**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC BÁCH KHO****A**

**KHOA CÔNG NGHỆ THÔNG TIN**



**PBL2: DỰ ÁN CƠ SỞ LẬP TRÌNH**

**Đề tài : Xây dựng ứng dụng đặt vé máy bay**

**SINH VIÊN THỰC HIỆN:**

**Tên sinh viên 1: Lê Thảo Vy LỚP: 22T\_Nhat2 Tên sinh viên 2: Lê Trần Diễm Ly LỚP: 22T\_Nhat2**

**GIẢNG VIÊN HƯỚNG DẪN: GVC.ThS. Trần Hồ Thuỷ Tiên.**

**Đà Nẵng 15/12/2023**

LỜI MỞ ĐẦU

Trong bối cảnh hiện nay, việc ứng dụng công nghệ thông tin vào các ngành nghề đã không còn là một xu hướng mà là một yêu cầu tất yếu, nhất là trong ngành hàng không - một lĩnh vực đòi hỏi sự chính xác và hiệu quả cao. Đồ án này, dưới sự hướng dẫn của quý thầy cô, được thực hiện với mục tiêu thiết kế và phát triển một ứng dụng đặt vé máy bay trực tuyến, tiên tiến và dễ sử dụng, thúc đẩy sự nhanh chóng và linh hoạt trong việc mua sắm và quản lý hành trình bay của khách hàng.

Sự kết hợp giữa lý thuyết và kỹ năng lập trình đã được áp dụng một cách sáng tạo để tạo ra một giải pháp đáp ứng nhu cầu thiết yếu của người dùng trong kỷ nguyên số. Bằng cách này, đồ án không chỉ phản ánh sự nỗ lực và sự hiểu biết sâu sắc của sinh viên về lập trình ứng dụng, mà còn đóng góp vào sự tiện lợi cho cộng đồng, đồng thời khẳng định tiềm năng của việc tích hợp công nghệ vào các dịch vụ hàng không.

Trân trọng cảm ơn sự chú ý và chúng em rất mong mỏi những ý kiến đóng góp quý báu của quý thầy cô để đồ án được hoàn thiện hơn.

MỤC LỤC

[1. GIỚI THIỆU ĐỀ TÀI 4](#_Toc153558538)

[2. PHÂN TÍCH CHỨC NĂNG HỆ THỐNG 4](#_Toc153558539)

[2.1. Tìm kiếm: 4](#_Toc153558540)

[2.2. Cập nhật: 4](#_Toc153558541)

[2.3. Thanh toán: 4](#_Toc153558542)

[2.4. Hiển thị: 4](#_Toc153558543)

[3. THIẾT KẾ CẤU TRÚC DỮ LIỆU 4](#_Toc153558544)

[3.1. Phát biểu bài toán 4](#_Toc153558545)

[3.2. Phân tích và ứng dụng cấu trúc dữ liệu trong hệ thống 5](#_Toc153558546)

[4. PHÂN TÍCH HƯỚNG ĐỐI TƯỢNG VÀ TRIỂN KHAI HỆ THỐNG 5](#_Toc153558547)

[4.1. Cấu trúc hệ thống hướng đối tượng 5](#_Toc153558548)

[4.1.1. Lớp Airplane 6](#_Toc153558549)

[4.1.2. Lớp Flight 7](#_Toc153558550)

[4.1.3. Lớp DateTime: 10](#_Toc153558551)

[4.1.4. Lớp TicketInfo 11](#_Toc153558552)

[4.1.5. Lớp CustomerInfo 14](#_Toc153558553)

[4.1.6. Lớp eTicket 16](#_Toc153558554)

[4.2. Kết quả 17](#_Toc153558555)

[4.2.1. Giao diện chính của chương trình 17](#_Toc153558556)

[4.2.2. Kết quả thực thi của chương trình: 19](#_Toc153558557)

[4.2.3. Nhận xét 24](#_Toc153558558)

[a. Kết luận 25](#_Toc153558559)

[b. Hướng phát triển 25](#_Toc153558560)

[[1] TS. Đặng Hoài Phương. Giáo trình *Lập trình hướng đối tượng*, trường Đại học Bách Khoa – ĐHĐN. 26](#_Toc153558561)

[[2] Mark Allen Weiss (1994). *Data Structures and Algorithm Analysis in C,* Pearson Education. 26](#_Toc153558562)

DANH MỤC HÌNH VẼ

[Hình 1. Mô hình cơ sở dữ liệu hệ thống 5](#_Toc153564021)

[Hình 2. File quản lý máy bay 7](#_Toc153564022)

[Hình 3. File quản lý chuyến bay 9](#_Toc153564023)

[Hình 4. File quản lý định dạng thời gian 11](#_Toc153564024)

[Hình 5. Hiển thị sơ đồ ghế của chuyến bay 12](#_Toc153564025)

[Hình 6. File quản lý đặt vé cho chuyến bay 13](#_Toc153564026)

[Hình 7. File quản lý thông tin khách hàng 15](#_Toc153564027)

[Hình 8. File quản lý xuất vé điện tử 16](#_Toc153564028)

[Hình 9. Giao diện menu chính 16](#_Toc153564029)

[Hình 10. Giao diện nhập thông tin chuyến bay 17](#_Toc153564030)

[Hình 11. Giao diện chọn chuyến bay 17](#_Toc153564031)

[Hình 12. Giao diện chọn vị trí ghế ngồi 17](#_Toc153564032)

[Hình 13. Giao diện thanh toán 18](#_Toc153564033)

[Hình 14. Hiển thị menu 19](#_Toc153564034)

[Hình 15. Hiển thị yêu cầu nhập thông tin chuyến bay 19](#_Toc153564035)

