TRƯỜNG ĐẠI HỌC BÌNH DƯƠNG

**KHOA CÔNG NGHỆ THÔNG TIN, ROBOT VÀ TRÍ TUỆ NHÂN TẠO**



**BÁO CÁO TIỂU LUẬN**

**CHUYÊN ĐỀ 1**

**ĐỀ TÀI: XÂY DỰNG HỆ THỐNG ĐẶT VÉ MÁY BAY BDUAIR**

| **Giảng viên hướng dẫn:** | Nguyễn Thanh Sơn  Hồ Ngọc Giàu  Dương Anh Tuấn  Nguyễn Hữu Quyền |
| --- | --- |
| **Sinh viên thực hiện:** | Tăng Đức Tiền - 20050034  Phạm Hoàng Phương- 22050042 |

**Bình Dương, năm 2025**

## NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

**…………………………………………………………………………………………...…………………………………………………………………………………………...…………………………………………………………………………………………...…………………………………………………………………………………………...…………………………………………………………………………………………...…………………………………………………………………………………………...…………………………………………………………………………………………...…………………………………………………………………………………………...…………………………………………………………………………………………...…………………………………………………………………………………………...…………………………………………………………………………………………...…………………………………………………………………………………………...…………………………………………………………………………………………...…………………………………………………………………………………………...…………………………………………………………………………………………...…………………………………………………………………………………………...…………………………………………………………………………………………...…………………………………………………………………………………………...…………………………………………………………………………………………...…………………………………………………………………………………………...…………………………………………………………………………………………...…………………………………………………………………………………………...…………………………………………………………………………………………...…………………………………………………………………………………………...**

## LỜI CẢM ƠN

Trước tiên, nhóm chúng em xin gửi lời cảm ơn chân thành đến các thầy/cô đã tận tình hướng dẫn và hỗ trợ trong suốt quá trình học tập và thực hiện tiểu luận môn Chuyên đề 1.

Sự giảng dạy nhiệt tình, những tài liệu quý báu cùng với các góp ý sâu sắc từ các thầy/cô đã giúp nhóm chúng em định hướng rõ ràng, nâng cao kiến thức và hoàn thiện bài tiểu luận một cách hiệu quả nhất.

Nhóm chúng em xin kính chúc quý thầy/cô luôn dồi dào sức khỏe, hạnh phúc và tiếp tục gặt hái nhiều thành công trong sự nghiệp giảng dạy và nghiên cứu.

Nhóm xin chân thành cảm ơn và mong sẽ còn có cơ hội được tiếp tục học hỏi từ các thầy/cô trong tương lai.

## MỤC LỤC

[**DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT 1**](#_heading=h.tyjcwt)

[**DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ 2**](#_heading=h.3dy6vkm)

[**PHẦN 1 – GIỚI THIỆU TỔNG QUAN ĐỀ TÀI 3**](#_heading=h.lnxbz9)

[1.1 Giới thiệu đề tài 3](#_heading=h.2tevje7v6m24)

[1.2 Đối tượng nghiên cứu 4](#_heading=h.eakav0bmzqpc)

[1.3 Phạm vi nghiên cứu 4](#_heading=h.di7fdb2habxf)

[1.4 Mục tiêu nghiên cứu 5](#_heading=h.m5w7avrgc5j1)

[1.5 Kết quả cần đạt 5](#_heading=h.bgm83yoruxzt)

[**PHẦN 2 – HIỆN TRẠNG VÀ PHÂN TÍCH YÊU CẦU 6**](#_heading=h.2jxsxqh)

[2.1 Hiện trạng đề tài 6](#_heading=h.s8pjjwwwd7oz)

[2.2 Các yêu cầu chức năng của đề tài 7](#_heading=h.w7m7fwmhgdru)

[2.3 Các yêu cầu phi chức năng của đề tài 8](#_heading=h.tvh7ht8d9wv2)

[**PHẦN 3 – MÔ HÌNH HÓA HỆ THỐNG 9**](#_heading=h.dv49w1ljbxl8)

[3.1 Mô hình hóa Use Case 9](#_heading=h.gzhf8ts6x0p0)

[3.1.1 Mô hình Use Case tổng quát 9](#_heading=h.6heea768m42b)

[3.1.2 Đặc tả use case 12](#_heading=h.sa9rxpxodsvb)

[3.1.2.1 Use case “DangKiTaiKhoan” 12](#_heading=h.2x1er73917vz)

[3.1.2.2 Use case “DangNhap” 14](#_heading=h.se6cquw6somj)

[3.1.2.3 Use case “QuanLiThongTinNguoiDung” 16](#_heading=h.gubs4i5vr99h)

[3.1.2.4 Use case “TimKiemChuyenBay” 19](#_heading=h.1zyh0hl3zook)

[3.1.2.5 Use case “XemChiTietChuyenBay” 21](#_heading=h.t2lomdd6ceab)

[3.1.2.6 Use case “ChonChuyenBay” 23](#_heading=h.3bhg3dxyilj)

[3.1.2.7 Use case “ChonHangGhe” 25](#_heading=h.cv2lyumrats2)

[3.1.2.8 Use case “DatVe” 26](#_heading=h.patcjsoz2zp)

[3.1.2.9 Use case “ThanhToan” 28](#_heading=h.7cj2rdoqttee)

[3.1.2.10 Use case “QuanLiMayBay” 30](#_heading=h.esex2qcg5hq8)

[3.1.2.11 Use case “QuanLiChuyenBay” 32](#_heading=h.qchlcdf1239i)

[3.1.2.12 Use case “QuanLiGheNgoi” 35](#_heading=h.pfuwz87j9pom)

[3.2 Mô hình hóa cấu trúc 38](#_heading=h.nld1j7ttir5s)

[3.2.1 Sơ đồ class 38](#_heading=h.hdqawygziyld)

[3.2.2 Sơ đồ thực thể ERD 41](#_heading=h.6yxmd4b1a8k7)

[3.3 Mô hình hóa hành vi 44](#_heading=h.r2tqt0hlz25o)

[3.3.1 Sơ đồ tuần tự 44](#_heading=h.r7vkk5ddw2mr)

[3.3.1.1 Quá trình đăng ký tài khoản của khách hàng 44](#_heading=h.5btk2mwaggx7)

[3.3.1.2 Quá trình đăng nhập của khách hàng 45](#_heading=h.14au6yyguee1)

[3.3.1.3 Quá trình thêm thông tin của khách hàng 46](#_heading=h.dd753xfa5xor)

[3.3.1.4 Quá trình sửa thông tin của khách hàng 47](#_heading=h.6jae1ru7cj49)

[3.3.1.5 Quá trình tìm kiếm chuyến bay của khách hàng 48](#_heading=h.y0clnvjdyqlr)

[3.3.1.6 Quá trình xem chi tiết chuyến bay của khách hàng 49](#_heading=h.7fbj8sv67hnf)

[3.3.1.7 Quá trình chọn chuyến bay của khách hàng 50](#_heading=h.tm0x2eooumdq)

[3.3.1.8 Quá trình chọn hạng ghế của khách hàng 51](#_heading=h.i8p713tduxj0)

[3.3.1.9 Quá trình chọn đặt vé của khách hàng 53](#_heading=h.xy76x4wnaybm)

[3.3.1.10 Quá trình thanh toán của khách hàng 55](#_heading=h.ab9sk5r7i5bp)

[3.3.1.11 Quá trình hủy vé của khách hàng 57](#_heading=h.ch620rt2mufe)

[**PHẦN 4 – THIẾT KẾ HỆ THỐNG 59**](#_heading=h.2xcytpi)

[4.1 Thiết kế cơ sở dữ liệu 59](#_heading=h.6zuojlbvvcxm)

[4.2 Thiết kế giao diện người dùng 60](#_heading=h.pc45yrv67mu2)

[4.3 Thiết kế backend 61](#_heading=h.7kf2krynvuug)

[**PHẦN 5 – TRIỂN KHAI HỆ THỐNG 62**](#_heading=h.j6bchohsz1vn)

[5.1 Đóng gói hệ thống bằng Docker 62](#_heading=h.9echdy1w9zg7)

[5.2 Cấu hình hạ tầng mạng (VPC) 62](#_heading=h.uv5178e48k7d)

[5.3 Cấu hình cơ sở dữ liệu 63](#_heading=h.y7tj4urejx32)

[5.4 Cài đặt và triển khai trên EC2 63](#_heading=h.8zq0rnivgfs9)

[**PHẦN 6 – KẾT QUẢ ĐẠT ĐƯỢC VÀ HƯỚNG PHÁT TRIỂN 64**](#_heading=h.4w863h9wecpf)

[6.1 Kết quả đạt được 64](#_heading=h.po44l2vjo6ls)

[6.2 Hướng phát triển 75](#_heading=h.ramd0osmxcp8)

[**TÀI LIỆU THAM KHẢO 76**](#_heading=h.1ci93xb)

## DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT

**CÁC CHỮ VIẾT TẮT**

2FA Two-Factor Authentication

CSDL Cơ sở dữ liệu

UML Unified Modeling Language

ERD Entity Relationship Diagram

## DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ

**DANH MỤC HÌNH**

Hình 3.1 Mô hình use case tổng quát của hệ thống đặt vé máy bay BDUAir.  10

Hình 3.2 Use case “DangKiTaiKhoan”.  13

Hình 3.3 Use case “DangNhap”.  15

Hình 3.4 Use case “QuanLiThongTinNguoiDung”.  16

Hình 3.5 Use case “TimKiemChuyenBay”.  19

Hình 3.6 Use case “XemChiTietChuyenBay”.  21

Hình 3.7 Use case “ChonChuyenBay”.  23

Hình 3.8 Use case “ChonHangGhe”.  25

Hình 3.9 Use case “DatVe”.  26

Hình 3.10 Use case “ThanhToan”.  28

Hình 3.11 Use case “QuanLiMayBay”.  30

Hình 3.12 Use case “QuanLiChuyenBay”.  32

Hình 3.13 Use case “QuanLiGheNgoi”.  35

Hình 3.14 Sơ đồ class của hệ thống đặt vé máy bay BDUAir.  38

Hình 3.15 Sơ đồ thực thể ERD của hệ thống đặt vé máy bay BDUAir.  41

Hình 3.16 Sơ đồ tuần tự quá trình đăng ký tài khoản của khách hàng.  44

Hình 3.17 Sơ đồ tuần tự quá trình đăng nhập của khách hàng.  45

Hình 3.18 Sơ đồ tuần tự quá trình thêm thông tin của khách hàng.  46

Hình 3.19 Sơ đồ tuần tự quá trình sửa thông tin của khách hàng.  47

Hình 3.20 Sơ đồ tuần tự quá trình tìm kiếm chuyến bay của khách hàng.  48

Hình 3.21 Sơ đồ tuần tự quá trình xem chi tiết chuyến bay của khách hàng.  49

Hình 3.22 Sơ đồ tuần tự quá trình chọn chuyến bay của khách hàng.  50

**DANH MỤC BẢNG**

Bảng 2.1 Chức năng nghiệp vụ của hệ thống.  8

Bảng 3.1 Danh sách Actor.  10

Bảng 3.2 Danh sách Use Case.  11

## PHẦN 1 – GIỚI THIỆU TỔNG QUAN ĐỀ TÀI

### 1.1 Giới thiệu đề tài

“Hệ thống đặt vé máy bay BDUAir” là một website trực tuyến cho phép người dùng dễ dàng tìm kiếm, đặt chỗ và thanh toán cho các chuyến bay, đảm bảo độ chính xác, an toàn và hiệu quả.

Hệ thống quản lý các chuyến bay từ nhiều hãng hàng không. Mỗi chuyến bay gồm thông tin như mã chuyến, hãng bay, điểm đi, điểm đến, thời gian khởi hành, đến nơi, giá vé, số ghế còn trống và hạng vé. Mỗi chuyến bay thuộc về một hãng hàng không, với mã và tên hãng tương ứng.

Người dùng có thể tìm chuyến bay theo ngày đi, điểm xuất phát, điểm đến và hạng vé. Sau khi đăng ký và đăng nhập, người dùng được phép đặt vé bằng cách cung cấp các thông tin cá nhân cần thiết. Sau khi đặt, hệ thống sẽ tạo đơn vé gồm mã đơn, thông tin người dùng, chuyến bay, số lượng vé, tổng tiền, trạng thái và thời gian đặt.

Đơn đặt vé có các trạng thái như: "Chờ thanh toán", "Đã thanh toán", "Đã huỷ",... Hệ thống cũng quản lý quyền người dùng (ví dụ: người dùng thường, quản trị viên) với thông tin về mã và tên quyền.

Ngoài ra, website cung cấp các tin tức liên quan đến hàng không như khuyến mãi, thông báo, cẩm nang,... Mỗi tin tức gồm mã, tiêu đề, hình ảnh, ngày đăng, tóm tắt, nội dung và thuộc một loại tin cụ thể.

Cuối cùng, hệ thống còn có chức năng hỗ trợ trực tuyến, giúp người dùng gửi thắc mắc và nhận tư vấn nhanh chóng.

### 1.2 Đối tượng nghiên cứu

Vì đề tài tập trung vào việc xây dựng một hệ thống đặt vé máy bay trực tuyến, các đối tượng nghiên cứu chính bao gồm:

* Người dùng hệ thống: Là hành khách có nhu cầu tìm kiếm, đặt vé và thanh toán cho các chuyến bay.
* Chuyến bay và hãng hàng không: Bao gồm các thông tin liên quan đến chuyến bay (mã chuyến, điểm đi, điểm đến, thời gian, giá vé, hạng vé...) và các hãng hàng không cung cấp dịch vụ. Đây là phần dữ liệu cốt lõi của hệ thống.
* Quy trình đặt vé và thanh toán: Bao gồm các bước người dùng thực hiện từ tìm kiếm chuyến bay, lựa chọn vé, nhập thông tin cá nhân đến thanh toán và nhận xác nhận. Việc nghiên cứu quy trình này giúp đảm bảo hệ thống hoạt động chính xác, an toàn và hiệu quả.
* Quản lý và phân quyền người dùng: Nghiên cứu cách tổ chức và kiểm soát quyền truy cập trong hệ thống, phân biệt các người dùng để bảo mật và dễ quản lý.

### 1.3 Phạm vi nghiên cứu

Phạm vi nghiên cứu của đề tài bao gồm:

* Phạm vi dữ liệu: Dữ liệu được sử dụng trong hệ thống bao gồm thông tin về các chuyến bay, hãng hàng không, người dùng, đơn đặt vé, thanh toán và tin tức.
* Phạm vi nội dung: Đề tài tập trung vào xây dựng một hệ thống đặt vé máy bay trực tuyến với các chức năng chính: tìm kiếm chuyến bay, đặt vé, thanh toán, quản lý tài khoản người dùng, phân quyền truy cập và hỗ trợ thông tin.
* Phạm vi công nghệ: Hệ thống được phát triển bằng Nodejs Express, sử dụng các framework và thư viện phù hợp để xử lý backend và tương tác với cơ sở dữ liệu. Giao diện người dùng được thiết kế đơn giản, thân thiện. Toàn bộ hệ thống được triển khai trên nền tảng AWS Cloud, nhằm đảm bảo khả năng mở rộng, độ tin cậy và hiệu suất hoạt động.

### 

### 1.4 Mục tiêu nghiên cứu

Đối với đề tài này, các mục tiêu chính sẽ được tiến hành nghiên cứu, tìm hiểu và giải quyết bao gồm:

* Phát triển giao diện người dùng thân thiện: Thiết kế một giao diện trực quan, dễ sử dụng, phù hợp với cả người dùng phổ thông không am hiểu công nghệ, nhằm giúp quá trình tìm kiếm và đặt vé diễn ra thuận tiện và nhanh chóng.
* Xây dựng hệ thống API Backend hiệu quả: Phát triển các API xử lý các chức năng chính như tìm kiếm chuyến bay, đặt vé và thanh toán. Hệ thống API cần đảm bảo khả năng xử lý nhanh chóng, ổn định.
* Thiết kế cơ sở dữ liệu rõ ràng và toàn vẹn: Xây dựng cơ sở dữ liệu với cấu trúc hợp lý để lưu trữ các thông tin liên quan đến chuyến bay, người dùng, đơn đặt vé và giao dịch.
* Triển khai hệ thống trên nền tảng điện toán đám mây: Sử dụng dịch vụ AWS Cloud để triển khai hệ thống, đảm bảo tính mở rộng và độ tin cậy.

### 1.5 Kết quả cần đạt

Kết quả cần đạt được đối với đề tài này là:

* Một hệ thống API backend hoàn chỉnh: Đảm bảo khả năng xử lý hiệu quả các chức năng như tìm kiếm, đặt vé, quản lý người dùng và thanh toán; hoạt động tốt kể cả khi có nhiều yêu cầu đồng thời.
* Một cơ sở dữ liệu được thiết kế rõ ràng, logic: Đáp ứng yêu cầu lưu trữ và truy xuất thông tin về chuyến bay, hãng bay, người dùng, đơn đặt vé và trạng thái giao dịch một cách nhất quán và bảo mật.
* Giao diện người dùng thân thiện: Giao diện web trực quan, dễ thao tác với trải nghiệm người dùng tốt, hỗ trợ cả trên máy tính và thiết bị di động.
* Triển khai thành công trên nền tảng AWS Cloud: Hệ thống có thể truy cập từ xa qua Internet, đảm bảo tính sẵn sàng, hiệu suất và khả năng mở rộng.

## PHẦN 2 – HIỆN TRẠNG VÀ PHÂN TÍCH YÊU CẦU

### 

### 2.1 Hiện trạng đề tài

Ngày nay, với sự phát triển nhanh chóng của công nghệ thông tin và Internet, nhu cầu di chuyển bằng đường hàng không ngày càng tăng cao. Người dùng mong muốn có thể tìm kiếm, so sánh và đặt vé máy bay một cách nhanh chóng, tiện lợi mà không cần phải đến trực tiếp các đại lý vé truyền thống. Chính vì thế, các hệ thống đặt vé máy bay trực tuyến đã trở thành một giải pháp phổ biến, giúp tiết kiệm thời gian và công sức cho người dùng.

Tuy nhiên, ở Việt Nam, nhiều website đặt vé vẫn còn hạn chế về mặt trải nghiệm người dùng, tốc độ xử lý chậm, giao diện phức tạp hoặc thiếu tính minh bạch về thông tin chuyến bay và giá vé. Ngoài ra, một số hệ thống chưa tối ưu tốt cho việc xử lý nhiều lượt đặt chỗ đồng thời hoặc không đảm bảo an toàn thông tin người dùng trong quá trình thanh toán.

Xuất phát từ nhu cầu thực tiễn đó, nhóm chúng em đã lựa chọn đề tài “Hệ thống đặt vé máy bay BDUAir” nhằm mang đến một giải pháp hiệu quả, tiện lợi và an toàn cho người dùng. Website không chỉ hỗ trợ tra cứu chuyến bay, đặt vé, thanh toán, mà còn tích hợp các tính năng như quản lý thông tin cá nhân, theo dõi lịch sử giao dịch và cập nhật tin tức liên quan đến hàng không.

Với giao diện thân thiện, dễ sử dụng cùng hệ thống xử lý nhanh chóng và đáng tin cậy, đề tài hướng đến việc nâng cao trải nghiệm người dùng cũng như góp phần thúc đẩy quá trình số hóa dịch vụ đặt vé máy bay tại Việt Nam.

### 2.2 Các yêu cầu chức năng của đề tài

#### 

| **STT** | **Công việc** | **Chức năng** |
| --- | --- | --- |
| 1 | Quản lý chuyến bay | Lưu trữ |
| 2 | Quản lý hãng hàng không | Lưu trữ |
| 3 | Quản lý người dùng | Lưu trữ |
| 4 | Quản lý đơn đặt vé | Lưu trữ |
| 5 | Quản lý trạng thái đặt vé | Lưu trữ |
| 6 | Quản lý quyền truy cập | Lưu trữ |
| 7 | Quản lý tin tức | Lưu trữ |
| 8 | Đăng nhập | Lưu trữ |
| 9 | Đăng ký | Lưu trữ |
| 10 | Tìm kiếm chuyến bay | Tra cứu |

*Bảng 2.1 Chức năng nghiệp vụ của hệ thống.*

Chức năng nghiệp vụ chính của hệ thống được chia thành hai nhóm chức năng cơ bản là chức năng lưu trữ và chức năng tra cứu:

* Chức năng lưu trữ: Lưu trữ và cập nhật thông tin khi có thao tác thêm, sửa, xóa dữ liệu từ người dùng hoặc admin, bao gồm các nghiệp vụ như quản lý chuyến bay, người dùng, hãng hàng không, đơn đặt vé, tin tức, quyền truy cập, v.v.
* Chức năng tra cứu: Cho phép người dùng tìm kiếm chuyến bay theo các tiêu chí như điểm đi, điểm đến, ngày khởi hành và hãng bay. Ngoài ra, admin có thể tra cứu thông tin người dùng, đơn đặt vé và các hãng hàng không nhằm phục vụ cho công tác quản trị hệ thống.

