



**BỘ MÔN: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**\*\*\***

**BÁO CÁO ĐỒ ÁN**

**-------**

**Đề tài: QUẢN LÝ BÁN VÉ MÁY BAY**

Mã môn học : **221DBMS330284**

Giáo viên hướng dẫn : **TS. Nguyễn Thành Sơn**

Sinh viên thực hiện : **Phan Thanh Tín**

TP. Hồ Chí Minh, tháng 11 năm 2023

MỤC LỤC

[**I.** **Đặc tả đề tài** 1](#_Toc151465688)

[**1.** **Mô tả bài toán** 1](#_Toc151465689)

[**2.** **Nghiệp vụ bài toán** 2](#_Toc151465690)

[**3.** **Chức năng** 4](#_Toc151465691)

[**II.** **Thiết kế CSDL** 7](#_Toc151465692)

[**1.** **Sơ đồ ERD** 7](#_Toc151465693)

[**2.** **Thiết kế cơ sở dữ liệu mức logic** 7](#_Toc151465694)

[**3.** **Các ràng buộc cần có** 8](#_Toc151465695)

[**4.** **Thành phần dữ liệu** 9](#_Toc151465696)

[**4.1.** **Tạo bảng và ràng buộc khóa chính** 9](#_Toc151465697)

[**4.2.** **Ràng buộc khóa ngoại** 11](#_Toc151465698)

[**4.3.** **Diagram** 13](#_Toc151465699)

[**4.4.** **Trigger** 13](#_Toc151465700)

[**III.** **Thiết kế các chức năng** 18](#_Toc151465701)

[**1.** **View** 18](#_Toc151465702)

[**2.** **Stored Procedure** 20](#_Toc151465703)

[**3.** **Function** 22](#_Toc151465704)

[**4.** **Transaction** 30](#_Toc151465705)

[**5.** **Kết nối CSDL** 33](#_Toc151465706)

[**IV.** **Tạo user và phân quyền** 34](#_Toc151465707)

[**V.** **Giao diện (Winform)** 37](#_Toc151465708)

[**1.** **Đăng nhập** 37](#_Toc151465709)

[**2.** **Hệ thống** 37](#_Toc151465710)

[**3.** **Bán vé** 38](#_Toc151465711)

[**4.** **Tra cứu chuyến bay** 38](#_Toc151465712)

[**5.** **Báo cáo doanh thu** 39](#_Toc151465713)

[**6.** **Quản lý chuyến bay** 39](#_Toc151465714)

[**7.** **Quản lý đơn giá** 40](#_Toc151465715)

[**8.** **Quản lý hạng vé** 40](#_Toc151465716)

[**9.** **Quản lý tuyến bay** 41](#_Toc151465717)

[**10.** **Quản lý sân bay** 41](#_Toc151465718)

[**11.** **Quản lý máy bay** 42](#_Toc151465719)

[**12.** **Quản lý nhân viên** 42](#_Toc151465720)

[**13.** **Quản lý khách hàng** 43](#_Toc151465721)

[**TÀI LIỆU THAM KHẢO** 44](#_Toc151465722)

1. **Đặc tả đề tài**
2. **Mô tả bài toán**

Hệ thống quản lý bán vé máy bay dành cho một chuỗi hãng hàng không cần được xây dựng để quản lý các hoạt động liên quan đến bán vé máy bay. Hệ thống này phải hỗ trợ hai đối tượng người dùng chính là nhân viên bán vé và quản lý, với các quyền truy cập khác nhau. Đối với hệ thống quản lý bán vé máy bay cần lưu trữ những dữ liệu sau:

Mỗi khách hàng có một mã để phân biệt với những người khác, có tên, CMND và số điện thoại để liên lạc. Khi đặt vé, mỗi vé sẽ chứa mã vé, mã khách hàng, mã chuyến bay, mã hạng vé và thông tin về việc nhân viên có mua vé. Mỗi khách hàng có thể đặt nhiều vé chuyến bay, nhưng mỗi vé chuyến bay chỉ thuộc về một khách hàng.

Mỗi nhân viên của hãng hàng không có một tên, địa chỉ, điện thoại và có một mã nhân viên duy nhất để phân biệt với các nhân viên khác. Có hai loại nhân viên là quản trị viên và bán vé. Một nhân viên có thể bán nhiều vé và một vé sẽ được bán bởi một nhân viên.

Mỗi tài khoản đăng nhập hệ thống sẽ có tên đăng nhập để phân biệt với những người khác, mật khẩu, mã nhân viên và loại tài khoản ứng với hai loại nhân viên. Mỗi nhân viên sẽ có một tài khoản và một tài khoản sẽ ứng với một nhân viên.

Mỗi chuyến bay sẽ có thông tin về chuyến bay, bao gồm mã chuyến bay, mã tuyến bay, mã máy bay, thời gian xuất phát và thời gian đến nơi. Thông tin về các tuyến bay sẽ bao gồm mã tuyến bay, sân bay xuất phát và sân bay đến. Mỗi chuyến bay thuộc về một tuyến bay cụ thể.

Mỗi máy bay có một mã máy bay để phân biệt với các máy bay, tên loại máy bay và số lượng ghế. Mỗi chuyến bay được thực hiện bởi một máy bay cụ thể, nhưng một máy bay có thể được sử dụng để thực hiện nhiều chuyến bay khác nhau.

Việc đặt vé sẽ bao gồm mã vé, mã hạng vé và thông tin về việc nhân viên có mua vé. Mỗi vé chuyến bay tương ứng với một hạng vé cụ thể, bao gồm mã hạng vé và tên hạng vé. Nhiều vé chuyến bay thuộc về một chuyến bay cụ thể.

Giá vé phụ thuộc vào các hạng vé và tuyến bay. Mỗi hạng vé sẽ có một giá vé cụ thể. Mỗi tuyến bay có thể có nhiều giá vé khác nhau.

Tình trạng vé sẽ lưu trữ thông tin trên các chuyến bay, bao gồm mã chuyến bay, mã hạng vé, tổng số ghế, số ghế trống. Một chuyến bay sẽ có nhiều tình trạng vé. Mỗi hạng vé sẽ có một tình trạng vé.

Mỗi sân bay sẽ mã sân bay để phân biệt với các sân bay khác, tên sân bay và tên thành phố của sân bay đó. Nhiều sân bay có thể tham gia vào nhiều tuyến bay và ngược lại, nhiều tuyến bay có thể liên quan đến nhiều sân bay.

Thông tin về giao dịch bán hàng, bao gồm tháng, năm, mã chuyến bay, số vé và số tiền bán vé. Mỗi giao dịch bán hàng liên quan đến một chuyến bay cụ thể.

1. **Nghiệp vụ bài toán**

Hệ thống quản lý bán vé máy bay là một ứng dụng phức tạp được thiết kế để quản lý các hoạt động liên quan đến việc bán và quản lý vé máy bay cho hãng hàng không hoặc đại lý bán vé. Dưới đây là một số nghiệp vụ quan trọng của hệ thống:

* Quản lý thông tin khách hàng:

Hệ thống cho phép thêm, sửa đổi và xóa thông tin của khách hàng, bao gồm tên, số CMND/CCCD, số điện thoại, địa chỉ và các thông tin liên hệ khác. Khách hàng có thể tạo tài khoản để theo dõi lịch sử đặt vé và cập nhật thông tin cá nhân.

* Đặt vé chuyến bay:

Khách hàng có thể tìm kiếm chuyến bay dựa trên điểm đi và điểm đến, ngày khởi hành, hạng vé, số lượng hành khách, và các yêu cầu khác. Hệ thống hiển thị thông tin về các chuyến bay có sẵn và giá vé tương ứng. Khách hàng có thể chọn chuyến bay, hạng vé, và thực hiện đặt vé.

* Quản lý giá vé:

Hệ thống tính toán tổng số tiền cần thanh toán dựa trên chọn lựa của khách hàng. Quản lý giá vé bao gồm cập nhật và hiển thị giá vé cho từng hạng vé trên từng chuyến bay và tuyến bay cụ thể.

* Quản lý thông tin chuyến bay:

Hệ thống lưu trữ thông tin về các chuyến bay, bao gồm lịch trình, sân bay đi và đến, số lượng ghế trống, và các thông tin khác. Cập nhật thời gian chuyến bay, sân bay thay đổi, và hủy chuyến bay (nếu cần).

* Quản lý hạng vé và hạng ghế:

Hệ thống cho phép quản lý các hạng vé khác nhau với giá cả và điều kiện khác nhau. Theo dõi và quản lý số lượng ghế trống trong từng hạng vé trên từng chuyến bay.

* Quản lý nhân viên:

Hệ thống lưu trữ thông tin về nhân viên, bao gồm tên, ngày sinh, giới tính, địa chỉ, số điện thoại, lươngvà thông tin khác. Quản lý quyền truy cập và tài khoản đăng nhập của nhân viên.

* Báo cáo và thống kê:

Hệ thống cung cấp các báo cáo và thống kê về doanh số bán hàng, lịch sử đặt vé, tỷ lệ sử dụng ghế trống, và các thông tin quản lý khác để hỗ trợ quản lý và ra quyết định.

* Quản lý tình trạng vé và giao dịch bán hàng:

Hệ thống theo dõi tình trạng vé, bao gồm số vé đã bán, số vé còn trống và các thông tin liên quan. Lưu trữ thông tin giao dịch bán hàng, bao gồm số vé bán, giá bán và các chi tiết thanh toán.

1. **Chức năng**

Để sử dụng phần mềm, người dùng cần đăng nhập vào hệ thống. Người dùng gồm có hai loại: dành cho nhân viên và quản lý.

**Đối với Nhân viên:**

**Bán vé máy bay:**

* + Mua vé: Nhân viên có thể thực hiện giao dịch mua vé cho khách hàng. Họ nhập thông tin khách hàng và chọn chuyến bay, hạng vé cùng với số lượng vé cần mua.
  + Hủy vé: Nếu cần, nhân viên có thể hủy vé cho khách hàng. Họ cần xác định vé cần hủy dựa trên thông tin giao dịch ban đầu.
  + Tìm kiếm: Nhân viên có khả năng tìm kiếm thông tin về chuyến bay, giá vé và tình trạng vé cho việc tư vấn và bán vé.

**Tra cứu chuyến bay:**

* + Tìm kiếm: Nhân viên có thể tra cứu thông tin về chuyến bay, bao gồm lịch trình, sân bay đi và đến, số ghế trống và các chi tiết liên quan để cung cấp cho khách hàng.

**Đối với Quản lý:**

**Bán vé máy bay:**

* Mua vé: Quản lý có thể thực hiện các giao dịch mua vé giống như nhân viên để hỗ trợ khách hàng hoặc xem lịch sử giao dịch.
* Hủy vé: Quản lý có thể thực hiện việc hủy vé cho khách hàng và xem thông tin chi tiết về các giao dịch hủy vé.
* Tìm kiếm: Tương tự như nhân viên, quản lý có khả năng tìm kiếm thông tin về chuyến bay và vé.

**Tra cứu chuyến bay:**

* Tìm kiếm: Quản lý có thể tìm kiếm thông tin về chuyến bay để cung cấp cho khách hàng hoặc theo dõi lịch trình chuyến bay.

**Báo cáo:**

* Xem báo cáo: Quản lý có quyền truy cập vào các báo cáo về doanh số bán hàng, lịch sử đặt vé, tỷ lệ sử dụng ghế trống và các thống kê khác để đánh giá hiệu suất kinh doanh.

**Quản lý chuyến bay:**

* Thêm, xóa, sửa: Quản lý có thể thêm mới, xóa hoặc chỉnh sửa thông tin về các chuyến bay, bao gồm lịch trình, sân bay đi và đến, số lượng ghế trống và các chi tiết khác.
* Tìm kiếm: Quản lý có khả năng tìm kiếm thông tin về chuyến bay để quản lý và điều chỉnh các thông tin cụ thể.

**Quản lý đơn giá:**

* Thêm, xóa, sửa: Quản lý có quyền thêm mới, xóa hoặc chỉnh sửa các thông tin về đơn giá vé theo từng tuyến bay và hạng vé.
* Tìm kiếm: Quản lý có khả năng tìm kiếm thông tin về đơn giá vé để điều chỉnh và cập nhật.

**Quản lý hạng vé:**

* Thêm, xóa, sửa: Quản lý có thể quản lý thông tin về các hạng vé, bao gồm tên, mô tả và các điều kiện của hạng vé.
* Tìm kiếm: Quản lý có khả năng tìm kiếm thông tin về hạng vé để cập nhật và quản lý.

**Quản lý tuyến bay:**

* Thêm, xóa, sửa: Quản lý có thể quản lý thông tin về các tuyến bay, bao gồm điểm đi và điểm đến, thời gian bay và các chi tiết khác.
* Tìm kiếm: Quản lý có khả năng tìm kiếm thông tin về tuyến bay để điều chỉnh và cập nhật.

**Quản lý sân bay:**

* Thêm, xóa, sửa: Quản lý có quyền quản lý thông tin về các sân bay, bao gồm tên, địa chỉ và các thông tin liên quan khác.
* Tìm kiếm: Quản lý có khả năng tìm kiếm thông tin về sân bay để quản lý và cập nhật.

