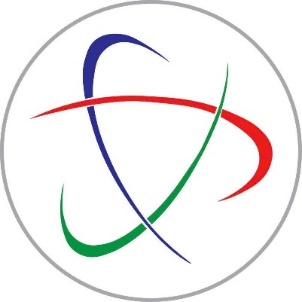
**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA ĐIỆN TỬ - VIỄN THÔNG**

****

**PBL1: LẬP TRÌNH**

**ĐỀ TÀI  
Chương trình quản lý tàu thuyền du lịch**

**GV hướng dẫn: TS. Văn Phú Tuấn**

**Sinh viên thực hiện:**

|  |  |  |
| --- | --- | --- |
| 1. **Phạm Đông Trà** | **Lớp: 24VDT** | **MSSV:106240366** |
| 1. **Hồ An Ninh** | **Lớp: 24DT4** | **MSSV:** |

**Đà Nẵng, 2025**

**BẢNG PHÂN CÔNG CÔNG VIỆC TRONG NHÓM**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **Nhiệm vụ** | **Khối lượng** |
| 1 | Phạm Đông Trà | * Viết chương trình * Viết báo cáo * Upload file lên GitHub * Quay video demo | 50% |
| 2 | Hồ An Ninh | * Chỉnh sửa báo cáo * Làm slides thuyết trình * Vẽ lưu đồ | 50% |

