**ĐẠI HỌC ĐÀ NẴNG**

**ĐẠI HỌC BÁCH KHOA**

**KHOA CÔNG NGHỆ THÔNG TIN**



**PBL2: DỰ ÁN CƠ SỞ LẬP TRÌNH**

**Đề tài: Ứng dụng quản lý sách và người dùng thư viện**

**SINH VIÊN THỰC HIỆN:**

**Trương Tấn Cường LỚP: 21TCLC\_DT1 NHÓM: DHP91**

**Nguyễn Nguyên Anh LỚP: 21TCLC\_DT1 NHÓM: DHP91**

**Ngô Thị Thanh Thúy LỚP: 21TCLC\_DT3 NHÓM: DHP91**

**GIẢNG VIÊN HƯỚNG DẪN: TS. Đặng Hoài Phương**

**Đà Nẵng 26-12-2022**

LỜI MỞ ĐẦU

Ngày nay, Công nghệ thông tin đang trên đà phát triển nhanh chóng, được ứng dụng trong nhiều lĩnh vực khác như kinh tế tài chính, công nghiệp, dịch vụ, quân sự và thậm chí trong cả các lĩnh vực văn hoá nghệ thuật ở khắp nơi trên thế giới. Nhân loại đang không ngừng cố gắng cải tiến và phát triển công nghệ thông tin, nhằm đáp ứng được nhu cầu tiến bộ xã hội, thúc đẩy sự phát triển toàn diện của thế giới.

Một trong những khía cạnh luôn được quan tâm là công tác quản lý. Với sự góp mặt của công nghệ thông tin, hoạt động của các doanh nghiệp trở nên vô cùng thuận tiện, hiệu quả, giúp tăng cao năng suất làm việc, tối ưu hoá nguồn lợi, thúc đẩy phát triển kinh tế chung. Do đó, công nghệ phần mềm cùng các sản phẩm phần mềm, đặc biệt là các ứng dụng quản lý, đã trở thành một trong những yếu tố đi đầu trong công cuộc phát triển kinh tế - xã hội. Sự ra đời các phần mềm ứng dụng quản lý là bước ngoặt trong công tác quản lý hàng hoá, vật tư, nhân sự,…giúp hạn chế tối đa sai sót, thất thoát; ngoài ra, công nghệ còn hỗ trợ trong việc thống kê dữ liệu, hoạch định chiến lược phát triển doanh nghiệp. Có thể nói, sự phát triển của khoa học công nghệ là yếu tố quan trọng thúc đẩy kinh tế phát triển.

Nắm bắt xu hướng đó, nhóm chúng em đã chọn đề tài “Ứng dụng quản lý sách và người dùng thư viện” . Với mục tiêu ứng dụng những kiến thức học phần đã học đưa vào công tác quản lý thực tế, chương trình được phát triển nhằm đáp ứng nhu cầu lưu trữ và quản lý thông tin sách và người dùng thư viện, đồng thời lưu trữ các thông tin mượn sách, hỗ trợ các thao tác tìm kiếm, tra cứu nhanh chóng.

Ngoài phần mở đầu và kết luận, nội dung báo cáo của nhóm chúng em được chia làm 4 chương:

Chương 1: GIỚI THIỆU ĐỀ TÀI

Chương 2: PHÂN TÍCH CHỨC NĂNG HỆ THỐNG

Chương 3: THIẾT KẾ CẤU TRÚC DỮ LIỆU

Chương 4: PHÂN TÍCH HƯỚNG ĐỐI TƯỢNG VÀ TRIỂN KHAI HỆ THỐNG

Chương 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

**MỤC LỤC**

[LỜI MỞ ĐẦU 2](#_Toc122423818)

[DANH MỤC HÌNH VẼ 4](#_Toc122423819)

[DANH MỤC BẢNG BIỂU 5](#_Toc122423820)

[1. GIỚI THIỆU ĐỀ TÀI 6](#_Toc122423821)

[2. PHÂN TÍCH CHỨC NĂNG HỆ THỐNG 7](#_Toc122423822)

[2.1. Định nghĩa và mô tả hệ thống 7](#_Toc122423823)

[2.1.1. Các thực thể có tác động tới hệ thống: 7](#_Toc122423824)

[2.1.2. Mô tả hoạt động của hệ thống: 7](#_Toc122423825)

[2.2. Mô hình nghiệp vụ 8](#_Toc122423826)

[2.2.1. Sơ đồ phân rã chức năng: 8](#_Toc122423827)

[2.2.2. Mô tả chi tiết các chức năng lá 9](#_Toc122423828)

[3. THIẾT KẾ CẤU TRÚC DỮ LIỆU 10](#_Toc122423829)

[3.1. Phát biểu bài toán 10](#_Toc122423830)

[3.2. Phân tích và ứng dụng cấu trúc dữ liệu trong hệ thống 10](#_Toc122423831)

[3.2.1 Danh sách liên kết đôi 10](#_Toc122423832)

[3.2.2 Cài đặt danh sách liên kết đôi 11](#_Toc122423833)

[3.2.3 Danh sách liên kết đôi và các cấu trúc dữ liệu khác 12](#_Toc122423834)

[4. PHÂN TÍCH HƯỚNG ĐỐI TƯỢNG VÀ TRIỂN KHAI HỆ THỐNG 13](#_Toc122423835)

[4.1. Cấu trúc hệ thống hướng đối tượng 13](#_Toc122423836)

[4.2. Kết quả 21](#_Toc122423837)

[4.2.1. Giao diện chính của chương trình 21](#_Toc122423838)

[4.2.2. Kết quả thực thi của chương trình 21](#_Toc122423839)

[4.2.3. Nhận xét 26](#_Toc122423840)

[5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 27](#_Toc122423841)

[a. Kết luận 27](#_Toc122423842)

[b. Hướng phát triển 27](#_Toc122423843)

[TÀI LIỆU THAM KHẢO 28](#_Toc122423844)

DANH MỤC HÌNH VẼ

Figure 2.1 Sơ đồ phân rã chức năng hệ thống 7

Figure 3.1 Mô tả cấu trúc danh sách liên kết đôi 10

Figure 4.1 Sơ đồ các lớp đối tượng trong cấu trúc dữ liệu 12

Figure 4.2 Lớp giao diện & Áp dụng Design Pattern Factory 13

Figure 4.3 Ví dụ Menu gồm có các lựa chọn (item) và số lựa chọn (number) là 4 14

Figure 4.4 Giao diện chính của chương trình 15

Figure 4.5 Sau khi chọn Thu vien sach ở giao diện chính 15

Figure 4.6 Sau khi chọn Danh sach nguoi dung ở giao diện chính 16

Figure 4.7 Sau khi chọn Muon tra sach ở giao diện chính 16

Figure 4.8 Tra cứu danh sách sách 16

Figure 4.9 Thêm sách 17

Figure 4.10 Sửa thông tin sách 17

Figure 4.11 Xóa sách 17

Figure 4.12 Tra cứu danh sách người dùng 17

Figure 4.13 Thêm người dùng 18

Figure 4.14 Sửa thông tin người dùng 18

Figure 4.15 Xóa người dùng 18

Figure 4.16 Mượn sách 18

Figure 4.17 Kiểm tra thông tin mượn sách 19

Figure 4.18 Kiểm tra thông tin mượn sách theo sách 19

Figure 4.19 Kiểm tra thông tin mượn sách theo người dùng 19

Figure 4.20 Trả sách 19

DANH MỤC BẢNG BIỂU

Table 3.1 Bảng so sánh DSLK đôi và một số cấu trúc dữ liệu khác 12

# GIỚI THIỆU ĐỀ TÀI

Hiện nay, khoa học đang phát triển với tốc độ nhanh chóng, đặc biệt trong lĩnh vực Công nghệ Thông tin. Nhu cầu và yêu cầu đặt ra cho các sản phẩm công nghệ thông tin theo đó ngày càng cao. Đối với các hệ thống phần mềm quản lý, cần đáp ứng được các yếu tố sau:

- Thời gian phát triển nhanh chóng

- Chất lượng tốt, vận hành hiệu quả

- Chi phí giá thành thấp

- Thuận tiện, dễ sử dụng

Việc đáp ứng được các yếu tố trên là cần thiết để đảm bảo sự thành công của một dự án phần mềm. Để thực hiện được điều đó, cần lưu ý:

- Phân tích, xác định đúng yêu cầu của hệ thống

- Lựa chọn môi trường, nền tảng phù hợp với yêu cầu

- Lựa chọn thuật toán, cấu trúc dữ liệu phù hợp

- Lựa chọn cơ sở dữ liệu, xây dựng hệ quản trị cơ sở dữ liệu phù hợp với nhu cầu và chi phí

Mỗi ý trên đều có vai trò quan trọng ảnh hưởng tới sản phẩm phần mềm. tuy nhiên, trong học phần PBL2 - Dự án cơ sở lập trình, chúng ta chỉ đi sâu vào phần thuật toán và cấu trúc dữ liệu, sử dụng phong cách lập trình hướng đối tượng. Với mục tiêu đặt ra: áp dụng được những kiến thức đã học, nghiên cứu xây dựng một phần mềm có tính thực tế trong đời sống.

Nắm bắt xu hướng đó, nhóm chúng em chọn đề tài “Ứng dụng quản lý sách và người dùng thư viện”, nhằm hỗ trợ người thủ thư trong việc quản lý sách, độc giả, cũng như các hoạt động mượn - trả sách. Ứng dụng được phát triển bằng ngôn ngữ lập trình C**++**, sử dụng giao diện console.

# PHÂN TÍCH CHỨC NĂNG HỆ THỐNG

## Định nghĩa và mô tả hệ thống

### Các thực thể có tác động tới hệ thống:

* Thủ thư: là người thực hiện công tác quản lý, trực tiếp thao tác với ứng dụng quản lý. Thủ thư quản lý thông tin các đầu mục sách có trong thư viện, quản lý thông tin những người dùng đã đăng kí với thư viện, và quản lý hoạt động mượn - trả sách của những người dùng thư viện.
* Đầu mục sách: ở đây chỉ những tài liệu nói chung, bao gồm sách, bài báo khoa học, luận văn,…được lưu trữ và quản lý trong thư viện. Mỗi sách sẽ có ID cụ thể và duy nhất trong hệ thống, được dán nhãn trên bìa sách
* Người dùng: chỉ những người đã đăng kí sử dụng thư viện. Người dùng sau khi đăng kí sẽ được cấp phát thẻ nhận dạng và ID tương ứng. Thông tin của người dùng được lưu trong hệ thống. Người dùng sử dụng thẻ nhận dạng và ID để thực hiện việc mượn - trả sách từ thư viện.