[Hình 16. Hiển thị danh sách chuyến bay lượt đi phù hợp với yêu cầu người dùng 19](#_Toc153564036)

[Hình 17. Hiển thị danh sách chuyến bay lượt về phù hợp với yêu cầu người dùng 19](#_Toc153564037)

[Hình 18. Hiển thị sơ đồ ghế ngồi của chuyến bay 20](#_Toc153564038)

[Hình 19. Giao diện yêu cầu người dùng nhập thông tin cá nhân 20](#_Toc153564039)

[Hình 20. Hiển thị số tiền cần thanh toán và các phương thức thanh toán 21](#_Toc153564040)

[Hình 21. Giao diện thanh toán trực tuyến và xác nhận thanh toán 21](#_Toc153564041)

[Hình 22. Giao diện thanh toán sau và xác nhận đặt vé 22](#_Toc153564042)

[Hình 23. Hiển thị vé điện tử 22](#_Toc153564043)

# GIỚI THIỆU ĐỀ TÀI

Đây là đề tài về quản lý đặt vé máy bay, chương trình cung cấp tiện ích giúp khách hàng có thể tìm được chuyến bay theo nhu cầu và thực hiện đặt vé, thanh toán trực tuyến. Điều này giúp tiết kiệm thời gian và công sức trong việc chọn mua vé máy bay thay vì phải mua trực tiếp tại sân bay nhưng vẫn đảm bảo tính bảo mật cao và dễ dàng thực hiện.

# PHÂN TÍCH CHỨC NĂNG HỆ THỐNG

## Tìm kiếm:

Tìm kiếm thông tin về chuyến bay phù hợp với yêu cầu người dùng: nơi đi, nơi

đến, ngày khởi hành.

## Cập nhật:

- Thông tin chuyến bay.

- Thông tin khách hàng.

## Thanh toán:

Xác định tổng số tiền cần phải thanh toán của khách hàng, đưa ra các phương thức thanh toán khác nhau cho khách hàng lựa chọn và xác nhận thanh toán theo yêu cầu.

## Hiển thị:

Hiển thị vé điện tử sau khi khách hàng hoàn thành các bước đặt vé và thanh toán.

# THIẾT KẾ CẤU TRÚC DỮ LIỆU

## Phát biểu bài toán

**Input:**

* Tìm kiếm chuyến bay phù hợp
* Điểm xuất phát
* Điểm đến
* Ngày đi
* Ngày về
* Số lượng hành khách
* Thông tin khách hàng
* Tên khách hàng
* E-mail
* SĐT
* Mã OTP
* Quản lý thông tin
* Chuyến bay
* Thông tin về hành khách (danh sách các khách hàng đã đặt vé)
* Xác nhận thông tin đặt vé

**Output:**

* Hiển thị:
* Danh sách chuyến bay phù hợp với yêu cầu
* Danh sách chỗ ngồi
* Tính toán số tiền phải thanh toán
* Vé điện tử với ID riêng cho mỗi vé
* Cập nhật:
* Thông tin chi tiết về chuyến bay
* Cập nhật những ghế có thể đặt( ghế hiển thị màu trắng)

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

* Sử dụng danh sách liên kết đơn để quản lý các máy bay và chuyến bay.
* Sử dụng mảng động để lưu các vị trí ghế đã được đặt.

A screenshot of a computer screen

Description automatically generated

Hình 1. Mô hình cơ sở dữ liệu hệ thống

# PHÂN TÍCH HƯỚNG ĐỐI TƯỢNG VÀ TRIỂN KHAI HỆ THỐNG

## Cấu trúc hệ thống hướng đối tượng

Chương trình quản lý đặt vé chuyến bay trên thể hiện rõ cấu trúc hệ thống hướng đối tượng thông qua việc sử dụng các lớp và mối quan hệ giữa chúng:

* Đối tượng: gồm 3 đối tượng là Airplane, Flight và Ticket. Các đối tượng này cùng hoạt động với nhau để quản lý thông tin về máy bay, chuyến bay, thông tin khách hàng và việc đặt vé điện tử.
* Mối quan hệ giữa các đối tượng: sử dụng 4 nguyên tắc cơ bản của OOP là đóng gói, kế thừa, đa hình và trừu tượng để quản lý và thao tác với các đối tượng khác nhau liên quan đến quá trình đặt vé chuyến bay.

### Lớp Airplane

*Thuộc tính private:*

* + airplaneID: Mã định danh của máy bay.
  + airplaneType: Loại máy bay.

*Thuộc tính protected:*

* + seatCount: Số lượng ghế trên máy bay(để có thể truy cập từ các lớp dẫn xuất).

*Thuộc tính public:*

* + head1: Con trỏ tĩnh trỏ đến đầu danh sách máy bay.
  + next: Con trỏ trỏ đến máy bay tiếp theo trong danh sách.

Hàm tạo và hàm hủy:

* + Airplane(): Hàm tạo mặc định.
  + Airplane(const std::string& id, const std::string& type, int seats): Hàm tạo với tham số để khởi tạo các thuộc tính của máy bay.
  + ~Airplane(): Hàm hủy, sử dụng virtual khi hàm con kế thừa, nó có thể định nghĩa lại hàm hủy.

Hàm getter:

* + getAirplaneID() const: Trả về mã định danh của máy bay.
  + getAirplaneType() const: Trả về loại máy bay.
  + getSeatCount() const: Trả về số lượng ghế trên máy bay.

