

# 上海海事大学

## 《面向对象程序设计课程设计》



设计题目 图书馆综合管理系统

学生姓名 骆明宇

指导教师 徐 明

所在班级 计算机 211 卓

所在学院 信息工程学院

提交日期 2022 年 12 月 29 日

# 目 录

1.	引言.....	1
1.1	背景.....	1
1.2	要解决的问题.....	2
1.3	小组成员分工.....	3
2.	系统框架.....	4
3.	类设计.....	5
4.	关键技术.....	7
4.1	用户登陆功能.....	7
4.2	密文密码存储功能。.....	7
4.3	管理员和用户功能.....	8
4.4	借出、归还图书功能.....	9
4.5	标记图书功能.....	10
4.6	离线存储功能。.....	10
4.6.1	介绍.....	10
4.6.2	文件添加数据.....	14
4.6.3	文件删除数据.....	15
4.6.4	文件数据覆写.....	16
4.6.5	文件数据修改.....	17
4.7	展示图书功能。.....	18
4.8	模糊搜索功能.....	18
4.9	设置、取消座位功能.....	19
4.10	查询座位上的同学信息.....	19
4.11	可视化座次表.....	20
4.12	添加学生、管理员账号功能.....	21
4.13	删除账户功能.....	21
4.14	修改密码功能.....	21
4.15	展示所有账户功能.....	21
4.16	账户查询功能.....	21
4.16	账户统计功能.....	22
4.17	日志功能.....	22
4.18	规整输出功能.....	23
4.19	时间显示功能.....	24
4	系统运行结果.....	26
5.1	运行环境.....	26
5.2	运行结果.....	27
5	调试和改进.....	33
6	结论.....	34
	参考文献.....	34



# 1. 引言

本图书馆综合管理系统总共分为三个子系统，分别是图书管理系统、用户管理系统和座位管理系统。

## 1.1 背景

本系统旨在完成一个综合系统，使得图书馆系统可以一站式运行。

**图书管理系统一旦开发成功，以下几个好处：**

- 1) 方便借书和还书。图书管理系统可以自动记录哪些书被借出去了，哪些书是可借的，这样就可以很方便地借书和还书。
- 2) 更好地跟踪和统计图书馆资源。图书管理系统可以自动跟踪图书馆藏书情况，包括图书的位置、状态和流通情况，这样可以帮助图书馆更好地管理和规划资源。
- 3) 保护图书馆资产。图书管理系统可以帮助图书馆管理员更好地管理和保护图书馆藏书，可以跟踪图书的流通情况到读者，并及时发现和解决任何问题。
- 4) 提高借阅者满意度。图书管理系统可以帮助借阅者更快地找到所需的书籍，并方便地借书还书，提高借阅者的满意度。
- 5) 节省人力和财力。图书管理系统可以自动完成许多繁琐的任务，如记录借书还书情况、跟踪图书馆藏书情况等，这样可以节省图书馆的人力和财力。

**座位管理系统一旦开发成功，以下几个好处：**

- 1) 提高效率：座位管理系统可以帮助快速安排座位，减少人工安排的时间，提高效率。
- 2) 提升用户体验：座位管理系统可以让用户随机选座，一键选座，提升用户体验。
- 3) 减少纠纷：通过座位管理系统可以清晰地记录座位使用情况，减少因座位产生的纠纷。
- 4) 提供分析数据：座位管理系统可以收集使用数据，为管理者提供分析座位使用情况的数据，帮助管理者做出更好的决策。
- 5) 节省成本：座位管理系统可以减少人工安排座位的成本，同时也可以帮助优化座位使用，节省成本。

**人员管理系统一旦开发成功，有以下几个好处：**

- 1) 减少人力成本：通过人员管理系统可以实现自动化管理，减少人力成本。

- 2) 提升工作效率：通过人员管理系统可以实现信息的快速传递，方便对人员信息的查询和管理，提升工作效率。
- 3) 减少信息错误：人员管理系统可以通过精确的数据管理来减少信息错误，提高信息准确性。
- 4) 增强信息安全：本人员管理系统针对密码使用 SHA-256 加密。密码通过密文进行存储，使用时比对密文。大大增强了信息安全。密码保护和权限管理保障了本系统的安全，防止信息泄露。
- 5) 增强决策能力：通过人员管理系统可以提供准确的人员信息数据，为决策者提供参考，增强决策能力。

综合以上三个系统，图书馆综合系统一旦开发成功，有以下几个好处：

- 1) 人们可以一站式的获取到所有信息。
- 2) 管理员可以一站式查看本图书馆运行情况。提高了工作效率。

所以，本系统旨在开发一个综合性的图书馆管理系统，该系统的难度等级：HARD。

## 1.2 要解决的问题

接着，描述本系统做出来会是什么一个样子，能提供什么样的功能。仍以“某某人的网上超市”为例。大致上，可以类似这样描述：

本系统要提供以下几个功能：

- (1) 用户登陆功能。每个用户可以拥有自己的账号，可以自己设置用户名和密码。
- (2) 密文密码存储功能。密码加密存储，不会出现明文密码，即使数据库被攻击也不会造成信息泄漏。
- (3) 管理员和用户功能。区分了管理员和用户，自动判断是否是管理员，然后进入不同的功能界面，实现不同的功能。管理员和用户的权限也不同。
- (4) 借出、归还图书功能。同学们和管理员们可以方便的借出、归还图书。
- (5) 标记图书功能。有的图书暂时不可以使用，或是维护或是其他原因，可以在不删除的情况下标明不可用。
- (6) 展示图书功能。可以方便的显示所有图书，便于管理员查看图书的个数，目前剩余个数，借出者。
- (7) 模糊搜索功能。有时候不会完整的记住书名，只需要输入其中一两个字就可以出现。

- (8) 设置、取消座位功能。同学们可以随机选择座位,并标记座位是你的。、
- (9) 查询座位上的同学信息。当管理员需要知道某个座位上具体做的是谁的时候,就可以使用这个功能。
- (10) 可视化座次表。当想要概括性的看一下这个座位总共有多少人的话,可以使用这个功能。
- (11) 添加学生、管理员账号功能。本系统可以提前添加学生和管理员用户。
- (12) 删除账户功能。有的同学因为某些原因,想要销户的话可以进行删除操作。
- (13) 修改密码功能。有的同学想要更新密码,觉得之前的密码不太好记,也可以重新修改。
- (14) 展示所有账户功能。管理员可以直接查看当前有多少个账户,并编辑。
- (15) 账户查询功能。提供模糊词搜索,只需要搜索一两个词就可以进行搜索。
- (16) 账户统计功能。可以显示目前总共有多少管理员和用户。
- (17) 日志功能。日志系统可以方便的复盘系统的运行情况。
- (18) 规整输出功能。由于中文占位与英文不同,使用具体函数对字符进行计算,对每个字符的输出长度进行规整。使得表看起来更加简洁凝练。

### 1.3 小组成员分工

描述小组成员组成以及各成员的详细分工情况。

表 1 课程设计任务分工安排

任务分工描述	实际完成人
用户管理系统的函数编写	龚敏
座位管理系统的函数编写	李飒
图书管理系统的函数编写	满振豪
其他函数的编写	骆明宇
上述函数的基于面向对象的重构	骆明宇
主函数的编写、整体架构	骆明宇
日志、时间戳系统	骆明宇
文件读写(离线存储)	骆明宇
SHA256 加密的实现	骆明宇
函数的修改和验证	龚敏
系统稳定性测试	李飒
系统功能测试	满振豪
本报告撰写	骆明宇