MỤC LỤC

MỞ ĐẦU 6

1. GIỚI THIỆU ĐỀ TÀI 7

1.1. Mô tả tổng quan về đề tài 7

1.2. Tính cấp thiết và ý nghĩa của đề tài 7

1.3. Mục tiêu và phạm vi nghiên cứu 7

1.3.1. Mục tiêu: 7

1.3.2. Phạm vi nghiên cứu: 8

2. CẤU TRÚC CHƯƠNG TRÌNH CỦA DỰ ÁN 8

2.1. Struct FoodItem 8

2.2. Struct RentalRecord 9

2.3. Struct Shift 9

2.4. Class Boat 9

2.4.1. Protected 9

2.4.2. Public 10

2.5. Class SmallBoat 11

2.5.1. Public 11

2.6. Class LargeBoat 11

2.6.1. Public 11

2.7. Class BoatManager 11

2.7.1. Privated 11

2.7.2. Public 12

2.8. Các hàm hỗ trợ 13

2.8.1. displayHeader 13

2.8.2. isValidBoatType 13

2.8.3 pressEnterToContinue 13

2.8.4.pressEnterToContinue2 13

2.8.5. displayMenu 13

2.8.6. generateTicketCode 13

2.8.7. isValidNumber 14

3. LƯU ĐỒ GIẢI THUẬT 15

3.1. Class Boat 15

3.1.1. Hàm bookSeats 15

3.1.2. Hàm cancelSeats 16

3.1.3. Hàm saveToFile 16

3.1.4. Hàm loadFromFile 17

3.2. Class SmallBoat 18

3.2.1. Hàm creatBoat 18

3.2.2. Hàm display 18

3.3. Class LargeBoat 19

3.3.1 Hàm createBoat 19

3.3.2. Hàm display 20

3.4. Class BoatManager 20

3.4.1. Hàm addBoat 20

3.4.2. Hàm deleteBoat 21

3.4.3. Hàm displayAllBoat 21

3.4.4. Hàm rentSeats 22

3.4.5. Hàm cancelRent 22

3.4.6. Hàm showAvailableBoat 23

3.4.7. Hàm displayRentalHistory 23

3.4.8. Hàm showMostRentedBoat 24

3.4.9. Hàm showSeatsPerShift 24

3.5. Các hàm hỗ trợ 25

3.5.1. generateTicketCode 25

4. KẾT QUẢ CHƯƠNG TRÌNH 26

4.1. Giao diện chính 26

4.2. Chức năng thêm thuyền 26

4.3. Chức năng xóa thuyền 27

4.4. Chức năng hiển thị tất cả các thuyền 27

4.5. Chức năng mua vé 28

4.6. Chức năng hủy vé 28

4.7. Chức năng hiển thị thuyền còn trống 29

4.8. Chức năng hiển thị lịch sử thuê/hủy 29

4.9. Chức năng hiển thị thuyền được thuê nhiều nhất 30

4.10. Chức năng hiển thị tổng số ghế được thuê trên một thuyền 30

4.11. Phân tích và đánh giá kết quả 30

5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 31

5.1. Kết luận 31

5.2. Đề xuất hướng phát triển 31

TÀI LIỆU THAM KHẢO 32

# MỞ ĐẦU

Trước khi đến với bài báo cáo, chúng em xin gửi lời cảm ơn chân thành nhất đến các thầy cô bộ môn PBL1 khoa Điện Tử- Viễn Thông và đặc biệt là thầy Văn Phú Tuấn. Thầy đã tận tình hướng dẫn và hỗ trợ chúng em trong suốt quá trình thực hiện đề tài cả về mặt lý thuyết lẫn thực hành. Nhờ có sự chỉ dẫn tận tâm của thầy, chúng em đã có cơ hội tiếp cận và rèn luyện các kỹ năng phân tích, lập trình và giải quyết các vấn đè gặp phải, cũng như việc hiểu rõ về các khái niệm mới trong môn học.

     Dù đã nỗ lực hoàn thành đề tài một cách tốt nhất trong khả năng, chắc chắn nhóm em vẫn không tránh khỏi những thiếu sót. Rất mong nhận được sự thông cảm, góp ý của giảng viên để chúng em có thể hoàn thiện hơn trong các lần thực hiện sau.

**Hồ An Ninh**

**Phạm Đông Trà**

# 1. GIỚI THIỆU ĐỀ TÀI

## 1.1. Mô tả tổng quan về đề tài

* Đề tài “Quản lý thuyền trên sông Hàn” hướng đến việc xây dựng một hệ thống phần mềm hỗ trợ quản lý các hoạt động cho thuê thuyền, bao gồm quản lý thông tin thuyền, đặt và hủy ghế, theo dõi lịch sử thuê, và quản lý thực đơn đồ ăn/uống đi kèm. Hệ thống được thiết kế để thay thế phương pháp quản lý thủ công, mang lại hiệu quả, chính xác và thuận tiện trong quản lý.
* Hệ thống cho phép thực hiện các nghiệp vụ như thêm/xóa thuyền, đặt ghế trên các ca làm việc, hủy vé, hiển thị thuyền còn trống, thống kê lịch sử thuê, và xác định thuyền được thuê nhiều nhất.
* Giao diện đơn giản, dễ sử dụng cho cả nhân viên quản lý và khách hàng.

## 1.2. Tính cấp thiết và ý nghĩa của đề tài

* Trong bối cảnh du lịch phát triển mạnh mẽ, đặc biệt tại các địa điểm như sông Hàn, việc quản lý dịch vụ cho thuê thuyền trở nên cần thiết để đáp ứng nhu cầu của khách hàng và tối ưu hóa hoạt động kinh doanh.
* Nhiều đơn vị hiện nay vẫn sử dụng phương pháp quản lý thủ công, dẫn đến các vấn đề như:
* Khó khăn trong việc theo dõi số ghế trống và lịch sử thuê.
* Dễ xảy ra sai sót trong quản lý vé và doanh thu.
* Mất thời gian xử lý các thao tác như đặt hoặc hủy ghế.

Do đó, đề tài mang ý nghĩa thực tiễn cao, góp phần:

* Tăng hiệu quả quản lý và giảm khối lượng công việc cho nhân viên.
* Cải thiện trải nghiệm khách hàng thông qua quy trình đặt vé nhanh chóng.
* Hỗ trợ hiện đại hóa quản lý dịch vụ du lịch, phù hợp với xu hướng công nghệ.

