

**UBND TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC SÀI GÒN**



ĐỒ ÁN CHUYÊN NGÀNH

ĐỀ TÀI

QUẢN LÝ CỬA HÀNG BÁN LAPTOP

NGÀNH: CÔNG NGHỆ THÔNG TIN

BẠC: ĐẠI HỌC

HÌNH THỨC: CHÍNH QUY

SINH VIÊN THỰC HIỆN: Dương Thanh Trục

MÃ SỐ SINH VIÊN: 3120410569

LỚP: DCT1201

GIẢNG VIÊN PHỤ TRÁCH: PGS. TS. Nguyễn Tuấn Đăng

NĂM HỌC: 2023-2024

TP. HỒ CHÍ MINH, THÁNG 12 NĂM 2023

LỜI CẢM ƠN

Em xin chân thành cảm ơn và lời tri ân sâu sắc đến các thầy cô của trường Đại Học Sài Gòn, đặc biệt là các thầy cô giảng viên khoa Công Nghệ Thông Tin đã cho cơ hội và tạo điều kiện cho em tiếp cận nghiên cứu để hoàn thành đồ án môn học này.

Em cũng xin đặc biệt gửi lời cảm ơn, tri ân sâu sắc đến thầy Nguyễn Tuấn Đăng – giảng viên khoa Công Nghệ Thông Tin đã tận tình giúp đỡ và hướng dẫn em trong quá trình nghiên cứu và hoàn thành đồ án môn học.

Mặc dù đã cố gắng hết sức trong quá trình làm đồ án nhưng vốn kiến thức và kinh nghiệm còn hạn chế nên không thể tránh khỏi những thiếu sót trong bài báo cáo. Do vậy em rất mong nhận được những lời góp ý chân thành của thầy cô để em có thể hoàn thành tốt hơn ở những bài báo cáo sắp tới.

Em xin chân thành cảm ơn thầy!

[illegible]

Mục lục

Giới thiệu	1
Chương 1: Tổng quan về đề tài.....	2
1.1. Mục tiêu	2
1.2. Đối tượng nghiên cứu	2
1.3. Phạm vi nghiên cứu/ứng dụng	2
1.4. Đóng góp mới (dự kiến).....	2
1.5. Lịch sử.....	4
Chương 2: Cơ sở lý thuyết	5
2.1. Lý thuyết cơ sở dữ liệu	5
2.2. Ngôn ngữ lập trình.....	6
2.2.1. Ngôn ngữ C#.....	6
a) Định nghĩa	6
b) Đặc trưng của ngôn ngữ C#.....	6
c) Ưu điểm, nhược điểm của C#.....	7
Chương 3: Phân tích và thiết kế cơ sở dữ liệu.....	9
3.1. Thiết kế cơ sở dữ liệu	9
3.2. Mô hình Class Diagram.....	15
Chương 4: Mô hình hoá yêu cầu.....	16
4.1. Mô hình luồng dữ liệu.....	16
4.2. Thiết kế hệ thống.....	17
a) Use Case tổng quát	17
b) Use Case chức năng.....	17
4.3. Lược đồ tuần tự.....	35
a) Chức năng quản lý bán hàng.....	35
b) Chức năng quản lý sản phẩm	36
c) Chức năng quản lý khuyến mãi.....	37
d) Chức năng quản lý nhân viên.....	38
e) Chức năng quản lý khách hàng	39
f) Chức năng quản lý kho.....	40
Chương 5: Thiết kế phần mềm	41
5.1. Chức năng đăng nhập.....	41
5.2. Chức năng bán hàng.....	43

5.3. Chức năng thanh toán	47
5.4. Chức năng quản lý sản phẩm.....	50
5.5. Chức năng quản lý khuyến mãi	52
5.6. Chức năng quản lý nhân viên	54
5.7. Chức năng quản lý khách hàng	56
5.8. Chức năng quản lý phiếu nhập kho	57
5.9. Chức năng thống kê	59
TỔNG KẾT	61
Phụ lục: Code xử lý.....	62
Tài liệu tham khảo	95

Giới thiệu

Một cửa hàng hoạt động tốt, thì cần phải quản lý rất nhiều thứ. Với cách quản lý thủ công, sẽ mất rất nhiều thời gian và công sức, gây nhiều khó khăn cho các chủ cửa hàng. Tuy nhiên, với sự phát triển của công nghệ hiện đại, việc sử dụng các phần mềm quản lý sẽ giúp quy trình làm việc rõ ràng hơn, mọi thứ được tự động hóa và phân bổ hợp lý. Phần mềm quản lý cửa hàng bán Laptop nhằm mục đích đơn giản hoá quy trình bán hàng và quản lý các đối tượng liên quan.

Chương 1: Tổng quan về đề tài

Đề tài nghiên cứu: Quản lý cửa hàng bán Laptop

1.1. Mục tiêu

- Mục tiêu tổng quát: Phát triển một hệ thống quản lý cửa hàng bán Laptop, giúp cải thiện quy trình kinh doanh và phục vụ khách hàng hiệu quả
- Mục tiêu cụ thể:
 - Tạo ra một ứng dụng quản lý cửa hàng bán Laptop dễ sử dụng và đa chức năng.
 - Quản lý kho với các chức năng thêm sửa xóa sản phẩm và theo dõi số lượng trong kho.
 - Tối ưu hoá quy trình bán hàng và quản lý đơn hàng từ việc đặt hàng đến giao hàng.
 - Cung cấp tính năng quản lý khách hàng.

1.2. Đối tượng nghiên cứu

Sử dụng ngôn ngữ lập trình C# để xây dựng ứng dụng quản lý cửa hàng bán Laptop.

1.3. Phạm vi nghiên cứu/ứng dụng

Nghiên cứu tập trung vào việc phát triển và triển khai hệ thống quản lý cửa hàng bán Laptop, bao gồm các khía cạnh sau:

Thiết kế giao diện người dùng trực quan và dễ sử dụng..

Tối ưu hóa quy trình đặt hàng và theo dõi đơn hàng.

Đảm bảo an toàn và bảo mật dữ liệu của khách hàng và doanh nghiệp.

Triển khai hệ thống và đào tạo nhân viên sử dụng ứng dụng.

1.4. Đóng góp mới (dự kiến)

Kế hoạch triển khai:

Lập kế hoạch và phân tích yêu cầu.

Thiết kế cơ sở dữ liệu và giao diện người dùng.

Phát triển và kiểm thử phần mềm.

Triển khai và cài đặt ứng dụng tại cửa hàng.

Đào tạo và hỗ trợ người sử dụng.

Đánh giá và điều chỉnh sau triển khai.

Nghiên cứu tổng quan:

Sự cần thiết của quản lý cửa hàng bán laptop:

Cửa hàng bán laptop là một trong những loại cửa hàng bán lẻ phổ biến và đa dạng sản phẩm. Quản lý hiệu quả giúp cải thiện khả năng cạnh tranh, tối ưu hóa lợi nhuận và nâng cao trải nghiệm của khách hàng.

Thị trường laptop:

Nghiên cứu cần xem xét thị trường laptop, bao gồm xu hướng, sự cạnh tranh, và sự thay đổi trong nhu cầu của người tiêu dùng. Điều này có thể giúp xác định chiến lược kinh doanh và cung cấp những sản phẩm phù hợp.

Quản lý tồn kho:

Tối ưu hóa việc quản lý tồn kho là yếu tố quan trọng trong quản lý cửa hàng bán laptop. Cần nghiên cứu về cách theo dõi và dự đoán nhu cầu, giảm thiểu lãng phí và đảm bảo sẵn sàng cung cấp sản phẩm.

Quản lý nhân sự:

Hiệu suất của nhân viên cửa hàng đóng vai trò quan trọng trong thành công của cửa hàng. Nghiên cứu về quản lý nhân sự, đào tạo, và thúc đẩy động viên có thể giúp nâng cao năng suất và hài lòng của nhân viên.

Quản lý tài chính:

Quản lý tài chính đúng cách là yếu tố quyết định trong việc duy trì sự tồn tại của cửa hàng. Nghiên cứu về quản lý chi phí, phân tích lợi nhuận, và quản lý nguồn vốn có thể giúp cải thiện tình hình tài chính.

Kỹ thuật và dịch vụ khách hàng:

Quản lý cửa hàng bán laptop cần xem xét các khía cạnh kỹ thuật như bảo trì, sửa chữa, và cập nhật sản phẩm. Cung cấp dịch vụ khách hàng xuất sắc có thể tạo sự khác biệt trong ngành.

Tiếp thị và quảng cáo:

Phương pháp tiếp thị và quảng cáo đóng vai trò quan trọng trong việc thu hút khách hàng. Nghiên cứu về chiến lược tiếp thị trực tuyến và offline có thể giúp tạo ra chiến dịch tiếp thị hiệu quả.

Pháp luật và quy định:

Nghiên cứu về các quy định và luật pháp liên quan đến bán lẻ và sản phẩm điện tử, như quyền bảo vệ người tiêu dùng và bản quyền, là cần thiết để đảm bảo tuân thủ và tránh rủi ro pháp lý.

Phân tích dữ liệu và quyết định dựa trên dữ liệu:

Sử dụng phân tích dữ liệu để đánh giá hiệu suất cửa hàng, dự đoán nhu cầu sản phẩm, và đưa ra quyết định kinh doanh thông minh.

1.5. Lịch sử

Ngành công nghiệp máy tính và bán laptop đã có sự phát triển đáng kể trong thế kỷ 20 và 21. Sự gia tăng đáng kể về sức mua của người tiêu dùng và sự phát triển của công nghệ đã làm cho laptop trở thành một sản phẩm phổ biến và thiết yếu trong cuộc sống hàng ngày.

Chương 2: Cơ sở lý thuyết

2.1. Lý thuyết cơ sở dữ liệu

Cơ sở dữ liệu (Database) là một hệ thống tổ chức, lưu trữ và quản lý dữ liệu. Nó giúp chúng ta lưu trữ thông tin một cách có tổ chức để dễ dàng truy cập và sử dụng. Dưới đây là một số khái niệm quan trọng trong lý thuyết cơ sở dữ liệu:

Bảng (Table): Là cấu trúc cơ bản của cơ sở dữ liệu, thường tương ứng với một loại dữ liệu cụ thể như danh sách người dùng, sản phẩm, hoặc đơn hàng. Bảng gồm các cột (fields) và hàng (rows) để lưu trữ dữ liệu.

Khóa chính (Primary Key): Là một hoặc nhiều cột trong bảng được sử dụng để định danh duy nhất từng dòng trong bảng. Điều này đảm bảo tính duy nhất của dữ liệu và giúp quản lý liên kết giữa các bảng.

Câu lệnh SQL (Structured Query Language): Là ngôn ngữ được sử dụng để tương tác với cơ sở dữ liệu. SQL cho phép bạn truy vấn dữ liệu, cập nhật dữ liệu, thêm mới và xóa dữ liệu.

SQL Server Management Studio (SSMS):

SSMS là một ứng dụng phát triển bởi Microsoft, được sử dụng để quản lý cơ sở dữ liệu SQL Server. Dưới đây là các chức năng và tính năng cụ thể của SSMS:

Kết nối cơ sở dữ liệu (Connect to Database): SSMS cho phép bạn thiết lập kết nối đến cơ sở dữ liệu SQL Server. Bạn cần cung cấp thông tin đăng nhập, như tên máy chủ, tên người dùng, và mật khẩu để truy cập cơ sở dữ liệu.

