

《专业方向综合实践》报告

|  |  |
| --- | --- |
| 题目： | 图书管理系统 |
| 专业： | 软件工程 |
| 班级： | 软件2003 |
| 日期： | 2023.11.13 |

|  |  |
| --- | --- |
| 成员 | 任务分工 |
| 曹伟（组长） | 前后端开发框架搭建、功能开发 |
| 王旭 | 数据库搭建、功能开发 |
| 赵骏杰 | 系统测试、功能开发 |
| 沈李 | 需求分析与概要设计、功能开发 |
| 兰富龙 | 软件架构设计、功能开发 |

**目录**

[一、方案选择 3](#_Toc866407384)

[1. 课题简介 3](#_Toc1769776228)

[2. 解决方案 3](#_Toc1980553046)

[二、开发工具与分工协作 4](#_Toc1158515622)

[1. 开发工具 4](#_Toc2085315252)

[2. 团队分工 4](#_Toc960321324)

[3. 迭代规划 4](#_Toc1780885263)

[4. 编码规范 5](#_Toc1859027002)

[三、需求分析 6](#_Toc927242411)

[1. 功能分析 6](#_Toc2021859045)

[2. 用例建模 7](#_Toc1751222834)

[3. 概念模型分析 8](#_Toc1538788903)

[四、概要设计 9](#_Toc279531900)

[五、详细设计 9](#_Toc1545907311)

[六、系统实现（编码） 9](#_Toc1807014571)

[1. 系统总体架构实现 9](#_Toc780158923)

[2. 登录注册 10](#_Toc1743353926)

[3. 图书录入 11](#_Toc282554614)

[4. 书籍检索 12](#_Toc809053981)

[5. 书籍借阅与预定 13](#_Toc2051289510)

[6. 书籍状态管理 15](#_Toc320325632)

[7. 用户权限管理 16](#_Toc2118877642)

[七、 系统测试 18](#_Toc255210893)

[1. 登录与注册 18](#_Toc804635592)

[2. 图书录入与修改 19](#_Toc805869585)

[3. 图书查询与检索 20](#_Toc70753466)

[4. 图书借阅与预定 20](#_Toc1595046271)

[5. 图书状态管理 20](#_Toc904311196)

[6.用户权限管理 21](#_Toc1016501353)

[八、总结与改进 21](#_Toc1105827986)

[九、参考文献 21](#_Toc1327479564)

[格式要求说明： 22](#_Toc741423465)

# 一、方案选择

## 1. 课题简介

系统开发的总目标是实现内部图书借阅管理的系统化、规范化和自动化。

主要功能模块：

（1）对图书进行录入和修改功能，也就是将图书的基本信息（如：书的编号、书名、作者、价格等）预先存入数据库中，供以后检索。

（2）借阅人进行注册和登入功能，包括记录借阅人的姓名、编号、班级、年龄、性别、地址、电话等信息。

（3）提供方便的查询方法。如：以书名、作者、出版社、出版时间（确切的时间、时间段、某一时间之前、某一时间之后）等信息进行图书检索，并能反映出图书的借阅情况；以借阅人编号对借阅人信息进行检索；以出版社名称查询出版社联系方式信息。

（4）图书借阅功能。

（5）提供对图书预订借阅的功能。当前某图书全部借出，使用预订借阅功能，当此图书归还时，预订借阅者可优先借阅，并会收到可借阅通知。

（6）提供旧书销毁功能，对于淘汰、损坏、丢失的书目可及时对数据库进行修改。

（7）能够对使用该管理系统的用户进行管理，按照不同的工作职能提供不同的功能授权。

该课题不仅满足了图书馆管理的基本需求，还通过引入自动化和信息技术手段，提高了管理效率、服务水平，为图书馆提供了现代化的管理工具，进而为用户提供更好的服务体验。

## **2. 解决方案**

为了实现图书管理系统，原定方案中，在前端方案选择中存在分支选项：

使用Thymeleaf集成于后端进行页面开发或使用React进行前端开发，前后端分离开发。

存在两套方案：

1. Mysql+SpringBoot+Mybatis+Thymeleaf
2. Mysql+SpringBoot+Mybatis+Redis+React

本课题选用方案2进行开发。由于前后端分离可以解决人力不足,分工不明确和前后端耦合度高等问题，且具有加快项目的开发速度,提高用户访问Web程序的响应速度的优势。[1]

Mysql完成数据库搭建，Springboot+Mybatis+Redis实现后端，React进行页面开发，前后端分离，前端自主持有路由，后端仅返回前端请求的数据。总体系统结构使用MVC结构进行开发。

# 二、开发工具与分工协作

## **1. 开发工具**

1.1 Idea 用于开发Java后端;

1.2 VSCode 用于开发Web前端;

1.3 Postman 用于测试后端API接口并协作分享API文档;

1.4 Github 作为代码仓库进行团队协作开发；

1.5 Github Desktop,SourceTree 团队成员个人使用的git客户端，用于本地分支管理与版本控制;

1.6 Freedgo 进行软件架构设计;

## **2. 团队分工**

功能模块的团队划分如表1所示：

表 1 团队分工表

|  |  |
| --- | --- |
| 负责人 | 功能模块 |
| 曹伟 | 图书录入修改；借阅人登录注册； |
| 王旭 | 书籍查询与检索 |
| 赵骏杰 | 书籍借阅与预定 |
| 沈李 | 书籍状态管理 |
| 兰富龙 | 用户权限管理 |

## 迭代规划

本课题采用迭代模型进行开发，迭代周期为一周，共两个迭代。

迭代1（2023.10.30 --- 2023.11.3）：

需求用例文档产出：沈李

软件总体架构设计：兰富龙

数据库设计搭建：王旭

前后端开发架构搭建：曹伟

代码仓库配置：赵骏杰

登录注册模块：曹伟

迭代2（2023.11.6 --- 2023.11.10）：

图书录入修改模块：曹伟

书籍查询与检索模块：王旭

书籍借阅与预定：赵骏杰

书籍状态管理：沈李

用户权限管理：兰富龙

## 编码规范

数据库设计规范：

数据库表名和字段名都使用小写字母，每个单词之间以下划线"\_" 分割，表名使用 "前缀\_表名"，其中前缀代表这个表代表的逻辑含义或者归属。比如loan前缀，代表贷款相关的表，partner代表合作伙伴相关的表等等。

数据库表的主键字段名使用 id，类型是 int 自增 或者 varchar(32) uuid 类型，如果这个表使用的外键做主键，则使用 外部{table\_name}\_id 作为主键，字段类型和目标表字段类型一致。

如果涉及到数据的字符串类型，一般字节长度为8,16,32,64,128,256 格式，一般为2的N次幂形式。对于日期的字符串格式，必须为varchar(10)，格式为'yyyy-mm-dd'格式，例如"2016-12-12"

对于状态字段，目前系统都是int表示，用0,1,2,3 等方式，这种方式非常不直观。后续的status或者其他等相关的表示状态，类型的字段默认是用varchar(16)或者varchar(32)类型，用英文表示，可读性会非常好。

接口API规范：

查询类api

method = GET

根据查询条件获取所有设备列表

url链接:{host}/deivce/server/all

请求参数:key1=value1 key2=value2

返回数据:json

成功情况:

{

'status':'success',

'code':'000000'

'message':'请求处理成功'

'content':[{'name:'='',...},{'name='',...}],

}

status 代表状态码，成功success，失败error

message代表处理结果描述信息结果

content 代表处理结果列表

meta 代表请求查询请求列表的meta信息

code 代表错误码,对于sucess状态,默认是000000,对于fail状态，默认是999999,可以修改的

# 三、需求分析

## **1. 功能分析**

对课程任务进行分析，得到项目所要完成的功能如下：

1.1 借阅人注册和登录功能。借阅人输入用户名、密码和邮箱进行注册，输入用户名和密码进行登录。

1.2 图书录入和修改功能。管理员输入图书名称、作者、出版社、价格、库存和出版时间对书籍进行录入。管理员可以修改图书的基本信息。

1.3 图书查询和检索功能。借阅人可以根据书名、作者、出版社、出版社时间以及状态（是否被借阅）进行图书的检索，可以根据出版社名查询相应的联系方式信息。管理员可以通过借阅人编号对借阅人信息进行检索。

1.4 图书借阅和预定功能。借阅人可以借阅暂未被其他人借阅的图书，如果当前图书全部处于被借阅的状态，借阅人可以对图书进行预定，当此图书归还时，预定借阅证可以即时收到邮件提醒。