## 1.3. Mục tiêu và phạm vi nghiên cứu

### 1.3.1. Mục tiêu:

* Phát triển hệ thống phần mềm quản lý thuyền với các chức năng cốt lõi:
  + Quản lý thông tin thuyền (thêm, xóa, hiển thị).
  + Quản lý đặt và hủy ghế trên các ca làm việc.
  + Quản lý thực đơn đồ ăn/uống đi kèm khi đặt vé.
  + Thống kê lịch sử thuê, thuyền được thuê nhiều nhất, và số ghế đã thuê trên mỗi ca.
  + Đảm bảo giao diện đơn giản, dễ thao tác cho người dùng.
  + Đảm bảo hệ thống có khả năng mở rộng và bảo trì trong tương lai.

### 1.3.2. Phạm vi nghiên cứu:

* Hệ thống được thiết kế cho các đơn vị cung cấp dịch vụ thuê thuyền quy mô nhỏ đến vừa, chẳng hạn như:
  + Các công ty du lịch tại sông Hàn.
  + Các đơn vị tổ chức tour trên sông.
* Không bao gồm các chức năng nâng cao như:
  + Đặt vé trực tuyến qua internet.
  + Tích hợp hệ thống thanh toán hoặc quét mã vạch/mã QR.
  + Giao diện đồ họa người dùng (GUI).

# 2. CẤU TRÚC CHƯƠNG TRÌNH CỦA DỰ ÁN

## 2.1. Struct FoodItem

|  |  |
| --- | --- |
| **Biến** | **Kiểu dữ liệu** |
| name | string |
| price | float |

* Lưu thông tin của các món ăn uống, gồm tên (name) và giá (price).

## 2.2. Struct RentalRecord

|  |  |
| --- | --- |
| **Biến** | **Kiểu dữ liệu** |
| boatID | string |
| ticketPrice | float |
| seatsRented | float |
| shiftIndex | int |
| action | string |
| foodItems | vector<FoodItem> |
| foodCost | float |

* Lưu thông tin của lịch sử ghi thuê, bao gồm ID thuyền đã thuê, mã số vé, số ghế đã thuê, ca thuê số bao nhiêu, hành động tương ứng với lần ghi thuê là hủy (CANCEL) hoặc thuê (RENT), danh sách món ăn và chi phí món ăn

## 2.3. Struct Shift

|  |  |
| --- | --- |
| **Biến** | **Kiểu dữ liệu** |
| startHour | int |
| endHour | int |
| pricePerHour | float |
| seatsBooked | float |

* Lưu thông tin của một ca thuê của một đối tượng thuyền. Bao gồm thời gian bắt đầu, thời gian kết thúc, giá cho một giờ và số ghế đã đặt đối với ca đó.

## 2.4. Class Boat

### 2.4.1. Protected

|  |  |
| --- | --- |
| **Biến** | **Kiểu dữ liệu** |
| boatID | string |
| boatName | string |
| type | char |
| weight | float |
| isAvailable | bool |
| seats | float |
| numshifts | int |
| shifts | vector<Shift> |
| rentalCount | int |

### 2.4.2. Public

* Các hàm getter: Trả về giá trị của các thuộc tính như boatID, boatName, type, seats, rentalCount, shifts, numShifts, và isAvailable.
* Hàm ảo thuần túy display có chức năng cho phép ghi đè trong các lớp con, phục vụ cho việc in thông tin của thuyền ra màn hình. Tùy thuộc vào loại thuyền mà sẽ có cách hiển thị khác nhau.
* Hàm ảo thuần túy creatBoat() tương tự hàm display, cũng dùng để ghi đè trong các lớp con nhằm tạo ra tính đa hình cho chương trình. Tùy thuộc vào loại thuyền mà sẽ có các giới hạn nhập( Nếu như là loại thuyền SmallBoat, số lượng ghế nhập phải bé hơn 50).
* Hàm setAvailability: Đặt trạng thái sẵn sàng của thuyền (true nếu còn ghế trống, false nếu hết).
* Hàm bookSeats: Đặt số ghế trên một ca làm việc, tăng rentalCount, cập nhật trạng thái thuyền nếu tất cả ca đã đầy.
* Hàm cancelSeats: Hủy số ghế đã đặt, cập nhật số ghế trống và trạng thái thuyền.
* Hàm saveToFile: Lưu thông tin thuyền vào file boats.txt, bao gồm boatID, boatName, type, weight, isAvailable, seats, numShifts, thông tin các ca và rentalCount.
* Hàm loadFromFile: Đọc thông tin thuyền từ file boats.txt để khởi tạo đối tượng.