Bảng điều khiển (Object Explorer): Đây là khung làm việc chính trong SSMS. Nó cho phép bạn truy cập và quản lý các đối tượng cơ sở dữ liệu như bảng, thủ tục lưu trữ, chức năng, và người dùng.

Trình soạn thảo truy vấn (Query Editor): SSMS cung cấp một trình soạn thảo truy vấn mạnh mẽ, giúp bạn viết và thử nghiệm câu lệnh SQL. Bạn có thể sử dụng trình soạn thảo này để truy vấn dữ liệu hoặc thực hiện các tác vụ như tạo bảng và thủ tục lưu trữ.

Tạo và quản lý bảo mật (Security): SSMS cho phép bạn quản lý quyền truy cập và bảo mật trong cơ sở dữ liệu bằng cách thiết lập các vai trò và phân quyền.

Sao lưu và phục hồi (Backup and Restore): SSMS cho phép bạn sao lưu dữ liệu và phục hồi cơ sở dữ liệu để bảo vệ dữ liệu khỏi mất mát.

Tối ưu hóa truy vấn (Query Optimization): SSMS cung cấp các công cụ và tính năng giúp bạn tối ưu hóa truy vấn SQL để cải thiện hiệu suất cơ sở dữ liệu.

Như vậy, lý thuyết cơ sở dữ liệu và SQL Server Management Studio là hai phần quan trọng khi làm việc với SQL Server. Hiểu về lý thuyết cơ sở dữ liệu giúp bạn thiết kế cơ sở dữ liệu chính xác và hiệu quả, trong khi SSMS là công cụ mạnh mẽ giúp bạn quản lý và tương tác với cơ sở dữ liệu một cách dễ dàng và hiệu quả.

2.2. Ngôn ngữ lập trình

2.2.1. Ngôn ngữ C#

a) Định nghĩa

C# (C Sharp, đọc là "xi-sáp") là một ngôn ngữ lập trình hướng đối tượng đa năng, mạnh mẽ được phát triển bởi Microsoft, C# là phần khởi đầu cho kế hoạch .NET của họ. Tên của ngôn ngữ bao gồm ký tự thăng theo Microsoft nhưng theo ECMA là C#, chỉ bao gồm dấu số thường. Microsoft phát triển C# dựa trên C++ và Java. C# được miêu tả là ngôn ngữ có được sự cân bằng giữa C++, Visual Basic, Delphi và Java.

C# được thiết kế chủ yếu bởi Anders Hejlsberg kiến trúc sư phần mềm nổi tiếng với các sản phẩm Turbo Pascal, Delphi, J++, WFC. Phiên bản gần đây nhất là 9.0, được phát hành vào năm 2020 cùng với Visual Studio 2019 phiên bản 16.8.

b) Đặc trưng của ngôn ngữ C#

C# loại bỏ một vài sự phức tạp và rối rắm của những ngôn ngữ như Java và c++, bao gồm việc loại bỏ những macro, những template, đa kế thừa, và lớp cơ sở ảo (virtual base class).

Ngôn ngữ C# đơn giản vì nó dựa trên nền tảng C và C++. Nếu chúng ta thân thiện với C và C++ hoặc thậm chí là Java, chúng ta sẽ thấy C# khá giống về diện mạo, cú pháp, biểu thức, toán tử và những chức năng khác được lấy trực tiếp từ ngôn ngữ C và C++, nhưng nó đã được cải tiến để làm cho ngôn ngữ đơn giản hơn.

C# là ngôn ngữ hiện đại

Điều gì làm cho một ngôn ngữ hiện đại? Những đặc tính như là xử lý ngoại lệ, thu gom bộ nhớ tự động, những kiểu dữ liệu mở rộng, và bảo mật mã nguồn là những đặc tính được mong đợi trong một ngôn ngữ hiện đại. C# chứa tất cả những đặc tính trên. Nếu là người mới học lập trình có thể chúng ta sẽ cảm thấy những đặc tính trên phức tạp và khó hiểu. Tuy nhiên, cũng đừng lo lắng chúng ta sẽ dần dần được tìm hiểu những đặc tính qua các nội dung khoá học này.

C# là một ngôn ngữ lập trình thuần hướng đối tượng

Lập trình hướng đối tượng (OOP: Object-oriented programming) là một phương pháp lập trình có 4 tính chất. Đó là tính trừu tượng (abstraction), tính đóng gói (encapsulation), tính đa hình (polymorphism) và tính kế thừa (inheritance). C# hỗ trợ cho chúng ta tất cả những đặc tính trên.

C# là một ngôn ngữ ít từ khóa

C# là ngôn ngữ sử dụng giới hạn những từ khóa. Phần lớn các từ khóa được sử dụng để mô tả thông tin. Chúng ta có thể nghĩ rằng một ngôn ngữ có nhiều từ khóa thì sẽ mạnh hơn. Điều này không phải sự thật, ít nhất là trong trường hợp ngôn ngữ C#, chúng ta có thể tìm thấy rằng ngôn ngữ này có thể được sử dụng để làm bất cứ nhiệm vụ nào.

c) Ưu điểm, nhược điểm của C#

Ưu điểm:

- Dễ học hơn c ++
- Dễ đọc mã hơn c ++
- Phát triển nhanh hơn và có khả năng ít lỗi hơn so với c ++ hoặc java (bạn có các loại không dấu, bạn có ref / out, bạn có thể tạo các loại giá trị của riêng mình, bạn có những thứ hữu ích khác mà java bỏ qua có nghĩa là bạn ít nhảy qua vòng có nghĩa là độ phức tạp mã ít cần thiết hơn).
- Tất cả mọi thứ được truyền bằng tham chiếu ngoại trừ các loại giá trị, theo mặc định
- Công cụ thu gom rác dọn dẹp đồ vật một khi chúng không còn được sử dụng, vì vậy bạn không phải tự mình theo dõi mọi thứ
- Chương trình được biên dịch thành các nhị phân riêng được tối ưu hóa cho nền tảng khi nó được chạy (và nó chạy chậm hơn một chút so với mã c ++).
- Dễ dàng tạo nhiều chủ đề
- Có nhiều phương tiện tạm dừng các luồng để chờ tín hiệu và như vậy
- lock (someObject) {code}, giống như java được đồng bộ hóa nhưng có thể được sử dụng ở bất cứ đâu và yêu cầu một đối tượng.
- P / Gọi dễ sử dụng hơn JNI rất nhiều - nhưng có lẽ chúng tôi sẽ không sử dụng nó vì chúng tôi muốn trở thành đa nền tảng
- Có một IDE miễn phí tuyệt vời (visual c # express) - nhưng chỉ trên windows (xem khuyết điểm).
- Bạn có thể tạo các loại giá trị, theo mặc định là truyền theo giá trị thay vì truyền bằng tham chiếu, bằng cách tạo cấu trúc thay vì một lớp.
- Bạn có các từ khóa 'ref' và 'out' cho phép bạn chuyển tham chiếu đến tham số cho hàm, với ý nghĩa là tham số đó phải được gán bởi hàm trước khi trả về. Về cơ bản, ref / out cho phép hàm sửa đổi biến được truyền dưới dạng tham số, giống như chuyển tham chiếu trong c ++.

- Đa nền tảng với mono và / nhưng mono vẫn đang được cải thiện
- Có số nguyên không dấu (c ++, java không)
- Nếu chương trình gặp sự cố, nó sẽ bật lên một hộp thoại cho người dùng biết mã ở đâu và tại sao (trên windows) hoặc ghi dấu vết ngăn xếp vào bàn điều khiển (với mono nếu chạy với --debug).
- Các chương trình được biên dịch thành các tệp .exe và không cần phải biên dịch lại cho các HĐH khác - mono có thể chạy trực tiếp exes .NET.

Nhược điểm:

- Sử dụng nhiều bộ nhớ hơn c ++
- Trình thu gom rác sử dụng chu kỳ và bộ nhớ CPU - (nhưng hầu như không có, ít nhất là thời gian CPU)
- Một số thứ không hoạt động trong các chức năng gọi đơn trong một số cửa sổ sẽ không tồn tại trên linux, việc triển khai các biểu mẫu đơn của windows cần có cách giải quyết trong mã (nhưng có lẽ chúng ta sẽ không sử dụng chúng)
- Mono chưa hoàn hảo
- Phải sử dụng .net 2.0 cho thuốc generic (trong số những thứ khác), 1.1. không có chúng. Không thể sử dụng bất cứ thứ gì mới hơn 2.0 vì hỗ trợ trả 3.0 trở lên cho windows 98, ME và có thể 2000.
- Không thể khai báo các chức năng được đồng bộ hóa
- Không có IDE tốt trên các cửa sổ không. Eclipse có plugin ac # nhưng nó có thể không hỗ trợ hầu hết các tính năng của Eclipse.
- Không có con trỏ, nhưng chúng được thay thế bởi các tham số ref và out, vì vậy chúng không cần nhiều
- Phương thức gọi thông qua các đại biểu chậm hơn đáng kể so với cách gọi phương thức bình thường (hoặc con trỏ hàm trong c ++, có lẽ)
- Yêu cầu (phiên bản chính xác của) .NET framework phải được cài đặt để chạy chương trình, đây là một bản tải xuống vài giờ khi quay số.

Chương 3: Phân tích và thiết kế cơ sở dữ liệu

3.1. Thiết kế cơ sở dữ liệu

Bảng 1: Tbl_NhaCungCap

STT	Tên thuộc tính	Kiểu dữ liệu	Diễn giải	Chú thích
1	idSupplier	nvarchar(50)	Mã nhà cung cấp	Khoá chính
2	nameSupplier	nvarchar(50)	Tên nhà cung cấp	

Bảng 2: Tbl_NhanVien

STT	Tên thuộc tính	Kiểu dữ liệu	Diễn giải	Chú thích
1	idEmployee	nvarchar(50)	Mã nhân viên	Khoá chính
2	nameEmployee	nvarchar(50)	Tên nhân viên	
3	serviceEmployee	nvarchar(50)	Vai trò nhân viên	
4	birthday	date	Sinh nhật	
5	phoneNumber	char(10)	Số điện thoại	
6	salaryEmployee	float	Lương nhân viên	
7	totalSales	float	Doanh thu	
8	quantityBillPay	int	Tổng hoá đơn	
9	idUser	nvarchar(50)	Mã người dùng	Khoá ngoại

Bảng 3: Tbl_SanPham

STT	Tên thuộc tính	Kiểu dữ liệu	Diễn giải	Chú thích
1	idProduct	nvarchar(50)	Mã sản phẩm	Khoá chính

2	nameProduct	nvarchar(max)	Tên sản phẩm	
3	unitPrice	float	Đơn giá	
4	quantityProduct	int	Số lượng sản phẩm	
5	statusProduct	nvarchar(max)	Tình trạng sản phẩm	
6	idTypeProduct	int	Mã danh mục sản phẩm	Khoá ngoại
7	idManufacture	int	Mã hãng sản phẩm	Khoá ngoại
8	image	nvarchar(max)	Hình ảnh	

Bảng 4: Tbl_TaiKhoan

STT	Tên thuộc tính	Kiểu dữ liệu	Diễn giải	Chú thích
1	idUser	nvarchar(50)	Mã người dùng	Khoá chính
2	UserName	nvarchar(50)	Tên tài khoản	
3	Password	nvarchar(100)	Mật khẩu	
4	idQuyền	int	Mã quyền	Khoá ngoại

