

Lab5 :图书管理系统

3220105929 秦际州

实验目的

- 设计并实现一个精简的图书管理程序，要求具有图书入库、查询、借书、还书、借书证管理等功能。

实验需求

• 1 后端实验需求

- 图书管理系统中所有应具备的功能模块都在接口LibraryManagementSystem中被声明：
 - 在实现交互界面之前，你首先需要为该接口做一个具体的实现(在实现类中重写接口中的方法)，具体实现位于类LibraryManagementSystemImpl中。
 - 在完成所有的接口后，你可以在主类Main中通过LibraryManagementSystem的一个实例来实现与用户的交互。

• 2 前端实验需求

- 你需要根据刚刚在基础部分实现的接口，使用BS体系完成一个精简的图书管理系统。
 - 后端使用Java实现，接收来自前端的请求，调用你所实现的接口实现处理逻辑，并给出响应。
 - 前端使用Vue3实现，与用户进行交互，向后端发出请求并处理响应。

• 3 系统功能验证

- 系统功能验证测试分为功能性测试和正确性测试。
 - 正确性测试通过测试用例进行评判，以验收时通过的测试用例数量占总测试用例数量的百分比来评定正确性测试部分的得分。
 - 当实现了前端，图书管理系统表现为一个完整可使用的程序时，进行功能性测试。功能性测试通过验收时随机运行模拟场景的结果进行评判，以软件使用时的交互友好程度、效率、正确性等指标来评定功能性测试部分的得分。

实验环境

- 电脑中已经安装了独立的JDK和Maven：
 - JDK 1.8.0及以上
 - Apache Maven 3.6.3及以上
- ```
PS C:\c\learn\DBS\lab5> java -version
java version "21.0.1" 2023-10-17 LTS
Java(TM) SE Runtime Environment (build 21.0.1+12-LTS-29)
Java HotSpot(TM) 64-Bit Server VM (build 21.0.1+12-LTS-29, mixed mode, sharing)
PS C:\c\learn\DBS\lab5> mvn -v
Apache Maven 3.9.6 (bc0240f3c744dd6b6ec2920b3cd08dcc295161ae)
Maven home: D:\apache-maven-3.9.6-bin\apache-maven-3.9.6
Java version: 21.0.1, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-21
Default locale: zh_CN, platform encoding: UTF-8
OS name: "windows 11", version: "10.0", arch: "amd64", family: "windows"
```
- 使用 VSCode + Java + Maven 作为开发环境

## 后端数据库系统实现

主要实现都是在 `LibraryMangementSystemImpl.java` 文件中实现与数据库的交互，主要实现如下：

## • 1 图书入库模块

- `ApiResult storeBook(Book book)`
- **要求：**向图书库中注册(添加)一本新书，并返回新书的书号。如果该书已经存在于图书库中，那么入库操作将失败。当且仅当书的<类别, 书名, 出版社, 年份, 作者>均相同时，才认为两本书相同。请注意，`book_id` 作为自增列，应该插入时由数据库生成。插入完成后，需要根据数据库生成的 `book_id` 值去更新 `book` 对象里的 `book_id`。
- **实现思路：**直接使用 `INSERT` 语句实现，由于判断是否相同会在插入语句中自行判断于是不需要过多判断，在使用插入语句时记录键值 `Statement.RETURN_GENERATED_KEYS` ,在插入结束时读取键值存在 `bookid` 中。

## • 2 图书增加库存模块

- `ApiResult incBookStock(int bookId, int deltaStock)`
- **要求：**为图书库中的某一本书增加库存。其中库存增量`deltaStock`可正可负，若为负数，则需要保证最终库存是一个非负数。
- **实现思路：**提前先使用 `SELECT` ,语句判断是都存在对应的书籍，并保证调整库存之后库存是非负数，之后使用 `UPDATE` 语句实现，判断库存是否为非负数，若为非负数则执行 `UPDATE` 语句，否则返回错误信息。

## • 3 图书批量入库模块

- `ApiResult storeBook(List<Book> books)`
- **要求：**批量入库图书，如果有一本书入库失败，那么就需要回滚整个事务(即所有的书都不能被入库)。
- **实现思路：**遍历传入的书籍列表，将书籍的分类、标题、出版社、出版年份、作者、价格和库存数量设置到预处理语句中。最后，执行批量插入操作并获取生成的键值。将生成的键值设置到书籍对象中。

## • 4 图书删除模块

- `ApiResult removeBook(int bookId)`
- **要求：**从图书库中删除一本书。如果还有人尚未归还这本书，那么删除操作将失败。
- **实现思路：**先通过 `SELECT` 判断书籍是否存在，是否有人借阅图书，再使用 `DELETE` 语句实现执行删除操作，否则返回错误信息。

## • 5 图书修改模块

- `ApiResult modifyBookInfo(Book book)`
- **要求：**修改已入库图书的基本信息，该接口不能修改图书的书号和存量。
- **实现思路：**先使用 `SELECT` 判断实际是否存在，再使用 `UPDATE` 语句实现执行修改操作，否则返回错误信息，对于数据的判断可以通过SQL自动判断。