### 

### 2.3 Các yêu cầu phi chức năng của đề tài

#### 

Các yêu cầu phi chức năng của hệ thống bao gồm:

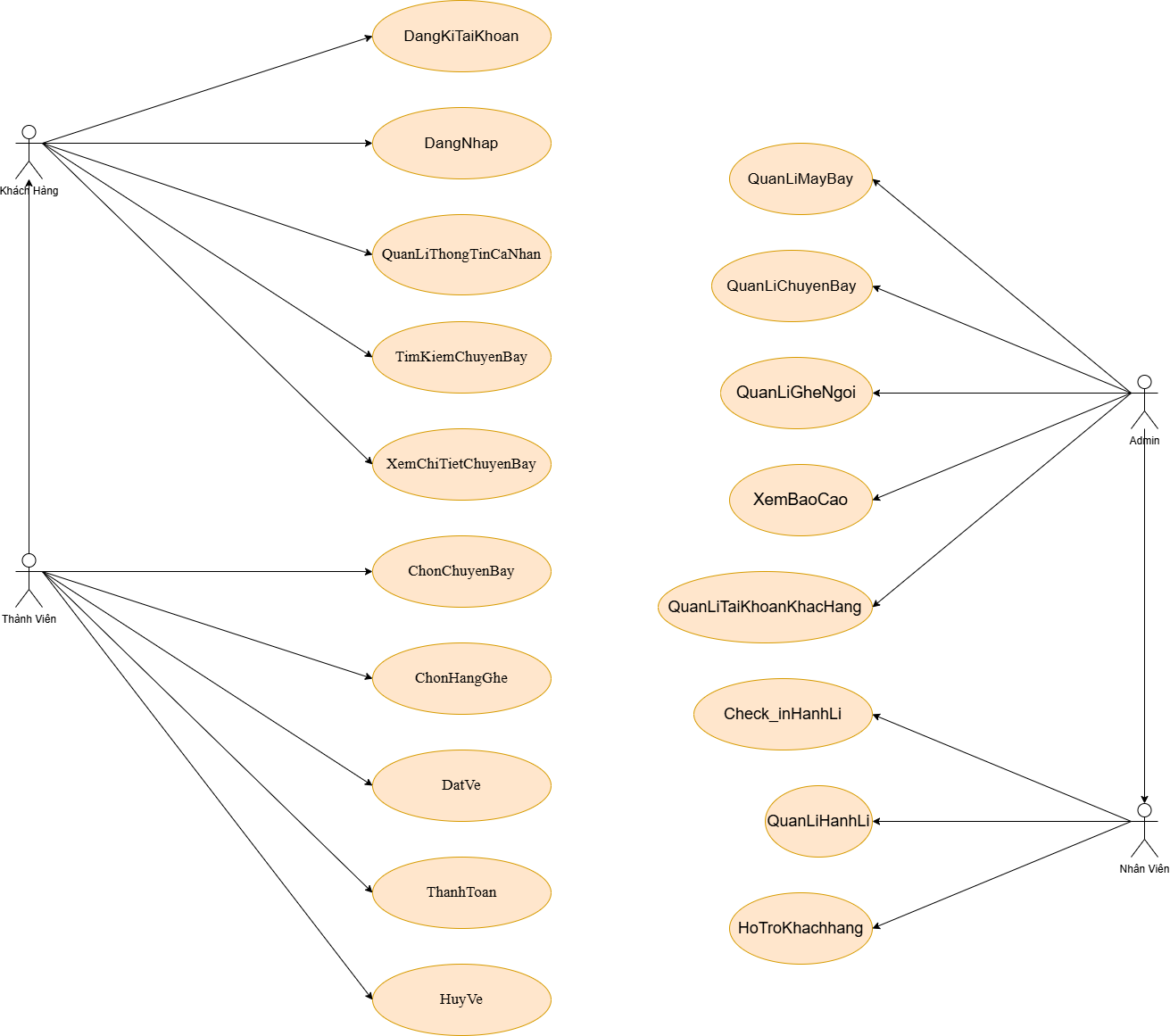
* Tính tiến hóa và khả năng mở rộng:
* Hệ thống cần được thiết kế linh hoạt để có thể mở rộng, chỉnh sửa và cập nhật dễ dàng theo yêu cầu phát triển sau này, như thêm các tính năng mới (ví dụ: tích hợp ví điện tử, chương trình tích điểm, v.v).
* Hiệu suất và tính ổn định:
* Website cần chạy ổn định và phản hồi nhanh, kể cả khi có nhiều người dùng truy cập cùng lúc. Mục tiêu là giữ thời gian phản hồi trung bình dưới 500 mili-giây.
* Hệ thống nên đảm bảo hoạt động tốt với ít nhất 500 người dùng truy cập đồng thời, đặc biệt vào giờ cao điểm (ví dụ như các đợt khuyến mãi hay lễ tết).
* Giao diện thân thiện và dễ sử dụng:
* Giao diện được xây dựng đơn giản, trực quan, giúp người dùng dễ dàng tra cứu thông tin chuyến bay, đặt vé và thanh toán, kể cả đối với những người không rành công nghệ.
* Tương thích với nhiều thiết bị và trình duyệt:
* Hệ thống sẽ được thiết kế để tương thích với các trình duyệt phổ biến hiện nay như Chrome, Firefox, Safari, Edge,...
* Giao diện hệ thống sẽ tự điều chỉnh để phù hợp với nhiều kích thước màn hình.

## PHẦN 3 – MÔ HÌNH HÓA HỆ THỐNG

### 

### 3.1 Mô hình hóa Use Case

#### 3.1.1 Mô hình Use Case tổng quát



*Hình 3.1 Mô hình use case tổng quát.*

| **Mã Actor** | **Actor** | **Ý nghĩa** |
| --- | --- | --- |
| A1 | Admin | Quản trị viên, có quyền cao nhất để quản lý toàn bộ hệ thống. (Admin bao gồm quyền truy cập của nhân viên). |
| A2 | Nhân viên | Người hỗ trợ khách hàng, xử lý hành lý và check-in. (Nhân viên bao gồm quyền truy cập của thành viên). |
| A3 | Thành viên | Là khách hàng đã đăng nhập và có thể sử dụng đầy đủ chức năng đặt vé. (Thành viên bao gồm quyền truy cập của khách hàng). |
| A4 | Khách hàng | Người dùng chưa đăng nhập, chỉ sử dụng được chức năng cơ bản |

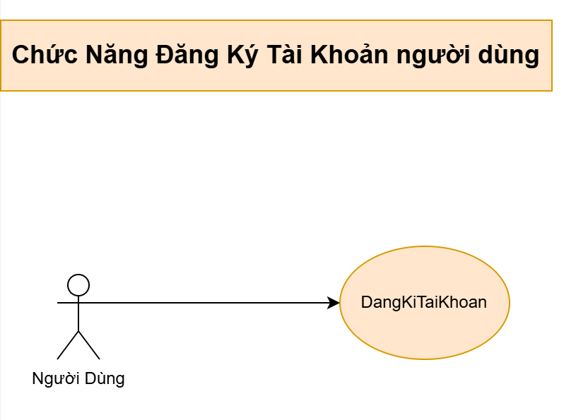
*Bảng 3.1 Danh sách Actor.*

| **STT** | **Use Case** | **Actor** | **Ý nghĩa** |
| --- | --- | --- | --- |
| 1 | DangKiTaiKhoan | A4 | Đăng ký tài khoản để sử dụng hệ thống |
| 2 | DangNhap | A4 | Đăng nhập vào hệ thống |
| 3 | QuanLiThongTinCaNhan | A3 | Thành viên có thể xem và chỉnh sửa thông tin cá nhân |
| 4 | TimKiemChuyenBay | A4 | Tìm kiếm chuyến bay theo điểm đi, điểm đến, ngày,... |
| 5 | XemChiTietChuyenBay | A4 | Xem thông tin chi tiết của chuyến bay đã chọn |
| 6 | ChonChuyenBay | A3 | Thành viên chọn chuyến bay muốn đặt |
| 7 | ChonHangGhe | A3 | Thành viên chọn hạng ghế trong chuyến bay |
| 8 | DatVe | A3 | Đặt vé cho chuyến bay đã chọn |
| 9 | ThanhToan | A3 | Tiến hành thanh toán vé đã đặt |
| 10 | HuyVe | A3 | Hủy vé đã đặt nếu có nhu cầu |
| 11 | QuanLiMayBay | A1 | Quản trị viên quản lý thông tin máy bay |
| 12 | QuanLiChuyenBay | A1 | Quản lý lịch trình và thông tin các chuyến bay |
| 13 | QuanLiGheNgoi | A1 | Quản lý danh sách và cấu hình chỗ ngồi trong máy bay |
| 14 | XemBaoCao | A1 | Xem báo cáo tổng hợp của hệ thống |
| 15 | QuanLiTaiKhoanKhachHang | A1 | Quản lý danh sách và thông tin người dùng hệ thống |
| 16 | Check\_inHanhLi | A2 | Nhân viên thực hiện quy trình check-in hành lý cho khách hàng |
| 17 | QuanLiHanhLi | A2 | Nhân viên theo dõi và xử lý thông tin hành lý |
| 18 | HoTroKhachHang | A2 | Nhân viên hỗ trợ khách hàng giải quyết các thắc mắc hoặc sự cố |

*Bảng 3.2 Danh sách Use Case.*

#### 3.1.2 Đặc tả use case

##### 3.1.2.1 Use case “DangKiTaiKhoan”



*Hình 3.2 Use case“DangKiTaiKhoan”.*

* **Tóm tắt:**

Use case này mô tả cách người dùng truy cập hệ thống, thực hiện thao tác đăng ký tài khoản để trở thành thành viên và sử dụng đầy đủ các chức năng của hệ thống.

* **Dòng sự kiện chính:**

Use case này bắt đầu khi một người dùng muốn đăng ký một tài khoản để trở thành một thành viên của hệ thống:

1. Người dùng truy cập vào giao diện chính của hệ thống.
2. Người dùng chọn chức năng "Đăng ký".
3. Hệ thống hiển thị form Đăng ký tài khoản mới.
4. Người dùng nhập các thông tin bắt buộc:  
   4.1. Họ tên.  
   4.2. Email.  
   4.3. Số điện thoại.  
   4.4. Tên đăng nhập.  
   4.5. Mật khẩu.
5. Người dùng nhấn nút "Đăng ký".
6. Hệ thống kiểm tra tính hợp lệ của dữ liệu nhập vào.
7. Nếu hợp lệ:  
   7.1. Hệ thống tạo mới tài khoản và lưu vào cơ sở dữ liệu.  
   7.2. Hiển thị thông báo đăng ký thành công.  
   7.3. Chuyển hướng người dùng đến màn hình đăng nhập.
8. Kết thúc usecase.

* **Dòng sự kiện khác:**

Dùng để xử lý các lỗi hoặc ngoại lệ trong dòng sự kiện chính, thường là do dữ liệu bị thiếu hoặc không hợp lệ:

* Dữ liệu nhập không đầy đủ: Hệ thống gửi thông báo "Thông tin nhập không đầy đủ. Vui lòng kiểm tra lại!".
* Mật khẩu không đủ mạnh: Hệ thống thông báo “Mật khẩu yếu. Vui lòng sử dụng mật khẩu có chứa ký tự đặc biệt, chữ số, chữ cái thường và viết hoa”.
* Tên đăng nhập đã tồn tại: Hệ thống thông báo "Tên đăng nhập đã tồn tại. Vui lòng chọn tên khác!".
* **Pre-conditions (Điều kiện tiên quyết):**

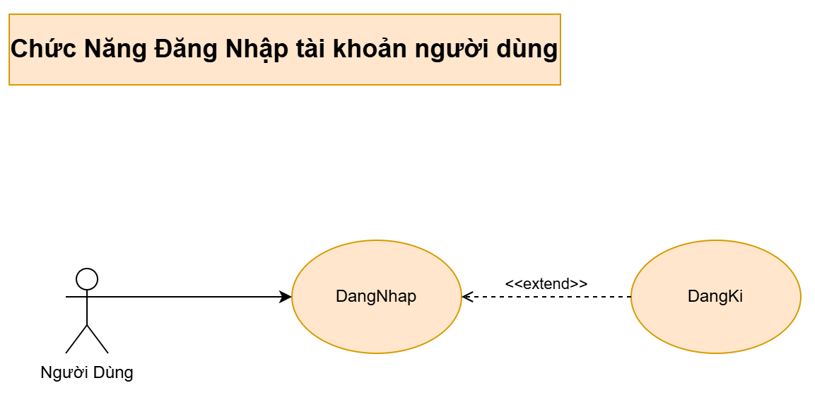
Người dùng chưa có tài khoản và chưa đăng nhập hệ thống.

* **Post-conditions (Kết quả đạt được):**

Một tài khoản mới được tạo trong hệ thống, người dùng có thể sử dụng tài khoản đó để đăng nhập và trở thành thành viên (A3).

##### 

##### 3.1.2.2 Use case “DangNhap”



*Hình 3.3 Use case“DangNhap”.*

* **Tóm tắt:**

Use case này mô tả cách người dùng truy cập hệ thống và thực hiện thao tác đăng nhập để sử dụng các chức năng phù hợp với vai trò (thành viên, nhân viên, admin).

* **Dòng sự kiện chính:**

Use case này bắt đầu khi một người dùng tiến hành đăng nhập để sử dụng các chức năng của hệ thống:

1. Người dùng truy cập vào giao diện chính của hệ thống.
2. Người dùng chọn chức năng "Đăng nhập".
3. Hệ thống hiển thị form Đăng nhập.
4. Người dùng nhập thông tin đăng nhập:  
   4.1. Tên đăng nhập.  
   4.2. Mật khẩu.
5. Người dùng nhấn nút "Đăng nhập".
6. Hệ thống kiểm tra thông tin:
7. Nếu hợp lệ:  
   7.1. Hệ thống xác định vai trò (A1/A2/A3).  
   7.2. Cấp quyền truy cập tương ứng.  
   7.3. Hiển thị thông báo đăng nhập thành công.  
   7.4. Chuyển hướng đến giao diện phù hợp
8. Kết thúc usecase.

* **Dòng sự kiện khác:**

Dùng để xử lý các lỗi hoặc ngoại lệ trong dòng sự kiện chính, thường là do dữ liệu bị thiếu hoặc không hợp lệ:

* Tài khoản không tồn tại: Hệ thống gửi thông báo "Tài khoản không tồn tại. Vui lòng kiểm tra lại!".
* Mật khẩu sai: Hệ thống gửi thông báo "Mật khẩu không chính xác. Vui lòng thử lại!".
* Tài khoản bị khóa hoặc chưa kích hoạt: Hệ thống gửi thông báo "Tài khoản đang bị khóa hoặc chưa kích hoạt. Vui lòng liên hệ quản trị viên!".
* Người dùng chưa có tài khoản: Có thể chọn chức năng Đăng ký, được mở rộng từ Use Case này.
* **Pre-conditions (Điều kiện tiên quyết):**

Người dùng đã có tài khoản hợp lệ trong hệ thống.

* **Post-conditions (Kết quả đạt được):**

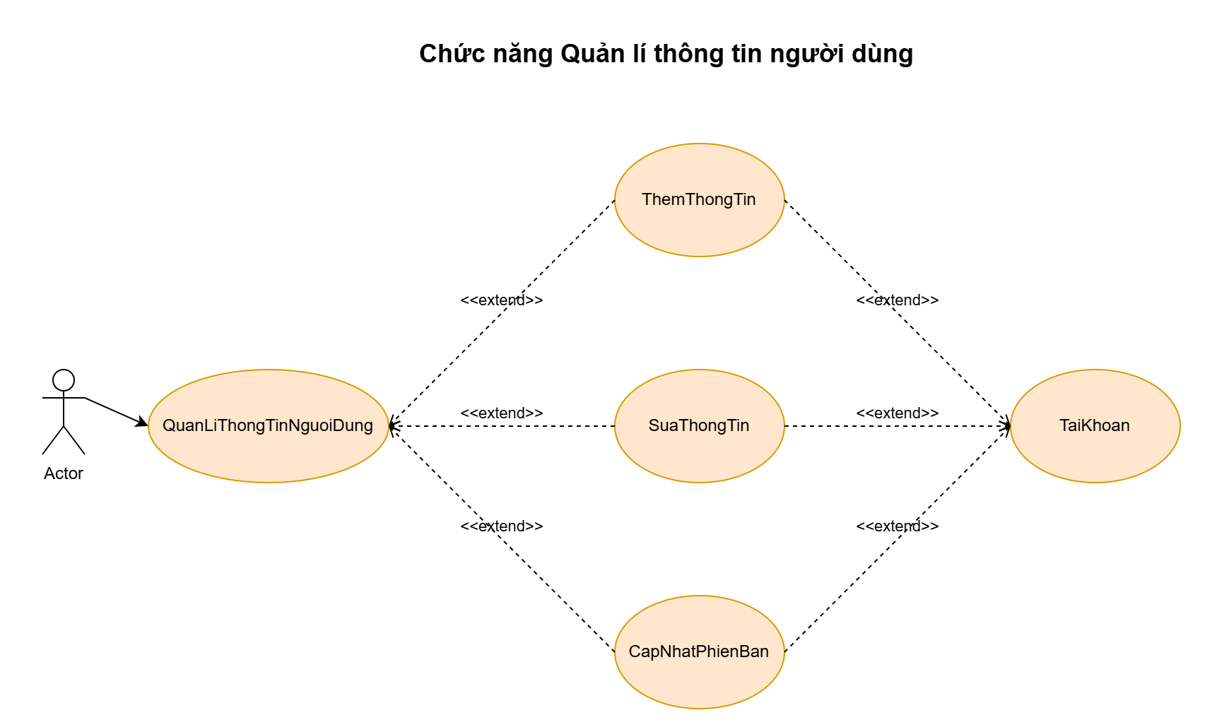
Người dùng được đăng nhập vào hệ thống, có thể truy cập các chức năng phù hợp với quyền của mình (A1/A2/A3).

### 

### 

##### 

##### 3.1.2.3 Use case “QuanLiThongTinNguoiDung”



*Hình 3.4 Use case“QuanLiThongTinNguoiDung”.*

* **Tóm tắt:**

Use case này mô tả cách người dùng truy cập hệ thống để xem, cập nhật, bổ sung hoặc chỉnh sửa thông tin cá nhân. Ngoài ra, người dùng có thể cập nhật phiên bản tài khoản hoặc cấu hình liên quan đến tài khoản của mình.

* **Dòng sự kiện chính:**

Use case này bắt đầu khi một người dùng tiến hành quản lý thông tin cá nhân của mình trên hệ thống:

1. Người dùng truy cập vào giao diện chính của hệ thống.
2. Người dùng chọn chức năng "Quản lý thông tin người dùng".
3. Hệ thống hiển thị danh sách các mục có thể thao tác:  
   3.1. Thêm Thông Tin.  
   3.2. Sửa Thông Tin.  
   3.3. Cập Nhật Phiên Bản.  
   3.4. Tài Khoản.
4. Người dùng chọn một trong các chức năng trên để thực hiện.
5. Thêm Thông Tin:  
   5.1. Người dùng nhấn vào Thêm Thông Tin.  
   5.2. Hệ thống hiển thị form nhập thông tin mới.  
   5.3. Người dùng điền đầy đủ thông tin và nhấn nút Lưu.  
   5.4. Nếu thông tin hợp lệ, hệ thống cập nhật và hiển thị thông báo "Thêm thành công".
6. Sửa Thông Tin:  
   6.1. Người dùng chọn mục Sửa Thông Tin.  
   6.2. Hệ thống hiển thị form có sẵn dữ liệu người dùng.  
   6.3. Người dùng chỉnh sửa thông tin mong muốn và nhấn Cập nhật.  
   6.4. Nếu dữ liệu hợp lệ, hệ thống lưu thay đổi và hiển thị "Cập nhật thành công".
7. Cập Nhật Phiên Bản:  
   7.1. Người dùng chọn mục Cập Nhật Phiên Bản (nếu có bản cập nhật tài khoản/tính năng mới).  
   7.2. Hệ thống hiển thị các phiên bản khả dụng.  
   7.3. Người dùng chọn phiên bản và xác nhận cập nhật.  
   7.4. Hệ thống cập nhật và thông báo "Phiên bản đã được cập nhật."
8. Tài Khoản:  
   8.1. Người dùng chọn mục Tài Khoản để xem thông tin đăng nhập hoặc các thiết lập cá nhân khác (đổi mật khẩu, email,…).  
   8.2. Người dùng có thể chỉnh sửa các thông tin liên quan tài khoản.  
   8.3. Sau khi chỉnh sửa, hệ thống kiểm tra và lưu thay đổi nếu hợp lệ.
9. Kết thúc usecase.

