

# 1 需求分析

## 1.1 项目背景

**目标用户：**图书馆的注册用户，希望通过在线图书室管理系统借还图书。

**项目目标：**提高用户体验，通过推荐用户可能感兴趣的图书，增加借阅频率和用户满意度。

## 1.2 功能需求

### 1.2.1 用户借书记录查询

系统需要记录和查询用户的借书历史。

借书记录应包括用户ID、图书ID、借书日期和还书日期等信息。

### 1.2.2 图书信息管理

系统需要管理图书信息，包括图书ID、书名、作者、出版社、出版日期等。

图书信息应便于检索和展示。

### 1.2.3 推荐模块（后期会完善）

在主页面上显示一个推荐模块，根据用户的借书记录推荐图书。

推荐算法应考虑到用户的历史借书偏好。

推荐结果应包括图书名称、作者和封面图片等信息。

### 1.2.4 用户界面

主页面应清晰、美观，易于导航。

推荐模块应位于显眼位置，方便用户查看。

## 1.3 非功能需求

#### 性能：

系统应能快速响应用户请求，推荐结果应在几秒钟内显示。

数据库查询应高效，能够处理大量用户和图书数据。

#### 安全性：

用户数据应加密存储，保护用户隐私。

系统应防止SQL注入等安全攻击。

#### 可扩展性：

系统应设计为可扩展，能够新增更多推荐算法和功能模块。

数据库结构应灵活，方便添加新的数据字段。

#### 易用性：

界面应简洁明了，用户操作直观。

提供帮助文档和用户指南，方便用户使用系统。

## 2 系统功能分析

### 2.1 系统功能概述

在线图书室管理系统旨在为用户提供便捷的图书借阅、查询和管理功能，同时为管理员提供高效的图书管理工具。系统应具备以下核心功能：

用户功能：

功能模块	功能描述	详细说明
图书搜索	用户可以通过书名、作者、类别等条件搜索图书。	用户可以在搜索框中输入关键词进行搜索，系统返回相关图书列表。
借阅	用户可以选择图书进行借阅。	用户点击借阅按钮后，系统检查图书库存并记录借阅信息，更新图书状态为“已借”。
预约	用户可以预约当前不可借的图书，并在图书可借时收到通知。	当图书库存为0时，用户可以预约图书，系统记录预约信息并在图书可借时通过邮件或短信通知用户。
图书推荐	系统根据用户的借阅历史推荐相关图书。	系统通过分析用户的借阅记录，推荐用户可能感兴趣的图书，推荐的图书显示在用户主页。

管理员功能：

功能模块	功能描述	详细说明
图书管理	管理员可以添加、修改、删除图书信息。	添加新图书：管理员输入书名、作者、类别、ISBN、库存等信息。修改图书信息：管理员可以修改图书的库存、状态等信息。删除图书：管理员可以下架不再提供的图书。
用户管理	管理员可以查看和管理用户列表，处理用户的借阅请求。	查看用户列表：管理员可以查看所有用户的注册信息和权限。管理用户权限：管理员可以添加、删除用户，修改用户权限。处理借阅请求：管理员可以处理用户的借阅和预约请求。
借阅记录管理	管理员可以查看所有用户的借阅记录，处理逾期图书。	查看借阅记录：管理员可以查看所有用户的借阅记录，包括借阅日期、归还日期等。处理逾期图书：管理员可以发送催还通知，处理逾期图书。处理归还和续借请求：管理员可以处理用户的归还和续借请求。
归还	管理员可以处理图书归还和续借请求。	归还图书：用户归还图书后，管理员确认归还并更新图书状态。续借图书：用户申请续借时，管理员可以批准或拒绝续借请求。

- 用户功能主要集中在图书的搜索、借阅、预约和推荐，旨在为用户提供便捷的图书借阅体验。
- 管理员功能则集中在图书和用户管理，以及借阅记录的处理，旨在确保系统的正常运行和图书资源的高效管理。