## • 6 图书查询模块

- `ApiResult queryBook(BookQueryConditions conditions)`
- **要求：**根据提供的查询条件查询符合条件的图书，并按照指定排序方式排序。查询条件包括：类别点查(精确查询)，书名点查(模糊查询)，出版社点查(模糊查询)，年份范围查，作者点查(模糊查询)，价格范围差。如果两条记录排序条件的值相等，则按`book_id`升序排序。
- **实现思路：**使用 `SELECT` 语句实现，执行查询操作，否则返回错误信息。困难：不知道查询时时使用了哪些条件，对于空的条件不能参与查询，于是遍历每一个条件判断是否为空，并拼装到查询字符串中。其次由于是模糊查询需要使用 `LIKE` 关键字，因此需要对查询字符串进行转义。最后由于要保证查询顺序，始终将 `book_id` 作为排序条件，并使用 `ORDER BY` 语句实现。

## • 7 借书模块

- `ApiResponse borrowBook(Borrow borrow)`
- **要求：**根据给定的书号、卡号和借书时间添加一条借书记录，然后更新库存。若用户此前已经借过这本书但尚未归还，那么借书操作将失败。
- **实现思路：**先使用 `SELECT` 判断书籍是否存在、借书证是否存在、是否有额外库存，再使用 `INSERT` 语句实现借书操作，否则返回错误信息，最后再更新库存。

## • 8 还书模块

- `ApiResponse returnBook(Borrow borrow)`
- **要求：**根据给定的书号、卡号和还书时间，查询对应的借书记录，并补充归还时间，然后更新库存。
- **实现思路：**先使用 `SELECT` 判断书籍是否存在、借书证是否存在、是否已经归还过，再使用 `UPDATE` 语句执行还书操作，并修改库存。

## • 9 借书记录查询模块

- `ApiResponse showBorrowHistory(int cardId)`
- **要求：**查询某个用户的借书记录，按照借书时间递减、书号递增的方式排序。
- **实现思路：**使用 `SELECT` 语句实现执行查询操作，否则返回错误信息。

## • 10 借书证注册模块

- `ApiResponse registerCard(Card card)`
- **要求：**注册一个借书证，若借书证已经存在，则该操作将失败。当且仅当<姓名, 单位, 身份>均相同时，才认为两张借书证相同。
- **实现思路：**使用 `INSERT` 语句实现，执行借书证注册操作，否则返回错误信息。

## • 11 借书证删除模块

- `ApiResponse removeCard(int cardId)`
- **要求：**删除借书证模块。如果该借书证还有未归还的图书，那么删除操作将失败。
- **实现思路：**先使用 `SELECT` 判断借书证是否存在，再使用 `DELETE` 语句实现，执行借书证删除操作，否则返回错误信息。

## • 12 借书证查询模块

- `ApiResponse showCards()`
- **要求：**列出所有的借书证。
- **实现思路：**使用 `SELECT` 语句实现，执行借书证查询操作，否则返回错误信息。

## • 13 借书证修改模块

- `ApiResponse modifyCard(Card card)`
- **要求：**修改已注册的借书证的基本信息，该接口不能修改借书证的卡号。
- **实现思路：**先使用 `SELECT` 判断借书证是否存在，再使用 `UPDATE` 语句实现，执行借书证修改操作，否则返回错误信息。

## 后端实验结果测试

直接使用提供的 `LibraryTest.java` 文件进行测试，结果如下：

🟢 测试运行时间: 2024/4/19 15:39:19

- 🟢 📦 bookRegisterTest()
- 🟢 📦 borrowAndReturnBookTest()
- 🟢 📦 bulkRegisterBookTest()
- 🟢 📦 incBookStockTest()
- 🟢 📦 modifyBookTest()
- 🟢 📦 parallelBorrowBookTest()
- 🟢 📦 queryBookTest()
- 🟢 📦 registerAndShowAndRemoveCardTest()
- 🟢 📦 removeBookTest()

测试均全部通过满足基本交互使用需求

## 前端界面及操作实现

根据实验要求所需要实现的部分如下：

- 1.学习前端请求知识，完成/card中 `ConfirmNewCard()`,`ConfirmModifyCard()``ConfirmRemoveCard()` ,的书写并向后端发送信号。
- 2.学习前端界面设计知识，自由设计书记管理界面，并能够实现像后端发送信号。
- 3.学习后端对于信号的处理并返回实现对应的需求。

### • /card 三个函数的实现

- `ConfirmNewCard()`
  - 使用axios发送POST请求到/card路径，将借书证的基本信息作为请求体发送。服务器返回成功或失败的消息，根据这些消息在页面上显示相应的提示信息。完成后，关闭对话框并重新查询借书证信息以刷新页面。在新建借书证的对话框中，当用户输入信息并点击“新建”按钮时，触发这个函数。函数发出POST请求将新的借书证信息发送到服务器，并在服务器返回结果后显示提示信息。
- `ConfirmModifyCard()`
  - 它通过axios向后端发送一个POST请求，将修改后的借书证信息发送给后端。如果请求成功，后端将更新数据库中的借书证信息，并返回一个成功消息。如果请求失败，后端将返回一个错误消息。
- `ConfirmRemoveCard()`
  - 首先，它使用axios发送一个DELETE请求到服务器，路径为"/card"，请求参数是{ cardID: this.toRemove }。如果删除成功，弹框会被关闭，并显示成功消息。如果删除失败，会显示错误消息。最后，执行一次QueryCards()方法来查询新的借书证信息。

### • 书籍管理界面的设计

- 由于大多使用 HTML 进行界面设计，在这里就不再叙述如何实现了，之后将在测试时讲解使用方式
- 书籍管理界面向后端传递信号的操作以及实现思路与/card中的十分相似，于是也不在过多叙述，唯一存在的区别是，/book中存在的操作过多，为了区分在传递信息时会使用 `book/u` `book/s` 等区分所需要的操作。

