

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN**



BÁO CÁO ĐỒ ÁN TIN HỌC

ĐỀ TÀI XÂY DỰNG CHƯƠNG TRÌNH QUẢN LÝ PHÒNG GAME

GVHD: HOÀNG KHUÊ

SVTH: TRẦN VĂN QUỐC THẮNG – DH52007101

VÕ XUÂN LAM – DH52006929

ĐẶNG XUÂN SANG – DH52006904

TRẦN NGUYỄN THANH SANG – DH52007102

ĐỖ TRẦN HẢI SƠN – DH52007024

Tp. Hồ Chí Minh, 12/2022

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN**



BÁO CÁO ĐỒ ÁN TIN HỌC

ĐỀ TÀI XÂY DỰNG CHƯƠNG TRÌNH QUẢN LÝ PHÒNG GAME

GVHD: HOÀNG KHUÊ

SVTH: TRẦN VĂN QUỐC THẮNG – DH52007101

VÕ XUÂN LAM – DH52006929

ĐẶNG XUÂN SANG – DH52006904

TRẦN NGUYỄN THANH SANG – DH52007102

ĐỖ TRẦN HẢI SƠN – DH52007024

Tp. Hồ Chí Minh, 12/2022

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Tp. Hồ Chí Minh, ngày ... tháng ... năm ...

Giảng viên hướng dẫn

Hoàng Khuê

LỜI CẢM ƠN

Lời đầu tiên chúng em xin gửi lời cảm ơn chân thành đến thầy HOÀNG KHUÊ trong bộ môn Công nghệ thông tin – Trường Đại học Công Nghệ Sài Gòn, người đã hướng dẫn tận tình và giúp đỡ cho chúng em để hoàn thành đồ án tin học này.

Ở đây, chúng em không chỉ học được kiến thức về sách vở mà chúng em còn học được các bài học, kỹ năng sống trước khi tạm biệt mái trường đại học thân yêu này và tiến ra biển đời mênh mông rộng lớn. Đặc biệt, không thể thiếu đó là người thầy HOÀNG KHUÊ, người đã đồng hành cùng chúng em trong suốt quá trình làm đồ án tin học, người đã bỏ thời gian quý báu, thậm chí là thời gian nghỉ ngơi để hướng dẫn, để định hướng đường đi nước bước cho chúng em. Chúng em thật chẳng biết dùng lời nào để diễn tả được công lao của thầy.

Trong quá trình học tập và tìm hiểu chúng em đã nỗ lực rất nhiều với mong muốn hoàn thành đồ án tin học một cách tốt nhất, nhưng những thiếu sót sẽ không thể tránh khỏi. Chúng em mong thầy có thể thông cảm và cho chúng em những ý kiến, đóng góp để chỉ dạy chúng em có thể hoàn thành đồ án tin học của mình một cách trọn vẹn nhất.

Sau cùng, chúng em xin kính chúc Thầy lời chúc sức khỏe, luôn hạnh phúc và thành công hơn nữa trong công việc cũng như trong cuộc sống.

Chúng em xin chân thành cảm ơn!

GIỚI THIỆU SƠ LƯỢC

1. Tên đề tài

Xây dựng chương trình quản lý phòng game.

2. Khảo sát nghiệp vụ

Trong những năm gần đây, công nghệ thông tin không ngừng phát triển một cách mạnh mẽ và nhanh chóng. Sự ra đời của công nghệ thông tin đã làm thay đổi bộ mặt xã hội, đời sống con người theo hướng tích cực, đóng góp to lớn cho sự phát triển của nhân loại.

Cũng với lẽ đó, chiếc máy tính đã trở thành thân thuộc với mỗi con người. Cùng với đó là nhu cầu về mạng Internet không thể tách rời. Điều này khiến cho tốc độ và mật độ các phòng game tăng lên nhanh chóng. Nhu cầu vào mạng Internet của con người ngày càng cao. Vì thế các phòng game khi được mở ra đòi hỏi phải có giải pháp hợp lý để quản lý phòng game một cách tốt nhất. Giải pháp tối ưu nhất là dùng các phần mềm về quản lý phòng game. Điều quan trọng nhất của một phần mềm quản lý và vấn đề bảo mật. Hệ thống chương trình sẽ được xây dựng theo 2 luồng truy cập đó là quản lý và nhân viên. Với người quản lý sẽ có đầy đủ các chức năng quản lý máy tính, quản lý nhân viên, xem tình trạng máy, xem lịch sử thanh toán, xem lịch sử đăng nhập, kiểm tra doanh thu, mở máy, thanh toán, thiết lập giá tiền. Với nhân viên chỉ có quyền truy cập vào 3 chức năng chính đó là xem tình trạng máy, mở máy và thanh toán. Quá trình đăng nhập được thực hiện thông qua 3 bước:

- Bước 1: Nhập tài khoản và mật khẩu.
- Bước 2: Nếu là quản lý thì cho phép truy cập các chức năng của quản lý. Nếu là nhân viên thì cho phép truy cập các chức năng của nhân viên.
- Bước 3: Hệ thống sẽ lưu thông tin tài khoản và thời gian đăng nhập vào file txt để quản lý.

Người quản lý có thể sử dụng hệ thống để thêm hoặc xóa thông tin máy tính khỏi danh sách máy tính đang được sở hữu với 2 loại máy đó là máy thường và máy cao cấp. Quá trình thêm máy được thực hiện thông qua 4 bước:

- Bước 1: Người quản lý sử dụng hệ thống lựa chọn đến chức năng thêm máy.
- Bước 2: Nhập thông tin số máy cần thêm.

- Bước 3: Nhập thông tin loại máy cần thêm (thường hoặc cao cấp).
- Bước 4: Hệ thống sẽ lưu thông tin máy tính vào file txt để quản lý.

Quá trình xóa máy được thực hiện thông qua 2 bước:

- Bước 1: Người quản lý sử dụng hệ thống lựa chọn đến chức năng xóa máy.
- Bước 2: Nhập mã máy tính cần xóa
- Bước 3: Hệ thống sẽ xóa thông tin máy tính đó ra khỏi file txt.

Ở đây, giá tiền sử dụng của máy thường và máy cao cấp sẽ là do người quản lý quyết định và cài đặt. Quá trình chỉnh sửa giá tiền sử dụng được thực hiện thông qua 3 bước:

- Bước 1: Người quản lý sử dụng hệ thống lựa chọn đến chức năng chỉnh sửa giá tiền.
- Bước 2: Nhập thông tin giá tiền máy thường và máy cao cấp.
- Bước 3: Khi kết thúc thao tác, hệ thống sẽ lưu thông tin giá tiền xuống file txt để quản lý.

Khi khách hàng có nhu cầu sử dụng máy tính tại phòng game, khách hàng sẽ ngồi vào một máy tính đang trống và nói với người quản lý hoặc nhân viên mở máy với số máy đó. Quá trình mở máy được thực hiện thông qua 3 bước:

- Bước 1: Khách hàng sẽ ngồi vào một máy tính đang trống và nói với người quản lý để mở máy với số máy đó.
- Bước 2: Khi đó người quản lý hoặc nhân viên sử dụng hệ thống và thao tác cho phép khách hàng truy cập vào máy có số máy đó để sử dụng.
- Bước 3: Khi kết thúc thao tác, hệ thống sẽ lưu thông tin số máy và thời gian bắt đầu sử dụng máy xuống file txt để quản lý.

Khi khách hàng không có nhu cầu sử dụng máy tính tại phòng game nữa thì đi tới quầy thu ngân và nói với người quản lý hoặc nhân viên thanh toán tiền với số máy đó. Quá trình thanh toán được thực hiện thông qua 4 bước:

- Bước 1: Khách hàng đi tới quầy thu ngân và nói với người quản lý hoặc nhân viên thanh toán tiền với số máy đó.
- Bước 2: Khi đó người quản lý hoặc nhân viên sử dụng hệ thống và thao tác tính tiền cho khách hàng với số máy đó.

- Bước 3: Số tiền, loại máy, số máy và thời gian sử dụng sẽ được hiển thị trên màn hình hệ thống. Khi đó người quản lý hoặc nhân viên sẽ thông báo giá tiền với khách hàng và nhận tiền từ khách.
- Bước 4: Sau khi khách hàng đã thanh toán thì người quản lý hoặc nhân viên sẽ xác nhận trong hệ thống là kết thúc phiên thanh toán đó. Thì hệ thống sẽ lưu lịch sử thanh toán của khách xuống file txt để quản lý.

Doanh thu sẽ được tổng kết vào thứ thứ 2 mỗi tuần. Quá trình kiểm tra doanh thu được thực hiện thông qua 3 bước:

- Bước 1: Người quản lý sử dụng hệ thống lựa chọn đến chức năng kiểm tra doanh thu.
- Bước 2: Lựa chọn muốn xem doanh thu 7 ngày gần nhất hoặc doanh thu ngày tùy chỉnh.
- Bước 3: Hệ thống sẽ truy cập cơ sở dữ liệu và xuất ra màn hình doanh thu.

Khi hết thời gian làm việc thì quản lý hoặc nhân viên sẽ lựa chọn chức năng đăng xuất khỏi hệ thống. Hệ thống sẽ lưu thông tin tài khoản và thời gian đăng xuất vào file txt để quản lý.

Người quản lý có thể sử dụng hệ thống để thêm hoặc xóa thông tin nhân viên khỏi danh sách nhân viên, xem lịch sử thanh toán và lịch sử đăng nhập.

3. Mục đích, yêu cầu

a. Mục đích

Nhằm giúp các chủ phòng game, cơ sở cho thuê phòng game quản lý hiệu quả công việc. Công nghệ thông tin phát triển nên nhu cầu cần có một chương trình để tối ưu hóa hiệu suất, nâng cao hiệu quả công việc, tiết kiệm thời gian, chi phí ghi chép bằng tay thủ công. Hệ thống được xây dựng trên nền tảng ứng dụng Desktop.

b. Yêu cầu

- Yêu cầu công nghệ:
 - Sử dụng ngôn ngữ lập trình C++.
 - Sử dụng danh sách liên kết đơn.
 - Sử dụng cây nhị phân.
 - Sử dụng github.

- Yêu cầu chức năng:
 - Chương trình có đầy đủ chức năng như: thêm máy tính, xóa máy tính, mở máy cho người chơi, thanh toán,...
 - Chương trình cho phép người dùng: quản lý máy tính và người chơi,...
 - Giao diện thân thiện với người dùng và dễ dàng thao tác.

4. Nội dung và phạm vi đề tài

a. Nội dung

- Tổng quan bài toán.
- Tổng quan về các công nghệ đang sử dụng.
- Phân tích và thiết kế chương trình.
- Kiểm tra và chạy thử.
- Kết quả thu được.

b. Phạm vi

- Lý thuyết về danh sách liên kết đơn.
- Bài toán quản lý phòng game.

5. Công nghệ, công cụ và ngôn ngữ lập trình

- Sử dụng ngôn ngữ lập trình C++.
- Sử dụng danh sách liên kết đơn.
- Sử dụng cây nhị phân.
- Sử dụng github.

6. Các kết quả chính dự kiến sẽ đạt được và ứng dụng

Xây dựng thành công chương trình quản lý phòng game đầy đủ chức năng đã nêu trên.