* **Dòng sự kiện khác:**

Dùng để xử lý các lỗi hoặc ngoại lệ trong dòng sự kiện chính, thường là do dữ liệu bị thiếu hoặc không hợp lệ:

* Thông tin nhập thiếu hoặc không hợp lệ: Hệ thống gửi thông báo "Vui lòng điền đầy đủ và đúng định dạng thông tin!".
* Không thể cập nhật thông tin: Hệ thống gửi thông báo "Có lỗi xảy ra trong quá trình cập nhật. Vui lòng thử lại sau!".
* Tài khoản chưa đăng nhập: Hệ thống gửi thông báo "Bạn cần đăng nhập để sử dụng chức năng này!".
* **Pre-conditions (Điều kiện tiên quyết):**

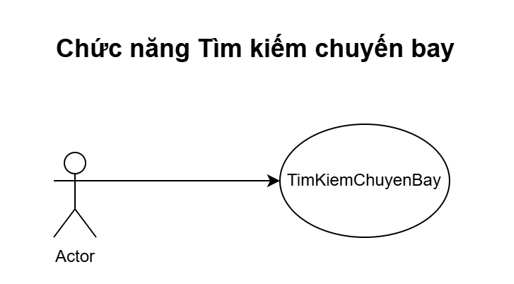
Người dùng đã đăng nhập vào hệ thống với vai trò phù hợp (A1/A2/A3).

* **Post-conditions (Kết quả đạt được):**

Các thay đổi được lưu vào cơ sở dữ liệu, và hệ thống hiển thị lại thông tin cập nhật mới nhất.

##### 

##### 3.1.2.4 Use case “TimKiemChuyenBay”



*Hình 3.5 Use case“TimKiemChuyenBay”.*

* **Tóm tắt:**

Use case này mô tả cách người dùng hoặc khách hàng sử dụng hệ thống để tìm kiếm các chuyến bay phù hợp với nhu cầu của họ như điểm đi, điểm đến, ngày bay,...

* **Dòng sự kiện chính:**

Use case này bắt đầu khi một người dùng tiến hành tìm kiếm các chuyến bay trên hệ thống:

1. Người dùng truy cập vào giao diện chính của hệ thống.
2. Người dùng chọn chức năng "Tìm kiếm chuyến bay".
3. Hệ thống hiển thị giao diện nhập thông tin tìm kiếm
4. Người dùng nhập đầy đủ thông tin cần thiết:  
   4.1. Điểm đi.  
   4.2. Điểm đến.  
   4.3. Ngày đi (và ngày về nếu có).  
   4.4. Số lượng hành khách.
5. Nhấn nút "Tìm kiếm".
6. Hệ thống xử lý truy vấn và tìm các chuyến bay phù hợp.
7. Nếu có kết quả:  
   7.1. Hiển thị danh sách chuyến bay phù hợp với thông tin đã nhập.  
   7.2. Người dùng có thể chọn xem chi tiết chuyến bay hoặc đặt vé.
8. Kết thúc usecase.

* **Dòng sự kiện khác:**

Dùng để xử lý các lỗi hoặc ngoại lệ trong dòng sự kiện chính, thường là do dữ liệu bị thiếu hoặc không hợp lệ:

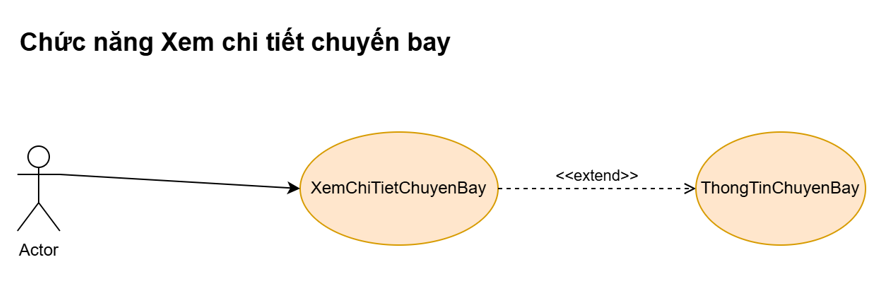
* Không nhập đủ thông tin: Hệ thống gửi thông báo "Vui lòng điền đầy đủ thông tin trước khi tìm kiếm!".
* Không có chuyến bay phù hợp: Hệ thống gửi thông báo "Không tìm thấy chuyến bay phù hợp.".
* **Pre-conditions (Điều kiện tiên quyết):**

Người dùng có thể không cần đăng nhập vẫn sử dụng được chức năng này. Giao diện tìm kiếm phải khả dụng trên hệ thống.

* **Post-conditions (Kết quả đạt được):**

Hệ thống hiển thị danh sách chuyến bay phù hợp với các tiêu chí tìm kiếm của người dùng. Người dùng có thể xem chi tiết, hoặc tiếp tục quy trình chọn chuyến bay để đặt vé nếu muốn.

##### 3.1.2.5 Use case “XemChiTietChuyenBay”



*Hình 3.6 Use case“XemChiTietChuyenBay”.*

* **Tóm tắt:**

Use case này mô tả cách người dùng hoặc khách hàng truy cập và xem thông tin chi tiết của một chuyến bay cụ thể, sau khi đã thực hiện thao tác tìm kiếm chuyến bay.

* **Dòng sự kiện chính:**

Use case này bắt đầu khi một người dùng tiến hành xem chi tiết một chuyến bay trên hệ thống:

1. Người dùng truy cập vào giao diện chính của hệ thống.
2. Hệ thống hiển thị danh sách chuyến bay phù hợp với điều kiện tìm kiếm.
3. Người dùng chọn một chuyến bay cụ thể từ danh sách.
4. Hệ thống hiển thị giao diện chi tiết chuyến bay, bao gồm:  
   4.1. Mã chuyến bay.  
   4.2. Thời gian khởi hành và đến nơi.  
   4.3. Sân bay đi và đến.  
   4.4. Loại máy bay.  
   4.5. Tình trạng chuyến bay.  
   4.6. Giá vé và các hạng ghế.
5. Người dùng có thể xem thêm các tiện ích mở rộng liên quan đến chuyến bay (nếu có).
6. Kết thúc usecase.

* **Dòng sự kiện khác:**

Dùng để xử lý các lỗi hoặc ngoại lệ trong dòng sự kiện chính, thường là do dữ liệu bị thiếu hoặc không hợp lệ:

* Chuyến bay không tồn tại hoặc đã bị huỷ: Hệ thống gửi thông báo "Chuyến bay không còn khả dụng!".
* Chưa chọn chuyến bay cụ thể: Hệ thống gửi thông báo "Vui lòng chọn chuyến bay để xem chi tiết!".
* **Pre-conditions (Điều kiện tiên quyết):**

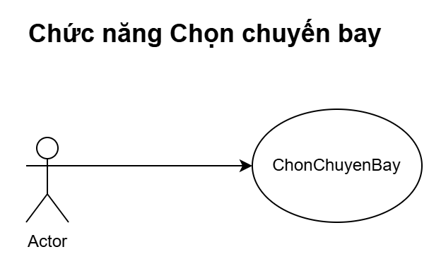
Người dùng có thể không cần đăng nhập vẫn sử dụng được chức năng này. Người dùng đã tìm kiếm chuyến bay.

* **Post-conditions (Kết quả đạt được):**

Thông tin chi tiết của chuyến bay được hiển thị thành công để người dùng tiếp tục thực hiện các hành động như chọn chuyến bay, đặt vé, v.v...

##### 

##### 3.1.2.6 Use case “ChonChuyenBay”



*Hình 3.7 Use case“ChonChuyenBay”.*

* **Tóm tắt:**

Use case này mô tả cách người dùng lựa chọn một chuyến bay cụ thể từ danh sách các chuyến bay được hiển thị sau khi tìm kiếm, để tiếp tục thực hiện đặt vé.

* **Dòng sự kiện chính:**

Use case này bắt đầu khi một người dùng tiến hành chọn một chuyến bay trên hệ thống:

1. Người dùng truy cập vào giao diện chính của hệ thống.
2. Hệ thống hiển thị danh sách các chuyến bay phù hợp với điều kiện tìm kiếm.
3. Người dùng chọn một chuyến bay cụ thể từ danh sách.
4. Hệ thống đánh dấu chuyến bay đó là được chọn để tiếp tục các bước tiếp theo như đặt vé hoặc xem thông tin chi tiết.
5. Kết thúc usecase.

* **Dòng sự kiện khác:**

Dùng để xử lý các lỗi hoặc ngoại lệ trong dòng sự kiện chính, thường là do dữ liệu bị thiếu hoặc không hợp lệ:

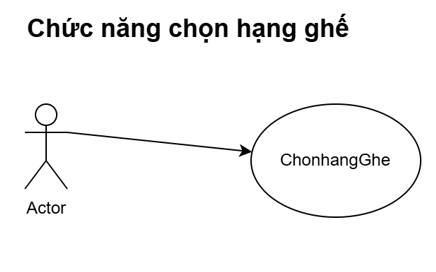
* Không có chuyến bay nào phù hợp với điều kiện tìm kiếm: Hệ thống gửi thông báo "Không tìm thấy chuyến bay phù hợp!".
* Người dùng chưa chọn chuyến bay mà cố gắng tiếp tục: Hệ thống hiển thị thông báo "Vui lòng chọn một chuyến bay!".
* **Pre-conditions (Điều kiện tiên quyết):**

Người dùng đã thực hiện thao tác tìm kiếm chuyến bay và đang ở giao diện danh sách chuyến bay.

* **Post-conditions (Kết quả đạt được):**

Một chuyến bay được chọn thành công, sẵn sàng để thực hiện các hành động tiếp theo như xem chi tiết, đặt vé...

##### 3.1.2.7 Use case “ChonHangGhe”



*Hình 3.8 Use case“ChonHangGhe”.*

* **Tóm tắt:**

Use case này mô tả cách người dùng chọn hạng ghế mong muốn (ví dụ: Phổ thông, Thương gia, Hạng nhất...) sau khi đã chọn chuyến bay, nhằm chuẩn bị cho việc đặt vé hoặc xem giá vé tương ứng.

* **Dòng sự kiện chính:**

Use case bắt đầu khi người dùng đã chọn một chuyến bay và muốn chọn hạng ghế:

1. Người dùng truy cập giao diện chọn hạng ghế.
2. Hệ thống hiển thị các hạng ghế khả dụng của chuyến bay đã chọn, bao gồm thông tin như:  
   2.1. Tên hạng ghế (Phổ thông, Thương gia…).  
   2.2. Giá vé tương ứng.  
   2.3. Số lượng chỗ còn trống.
3. Người dùng chọn một hạng ghế phù hợp.
4. Hệ thống lưu lại lựa chọn của người dùng. tiếp tục đến bước tiếp theo.
5. Kết thúc usecase.

* **Dòng sự kiện khác:**

Xử lý ngoại lệ hoặc lỗi xảy ra trong quá trình chọn hạng ghế:

* Không còn ghế trống ở hạng ghế được chọn: Hệ thống gửi thông báo "Hạng ghế này đã hết chỗ, vui lòng chọn hạng khác!".
* **Pre-conditions (Điều kiện tiên quyết):**

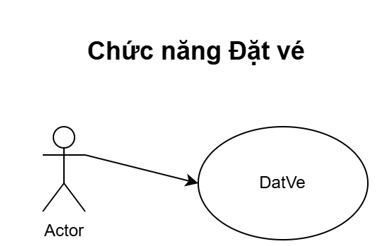
Người dùng đã chọn chuyến bay cụ thể và chuyển sang bước chọn hạng ghế.

* **Post-conditions (Kết quả đạt được):**

Hệ thống ghi nhận thành công hạng ghế mà người dùng đã chọn, sẵn sàng cho bước tiếp theo trong quy trình đặt vé.

##### 

##### 3.1.2.8 Use case “DatVe”



*Hình 3.9 Use case“DatVe”.*

* **Tóm tắt:**

Use case này mô tả quy trình người dùng thực hiện đặt vé máy bay sau khi đã chọn chuyến bay và hạng ghế phù hợp.

* **Dòng sự kiện chính:**

Use case này bắt đầu khi người dùng quyết định đặt vé cho chuyến bay đã chọn:

1. Người dùng truy cập vào giao diện đặt vé.
2. Hệ thống yêu cầu người dùng cung cấp thông tin cần thiết, bao gồm:  
   3.1. Họ tên hành khách.  
   3.2. Thông tin liên hệ (số điện thoại, email).  
   3.3. Số giấy tờ tùy thân (CMND/CCCD/Hộ chiếu).
3. Người dùng kiểm tra lại thông tin chuyến bay và hạng ghế.
4. Hệ thống hiển thị tổng chi phí vé.
5. Người dùng xác nhận đặt vé.
6. Hệ thống thực hiện đặt chỗ và thông báo đặt vé thành công.
7. Hệ thống gửi vé điện tử về email hoặc hiển thị mã đặt chỗ.
8. Kết thúc usecase.

* **Dòng sự kiện khác:**

Dùng để xử lý lỗi hoặc các tình huống ngoại lệ:

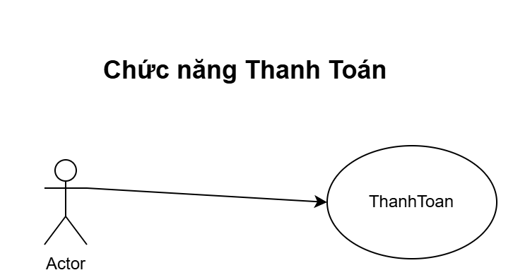
* Thiếu thông tin hành khách: Hệ thống gửi thông báo "Vui lòng nhập đầy đủ thông tin hành khách!".
* Đặt vé thất bại do hết chỗ/ngắt kết nối hệ thống: Hệ thống hiển thị thông báo "Đặt vé không thành công, vui lòng thử lại sau!".
* **Pre-conditions (Điều kiện tiên quyết):**

Người dùng đã chọn chuyến bay và hạng ghế, sẵn sàng tiến hành đặt vé.

* **Post-conditions (Kết quả đạt được):**

Hệ thống ghi nhận thành công giao dịch đặt vé, thông tin vé được lưu trữ và gửi đến người dùng.

##### 3.1.2.9 Use case “ThanhToan”



*Hình 3.10 Use case“ThanhToan”.*

* **Tóm tắt:**

Use case này mô tả quá trình người dùng tiến hành thanh toán cho vé máy bay đã đặt sau khi hoàn tất các bước trước đó.

* **Dòng sự kiện chính:**

Use case bắt đầu khi người dùng chuyển sang bước thanh toán sau khi đặt vé:

1. Người dùng chọn hình thức thanh toán .
2. Hệ thống hiển thị giao diện nhập thông tin thanh toán tương ứng.
3. Người dùng nhập đầy đủ thông tin thanh toán.
4. Hệ thống xử lý giao dịch và kết nối với cổng thanh toán.
5. Hệ thống xác nhận thanh toán thành công và thông báo cho người dùng.
6. Vé máy bay được xác nhận chính thức và gửi về email của người dùng hoặc hiển thị mã đặt chỗ trên hệ thống.
7. Kết thúc usecase.

* **Dòng sự kiện khác:**

Xử lý ngoại lệ hoặc lỗi xảy ra trong quá trình chọn hạng ghế:

* Giao dịch bị từ chối: Hệ thống hiển thị thông báo "Thanh toán thất bại, vui lòng kiểm tra lại thông tin hoặc thử lại sau!".
* Người dùng thoát giữa chừng: Hệ thống quay lại bước xác nhận thanh toán và thông báo "Bạn chưa hoàn tất thanh toán!".
* **Pre-conditions (Điều kiện tiên quyết):**

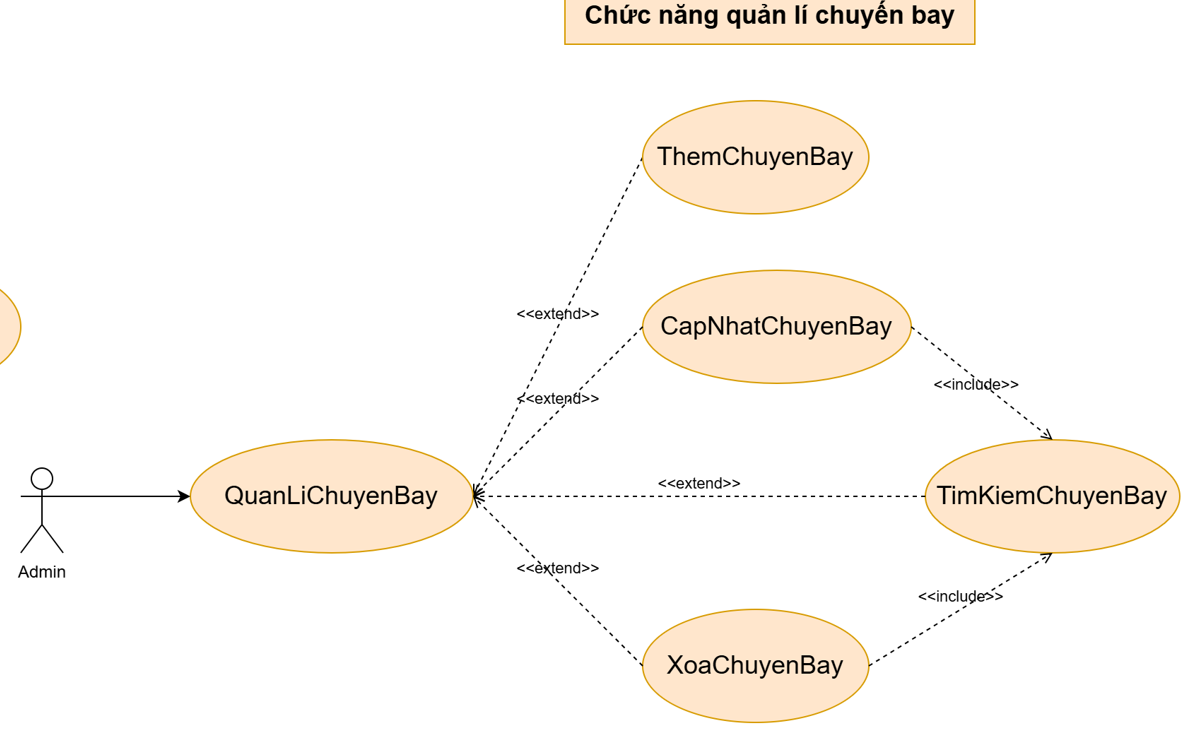
Người dùng đã hoàn tất việc đặt vé và sẵn sàng thanh toán.

* **Post-conditions (Kết quả đạt được):**

Hệ thống xác nhận giao dịch thanh toán thành công và chính thức giữ chỗ cho vé máy bay đã đặt.

##### 

##### 3.1.2.10 Use case “QuanLiMayBay”



*Hình 3.11 Use case“QuanLiMayBay”.*

* **Tóm tắt:**

Use case này mô tả cách Admin thực hiện việc quản lý danh sách máy bay trong hệ thống. Chức năng này bao gồm hai hành động chính: Thêm máy bay mới vào hệ thống hoặc xóa máy bay không còn sử dụng.

* **Dòng sự kiện chính:**

Use case bắt đầu khi Admin đăng nhập và truy cập chức năng quản lý máy bay:

1. Admin chọn chức năng “Quản lý máy bay” trên giao diện quản trị.
2. Hệ thống hiển thị danh sách máy bay hiện có và các tùy chọn quản lý.
3. Thêm máy bay:  
   3.1. Hệ thống yêu cầu nhập thông tin máy bay (mã số, loại, số ghế, hãng hàng không…).  
   3.2. Admin điền thông tin và xác nhận.  
   3.3. Hệ thống kiểm tra tính hợp lệ và lưu thông tin vào cơ sở dữ liệu.
4. Xóa máy bay:  
   4.1. Admin chọn máy bay muốn xóa từ danh sách.  
   4.2. Hệ thống hiển thị cảnh báo xác nhận.  
   4.3. Nếu Admin xác nhận, hệ thống xóa máy bay khỏi cơ sở dữ liệu.
5. Hệ thống thông báo kết quả thao tác và quay lại màn hình danh sách.
6. Kết thúc usecase.