### Mô tả hoạt động của hệ thống:

Khi thư viện nhập về các đầu mục sách mới, người thủ thư thực hiện nhập thông tin các đầu mục sách mới vào hệ thống. Hệ thống sẽ kiểm tra xem sách với thông tin đã nhập có tồn tại sẵn trong thư viện hay chưa. Nếu đã tồn tại //thì cộng dồn số bản sao của sách đó. Ngược lại tạo mục lưu trữ mới, cấp ID cho đầu mục sách mới.

Trong quá trình vận hành, nếu phát hiện sai sót về thông tin trong bản ghi, người thủ thư có thể chỉnh sửa bản ghi thông qua ID được cấp của đầu mục sách. Nếu có một bản sao nào đó bị thất lạc hoặc không thể thu hồi, cần chỉnh sửa lại số bản sao trong bản ghi hệ thống.

Khi một người muốn đăng kí trở thành người dùng thư viện, họ cần cung cấp thông tin cho thủ thư. Thủ thư sẽ kiểm tra xem người này đã đăng kí trước đó hay chưa, nếu đã đăng kí, thủ thư thông báo và tiến hành cấp lại thẻ nếu người đó sơ ý làm mất thẻ, quên tài khoản,…, nếu chưa thì tiến hành đăng kí mới và cấp thẻ.

Khi một người dùng muốn mượn hoặc trả sách, người đó cần mang theo thẻ người dùng, và sách muốn mượn/trả đến quầy. Thủ thư ghi nhận thông tin ID sách và ID người dùng để tiến hành lập đơn mượn/thực hiện thủ tục trả sách trên hệ thống.

## Mô hình nghiệp vụ

### Function DiagramSơ đồ phân rã chức năng:

Figure 2.1 Sơ đồ phân rã chức năng hệ thống

### Mô tả chi tiết các chức năng lá

1.1. Thêm sách mới: Dựa vào thông tin của sách mới nhập về thư viện, thêm các bản ghi đầu mục sách vào hệ thống.

1.2. Chỉnh sửa thông tin sách: Nếu có bất cứ thay đổi nào về thông tin hay số bản sao đầu mục sách lưu trữ trong thư viện, cần cập nhật bản ghi trong hệ thống.

1.3. Xoá sách: Nếu có sách nào đó không còn bản sao nào hoặc cần được xoá bỏ khỏi thư viện, thực hiện xoá bản ghi đầu mục sách trong hệ thống.

2.1. Thêm người dùng mới: Đăng kí cho một người trở thành người dùng, lưu thông tin của người dùng đó vào hệ thống.

2.2. Chỉnh sửa thông tin người dùng: Nếu người dùng có thay đổi về thông tin cá nhân có ảnh hưởng đến lưu trữ trong hệ thống, cần cập nhật lại thông tin của người dùng đó.

2.3. Xoá người dùng: Nếu người dùng không hoạt động trong thời gian dài, không thể liên lạc, cần cân nhắc xoá người dùng khỏi hệ thống.

3.1. Lập đơn mượn: Thực hiện thủ tục cho một người dùng mượn sách, lưu trữ lại vào hệ thống.

3.2. Kiểm tra thông tin mượn/trả: Kiểm tra một người dùng nào đó đang mượn những cuốn sách nào, và một đầu mục sách có các bản sao đang được mượn bởi những người dùng nào

3.3. Trả sách: Thực hiện thủ tục cho một người dùng trả sách.

# THIẾT KẾ CẤU TRÚC DỮ LIỆU

## Phát biểu bài toán

* Xây dựng chương trình quản lý sách và người dùng trong thư viện sử dụng ngôn ngữ C**++** và cấu trúc dữ liệu kiểu danh sách liên kết đôi.
* Input: Đọc từ file sachbase.txt, userbase.txt, borrowing.txt thông tin về sách có trong thư viện, người dùng đã mượn sách và danh sách sách mượn.
* Cấu trúc file sachbase.txt:

Gồm các dòng cho trước với mỗi dòng có cấu trúc

<id sách>|<tên sách>|<tên tác giả>|<số bản>

* Cấu trúc file userbase.txt:

Gồm các dòng cho trước với mỗi dòng có cấu trúc:

<id người dùng>|<tên người dùng>|<Số CMND>

* Cấu trúc file borrowing.txt:

Gồm các dòng cho trước với mỗi dòng có cấu trúc:

<id sách>|<id người dùng>

* Output: Ghi lại vào các file trên nội dung sau khi người dùng sử dụng chương trình

## Phân tích và ứng dụng cấu trúc dữ liệu trong hệ thống

### 3.2.1 Danh sách liên kết đôi

- **Con trỏ** là một biến dùng để lưu trữ địa chỉ của một biến khác. Biến khác có thể là kiểu số nguyên, số thực, ký tự, xâu, một mảng, một hàm hay một kiểu dữ liệu do người dùng tự định nghĩa và nó cũng có thể dùng để trỏ tới địa chỉ của một con trỏ khác. [1]

- **Danh sách liên kết** là một cấu trúc dữ liệu cho phép quản lý một nhóm các node tạo thành một chuỗi. Mỗi node gồm có thông tin và con trỏ để trỏ đến node kế tiếp. Có 3 loại danh sách liên kết cơ bản:

**+** Danh sách liên kết đơn

**+** Danh sách liên kết đôi

**+** Danh sách liên kết vòng

Ngoài 3 kiểu danh sách liên kết trên, ta còn có các biến thể khác của danh sách liên kết bằng cách kết hợp chúng với nhau.

- **Danh sách liên kết đôi** là một danh sách liên kết trong đó mỗi node mang thông tin và 2 con trỏ: con trỏ trỏ đến node kế tiếp và con trỏ trỏ đến node ngay trước nó.

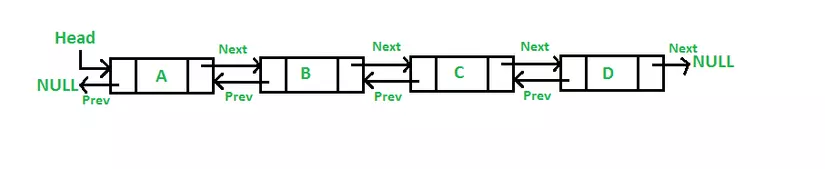


Figure 3.1 Mô tả cấu trúc danh sách liên kết đôi

### 3.2.2 Cài đặt danh sách liên kết đôi

- Danh sách liên kết đôi trong hướng đối tượng:

template <class U>

class Node

{

private:

U data;

Node \*next; //Con trỏ trỏ đến node tiếp theo

Node \*prev; //Con trỏ trỏ đến node liền trước

public:

//Các phương thức quản lý Node

};

template <class T>

class DSLK

{

private:

int size;

T \*head; //Con trỏ trỏ đến node đầu danh sách

T \*tail; //Con trỏ trỏ đến node cuối danh sách

public:

//Các phương thức quản lý DSLK

};

### 3.2.3 Danh sách liên kết đôi và các cấu trúc dữ liệu khác

|  |  |  |
| --- | --- | --- |
| **Danh sách liên kết đôi** | **Danh sách liên kết đơn** | **Mảng** |
| Các phần tử được cấp phát vùng nhớ không nhất thiết kề nhau | Tương tự danh sách liên kết đôi | Các phần tử được cấp phát vùng nhớ kề nhau |
| Thêm hoặc xóa phần tử không cần cấp phát lại vùng nhớ cho các phần tử đã có | Tương tự danh sách liên kết đôi | Thêm phần tử vượt quá kích thước mảng phải cấp phát lại vùng nhớ cho cả mảng |
| Muốn truy cập phần tử phải duyệt từ đầu hoặc cuối danh sách | Muốn truy cập phần tử phải duyệt từ đầu danh sách | Có thể truy cập phần tử bất kỳ mà không cần duyệt danh sách |
| Cài đặt phức tạp | Cài đặt phức tạp | Cài đặt dễ dàng |

Table 3.1 Bảng so sánh DSLK đôi và một số cấu trúc dữ liệu khác

Khi chương trình cần lưu danh sách với số lượng rất lớn phần tử, bộ nhớ có thể không đảm bảo đủ vùng nhớ liên tiếp nhau để ta có thể sử dụng mảng. Vì vậy trong chương trình quản lý thư viện đang được xây dựng, sử dụng danh sách liên kết đôi sẽ cho ta lợi về mặt bộ nhớ khi ta có thể cấp phát vùng nhớ không kề nhau cho các phần tử có trong danh sách. Bên cạnh đó, sử dụng cấu trúc dữ liệu này còn cho phép ta duyệt từ đầu hoặc cuối danh sách tăng tốc độ truy cập.

# PHÂN TÍCH HƯỚNG ĐỐI TƯỢNG VÀ TRIỂN KHAI HỆ THỐNG

## Class DiagramCấu trúc hệ thống hướng đối tượng

Figure 4.1 Sơ đồ các lớp đối tượng trong cấu trúc dữ liệu

- Lớp Node: là lớp template gồm các Node trong danh sách liên kết đôi, có phần dữ liệu **data** thuộc kiểu dữ liệu bất kì đảm bảo được một số điều kiện; **next** và **prev** là con trỏ trỏ đến Node kề trước hoặc kề sau nó.

Các phương thức của lớp Node:

+ Hàm tạo **Node(const U&)** tạo Node từ một đối tượng thuộc kiểu dữ liệu U;

+ Hàm huỷ ~Node() trả con trỏ **next** và **prev** về NULL.