**Quản lý máy bay:**

* Thêm, xóa, sửa: Quản lý có thể quản lý thông tin về các máy bay, bao gồm tên, loại máy bay và các chi tiết liên quan khác.
* Tìm kiếm: Quản lý có khả năng tìm kiếm thông tin về máy bay để điều chỉnh và cập nhật.

**Quản lý nhân viên:**

* Thêm, xóa, sửa: Quản lý có thể thực hiện quản lý thông tin về nhân viên, bao gồm tên, ngày sinh, giới tính, địa chỉ, số điện thoại, mã công việc, lương, ngày tuyển dụng và các thông tin khác.
* Tìm kiếm: Quản lý có khả năng tìm kiếm thông tin về nhân viên để quản lý và cập nhật.

1. **Thiết kế CSDL**
2. **Sơ đồ ERD**

**A computer screen shot of a diagram

Description automatically generated**

1. **Thiết kế cơ sở dữ liệu mức logic**

Từ sơ đồ thực thể kết hợp (ERD), ta có các lược đồ quan hệ:

* Customers(idCus, nameCus, identitycardCus, phoneCus)
* Staffs(idStaffs, nameStaffs, addressStaffs, phoneStaffs)
* Accounts(userAcc, passAcc, idStaffs, typeAcc)
* Planes(idPlane, namePlane, seatsPlane)
* Airports(idAirport, nameAirport, cityAirport)
* FlightRoutes(idFlightRoutes, idAirportToGo, idAirportToCome)
* TicketClasses(idTicketClass, nameTicketClass)
* Prices(idFlightRoutes, idTicketClass, unitPrice)
* Flights(idFlights, idFlightRoutes, idPlane, timeToGo, timeToCome)
* TicketFlights(idTicket, idCus, idFlights, idTicketClass, ifStaffs)
* TicketStatus(idFlights, idTicketClass, totalSeats, emptySeats)
* Sales(monthSales, yearSales, idFlights, countTicket, sale)

1. **Các ràng buộc cần có**

|  |  |  |
| --- | --- | --- |
| STT | BẢNG | RÀNG BUỘC |
| 1 | Customers | Khóa chính idCus,  NOT NULL nameCus,  NOT NULL identitycardCus,  NOT NULL phoneCus,  CHECK phoneCus(=10) |
| 2 | Staffs | Khóa chính idStaffs,  NOT NULL nameStaffs,  NOT NULL addressStaffs,  NOT NULL phoneStaffs,  CHECK phoneStaffs(=10) |
| 3 | Accounts | Khóa chính userAcc,  Khóa ngoại idStaffs đến bảng Staffs  NOT NULL passAcc,  NOT NULL idStaffs,  NOT NULL typeAcc,  CHECK typeAcc(1 hoặc 0) |
| 4 | Planes | Khóa chính idPlane,  NOT NULL namePlane,  NOT NULL seatsPlane |
| 5 | Airports | Khóa chính idAirport,  NOT NULL nameAirport,  NOT NULL cityAirport |
| 6 | FlightRoutes | Khóa chính idFlightRoutes,  Khóa ngoại idAirportToGo đến idAirport bảng Airports,  Khóa ngoại idAirportToCome đến idAirport bảng Airports,  NOT NULL idAirportToGo,  NOT NULL idAirportToCome |
| 7 | TicketClasses | Khóa chính idTicketClass,  NOT NULL nameTicketClass |
| 8 | Prices | Khóa chính (idFlightRoutes, idTicketClass),  Khóa ngoại idTicketClass đến bảng TicketClasses,  Khóa ngoại idFlightRoutes đến bảng FlightRoutes,  NOT NULL unitPrice |
| 9 | Flights | Khóa chính idFlights,  Khóa ngoại idFlightRoutes đến bảng FlightRoutes,  Khóa ngoại idPlane đến bảng Planes,  NOT NULL idFlightRoutes,  NOT NULL idPlane,  NOT NULL timeToGo,  NOT NULL timeToCome |
| 10 | TicketFlights | Khóa chính idTicket,  Khóa ngoại idCus đến bảng Customers,  Khóa ngoại idFlights đến bảng Flights,  Khóa ngoại idTicketClass đến bảng TicketClasses,  Khóa ngoại idStaffs đến bảng Staffs  NOT NULL idCus,  NOT NULL idFlights,  NOT NULL idTicketClass,  NOT NULL idStaffs |
| 11 | TicketStatus | Khóa chính (idFlights, id TicketClass),  Khóa ngoại idFlights đến bảng Flights,  Khóa ngoại idTicketClass đến bảng TicketClasses,  NOT NULL totalSeats,  NOT NULL emptySeats,  CHECK totalSeats(>0) |
| 12 | Sales | Khóa chính(monthSales, yearSales, idFlights),  Khóa ngoại idFlights đến bảng Flights,  NOT NULL idFlights,  NOT NULL countTicket,  NOT NULL sale |

1. **Thành phần dữ liệu**
   1. **Tạo bảng và ràng buộc khóa chính**

**Customers**

CREATE TABLE Customers(

idCus CHAR(10) CONSTRAINT PK\_idCus PRIMARY KEY,

nameCus NVARCHAR(50) NOT NULL,

identitycardCus VARCHAR(12) NOT NULL,

phoneCus VARCHAR(10) NOT NULL

)

**Staffs**

CREATE TABLE Staffs(

idStaffs CHAR(10) CONSTRAINT PK\_idStaffs PRIMARY KEY,

nameStaffs NVARCHAR(50) NOT NULL,

addressStaffs NVARCHAR(50) NOT NULL,

phoneStaffs VARCHAR(10) NOT NULL

)

**Accounts**

CREATE TABLE Accounts(

userAcc CHAR(100) CONSTRAINT PK\_userAcc PRIMARY KEY,

passAcc CHAR(100) NOT NULL,

idStaffs CHAR(10) NOT NULL,

typeAcc INT NOT NULL -- 1 là admin, 0 là nhân viên

)

**Planes**

CREATE TABLE Planes(

idPlane CHAR(10) CONSTRAINT PK\_idPlane PRIMARY KEY,

namePlane NVARCHAR(50) NOT NULL,

seatsPlane INT NOT NULL

)

**Airports**

CREATE TABLE Airports(

idAirport CHAR(10) CONSTRAINT PK\_idAirport PRIMARY KEY,

nameAirport NVARCHAR(50) NOT NULL,

cityAirport NVARCHAR(50) NOT NULL

)

**FlightRoutes**

CREATE TABLE FlightRoutes(

idFlightRoutes CHAR(10) CONSTRAINT PK\_idFlightRoutes PRIMARY KEY,

idAirportToGo CHAR(10) NOT NULL,

idAirportToCome CHAR(10) NOT NULL,

)

**TicketClasses**

CREATE TABLE TicketClasses(

idTicketClass CHAR(10) CONSTRAINT PK\_idTicketClass PRIMARY KEY,

nameTicketClass NVARCHAR(50) NOT NULL

)

**Prices**

CREATE TABLE Prices(

idFlightRoutes CHAR(10),

idTicketClass CHAR(10),

unitPrice DECIMAL(18, 0) NOT NULL,

CONSTRAINT PK\_Prices PRIMARY KEY (idFlightRoutes, idTicketClass)

)

**Flights**

CREATE TABLE Flights(

idFlights CHAR(10) CONSTRAINT PK\_idFlight PRIMARY KEY,

idFlightRoutes CHAR(10) NOT NULL,

idPlane CHAR(10) NOT NULL,

timeToGo DATETIME NOT NULL,

timeToCome DATETIME NOT NULL

)

**TicketFlights**

CREATE TABLE TicketFlights(

idTicket CHAR(10) CONSTRAINT PK\_idTicket PRIMARY KEY,

idCus CHAR(10) NOT NULL,

idFlights CHAR(10) NOT NULL,

idTicketClass CHAR(10) NOT NULL,

idStaffs CHAR(10) NOT NULL

)

**TicketStatus**

CREATE TABLE TicketStatus(

idFlights CHAR(10),

idTicketClass CHAR(10),

totalSeats INT NOT NULL,

emptySeats INT NOT NULL,

CONSTRAINT PK\_TicketStatus PRIMARY KEY (idFlights, idTicketClass)

)

**Sales**

CREATE TABLE Sales(

monthSales INT,

yearSales INT,

idFlights CHAR(10) NOT NULL,

countTicket INT NOT NULL,

sale DECIMAL(18, 0) NOT NULL,

CONSTRAINT PK\_Sales PRIMARY KEY (monthSales, yearSales, idFlights)

)

* 1. **Ràng buộc khóa ngoại**

**--Accounts**

ALTER TABLE dbo.Accounts ADD CONSTRAINT FK\_Accounts\_idStaffs FOREIGN KEY(idStaffs) REFERENCES dbo.Staffs(idStaffs)

**--FlightRoutes**

ALTER TABLE dbo.FlightRoutes ADD CONSTRAINT FK\_FlightRoutes\_idAirportToGo FOREIGN KEY(idAirportToGo) REFERENCES dbo.Airports(idAirport)

ALTER TABLE dbo.FlightRoutes ADD CONSTRAINT FK\_FlightRoutes\_idAirportToCome FOREIGN KEY(idAirportToCome) REFERENCES dbo.Airports(idAirport)

**--Prices**

ALTER TABLE dbo.Prices ADD CONSTRAINT FK\_Prices\_idTicketClass FOREIGN KEY(idTicketClass) REFERENCES dbo.TicketClasses(idTicketClass)

ALTER TABLE dbo.Prices ADD CONSTRAINT FK\_Prices\_idFlightRoutes FOREIGN KEY(idFlightRoutes) REFERENCES dbo.FlightRoutes(idFlightRoutes)

**--Flights**

ALTER TABLE dbo.Flights ADD CONSTRAINT FK\_Flights\_idFlightRoutes FOREIGN KEY(idFlightRoutes) REFERENCES dbo.FlightRoutes(idFlightRoutes)

ALTER TABLE dbo.Flights ADD CONSTRAINT

FK\_Flights\_idPlane FOREIGN KEY(idPlane) REFERENCES dbo.Planes(idPlane)

**--TicketStatus**

ALTER TABLE dbo.TicketStatus ADD CONSTRAINT FK\_TicketStatus\_idFlights FOREIGN KEY(idFlights) REFERENCES dbo.Flights(idFlights)

ALTER TABLE dbo.TicketStatus ADD CONSTRAINT FK\_TicketStatus\_idTicketClass FOREIGN KEY(idTicketClass) REFERENCES dbo.TicketClasses(idTicketClass)

**--TicketFlights**

ALTER TABLE dbo.TicketFlights ADD CONSTRAINT FK\_TicketFlights\_idCus FOREIGN KEY(idCus) REFERENCES dbo.Customers(idCus)

ALTER TABLE dbo.TicketFlights ADD CONSTRAINT FK\_TicketFlights\_idFlights FOREIGN KEY(idFlights) REFERENCES dbo.Flights(idFlights)

ALTER TABLE dbo.TicketFlights ADD CONSTRAINT FK\_TicketFlights\_idTicketClass FOREIGN KEY(idTicketClass) REFERENCES dbo.TicketClasses(idTicketClass)

ALTER TABLE dbo.TicketFlights ADD CONSTRAINT FK\_TicketFlights\_idStaffs FOREIGN KEY(idStaffs) REFERENCES dbo.Staffs(idStaffs)

**--Sales**

ALTER TABLE dbo.Sales ADD CONSTRAINT FK\_Sales\_idFlights FOREIGN KEY(idFlights) REFERENCES dbo.Flights(idFlights)

* 1. A computer screen shot of a computer

     Description automatically generated**Diagram**
  2. **Trigger**

**-- Trigger xóa tài khoản khi xóa nhân viên**

CREATE TRIGGER UTG\_DeleteAccount

ON dbo.Accounts AFTER DELETE

AS

BEGIN

DECLARE @id CHAR(10)

SELECT @id = Deleted.idStaffs FROM Deleted

DELETE dbo.Staffs WHERE idStaffs = @id

END

GO

**-- Trigger kiểm tra trùng lặp tên sân bay**

CREATE TRIGGER UTG\_CheckAirport

ON dbo.Airports AFTER INSERT, UPDATE

AS

BEGIN

DECLARE @nameAirport NVARCHAR(50), @count INT

SELECT @nameAirport = Inserted.nameAirport FROM Inserted

SELECT @count = COUNT(\*) FROM dbo.Airports WHERE nameAirport = @nameAirport

IF (@count > 1)

ROLLBACK TRAN

END

GO

**-- Trigger kiểm tra trùng lặp tên máy bay**

CREATE TRIGGER UTG\_CheckNamePlane

ON dbo.Planes AFTER INSERT, UPDATE

AS

BEGIN

DECLARE @namePlane NVARCHAR(50), @count INT

SELECT @namePlane = Inserted.namePlane FROM Inserted

SELECT @count = COUNT(\*) FROM dbo.Planes WHERE namePlane = @namePlane

IF (@count > 1)

ROLLBACK TRAN

END

GO

**-- Trigger kiểm tra trùng lặp tên hạng vé**

CREATE TRIGGER UTG\_CheckNameTịcketClass

ON dbo.TicketClasses AFTER INSERT, UPDATE

AS

BEGIN

DECLARE @nameTC NVARCHAR(50), @count INT

SELECT @nameTC = Inserted.nameTicketClass FROM Inserted

SELECT @count = COUNT(\*) FROM dbo.TicketClasses WHERE nameTicketClass = @nameTC

IF (@count > 1)