* **Dòng sự kiện khác:**

Xử lý ngoại lệ hoặc lỗi xảy ra trong quá trình chọn hạng ghế:

* Nếu thông tin máy bay nhập vào không hợp lệ (thiếu dữ liệu, trùng mã số,…), hệ thống báo lỗi cụ thể và yêu cầu nhập lại.
* Nếu cố gắng xóa máy bay đang được sử dụng trong chuyến bay sắp tới, hệ thống hiển thị thông báo lỗi: “Không thể xóa máy bay đang hoạt động!”.
* **Pre-conditions (Điều kiện tiên quyết):**

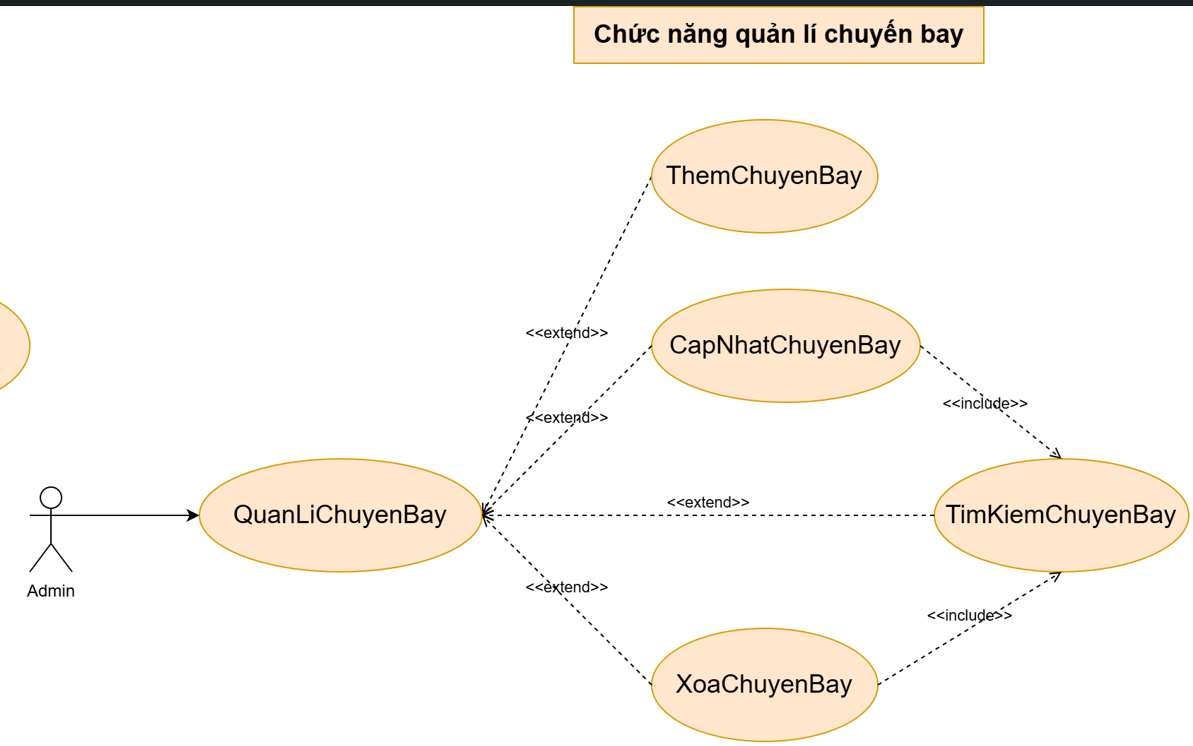
Admin đã đăng nhập thành công vào hệ thống với quyền quản trị.

* **Post-conditions (Kết quả đạt được):**

Hệ thống ghi nhận thay đổi về danh sách máy bay (đã thêm mới hoặc đã xóa), cập nhật dữ liệu phục vụ các chức năng liên quan (đặt vé, lập lịch bay…).

##### 

##### 3.1.2.11 Use case “QuanLiChuyenBay”



*Hình 3.12 Use case“QuanLiChuyenBay”.*

* **Tóm tắt:**

Use case này mô tả cách Admin thực hiện việc quản lý các chuyến bay trong hệ thống. Bao gồm các thao tác thêm mới, cập nhật, xóa hoặc tìm kiếm chuyến bay nhằm duy trì dữ liệu chính xác và hỗ trợ quy trình đặt vé.

* **Dòng sự kiện chính:**

Use case bắt đầu khi Admin đăng nhập và truy cập chức năng quản lý chuyến bay:

1. Admin chọn chức năng “Quản lý chuyến bay” trong giao diện quản trị.
2. Hệ thống hiển thị các tùy chọn quản lý chuyến bay.
3. Thêm chuyến bay:  
   3.1. Admin chọn “Thêm chuyến bay”.  
   3.2. Hệ thống yêu cầu nhập thông tin chuyến bay (mã số, giờ khởi hành, điểm đi, điểm đến, máy bay sử dụng…).  
   3.3. Sau khi nhập đầy đủ và hợp lệ, hệ thống lưu lại dữ liệu.
4. Cập nhật chuyến bay:  
   4.1. Admin tìm chuyến bay cần sửa đổi (bao gồm use case “TimKiemChuyenBay”).  
   4.2. Hệ thống hiển thị thông tin hiện tại, cho phép chỉnh sửa.  
   4.3. Admin xác nhận và hệ thống cập nhật dữ liệu mới.
5. Xóa chuyến bay:  
   5.1. Admin tìm và chọn chuyến bay muốn xóa (bao gồm use case “TimKiemChuyenBay”).  
   5.2. Hệ thống yêu cầu xác nhận.  
   5.3. Sau khi xác nhận, hệ thống xóa chuyến bay nếu không còn liên quan đến đặt vé.
6. Kết thúc usecase.

* **Dòng sự kiện khác:**

Xử lý ngoại lệ hoặc lỗi xảy ra trong quá trình quản lý:

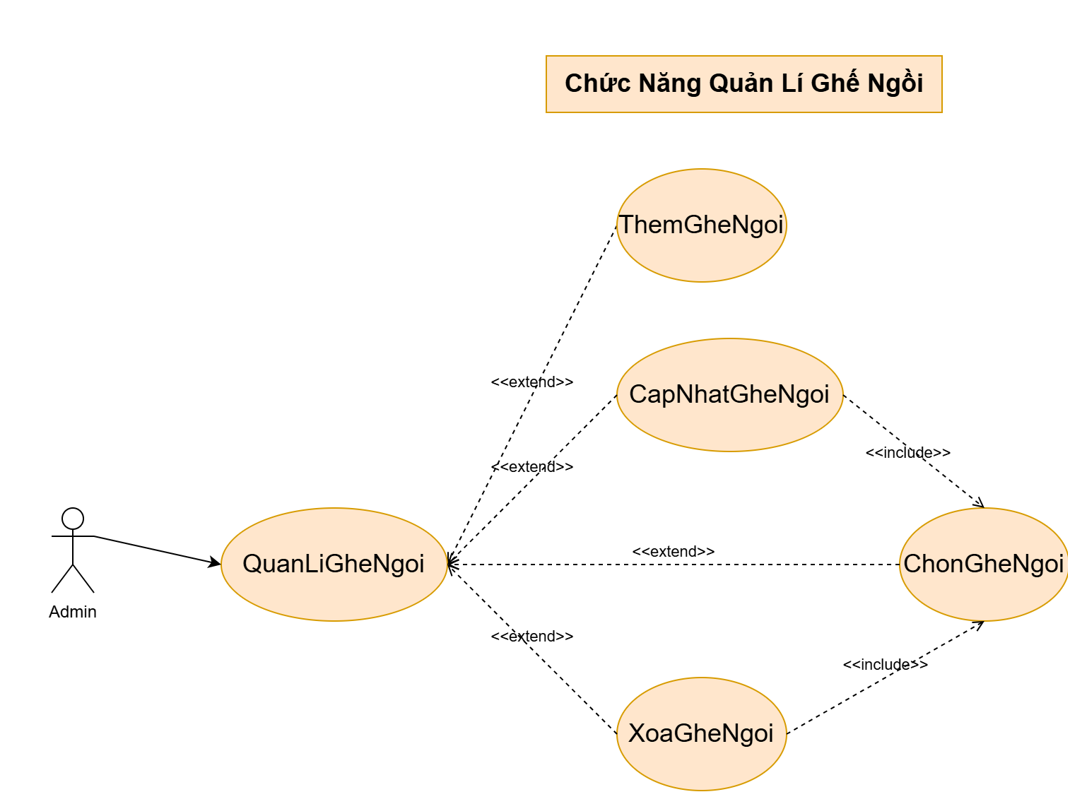
* Thêm hoặc cập nhật thông tin không hợp lệ (giờ sai định dạng, mã chuyến bay trùng…): hệ thống hiển thị thông báo lỗi cụ thể.
* Không tìm thấy chuyến bay trong quá trình tìm kiếm: hệ thống hiển thị thông báo “Không tìm thấy chuyến bay phù hợp!”.
* Cố gắng xóa chuyến bay đang có vé đặt: hệ thống báo lỗi “Không thể xóa chuyến bay đang được đặt vé!”.
* **Pre-conditions (Điều kiện tiên quyết):**

Admin đã đăng nhập thành công và có quyền truy cập chức năng quản lý chuyến bay.

* **Post-conditions (Kết quả đạt được):**

Hệ thống ghi nhận thành công các thao tác quản lý chuyến bay (thêm, sửa, xóa), đảm bảo dữ liệu chuyến bay được cập nhật chính xác, phục vụ quá trình đặt vé và tra cứu.

##### 3.1.2.12 Use case “QuanLiGheNgoi”



*Hình 3.13 Use case“QuanLiGheNgoi”.*

* **Tóm tắt:**

Use case này mô tả cách Admin thực hiện các thao tác quản lý ghế ngồi trên hệ thống bao gồm: thêm mới ghế, cập nhật thông tin ghế, xóa ghế không còn sử dụng và hỗ trợ người dùng trong việc chọn ghế ngồi phù hợp cho chuyến bay.

* **Dòng sự kiện chính:**

Use case bắt đầu khi Admin truy cập chức năng quản lý ghế ngồi trong hệ thống quản trị:

1. Admin đăng nhập và chọn chức năng “Quản lý ghế ngồi”.
2. Hệ thống hiển thị giao diện quản lý ghế cùng các chức năng hỗ trợ.
3. Thêm ghế ngồi:  
   3.1. Admin chọn “Thêm ghế ngồi”.  
   3.2. Hệ thống yêu cầu nhập các thông tin: số ghế, loại ghế, vị trí, hạng ghế, tình trạng…  
   3.3. Sau khi nhập và xác nhận, hệ thống lưu thông tin ghế vào cơ sở dữ liệu.
4. Cập nhật ghế ngồi:  
   4.1. Admin tìm và chọn ghế cần chỉnh sửa (bao gồm use case “ChonGheNgoi”).  
   4.2. Hệ thống hiển thị thông tin ghế hiện tại để chỉnh sửa.  
   4.3. Sau khi cập nhật và xác nhận, hệ thống lưu thay đổi.
5. Xóa ghế ngồi:  
   5.1. Admin chọn ghế muốn xóa (bao gồm use case “ChonGheNgoi”).  
   5.2. Hệ thống yêu cầu xác nhận.  
   5.3.Nếu ghế không nằm trong chuyến bay đang được sử dụng, hệ thống sẽ xóa ghế khỏi dữ liệu.
6. Kết thúc usecase.

* **Dòng sự kiện khác:**

Xử lý ngoại lệ hoặc lỗi xảy ra trong quá trình quản lý:

* Nếu nhập thiếu hoặc sai định dạng thông tin ghế (số ghế trùng, vị trí không hợp lệ,...): hệ thống báo lỗi và yêu cầu nhập lại.
* Nếu cố gắng xóa ghế đang được sử dụng trong một chuyến bay đã có hành khách đặt chỗ: hệ thống thông báo lỗi “Không thể xóa ghế đang được sử dụng!”.
* Nếu không tìm thấy ghế cần cập nhật hoặc xóa: hệ thống hiển thị “Không tìm thấy ghế phù hợp!”.
* **Pre-conditions (Điều kiện tiên quyết):**

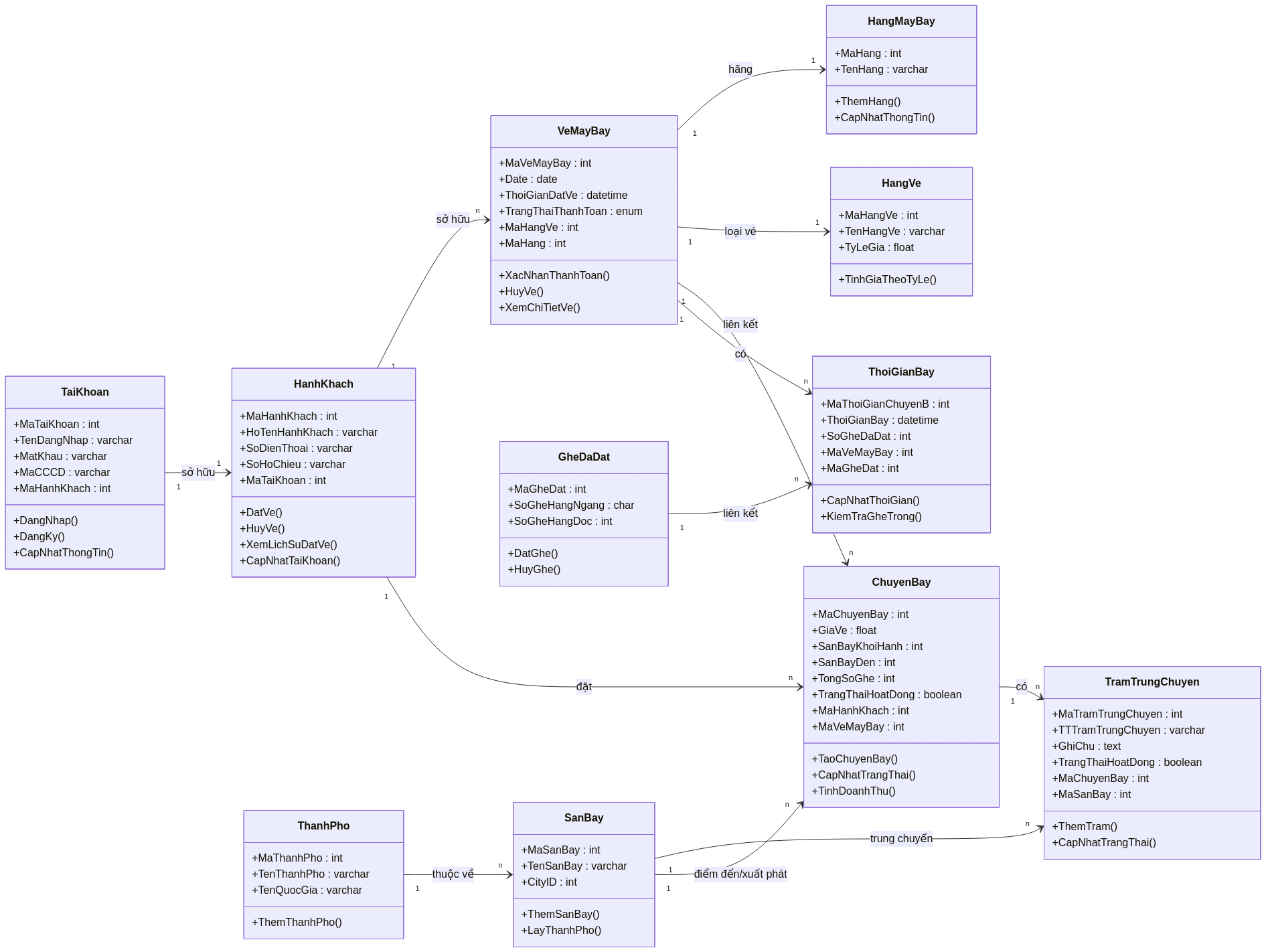
Admin đã đăng nhập thành công và có quyền truy cập vào chức năng quản lý ghế ngồi.

* **Post-conditions (Kết quả đạt được):**

Hệ thống cập nhật thành công thông tin liên quan đến các ghế ngồi (thêm, sửa, xóa), đảm bảo dữ liệu sẵn sàng phục vụ cho quy trình chọn ghế và đặt vé của người dùng.

### 3.2 Mô hình hóa cấu trúc

#### 3.2.1 Sơ đồ class



*Hình 3.14 Sơ đồ class*

Các lớp (Class) chính:

* HanhKhach (Hành khách): Đại diện cho người dùng đặt vé máy bay, chứa thông tin cá nhân như họ tên, số điện thoại, số hộ chiếu và mã tài khoản liên kết. Hành khách có thể đặt vé, hủy vé và xem lịch sử.
* TaiKhoan (Tài khoản): Quản lý thông tin đăng nhập, gồm tên đăng nhập, mật khẩu, mã CCCD và liên kết đến hành khách. Tài khoản có chức năng đăng nhập, đăng ký và cập nhật thông tin.
* VeMayBay (Vé máy bay): Chứa thông tin về vé như mã vé, thời gian đặt vé, trạng thái thanh toán, mã hạng vé và mã hạng máy bay. Vé có thể được xác nhận, hủy và xem chi tiết.
* ChuyenBay (Chuyến bay): Đại diện cho một chuyến bay cụ thể, gồm giá vé, sân bay khởi hành, sân bay đến, tổng số ghế, trạng thái hoạt động và mã vé. Có thể tạo chuyến bay, cập nhật trạng thái và tính doanh thu.
* ThoiGianBay (Thời gian bay): Ghi lại chi tiết thời gian cho chuyến bay cụ thể và liên kết đến vé và ghế. Có thể cập nhật thời gian và kiểm tra ghế trống.
* GheDaDat (Ghế đã đặt): Quản lý việc đặt ghế, bao gồm mã ghế, số ghế, số ghế hàng ngang/dọc, liên kết với thời gian bay. Có thể đặt ghế và hủy ghế.
* SanBay (Sân bay): Chứa thông tin mã và tên sân bay, và liên kết với thành phố.
* ThanhPho (Thành phố): Ghi lại tên thành phố và quốc gia, liên kết với sân bay.
* HangMayBay (Hãng máy bay): Quản lý thông tin hãng như tên và mã hãng. Có thể thêm hãng và cập nhật.
* HangVe (Hạng vé): Chứa loại hạng vé, tỷ lệ giá, và tên hạng vé. Dùng để tính giá theo tỷ lệ.
* TramTrungChuyen (Trạm trung chuyển): Ghi nhận các điểm trung chuyển của sân bay trong chuyến bay. Có thể thêm trạm và cập nhật trạng thái.

Mối quan hệ giữa các lớp:

* TaiKhoan - HanhKhach: Một tài khoản sở hữu một hành khách (1-1).
* HanhKhach - VeMayBay: Một hành khách có thể đặt nhiều vé máy bay (1-n).
* VeMayBay - ThoiGianBay: Mỗi vé liên kết với một thời gian bay cụ thể (1-1).
* ThoiGianBay - GheDaDat: Mỗi thời gian bay có thể có nhiều ghế đã đặt (1-n).
* GheDaDat - Ghe: Mỗi ghế đã đặt chỉ định một ghế cụ thể trong chuyến bay (1-1, implied by structure).
* ChuyenBay - ThoiGianBay: Một chuyến bay có nhiều khung giờ bay khác nhau (1-n).
* ChuyenBay - SanBay: Một chuyến bay xuất phát từ một sân bay và kết thúc tại một sân bay khác (1-1).
* SanBay - ThanhPho: Mỗi sân bay thuộc về một thành phố (1-1).
* ChuyenBay - VeMayBay: Một chuyến bay có thể liên kết với nhiều vé (1-n).
* ChuyenBay - TramTrungChuyen: Một chuyến bay có thể đi qua nhiều trạm trung chuyển (1-n).
* VeMayBay - HangVe: Mỗi vé thuộc một loại hạng vé (1-1).
* VeMayBay - HangMayBay: Mỗi vé thuộc về một hãng máy bay (1-1).

