

《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**

**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。

Mysql完成数据库搭建，Springboot+Mybatis+Redis实现后端控制器，React进行页面开发，前后端分离，前端自主持有路由，并单独存在于一个服务器地址，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 |

1. **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

1. **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. xxx**

1.1 …

1.2 …

**2. xxx**

需求分析的具体要求：（1）针对项目的基本要求（不限于、不拘于这些要求），详细说明项目所要完成的每项功能（必须足够的详细），用**软件层次图**描述功能之间的依赖关系。（2）确定系统的参与者及其关联的功能，用UML画出系统的**用例图**（建议用Visio）。（3）确定系统中需要存储的数据对象，建立系统的实体-联系图。

**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1所示；

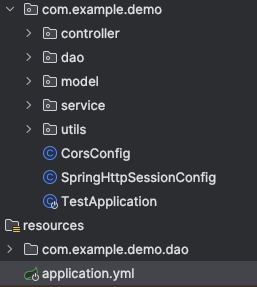


图 1 后端目录结构

1. **登录注册**

登录界面如图1所示，注册界面如图2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3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；

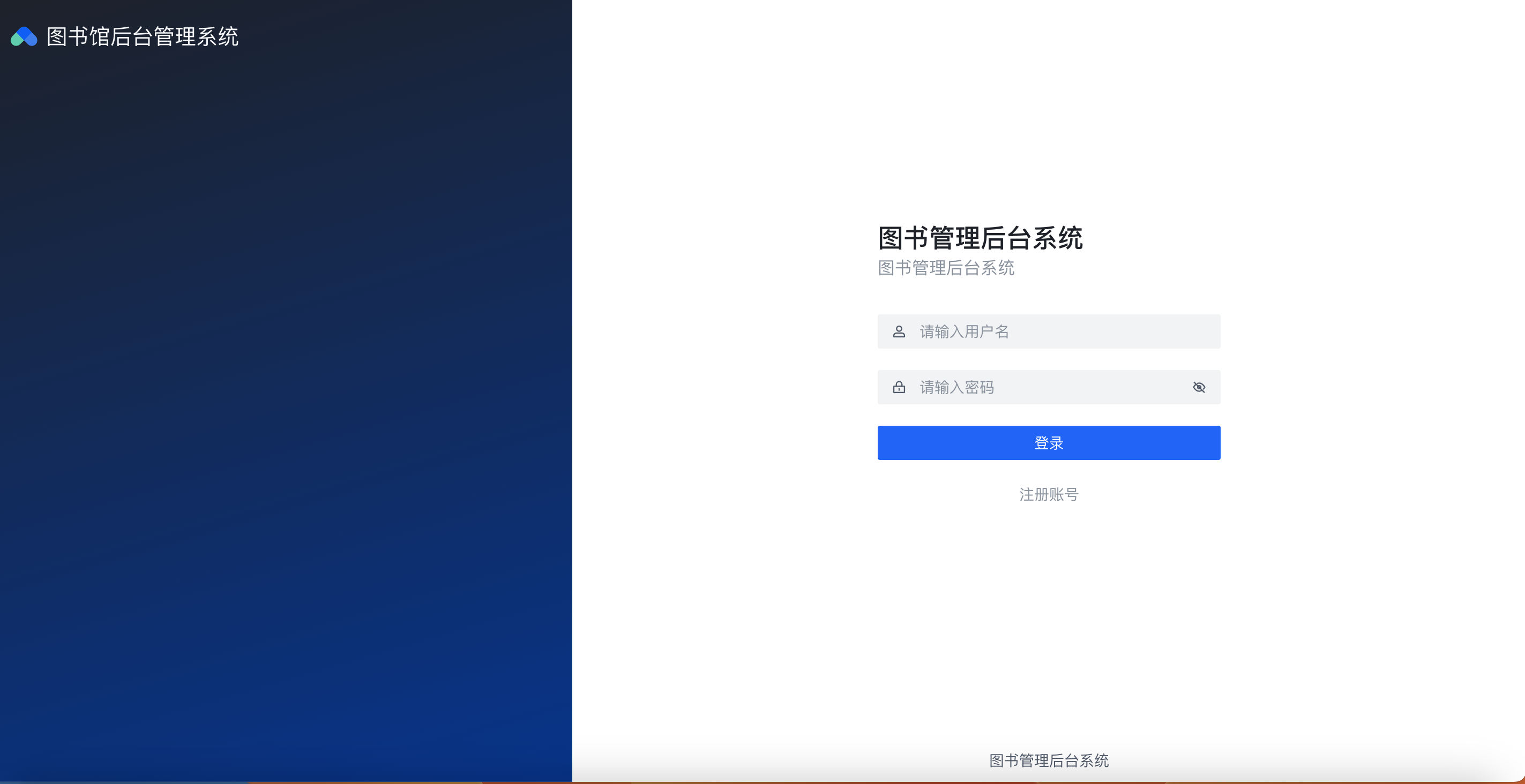


图 2 登录页面

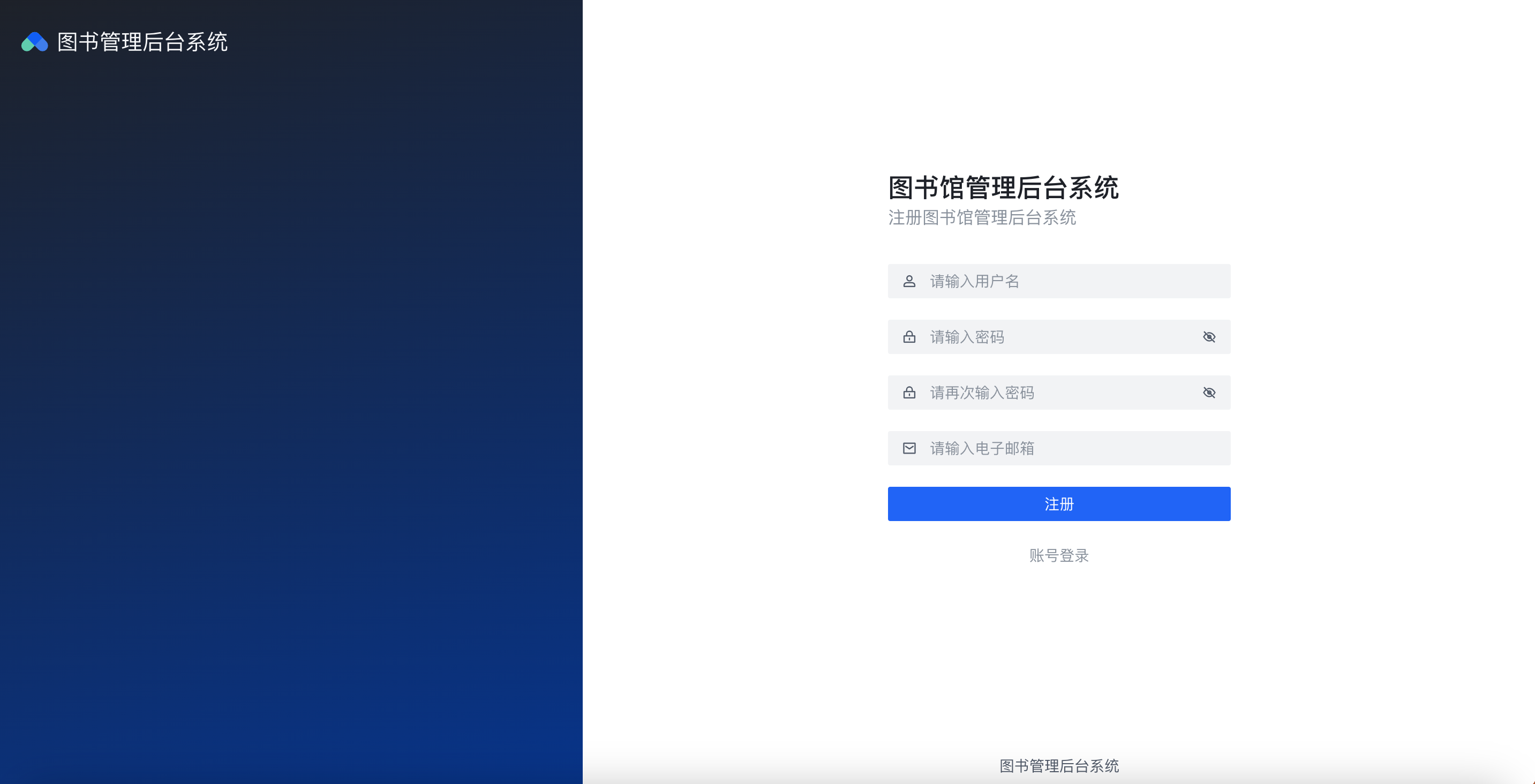