ROLLBACK TRAN

END

GO

**-- Trigger thêm danh sách doanh thu khi thêm, sửa chuyến bay**

CREATE TRIGGER UTG\_AddSaleFromFlight

ON dbo.Flights AFTER INSERT, UPDATE

AS

BEGIN

DECLARE @timetogonew DATETIME, @timetocomenew DATETIME, @monthnew INT, @yearnew INT, @timetogoold DATETIME, @timetocomeold DATETIME, @monthold INT, @yearold INT, @idFlightnew CHAR(10), @idFlightold CHAR(10)

SELECT @timetogonew = Inserted.timeToGo, @timetocomenew = Inserted.timeToCome, @idFlightnew = Inserted.idFlights FROM Inserted

SELECT @timetogoold = Deleted.timeToGo, @timetocomeold = Deleted.timeToCome, @idFlightold = Deleted.idFlights FROM Deleted

SET @monthnew = (SELECT MONTH(@timetogonew))

SET @yearnew = (SELECT YEAR(@timetogonew))

SET @monthold = (SELECT MONTH(@timetogoold))

SET @yearold = (SELECT YEAR(@timetogoold))

BEGIN TRAN

IF (@timetogonew < (SELECT GETDATE()))

BEGIN

ROLLBACK

RETURN

END

IF (@timetogonew >= @timetocomenew)

BEGIN

ROLLBACK

RETURN

END

INSERT dbo.Sales(monthSales, yearSales, idFlights, countTicket, sale) VALUES (@monthnew, @yearnew, @idFlightnew, 0, 0)

DELETE dbo.Sales WHERE monthSales = @monthold AND yearSales = @yearold AND idFlights = @idFlightold

IF (@@ERROR <> 0)

BEGIN

ROLLBACK

RETURN

END

COMMIT TRAN

END

GO

**-- Trigger xóa danh sách doanh thu khi thêm, sửa chuyến bay**

CREATE TRIGGER UTG\_DeleteSaleFromFlight

ON dbo.Flights INSTEAD OF DELETE

AS

BEGIN

DECLARE @timetogoold DATETIME, @timetocomeold DATETIME, @monthold INT, @yearold INT, @idFlight CHAR(10)

SELECT @timetogoold = Deleted.timeToGo, @timetocomeold = Deleted.timeToCome, @idFlight = Deleted.idFlights FROM Deleted

SET @monthold = (SELECT MONTH(@timetogoold))

SET @yearold = (SELECT YEAR(@timetogoold))

BEGIN TRAN

DELETE dbo.Sales WHERE monthSales = @monthold AND yearSales = @yearold AND @idFlight = @idFlight

DELETE dbo.Flights WHERE idFlights = @idFlight

IF (@@ERROR <> 0)

BEGIN

ROLLBACK

RETURN

END

COMMIT TRAN

END

GO

**--Triger sửa doanh thu và sửa tình trạng vé khi bán vé**

CREATE TRIGGER UTG\_SaleAndTicketStatusBookingTicket

ON dbo.TicketFlights AFTER INSERT

AS

BEGIN

DECLARE @idFlight CHAR(10), @idTC CHAR(10), @timetogo DATETIME, @month INT, @year INT, @price DECIMAL(18, 0)

SELECT @idFlight = Inserted.idFlights, @idTC = Inserted.idTicketClass

FROM Inserted

SELECT @timetogo = timeToGo FROM dbo.Flights WHERE idFlights = @idFlight

SET @month = (SELECT MONTH(@timetogo))

SET @year = (SELECT YEAR(@timetogo))

SET @price = (SELECT dbo.UF\_GetPriceByIdFlightAndIdTicketClass(@idFlight, @idTC))

BEGIN TRAN

UPDATE dbo.TicketStatus SET emptySeats = emptySeats - 1

WHERE idFlights = @idFlight AND idTicketClass = @idTC

UPDATE dbo.Sales SET sale = sale + @price, countTicket = countTicket + 1

WHERE monthSales = @month AND yearSales = @year AND idFlights = @idFlight

IF (@@ERROR <> 0)

BEGIN

ROLLBACK

RETURN

END

COMMIT TRAN

END

GO

**-- Trigger sửa doanh thu và sửa tình trạng vé khi hủy vé**

CREATE TRIGGER UTG\_SaleAndTicketStatusCancelTicket

ON dbo.TicketFlights INSTEAD OF DELETE

AS

BEGIN

DECLARE @idFlight CHAR(10), @idTC CHAR(10), @idTicket CHAR(10),

@timetogo DATETIME, @month INT, @year INT, @price DECIMAL(18, 0)

SELECT @idFlight = Deleted.idFlights, @idTC = Deleted.idTicketClass, @idTicket = Deleted.idTicket

FROM Deleted

SELECT @timetogo = timeToGo FROM dbo.Flights WHERE idFlights = @idFlight

SET @month = (SELECT MONTH(@timetogo))

SET @year = (SELECT YEAR(@timetogo))

SET @price = (SELECT dbo.UF\_GetPriceByIdFlightAndIdTicketClass(@idFlight, @idTC))

BEGIN TRAN

UPDATE dbo.TicketStatus SET emptySeats = emptySeats + 1

WHERE idFlights = @idFlight AND idTicketClass = @idTC

UPDATE dbo.Sales SET sale = sale - @price, countTicket = countTicket - 1

WHERE monthSales = @month AND yearSales = @year AND idFlights = @idFlight

DELETE dbo.TicketFlights WHERE idTicket = @idTicket

IF (@@ERROR <> 0)

BEGIN

ROLLBACK

RETURN

END

COMMIT TRAN

END

GO

1. **Thiết kế các chức năng**
2. **View**

**-- View danh sách vé chuyến bay**

CREATE VIEW UV\_TicketFlightForDisplay

AS

SELECT TicketFlights.idTicket, nameCus, identitycardCus, phoneCus, idFlights, nameTicketClass, Prices.Price

FROM dbo.TicketFlights

INNER JOIN dbo.Customers

ON Customers.idCus = TicketFlights.idCus

INNER JOIN dbo.TicketClasses

ON TicketClasses.idTicketClass = TicketFlights.idTicketClass

INNER JOIN (SELECT dbo.UF\_GetPriceByIdFlightAndIdTicketClass(idFlights, idTicketClass) AS Price, idTicket FROM dbo.TicketFlights) AS Prices

ON Prices.idTicket = TicketFlights.idTicket

Trên C#:

A close-up of a computer screen

Description automatically generated

**-- View danh sách nhân viên**

CREATE VIEW UV\_StaffForDisplay

AS

SELECT Accounts.idStaffs, nameStaffs, addressStaffs, phoneStaffs, userAcc, typeAcc

FROM dbo.Accounts INNER JOIN dbo.Staffs ON Staffs.idStaffs = Accounts.idStaffs

Trên C#:

A computer screen shot of a computer code

Description automatically generated

**-- View danh sách tuyến bay**

CREATE VIEW UV\_FlightRouteForDisplay

AS

SELECT idFlightRoutes, AirportToGo.idAirport AS idAirportToGo, AirportToGo.nameAirport AS nameAirportToGo,

AirportToCome.idAirport AS idAirportToCome, AirportToCome.nameAirport AS nameAirportToCome

FROM dbo.FlightRoutes

INNER JOIN dbo.Airports AS AirportToGo

ON AirportToGo.idAirport = FlightRoutes.idAirportToGo

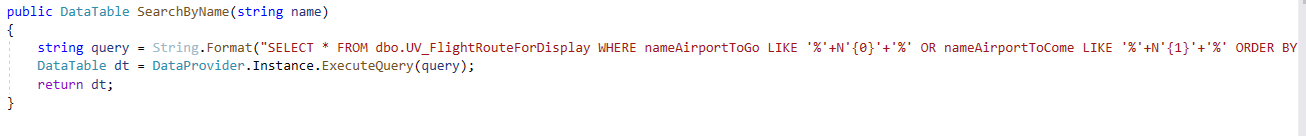
INNER JOIN dbo.Airports AS AirportToCome

ON AirportToCome.idAirport = FlightRoutes.idAirportToCome

Trên C#:

A computer screen shot of a code

Description automatically generated



**-- View danh sách doanh thu theo cá nhân nhân viên**

CREATE VIEW UV\_SaleByStaff

AS

SELECT TicketFlights.idStaffs, nameStaffs, phoneStaffs, COUNT(\*) AS countTicket, SUM(GetPrice.totalPrice) AS totalSale

FROM dbo.TicketFlights INNER JOIN

(SELECT idTicket, dbo.UF\_GetPriceByIdFlightAndIdTicketClass(idFlights, idTicketClass) AS totalPrice, idStaffs

FROM dbo.TicketFlights) AS GetPrice

ON GetPrice.idTicket = TicketFlights.idTicket

INNER JOIN dbo.Staffs

ON Staffs.idStaffs = GetPrice.idStaffs

GROUP BY TicketFlights.idStaffs, nameStaffs, phoneStaffs

GO

Trên C#:

A computer screen shot of a computer code

Description automatically generated

1. **Stored Procedure**

**-- Stored Procedure tìm kiếm thông tin nhân viên bằng tên**

CREATE PROC USP\_SearchStaffByName

@name NVARCHAR(50)

AS

SELECT Accounts.idStaffs, nameStaffs, addressStaffs, phoneStaffs, userAcc, typeAcc

FROM dbo.Accounts INNER JOIN dbo.Staffs ON Staffs.idStaffs = Accounts.idStaffs

WHERE nameStaffs LIKE '%'+@name+'%'

ORDER BY Accounts.idStaffs

GO

Trên C#:

A close-up of a computer screen

Description automatically generated

**-- Stored Procedure lấy danh sách chuyến bay theo mã chuyến bay**

CREATE PROC USP\_GetFlightByIdFlight

@idFlight CHAR(10)

AS

SELECT idFlights, Flights.idFlightRoutes, idPlane, timeToGo, timeToCome,

AirportToGo.nameAirport AS AirportToGo, AirportToCome.nameAirport AS AirportToCome

FROM dbo.Flights

INNER JOIN dbo.FlightRoutes

ON FlightRoutes.idFlightRoutes = Flights.idFlightRoutes

INNER JOIN dbo.Airports AS AirportToGo

ON AirportToGo.idAirport = FlightRoutes.idAirportToGo

INNER JOIN dbo.Airports AS AirportToCome

ON AirportToCome.idAirport = FlightRoutes.idAirportToCome

WHERE idFlights = @idFlight

GO

Trên C#:

A computer code with text

Description automatically generated

**-- Stored Procedure lấy danh sách chuyến bay theo sân bay và thời gian**

CREATE PROC USP\_GetFlightByAirportAndTime

@idAirportToGo CHAR(10), @idAirportToCome CHAR(10),

@timeToGo DATETIME, @timeToCome DATETIME

AS

SELECT idFlights, timeToGo, timeToCome

FROM dbo.Flights

INNER JOIN dbo.FlightRoutes

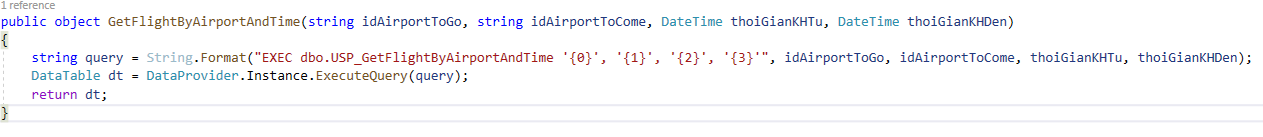
ON FlightRoutes.idFlightRoutes = Flights.idFlightRoutes

WHERE idAirportToGo = @idAirportToGo AND idAirportToCome = @idAirportToCome AND

(timeToGo >= @timeToGo AND timeToGo <= @timeToCome)

GO

Trên C#:



**-- Stored Procedure tìm kiếm thông tin vé của khách hàng từ số điện thoại**

CREATE PROC USP\_SearchTicketFlightByPhone

@phone VARCHAR(10)

AS

SELECT TicketFlights.idTicket, nameCus, identitycardCus, phoneCus, idFlights, nameTicketClass, Prices.Price

FROM dbo.TicketFlights

INNER JOIN dbo.Customers

ON Customers.idCus = TicketFlights.idCus

INNER JOIN dbo.TicketClasses

ON TicketClasses.idTicketClass = TicketFlights.idTicketClass

INNER JOIN (SELECT dbo.UF\_GetPriceByIdFlightAndIdTicketClass(idFlights, idTicketClass) AS Price, idTicket FROM dbo.TicketFlights) AS Prices

ON Prices.idTicket = TicketFlights.idTicket

WHERE phoneCus LIKE '%'+@phone+'%'

GO

Trên C#:

A blue and black text

Description automatically generated

**-- Stored Procedure lấy danh sách hạng vé của chuyến bay từ mã chuyến bay**

CREATE PROC USP\_GetTicketClassForFlight

@idFlight CHAR(10)

AS

SELECT idFlights, nameTicketClass, totalSeats, emptySeats

FROM dbo.TicketStatus INNER JOIN dbo.TicketClasses

ON TicketClasses.idTicketClass = TicketStatus.idTicketClass

WHERE idFlights = @idFlight

GO

Trên C#:

A screen shot of a computer code

Description automatically generated

**-- Stored Procedure lấy danh sách tình trạng vé từ mã máy bay**

CREATE PROC USP\_GetTicketStatusByIdFlight

@idFlight CHAR(10)

AS

SELECT nameTicketClass, totalSeats, emptySeats

FROM dbo.TicketStatus

INNER JOIN dbo.TicketClasses

ON TicketClasses.idTicketClass = TicketStatus.idTicketClass

WHERE idFlights = @idFlight

GO

Trên C#:

A screen shot of a computer code

Description automatically generated

1. **Function**

**-- Function kiểm tra đăng nhập**

create function UF\_Login(@username char(100), @password char(100))

returns int

as

begin

Declare @checkUser INT = (SELECT count(\*) FROM dbo.Accounts WHERE userAcc=@username)

Declare @checkPass INT = (SELECT count(\*) FROM dbo.Accounts WHERE userAcc=@username and passAcc=@password)

DECLARE @check INT = 1

IF (@checkUser != 1)

SET @check = 0

IF (@checkPass != 1)

SET @check = -1

return @check

end

GO

Trên C#:  
**A screenshot of a computer code

Description automatically generated**

**-- Function tạo mã khách hàng**

CREATE FUNCTION UF\_CreateIdCustomer()

RETURNS CHAR(10)

AS

BEGIN

DECLARE @IdCus CHAR(10)

DECLARE @count INT = (SELECT COUNT(\*) FROM dbo.Customers)

IF @count = 0

RETURN 'KH0000'

SET @count = (SELECT CAST((SELECT SUBSTRING((SELECT TOP(1) idCus FROM dbo.Customers ORDER BY idCus DESC), 3, 5)) AS INT) + 1)

SET @IdCus = 'KH' + CAST(@count AS CHAR(10))

DECLARE @temp INT = @count

DECLARE @strSoKhong CHAR(4) = ''

DECLARE @dem INT = 0

WHILE @temp > 0

BEGIN

SET @temp = @temp / 10

SET @dem = @dem + 1

END

DECLARE @i INT = 0

WHILE @i < (4 - @dem)

BEGIN

SET @IdCus = (SELECT STUFF(@IdCus, 3, 0, '0'))

SET @i = @i + 1

END

RETURN @IdCus

END

GO

**-- Function tạo mã nhân viên**

CREATE FUNCTION UF\_CreateIdStaff()

RETURNS CHAR(10)

AS

BEGIN

DECLARE @IdStaff CHAR(10)

DECLARE @count INT = (SELECT COUNT(\*) FROM dbo.Staffs)

IF @count = 0

RETURN 'NV0000'

SET @count = (SELECT CAST((SELECT SUBSTRING((SELECT TOP(1) idStaffs FROM dbo.Staffs ORDER BY idStaffs DESC), 3, 5)) AS INT) + 1)

SET @IdStaff = 'NV' + CAST(@count AS CHAR(10))

DECLARE @temp INT = @count

DECLARE @strSoKhong CHAR(4) = ''

DECLARE @dem INT = 0

WHILE @temp > 0

BEGIN

SET @temp = @temp / 10

SET @dem = @dem + 1

END

DECLARE @i INT = 0

WHILE @i < (4 - @dem)

BEGIN

SET @IdStaff = (SELECT STUFF(@IdStaff, 3, 0, '0'))

SET @i = @i + 1

END

RETURN @IdStaff

END

GO

**-- Function tạo mã máy bay**

CREATE FUNCTION UF\_CreateIdPlane()

RETURNS CHAR(10)

AS

BEGIN

DECLARE @IdPlane CHAR(10)

DECLARE @count INT = (SELECT COUNT(\*) FROM dbo.Planes)

IF @count = 0

RETURN 'MB0000'

SET @count = (SELECT CAST((SELECT SUBSTRING((SELECT TOP(1) idPlane FROM dbo.Planes ORDER BY idPlane DESC), 3, 5)) AS INT) + 1)

SET @IdPlane = 'MB' + CAST(@count AS CHAR(10))

DECLARE @temp INT = @count

DECLARE @strSoKhong CHAR(4) = ''

DECLARE @dem INT = 0

WHILE @temp > 0

BEGIN

SET @temp = @temp / 10

SET @dem = @dem + 1

END

DECLARE @i INT = 0

WHILE @i < (4 - @dem)

BEGIN

SET @IdPlane = (SELECT STUFF(@IdPlane, 3, 0, '0'))

SET @i = @i + 1

END

RETURN @IdPlane

END

GO

Trên C#:

A close-up of a sign

Description automatically generated

**-- Function tạo mã sân bay**

CREATE FUNCTION UF\_CreateIdAirport()

RETURNS CHAR(10)

AS

BEGIN

DECLARE @IdAirport CHAR(10)

DECLARE @count INT = (SELECT COUNT(\*) FROM dbo.Airports)

IF @count = 0

RETURN 'SB0000'

SET @count = (SELECT CAST((SELECT SUBSTRING((SELECT TOP(1) idAirport FROM dbo.Airports ORDER BY idAirport DESC), 3, 5)) AS INT) + 1)

SET @IdAirport = 'SB' + CAST(@count AS CHAR(10))

DECLARE @temp INT = @count

DECLARE @strSoKhong CHAR(4) = ''

DECLARE @dem INT = 0

WHILE @temp > 0

BEGIN

SET @temp = @temp / 10

SET @dem = @dem + 1

END

DECLARE @i INT = 0

WHILE @i < (4 - @dem)

BEGIN

SET @IdAirport = (SELECT STUFF(@IdAirport, 3, 0, '0'))

SET @i = @i + 1

END

RETURN @IdAirport

END

GO

Trên C#:

A close-up of a white background

Description automatically generated

**-- Function tạo mã tuyến bay**

CREATE FUNCTION UF\_CreateIdFlightRoute()

RETURNS CHAR(10)

AS

BEGIN

DECLARE @IdFR CHAR(10)

DECLARE @count INT = (SELECT COUNT(\*) FROM dbo.FlightRoutes)

IF @count = 0

RETURN 'TB0000'

SET @count = (SELECT CAST((SELECT SUBSTRING((SELECT TOP(1) idFlightRoutes FROM dbo.FlightRoutes ORDER BY idFlightRoutes DESC), 3, 5)) AS INT) + 1)

SET @IdFR = 'TB' + CAST(@count AS CHAR(10))

DECLARE @temp INT = @count

DECLARE @strSoKhong CHAR(4) = ''

DECLARE @dem INT = 0

WHILE @temp > 0

BEGIN

SET @temp = @temp / 10

SET @dem = @dem + 1

END

DECLARE @i INT = 0

WHILE @i < (4 - @dem)

BEGIN

SET @IdFR = (SELECT STUFF(@IdFR, 3, 0, '0'))

SET @i = @i + 1

END

RETURN @IdFR

END

GO

Trên C#:

A close-up of a computer screen

Description automatically generated

**-- Function tạo mã hạng vé**

CREATE FUNCTION UF\_CreateIdTicketClass()

RETURNS CHAR(10)

AS

BEGIN

DECLARE @IdTC CHAR(10)

DECLARE @count INT = (SELECT COUNT(\*) FROM dbo.TicketClasses)

IF @count = 0

RETURN 'HV0000'

SET @count = (SELECT CAST((SELECT SUBSTRING((SELECT TOP(1) idTicketClass FROM dbo.TicketClasses ORDER BY idTicketClass DESC), 3, 5)) AS INT) + 1)

SET @IdTC = 'HV' + CAST(@count AS CHAR(10))

DECLARE @temp INT = @count

DECLARE @strSoKhong CHAR(4) = ''

DECLARE @dem INT = 0

WHILE @temp > 0

BEGIN

SET @temp = @temp / 10

SET @dem = @dem + 1

END

DECLARE @i INT = 0

WHILE @i < (4 - @dem)

BEGIN

SET @IdTC = (SELECT STUFF(@IdTC, 3, 0, '0'))

SET @i = @i + 1

END

RETURN @IdTC

END

GO

Trên C#:

A close-up of a white background

Description automatically generated

**-- Function tạo mã chuyến bay**

CREATE FUNCTION UF\_CreateIdFlight()

RETURNS CHAR(10)

AS

BEGIN

DECLARE @IdFlight CHAR(10)

DECLARE @count INT = (SELECT COUNT(\*) FROM dbo.Flights)

IF @count = 0

RETURN 'CB0000'

SET @count = (SELECT CAST((SELECT SUBSTRING((SELECT TOP(1) idFlights FROM dbo.Flights ORDER BY idFlights DESC), 3, 5)) AS INT) + 1)

SET @IdFlight = 'CB' + CAST(@count AS CHAR(10))

DECLARE @temp INT = @count

DECLARE @strSoKhong CHAR(4) = ''

DECLARE @dem INT = 0

WHILE @temp > 0

BEGIN

SET @temp = @temp / 10

SET @dem = @dem + 1

END

DECLARE @i INT = 0

WHILE @i < (4 - @dem)

BEGIN

SET @IdFlight = (SELECT STUFF(@IdFlight, 3, 0, '0'))

SET @i = @i + 1

END

RETURN @IdFlight

END

GO

Trên C#:

A close-up of a computer screen

Description automatically generated

**-- Function tạo mã vé máy bay**

CREATE FUNCTION UF\_CreateIdTicketFlight()

RETURNS CHAR(10)

AS

BEGIN

DECLARE @IdTF CHAR(10)

DECLARE @count INT = (SELECT COUNT(\*) FROM dbo.TicketFlights)

IF @count = 0

RETURN 'VE0000'

SET @count = (SELECT CAST((SELECT SUBSTRING((SELECT TOP(1) idTicket FROM dbo.TicketFlights ORDER BY idTicket DESC), 3, 5)) AS INT) + 1)

SET @IdTF = 'VE' + CAST(@count AS CHAR(10))

DECLARE @temp INT = @count

DECLARE @strSoKhong CHAR(4) = ''

DECLARE @dem INT = 0

WHILE @temp > 0

BEGIN

SET @temp = @temp / 10

SET @dem = @dem + 1

END

DECLARE @i INT = 0

WHILE @i < (4 - @dem)

BEGIN

SET @IdTF = (SELECT STUFF(@IdTF, 3, 0, '0'))

SET @i = @i + 1

END

RETURN @IdTF

END

GO

Dùng trong **Transaction bán vé máy bay**

A screenshot of a computer program

Description automatically generated

**-- Function lấy giá vé của chuyến bay**

CREATE FUNCTION UF\_GetPriceByIdFlightAndIdTicketClass(@idFlight CHAR(10), @idTicketClass CHAR(10))

RETURNS DECIMAL(18, 0)

AS

BEGIN

DECLARE @price DECIMAL(18, 0)

SELECT @price = unitPrice

FROM dbo.Flights

INNER JOIN dbo.Prices

ON Prices.idFlightRoutes = Flights.idFlightRoutes

WHERE idFlights = @idFlight AND idTicketClass = @idTicketClass

RETURN @price

END

GO

Trên C#:

A computer screen shot of a computer code

Description automatically generated

**-- Function lấy số ghế trống từ mã chuyến bay và hạng vé**

CREATE FUNCTION UF\_GetEmptySeatsByIdFlightAndIdTicketClass(@idFlight CHAR(10), @idTicketClass CHAR(10))

RETURNS INT

AS

BEGIN

DECLARE @empty INT

SELECT @empty = emptySeats

FROM dbo.TicketStatus

WHERE idFlights = @idFlight AND idTicketClass = @idTicketClass

IF @empty > 0

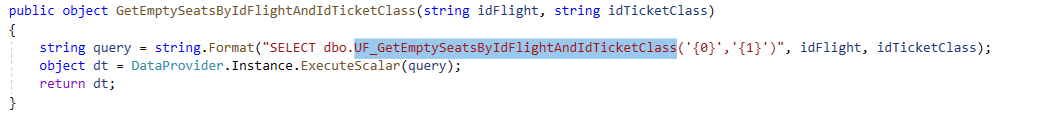
RETURN @empty

RETURN 0

END

GO

Trên C#:



**-- Function lấy số ghế còn lại của máy bay từ mã chuyến bay**

CREATE FUNCTION UF\_GetSeatsOfPlaneByIdFlight(@idFlight CHAR(10))

RETURNS INT

AS

BEGIN

DECLARE @totalseats INT = (SELECT seatsPlane FROM dbo.Planes

WHERE idPlane = (SELECT idPlane FROM dbo.Flights WHERE idFlights = @idFlight))

DECLARE @insertseats INT = (SELECT SUM(totalSeats) FROM dbo.TicketStatus WHERE idFlights = @idFlight)

RETURN @totalseats - @insertseats

END

GO

Trên C#:

A close-up of a computer screen

Description automatically generated

1. **Transaction**

**-- Transaction thêm nhân viên**

CREATE PROC USP\_InsertStaff

@name NVARCHAR(50), @address NVARCHAR(50), @phone VARCHAR(20),

@user CHAR(100), @type INT

AS

DECLARE @id CHAR(10) = (SELECT dbo.UF\_CreateIdStaff())

DECLARE @len INT = (SELECT LEN(@phone))

BEGIN TRAN

IF (@len <> 10)