## 2. 系统框架



图 2.1 系统框架图

## (1) 数据架构:

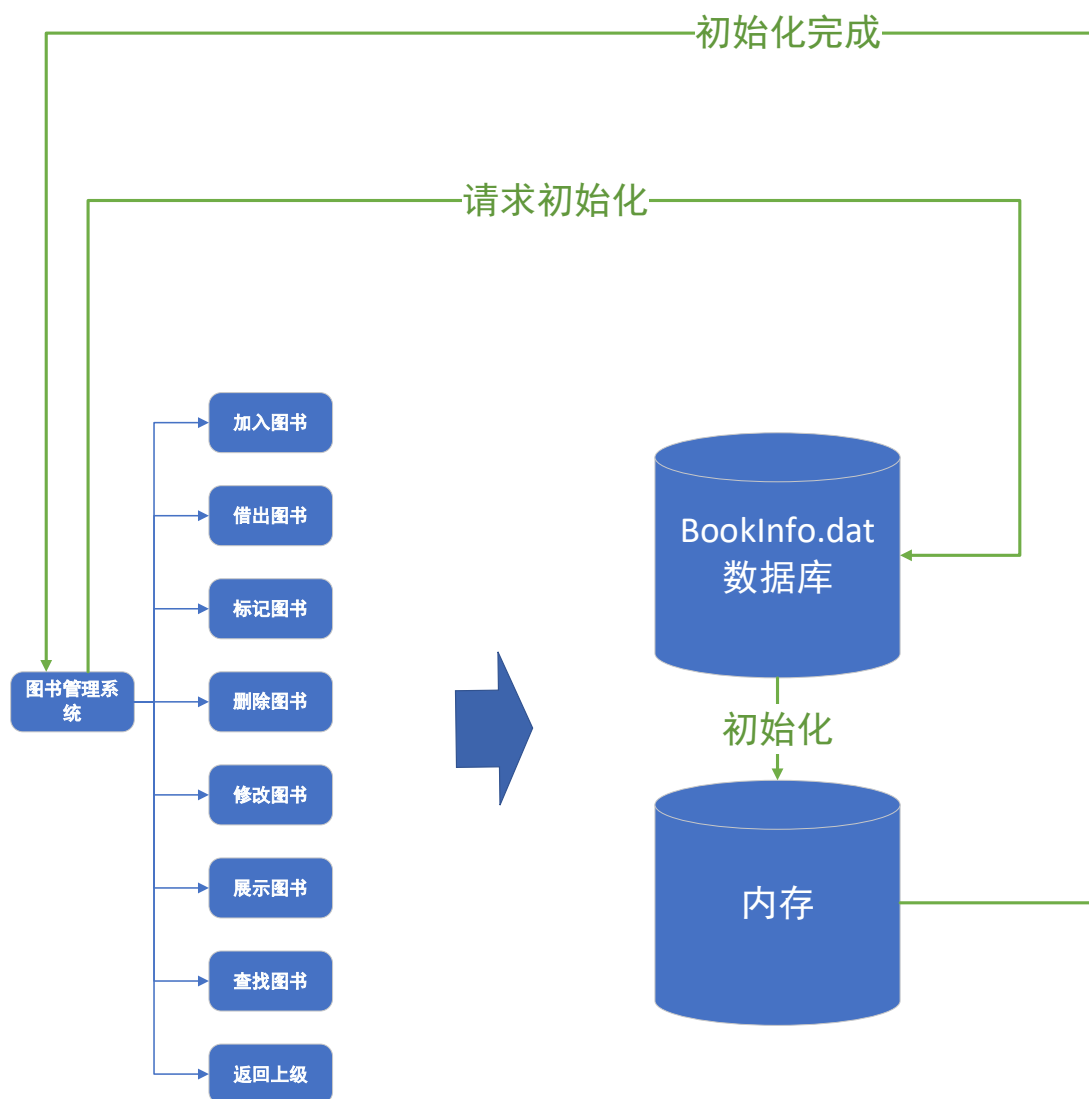


图 2.1 图书管理系统框架图

本系统大致数据架构如图所示。数据同时存储在数据库和内存中。在系统启动时，会从数据库中读取所有数据到内存中。在后续操作的时候，读取都是从内存读取，涉及到写入的内容的时候，写双份数据。一份写到内存中，一份同步更新到数据库。删除数据的时候亦是如此。

其他系统和此系统原理相同，就不过多赘述了。

### 3. 类设计

本程序总共有 7 个类，分别是 Function, Books, BookSystem, Users, UserSystem, SeatSystem, student。

Function 类是编写的一些小函数例如：模糊查找函数，日志函数，放在类中进行调用。



Books 是 BookSystem 的子类。Books 相当于一个子节点，而 BookSystem 中使用了很很多 Books，是 Books 的集合。Book 中，所有的值都是 private 的，因此，所有的数据都是安全的状态。数据不能被直接的调用，所有的函数调用都要通过 Get..()函数和 Set..()函数。

Users 是 UserSystem 的子类。Users 相当于一个子节点，而 UserSystem 中使用了很很多 Users，是 Users 的集合。Users 中，所有的值都是 private 的，因此，所有的数据都是安全的状态。数据不能被直接的调用，所有的函数调用都要通过 Get..()函数和 Set..()函数。

使用面向对象思想可以有效地保护数据的安全。在面向对象编程中，通常可以将数据封装在类的私有成员中，使得外部代码无法直接访问这些数据。这有助于防止意外的数据修改，保护数据的完整性。

此外，使用面向对象思想还可以使用访问控制修饰符来控制对数据的访问权限。例如，可以将数据定义为私有成员，然后通过公共方法访问和修改这些数据。这有助于将数据的访问权限严格限制在类的内部，从而提高数据的安全性。

另外，使用面向对象思想还可以使用抽象类和接口来提供数据的抽象视图。通过抽象类和接口，可以将数据的具体实现隐藏起来，只提供必要的接口。这有助于将数据的访问权限严格限制在类的内部，从而提高数据的安全性。