1.5 旧书销毁功能。对于淘汰、损毁和丢失的图书，可以修改其状态为不可借阅状态，并可对其详情进行备注。

1.6 用户权限管理功能。用户的权限分为3个级别。1级权限为借阅人，仅能借阅、预定和归还图书。2级为图书管理员，在1级基础上，可以对图书进行录入、修改和销毁操作。3级为最高管理员，在2级基础上，可以修改所有用户账号信息和权限。

由上述的功能，可以得到相应的软件层次图，如图3.1所示。

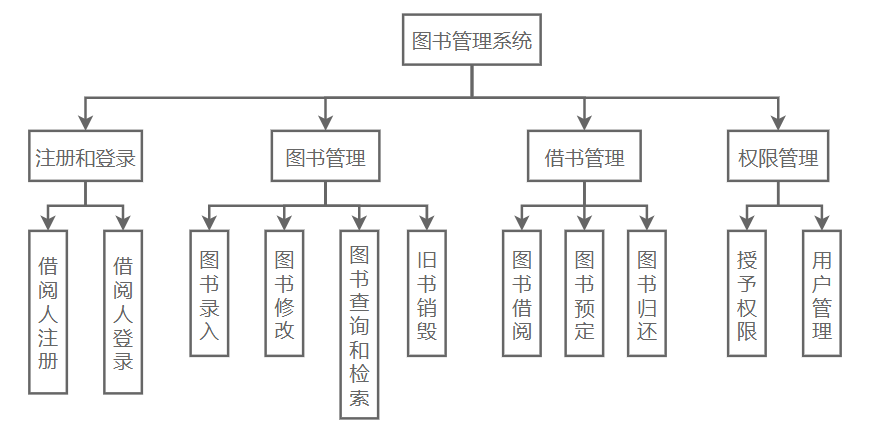


图 3.1 软件层次图

整个图书管理系统由四个模块构成，分别为注册和登录、图书管理、借书管理和选项管理模块。在注册和登录模块，借阅人可以进行注册和登录操作。在图书管理模块，管理员可以对图书进行录入、修改、检索和旧书销毁的操作。在借书管理模块，借阅人可以借阅、预定和归还图书。在选项管理模块，最高管理员可以对所有用户的权限进行授予，并且管理所有的用户。

## **2. 用例建模**

在本系统中一共有三个参与者，分别为借阅人、图书管理员和最高管理员。

借阅人拥有注册用户、登录用户、图书借阅、图书预定和图书归还用例。其中图书借阅和图书预定包含图书查询用例，“我”的借阅查询是图书借阅的扩展用例，编辑个人信息是注册账户的扩展用例。

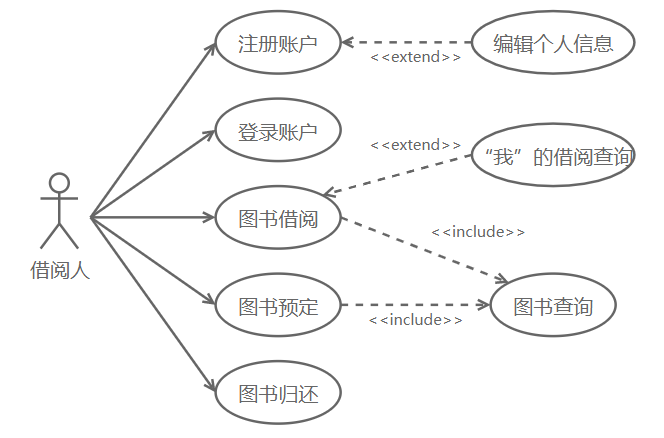


图 3.2 借阅人的用例图

图书管理员拥有借阅人的所有用例，除此之外，图书管理员拥有图书录入、图书修改、图书查询和检索和旧书销毁用例。其中图书查询和检索包括图书查询、借阅人查询和出版社联系方式查询的用例。

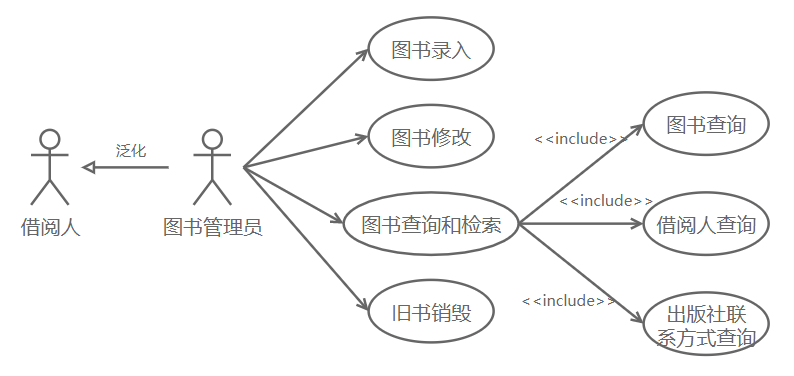


图 3.3 图书管理员的用例图

最高管理员拥有图书管理员的所有用例，除此之外，最高管理员拥有用户管理和授予权限的用例。

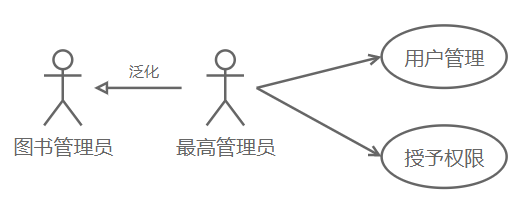


图 3.4 最高管理员的用例图

## **3. 概念模型分析**

根据图书管理系统的的适用对象和所需完成的功能，可以得到如图3.5所示的实体-联系图。

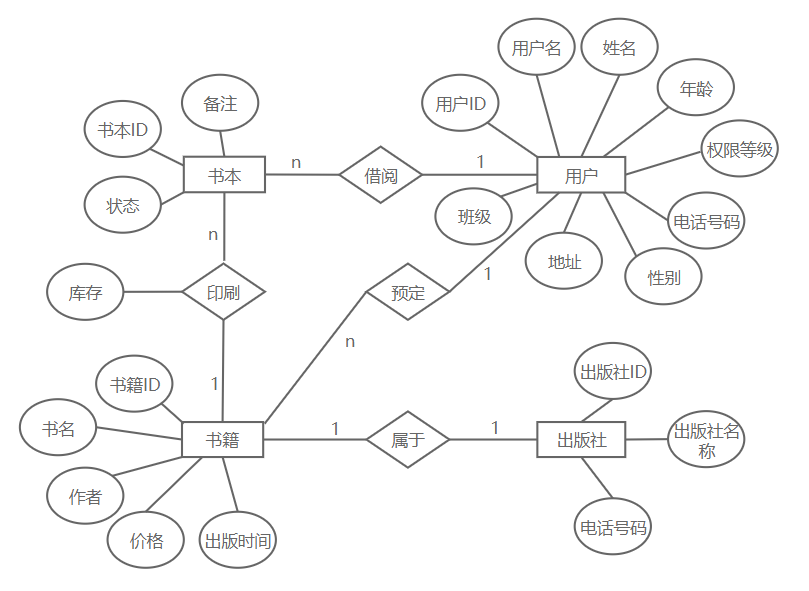


图 3.5 实体-联系图

图书管理系统一共有四个实体，分别为用户、书本、书籍和出版社。用户拥有用户ID、用户名、姓名、年龄、权限等级、电话号码、性别、地址和班级的属性。书本拥有书本ID、状态和备注的属性。书籍拥有书籍ID、书名、作者、价格和出版时间的属性。出版社拥有出版社ID、出版社名称和电话号码的属性。

这四个实体彼此拥有联系。其中一个书籍属于一个出版社，一个书籍可以印刷多个书本，该联系拥有属性库存，一个用户可以借阅多本书本以及一个用户可以预定多种书籍。

# 四、概要设计

**1. xxx**

1.1 …

1.2 …

**2. xxx**

概要设计的具体要求：（1）针对需求分析中建立的用例图，详述每个用例的执行过程（包括那些重要的步骤，剧本）。（2）依据需求陈述，确定系统的中类和对象，用UML建立系统的**初级类图**（**领域类图，**不用完整的列出类的属性和方法，只需列出最主要的即可）。（3）针对每个用例，确定参与的类及其类之间的交互过程，用UML建立每个用例的**顺序图/协作图**。（4）设计数据存储方案，确定数据库表的格式。*（5）确定项目的实施方案，系统运行的各个组件及物理部署，用UML画出系统得****组件图和部署图【可选】***。