### • 后端对于信号的处理

后端对于型号的处理全部在了 `handle.java` 中实现，其中分成了三个不同的 `handler` 分别处理各个界面需要的操作，如下将分析/book的 `post delete option get` 操作 其他的十分类似：

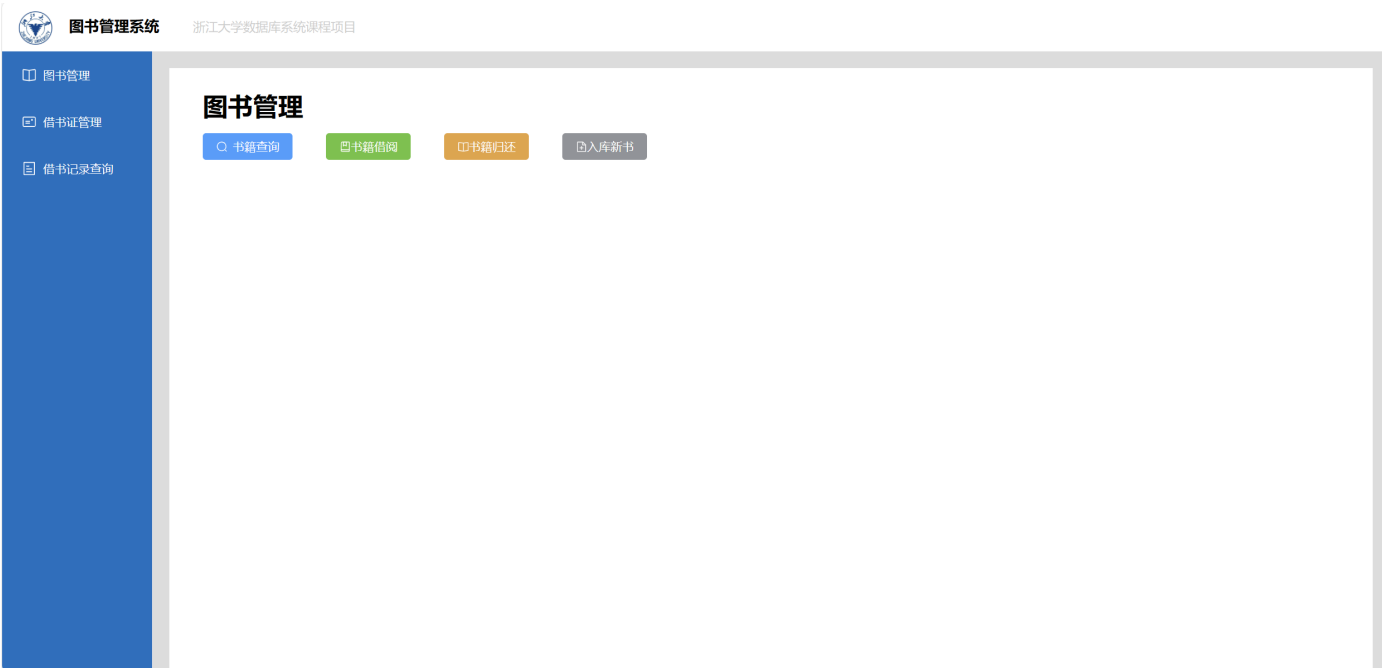
- `private void handleDeleteRequest(HttpExchange exchange) throws IOException`
  - 当收到一个删除请求时，它会解析请求的URI，从中提取书ID，然后连接到数据库并调用Library的removeBook方法来删除书籍。如果删除成功，它将返回一个成功消息，否则返回一个错误消息。最后，它将响应返回给客户端。
- `private void handlePostRequest(HttpExchange exchange) throws IOException`
  - 读取请求体：

- `handlePostRequest` 方法通过 `HttpExchange` 对象获取到请求体的输入流 (`InputStream`)。使用 `InputStreamReader` 和 `BufferedReader` 对请求体进行逐行读取，并将每一行添加到 `StringBuilder` 中以构建完整的请求体字符串。
  - 解析请求URL:
    - 从请求的URI中提取了一个特定位置的字符，即索引为6的字符，用于确定请求的类型。根据请求类型执行不同的操作：根据解析得到的请求类型字符执行相应的操作。这里使用了字符索引来确定不同类型的操作：
      - 如果索引为6的字符为 'm'，则执行修改图书信息的操作。
      - 如果索引为6的字符为 's'，则执行增加图书库存的操作。
      - 如果索引为6的字符为 'b'，则执行借阅图书的操作。
      - 如果索引为6的字符为 'r'，则执行归还图书的操作。
      - 如果索引为6的字符为 'n'，则执行存储新书籍信息的操作。
      - 如果索引为6的字符为 'u'，则不执行任何操作，直接发送成功响应。
  - 执行操作并发送响应：
    - 根据不同的操作类型，从请求体中提取所需的数据。
    - 使用提取的数据执行相应的操作（如修改图书信息、增加库存、借书、还书等）。
  - 根据操作的结果，设置响应头和响应体：
    - 如果操作成功，则发送 "success" 字符串。
    - 如果操作失败，则发送相应的错误消息。
  - 最后，关闭输出流并释放资源。
- `private void handleOptionRequest(HttpExchange exchange) throws IOException`
    - 当收到一个OPTION请求时，它会返回一个允许的响应。
  - `private void handleGetRequest(HttpExchange exchange) throws IOException`
    - 用于处理GET请求并返回图书查询结果。代码中使用了正则表达式来提取查询参数，然后使用这些参数来创建 `BookQueryConditions` 对象并调用 `library.queryBook()` 方法来执行查询。查询结果被序列化为JSON格式并发送回客户端。
    - 这个处理GET请求的方法被用来处理用户通过HTTP GET请求进行的图书查询。这些请求的URL格式如下：
 

```
http://localhost:8080/api/v1/books?category=children&title=cooking&press=penguin&minpubilshyear=2000&maxpubilshyear=2020&author=john&minprice=0.99&maxprice=19.99&sortby=publishyear&sortorder=asc
```
    - 代码中使用了硬编码的字符串，如 "application/json"、"category="、"title="等，这些应该定义为常量以便于管理和修改。正则表达式使用可能会导致性能问题，特别是在处理大量请求时。可以考虑使用更高效的查询参数解析方法。代码中使用了 `System.out.println` 进行调试，在生产环境中应该移除或使用日志框架来记录信息。

