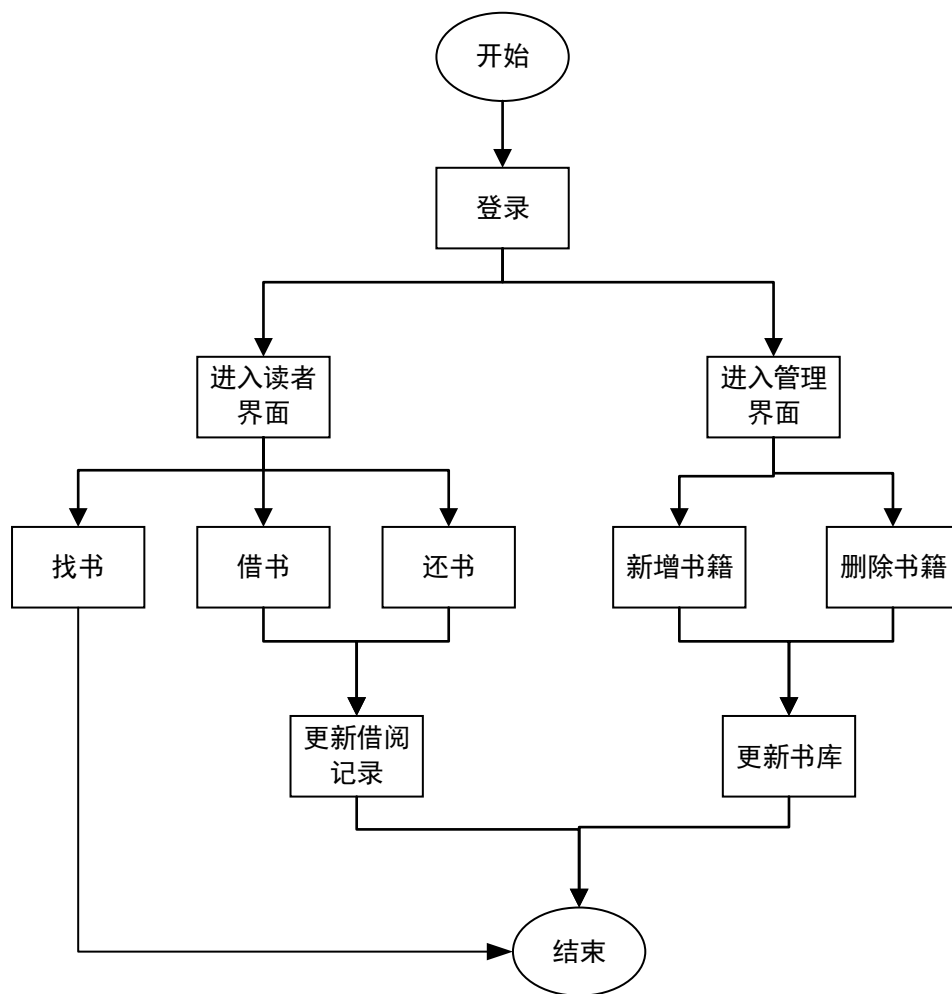


图9 用户流程图

(三) 原型设计与跳转逻辑

1. 登录界面

用户在该界面输入自己的图书管理系统账号与密码，从而进入不同的界面。



图 10 登录界面

2. 读者端

此为读者端首页，分为找书、借书、还书三个主要板块，点击分别可以跳转对应界面；此外点击顶部“我的借阅情况”可以跳转到该账号名下的借阅记录。



图 11 读者端首页

此为读者找书界面，左侧索引栏根据类型分成了四个一级目录，一级目录下各有若干二三级目录。点击可在右侧显示对应种类的书籍，当符合目录类别的书籍数目过多时，可滑动右侧界面查看。此外右侧上方居中处有搜索栏，点击可输入书名、作者名等实现精确搜索。右上角有退出按键，点击可跳转到读者端首页。