BEGIN

ROLLBACK

RETURN

END

INSERT dbo.Staffs(idStaffs, nameStaffs, addressStaffs, phoneStaffs) VALUES (@id, @name, @address, @phone)

IF (@@ERROR <> 0)

BEGIN

ROLLBACK

RETURN

END

INSERT dbo.Accounts(userAcc, passAcc, idStaffs, typeAcc) VALUES (@user, '1', @id, @type)

IF (@@ERROR <> 0)

BEGIN

ROLLBACK

RETURN

END

COMMIT TRAN

GO

Trên C#:

A close up of a computer screen

Description automatically generated

**-- Transaction sửa thông tin nhân viên**

CREATE PROC USP\_UpdateStaff

@id CHAR(10), @name NVARCHAR(50), @address NVARCHAR(50), @phone VARCHAR(20),

@user CHAR(100), @type INT

AS

DECLARE @len INT = (SELECT LEN(@phone))

BEGIN TRAN

IF (@len <> 10)

BEGIN

ROLLBACK

RETURN

END

UPDATE dbo.Staffs SET nameStaffs = @name, addressStaffs = @address, phoneStaffs = @phone

WHERE idStaffs = @id

IF (@@ERROR <> 0)

BEGIN

ROLLBACK

RETURN

END

UPDATE dbo.Accounts SET userAcc = @user, typeAcc = @type

WHERE idStaffs = @id

IF (@@ERROR <> 0)

BEGIN

ROLLBACK

RETURN

END

COMMIT TRAN

GO

Trên C#:

A white background with red and yellow dots

Description automatically generated

**-- Transaction bán vé máy bay**

CREATE PROC USP\_BookingTicket

@nameCus NVARCHAR(50), @identityCus VARCHAR(12), @phoneCus VARCHAR(10),

@idFlight CHAR(10), @idTicketClass CHAR(10), @idStaff CHAR(10)

AS

DECLARE @count INT, @idCus CHAR(10)

SELECT @count = COUNT(\*) FROM dbo.Customers WHERE identitycardCus = @identityCus

BEGIN TRAN

IF (@count > 0)

SET @idCus = (SELECT idCus FROM dbo.Customers WHERE identitycardCus = @identityCus)

ELSE

BEGIN

SET @idCus = (SELECT dbo.UF\_CreateIdCustomer())

INSERT dbo.Customers( idCus, nameCus, identitycardCus, phoneCus)

VALUES(@idCus, @nameCus, @identityCus, @phoneCus)

END

INSERT dbo.TicketFlights (idTicket, idCus, idFlights, idTicketClass, idStaffs)

VALUES ((SELECT dbo.UF\_CreateIdTicketFlight()), @idCus, @idFlight, @idTicketClass, @idStaff)

IF (@@ERROR <> 0)

BEGIN

ROLLBACK

RETURN

END

COMMIT TRAN

GO

Trên C#:

**A white background with red and blue text

Description automatically generated**

1. **Kết nối CSDL**

Kết nối cơ sở dữ liệu khi thao tác giữa Manager và Staff trên C#

private string strmanager = "Data Source=.\\RUBIE;Initial Catalog=VNJET;User ID=Manager;Password=Manager;Connect Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;MultiSubnetFailover=False";

private string strstaff = "Data Source=.\\RUBIE;Initial Catalog=VNJET;User ID=Staff;Password=Staff;Connect Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;MultiSubnetFailover=False";

private string connectionStr = "Data Source=.\\RUBIE;Initial Catalog=VNJET;Integrated Security=True";

public void Decentralize(int type)

{

if (type == 1)

connectionStr = strmanager;

else

connectionStr = strstaff;

}

**A screenshot of a computer code

Description automatically generated**

1. **Tạo user và phân quyền**

**-- Tạo user và phân quyền cho quản lý**

CREATE LOGIN Manager WITH PASSWORD = 'Manager'

CREATE ROLE manager

GO

GRANT SELECT, INSERT, DELETE, UPDATE ON dbo.Customers TO manager WITH GRANT OPTION

GRANT SELECT, INSERT, DELETE, UPDATE ON dbo.Staffs TO manager WITH GRANT OPTION

GRANT SELECT, INSERT, DELETE, UPDATE ON dbo.Accounts TO manager WITH GRANT OPTION

GRANT SELECT, INSERT, DELETE, UPDATE ON dbo.Planes TO manager WITH GRANT OPTION

GRANT SELECT, INSERT, DELETE, UPDATE ON dbo.Airports TO manager WITH GRANT OPTION

GRANT SELECT, INSERT, DELETE, UPDATE ON dbo.FlightRoutes TO manager WITH GRANT OPTION

GRANT SELECT, INSERT, DELETE, UPDATE ON dbo.TicketClasses TO manager WITH GRANT OPTION

GRANT SELECT, INSERT, DELETE, UPDATE ON dbo.Prices TO manager WITH GRANT OPTION

GRANT SELECT, INSERT, DELETE, UPDATE ON dbo.Flights TO manager WITH GRANT OPTION

GRANT SELECT, INSERT, DELETE, UPDATE ON dbo.TicketFlights TO manager WITH GRANT OPTION

GRANT SELECT, INSERT, DELETE, UPDATE ON dbo.TicketStatus TO manager WITH GRANT OPTION

GRANT SELECT, INSERT, DELETE, UPDATE ON dbo.Sales TO manager WITH GRANT OPTION

GO

GRANT EXECUTE ON dbo.UF\_CreateIdCustomer TO manager

GRANT EXECUTE ON dbo.UF\_CreateIdStaff TO manager

GRANT EXECUTE ON dbo.UF\_CreateIdPlane TO manager

GRANT EXECUTE ON dbo.UF\_CreateIdAirport TO manager

GRANT EXECUTE ON dbo.UF\_CreateIdFlightRoute TO manager

GRANT EXECUTE ON dbo.UF\_CreateIdTicketClass TO manager

GRANT EXECUTE ON dbo.UF\_CreateIdFlight TO manager

GRANT EXECUTE ON dbo.UF\_CreateIdTicketFlight TO manager

GRANT EXECUTE ON dbo.UF\_GetPriceByIdFlightAndIdTicketClass TO manager

GRANT EXECUTE ON dbo.UF\_GetEmptySeatsByIdFlightAndIdTicketClass TO manager

GRANT EXECUTE ON dbo.UF\_GetSeatsOfPlaneByIdFlight TO manager

GO

GRANT EXECUTE ON dbo.USP\_InsertStaff TO manager

GRANT EXECUTE ON dbo.USP\_UpdateStaff TO manager

GRANT EXECUTE ON dbo.USP\_SearchStaffByName TO manager

GRANT EXECUTE ON dbo.USP\_GetTicketClassForFlight TO manager

GRANT EXECUTE ON dbo.USP\_GetFlightByIdFlight TO manager

GRANT EXECUTE ON dbo.USP\_GetFlightByAirportAndTime TO manager

GRANT EXECUTE ON dbo.USP\_SearchTicketFlightByPhone TO manager

GRANT EXECUTE ON dbo.USP\_BookingTicket TO manager

GRANT EXECUTE ON dbo.USP\_GetTicketStatusByIdFlight TO manager

GO

GRANT SELECT ON dbo.UV\_StaffForDisplay TO manager

GRANT SELECT ON dbo.UV\_FlightRouteForDisplay TO manager

GRANT SELECT ON dbo.UV\_TicketFlightForDisplay TO manager

GRANT SELECT ON dbo.UV\_SaleByStaff TO manager

GO

**-- Tạo user và phân quyền cho nhân viên**

CREATE LOGIN Staff WITH PASSWORD = 'Staff'

CREATE ROLE staff

GO

GRANT SELECT, INSERT, DELETE, UPDATE ON dbo.Customers TO staff WITH GRANT OPTION

GRANT SELECT ON dbo.Accounts TO staff WITH GRANT OPTION

GRANT SELECT ON dbo.Planes TO staff WITH GRANT OPTION

GRANT SELECT ON dbo.Airports TO staff WITH GRANT OPTION

GRANT SELECT ON dbo.FlightRoutes TO staff WITH GRANT OPTION

GRANT SELECT ON dbo.TicketClasses TO staff WITH GRANT OPTION

GRANT SELECT ON dbo.Prices TO staff WITH GRANT OPTION

GRANT SELECT ON dbo.Flights TO staff WITH GRANT OPTION

GRANT SELECT, INSERT, DELETE, UPDATE ON dbo.TicketFlights TO staff WITH GRANT OPTION

GRANT SELECT, INSERT, DELETE, UPDATE ON dbo.TicketStatus TO staff WITH GRANT OPTION

GRANT SELECT, INSERT, DELETE, UPDATE ON dbo.Sales TO staff WITH GRANT OPTION

GO

GRANT EXECUTE ON dbo.UF\_CreateIdCustomer TO staff

GRANT EXECUTE ON dbo.UF\_CreateIdTicketFlight TO staff

GRANT EXECUTE ON dbo.UF\_GetPriceByIdFlightAndIdTicketClass TO staff

GRANT EXECUTE ON dbo.UF\_GetEmptySeatsByIdFlightAndIdTicketClass TO staff

GRANT EXECUTE ON dbo.UF\_GetSeatsOfPlaneByIdFlight TO staff

GO

GRANT EXECUTE ON dbo.USP\_GetTicketClassForFlight TO staff

GRANT EXECUTE ON dbo.USP\_GetFlightByIdFlight TO staff

GRANT EXECUTE ON dbo.USP\_GetFlightByAirportAndTime TO staff

GRANT EXECUTE ON dbo.USP\_SearchTicketFlightByPhone TO staff

GRANT EXECUTE ON dbo.USP\_BookingTicket TO staff

GRANT EXECUTE ON dbo.USP\_GetTicketStatusByIdFlight TO staff

GO

GRANT SELECT ON dbo.UV\_FlightRouteForDisplay TO staff

GRANT SELECT ON dbo.UV\_TicketFlightForDisplay TO staff

GO

Trigger tạo tài khoản:

CREATE TRIGGER [dbo].[Trigger\_CreateSQLAccount] ON [dbo].[Accounts]

AFTER INSERT

AS

BEGIN

DECLARE @userName nvarchar(100), @passWord nvarchar(100), @idStaffs nvarchar(10), @typeAcc int

SELECT @userName = i.userAcc, @passWord = i.passAcc, @idStaffs = i.idStaffs, @typeAcc = i.typeAcc

FROM inserted i

DECLARE @sqlString nvarchar(2000), @macv nvarchar(10)

SET @sqlString = 'CREATE LOGIN [' + @userName + '] WITH PASSWORD=''' + @passWord + ''', DEFAULT\_DATABASE=[VNJET], CHECK\_EXPIRATION=OFF, CHECK\_POLICY=OFF'

EXEC (@sqlString)

SET @sqlString = 'CREATE USER ' + @userName + ' FOR LOGIN ' + @userName

EXEC (@sqlString)

IF (@typeAcc = 1)

SET @sqlString = 'ALTER SERVER ROLE sysadmin ADD MEMBER ' + @userName;

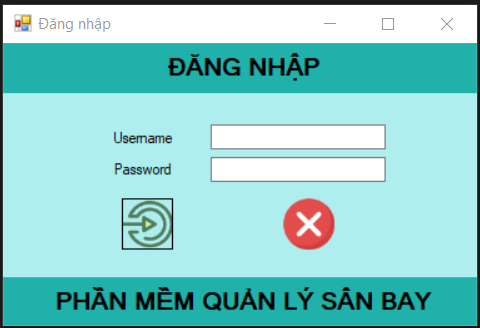
ELSE

SET @sqlString = 'ALTER ROLE staff ADD MEMBER ' + @userName;

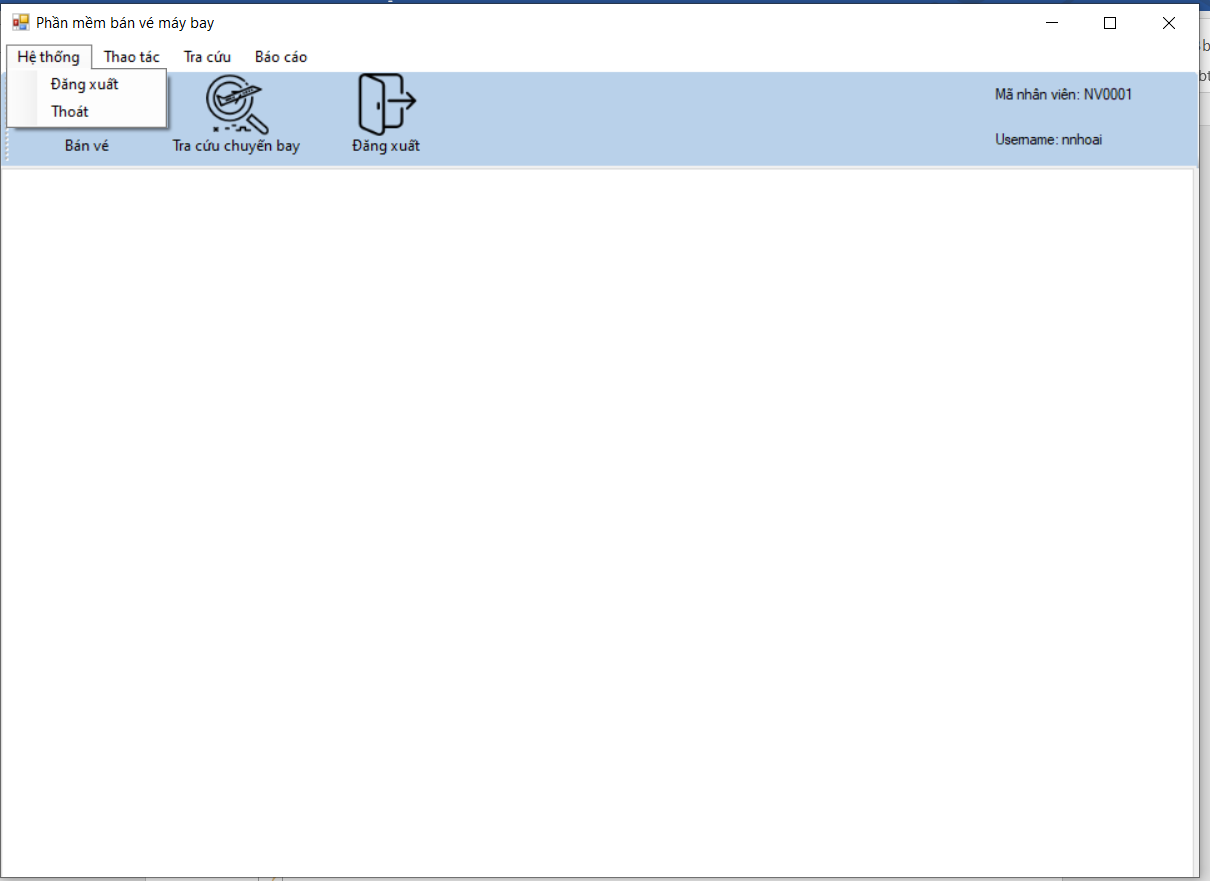
EXEC (@sqlString)