Các phương thức quản lý danh sách máy bay:

* + viewAirplanes() const: Hiển thị thông tin về tất cả máy bay.
  + findAirplaneByID(const std::string &id): Tìm máy bay theo mã định danh.
  + addAirplane(const std::string &id, const std::string &type, const int& seats): Thêm máy bay mới vào danh sách với dữ liệu từ file txt
  + removeAirplane(const std::string& planeID): Loại bỏ máy bay khỏi danh sách.

Phương thức đọc/ghi dữ liệu:

* + virtual void readDataFromFile(const std::string &filename): Hàm ảo để đọc dữ liệu từ tệp.
  + virtual void saveDataToFile(const std::string& filename)const: Hàm ảo để lưu dữ liệu vào tệp.

Sử dụng hàm ảo trong lớp cơ sở và định nghĩa lại nó trong lớp dẫn xuất, ngăn chặn việc gọi lại hàm của lớp cha



Hình 2. File quản lý máy bay

### Lớp Flight

* + Lớp Flight kế thừa public từ lớp Airplane: Flight sẽ kế thừa tất cả thuộc tính và phương thức protected và public của Airplane

*Thuộc tính private:*

* + flightID: Mã định danh của chuyến bay.
  + from: Điểm xuất phát của chuyến bay.
  + to: Điểm đến của chuyến bay.
  + departureDateTime: Đối tượng kiểu DateTime đại diện cho thời điểm xuất phát của chuyến bay.
  + flightStatus: Trạng thái của chuyến bay
  + initialTotalSeats: Số lượng ghế ban đầu trên máy bay.
  + price: Giá vé của chuyến bay.
  + airplaneID: Mã định danh của máy bay.
  + availableSeats: Số lượng ghế còn trống trên máy bay.

*Thuộc tính public:*

* + head: Con trỏ tĩnh trỏ đến đầu danh sách chuyến bay.
  + next: Con trỏ trỏ đến chuyến bay tiếp theo trong danh sách.

Hàm tạo:

* + Hàm tạo không đối số và hàm tạo với đối số để khởi tạo các thuộc tính của chuyến bay.

Hàm getter:

* + Các hàm lấy giá trị của các thuộc tính như getFlightID(), getAvailableSeats(), getFlightStatus(), getPrice(), getFrom(), getTo(), getDepartureDateTime(), getHead(), và getNext().

Hàm đọc và hiển thị dữ liệu:

* + readDataFromFile(const string&filename): Đọc dữ liệu từ tệp và cập nhật thông tin của chuyến bay.
  + displayFlightDetails(): Hiển thị thông tin chi tiết về chuyến bay.

Hàm thêm chuyến bay:

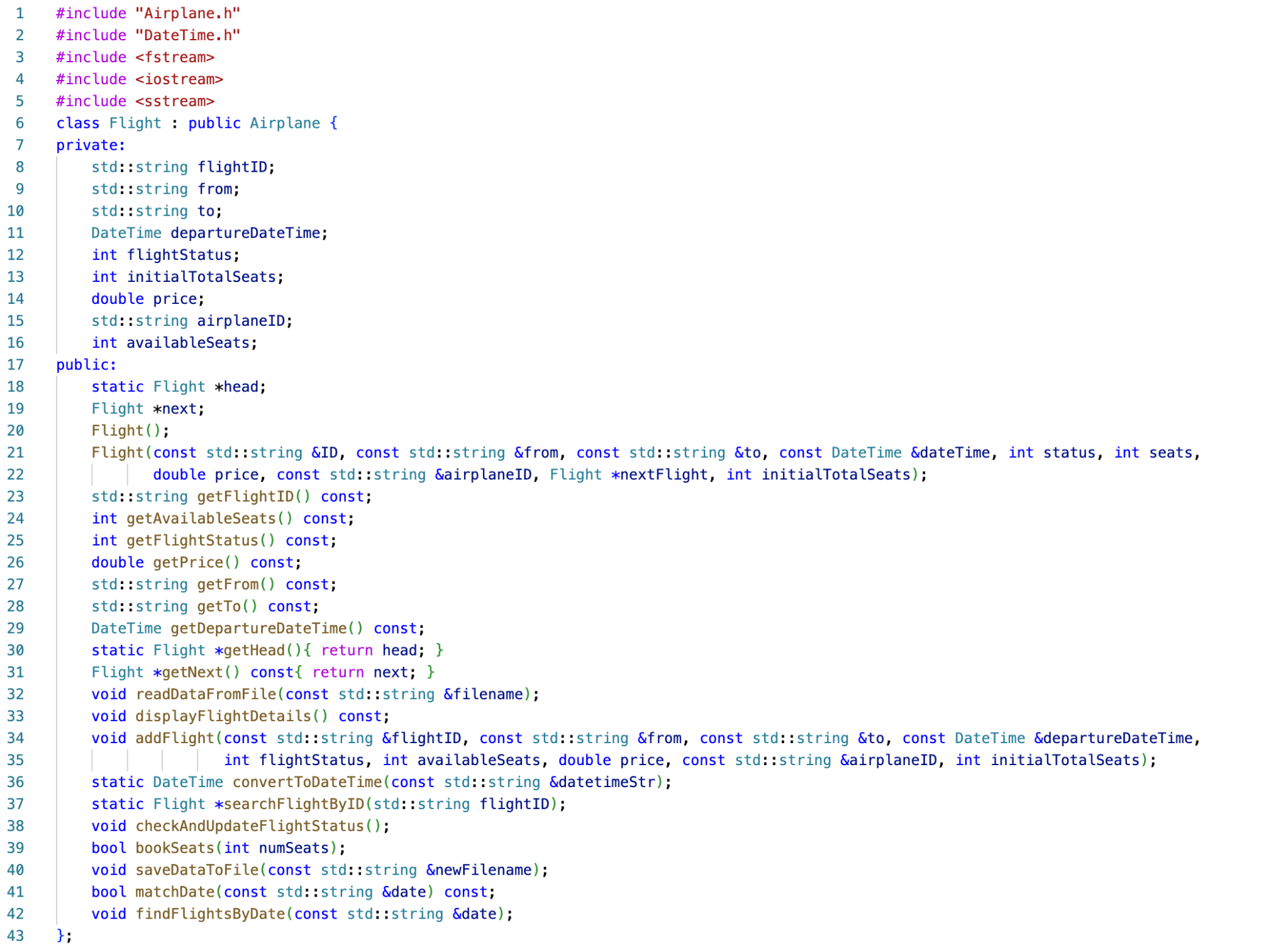
* + addFlight(): Thêm chuyến bay mới vào danh sách, với dữ liệu được đọc từ file txt.

