



**阅读管理系统**

**设计报告**

**学 号 2162810203**

**姓 名 杨婧蕙**

**指导教师 吕嘉**

# 一. 需求分析和功能模块划分

## 1.1 功能模块划分

### 1.1.1功能模块图

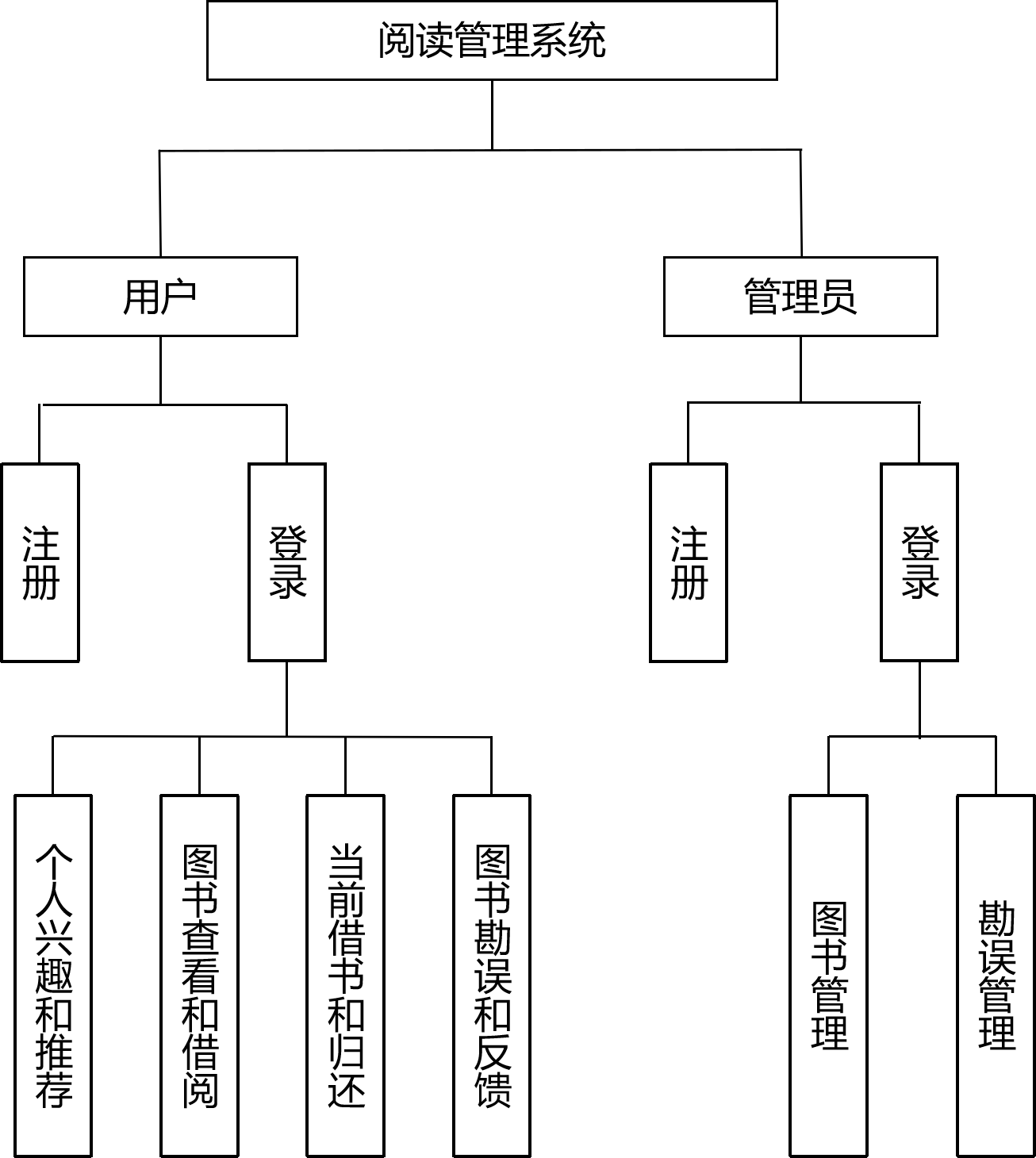


图1 功能模块图

### **1.1.2用户模块**

包括用户注册、用户登录模块。用户登录后可在个人兴趣和推荐模块查找系统推荐的图书并进行借阅；可在图书查看和借阅模块按书名和书号进行查找，借阅图书和查看借书情况；可在当前借书和归还模块查看当前借阅图书，并归还图书；也可在图书勘误和反馈模块提交勘误声明，查看勘误是否审核完成。