## 前端操作说明以及测试

### • 图书管理界面



- 入库功能测试

- 点击入库新书



- 手动入库 点击添加新书 输入书籍信息

书籍信息填写

×

书籍种类：

标题：

出版社：

出版年份：

作者：

价格：

库存：

取消

确定

◦ 观察到我们新添加的书籍，可以书籍信息进行修改和删除，当确认无误后可以选择提交正式入库

新书入库

^

➕添加新书

🗑清空列表

📄选取文件 只能上传JSON文件

|   | 书籍种类 | 标题  | 出版社 | 出版年份 | 作者  | 价格 | 库存  | 操作                                        |
|---|------|-----|-----|------|-----|----|-----|-------------------------------------------|
| 1 | 123  | 123 | abc | 2022 | bcd | 12 | 232 | <div><div>✎ 修改</div><div>🗑 删除</div></div> |

➕确认入库

取消

✔ 书籍插入成功!

- 库存中就已经有插入的新书了

| 书籍编号 | 书籍种类       | 标题              | 出版社     | 出版年份 | 作者      | 价格    | 库存  | 操作                                                   |
|------|------------|-----------------|---------|------|---------|-------|-----|------------------------------------------------------|
| 1    | Dictionary | Le Petit Prince | Press-E | 2009 | Nonehyo | 63.43 | 0   | <div><div>修改</div><div>删除</div><div>调整库存</div></div> |
| 2    | 123        | 123             | abc     | 2022 | bcd     | 12.0  | 232 | <div><div>修改</div><div>删除</div><div>调整库存</div></div> |

- 批量入库测试

- 此外可以点击上传文件，只允许时json文件(提供一个测试文件，为：librarymanagementsystem\src\test\java\test.json)

新书入库

添加新书

清空列表

选取文件

只能上传JSON文件

test.json

|   | 书籍种类             | 标题                      | 出版社     | 出版年份 | 作者       | 价格     | 库存 | 操作                                    |
|---|------------------|-------------------------|---------|------|----------|--------|----|---------------------------------------|
| 1 | Computer Science | Computer Networking     | Press-E | 1998 | Immortal | 215.12 | 83 | <div><div>修改</div><div>删除</div></div> |
| 2 | History          | Gone with the wind      | Press-D | 2017 | Nonehyo  | 3.97   | 5  | <div><div>修改</div><div>删除</div></div> |
| 3 | Nature           | Operating System        | Press-D | 2001 | ZaiZai   | 24.0   | 42 | <div><div>修改</div><div>删除</div></div> |
| 4 | Magazine         | Miserable World         | Press-D | 2003 | DaDa     | 221.51 | 45 | <div><div>修改</div><div>删除</div></div> |
| 5 | Horror           | The Old Man and the Sea | Press-B | 2016 | DouDou   | 33.91  | 63 | <div><div>修改</div><div>删除</div></div> |
| 6 | Novel            | Miserable World         | Press-G | 2019 | Erica    | 193.0  | 73 | <div><div>修改</div><div>删除</div></div> |
| 7 | Horror           | How steel is made       | Press-E | 2005 | Yuuku    | 188.03 | 29 | <div><div>修改</div><div>删除</div></div> |