MỤC LỤC

CHƯƠNG 1: MỞ ĐẦU.....	1
1.1. Lý do chọn đề tài	1
1.2. Hướng tiếp cận của đề tài	1
1.3. Mục tiêu nghiên cứu	1
1.4. Đối tượng và phạm vi nghiên cứu	1
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.....	2
2.1. Danh sách liên kết	2
2.1.1. Định nghĩa, tổng quan về danh sách liên kết đơn	2
2.1.2. Phân loại danh sách liên kết	2
2.1.3. Danh sách liên kết đơn	2
2.2. Cây nhị phân.....	7
2.2.1. Biểu diễn cây nhị phân	7
2.2.2. Các thao tác trên cây nhị phân	8
2.3. Ngôn ngữ lập trình C++	12
2.3.1. Giới thiệu	12
2.3.2. Các đặc điểm của ngôn ngữ lập trình C++	12
2.4. Github	13
2.4.1. Tổng quan về Github	13
2.4.2. Khái niệm	13
2.4.3. Tính năng.....	13
2.4.4. Lợi ích Github mang lại cho lập trình viên	13
2.4.5. Các thuật ngữ cơ bản của Github	14

CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ.....	15
3.1. Đặc tả bài toán.....	15
3.2. Yêu cầu hệ thống.....	15
3.3. Phân tích và thiết kế chương trình.....	15
3.4. Cấu trúc dữ liệu chương trình	15
3.5. Các chức năng của chương trình	19
3.5.1. Thêm máy tính.....	19
3.5.2. Xóa máy tính	20
3.5.3. Thêm nhân viên	21
3.5.4. Xóa nhân viên.....	22
3.5.5. Xem danh sách nhân viên.....	22
3.5.6. Xem danh sách máy đầy.....	23
3.5.7. Xem danh sách máy trống	23
3.5.8. Xem danh sách tất cả máy	24
3.5.9. Xem lịch sử.....	24
3.5.10. Mở máy	25
3.5.11. Thiết lập giá tiền.....	25
CHƯƠNG 4: THỬ NGHIỆM ỨNG DỤNG	26
4.1 Giao diện đăng nhập.....	26
4.2. Giao diện sau khi đăng nhập	27
4.3. Giao diện quản lí máy tính	28
4.3.1. Giao diện thêm máy tính	28
4.3.2. Giao diện xóa máy tính	29
4.4. Giao diện quản lí nhân viên.....	29
4.4.1 Giao diện thêm nhân viên.....	30

4.4.2. Giao diện xóa nhân viên	31
4.4.3. Giao diện xem danh sách nhân viên	31
4.5. Giao diện xem tình trạng máy	32
4.5.1. Giao diện xem máy trống	33
4.5.2. Giao diện xem máy đầy	33
4.5.3. Giao diện xem tất cả máy	34
4.6. Giao diện xem lịch sử thanh toán	34
4.7. Giao diện xem lịch sử đăng nhập	35
4.8. Giao diện kiểm tra doanh thu	35
4.8.1. Giao diện kiểm tra doanh thu 7 ngày gần nhất	36
4.8.2. Giao diện kiểm tra doanh thu ngày tùy chỉnh	36
4.9. Giao diện mở máy trực tiếp	37
4.10. Giao diện thanh toán	38
4.11. Giao diện thiết lập giá tiền	38
4.12. Giao diện thông tin giới thiệu	39
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	40
5.1. Kết luận	40
5.2. Hướng phát triển	40
TÀI LIỆU THAM KHẢO	41

CHƯƠNG 1: MỞ ĐẦU

1.1. Lý do chọn đề tài

Chúng ta, thế hệ đang sống trong thời đại công nghiệp hóa, hiện đại hóa. Thời đại mà công nghệ thông tin đang ngày một phát triển như vũ bão. Nó có mặt hầu hết trong tất cả các lĩnh vực mà chúng ta đang sinh hoạt, làm việc và cũng như giải trí...Nó giúp đỡ cho chúng ta từ nông nghiệp, công nghiệp, kinh tế, giáo dục, y tế.

Chính vì thế chương trình quản lý phòng game là một giải pháp tối ưu nhằm:

- Tiết kiệm thời gian tối đa.
- Tiết kiệm chi phí.
- Quản lý hiệu quả, khoa học.
- Tạo ra sự tiện ích cho chủ phòng game.

1.2. Hướng tiếp cận của đề tài

- Nắm vững kiến thức nền tảng của ngôn ngữ lập trình C++, tiến đến phát triển giao diện.
- củng cố lại kiến thức về danh sách liên kết đơn, cây nhị phân.
- Tiến hành kiểm tra và chạy thử phần mềm.

1.3. Mục tiêu nghiên cứu

- Phát triển thành công chương trình quản lý phòng game.
- Khiến ứng dụng trở nên có ích hơn cho xã hội.

1.4. Đối tượng và phạm vi nghiên cứu

- Đối tượng : Các chủ phòng game trung bình và nhỏ.
- Phạm vi ứng dụng : Mong muốn phần mềm có thể có mặt hầu hết trong các phòng game trung bình và nhỏ.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Danh sách liên kết

2.1.1. Định nghĩa, tổng quan về danh sách liên kết đơn

Danh sách liên kết là danh sách mà các phần tử (Node) liên kết với nhau nhờ vào vùng liên kết của chúng. Mỗi node bao gồm 2 phần: phần Data dùng để chứa dữ liệu cần xử lý và phần liên kết dùng để liên kết tới các node khác.

2.1.2. Phân loại danh sách liên kết

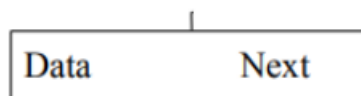
Tùy cách liên kết giữa các phần tử, danh sách liên kết gồm có nhiều loại khác nhau:

- Danh sách liên kết đơn: mỗi phần tử liên kết với phần tử đứng sau nó trong danh sách.
- Danh sách liên kết đôi/kép: mỗi phần tử liên kết với các phần tử đứng trước và sau nó trong danh sách.
- Danh sách liên kết vòng: phần tử cuối danh sách liên kết với phần tử đầu danh sách.

2.1.3. Danh sách liên kết đơn

2.1.3.1. Khái niệm

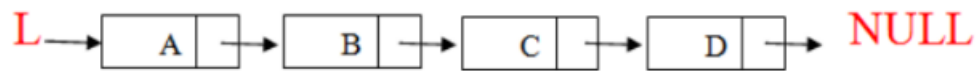
Danh sách liên kết đơn là một tập hợp các Node được phân bố động, được sắp xếp theo cách sao cho mỗi Node chứa một giá trị (Data) và một con trỏ (Next). Con trỏ sẽ trỏ đến phần tử kế tiếp của danh sách liên kết đó. Nếu con trỏ mà trỏ đến NULL, nghĩa là đó là phần tử cuối cùng của link list. Các phần tử trong danh sách không thể truy cập trực tiếp, muốn truy cập bất kỳ phần tử nào đó trong danh sách phải xuất phát từ phần tử đầu tiên. Mỗi Node trong danh sách có thể được biểu diễn như sau:



Cú pháp:

```
Struct Node
{
    Kiểu_Dữ_Liệu Data;
    Node *Next;
};
```

Có thể hình dung danh sách liên kết qua hình vẽ như sau:



2.1.3.2. Thao tác trên 1 Node

- Khai báo biến con trỏ p là biến con trỏ kiểu Node: Node *p
- Cấp phát bộ nhớ cho Node mới – được trỏ bởi con trỏ p:
- Node *p = new Node;
- Truy cập vào trường info: p->info
- Truy cập vào trường next: p->next
- Xóa Node được trỏ bởi p: delete p

2.1.3.3. Các thao tác trên danh sách liên kết đơn

Thao tác trên danh sách liên kết khác với thao tác trên mảng khi cần thêm Node vào danh sách liên kết thì ta xin cấp phát bộ nhớ cho Node và nối Node đó vào danh sách liên kết và khi không cần sử dụng thì ta giải phóng nó khỏi danh sách. Truy cập trên danh sách cũng khác so với trên mảng. Nếu ở mảng ta truy cập trực tiếp thông qua chỉ số mảng (thông qua địa chỉ ô nhớ) thì ở danh sách liên kết đơn ta phải truy cập một cách tuần tự. Khi làm việc trên danh sách liên kết đơn cũng tương tự như vậy nên ta cần chú ý những điểm sau:

- Danh sách luôn có con trỏ đầu danh sách.
- Danh sách luôn có giá trị báo kết thúc danh sách.
- Trường Next của mỗi Node chỉ chứa địa chỉ của Node sau đó, trừ Node cuối.
- Không tách danh sách thành 2 danh sách con nếu danh sách phần sau chưa có con trỏ tới.
- Khởi tạo danh sách rỗng.

- **Thêm một phần tử (node) vào danh sách:**

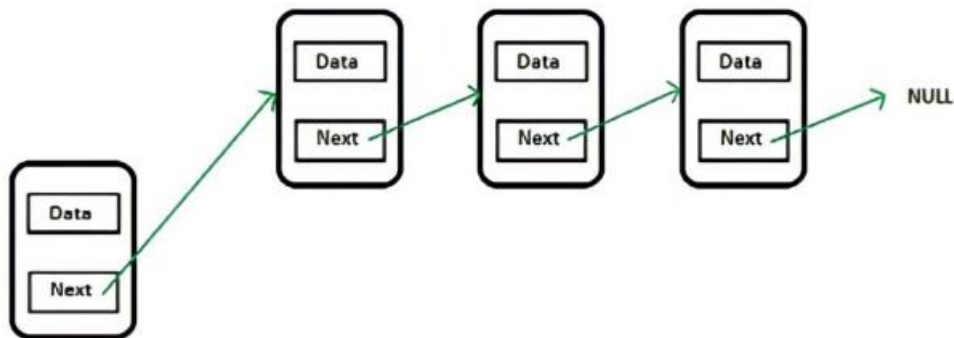
• Trường hợp 1: Thêm phần tử vào đầu danh sách:

- Bước 1: Cho con trỏ của node cần thêm (node p) liên kết với node đầu danh sách pHead.

$p \rightarrow pNext = l.pHead;$

- Bước 2: Cập nhật lại node đầu pHead là node vừa thêm (node p).

$l.pHead = p;$



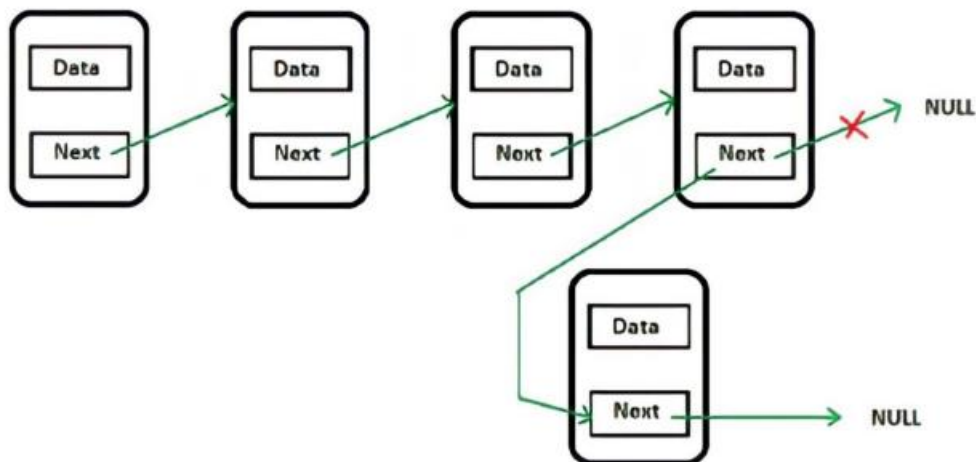
• Trường hợp 2: Thêm phần tử vào cuối danh sách:

- Bước 1: Cho con trỏ của node cuối danh sách pTail liên kết với node cần thêm (node p).