# 五、详细设计

**1. xxx**

1.1 …

1.2 …

**2. xxx**

详细设计的具体要求：（1）完善在概要设计中建立的**类图（实现类图）**，详细注明类的所有属性和方法，注明类与类之间的关系、多重性。*（2）设计每个功能的图形界面（采用UML绘制或者VS绘制）【****可选****】*。（3）针对每个用例，设计完成该用例的算法，并用UML建立每个用例的**活动图/流程图**。*（4）在概要设计的基础上，完善数据库的建模，并写出建立****数据库及相关表****的SQL语句****【可选】****。*

# 六、系统实现（编码）

## **1. 系统总体架构实现**

本课题采用前后端分离进行开发，系统架构为MVC架构，后端用Springboot+Mybatis+Redis+Mysql承担Model操作与Controller数据流控制，前端由React构建web应用承担View视图的责任；后端目录结构如图6.1所示；

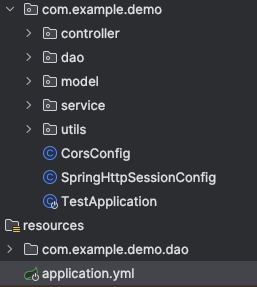


图 6.1 后端目录结构

## 登录注册

登录界面如图6.2所示，注册界面如图6.3所示；

登录与注册模块主要包含用户的注册信息录入、用户登录、用户登录状态校验三个核心功能；

其中用户的密码将会增加自定义盐值并统一转为md5加密值存入数据库以确保安全，后续登录校验都将使用加密后密码进行，避免用户敏感信息明码存于数据库中，其核心加密代码如图6.4所示；

用户的登录状态校验利用session来进行，当用户登录时在session中加入userId字段作为标识，后续涉及仅登录用户或权限操作时将利用session进行登录校验以及权限校验；