### **1.1.3管理员模块**

包括管理员注册和登录模块。登录后可以在图书管理模块对图书信息进行增删改查等操作，也可以在勘误管理模块对用户的勘误申请进行审核，并对勘误申请驳回或根据申请修改图书信息。

## **1.2 业务实体分析**

### **1.2.1 E-R图**

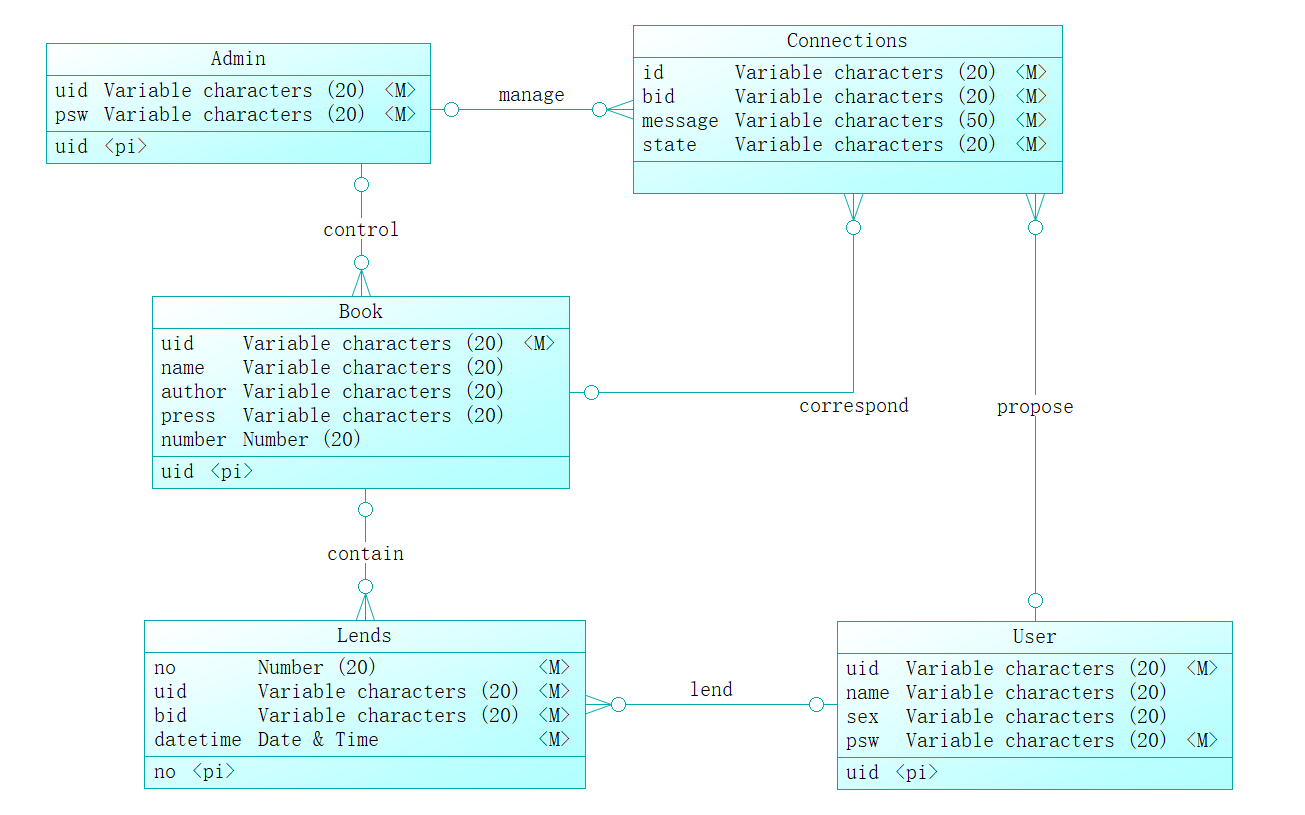


图2 ER图

### **1.2.2模块说明**

1. 实体（Entities）: 用户（User）、管理员（Admin）、图书（Book）、借阅记录（Lends）、勘误声明（Corrections）

（2）属性（Attributes）:

用户：用户ID（id，主键），用户名（name），性别（sex）密码（psw）。

管理员：管理员ID（id，主键），密码（psw）。

图书：书号（id，主键），书名（name），作者（author）、出版社（press）、库存（number）。

借阅记录：记录（no，主键），用户ID（uid，外键），书号（bid，外键），借阅时间（datetime）。

勘误声明：用户ID（id，外键），书号（bid，外键），勘误内容（message），审核状态（state）。

（3）关系（Relationships）:

用户和借阅记录：一对多关系，一个用户可以有多条借阅记录，但一条借阅记录只属于一个用户。

图书和借阅记录：一对多关系，一本图书可以有多条借阅记录，但一条借阅记录只对应一本图书。

用户和勘误申请：一对多关系，一个用户可以提交多个勘误声明，但一个勘误声明只属于一个用户。

图书和勘误申请：一对多关系，一本图书可以有多个勘误声明，但一个勘误声明只对应一本图书。

管理员和勘误申请：一对多关系，一个管理员可以管理多个勘误声明，但一个勘误声明只由一个管理员管理。

图书和管理员：一对多关系，一个管理员可以管理多本图书，但一本图书只由一个管理员管理。

## **1.3 业务过程分析**

### **1.3.1用例图**

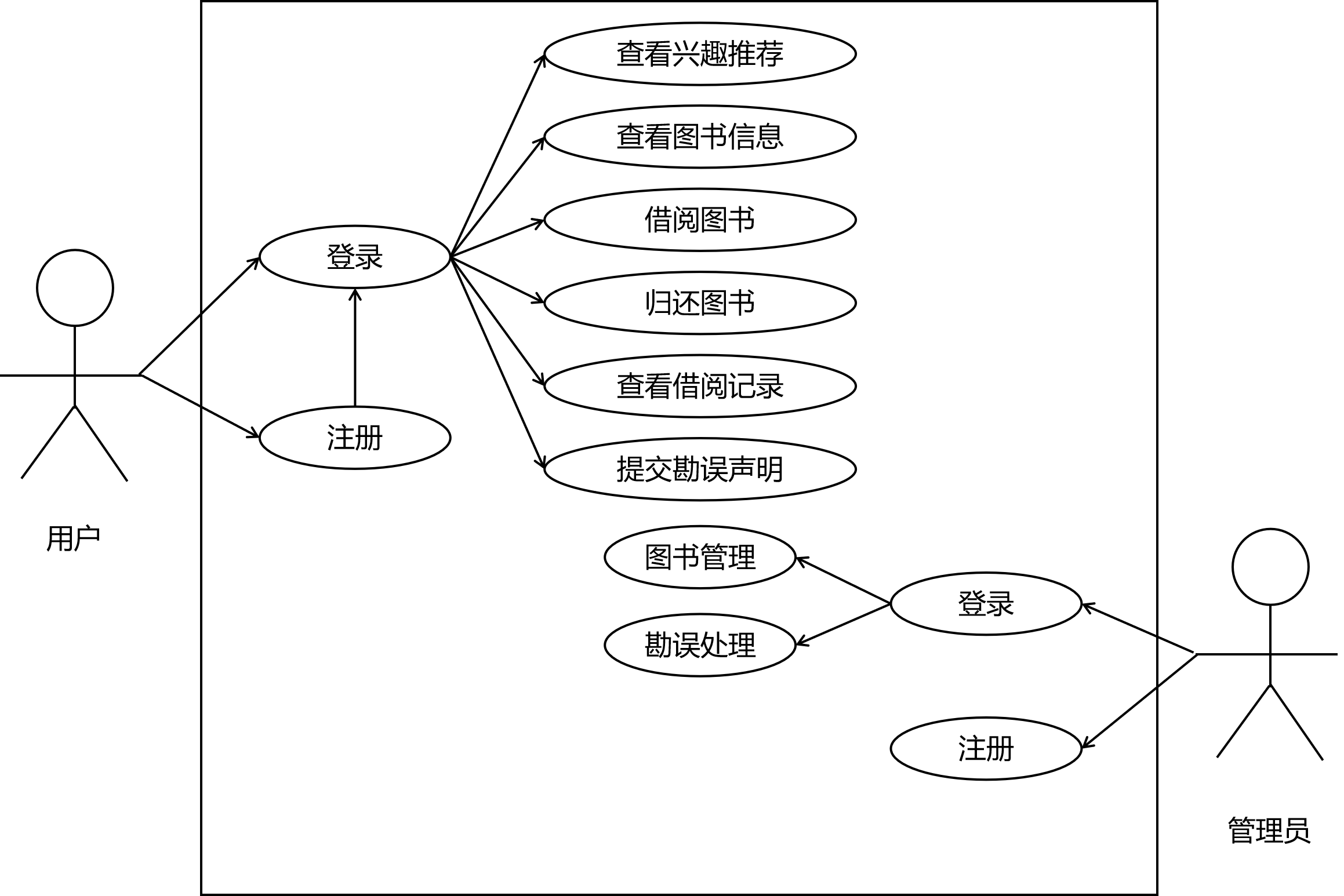


图3 用例图

