

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI TP. HỒ CHÍ MINH**

**VIỆN ĐÀO TẠO CHẤT LƯỢNG CAO**



**BÁO CÁO THUYẾT TRÌNH**

**DỰ ÁN QUẢN LÝ THƯ VIỆN**

**MÔN HỌC: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

**Giảng viên hướng dẫn: TS. Nguyễn Thị Khánh Tiên**

**THÀNH VIÊN NHÓM:**

**Nguyễn Đức Thiệp**

**Hồ Chí Nhân**

**Đỗ Kim Lâm**

**MSSV: 21H1120054**

**MSSV: 21H1120013**

**MSSV: 21H1120042**

***TP.Hồ Chí Minh, ngày 15 tháng 12 năm 2022***

**MỤC LỤC**

CHƯƠNG 1: TỔNG QUAN 1

1. Lý do chọn đề tài 1
2. Mục tiêu 1
3. Giới hạn và phạm vi 1
4. Phương pháp xây dựng 2

CHƯƠNG 2: LÝ THUYẾT CƠ SỞ 2

1. Tìm hiểu về C++ 2
2. Tìm hiểu về lập trình Hướng Đối Tượng 2

CHƯƠNG 3: THIẾT KẾ CHƯƠNG TRÌNH 3

1. Xác định các yêu cầu hệ thống 3
2. Yêu cầu nghiệp vụ 3
3. Yêu cầu phi chức năng 3
4. Diagram 4
5. Thực hiện 4
6. Xây dựng các lớp (class) 4
7. Xây dựng các hàm đọc/ghi file 15

3. Đăng nhập ADMIN và USER 17

4. Quyền hạn ADMIN 18

4.1 Quản lý đọc giả (READER) 18

4.2 Quản lý sách (BOOK) 25

* 1. Quản lý mượn – trả sách (BORROW-RETURN) 30

4.4 Các chức năng thống kê 35

5. Quyền hạn USER 36

5.1 Về đọc giả 36

5.2 Về sách 37

5.3 Về phiếu trả - mượn 37

5.4 Về thống kê 37

**CHƯƠNG 1: TỔNG QUAN**

1. **Lý do chọn đề tài**

Hiện nay, thư viện là một hệ thống khá phổ biến trong các trường học cũng như các địa điểm tra cứu thông tin. Thư viện là một kho sưu tập các nguồn thông tin, được chọn lựa bởi các chuyên gia và có thể được tiếp cận để tham khảo hay mượn. Cùng với sự phát triển của các thư viện là nhu cầu của độc giả tăng lên, và số lượng sách trong thư viện cũng tăng lên rất nhiều so với những hệ thống thư viện đơn giản và nhỏ lẻ trước đây.

Và, cũng một yêu cầu được đặt ra cùng với sự phát triển đó là làm thế nào để quản lý các thông tin trong thư viện một cách tốt nhất và có hiệu quả nhất. Vì vậy, nhóm chúng em xin giới thiệu 1 hệ thống quản lý thư viện, đáp ứng được 1 số chức năng cơ bản cần thiết của một hệ thống quản lý thư viện là quản lý tài liệu, độc giả, quản lý cập nhật và quản lý quá trình mượn và trả sách trong thư viện. Hệ thống mà nhóm chúng em đang giới thiệu tập trung chủ yếu vào quản lý dữ liệu (tài liệu, bạn đọc, tài khoản thư viện) và quản lý mượn, trả sách, …

1. **Mục tiêu**

Cung cấp một chương trình có thể quản lý thư viện, cụ thể là quản lý việc mượn và trả sách. Bên cạnh đó là quản lý sách, quản lý người đọc, quản lý thẻ/tài khoản thư viện, ...

Nắm vững được kiến thức cơ bản về lập trình hướng đối tượng nói riêng cũng như lập trình C++ nói chung.

Truy xuất dữ liệu có sẵn từ các file để giảm thiểu tối đa thời gian nhập, xuất danh sách. Lưu lại các dữ liệu đã bị thay đổi để hỗ trợ cho lần sau.

1. **Giới hạn và phạm vi**

Xây dựng chương trình thư viện với các chức năng nhập, xuất, lưu trữ  
thông tin người đọc, sách, phiếu trả-mượn,….

Phạm vi sử dụng: sử dụng trực tiếp trên máy tính thông qua trình biên dịch.

Đối tượng sử dụng: người quản trị viên (ADMIN) và người dùng (USER).

1. **Phương pháp xây dựng**

Để thực hiện, xây dựng chương trình này thì nhóm em đã sử dụng kiến thức được học từ các môn lập trình trước và chủ yếu là các kiến thức môn Lập trình hướng đối tượng. Đồng thời nhóm cũng tìm hiểu từ thực tế thư viện tại trường Đại học Giao thông vận tải TP. Hồ Chí Minh, các nguồn tài liệu trên Internet một số tài liệu về quản lý bằng ngôn ngữ C++.

**CHƯƠNG 2: LÝ THUYẾT CƠ SỞ**

* 1. **Tìm hiểu về C++**
* Ngôn ngữ **C++** được **Bjarne Stroustrup**phát triển từ ngôn ngữ C từ **cuối thập niên 1970**.
* **C++** là một phiên bản **mở rộng** của ngôn ngữ C, kết hợp tất cả các tính năng đã có của C.
* **C++** được coi như là ngôn ngữ **bậc trung** (**middle-level**), kết hợp các đặc điểm và tính năng của ngôn ngữ bậc cao và bậc thấp.
* **C++** có thể dùng để lập trình **nhúng**, lập trình **hệ thống**, hoặc những **ứng dụng**, **game**…
* **C++** là ngôn ngữ **"đa hướng"**. Nghĩa là nó hướng cấu trúc giống C và có thêm một tính năng cực kỳ quan trọng đó là tính năng **hướng đối tượng**. Các bạn sẽ được học phần hướng đối tượng của C++ trong serial **Lập trình hướng đối tượng C++**.
* **C++** là một trong những ngôn ngữ lập trình **phổ biết trên thế giới.**

1. **Tìm hiểu về lập trình Hướng Đối Tượng (Object-oriented programming, viết tắt: OOP)**

* Khái niệm: lập trình hướng đối tượng là một phương pháp lập trình dựa trên khái niệm về lớp và đối tượng. OOP tập trung vào các đối tượng thao tác hơn là logic để thao tác chúng.
* Đối tượng bao gồm 2 thành phần chính:
* Thuộc tính (Attribute): là những thông tin, đặc điểm của đối tượng.
* Phương thức (Method): là những hành vi mà đối tượng có thể thực hiện.
* Lớp là sự trừu tượng hóa của đối tượng. Những đối tượng có những đặc tính tương tự nhau sẽ được tập hợp thành một lớp.
* 4 Đặc tính cơ bản lập trình hương đối tượng:

### Tính đóng gói **(Encapsulation)**cho phép che giấu thông tin và những tính chất xử lý bên trong của đối tượng. Các đối tượng khác không thể tác động trực tiếp đến dữ liệu bên trong và làm thay đổi trạng thái của đối tượng mà bắt buộc phải thông qua các phương thức công khai do đối tượng đó cung cấp. Tính chất này giúp tăng tính bảo mật cho đối tượng và tránh tình trạng dữ liệu bị hư hỏng ngoài ý muốn.

### **Tính kế thừa (Inheritance)** là tính chất được sử dụng khá nhiều. Tính kế thừa cho phép xây dựng một lớp mới (lớp Con), kế thừa và tái sử dụng các thuộc tính, phương thức dựa trên lớp cũ (lớp Cha) đã có trước đó. Các lớp Con kế thừa toàn bộ thành phần của lớp Cha và không cần phải định nghĩa lại. Lớp Con có thể mở rộng các thành phần kế thừa hoặc bổ sung những thành phần mới.

### **Tính đa hình (Polymorphism)** cho phép các đối tượng khác nhau thực thi chức năng giống nhau theo những cách khác nhau.

### **Tính trừu tượng (Abstraction)** Tính trừu tượng giúp loại bỏ những thứ phức tạp, không cần thiết của đối tượng và chỉ tập trung vào những gì cốt lõi, quan trọng.

**CHƯƠNG 3: THIẾT KẾ CHƯƠNG TRÌNH**

* + 1. **Xác định các yêu cầu hệ thống**

1. **Yêu cầu nghiệp vụ**

Đối với ADMIN: là người có quyền cao nhất của hệ thống, ngoài việc được sử dụng các chức năng quản lý người đọc, quản lý sách,… còn có quyền tạo và cấp quyền cho USER.

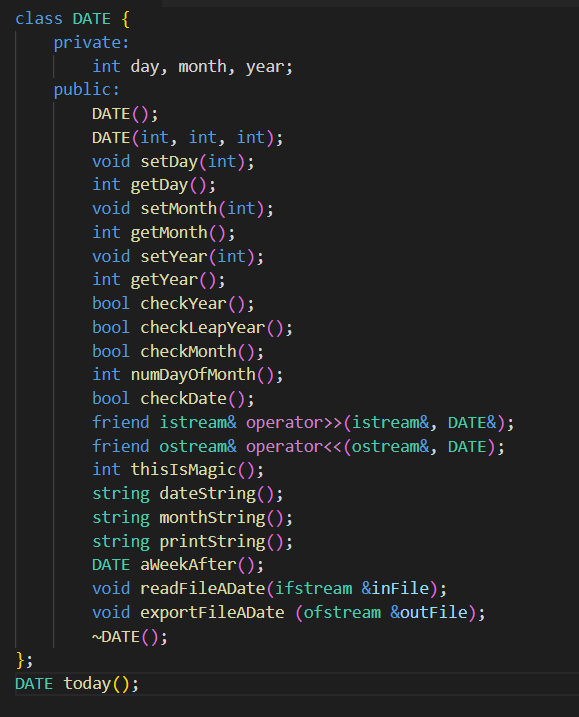
Đối với USER: là người có quyền tìm hiểu danh sách sách, tìm kiếm sách, quản lý account của mình,…

1. **Yêu cầu phi chức năng**

* Về giao diện:
* Giao diện bắt mắt, thân thiện người dùng.
* Menu tối giản, font chữ dễ nhìn.
* Về bảo mật:
* Bảo mật dữ liệu an toàn.
* Dữ liệu lưu trữ thời gian thực
* Về thương thích:
* Tương thích các trình biên dịch có setup mặc định.
  + 1. **Diagram**
    2. **Thực hiện**

1. **Xây dựng các lớp (class)**
   1. **. Lớp DATE**

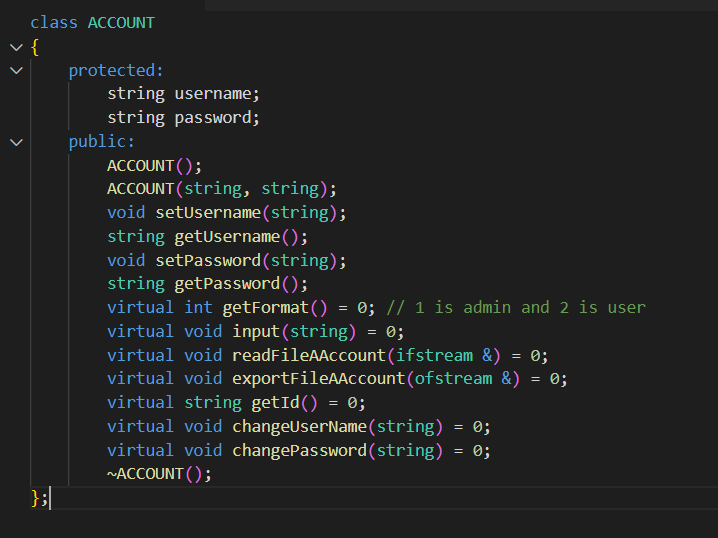
Được xây dựng nhằm thuận tiện dùng trong việc nhập, xuất dữ liệu kiểu có kiểu ngày/tháng/năm trong các lớp khác. Ngoài ra, lớp **DATE** còn bao gồm các phương thức: khởi tạo, get/set, kiểm tra năm nhuận, hàm **today()** (trả về ngày hiện tại), chuyển ngày tháng về dạng chuỗi (1st, 2nd, ...), **hàm thisIsMagic()** (với 1 ngày bất kỳ, hàm sẽ trả về một con số bất kỳ nhưng vẫn đảm bảo khoảng cách giữa 2 ngày), ...

****

**1.2. Lớp ACCOUNT (có 2 lớp con: ADMIN và USER)**

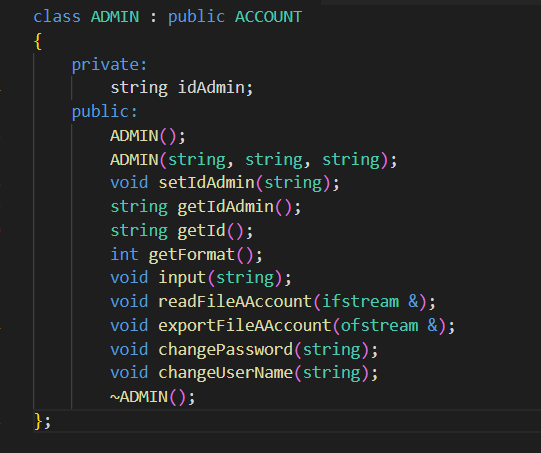
Dùng để xây dựng tài khoản cho các cá nhân muốn sử dụng các chức năng của chương trình (có 2 lớp con là **ADMIN** và **USER** (phân biệt bởi idAdmin và idUser) kế thừa trực tiếp các thuộc tính và phương thức của **ACCOUNT**.

Bao gồm 2 thuộc tính: tên đăng nhập (username) và mật khẩu (password) và các phương thức và phương thức thuần ảo: get/set, tạo tài khoản, thay đổi tên đăng nhập/mật khẩu.