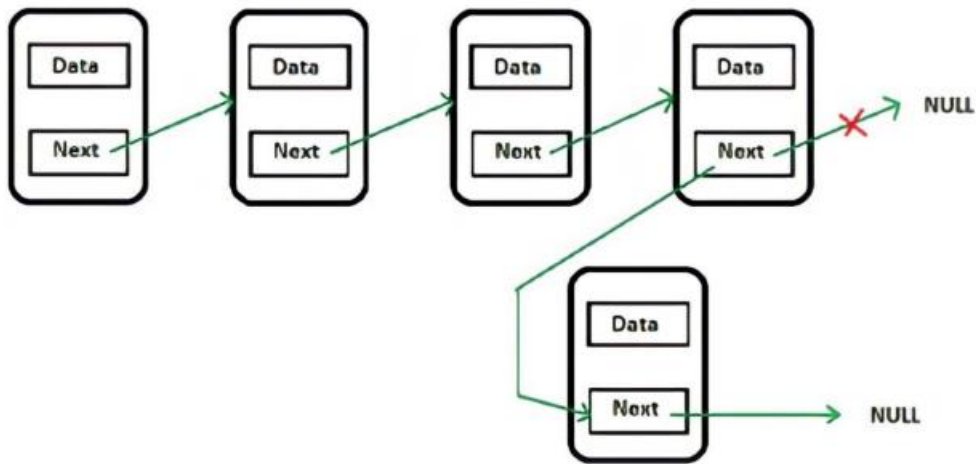
$l.pTail \rightarrow pNext = p;$

- Bước 2: Cập nhật lại node cuối pTail chính là node mới thêm (node p).

$l.pTail = p;$



- Trường hợp 3: Thêm phần tử vào vị trí bất kỳ:



- **Xóa một phần tử ra khỏi danh sách:**

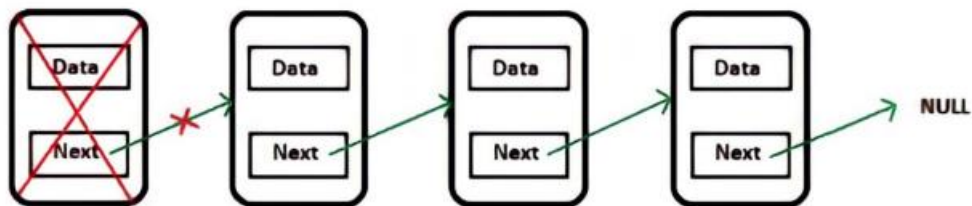
- Trường hợp 1: Xóa phần tử (p) đầu danh sách:

- Bước 1: Cho con trỏ node đầu danh sách (pHead) trở đến node sau nó (pNext).

$l.pHead = l.pHead \rightarrow pNext;$

- Bước 2: Xóa đi node (p) nằm ở đầu danh sách.

`delete p;`



- Trường hợp 2: Xóa phần tử cuối danh sách:

- Bước 1 : Nếu danh sách có một phần tử thực hiện hàm xóa đầu (Trường hợp 1).
- Bước 2 : Nếu danh sách có nhiều phần tử ta thực hiện duyệt danh sách từ đầu đến kế cuối.

$for(NODE *k = l.pHead; k \neq NULL; k = k \rightarrow pNext)$

- Bước 3 : Phát hiện phần tử cần xóa ở vị trí kế cuối.

$if(k \rightarrow pNext == l.pTail)$

- Bước 4 : Xóa đi phần tử cuối.

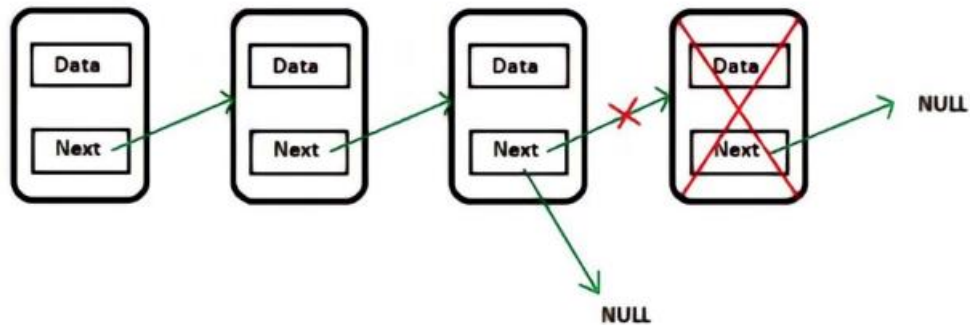
`delete l.pTail;`

- Bước 5 : Cho con trỏ kế cuối trở về NULL.

`k->pNext = NULL;`

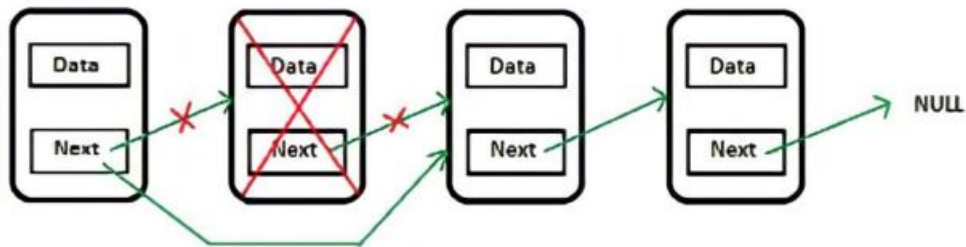
- Bước 6 : Cập nhập lại pTail.

`l.pTail = k;`



- Trường hợp 3 : Xóa phần tử ở vị trí bất kì trong danh sách :

- Bước 1 : Nếu node cần xóa nằm đầu danh sách thì xóa đầu (Trường hợp 1).
- Bước 2 : Nếu node cần xóa nằm cuối danh sách thì xóa cuối (Trường hợp 2).
- Bước 3 : Nếu node cần xóa không nằm ở đầu hoặc cuối danh sách:
 - Duyệt danh sách liên kết đơn.
 - Phát hiện node cần xóa ta cập nhập mới liên kết của node trước node cần xóa với node sau node cần xóa.
 - Xóa phần tử cần xóa.



- Tìm kiếm phần tử trong danh sách :

- Bước 1: Cho con trỏ p trở vào đầu danh sách liên kết (l) và

khởi tạo 1 biến int found = 0

```
Node *p = l.pHead;

int found = 0;
```

- Bước 2: Thực hiện duyệt các phần tử từ đầu đến cuối danh sách kiểm tra nếu có data = X thì trả về địa chỉ của Node đó, ngược lại thì cho con trỏ p trở đến vị trí phần tử kế tiếp

```
while ((p != NULL)&&(p->data != x)) {
    p = p->pNext;
}

return p;
```

2.2. Cây nhị phân

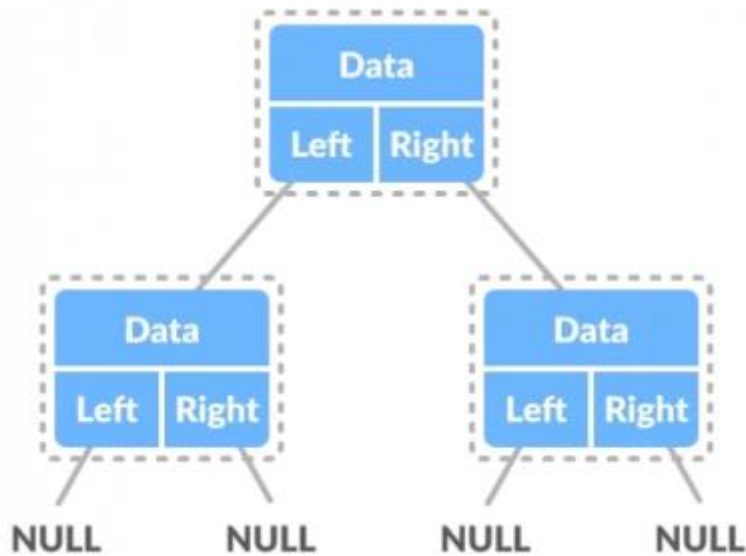
2.2.1. Biểu diễn cây nhị phân

Cây nhị phân là một cấu trúc dữ liệu bao gồm các nút được kết nối với nhau theo quan hệ cha con với mỗi nút cha có tối đa 2 nút con. Mỗi nút của cây nhị phân sẽ lưu các thông tin sau:

- Giá trị lưu tại nút đó. Giá trị này có thể là bất kỳ kiểu dữ liệu nào. Ví dụ, cây nhị phân lưu trữ các số nguyên thì kiểu dữ liệu là int.
- Địa chỉ nút gốc của cây con bên trái.
- Địa chỉ nút gốc của cây con bên phải.

Chúng ta có thể sử dụng danh sách liên kết để biểu diễn cây nhị phân. Đầu tiên, cần định nghĩa một nút trong cấu trúc dữ liệu dạng cây.

```
struct tNode{
    int data;
    tNode *pLeft, *pRight;
};
```



Để lưu trữ cây, chúng ta chỉ cần xác định nút gốc của cây.

```
tNode *root;
```

2.2.2. Các thao tác trên cây nhị phân

Có nhiều thao tác trên cây nhị phân như:

- Chèn một nút mới vào cây nhị phân
- Xóa một nút ra khỏi cây nhị phân
- Duyệt cây nhị phân
- Tìm kiếm trên cây nhị phân
- ...

- **Chèn một nút mới vào cây nhị phân để làm con của nút p:**

```
void insertNode(tNode *p, int value){  
    tNode *node = newNode(value);  
    if (p->pLeft == NULL){  
        p->pLeft = node;  
    }else if (p->pRight == NULL) {  
        p->pRight = node;  
    }else{  
        node->pLeft = p->pLeft;  
        p->pLeft = node;  
    }  
}
```

- **Chèn nút mới vào vị trí gốc của cây:**

```
void insertNewRoot(int value){  
    tNode *node = newNode(value);  
    node->pLeft = root;  
    root = node;  
}
```

- **Duyệt cây nhị phân:**

Có 3 kiểu duyệt chính có thể áp dụng trên cây nhị phân:

- **Duyệt theo thứ tự trước NodeLeftRight (NLR):**

Xử lý nút đang duyệt trước, sau đó mới duyệt đến các cây con bên trái và bên phải.

```
void NLR(tNode *root){  
    if(root!=NULL){  
        if(root!=NULL){  
            cout<<root->data<<" ";  
        }  
        NLR(root->pLeft);  
        NLR(root->pRight);  
    }  
}
```

- **Duyệt theo thứ tự giữa LeftNodeRight (LNR):**

Xử lý cây con bên trái trước, rồi mới xử lý nút đang duyệt, cuối cùng xử lý cây con bên phải.

```
void LNR(tNode *root){
    if(root!=NULL){
        LNR(root->pLeft);
        if(root!=NULL){
            cout<<root->data<<" ";
        }
        LNR(root->pRight);
    }
}
```

- **Duyệt theo thứ tự sau LeftRightNode (LRN):**

Xử lý cây con bên trái trước, rồi xử lý cây con bên phải, cuối cùng mới xử lý nút đang duyệt.

```
void LRN(tNode *root){
    if(root!=NULL){
        LRN(root->pLeft);
        LRN(root->pRight);
        if(root!=NULL){
            cout<<root->data<<" ";
        }
    }
}
```

2.3. Ngôn ngữ lập trình C++

2.3.1. Giới thiệu

C++ là một ngôn ngữ lập trình được phát triển bởi Bjarne Stroustrup vào năm 1979 tại Bell Labs. C++ được coi là ngôn ngữ bậc trung (middle-level) như một phần mở rộng của ngôn ngữ lập trình C, hoặc “C với các lớp Class” vì nó bao gồm sự kết hợp của cả các tính năng của ngôn ngữ cấp cao và cấp thấp.