## 2.2 根据不同用户的需求

同样，我们根据不同用户的需求，还涉及了不同的功能，具体如下所示：

### 普通用户功能

功能模块	功能描述	详细说明
便捷的图书搜索和借阅功能	用户可以通过书名、作者、类别等条件搜索图书。	用户可以在搜索框中输入关键词进行搜索，系统返回相关图书列表。
	用户可以选择图书进行借阅或预约。	用户点击借阅按钮后，系统检查图书库存并记录借阅信息；用户可以预约当前不可借的图书。
查看借阅记录、预约图书及获取图书推荐	用户可以查看自己的借阅记录和归还状态。	用户可以在个人中心查看自己的借阅历史，包括借阅日期、归还日期等信息。
	用户可以预约图书，并在图书可借时收到通知。	当图书库存为0时，用户可以预约图书，系统记录预约信息并在图书可借时通过邮件或短信通知用户。
	系统根据用户的借阅历史推荐相关图书。	系统通过分析用户的借阅记录，推荐用户可能感兴趣的图书，推荐的图书显示在用户主页。

### 管理员功能

功能模块	功能描述	详细说明
高效的图书管理和用户管理功能	管理员可以添加、修改、删除图书信息。	添加新图书：管理员输入书名、作者、类别、ISBN、库存等信息。修改图书信息：管理员可以修改图书的库存、状态等信息。删除图书：管理员可以下架不再提供的图书。
	管理员可以查看和管理用户列表，处理用户的借阅请求。	查看用户列表：管理员可以查看所有用户的注册信息和权限。管理用户权限：管理员可以添加、删除用户，修改用户权限。处理借阅请求：管理员可以处理用户的借阅和预约请求。
查看和管理借阅记录、处理逾期图书	管理员可以查看所有用户的借阅记录，处理逾期图书。	查看借阅记录：管理员可以查看所有用户的借阅记录，包括借阅日期、归还日期等。处理逾期图书：管理员可以发送催还通知，处理逾期图书。
	管理员可以处理图书归还和续借请求。	归还图书：用户归还图书后，管理员确认归还并更新图书状态。续借图书：用户申请续借时，管理员可以批准或拒绝续借请求。

普通用户功能主要集中在图书的搜索、借阅、预约、借阅记录查看和图书推荐，旨在为用户提供便捷的图书借阅体验。

管理员功能则集中在图书和用户管理，以及借阅记录的处理，旨在确保系统的正常运行和图书资源的高效管理。

## 3 系统模块构建

### 3.1 用户使用流程

用户注册/登录：



用户通过注册/登录进入系统。

用户输入用户名和密码进行登录，支持新用户注册。

图书搜索：



用户可以根据书名、作者、类别等条件搜索图书。



搜索结果应包含图书的详细信息，如书名、作者、出版社等。

图书借阅/预约：



用户可以选择图书进行借阅或预约。

借阅时，系统应检查图书的库存状态，并记录借阅信息。

预约时，系统应记录预约信息，并在图书可借时通知用户。

借阅记录管理：



用户可以查看自己的借阅记录和归还状态。

系统应提供归还功能，用户可以在线确认归还图书。

## 3.2 管理员功能模块

管理员注册：



### 3.2.1 图书管理

添加新图书：



输入书名、作者、类别、库存等信息。

修改图书信息：



修改图书的库存、状态等信息。

删除图书：



下架不再提供的图书。

### 3.2.2 用户管理

查看用户列表：



管理员可以查看所有用户的注册信息和权限。

管理用户权限：



管理员可以添加、删除用户，修改用户权限。

### 3.2.3 借阅记录管理

查看所有用户的借阅记录：



管理员可以查看所有用户的借阅记录，包括借阅日期、归还日期等。

## 4 设计要求

### 4.1 局部概念设计

#### 图书模块：

属性：

书名 (book\_name)

作者 (author)

ISBN

库存 (Stock)

