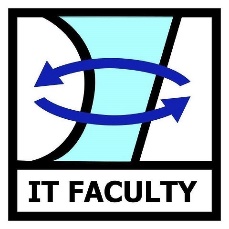
**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA CÔNG NGHỆ THÔNG TIN**

**---🙣\*🙡---**

****

**BÁO CÁO ĐỒ ÁN**

**PBL2: DỰ ÁN CƠ SỞ LẬP TRÌNH**

**HỆ THỐNG QUẢN LÝ SIÊU THỊ**

**(MINI MARKET MANAGEMENT SYSTEM)**

GIẢNG VIÊN HƯỚNG DẪN:  **TS. ĐẶNG HOÀI PHƯƠNG**

SINH VIÊN THỰC HIỆN:

**LÊ VÕ NHẬT MINH LỚP: 22T\_DT4**

**NHÓM: 22.13**

**NGUYỄN HỮU PHÚC LỚP: 22T\_DT4**

**NHÓM: 22.13**

Đà Nẵng, ngày 23 tháng 12 năm 2023

**LỜI MỞ ĐẦU**

Trong thời đại công nghệ tiên tiến, công nghệ thông tin (IT) đã trở thành một ngành nghề có vai trò quan trọng đối với sự tồn tại và phát triển của đất nước. Với việc ứng dụng rộng rãi các thành tựu của ngành công nghệ thông tin vào nhiều lĩnh vực, việc giải quyết các vấn đề bằng cấu trúc dữ liệu và giải thuật luôn được các nhà phát triển đặc biệt quan tâm. Điều này đòi hỏi phải liên tục củng cố và cải tiến công nghệ, tối ưu các thuật toán để xây dựng thêm nhiều chức năng.

Với mong muốn xây dựng một phương pháp lập trình trực quan, mô tả trung thực các hệ thống trong thế giới thực, phương pháp *“Lập trình hướng đối tượng”* đã ra đời. Lập trình hướng đối tượng là phương pháp lập trình xây dựng chương trình dựa trên đối tượng. Hơn nữa, một trong những nguyên tắc cơ bản quan trọng ảnh hưởng trực tiếp đến các thuật toán là *"Cấu trúc dữ liệu"*.

Nhằm nghiên cứu, xây dựng cấu trúc dữ liệu và giải thuật một cách hợp lý, đồng thời đào sâu, mở rộng kiến ​​thức về lập trình hướng đối tượng, nhóm chúng em đã chọn đề tài để nghiên cứu mang tên *“Hệ thống quản lý siêu thị”.*

Đặc biệt, chúng em xin gửi lời cảm ơn sâu sắc đến thầy Đặng Hoài Phương. Người đã giúp đỡ, hướng dẫn, giải đáp thắc mắc và chỉ ra những sai sót trong quá trình làm đồ án để chúng em có thể hoàn thành đồ án này.

Đà Nẵng, ngày 23 tháng 12 năm 2022

**Nhóm sinh viên thực hiện đề tài**

**1. Trần Viết Sơn**

**2. Trần Văn Đức Sơn**

**3. Nguyễn Thanh Minh**

**MỤC LỤC**

[**MỤC LỤC** 3](#_Toc122674579)

[1. GIỚI THIỆU ĐỀ TÀI 5](#_Toc122674580)

[1.1 Đặt vấn đề 5](#_Toc122674581)

[1.2 Giới thiệu đề tài: 5](#_Toc122674582)

[2. PHÂN TÍCH CHỨC NĂNG HỆ THỐNG 6](#_Toc122674583)

[2.1. Hiển thị, cập nhật giá sân: 7](#_Toc122674584)

[2.2 Quản lí danh sách khách hàng 8](#_Toc122674585)

[2.3 Quản lí danh sách đặt sân 9](#_Toc122674586)

[3. THIẾT KẾ CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN 10](#_Toc122674587)

[3.1. Phát biểu bài toán 10](#_Toc122674588)

[3.2. Phân tích và ứng dụng cấu trúc dữ liệu và thuật toán trong hệ thống 10](#_Toc122674589)

[a) Phân tích cấu trúc dữ liệu và thuật toán: 10](#_Toc122674590)

[b) Ứng dụng cấu trúc dữ liệu và thuật toán trong hệ thống 12](#_Toc122674591)

[4. PHÂN TÍCH HƯỚNG ĐỐI TƯỢNG VÀ TRIỂN KHAI HỆ THỐNG 18](#_Toc122674592)

[4.1. Cấu trúc hệ thống hướng đối tượng (class, object, relation) 18](#_Toc122674593)

[a) Lớp (class) 18](#_Toc122674594)

[b) Đối tượng (object) 24](#_Toc122674595)

[c) Quan hệ giữa các lớp (relation) 25](#_Toc122674596)

[4.2. Tổ chức chương trình 26](#_Toc122674597)

[4.2.1. Giao diện chính của chương trình 26](#_Toc122674598)

[4.2.2. Kết quả thực thi của chương trình 28](#_Toc122674599)

[4.2.3. Nhận xét 34](#_Toc122674600)

[5.  KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 34](#_Toc122674601)

[5.1. Kết luận 34](#_Toc122674602)

[5.2. Hướng phát triển 34](#_Toc122674603)

**DANH MỤC HÌNH VẼ**

Hình 1.1. Sơ đồ khối chức năng chương trình..................................................................6  
Hình 1.2. Mô tả chức năng quản lí danh sách sân……………………..……….............7  
Hình 1.3. Mô tả chức năng quản lí danh sách khách hàng..................................................8  
Hình 1.4.Mô tả chức năng quản lí danh sách đặt sân ………...............................................9  
Hình 2.1. Biểu diễn danh sách liên kết kép……….…………………….……..............11  
Hình 2.2. Mô tả thuật toán tìm kiếm tuần tự..............……........................................12  
Hình 3.1. Mô tả lớp sân……………………...……………………….…......................19  
Hình 3.2. Mô tả các lớp khách hàng…………………………………………………..21  
Hình 3.3. Mô tả các lớp giá sân theo giờ………………………………………….….22  
Hình 3.4. Mô tả lịch đặt sân của mỗi sân trong ngày cụ thể ……........................................23  
Hình 3.5. Mô tả lớp đặt sân………………………………………………………………24  
Hình 3.6. Quan hệ giữa các class…………………...........................................................25  
Hình 3.7. Quan hệ class quản lý đặt sân…………...........................................................26  
Hình 3.8. Giao diện bắt đầu…………….......................................................................26  
Hình 3.9. Giao diện Menu lựa chọn……………….......................................................27  
Hình 3.10. Menu lựa chọn chức năng hiển thị……………………….……...................27  
Hình 3.11. File dữ liệu bản ghi đặt sân(ban\_ghi\_dat\_san.txt)……………………......28  
Hình 3.12. File dữ liệu danh sách khách hàng (ds\_khach\_hang.txt)………………….28  
Hình 3.13. File dữ liệu giá sân 5 theo giờ (giasan5.txt).……..……………………29  
Hình 3.14. Menu lựa chọn hiển thị………………………………………………….........30

Hình 3.15. Tìm kiếm mã sân trống theo giờ.…………………………………………....30  
Hình 3.16. Giao diện thực hiện chức năng hiển thị giá sân theo giờ……………..............31  
Hình 3.17. Giao diện thực hiện chức năng hiển thị thông tin đặt sân.……………….....31  
Hình 3.18. Giao diện thực hiện chức năng đặt sân…………………..………………….32  
Hình 3.19. Giao diện thực hiện chức năng xóa bằng diện thoại …………….…………...32

Hình 3.20. Giao diện thực hiện chức năng hủy sân………………….................................33  
Hình 3.21. Giao diện thực hiện chức năng cập nhật giá sân.……………………..…....33