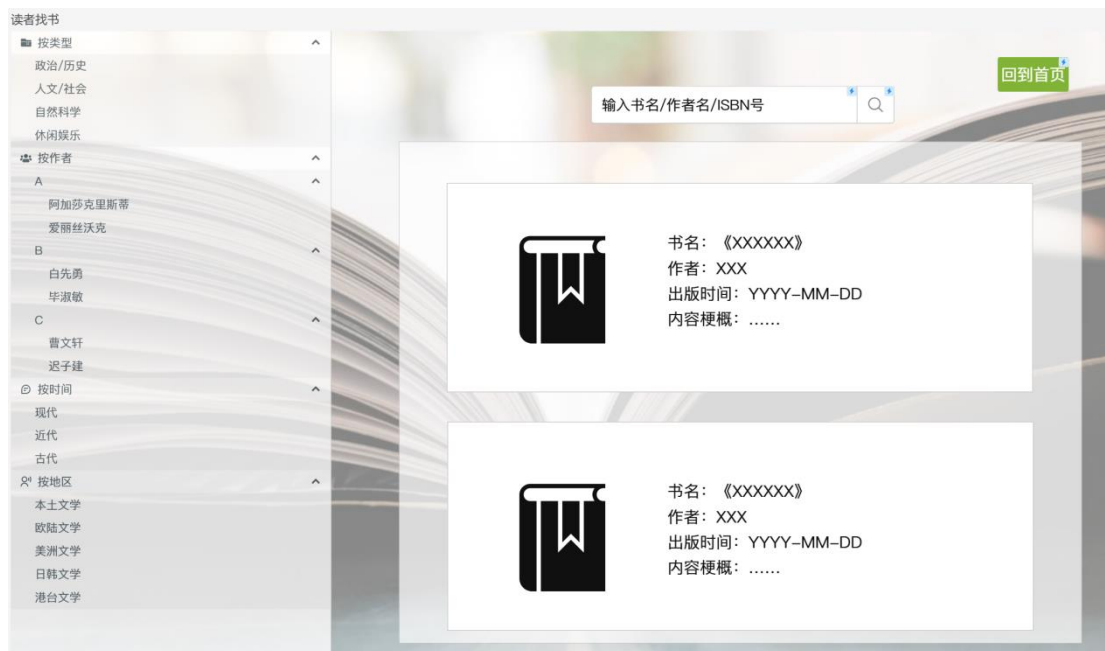


图 12 读者找书界面

此为借书界面，读者在此点击“书名”后搜索框可跳转到找书界面搜索目标书籍，选定后目标书籍后跳转回本界面，再自主填写自己的姓名、电话等个人信息。借出时间系统自动填写当天时间，归还时间系统自动关联以当天为起点的十四天后。