### 

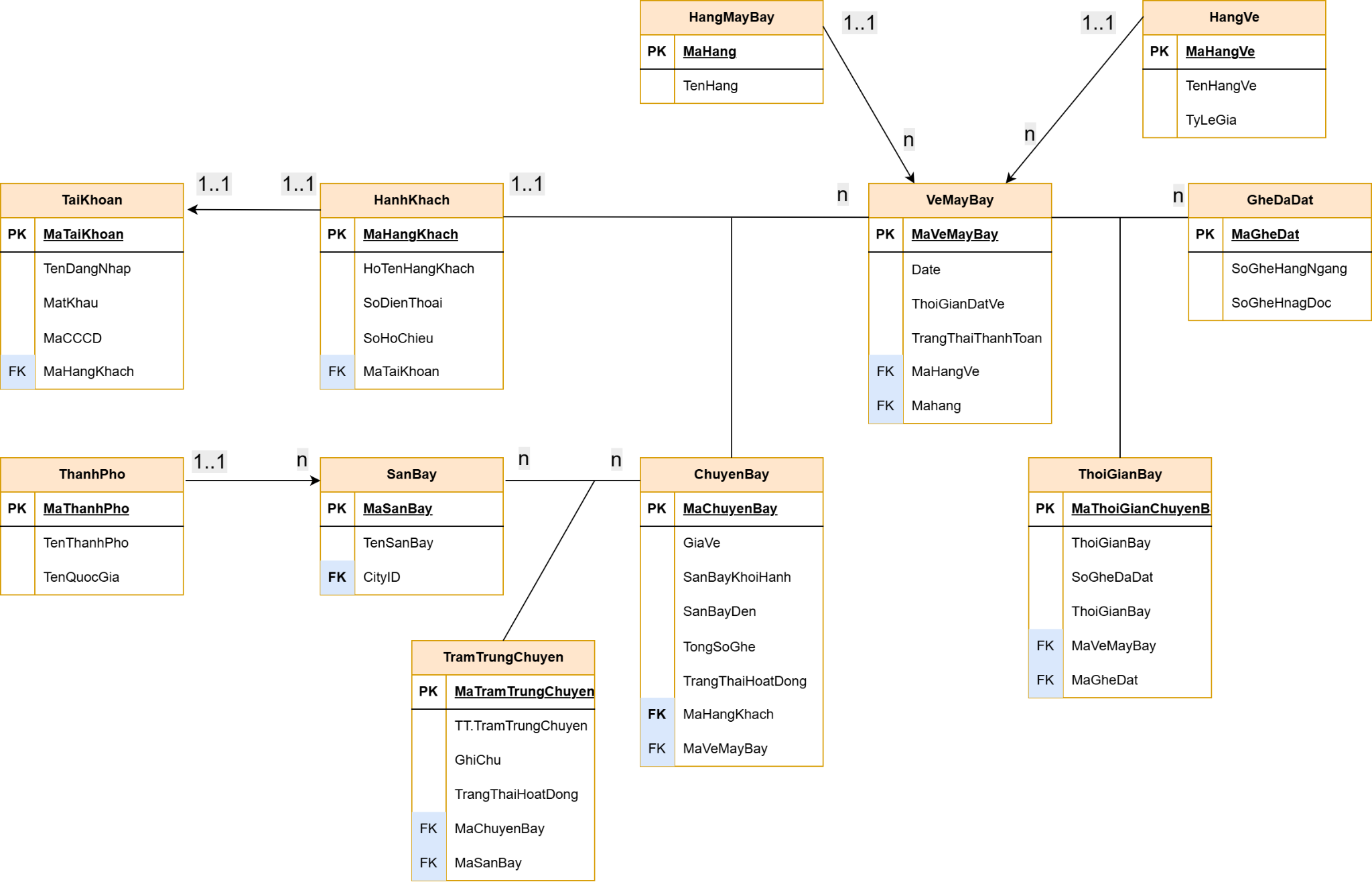
### 

### 

### 

### 

#### 3.2.2 Sơ đồ thực thể ERD



*Hình 3.15 Sơ đồ thực thể ERD*

Các thực thể (Entity) chính:

* Tài Khoản (TaiKhoan): Lưu thông tin đăng nhập của người dùng như tên đăng nhập, mật khẩu, mã CCCD. Mỗi tài khoản sẽ được liên kết với một hành khách cụ thể.
* Hành Khách (HanhKhach): Lưu trữ thông tin cá nhân của người dùng như họ tên, số điện thoại, số hộ chiếu. Mỗi hành khách chỉ gắn với một tài khoản.
* Hãng Máy Bay (HangMayBay): Danh sách các hãng hàng không hoạt động trong hệ thống, ví dụ như Vietnam Airlines, Vietjet Air,…
* Hạng Vé (HangVe): Lưu thông tin về các loại hạng vé như phổ thông, thương gia, hạng nhất, và tỷ lệ giá tương ứng của mỗi hạng.
* Vé Máy Bay (VeMayBay): Quản lý thông tin vé bao gồm ngày đặt, thời gian đặt vé, trạng thái thanh toán. Vé sẽ thuộc một hãng máy bay và một hạng vé.
* Ghế Đã Đặt (GheDaDat): Ghi lại thông tin vị trí ghế mà người dùng đã chọn theo hàng ngang và hàng dọc.
* Thời Gian Bay (ThoiGianBay): Quản lý thời gian bay cụ thể gắn với vé máy bay và ghế ngồi đã đặt.
* Thành Phố (ThanhPho): Danh sách thành phố và quốc gia tương ứng. Dùng để xác định vị trí của các sân bay.
* Sân Bay (SanBay): Thông tin các sân bay, mỗi sân bay thuộc một thành phố cụ thể.
* Chuyến Bay (ChuyenBay): Thông tin chi tiết về chuyến bay như giá vé, điểm khởi hành, điểm đến, số ghế và trạng thái hoạt động.
* Trạm Trung Chuyển (TramTrungChuyen): Quản lý các điểm dừng trung gian của chuyến bay, nếu có, và các ghi chú liên quan.

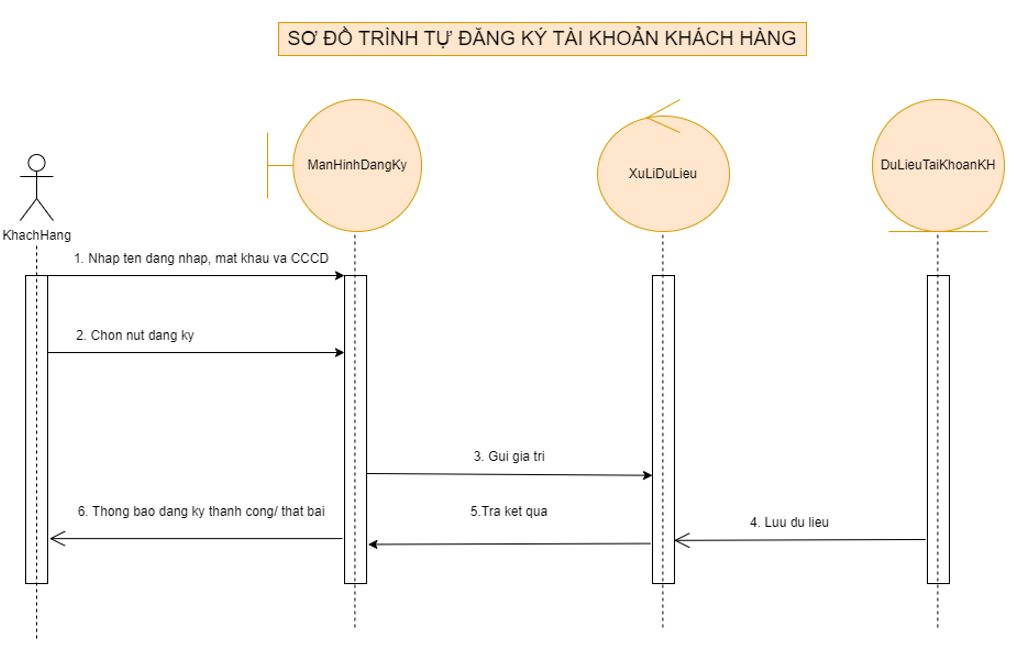
Mối quan hệ giữa các thực thể:

* Tài khoản – Hành khách: Mối quan hệ 1-1: Mỗi tài khoản tương ứng với một hành khách, và ngược lại.
* Hành khách – Vé máy bay: Mối quan hệ 1-n: Một hành khách có thể đặt nhiều vé máy bay khác nhau.
* Hãng máy bay – Vé máy bay: Mối quan hệ 1-n: Mỗi vé sẽ thuộc một hãng máy bay cụ thể.
* Hạng vé – Vé máy bay: Mỗi vé máy bay gắn với một hạng vé duy nhất.
* Vé máy bay – Ghế đã đặt: Mối quan hệ 1-n: Một vé có thể bao gồm nhiều ghế ngồi được chọn.
* Vé máy bay – Thời gian bay: Mối quan hệ 1-n: Một vé máy bay có thể gắn với nhiều thời điểm bay khác nhau tùy theo hành trình hoặc đặt lại.
* Thành phố – Sân bay: Mối quan hệ 1-n: Mỗi thành phố có thể có nhiều sân bay.
* Chuyến bay – Sân bay: Mỗi chuyến bay sẽ chỉ định rõ sân bay khởi hành và sân bay đến.
* Chuyến bay – Trạm trung chuyển: Mối quan hệ 1-n: Một chuyến bay có thể có nhiều trạm dừng trung chuyển.
* Chuyến bay – Vé máy bay: Một vé máy bay sẽ được phát hành dựa trên một chuyến bay cụ thể.

### 3.3 Mô hình hóa hành vi

#### 3.3.1 Sơ đồ tuần tự

##### 3.3.1.1 Quá trình đăng ký tài khoản của khách hàng



*Hình 3.16 Sơ đồ tuần tự quá trình đăng ký tài khoản của khách hàng*

* **Dòng sự kiện chính:**

Sơ đồ thể hiện quá trình khách hàng đăng ký tài khoản vào hệ thống như sau:

1. Khách hàng nhập tên đăng nhập, mật khẩu và CCCD vào giao diện.
2. Sau đó nhấn nút đăng ký để gửi yêu cầu.
3. Giao diện đăng ký gửi dữ liệu tài khoản đến bộ phận xử lý dữ liệu.
4. Bộ xử lý lưu dữ liệu vào cơ sở dữ liệu tài khoản khách hàng.
5. Kết quả xử lý (thành công hoặc thất bại) được trả về cho bộ xử lý.
6. Hệ thống sẽ hiển thị thông báo đăng ký thành công hoặc thất bại.

##### 3.3.1.2 Quá trình đăng nhập của khách hàng

#### 

*Hình 3.17 Sơ đồ tuần tự quá trình đăng nhập của khách hàng.*

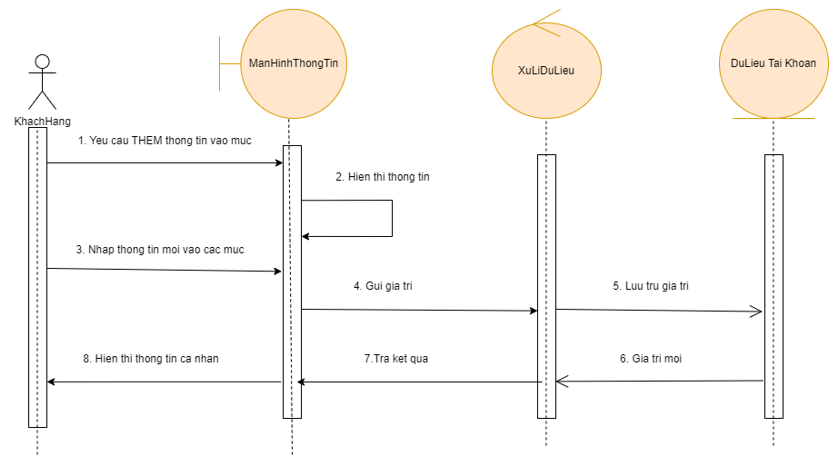
#### 

* **Dòng sự kiện chính:**

Sơ đồ thể hiện quá trình khách hàng đăng nhập vào hệ thống như sau:

1. Khách hàng nhập tên đăng nhập và mật khẩu vào giao diện đăng nhập.
2. Sau đó nhấn nút đăng nhập để gửi yêu cầu.
3. Giao diện đăng nhập gửi dữ liệu tài khoản đến bộ phận xử lý dữ liệu.
4. Bộ xử lý truy vấn thông tin từ cơ sở dữ liệu tài khoản khách hàng.
5. Kết quả truy vấn (hợp lệ hoặc không hợp lệ) được trả về cho bộ xử lý.
6. Bộ xử lý gửi kết quả xác thực về giao diện.
7. Hệ thống hiển thị thông báo đăng nhập thành công hoặc thất bại cho khách hàng.

##### 3.3.1.3 Quá trình thêm thông tin của khách hàng



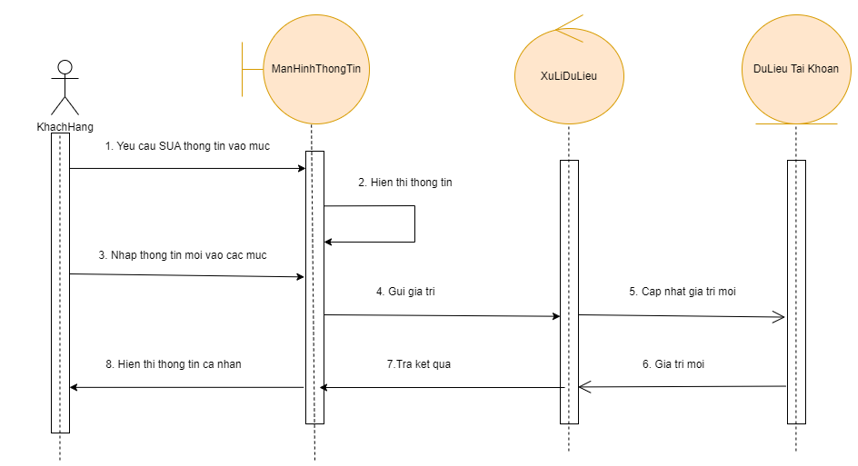
*Hình 3.18 Sơ đồ tuần tự quá trình thêm thông tin của khách hàng.*

* **Dòng sự kiện chính:**

Sơ đồ thể hiện quá trình khách hàng thêm thông tin cá nhân vào hệ thống như sau:

1. Khách hàng yêu cầu thêm thông tin vào các mục trên giao diện.
2. Giao diện hiển thị các mục thông tin cần nhập.
3. Khách hàng nhập thông tin mới vào các mục tương ứng.
4. Giao diện gửi dữ liệu thông tin mới đến bộ phận xử lý dữ liệu.
5. Bộ xử lý lưu dữ liệu mới vào cơ sở dữ liệu tài khoản khách hàng.
6. Dữ liệu mới được xác nhận và trả lại cho bộ xử lý.
7. Kết quả xử lý được trả về giao diện.
8. Hệ thống hiển thị thông tin cá nhân đã cập nhật cho khách hàng.

##### 3.3.1.4 Quá trình sửa thông tin của khách hàng



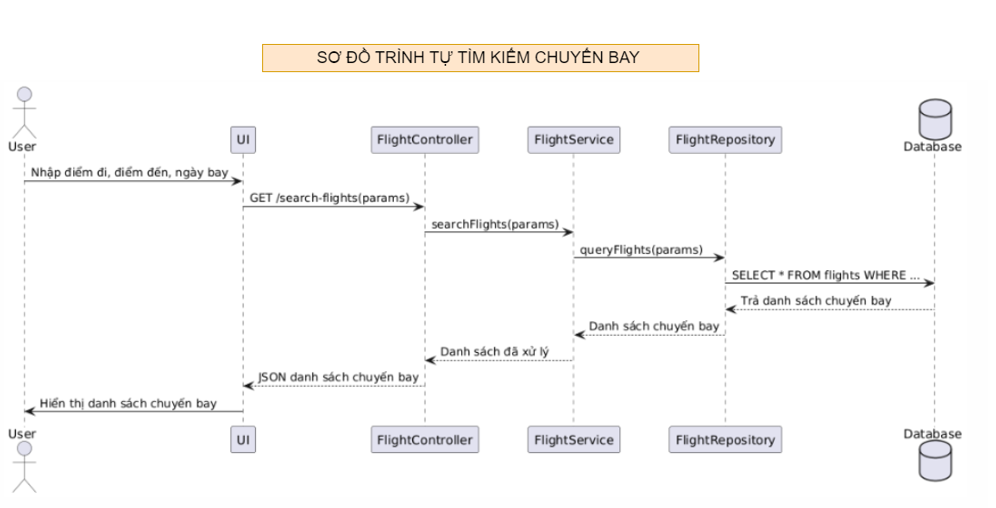
*Hình 3.19 Sơ đồ tuần tự quá trình sửa thông tin của khách hàng.*

* **Dòng sự kiện chính:**

Sơ đồ thể hiện quá trình khách hàng chỉnh sửa thông tin cá nhân trên hệ thống như sau:

1. Khách hàng yêu cầu sửa thông tin vào các mục trên giao diện.
2. Giao diện hiển thị các mục thông tin hiện có để khách hàng chỉnh sửa.
3. Khách hàng nhập thông tin mới vào các mục tương ứng.
4. Giao diện gửi dữ liệu thông tin mới đến bộ phận xử lý dữ liệu.
5. Bộ xử lý cập nhật dữ liệu mới vào cơ sở dữ liệu tài khoản khách hàng.
6. Dữ liệu mới được xác nhận và trả lại cho bộ xử lý.
7. Kết quả xử lý được trả về giao diện.
8. Hệ thống hiển thị thông tin cá nhân đã được chỉnh sửa cho khách hàng.

##### 3.3.1.5 Quá trình tìm kiếm chuyến bay của khách hàng



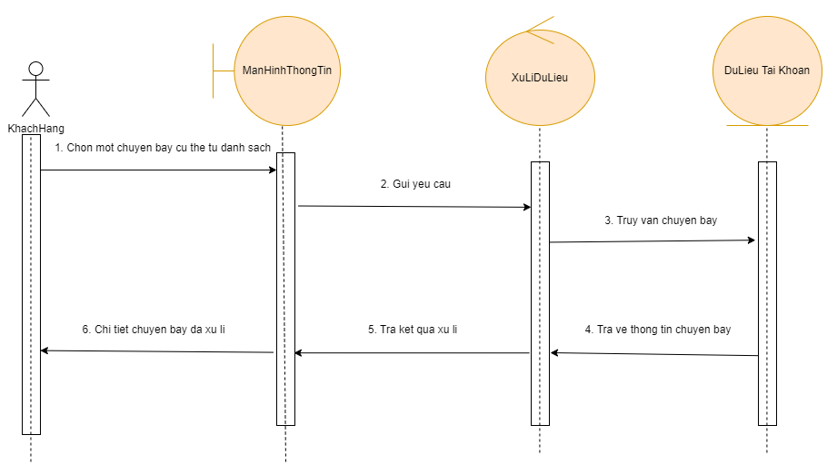
*Hình 3.20 Sơ đồ tuần tự quá trình tìm kiếm chuyến bay của khách hàng.*

* **Dòng sự kiện chính:**

Sơ đồ thể hiện quá trình khách hàng tìm kiếm chuyến bay trên hệ thống như sau:

1. Khách hàng nhập điểm đi, điểm đến và ngày bay vào giao diện tìm kiếm.
2. Giao diện gửi yêu cầu tìm kiếm chuyến bay đến bộ điều khiển.
3. Bộ điều khiển chuyển tiếp yêu cầu tìm kiếm đến dịch vụ xử lý.
4. Dịch vụ xử lý gửi truy vấn đến kho dữ liệu chuyến bay.
5. Kho dữ liệu thực hiện truy vấn cơ sở dữ liệu để lấy danh sách chuyến bay phù hợp.
6. Cơ sở dữ liệu trả về danh sách các chuyến bay thỏa điều kiện.
7. Kho dữ liệu trả danh sách chuyến bay về cho dịch vụ xử lý.
8. Dịch vụ xử lý gửi danh sách chuyến bay đã xử lý về bộ điều khiển.
9. Bộ điều khiển trả dữ liệu (ở dạng JSON) về giao diện người dùng.
10. Hệ thống hiển thị danh sách chuyến bay cho khách hàng.

##### 3.3.1.6 Quá trình xem chi tiết chuyến bay của khách hàng



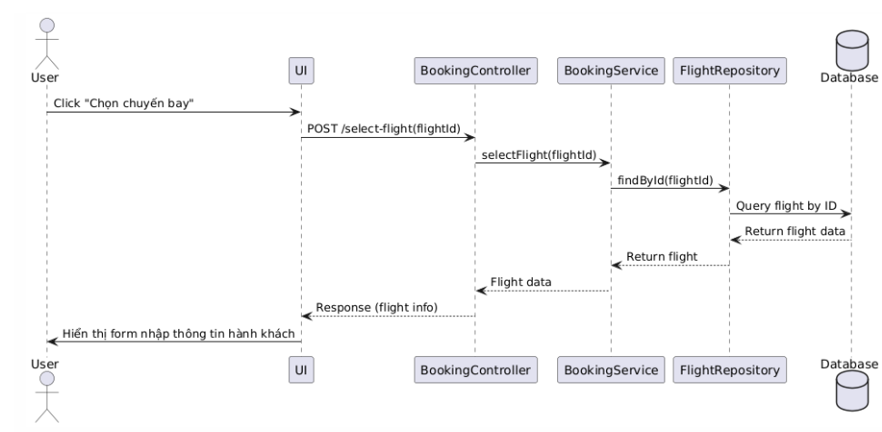
*Hình 3.21 Sơ đồ tuần tự quá trình xem chi tiết chuyến bay của khách hàng.*

* **Dòng sự kiện chính:**

Sơ đồ thể hiện quá trình khách hàng xem chi tiết một chuyến bay từ danh sách như sau:

1. Khách hàng chọn một chuyến bay từ danh sách hiển thị trên giao diện.
2. Giao diện gửi yêu cầu xem thông tin chi tiết chuyến bay đến bộ phận xử lý dữ liệu.
3. Bộ xử lý gửi truy vấn đến cơ sở dữ liệu tài khoản để lấy thông tin chuyến bay tương ứng.
4. Cơ sở dữ liệu trả về thông tin chi tiết của chuyến bay.
5. Bộ xử lý nhận kết quả xử lý và trả về cho giao diện.
6. Hệ thống hiển thị thông tin chi tiết của chuyến bay cho khách hàng.