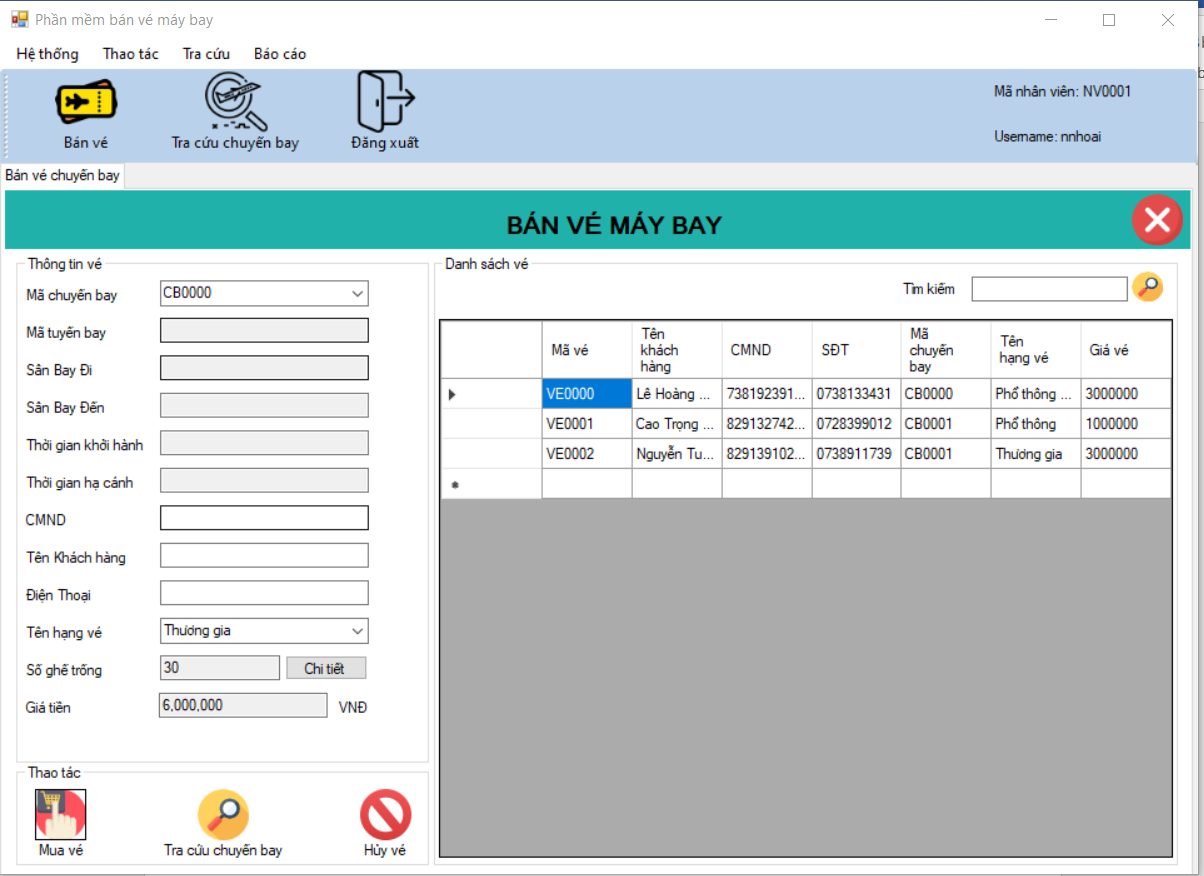
END

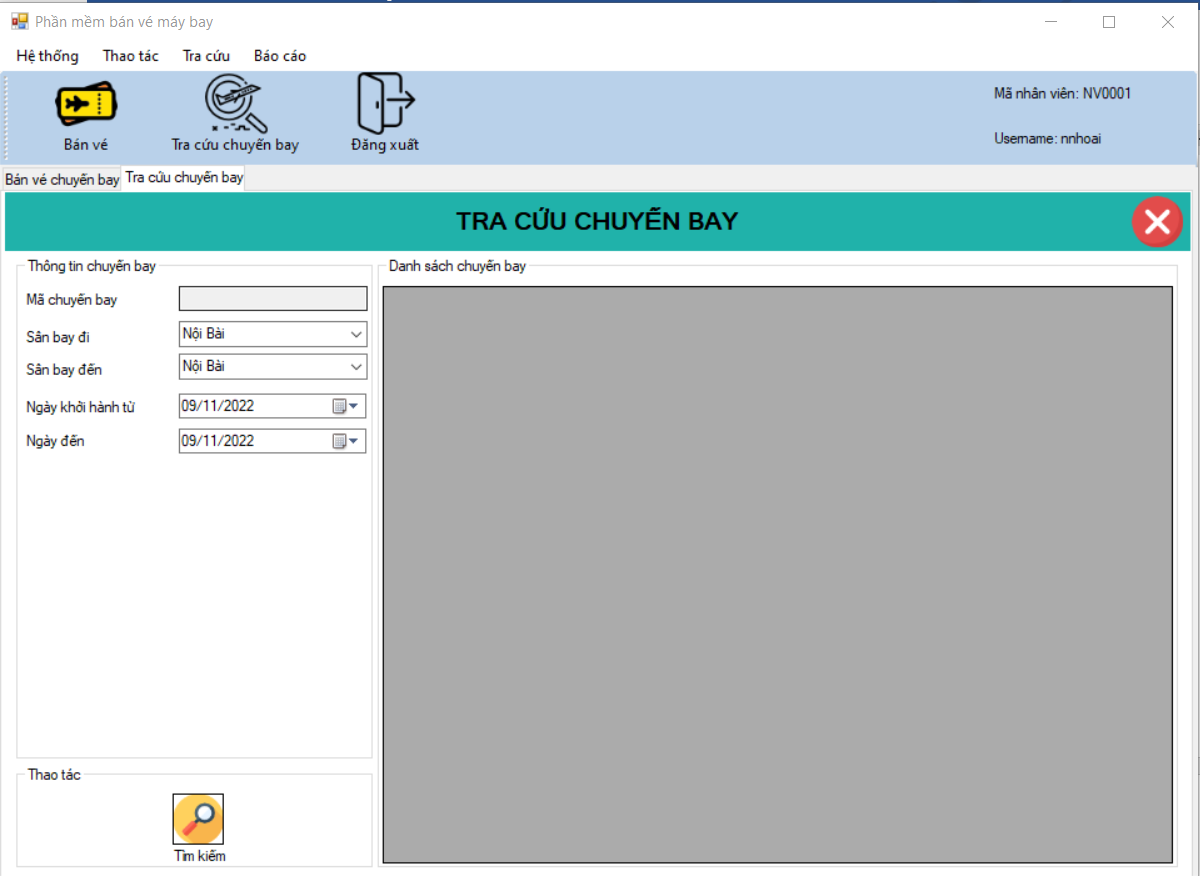
1. **Giao diện (Winform)**
2. **Đăng nhập**