页面底部有两个按键，单击“确认退出”则跳转到借阅记录界面，同时借阅记录更新新借阅书籍信息。单击“不借了，退出”则直接回到读者端首页。

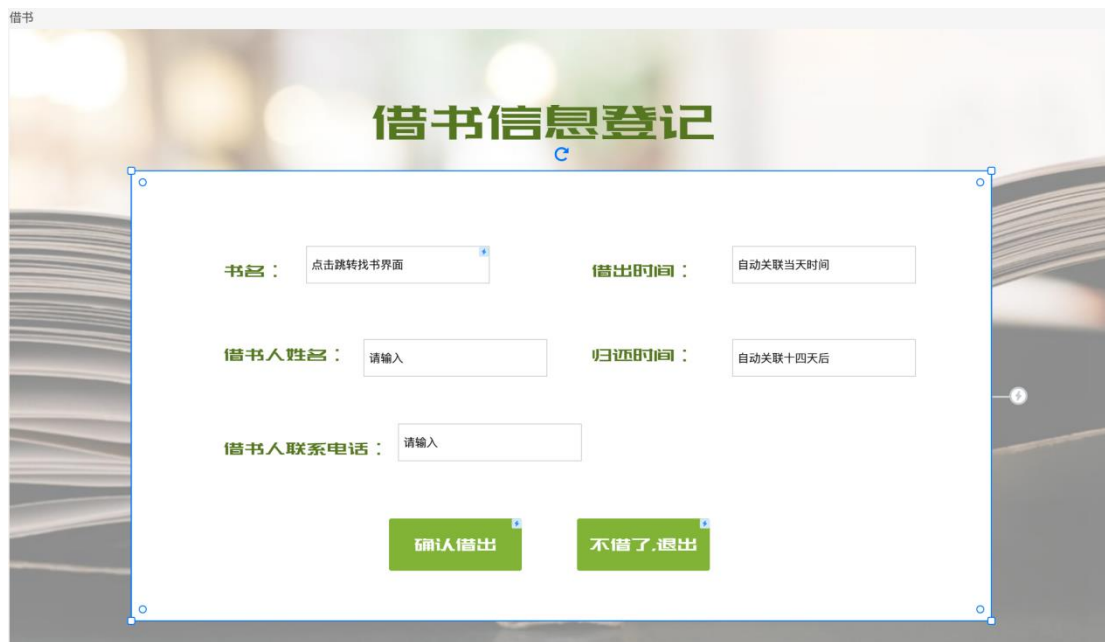


图 13 借书界面

此为还书界面，读者点击“书名”后输入框，跳转借阅记录界面，选定已借阅的书后再跳转回本界面，同时借书人姓名、联系电话自动关联借阅记录里的信息。同时系统自动关联归还时间，并且计算出逾期的费用。

页面底部有两个按键，单击“确认归还”则跳转到借阅记录界面，同时借阅记

录删除已退还书籍信息。单击“不还了，退出”则直接回到读者端首页。

还书

还书信息登记

书名：	<input type="text" value="点击跳转“我的借阅”界面"/>	归还时间：	<input type="text" value="自动关联当天时间"/>
借书人姓名：	<input type="text" value="自动关联借阅信息"/>	逾期费用：	<input type="text" value="自动计算，未逾期则显示为0"/>
借书人联系电话：	<input type="text" value="自动关联借阅信息"/>		

图 14 还书界面

此为借阅记录界面，读者在此可查看自己的借阅记录，点击顶部“首页”即可回到首页。

我的借阅情况

首页 我的借阅情况 个人信息编辑 关于我们

书名：《XXXXXX》
作者：XXX
出版时间：YYYY-MM-DD
内容梗概：.....

书名：《XXXXXX》
作者：XXX
出版时间：YYYY-MM-DD
内容梗概：.....

图 15 借阅记录界面

3. 图书管理员端

此为管理端首页，单击找书、增添新书、删除旧书，依次可以进入相应界面。



图 16 管理端首页

此为管理员找书界面，左侧索引栏根据类型分成了四个一级目录，一级目录下各有若干二三级目录。点击可在右侧显示对应种类的书籍，当符合目录类别的书籍数目过多时，可滑动右侧界面查看。此外右侧上方居中处有搜索栏，点击可输入书名、作者名等实现精确搜索。右上角有退出按键，点击可跳转到管理端首页。



图 17 管理员找书界面

此为登记新书界面，管理员依次填入书籍相关信息，“登记人”一栏会由系统自动关联该管理员账号的个人信息。

登记完毕后单击“确认登记”，跳转管理端首页。

登记新书

登记新书

书名：

ISBN号：

作者：

登记人：

标题：

自动关联管理员账号个人信息

确认登记

图 18 登记新书界面

此为删除旧书界面，单击“书名”后搜索栏跳转管理员找书界面，选定书籍后跳转回该页面。ISBN 号将由系统结合书名自动关联，“登记人”一栏会由系统自动关联该管理员账号的个人信息。

登记完毕后单击“确认注销”，跳转管理端首页。

注销旧书

注销旧书

书名：

ISBN号：

登记人：

点击跳转书库

自动关联

自动关联管理员账号个人信息

确认注销

图 19 删除图书界面

（四）后端设计

1. 用户模型

用户模型是后端数据库中的一个核心表，用于存储用户信息。这个模型包含

以下字段：

用户 ID：唯一标识每个用户。

用户名：用户的唯一标识符，用于登录。

密码：用户的登录密码，必须以加密形式存储，以保证安全性。

权限等级：用户的角色或权限等级，例如普通读者、管理员等。

邮箱：用户的联系邮箱，可用于发送通知和密码重置。

电话号码：用户的联系电话，可用于紧急联系和验证。

创建时间：用户账户创建的时间戳。

最后登录时间：用户最后一次成功登录的时间戳。

2. 认证服务

认证服务是处理用户登录请求的后端逻辑。它包括以下几个步骤：

接收登录请求：用户提交用户名和密码到后端。

验证用户信息：后端查询用户模型，检查用户名是否存在，并且密码是否匹配（需要对密码进行解密或使用哈希比较）。

生成认证令牌：如果用户验证成功，后端会生成一个认证令牌，如 JSON Web Token (JWT)。这个令牌包含了用户的身份信息和权限等级，并且是签名的，以确保其不被篡改。