Phương thức tĩnh:

* + convertToDateTime(): Chuyển đổi một chuỗi đại diện cho ngày giờ thành đối tượng DateTime.
  + searchFlightByID(string ID): Tìm kiếm chuyến bay theo mã định danh.

Các hàm khác:

* + checkAndUpdateFlightStatus(): Kiểm tra và cập nhật trạng thái của chuyến bay.
  + bookSeats(int numSeats): Đặt vé cho số lượng ghế được chỉ định và cập nhật số lượng ghế còn trống.
  + saveDataToFile(const string&filename): Lưu dữ liệu của chuyến bay vào một tệp mới.
  + matchDate(): So sánh ngày của chuyến bay với một ngày được chỉ định.
  + findFlightsByDate(const string&date): Tìm kiếm và hiển thị các chuyến bay theo ngày.

****

Hình 3. File quản lý chuyến bay

### Lớp DateTime:

*Thuộc tính private:*

* day, month, year, hour, minute: biểu thị các giá trị ngày, tháng, năm, giờ, phút.

*Thuộc tính public:*

Hàm tạo:

* DateTime(): hàm tạo mặc định.
* DateTime(int, int, int, int, int): hàm tạo có tham số.

Hàm getter:

* Các hàm lấy giá trị thuộc tính bao gồm: getDay(), getMonth(), getYear(), getHour(), getMinute(), getDate()(trả về giá trị ngày, tháng, năm), getTime()(trả về giá trị giờ và phút), getCurrentDateTime()(trả về thời gian hiện tại), getTM()(trả về thông tin thời gian dưới định dạng chuẩn).

Các phương thức kiểm tra định dạng:

* Hàm isLeapYear(): hàm kiểm tra năm hiện tại có phải là năm nhuận hay không.
* Hàm isValid DateTime(): hàm kiểm tra định dạng đúng của thời gian.



Hình 4. File quản lý định dạng thời gian

### Lớp TicketInfo

*Thuộc tính private:*

* + departureCity và destinationCity: điểm xuất phát và điểm đến của chuyến đi.
  + isRoundTrip: Biến bool xác định hành khách có muốn đặt vé chuyến bay khứ hồi hay không.
  + departureDate và returnDate: Ngày xuất phát và ngày trở về( nếu có đặt vé khứ hồi).
  + selectedOutboundFlight và selectedReturnFlight: Đối tượng lớp Flight đại diện cho chuyến đi và chuyến về(nếu khứ hồi).
  + selectedOutboundSeat và selectedReturnSeat: được chọn cho chuyến đi và chuyến về(nếu khứ hồi).
  + amount: Tổng số lượng vé mà hành khách muốn đặt trong 1 chuyến bay(bao gồm cả vé người lớn và vé trẻ em).
  + price: Giá tiền của một vé trong chuyến bay.

*Thuộc tính public:*

Hàm tạo và hàm hủy:

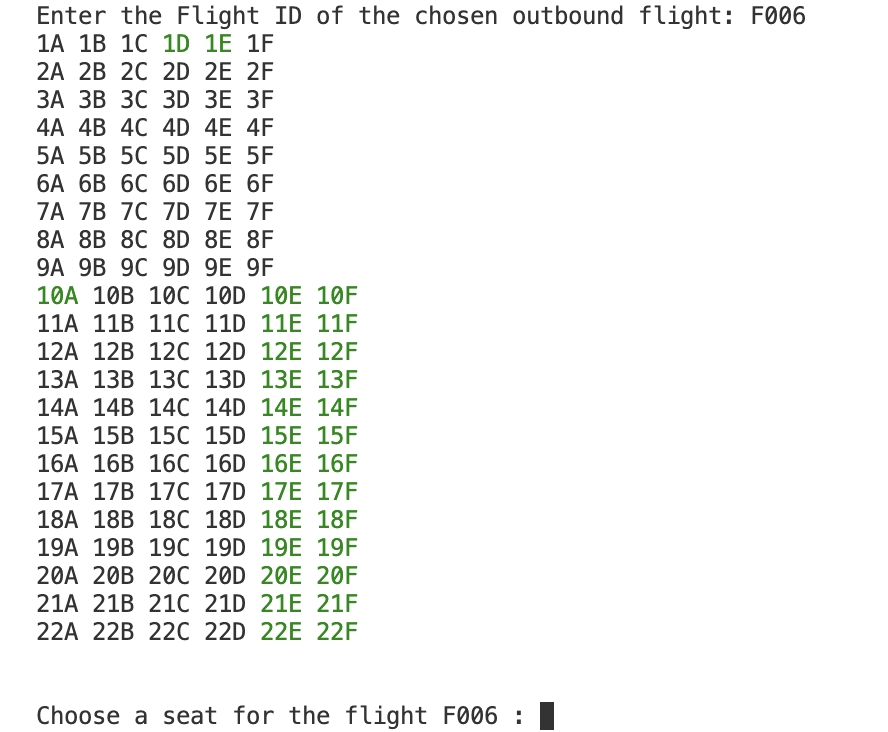
* + TicketInfo(): Hàm tạo mặc định.
  + ~TicketInfo(): Hàm hủy.