Ngôn ngữ đã được mở rộng đáng kể theo thời gian và C++ hiện nay có nhiều tính năng như:

- Lập trình tổng quát.
- Lập trình hướng đối tượng.
- Lập trình thủ tục.
- Ngôn ngữ đa mẫu hình tự do có kiểu tĩnh.
- Dữ liệu trừu tượng.
- Lập trình đa hình.

2.3.2. Các đặc điểm của ngôn ngữ lập trình C++

Tính phổ biến: C++ là một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới.

Tính thực thi nhanh: Nếu bạn đã sành sỏi về C++ thì bạn có thể lập trình rất nhanh. Một trong những mục tiêu của C++ chính là khả năng thực thi. Và nếu bạn cần thêm các tính năng cho chương trình, C++ cho phép bạn sử dụng ngôn ngữ Assembly (Hợp ngữ) – Ngôn ngữ lập trình bậc thấp nhất dùng để giao tiếp trực tiếp với phần cứng của máy tính.

Thư viện đầy đủ: Có rất nhiều tài nguyên sử dụng cho người lập trình bằng C++, bao gồm cả đồ họa API, 2D, 3D, vật lý các thiết bị âm thanh hỗ trợ giúp cho lập trình viên dễ dàng thực thi.

Đa mô hình: C++ cũng cho phép bạn lập trình theo cấu trúc tuyến tính, hướng chức năng, hướng đối tượng đa dạng tùy theo yêu cầu của người lập trình.

2.4. Github

2.4.1. Tổng quan về Github

GitHub được viết bằng Ruby on Rails và Erlang do Tom Preston-Werner, Chris Wanstrath, và PJ Hyett phát triển trang web được đưa ra và chạy chính thức vào tháng 4 năm 2008.

Tính đến thời điểm tháng 3 năm 2018 Github đang là dịch vụ máy chủ lưu trữ các mã nguồn lập trình lớn nhất thế giới. Với hơn 25 triệu người dùng và hơn 80 triệu mã nguồn dự án, Github đã trở thành một phần không thể thiếu đối với cộng đồng phát triển mã nguồn mở và cộng đồng lập trình viên trên toàn thế giới.

2.4.2. Khái niệm

GitHub là một dịch vụ nổi tiếng cung cấp kho lưu trữ mã nguồn Git cho các dự án phần mềm. Github có đầy đủ những tính năng của Git, ngoài ra nó còn bổ sung những tính năng về social để các developer tương tác với nhau. Git là công cụ giúp quản lý source code tổ chức theo dạng dữ liệu phân tán.

2.4.3. Tính năng

Follow User : tính năng này giúp bạn có thể theo dõi những hoạt động của người khác.

Watch Project : đây là tính năng giúp mọi người có thể theo dõi được các hoạt động của nhiều chương trình khác nhau. Nhờ đó họ có thể hiểu xem quá trình để mọi người phát triển các phần mềm như thế nào, hoạt động nhờ project này ra làm sao. Wiki, thống kê, issue... được đặt vào các namespace và coi là các user.

2.4.4. Lợi ích Github mang lại cho lập trình viên

Dễ dàng Markdown: bạn có thể chỉnh sửa được các cách hiển thị của document, ví dụ như định dạng in nghiêng, in đậm, tạo list, thêm hình... Có thể tracking về sự thay đổi qua nhiều version: nếu bạn có nhiều member cùng thực hiện một dự án, sử dụng Github sẽ giúp bạn lưu lại thay đổi mà bạn đã đẩy lên repository. Có thể nói rằng chức năng này khá giống với Google Drive.

Quản lý các source code cực kỳ dễ dàng: nếu bạn muốn tạo ra repo, các nguồn code từ repo đó đều đã được lưu sẵn ở trên Github. Nó giúp bạn có thể coi lại quá trình mà mình đã làm việc thông qua những bình luận sau mỗi lần mà bạn commit. Trên thực tế, nhiều người có thể cùng làm một repo chứ không phải chỉ một người.

2.4.5. Các thuật ngữ cơ bản của Github

- **git init:** tạo một repo trên máy của bạn.
- **git add:** cập nhập lại project.
- **git clone:** sao chép một repo trên server về máy.
- **git commit:** xác nhận và lưu những thay đổi của project.
- **git push:** cập nhập những thay đổi của project và gửi lên server.
- **git checkout:** tạo một nhánh mới và chuyển qua nhánh đó hoặc chuyển nhánh.
- **git remote:** Xem danh sách remote, kết nối repo của bạn đến máy chủ remote, đổi tên remote.
- **git branch:** tạo nhánh mới, xóa nhánh, kiểm tra nhánh hiện tại.
- **git pull:** lấy source mới nhất trên server về và tiến hành trộn.
- **git merge:** để trộn một nhánh vào nhánh đang hoạt động.
- **git log:** hiển thị danh sách các commit trên một nhánh và các thông tin của nó.
- **git diff:** xem xung đột với tệp cơ sở, với nhánh được trộn trước khi trộn.

CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ

3.1. Đặc tả bài toán

Ngày nay khi mà khoa học kỹ thuật phát triển, đặc biệt là sự bùng nổ công nghệ thông tin thì việc quản lý phòng game sẽ dễ dàng hơn nhiều. Với sự phát triển mạnh của công nghệ thông tin như hiện nay dẫn đến nhu cầu xây dựng chương trình quản lý phòng game. Yêu cầu của bài toán là tạo ra chương trình có thể thực hiện các thao tác quản lý phòng game một cách dễ dàng, tiện lợi dựa trên sự trợ giúp của máy tính. Mọi công việc phải được thao tác trên một vùng dữ liệu chung để đảm bảo việc đồng bộ với nhau trong khâu quản lý.

Một người cần quản lý thông tin về các máy tính của mình. Mỗi máy tính gồm các thuộc tính: mã máy, số máy, kiểu máy, tình trạng máy. Với mỗi nhân viên sẽ có các thuộc tính của nhân viên: mã nhân viên, họ tên, tài khoản, mật khẩu, số điện thoại.

3.2. Yêu cầu hệ thống

Quản lý phòng game là chương trình quản lý thông tin thời gian của khách thuê máy tính trong quá trình thuê máy tính ở phòng game. Chương trình có thể thực hiện các công việc : thêm/xóa máy tính, xem danh sách máy tính, thanh toán, xem tình trạng máy, xem lịch sử thanh toán, xem lịch sử đăng nhập. Chương trình được viết bằng ngôn ngữ C++ và dựa trên cấu trúc lưu trữ của danh sách liên kết đơn và cây nhị phân.

3.3. Phân tích và thiết kế chương trình

3.4. Cấu trúc dữ liệu chương trình

Để chương trình quản lý phòng game được chặt chẽ thì mỗi máy tính, nhân viên, cũng như mỗi máy được thuê có thuộc tính như sau:

- Máy tính:

```

1  class MayTinh {
2      private:
3          string maMay; // mã máy
4
5      public:
6          int soMay; // số máy
7          int kieuMay; // kiểu máy: -0:thường 1:cao cấp
8          bool tinhTrang = 0; // tình trạng máy: -0:trống 1:đã được sử dụng
9          int gioBD = 0, phutBD = 0, giayBD = 0; // thời gian bắt đầu (giờ:phút:giây)
10         int ngayBD = 0, thangBD = 0, namBD = 0; // thời gian bắt đầu (ngày/tháng/năm)
11         int gioKT = 0, phutKT = 0, giayKT = 0; // thời gian kết thúc (giờ:phút:giây)
12         int ngayKT = 0, thangKT = 0, namKT = 0; // thời gian kết thúc (ngày/tháng/năm)
13         long giaTien = 0, giaTienThuong = 0, giaTienCaoCap = 0; // giá tiền
14         void themMayTinh(MayTinh[], int &, int &); // thêm một máy tính
15         void xoaMayTinh(MayTinh[], int &, int &); // xóa một máy tính
16         void inMotMayTheoChieuNgang(); // in một máy theo chiều ngang
17         void inMotMayTheoChieuNgangCoThoiGian(); // in một máy theo chiều ngang có thời gian truy cập
18         void docMotMay(ifstream &); // đọc thông tin một máy tính từ file
19         void ghiMotMay(ofstream &); // ghi thông tin một máy tính ra file
20         void docGiaTien(); // đọc giá tiền từ file
21         void ghiGiaTien(); // ghi giá tiền ra file
22         void thietLapGiaTien(); // thiết lập giá tiền
23
24         /** Getters, setters
25         string getter_MaMay() { return maMay; };
26         void setter_MaMay(string MAMAY) { maMay = MAMAY; };
27     };

```

- Nhân viên:

```

1  struct NhanVien {
2      int ma;
3      string ten;
4      string ho;
5      string taiKhoan;
6      string matKhai;
7      string soDienThoai;
8  };
9
10 struct NodeT {
11     NhanVien nv;
12     NodeT *left;
13     NodeT *right;
14 };
15 typedef struct NodeT *DanhSachNhanVien;

```

- Thuê một máy:

```

1  class ThueMotMay {
2      public:
3          MayTinh *maytinh = NULL;           // trỏ đến máy
4          void chonMay(MayTinh[], int);       // chọn máy
5          void chonMayCanThanhToan(MayTinh[], int); // chọn máy cần thanh toán
6          void docMotNguoiThueTrucTiep(ifstream &, MayTinh[], int); // đọc một người thuê trực tiếp từ file
7          void ghiMotNguoiThueTrucTiep(ofstream &); // ghi một người thuê trực tiếp ra file
8  };

```

- Thuê nhiều máy:

```

1  struct Node {
2      ThueMotMay data;
3      Node *pNext;
4  };
5
6  class ThueNhiềuMay {
7  public:
8      Node *pHead;
9      Node *pTail;
10     void taoListThueMotMay();           // tạo List
11     void themNodeVaoCuoi(ThueMotMay);   // thêm Node vào cuối
12     void xoaNodeDauTien();               // xóa Node đầu tiên
13     void xoaNodeCuoiCung();              // xóa Node cuối cùng
14     void giaiPhongDanhSachNguoiThueTrucTiep();
15     void moMayTrucTiep(MayTinh[], int);  // mở máy trực tiếp
16     void thanhToan(MayTinh[], int);      // thanh toán
17 };

```

- Admin:

```

1  class Admin {
2  private:
3      string taiKhoanAdmin = "admin",
4          matKhuAdmin = "admin";
5
6  public:
7      bool laAdmin(string taiKhoan, string matKhu) {
8          if (taiKhoan == taiKhoanAdmin && matKhu == matKhuAdmin)
9              return true;
10             return false;
11         }
12         string getTaiKhoanAdmin() { return taiKhoanAdmin; }
13         string getMatKhuAdmin() { return matKhuAdmin; }
14     };