返回认证令牌：将生成的 JWT 返回给用户，用户在后续的请求中需要携带这个令牌以验证身份。

3. 权限控制

权限控制是确保用户只能访问他们被授权访问的资源的机制。在用户登录后，每次请求资源时进行检查。

角色定义：定义不同的角色和对应的权限，例如管理员可以添加、编辑、删除图书，而普通用户只能借阅和归还图书。

权限检查：在用户请求资源时，后端会检查用户的权限等级是否允许该操作。

中间件/拦截器：在后端框架中使用中间件或拦截器来处理权限检查，这样可以在请求到达具体业务逻辑之前就拦截非法请求。

动态权限：在某些情况下，权限可能会动态变化，系统需要能够处理这种变化，及时更新用户的权限状态。

4. 安全性考虑

密码存储：密码应该使用强哈希算法（如 **bcrypt**）进行加密存储，绝不应该以明文形式存储。

令牌管理：JWT 应该有一个合理的过期时间，并且在用户登出时应该能够被撤销。

防止攻击：实现防止常见的安全攻击，如 SQL 注入、跨站请求伪造（CSRF）、跨站脚本攻击（XSS）等。

日志记录：记录所有认证尝试和权限变更，以便在发生安全事件时进行审计。

（五）接口设计

1. POST /login: 用户登录

请求格式：

```
json
{
  "username": "string",
  "password": "string"
}
```

响应格式：

```
json
{
  "status": "success",
  "token": "string",
  "message": "User logged in successfully."
}
```

交互：

前端视图：显示登录表单，收集用户名和密码。

后端视图：验证用户名和密码，生成令牌（Token），返回登录状态和令牌。

2. GET /user/profile: 获取用户信息

请求格式：

json

```
{  
  "token": "string"  
}
```

响应格式:

json

```
{  
  "status": "success",  
  "user": {  
    "id": "integer",  
    "username": "string",  
    "email": "string",  
    "joined": "date"  
  },  
  "message": "User profile retrieved successfully."  
}
```

交互:

前端视图: 请求用户信息, 显示用户个人资料。

后端视图: 根据令牌验证用户身份, 返回用户信息。

3. 图书接口

(1) GET /books: 查询图书

请求格式:

json

```
{  
  "query": "string",  
  "page": "integer",  
  "limit": "integer"  
}
```

响应格式:


```
json
{
  "status": "success",
  "books": [
    {
      "id": "integer",
      "title": "string",
      "author": "string",
      "isbn": "string",
      "status": "string"
    }
  ],
  "total": "integer",
  "message": "Books retrieved successfully."
}
```

交互：

前端视图：显示图书搜索结果。

后端视图：根据查询条件检索图书，分页返回结果。

(2) POST /books：添加图书

请求格式：

```
json
{
  "title": "string",
  "author": "string",
  "isbn": "string",
  "status": "string"
}
```

响应格式：

```
json
{
```

```
"status": "success",
"book": {
  "id": "integer",
  "title": "string",
  "author": "string",
  "isbn": "string",
  "status": "string"
},
"message": "Book added successfully."
}
```

交互：

前端视图：显示添加图书表单，收集图书信息。

后端视图：将新图书信息插入数据库。

(3) PUT /books/{id}：更新图书信息

请求格式：

json

```
{
  "title": "string",
  "author": "string",
  "isbn": "string",
  "status": "string"
}
```

响应格式：

json

```
{
  "status": "success",
  "book": {
    "id": "integer",
    "title": "string",
    "author": "string",
```

```
    "isbn": "string",
    "status": "string"
  },
  "message": "Book updated successfully."
}
```

交互：

前端视图：显示图书编辑表单，收集更新信息。

后端视图：根据 ID 更新图书信息。

（4）DELETE /books/{id}：删除图书

请求格式：

```
json
{
  "id": "integer"
}
```

响应格式：

```
json
{
  "status": "success",
  "message": "Book deleted successfully."
}
```