状态 (Status: 可借/已借/预约)

#### 用户模块：

属性：

用户ID (UserID)

用户名 (Username)

密码 (Password)

邮箱 (Email)

借阅记录 (BorrowRecords)

预约记录 (Reservations)

#### 借阅模块：

属性：

借阅ID (BorrowID)

用户ID (UserID)

图书ID (BookID)



借阅日期 (BorrowDate)

归还日期 (ReturnDate)

## 4.2 全局概念结构设计

---

用户表 (users) :

字段:

UserID (主键)

Username

Password

Email

图书表 (Books) :

字段:

BookID (主键)

Title

Author

Category

ISBN

Stock

Status

借阅表 (borrows) :

字段:

BorrowID (主键)

UserID (外键)

BookID (外键)

BorrowDate

ReturnDate

管理员信息表 (manager\_info) :

字段:

AdminID (主键)

Username

Password

## 4.3 数据库的物理设计

---

数据库名称: book

表结构:

**图书表 (Books) :**

BookID (主键)

Title

Author

Category

ISBN

Stock

Status

**用户表 (Users) :**

UserID (主键)

Username

Password

Email

**借阅表 (BorrowRecords) :**

BorrowID (主键)

UserID (外键)

BookID (外键)

BorrowDate

ReturnDate

**管理员表 (Admins) :**

AdminID (主键)