Bảng 5: Tbl_PhieuNhapKho

STT	Tên thuộc tính	Kiểu dữ liệu	Diễn giải	Chú thích
1	idReceipt	int	Mã biên lai	Khoá chính
2	idReceipt	nvarchar(50)	Mã nhà cung cấp	Khoá ngoại
3	dateReceipt	date	Ngày biên lai	
4	sumPay	float	Tổng tiền	

Bảng 6: Tbl_PhieuYeuCauNhapHang

STT	Tên thuộc tính	Kiểu dữ liệu	Diễn giải	Chú thích
1	idCoupon	int	Mã phiếu mua hàng	Khoá chính
2	dateAdded	date	Ngày thêm	
3	intoMoney	float	Tiền mua	
4	idSupplier	nvarchar(50)	Mã nhà cung cấp	Khoá ngoại

Bảng 7: Tbl_KhuyenMai

STT	Tên thuộc tính	Kiểu dữ liệu	Diễn giải	Chú thích
1	idPromotion	int	Mã khuyến mãi	Khoá chính
2	namePromotions	nvarchar(max)	Tên khuyến mãi	
3	status	nvarchar(50)	Tình trạng	
4	startDay	date	Ngày bắt đầu	
5	endDay	date	Ngày kết thúc	

Bảng 8: Tbl_KhachHang

STT	Tên thuộc tính	Kiểu dữ liệu	Diễn giải	Chú thích
1	idCustomer	nvarchar(50)	Mã khách hàng	Khoá chính
2	nameCustomer	nvarchar(50)	Tên khách hàng	
3	address	nvarchar(max)	Địa chỉ	
4	phoneNumber	nvarchar(10)	Số điện thoại	

5	email	nvarchar(50)	Email	
---	-------	--------------	-------	--

Bảng 9: Tbl_HoaDon

STT	Tên thuộc tính	Kiểu dữ liệu	Diễn giải	Chú thích
1	idBill	int	Mã hoá đơn	Khoá chính
2	dateBill	date	Ngày tạo	
3	sumPay	float	Tổng tiền	
4	typePay	nvarchar(50)	Loại thanh toán	
5	discountMoney	float	Tiền đã giảm	
6	pay	float	Tiền phải trả	
7	status	nvarchar(max)	Trạng thái	
8	idEmployee	nvarchar(50)	Mã nhân viên	Khoá ngoại
9	idCustomer	nvarchar(50)	Mã khách hàng	Khoá ngoại

Bảng 10: Tbl_HangSP

STT	Tên thuộc tính	Kiểu dữ liệu	Diễn giải	Chú thích
1	idManufacture	int	Mã hãng sản phẩm	Khoá chính
2	nameManufacture	nvarchar(max)	Tên hãng sản phẩm	

Bảng 11: Tbl_DanhMucLapTop

STT	Tên thuộc tính	Kiểu dữ liệu	Diễn giải	Chú thích
1	idTypeProduct	int	Mã danh mục laptop	Khoá chính

2	nameTypeProduct	nvarchar(50)	Tên danh mục laptop	
---	-----------------	--------------	---------------------	--

Bảng 12: Tbl_ChiTietPhieuYcNhapHang

STT	Tên thuộc tính	Kiểu dữ liệu	Diễn giải	Chú thích
1	idProduct	nvarchar(50)	Mã sản phẩm	Khoá chính, Khoá ngoại
2	idCoupon	int	Mã phiếu mua hàng	Khoá chính, Khoá ngoại
3	quantityImport	int	Số lượng nhập	
4	unitPrice	float	Đơn giá	
5	intoMoney	float	Tổng tiền	

Bảng 13: Tbl_ChiTietNhapKho

STT	Tên thuộc tính	Kiểu dữ liệu	Diễn giải	Chú thích
1	idReceipt	int	Mã biên lai	Khoá chính, Khoá ngoại
2	quantityRequest	int	Số lượng yêu cầu	
3	quantityActual	int	Số lượng có sẵn	
4	unitPrice	float	Đơn giá	
5	intoMoney	float	Tổng tiền	
6	idProduct	nvarchar(50)	Mã sản phẩm	Khoá chính, Khoá ngoại

Bảng 14: Tbl_ChiTietKhuyenMai

STT	Tên thuộc tính	Kiểu dữ liệu	Diễn giải	Chú thích
1	discount	int	Phần trăm giảm giá	

2	status	nvarchar(50)	Trạng thái	
3	idProduct	nvarchar(50)	Mã sản phẩm	Khoá chính, Khoá ngoại
4	idPromotion	int	Mã khuyến mãi	Khoá chính, Khoá ngoại

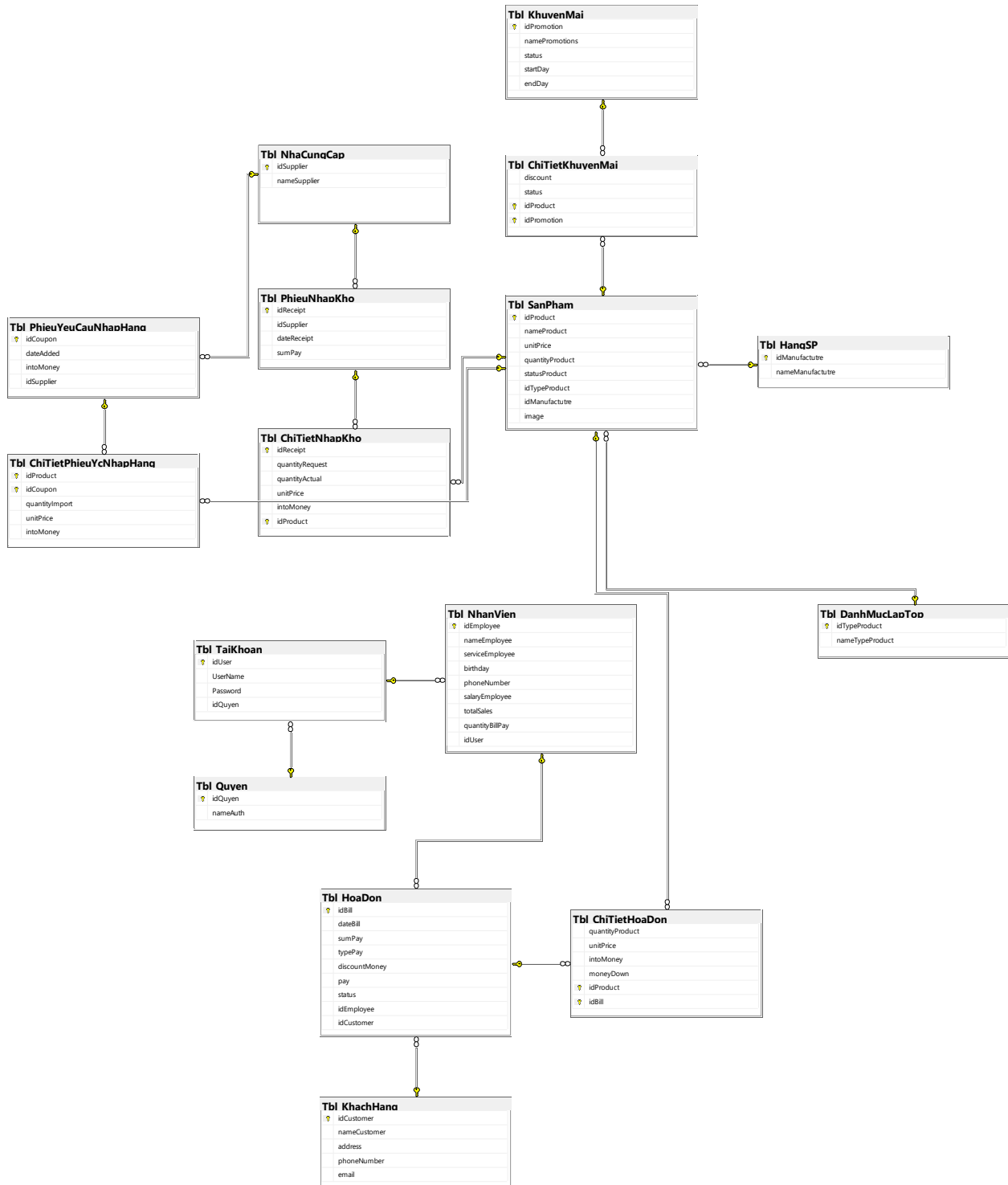
Bảng 15: Tbl_ChiTietHoaDon

STT	Tên thuộc tính	Kiểu dữ liệu	Diễn giải	Chú thích
1	quantityProduct	int	Số lượng sản phẩm	Khoá chính, Khoá ngoại
2	unitPrice	float	Đơn giá	
3	intoMoney	float	Tổng tiền	
4	moneyDown	float	Tiền đã giảm	
5	idProduct	nvarchar(50)	Mã sản phẩm	Khoá chính, Khoá ngoại
6	idBill	nvarchar(50)	Mã hoá đơn	Khoá chính, Khoá ngoại

Bảng 16: Tbl_Quyền

STT	Tên thuộc tính	Kiểu dữ liệu	Diễn giải	Chú thích
1	idQuyền	int	Mã quyền	Khoá chính
2	nameAuth	nvarchar(50)	Tên quyền	

3.2. Mô hình Class Diagram



4.2. Thiết kế hệ thống

a) Use Case tổng quát



b) Use Case chức năng

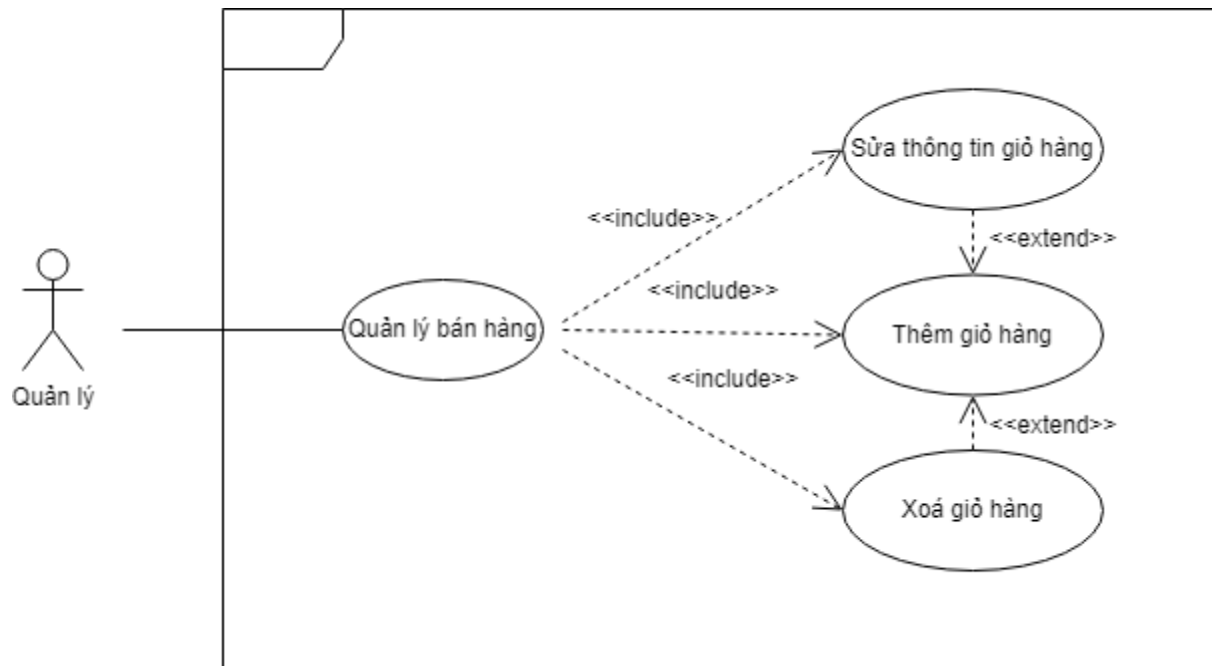
Use Case chức năng Quản lý bán hàng:

Use Case Number:	UC1
Use Case Name:	Quản lý bán hàng

Actor (s):	Quản lý	
Maturity:	Focused	
Basic Course of Events:	Actor action	System Response
	1. Đăng nhập dưới quyền của admin	
		2. Hệ thống kiểm tra tài khoản và mật khẩu nhập vào. E1
		3. Hệ thống sẽ điều hướng đến giao diện trang chủ
	4. Admin nhấn vào nút Bán hàng	
		5. Hệ thống sẽ hiển thị các thông tin trong mục bán hàng
	6. Admin có thể thao tác hành động. A1	
	7. Chọn vào textbox bên cạnh icon kính lúp	
	8. Nhập tên sản phẩm cần tìm	
		9. Hiển thị thông tin sản phẩm tìm thấy
	10. Admin chọn một chức năng. A2, A3	
	11. Chọn nút Hoá đơn	

		12. Hệ thống hiển thị ra thông tin danh sách và chi tiết hoá đơn
	13. Chọn thanh toán, khách hàng và nhấn thanh toán	
		14. Hệ thống hiển thị thông báo thanh toán thành công và in ra thông tin hoá đơn
Alternative Paths:	A1	
	Actor Action	System Response
	1. Chọn sản phẩm, nhập số lượng rồi chọn thêm vào giỏ hàng	
		2. Hệ thống hiển thị thông báo thêm thành công
	3. Chọn thanh toán, khách hàng và nhấn thanh toán	
		4. Hệ thống hiển thị thông báo thanh toán thành công và in ra thông tin hoá đơn
	A2	
	Actor Action	System Response
	1. Chọn sản phẩm, nhập số lượng rồi chọn thêm vào giỏ hàng	

		2. Hệ thống hiển thị thông báo thêm thành công
	3. Nhập lại số lượng sản phẩm và nhấn sửa CT	
		4. Hệ thống hiển thị thông báo sửa thành công
	A3	
	Actor Action	System Response
	1. Chọn sản phẩm, nhập số lượng rồi chọn thêm vào giỏ hàng	
		2. Hệ thống hiển thị thông báo thêm thành công
	3. Nhấn Xóa CT	
		4. Hệ thống hiển thị thông báo xóa thành công
Exception Paths:	E1. Hệ thống sẽ kiểm tra nếu tài khoản và mật khẩu sai thì thông báo để người dùng nhập lại	

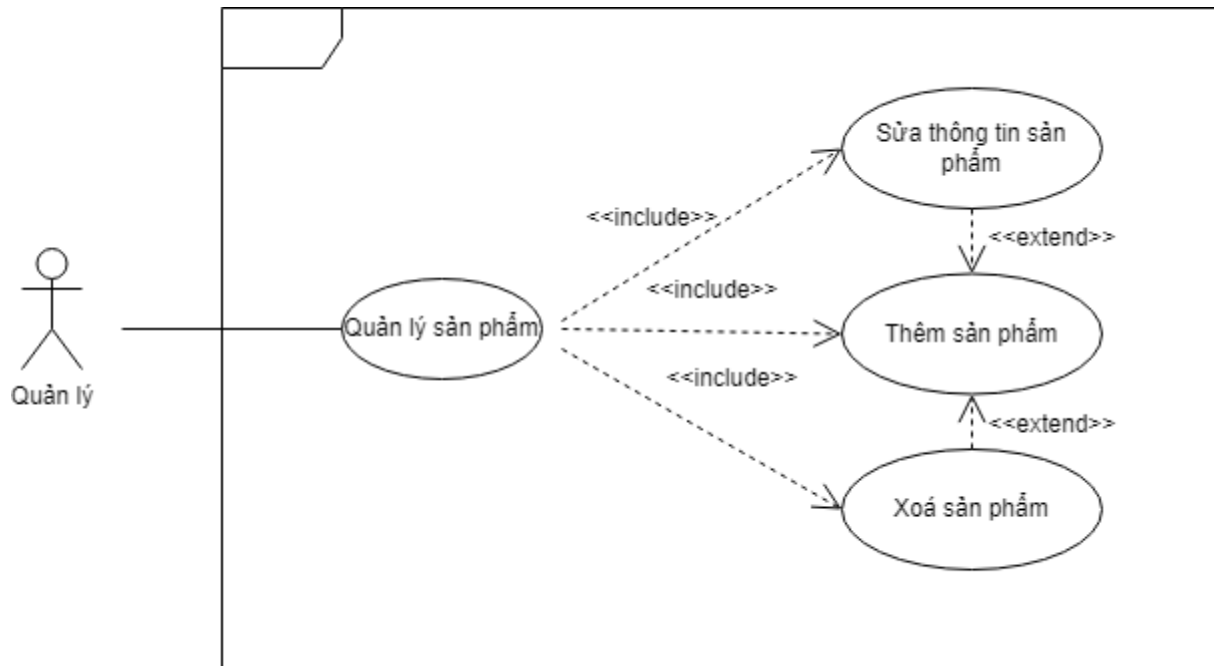


Use Case chức năng quản lý sản phẩm:

Use Case Number:	UC2	
Use Case Name:	Quản lý sản phẩm	
Actor (s):	Quản lý	
Maturity:	Focused	
Basic Course of Events:	Actor action	System Response
	1. Đăng nhập dưới quyền của admin	
		2. Hệ thống kiểm tra tài khoản và mật khẩu nhập vào. E1
		3. Hệ thống sẽ điều hướng đến giao diện trang chủ

	4. Admin nhấn vào nút Sản phẩm	
		5. Hệ thống sẽ hiển thị các thông tin trong mục sản phẩm
	6. Admin có thể thao tác hành động. A1	
	7. Chọn vào textbox bên cạnh icon kính lúp	
	8. Nhập tên sản phẩm cần tìm	
		9. Hiển thị thông tin sản phẩm tìm thấy
	10. Admin chọn một chức năng. A2, A3	
	11. Chọn nút Loại sản phẩm	
		12. Hệ thống hiển thị ra thông tin danh sách sản phẩm từ loại sản phẩm
Alternative Paths:	A1	
	Actor Action	System Response
	1. Nhập thông tin sản phẩm và chọn thêm	
		2. Hệ thống hiển thị thông báo thêm thành công
	A2	
	Actor Action	System Response

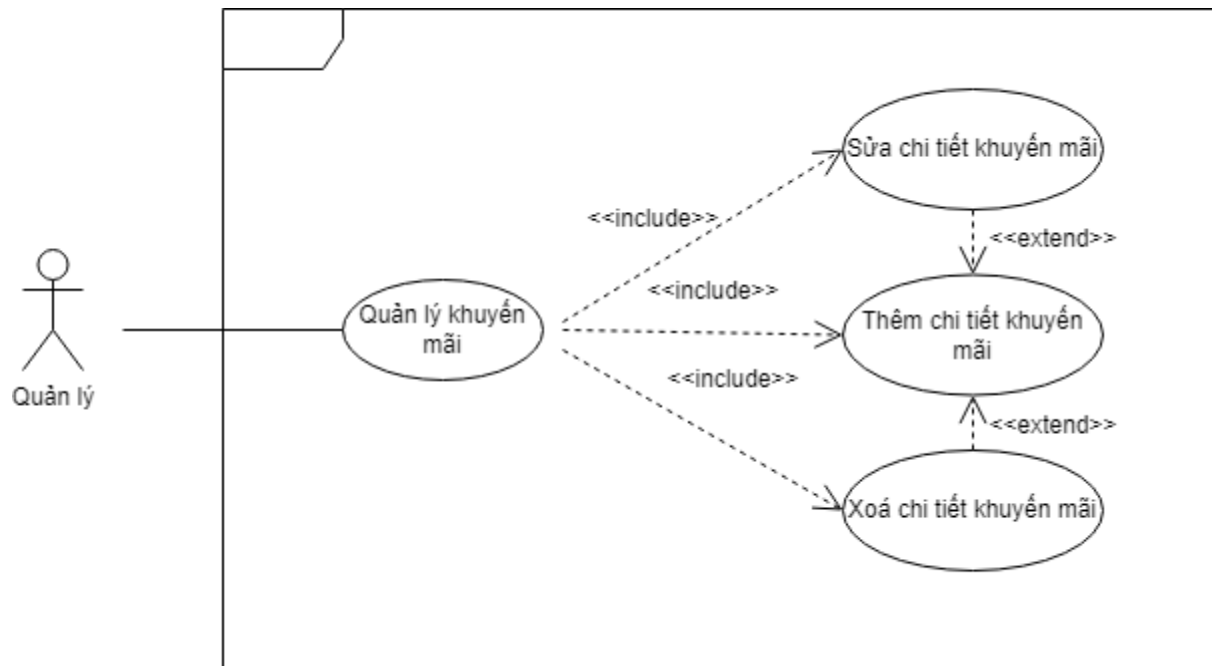
	1. Chọn sản phẩm, nhập số lượng rồi chọn sửa	
		2. Hệ thống hiển thị thông báo sửa thành công
	A3	
	Actor Action	System Response
	1. Chọn sản phẩm rồi nhấn xoá	
		2. Hệ thống hiển thị thông báo xoá thành công
Exception Paths:	E1. Hệ thống sẽ kiểm tra nếu tài khoản và mật khẩu sai thì thông báo để người dùng nhập lại	



Use Case chức năng quản lý khuyến mãi:

Use Case Number:	UC3	
Use Case Name:	Quản lý khuyến mãi	
Actor (s):	Quản lý	
Maturity:	Focused	
Basic Course of Events:	Actor action	System Response
	1. Đăng nhập dưới quyền của admin	
		2. Hệ thống kiểm tra tài khoản và mật khẩu nhập vào. E1
		3. Hệ thống sẽ điều hướng đến giao diện trang chủ
	4. Admin nhấn vào nút Khuyến mãi	
		5. Hệ thống sẽ hiển thị các thông tin danh sách và chi tiết khuyến mãi
	6. Admin có thể thao tác hành động. A1, A2	
	7. Chọn vào textbox bên cạnh nút tìm kiếm	
	8. Nhập tên khuyến mãi cần tìm	
		9. Hiển thị thông tin khuyến mãi tìm thấy

Alternative Paths:	A1	
	Actor Action	System Response
	1. Chọn tạo CTKM	
		2. Hệ thống hiển thị giao diện chương trình khuyến mãi
	3. Nhập các thông tin cần thiết và các thao tác sau đó chọn lưu để hoàn tất tạo CTKM	
		4. Hệ thống hiển thị thông báo thêm thành công
	A2	
	Actor Action	System Response
	1. Chọn xuất CTKM	
		2. Hệ thống hiển thị thông báo xuất thành công
Exception Paths:	E1. Hệ thống sẽ kiểm tra nếu tài khoản và mật khẩu sai thì thông báo để người dùng nhập lại	