总之，使用面向对象思想可以通过封装、访问控制修饰符、抽象类和接口等方式有效地保护数据的安全。

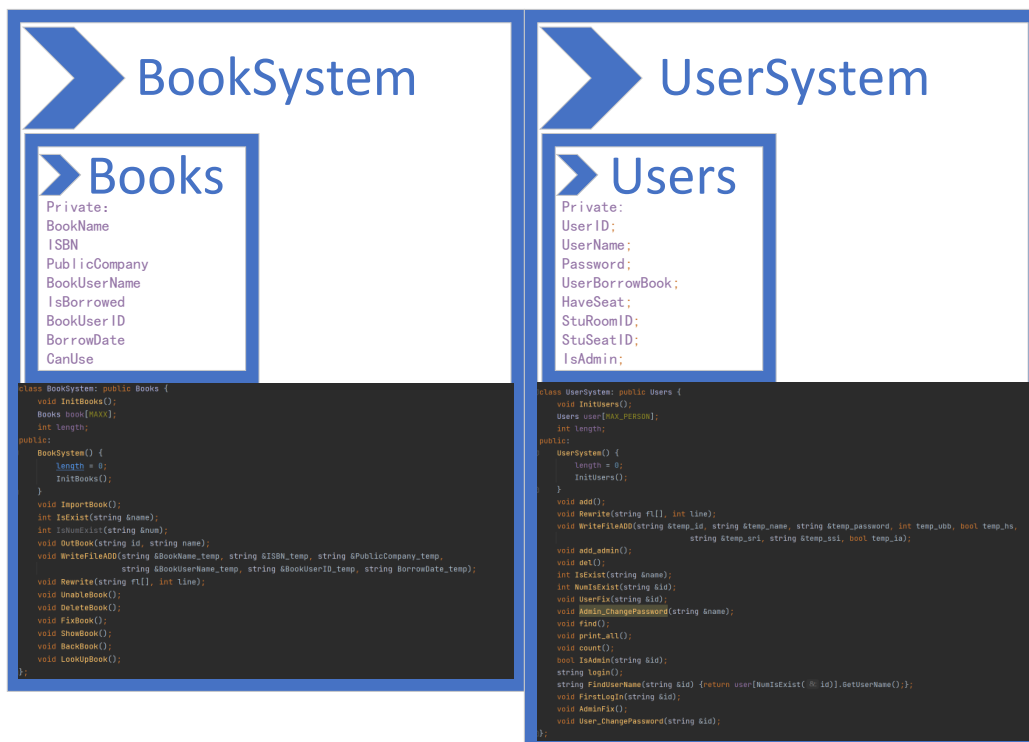


图 3 部分类设计图

`Menu` 类存放着所有的菜单，当需要使用菜单的时候，要使用 `Menu::` 进行调用。

首先，将函数写入类中可以使代码更加结构化和清晰。将相关函数放在一起可以使得代码更容易理解和维护。

其次，将函数写入类中可以使用继承来重用代码。子类可以继承父类中的函数，并在必要时对其进行修改。这可以大大减少代码的冗余，提高代码的可维护性。

此外，将函数写入类中还可以使用封装来隐藏实现细节。将函数定义为类的私有成员可以使外部代码无法直接访问这些函数，只能通过类提供的公共接口来调用。这有助于保护类的内部实现不被意外更改，并使得类更加稳定和可靠。

## 4. 关键技术

### 4.1 用户登陆功能

每个用户可以拥有自己的账号，可以自己设置用户名和密码。

用户登陆功能主要分为几个部分：

1. 用户已经有账号密码时：

本系统已经集成了本年级所有同学的账号，密码以及姓名等信息。初始密码均为 123456。同学们登陆进去的账号都是普通用户权限，而老师登陆后可以直接使用自己的账号进行登陆。

2. 用户输错密码时：系统可以自动判断，用户输入的密码和后台的密码进行比对。如果密码相同的话，再判断是否是管理员，通过不同的判断进入相对应的函数。

3. 当用户的账号不在数据库中时：系统会自动询问是否添加新用户，然后可以直接进入添加用户界面。此时添加的用户默认权限为普通用户。

### 4.2 密文密码存储功能。

密码加密存储，不会出现明文密码，即使数据库被攻击也不会造成信息泄漏。

### 1. 本系统采用 SHA-256 加密方法加密。

SHA-256 是一个加密散列函数，对任何给定的输入产生一个固定大小的输出（256 比特）。它是一个单向函数，这意味着对一个给定的输入很容易计算出输出，但从输出中生成原始输入是不可行的。这使得它对于验证数据的完整性非常有用，因为对输入的任何改变都会导致不同的输出。

值得注意的是，虽然 SHA-256 是一个广泛使用的安全加密哈希函数，但仍有可能发生碰撞，即两个不同的输入产生相同的输出。然而，这种情况发生的概率极低，而且就所有实际目的而言，这被认为是不可行的。

就本系统而言，存储的加密函数只能被验证是否正确，而不能被倒推，得知具体的明文密码，即使获取到数据库后台的密码也无法登入用户界面。

### 2. 实现过程：

本系统参考了 okdshin 的 SHA-256<sup>[1]</sup>实现过程。

系统使用 `picosha2::hash256_hex_string(x)` 进行调用，x 是一个 string 值，返回的是 x 的 SHA256 的 string 值。使用时，一般获取输入值后 `pw = picosha2::hash256_hex_string(pw)`。使用的时候直接覆盖在原变量上，使系统更加具有安全性。

## 4.3 管理员和用户功能

本系统区分了管理员和用户，自动判断是否是管理员，然后进入不同的功能界面，实现不同的功能。管理员和用户的权限也不同。

Login()函数：

While True:

    输入账号密码

    判断账号是否在数据库中

        在：

            判断密码是否正确

                正确：登陆成功

                不正确：重新输入密码

不在：

询问是否需要添加用户

需要：添加用户

不需要：重新输入密码

同时：主函数中可以显示对应的权限级别。普通用户无法使用管理员账户上的功能。这种隔离的设计，可以极大的保护系统安全。

并且 Menu()函数也同时会把当前登入账号的 name 和 id 传入其子函数，这样可以保证子函数也都可以获取到当前登入的账号。实现更多的功能，使得一些函数不用手动再次输入名字，例如下文的 seat, book 和 user 系统。

## 4.4 借出、归还图书功能

同学们和管理员们可以方便的借出、归还图书。

借出图书时、只需要输入书名，即可在你的名下借出，并在数据库中更新他们，加盖时间戳和写进日志。

归还图书时，如果有的同学因为有事，需要他人代还，可以通过其他用户或者管理员用户进行还书操作。还书时不验证姓名和学号。

借出图书时：

用户首先输入需要借阅的图书

通过 IsExist()函数进行判断，判断是否图书馆中有这本书

如果有：

从内存中找到这个书的位置，并进行标记，打上时间戳，加上日志。

更新数据库

如果没有：

直接输出查无此书。

归还图书时：

直接输入归还的图书名即可归还图书

此操作也会同步更改到数据库中

## 4.5 标记图书功能

有的图书暂时不可以使用，或是维护或是其他原因，可以在不删除的情况下标明不可用。管理员则可以在系统内标记出无法使用的图书，这个修改只会在内存中进行。因为一般对图书做维护都不会拖到第二个工作日，在内存中存储的信息在系统重启后不会从数据库中再次写入内存。

实现方法：即输入书本名称后，启动对书本可用性进行标记。

## 4.6 离线存储功能。

### 4.6.1 介绍

离线存储功能，即文件读写功能。是不同的系统通过文件的读写来实现数据库的增删改和初始化。

在 C++ 中，你可以使用标准库中的 `fstream` 类来读写文件。这个类提供了三个流对象：

`ifstream`：用于从文件读取数据的输入流对象。

`ofstream`：用于向文件写入数据的输出流对象。

`fstream`：既可以用于读取文件，也可以用于写入文件的流对象。

下面是一个简单的例子，展示了如何使用 `fstream` 类来读写文件：

```
#include <fstream>
```

```
#include <iostream>
```

```
int main() {
```

```
    // 以写模式打开文件
```

```
    std::ofstream out("output.txt");
```

```
    // 向文件写入数据
```

```
    out << "Hello, world!" << std::endl;
```

```
out.close(); // 关闭文件

// 以读模式打开文件
std::ifstream in("output.txt");
// 从文件读取一行数据
std::string line;
std::getline(in, line);
std::cout << line << std::endl; // 输出读取到的数据
in.close(); // 关闭文件

return 0;
}
```

在上面的代码中，我们使用了 `ofstream` 类的对象 `out` 来打开文件 `output.txt`，并以写模式打开。然后使用流插入运算符 `<<` 来写入数据。写入完成后，调用 `close` 函数关闭文件。

然后我们使用 `ifstream` 类的对象 `in` 来打开文件 `output.txt`，并以读模式打开。然后使用 `std::getline` 函数从文件中读取一行数据。最后，调用 `close` 函数关闭文件。