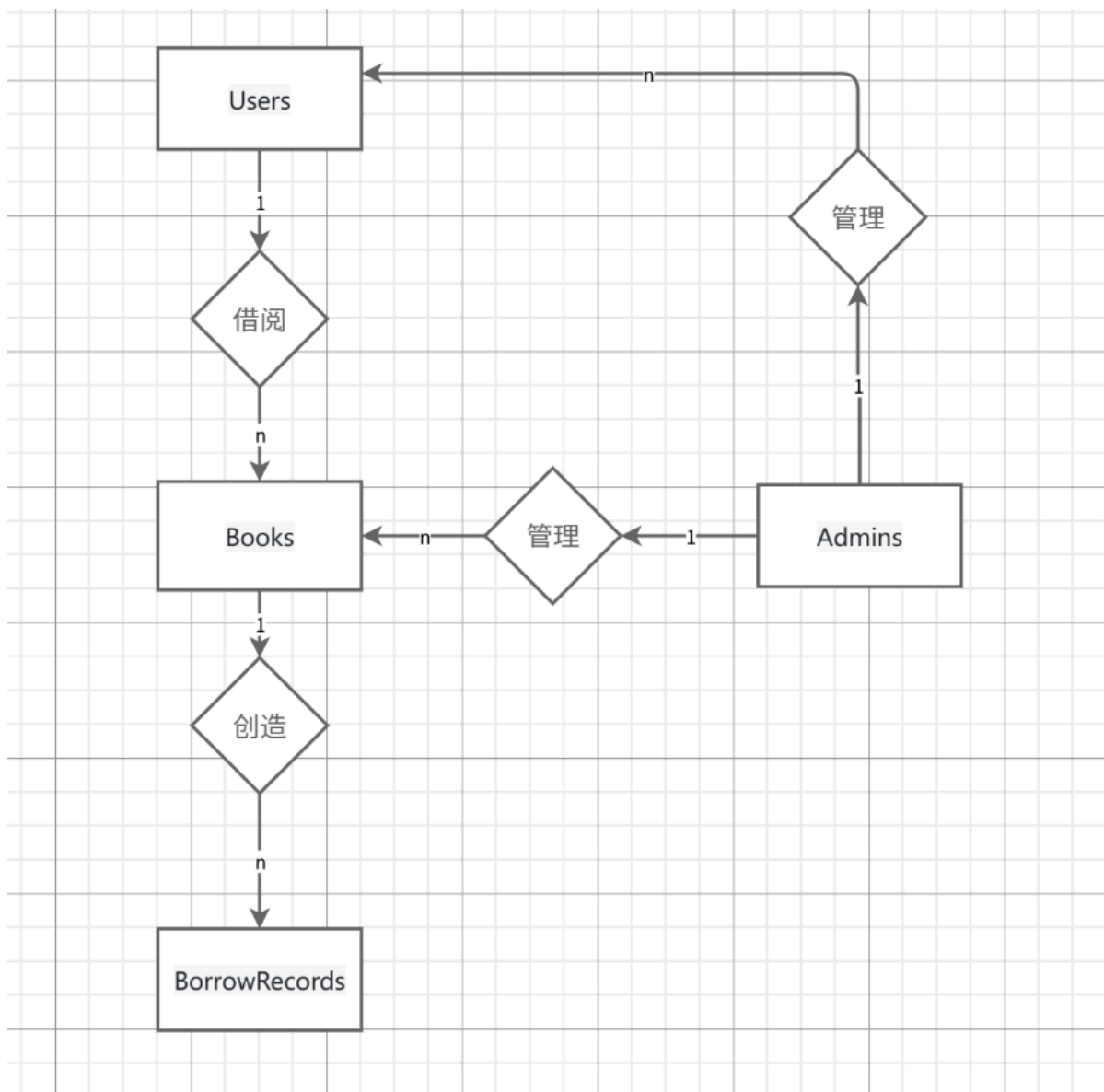
Username

Password

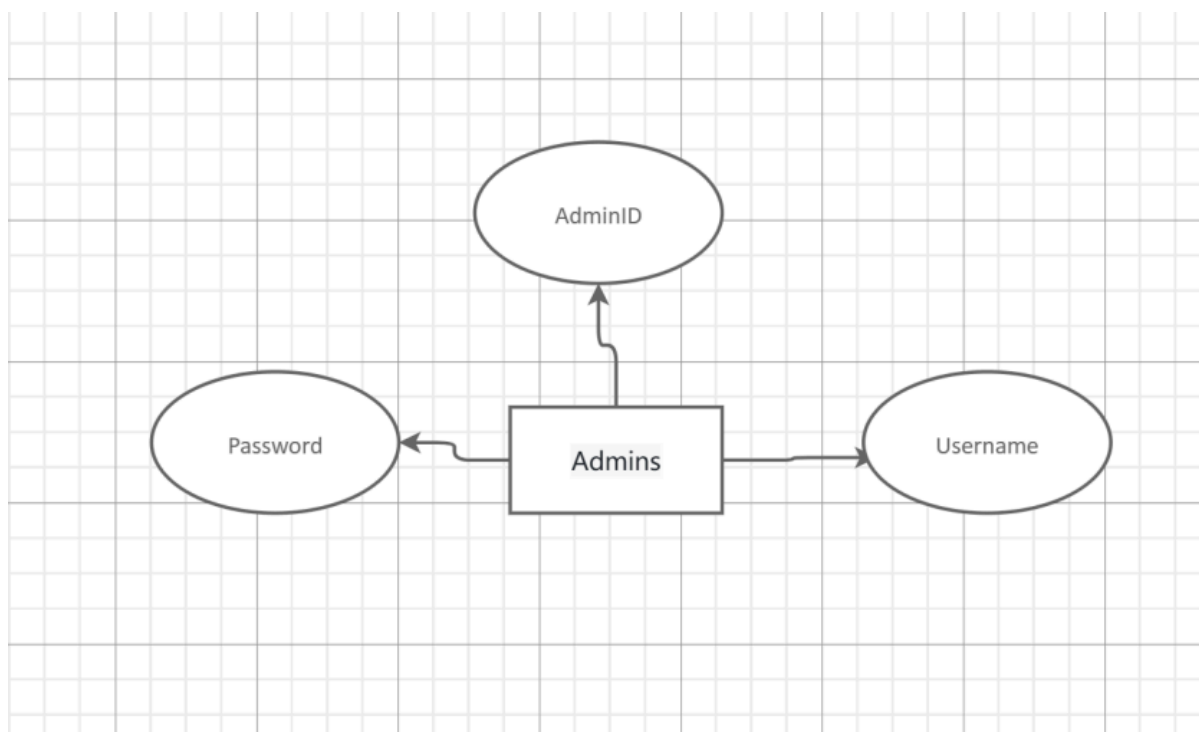