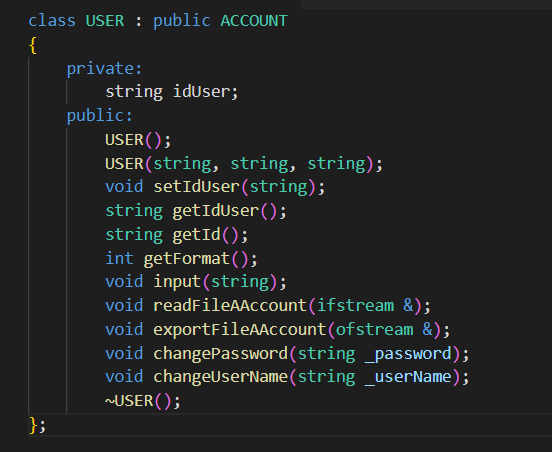
**1.2.1 Lớp ADMIN**

Kế thừa từ lớp ACCOUNT với thuộc tính phân biệt idAdmin cùng với các phương thức get/set, định nghĩa các phương thức thuần ảo ở ACCOUNT.



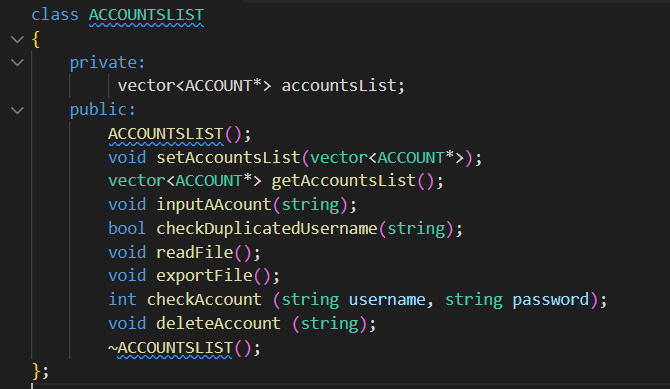
**1.2.1 Lớp USER**

Kế thừa từ lớp ACCOUNT với thuộc tính phân biệt idUser cùng với các phương thức get/set, định nghĩa các phương thức thuần ảo ở ACCOUNT.



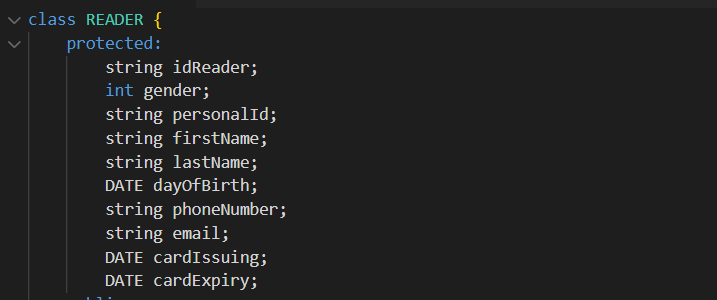
**1.3. Lớp ACCOUNTSLIST**

Bao gồm một danh sách các **ACCOUNT**, có tác dụng lưu trữ danh sách các tài khoản tiện cho việc lưu trữ và xử lý của hệ thống.



**1.4. Lớp READER (có 2 lớp con: STUDENT và TEACHER)**

Bao gồm các thuộc tính:



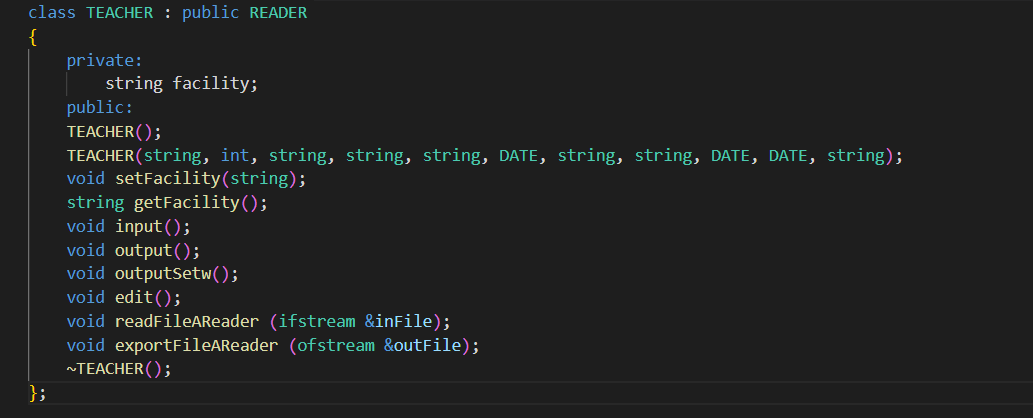
Cùng với các phương thức và phương thức thuần ảo:



Là lớp dùng để truy xuất thông tin của người đọc/mượn sách. Lớp này bao gồm 2 lớp con: lớp **STUDENT**(có thêm thuộc tính **lớp học**) và lớp **TEACHER**(có thêm thuộc tính **khoa**), kế thừa trực tiếp từ các thuộc tính và phương thức của **READER**.

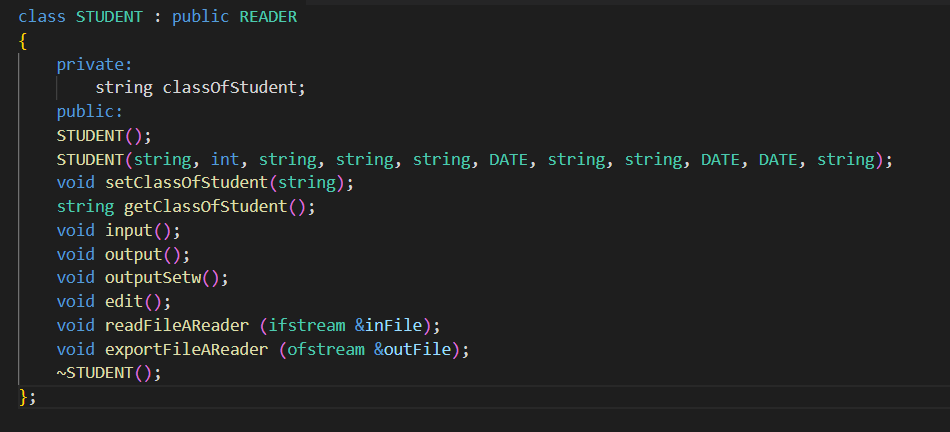
**1.4.1. Lớp TEACHER**

Kế thừa từ lớp READER với thuộc tính phân biệt facility(khoa) cùng với các phương thức get/set, định nghĩa các phương thức thuần ảo ở READER.

****

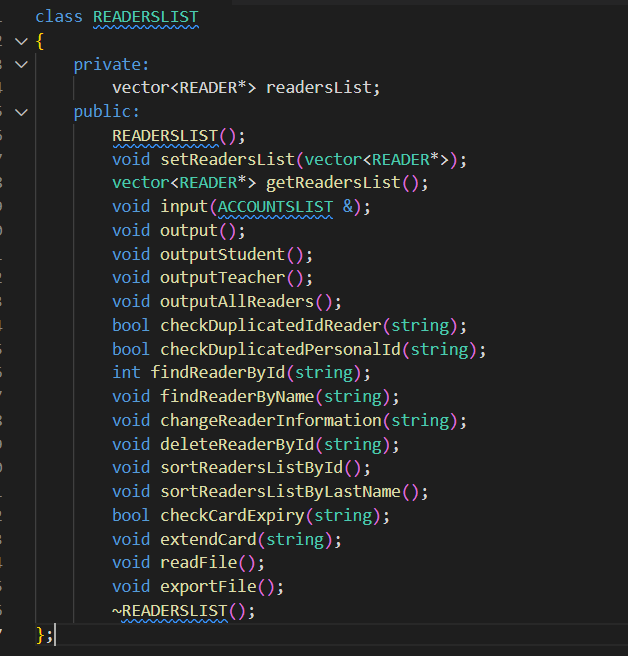
**1.4.2. Lớp STUDENT**

Kế thừa từ lớp READER với thuộc tính phân biệt classOfStudent(lớp) cùng với các phương thức get/set, định nghĩa các phương thức thuần ảo ở READER.



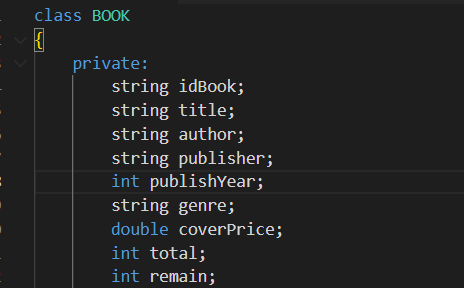
**1.5. Lớp READERSLIST**

Bao gồm một danh sách các **READER**, có tác dụng lưu trữ danh sách bạn đọc tiện cho việc truy xuất cũng như in danh sách độc giả.

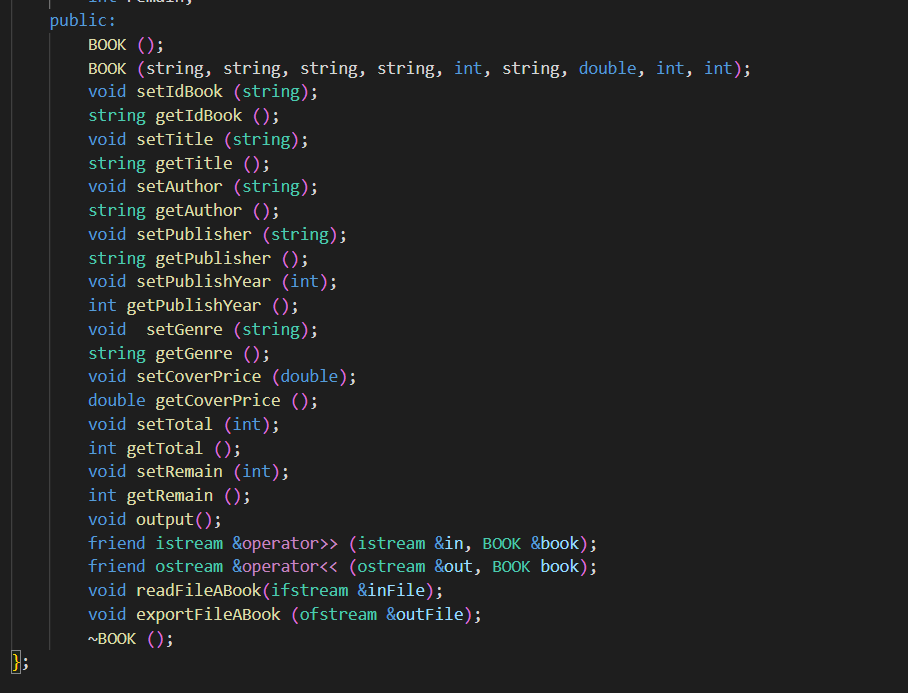
****

**1.6. Lớp BOOK**

Dùng để khai báo thông tin sách, gồm các thuộc tính:

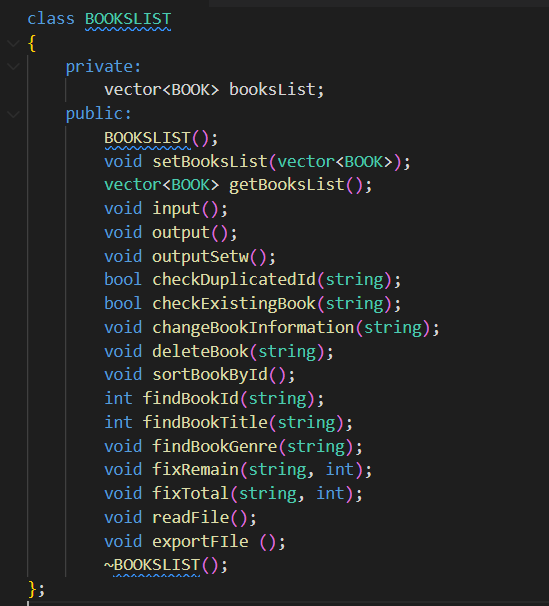


Lớp **BOOK** bao gồm các phương thức và các phương thức thuần ảo:



**1.7. Lớp BOOKSLIST**

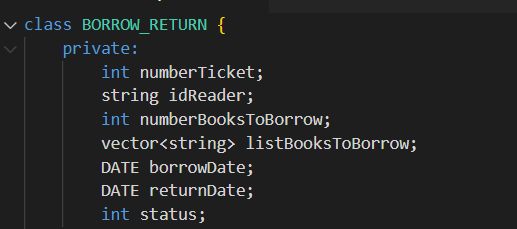
Bao gồm một danh sách các **BOOK**, có tác dụng lưu trữ danh sách các loại sách tiện cho việc lưu trữ của hệ thống và truy xuất của admin.



**1.8. Lớp BORROW\_RETURN**

Dùng để khai báo phiếu trả-mượn, lưu trữ danh sách tên sách mượn, thời gian mượn-trả.

Bao gồm các thuộc tính:

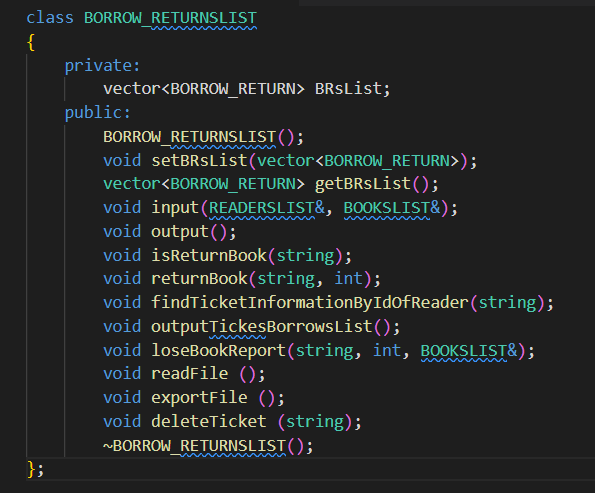


Các hàm cơ bản bao gồm:



**1.9. Lớp BORROW\_RETURNSLIST**

Bao gồm một danh sách các **BORROW\_RETURN**, có tác dụng lưu trữ danh sách các phiếu tiện cho việc mượn và trả sách của độc giả và truy xuất thông tin của admin.