### **1.3.2模块说明**

用户用例（Use Cases）:登录、注册、查看兴趣推荐、查看图书信息、借阅图书、查看借阅记录、提交勘误声明

管理员用例（Use Cases）:登录、注册、图书管理、勘误处理

# **二、系统分析与设计**

## **2.1软件体系结构设计**

### **2.1.1组件图**

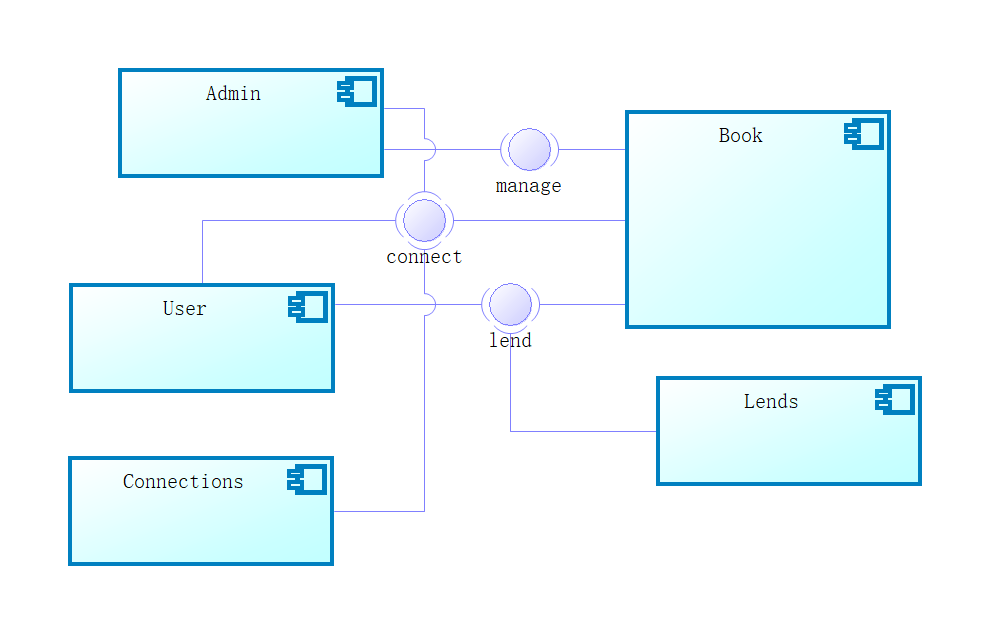


图4 组件图

### **2.1.2总体结构描述**

阅读管理系统采用三层架构，包括用户界面层(UI)，业务逻辑层(BLL)，数据访问层(DAL)。每一层都有着明确定义的责任，这有助于提高系统的可维护性、可扩展性和可重用性。

## **2.2功能模块整体设计**

### **2.2.1功能描述：**

系统具备用户和管理员两大功能模块。用户模块包括用户的注册、登录、兴趣推荐、借阅图书、提交勘误声明等功能。管理员模块包括管理员的登录、管理图书、处理用户勘误申请等功能。系统还提供了导航功能，使用户和管理员能够方便地浏览和使用各项功能。

### **2.2.2导航文字描述：**

用户通过登录后可以访问兴趣推荐，借阅图书，提交勘误声明等。管理员登录后可以管理图书信息，处理用户的勘误申请。导航栏包括个人兴趣和推荐、图书查看和借阅、当前借书和归还、图书勘误和反馈、图书管理、勘误处理等链接，以便用户和管理员轻松切换功能。