```

3.5. Các chức năng của chương trình

3.5.1. Thêm máy tính

```

1 void MayTinh::themMayTinh(MayTinh nhieuMay[], int &n, int &soMay) {
2     MayTinh mt;
3     int soMayTam;
4     mt.maMay = taoMaMayTinh(nhieuMay, n);
5     string fileName = "../File/maytinh/danh sach may tinh.txt";
6     int x =kiemTraFileTrong(fileName);
7     if (x != -1) {
8         do {
9             cout << "\n(?) Nhập số máy: ";
10            cin >> mt.soMay;
11            soMayTam = n + 1;
12            if (kiemTraTrungSoMay(nhieuMay, n, mt.soMay) != -1) {
13                cout << bright_red << "\n\t(!) Số máy đã tồn tại. Xin hãy nhập lại" << reset << "\n";
14            } else if (mt.soMay > soMayTam) {
15                cout << bright_red << "\n\t(!) Số máy chỉ được lớn hơn số máy lớn nhất hiện tại 1 đơn vị. Xin hãy nhập lại" << reset << "\n";
16            }
17        } while (kiemTraTrungSoMay(nhieuMay, n, mt.soMay) != -1 || mt.soMay > soMayTam);
18    } else {
19        do {
20            cout << "\n(?) Nhập số máy: ";
21            cin >> mt.soMay;
22            if (mt.soMay != 1) {
23                cout << bright_red << "\n\t(!) Số máy khởi đầu mặc định sẽ là 1. Xin hãy nhập lại" << reset << "\n";
24            }
25        } while (mt.soMay != 1);
26    }
27    soMay = mt.soMay;
28    do {
29        cout << "\n(?) Nhập kiểu máy (0: Thường - 1: Cao cấp): ";
30        cin >> mt.kieuMay;
31        if (mt.kieuMay < 0 || mt.kieuMay > 1) {
32            cout << bright_red << "\n\t(!) Kiểu máy không hợp lệ. Xin hãy nhập lại" << reset << "\n";
33        }
34    } while (mt.kieuMay < 0 || mt.kieuMay > 1);
35    for (int i = n - 1; i >= 0; i--)
36        nhieuMay[i + 1] = nhieuMay[i];
37    nhieuMay[0] = mt;
38    n++;
39    cout << bright_green << "\n\t(!) Thêm máy tính thành công" << reset << "\n";
40    cout << "\n\t(Nhấn ENTER để tiếp tục...)";
41    while (true) {
42        int key = _getch();
43        if (key == KEY_ENTER)
44            break;
45    }
46 }

```

3.5.2. Xóa máy tính

```

1 void MayTinh::xoaMayTinh(MayTinh nhieuMay[], int &n, int &kiemTra) {
2     string str;
3     cout << "\n(?) Nhập mã máy tính cần xóa: ";
4     cin >> str;
5     int vitri = kiemTraTrungMaMayTinh(nhieuMay, n, str);
6     // Dời
7     if (vitri < 0) {
8         cout << bright_red << "\n\t(!) Mã máy tính không tồn tại. Xin hãy nhập lại" << reset << "\n";
9         kiemTra = 0;
10        cout << "\n\t[Nhấn ENTER để tiếp tục...]";
11        while (true) {
12            int key = _getch();
13            if (key == KEY_ENTER)
14                break;
15        }
16    } else {
17        for (int i = vitri; i < n - 1; i++) {
18            nhieuMay[i].maMay = nhieuMay[i + 1].maMay;
19            nhieuMay[i].soMay = nhieuMay[i + 1].soMay;
20            nhieuMay[i].kieuMay = nhieuMay[i + 1].kieuMay;
21            nhieuMay[i].tinhTrang = nhieuMay[i + 1].tinhTrang;
22        }
23        // Giảm số lượng
24        MayTinh tam = nhieuMay[n - 1];
25        n--;
26        cout << bright_green << "\n\t(!) Xóa máy tính thành công" << reset << "\n";
27        kiemTra = 1;
28        cout << "\n\t[Nhấn ENTER để tiếp tục...]";
29        while (true) {
30            int key = _getch();
31            if (key == KEY_ENTER)
32                break;
33        }
34    }
35 }

```

3.5.3. Thêm nhân viên

```

1 void nhapMotNhanVien(DanhSachNhanVien &dsnv) {
2     NhanVien nv;
3     do {
4         cout << "\n(?) Nhập tài khoản:  \t\t";
5         getline(cin, nv.taiKhoan);
6         if (nv.taiKhoan == "admin") {
7             cout << bright_red << "\n\t(!) Tài khoản không được trùng với tài khoản Admin" << reset << "\n";
8         } else if (nv.taiKhoan.length() > 30) {
9             cout << bright_red << "\n\t(!) Tài khoản không được lớn hơn 30 kí tự. Xin hãy nhập lại" << reset << "\n";
10            } else if (kiemTraTaiKhoanNhanVienTrung(nv.taiKhoan) == true)
11            {
12                cout << bright_red << "\n\t(!) Tài khoản đã tồn tại. Xin hãy nhập lại" << reset << "\n";
13            } while (nv.taiKhoan == "admin" || nv.taiKhoan.length() > 30 || kiemTraTaiKhoanNhanVienTrung(nv.taiKhoan) == true);
14            xoaKhoangTrangThua(nv.taiKhoan);
15        }
16        do {
17            cout << "\n(?) Nhập mật khẩu:  \t\t";
18            getline(cin, nv.matKhu);
19            if (nv.matKhu.length() > 30) {
20                cout << bright_red << "\n\t(!) Mật khẩu không được lớn hơn 30 kí tự. Xin hãy nhập lại" << reset << "\n";
21            } while (nv.matKhu.length() > 30);
22            xoaKhoangTrangThua(nv.matKhu);
23        }
24        do {
25            cout << "\n(?) Nhập số điện thoại (10 số):\t";
26            getline(cin, nv.soDienThoai);
27            if (nv.soDienThoai.length() <= 0 || nv.soDienThoai.length() > 10 || nv.soDienThoai.length() != 10) {
28                cout << bright_red << "\n\t(!) Số điện thoại không hợp lệ. Xin hãy nhập lại" << reset << "\n";
29            } while (nv.soDienThoai.length() <= 0 || nv.soDienThoai.length() > 10 || nv.soDienThoai.length() != 10);
30        }
31        nv.ma = taoMaNhanVien(dsnav);
32        do {
33            cout << "\n(?) Nhập tên:  \t\t\t";
34            cin >> nv.ten;
35            if (nv.ten.length() > 8) {
36                cout << bright_red << "\n\t(!) Tên không được lớn hơn 8 kí tự. Xin hãy nhập lại" << reset << "\n";
37            } while (nv.ten.length() > 8);
38            xoaKhoangTrangThua(nv.ten);
39            vietHoaKiTuDauMoiTu(nv.ten);
40            cin.ignore();
41        }
42        do {
43            cout << "\n(?) Nhập họ:  \t\t\t";
44            cin >> nv.ho;
45            if (nv.ho.length() > 8) {
46                cout << bright_red << "\n\t(!) Họ không được lớn hơn 8 kí tự xin hãy nhập lại" << reset << "\n";
47            } while (nv.ho.length() > 8);
48            xoaKhoangTrangThua(nv.ho);
49            vietHoaKiTuDauMoiTu(nv.ho);
50            cin.ignore();
51        }
52        themNhanVien(dsnav, taoNodeNhanVien(nv));
53        system("cls");
54        cout << bright_green << "\n\t(!) Tạo tài khoản thành công. Mã tài khoản là " << nv.ma << reset << "\n";
55    }
56 }
57
58 void themNhanVien(DanhSachNhanVien &dsnv, DanhSachNhanVien nv) {
59     if (dsnv) {
60         if (dsnv->nv.ma == nv->nv.ma) // Trùng
61             return;
62         if (nv->nv.ma < dsnv->nv.ma)
63             themNhanVien(dsnav->left, nv);
64         else
65             themNhanVien(dsnav->right, nv);
66     } else
67         dsnv = nv;
68 }
69 }

```


3.5.4. Xóa nhân viên

```

1 void xoaMotNhanVien(DanhSachNhanVien &dsnv) {
2     int ma;
3     cout << "\n(?) Nhập mã nhân viên cần xóa: ";
4     cin >> ma;
5     xoaNhanVien(dsnv, ma);
6 }
7
8 void xoaNhanVien(DanhSachNhanVien &dsnv, int ma) {
9     if (dsnv) {
10         if (ma > dsnv->nv.ma)
11             xoaNhanVien(dsnv->right, ma);
12         else if (ma < dsnv->nv.ma)
13             xoaNhanVien(dsnv->left, ma);
14         else {
15             DanhSachNhanVien p = dsnv;
16             if (!dsnv->left)
17                 dsnv = dsnv->right;
18             else if (!dsnv->right)
19                 dsnv = dsnv->left;
20             else
21                 timNútNhoNhatBenPhai(p, dsnv->right);
22             delete p;
23             cout << bright_green << "\n\t(!) Xóa thành công" << reset << "\n";
24         }
25     } else
26         cout << bright_red << "\n\t(!) Không tìm thấy nhân viên cần xóa" << reset << "\n";
27 }

```

3.5.5. Xem danh sách nhân viên

```

1 void inMotNhanVien(DanhSachNhanVien dsnv) {
2     if (dsnv) {
3         cout << on_bright_blue << setw(6) << left << "|" << to_string(dsnv->nv.ma) << "|" << reset;
4         cout << on_bright_blue << setw(20) << left << " " << dsnv->nv.ho << " " << dsnv->nv.ten << "|" << reset;
5         cout << on_bright_blue << setw(30) << left << " " << dsnv->nv.taikhoan << "|" << reset;
6         cout << on_bright_blue << setw(30) << left << " " << dsnv->nv.matkhau << "|" << reset;
7         cout << on_bright_blue << setw(14) << left << " " << dsnv->nv.soDienThoai << "|" << reset << "\n";
8         inMotNhanVien(dsnv->left);
9         inMotNhanVien(dsnv->right);
10    }
11 }
12
13 void inDanhSachNhanVien(DanhSachNhanVien dsnv) {
14     cout << on_bright_blue << "*-----*-----*-----*-----*-----*-----* << reset << "\n";
15     cout << on_bright_blue << setw(7) << left << "| Mã"
16     << "|" << reset;
17     cout << on_bright_blue << setw(23) << left << " Họ tên"
18     << "|" << reset;
19     cout << on_bright_blue << setw(33) << left << " Tài khoản"
20     << "|" << reset;
21     cout << on_bright_blue << setw(34) << left << " Mật khẩu"
22     << "|" << reset;
23     cout << on_bright_blue << setw(18) << left << " Số điện thoại"
24     << "|" << reset << "\n";
25     cout << on_bright_blue << "*-----*-----*-----*-----*-----*-----* << reset << "\n";
26     inMotNhanVien(dsnv);
27     cout << on_bright_blue << "*-----*-----*-----*-----*-----*-----* << reset << "\n";
28 }

```

3.5.6. Xem danh sách máy đầy

```

1 void xuatDanhSachCacMayDay(MayTinh nhieuMay[], int n) {
2     if (n != 0) {
3         // nếu không có máy nào được bật
4         if (!kiemTraTrangThai(nhieuMay, n))
5             cout << bright_red << "\n\t(!) Không có máy đang được sử dụng" << reset << "\n";
6         else {
7             cout << on_bright_blue << "*-----*-----*-----*-----*-----*-----*-----*-----*-----*" << reset << "\n";
8             cout << on_bright_blue << setw(12) << left << "| Mã máy"
9                 << "|" << reset;
10            cout << on_bright_blue << setw(13) << left << "| Số máy"
11                << "|" << reset;
12            cout << on_bright_blue << setw(23) << left << "| Kiểu máy"
13                << "|" << reset;
14            cout << on_bright_blue << setw(23) << left << "| Tình trạng"
15                << "|" << reset;
16            cout << on_bright_blue << setw(30) << left << "| Thời gian bắt đầu"
17                << "|" << reset << "\n";
18            cout << on_bright_blue << "*-----*-----*-----*-----*-----*-----*-----*-----*-----*\n";
19            for (int i = 0; i < n; i++)
20                if (nhieuMay[i].tinhTrang == 1)
21                    nhieuMay[i].inMotMayTheoChieuNgangCoThoiGian();
22            cout << on_bright_blue << "*-----*-----*-----*-----*-----*-----*-----*-----*-----*" << reset << "\n";
23        }
24    } else
25        cout << bright_red << "\n\t(!) Danh sách máy tính trống" << reset << "\n";
26 }