## 4.4 ER图的绘制

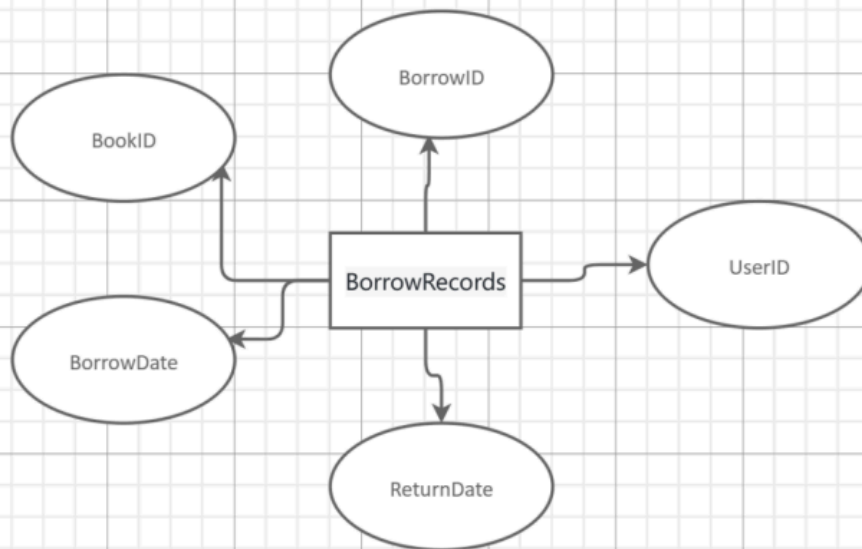
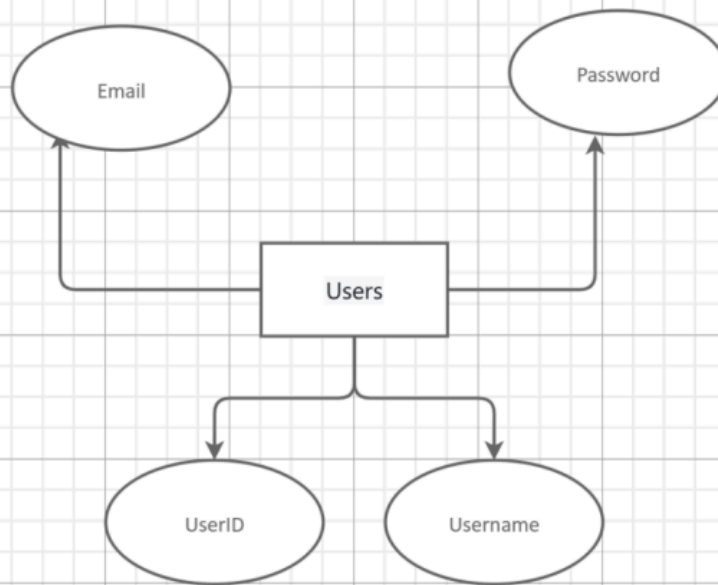
---