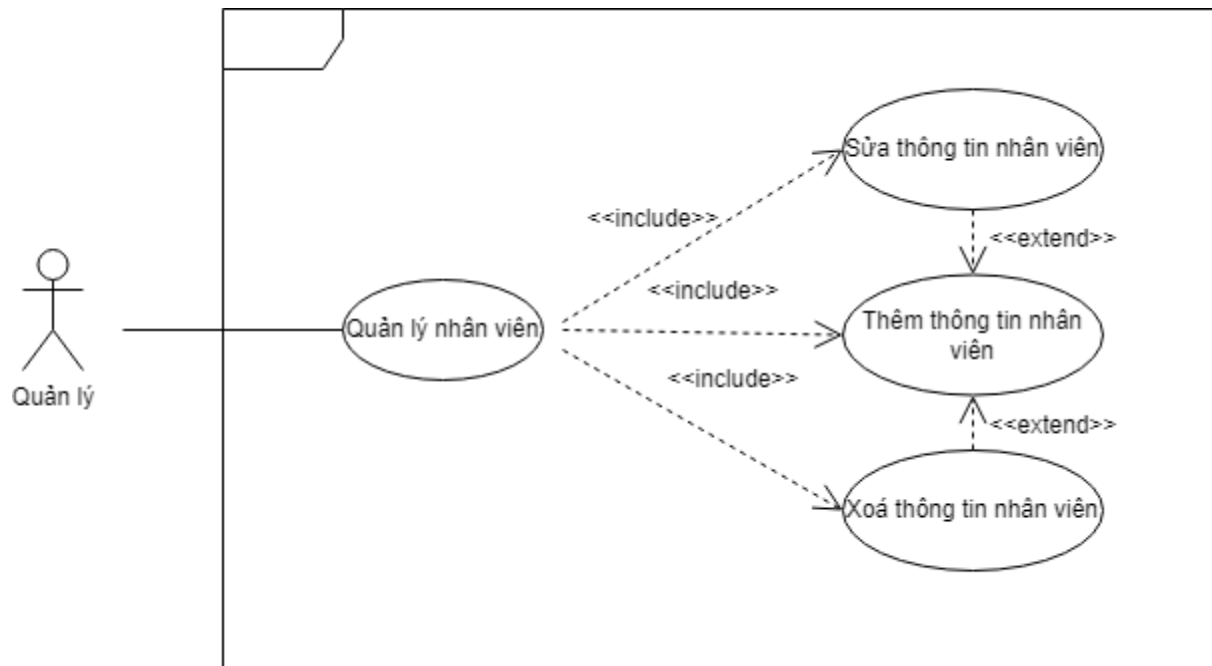


Use Case chức năng quản lý nhân viên:

Use Case Number:	UC4	
Use Case Name:	Quản lý nhân viên	
Actor (s):	Quản lý	
Maturity:	Focused	
Basic Course of Events:	Actor action	System Response
	1. Đăng nhập dưới quyền của admin	
		2. Hệ thống kiểm tra tài khoản và mật khẩu nhập vào. E1
		3. Hệ thống sẽ điều hướng đến giao diện trang chủ

	4. Admin nhấn vào nút Nhân viên	
		5. Hệ thống sẽ hiển thị các thông tin về nhân viên
	6. Admin có thể thao tác hành động. A1, A2	
	7. Chọn vào textbox ở sau label Từ khoá	
	8. Nhập tên nhân viên cần tìm	
		9. Hiển thị thông tin nhân viên tìm thấy
Alternative Paths:	A1	
	Actor Action	System Response
	1. Nhập thông tin nhân viên và chọn thêm	
		2. Hệ thống hiển thị thông báo thêm thành công
	3. Chọn thông tin nhân viên cần sửa	
		4. Hệ thống hiển thị thông tin của nhân viên
	5. Nhập thông tin mới của nhân viên và nhấn sửa	
		6. Hệ thống hiển thị thông báo sửa thành công

	7. Chọn nhân viên cần xoá và chọn xoá	
		8. Hệ thống hiển thị thông báo xoá thành công
	A2	
	Actor Action	System Response
	1. Chọn nút Phân quyền	
		2. Hệ thống hiển thị thông tin nhân viên
	3. Chọn nhân viên cần phân quyền, vai trò của nhân viên và nhấn lưu	
		4. Hệ thống hiển thị thông báo lưu thành công
Exception Paths:	E1. Hệ thống sẽ kiểm tra nếu tài khoản và mật khẩu sai thì thông báo để người dùng nhập lại	

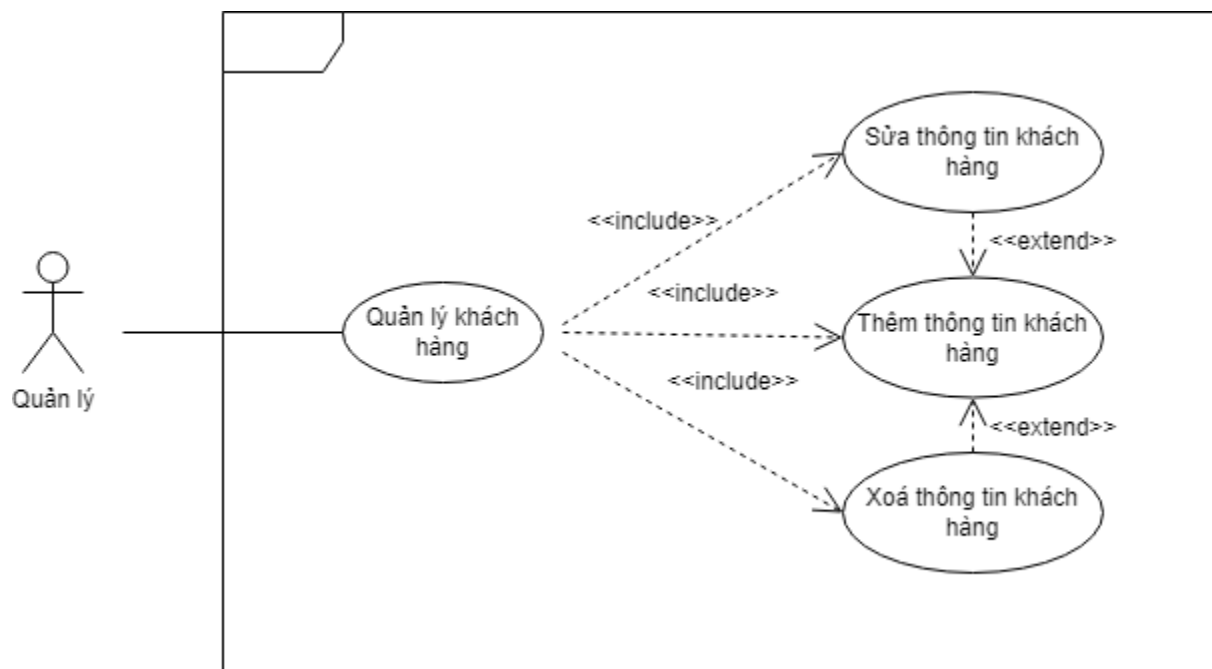


Use Case chức năng quản lý khách hàng:

Use Case Number:	UC5	
Use Case Name:	Quản lý khách hàng	
Actor (s):	Quản lý	
Maturity:	Focused	
Basic Course of Events:	Actor action	System Response
	1. Đăng nhập dưới quyền của admin	
		2. Hệ thống kiểm tra tài khoản và mật khẩu nhập vào. E1

		3. Hệ thống sẽ điều hướng đến giao diện trang chủ
	4. Admin nhấn vào nút Khách hàng	
		5. Hệ thống sẽ hiển thị các thông tin về khách hàng
	6. Admin có thể thao tác hành động. A1	
	7. Chọn vào textbox ở sau label Từ khoá	
	8. Nhập tên khách hàng cần tìm	
		9. Hiển thị thông tin khách hàng tìm thấy
Alternative Paths:	A1	
	Actor Action	System Response
	1. Nhập thông tin khách hàng và chọn thêm	
		2. Hệ thống hiển thị thông báo thêm thành công
	3. Chọn thông tin khách hàng cần sửa	
		4. Hệ thống hiển thị thông tin của khách hàng
	5. Nhập thông tin mới của khách hàng và nhấn sửa	

		6. Hệ thống hiển thị thông báo sửa thành công
	7. Chọn khách hàng cần xoá và chọn xoá	
		8. Hệ thống hiển thị thông báo xoá thành công
Exception Paths:	E1. Hệ thống sẽ kiểm tra nếu tài khoản và mật khẩu sai thì thông báo để người dùng nhập lại	

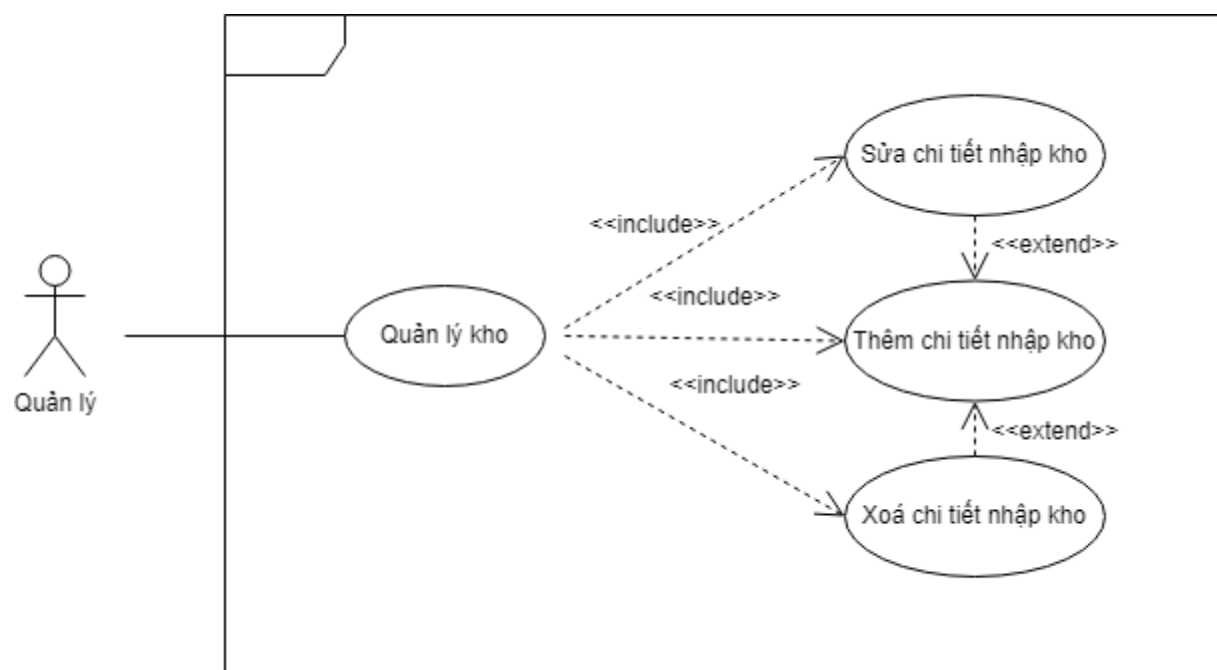


Use Case chức năng quản lý kho:

Use Case Number:	UC6
Use Case Name:	Quản lý kho
Actor (s):	Quản lý

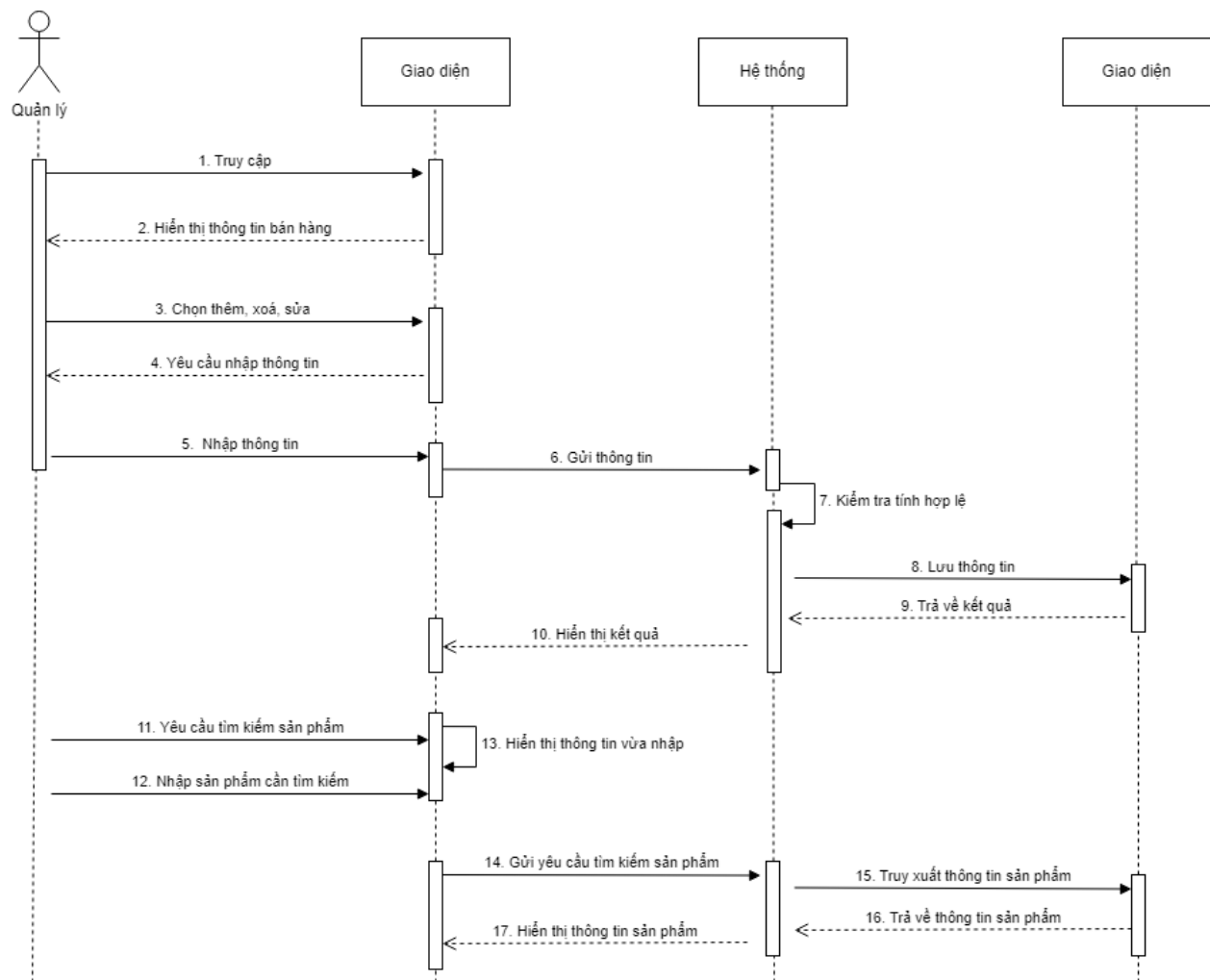
Maturity:	Focused	
Basic Course of Events:	Actor action	System Response
	1. Đăng nhập dưới quyền của admin	
		2. Hệ thống kiểm tra tài khoản và mật khẩu nhập vào. E1
		3. Hệ thống sẽ điều hướng đến giao diện trang chủ
	4. Admin nhấn vào nút Kho	
		5. Hệ thống sẽ hiển thị các thông tin phiếu nhập và chi tiết phiếu nhập kho
	6. Admin có thể thao tác hành động. A1, A2	
	7. Chọn vào textbox bên cạnh nút tìm kiếm	
	8. Nhập tên phiếu nhập cần tìm	
Alternative Paths:		9. Hiển thị thông tin phiếu nhập tìm thấy
	A1	
	Actor Action	System Response
	1. Chọn tạo PNK	

		2. Hệ thống hiển thị giao diện phiếu nhập kho
	3. Nhập các thông tin cần thiết và các thao tác sau đó chọn thanh toán để nhập sản phẩm vào kho	
		4. Hệ thống hiển thị thông báo thanh toán thành công
	A2	
	Actor Action	System Response
	1. Chọn xuất PNK	
		2. Hệ thống hiển thị thông báo xuất thành công
Exception Paths:	E1. Hệ thống sẽ kiểm tra nếu tài khoản và mật khẩu sai thì thông báo để người dùng nhập lại	