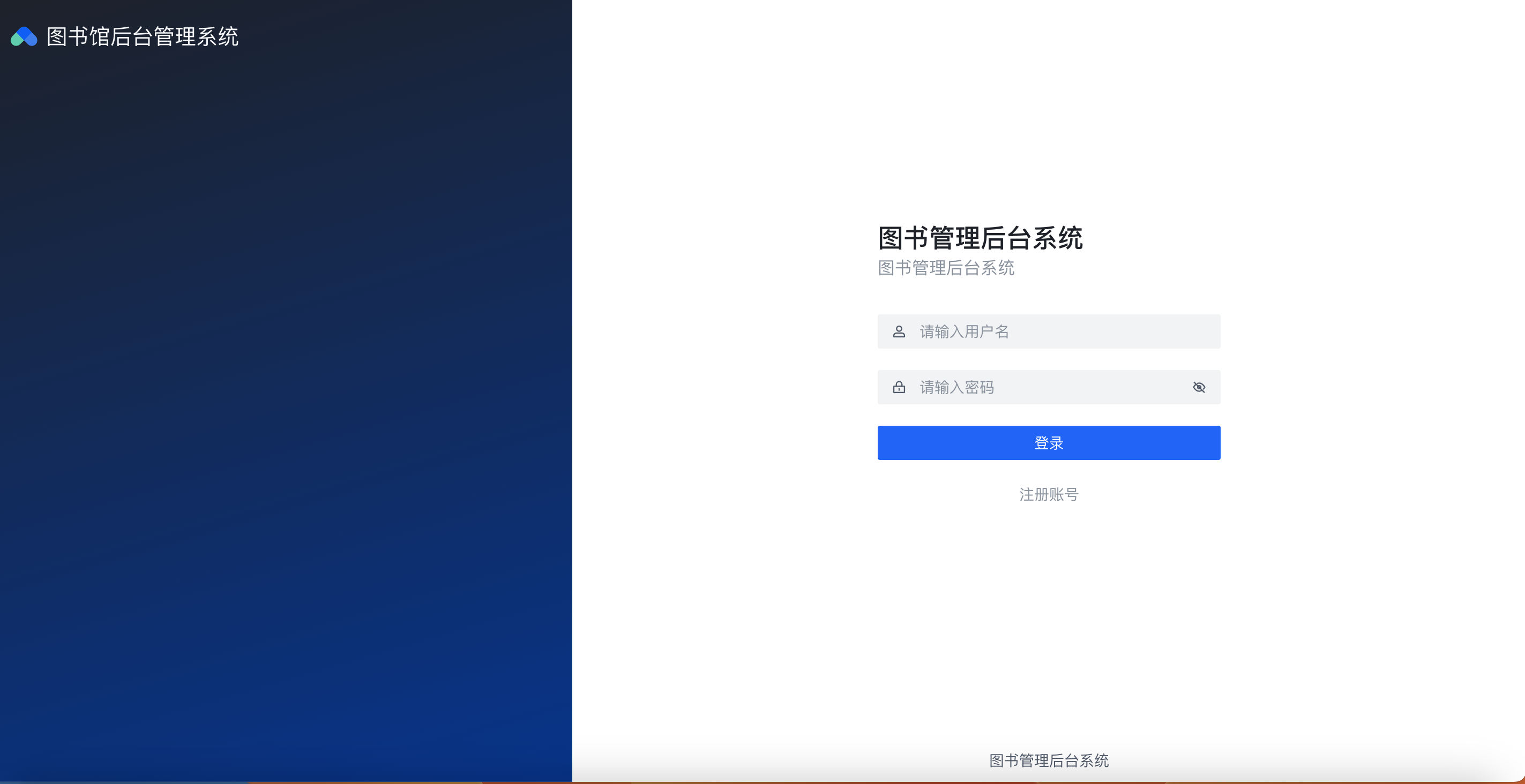


图 6.2 登录页面

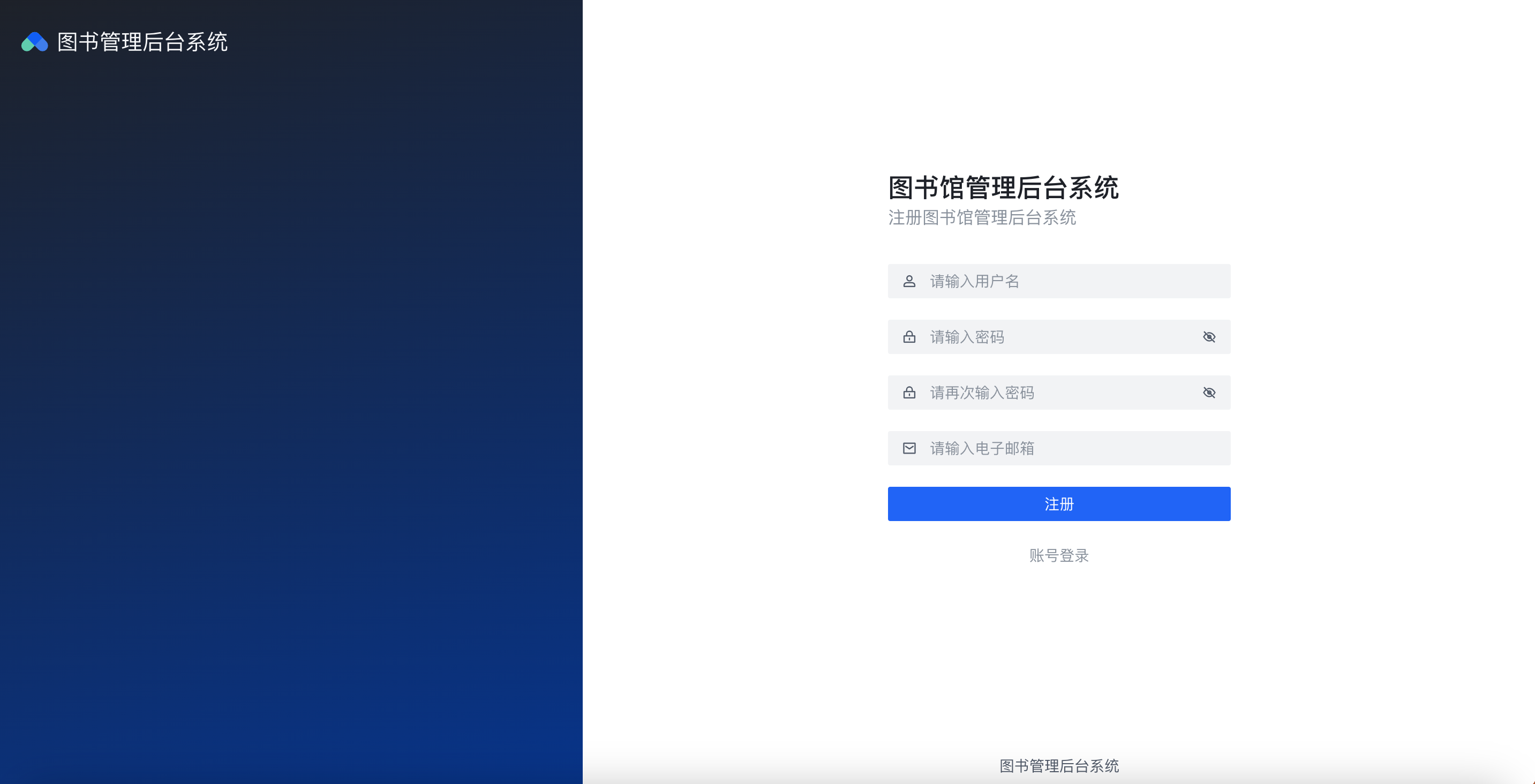
****

图 6.3 注册页面



图 6.4 用户密码加密逻辑

## 图书录入

图书录入界面如图6.5所示；

该模块根据管理员输入信息进行图书信息录入，总体结构分为两层：书籍为第一层，用于记录该书籍相关信息，有唯一的书籍编号作为主键；书本为第二层，用于记录该书籍的每一本具体书本的独立信息，包括是否在库是否损毁等，有对应书籍编号与独立的书本编号作为主键，即（书籍编号，书本编号）；