##### 3.3.1.7 Quá trình chọn chuyến bay của khách hàng



*Hình 3.22 Sơ đồ tuần tự quá trình chọn chuyến bay của khách hàng.*

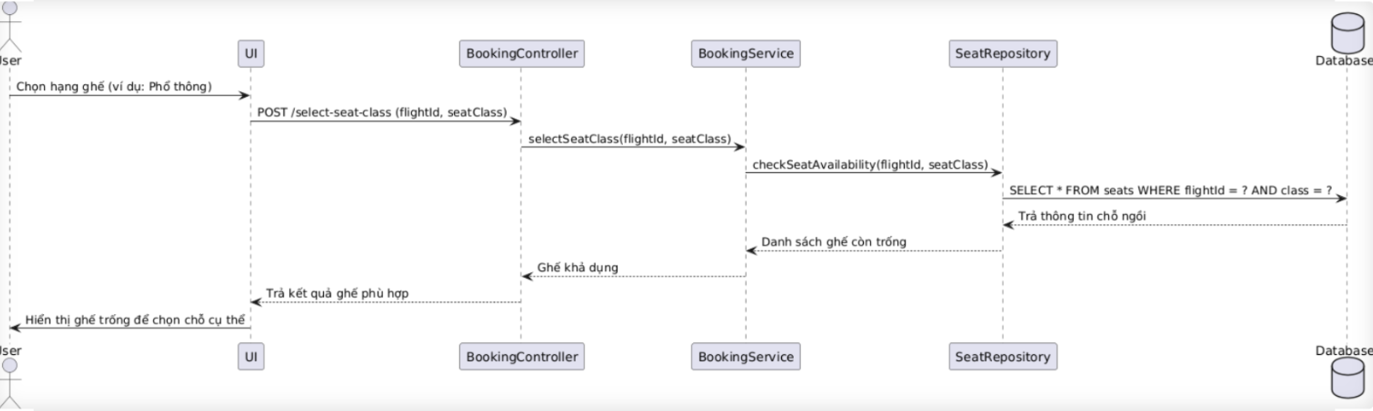
* **Dòng sự kiện chính:**

Sơ đồ thể hiện quá trình khách hàng chọn chuyến bay để đặt vé như sau:

1. Khách hàng nhấn nút “Chọn chuyến bay” trên giao diện.
2. Giao diện gửi yêu cầu chọn chuyến bay kèm flightId đến bộ điều khiển.
3. Bộ điều khiển chuyển tiếp yêu cầu đến dịch vụ đặt vé.
4. Dịch vụ đặt vé gửi truy vấn đến kho dữ liệu chuyến bay để tìm chuyến bay theo flightId.
5. Kho dữ liệu thực hiện truy vấn cơ sở dữ liệu để lấy dữ liệu chuyến bay.
6. Cơ sở dữ liệu trả về dữ liệu chuyến bay cho kho dữ liệu.
7. Kho dữ liệu chuyển dữ liệu chuyến bay về cho dịch vụ đặt vé.
8. Dịch vụ đặt vé trả dữ liệu chuyến bay về bộ điều khiển.
9. Bộ điều khiển gửi thông tin chuyến bay về giao diện người dùng.
10. Hệ thống hiển thị form nhập thông tin hành khách cho khách hàng.

##### 

##### 3.3.1.8 Quá trình chọn hạng ghế của khách hàng



*Hình 3.23 Sơ đồ tuần tự quá trình chọn hạng ghế của khách hàng.*

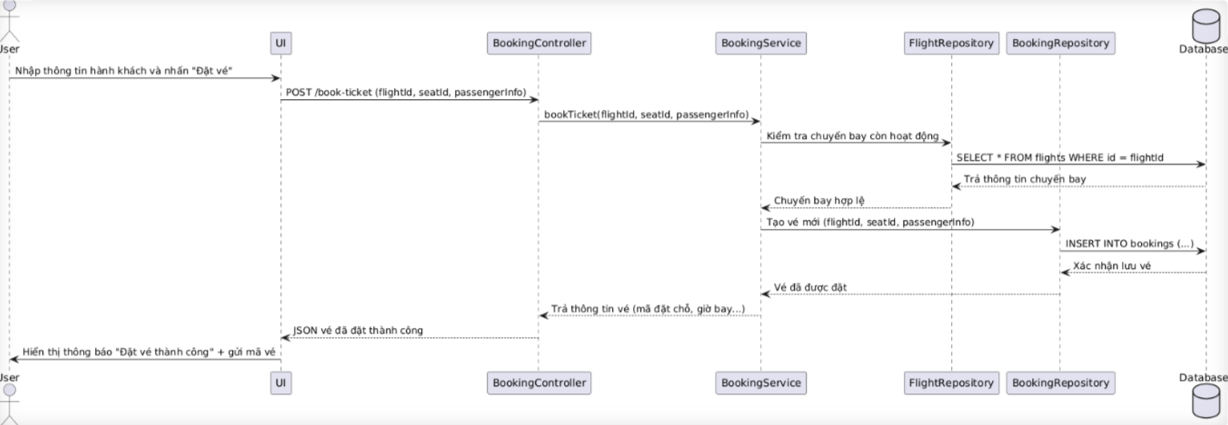
* **Dòng sự kiện chính:**

Sơ đồ thể hiện quá trình khách hàng chọn hạng ghế và xem danh sách ghế trống phù hợp như sau:

1. Khách hàng chọn hạng ghế (ví dụ: Phổ thông) trên giao diện.
2. Giao diện gửi yêu cầu chọn hạng ghế kèm theo flightId và seatClass đến bộ điều khiển đặt vé.
3. Bộ điều khiển chuyển tiếp yêu cầu đến dịch vụ đặt vé.
4. Dịch vụ đặt vé gửi yêu cầu kiểm tra tình trạng ghế đến kho dữ liệu chỗ ngồi với flightId và seatClass.
5. Kho dữ liệu thực hiện truy vấn cơ sở dữ liệu để lấy danh sách các ghế còn trống thuộc chuyến bay và hạng ghế tương ứng.
6. Cơ sở dữ liệu trả về thông tin danh sách chỗ ngồi còn trống cho kho dữ liệu.
7. Kho dữ liệu chuyển danh sách ghế khả dụng về cho dịch vụ đặt vé.
8. Dịch vụ đặt vé trả danh sách ghế phù hợp về cho bộ điều khiển.
9. Bộ điều khiển gửi kết quả về giao diện người dùng.
10. Hệ thống hiển thị danh sách ghế trống để khách hàng chọn chỗ cụ thể.

##### 

##### 3.3.1.9 Quá trình chọn đặt vé của khách hàng



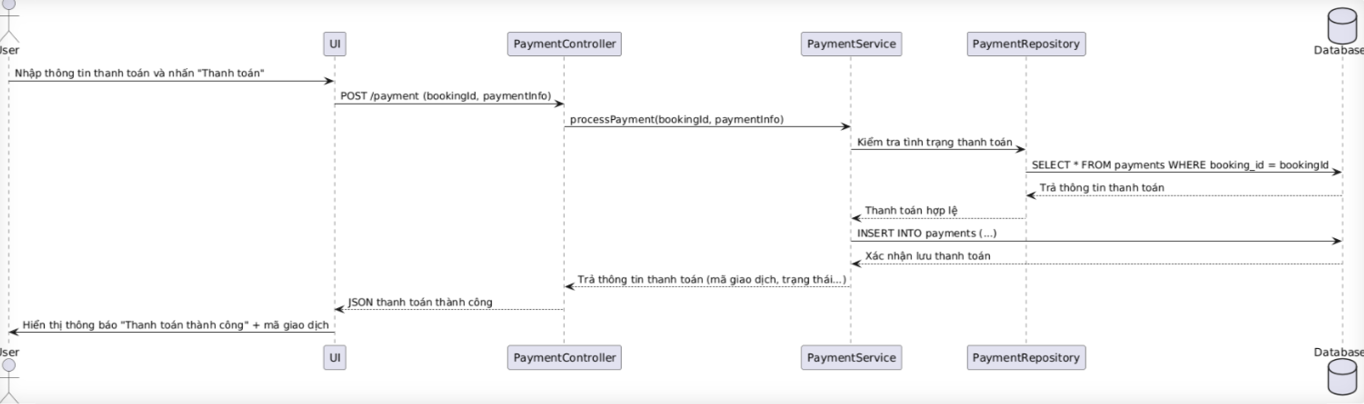
*Hình 3.24 Sơ đồ tuần tự quá trình đặt vé của khách hàng.*

* **Dòng sự kiện chính:**

Sơ đồ thể hiện quá trình khách hàng đặt vé máy bay trên hệ thống như sau:

1. Khách hàng nhập thông tin hành khách và nhấn "Đặt vé" trên giao diện.
2. Giao diện gửi yêu cầu đặt vé đến hệ thống, bao gồm các thông tin: flightId, seatId, và passengerInfo.
3. Hệ thống tiếp nhận yêu cầu và kiểm tra chuyến bay còn hoạt động hay không.
4. Nếu chuyến bay hợp lệ, hệ thống tiến hành tạo vé mới với các thông tin đã cung cấp.
5. Dữ liệu vé mới được lưu vào cơ sở dữ liệu, xác nhận việc đặt vé thành công.
6. Hệ thống trả thông tin vé đã đặt (mã đặt chỗ, giờ bay...) về lại cho giao diện.
7. Giao diện hiển thị thông báo "Đặt vé thành công" kèm mã vé cho khách hàng.

##### 3.3.1.10 Quá trình thanh toán của khách hàng



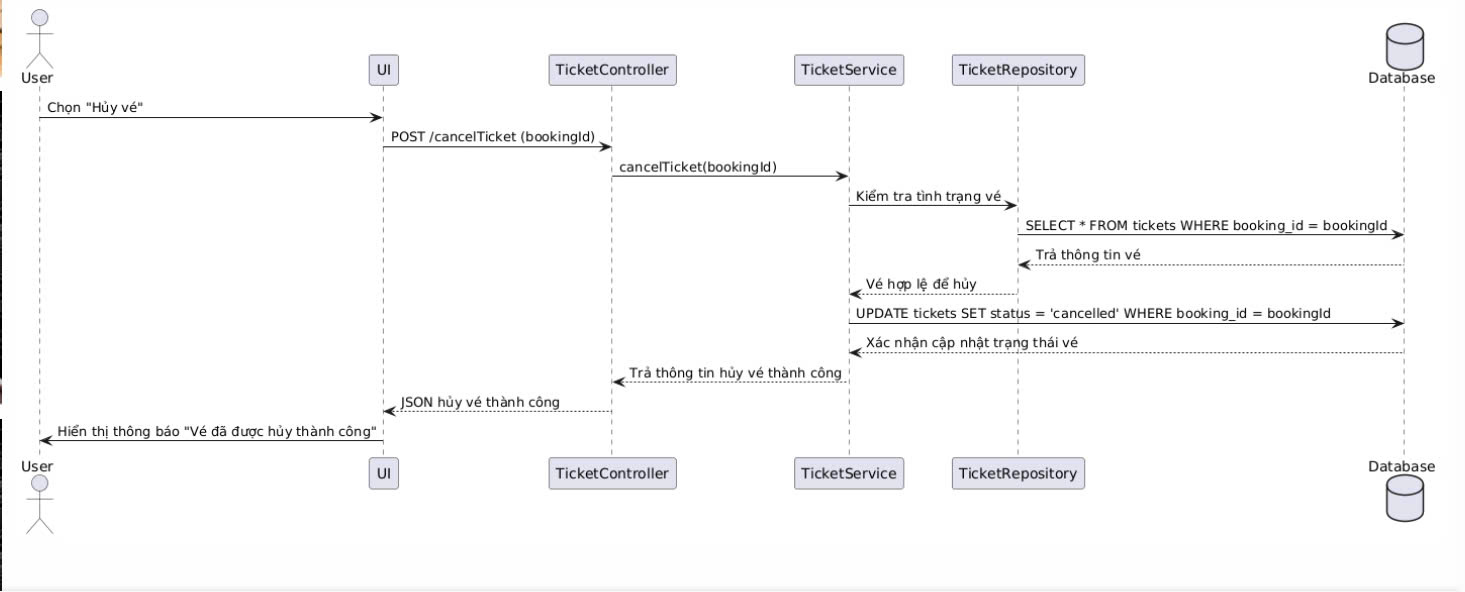
*Hình 3.25 Sơ đồ tuần tự quá trình thanh toán của khách hàng.*

* **Dòng sự kiện chính:**

Sơ đồ thể hiện quá trình khách hàng thanh toán trên hệ thống như sau:

1. Khách hàng nhập thông tin thanh toán và nhấn "Thanh toán" trên giao diện.
2. Giao diện gửi yêu cầu thanh toán đến hệ thống, bao gồm các thông tin: bookingId và paymentInfo.
3. Hệ thống tiếp nhận yêu cầu và kiểm tra trạng thái thanh toán của vé.
4. Nếu thanh toán hợp lệ, hệ thống tiến hành lưu thông tin thanh toán mới.
5. Dữ liệu thanh toán được lưu vào cơ sở dữ liệu, xác nhận thanh toán thành công.
6. Hệ thống trả thông tin thanh toán (mã giao dịch, trạng thái...) về lại cho giao diện.
7. Giao diện hiển thị thông báo "Thanh toán thành công" kèm mã giao dịch cho khách hàng.

##### 3.3.1.11 Quá trình hủy vé của khách hàng



*Hình 3.26 Sơ đồ tuần tự quá trình hủy vé của khách hàng.*

* **Dòng sự kiện chính:**

Sơ đồ thể hiện quá trình khách hàng hủy vé trên hệ thống như sau:

1. Khách hàng chọn "Hủy vé" trên giao diện.
2. Giao diện gửi yêu cầu hủy vé đến hệ thống, bao gồm thông tin bookingId.
3. Hệ thống tiếp nhận yêu cầu và kiểm tra trạng thái của vé tương ứng.
4. Nếu vé hợp lệ để hủy, hệ thống cập nhật trạng thái của vé thành "cancelled".
5. Thông tin cập nhật được lưu vào cơ sở dữ liệu, xác nhận hủy vé thành công.
6. Hệ thống trả thông tin hủy vé thành công về cho giao diện (dưới dạng JSON).
7. Giao diện hiển thị thông báo "Vé đã được hủy thành công" cho khách hàng.

## PHẦN 4 – THIẾT KẾ HỆ THỐNG

### 4.1 Thiết kế cơ sở dữ liệu

Cơ sở dữ liệu (CSDL) được thiết kế bằng MySQL với các bảng chính để lưu trữ và quản lý thông tin. CSDL được tích hợp với Backend thông qua thư viện MySQL2 hoặc Sequelize ORM để giao tiếp giữa Node.js (Express) và MySQL, đảm bảo dữ liệu được xử lý hiệu quả và bảo mật.

Các bảng chính trong CSDL bao gồm:

* Bảng users: Quản lý tài khoản người dùng. Các trường chính: id (khóa chính), username (duy nhất), password, role (phân quyền), email, created\_at.
* Bảng cities: Lưu tên thành phố và quốc gia. Trường chính: id, city\_name, country.
* Bảng airports: Quản lý sân bay. Gồm id, name, airport\_code, city\_id (liên kết với bảng cities).
* Bảng airlines: Thông tin hãng hàng không. Trường chính: id, airline\_name, airline\_code, is\_active.
* Bảng flights: Quản lý chuyến bay. Các trường: flight\_code (duy nhất), airline\_id, origin\_airport, destination\_airport, departure\_time, price, total\_seats.
* Bảng bookings: Lưu thông tin đặt vé. Trường chính: id, user\_id, flight\_id, total\_tickets, total\_amount, status, booking\_time.
* Bảng passengers: Lưu danh sách hành khách theo từng đơn đặt. Gồm: first\_name, last\_name, gender, date\_of\_birth, booking\_id.
* Bảng cart: Giỏ hàng tạm của người dùng. Trường chính: user\_id, flight\_id, seat\_number, added\_time.
* Bảng reports: Quản lý các báo cáo do người dùng tạo. Gồm: title, content, created\_at, created\_by.
* Bảng ticket\_suggestions: Lưu đề xuất vé máy bay nổi bật. Trường chính: origin\_city\_name, destination\_city\_name, image\_url, price.

### 4.2 Thiết kế giao diện người dùng

Giao diện người dùng được xây dựng bằng HTML, CSS và JavaScript, đảm bảo giao diện thân thiện, trực quan và dễ sử dụng. Hệ thống được thiết kế để hiển thị và tương tác với các API từ backend.

Các chức năng chính của Frontend bao gồm:

* Tìm kiếm và đặt vé máy bay: Giao diện cho phép người dùng chọn điểm đi, điểm đến, ngày khởi hành và tìm chuyến bay phù hợp; hỗ trợ đặt vé với thao tác chọn chuyến, điền thông tin hành khách và thanh toán.
* Quản lý đơn đặt vé: Hiển thị danh sách đơn đặt với thông tin chuyến bay, hành khách, trạng thái và tổng tiền; cho phép người dùng xem chi tiết, hủy vé hoặc cập nhật liên hệ nếu cần.
* Giỏ hàng: Lưu tạm thời các vé đã chọn trước khi thanh toán; hỗ trợ chỉnh sửa, xóa và tiến hành thanh toán từ giao diện giỏ hàng.
* Quản lý hệ thống (dành cho quản lý): Giao diện cho phép quản lý kiểm soát danh sách chuyến bay (thêm, sửa, xóa), tài khoản người dùng, tài khoản nhân viên và phân quyền truy cập; đồng thời điều chỉnh các đề xuất vé máy bay hiển thị trên trang chính.
* Phân tích và báo cáo: Hiển thị biểu đồ doanh thu, số lượng vé bán ra, tỷ lệ hủy và các tuyến bay phổ biến theo từng khoảng thời gian; hỗ trợ lọc theo ngày, tháng hoặc năm và phục vụ công tác thống kê doanh thu cho quản lý.

### 4.3 Thiết kế backend

Backend được xây dựng bằng Nodejs với framework Express, giúp xử lý logic nghiệp vụ và giao tiếp với cơ sở dữ liệu MySQL. Dựa trên cấu trúc hướng đối tượng, hệ thống được chia thành các module chính, mỗi module quản lý một thực thể trong hệ thống, bao gồm: Flight (Chuyến bay), Booking (Đơn đặt vé), Passenger (Hành khách), User (Tài khoản người dùng), và Report (Báo cáo).

Các chức năng chính của Backend bao gồm:

* Quản lý chuyến bay: Cung cấp API để tạo mới chuyến bay, cập nhật thông tin (giờ bay, sân bay đi/đến, hãng bay, số ghế, giá vé), xóa chuyến bay, và truy vấn danh sách các chuyến bay theo điều kiện tìm kiếm.
* Quản lý đơn đặt vé: Bao gồm việc tạo mới đơn đặt vé, hiển thị danh sách đơn đặt, xem chi tiết hành khách, cập nhật trạng thái đơn (đang xử lý, xác nhận, hủy), và hỗ trợ tính toán tổng tiền.
* Quản lý hành khách: Cung cấp các API để lưu và hiển thị thông tin hành khách theo từng đơn đặt vé, bao gồm tên, giới tính, ngày sinh và quốc tịch.
* Quản lý người dùng: Hỗ trợ đăng ký, đăng nhập, phân quyền (người dùng thường, quản trị viên), và hiển thị thông tin tài khoản người dùng. QTV có thể sửa, xóa hoặc khóa tài khoản nếu cần.
* Quản lý đề xuất vé: Cho phép QTV tạo, chỉnh sửa hoặc xóa các đề xuất vé nổi bật trên trang chính, với thông tin thành phố đi, thành phố đến, giá vé, hình ảnh và thứ tự hiển thị.
* Phân tích và báo cáo: Tích hợp API thống kê doanh thu, số lượng vé bán theo trạng thái, tuyến bay phổ biến và biểu đồ theo từng khoảng thời gian (ngày, tháng, năm), phục vụ cho việc theo dõi hiệu quả kinh doanh.

## 

## PHẦN 5 – TRIỂN KHAI HỆ THỐNG

Hệ thống web bán vé máy bay được triển khai trên nền tảng AWS Cloud nhằm đảm bảo khả năng mở rộng, tính sẵn sàng cao và dễ dàng quản lý. Quá trình triển khai gồm các bước chính như sau:

### 5.1 Đóng gói hệ thống bằng Docker

Frontend và Backend được đóng gói thành hai Docker container riêng biệt:

* Frontend: Dự án React được build bằng lệnh npm run build, sau đó đóng gói thành Docker image với Nginx để phục vụ nội dung tĩnh.
* Backend: Dự án Node.js Express sử dụng MySQL driver, đóng gói thành Docker image với các cấu hình kết nối đến RDS (qua biến môi trường).

### 5.2 Cấu hình hạ tầng mạng (VPC)

Tạo một Virtual Private Cloud (VPC) mới để quản lý toàn bộ tài nguyên mạng của hệ thống:

Cấu hình subnet gồm:

* 2 Private Subnet: Dùng để triển khai backend và RDS (không truy cập trực tiếp từ internet).
* 1 Public Subnet: Dùng để triển khai frontend và cho phép truy cập từ bên ngoài.