交互：

前端视图：提供删除图书的选项。

后端视图：根据 ID 从数据库中删除图书。

（4）借阅接口

POST /borrow：借阅图书

请求格式：

```
json
{
  "user_id": "integer",
  "book_id": "integer"
}
```

```
}
```

响应格式:

json

```
{
  "status": "success",
  "borrowing": {
    "id": "integer",
    "user_id": "integer",
    "book_id": "integer",
    "date": "date"
  },
  "message": "Book borrowed successfully."
}
```

交互:

前端视图: 显示借阅图书的选项。

后端视图: 创建借阅记录, 更新图书状态。

(5) POST /return: 归还图书

请求格式:

json

```
{
  "user_id": "integer",
  "book_id": "integer"
}
```

响应格式:

json

```
{
  "status": "success",
  "message": "Book returned successfully."
}
```

交互：

前端视图：显示归还图书的选项。

后端视图：更新借阅记录，恢复图书状态。

（6）搜索接口

GET /search: 根据条件搜索图书

请求格式：

json

```
{  
  "query": "string",  
  "page": "integer",  
  "limit": "integer"  
}
```

响应格式：

json

```
{  
  "status": "success",  
  "books": [  
    {  
      "id": "integer",  
      "title": "string",  
      "author": "string",  
      "isbn": "string",  
      "status": "string"  
    }  
  ],  
  "total": "integer",  
  "message": "Search results retrieved successfully."  
}
```

交互：

前端视图：显示搜索框，收集搜索关键词。

后端视图：根据搜索条件检索图书，分页返回结果。

（7）通知接口

GET /notifications: 获取通知信息

请求格式：

```
json
{
  "user_id": "integer"
}
```

响应格式：

```
json
{
  "status": "success",
  "notifications": [
    {
      "id": "integer",
      "message": "string",
      "date": "date"
    }
  ],
  "message": "Notifications retrieved successfully."
}
```

交互：

前端视图：显示通知列表。

后端视图：根据用户 ID 检索通知信息。

（六）数据存储

1.数据库选择

为了满足本系统的需求，我们选择 MySQL 数据库作为数据存储解决方案。MySQL 是一个流行的关系型数据库管理系统，以其高性能、可靠性和易用性而闻名。它支持多种操作系统，并且提供了强大的数据加密和备份功能，确保数据

的安全性和完整性。

进一步详细到数据库的具体配置和优化策略：

(1) 存储引擎选择：

推荐使用 InnoDB 存储引擎，它支持事务处理、行级锁定，并提供更好的并发性能和崩溃恢复能力。

(2) 字符集和排序规则：

设置服务器端使用的字符集为 utf8mb4，这是 MySQL 中支持最多 Unicode 字符的字符集。

设置排序规则为 utf8mb4_unicode_ci，以支持大小写不敏感的比较。

(3) 连接设置：

max_connections：设置最大连接数，建议设置为上限连接数的 80% 左右。

table_open_cache：设置表的缓存数量，与 max_connections 有关，建议至少为并行连接数乘以查询中联接的表的最大数量。

thread_cache_size：线程缓存数量，用于缓存线程以复用。

(4) 缓冲区大小：

sort_buffer_size：查询排序时缓冲区大小，可增大此值为 16M，以提高排序效率。

(5) 日志设置：

log-output：设置 MySQL 日志的输出方式。

general-log：启用常规查询日志。

slow-query-log：启用慢查询日志，以识别和优化低效查询。

(6) 错误日志和安全性：

log-error：设置错误日志文件路径。

log_bin_trust_function_creators：控制是否可以信任存储函数创建者。

(7) 性能优化策略

A. 索引优化：

确保查询至少达到 range 级别，最好能达到 ref。避免全表扫描（ALL），这通常意味着没有使用索引。

遵守最左前缀法则，即查询从索引的最左前列开始并且不跳过索引中的列。不在索引列上做任何操作（计算、函数、类型转换），以免导致索引失效而转向全表扫描。

B. 查询优化：

使用 EXPLAIN 关键字分析查询，以确定是否有效使用了索引。

优化查询语句，减少不必要的联接和子查询。

C. 存储优化：

根据数据访问模式合理设计表结构，避免过多的冗余数据。

D. 数据库结构优化：

定期检查和优化数据库结构，如归档旧数据，以提高性能。

E. 硬件和部署优化：

考虑使用主从复制和负载均衡技术，如 Keepalived，以提高数据库的可用性和扩展性。

F. 安全配置：

配置审计以监控和记录数据库活动，增强数据库的安全性。

G. 数据备份：

定期进行本地和异地备份，确保数据的安全性和可恢复性。

2. 数据库设计

(1) 用户表 (Users)