# 

# GIỚI THIỆU ĐỀ TÀI

## 1.1 Đặt vấn đề

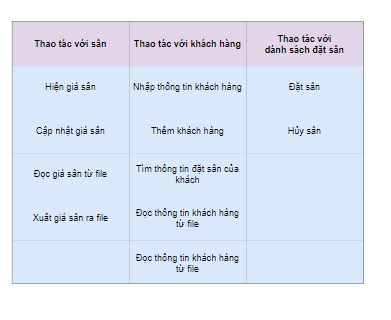
Bài toán quản lý luôn là vấn đề ở tất cả các lĩnh vực, từ quản lý nhân sự, quản lý kinh doanh, quản lý tài chính, … luôn được quan tâm. Một trong số đó, vấn đề về giải pháp quản lý mô hình cho thuê sân bóng đá cũng được đặt ra khi khách hàng của các sân thường gặp phải tình trạng thiếu sân khi không đặt trước. Các chủ sân thường chủ quan do số lượng sân thấp, tin rằng việc ghi các bản viết tay cũng đủ để lưu kiểm soát thông tin của tất cả sân bóng. Tuy nhiên, thực chất sẽ rất khó quản lý thời gian cho thuê của các sân khi số lượng đặt trước sân quá lớn, việc lưu trữ các mẫu đặt sân cũng gặp nhiều khó khăn … Để đạt được mục đích quản lý hiệu quả, việc áp dụng công nghệ vào quản lý cũng là một giải pháp hiệu quả, chính xác và tiết kiệm thời gian cho chủ sở hữu và khách hàng.

## Giới thiệu đề tài:

Việc tin học hóa quản lý sân bóng đá mang rất nhiều lợi ích so với việc ghi chép thủ công. Người chủ điều hành sẽ quản lí các sân bóng một cách chính xác và thuận tiện dựa trên những chức năng mà hệ thống đã được lập trình sẵn như kiểm tra tình trạng từng sân, quản lí thông tin của sân bóng, kiểm tra thông tin của người đặt sân, thay đổi thông tin của sân bóng, lựa chọn sân phù hợp còn trống để khách hàng đặt sân,... Với hệ thống này, việc quản lí sẽ vô cùng thuận lợi và dễ dàng, giúp tăng nâng suất cũng như hạn chế tối thiểu những sai sót trong công việc.

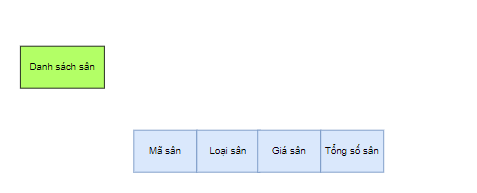
# 2. PHÂN TÍCH CHỨC NĂNG HỆ THỐNG

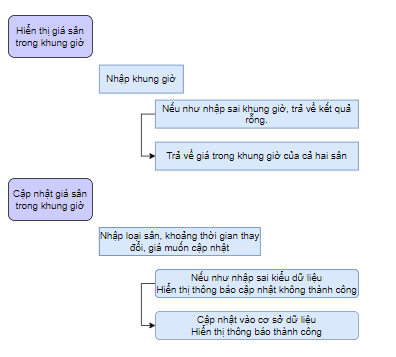
* Hệ thống được viết thành một chương trình đơn giản, người sử dụng là người quản lý sân. Chương trình được thiết kế đáp ứng tương đối đủ các nhu cầu của người sử dụng
* Hệ thống sẽ có 3 chức năng tổng quát:
* *Chức năng 1*: Thao tác đặt, hủy, sửa thông tin đặt sân
* *Chức năng 2*: Xem, thêm và tìm kiếm thông tin khách hàng, tình
* *Chức năng 3*: Hiển thị và cập nhật giá của từng loại sân



Hình 1.1: Sơ đồ khối chức năng chương trình

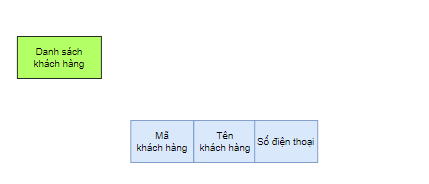
## 2.1. Hiển thị, cập nhật giá sân:

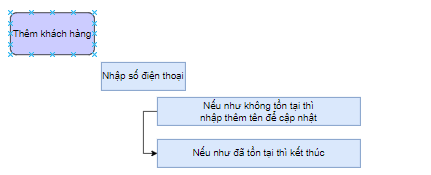


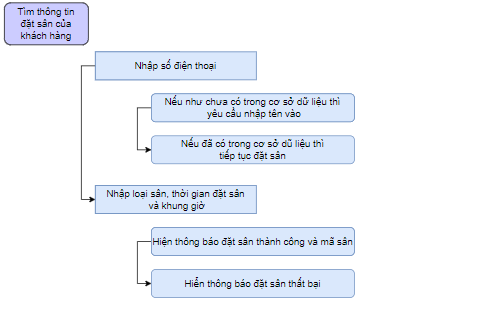


Hình 1.2: Mô tả chức năng quản lí danh sách sân

## 2.2 Quản lí danh sách khách hàng

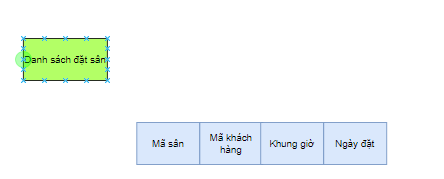




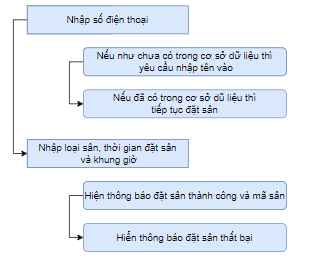


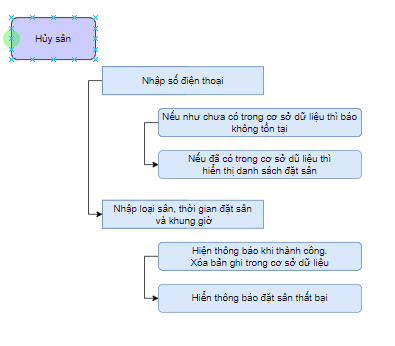
***Hình 1.3:*** *Mô tả chức năng quản lí danh sách khách hàng*

## 2.3 Quản lí danh sách đặt sân









***Hình 1.4:*** *Mô tả chức năng quản lí danh sách đặt sân*

# 3. THIẾT KẾ CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN

# 3.1. Phát biểu bài toán

Chủ doanh nghiệp sẽ quản lí nhiều sân bóng, mỗi sân bóng sẽ chứa các thông tin riêng bao gồm: thông tin sân bóng, thời gian sân được thuê, thông tin khách hàng, hóa đơn đặt sân. Để có thể kiểm soát thông tin tất cả sân bóng, phải xây dựng chương trình quản lí cả bốn thành phần(sân bóng, khách hàng, thời gian, hóa đơn).

Cấu trúc sân bóng chứa: mã sân, chú thích của sân, giá sân theo khung giờ, loại sân (sân 5 hay sân 7).

Cấu trúc thông tin khách hàng chứa: mã khách hàng, tên khách hàng, số điện thoại.

Cấu trúc thời gian chứa: thời gian chính xác ở thời điểm hiện tại bao gồm các thông số về giờ, ngày, tháng, năm.

Cấu trúc hóa đơn chứa: mã khách hàng, mã sân bóng, thời gian đặt sân.

Cấu trúc bắt đầu: chức hàm đọc và ghi dữ liệu file .txt.