Hàm getter:

* + Các hàm getter để lấy thông tin về các thuộc tính như chuyến đi, chuyến về, số lượng ghế đã đặt, giá vé.

Các phương thức quản lý thông tin vé:

* + displayOptions(): Hiển thị các tùy chọn cho người dùng.
  + chooseFlight(): Cho phép người dùng chọn chuyến bay.
  + getLineByFlightID(const string&): Tìm và trả về dòng chứa thông tin các ghế đã được đặt trong file flight\_seats.txt dựa trên ID của chuyến bay.
  + printSeatMapForFlight(const Flight& flight): Hiển thị sơ đồ ghế cho chuyến bay được chọn( với màu xanh là ghế đã có hành khách lựa chọn, màu trắng đối với ghế còn trống).
  + isSeatInRange(const string &seat, int seatCount): kiểm tra ghế được chọn có phù hợp với chuyến bay hay không.



Hình 5. Hiển thị sơ đồ ghế của chuyến bay

* + chooseSeats() và chooseSeatsForFlight(const Flight &flight, string \*seconstats): Cho phép người dùng chọn ghế.
  + updateAvailableSeats(): Cập nhật số lượng ghế còn trống sau khi có vé được đặt.
  + saveTicketInfoToFile(const std::string &filename) const: Lưu thông tin vé vào tệp.
  + readTicketInfoFromFile(const std::string &filename): Đọc thông tin vé từ tệp.
  + calculatePrice(): Tính toán tổng giá tiền cho các vé đã chọn.



Hình 6. File quản lý đặt vé cho chuyến bay

### Lớp CustomerInfo

*Thuộc tính protected:*

* Name, Phone, Email, YearBirth: các thuộc tính thể hiện thông tin của khách hàng.
* \*nameList: mảng con trỏ được cấp phát động để lưu trữ thuộc tính Tên khách hàng.

*Thuộc tính public:*

Hàm tạo và hàm huỷ:

* Customer(): hàm dựng mặc định
* ~Customer(): hàm huỷ

Hàm getter và setter:

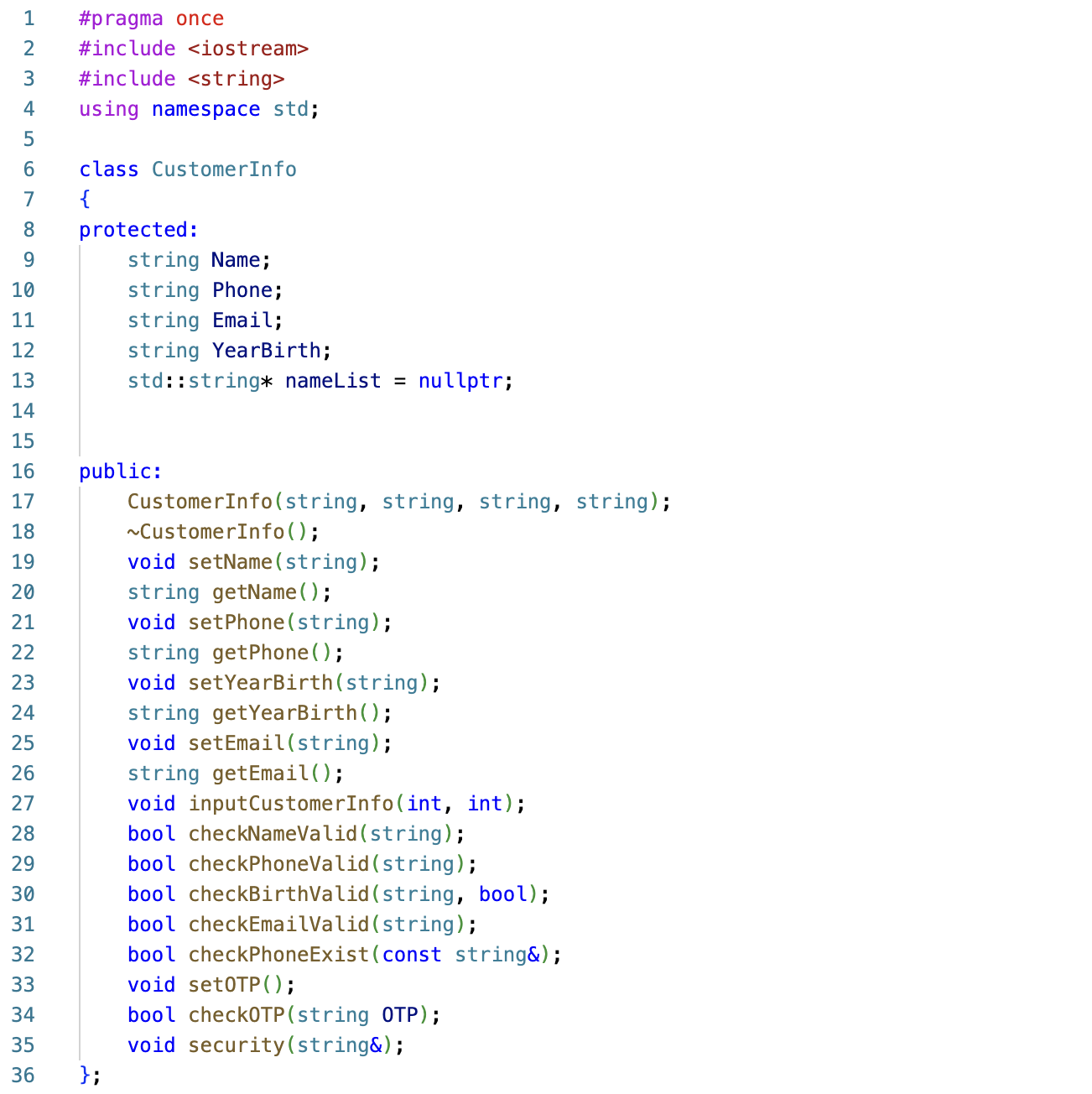
* getName(), getPhone(), getEmail(), getYearBirth(): lấy các thuộc tính thể hiện thông tin khách hàng.
* setName(string), setPhone(string), setEmail(string), setYearBirth(string): gán giá trị các thuộc tính thể hiện thông tin khách hàng.
* setOTP(): tạo mã OTP để tăng tính bảo mật cho thông tin khách hàng.