+ **toNext(), toPrev()** lần lượt trả về **next** và **prev**, để lấy địa chỉ Node tiếp theo hoặc Node liền trước

+ **setNext(Node\*), setPrev(Node\*)** dùng để thay đổi **next** và **prev** trong các thủ tục Insert, Remove

+ **getData()** trả về tham chiếu tới đối tượng **data**, cho phép gọi các phương thức tương ứng của kiểu dữ liệu U

- Lớp DSLK: là lớp template quản lý một danh sách các Node có **data** thuộc cùng kiểu dữ liệu, ở đây kí hiệu là T, với T = Node<U>; **size** lưu độ dài của danh sách, **head** và **tail** lần lượt là con trỏ trỏ tới Node đầu danh sách và Node cuối danh sách.

\*Để có thể thực hiện các phương thức của lớp DSLK, lớp kiểu dữ liệu U cần đảm bảo các phương thức và toán tử sau:

+ **getID()** có kiểu trả về string

+ **update()** có kiểu trả về void, cho phép nhập và thay đổi dữ liệu từ bàn phím

+ **operator == (const U&)** có kiểu trả về bool, thực hiện so sánh bằng giữa hai đối tượng cùng lớp

Các phương thức của lớp DSLK:

+ **DSLK()** khai báo một danh sách rỗng

+ **~DSLK()** thực hiện giải phóng các Node

+ **getSize(), getHead(), getTail()** lần lượt trả về **size, head, tail** của đối tượng

+ **find<U>(U& input)** trả về con trỏ trỏ tới Node có **data == input**, nếu không có Node thoả mãn trả về NULL, được sử dụng trong các phương thức **insert** và **remove** của lớp template DSLK

+ **insert<U>(U& input)** thêm một Node mới vào danh sách và trả về 1, Node mới có **data = input**, nếu đã có Node nào đó có **data == input** (f**ind(input)!=NULL**) thì không thực hiện thêm vào, trả về 0

+ **remove<U>(U& input)** xoá một Node có **data == input** và trả về 1, nếu không có Node nào thoả mãn thì không thực hiện xoá và trả về 0

+ **find\_id(const string& input)** tìm một Node nào đó có phần **data.getID() == input** và trả về tham chiếu đến **data** của Node đó. Phương thức này được gọi bởi các hàm bên ngoài lớp template DSLK, dùng tìm kiếm bản ghi dựa trên mã ID.

- Lớp Sach: là lớp gồm các đối tượng lưu dữ liệu của một đầu mục sách. Các thuộc tính của một đối tượng bao gồm **tenSach** - tên sách, **tenTacGia** - tên tác giả, **ma** - mã ID của đầu mục sách, **soBan** - số bản sao của sách đó trong thư viện; ngoài ra có một đối tượng **list** là một DSLK<Node<User\*>>: danh sách liên kết quản lý các con trỏ kiểu User, với User là lớp gồm các đối tượng người dùng. **list** có vai trò quản lý danh sách những người dùng đang mượn các bản sao của đầu sách này.

Các phương thức:

+ **Sach(string, string, int)** tạo một đối tượng Sach mới với tham số truyền vào gồm tên sách, tên tác giả và số bản sao, ID sẽ được cập nhật tự động

+ **update()** thực hiện thay đổi thông tin tên sách, tên tác giả, số bản, nhập từ bàn phím

+ **operator == (const Sach&)** trả về 1 nếu hai đối tượng Sach trùng tên sách và tên tác giả, nếu không trả về 0

+ **setID(const string&), getID()** lần lượt gán mã ID và lấy mã ID của sách

+ **getList()** trả về tham chiếu tới **list**, để thực hiện các thao tác thuộc lớp DSLK

+ **borrowList()** in danh sách người dùng có mượn các bản sao của đối tượng sách này ra màn hình

- Lớp User: là lớp gồm các đối tượng người dùng. Các thuộc tính của một đối tượng gồm **tên -**  tên người dùng, **cmd -** mã số CMND, **uid** - mã ID được cấp bởi thư viện, ngoài ra có đối tượng **list** là một DSLK<Node<Sach\*>>: danh sách liên kết quản lý các con trỏ kiểu Sach, lưu danh sách những đầu sách mà người dùng này đang mượn.

Các phương thức:

+ **User(string, string)** tạo một đối tượng User mới với tham số truyền vào gồm tên người dùng, và số CMND, ID sẽ được cập nhật tự động

+ **update()** thực hiện thay đổi thông tin tên người dùng, số CMND, nhập từ bàn phím

+ **operator == (const Sach&)** trả về 1 nếu hai đối tượng User trùng số CMND, ngược lại trả về 0

+ **setID(const string&), getID()** lần lượt gán mã ID và lấy mã ID của user

+ **getList()** trả về tham chiếu tới **list**, để thực hiện các thao tác thuộc lớp DSLK

+ **borrowList()** in danh sách sách mà đối tượng người dùng này đang mượn ra màn hình

Nhận xét về quan hệ giữa các lớp đối tượng:

- User - Sach có quan hệ nhiều nhiều: một người dùng có thể mượn nhiều sách, một đầu mục sách có thể có nhiều bản sao được nhiều người mượn tại một thời điểm

- Khuôn Node có 4 thể hiện:

+ Node<Sach>: Node chứa một đối tượng sách

+ Node<User>: Node chứa một đối tượng người dùng

+ Node<Sach\*>: Node chứa một con trỏ đến đối tượng sách khác

+ Node<User\*>: Node chứa một con trỏ đến đối tượng người dùng khác

- Tương ứng, khuôn DSLK có 4 thể hiện:

+ DSLK<Node<Sach>>: Lớp danh sách quản lý các đối tượng sách

+ DSLK<Node<User>>: Lớp danh sách quản lý các đối tượng người dùng

+ DSLK<Node<Sach\*>>: Lớp danh sách quản lý các con trỏ tới các đối tượng sách, dùng truy cập nhanh các sách mà một người dùng mượn

+ DSLK<Node<User\*>>: Lớp danh sách quản lý các con trỏ tới các đối tượng người dùng, dùng truy cập nhanh các người dùng đang mượn một đầu sách

\*Một số hàm không thành viên & toán tử đa năng hoá không thành viên:

Các toán tử hỗ trợ việc nhập xuất qua màn hình console, nhập xuất qua file đối tượng sách:

**operator >> (istream&, Sach&); operator << (ostream&, const Sach&)**

**operator >> (ifstream&, Sach&); operator << (ofstream&, const Sach&)**

Đối tượng người dùng cũng có các toán tử được đa năng hoá tương ứng:

**operator >> (istream&, User&); operator << (ostream&, const User&)**

**operator >> (ifstream&, User&); operator << (ofstream&, const User&)**

Ngoài ra, có một số hàm cần sử dụng trong quá trình nhập liệu từ màn hình:

+ **string increment(string&)** nhận vào một chuỗi chỉ gồm các kí tự số, trả về chuỗi kề sau (ví dụ: increment(“00080”) sẽ trả về “00081”) dùng cấp ID tự động

+ **bool checkcmd(string&)** nhận vào một chuỗi, kiểm tra có phải số CMND hợp lệ: chứa đúng 9 chữ số. Nếu hợp lệ trả về 1, ngược lại trả về 0

Các hàm đọc/ghi danh sách từ file:

**+ void readFile(ifstream&, DSLK<Node<Sach>>&)**

**+ void readFile(ifstream&, DSLK<Node<User>>&)**

**+ void readBorrowlist(ifstream&, DSLK<Node<User>>&, DSLK<Node<Sach>>&)**

**+ void writeFile(ofstream&, DSLK<Node<User>>&)**

**+ void writeFile(ofstream&, DSLK<Node<Sach>>&)**

**+ void saveBorrowlist(ofstream&, DSLK<Node<User>>&)**

Các hàm phục vụ thủ tục mượn/trả:

**+ bool borrowBook(User&,Sach&)**

**+ bool returnBook(User&,Sach&)**

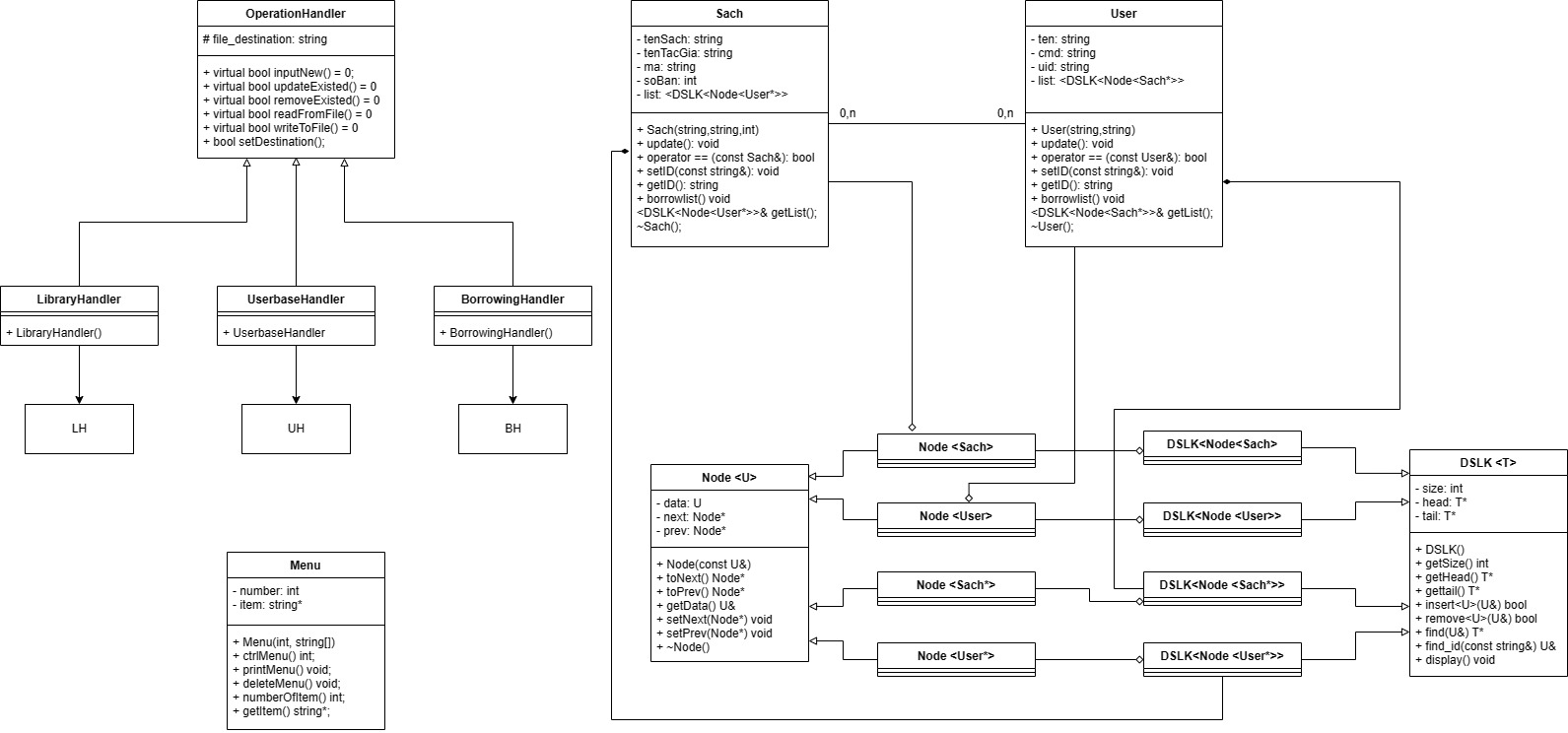


Figure 4.2 Lớp giao diện & Áp dụng Design Pattern Factory

- Lớp Menu là lớp dùng để thao tác với giao diện và in Menu cho người dùng lựa các hành động đối với chương trình. Trong đó:

**Thuộc tính:**

**+ number**: Số lựa chọn của một menu (kiểu int)

**+ item**: Các lựa chọn (mảng string)

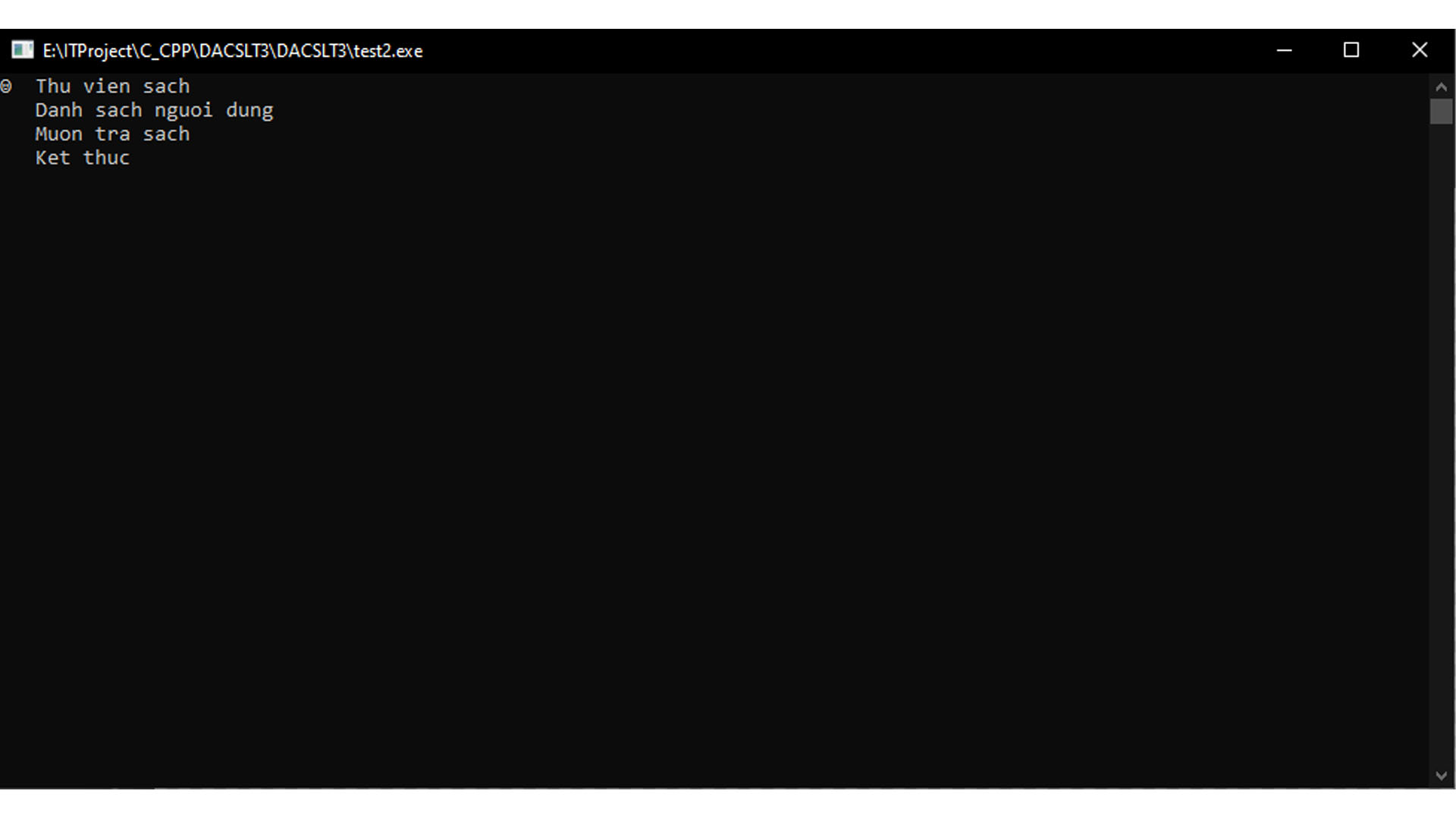


Figure 4.3 Ví dụ Menu gồm có các lựa chọn (item) và số lựa chọn (number) là 4

**Phương thức:**

**+ Menu()**: khởi tạo menu với n lựa chọn được lưu trong mảng string

**+ ctrlMenu():** Phương thức để người dùng điều khiển menu. Cho phép người dùng sử dụng các phím mũi tên để có thể di chuyển giữa các lựa chọn trong menu. Phương thức sau đó trả về giá trị là dòng của Menu mà người dùng chọn để tiếp tục thao tác với chương trình.

**+ printMenu():** In ra các lựa chọn của đối tượng menu được sử dụng.

**+ deleteMenu():** Xóa menu

**+ numberOfitem():** Trả về số lượng các lựa chọn của một menu

**+ getItem():** Kiểu mảng string, trả về this -> item

Để điều khiển menu sử dụng được các phím mũi tên, có thể sử dụng các hàm và cấu trúc có sẵn trong thư viện windows.h của C**++**.

**Các cấu trúc**:

**+ CONSOLE\_CURSOR\_INFO**: Cấu trúc mang thông tin của con trỏ chuột. Gồm 2 thuộc tính:

* dwSize: Kiểu DWORD (unsigned long). Tỷ lệ phần trăm của ô ký tự được lấp đầy bởi con trỏ chuột. Giá trị nằm từ 1 đến 100.
* bVisible: Kiểu bool. Khả năng hiển thị của con trỏ. Nếu con trỏ hiển thị, thành viên này là TRUE.

**+ HANDLE**: Con trỏ void để điều khiển các handle của cửa sổ

**+ COORD**: Cấu trúc gồm 2 thuộc tính kiểu SHORT là X, Y để biểu diễn tọa độ của phần tử trên cửa sổ

**Các hàm**:

**+ SetConsoleCursorPosition(HANDLE, COORD)**: Truyền vào hai tham số kiểu HANDLE và COORD để đưa con trỏ nhấp nháy tới vị trí X, Y trên cửa sổ. Trả về kiểu bool.

**+ GetConsoleCursorInfo(HANDLE, CONSOLE\_CURSOR\_INFO\*):** Truy xuất thông tin về kích thước và khả năng hiển thị của con trỏ ra màn hình

**+ SetConsoleCursorInfo(HANDLE, CONSOLE\_CURSOR\_INFO\*):** Đặt kích thước và khả năng hiển thị của con trỏ ra màn hình

- Lớp OperationHandler: là lớp trừu tượng, đại diện cho một *nhánh nghiệp vụ*, quản lý các thao tác nghiệp vụ của hệ thống. Một *nhánh nghiệp vụ* có thể và chỉ thực hiện các thao tác nghiệp vụ trên một nhánh chức năng của mô hình nghiệp vụ (xem hình 2.1. Sơ đồ phân rã chức năng hệ thống). Theo đó, sẽ có 3 lớp con kế thừa lớp trừu tượng này:

+ Lớp LibraryHandler: có các phương thức thực hiện các thao tác nghiệp vụ quản lý sách

+ Lớp UserbaseHandler: có các phương thức thực hiện các thao tác nghiệp vụ quản lý người dùng

+ Lớp BorrowingHandler: có các phương thức thực hiện các thao tác nghiệp vụ quản lý mượn sách

Mỗi đối tượng thuộc lớp có một thuộc tính **fileDestination** kiểu string, chứa địa chỉ file sẽ thực hiện thao tác đọc dữ liệu/ghi dữ liệu. Địa chỉ này được đặt mặc định trong hàm tạo của mỗi lớp, và có thể thay đổi bằng phương thức **setDestination().**

Áp dụng Design pattern Factory [1], chương trình thực hiện quản lý và thao tác với các đối tượng thuộc 3 lớp con của OperationHandler, tuỳ vào yêu cầu của người dùng. Tuy nhiên việc tồn tại nhiều hơn 1 đối tượng của các lớp trên là không cần thiết, ngoài ra việc lặp lại chu trình tạo đối tượng -> thao tác -> huỷ đối tượng sẽ trở nên rườm rà. Do đó, trong chương trình 3 đối tượng LH, UH, BH được khởi tạo ngay từ lúc bắt đầu, lần lượt thuộc lớp LibraryHandler, UserbaseHandler, BorrowingHandler. Tuỳ vào lựa chọn của người dùng, hàm **switcher()** sẽ chuyển trả về địa chỉ tới đối tượng tương ứng, lưu bởi một biến con trỏ **handler,** kiểu OperationHandler. Các thao tác nghiệp vụ được gọi bởi đúng một biến **handler**, và thao tác thực hiện tuỳ vào thời điểm hoàn cảnh gọi hàm.