## 3.2. Phân tích và ứng dụng cấu trúc dữ liệu và thuật toán trong hệ thống

### a) Phân tích cấu trúc dữ liệu và thuật toán:

* Danh sách liên kết kép

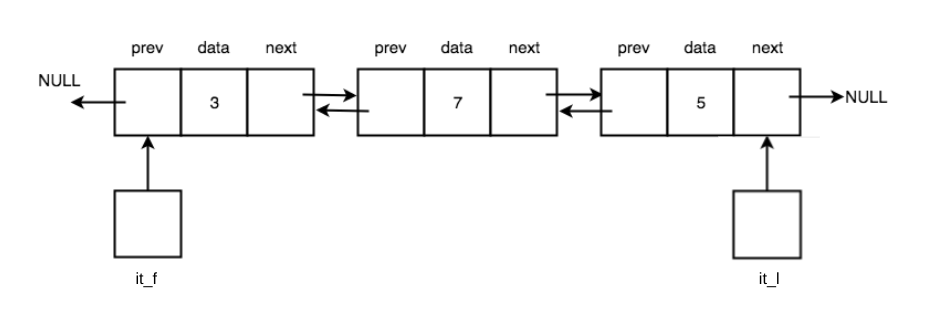
Danh sách liên kết kép là kiểu cấu trúc dữ liệu chứa một danh sách các phần tử, mà mỗi phần tử (Node) chứa 3 miền giá trị:

* Prev : con trỏ trỏ tới phần tử đứng trước nó, mô tả liên kết giữa phần tử này tới phần tử đằng trước nó
* Data : Dữ liệu của phần tử, nói cách khác là nơi lưu trữ giá trị của phần tử này trong danh sách liên kết
* Next : con trỏ trỏ tới phần tử đứng sau nó, mô tả liên kết giữa phần tử này tới phần tử đằng sau nó.

Danh sách liên kết kép có các phần tử kết nối với nhau giữa hai phần tử liền kề nhau.

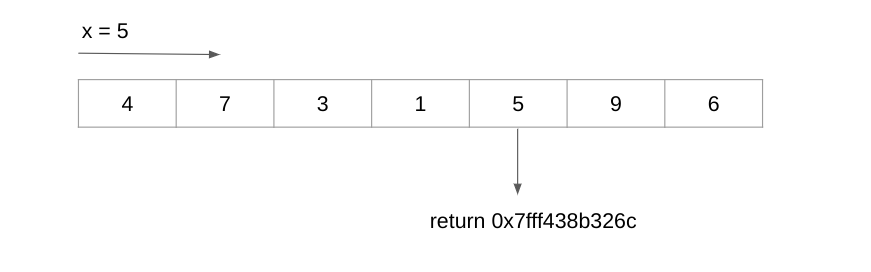
Bên cạnh đó, danh sách liên kết còn lưu thêm các trường giá trị như

* it\_f : Con trỏ trỏ tới phần tử đầu tiên của danh sách liên kết
* it\_l : Con trỏ trỏ tới phần tử cuối cùng của danh sách liên kết
* csize : Kích thước hiện tại của danh sách liên kết



***Hình 2.1:*** *Biểu diễn danh sách liên kết kép*

* Danh sách liên kết đôi ban đầu chứa 3 miền giá trị: csize, it\_f, it\_l.
* Ban đầu, it\_f = it\_l = NULL, csize = 0.
* Khi thêm vào giá trị đầu tiên, it\_f = it\_l = con trỏ tới phần tử đầu tiên, csize = 1.
* Khi thêm mới các giá trị khác, phần tử mới tạo liên kết với các phần tử liền kề nó, nếu như thêm mới tại đầu hoặc cuối danh sách, cập nhật lại giá trị của it\_f hoặc it\_l.
* **Thuật toán tìm kiếm tuần tự (Linear search)**
* Thuật toán tìm kiếm tuần tự là phương pháp tìm kiếm một phần tử đã cho trong danh sách bằng cách duyệt lần lượt từng phần của danh sách cho đến khi tìm thấy giá trị mong muốn hoặc bằng cách duyệt toàn bộ danh sách.
* Ý tưởng:
  + Với giá trị x cần tìm trong danh sách liên kết, ta bắt đầu tạo vòng lặp để duyệt từ đầu đến cuối danh sách.
  + Trong quá trình lặp, nếu gặp phải một phần tử thỏa mãn thuộc tính = x thì lập tức trả về con trỏ trỏ tới phần tử đó.
  + Nếu đã duyệt toàn bộ phần tử mà không gặp phần tử nào bằng với x thì trả về NULL.



***Hình 2.2:*** *Mô tả thuật toán tìm kiếm tuần tự*

* Độ phức tạp trong tình huống xấu nhất là O(n) khi phần tử tìm kiếm nằm cuối danh sách hoặc không có trong danh sách.
* Thời gian chạy tốt nhất: O(1) khi phần tử cần tìm nằm ngay đầu danh sách.
* Độ phức tạp không gian: O(n).

### b) Ứng dụng cấu trúc dữ liệu và thuật toán trong hệ thống

**Danh sách liên kết kép**

* Tạo một khuôn mẫu lớp **l**isttđể xây dựng cấu trúc dữ liệu cho chương trình, trong đó dùng kiểu dữ liệu class để định nghĩa một lớp đối tượng, và dùng cấu trúc dữ liệu danh sách liên kết kép để tạo và lưu trữ một danh sách lớp đối tượng theo một khuôn mẫu lớp (Class Template).
* **Khai báo khuôn mẫu lớp**

**template <class T>**

**class listt**

* **Khai báo thuộc tính private**

**private:**

**node<T>\* it\_f;**

**node<T>\* it\_l;**

**int csize;**

* **Định nghĩa cấu trúc một Node**

Các node: Mỗi node bao gồm phần dữ liệu, một con trỏ prev để trỏ đến node ngay trước nó và một con trỏ next để trỏ tới node ngay tiếp sau nó.

**template <class T>**

**struct node{**

**T data; // Miền lưu giá trị**

**node\* next;// Lưu con trỏ trỏ tới phần tử liền sau**

**node\* prev;// Lưu con trỏ trỏ tới phần tử liền trước**

**template<class U>**

**friend istream& operator >> (istream& in, node<U>& a);**

**template<class U>**

**friend ostream& operator << (ostream& out, node<U>& a);**

**};**

* **Hàm dựng** không đối số (mặc định):

**listt():it\_f(NULL),it\_l(NULL),csize(0)**

**{**

**}**

* Hàm dựng với kích thước có sẵn và giá trị ban đầu:

**listt(int csize, int val = 0):csize(csize){**

**for (int i =0; i<csize; i++){**

**push\_back(val);**

**}**

**}**

* **Hàm hủy**

**~listt(){**

**while (pop\_back()){}**

/\**Hàm pop\_back() (sẽ được giải thích ở bên dưới) là hàm xóa phần tử và giải phóng vùng nhớ cho phần tử cuối cùng, trả về 1 nếu xóa thành công. Trả về 0 nếu kích thước =0, vì vậy nên dòng lệnh trên hoạt động được \*/*

**}**

* Thêm một phần tử vào cuối danh sách:

**void push\_back(T a){**

**if (empty()){**

*/\** *nếu danh sách liên kết đang trống,*

*tạo mới một phần tử và gán itf = itl = node\*/*

**node<T>\* x = new node<T>;**

**x->data = a;**

**it\_l = x;**

**it\_f = x;**

**x->next = NULL;**

**x->prev = NULL;**

**csize=1;**

**return;**

**}**

**node<T>\* x = new node<T>;**

**x->data = a;**

**x->prev = it\_l;**

**x->next = NULL;**

**it\_l->next = x;**

**it\_l = x;**

**csize++;**

**}**

* Thêm phần tử vào đầu danh sách:

**void push\_front(T a){**

**if (empty()){**

/\* *Tương tự như push\_back, nhưng thêm vào đầu danh sách\*/*

**node<T>\* x = new node<T>;**

**x->data = a;**

**it\_l = x;**

**it\_f = x;**

**x->next = NULL;**

**x->prev = NULL;**

**return;**

**}**

**node<T>\* x = new node<T>;**

**it\_f->prev = x;**

**x->data = a;**

**x->next = it\_f;**

**x->prev = NULL;**

**it\_f = x;**

**csize++;**

**}**

* Xóa phần tử ở cuối danh sách:

**bool pop\_back(){**

**if(empty()) return 0;**

**auto it = it\_l;**

**if (it\_l == it\_f) {**

// *nếu như danh sách chỉ còn 1 phần tử,*

***it\_f và it\_l phải mang giá trị NULL***

**it\_f = NULL;**

**it\_l = NULL;**

**delete it;**

**return 1;**

**}**

**else {**

**it\_l = it\_l->prev;**

**it\_l->next=NULL;**

**}**

**csize--;**

**delete it;**

**return 1;**

**}**

* Giá trị cuối cùng của danh sách

**T& back(){**

**return it\_l->data;**

**}**

* Con trỏ trỏ vào phần tử đầu danh sách:

**node<T>\* begin() const {**

**return it\_f;**

**}**

* Truy xuất phần tử thứ i trong danh sách:

// *Sử dụng đa năng hóa toán tử*

**T& operator [] (int a){// *trả về tham chiếu để tiện thay đổi giá trị***

**if (a<0 || a>=csize){**

***/\*nếu phần tử thứ i ra khỏi phạm vi truy xuất,***

***lập tức kết thúc ctrinh\*/***

**exit(0);**

**}**

**auto it = begin();**

**for (it= begin(); it!=end(); it = it->next){**

**if (a) a--;**

**else break;**

**}**

**return it->data;**

**}**

* Trả về kích thước danh sách:

**int size(){**

**return csize;**

**}**

* Tìm kiếm phần tử:

**node<T>\* find(T a){**

*/\*Trả về địa chỉ của node mang giá trị a\*/*

**for (auto it = begin(); it!=end(); it= it->next){**

**if (a==it->data) return it;**

**}**

**return NULL;**

*/\*Nếu không tìm thấy, trả về NULL\*/*

**}**

* **Xoá 1 node** mang giá trị a:

// *Hàm sẽ xóa phần tử đầu tiên trong danh sách liên kết có giá trị a và xóa nó*

**bool erase(const T& a){**

**for (auto it = begin(); it!=end(); it=it->next){**

**if (a==it->data){ // *nếu tìm thấy giá trị cần xóa (it)***

**if (it == it\_f) pop\_front();**

*// nếu giá trị cần xóa nằm ở đầu danh sách, gọi hàm pop\_front()*

**else if (it == it\_l) pop\_back();**

*// nếu giá trị cần xóa nằm ở cuối danh sách, gọi hàm pop\_back()*

**else {**

**csize--;**

**auto tmp = it->next;**

**it->prev->next = tmp;**

**tmp->prev = it->prev;**

// *3 câu lệnh trên mục đích liên kết*

*hai node bên trái và phải của it lại với nhau*

**delete it;**

**}**

**return 1;**

**}**

**}**

**return 0;**

**}**

* Tìm kiếm sân theo mã sân:
* Sử dụng thuật toán tìm kiếm tuần tự

**san\* search\_san(int id){**

**for (auto it= ds\_san.begin(); it!=NULL; it=it->next){**

**if (it->data->id()==id){**

**return (it->data);**

**}**

**}**

**return NULL;**

**}**

* Tìm kiếm khách hàng theo mã khách:
* Sử dụng thuật toán tìm kiếm tuần tự

**khach\_hang\* search\_kh(int id){**

**for (auto it= ds\_khachhang.begin(); it!=NULL; it=it->next){**

**if (it->data.id()==id){**

**return &(it->data);**

**}**

**}**

**return NULL; //trả về NULL nếu không tìm thấy**

**}**

* Tìm kiếm khách hàng theo số điện thoại:
* Sử dụng thuật toán tìm kiếm tuần tự

**khach\_hang\* search\_kh(string& sdt){**

**for (auto it= ds\_khachhang.begin(); it!=NULL; it=it->next){**

**if (it->data==sdt) {**

**return &(it->data);**

**}**

**}**

**return NULL;**

**}**

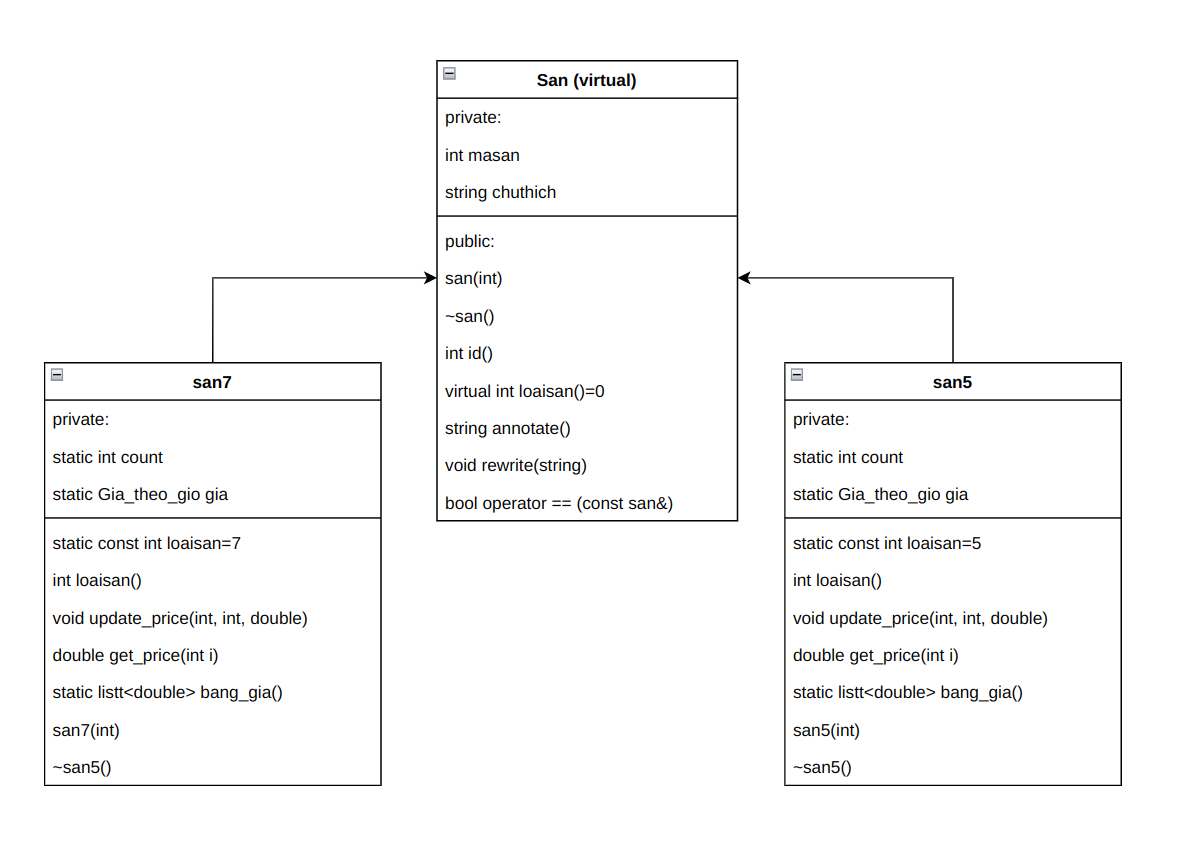
# 4. PHÂN TÍCH HƯỚNG ĐỐI TƯỢNG VÀ TRIỂN KHAI HỆ THỐNG

## 4.1. Cấu trúc hệ thống hướng đối tượng (class, object, relation)

Xây dựng mã nguồn chương trình theo các lớp và đối tượng đáp ứng bốn tính chất (đóng gói, kế thừa, đa hình, trừu tượng) của *“Lập trình hướng đối tượng”* (tiếng Anh: Object-oriented programming, viết tắt: OOP).

### a) Lớp (class)

* Các lớp dữ liệu chứa các thuộc tính private đặc trưng cho từng đối tượng đảm bảo tính chất đóng gói của ***“Lập trình hướng đối tượng”***.
* Các đối tượng được khởi tạo mặc định, bên cạnh đó còn được khởi tạo với dữ liệu nhập từ file,không những thế còn được tạo và truy cập thông qua các hàm Get – Set. Đáp ứng tính đa hình của ***“Lập trình hướng đối tượng”***, đa hình với nạp chồng phương thức.
* Từng lớp chứa các phương thức riêng phù hợp cho việc truy xuất dữ liệu của từng đối tượng.