```

3.5.7. Xem danh sách máy trống

```

1 void xuatDanhSachCacMayTrong(MayTinh nhieuMay[], int n) {
2     if (n != 0) {
3         // nếu hết máy
4         if (kiemTraHetMay(nhieuMay, n))
5             cout << bright_red << "\n\t(!) Hết máy" << reset << "\n";
6         else {
7             cout << on_bright_blue << "*-----*-----*-----*-----*-----*-----*-----*-----*-----*" << reset << "\n";
8             cout << on_bright_blue << setw(12) << left << "| Mã máy"
9                 << "|" << reset;
10            cout << on_bright_blue << setw(13) << left << "| Số máy"
11                << "|" << reset;
12            cout << on_bright_blue << setw(23) << left << "| Kiểu máy"
13                << "|" << reset;
14            cout << on_bright_blue << setw(23) << left << "| Tình trạng"
15                << "|" << reset << "\n";
16            cout << on_bright_blue << "*-----*-----*-----*-----*-----*-----*-----*-----*-----*" << reset << "\n";
17            for (int i = 0; i < n; i++)
18                if (nhieuMay[i].tinhTrang == 0)
19                    nhieuMay[i].inMotMayTheoChieuNgang();
20            cout << on_bright_blue << "*-----*-----*-----*-----*-----*-----*-----*-----*-----*" << reset << "" << reset << "\n";
21        }
22    } else
23        cout << bright_red << "\n\t(!) Danh sách máy tính trống" << reset << "\n";
24 }

```

3.5.8. Xem danh sách tất cả máy

```

1 void xuatDanhSachCacMay(MayTinh nhieuMay[], int n) {
2     if (n != 0) {
3         cout << on_bright_blue << "*-----*-----*-----*-----*-----*-----*-----* << reset << "\n";
4         cout << on_bright_blue << setw(12) << left << "| Mã máy"
5         << "|" << reset;
6         cout << on_bright_blue << setw(13) << left << "| Số máy"
7         << "|" << reset;
8         cout << on_bright_blue << setw(23) << left << "| Kiểu máy"
9         << "|" << reset;
10        cout << on_bright_blue << setw(23) << left << "| Tình trạng"
11        << "|" << reset;
12        cout << on_bright_blue << setw(30) << left << "| Thời gian bắt đầu"
13        << "|" << reset << "\n";
14        cout << on_bright_blue << "*-----*-----*-----*-----*-----*-----*-----* << reset << "\n";
15        for (int i = 0; i < n; i++)
16            nhieuMay[i].inMotMayTheoChieuNgangCoThoiGian();
17        cout << on_bright_blue << "*-----*-----*-----*-----*-----*-----*-----* << reset << "\n";
18    } else
19        cout << bright_red << "\n\t(!) Danh sách máy tính trống" << reset << "\n";
20 }

```

3.5.9. Xem lịch sử

```

1 void docLichSu(string &fileName) {
2     ifstream fileIn(fileName);
3     if (fileIn.fail()) {
4         cout << bright_red << "\n\t(!) Không tìm thấy tập tin" << reset << "\n";
5         cout << "\n\t[Nhấn ENTER để tiếp tục...]";
6         while (true) {
7             int key = _getch();
8             if (key == KEY_ENTER)
9                 break;
10        }
11    } else {
12        string line;
13        vector<string> list;
14        while (getline(fileIn, line)) {
15            list.push_back(line);
16        }
17        fileIn.close();
18        for (int i = 0; i < list.size(); i++)
19            cout << on_bright_blue << list[i] << reset << "\n";
20    }
21 }

```

3.5.10. Mở máy

```

1 void ThueNhiềuMay::moMayTrucTiep(MayTinh nhieuMay[], int n) {
2     if (n != 0) {
3         if (kiemTraHetMay(nhieuMay, n)) { // nếu hết máy
4             system("cls");
5             cout << bright_red << "\n\t(!) Hết máy" << reset << "\n";
6         } else {
7             int gioBD = 0, phutBD = 0, giayBD = 0, ngayBD = 0, thangBD = 0, namBD = 0;
8             ThueMotMay r;
9             r.chonMay(nhieuMay, n);
10            r.maytinh->tinhTrang = 1; // cập nhật lại tình trạng máy → 1 (máy đã có người sử dụng)
11            layThoiGianHeThong(gioBD, phutBD, giayBD, ngayBD, thangBD, namBD);
12            r.maytinh->gioBD = gioBD;
13            r.maytinh->phutBD = phutBD;
14            r.maytinh->giayBD = giayBD;
15            r.maytinh->ngayBD = ngayBD;
16            r.maytinh->thangBD = thangBD;
17            r.maytinh->namBD = namBD;
18            cout << bright_green << "\n\t(!) Mở máy thành công" << reset << "\n";
19            cout << "\n\t[Nhấn ENTER để tiếp tục...]";
20            while (true) {
21                int key = _getch();
22                if (key == KEY_ENTER)
23                    break;
24            }
25        }
26    } else {
27        cout << bright_red << "\n\t(!) Danh sách máy tính trống" << reset << "\n";
28        cout << "\n\t[Nhấn ENTER để tiếp tục...]";
29        while (true) {
30            int key = _getch();
31            if (key == KEY_ENTER)
32                break;
33        }
34    }
35 }

```

3.5.11. Thiết lập giá tiền

```

1 void MayTinh::thietLapGiaTien() {
2     cout << on_bright_blue << "\n*----- GIÁ TIỀN HIỆN TẠI -----*" << reset << "\n";
3     cout << on_bright_blue << setw(17) << left << "| Máy thường:" << reset;
4     cout << on_bright_blue << setw(24) << right << to_string(giaTienThuong) + " VND"
5     << " |" << reset << "\n";
6     cout << on_bright_blue << setw(17) << left << "| Máy cao cấp:" << reset;
7     cout << on_bright_blue << setw(23) << right << to_string(giaTienCaoCap) + " VND"
8     << " |" << reset;
9     cout << on_bright_blue << "\n*-----*" << reset << "\n";
10    cout << "\n(?) Nhập giá tiền máy thường / 1 giờ: ";
11    cin >> giaTienThuong;
12    cout << "\n(?) Nhập giá tiền máy cao cấp / 1 giờ: ";
13    cin >> giaTienCaoCap;
14    cout << bright_green << "\n\t(!) Thiết lập giá tiền thành công" << reset << "\n";
15    ghiGiaTien();
16 }

```

CHƯƠNG 4: THỬ NGHIỆM ỨNG DỤNG

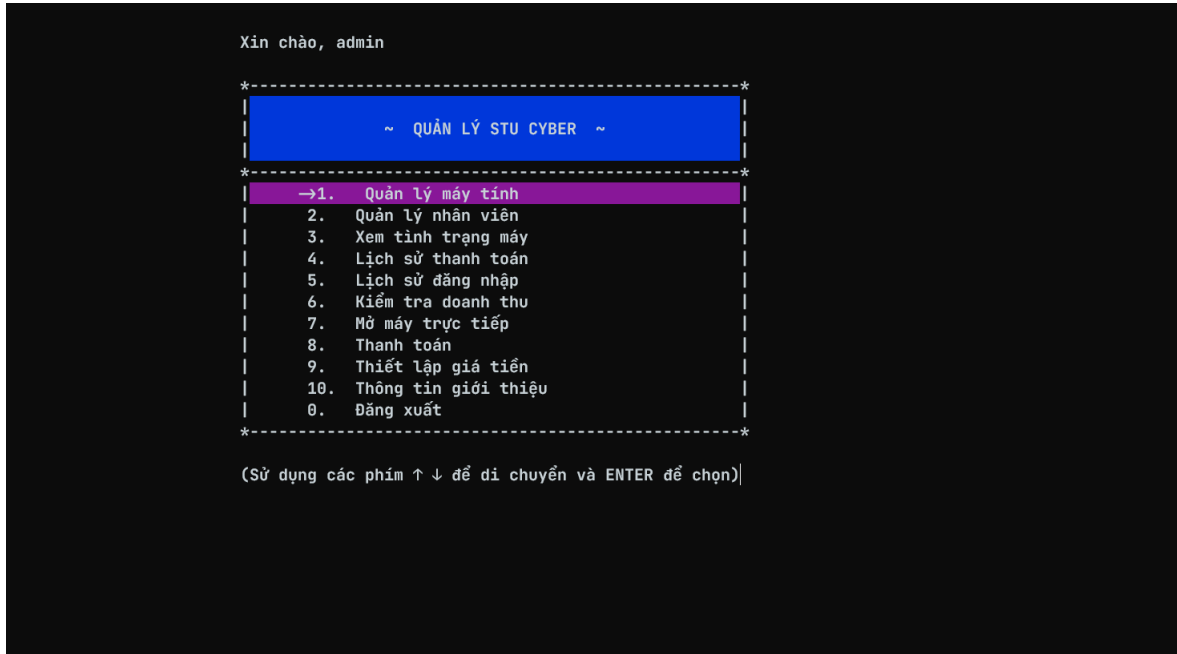
4.1 Giao diện đăng nhập

Cũng như bao ứng dụng, chương trình khác. Đầu tiên chúng ta tiến hành đăng nhập để có quyền truy cập vào hệ thống. Tài khoản và mật khẩu đã được lưu trước đó, chỉ có quản lý và nhân viên mới biết và có quyền truy cập.

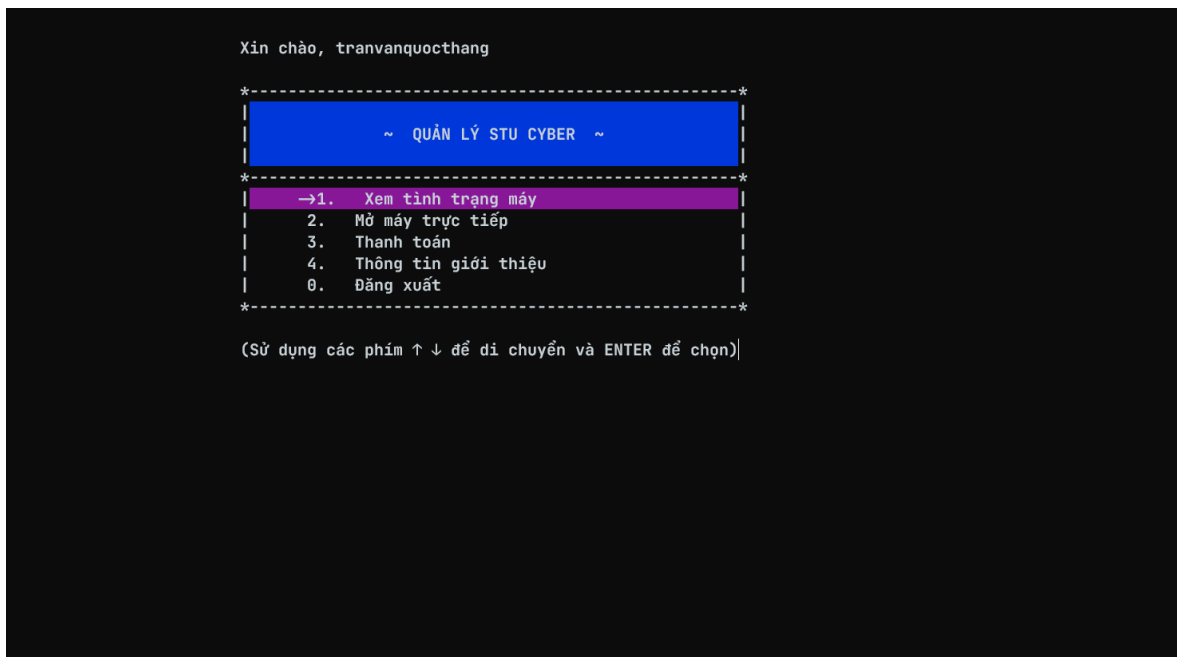


4.2. Giao diện sau khi đăng nhập

Đây là giao diện người quản lý sẽ sử dụng. Có đầy đủ các chức năng hiệu quả để quản lý phòng game.