## **2.3数据库设计**

### **2.3.1关系模型：**

User(id, name, sex, psw)

Book(id, name, author, press, number)

Lends(no, uid, bid, datetime)

Corrections(id, bid, message, state)

Admin(id, psw)

### **2.3.2数据表和关系模型文字描述：**

User: 存储用户信息，主键为id，包括用户名(name)、性别(sex)和密码(psw)。

Book: 存储图书信息，主键为id，包括书名(name)、作者(author)、出版社(press)和库存数量(number)。

Lends: 存储借阅记录信息，主键为no，外键分别为uid和bid，记录借阅时间(datetime)。

Corrections: 存储勘误声明信息，主键为id，外键分别为uid和bid，包括勘误内容(message)和审核状态(state)。

Admin: 存储管理员信息，主键为id，包括密码(psw)。

## **2.4模型类设计**

### **2.4.1类图**

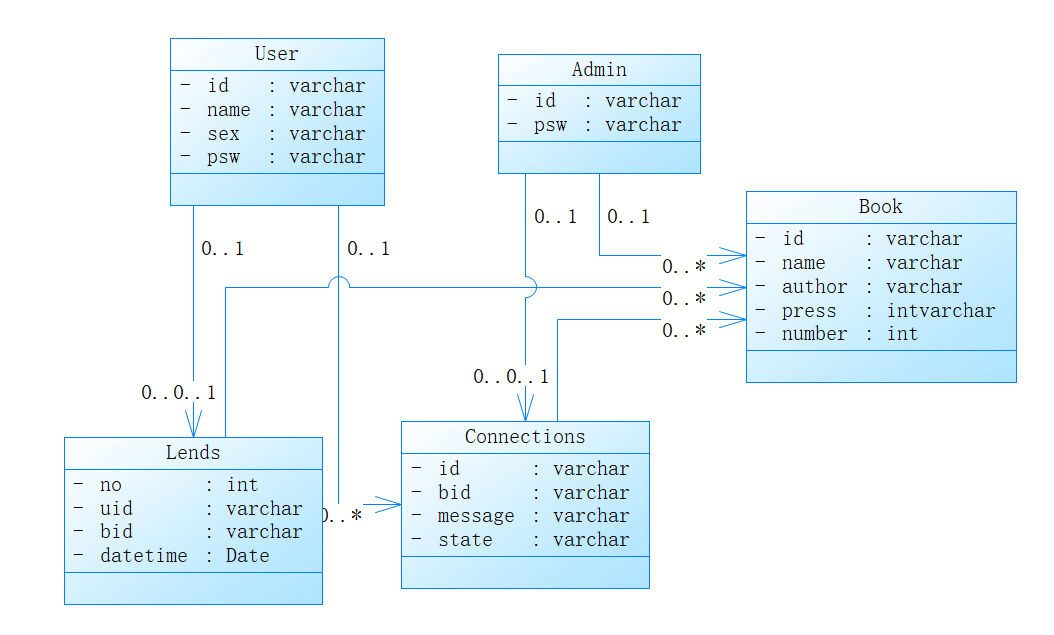


图5 类图

### **2.4.2类、属性和类之间关系文字描述：**

User类：

属性：id, name, sex, psw

关系：拥有多个Lends

Book类：

属性：id, name, author, press, number

关系：拥有多个Connections

Lends类：