1. **Xây dựng các hàm đọc/ghi file**

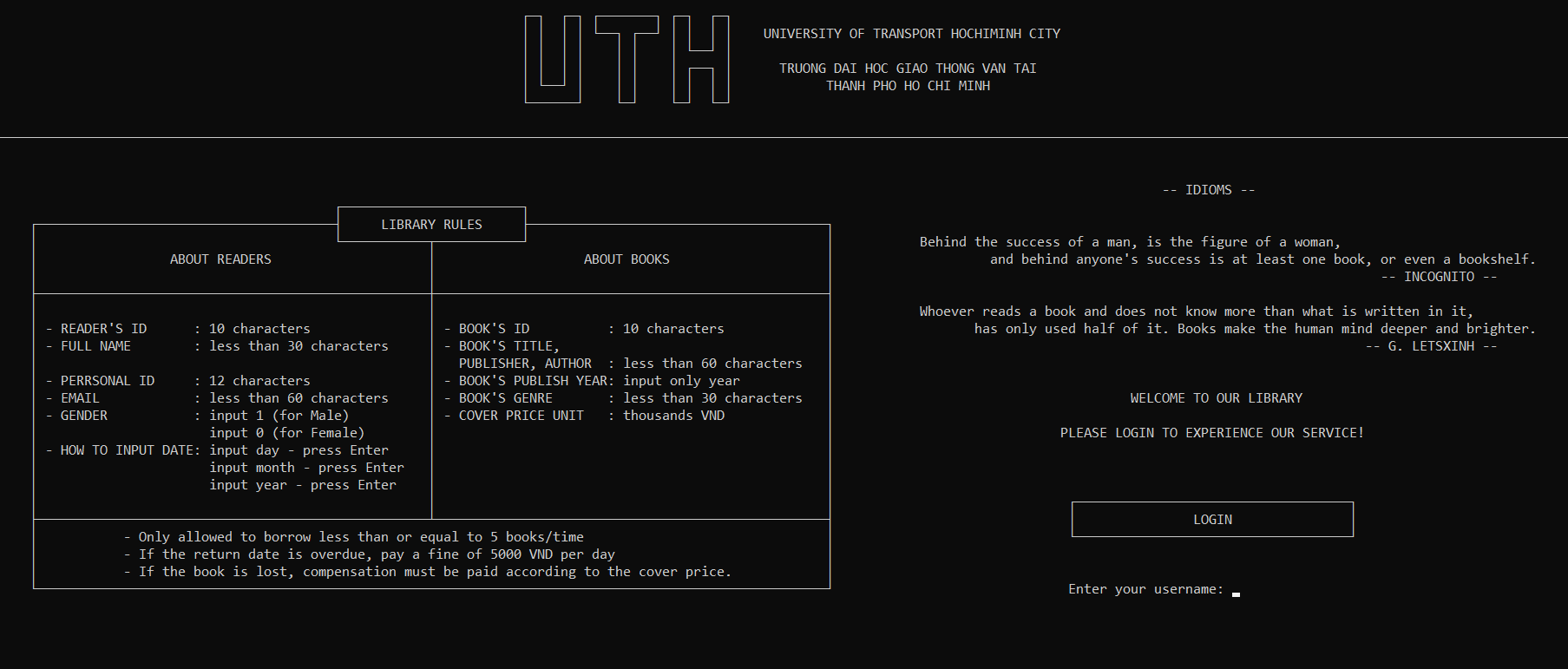
* Đọc file
* Dùng biến kiểu **ifstream** để tiến hành mở file và đọc file. Nội dung tương ứng từng cột sẽ được đọc, dấu “,” là dấu phân cách các cột. Nếu thông tin là DATE thì đọc lần lượt qua dấu phân cách “/” ngày, tháng, năm.
* Khi đọc tới cột cuối thì tiến hành đọc xuống dòng (\n) để qua đối tượng tiếp theo. Đọc và lưu từng đối tượng vào danh sách tương ứng. Lặp cho đến khi con trỏ file di chuyển tới cuối file.
* Sau khi làm xong thì đóng file.
* Ghi file
* Dùng biến kiểu **ofstream** để mở file .txt. Nếu không mở được thì báo lỗi. Ngược lại dùng vòng lặp để ghi file. Để ghi xuống dòng có thể dùng toán tử **<<** và kí tự **‘\n’**.
* Dùng vòng lặp để tiến hành lưu từng thông tin tương ứng với từng cột bằng toán tử **<<**. Và để con trỏ file qua cột khác thì dùng kí tự **‘,’**. Lặp đến khi danh sách sinh viên rỗng thì dừng.
* Sau khi làm xong thì đóng file.

VD: Đọc/Ghi file STUDENT



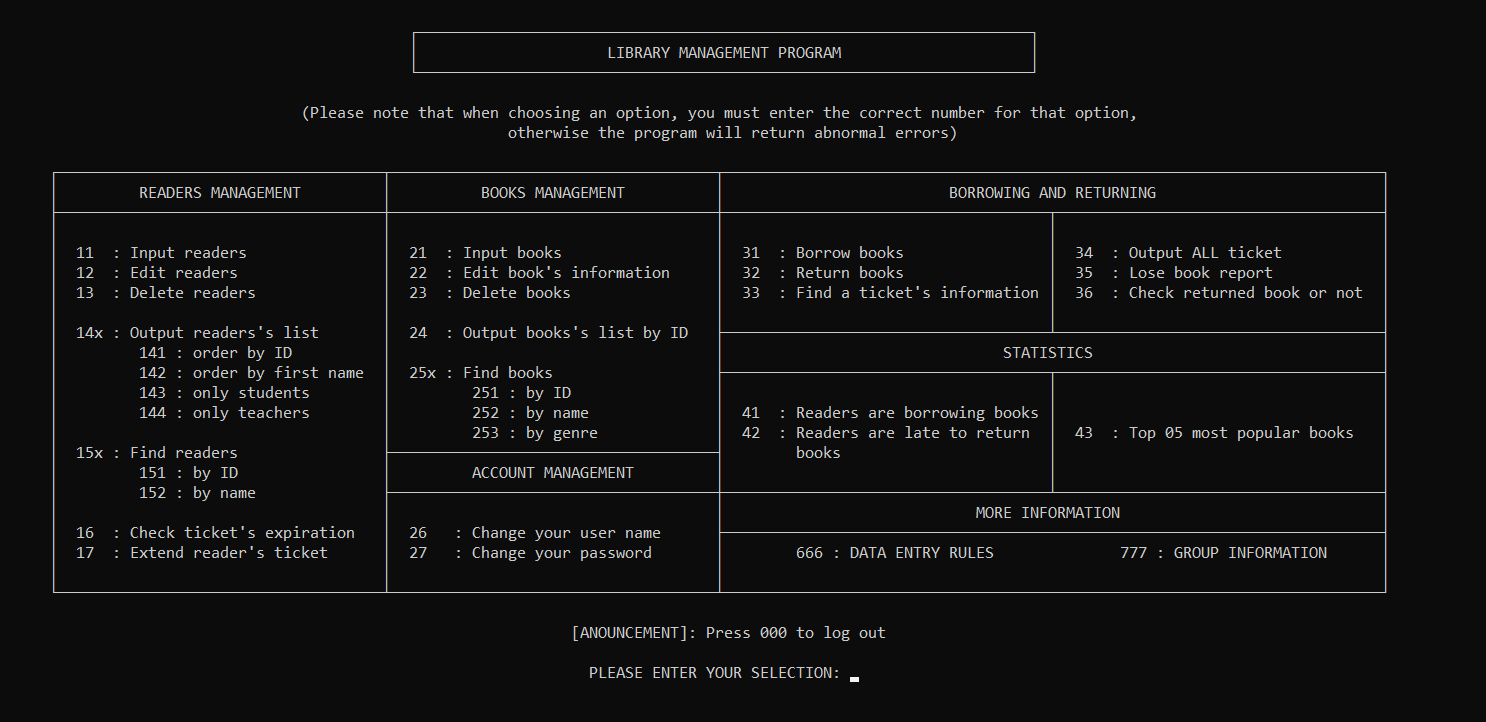
**3.** **Đăng nhập ADMIN và USER**

Khi khởi chạy chương trình, chương trình sẽ hiện ra khung đăng nhập yêu cầu người dùng nhập tên đăng nhập và mật khẩu của mình.

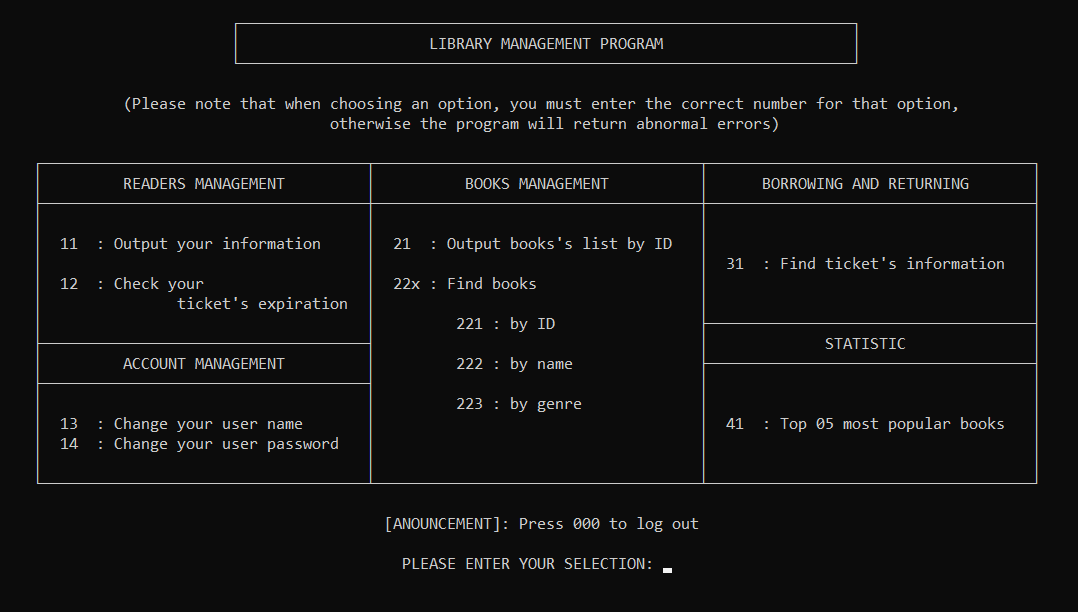


Sau khi đăng nhập thành công, chương trình sẽ đưa ra menu với các chức năng tương ứng với người truy cập (chức năng của admin sẽ khác với user).

Menu của admin:



Menu của user:



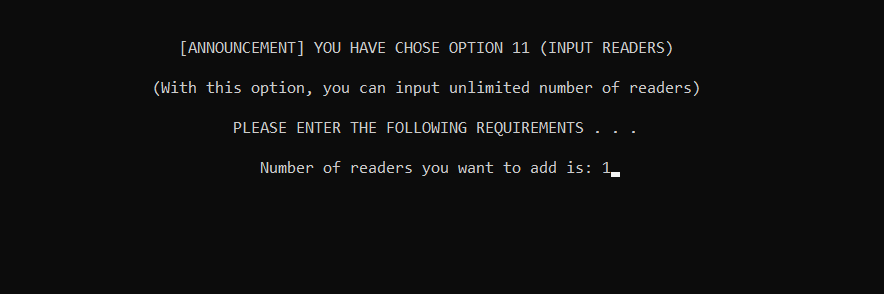
**4. Quyền hạn ADMIN**

**4.1 Quản lý đọc giả (READER)**

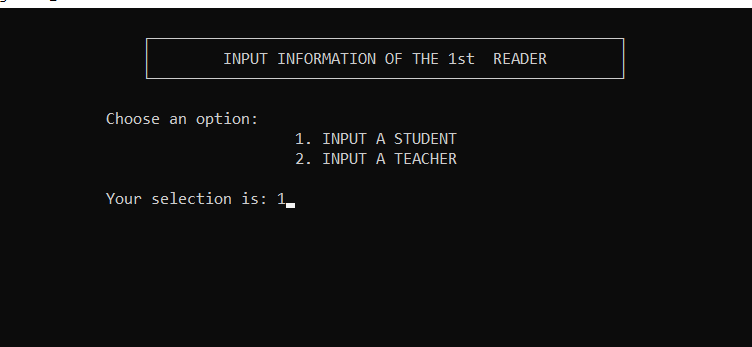
**11. Thêm đọc giả**

Với chức năng này, khi có 1 người đọc mới (không có trong danh sách) liên hệ để tạo tài khoản, họ sẽ cung cấp các thông tin cơ bản (với số lượng đã được nhập trước đó) và admin sẽ nhập các thông tin đó vào hệ thống.

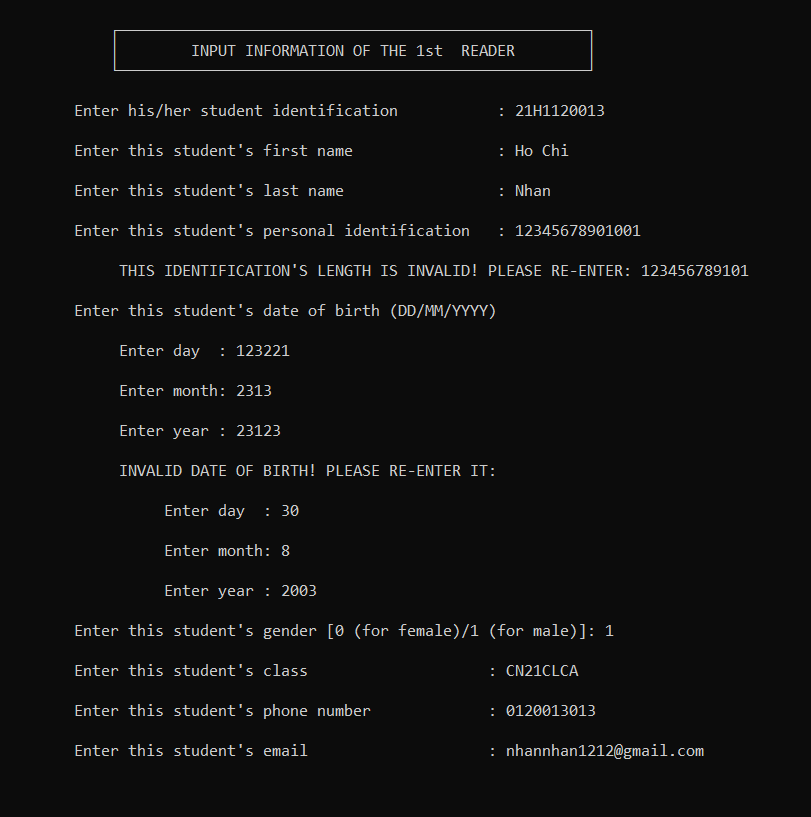
Nhóm có sử dụng các chức năng hỗ trợ như, kiểm tra mã người đọc, mã CCCD (có trùng hay không), kiểm tra mã sách, ... nhằm tăng tính hợp lệ cho danh sách người đọc.