## 2.5. Class SmallBoat

Đây là lớp con của lớp cơ sở Boat.

### 2.5.1. Public

* Hàm khởi tạo constructor, có chức năng đặt giá trị Type ban đầu của đối tượng là ‘S’.
* Hàm creatBoat: là hàm ghi đè của hàm thuần ảo creatBoat của lớp Boat, có chức năng nhập thông tin thuyền nhỏ (trọng lượng tối đa 50 tấn, số ghế tối đa 100). Kiểm tra đầu vào hợp lệ (ID, tên, trọng lượng, số ghế, số ca, giờ ca, giá ca).
* Hàm display: tương tự hàm creatBoat, cũng là hàm ghi đè của hàm thuần ảo display của lớp Boat, có chức năng hiển thị thông tin thuyền nhỏ, bao gồm ID, tên, loại (‘S’), trọng lượng, số ghế, trạng thái, thông tin ca, và số lần thuê.

## 2.6. Class LargeBoat

Đây là lớp con của lớp cơ sở Boat.

### 2.6.1. Public

* Hàm khởi tạo constructor, có chức năng đặt giá trị Type ban đầu của đối tượng là ‘L’.
* Hàm creatBoat: là hàm ghi đè của hàm thuần ảo creatBoat của lớp Boat, có chức năng nhập thông tin thuyền lớn (trọng lượng tối thiểu 50 tấn, số ghế tối thiểu 100). Kiểm tra đầu vào hợp lệ (ID, tên, trọng lượng, số ghế, số ca, giờ ca, giá ca).
* Hàm display: tương tự hàm creatBoat, cũng là hàm ghi đè của hàm thuần ảo display của lớp Boat, có chức năng hiển thị thông tin thuyền lớn, bao gồm ID, tên, loại ( ‘L’), trọng lượng, số ghế, trạng thái, thông tin ca, và số lần thuê.

## 2.7. Class BoatManager

### 2.7.1. Privated

|  |  |
| --- | --- |
| **Biến** | **Kiểu dữ liệu** |
| boats | vector<Boat\*> |
| rentalHistory | vector<RentalRecord> |
| foodMenu | vector<FoodItem> |
| password | const string |

* Khởi tạo giá trị cho foodMenu gồm: Banh Mi giá 15000, Coca Cola giá 10000, Trai Cay giá 30000 (VND)
* Khởi tạo giá trị mặc định cho mật khẩu máy chủ: password = BKDaNang