****

图 3 注册页面



图 4 用户密码加密逻辑

1. **图书录入**

图书录入界面如图4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5所示；

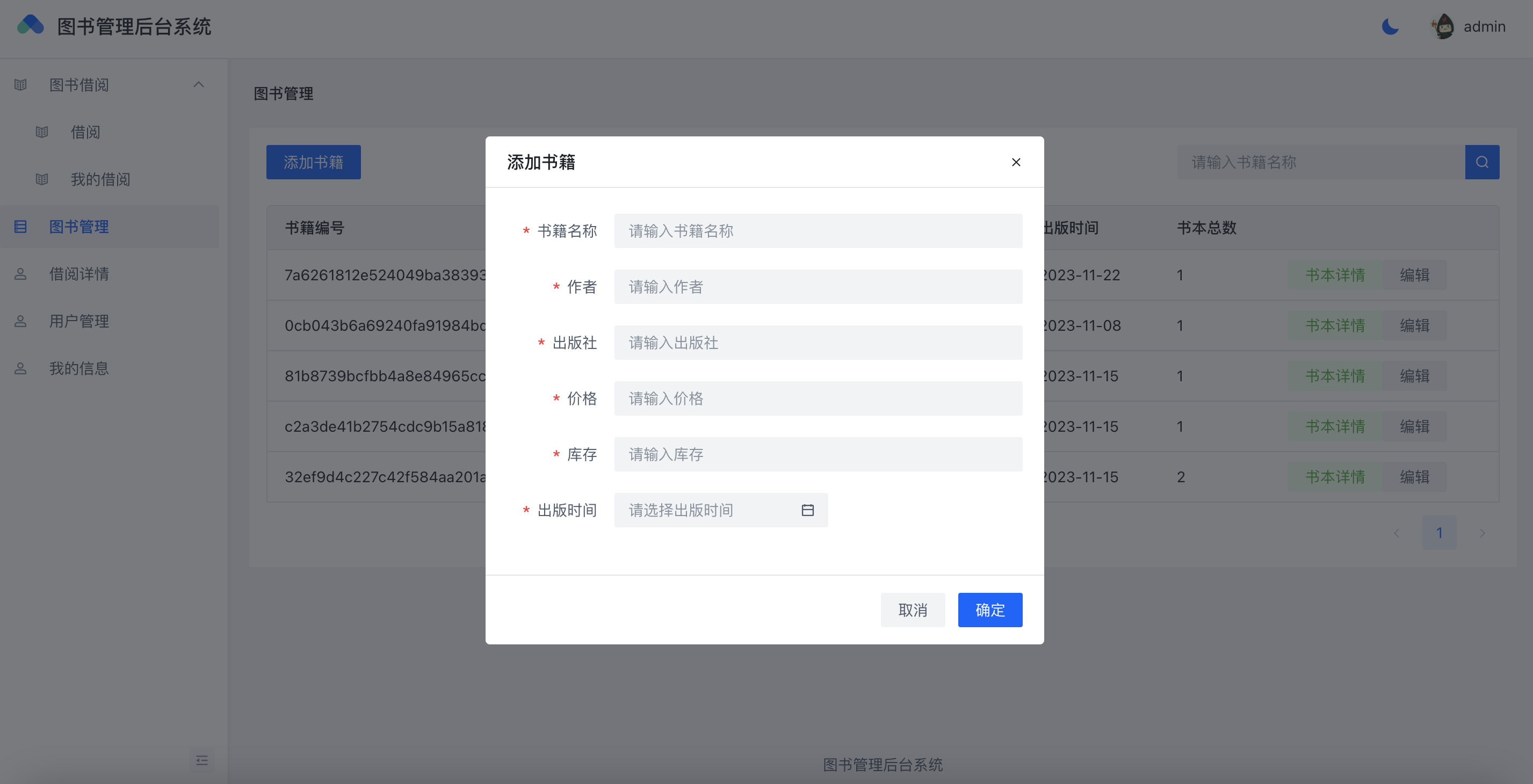


图 5 图书录入页面



图 6 图书录入核心逻辑

1. **书籍检索**

书籍检索模块与借阅模块在同一个页面，如图6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7所示；