## Kết quả

### Giao diện chính của chương trình

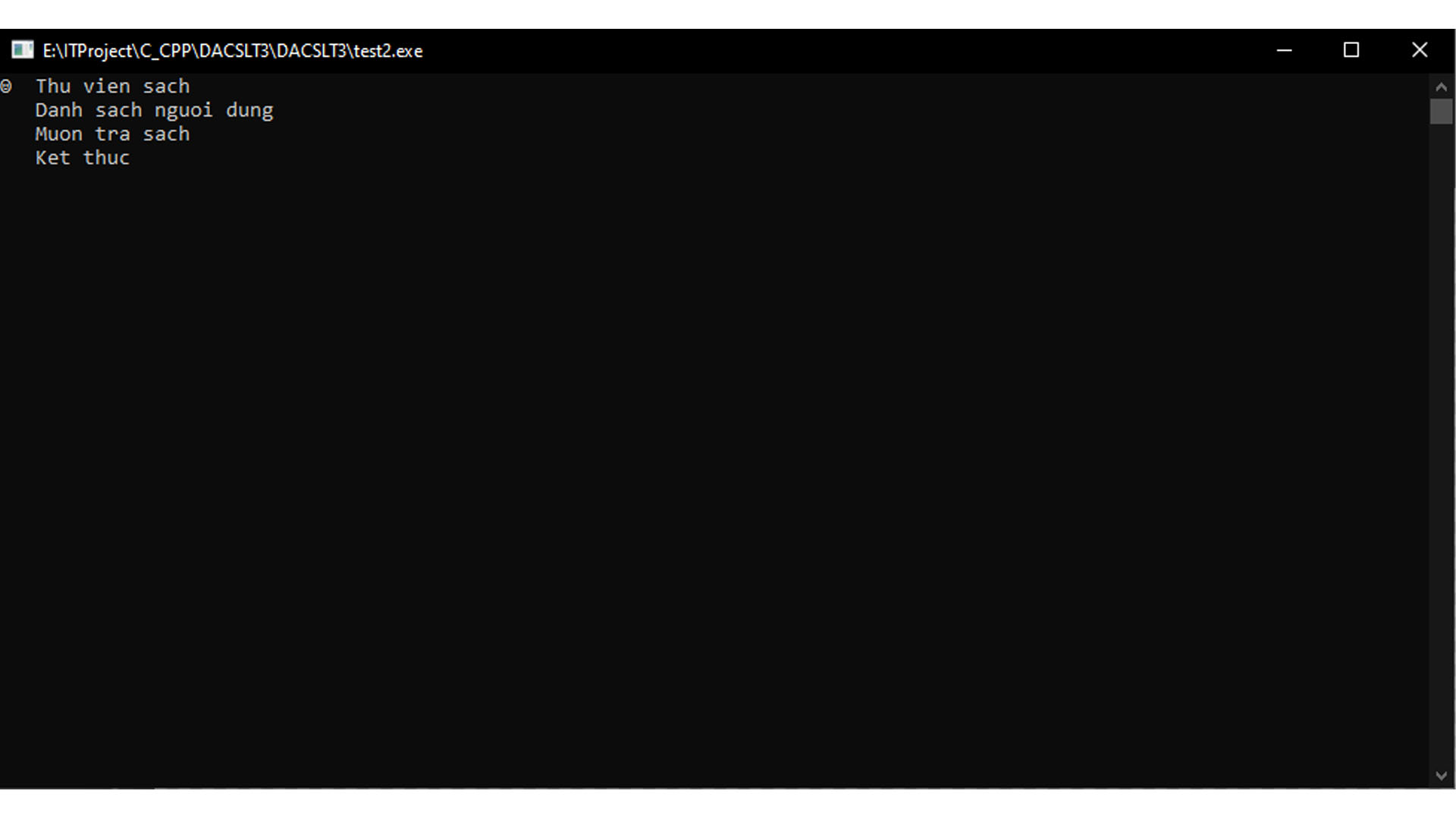


Figure 4.4 Giao diện chính của chương trình

### Kết quả thực thi của chương trình



Figure 4.5 Sau khi chọn Thu vien sach ở giao diện chính

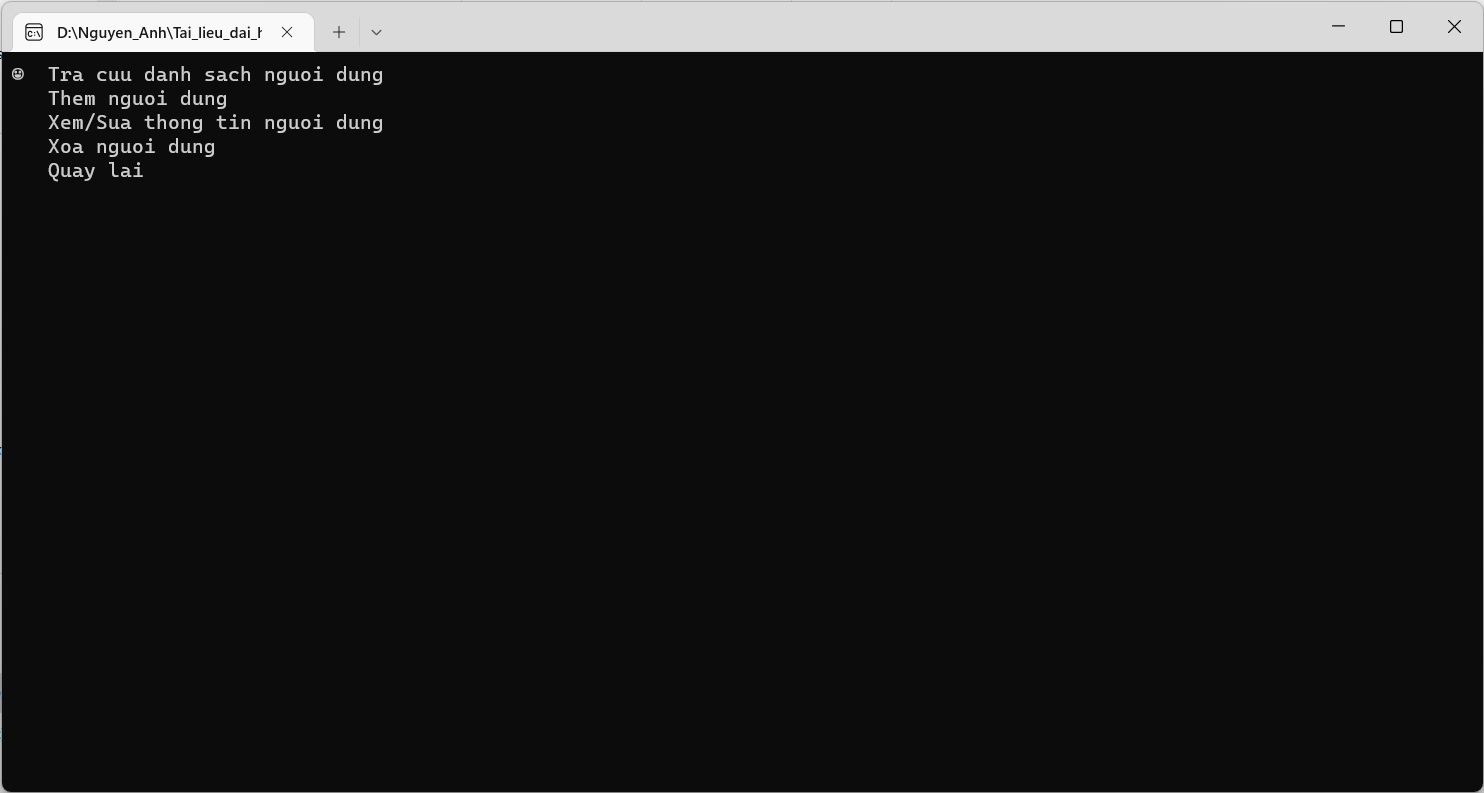


Figure 4.6 Sau khi chọn Danh sach nguoi dung ở giao diện chính

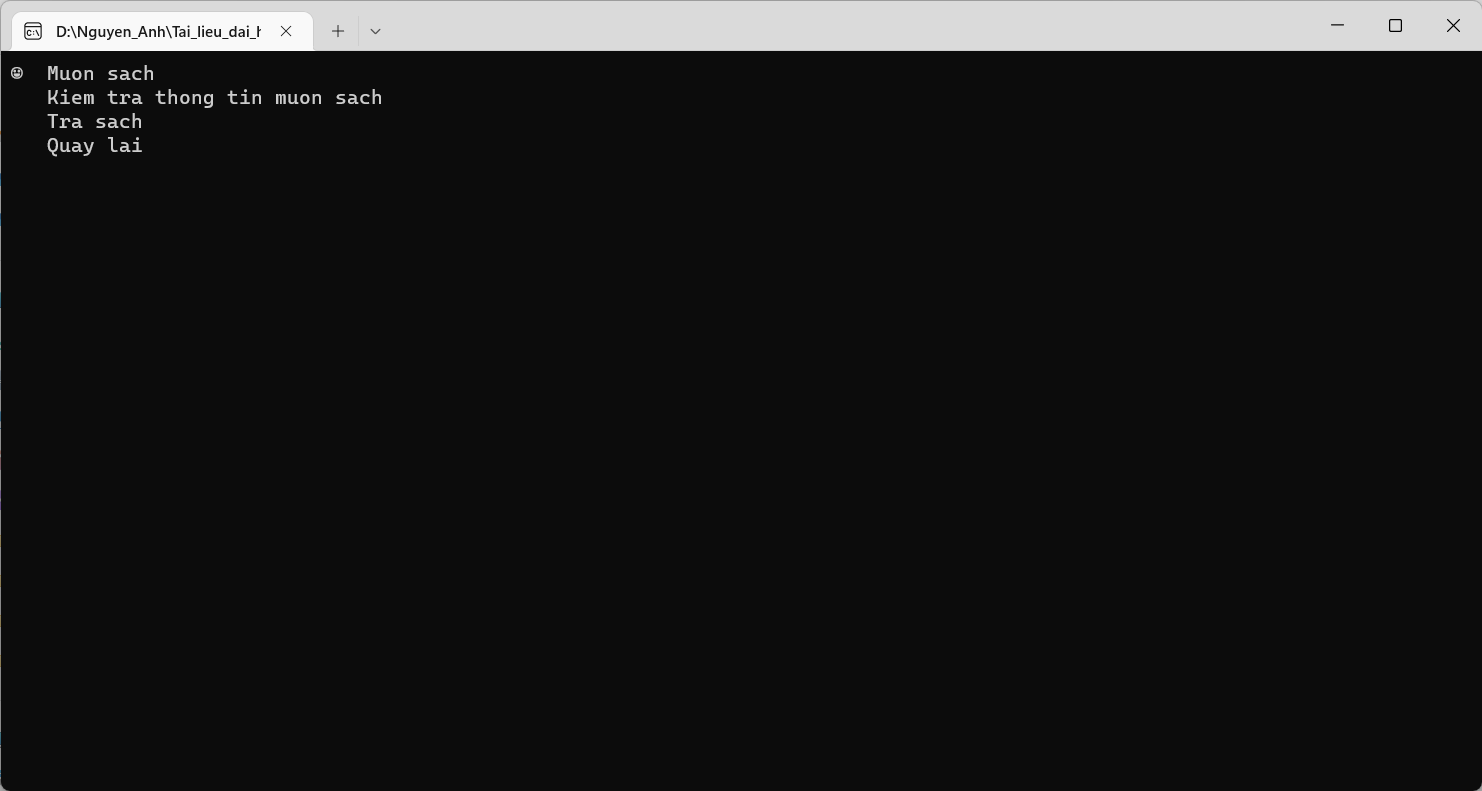


Figure 4.7 Sau khi chọn Muon tra sach ở giao diện chính

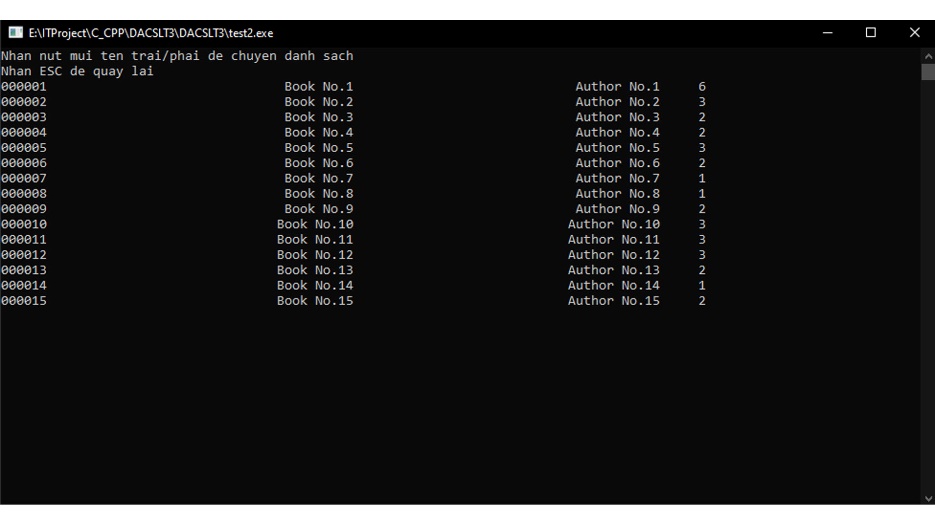


Figure 4.8 Tra cứu danh sách sách



Figure 4.9 Thêm sách



Figure 4.10 Sửa thông tin sách



Figure 4.11 Xóa sách

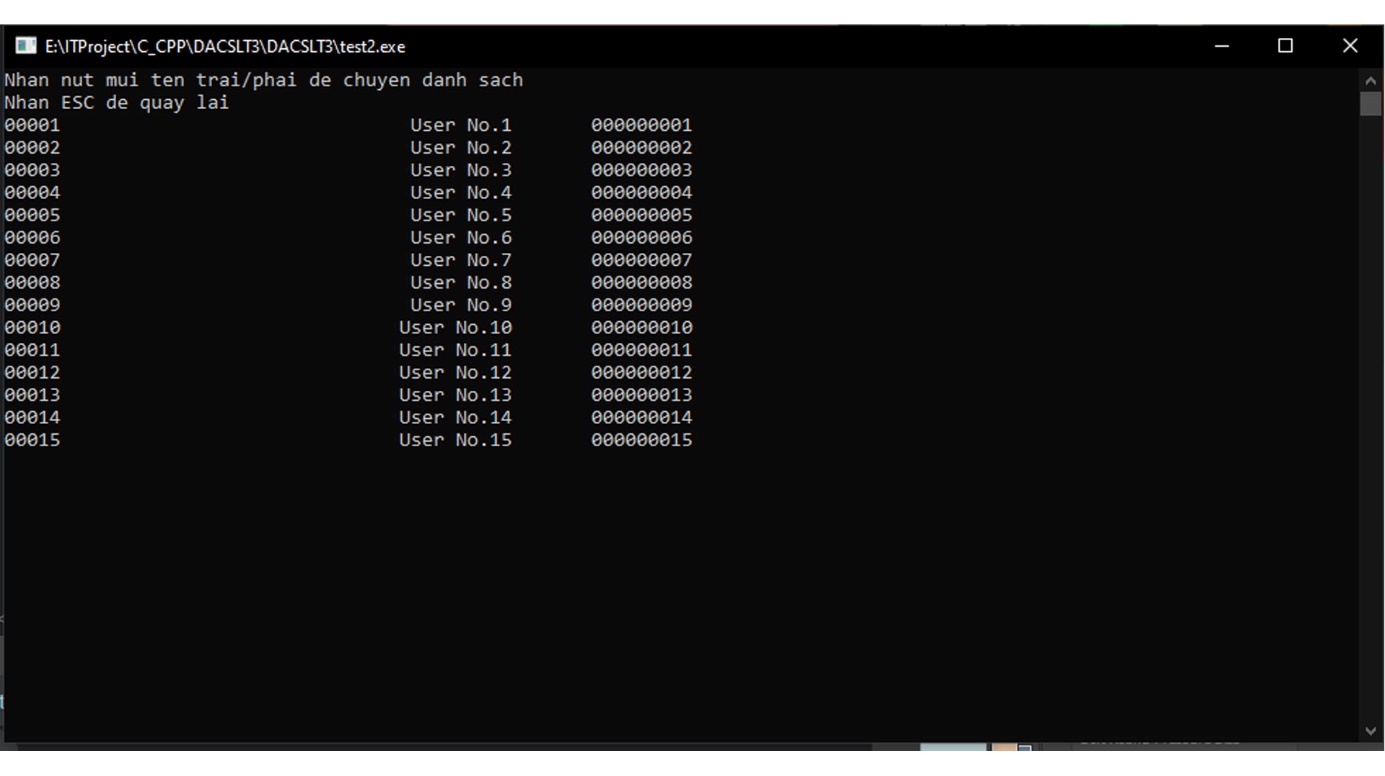


Figure 4.12 Tra cứu danh sách người dùng



Figure 4.13 Thêm người dùng

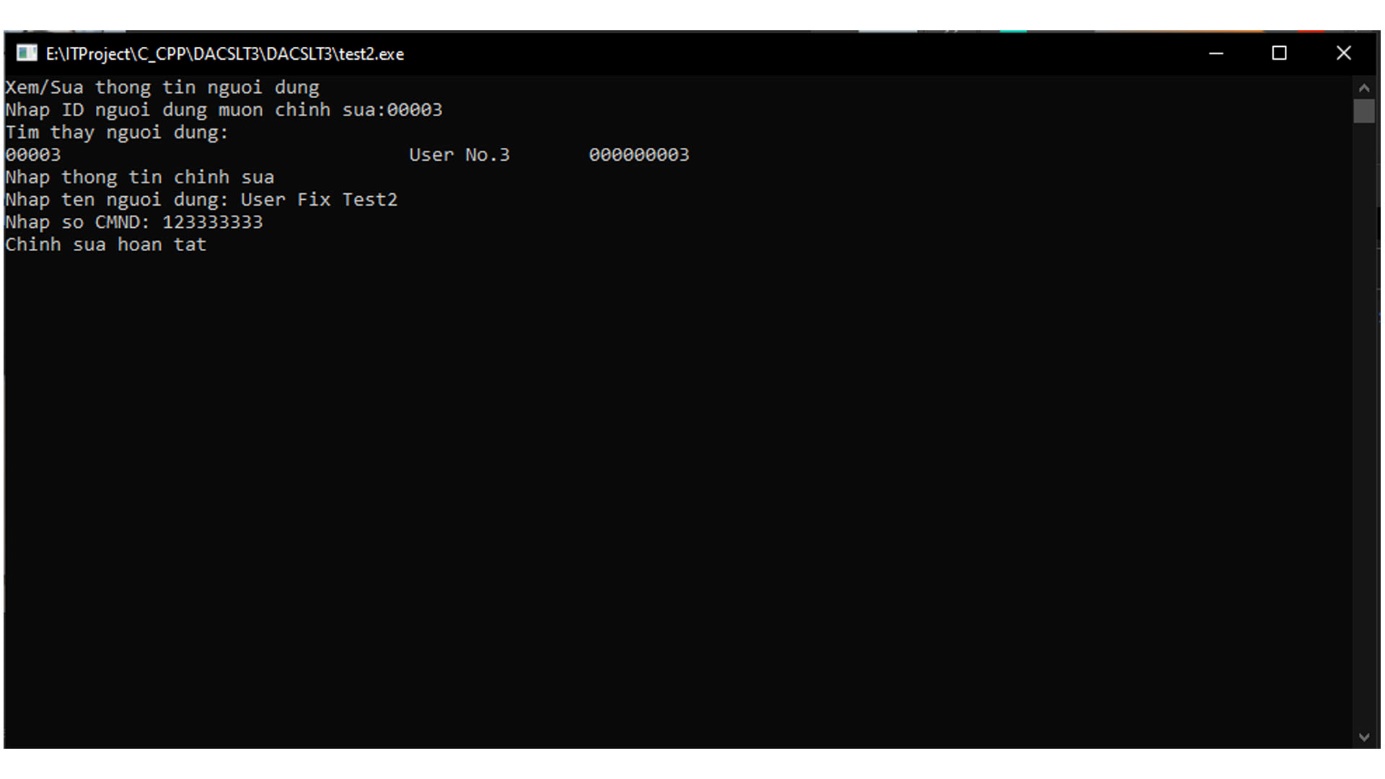


Figure 4.14 Sửa thông tin người dùng

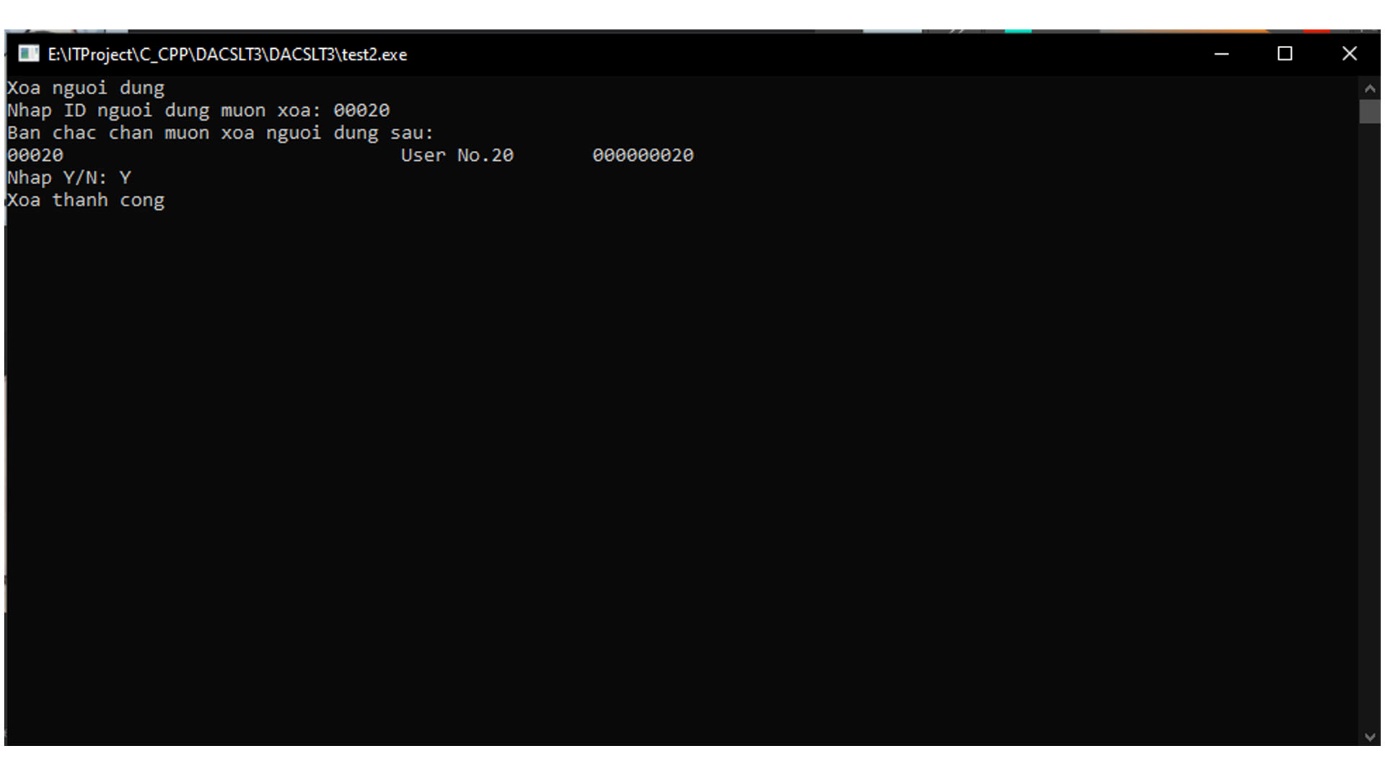