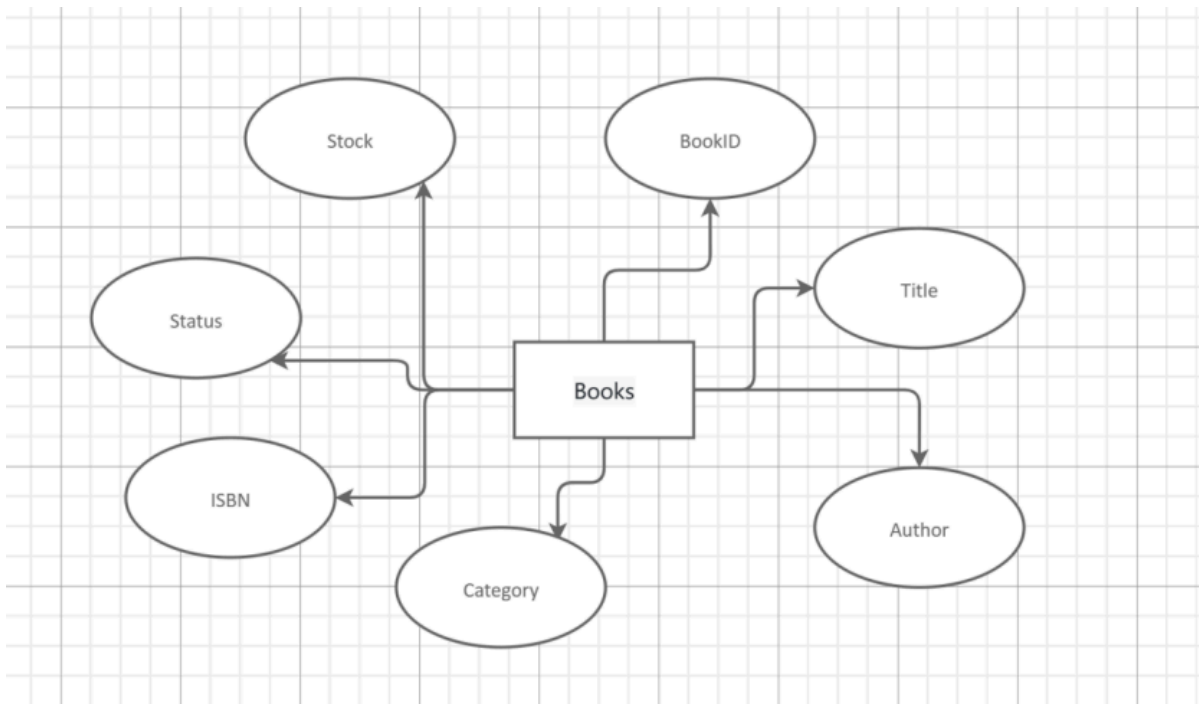
### 4.4.1 全局ER图



#### 4.4.2 局部ER图







### 4.4.3 关系模式转换与范式理论的论述

在ER图设计完成后，将其转换为关系模式时，建议结合范式理论进行优化，以确保数据库设计的合理性和高效性。

#### 4.4.3.1 关系模式转换

- 用户表 (Users) :

```
CREATE TABLE Users (  
    UserID INT PRIMARY KEY,  
    Username VARCHAR(50) NOT NULL UNIQUE,  
    Password VARCHAR(255) NOT NULL,  
    Email VARCHAR(100) NOT NULL UNIQUE  
);
```

- 图书表 (Books) :

```
CREATE TABLE Books (  
    BookID INT PRIMARY KEY,  
    Title VARCHAR(255) NOT NULL,  
    Author VARCHAR(100) NOT NULL,  
    Category VARCHAR(50),  
    ISBN VARCHAR(20) NOT NULL UNIQUE,  
    Stock INT NOT NULL CHECK (Stock >= 0),  
    Status ENUM('可借', '已借', '预约') NOT NULL  
);
```

- 借阅表 (BorrowRecords) :

```
CREATE TABLE BorrowRecords (  
    BorrowID INT PRIMARY KEY,  
    UserID INT,  
    BookID INT,  
    BorrowDate DATE NOT NULL,  
    ReturnDate DATE,  
    FOREIGN KEY (UserID) REFERENCES Users(UserID),  
    FOREIGN KEY (BookID) REFERENCES Books(BookID)  
);
```