我们还可以使用文件指针来读写文件。文件指针是一个指向文件的指针，你可以使用它来对文件进行读写操作。下面是一个使用文件指针读写文件的例子：

```
#include <cstdio>

#include <iostream>

int main() {
    // 以写模式打开文件
    FILE* out = fopen("output.txt", "w");
    // 向文件写入数据
    fprintf(out, "Hello, world!\n");
    fclose(out); // 关闭文件
}
```

```
// 以读模式打开文件
FILE* in = fopen("output.txt", "r");
// 从文件读取一行数据
char line[1024];
fgets(line, 1024, in);
std::cout << line; // 输出读取到的数据
fclose(in); // 关闭文件

return 0;
}
```

在上面的代码中,我们使用 `fopen` 函数打开文件 `output.txt`,并以写模式打开。然后使用 `fprintf` 函数向文件写入数据。写入完成后,调用 `fclose` 函数关闭文件。

然后我们使用 `fopen` 函数打开文件 `output.txt`,并以读模式打开。然后使用 `fgets` 函数从文件中读取一行数据。最后,调用 `fclose` 函数关闭文件。

无论使用哪种方法读写文件,都需要注意检查文件是否打开成功,以及在读写过程中是否出现错误。

最后,还有一种方法可以在 C++ 中读写文件,那就是使用 `std::fstream` 类的模板特化 `std::fstream<char>` 来读写文件。下面是一个例子:

```
#include <fstream>
#include <iostream>

int main() {
    // 以写模式打开文件
    std::fstream<char> out("output.txt", std::ios::out);
    // 向文件写入数据
    out << "Hello, world!" << std::endl;
    out.close(); // 关闭文件

    // 以读模式打开文件
    std::fstream<char> in("output.txt", std::ios::in);
```

```

// 从文件读取一行数据

std::string line;

std::getline(in, line);

std::cout << line << std::endl; // 输出读取到的数据

in.close(); // 关闭文件

return 0;

}

```

在上面的代码中，我们使用了 `std::fstream` 类的模板特化 `std::fstream<char>` 的对象 `out` 来打开文件 `output.txt`，并以写模式打开。然后使用流插入运算符 `<<` 来写入数据。写入完成后，调用 `close` 函数关闭文件。

然后我们使用 `std::fstream` 类的模板特化 `std::fstream<char>` 的对象 `in` 来打开文件 `output.txt`，并以读模式打开。然后使用 `std::getline` 函数从文件中读取一行数据。最后，调用 `close` 函数关闭文件。

最后，还有一点需要注意的是，在使用 `fstream` 类和 `std::fstream` 类的模板特化 `std::fstream<char>` 读写文件时，可以使用流的一些方法来检查文件是否打开成功和读写过程中是否出现错误。例如，可以使用流的 `fail` 和 `bad` 方法来检查是否出现错误，使用流的 `good` 方法来检查流是否处于良好状态。

例如，你可以这样检查文件是否打开成功：

```

std::fstream<char> in("input.txt", std::ios::in);

if (!in.good()) {

    std::cerr << "Error: failed to open file." << std::endl;

    return 1;

}

```

你也可以在读写过程中使用流的 `fail` 和 `bad` 方法来检查是否出现错误：

```

std::fstream<char> out("output.txt", std::ios::out);

out << "Hello, world!" << std::endl;

if (out.fail()) {

    std::cerr << "Error: failed to write to file." << std::endl;

    return 1;

}

```



```
}
```

## 4.6.2 文件添加数据

以上介绍了文件的读写，在本系统中，使用了上述的功能：

例如，BookInfo.dat 里存储的是图书管理系统的数据库。

```
活着 978-7-5302-2153-2 北京十月文艺出版社 test 1 2022-12-20@16:56:25
杀死一只知更鸟 978-7-5447-6650-2 译林出版社 NULL NULL NULL
追风筝的人 978-7-208-06164-4 上海人民出版社 NULL NULL NULL
月亮与六便士 978-7-5442-8208-6 南海出版公司 NULL NULL NULL
平凡的世界 978-7-5302-2139-6 北京十月文艺出版社 NULL NULL NULL
小妇人 978-7-5447-6678-4 译林出版社 NULL NULL NULL
爱玛 978-7-5594-6076-9 江苏凤凰文艺出版社 NULL NULL NULL
云边有个小卖部 978-7-5404-8764-5 湖南文艺出版社 NULL NULL NULL
山茶文具店 978-7-5404-8533-7 湖南文艺出版社 NULL NULL NULL
解忧杂货店 978-7-5442-7087-8 南海出版公司 NULL NULL NULL
人间值得 978-7-5477-3431-5 北京日报出版社 NULL NULL NULL
悲惨世界 978-7-02-014105-0 人民文学出版社 NULL NULL NULL
沉默的大多数 978-7-5302-1659-0 北京十月文艺出版社 NULL NULL NULL
安娜卡列妮娜 978-7-5302-1659-1 北京十月文艺出版社 NULL NULL NULL
外婆的道歉信 978-7-5302-1659-2 北京十月文艺出版社 NULL NULL NULL
呐喊 978-7-5302-1659-3 北京十月文艺出版社 NULL NULL NULL
秘密花园 978-7-5302-1659-4 北京十月文艺出版社 NULL NULL NULL
你当像鸟飞往你的山 978-7-5302-1659-5 湖南文艺出版社 NULL NULL NULL
非暴力沟通 978-7-5302-1659-6 湖南文艺出版社 NULL NULL NULL
别输在不会表达上 978-7-5302-1659-7 湖南文艺出版社 NULL NULL NULL
谈话的力量 978-7-5302-1659-8 湖南文艺出版社 NULL NULL NULL
沟通的艺术 978-7-5302-1659-9 湖南文艺出版社 NULL NULL NULL
关键对话 978-7-5404-1738-0 北京十月文艺出版社 NULL NULL NULL
人性的弱点 978-7-5404-1738-1 南海出版公司 NULL NULL NULL
掌控谈话 978-7-5404-1738-2 湖南文艺出版社 NULL NULL NULL
学会提问 978-7-5404-1738-3 湖南文艺出版社 NULL NULL NULL
```

图 4.6.2 BookInfo.dat 数据库

依次存储的分别是书名、ISBN 号、出版社、借阅人、学号和时间戳。

系统初始化的时候会从文件中读取所有的数值，读取的方式如下：

//建立一个类对象

```
ifstream fin;
```

//打开文件

```
fin.open("bookInfo.dat");
```

//判断文件是否打开，增加程序鲁棒性

```
if (fin.is_open()) {
```

```
    for (int i = 0; i++) {
```

        //遍历，依次读入到内存中

```

    string a, b, c, d, e, f;
    fin >> a >> b >> c >> d >> e >> f;
    book[length++].ImportBook(a, b, c, d, e, f);
    if (fin.eof())
    {
        cout << "图书系统共初始化" << length << "本书，初始化完毕。" <<
endl;

        break;
    }
}
//关闭文件
fin.close();
} else {cout << "bookInfo.dat 文件打开失败" << endl;}