Figure 4.15 Xóa người dùng

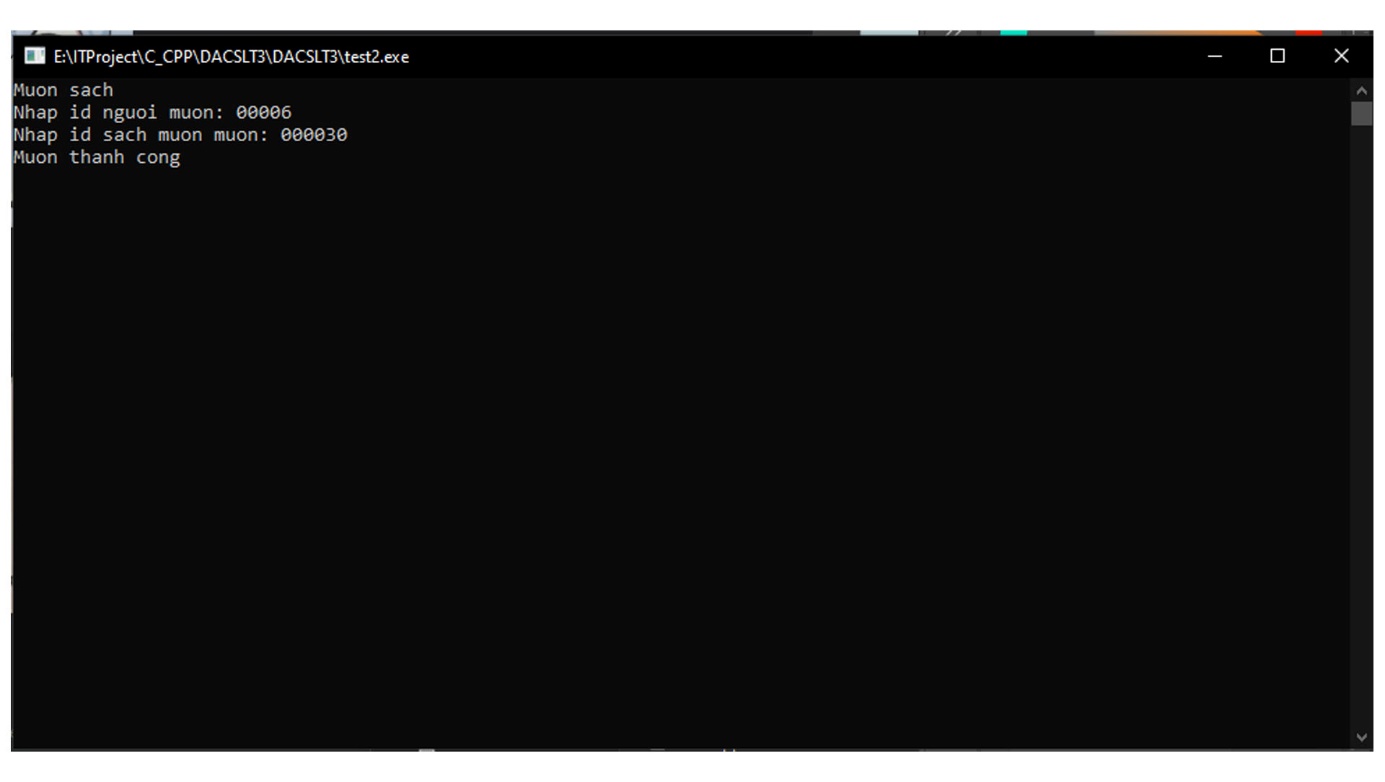


Figure 4.16 Mượn sách



Figure 4.17 Kiểm tra thông tin mượn sách

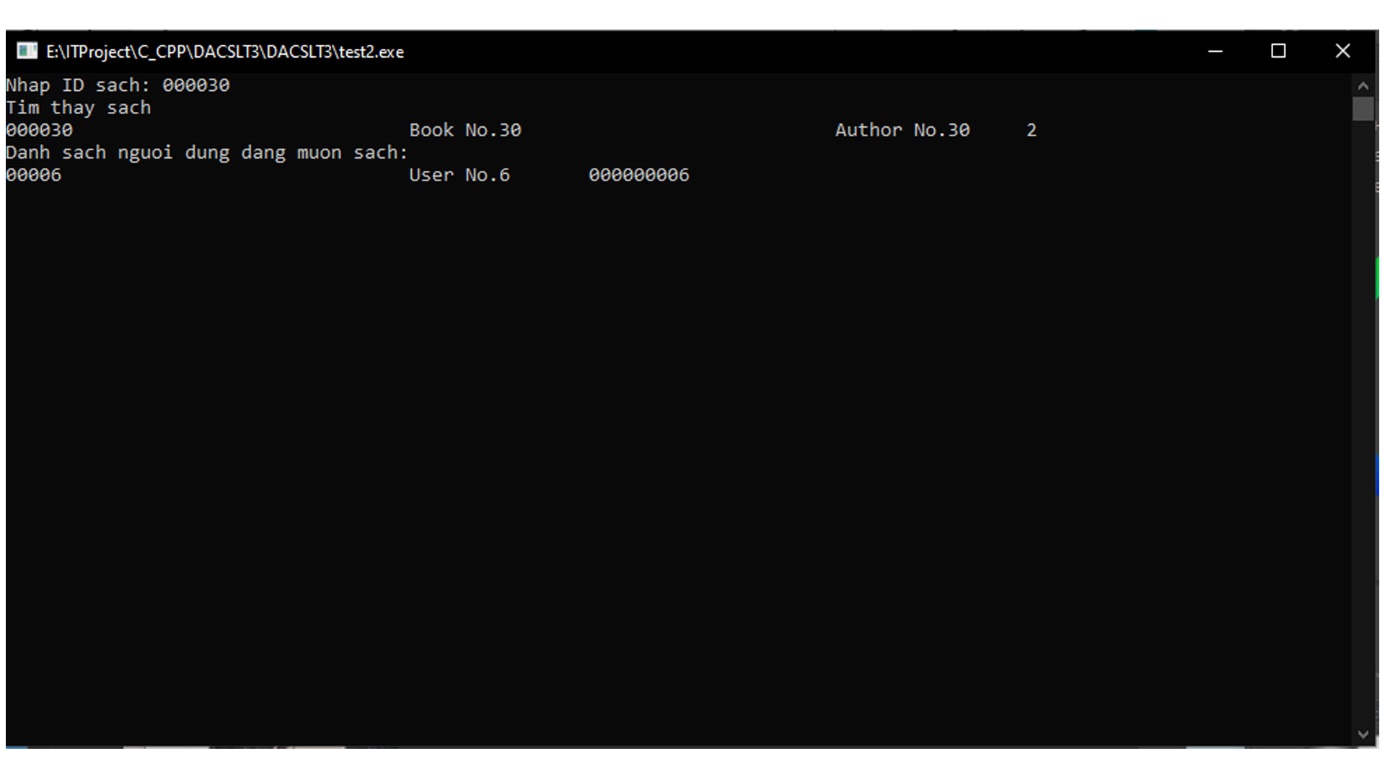


Figure 4.18 Kiểm tra thông tin mượn sách theo sách

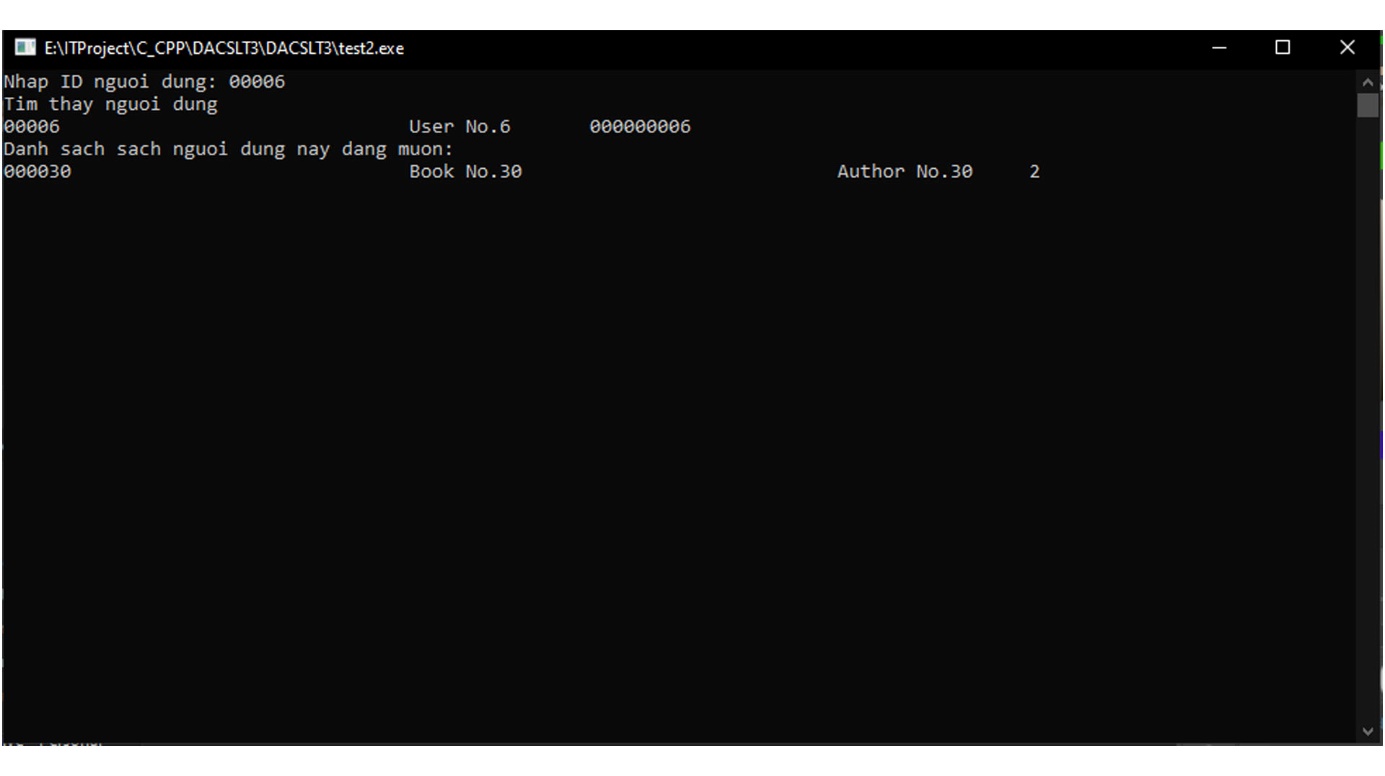


Figure 4.19 Kiểm tra thông tin mượn sách theo người dùng

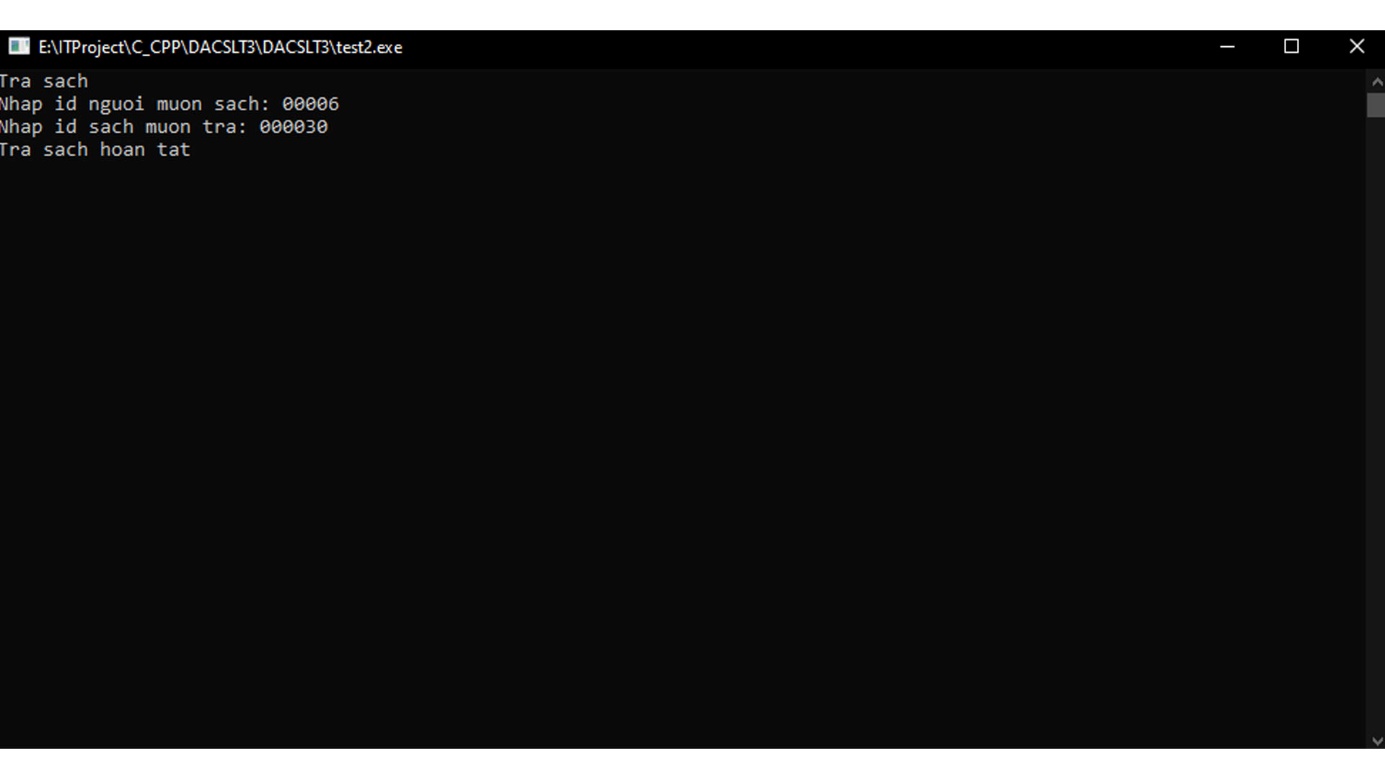


Figure 4.20 Trả sách

### Nhận xét

- Chương trình thực hiện đúng những yêu cầu đặt ra

- Các thao tác chưa được thuận tiện, chỉ có phương thức tìm kiếm bằng ID

- Các thao tác tìm kiếm, thêm, sửa, xoá danh sách đều có độ phức tạp O(n)

- Giao diện chương trình còn chưa được đẹp mắt

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

* + Ưu điểm của chương trình:

- Chương trình phần mềm được lập trình theo phương pháp hướng đối tượng, ứng dụng các tính chất: đóng gói, kế thừa, đa hình và trừu tượng.