属性：no, uid, bid, datetime

关系：关联到一个User和一个Book

Correction类：

属性：id, bid, message, state, uid

关系：关联到一个User和一个Book

Admin类：

属性：id, psw

关系：拥有多个Connections

## **2.5 业务过程设计**

### **2.5.1程序流程图**

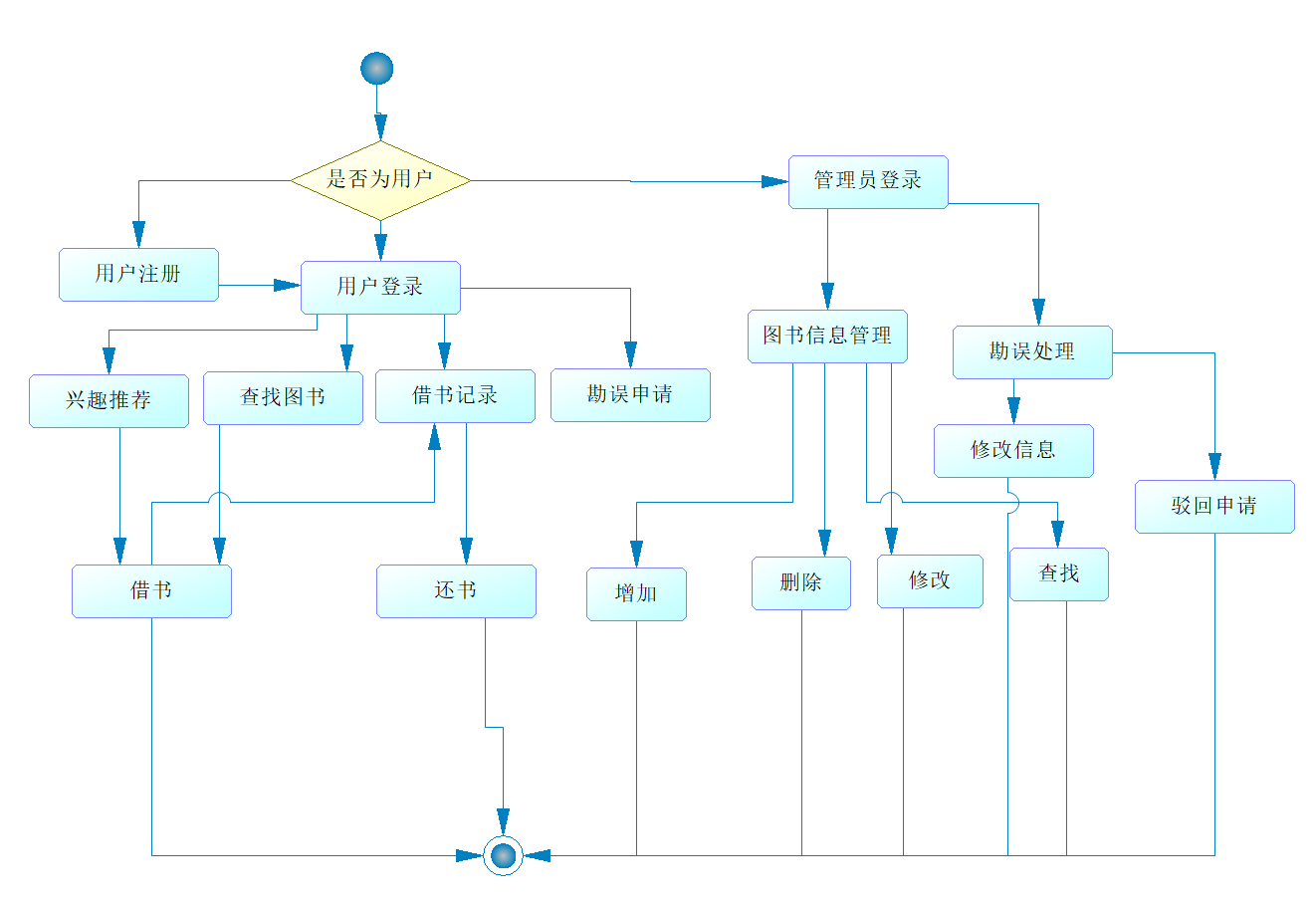


图6 程序流程图

### **2.5.2程序流程说明**

兴趣推荐流程：用户登录，进入个人兴趣和推荐模块，按感兴趣的关键字或词搜索感兴趣的图书，也可以借阅选中的图书。

借阅图书流程：用户登录，进入图书查看和借阅模块，（按书号和书名查询图书），选择图书，确认借阅，记录借阅信息。

查看借书记录流程：用户登录，进入当前借书和归还模块，查看借书记录。

还书流程：用户登录，进入当前借书和归还模块，选中准备归还的图书，确认还书。

提交勘误流程：用户登录，进入图书勘误和反馈模块，提交勘误信息，查看勘误审核情况。

图书信息管理流程：管理员登录，进入图书管理模块，（按书号和书名查询图书），选择图书，对该行图书信息做出增删改操作。

勘误处理流程：管理员登录，进入勘误处理模块，根据勘误申请对图书信息进行增删改查，或者驳回勘误申请。

# **三. 系统实现**

## **3.1 关键技术说明**

系统应用的关键技术包括：C#语言、ASP.NET Core框架、Entity Framework Core用于数据库操作、SQL Server 2008作为数据库管理系统。