```

依次读取文件中的书本，并初始化到内存中，这里举例的是 `BookSystem`，而 `UserSystem` 初始化的时候亦是如此。

### 4.6.3 文件删除数据

接下来是删除，当需要删除的时候，我们先创建一个 `string` 数组，用来存储文件中的数据，我们一列列的读入数据到 `string` 数组中，然后判断这一列的数据是否需要删除的那一行，如果是的话，就不把他加入到 `string` 数组中，如果不是的话，就直接把他加入到 `string` 数组中，此时 `string` 数组里面就有了文件的整个数据。再把整个 `string` 数组覆写到文件中。实现代码如下：

```

stringstream ss; // 定义一个流字符串对象
fstream fff; // 定义文件流对象
fff.open("bookInfo.dat"); // 打开文件
string fl[MAXX_FILE]; // 定义字符串数组用于保存文件中的每行内容
int line = 0; // 记录文件中的行数
string temp_name, temp_isbn, temp_public, temp_username, temp_userid, temp_date,

```

```

white_space = " "; // 定义临时字符串用于保存文件中的每个字段
if (fff.is_open()) { // 如果文件打开成功
    while (true) { // 循环读取文件
        fff >> temp_name >> temp_isbn >> temp_public >> temp_username >>
temp_userid >> temp_date; // 读取一行文件的内容
        if (temp_name != name) { // 如果读取的行的书名不是所要删除的书名
            ss << temp_id << " " << temp_name << " " << temp_password << " " <<
temp_ubb << " " << temp_hs << " " << temp_sri << " " << temp_ssi << " " <<
temp_ia; // 使用流字符串进行输入
            fl[line++] = ss.str(); // 保存
        }
        if (fff.eof()) {break;} // 如果到了文件末尾，退出循环
    }
    fff.close(); // 关闭文件
    Rewrite(fl, line); // 调用函数 Rewrite，将新内容写入文件
}

```

#### 4.6.4 文件数据覆写

上述代码中的 Rewrite 函数，是负责把 fl 的 string 数组重写写到文件中。

// 使用 fstream 类型的变量 ff 打开文件 "bookInfo.dat"

```
fstream ff;
```

```
ff.open("bookInfo.dat", ios::out | ios::trunc);
```

// 如果文件已成功打开

```
if (ff.is_open()) {
```

```
    // 使用循环写入 fl 数组的每一行
```

```
    for (int i = 0; i < line; ++i)
```

```
    {
```

```
        // 将 fl[i] 写入文件
    }
```

```
        ff << fl[i];  
        // 如果不是最后一行，则写入换行符  
        if (i != line - 1)  
            ff << endl;  
    }  
} else {  
    // 如果文件未能成功打开，输出 "orz"  
    cout << "orz" << endl;  
}  
  
// 关闭文件  
ff.close();  
// 输出 "数据库更新成功"  
cout << "数据库更新成功" << endl;
```

这段代码实现了将一个名为 `fl` 的字符串数组写入到名为 `"bookInfo.dat"` 的文件中。首先，使用 `fstream` 类型的变量 `ff` 打开文件，并使用 `ios::out` 和 `ios::trunc` 标志指定以写入模式打开文件，并清除文件内容。

然后，使用 `ff.is_open()` 检查文件是否已成功打开。如果文件已打开，则使用一个循环将 `fl` 数组的每一行写入文件。最后，使用 `ff.close()` 关闭文件，并输出 "数据库更新成功"。

#### 4.6.5 文件数据修改

与删除类似，只不过删除是不把要删除的数据添加到字符串数组中。文件修改是把要修改的数据，修改完后添加到字符串数组中。

最后在实现一遍文件覆写功能。由于功能相似，这里就不再过多赘述。

## 4.7 展示图书功能。

可以方便的显示所有图书，便于管理员查看图书的个数，目前剩余个数，借出者。

这部分主要是一个循环，i 从 0 到 length，全部输出。当遇到 NULL 的时候，不输出即可。

```
if (length == 0) {  
    cout << "书架为空" << endl;  
}  
else {  
    //这里输出表头  
    for (int i = 0; i < length; ++i) {  
        //这里输出每本书的信息  
        cout << endl;  
    }  
}
```

## 4.8 模糊搜索功能

有时候我们不没有记住完整的记住书名，这个功能的加入只需要输入其中一两个字就可以出现。

具体的实现过程是判断输入的字符串是否是书本的字串。通过 FuzzySearch() 函数进行实现。

```
// 定义一个 string::size_type 类型的变量 idx  
string::size_type idx;
```

```
// 使用 find 方法在 a 中查找 b，并将结果存储在 idx 中  
idx = a.find(b);
```

```
// 如果 idx 等于 string::npos, 则返回 false
if (idx == string::npos) return false;
// 否则, 返回 true
else return true;
```

首先, 定义一个 `string::size_type` 类型的变量 `idx`。然后, 使用 `string` 类型的 `find` 方法在 `a` 中查找 `b`。`find` 方法返回 `b` 在 `a` 中出现的位置, 如果未找到, 则返回 `string::npos`。

接下来, 使用一个条件语句检查 `idx` 的值是否等于 `string::npos`。如果等于, 则返回 `false`; 否则, 返回 `true`。

## 4.9 设置、取消座位功能

同学们可以随机选择座位, 并标记座位是你的。

代码的实现过程如下:

首先, 定义了一个 `time_t` 类型的变量 `t`, 并使用 `srand` 函数将其初始化为当前时间。接下来, 使用一个嵌套循环枚举座位数组中的每一个位置。如果当前位置为空, 则跳出循环。如果已经枚举完所有位置, 则输出 "本教室学生已满!" 并退出函数。

然后, 使用一个 `do-while` 循环随机生成一个座位号。如果对应的座位已有人, 则继续生成座位号。最后, 使用一个循环检查该学生的学号是否与其他学生的学号重复, 如果重复, 则重新输入该学生的信息, 直到学号不再重复为止。最后, 将该学生的信息存储到学生类中, 并输出 "操作成功!"。

## 4.10 查询座位上的同学信息

当管理员需要知道某个座位上具体做的是谁的时候, 就可以使用这个功能。

首先, 输出提示信息, 要求用户输入行号和列号。然后, 使用 `cin` 读入用户输入的行号和列号, 并将其减一, 因为数组的索引从零开始。接下来, 使用一个条件语句检查指定位置上的学生的学号和姓名是否为空。如果为空, 则输出 "

该座位目前为空"; 否则, 输出该学生的学号和姓名。

// 定义行号和列号

```
int r,c;
```

// 输出提示信息, 要求用户输入行号和列号

```
cout<<"请输入行号 row 和列号 col: ";
```

// 读入用户输入的行号和列号

```
cin>>r>>c;
```

// 将行号和列号减一, 因为数组的索引从零开始

```
r = r-1;
```

```
c = c-1;
```

// 检查指定位置上的学生的学号和姓名是否为空

```
if ( cstu[r*8+c].UserID == "" || cstu[r*8+c].UserName == "")
```

```
{
```

```
    // 如果为空, 则输出 "该座位目前为空"
```

```
    cout<<"该座位目前为空."<<endl;
```

```
}
```

```
else
```

```
{
```

```
    // 否则, 输出该学生的学号和姓名
```

```
    cout<<"学号"<<cstu[r*8+c].UserID<<", 姓名:"<<cstu[r*8+c].UserName<<endl;
```

```
}
```

```
}
```

## 4.11 可视化座次表

当想要概括性的看一下这个座位总共有多少人的话, 可以使用这个功能。

首先，定义了两个变量 `r` 和 `c`，用于枚举座位数组中的每一个位置。然后，使用一个嵌套循环枚举座位数组中的每一个位置。如果当前位置有人，则使用 `cout` 输出该位置上学生的学号；否则，输出 "\*\*\*\*\*"。最后，输出两个空行。