### 2.7.2. Public

* Hàm khởi tạo constructor, dùng để tải dữ liệu ban đầu của file lên
* Hàm addBoat: Có chức năng thêm thuyền mới (nhỏ hoặc lớn), kiểm tra nhằm đảm bảo ID và tên không trùng lặp, lưu thông tin của thuyền vừa nhập vào file.
* Hàm deleteBoat: Có chức năng xóa thuyền theo ID hoặc tên. Đảm bảo việc nhập chính xác mật khẩu của máy chủ và thuyền phải không có bản ghi thuê thì mới được xóa, sau đó lưu cập nhật vào file.
* Hàm rentSeats: Đặt ghế trên thuyền bằng cách nhập ID thuyền. Sau đó sẽ lựa chọn ca, số ghế, và món ăn (nếu có) phù hợp. Tạo mã vé duy nhất, lưu bản ghi thuê, cập nhật số ghế vào file.
* Hàm cancelRent: Có chức năng hủy ghế theo mã vé, sau đó cập nhật số ghế trống, thêm bản ghi hủy, lưu vào file.
* Hàm howAvailableBoats: Hiển thị các thuyền còn ghế trống.
* Hàm displayRentalHistory: Hiển thị lịch sử thuê/hủy, bao gồm thông tin thuyền, vé, ghế, ca, món ăn, và chi phí.
* Hàm showMostRentedBoat: Hiển thị thuyền có số lần thuê cao nhất.
* Hàm showSeatsPerShift: Hiển thị số ghế đã thuê trên mỗi ca của một thuyền.
* Hàm saveBoatsToFile: có chức năng lưu danh sách thuyền vào file “boats.txt” nhằm bảo mật và lưu trữ thông tin, phục vụ cho việc quản lý về lâu dài.
* Hàm saveRentalHistoryToFile: có chức năng lưu danh sách đặt vé/ hủy vé vào file “rental\_History.txt” nhằm bảo mật và lưu trữ thông tin, phục vụ cho việc quản lý về lâu dài.
* Các hàm loadBoatsFromFile/loadRentalHistoryFromFile: Hàm hỗ trợ việc đọc file “boats.txt”/”rental\_history.txt” đã tồn tại trước đó. Đảm bảo cho việc hoạt động liên tục của chương trình kể cả khi khởi động lại.

## 2.8. Các hàm hỗ trợ

### 2.8.1. displayHeader

* Hiển thị tiêu đề chương trình, nhằm tăng tính tiếp cận người dùng.

### 2.8.2. isValidBoatType

* Có chức năng kiểm tra xem liệu loại thuyền nhập vào có hợp lệ không ( là một trong hai giá trị ‘S’ và ‘L’). Nếu đúng chương trình sẽ trả về true, nếu sai sẽ trả về false.

### 2.8.3 pressEnterToContinue

* Hiển thị dòng lệnh yêu cầu người dùng cần nhấn enter để tiếp tục chương trình.

### 2.8.4. pressEnterToContinue2

* Có chức năng tương tự hàm pressEnterToContinue, tuy nhiên khác ở chỗ hàm sẽ không xóa bộ đệm, phù hợp cho một số trường hợp đặc biệt mà việc xóa bộ nhớ đệm sẽ gây vấn đề. Ví dụ cụ thể, ở hàm deleteBoat của class BoatManager, sau khi nhập phương thức xóa là ID hoặc tên của thuyền, chương trình sẽ sử dụng getline để đọc đầu vào. Nếu ở đây người dùng chỉ nhấn Enter mà không nhập bất kì giá trị nào khác, chương trình sẽ kết thúc và kí tự \n sẽ được đọc và để lại trong bộ nhớ đệm. Nếu dùng pressEnterToContinue, cin.ignore sẽ xóa \n và yêu cầu nhấn Enter thêm lần nữa, dẫn đến trải nghiệm người dùng không tốt (phải nhấn Enter hai lần). Do đó, pressEnterToContinue2 được sử dụng để tránh việc xóa bộ đệm dư thừa, chỉ cần một lần nhấn Enter để tiếp tục.

### 2.8.5. displayMenu

* Có chức năng hiển thị menu chính, bao gồm các chức năng chính của chương trình.

### 2.8.6. generateTicketCode

* Có chức năng tạo một vé duy nhất bằng hàm srand và rand. Cách hoạt động:

1. Đặt hạt giống (seed) cho hàm srand() với tham số mà số giây kể từ ngày 1/1/1970 tới bây giờ. Với mỗi lần gọi hàm generateTicketCode, chúng ta sẽ chạy srand một lần. Vì vậy hạt giống sẽ luôn khác nhau, đảm bảo cho việc tạo chuỗi giả ngẫu nhiên sẽ không trùng lặp.
2. chạy vòng lặp để tìm ra mã vé. Dùng rand() để chọn tạo ra chuỗi giả ngẫu nhiên, rồi ánh xạ sang chuỗi kí tự chars đã được xác định bằng ***rand() % chars.length()*** . Sau mỗi vòng lặp, hàm rand() sẽ tăng tới giá trị tiếp theo của chuỗi giả ngẫu nhiên, do đó đảm bảo được tính độc nhất của vé tàu.