确认入库

取消

- 再次确认无误点击提交

| 书籍编号 | 书籍种类             | 标题                       | 出版社     | 出版年份 | 作者        | 价格     | 库存  | 操作                                                   |
|------|------------------|--------------------------|---------|------|-----------|--------|-----|------------------------------------------------------|
| 1    | Dictionary       | Le Petit Prince          | Press-E | 2009 | Nonehyo   | 63.43  | 0   | <div><div>修改</div><div>删除</div><div>调整库存</div></div> |
| 2    | 123              | 123                      | abc     | 2022 | bcd       | 12.0   | 232 | <div><div>修改</div><div>删除</div><div>调整库存</div></div> |
| 3    | Computer Science | Computer Networking      | Press-E | 1998 | Immortal  | 215.12 | 83  | <div><div>修改</div><div>删除</div><div>调整库存</div></div> |
| 4    | History          | Gone with the wind       | Press-D | 2017 | Nonehyo   | 3.97   | 5   | <div><div>修改</div><div>删除</div><div>调整库存</div></div> |
| 5    | Nature           | Operating System         | Press-D | 2001 | ZaiZai    | 24.0   | 42  | <div><div>修改</div><div>删除</div><div>调整库存</div></div> |
| 6    | Magazine         | Miserable World          | Press-D | 2003 | DaDa      | 221.51 | 45  | <div><div>修改</div><div>删除</div><div>调整库存</div></div> |
| 7    | Horror           | The Old Man and the Sea  | Press-B | 2016 | DouDou    | 33.91  | 63  | <div><div>修改</div><div>删除</div><div>调整库存</div></div> |
| 8    | Novel            | Miserable World          | Press-G | 2019 | Erica     | 193.0  | 73  | <div><div>修改</div><div>删除</div><div>调整库存</div></div> |
| 9    | Horror           | How steel is made        | Press-E | 2005 | Yuuku     | 188.03 | 29  | <div><div>修改</div><div>删除</div><div>调整库存</div></div> |
| 10   | Horror           | The Old Man and the Sea  | Press-H | 2012 | DaDa      | 187.54 | 80  | <div><div>修改</div><div>删除</div><div>调整库存</div></div> |
| 11   | Philosophy       | Database System Concepts | Press-H | 2016 | ColaOtaku | 208.46 | 33  | <div><div>修改</div><div>删除</div><div>调整库存</div></div> |
| 12   | Autobiography    | How steel is made        | Press-E | 2015 | Hqs       | 229.92 | 97  | <div><div>修改</div><div>删除</div><div>调整库存</div></div> |

- 库存修改测试

| 书籍编号 | 书籍种类       | 标题              | 出版社     | 出版年份 | 作者      | 价格    | 库存 | 操作                                                   |
|------|------------|-----------------|---------|------|---------|-------|----|------------------------------------------------------|
| 1    | Dictionary | Le Petit Prince | Press-E | 2009 | Nonehyo | 63.43 | 0  | <div><div>修改</div><div>删除</div><div>调整库存</div></div> |

- 点击修改库存 增加100本



## 修改库存



修改1号书的库存吗?

调整库存

增加



100



取消

确定修改

结果成功

浙江大学数据库系统课程项目

✔ 库存调整成功!

| 书籍编号 | 书籍种类       | 标题              | 出版社     | 出版年份 | 作者      | 价格    | 库存  | 操作                                                               |
|------|------------|-----------------|---------|------|---------|-------|-----|------------------------------------------------------------------|
| 1    | Dictionary | Le Petit Prince | Press-E | 2009 | Nonehyo | 63.43 | 100 | <a href="#">✎ 修改</a> <a href="#">🗑 删除</a> <a href="#">📦 调整库存</a> |

再减少库存100本

## 修改库存



修改1号书的库存吗?

调整库存

减少



100



取消

确定修改

结果

| 书籍编号 | 书籍种类       | 标题              | 出版社     | 出版年份 | 作者      | 价格    | 库存 | 操作                                                               |
|------|------------|-----------------|---------|------|---------|-------|----|------------------------------------------------------------------|
| 1    | Dictionary | Le Petit Prince | Press-E | 2009 | Nonehyo | 63.43 | 0  | <a href="#">✎ 修改</a> <a href="#">🗑 删除</a> <a href="#">📦 调整库存</a> |

再次减少 出现报错

✖ 库存调整失败: 库存小于0!

## 修改图书信息测试

点击修改信息

修改书籍信息(书籍ID: 3)



书籍种类:

Computer Science

标题:

Computer Networking

出版社:

Press-E

出版年份:

1998

作者:

Immortal

价格:

215.12

取消

确定

◦ 修改

书籍种类:

Computer

✕

标题:

Computer

出版社:

Press-F

出版年份:

2020

作者:

kkkk

价格:

100

取消

确定

◦ 结果

|   |          |          |         |      |      |       |    |                 |                 |                   |
|---|----------|----------|---------|------|------|-------|----|-----------------|-----------------|-------------------|
| 3 | Computer | Computer | Press-F | 2020 | kkkk | 100.0 | 83 | <div>✎ 修改</div> | <div>🗑 删除</div> | <div>📦 调整库存</div> |
|---|----------|----------|---------|------|------|-------|----|-----------------|-----------------|-------------------|

• 图书查询测试

- 筛选界面输入需要查询的条件之后，即可查询

查询条件筛选

书籍种类

Q 输入书籍种类

书籍标题

Q 输入书籍标题

出版社名称

Q 输入出版社名

书籍作者

Q 输入作者姓名

年代范围

📅 选择年代范围

—

📅 选择年代范围

价格范围

Q 选择价格范围

—

Q 选择价格范围

排列顺序

书籍序号

▼

ASC

▼

Q 查询

取消

- 查询案例1：精准查询

查询条件筛选

书籍种类

Q

输入书籍种类

书籍标题

Q

Gone with the wind

出版社名称

Q

输入出版社名

书籍作者

Q

输入作者姓名

年代范围

📅

选择年代范围

——

📅

选择年代范围

价格范围

Q

选择价格范围

——

Q

选择价格范围

排列顺序

书籍序号

▼

ASC

▼

Q 查询

取消

| 书籍编号 | 书籍种类     | 标题                 | 出版社     | 出版年份 | 作者      | 价格     | 库存 | 操作                                                         |
|------|----------|--------------------|---------|------|---------|--------|----|------------------------------------------------------------|
| 4    | History  | Gone with the wind | Press-D | 2017 | Nonehyo | 3.97   | 5  | <div><div>🔗 修改</div><div>🗑 删除</div><div>📦 调整库存</div></div> |
| 27   | Magazine | Gone with the wind | Press-E | 2016 | Nonehyo | 147.85 | 94 | <div><div>🔗 修改</div><div>🗑 删除</div><div>📦 调整库存</div></div> |
| 48   | Others   | Gone with the wind | Press-E | 2022 | Nonehyo | 198.48 | 3  | <div><div>🔗 修改</div><div>🗑 删除</div><div>📦 调整库存</div></div> |