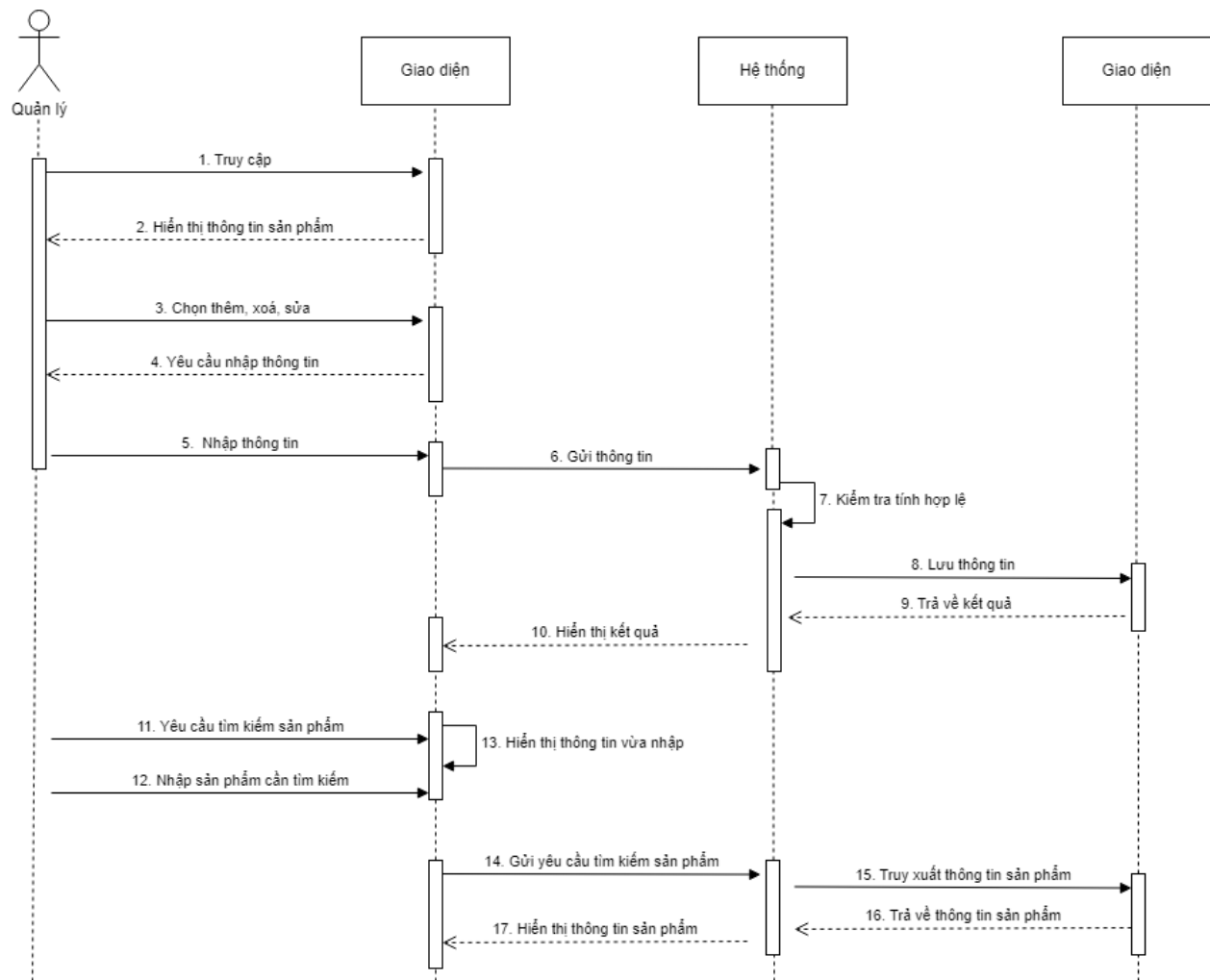


4.3. Lược đồ tuần tự

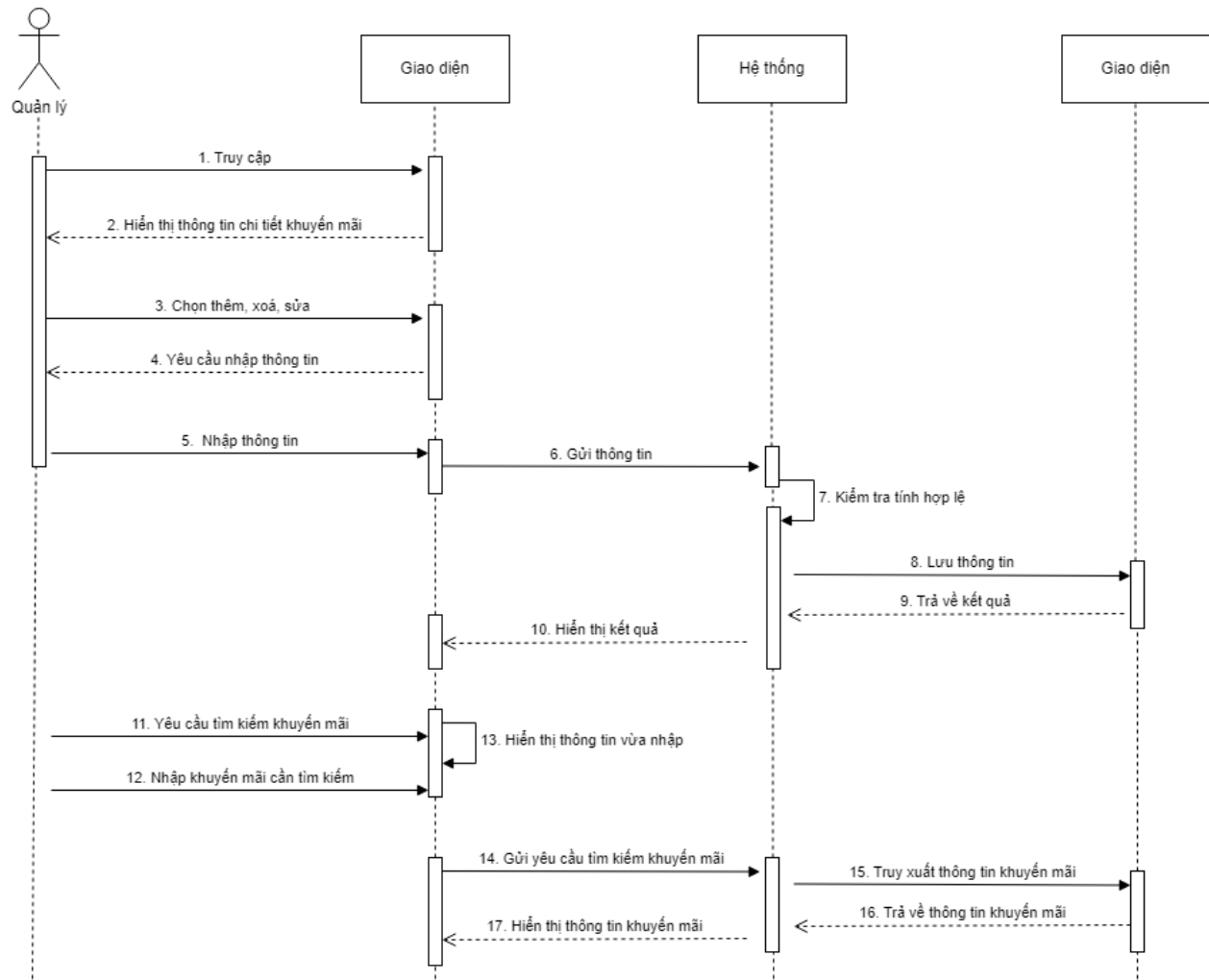
a) Chức năng quản lý bán hàng



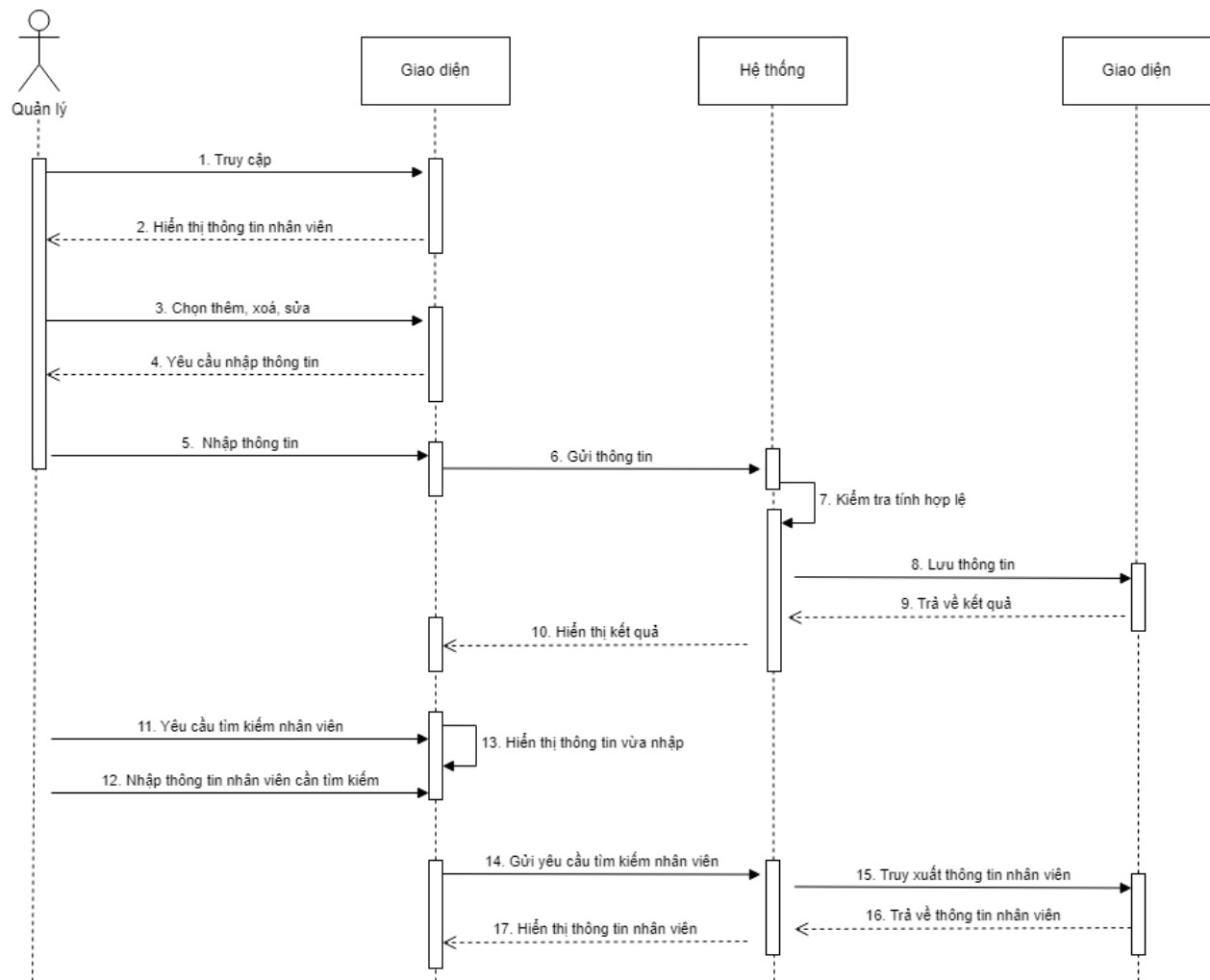
b) Chức năng quản lý sản phẩm



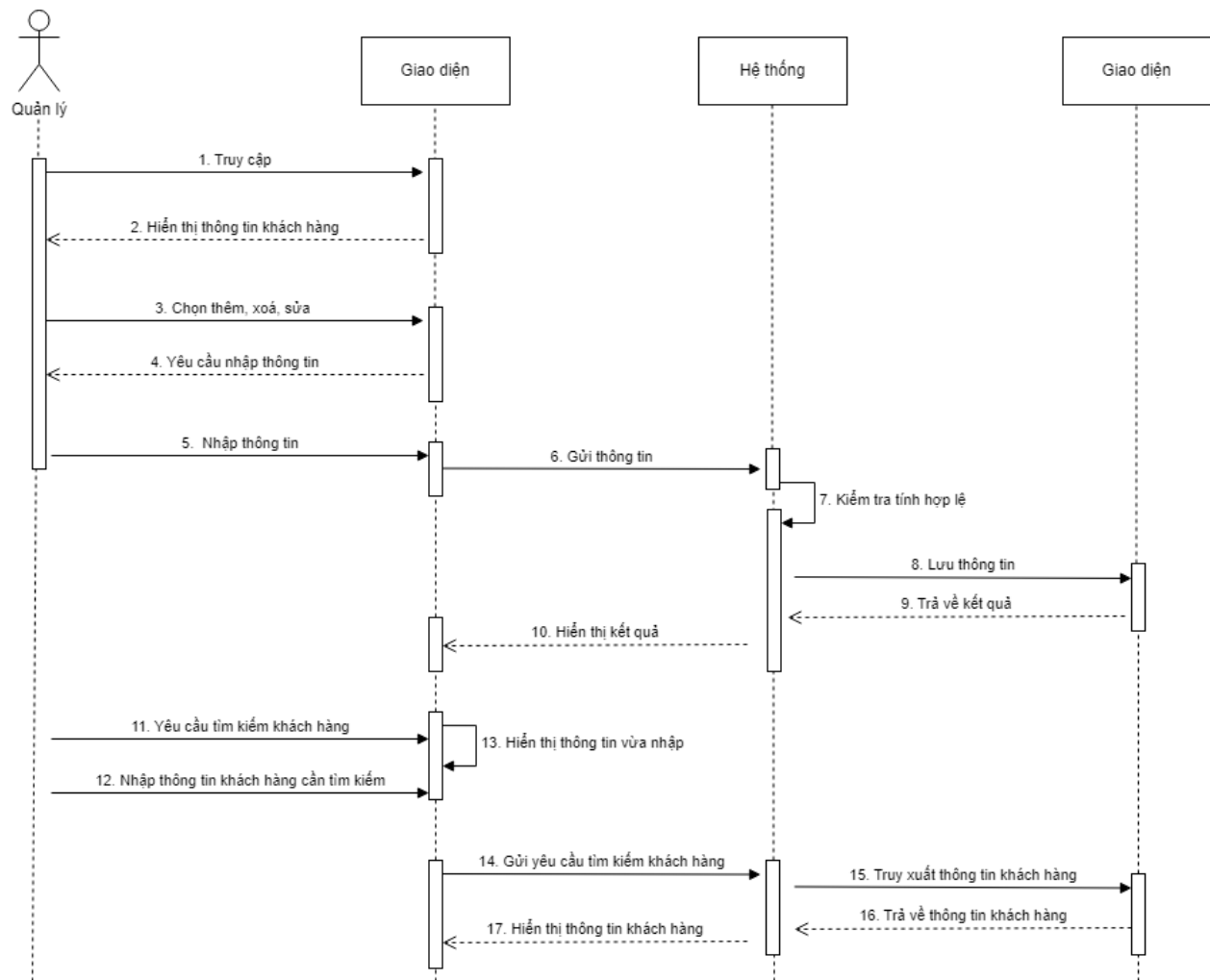
c) Chức năng quản lý khuyến mãi



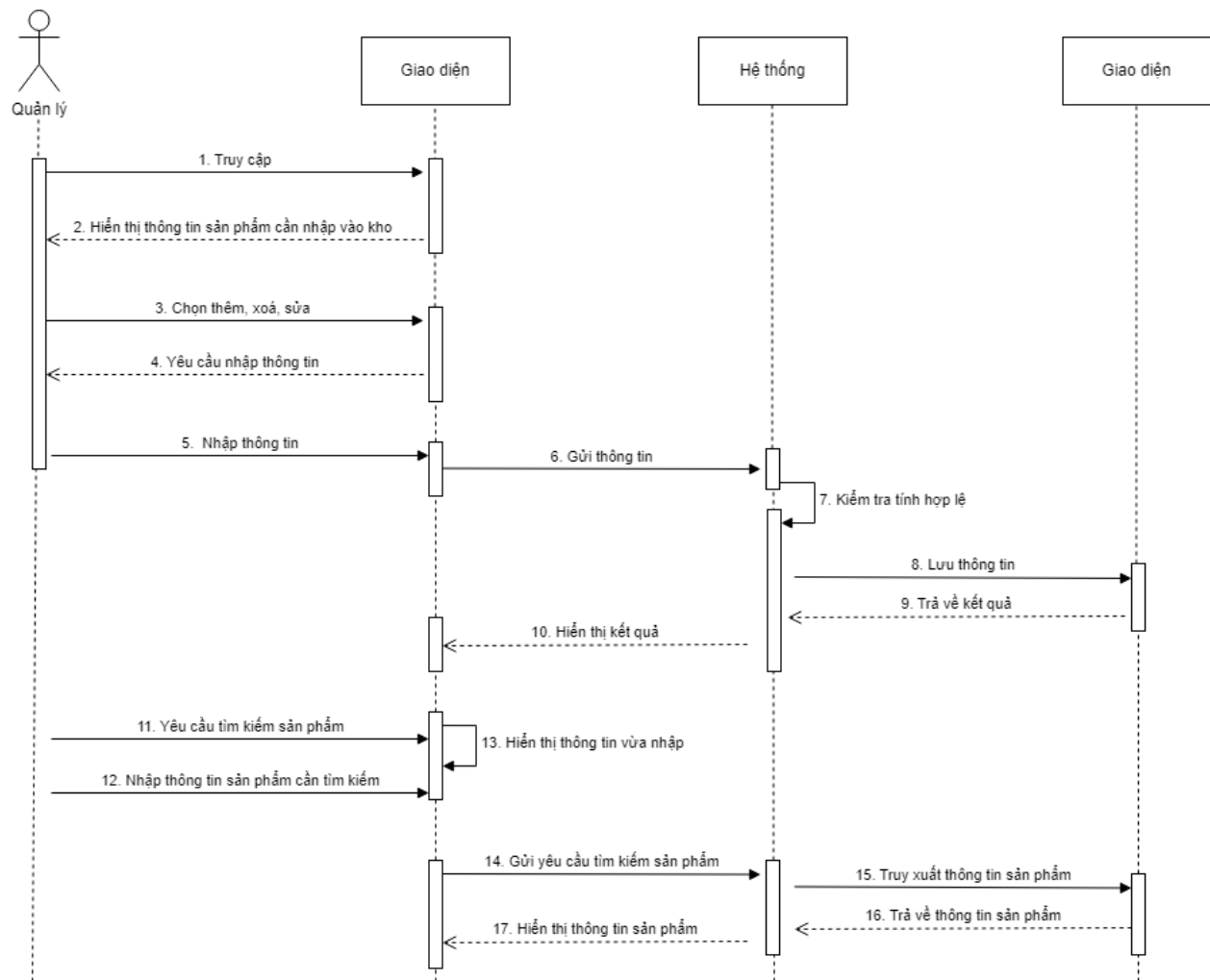
d) Chức năng quản lý nhân viên



e) Chức năng quản lý khách hàng



f) Chức năng quản lý kho



Chương 5: Thiết kế phần mềm

5.1. Chức năng đăng nhập



Đăng nhập

Đăng nhập hệ thống

Thông tin đăng nhập

Tài khoản

Mật khẩu

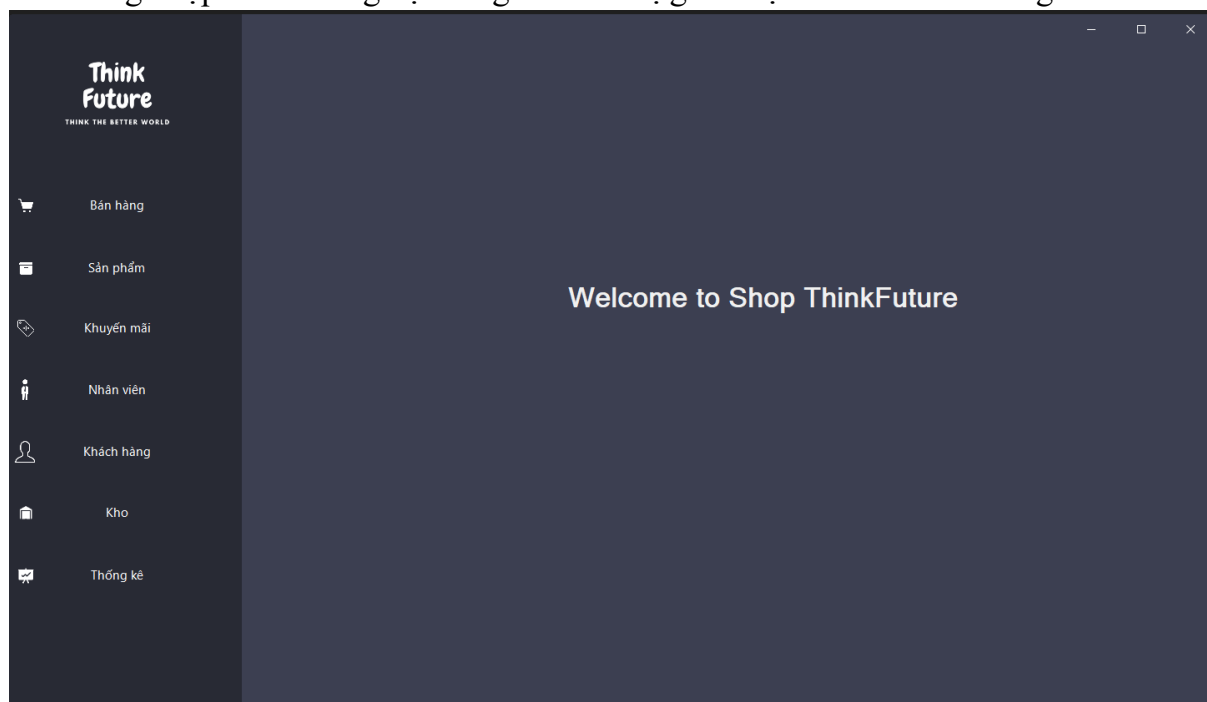
Đăng nhập Quên mật khẩu

Thoát chương trình

Màn hình đăng nhập xuất hiện khi khởi chạy chương trình.

Có 2 textfield gồm tài khoản và mật khẩu để người dùng nhập thông tin đăng nhập và có 3 button Đăng nhập, quên mật khẩu và thoát chương trình.

Nếu đăng nhập thành công hệ thống sẽ hiển thị giao diện chính của chương trình



Khi người dùng đăng nhập thành công thì sẽ xuất hiện menu chính như hình trên. Menu bên trái hiện các chức năng chính của phần mềm để người dùng có thể lựa chọn và tương tác.

5.2. Chức năng bán hàng

Think
Future
THINK THE BETTER WORLD

Bán hàng

Sản phẩm

Khuyến mãi

Nhân viên

Khách hàng

Kho

Thống kê

Bán hàng

Hoà đơn

Mua hàng

Danh sách sản phẩm

ID	Tên sản phẩm	Đơn giá	Số lượng tồn	Loại sản phẩm	Hãng sản phẩm
SP001	Tuf Gaming F...	21,000,000	11	Laptop gaming	Asus
SP002	Laptop Acer S...	22,000,000	5	Ultrabook m...	Acer
SP003	Laptop Dell Pr...	23,000,000	6	Laptop văn p...	Dell
SP004	Tuf Gaming A...	24,000,000	18	Laptop gaming	Asus
SP005	Macbook Pro ...	25,000,000	5	Ultrabook m...	Apple
SP006	Laptop Asus ...	27,899,000	5	Laptop gaming	Asus
SP007	Laptop Asus ...	33,990,000	5	Laptop gaming	Asus