Tạo Route Table:

* Gán route table mặc định cho các public subnet để cho phép truy cập internet.
* Tạo Internet Gateway, liên kết với VPC và gắn vào public route table.

### 5.3 Cấu hình cơ sở dữ liệu

Sử dụng Amazon Aurora (MySQL-compatible) để tạo Cơ sở dữ liệu quan hệ (RDS):

* Đặt instance vào private subnet để bảo mật.
* Kết nối backend với RDS thông qua endpoint riêng, sử dụng cổng mặc định MySQL (3306).

Tạo Security Groups:

* SG-DB: Cho phép truy cập từ private subnet (nơi backend chạy).
* SG-Backend: Chỉ chấp nhận kết nối từ frontend hoặc các subnet nội bộ.
* SG-Frontend: Cho phép truy cập HTTP từ Internet thông qua port 8080.

### 5.4 Cài đặt và triển khai trên EC2

Khởi tạo EC2 instance (Amazon Linux 2) trong:

* Private Subnet để chạy container backend.
* Public Subnet để chạy container frontend.

Các bước thực hiện trên EC2:

1. Cài đặt Docker và Docker Compose.
2. Kéo image frontend/backend từ Docker Hub hoặc build trực tiếp từ source code.
3. Chạy các container với cấu hình:

3.1. Backend chạy trên port 3000, liên kết tới RDS bằng địa chỉ nội bộ.

3.2. Frontend chạy trên port 8080, proxy các yêu cầu API đến backend qua địa chỉ nội bộ trong subnet.

1. Mở cổng 8080 trong security group của frontend để cho phép người dùng truy cập website.

## 

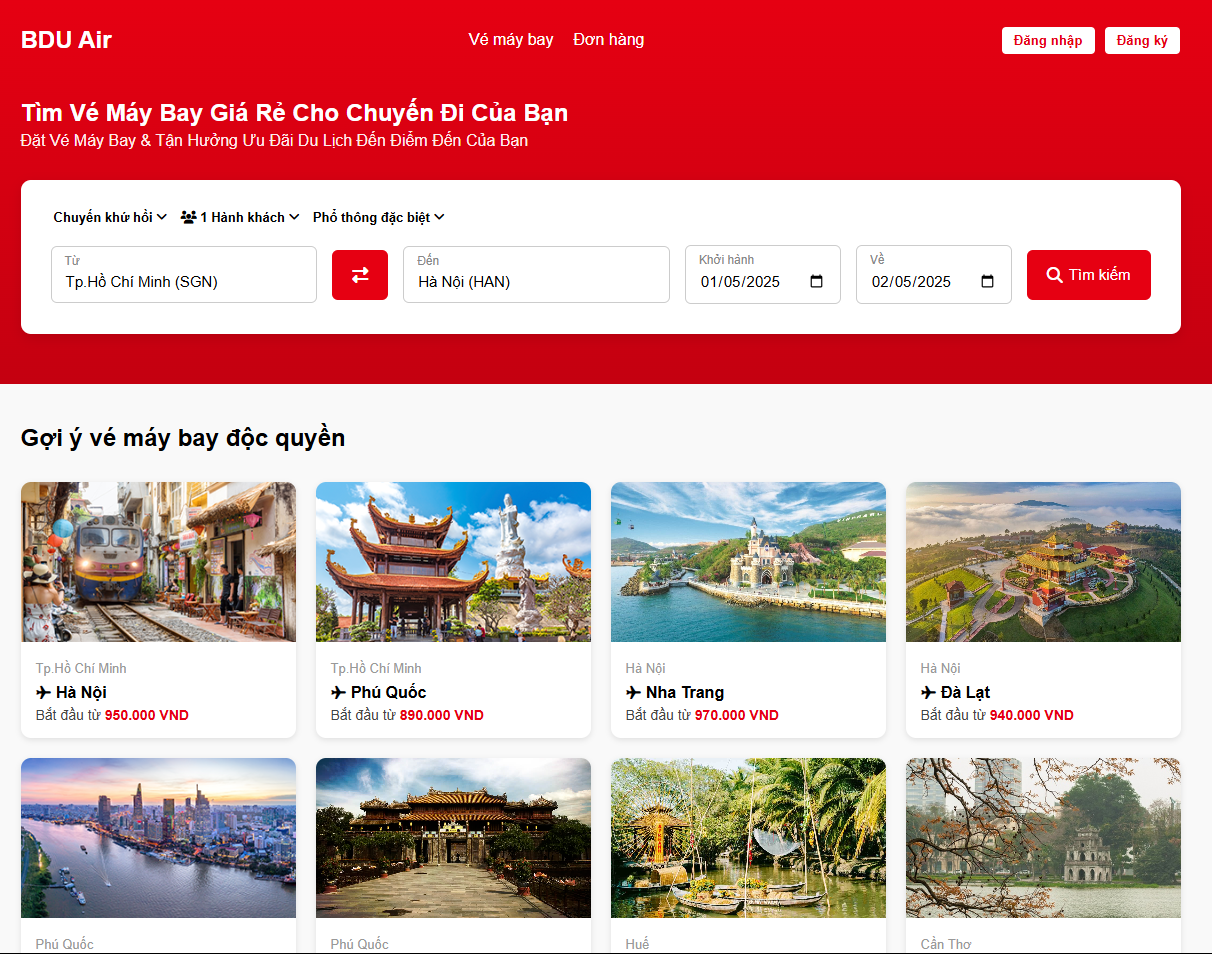
## PHẦN 6 – KẾT QUẢ ĐẠT ĐƯỢC VÀ HƯỚNG PHÁT TRIỂN

### 6.1 Kết quả đạt được

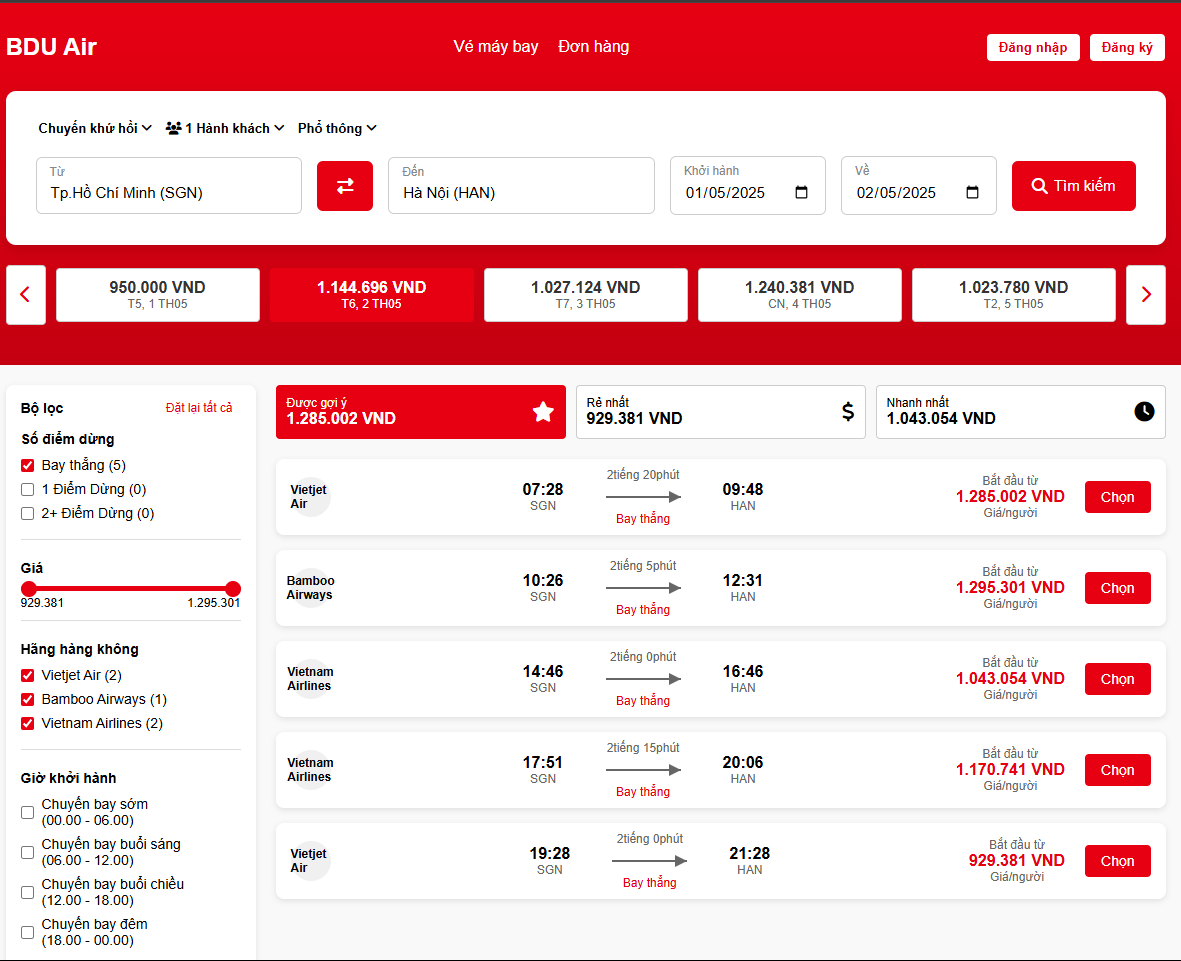
Hệ thống web bán vé máy bay đã được xây dựng và triển khai hoàn chỉnh, tích hợp đầy đủ các thành phần cốt lõi nhằm hỗ trợ hiệu quả cho việc đặt vé, quản lý chuyến bay và vận hành tổng thể của hệ thống, bao gồm:

* Backend: Phát triển bằng Nodejs và Express, xử lý logic nghiệp vụ chính xác và hiệu quả qua các module quản lý như chuyến bay, vé máy bay, tài khoản người dùng, đơn đặt vé và thống kê. Các API được thiết kế rõ ràng, bảo mật, hỗ trợ xác thực và phân quyền người dùng, đồng thời đảm bảo tính toàn vẹn dữ liệu khi thao tác với cơ sở dữ liệu.
* Frontend: Xây dựng bằng HTML, CSS và JavaScript (React), giao diện thân thiện và dễ sử dụng, cho phép người dùng đặt vé, tra cứu chuyến bay, đăng nhập, đăng ký và quản lý thông tin cá nhân. Đối với quản trị viên, giao diện hỗ trợ thêm/sửa/xóa chuyến bay, quản lý người dùng, điều chỉnh hệ thống đề xuất vé và theo dõi báo cáo doanh thu.
* Cơ sở dữ liệu: Sử dụng Amazon Aurora (MySQL-compatible) triển khai trên AWS RDS, với các bảng chính như Flight, Ticket, Booking, User và RevenueReport. Cơ sở dữ liệu được bảo mật thông qua hệ thống mạng riêng (private subnet), hỗ trợ truy vấn hiệu quả từ backend và đảm bảo tính nhất quán dữ liệu trong toàn hệ thống.

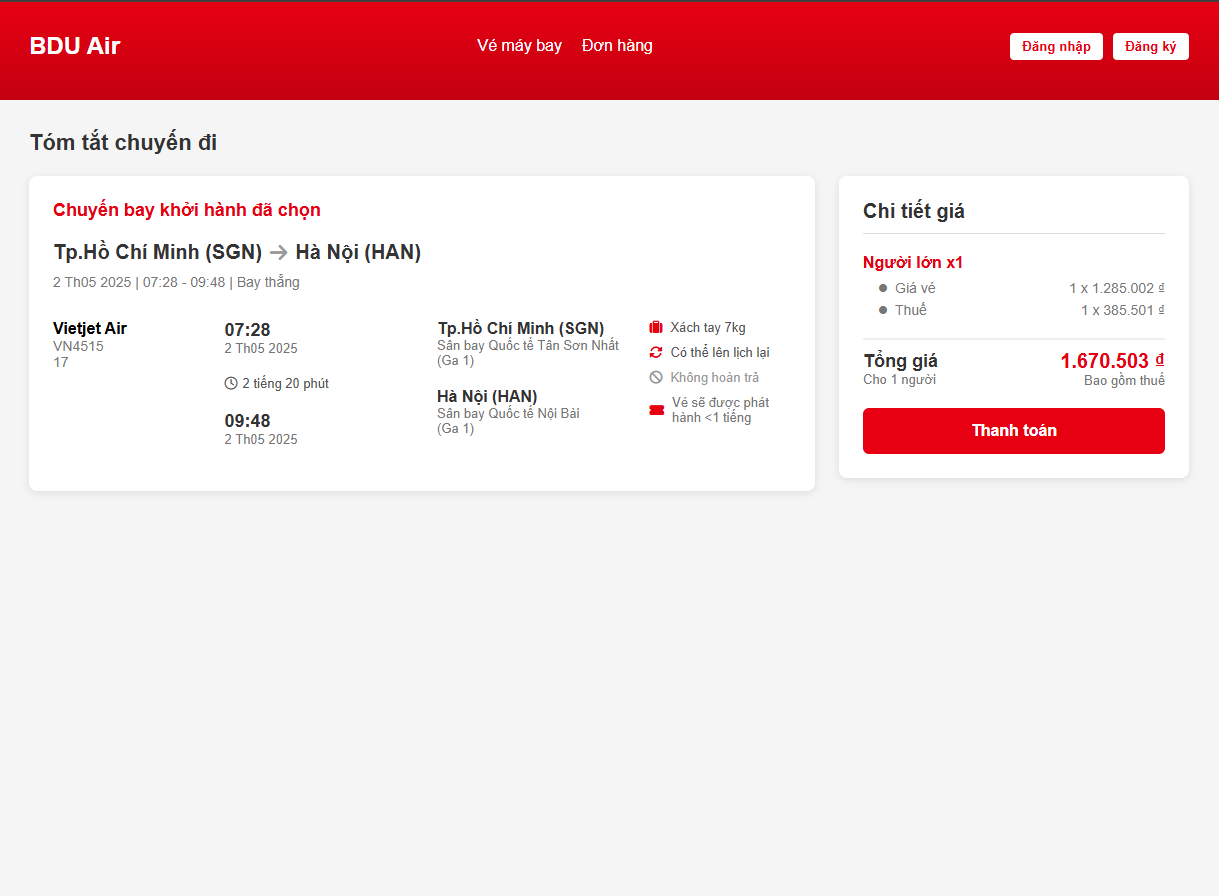
Tổng thể, hệ thống đã đáp ứng tốt các chức năng thiết yếu của một nền tảng đặt vé máy bay trực tuyến. Dưới đây là một số hình ảnh minh họa cho các kết quả đã đạt được:



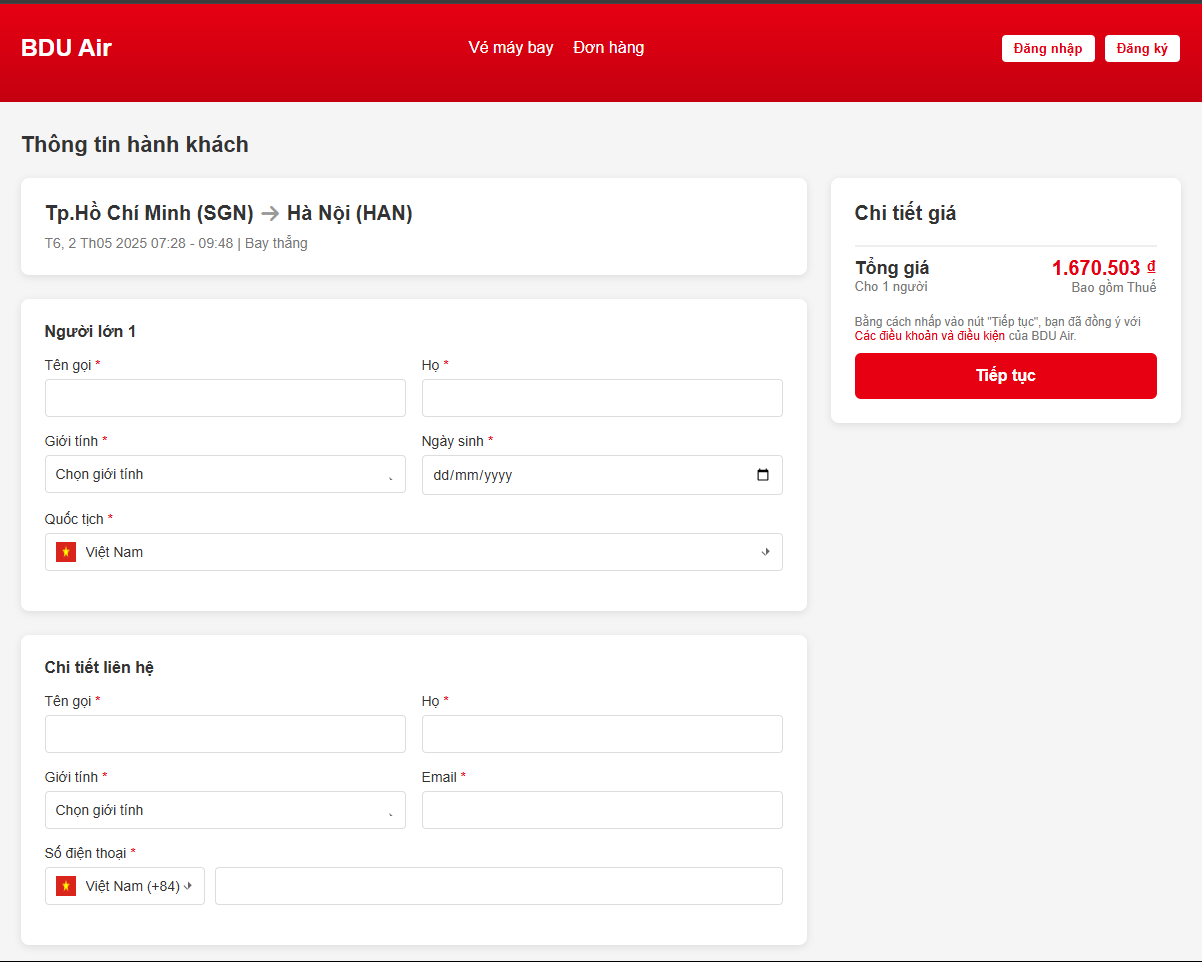
*Hình 6.1 Giao diện trang chủ của trang web.*

**

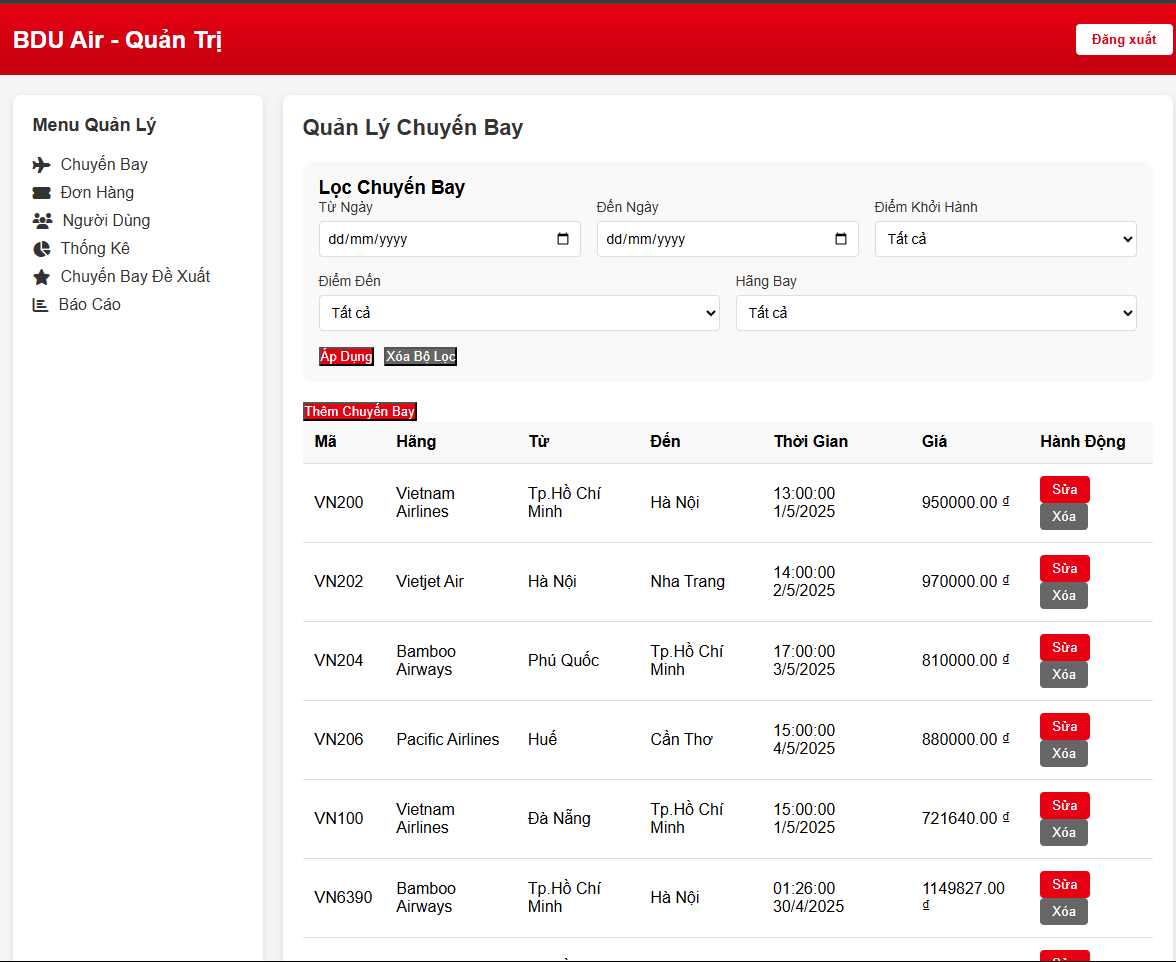
*Hình 6.2 Chức năng tìm vé của trang web.*