◦ 查询案例2：模糊查询

查询条件筛选

书籍种类

Q

输入书籍种类

书籍标题

Q

Gone

出版社名称

Q

输入出版社名

书籍作者

Q

输入作者姓名

年代范围

📅

选择年代范围

——

📅

选择年代范围

价格范围

Q

选择价格范围

——

Q

选择价格范围

排列顺序

书籍序号

▼

ASC

▼

Q 查询

取消

| 书籍编号 | 书籍种类     | 标题                 | 出版社     | 出版年份 | 作者      | 价格     | 库存 | 操作                                                         |
|------|----------|--------------------|---------|------|---------|--------|----|------------------------------------------------------------|
| 4    | History  | Gone with the wind | Press-D | 2017 | Nonehyo | 3.97   | 5  | <div><div>🔗 修改</div><div>🗑 删除</div><div>📦 调整库存</div></div> |
| 27   | Magazine | Gone with the wind | Press-E | 2016 | Nonehyo | 147.85 | 94 | <div><div>🔗 修改</div><div>🗑 删除</div><div>📦 调整库存</div></div> |
| 48   | Others   | Gone with the wind | Press-E | 2022 | Nonehyo | 198.48 | 3  | <div><div>🔗 修改</div><div>🗑 删除</div><div>📦 调整库存</div></div> |

◦ 查询案例3：多条件查询 并排序

查询条件筛选

书籍种类

Q

输入书籍种类

书籍标题

Q

输入书籍标题

出版社名称

Q

A

书籍作者

Q

输入作者姓名

年代范围

📅

2021

——

📅

2025

价格范围

Q

10

——

Q

100

排列顺序

书籍序号

▼

降序

▼

Q 查询

取消

| 书籍编号 | 书籍种类   | 标题              | 出版社     | 出版年份 | 作者    | 价格    | 库存  | 操作                                                         |
|------|--------|-----------------|---------|------|-------|-------|-----|------------------------------------------------------------|
| 24   | Horror | Eugenie Grandet | Press-A | 2022 | Yuuku | 25.74 | 2   | <div><div>🔗 修改</div><div>🗑 删除</div><div>📦 调整库存</div></div> |
| 2    | 123    | 123             | abc     | 2022 | bcd   | 12.0  | 232 | <div><div>🔗 修改</div><div>🗑 删除</div><div>📦 调整库存</div></div> |

◦ 查询案例4：类别只允许精准查询

查询条件筛选

书籍种类

Q Dictionary

书籍标题

Q 输入书籍标题

出版社名称

Q 输入出版社名

书籍作者

Q 输入作者姓名

年代范围

选择年代范围

选择年代范围

价格范围

选择价格范围

选择价格范围

排列顺序

书籍序号

ASC

查询

取消

| 书籍编号 | 书籍种类       | 标题                 | 出版社     | 出版年份 | 作者      | 价格     | 库存 | 操作                                          |
|------|------------|--------------------|---------|------|---------|--------|----|---------------------------------------------|
| 1    | Dictionary | Le Petit Prince    | Press-E | 2009 | Nonehyo | 63.43  | 0  | <div>修改</div> <div>删除</div> <div>调整库存</div> |
| 22   | Dictionary | Le Petit Prince    | Press-D | 2003 | DouDou  | 107.1  | 96 | <div>修改</div> <div>删除</div> <div>调整库存</div> |
| 23   | Dictionary | Operating System   | Press-B | 2015 | ZaiZai  | 181.13 | 85 | <div>修改</div> <div>删除</div> <div>调整库存</div> |
| 33   | Dictionary | Analysis of Dreams | Press-A | 2003 | Nonehyo | 110.08 | 32 | <div>修改</div> <div>删除</div> <div>调整库存</div> |
| 42   | Dictionary | Algorithms         | Press-F | 2020 | SoonWhy | 104.56 | 98 | <div>修改</div> <div>删除</div> <div>调整库存</div> |

借书测试

- 点击借书按钮 输入对应信息

书籍借阅

书籍归还

入库新书

借书登记

借书证号

1

输入书籍编号

23

选择借阅日期

2024/04/19 16:48

借阅

取消

- 成功之后就能查到借书的信息

1

查询

| 借书证ID | 图书ID | 借出时间             | 归还时间 |
|-------|------|------------------|------|
| 1     | 23   | 2024.04.19 16:48 | 未归还  |

- 再次借阅

书籍借阅失败：用户借阅此书但未归还！

还书测试

- 点击还书按钮 归还刚才接的书

还书登记

借书证号

📖 1

输入书籍编号

📖 23

选择归还日期

🕒 2024/04/19 16:51

📖 确认归还

取消

成功

1

查询

| 借书证ID | 图书ID | 借出时间             | 归还时间             |
|-------|------|------------------|------------------|
| 1     | 23   | 2024.04.19 16:48 | 2024.04.19 16:51 |