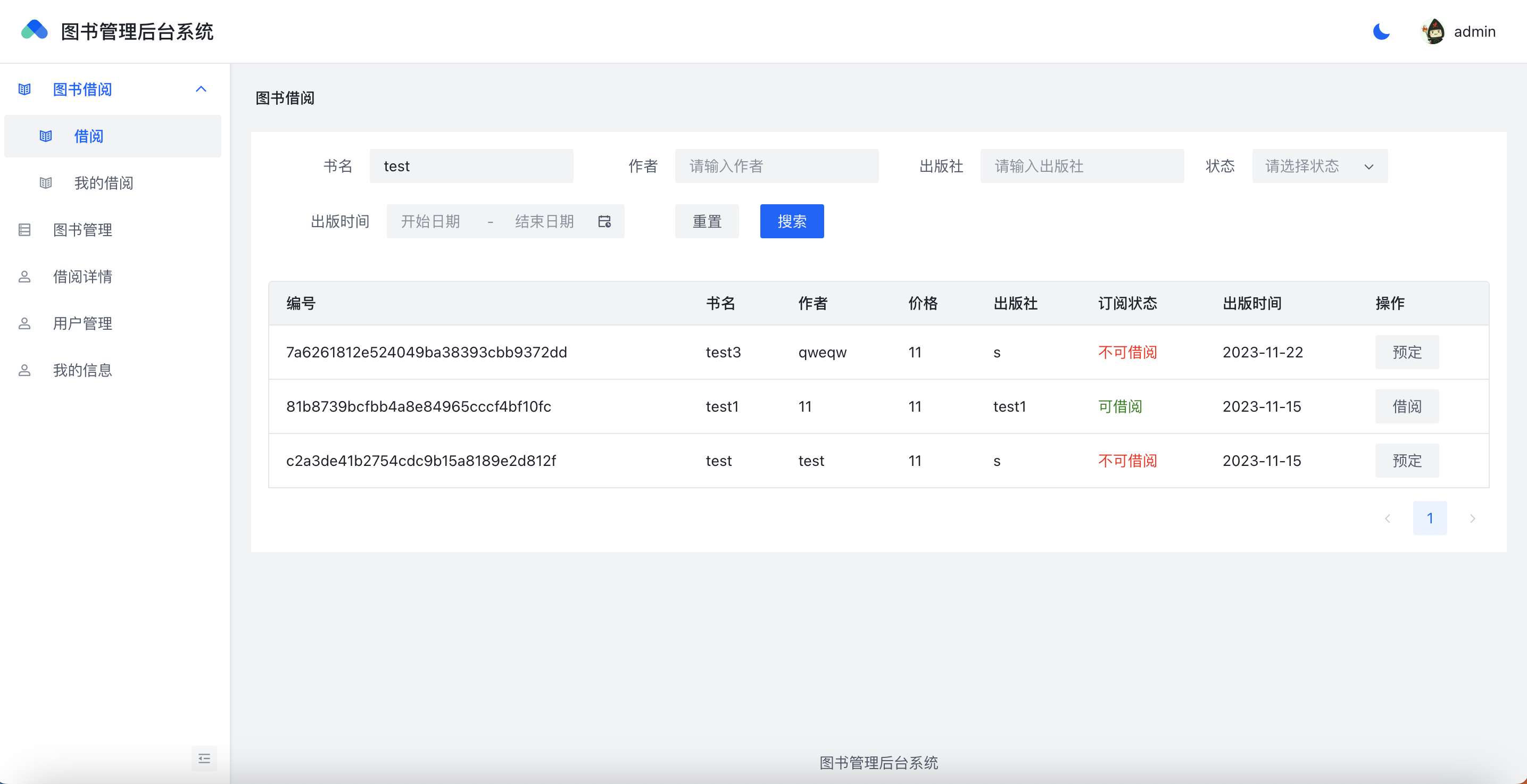


图 7 书籍检索（借阅）页面

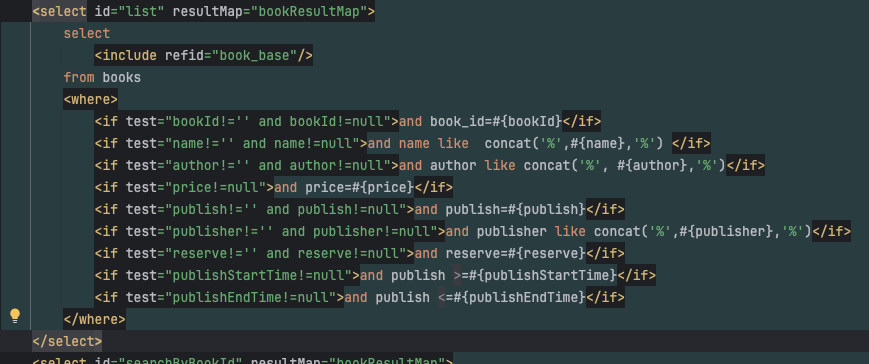


图 8 书籍搜索逻辑

1. **书籍借阅与预定**

书籍借阅界面如图6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8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10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11所示；