Nhập số lượng đọc giả cần thêm:

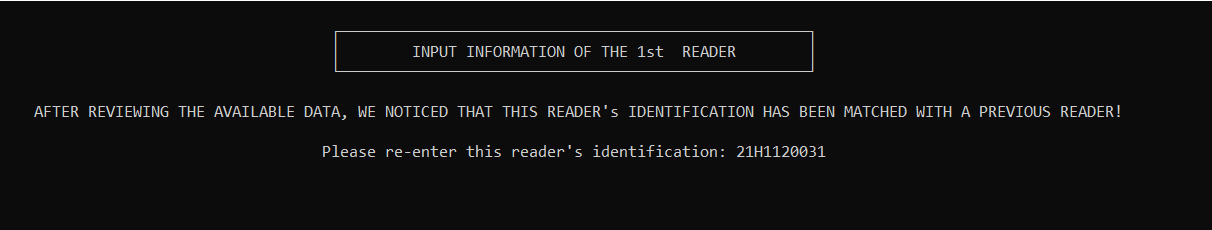
Chọn nhập STUDENT hay TEACHER:



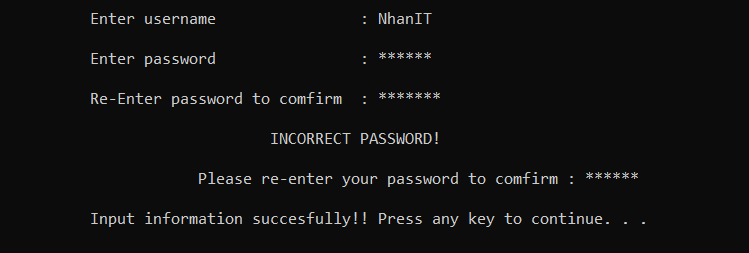
Nhập thông tin của đọc giả:



Nếu trùng ID chương trình sẽ bắt nhập lại:



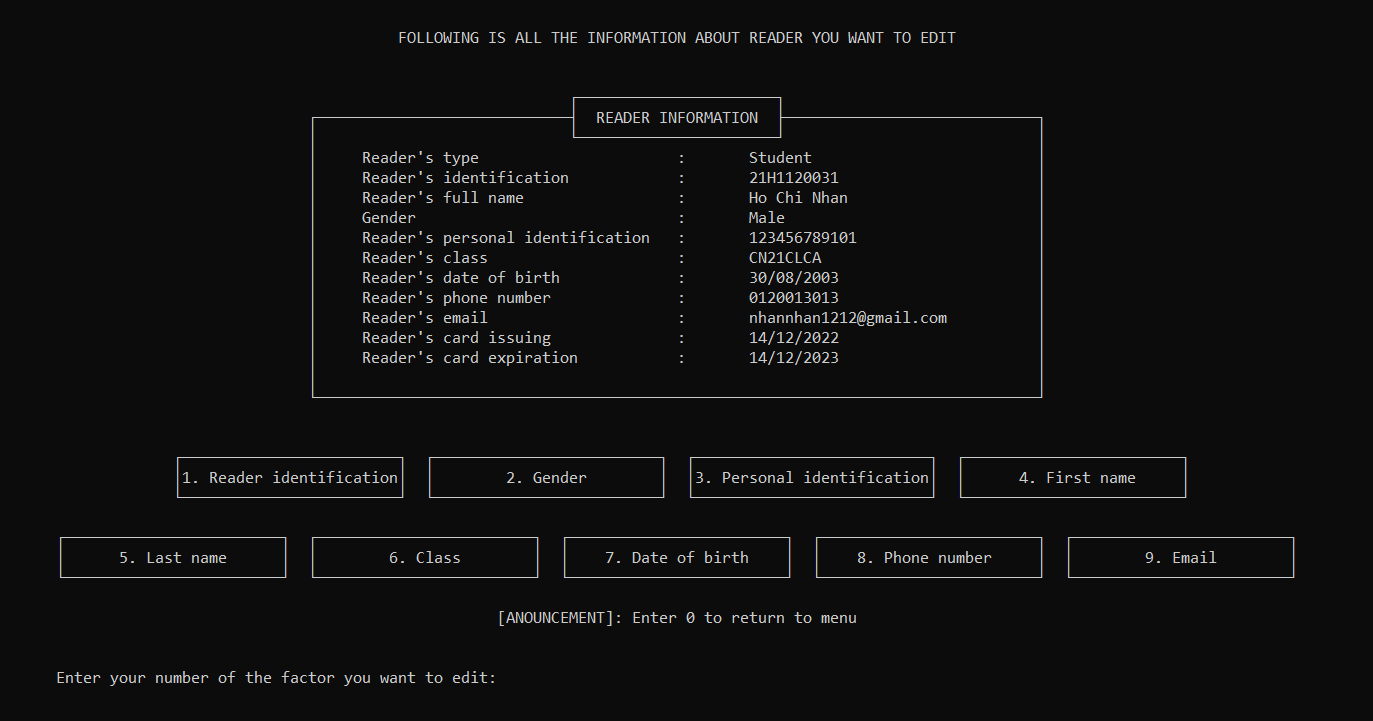
Nhập Username và password cho Account:



**12. Sửa thông tin độc giả**

Ý tưởng: admin nhập vào mã của độc giả, hệ thống sẽ truy xuất đến đối tượng độc giả này và đưa ra các tùy chọn để chỉnh sửa.

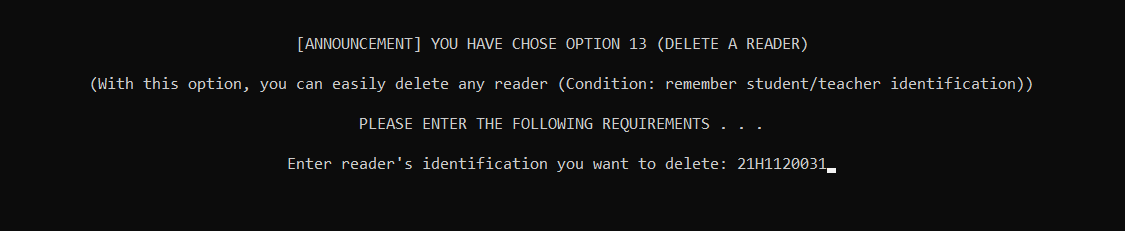
Giao diện chỉnh sửa thông tin:



**13. Xóa độc giả**

Ý tưởng: admin nhập vào mã của độc giả, hệ thống sẽ truy xuất đến đối tượng độc giả này và xóa tất cả các thông tin.

Nhập ID READER cần xoá:



Xoá thành công:

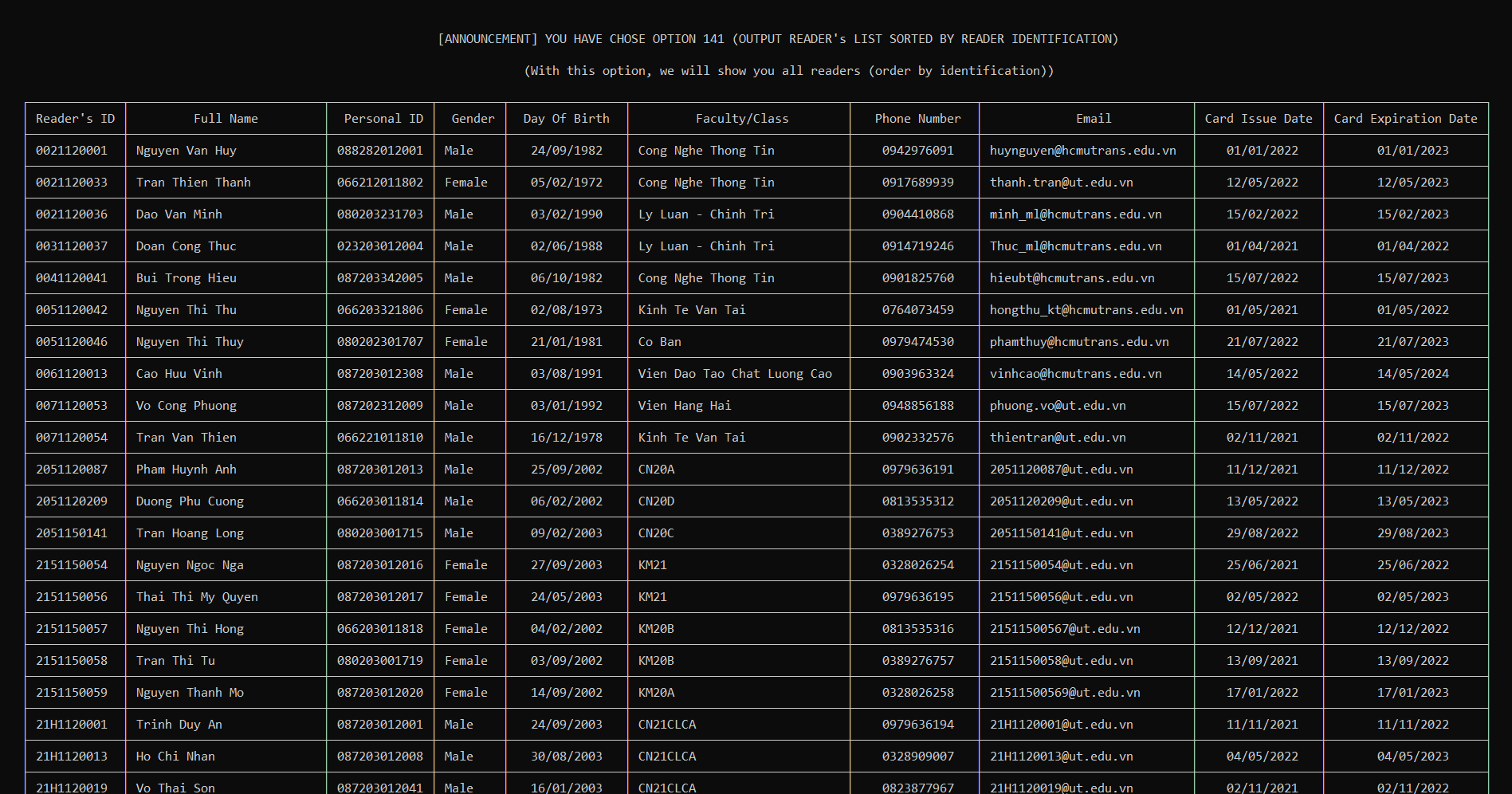


**14. Xuất danh sách đọc giả**

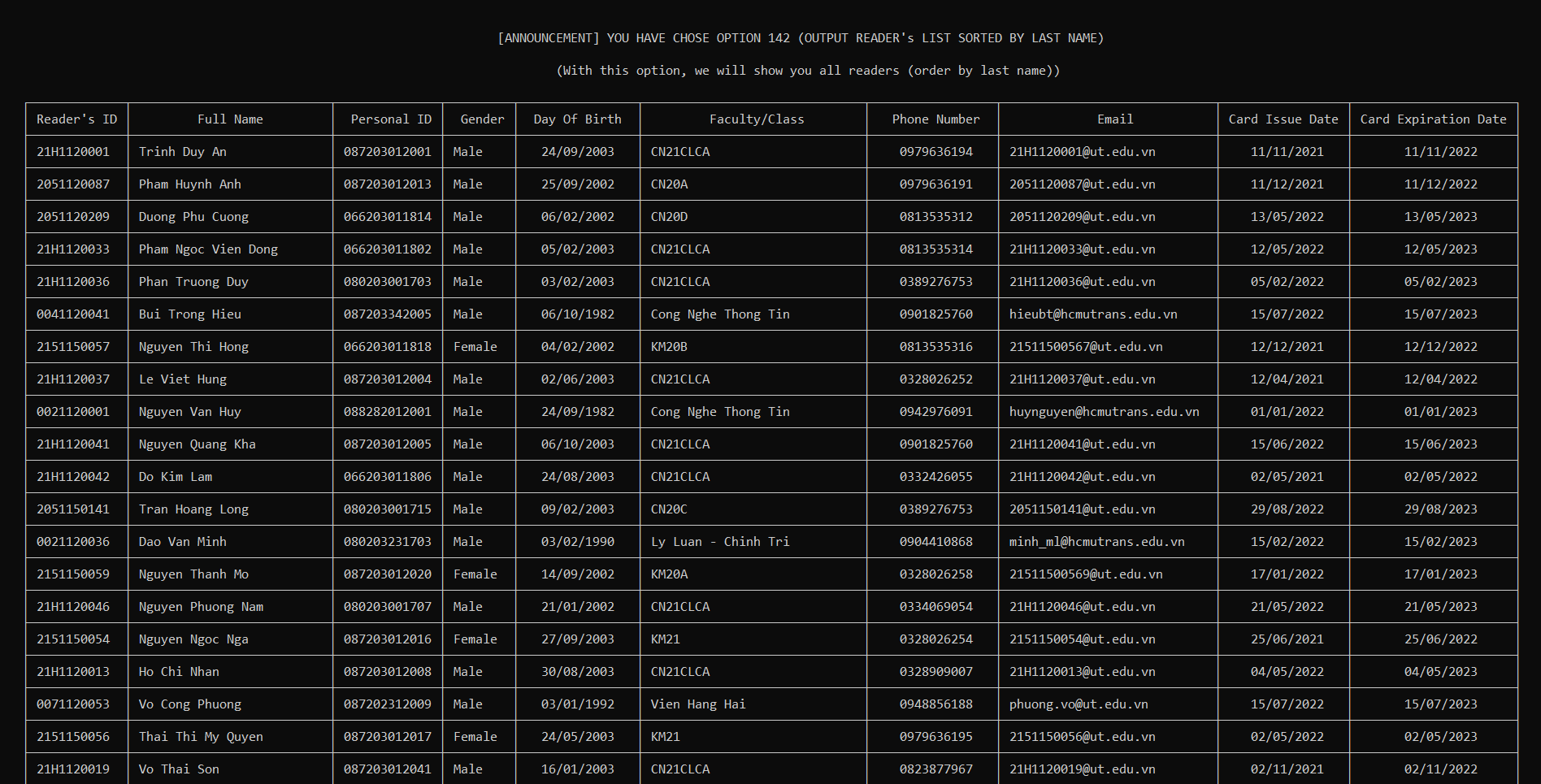
Ý tưởng: hệ thống sẽ truy xuất đến danh sách các đọc giả và sắp xếp danh sách (theo tên, theo mã) và xuất ra danh sách.