Hình 3.1: Mô tả lớp sân

* Lớp sân 5, sân 7:
* Đối với mỗi sân, cần có thông tin để truy xuất như mã sân, ghi chú sân, loại sân (category), giá tiền theo giờ của sân…
* Các điểm khác nhau của hai sân là về loại sân và giá tiền theo giờ.
* Tạo lớp trừu tượng chứa các thuộc tính chung và các hàm thao tác trên các thuộc tính chung đó.
* Lớp san: lớp trừu tượng:
* Quản lý các thuộc tính mã sân, ghi chú sân:

protected:

* int masan;
* int chuthich;
* Các phương thức với quyền truy cập public:
* san()
* virtual loaisan()=0;
* string annotate();
* void rewrite(string)
* bool operator == (const san&)
* ~san()
* Lớp dẫn xuất từ lớp san: san5, san7
  + Lớp san5:
* Quản lý các thuộc tính loại sân, giá sân theo giờ, số lượng sân theo loại:

private:

* static int cnt
* static Gia\_theo\_gio gia

public:

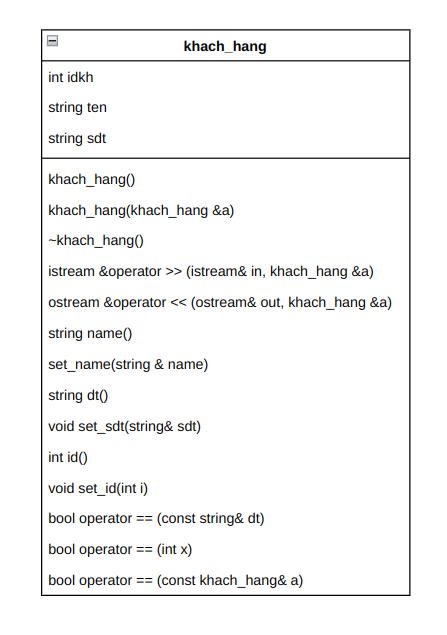
* static const int loai\_san=5;
* Các thuộc tính giá theo giờ và biến đếm là loại thuộc tính static, vì các sân cùng loại có cùng một mệnh giá.
* Các phương thức với quyền truy cập public:
* int loaisan()
* static void update(int,int,double dong)
* static double get\_price()
* static listt<double> bang\_gia()
* san5(int)
* ~san5()
  + Lớp san7:
* Quản lý các thuộc tính loại sân, giá sân theo giờ, số lượng sân theo loại:

private:

* static int cnt
* static Gia\_theo\_gio gia

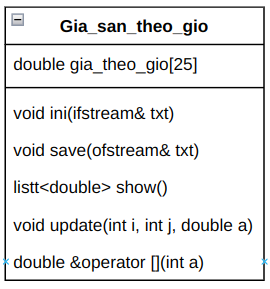
public:

* static const int loai\_san=7;
* Các thuộc tính giá theo giờ và biến đếm là loại thuộc tính static, vì các sân cùng loại có cùng một mệnh giá.
* Các phương thức với quyền truy cập public:
* int loaisan()
* static void update(int,int,double dong)
* static double get\_price()
* static listt<double> bang\_gia()
* san7(int)
* ~san7()
* Lớp khach\_hang
* Lớp chứa thông tin của một đối tượng khách hàng và các thao tác trên thông tin khách hàng đó
* Các thuộc tính:
* int idkh (chứa mã khách hàng, là khóa)
* string ten (chứa tên của khách hàng tương ứng)
* string sdt (chứa số điện thoại của khách hàng, là khóa)
* Các phương thức với quyền truy cập public:

****

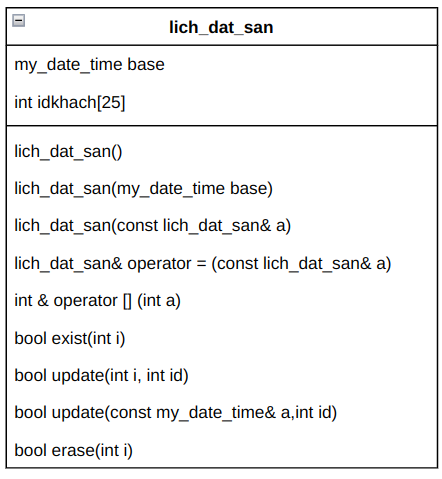
Hình 3.2: Mô tả lớp khách hàng

* Lớp hoadon
* Lớp lưu một bản ghi về đặt sân.
* int idkhach (mã khách hàng, bắt buộc phải tồn tại trong danh sách khách hàng)
* int idsan (mã sân, bắt buộc phải tồn tại trong danh sách sân)
* my\_date\_time h (khung giờ đặt sân bắt đầu từ h.hour, kéo dài 1h)
* Khóa là idsan và h.
* Các phương thức:
* hoadon() (khởi tạo mặc định)
* hoadon(int,int,my\_date\_time) (khởi tạo với các tham số idkhach, idsan, thời gian)
* bool operator == (đa năng hóa toán tử ==, kiểm tra hai bản ghi có bằng nhau hay không)
* int loai\_san() (trả về loại sân mà khách hàng đã đặt trong bản ghi này)
* Lớp Gia\_san\_theo\_gio:
* Class được tạo ra nhằm xử lý các thông tin về giá của sân theo hai loại sân là sân 5 và sân 7. Chỉ có 2 đối tượng Gia\_san\_theo\_gio được tạo ra trong đề tài này, tuy nhiên việc tạo ra class này là nhằm giảm chi phí để cập nhật hay lấy giá sân.
* Nếu mỗi sân có một thuộc tính giá sân, thì việc cập nhật giá sân sẽ phải thực hiện trên tất cả các sân. Tuy nhiên việc này lại không hiệu quả về tài nguyên thời gian và bộ nhớ. Mặt khác, các sân cùng loại sẽ có cùng giá tiền. Vì vậy giải pháp cho đề tài này là mỗi loại sân có một biến static Gia\_san\_theo\_gio để lưu thông tin và cập nhật thông tin giá sân thì vấn đề nêu trên sẽ được giải quyết

****

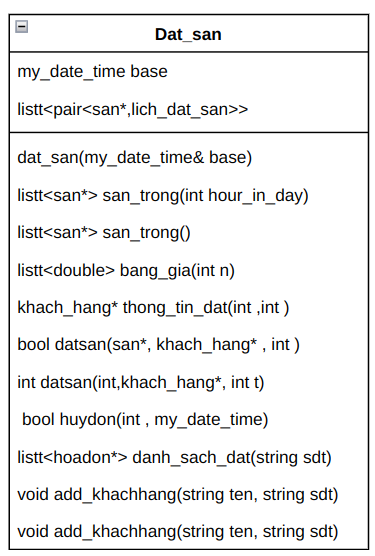
Hình 3.3: Mô tả lớp giá sân theo giờ

* Lớp lich\_dat\_san
* Đối với thao tác tìm sân trống trong một khung giờ, sẽ rất khó khăn nếu như duyệt qua toàn bộ hoadon và so sánh với từng sân trong từng khung giờ trong ngày. Vì vậy cần có giải pháp thuận tiện hơn, để tối ưu về thời gian thực thi cho truy vấn tìm sân trống. Trong thực tế, câu hỏi “Còn sân trống nào vào giờ h hay không” rất thường xuyên được đặt ra, vì vậy cần có giải pháp hữu hiệu cho những truy vấn kiểu này.
* Class có quan hệ 1-1 với một sân, được tạo ra với hai thuộc tính là my\_date\_time base và int idkhach[25]. Base là thuộc tính lưu ngày mà lich\_dat\_san đang mô tả, idkhach[i] là mã của vị khách hàng đã đặt sân trong giờ thứ i trong ngày base.
* Class lich\_dat\_san chỉ được sử dụng ở class dat\_san trong đề tài này.