当录入图书信息时会要求填写书籍库存数量，填入后将在后台创建对应数量的独立书本，后续书本状态的管理会利用该结构，该部分核心代码如图6.6所示；

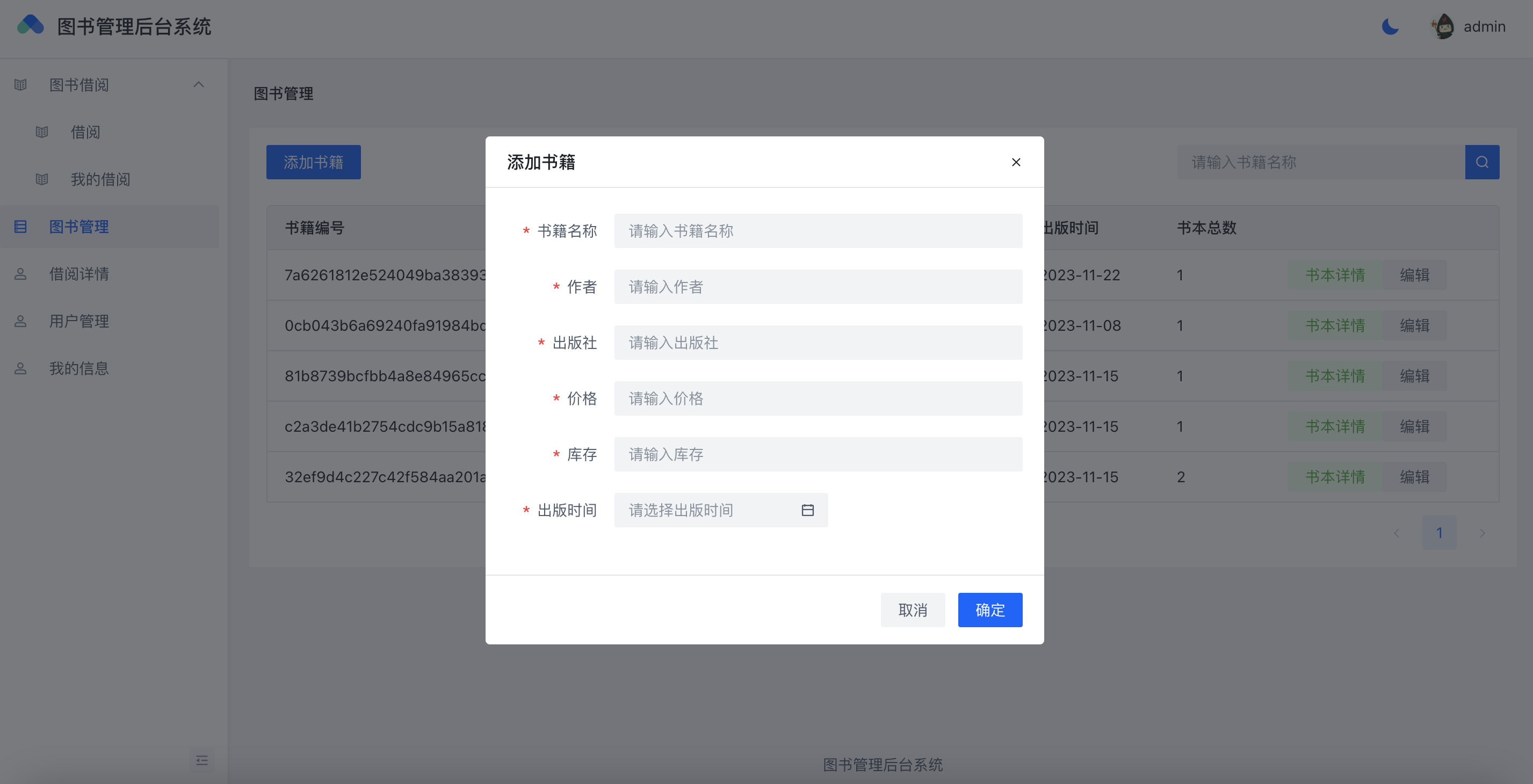


图 6.5 图书录入页面



图 6.6 图书录入核心逻辑

## 书籍检索

书籍检索模块与借阅模块在同一个页面，如图6.7所示为整体页面；

该模块主要通过各关键词的模糊搜索获取用户所需的书籍列表并返回给前端进行显示，其核心数据库搜索逻辑如图6.8所示；

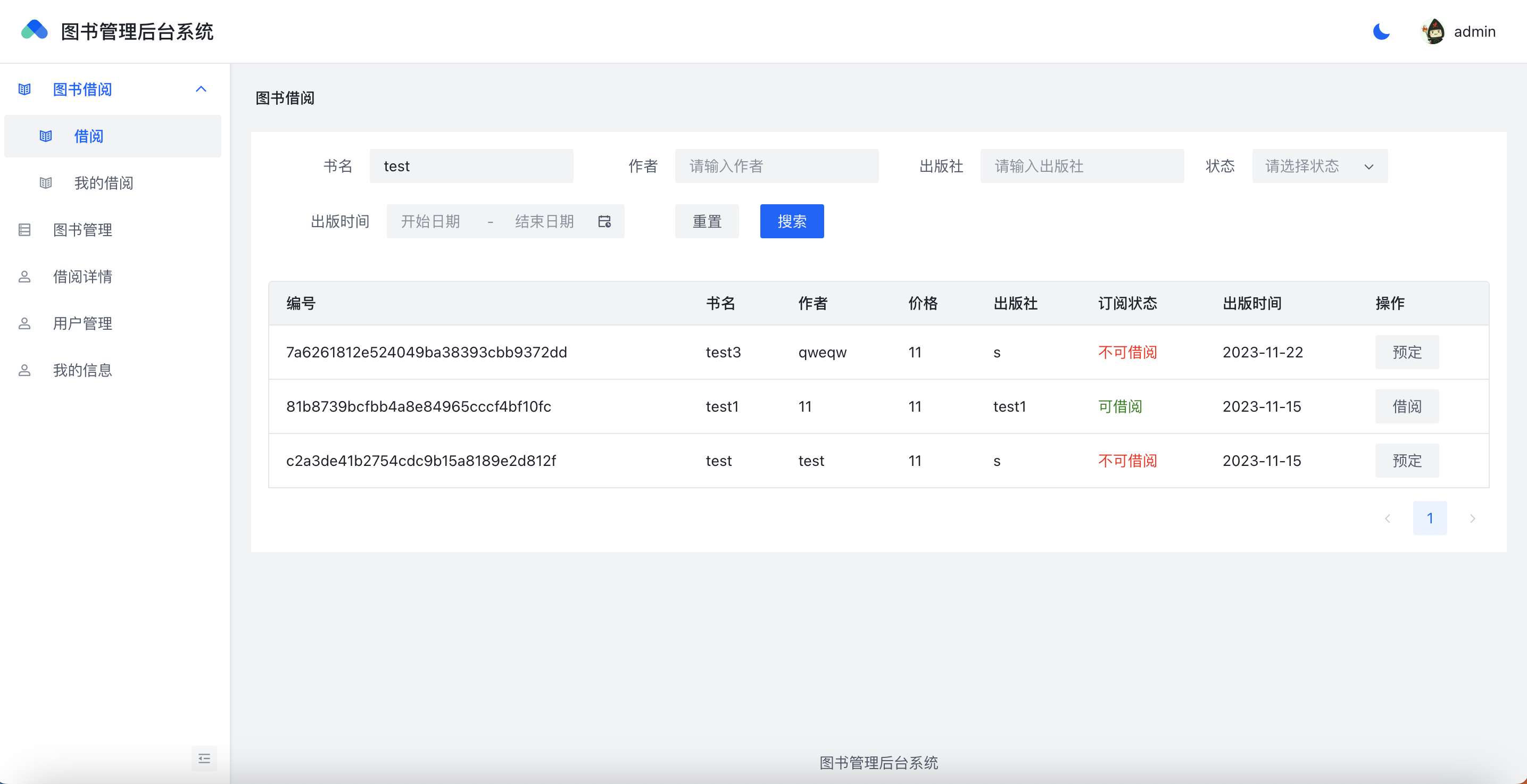


图 6.7 书籍检索（借阅）页面

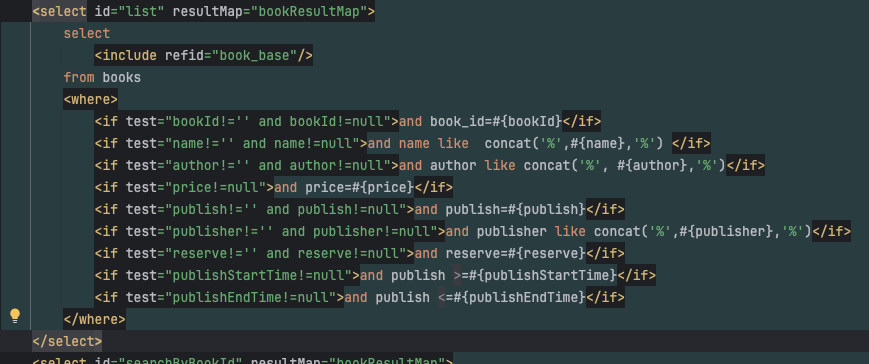


图 6.8 书籍搜索逻辑

## 书籍借阅与预定

书籍借阅界面如图6.9所示与书籍检索属于同一页面，其中书籍根据库存数量以及库存内书本状态分为可借阅与不可借阅状态，不可借阅的图书可以进行预定，可借阅的图书可直接借阅，归还书籍界面如图6.10所示；

借阅书籍以及归还书籍皆需要考虑并发冲突因此对于具体service加锁避免产生冲突问题，同时归还书籍会触发对预定该书籍用户的提醒逻辑，借阅核心代码如图9所示，归还核心代码如图6.11所示；

当不可借阅当书籍有用户归还了该书籍后将会触发预定书籍提示逻辑，具体实现方案为根据预定用户的邮件地址群体发送对应提示邮件；邮件发送逻辑如图6.12所示；

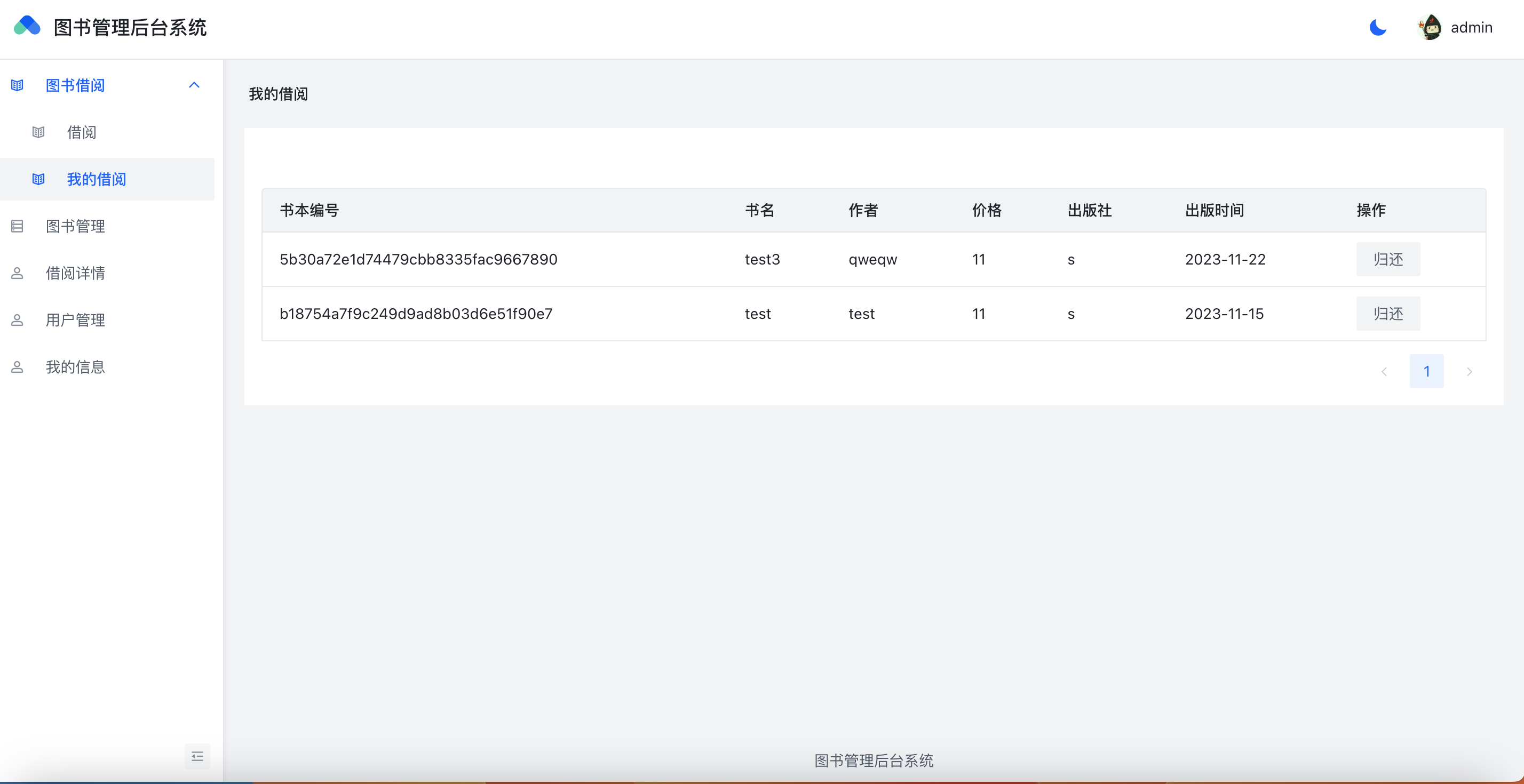
****

图 6.9 归还图书页面

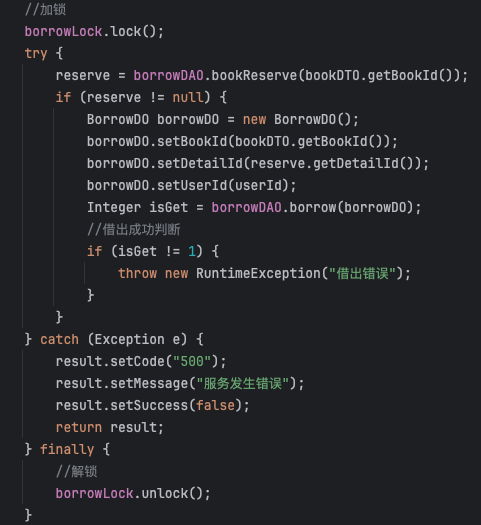


图 6.10 借阅书籍核心逻辑



图 6.11 归还书籍核心逻辑



图 6.12 发送提醒邮件逻辑

## 书籍状态管理

书籍状态管理具体事实上是管理每一本书本的状态，核心状态有可借阅与不可借阅两种，而书籍毁损等原因均作为每条书本数据的备注存在，因此当管理员想要对淘汰、损坏、丢失的书目进行条目修改时只需将该具体书本的状态置为不可借阅并备注原因即可；状态管理界面如图6.13所示；

当某一书籍所有现存书本的状态都为不可借阅，该书籍将在借阅界面显示为不可借阅，知道用户将已借的该书籍归还或者管理员将不可借阅的书本重新设置为可借阅状态后该书籍才能再次被借阅；具体数据库更新语句如图6.14所示；