Các phương thức kiểm tra định dạng:

* checkNameValid(string), checkPhoneValid(string), checkBirthValid(string), checkEmailValid(string), checkOTP(string OTP): các hàm kiểm tra thông tin nhập vào đã đúng định dạng hay chưa.
* inputCustomerInfo(int): xử lý các thông tin nhập sai định dạng để thu được thông tin cuối cùng trước khi lưu dữ liệu vào file.

Phương thức lưu dữ liệu vào file:

* saveToTxtFile(): lưu thông tin khách hàng đã nhập vào file text(đã được xử lý định dạng).



Hình 7. File quản lý thông tin khách hàng

### Lớp eTicket

Lớp eTicket kế thừa 2 lớp là CustomerInfo và TicketInfo.

*Thuộc tính private:*

* TicketID: thuộc tính thể hiện mã vé điện tử.

*Thuộc tính public:*

Hàm tạo:

* eTicket(const CustomerInfo&, const TicketInfo&): hàm tạo có tham số.

Các phương thức thành viên:

* TicketToFile(int): hàm lưu thông tin vé vào file text.
* Display(): hàm hiển thị các vé điện tử ra màn hình.
* DisplayTicket(const& string): hàm định dạng thông tin hiển thị của 1 vé điện tử.
* checkAccValid(string): hàm kiểm tra định dạng của số thẻ ngân hàng.
* Paymen(): hàm thực hiện chức năng thanh toán.
* ticketList(int): lưu tất cả các vé đã đặt vào file text.
* FindTicket(const string&): tìm vé thông qua ticket ID.

A screen shot of a computer program

Description automatically generated

Hình 8. File quản lý xuất vé điện tử

## Kết quả

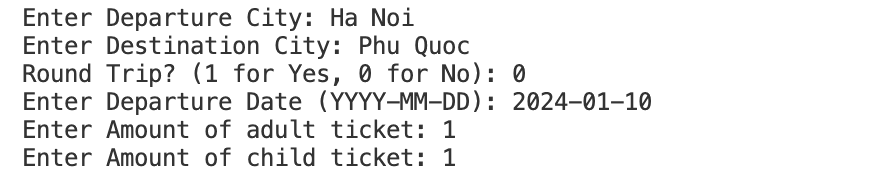
### Giao diện chính của chương trình

* Giao diện menu chính:A close-up of a white background

  Description automatically generated

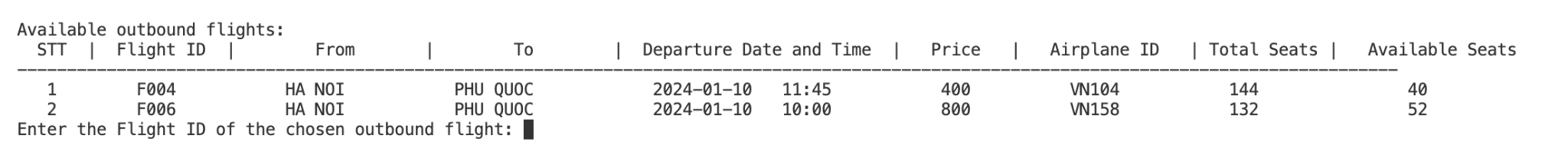
Hình 9. Giao diện menu chính

* Giao diện nhập thông tin chuyến bay:



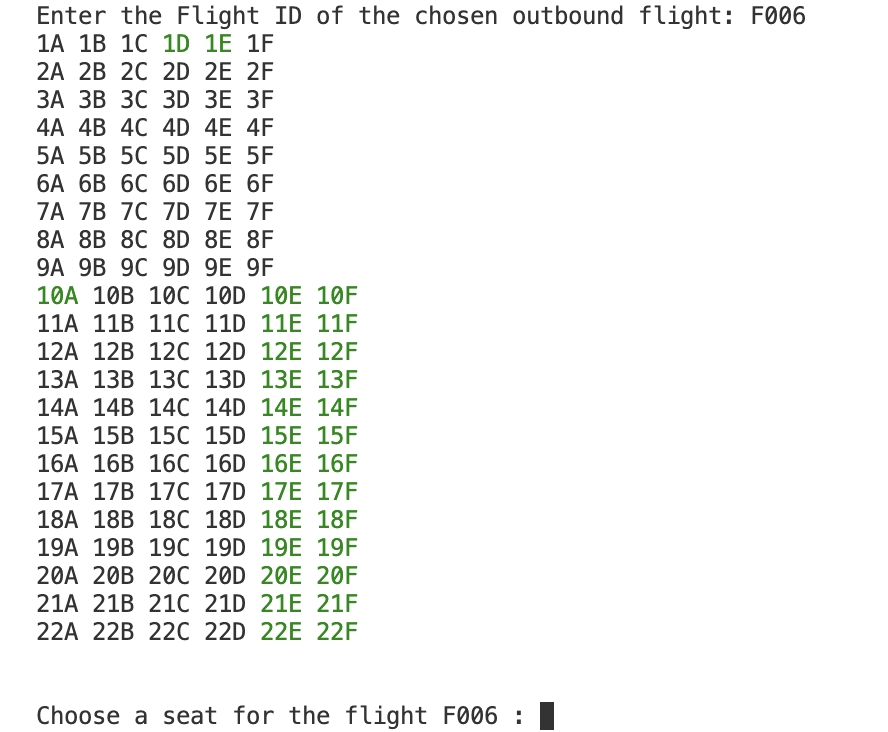
Hình 10. Giao diện nhập thông tin chuyến bay