## **3.2 关键功能设计**

### **3.2.1登录注册功能**

用户和管理员输入正确的用户名和密码，系统验证通过后允许登录。用户可以注册，保证性别正确，且没有信息缺漏即可注册。



图7 登录注册界面

### **3.2.2主界面**

显示用户登录和管理员登录，并提供导航链接到各功能模块，提升项目的实用性。

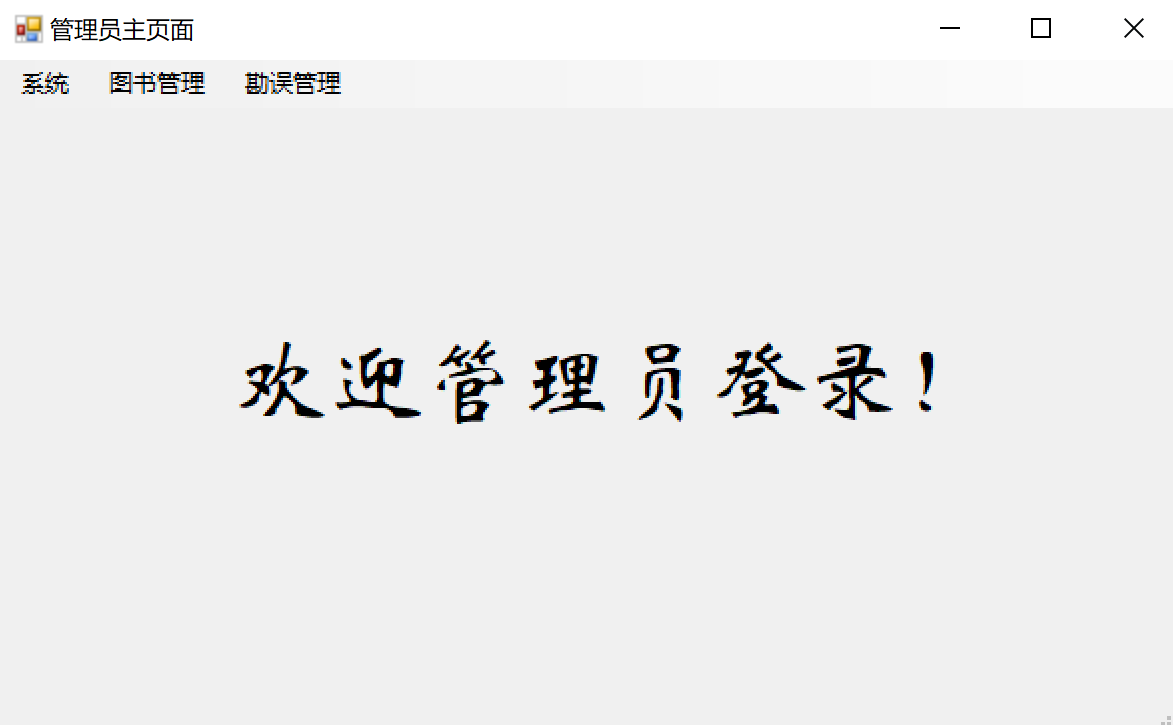


图8 用户端、管理端主界面

### **3.2.3 CRUD功能**

提供对图书信息、借阅记录以及用户提交的勘误申请的增、删、改、查等功能。



图9 用户和管理员的图书信息界面

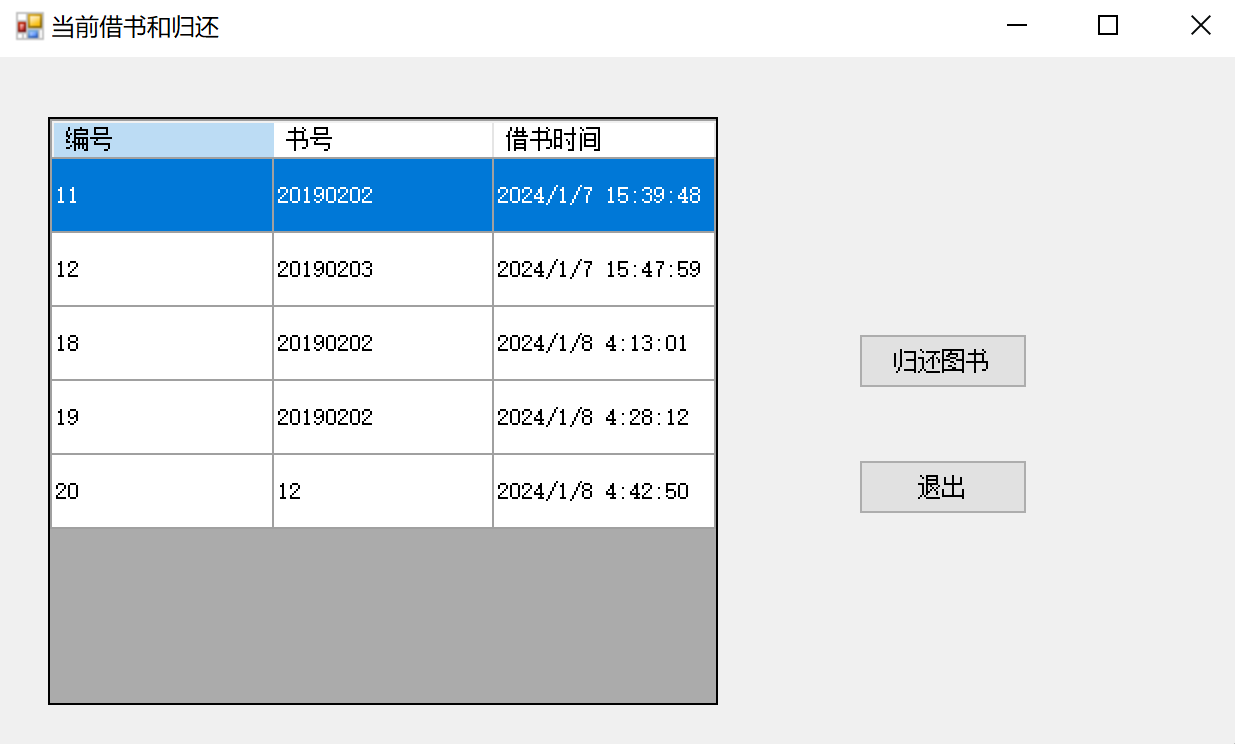


图10 借书还书界面 图11 勘误管理界面

## **3.3关键业务类或者业务过程实现说明**

### **3.3.1用户兴趣类**

searchBooksByKeyword(keyword: string): List<Book>根据关键字搜索感兴趣的图书。

borrowBook(book: Book): BorrowInfo 借阅选中的图书。

### **3.3.2图书借阅类**

searchBooksByNumberAndTitle(number: string, title: string): List<Book>按书号和书名查询图书。

borrowBook(user: User, book: Book): BorrowInfo确认借阅，记录借阅信息。

### **3.3.3用户记录类**

viewBorrowRecords(user: User): List<BorrowInfo>查看借书记录。

### **3.3.4图书归还类**

returnBook(user: User, book: Book): ReturnInfo确认还书。