## 4.12 添加学生、管理员账号功能

本系统可以添加学生和管理员用户。添加操作是在对象数组后面再添加一个数组元素。并且在数据库中添加。

## 4.13 删除账户功能

有的同学因为某些原因，想要销户的话可以进行删除操作。同时也删除数据库。

## 4.14 修改密码功能

有的同学想要更新密码，觉得之前的密码不太好记，也可以重新修改。此操作可以在内存中和数据库中同时修改。

## 4.15 展示所有账户功能

管理员可以直接查看当前有多少个账户，并编辑。遍历整个内存并输出。

## 4.16 账户查询功能

提供模糊词搜索，只需要搜索一两个词就可以进行模糊搜索。调用了 `FuzzySearch` 函数。



```
while (true) {  
    // 输出头一行，显示以下每列是干什么的。  
    for (int i = 0; i < length; i++) // 遍历所有对象  
    {  
        //如果符合模糊搜索的条件  
        if(FuzzySearch(user[i].GetUserID(),aa)||  
FuzzySearch(user[i].GetUserName(), aa))  
        {  
            //输出当前对象的所有信息  
        }  
    }  
    if (!flag) cout << "请换个关键词进行搜索" << endl;  
    else break;  
}
```

## 4.16 账户统计功能

可以显示目前总共有多少管理员和用户。即遍历一遍，输出对象数组长度和管理员个数。管理员个数统计可以定义一个计数器。

## 4.17 日志功能

日志系统可以方便的复盘系统的运行情况。

```

[2022-12-21@21:48:56] UserLogin 202110310087 08a45e752645753ed9455c2fd8ec4459ab27908e051e3d67b5020908a2ada
[2022-12-21@22:21:53] BookSystemInitSuccess NULL NULL
[2022-12-21@22:34:18] BookSystemInitSuccess NULL NULL
[2022-12-21@22:34:18] UserSystemInitSuccess NULL NULL
[2022-12-21@22:34:24] AdminLogin 1 6b86b273ff34fce19d6b804eff5a3f5747ada4eaa22f1d49c01e52ddb7875b4b
[2022-12-21@22:34:55] BookSystemUnableBookSuccess 我们仨 NULL
[2022-12-21@22:35:24] BookSystemInitSuccess NULL NULL
[2022-12-21@22:35:24] UserSystemInitSuccess NULL NULL
[2022-12-21@22:35:39] LoginFailed 001 6b86b273ff34fce19d6b804eff5a3f5747ada4eaa22f1d49c01e52ddb7875b4b
[2022-12-21@22:35:42] AdminLogin 1 6b86b273ff34fce19d6b804eff5a3f5747ada4eaa22f1d49c01e52ddb7875b4b
[2022-12-21@22:35:48] BookSystemInitSuccess NULL NULL

```

图 4.17 部分日志数据

以上的登陆日志写了每个人怎么登陆的，登陆状态等参数。

## 4.18 规整输出功能

由于中文占位与英文不同，使用具体函数对字符进行计算，对每个字符的输出长度进行规整。使得表看起来更加简洁凝练。

使用了 `GetWidth()`函数，使得其可以动态的调整中英文之间的输出大小的关系。

// 函数 `GetWidth` 计算字符串 `a` 的宽度

// 参数 `num` 表示原来的宽度

// 返回值表示新的宽度

```
int GetWidth(int num, string a) {
```

```
    // 如果 a 的第一个字符是字母或数字，直接返回 num
```

```
    if ((a[0] >= 65 && a[0] <= 90) || (a[0] >= 97 && a[0] <= 122) || (a[0] >= 48 &&
a[0] <= 57)) return num;
```

```
    // 否则，返回 num 加上 a 字符串的长度乘上 0.48
```

```
    else return int(num + (0.48 * a.length()));
```

```
}
```

这段代码的功能是根据输入的字符串计算宽度。是否使用此函数对输出有着很大的影响，见下图：

书名	ISBN	出版社	书名	ISBN	出版社
活着	978-7-5302-2153-2	北京十月文艺出版社	活着	978-7-5302-2153-2	北京十月文艺出版社
杀死一只知更鸟	978-7-5447-6650-2	译林出版社	杀死一只知更鸟	978-7-5447-6650-2	译林出版社
追风筝的人	978-7-208-06164-4	上海人民出版社	追风筝的人	978-7-208-06164-4	上海人民出版社
月亮与六便士	978-7-5442-8208-6	南海出版公司	月亮与六便士	978-7-5442-8208-6	南海出版公司
平凡的世界	978-7-5302-2139-6	北京十月文艺出版社	平凡的世界	978-7-5302-2139-6	北京十月文艺出版社
小妇人	978-7-5447-6678-4	译林出版社	小妇人	978-7-5447-6678-4	译林出版社
爱玛	978-7-5594-6076-9	江苏凤凰文艺出版社	爱玛	978-7-5594-6076-9	江苏凤凰文艺出版社
云边有个小卖部	978-7-5404-8764-5	湖南文艺出版社	云边有个小卖部	978-7-5404-8764-5	湖南文艺出版社
山茶文具店	978-7-5404-8533-7	湖南文艺出版社	山茶文具店	978-7-5404-8533-7	湖南文艺出版社
解忧杂货店	978-7-5442-7087-8	南海出版公司	解忧杂货店	978-7-5442-7087-8	南海出版公司
人间值得	978-7-5477-3431-5	北京日报出版社	人间值得	978-7-5477-3431-5	北京日报出版社
悲惨世界	978-7-02-014105-0	人民文学出版社	悲惨世界	978-7-02-014105-0	人民文学出版社
沉默的大多数	978-7-5302-1659-0	北京十月文艺出版社	沉默的大多数	978-7-5302-1659-0	北京十月文艺出版社
安娜卡列妮娜	978-7-5302-1659-1	北京十月文艺出版社			
外婆的道歉信	978-7-5302-1659-2	北京十月文艺出版社			
呐喊	978-7-5302-1659-3	北京十月文艺出版社			
秘密花园	978-7-5302-1659-4	北京十月文艺出版社			

图 4.18 规整输出功能使用前后对比图

## 4.19 时间显示功能

时间显示可以帮助在日志里得到具体的时间。使用 `GetDate()` 函数。

这段代码的功能是获取当前的日期和时间，并返回一个格式化后的字符串。

在添加注释时，你可以考虑描述代码的功能，以及每个变量和函数的作用。

这可以帮助你和其他人理解代码，并使代码更易于维护。

// 函数 `GetDate` 返回当前的日期和时间

```
string GetDate() {
```

```
    // 定义 tm 结构体指针 newtime，用于保存当前时间
```

```
    struct tm *newtime;
```

```
    // 定义字符数组 tmpbuf，用于保存格式化后的时间
```

```
    char tmpbuf[128];
```

```
    // 定义字符串变量 a，用于保存时间
```

```
    string a;
```

```
    // 定义 time_t 类型变量 lt1，用于保存当前时间
```

```
    time_t lt1;
```

```
    // 获取当前时间
```

```
    time( &ltlt1 );
```

```
    // 将当前时间转化为本地时间
```

```
    newtime=localtime(&ltlt1);
```

```
// 使用 strftime 函数格式化时间
// %F 表示年-月-日, %T 表示时:分:秒
strftime( tmpbuf, 128, "%F@%T", newtime);
// 将格式化后的时间保存到字符串 a 中
a = tmpbuf;
// 返回字符串 a
return a;
}
```