* Giao diện chọn chuyến bay:



Hình 11. Giao diện chọn chuyến bay

* Giao diện chọn vị trí ghế ngồi:



Hình 12. Giao diện chọn vị trí ghế ngồi

* Giao diện nhập thông tin khách hàng:

A close up of a website

Description automatically generated

Hình 13. Giao diện nhập thông tin khách hàng

* Giao diện thanh toán:

A screenshot of a computer

Description automatically generated

Hình 14. Giao diện thanh toán

### Kết quả thực thi của chương trình:

* Khi truy cập ứng dụng, người dùng sẽ lựa chọn các chức năng do chương trình chung cấp thông qua menu:

A close-up of a white background

Description automatically generated

Hình 15. Hiển thị menu

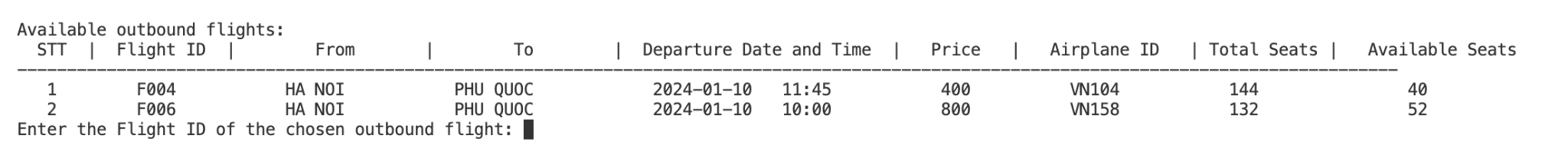
* Tiếp theo, người dùng sẽ nhập các lựa chọn với chuyến bay mà khách hàng mong muốn, bao gồm: nơi đi, nơi đến, ngày đi, khứ hồi hoặc không khứ hồi, ngày về (nếu có khứ hồi), số lượng vé. Dưới đây là minh hoạ:

A black text on a white background

Description automatically generated

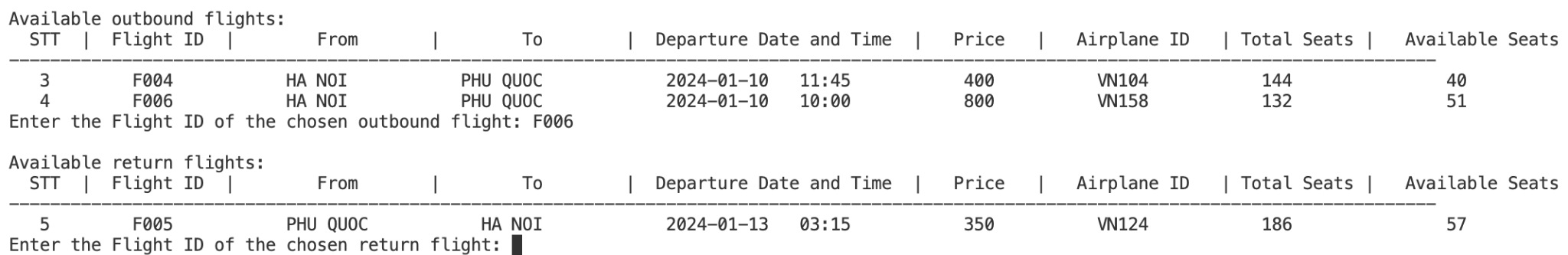
Hình 16. Hiển thị yêu cầu nhập thông tin chuyến bay

* Sau khi đã có các thông tin về chuyến bay mong muốn, ứng dụng sẽ đọc dữ liệu từ file text và hiển thị ra màn hình danh sách những chuyến bay phù hợp để khách hàng lựa chọn:



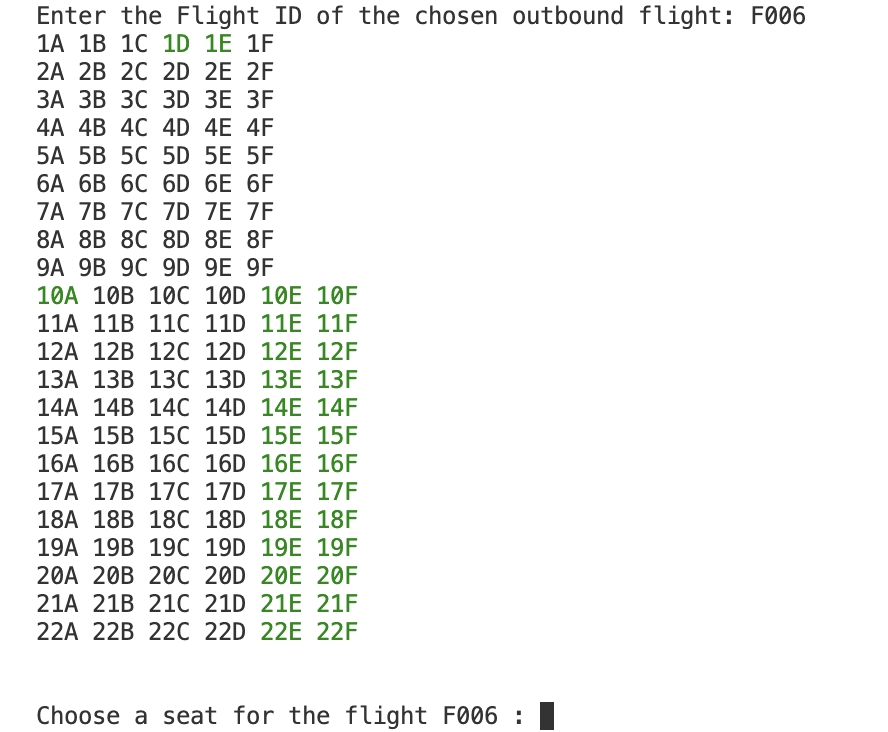
Hình 17. Hiển thị danh sách chuyến bay lượt đi phù hợp với yêu cầu người dùng