chars = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"

1. Mã vé được tạo ra bằng cách thêm vào phía sau “TICKET” 06 các ký tự đã được ánh xạ ở trên.
2. Để đảm bảo tính độc nhất, chương trình sẽ chạy và kiểm tra xem trong có tồn tại mã vé nào trúng với mã vé vừa được tạo ra hay không. Nếu có sẽ tạo ra mã vé mới.

### 2.8.7. isValidNumber

* Hàm có chức năng kiểm tra dữ liệu nhập vào có phải là số hợp lệ hay không.

# 3. LƯU ĐỒ GIẢI THUẬT

## 3.1. Class Boat

### 3.1.1. Hàm bookSeats

A black background with white squares and circles

AI-generated content may be incorrect.

### 3.1.2. Hàm cancelSeats

A black background with white squares and circles

AI-generated content may be incorrect.

### 3.1.3. Hàm saveToFile

A white background with black text

AI-generated content may be incorrect.

### 3.1.4. Hàm loadFromFile

A screenshot of a computer screen

AI-generated content may be incorrect.

## 3.2. Class SmallBoat

### 3.2.1. Hàm creatBoat

A black background with white squares

AI-generated content may be incorrect.

### 3.2.2. Hàm display



## 3.3. Class LargeBoat

### 3.3.1 Hàm createBoat

A black background with white squares

AI-generated content may be incorrect.

### 3.3.2. Hàm display



## 3.4. Class BoatManager

### 3.4.1. Hàm addBoat

A black background with white rectangles and text

AI-generated content may be incorrect.

### 3.4.2. Hàm deleteBoat

A black background with white rectangles

AI-generated content may be incorrect.

### 3.4.3. Hàm displayAllBoat

A white background with black text

AI-generated content may be incorrect.

### 3.4.4. Hàm rentSeats

A black background with white rectangles

AI-generated content may be incorrect.

### 3.4.5. Hàm cancelRent

A black and white rectangle with black text

AI-generated content may be incorrect.

### 3.4.6. Hàm showAvailableBoat

A screenshot of a cell phone

AI-generated content may be incorrect.

### 3.4.7. Hàm displayRentalHistory

A black background with a white diamond

AI-generated content may be incorrect.

### 3.4.8. Hàm showMostRentedBoat

A black background with white rectangles and black text

AI-generated content may be incorrect.

### 3.4.9. Hàm showSeatsPerShift

A black background with white rectangles

AI-generated content may be incorrect.

## 3.5. Các hàm hỗ trợ

### 3.5.1. generateTicketCode

A screenshot of a phone

AI-generated content may be incorrect.

# 4. KẾT QUẢ CHƯƠNG TRÌNH

## 4.1. Giao diện chính

A screenshot of a computer

AI-generated content may be incorrect.

## 4.2. Chức năng thêm thuyền

A screenshot of a video game

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

## 4.3. Chức năng xóa thuyền

A screenshot of a computer

AI-generated content may be incorrect.

## 4.4. Chức năng hiển thị tất cả các thuyền

A screenshot of a computer

AI-generated content may be incorrect.’

## 4.5. Chức năng mua vé

A screenshot of a computer

AI-generated content may be incorrect.

## 4.6. Chức năng hủy vé

A screen shot of a computer screen

AI-generated content may be incorrect.

## 4.7. Chức năng hiển thị thuyền còn trống

A screenshot of a computer program

AI-generated content may be incorrect.

## 4.8. Chức năng hiển thị lịch sử thuê/hủy

A screen shot of a computer

AI-generated content may be incorrect.