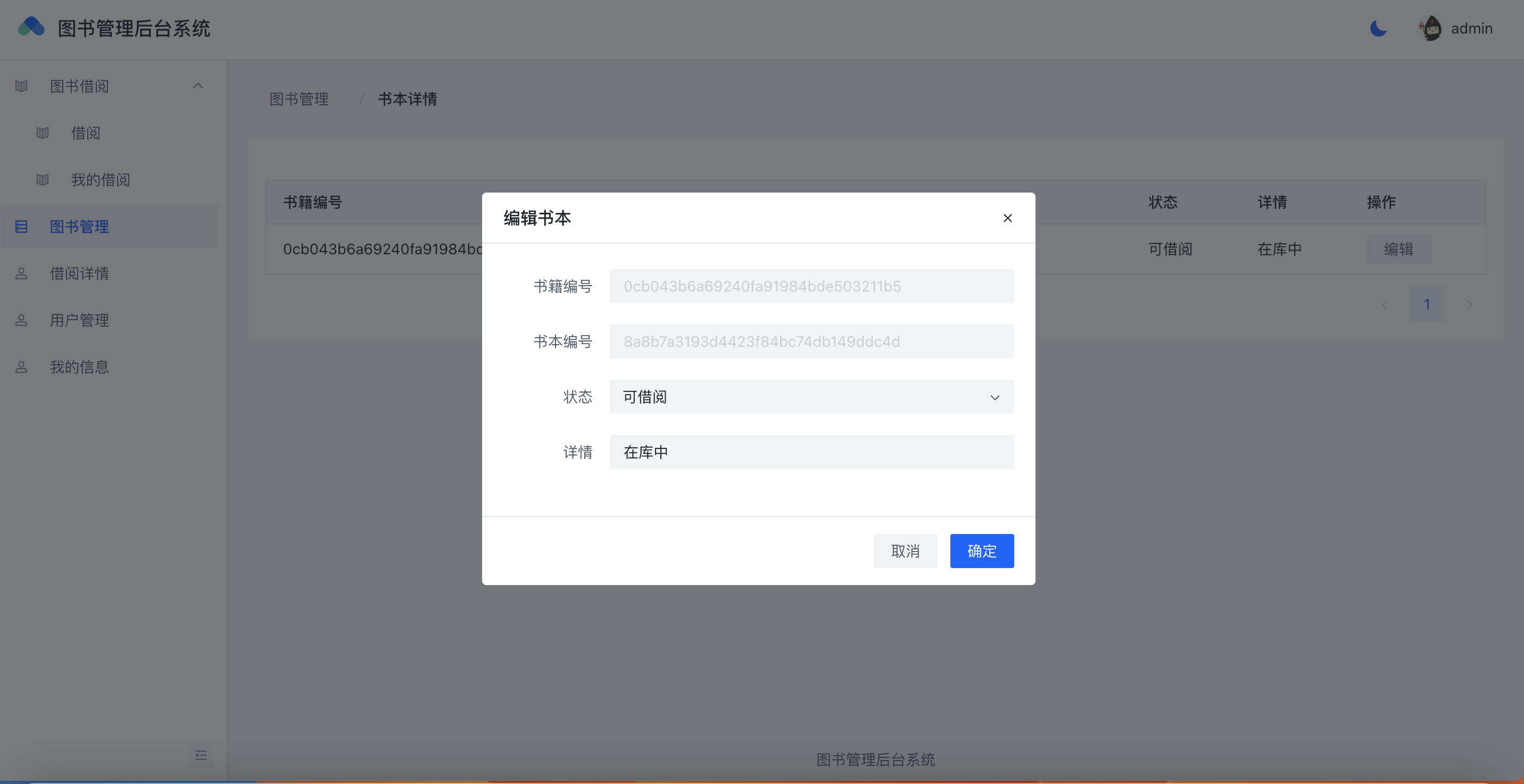


图 6.13 书籍状态管理界面

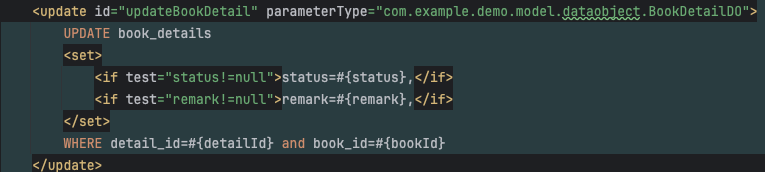


图 6.14 书籍状态信息更新

## 用户权限管理

用户权限分为3个等级：

1级权限是普通用户，仅限借阅与归还书籍；

2级权限是书籍管理员，在1级的基础上允许对书籍进行录入与状态管理；

3级权限是最高管理员，在2级的基础上可以额外修改所有用户账号信息和权限；

用户权限数据库更新语句如图6.15所示；

在用户登录后前端会将用户权限信息记录在本地，而后由路由中各页面对应的权限等级进行页面显示，具体逻辑如图6.16、图6.17和图6.18所示；

****

图 6.15 权限更新

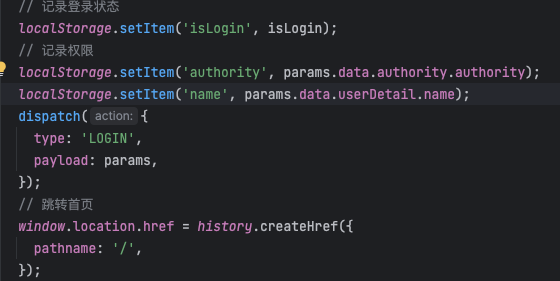


图 6.16 存储用户权限

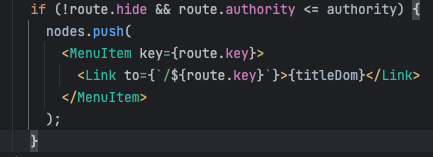


图 6.17 根据权限显示对应目录内容

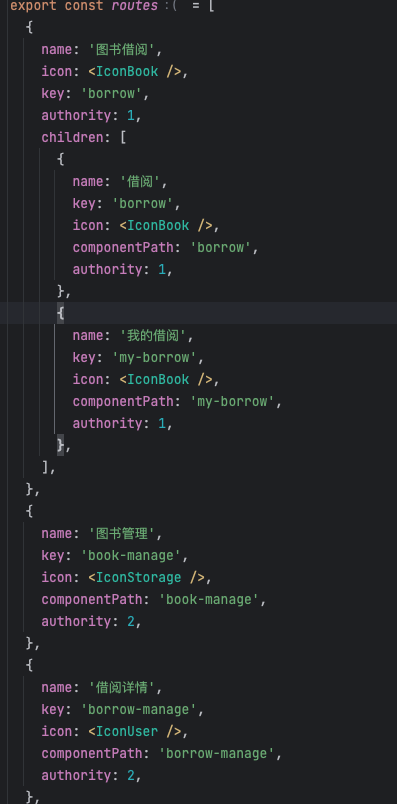


图 6.18 前端路由权限映射

# 系统测试

## 登录与注册

对于注册功能：

预设条件：用户名：4-20位字符，密码：6-20位数字字母下划线组合（不需要同时有数字字母下划线），邮箱必须是正常邮箱的格式。可划分等价类如下：

符合用户名、密码、邮箱要求的正常情况。

对应用例1：用户名 "valid\_user"，密码 "Pass\_123"，邮箱["user@example.com"](mailto:\"user@example.com\")

用户名长度小于4位。

对应用例2：用户名 "usr"，密码 "Pass\_123"，邮箱 ["user@example.com"](mailto:\"user@example.com\")

用户名长度大于20位。

对应用例3：用户名 "usernamewithmorethan20characters"，密码 "Pass\_123"邮箱 "user@example.com"

密码长度小于6位。

对应用例4：用户名 "user1234"，密码 "P\_12"，邮箱 "user@example.com"

密码长度大于20位。

对应用例5：用户名 "user1234"，密码"Password\_with\_more\_than\_20\_characters"，邮箱 "user@example.com"

密码中有非数字字母下划线。

对应用例6：用户名 "user1234"，密码 "Pass\_123@"，邮箱 "user@example.com"

邮箱不符合正常邮箱格式。

对应用例7：用户名 "user1234"，密码 "Pass\_123"，邮箱 "invalid\_email"