****

Hình 3.4: Mô tả lịch đặt sân của mỗi sân trong ngày cụ thể

* Lớp dat\_san
* Dựa theo nhu cầu sử dụng và để dễ dàng quản lý dữ liệu trong đề tài, class dat\_san được tạo ra nhằm giải quyết hầu hết các vấn đề mà giao diện yêu cầu. Nói cách khác, dat\_san là lớp để quản lý các thông tin về đặt sân, hủy sân, thông tin giá sân, thông tin khách hàng… trong một ngày cụ thể.
* Các thuộc tính của Class Dat\_san bao gồm:
* my\_date\_time base (lưu ngày mà lớp dat\_san đang thao tác)
* listt <pair<san\*,lich\_dat\_san>> (mỗi phần tử lưu quan hệ 1-1 giữa lich\_da\_san và sân)



Hình 3.5: mô tả lớp đặt sân

* Dữ liệu và các hàm ngoài main
* listt<san\*> ds\_san
* listt<hoadon> ds\_hoadon;
* listt<khach\_hang> ds\_khachhang;

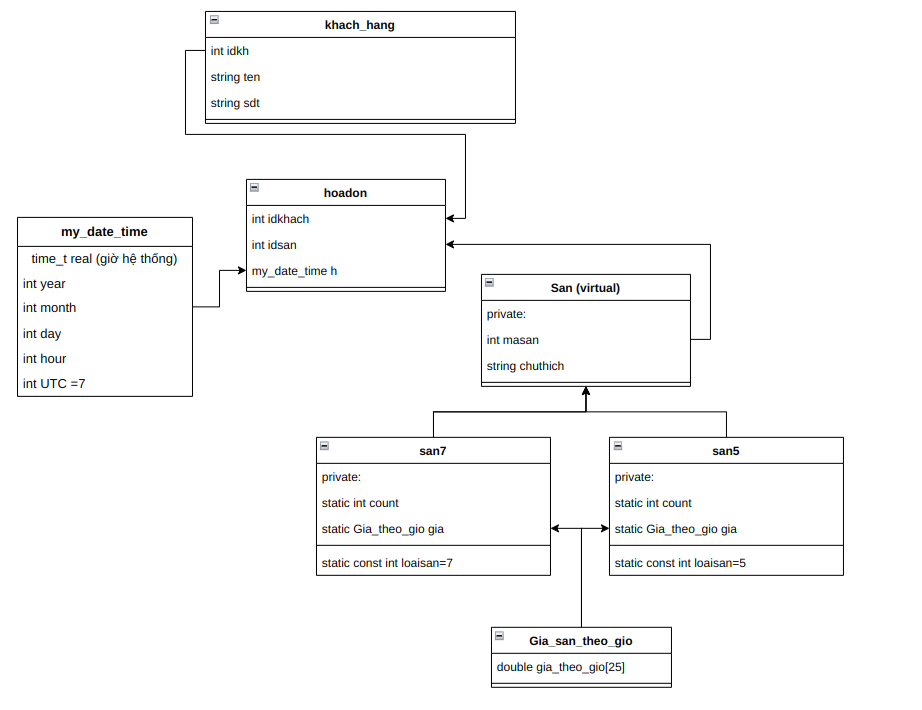
- Các phương thức khác:

* khachhang\* search\_kh(int) (tìm kiếm khách hàng theo id khách)
* khachhang\* search\_kh(string) (tìm kiếm khách hàng theo số điện thoại)
* san\* search\_san(int) (tìm kiếm sân theo mã sân)
* void init() (hàm khởi tạo chương trình, bao gồm load thông tin từ file txt)
* void save() (hàm lưu chương trình, lưu những thay đổi ra file txt)

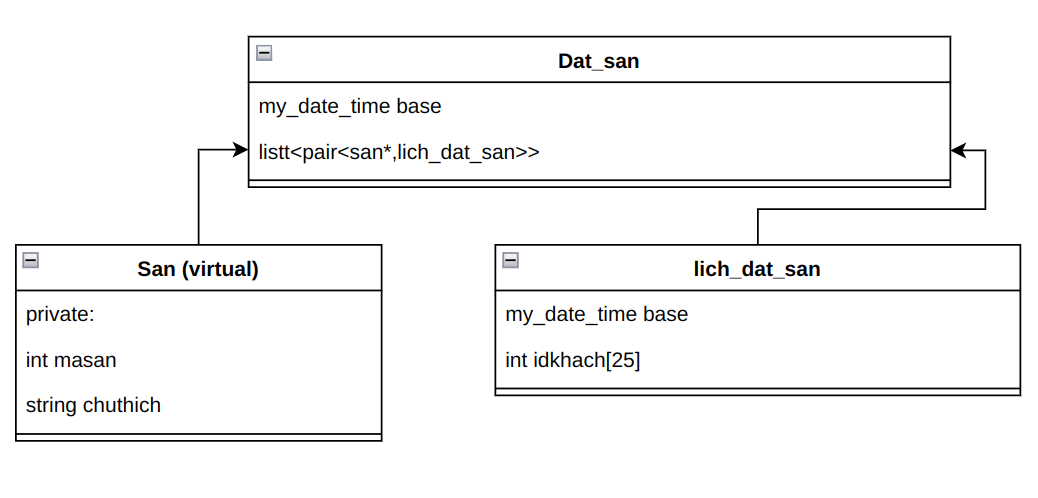
### b) Đối tượng (object)

* Các lớp dữ liệu tập hợp ba đối tượng: sân, khách hàng, hóa đơn
* Đối tượng sân chứa các thuộc tính bao gồm mã sân, chú thích của sân, giá sân theo khung giờ, loại sân (sân 5 hay sân 7).
* Đối tượng khách hàng chứa các thuộc tính bao gồm mã khách hàng, tên khách hàng, số điện thoại.
* Đối tượng hóa đơn chứa các thuộc tính bao gồm mã khách hàng, mã sân bóng, thời gian đặt sân.
* Các đối tượng được trên được quản lí bởi lớp đối tượng đặt sân, lớp đặt sân sẽ chứa các phương thức để tương tác và truy cập vào các đối tượng trên.