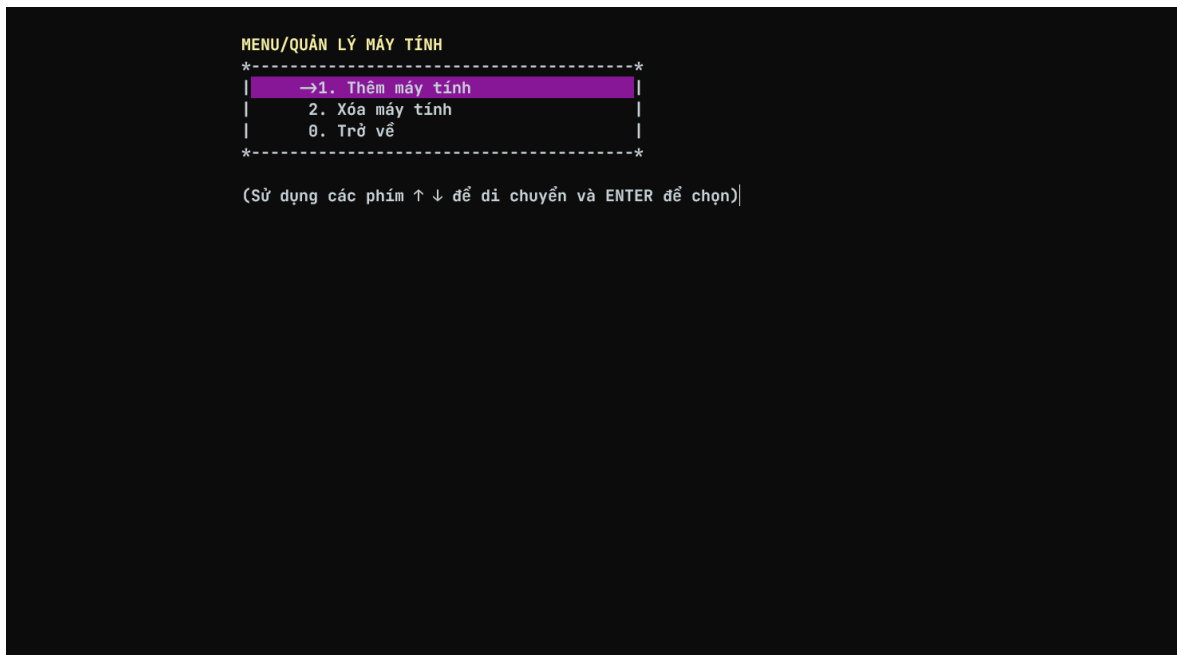


Đây là giao diện nhân viên sẽ sử dụng. Tương tự như của người quản lý nhưng sẽ bị lược bỏ một số chức năng của quản lý.



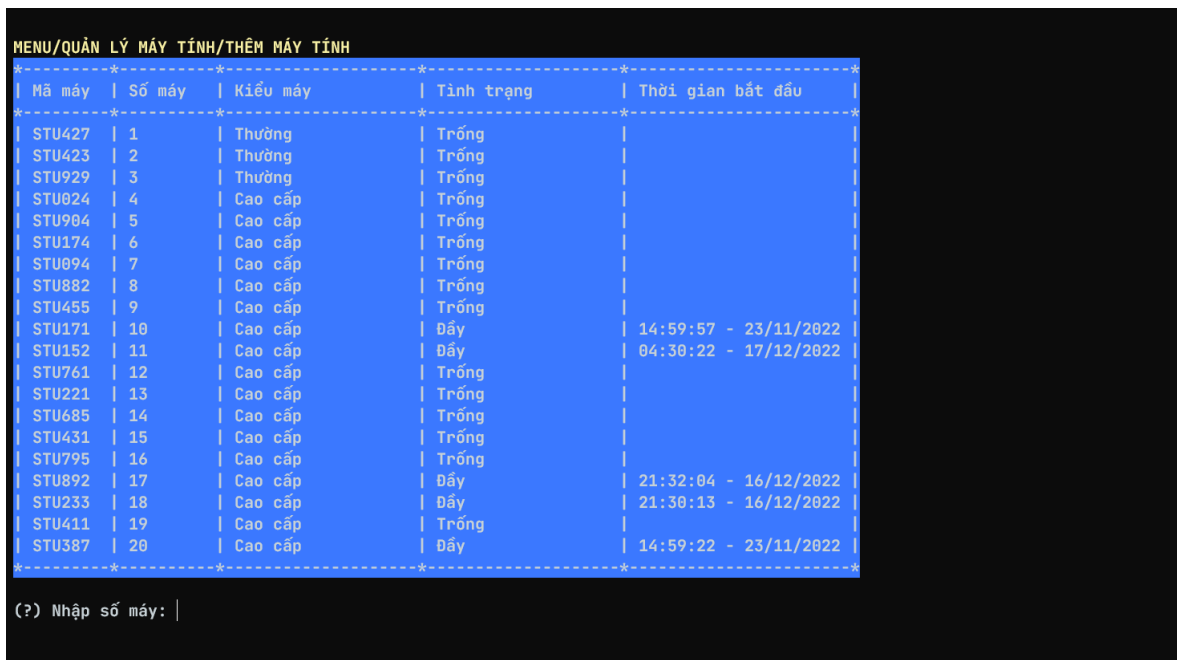
4.3. Giao diện quản lý máy tính

Có 2 tùy chọn là thêm hoặc xóa máy tính.



4.3.1. Giao diện thêm máy tính

Đây là giao diện thêm máy tính mới. Phải nhập thông tin số máy và kiểu máy cho máy mới.



4.3.2. Giao diện xóa máy tính

Tương tự giao diện thêm máy tính.

MENU/QUẢN LÝ MÁY TÍNH/XÓA MÁY TÍNH

Mã máy	Số máy	Kiểu máy	Tình trạng	Thời gian bắt đầu
STU427	1	Thường	Trống	
STU423	2	Thường	Trống	
STU929	3	Thường	Trống	
STU024	4	Cao cấp	Trống	
STU904	5	Cao cấp	Trống	
STU174	6	Cao cấp	Trống	
STU094	7	Cao cấp	Trống	
STU882	8	Cao cấp	Trống	
STU455	9	Cao cấp	Trống	
STU171	10	Cao cấp	Đầy	14:59:57 - 23/11/2022
STU152	11	Cao cấp	Đầy	04:30:22 - 17/12/2022
STU761	12	Cao cấp	Trống	
STU221	13	Cao cấp	Trống	
STU685	14	Cao cấp	Trống	
STU431	15	Cao cấp	Trống	
STU795	16	Cao cấp	Trống	
STU892	17	Cao cấp	Đầy	21:32:04 - 16/12/2022
STU233	18	Cao cấp	Đầy	21:30:13 - 16/12/2022
STU411	19	Cao cấp	Trống	
STU387	20	Cao cấp	Đầy	14:59:22 - 23/11/2022

(?) Nhập mã máy tính cần xóa: |

4.4. Giao diện quản lý nhân viên

Có 3 tùy chọn là thêm nhân viên, xóa nhân viên hoặc xem danh sách nhân viên.

MENU/QUẢN LÝ NHÂN VIÊN

→1. Thêm nhân viên
2. Xóa nhân viên
3. Xem danh sách
0. Trở về

(Sử dụng các phím ↑ ↓ để di chuyển và ENTER để chọn)|

4.4.1 Giao diện thêm nhân viên

Đây là giao diện thêm nhân viên. Phải nhập thông tin họ tên, tài khoản, mật khẩu và số điện thoại cho nhân viên mới.

```

MENU/QUẢN LÝ NHÂN VIÊN/THÊM NHÂN VIÊN
*-----*
| Mã | Họ tên | Tài khoản | Mật khẩu | Số điện thoại |
*-----*
| 137 | Lam Vo | voxuanlam | 123 | 0963538137 |
| 276 | Thang Tran | tranvanquochang | 1234 | 0866085276 |
| 155 | Test Test | test | test | 1231231231 |
| 377 | Sang Tran | trannguyenthanhsang | 3123 | 0778547377 |
| 622 | Sang Dang | dangxuansang | 1444 | 0912697622 |
| 875 | Son Do | dotranhaison | 123 | 0338874875 |
*-----*

(?) Nhập tài khoản: |

```

4.4.2. Giao diện xóa nhân viên

Người quản lý bắt buộc nhập mã nhân viên cần xóa để xóa nhân viên đó ra khỏi danh sách nhân viên.

MENU/QUẢN LÝ NHÂN VIÊN/XÓA NHÂN VIÊN				
Mã	Họ tên	Tài khoản	Mật khẩu	Số điện thoại
137	Lam Vo	voxuanlam	123	0963538137
276	Thang Tran	tranvanquochang	1234	0866085276
155	Test Test	test	test	1231231231
377	Sang Tran	trannguyenthanhsang	3123	0778547377
622	Sang Dang	dangxuansang	1444	0912697622
875	Son Do	dotranhaison	123	0338874875

(?) Nhập mã nhân viên cần xóa: |

4.4.3. Giao diện xem danh sách nhân viên

Chương trình sẽ hiển thị thông tin của từng nhân viên.

MENU/QUẢN LÝ NHÂN VIÊN/XEM DANH SÁCH

Mã	Họ tên	Tài khoản	Mật khẩu	Số điện thoại
137	Lam Vo	voxuanlam	123	0963538137
276	Thang Tran	tranvanquocthang	1234	0866085276
155	Test Test	test	test	1231231231
377	Sang Tran	trannguyenthanhsang	3123	0778547377
622	Sang Dang	dangxuansang	1444	0912697622
875	Son Do	dotranhaison	123	0338874875

[Nhấn ENTER để tiếp tục...]

4.5. Giao diện xem tình trạng máy

Có 3 tùy chọn là tình trạng máy trống, máy đầy hoặc tắt cả.

MENU/TÌNH TRẠNG MÁY

→1. Máy trống
2. Máy đầy
3. Tắt cả
0. Trở về

(Sử dụng các phím ↑ ↓ để di chuyển và ENTER để chọn)

4.5.1. Giao diện xem máy trống

Chương trình chỉ hiển thị thông tin và trạng thái những máy chưa được sử dụng.

MENU/TÌNH TRẠNG MÁY/MÁY TRỐNG			
Mã máy	Số máy	Kiểu máy	Tình trạng
STU427	1	Thường	Trống
STU423	2	Thường	Trống
STU929	3	Thường	Trống
STU024	4	Cao cấp	Trống
STU904	5	Cao cấp	Trống
STU174	6	Cao cấp	Trống
STU094	7	Cao cấp	Trống
STU882	8	Cao cấp	Trống
STU455	9	Cao cấp	Trống
STU761	12	Cao cấp	Trống
STU221	13	Cao cấp	Trống
STU685	14	Cao cấp	Trống
STU431	15	Cao cấp	Trống
STU795	16	Cao cấp	Trống
STU411	19	Cao cấp	Trống

[Nhấn ENTER để tiếp tục...]