- **预约表 (Reservations) :**

```
CREATE TABLE Reservations (  
    ReservationID INT PRIMARY KEY,  
    UserID INT,  
    BookID INT,  
    ReservationDate DATE NOT NULL,  
    FOREIGN KEY (UserID) REFERENCES Users(UserID),  
    FOREIGN KEY (BookID) REFERENCES Books(BookID)  
);
```

- **管理员表 (Admins) :**

```
CREATE TABLE Admins (  
    AdminID INT PRIMARY KEY,  
    Username VARCHAR(50) NOT NULL UNIQUE,  
    Password VARCHAR(255) NOT NULL  
);
```

---

#### 4.4.3.2 范式理论的论述

在关系模式设计中，应用范式理论可以避免数据冗余、提高数据一致性和查询效率。

##### 第一范式 (1NF)

- 确保每个属性都是原子的，不可再分。
  - 例如，用户表中的 `Email` 字段不能存储多个邮箱地址。

##### 第二范式 (2NF)

- 满足1NF，并且每个非主键属性完全依赖于主键。
  - 例如，借阅表中的 `BorrowDate` 和 `ReturnDate` 完全依赖于 `BorrowID`，而不是部分依赖。

第三范式 (3NF)

- 满足2NF，并且非主键属性之间不存在传递依赖。
  - 例如，图书表中的 `Category` 不应依赖于 `Author`，而是直接依赖于 `BookID`。

第四范式 (4NF)

- 满足3NF，并且消除多值依赖。
  - 例如，用户表中的 `BorrowRecords` 和 `Reservations` 不应直接存储在用户表中，而是通过外键关联到单独的表。

5 界面设计

5.1 用户登录页面

下面是登录功能页面元素和功能说明表：

页面元素	描述	行为/功能
用户名输入框	用户输入其用户名。文本输入框类型。支持输入字母、数字和特殊字符。必填项。	用于输入用户名，系统通过此字段验证用户身份。前端验证：非空，长度范围（如：620字符）。
密码输入框	用户输入其密码。密码输入框类型（隐藏输入内容）。支持输入字母、数字和特殊字符。必填项。	用于输入密码，系统通过此字段验证用户身份。前端验证：非空，密码长度范围（如：816字符）。
登录按钮	用户点击登录系统。按钮样式：突出显示，通常为“登录”文本。支持点击事件。	提交用户名和密码到后端进行验证。验证成功：跳转至系统主界面。验证失败：显示错误信息（如：“用户名或密码错误”）。
注册按钮	允许用户跳转至注册页面。按钮文本：“注册”。支持点击事件。	点击后跳转至注册页面。注册页面包含用户名、密码、邮箱等注册信息输入框和提交按钮。

5.2 管理员页面

下是管理员管理功能的页面元素和功能说明，以表格形式整理如下：

1.管理员管理功能页面元素和功能说明表

模块	功能	页面元素	描述
导航栏	图书管理 用户管理 借阅记录管理	导航菜单项：图书管理、用户管理、借阅记录管理。	管理员通过导航栏快速切换到不同管理模块。

2.图书管理模块

功能	页面元素	描述
添加新图书	书名输入框 作者输入框 类别选择框 库存数量输入框 添加按钮	管理员输入书名、作者、类别、库存等新图书信息，点击“添加”按钮保存图书信息到系统中。
修改图书信息	图书列表（包含书名、作者、类别、库存等信息） 修改按钮 库存输入框 状态选择框	管理员选择图书后，可以修改库存或状态（如：“在库”、“借出”、“下架”）。点击“保存”按钮更新。
删除图书	图书列表（包含书名、作者等信息） 删除按钮	管理员选择图书后，点击“删除”按钮将图书标记为下架，不再提供给用户借阅。

### 3.用户管理模块

功能	页面元素	描述
查看用户列表	用户列表（包含用户名、注册时间、权限等信息）	管理员可以查看所有用户的注册信息、权限状态。
管理用户权限	用户列表 添加用户按钮 删除用户按钮 权限选择框（如：普通用户、管理员）	管理员可以添加新用户、删除现有用户，或修改用户的权限（如提升为管理员或降级为普通用户）。