user_id (INT, PRIMARY KEY, AUTO_INCREMENT): 唯一标识用户的 ID。

username (VARCHAR): 用户名，用于登录系统。

password (VARCHAR): 加密存储的用户密码。

email (VARCHAR): 用户的电子邮件地址。

create_time (DATETIME): 用户账户创建时间。

update_time (DATETIME): 用户信息最后更新时间。

permission_level (INT): 用户的权限等级，用于访问控制。

(2) 图书表 (Books)

book_id (INT, PRIMARY KEY, AUTO_INCREMENT): 唯一标识图书的 ID。

title (VARCHAR): 书名。

author (VARCHAR): 作者名。

ISBN (VARCHAR): 国际标准书号。

publisher (VARCHAR): 出版社。

category (VARCHAR): 图书分类。

status (ENUM): 图书状态（在库、借出、预定）。

(3) 借阅记录表 (BorrowingRecords)

record_id (INT, PRIMARY KEY, AUTO_INCREMENT): 唯一标识借阅记录的 ID。

user_id (INT, FOREIGN KEY REFERENCES Users(user_id)): 借阅用户的 ID。

book_id (INT, FOREIGN KEY REFERENCES Books(book_id)): 借阅图书的 ID。

borrow_date (DATETIME): 借阅日期。

return_date (DATETIME): 归还日期。

overdue_status (BOOLEAN): 是否逾期。

3. 数据库访问层

(1) UserDAO: 用户数据访问对象

UserDAO 是专门负责与用户表交互的类，它封装了所有与用户相关的数据库操作。以下是 UserDAO 包含的关键功能：

添加新用户：将新用户的信息插入到用户表中。

验证用户：根据提供的用户名和密码验证用户身份。

更新用户信息：允许已认证的用户更新其个人资料信息，如电子邮件地址、联系方式等。

删除用户：在用户请求注销账户时，从数据库中删除用户记录。

查询用户：根据用户 ID 或其他标识符检索用户信息。

重置密码：为忘记密码的用户重置密码。

用户权限管理：管理用户的角色和权限，确保用户只能访问他们被授权的资源。

登录日志记录：记录用户的登录历史，用于安全审计和监控。

(2) BookDAO: 图书数据访问对象