4.5.2. Giao diện xem máy đầy

Chương trình chỉ hiển thị thông tin và trạng thái những máy đang được sử dụng.

MENU/TÌNH TRẠNG MÁY/MÁY ĐẦY				
Mã máy	Số máy	Kiểu máy	Tình trạng	Thời gian bắt đầu
STU171	10	Cao cấp	Đầy	14:59:57 - 23/11/2022
STU152	11	Cao cấp	Đầy	04:30:22 - 17/12/2022
STU892	17	Cao cấp	Đầy	21:32:04 - 16/12/2022
STU233	18	Cao cấp	Đầy	21:30:13 - 16/12/2022
STU387	20	Cao cấp	Đầy	14:59:22 - 23/11/2022

[Nhấn ENTER để tiếp tục...]

4.5.3. Giao diện xem tất cả máy

Chương trình sẽ hiển thị thông tin và trạng thái tất cả các máy.

MENU/TÌNH TRẠNG MÁY/TẤT CẢ				
Mã máy	Số máy	Kiểu máy	Tình trạng	Thời gian bắt đầu
STU427	1	Thường	Trống	
STU423	2	Thường	Trống	
STU929	3	Thường	Trống	
STU024	4	Cao cấp	Trống	
STU904	5	Cao cấp	Trống	
STU174	6	Cao cấp	Trống	
STU094	7	Cao cấp	Trống	
STU882	8	Cao cấp	Trống	
STU455	9	Cao cấp	Trống	
STU171	10	Cao cấp	Đầy	14:59:57 - 23/11/2022
STU152	11	Cao cấp	Đầy	04:30:22 - 17/12/2022
STU761	12	Cao cấp	Trống	
STU221	13	Cao cấp	Trống	
STU685	14	Cao cấp	Trống	
STU431	15	Cao cấp	Trống	
STU795	16	Cao cấp	Trống	
STU892	17	Cao cấp	Đầy	21:32:04 - 16/12/2022
STU233	18	Cao cấp	Đầy	21:30:13 - 16/12/2022
STU411	19	Cao cấp	Trống	
STU387	20	Cao cấp	Đầy	14:59:22 - 23/11/2022

[Nhấn ENTER để tiếp tục...]

4.6. Giao diện xem lịch sử thanh toán

Chương trình sẽ hiển thị 2 mốc thời gian bắt đầu và thời gian kết thúc, số máy, kiểu máy và số tiền thanh toán.

MENU/LỊCH SỬ THANH TOÁN				
Thời gian bắt đầu	Thời gian kết thúc	Số máy	Kiểu máy	Số tiền
01:23:43 - 19/11/2022	22:15:48 - 19/11/2022	4	Cao cấp	751250 VND
01:23:46 - 19/11/2022	22:17:58 - 19/11/2022	5	Cao cấp	752520 VND
22:04:58 - 19/11/2022	22:18:14 - 19/11/2022	10	Cao cấp	7960 VND
22:02:19 - 19/11/2022	22:19:01 - 19/11/2022	15	Cao cấp	10020 VND
22:07:33 - 19/11/2022	22:19:08 - 19/11/2022	16	Cao cấp	6950 VND
01:23:57 - 19/11/2022	13:23:27 - 23/11/2022	11	Cao cấp	3887700 VND
22:06:36 - 19/11/2022	13:26:41 - 23/11/2022	12	Cao cấp	3144050 VND
01:23:48 - 19/11/2022	13:27:12 - 23/11/2022	13	Cao cấp	3890040 VND
01:23:50 - 19/11/2022	13:29:33 - 23/11/2022	14	Cao cấp	3891430 VND
01:23:53 - 19/11/2022	13:30:00 - 23/11/2022	19	Cao cấp	3891670 VND
23:03:17 - 15/12/2022	23:03:34 - 15/12/2022	1	Thường	85 VND
23:10:20 - 15/12/2022	23:10:23 - 15/12/2022	3	Thường	15 VND
23:50:56 - 15/12/2022	23:50:59 - 15/12/2022	3	Thường	15 VND
20:36:18 - 20/12/2022	20:36:45 - 20/12/2022	1	Thường	135 VND
14:59:51 - 23/11/2022	20:38:46 - 20/12/2022	2	Thường	11765675 VND
20:40:07 - 20/12/2022	20:40:35 - 20/12/2022	19	Cao cấp	280 VND

[Nhấn ENTER để tiếp tục...]

4.7. Giao diện xem lịch sử đăng nhập

Chương trình sẽ hiển thị mốc thời gian cụ thể, tài khoản thực hiện thao tác và trạng thái đăng nhập hoặc đăng xuất.

MENU/LỊCH SỬ ĐĂNG NHẬP

Thời gian	Tài khoản	Trạng thái
00:44:57 - 20/12/2022	admin	Đăng nhập
00:45:06 - 20/12/2022	admin	Đăng xuất
00:45:10 - 20/12/2022	admin	Đăng nhập
00:47:05 - 20/12/2022	admin	Đăng nhập
00:47:24 - 20/12/2022	admin	Đăng xuất
00:48:28 - 20/12/2022	tranvanquochang	Đăng nhập
01:01:48 - 20/12/2022	admin	Đăng nhập
01:03:51 - 20/12/2022	admin	Đăng nhập
01:05:15 - 20/12/2022	admin	Đăng nhập
01:07:44 - 20/12/2022	admin	Đăng nhập
01:10:42 - 20/12/2022	admin	Đăng xuất
01:10:48 - 20/12/2022	tranvanquochang	Đăng nhập
01:13:13 - 20/12/2022	admin	Đăng nhập
01:25:54 - 20/12/2022	admin	Đăng nhập
19:20:59 - 20/12/2022	admin	Đăng nhập
19:21:23 - 20/12/2022	admin	Đăng xuất
19:21:26 - 20/12/2022	admin	Đăng nhập
20:34:01 - 20/12/2022	admin	Đăng nhập
20:34:16 - 20/12/2022	admin	Đăng xuất

[Nhấn ENTER để tiếp tục...]

4.8. Giao diện kiểm tra doanh thu

Có 2 loại là kiểm tra doanh thu 7 ngày gần nhất và doanh thu ngày tùy chỉnh.

MENU/KIỂM TRA DOANH THU

→1. Doanh thu 7 ngày gần nhất
2. Doanh thu ngày tùy chỉnh
0. Trở về

(Sử dụng các phím ↑ ↓ để di chuyển và ENTER để chọn)

4.8.1. Giao diện kiểm tra doanh thu 7 ngày gần nhất

Chương trình sẽ xuất ra doanh thu từng ngày từ ngày hiện tại lùi lại thêm 6 ngày.

MENU/KIỂM TRA DOANH THU/DOANH THU 7 NGÀY GẦN NHẤT	
Thời gian	Tổng doanh thu
21/12/2022 (Hôm nay)	0 VND
20/12/2022	11766090 VND
19/12/2022	0 VND
18/12/2022	0 VND
17/12/2022	0 VND
16/12/2022	0 VND
15/12/2022	115 VND

[Nhấn ENTER để tiếp tục...]

4.8.2. Giao diện kiểm tra doanh thu ngày tùy chỉnh

Người quản lý sẽ nhập ngày/tháng/năm cần xem doanh thu. Sau đó chương trình sẽ xuất ra doanh thu của ngày đó.

MENU/KIỂM TRA DOANH THU/DOANH THU NGÀY TÙY CHỈNH	
Thời gian	Tổng doanh thu
20/12/2022	11766090 VND

[Nhấn ENTER để tiếp tục...]

4.9. Giao diện mở máy trực tiếp

Nếu danh sách máy tính trống thì sẽ xuất thông báo danh sách máy tính trống.

Nếu danh sách máy tính đều đang được sử dụng thì sẽ xuất thông báo hết máy.

Người quản lý hoặc nhân viên sẽ nhập số máy cần mở để mở máy.

MENU/MỞ MÁY TRỰC TIẾP

Mã máy	Số máy	Kiểu máy	Tình trạng
STU427	1	Thường	Trống
STU423	2	Thường	Trống
STU929	3	Thường	Trống
STU024	4	Cao cấp	Trống
STU904	5	Cao cấp	Trống
STU174	6	Cao cấp	Trống
STU094	7	Cao cấp	Trống
STU882	8	Cao cấp	Trống
STU455	9	Cao cấp	Trống
STU761	12	Cao cấp	Trống
STU221	13	Cao cấp	Trống
STU685	14	Cao cấp	Trống
STU431	15	Cao cấp	Trống
STU795	16	Cao cấp	Trống
STU411	19	Cao cấp	Trống

(?) Nhập số máy: |

4.10. Giao diện thanh toán

Người quản lý hoặc nhân viên nhập số máy cần thanh toán, sau đó sẽ chương trình sẽ hiển thị thông tin hóa đơn cần thanh toán.

MENU/THANH TOÁN				
Mã máy	Số máy	Kiểu máy	Tình trạng	Thời gian bắt đầu
STU427	1	Thường	Đầy	04:00:40 - 21/12/2022
STU171	10	Cao cấp	Đầy	14:59:57 - 23/11/2022
STU152	11	Cao cấp	Đầy	04:30:22 - 17/12/2022
STU892	17	Cao cấp	Đầy	21:32:04 - 16/12/2022
STU233	18	Cao cấp	Đầy	21:30:13 - 16/12/2022
STU387	20	Cao cấp	Đầy	14:59:22 - 23/11/2022

(?) Nhập số máy: |

4.11. Giao diện thiết lập giá tiền

Người quản lý phải bắt buộc nhập giá tiền giá máy mới (thường và cao cấp).

MENU/THIẾT LẬP GIÁ TIỀN	
GIÁ TIỀN HIỆN TẠI	
Máy thường:	18000 VND
Máy cao cấp:	36000 VND

(?) Nhập giá tiền máy thường / 1 giờ: |

4.12. Giao diện thông tin giới thiệu

```
MENU/THÔNG TIN GIỚI THIỆU
```

```
*----- ĐÀI HỌC CÔNG NGHỆ SÀI GÒN -----*
```

```
| * Đồ án:      Quản Lý Phòng Game |
```

```
| |
```

```
| * Giảng viên: Hoàng Khuê |
```

```
| |
```

```
| * Sinh viên:   1. Trần Văn Quốc Thắng - DH52007101 - D20_TH11 |
```

```
|                2. Võ Xuân Lam - DH52006929 - D20_TH11 |
```

```
|                3. Đặng Xuân Sang - DH52006904 - D20_TH11 |
```

```
|                4. Trần Nguyễn Thanh Sang - DH52007102 - D20_TH11 |
```

```
|                5. Đỗ Trần Hải Sơn - DH52007024 - D20_TH11 |
```

```
*----- 2022 -----*
```

```
[Nhấn ENTER để tiếp tục...]|
```

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận

Chương trình đã được phát triển thành công. Đảm bảo đầy đủ các tính năng cơ bản để quản lí phòng game một cách hiệu quả, thuận lợi.

Chương trình hoạt động tốt trên máy tính, laptop. Hiệu suất nhanh chóng ổn định, giao diện thân thiện, dễ sử dụng, dễ thao tác.

5.2. Hướng phát triển

Xây dựng hoàn thiện chương trình, thêm các tính năng như nạp giờ chơi,...

Tìm hiểu kết hợp nhiều ngôn ngữ khác để phát triển chương trình thành phần mềm, ứng dụng.

TÀI LIỆU THAM KHẢO

- [1]. <https://teky.edu.vn/blog/github-la-gi>
- [2]. <https://glints.com/vn/blog/lap-trinh-cpp-la-gi/#.Y6INQXZByMo>
- [3]. <https://github.com/hugorplobo/colors.hpp>