Chương trình còn hỗ trợ chức năng xuất thông tin chỉ các sinh viên hoặc giáo viên.

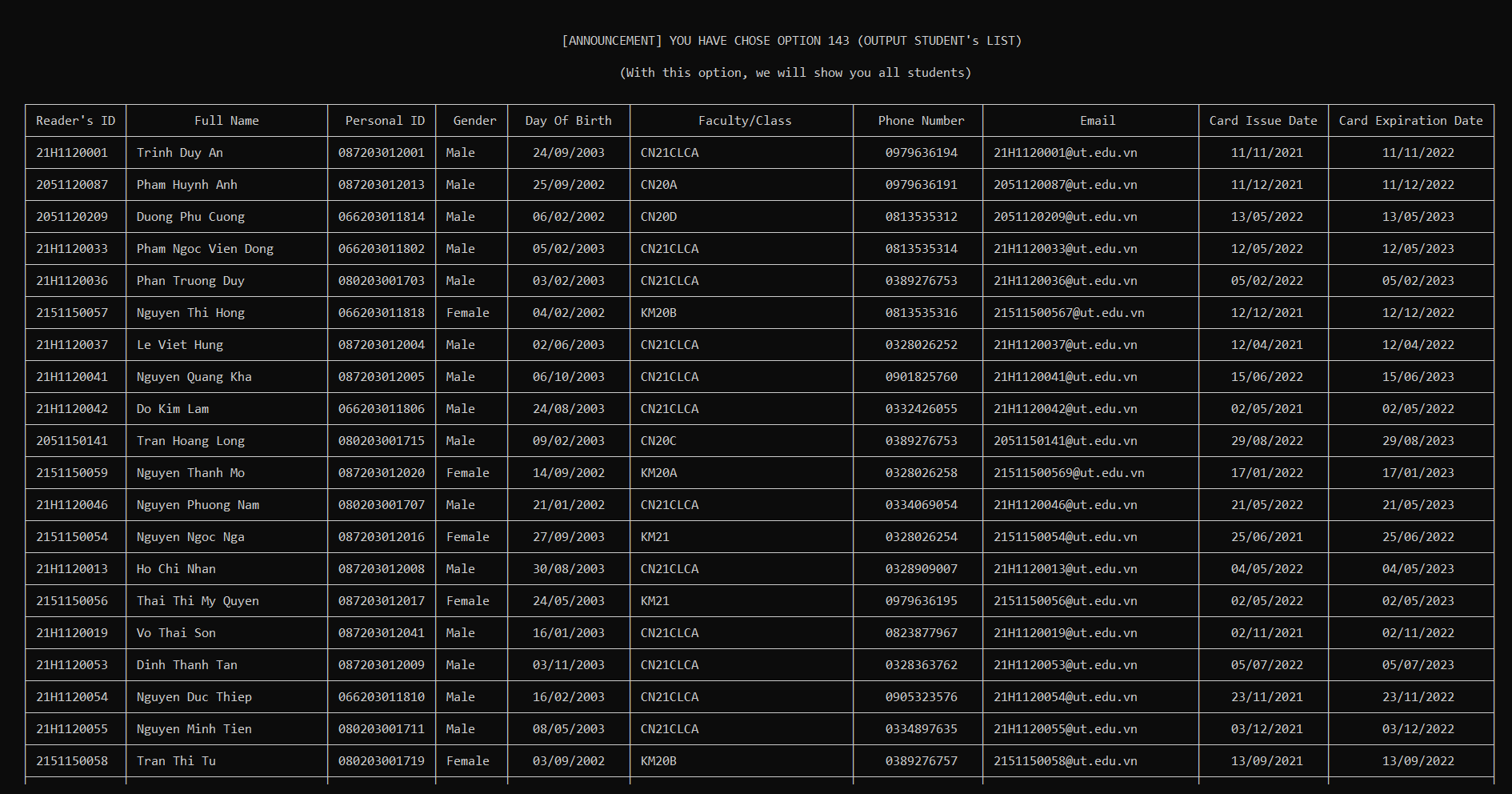
**141. Sắp xếp theo ID**

****

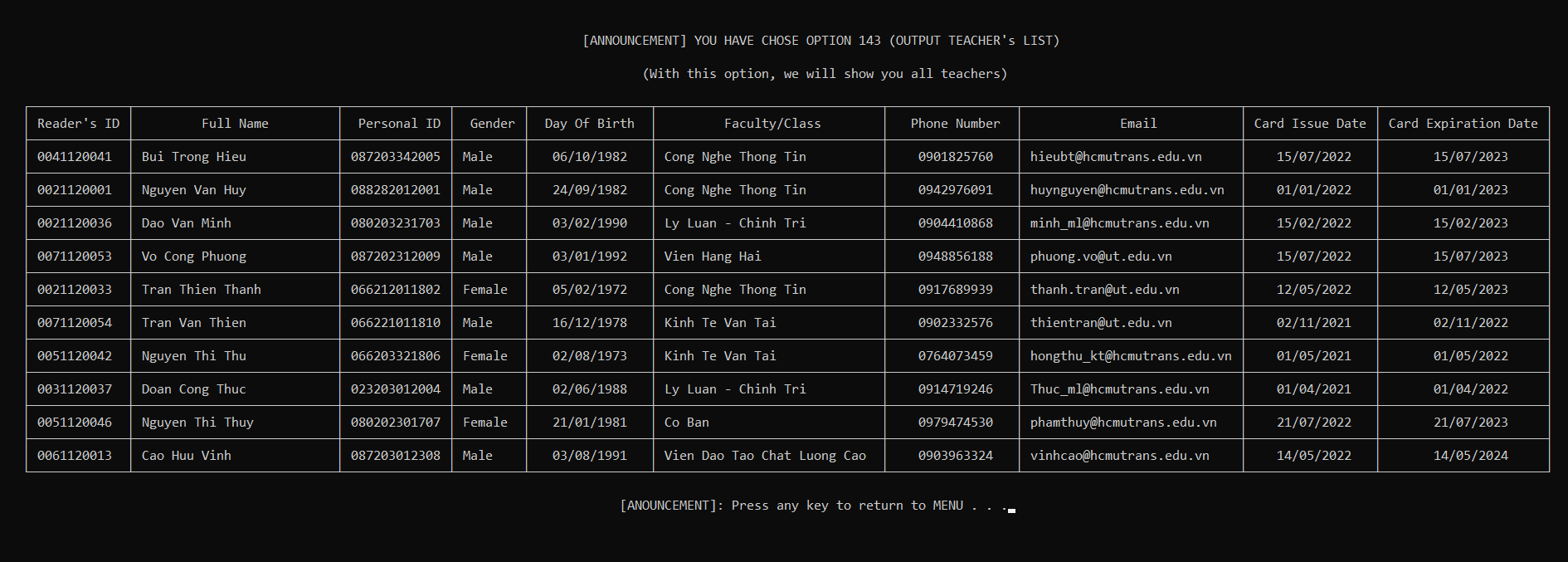
**142. Sắp xếp theo tên**

****

**143. Chỉ xuất Student**

****

**144. Chỉ xuất Teacher**

****

**15. Tìm kiếm thông tin đọc giả**

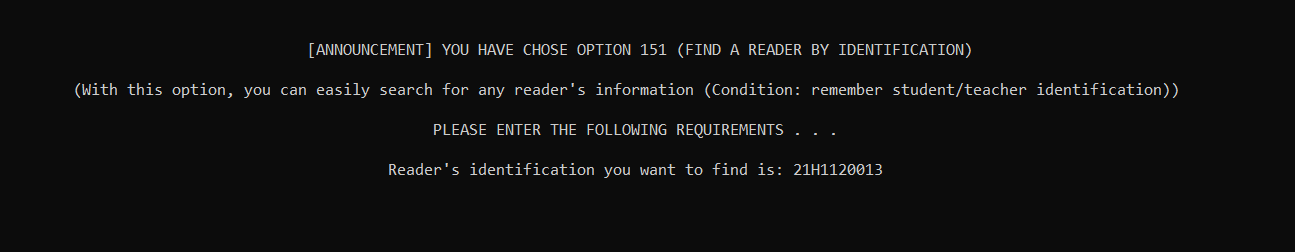
Ý tưởng: nhận vào mã (hoặc tên/tiêu đề) độc giả/sách, duyệt qua danh sách, độc giả/sách nào có thông tin giống với yêu cầu của admin thì sẽ được xuất thông tin.

Đầu vào: danh sách độc giả/sách của thư viện cùng với yêu cầu mà admin nhập vào (có thể là mã, tên độc giả, tiêu đề sách).

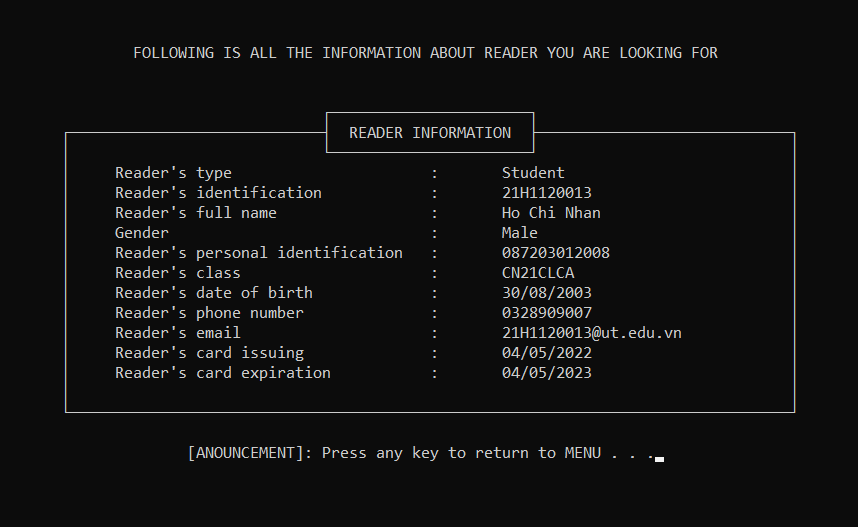
Đầu ra: thông tin độc giả/sách theo yêu cầu của admin.

**151. Theo ID**

Nhập ID READER cần tìm kiếm:

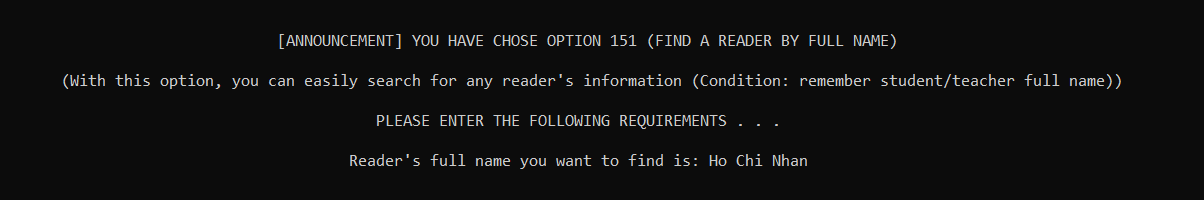


Kết quả:

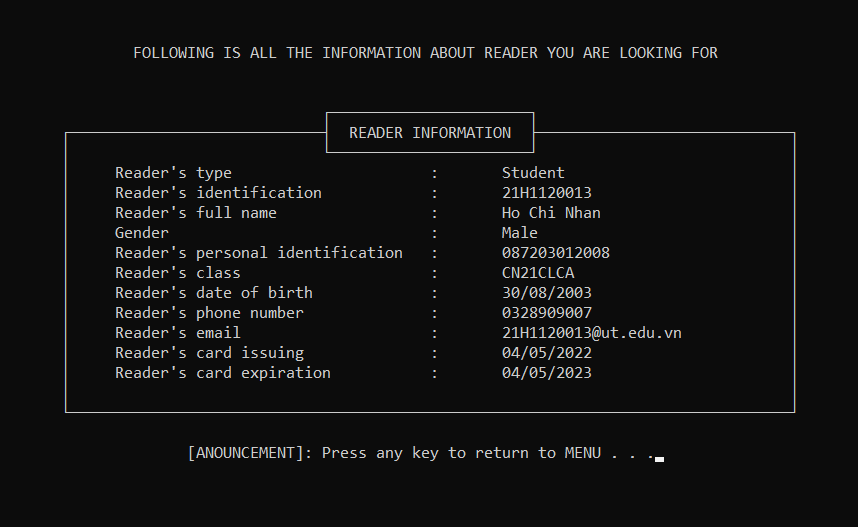


**152. Theo tên**

Nhập tên cần tìm kiếm:

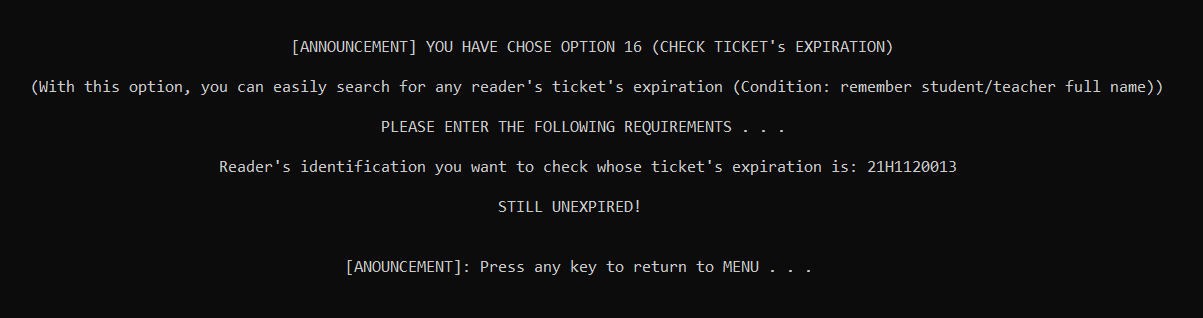


Kết quả:



**16. Kiểm tra hạn thẻ của đọc giả**

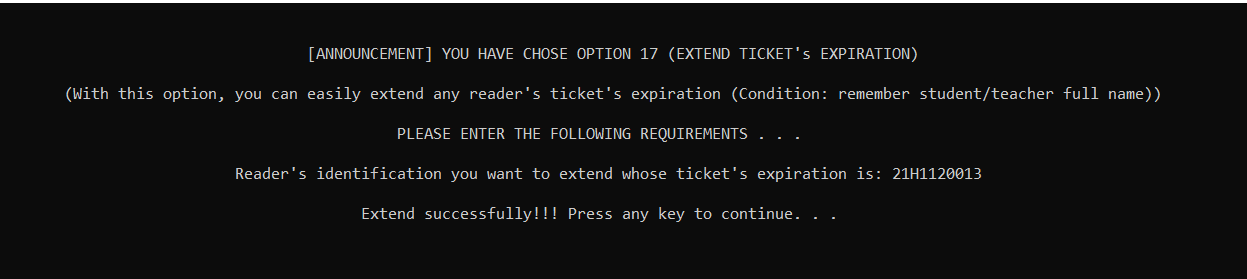
Nhập ID để kiểm tra xem thẻ còn hạn không:



**17. Gia hạn thời hạn thẻ cho đọc giả**

Với chức năng này, độc giả nào nếu muốn gia hạn thẻ thì có thể liên hệ với admin, nhiệm vụ của admin là nhập mã độc giả vào và hệ thống sẽ tự động gia hạn thời hạn thẻ của độc giả thêm 1 năm.

Nhập ID để gia hạn thẻ:



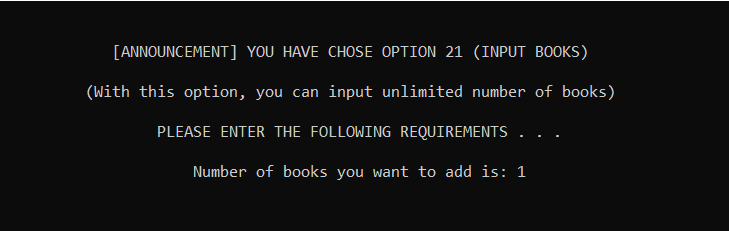
**4.2 Quản lý sách (BOOK)**

**21. Thêm sách**

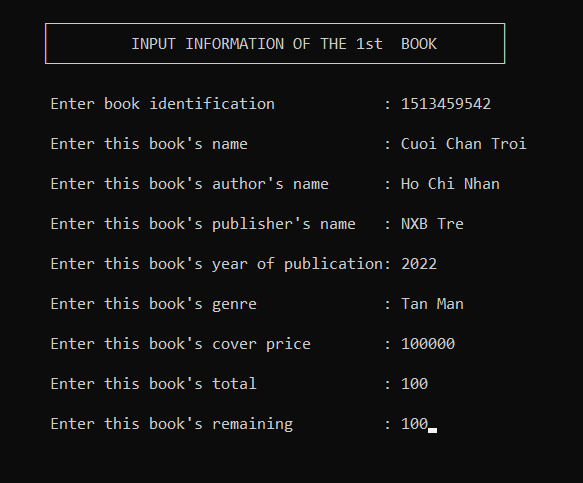
Với chức năng này, khi có 1 sách mới (không có trong danh sách) liên hệ Admin nhập các thông tin cơ bản (với số lượng đã được nhập trước đó).

Nhóm có sử dụng các chức năng hỗ trợ như kiểm tra mã sách (có trùng hay không), ... nhằm tăng tính hợp lệ cho danh sách sách.

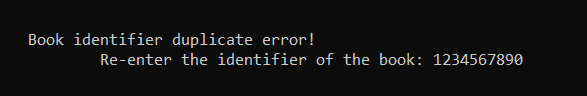
Nhập số lượng sách muốn thêm:



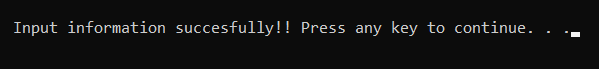
Nhập thông tin sách:



Nếu trùng ID sách:



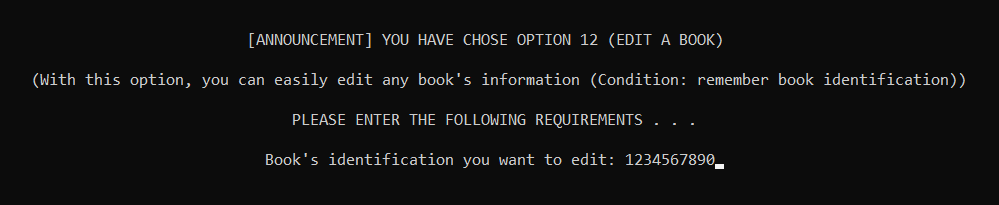
Kết quả:



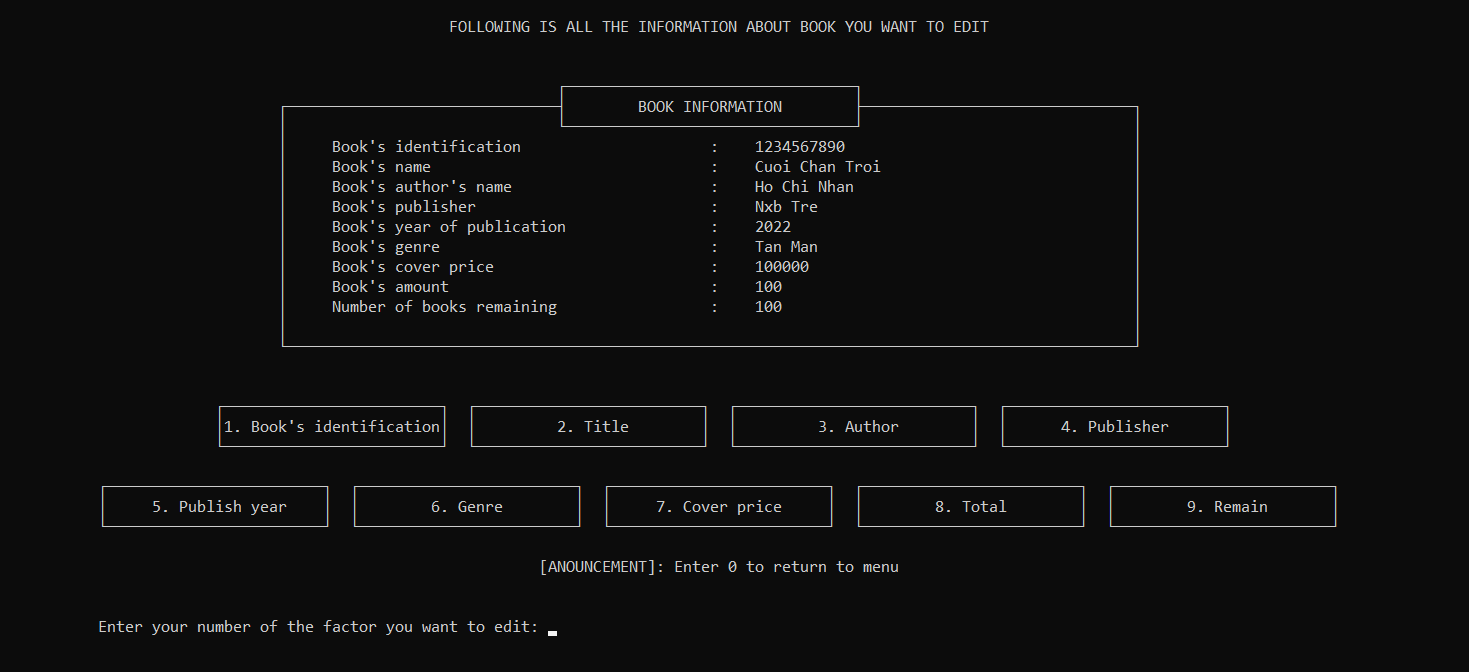
**22. Sửa thông tin sách**

Ý tưởng: admin nhập vào mã của sách, hệ thống sẽ truy xuất đến đối tượng sách này và đưa ra các tùy chọn để chỉnh sửa.

Nhập ID sách cần sửa:

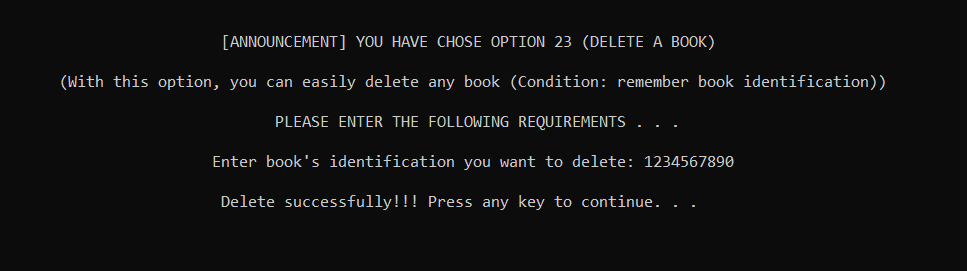


Giao diện chỉnh sửa sách:



**23. Xóa sách**

Ý tưởng: admin nhập vào mã của sách, hệ thống sẽ truy xuất đến đối tượng sách này và xóa tất cả các thông tin.

Nhập ID sách cần xoá:

**24. Xuất danh sách sách**

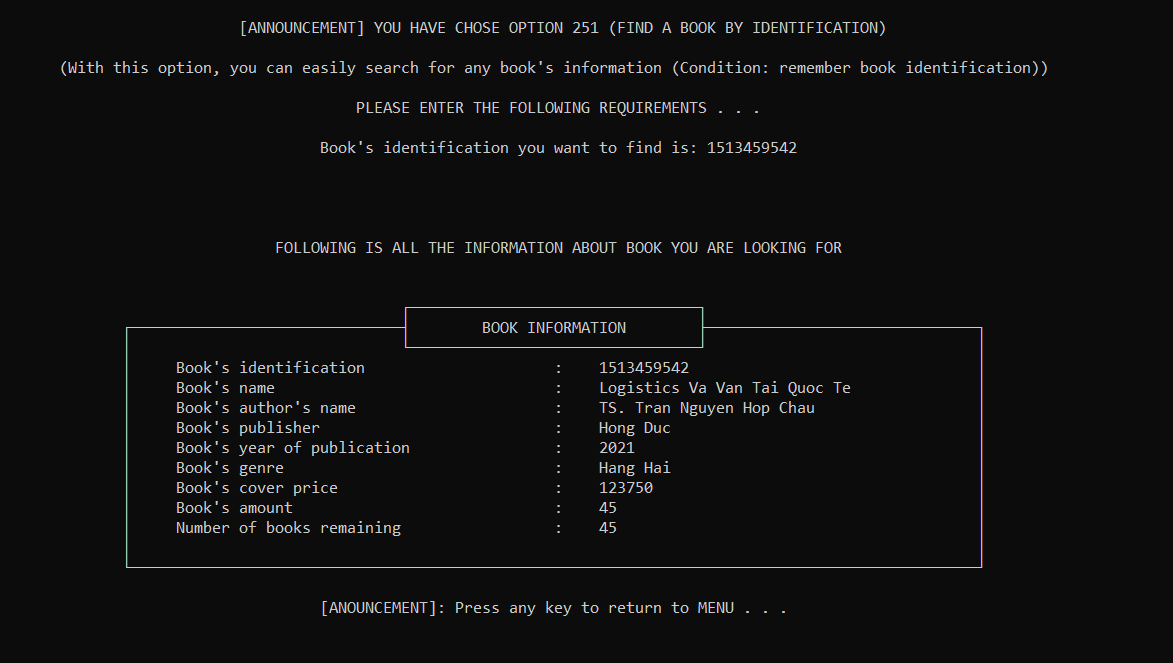
Ý tưởng: hệ thống sẽ truy xuất đến danh sách các sách và xuất ra danh sách.