再次归还

📖 书籍归还失败：书籍重复归还！

还书登记

借书证号

📖 1

输入书籍编号

📖 23

选择归还日期

🕒 2024/04/19 16:52

📖 确认归还

取消

书籍删除测试

删除书籍

删除书籍

确定删除4号书籍吗？

取消

删除

结果成功删除

|   |          |                  |         |      |        |       |    |      |      |        |
|---|----------|------------------|---------|------|--------|-------|----|------|------|--------|
| 3 | Computer | Computer         | Press-F | 2020 | kkkk   | 100.0 | 83 | 📖 修改 | 📖 删除 | 📖 调整库存 |
| 5 | Nature   | Operating System | Press-D | 2001 | ZaiZai | 24.0  | 42 | 📖 修改 | 📖 删除 | 📖 调整库存 |

-删除未归还的书籍

# 删除书籍



确定删除1号书籍吗?

取消

























删除

-删除结果

✖ 书籍删除失败：还有用户未归还此书!

## • 借书证界面

### 借书证管理

|                                                                                                                                                                                                                                                                       |                                                                                                                                                                                                                                                                                 |                                                                                                                                                                                                                                                                                 |                                                                                                                                                                                                                                                                       |                                                                                                                                                                                                                                                                                     |                                                                                                                                                                                                                                                                                     |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <div>No. 1</div> <div>姓名: User00000<br/>部门: English Language<br/>类型: Student</div> <div></div>  | <div>No. 2</div> <div>姓名: User00001<br/>部门: Ideological &amp; Political<br/>类型: Student</div> <div></div> | <div>No. 3</div> <div>姓名: User00002<br/>部门: Ideological &amp; Political<br/>类型: Teacher</div> <div></div> | <div>No. 4</div> <div>姓名: User00003<br/>部门: English Language<br/>类型: Teacher</div> <div></div>  | <div>No. 5</div> <div>姓名: User00004<br/>部门: Ideological &amp; Political<br/>类型: Student</div> <div></div> | <div>No. 6</div> <div>姓名: User00005<br/>部门: Ideological &amp; Political<br/>类型: Student</div> <div></div> |
| <div>No. 7</div> <div>姓名: User00006<br/>部门: Civil Engineering<br/>类型: Teacher</div> <div></div> | <div>No. 8</div> <div>姓名: User00007<br/>部门: Civil Engineering<br/>类型: Teacher</div> <div></div>           | <div>No. 9</div> <div>姓名: User00008<br/>部门: Environmental Science<br/>类型: Teacher</div> <div></div>       | <div>No. 10</div> <div>姓名: User00009<br/>部门: Computer Science<br/>类型: Teacher</div> <div></div> | <div>No. 11</div> <div>姓名: User00010<br/>部门: Management<br/>类型: Teacher</div> <div></div>                 | <div>No. 12</div> <div>姓名: User00011<br/>部门: Environmental Science<br/>类型: Teacher</div> <div></div>      |

## ◦ 增加借书证测试

- 输入信息

## 新建借书证



姓名: abc

部门: abc

类型: 学生 ^

教师

学生

取消

确定

-结果

# No. 17

姓名: abc

部门: abc

类型: Student



### 。借书证修改测试

- 选择修改借书证



×

修改信息(借书证ID: 1)

姓名:

11111

⊗

部门:

English Language

类型:

Student

∨

取消

确定

▪ 修改结果

No. 1

姓名: 11111

部门: English  
Language

类型: Student

✎

🗑

◦ 借书证删除测试

▪ 删除未还书的借书证

删除借书证

×

确定删除1号借书证吗?

取消

删除

▪ 结果

✖ 借书证删除失败：用户仍然有未归还的书籍!

删除借书证

删除借书证

确定删除4号借书证吗?