### c) Quan hệ giữa các lớp (relation)



Hình 3.6: Quan hệ giữa các class

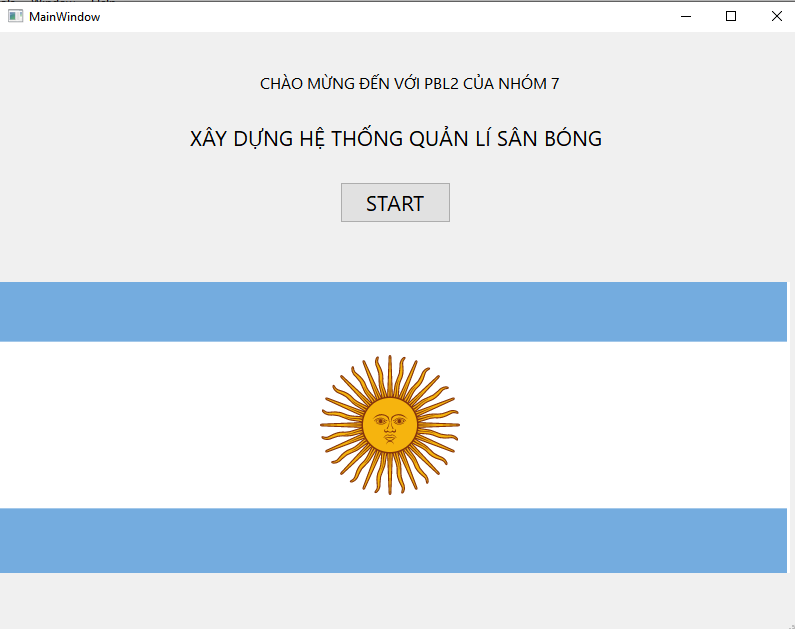
****

Hình 3.7: Quan hệ class quản lý đặt sân

## 4.2. Tổ chức chương trình

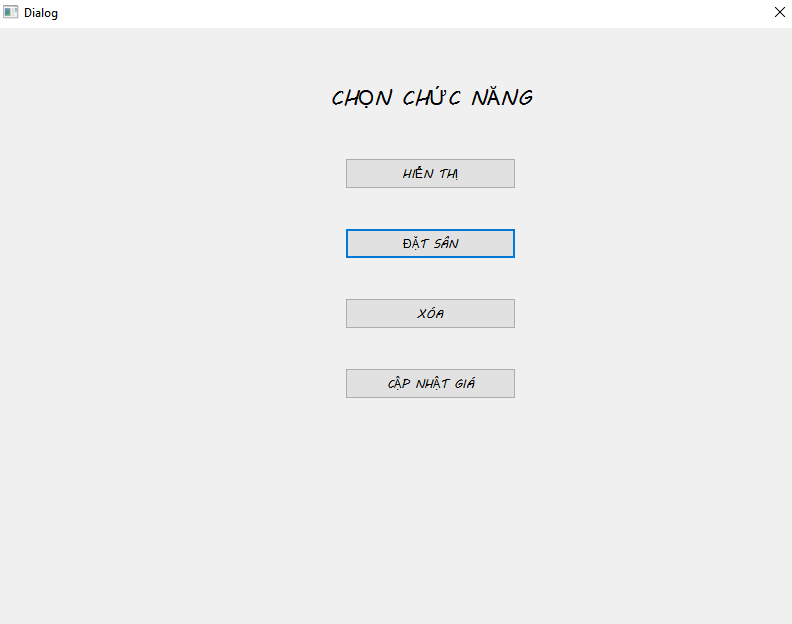
### 4.2.1. Giao diện chính của chương trình

1. Màn hình bắt đầu

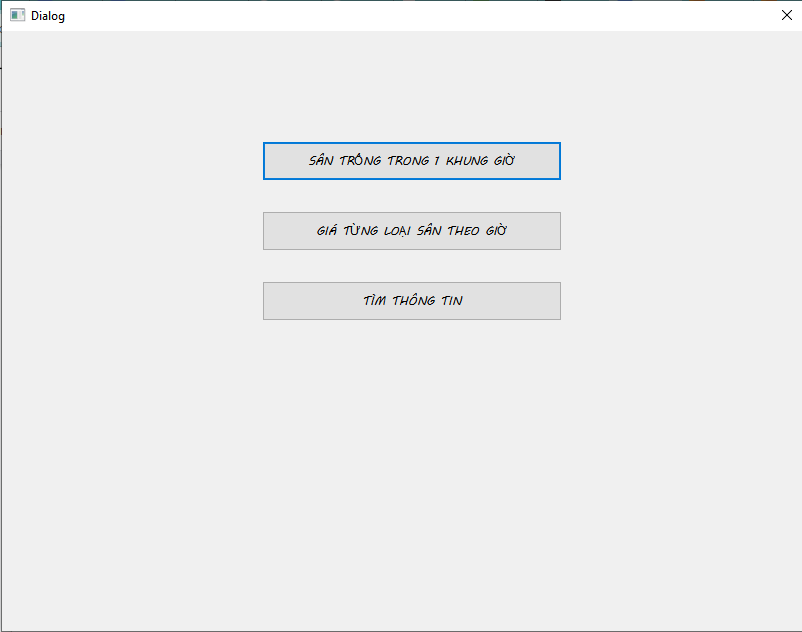


Hình 3.8: *Giao diện bắt đầu*

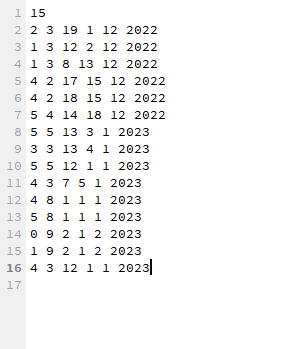
1. *Menu thao tác*

**

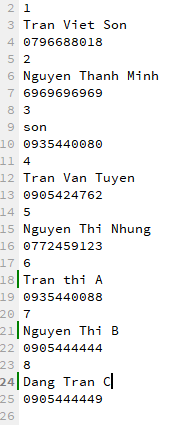
Hình 3.9: *Giao diện Menu lựa chọn*

** Hình 3.10: Menu lựa chọn chức năng hiển thị

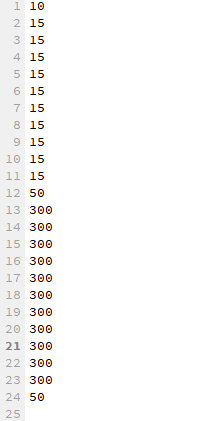
### 4.2.2. Kết quả thực thi của chương trình