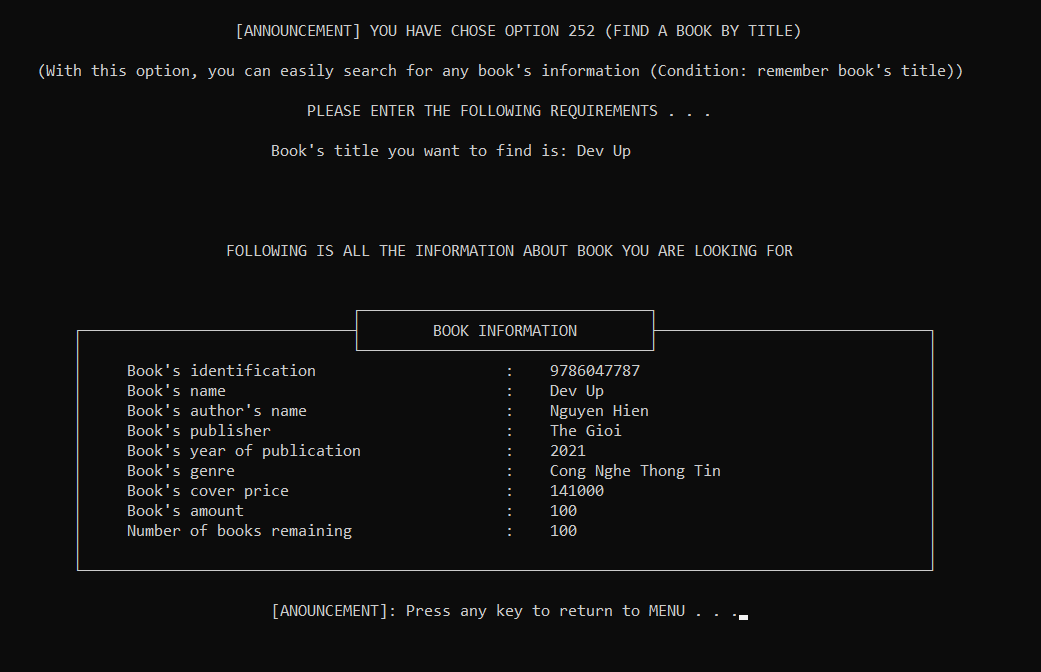
**25. Tìm kiếm thông tin sách**

Ý tưởng: nhận vào mã (hoặc tên/tiêu đề) sách, duyệt qua danh sách, độc sách nào có thông tin giống với yêu cầu của admin thì sẽ được xuất thông tin.

**251. Theo ID**

****

**252. Theo tên**

****

**253. Theo loại**

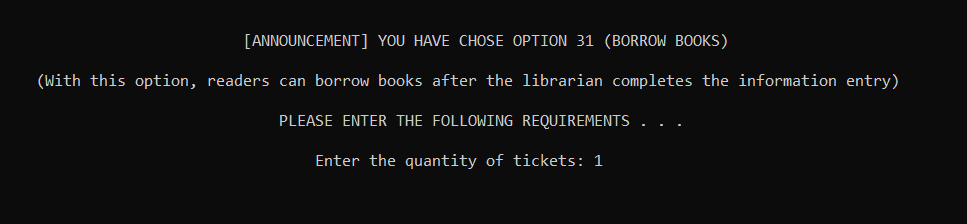
****

**4.3 Quản lý mượn – trả sách (BORROW-RETURN)**

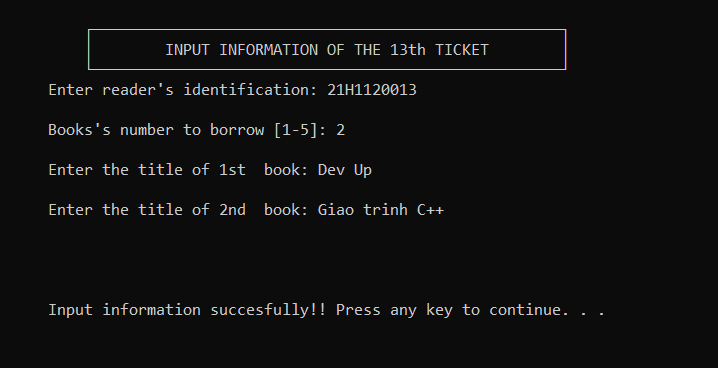
**31. Mượn sách**

Ý tưởng: Nhập số lượng sách mượn lưu vào phiếu trả mượn để quản lý

Nhập số lượng phiếu trả mượn:

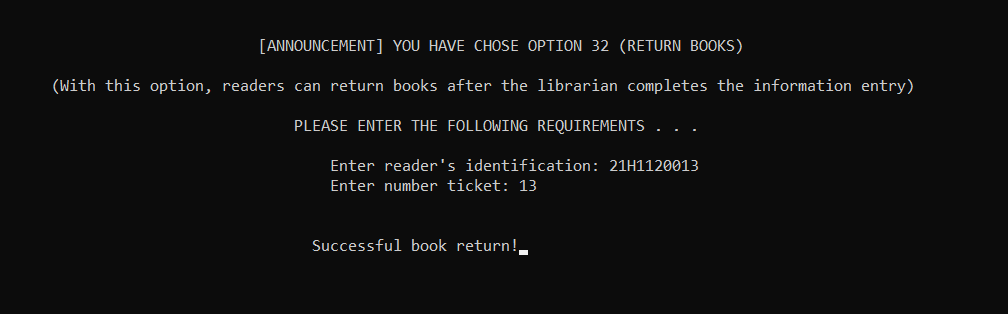


Nhập ID READER và danh sách sách mượn:



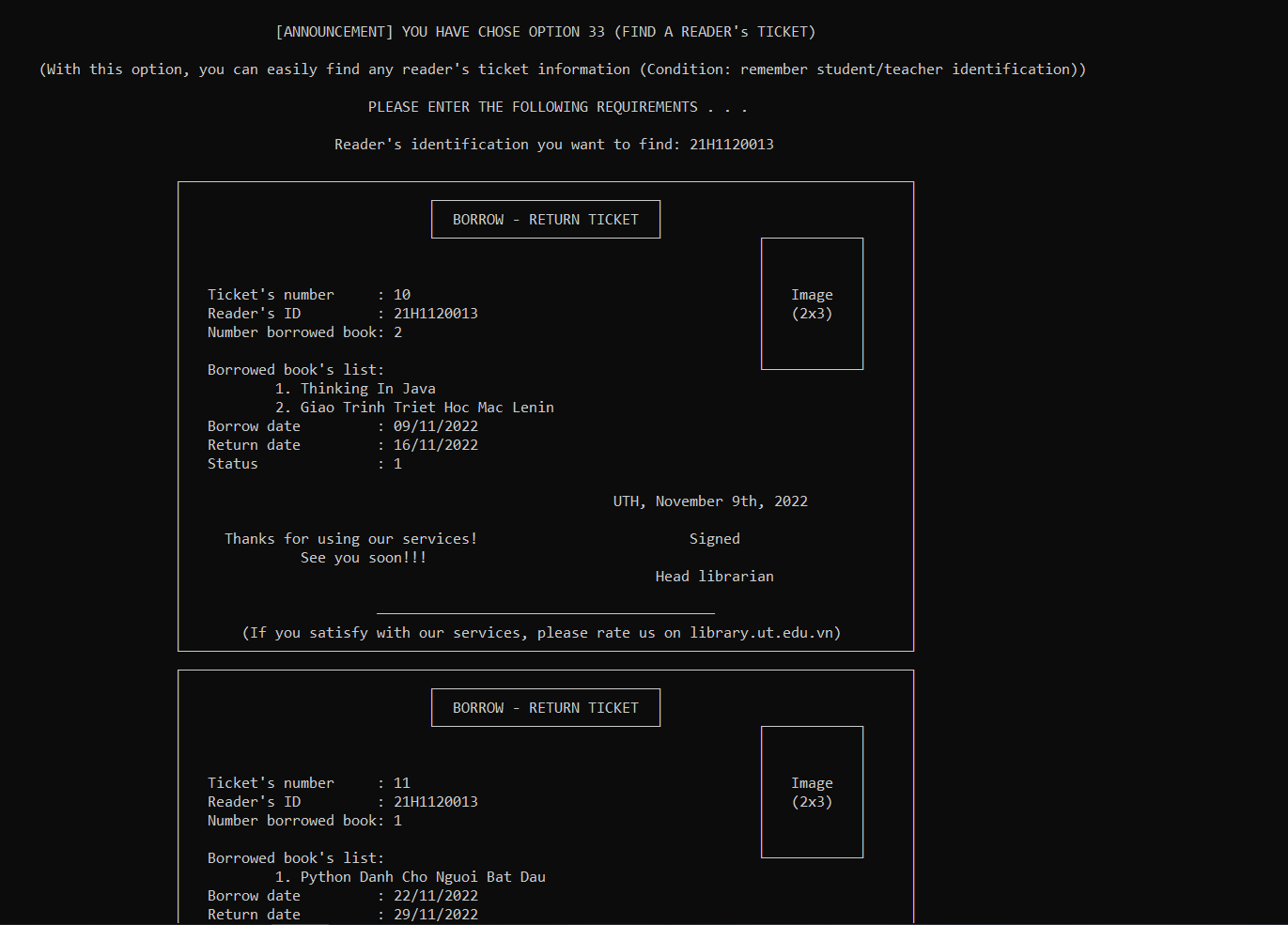
**32. Trả sách**

Ý tưởng: duyệt qua danh sách phiếu mượn, kiểm tra và xoá phiếu mượn.



**33. Tìm phiếu mượn**

Ý tưởng: duyệt danh sách phiếu trả mượn và in ra thông tin phiếu cần tìm



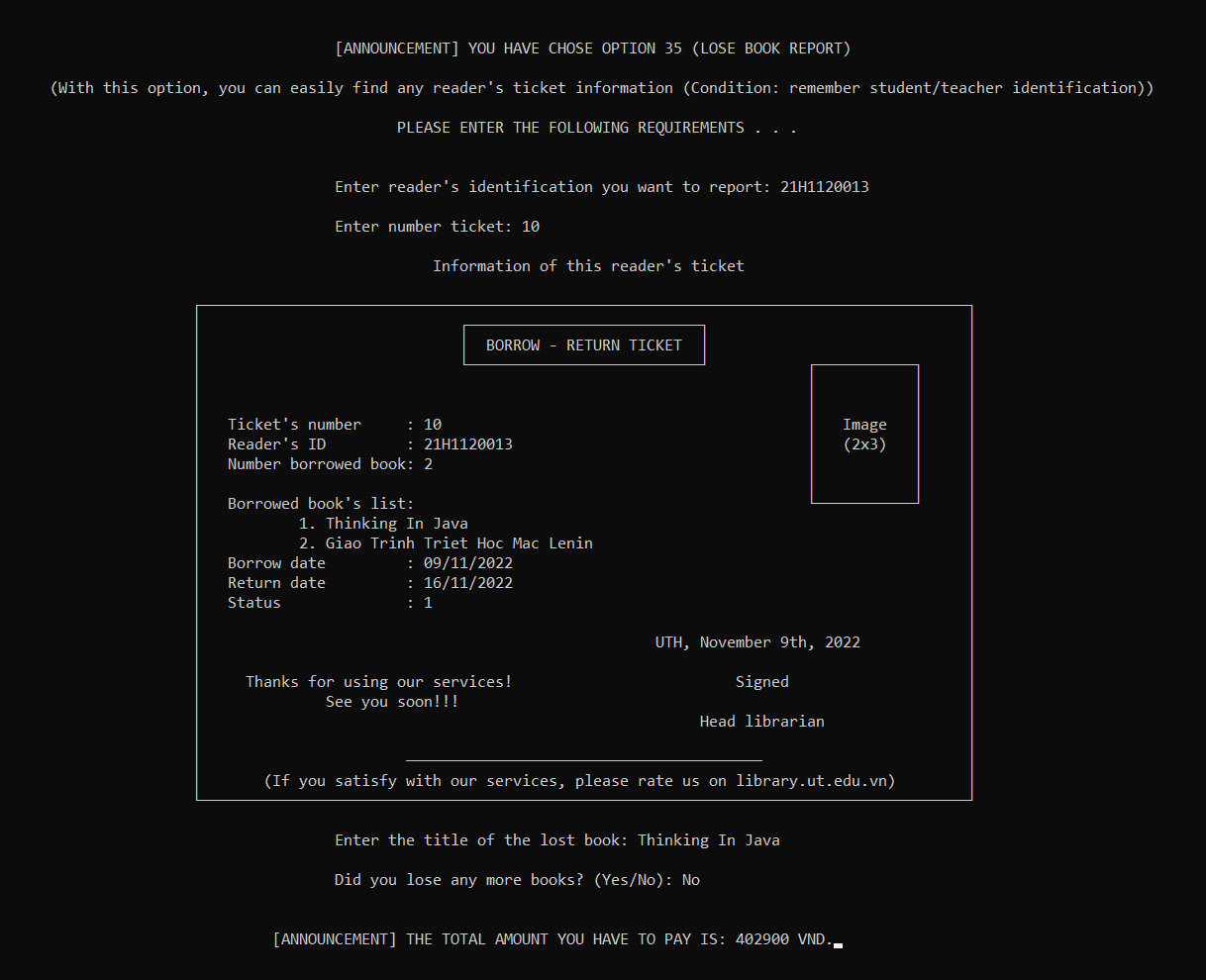
**34. Xuất tất cả phiếu mượn**

Ý tưởng:Duyệt qua danh sách phiếu mượn và in thông tin tất cả các phiếu mượn.