用例8:不输入用户名、密码或邮箱

用例9:密码二次确认与第一次输入不一致

对于登录功能：

用例10:正常输入正确账号密码

用例11:输入不正确账号，正确密码

用例12:输入正确账号，不正确密码

根据以上用例，对相应系统模块对验收表如表2所示；

表 2 登录与注册验收表

|  |  |  |
| --- | --- | --- |
| 用例 | 测试结果 | 预期结果 |
| 用例1 | 通过 | 正常注册 |
| 用例2 | 通过 | 无法注册并给出对应提示 |
| 用例3 | 通过 |
| 用例4 | 通过 |
| 用例5 | 通过 |
| 用例6 | 通过 |
| 用例7 | 通过 |
| 用例8 | 通过 |
| 用例9 | 通过 |
| 用例10 | 通过 | 正常登录 |
| 用例11 | 通过 | 无法登陆并给出对应提示 |
| 用例12 | 通过 |

验收意见：当前模块已基本完善，无进一步修改意见。

## 图书录入与修改

图书录入与修改的等价类如下：

完整填写所有信息

用例1:名称 testBook、作者admin、出版社ujss、价格50、库存10和出版时间2020.1.5

有缺失的填写信息

用例2:在用例1的基础上轮流空缺某些信息即可

正常修改图书的某信息

用例3:将用例1的书籍名称改为BookTest

将字段修改为空值

用例4:将用例1的书籍名称改为空

修改图书的库存数

用例5:修改用例1的库存数为11

根据以上用例，对相应系统模块对验收表如表3所示；

表 3 图书录入与修改验收表

|  |  |  |
| --- | --- | --- |
| 用例 | 测试结果 | 预期结果 |
| 用例1 | 通过 | 正常添加 |
| 用例2 | 通过 | 无法添加并提示 |
| 用例3 | 通过 | 成功修改 |
| 用例4 | 通过 | 不允许修改并提示 |
| 用例5 | 不通过 | 修改成功并且书籍详情内多出一本书本 |

验收意见：图书无法修改库存，也就是无法新入库已有记录的书籍，且没有书籍重复检测，应当对重复书籍进行提示并提供已有书籍增加库存的通道。

## 图书查询与检索

对于图书查询与检索有如下等价类：

正常情况下的查询。

用例1:查询书籍名称BookTest

用例2:查询作者：admin

用例3:查询出版社：ujss

用例4:查询状态：可借阅

用例5:查询出版时间：1999.1.1-2023.1.1

用例6:查询书籍名称：test

用例7:查询书籍名称为BookTest且作者为admin的书籍

无匹配结果的查询。

用例8:查询书籍名称：null

根据以上用例，对相应系统模块对验收表如表4所示

表 4 图书查询检索验收表

|  |  |  |
| --- | --- | --- |
| 用例 | 测试结果 | 预期结果 |
| 用例1 | 通过 | 显示对应书籍 |
| 用例2 | 通过 |
| 用例3 | 通过 |
| 用例4 | 通过 |
| 用例5 | 通过 |
| 用例6 | 通过 | 显示所有名称中含有test的书籍 |
| 用例7 | 通过 | 显示该书籍 |
| 用例8 | 通过 | 无书籍显示 |