## 4.20 流字符串 stringstream

由于本系统多处需要将字符串相加, 所以使用流字符串可以更好的减少代码的行数, 增加代码的可读性。

流字符串需要包涵头文件 `#include <sstream>`。以下为一个例子:

```
#include <sstream>

int main() {
    stringstream ss;
    ss << 1 << 2 << 3 << 4 << 5;
    string str = ss.str();
    cout << str << endl;
    return 0;
}
```

以上代码使用字符串流 (stringstream) 来生成字符串 12345。

## 4.21 多文件编程

在 C++ 中, 我们可以使用多个文件来组织程序代码。这种方法称为多文件编程。

下面是一个多文件程序的基本结构:

1. 一个主文件，包含程序的入口函数 `main` 和其他代码。
2. 一个或多个头文件，包含函数原型、常量定义、类定义等信息。
3. 一个或多个源文件，包含函数实现、变量定义等信息。

在多文件编程中，可以使用 `#include` 命令来在主文件中包含头文件，以便在主文件中使用头文件中的信息。同时，可以使用 `#include` 命令在头文件中包含其他头文件，以便在头文件中使用其他头文件中的信息。

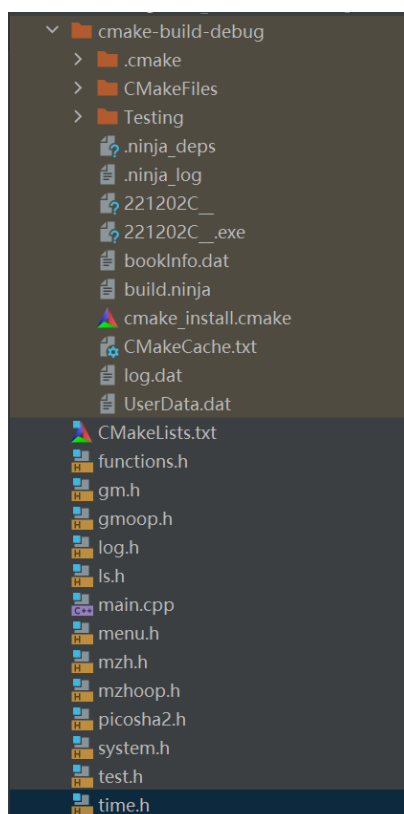


图 4.21 本程序多文件编程图

## 4 系统运行结果

### 5.1 运行环境

硬件：MacBook Pro M1 版本 16+512G

系统：macOS Ventura 13.1

软件：CLion 22.2.1

NotePad++



图 4.17 系统运行环境图

## 5.2 运行结果

### 5.3.1 系统主界面

系统流程图，描述：

- (1) 进入系统后，系统首先会从数据库中读取加载到内存中，之后所有的变动都会同步在数据库和内存中修改。加载到内存的原因是因为这样速度较快。三个子系统分别初始化完成后会输出如下图所示：

```
图书系统共初始化100本书，初始化完毕。
人员管理系统共初始化232个用户共计3个管理员，初始化完毕。
座位系统共初始化64个座位，初始化完毕。

***欢迎使用本图书馆管理系统***
当前时间为：2022-12-20 21:48:31
```

图 5.3.1.1 系统初始化图

- (2) 用户进行登陆。根据账号判断是否是管理员。
- (3) 如果是管理员：

```

*****请登录*****
请输入学号/工号: 001
请输入密码: admin
登陆成功: 欢迎管理员徐明老师
*****
2022-12-20 21:50:00
欢迎管理员: 徐明老师
工号: 001
*****
    
```

图 5.3.1.2 管理员登陆

- a. 进入菜单，如果输入 1 进入图书管理系统：

```

*****
** 图书馆综合管理系统（管理员端） **
*****
*****1.图书管理*****
*****2.座位管理*****
*****3.人员管理*****
*****0.返回上级*****
*****
    
```

图 5.3.1.3 图书馆综合管理系统（管理员端）

```

*****
2022-12-20 21:54:20
欢迎管理员: 徐明老师
工号: 001
*****
** 图书管理系统（管理员端） **
*****
*****1.加入图书*****
*****2.借出图书*****
*****3.标记图书*****
*****4.删除图书*****
*****5.修改图书*****
*****6.展示图书*****
*****7.归还图书*****
*****8.查找图书*****
*****0.返回上级*****
*****
    
```

图 5.3.1.4 图书管理系统（管理员端）

- i. 输入 1，则添加图书，添加图书会在缓存内添加，同时也会写文件，添加到 BookInfo.dat 文件中。此文件作为此系统的数据库使用。

```

1
请输入书籍的名字:
课设4.0要分教程
请输入书ISBN号:
4444-5140-4444
请输入出版社:
朕的江山不用出版社
导入日期: 2022-12-20 21:59:42
添加成功
数据库添加成功

```

图 5.3.1.5 添加图书

99	唐诗三百首	978-4-2005-2377-6	人民文学出版社	NULL	NULL	NULL
100	海子诗全集	978-4-2005-2377-7	人民文学出版社	NULL	NULL	NULL
101	课设4.0要分教程	4444-5140-4444	朕的江山不用出版社	NULL	NULL	NULL

图 5.3.1.6 数据库

- ii. 输入 2，则借出图书，输入书名可以进行借阅。借出操作会同时在内存和数据库中体现。

```

2
输入你要借阅的图书
三体
借阅成功
数据库更新成功

```

图 5.3.1.7 借阅图书

97	第七重奏	978-4-5405-1873-8	北京日报出版社	NULL	NULL	NULL
98	恶意	978-4-5405-1873-7	北京日报出版社	NULL	NULL	NULL
99	三体	978-4-5405-1873-8	北京日报出版社	徐明老师	001	2022-12-20 22:01:57
90	西游记	978-4-5405-1873-9	人民文学出版社	NULL	NULL	NULL
91	红楼梦	978-4-2005-2377-0	人民文学出版社	NULL	NULL	NULL

图 5.3.1.8 数据库更新

- iii. 输入 3，则标记不可用图书。用来在维护图书的时候使用。

```

3
请输入要标记不可用图书
我们仨
已标记为不可用

```

图 5.3.1.9 标记不可用图书

- iv. 输入 4，删除图书。用于图书丢失的时候使用。此操作会同步记录在内存和数据库中，属于不可逆操作。



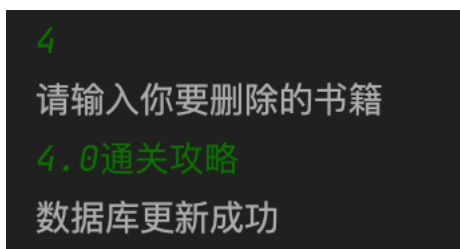


图 5.3.1.10 删除图书

- v. 输入 5，修改图书。有的时候书籍录入错误，可以在这里修改图书。



图 5.3.1.11 修改图书

- vi. 输入 6，展示图书。从内存中遍历显示所有的书，方便看到有哪些书可以看。

书名	ISBN	出版社	借阅人姓名	借阅人学号	借阅日期
活着	978-7-5302-2153-2	北京十月文艺出版社	test	1	2022-12-20@16:56:25
杀死一只知更鸟	978-7-5447-6650-2	译林出版社			
追风筝的人	978-7-208-06164-4	上海人民出版社			