### **3.3.5图书勘误类**

submitCorrection(user: User, correctionInfo: CorrectionInfo): CorrectionStatus提交勘误信息。

viewCorrectionStatus(user: User): CorrectionStatus查看勘误审核情况。

### **3.3.6管理员图书管理类**

searchBooksByNumberAndTitle(number: string, title: string): List<Book>按书号和书名查询图书。

manageBookInfo(book: Book, operation: OperationType): Book对图书信息进行增删改操作。

### **3.3.7管理员勘误处理类**

processCorrection(correctionInfo: CorrectionInfo, action: CorrectionAction): CorrectionStatus根据勘误申请进行增删改查操作，或者驳回勘误申请。

这些方法提供了系统中关键的业务逻辑，实现了用户兴趣、图书借阅、用户记录、图书归还、图书勘误、管理员图书管理以及管理员勘误处理等功能。

# **四、 系统实施和部署**

## **4.1 数据库部署**

部署服务器: 在服务器上安装SQL Server 2008，并创建数据库。

生成数据库: 使用Entity Framework Code First方式生成数据库表结构。

初始化数据: 插入初始用户、图书信息。

## **4.2 其他文件部署及应用程序安装**

（1）其他文件部署：

将其他文件如配置文件、资源文件或第三方库进行部署，并将其打包成安装包。确保这些文件能够被正确地引用和访问。

（2）应用程序安装：

使用 Visual Studio 2019 可以创建安装程序 (Installer) 或者打包部署项目。可以使用 ClickOnce 部署技术创建简单的应用程序安装程序。