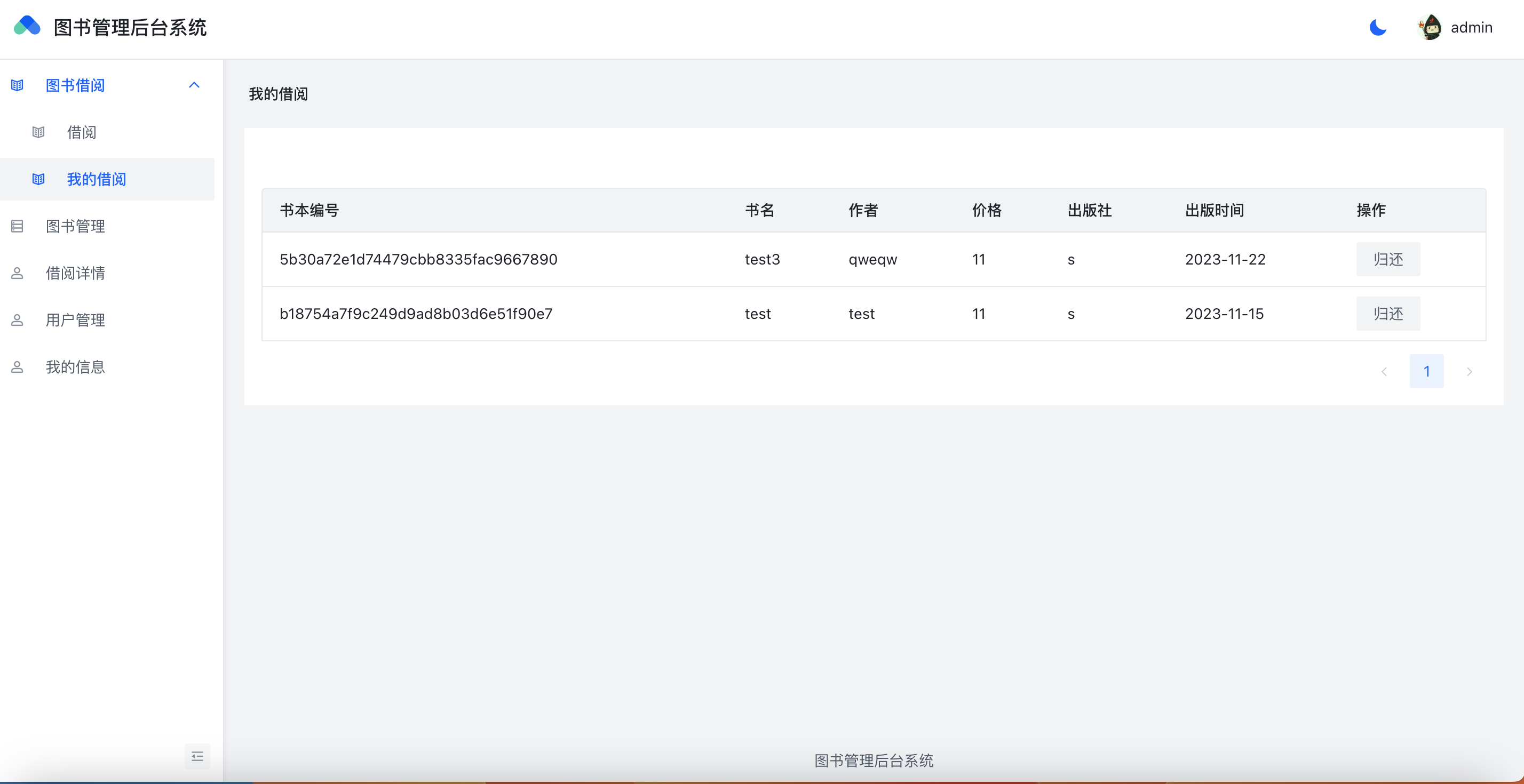
****

图 9 归还图书页面

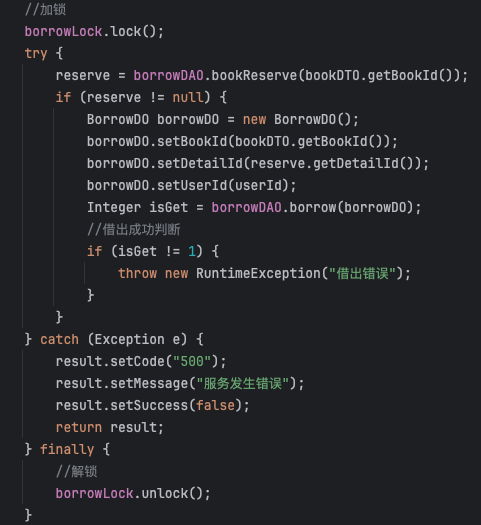


图 10 借阅书籍核心逻辑



图 11 归还书籍核心逻辑



图 12 发送提醒邮件逻辑

1. **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12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13所示；