- Chương trình đáp ứng được các thao tác cơ bản của việc quản lý: thêm, sửa, xoá, tra cứu và lưu trữ thông tin sách, thông tin người dùng, thông tin đơn mượn.

- Việc sử dụng phương pháp lập trình hướng đối tượng giúp mô hình hoá các hệ thống phức tạp thành tập hợp cấu trúc đơn giản hơn; giúp dễ dàng chỉnh sửa, thay đổi trong trường hợp cần cập nhật, thuận tiện cho việc mở rộng, phát triển trong tương lai; cung cấp tính bảo mật thông tin, an toàn dữ liệu.

* + Hạn chế:

- Tính chuyên nghiệp chưa cao: chưa có tính năng tìm kiếm theo tên, sắp xếp, thống kê; chưa ứng dụng cơ sở dữ liệu, do đó quản lý dữ liệu có phần chưa chặt chẽ.

- Tính thẩm mỹ chưa tốt: sử dụng giao diện console không bắt mắt, không trực quan, gây khó khăn trong quá trình sử dụng

## Hướng phát triển

- Mở rộng thông tin quản lý trong các bản ghi, phân loại sách theo danh mục, phân nhóm người dùng theo độ tuổi, ngành nghề,…

- Xây dựng các tính năng bổ sung, các tính năng thống kê

- Ứng dụng cơ sở dữ liệu để quản lý dữ liệu tốt hơn

- Phát triển giao diện đồ hoạ trực quan để hỗ trợ thao tác thuận tiện, dễ dàng

# **TÀI LIỆU THAM KHẢO**

|  |  |
| --- | --- |
| [1] | S. Sanhi, Data Structures, Algorithms and Application in C++, University Press (India), 2009. |
| [2] | E. R. Eric Freeman, Head First Design Patterns, O'Reilly, 2014. |
| [3] | B. Eckel, Thinking in C++, Prentice Hall Inc., 2000. |
| [4] | C. Kormanyos, Real-Time C++ Efficient Object-Oriented and Template Microcontroller Programming, Springer, 2015. |
| [5] | "GP Coder," 30 8 2018. [Online]. Available: https://gpcoder.com/4164-gioi-thieu-design-patterns/. [Accessed 21 10 2022]. |