****

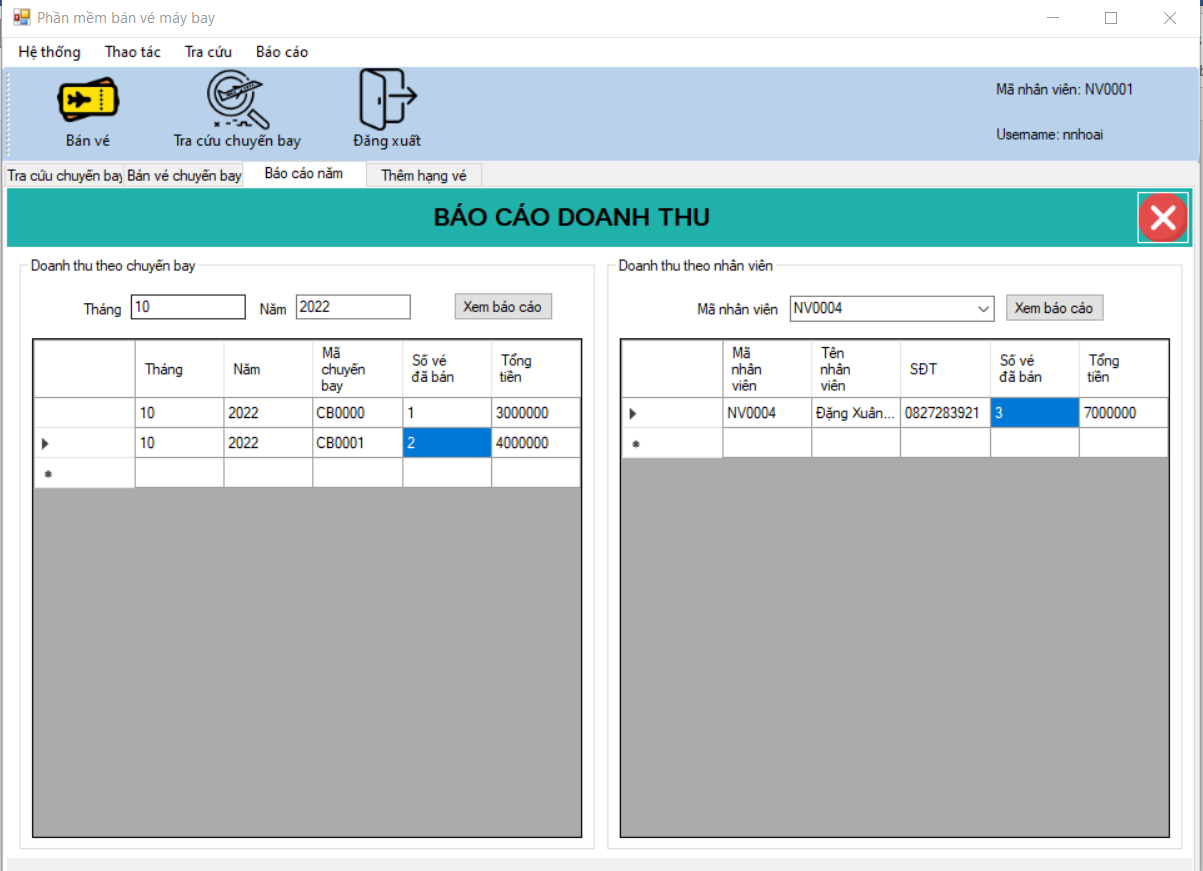
1. **Hệ thống**

****

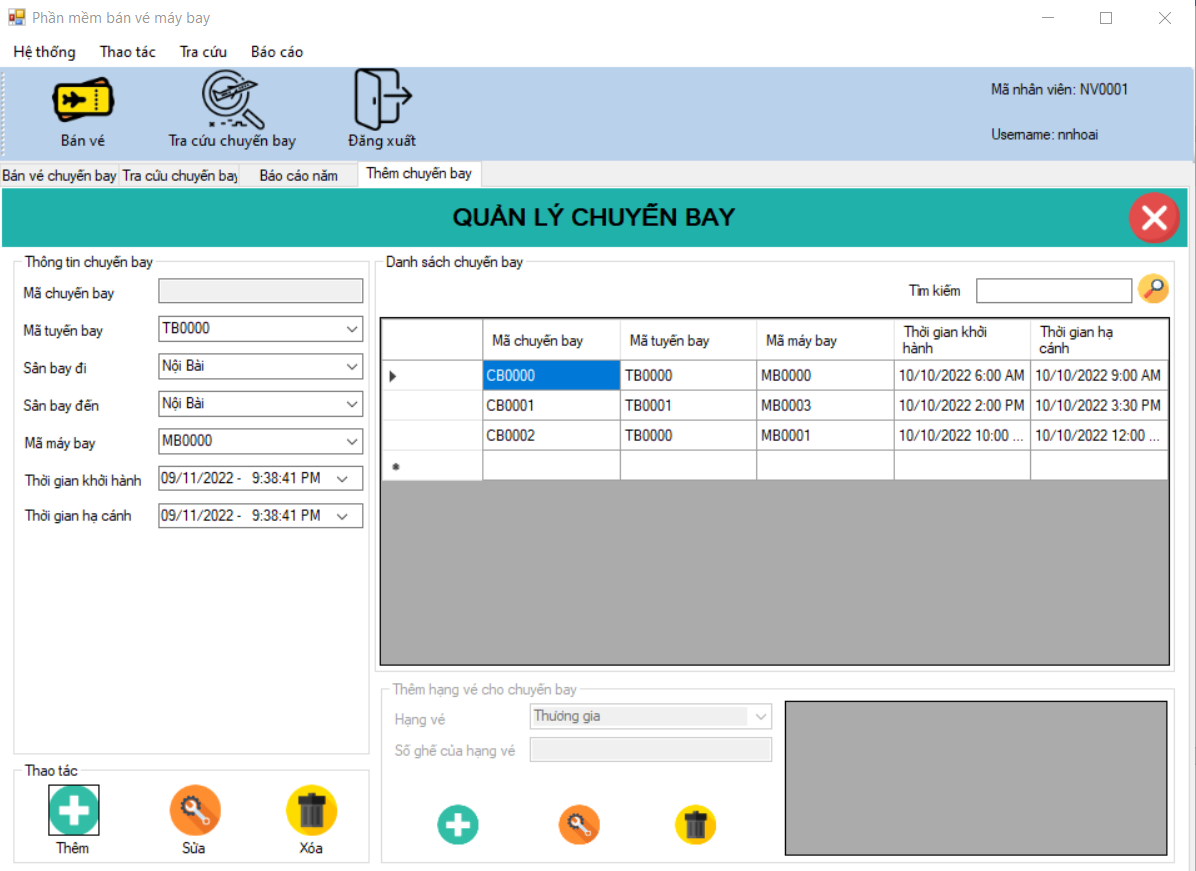
1. **Bán vé**
2. **Tra cứu chuyến bay**

****

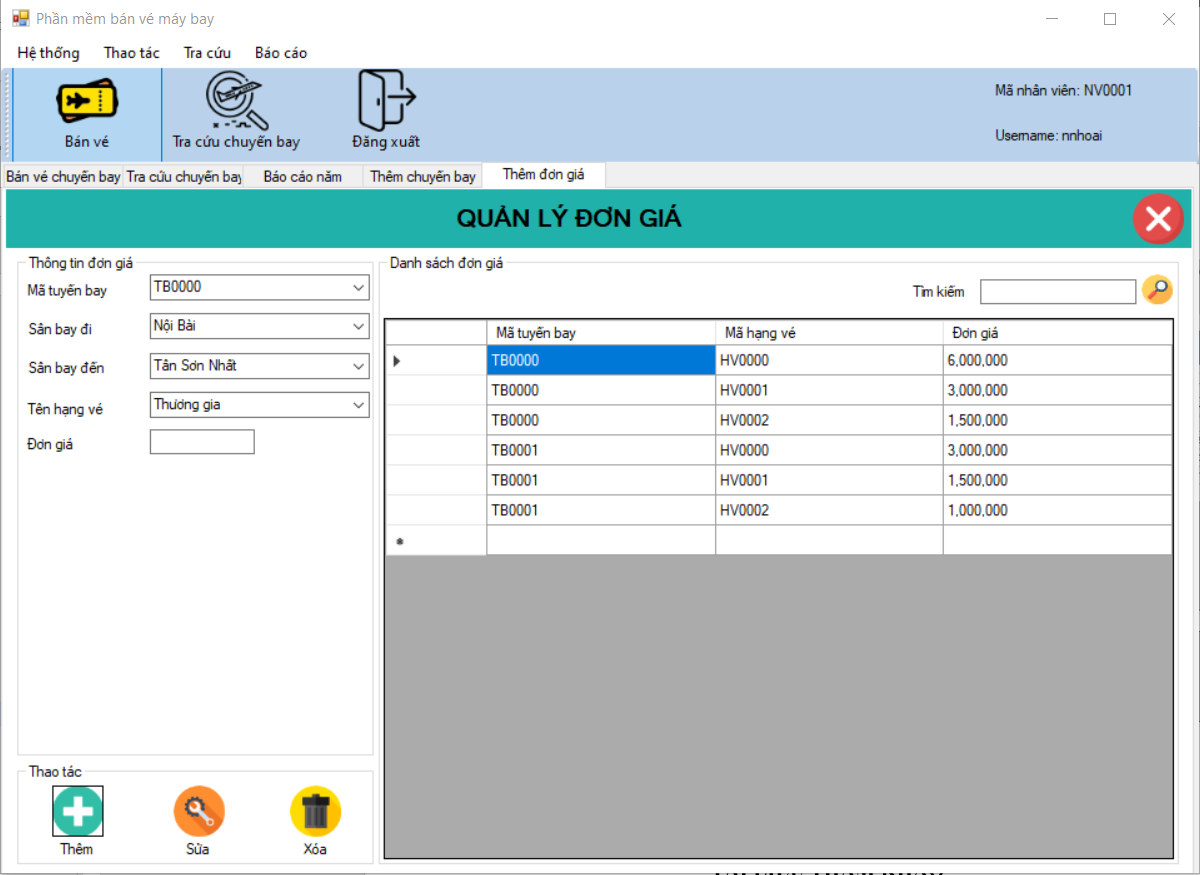
1. **Báo cáo doanh thu**

****

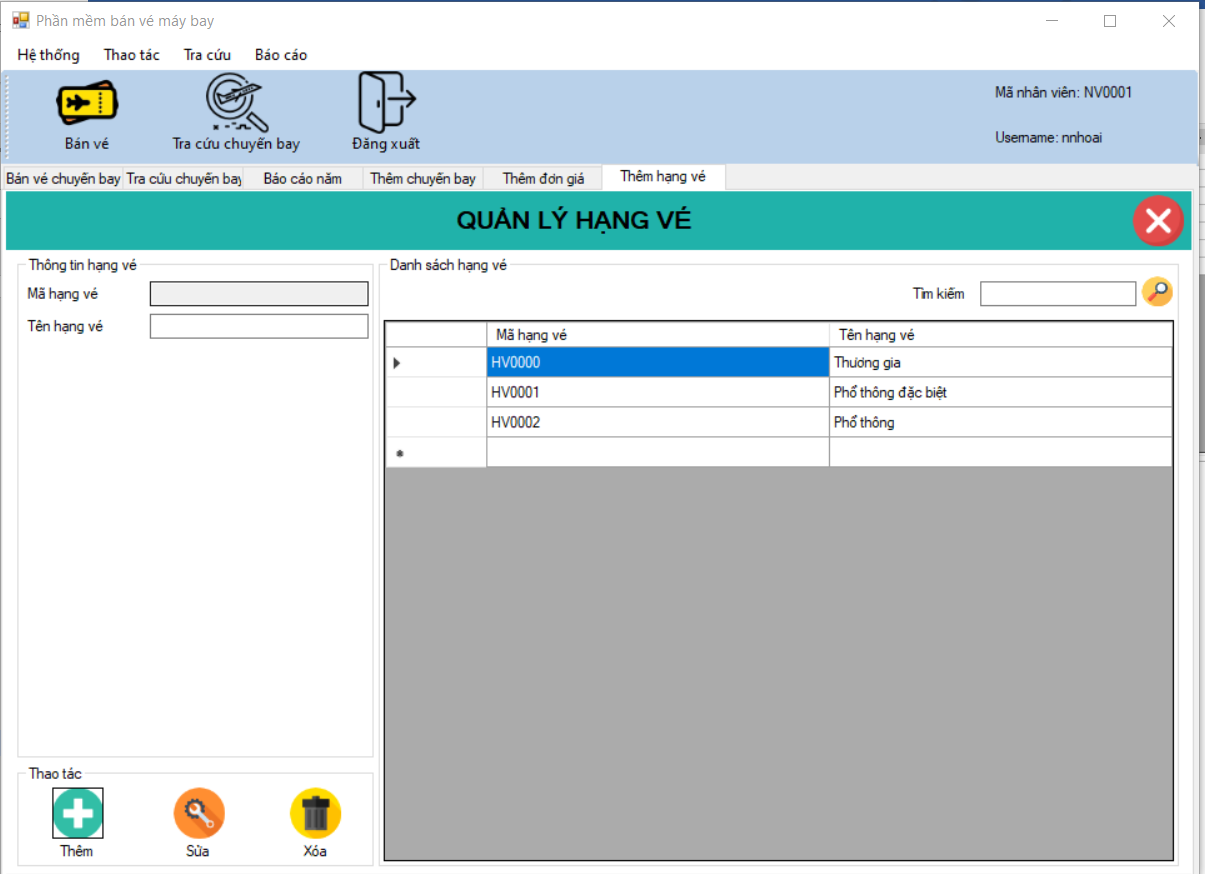
1. **Quản lý chuyến bay**

****

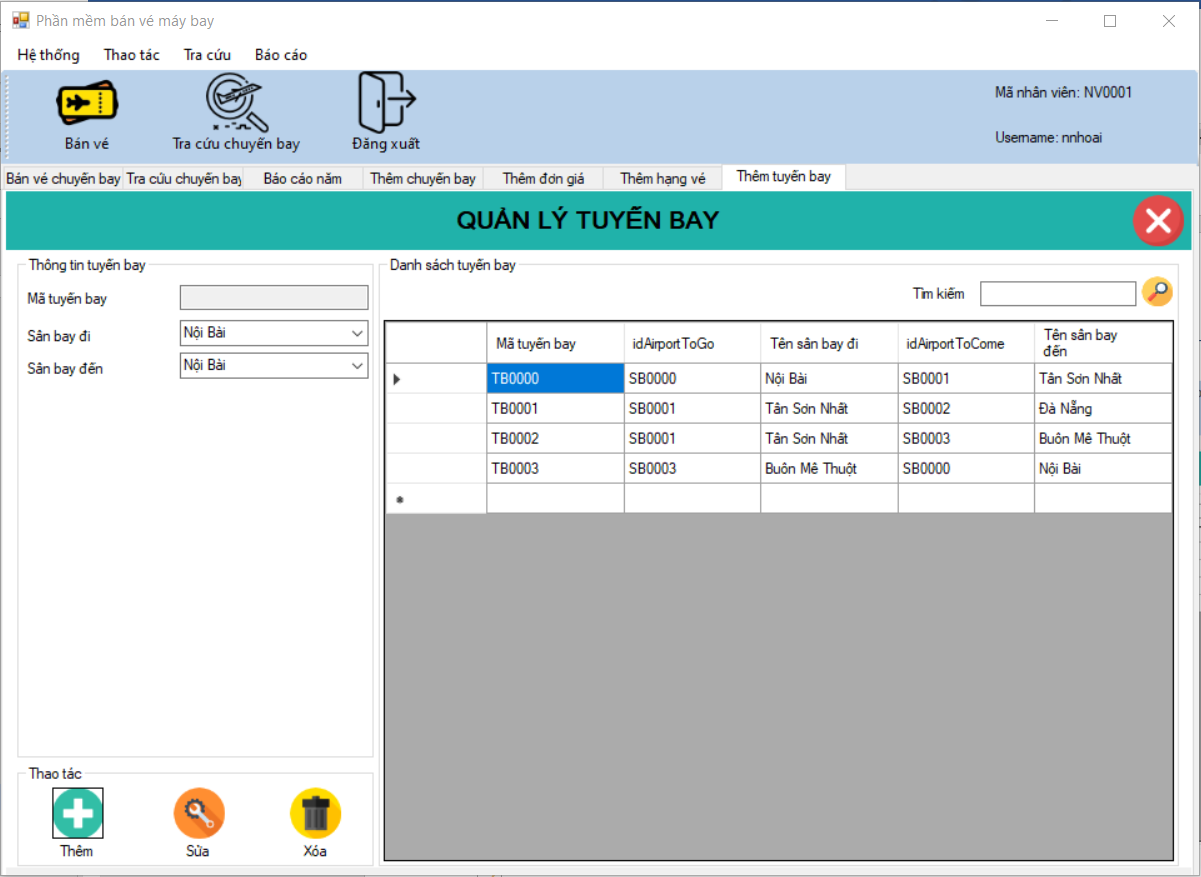
1. **Quản lý đơn giá**

****

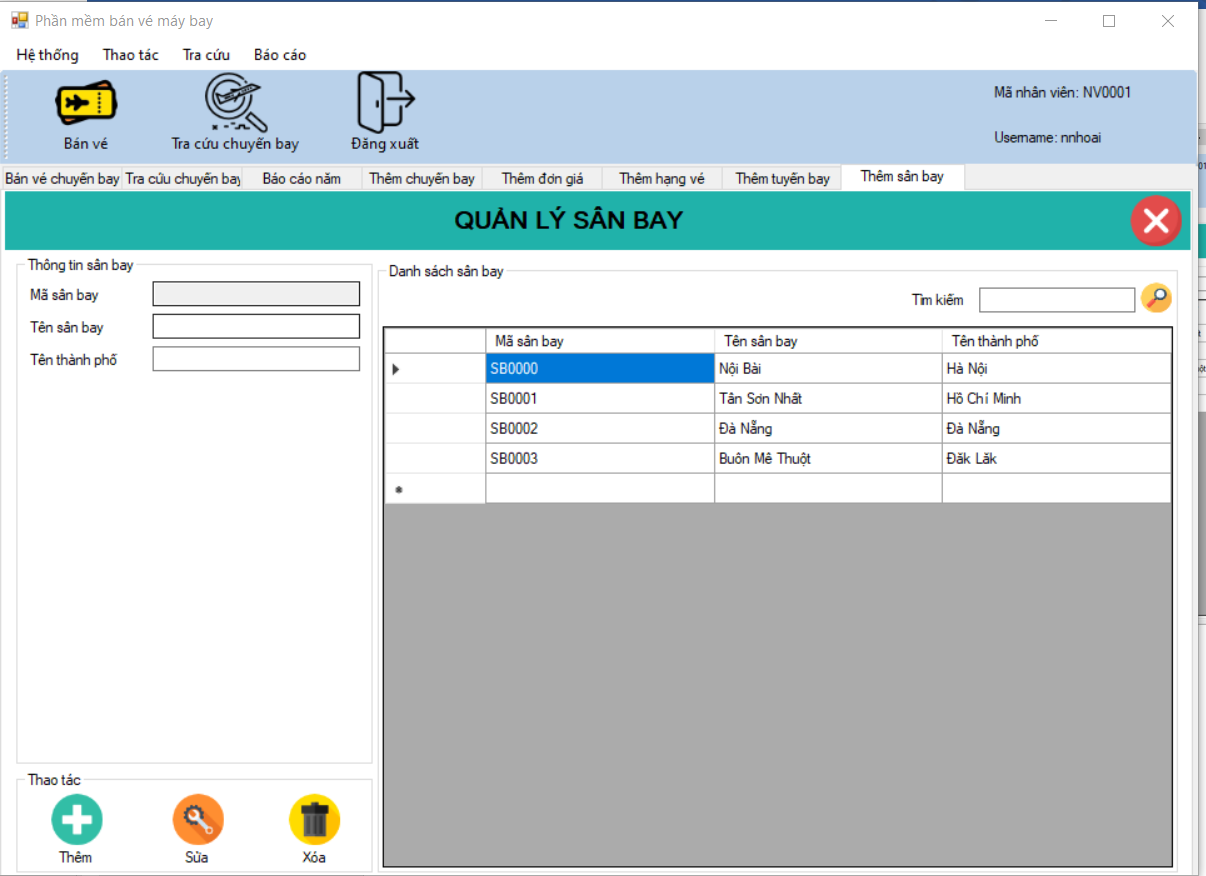