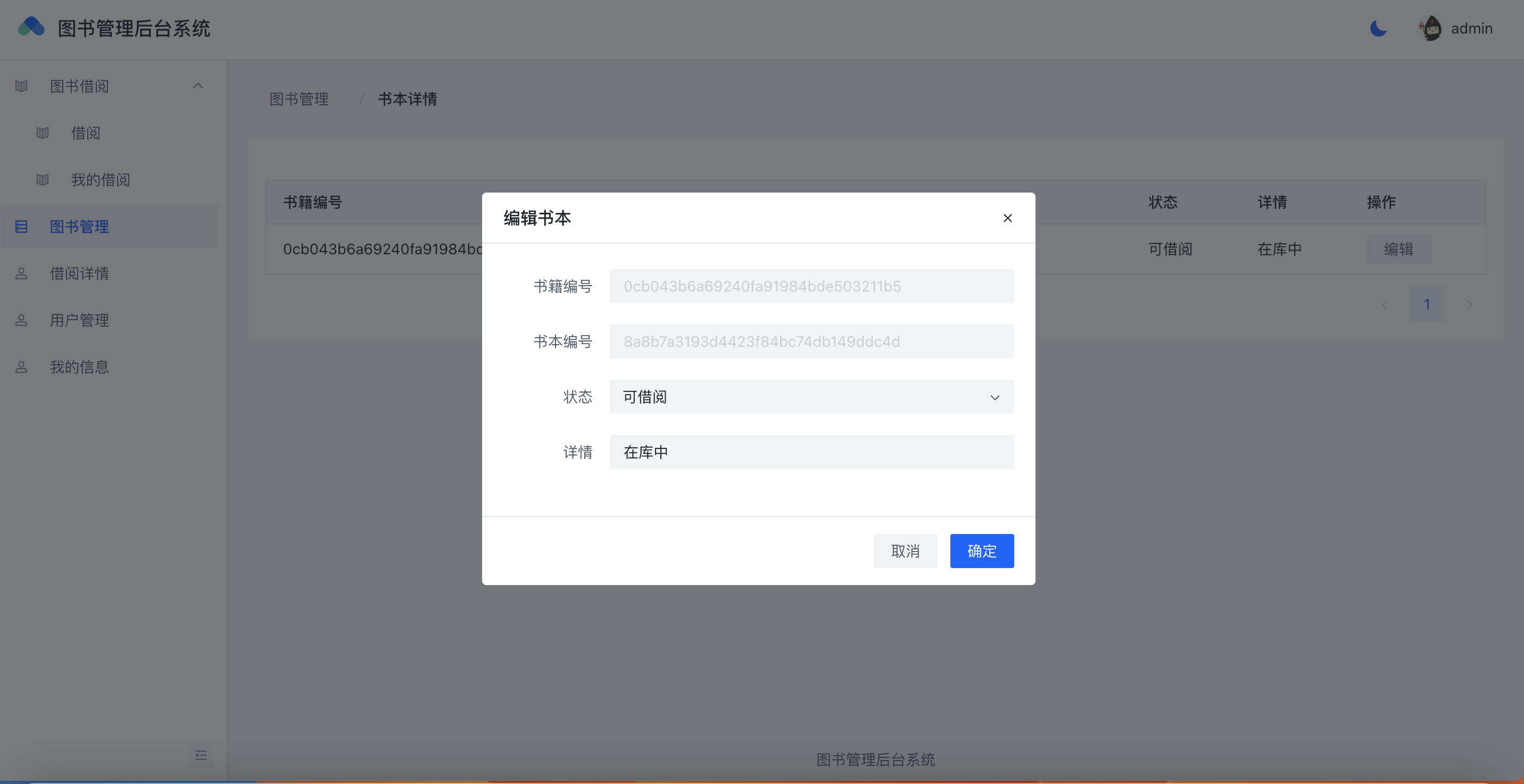


图 13 书籍状态管理界面

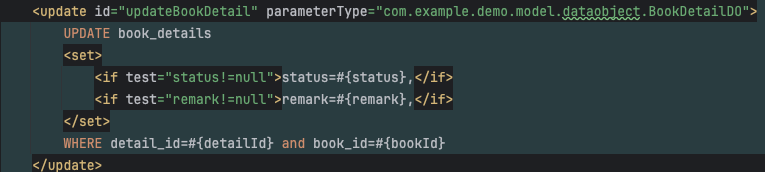


图 14 书籍状态信息更新

1. **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14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15、图16和图17所示；