* Nếu là chuyến bay khứ hồi, chương trình sẽ tiếp tục hiển thị danh sách các chuyến bay lượt về phù hợp với nhu cầu để khách hàng tiếp tục lựa chọn:



Hình 18. Hiển thị danh sách chuyến bay lượt về phù hợp với yêu cầu người dùng

* Khi đã chọn xong chuyến bay lượt đi và lượt về (nếu có), ứng dụng sẽ hiển thị sơ đồ chỗ ngồi để khách hàng lựa chọn vị trí ghế thích hợp:



Hình 19. Hiển thị sơ đồ ghế ngồi của chuyến bay

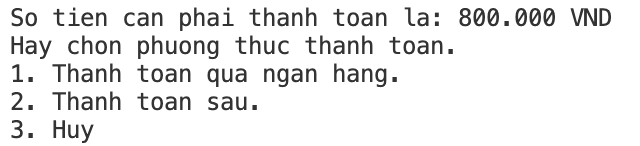
* Bước tiếp theo sau khi đã chọn xong chuyến bay và ghế ngồi, khách hàng sẽ phải nhập thông tin cá nhân, bao gồm: số điện thoại, mã OTP (chuyển các ký tự được nhập vào thành \* để bảo vệ thông tin người dùng), tên, địa chỉ email và năm sinh. Đối với trẻ em, chỉ cần nhập tên và năm sinh. Nếu có thông tin nào nhập sai định dạng thì chương trình sẽ yêu cầu nhập lại:

A close up of a website

Description automatically generated

Hình 20. Giao diện yêu cầu người dùng nhập thông tin cá nhân

* Khi khách hàng hoàn thành bước nhập thông tin cá nhân, ứng dụng sẽ tính toán số tiền mà khách hàng cần phải trả, khách hàng có thể lựa chọn thanh toán trực tuyến thông qua ngân hàng hoặc thanh toán sau:



Hình 21. Hiển thị số tiền cần thanh toán và các phương thức thanh toán

* Nếu khách hàng chọn thanh toán trực tuyến qua ngân hàng, ứng dụng sẽ chuyển tới bước nhập lựa chọn ngân hàng, nhập thông tin thẻ và xác nhận thanh toán, và khi khách hàng chọn thanh toán sau, ứng dụng sẽ xác nhận thông tin này và chuyển đến bước tiếp theo:

***A screenshot of a computer

Description automatically generated***

Hình 22. Giao diện thanh toán trực tuyến và xác nhận thanh toán

***A white background with black text

Description automatically generated***

Hình 23. Giao diện thanh toán sau và xác nhận đặt vé

* Cuối cùng, ứng dụng sẽ hiển thị ra màn hình vé điện tử mà khách hàng đã đặt. Mỗi vé có 2 phần chính: phần mã vé là duy nhất đối với mỗi vé, phần nội dung vé sẽ bao gồm tên khách hàng và thông tin chuyến bay mà khách hàng đã đặt. Đến đây, khách hàng đã đặt thành công vé máy bay thông qua ứng dụng:

A close up of a ticket

Description automatically generated

Hình 24. Hiển thị vé điện tử

### Nhận xét

Chương trình quản lý đặt vé máy bay trên đã đảm bảo cấu trúc hướng đối tượng và có khả năng quản lý thông tin của các đối tượng khác nhau, tuy nhiên vẫn còn các thiếu sót về mặt logic và triển khai giao diện người dùng.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

Trong thời gian thực hiện đồ án, nhóm chúng em đã có nhiều cơ hội tiếp cận những kiến thức mới và củng cố kiến thức cũ, đồng thời là sự trau dồi các kỹ năng khác như làm việc nhóm, thuyết trình, phân bổ thời gian thực hiện kế hoạch một cách cụ thể. Tuy nhiên sản phẩm của chúng em vẫn còn nhiều thiếu sót, rất mong được thầy cô đóng góp ý kiến để có thể phát triển hơn.

## Hướng phát triển

* Tiếp thu các ý kiến đóng góp để nội dung đồ án ngày càng hoàn thiện.
* Nghiên cứu và học hỏi nhiều hơn các kiến thức mới, sử dụng linh hoạt những điều đã học vào thực hiện đồ án.
* Xây dựng một ứng dụng hoàn chỉnh, tối ưu nhất. Có thể ứng dụng vào thực tiễn.

TÀI LIỆU THAM KHẢO

## [1] TS. Đặng Hoài Phương. Giáo trình *Lập trình hướng đối tượng*, trường Đại học Bách Khoa – ĐHĐN.

## [2] Mark Allen Weiss (1994). *Data Structures and Algorithm Analysis in C,* Pearson Education.