1. **Quản lý hạng vé**

****

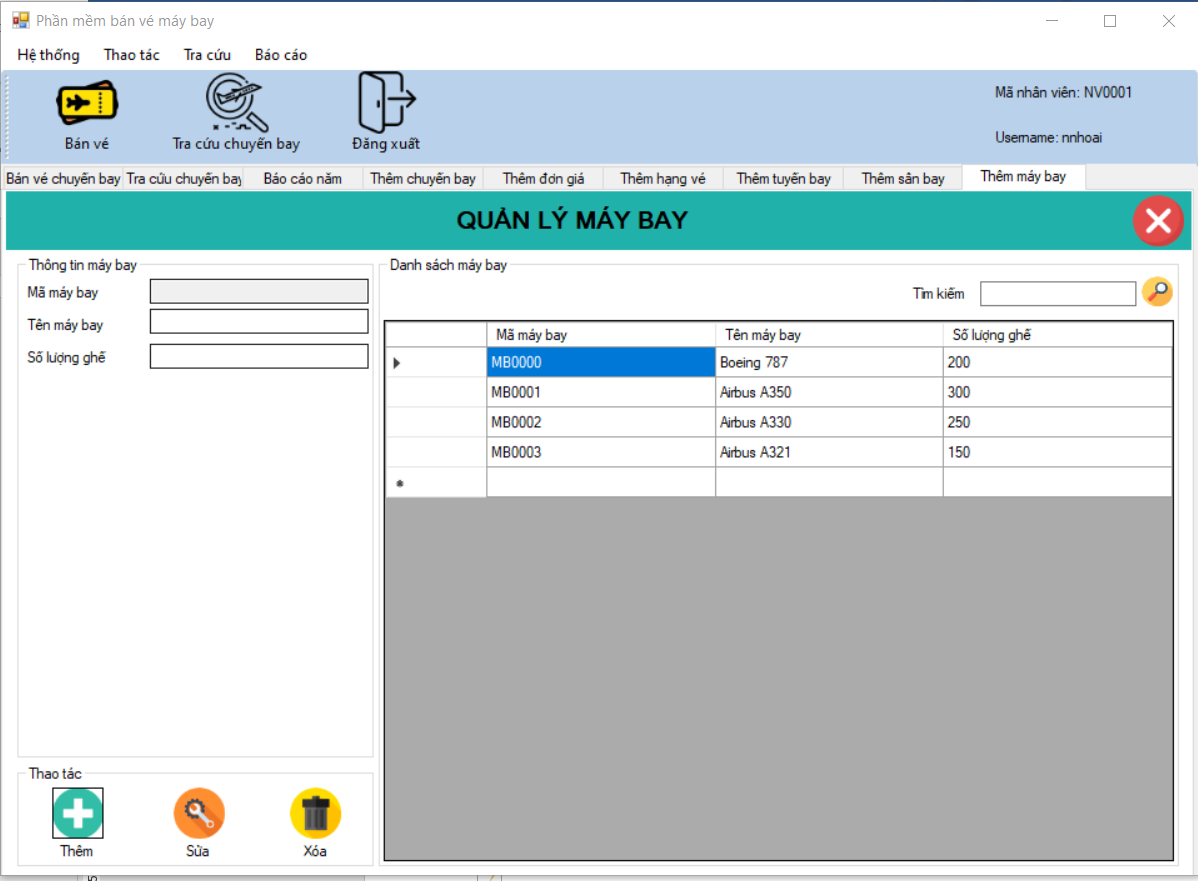
1. **Quản lý tuyến bay**

****

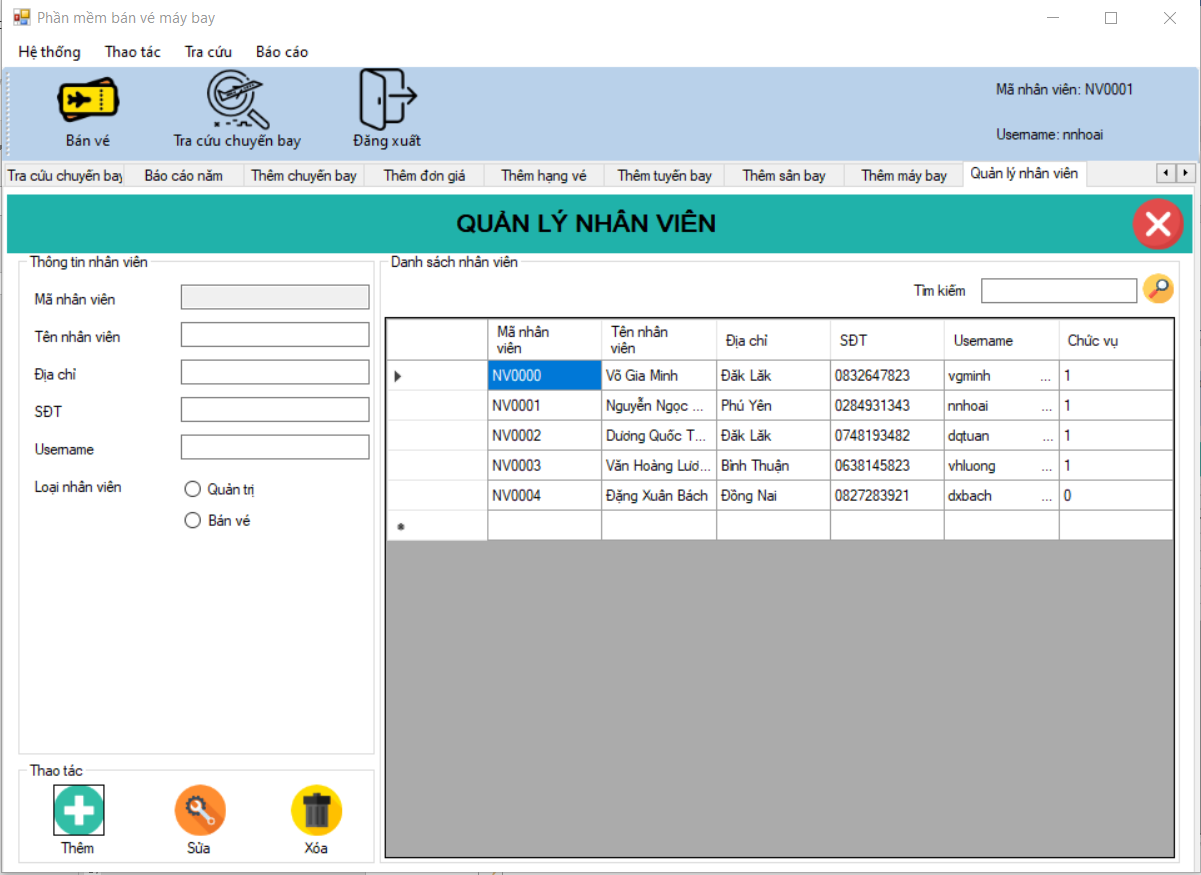
1. **Quản lý sân bay**

****

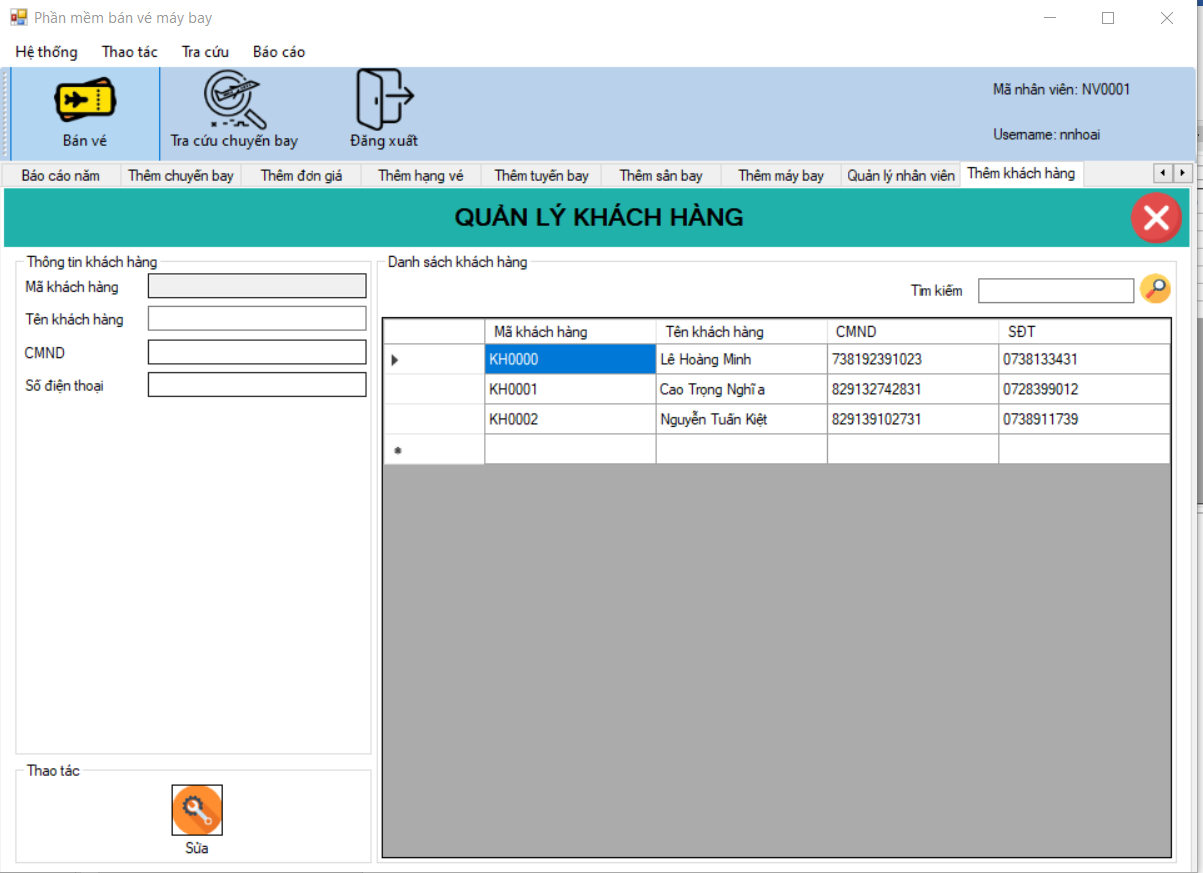
1. **Quản lý máy bay**

****

1. **Quản lý nhân viên**

****

1. **Quản lý khách hàng**

****

**TÀI LIỆU THAM KHẢO**

1. Khóa học ‘Hệ quản trị cơ sở dữ liệu’ – thầy Nguyễn Thành Sơn