### 4.借阅记录管理模块

功能	页面元素	描述
查看所有借阅记录	借阅记录列表（包含用户名、书名、借阅日期、归还日期等信息）	管理员可以查看所有用户的借阅记录，了解图书的借阅情况。
处理归还和续借请求	归还请求列表 续借请求列表 确认归还按钮 批准续借按钮	管理员可以处理用户的归还请求，点击“确认归还”更新图书状态；或批准续借请求延长借阅时间。

## 6 相关SQL代码

#### 1. 从 `manager_info` 表中查询管理员信息：

```
SELECT * FROM manager_info WHERE manager_id = ?
```

#### 2. 从 `users` 表中查询用户信息：

```
SELECT * FROM users WHERE user_name = ?
```

#### 3. 从 `users` 表中查询用户的 `user_id`：



```
SELECT user_id FROM users WHERE user_name = ?
```

4. 向 `users` 表中插入新用户数据:

```
INSERT INTO users (user_name, pwd, college, num, email) VALUES (?, ?, ?, ?, ?)
```

5. 查询所有书籍:

```
SELECT * FROM books
```

6. 查询所有用户:

```
SELECT * FROM users
```

7. 查询特定用户:

```
SELECT * FROM users WHERE user_id = ?
```

8. 更新用户信息 (不包含密码) :

```
UPDATE users SET user_name = ?, college = ?, num = ?, email = ? WHERE user_id = ?
```

9. 更新用户信息 (包含密码) :

```
UPDATE users SET user_name = ?, pwd = ?, college = ?, num = ?, email = ? WHERE user_id = ?
```

10. 删除用户:

```
DELETE FROM users WHERE user_id = ?
```

11. 插入管理员信息:

```
INSERT INTO manager_info (manager_id, password) VALUES (?, ?)
```

12. 插入新书籍:

```
INSERT INTO books (book_id, book_name, author, publish_com, publish_date)
VALUES (?, ?, ?, ?, ?)
```

13. 删除书籍:

- 检查书籍是否存在:

```
SELECT * FROM books WHERE book_id = ?
```

- 删除指定书籍:

```
DELETE FROM books WHERE book_id = ?
```

14. 查看书籍详情:

```
SELECT * FROM books WHERE book_id = ?
```

15. 获取借阅该书籍的读者信息:

```
SELECT * FROM borrows WHERE book_id = ?
```

16. 获取借阅该书籍的读者姓名:

```
SELECT user_name FROM borrows WHERE book_id = ?
```

17. 更新历史记录中的书籍状态和归还日期:

```
UPDATE histroys SET status = ?, date_return = ? WHERE book_id = ? AND
user_name = ? AND status = ?
```

18. 删除借阅记录:

```
DELETE FROM borrows WHERE book_id = ?
```

19. 查看用户详情:

```
SELECT * FROM users WHERE user_id = ?
```

20. 修改书籍信息：

- 获取书籍当前信息：

```
SELECT * FROM books WHERE book_id = ?
```

- 更新书籍信息：

```
UPDATE books SET book_name = ?, author = ?, publish_com = ?,  
publish_date = ? WHERE book_id = ?
```

21. 查看读者信息：

```
SELECT * FROM users WHERE user_name = ?
```

22. 修改读者信息（不包含密码）：

```
UPDATE users SET user_name = ?, college = ?, num = ?, email = ? WHERE user_id  
= ?
```

23. 修改读者信息（包含密码）：

```
UPDATE users SET user_name = ?, pwd = ?, college = ?, num = ?, email = ?  
WHERE user_id = ?
```

24. 读者查询书籍（按书名）：

```
SELECT * FROM books WHERE book_name LIKE ?
```

25. 读者查询书籍（按作者）：

```
SELECT * FROM books WHERE author LIKE ?
```