****

图 15 权限更新

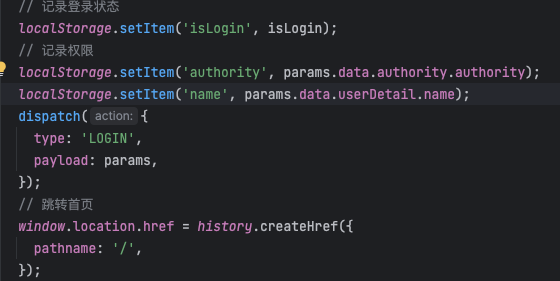


图 16 存储用户权限

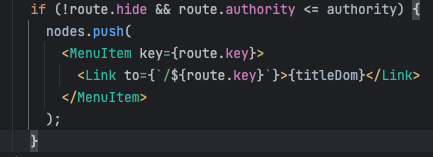


图 17 根据权限显示对应目录内容

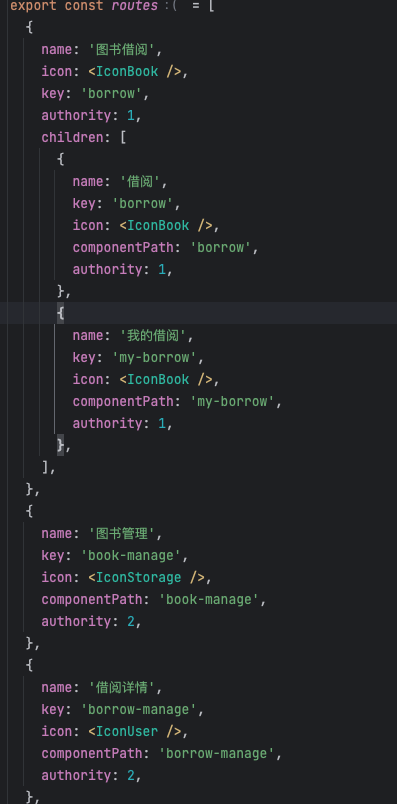


图 18 前端路由权限映射

**七、系统测试**

**1. xxx**

1.1 …

1.2 …

**2. xxx**

测试的具体要求：（1）针对需求说明中的每项功能，编写测试用例。（2）针对最核心的功能完成白盒测试和黑盒测试，其它功能只需要完成黑盒测试。（3） 每项测试必须采用等价划分、边界值分析等多种技术。（4）写出对每个测试结果的分析，以及对系统的改进意见。

**八、总结与改进**

**1. xxx**

**2. xxx**

总结：使用什么技术和方法，设计和实现了一个什么系统。

改进：从哪些方面，系统可以进一步改进

**九、参考文献**

**格式要求说明：**

1、所有**中文字体均采用楷体**，**英文字体采用Times New Roman**。

2、一级标题为四号加粗（一级标题的内容根据研讨内容的需求来设置，但最后三个标题不允许变动），二级、三级标题以及正文字体大小为小四号。

3、**表格字体与正文字体一致，在正文中居中，每个表均有标题，以表1，表2，…来命名，且表的标题在表格上方。此外每个表在正文中均要有引用，比如“如表1所示”。**

4、**图片或截图要清晰，在正文中居中，每个图均有标题，以图1，图2，….来命名，且图的标题在图片的下方。此外，每个图在正文中均要有引用，比如“图2展示了程序的运行结果”。**

5、**参考文献格式不限，但要保持统一**，即会议引用保持一致、期刊引用保持一致。

6、打印要求：**小册子双面打印，中缝装订**（打印店的工作人员会知道）。如果内容页数太多（小册子打印不下），可选择胶装。