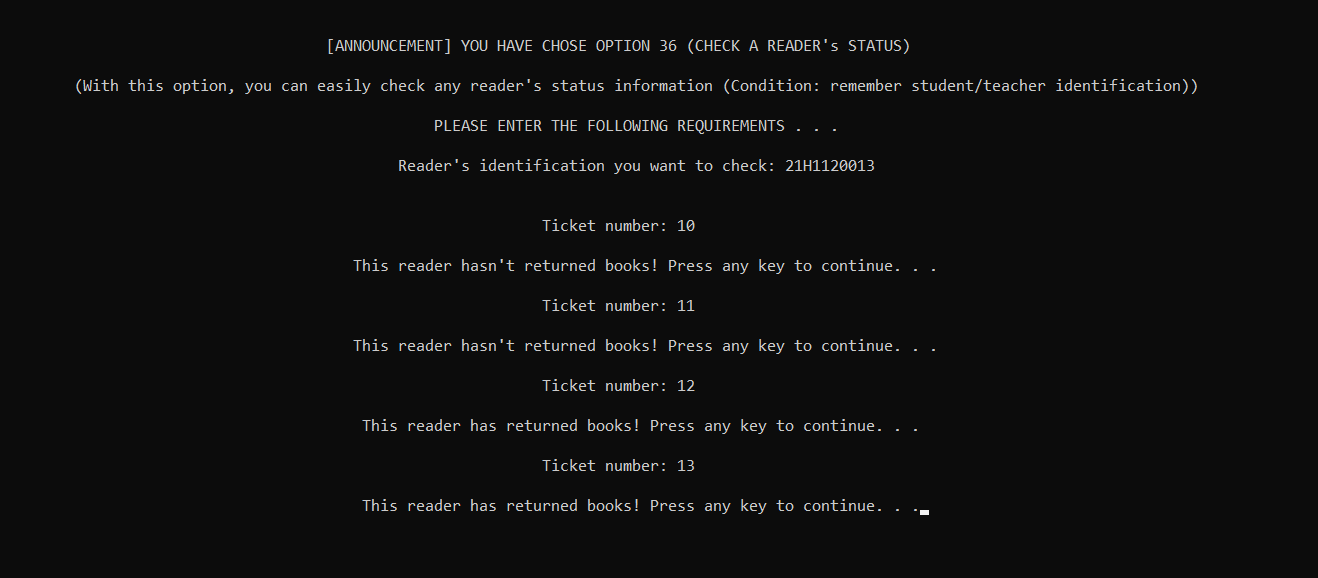
**35. Báo cáo mất sách**

Ý tưởng: duyệt danh sách phiếu mượn, nhập tên sách mất và duyệt qua danh sách sách mượn để kiểm tra, xuất tiền phải trả cho việc mất sách

****

**36. Kiểm tra độc giả có trả sách hay chưa**

Ý tưởng:Duyệt qua danh sách phiếu mượn và kiểm tra trạng thái phiếu mượn.

****

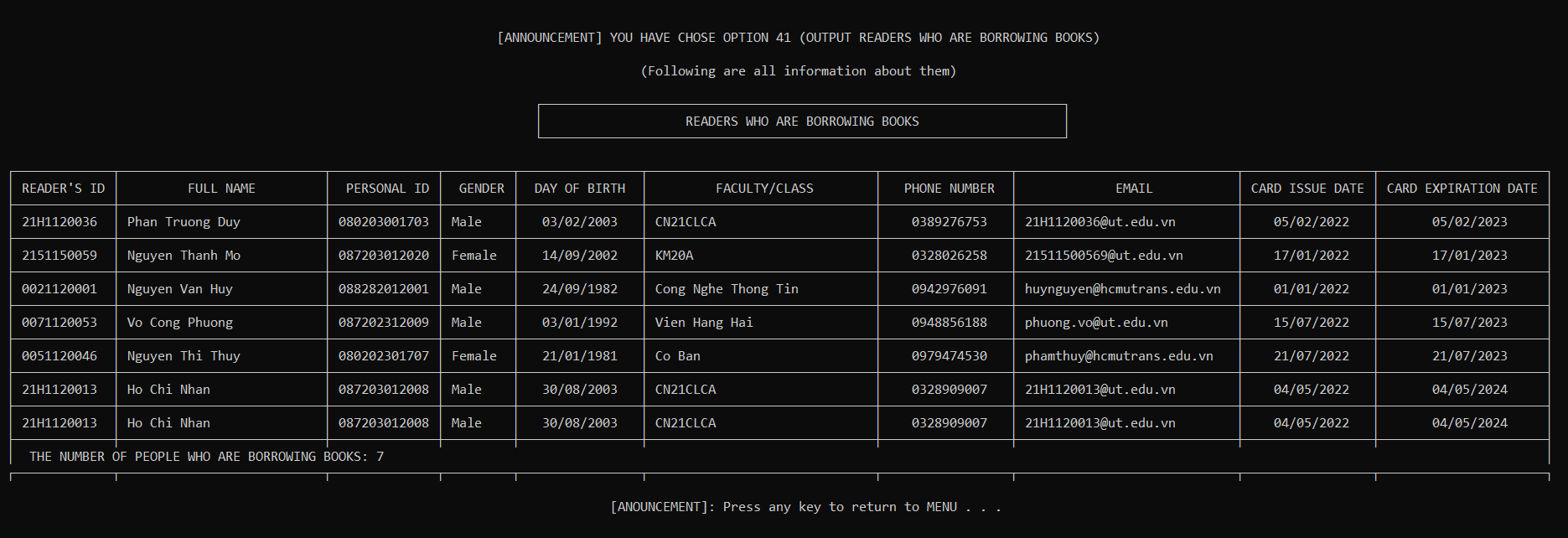
**4.4 Các chức năng thống kê**

**41. Danh sách đọc giả đang mượn sách**

Ý tưởng: duyệt qua danh sách, thẻ của độc giả nào có thuộc tính status = 1 (tức là chưa trả hết sách) thì sẽ được xuất thông tin.

Đầu vào: danh sách độc giả của thư viện.

Đầu ra: thông tin tất cả các độc giả đang mượn sách.

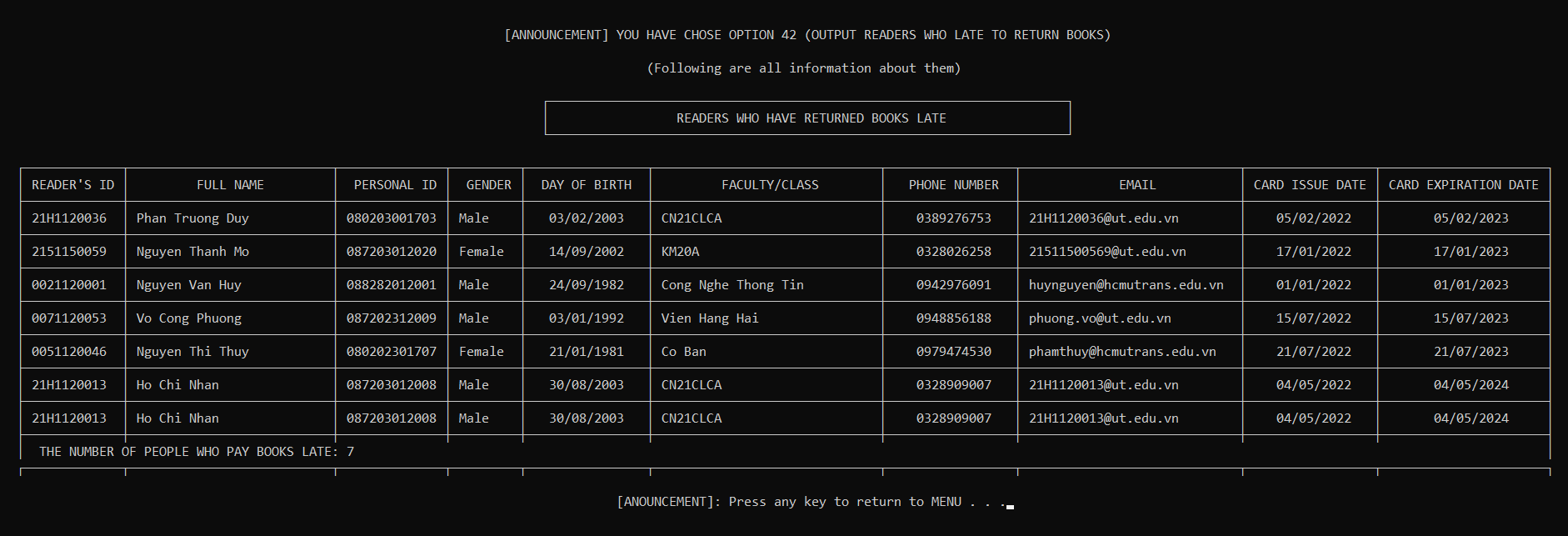


**42. Danh sách trễ hạn trả sách**

Ý tưởng: duyệt qua danh sách, thẻ của độc giả nào có thuộc tính status = 1 (tức là chưa trả hết sách) và borrowDate là ngày cách đây hơn 1 năm thì sẽ được xuất thông tin.

Đầu vào: danh sách độc giả của thư viện.

Đầu ra: thông tin tất cả các độc giả đã quá hạn trả sách.

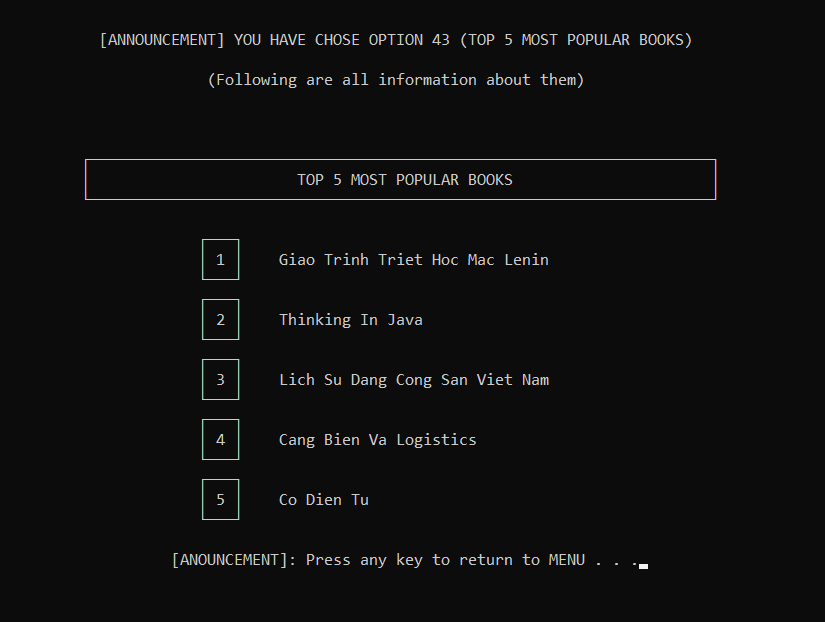


**43. Top 5 sách được mượn nhiều nhất**

Ý tưởng: duyệt qua danh sách sách, kiểm tra số lượng sách gốc và hiện tại để so sánh và xuất ra kết quả

Đầu vào: danh sách sách của thư viện.

Đầu ra: Top 5 sách được mượn nhiều nhất.



1. **Quyền hạn USER**

Các chức năng của User là các chức năng lấy từ Admin nhưng chỉ giới hạn ở việc đọc, không được quyền thêm và chỉnh sửa.

**5.1. Về đọc giả**

**11. Xuất thông tin của chính User**

Ý tưởng: nhận vào thông tin của user và xuất các thông tin theo một định dạng nhất định.

Đầu vào: dữ liệu của user đó, việc lập trình bằng phương pháp hướng đối tượng giúp đảm bảo dữ liệu của user không thể bị thay đổi trong hàm.

Đầu ra: Thông tin cá nhân của user.

**12. Kiểm tra thời hạn thẻ của User**

Ý tưởng: nhận vào thông tin của user, kiểm tra ngày hết hạn thẻ (so với hiện tại) và đưa ra kết luận.

Đầu vào: dữ liệu của user đó, việc lập trình bằng phương pháp hướng đối tượng giúp đảm bảo dữ liệu của user không thể bị thay đổi trong hàm.

Đầu ra: Nhận xét về tình trạng còn hay hết hạn của thẻ của user.

**5.2 Về sách**

**21. Xuất danh sách sách**

**22. Tìm sách**

**221. Theo ID**

**222. Theo tên**

**223. Theo thể loại**

**5.3 Về phiếu trả - mượn**

**31. Xuất thông tin phiếu trả mượn**

**5.4 Về thống kê**

**41. Top 5 sách được mượn nhiều nhất**

**CHƯƠNG 4: TỔNG KẾT**

1. Công cụ hỗ trợ

* + Hệ điều hành Windows 10/11
  + Công cụ lập trình: CodeBlock, DevC++, VS Code(C++ Extension).
  + Công cụ khác: Word,  Lucidchart, …

2. Ngôn ngữ viết: C++

### 3. Phân tích ưu/nhược điểm:

* **Ưu điểm:**
* Giao diện đẹp, bắt mắt, dễ dàng tiếp cận.
* Khả năng bảo mật cao.
* Quản lý các đối tượng một cách dễ dàng, nhanh chóng.
* Khả năng lưu trữ ngay lập tức, tránh tình trạng mất dữ liệu.
* Hỗ trợ nhiều chức năng, lựa chọn bổ sung.
* Các hàm hỗ trợ kiểm tra, nhập xuất tránh các lỗi về cú pháp, trùng lặp dữ liệu.
* **Nhược điểm:**
* Một số phần code chưa được tối ưu hoá.
* Chương trình chưa hỗ trợ hiển thị màu.
* Chương trình có thể chạy chậm ở 1 số thời điểm do lượng code lớn.
* Một số liên kết giữa các đối tượng chưa được hoàn chỉnh
* **Kinh nghiệm thu được:**
* Hiểu được làm việc một mình là không thể và làm việc nhóm quan trọng như thế nào. Cảm ơn các bạn trong nhóm đã hỗ trợ nhau hoàn thành dự án này.
* Có cơ hội vận dụng và trao dồi kĩ năng code của bản thân thông qua việc học hỏi và đúc kết từ các thành viên trong nhóm.
* Trải nghiệm tự mình xây dựng 1 dự án thực tế.

**4. Kết luận**

Nhóm đã tạo ra chương trình đáp ứng được yêu cầu đặt ra bằng những kiến thức đã học như: tính trừu tượng, tính đóng gói, tính kế thừa, tính đa hình, quá tải toán tử, đọc – ghi file, ... Thông qua bài tập này, nhóm đã có thể ứng dụng được kiến thức đã học để thực hiện hóa những mục tiêu, yêu cầu thực tế. Ngoài ra, mỗi thành viên luôn hỗ trợ lẫn nhau trong lúc làm, nó bù đắp những thiếu sót của nhau.

Thông qua bài tập, dù còn nhiều thiếu sót nhóm đã học hỏi và rút được nhiều kinh nghiệm trong việc học tập của mỗi bản thân cùng với đó là củng cố khả năng làm việc nhóm. Nhóm chúng em xin cảm ơn cô vì đã tạo điều kiện để chúng em có thể học hỏi cũng như trau dồi kinh nghiệm cho bản thân. Nhóm chúng em sẽ cố gắng phát triển chương trình trong tương lai.