取消

删除

借书记录界面

借书记录查询

输入借书证ID

查询

借书记录查询

1

查询

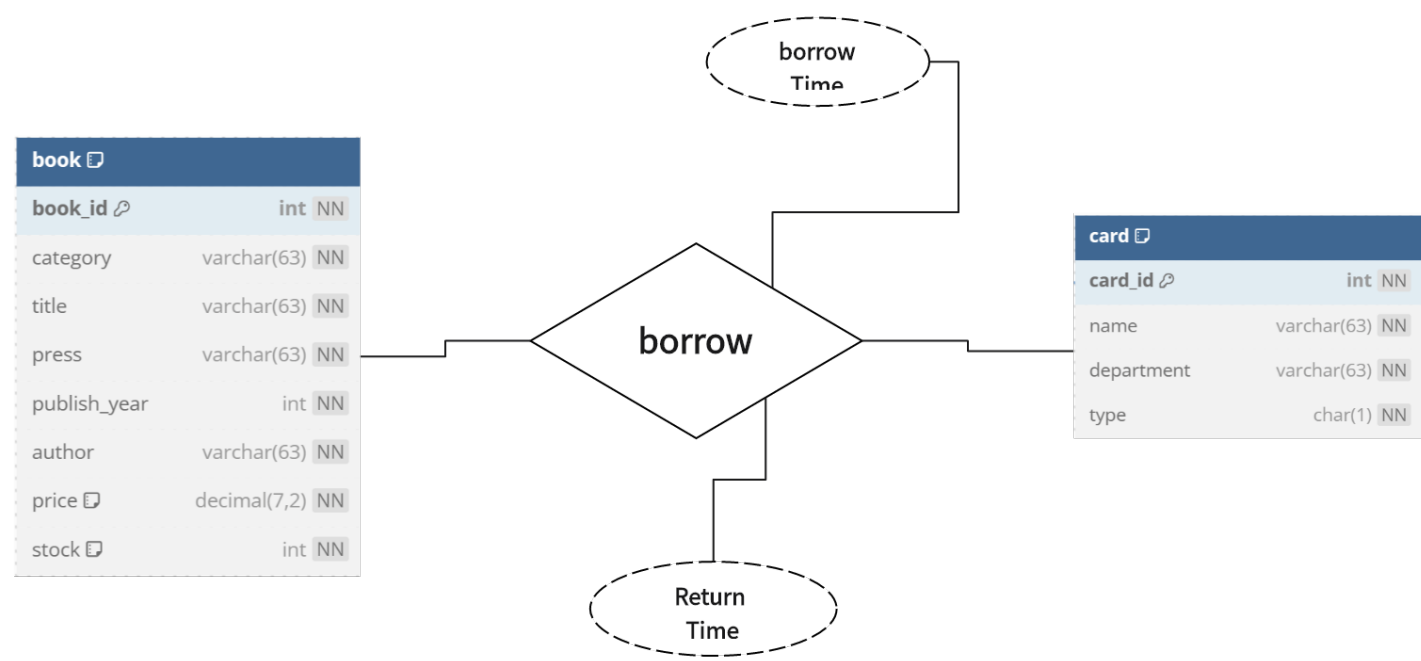
| 借书证ID | 图书ID | 借出时间             | 归还时间             |
|-------|------|------------------|------------------|
| 1     | 23   | 2024.04.19 16:48 | 2024.04.19 16:51 |
| 1     | 23   | 2024.04.19 16:53 | 未归还              |

遇到问题及解决方法

- 问题：在el-update无法上传成功文件
  - 仔细学习之后发现，在上传文件时会向后端发送一个post的信息，需要得到后端的确认，于是我重新改变了传向的对象，并返回一个success的信号，这样就能正常接受到信息了。
- 问题：在使用el-table时需要使用本行前面的所有数据 不知道怎么获取？
  - 通过阅读了官方的文档之后，可以发现在template中添加一个scope就可以通过这个来回去同行内的其他信息
- 问题：在时间的传递过程中一般都在使用unix的时间戳，但是我在现实的时候需要使用的是大家都能看懂的形式？
  - 通过查阅资料之后可以对时间戳进行格式修改调用库来进行转化
- 问题：当没有还书时时间默认为0，转化时间戳后发现，变成1970年的一个时间？
  - 在转化时间戳时，不转化等于0的时间戳，在表格显示之前添加一个判断的函数，当为0时自动输出未换书的字样。
- 问题 思考体中并行的问题
  - 将在思考题中回答

# 思考题

## 1.er图



## 2.描述SQL注入攻击的原理(并简要举例)。在图书管理系统中，哪些模块可能会遭受SQL注入攻击？如何解决？

- SQL注入是一种常见的网络安全漏洞，它允许攻击者通过在应用程序的输入字段中注入恶意的SQL代码来执行未经授权的数据库操作SQL注入攻击的原理基于应用程序未能正确过滤用户输入，导致恶意用户能够在SQL查询中插入恶意代码，从而影响数据库的执行。
- 举例：

```
username = getRequestParameter("username");
password = getRequestParameter("password");
sqlQuery = "SELECT * FROM users WHERE username='" + username + "' AND password='" + password + "'";

//输入
' OR '1'='1

//代码变成
SELECT * FROM users WHERE username=' ' OR '1'='1' AND password=' ' OR '1'='1'
```

- 在图书管理系统中的查询操作主要是**查询模块**时，可能通过SQL注入来影响整个图书管理系统。
- 解决方法
  - 使用 `prepareStatement` 于是查询时使用 `?` 替代需要用户输入的部分，`?` 的内容统一使用语句代替如：

```

PreparedStatement ps = conn
 .prepareStatement(
 "INSERT INTO book (category,title,press,publish_year,author,price,stock) VALUES (
 Statement.RETURN_GENERATED_KEYS);
ps.setString(1, book.getCategory());
ps.setString(2, book.getTitle());
ps.setString(3, book.getPress());
ps.setInt(4, book.getPublishYear());
ps.setString(5, book.getAuthor());
ps.setDouble(6, book.getPrice());
ps.setInt(7, book.getStock());
ps.executeUpdate();

```

### 3.在InnoDB的默认隔离级别(RR, Repeated Read)下，当出现并发访问时，如何保证借书结果的正确性？

- 在提示下了解了InnoDB默认隔离级别(RR, Repeated Read)实现的原理，以及快照读和当前读的区别
- 于是更具所了解到的知识，增加修改成如下,确保所查到的数据不是快照读，避免了出现异常


```

String sql = "SELECT stock FROM book WHERE book_id=? for update;";
sql = "SELECT * FROM borrow WHERE card_id=? AND book_id=? AND return_time=0 for update;";
ps = conn.prepareStatement(sql);
sql = "INSERT INTO borrow (card_id,book_id,borrow_time) VALUES (?,?,?);";
ps = conn.prepareStatement(sql);

```

- 测试也通过了

✓ 测试运行时间: 2024/4/19 17:44:52

✓  parallelBorrowBookTest()