Hình 3.11: *File dữ liệu bản ghi đặt sân(ban\_ghi\_dat\_san.txt)*



Hình 3.12: *File dữ liệu khách hàng (ds\_khach\_hang.txt)*

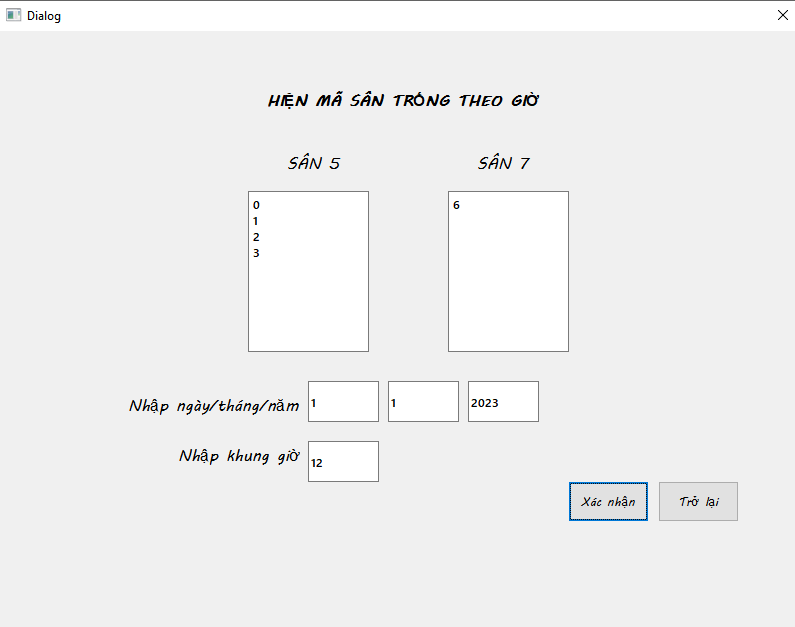
**

Hình 3.13: *File dữ liệu giá sân 5 theo giờ (giasan5.txt)*

1. **Thực hiện chức năng** hiển thị dữ liệu



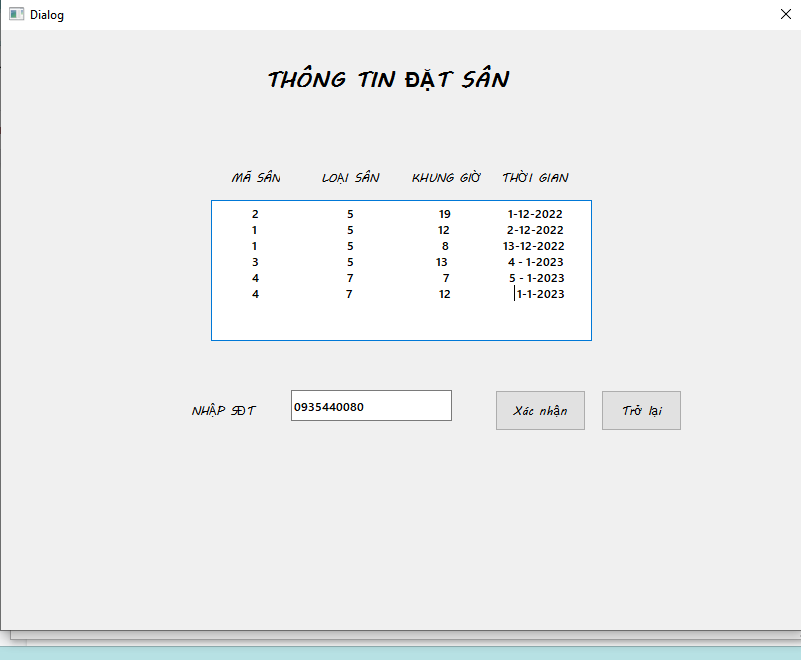
Hình 3.14: *Menu lựa chọn hiển thị*



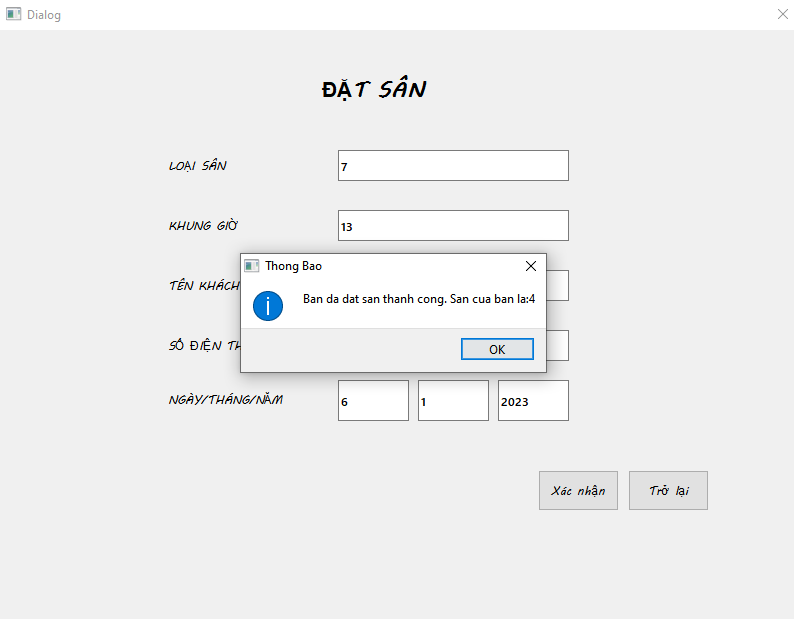
Hình 3.15: *Tìm kiếm mã sân trống theo giờ*



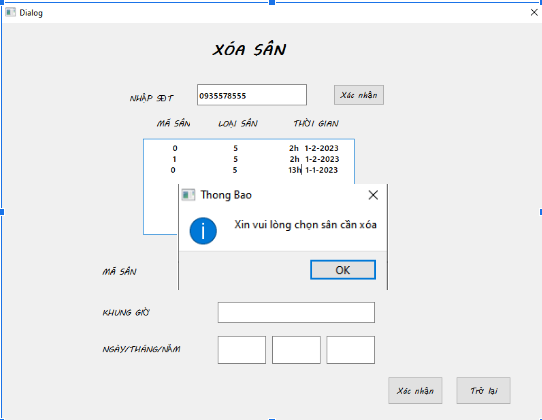
Hình 3.16: *Giao diện thực hiện chức năng hiển thị giá sân theo giờ*



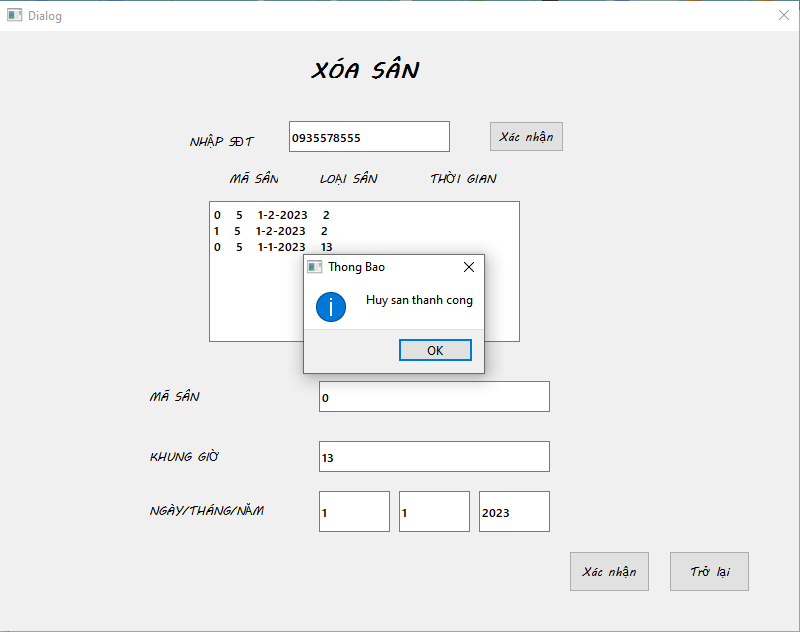
Hình 3.17: *Giao diện thực hiện chức năng hiển thị thông tin đặt sân*

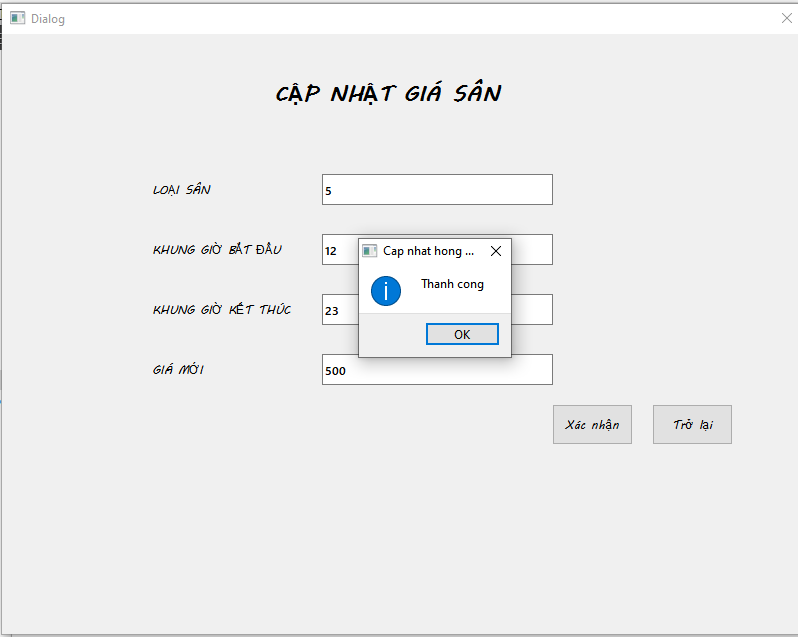


Hình 3.18: *Giao diện thực hiện chức năng đặt sân*



Hình 3.19: *Giao diện thực hiện chức năng xóa sân bằng số diện thoại*

 Hình 3.20: *Giao diện thực hiện chức năng hủy sân*



Hình 3.21: *Giao diện thực hiện chức năng cập nhật giá sân*

### 4.2.3. Nhận xét

* Chương trình cho kết quả đúng yêu cầu.
* Đọc file input bằng cách nhập tên file từ bàn phím.
* Giao diện chính của chương trình chưa đặc biệt, đơn giản.

# 5.  KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 5.1. Kết luận

Sau một khoảng thời gian dài phát triển và thử nghiệm thực tế, chương trình đã đáp ứng hầu hết mọi nhu cầu của đối tượng. Trong quá trình nghiên cứu và thực hiện đồ án, chúng em đã tìm hiểu và nắm vững cấu trúc dữ liệu danh sách liên kết đôi, struct, lập trình hướng đối tượng và các kiến ​​thức liên quan khác... Cuối cùng, chúng em đã hoàn thành chương trình: “Hệ thống quản lý sân bóng". Tuy đã chương trình đã hoàn thiện nhưng vẫn còn nhiều giải pháp tối ưu hơn về mặt thuật toán lẫn về giao diện nên cần được tiếp tục phát triển trong tương lai.

## 5.2. Hướng phát triển

* Tiếp thu ý kiến, phát triển thêm các tính năng mới.
* Phát triển và đổi mới giao diện.
* Tiếp tục phát triển thuật toán nhằm tối ưu thời gian xử lí chương trình.

**TÀI LIỆU THAM KHẢO**

[1] Lê Minh Hoàng, *Giải thuật & Lập trình*, Đại học Sư phạm Hà Nội, 2003.

[2] Nguyễn Thanh Bình, Phạm Minh Tuấn, Đặng Thiên Bình, *Phân tích và thiết  
kế giải thuật*, NXB GD, 2016.

[3]Lê Thị Mỹ Hạnh, Giáo trình môn học Lập trình hướng đối tượng, 2002.