图 5.3.1.12 展示图书

- vii. 输入 7，查找图书。输入书的一到两个字即可进行搜索。

输入你要查找的图书的关键词，支持模糊查找					
书名	ISBN	出版社	借阅人姓名	借阅人学号	借阅日期
追风筝的人	978-7-208-06164-4	上海人民出版社			
平凡的世界	978-7-5302-2139-6	北京十月文艺出版社			
沉默的大多数	978-7-5302-1659-0	北京十月文艺出版社			
外婆的道歉信	978-7-5302-1659-2	北京十月文艺出版社			
你当像鸟飞往你的山	978-7-5302-1659-5	湖南文艺出版社			
谈话的力量	978-7-5302-1659-8	湖南文艺出版社			
沟通的艺术	978-7-5302-1659-9	湖南文艺出版社			
人性的弱点	978-7-5404-1738-1	南海出版公司			
逻辑思考的100个关键	978-7-5404-1738-9	人民文学出版社			

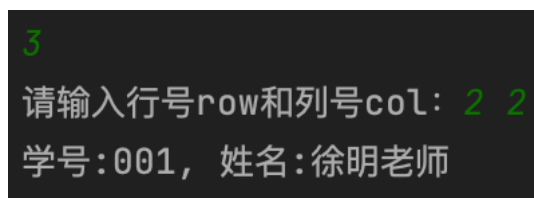
图 5.3.1.13 图书关键字模糊查找

- viii. 输入 0，返回上一级。
- b. 如果输入 2，进入座位管理系统。
- i. 输入 1，进行座位的设置。只会设置当前账号的位置。为了安全设计，我们无法设置别人账户的位置。这个时候按 5，显示座次表，会如下所示：

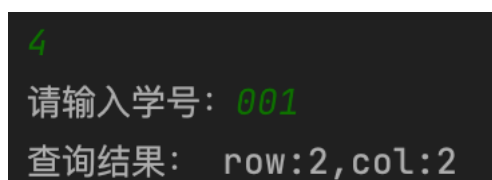
ii. 输入 2，会取消座位的设置。这个时候按 5，显示座次表，会如下所示：



iii. 输入 3，则会显示查询指定行列的座位占用情况。



iv. 输入 4，则会通过学号查询同学当前的位置。



- v. 输入 0，则会返回上一级系统。
- c. 若输入 3，则会进入人员管理系统。

```
*****
2022-12-20@23:06:44
欢迎管理员：徐明老师
工号：001
*****
**   人员管理系统（管理员端）   **
*****
*****1.添加学生账户*****
*****2.添加管理员账户*****
*****3.删除账户*****
*****4.修改密码*****
*****5.展示所有账户*****
*****6.账户查询*****
*****7.编辑账户信息*****
*****8.统计账户个数*****
*****0.返回上级*****
*****
```

图 5.3.1.18 人员管理系统（管理员端）

- i. 输入 1，可以添加学生账户。同时会更新在内存和数据库中。
- ii. 输入 2，可以添加管理员账户。同时会更新在内存和数据库中。
- iii. 输入 3，可以删除一个选中的账户。同时会更新在内存和数据库中。此修改不可逆。
- iv. 输入 4，可以修改账户的密码。同时会更新在内存和数据库中。
- v. 输入 5，可以展示所有的账户。

学号或工号	用户名	借阅图书数量	是否有座位	房间	座位号	是否管理员
1	test_admin	0	0	NULL	NULL	1
2	test_user	0	0	NULL	NULL	0
001	徐明老师	0	0	NULL	NULL	1
002	骆明宇	0	0	NULL	NULL	1
003	admin	0	0	NULL	NULL	1
202110310001	曹丹	0	0	NULL	NULL	0
202110310002	华思婷	0	0	NULL	NULL	0
202110310003	李欣怡	0	0	NULL	NULL	0
202110310004	唐飞杨	0	0	NULL	NULL	0
202110310005	石中钰	0	0	NULL	NULL	0
202110310006	王毅鑫	0	0	NULL	NULL	0
202110310007	丁鹏	0	0	NULL	NULL	0
202110310008	李文雯	0	0	NULL	NULL	0
202110310009	陈肖	0	0	NULL	NULL	0
202110310010	张欣雨	0	0	NULL	NULL	0
202110310011	董云海	0	0	NULL	NULL	0
202110310012	吴林为	0	0	NULL	NULL	0

图 5.3.1.19 人员管理系统展示

d. 相应的，学生端拥有专属菜单，但是相对应的功能会有所删减。

<pre> ***** 2022-12-20@23:51:57 欢迎用户: test_user 学号: 2 ***** ** 图书借阅系统 (学生端) ** ***** *****1. 借出图书***** *****2. 展示图书***** *****3. 归还图书***** *****4. 查找图书***** *****0. 返回上级***** ***** </pre>	<pre> ***** 2022-12-20@23:52:12 欢迎用户: test_user 学号: 2 ***** ** 座位管理系统 (学生端) ** ***** *****1. 设置座位***** *****2. 取消座位***** *****3. 显示座次表***** *****0. 返回上级***** ***** </pre>	<pre> ***** 2022-12-20@23:52:40 欢迎用户: test_user 学号: 2 ***** ** 人员管理系统 (学生端) ** ***** *****1. 修改密码***** *****2. 我的账户***** *****3. 修改姓名***** *****0. 返回上级***** ***** </pre>
---	--	---

图 5.3.1.13 用户菜单展示（学生端）

## 5 调试和改进

1. 我们遇到的问题是面向对象化编程的实现。一开始我们都是使用数据结构的思想进行编程，认为后期可以通过替换几个代码或者增加几行代码就可以实现。但是事实并不是这样，面向对象思想和平时普通的思想不尽相同。所以我们在最后重写了 70% 的代码使得本项目具有面向对象的思想。
2. 我们在实现本项目的过程中，同样遇到了很多问题，原因是我们的基础不太扎实。有时会犯一些非常低级的错误。比如说==会经常打成=，导致代码运

行出错。

3. 一开始，由于整个系统很多地方要使用字符串的地方，一开始使用费时费力的一长串代码字符串挨个相加的方式，后来发现这样的代码过于冗余。所以改用字符串流的方式，这样可以减少代码的冗余量，使得代码“信达雅”。

想要本项目可以获得良好的运用需要改进以下几点：

- a) 抛弃文件读写来充当数据库，使用专业的数据库软件对内容进行控制。这样可以提高程序的运行效率。同时也可以保证数据库的数据安全。
- b) 完成可视化页面，想要好的效果，就必须有很好的可视化页面，并且要有好的交互逻辑。

## 6 结论

本课程设计旨在设计一个图书馆综合管理系统。使得用户和管理员可以方便快捷，不出错误的高效完成图书馆的一系列工作。

我们设计的很多功能，区分了学生和管理员，使得大家可以各司其职，管理员功能多，学生系统方便快捷，不需要重复输入自己的名字，系统会自动从账号中获取。

我们设计了加密系统，在大家对自己的信息安全很重视的年代，我们使用了目前无法破解的 SHA-256 算法对大家的密码进行加密。由于密码验证只需要单向验证是否正确，并不需要字符串的更高级的搜索，所以我们可以得以使用。相信在未来，本项目可以使用完全数据加密，加密后可以查找和读写操作。

本项目在设计之初，本着不能出 Bug 的原则，成功通过了同组小伙伴们“暴力验证”的测试。确保了系统的稳定性和鲁棒性。同时也感谢各位小组成员的鼎力支持。

## 参考文献

- [1] PicoSHA2 - a C++ SHA256 hash generator, <https://github.com/okdshin/PicoSHA2>
- [2] C++ Reference, <https://cplusplus.com/>
- [3] 启明星辰网络空间安全学院.CTF 安全竞赛入门[M].北京.清华大学出版社.2020:72-124