## 4.9. Chức năng hiển thị thuyền được thuê nhiều nhất

A screenshot of a computer

AI-generated content may be incorrect.

## 4.10. Chức năng hiển thị tổng số ghế được thuê trên một thuyền

A screenshot of a computer

AI-generated content may be incorrect.

## 4.11. Phân tích và đánh giá kết quả

Chương trình có menu đầy đủ với 9 lựa chọn trong việc quản lý và mua vé tàu thuyền: quản lý tàu thuyền, quản lý mua vé tàu và hủy vé tàu. Với mỗi lựa chọn, sẽ có các chức năng riêng biệt. Trong đó có chức năng quay lại menu chính để giúp người dùng sử dụng các tính năng khác. Nhìn chung, các kết quả của chương trình đã đáp ứng được các nhu cầu cơ bản của một chương trình quản lý và mua vé tàu thuyền.

# 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 5.1. Kết luận

Chương trình đáp ứng các chức năng cơ bản của hệ thống quản lý thuyền, hỗ trợ lưu trữ và truy xuất thông tin nhanh chóng, hiệu quả hơn so với phương pháp thủ công. Các tính năng như đặt/hủy ghế, quản lý thực đơn, và thống kê giúp cải thiện quy trình vận hành dịch vụ.

Hạn chế bao gồm: giao diện console chưa thân thiện, và thiếu các tính năng nâng cao như thanh toán hoặc thông báo.

    Trong tương lai, nhóm mong muốn tiếp tục phát triển chương trình theo hướng hoàn thiện hơn: nâng cao độ ổn định, bổ sung các chức năng nâng cao, cải tiến giao diện người dùng và tích hợp thêm các công nghệ hiện đại để phù hợp với nhu cầu thực tế.

## 5.2. Đề xuất hướng phát triển

* Sửa các lỗi, bugs còn tồn đọng trong chương trình.
* Thêm tính năng gia hạn vé, thông báo ca sắp hết ghế, và báo cáo doanh thu.
* Tích hợp thanh toán trực tuyến và đặt vé qua internet.
* Phát triển giao diện đồ họa (GUI) để cải thiện trải nghiệm người dùng.
* Hỗ trợ quản lý khách hàng (thông tin cá nhân, lịch sử đặt vé).
* Tối ưu hóa chương trình giúp chương trình chạy nhanh và tiết kiệm bộ nhớ hơn.

# 

# TÀI LIỆU THAM KHẢO

[1]. “Reading and writing C++ vector to a file.” Stack overflow. https://stackoverflow.com/questions/2469531/reading-and-writing-c-vector-to-a-file (accessed March 18, 2010).

[2]. “Inheritance in C++.” GeeksForGeeks. https://www.geeksforgeeks.org/cpp/inheritance-in-c/ (accessed May 15, 2025).

[3]. “How do you generate random strings in C++” Stack overflow. https://stackoverflow.com/questions/2146792/how-do-you-generate-random-strings-in-c (accessed Jan 27, 2010).

[4]. “File Handling through C++ Classes.” GeeksForGeeks. https://www.geeksforgeeks.org/cpp/file-handling-c-classes/ (accessed May 15, 2025).

[5]. “C++ Polymorphism.” GeeksForGeeks. <https://www.geeksforgeeks.org/cpp/cpp-polymorphism/>(accessed May 15, 2025).

[6]. “C++ String to Float/Double and Vice-Versa.” GeeksForGeeks. https://www.geeksforgeeks.org/cpp/cpp-string-to-float-double-and-vice-versa/ (accessed Apr 28, 2025).

[7]. “[STL]. Vector Trong C++.” 28tech. https://blog.28tech.com.vn/stl-vector-trong-c (accessed Aug 21, 2021).

[8]. “[C++]. Kiểu Dữ Liệu Suy Luận Auto Trong C++” 28tech. https://blog.28tech.com.vn/c-kieu-du-lieu-suy-luan-auto-trong-c (accessed Aug 19, 2021).