验收意见：查询与检索无功能性问题。

## 图书借阅与预定

对于图书借阅与预定模块将先进行功能黑盒测试，后对核心接口（借书，还书）利用postman进行额外白盒测试。

以下是正常黑盒测试用例：

用例1:选择可借阅的图书进行借阅

用例2:选择不可借阅的图书预定

用例3:归还已借图书

下面是针对两个接口的白盒测试方案以及用例：

针对借书接口，存在逻辑判断：用户是否登录；书籍id是否为空；该书籍是否还有库存；

针对还书接口，存在逻辑判断：用户是否登录；书籍id是否为空以及书本id是否为空；

因此有以下用例：

用例4:borrow（借书）接口参数：{

"bookId": "c2a3de41b2754cdc9b15a8189e2d812f",

}，携带登录的即时cookie

用例5:borrow接口参数：{

"bookId": "c2a3de41b2754cdc9b15a8189e2d812f",

}，不携带cookie

用例6:borrow接口参数：{

"bookId": "",

}，携带登录的即时cookie

用例7:return（还书）接口参数：{

"detailId":"b18754a7f9c249d9ad8b03d6e51f90e7",

"bookInfo":{

"id": null,

"bookId": "c2a3de41b2754cdc9b15a8189e2d812f",

"name": "test",

"author": "test",

"publisher": "s",

"price": 11,

"publish": "2023-11-14T16:00:00.000+00:00",

"reserve": 1

}

}，携带cookie

用例8:return接口参数：{

"detailId":"b18754a7f9c249d9ad8b03d6e51f90e7",

"bookInfo":{

"id": null,

"bookId": "c2a3de41b2754cdc9b15a8189e2d812f",

"name": "test",

"author": "test",

"publisher": "s",

"price": 11,

"publish": "2023-11-14T16:00:00.000+00:00",

"reserve": 1

}

}，不携带cookie

用例9:return接口参数：{

"detailId":"",

"bookInfo":{

"id": null,

"bookId": "c2a3de41b2754cdc9b15a8189e2d812f",

"name": "test",

"author": "test",

"publisher": "s",

"price": 11,

"publish": "2023-11-14T16:00:00.000+00:00",

"reserve": 1

}

}，携带cookie

用例10:return接口参数：{

"detailId":"b18754a7f9c249d9ad8b03d6e51f90e7",

"bookInfo":{

"id": null,

"bookId": "",

"name": "test",

"author": "test",

"publisher": "s",

"price": 11,

"publish": "2023-11-14T16:00:00.000+00:00",

"reserve": 1

}

}，携带cookie

用例11:归还已被预定的的书籍

用例4和用例7的postman测试截图分别如图7.1和图7.2所示；

根据以上用例，对相应系统模块对验收表如表5所示；

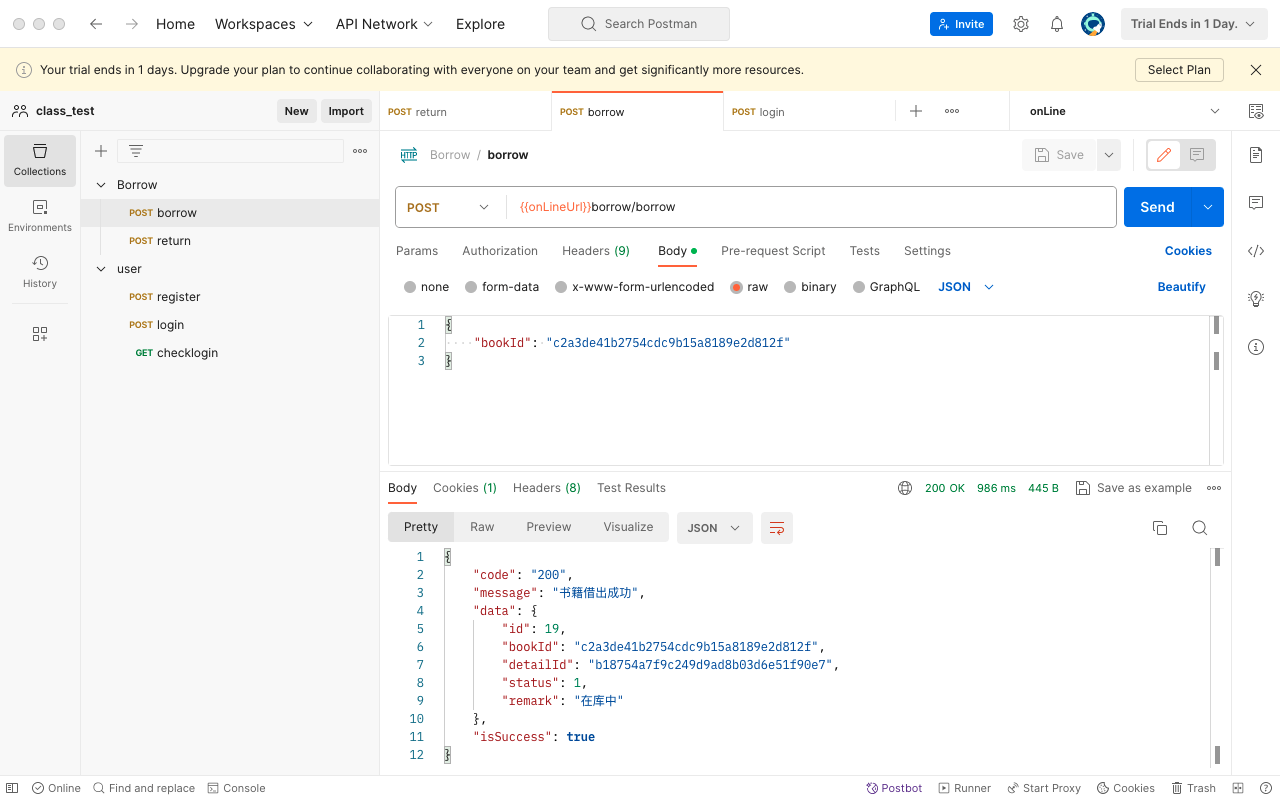


图 7.1 borrow接口用例4测试结果

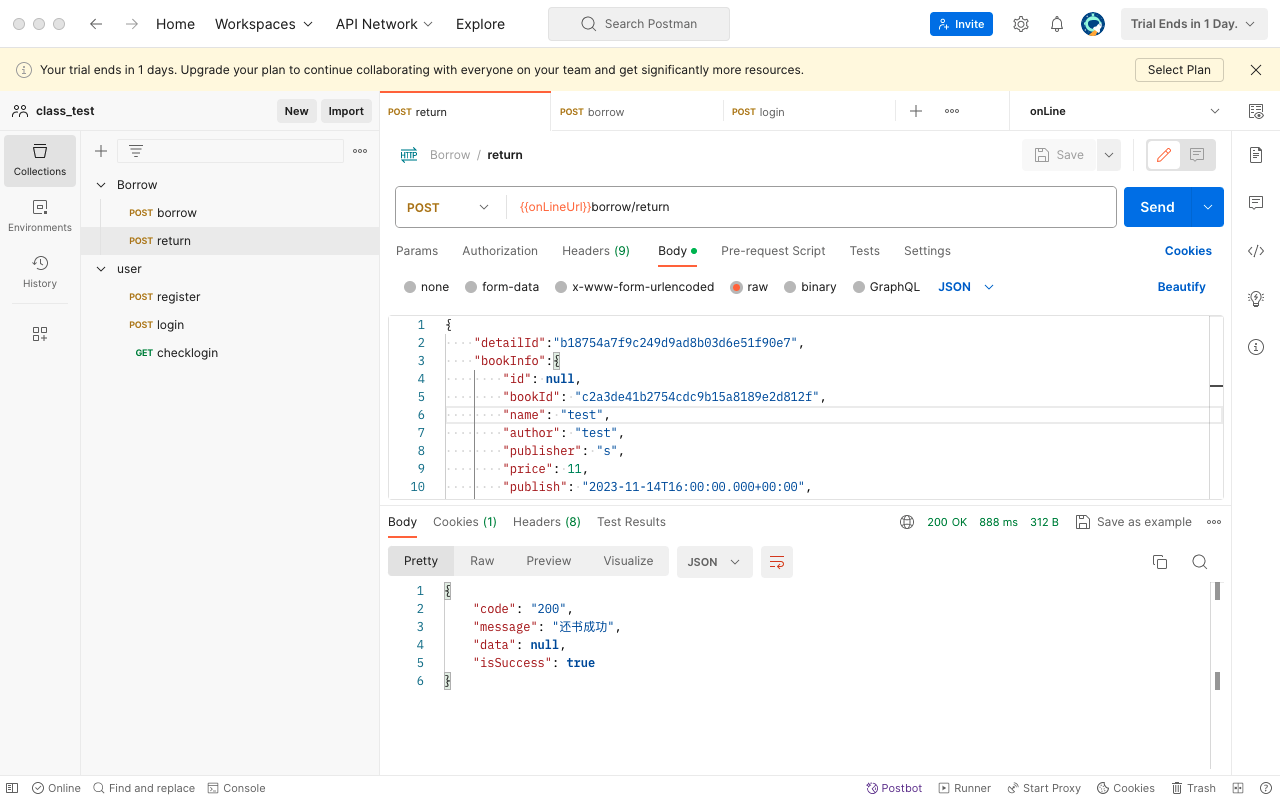


图 7.2 return接口用例7测试结果

表 5 图书借阅与预定验收表

|  |  |  |
| --- | --- | --- |
| 用例 | 测试结果 | 预期结果 |
| 用例1 | 通过 | 成功借阅 |
| 用例2 | 通过 | 成功预定 |
| 用例3 | 通过 | 成功归还 |
| 用例4 | 通过 | 接口返回200，成功借阅 |
| 用例5 | 通过 | 接口返回401，尚未登录 |
| 用例6 | 通过 | 接口返回400，未选择正确书本 |
| 用例7 | 通过 | 接口返回200，还书成功 |
| 用例8 | 通过 | 接口返回401，尚未登录 |
| 用例9 | 通过 | 接口返回400，未选择正确书本 |
| 用例10 | 通过 | 接口返回400，未选择正确书本 |
| 用例11 | 通过 | 预定用户收到提醒邮件 |

验收意见：功能实现已吻合需求，但是仍存在可优化空间，当后台管理员将不可借阅书籍置为可借阅时，原本预定该书籍的用户并不会收到提示，可以在每次修改图书状态为可借阅时进行额外操作，向已预定用户发送提示。

## 图书状态管理

对图书状态管理有如下用例：

用例1:将有多本库存的书籍全部置为不可借阅

用例2:将有多本库存的书籍部分置为不可借阅

用例3:改变书本备注

根据以上用例，对相应系统模块对验收表如表6所示；

表 6 图书状态管理验收表

|  |  |  |
| --- | --- | --- |
| 用例 | 测试结果 | 预期结果 |
| 用例1 | 通过 | 该书籍在借阅界面不可借阅 |
| 用例2 | 通过 | 该书籍在借阅界面可借阅 |
| 用例3 | 通过 | 书本详情备注对应改变 |

验收意见：只能对书本单独操作，缺少批量操作的手段，可优化。

## 6.用户权限管理

对用户权限管理有如下用例：

用例1:修改用户test权限为1

用例2:修改用户test权限为2

用例3:修改用户test权限为3

用例4:用最低权限账户登录，直接修改路由链接进入需要最高权限的用户管理页面

根据以上用例，对相应系统模块对验收表如表7所示；

表 7 用户权限管理验收表

|  |  |  |
| --- | --- | --- |
| 用例 | 测试结果 | 预期结果 |
| 用例1 | 通过 | 该用户只能进入图书借阅相关页面和个人信息修改页面 |
| 用例2 | 通过 | 该用户不能进入用户管理页面 |
| 用例3 | 通过 | 该用户可进入所有页面包括用户管理页面 |
| 用例4 | 不通过 | 用户无法直接进入用户管理页面 |

验收意见：权限管理实现方案仅实现了页面隐藏，而没有在具体接口层面对权限做出身份验证权限验证，导致用户若知道对应页面的路由名称，可直接利用无视权限链接进入对应页面并进行越权操作。

必须对所有需要高级权限的接口做好权限验证操作，包括数据请求与操作请求等接口，规避越权风险。

# 八、总结与改进

## 1.总结

系统的总体目标是建立一个内部图书借阅管理系统，旨在实现系统化、规范化和自动化的图书借阅流程。主要功能模块包括对图书的录入和修改、借阅人的注册和登录、方便的查询方法、图书的借阅和预订、旧书销毁功能以及用户权限管理。数据库采用Mysql，后端使用Spring Boot、MyBatis、Redis，前端采用React，实现了前后端分离的架构。

## 2.改进点

在测试过程中，发现了一些需要优化的点：

1.图书库存修改问题：目前系统存在无法修改图书库存的问题，特别是对于已有记录的书籍。应添加功能，使管理员能够对已有书籍进行库存的修改。

2.图书重复检测：缺乏对重复书籍的检测机制，系统应提供提示并避免添加相同的书籍信息。

3.预定通知问题：需要在管理员将不可借阅书籍置为可借阅时，向已预定用户发送通知，确保他们能够及时借阅。

4.批量操作功能：缺少对图书的批量操作功能，这可通过系统添加批量操作的手段来提高管理员效率。

5.权限验证不完整：目前权限管理仅实现了页面隐藏，缺乏在接口层面对权限进行身份验证，存在越权操作的潜在风险，应加强权限验证的细节实现。

# 九、参考文献

[1]Web项目前后端分离的设计与实现——《软件工程》2020年 第4期 | 王建 罗政 张希 张梦琪 张科 马文成 长江大学地球科学学院 湖北武汉430100 长江大学地球物理与石油资源学院 湖北武汉430100