BookDAO 管理与图书表相关的所有数据库操作，包括：

添加新书：将新图书的详细信息添加到图书表中。

查询图书：根据书名、作者、ISBN 等条件查询图书。

更新图书状态：更新图书的借阅状态，如在借出、可借阅、维修中等。

删除图书：从数据库中移除不再可用的图书记录。

图书库存管理：增加或减少图书的库存数量。

图书分类管理：管理图书的分类信息，便于用户检索。

图书推荐：根据用户的借阅历史推荐图书。

(3) BorrowingRecordDAO：借阅记录数据访问对象

BorrowingRecordDAO 负责处理所有与借阅记录相关的数据库操作：

创建借阅记录：在用户借阅图书时，创建一个新的借阅记录。

查询借阅历史：允许用户和管理员查询过去的借阅记录。

更新借阅状态：更新借阅记录的状态，如续借、逾期、归还等。

逾期管理：标记逾期未还的图书，并通知用户。

借阅统计：生成借阅统计报告，如最常借阅的图书、最活跃的借阅者等。

4. E-R 图

在图书管理系统的数据库设计中，我们可以定义以下实体和关系，从而完成 E-R 图的刻画。

- **图书实体**：包含图书的基本信息（如书名、作者、出版社等）和附加信息（如图书封面、图书简介等）。同时，图书实体还可以与借阅记录实体建立关系，表示该图书被哪些用户借阅过；
- **用户实体**：包含用户的基本信息（如姓名、性别、年龄等）和附加信息（如联系方式、借阅证号等）。同时，用户实体还可以与借阅记录实体建立关系，表示该用户借阅过哪些图书；
- **管理员实体**：包含管理员的账号信息，赋予更多的管理权限，例如接收采购目录、查询所有图书的统计信息（动态信息、损毁和保存状态）、添加新书目或删除旧书目等操作，用户通过图书的借阅向管理员发送信息，管理员可以选择推荐书目；
- **书籍采购员**：专门负责书籍采购工作，新书目采购以及原有书目的补充；
- **借阅记录关系**：包含借阅的基本信息（如借阅时间）和附加信息（如借阅的图书名称、借阅的图书数量等）。同时，借阅记录关系还可以与图书实体和用户实体建立关系，表示该借阅记录对应的图书和用户信息；
- **归还记录关系**：包含归还的基本信息（如归还时间）。同时，归还记录关系还可以与图书实体和用户实体建立关系，表示该归还记录对应的图书和用户信息；
- **管理关系**：包含图书更改信息，如新增书目信息和注销书目信息，通过管理关系将管理员实体与图书实体两者建立联系，使管理员通过管理关系来使用自己的管理图书权限；
- **采购关系**：记录采购书籍目录，通过采购关系将采购书籍信息从采购员实体传达到管理员实体。

下面是有关数据存储分析的 E-R 图

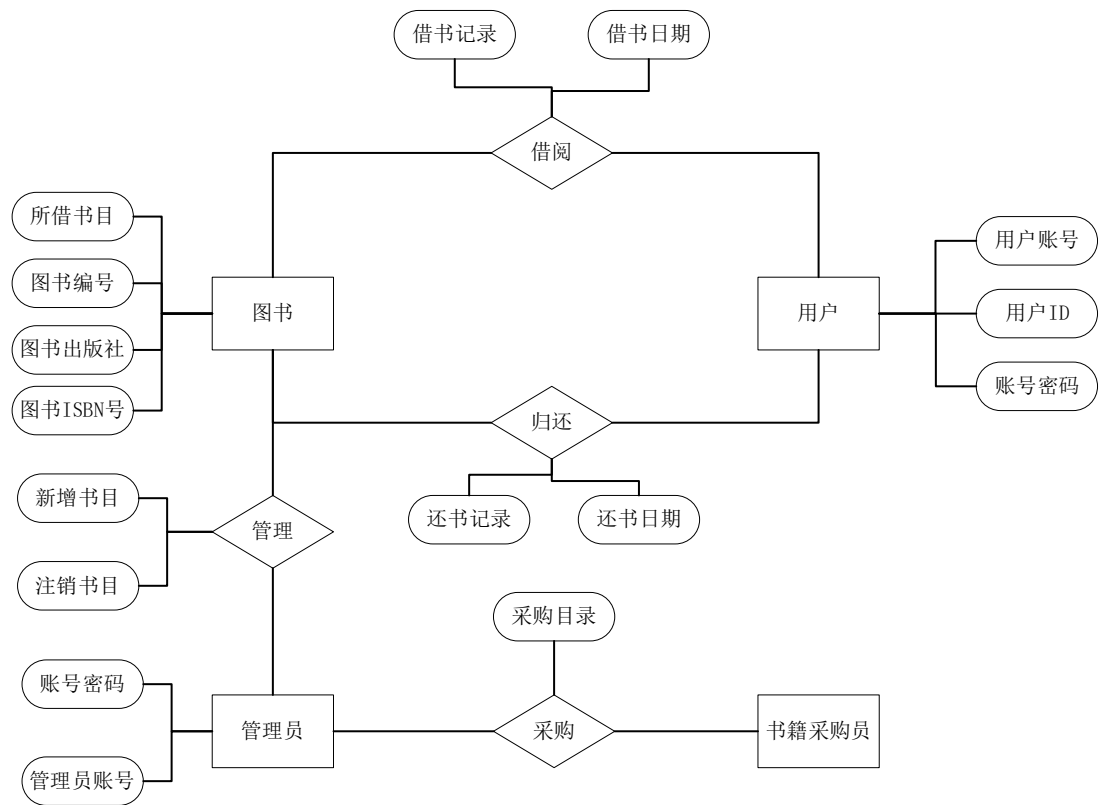


图 20 E-R 图