**

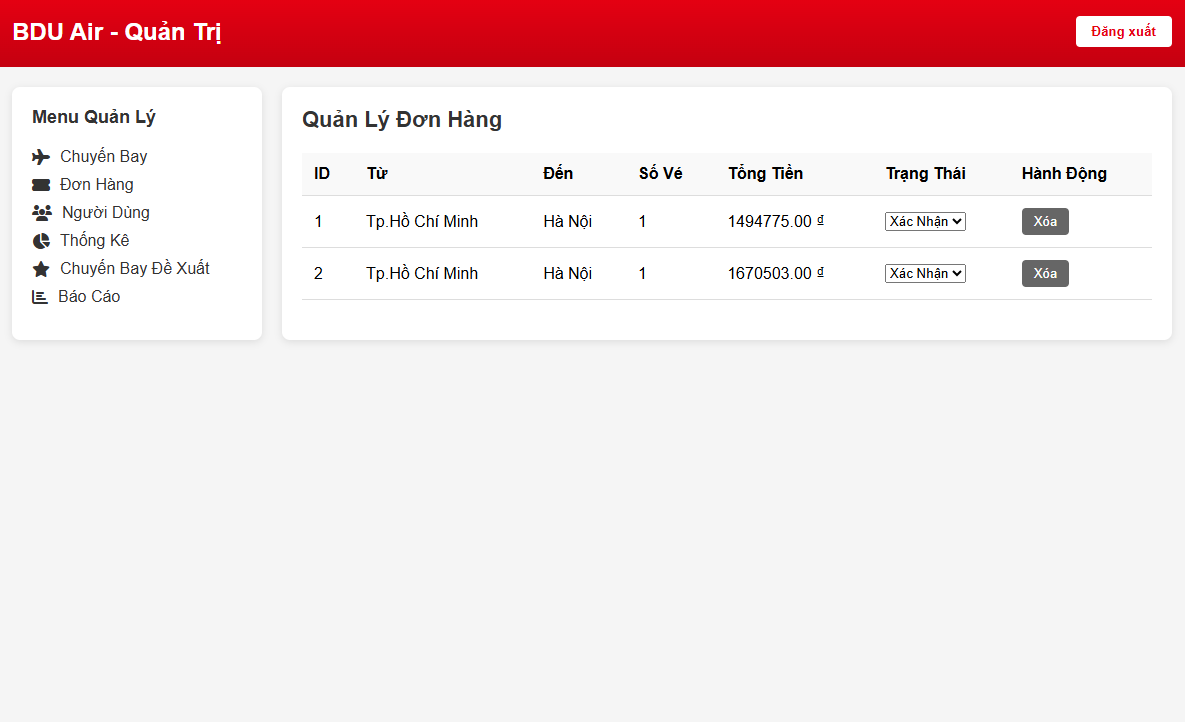
*Hình 6.3 Chức năng chọn vé của trang web.*

**

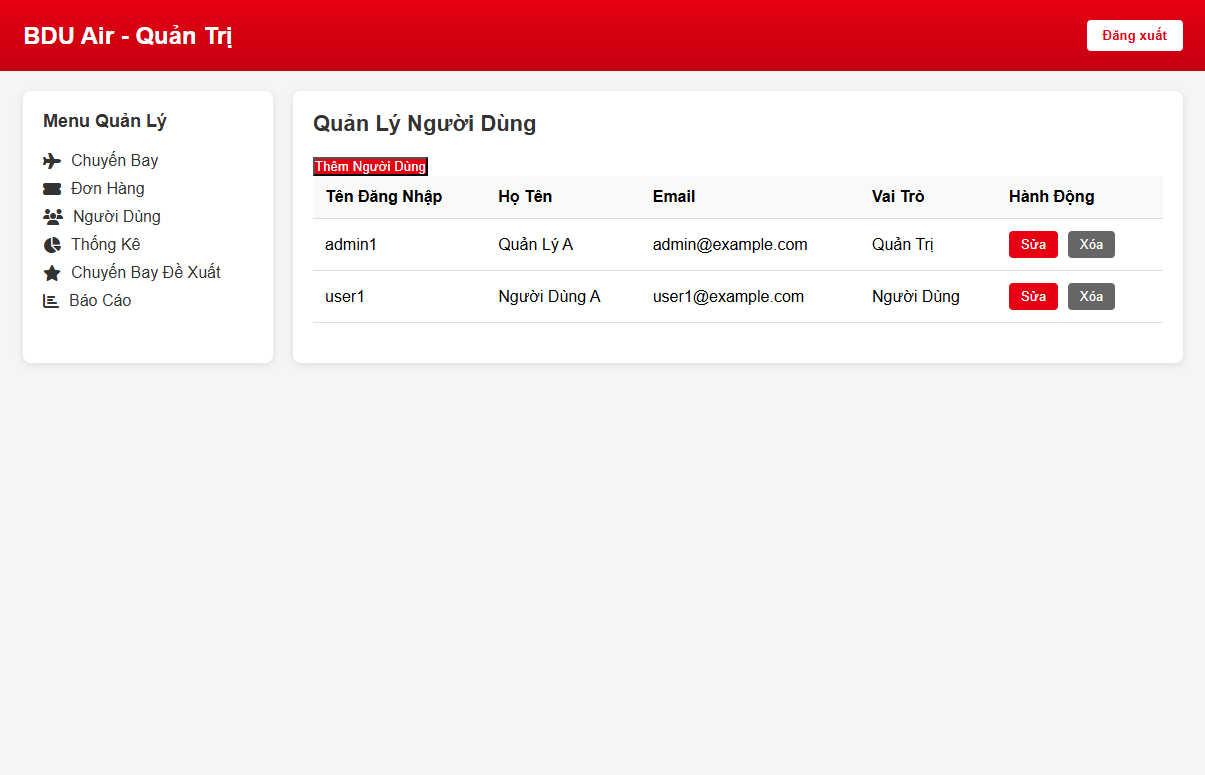
*Hình 6.4 Chức năng thanh toán của trang web.*

**

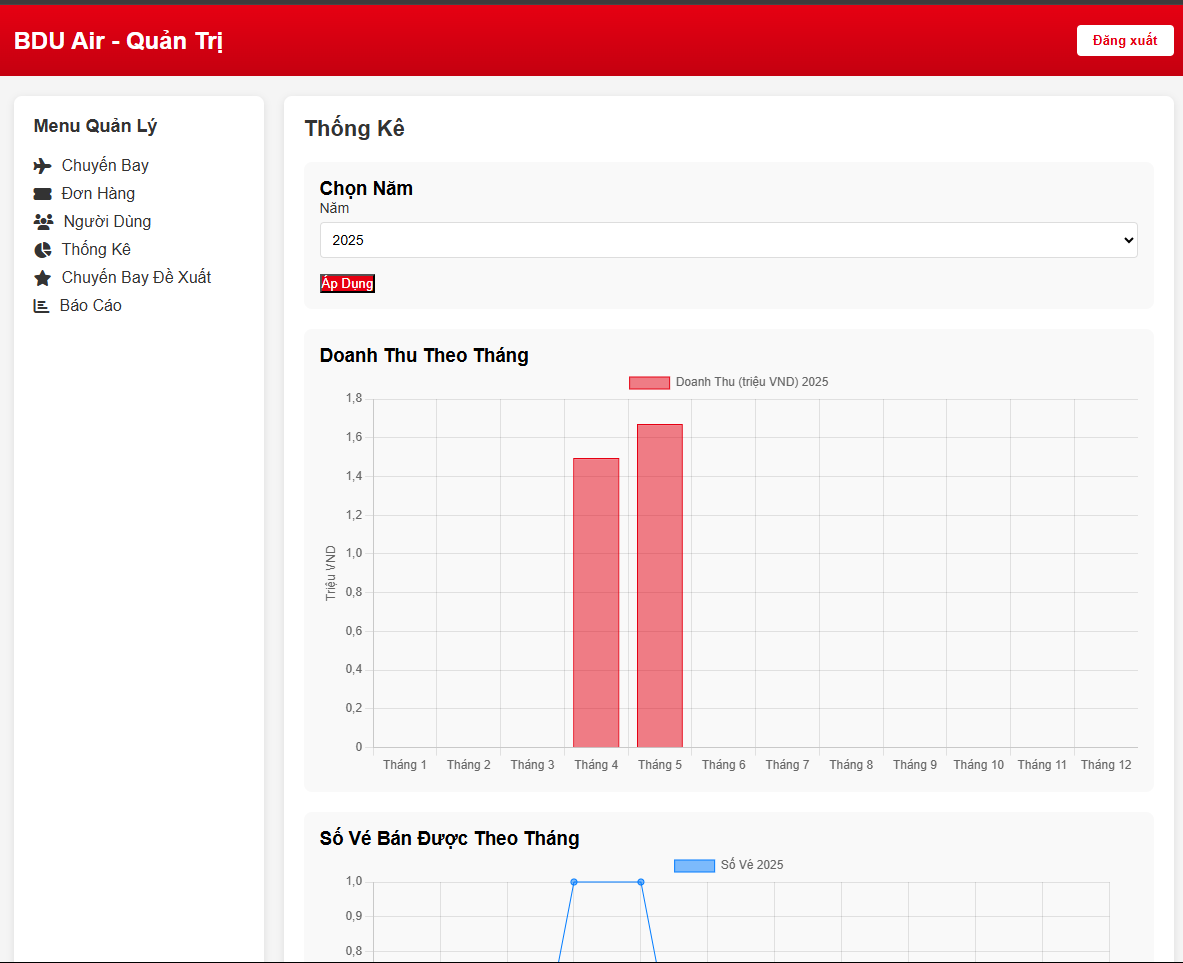
*Hình 6.5 Chức năng quản lý chuyến bay của trang web.*

**

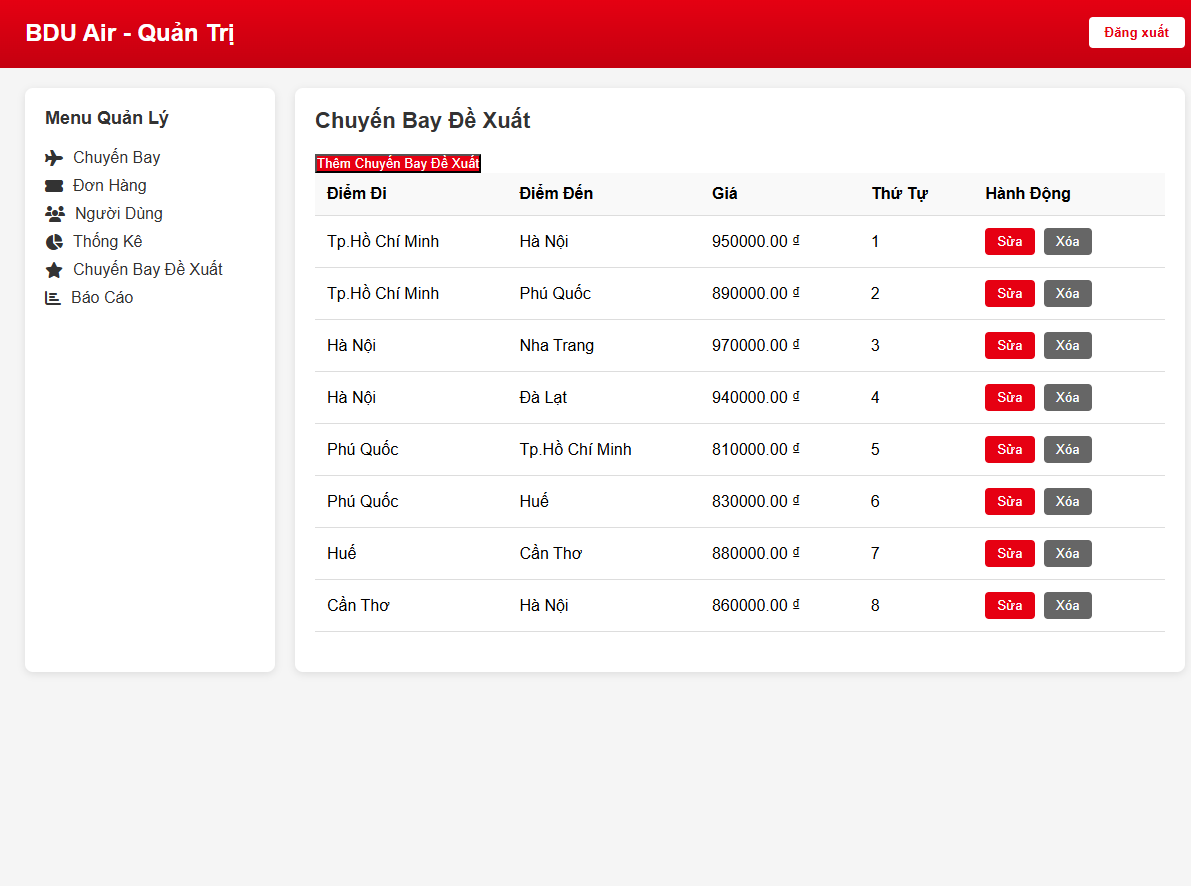
*Hình 6.6 Chức năng quản lý đơn hàng của trang web.*

**

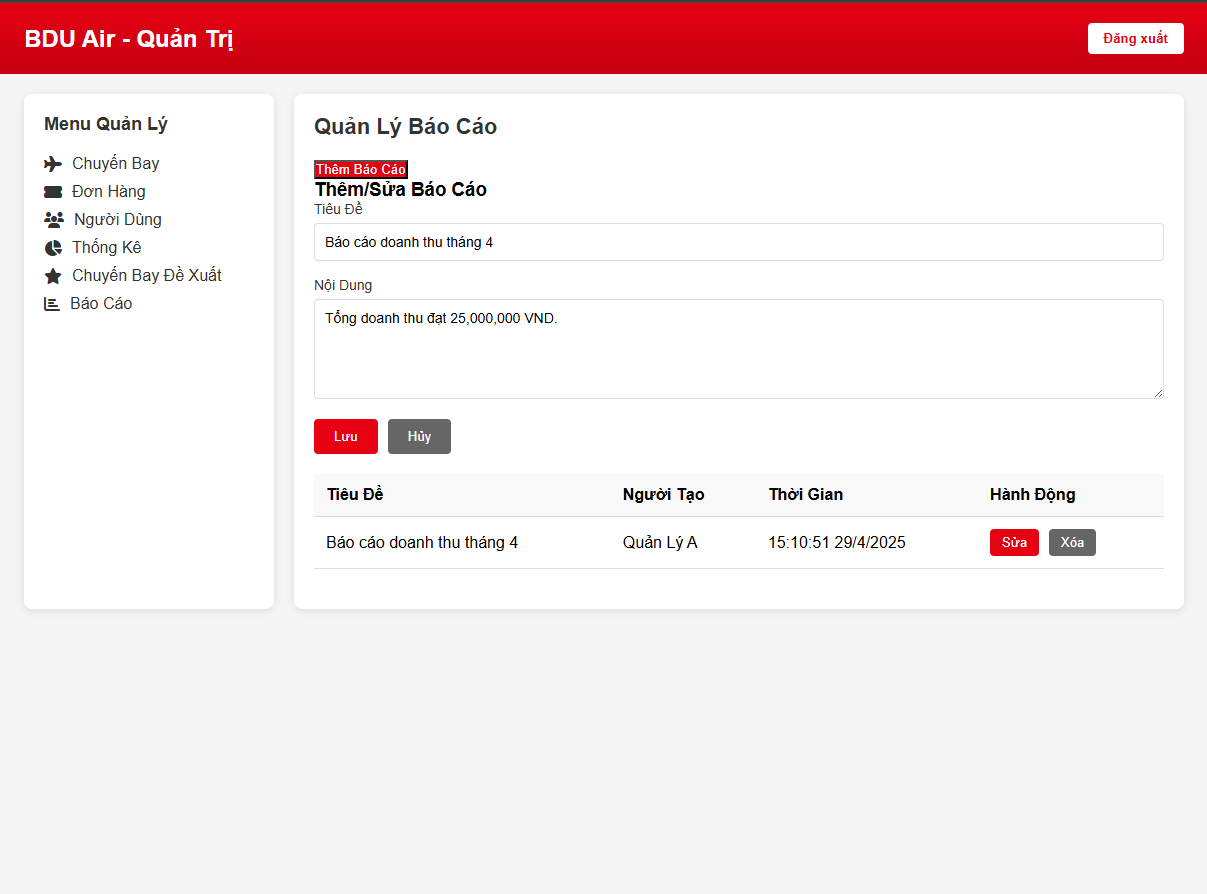
*Hình 6.7 Chức năng quản lý người dùng của trang web.*

**

*Hình 6.8 Chức năng thống kê doanh thu của trang web.*

**

*Hình 6.9 Chức năng quản lý các chuyến bay đề xuất của trang web.*

**

*Hình 6.10 Chức năng quản lý báo cáo của trang web.*

### 

### 6.2 Hướng phát triển

Mặc dù hệ thống đã vận hành ổn định và đáp ứng tốt các chức năng cơ bản, vẫn còn nhiều tiềm năng để mở rộng và cải tiến trong tương lai. Một số hướng phát triển cụ thể như sau:

* Tối ưu hiệu suất: Cải thiện logic xử lý ở backend, tối ưu truy vấn cơ sở dữ liệu và bộ nhớ đệm (caching) để tăng tốc độ phản hồi khi tra cứu chuyến bay hoặc đặt vé.
* Cải thiện trải nghiệm người dùng: Nâng cấp giao diện người dùng (UI/UX) để hỗ trợ tốt hơn trên thiết bị di động, đồng thời tinh chỉnh luồng đặt vé và tra cứu chuyến bay để dễ sử dụng hơn.
* Tăng cường bảo mật: Triển khai các biện pháp bảo mật nâng cao như xác thực đa yếu tố (2FA), mã hóa dữ liệu nhạy cảm, hạn chế truy cập theo vai trò, và giám sát hoạt động bất thường.
* Báo cáo và thống kê nâng cao: Phát triển hệ thống báo cáo doanh thu, lượt đặt vé theo tuyến bay, khung giờ cao điểm, và các chỉ số hiệu suất giúp quản trị viên có cái nhìn trực quan về tình hình kinh doanh.
* Khả năng mở rộng: Thiết kế và triển khai kiến trúc linh hoạt, hỗ trợ mở rộng hệ thống theo chiều ngang khi lượng người dùng và dữ liệu tăng cao, đảm bảo tính ổn định và khả năng xử lý đồng thời.

## TÀI LIỆU THAM KHẢO

**Tiếng Việt**

1. Quế, T. Đ., & Sơn, N. M. (2007). Phân tích thiết kế hệ thống thông tin.
2. UML: Unified Modeling Language (ngôn ngữ mô hình hóa thống nhất). (2023). Truy cập tại: <https://viblo.asia/p/uml-unified-modeling-language-ngon-ngu-mo-hinh-hoa-thong-nhat-pgjLNQGjJ32>
3. Tìm hiểu về mô hình 3 lớp (3 layer). (2020). Truy cập tại: <https://viblo.asia/p/tim-hieu-ve-mo-hinh-3-lop-3-layer-Qbq5Qmyz5D8>
4. Bản vẽ use case (use case diagram). (2014). Truy cập tại: <https://iviettech.vn/blog/543-ban-ve-use-case-use-case-diagram.html>
5. Biểu đồ lớp UML. (2020). Truy cập tại: <https://viblo.asia/p/bieu-do-lop-uml-Az45bDaVZxY>
6. Mọi thứ bạn cần biết về sơ đồ tuần tự. (2022). Truy cập tại: <https://blog.visual-paradigm.com/vn/everything-you-need-to-know-about-sequence-diagrams/>

**Tiếng Anh**

* 1. UML sequence diagram tutorial. (n.d.). Truy cập tại: <https://www.lucidchart.com/pages/uml-sequence-diagram>