SP008	Laptop ASUS ...	102,900,000	105	Laptop gaming	Asus
SP009	Laptop Acer ...	14,690,000	5	Laptop gaming	Acer
SP010	Laptop Gamin...	19,990,000	5	Laptop gaming	Acer
SP011	Laptop Acer P...	40,990,000	5	Laptop gaming	Acer
SP012	Laptop Asus ...	94,990,000	5	Laptop gaming	Asus

Giỏ hàng

ID	Tên sản phẩm	Đơn giá	Số lượng mua	Tổng tiền	Tiền giảm	Thành tiền
----	--------------	---------	--------------	-----------	-----------	------------

Hãng

Mã sản phẩm

Tên sản phẩm

SL Mua

Giá bán

% Giảm

Thành tiền

Thêm vào giỏ hàng

Sửa CT

Xoá CT

Thanh toán

Khi nhấn vào nút Bán hàng sẽ ra được giao diện như hình trên.

Think
Future
THINK THE BETTER WORLD

Bán hàng

Sản phẩm

Khuyến mãi

Nhân viên

Khách hàng

Kho

Thống kê

Bán hàng

Hoà đơn

Mua hàng

Danh sách sản phẩm

ID	Tên sản phẩm	Đơn giá	Số lượng tồn	Loại sản phẩm	Hãng sản phẩm
SP001	Tuf Gaming F...	21,000,000	11	Laptop gaming	Asus
SP002	Laptop Acer S...	22,000,000	5	Ultrabook m...	Acer
SP003	Laptop Dell Pr...	23,000,000	6	Laptop văn p...	Dell
SP004	Tuf Gaming A...	24,000,000	18	Laptop gaming	Asus
SP005	Macbook Pro ...	25,000,000	5	Ultrabook m...	Apple
SP006	Laptop Asus ...	27,899,000	5	Laptop gaming	Asus
SP007	Laptop Asus ...	33,990,000	5	Laptop gaming	Asus
SP008	Laptop ASUS ...	102,900,000	105	Laptop gaming	Asus
SP009	Laptop Acer ...	14,690,000	5	Laptop gaming	Acer
SP010	Laptop Gamin...	19,990,000	5	Laptop gaming	Acer
SP011	Laptop Acer P...	40,990,000	5	Laptop gaming	Acer
SP012	Laptop Asus ...	94,990,000	5	Laptop gaming	Asus

Giỏ hàng

ID	Tên sản phẩm	Đơn giá	Số lượng mua	Tổng tiền	Tiền giảm	Thành tiền
----	--------------	---------	--------------	-----------	-----------	------------

Hãng

Mã sản phẩm

Tên sản phẩm

SL Mua

Giá bán

% Giảm

Thành tiền

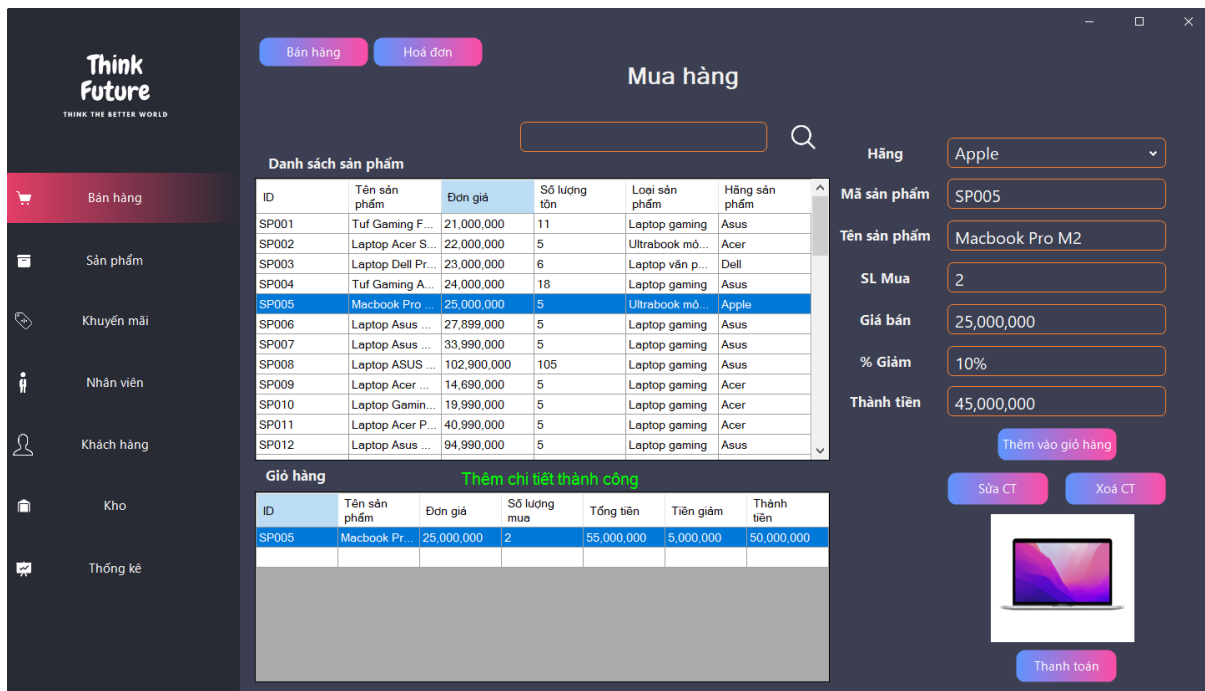
Thêm vào giỏ hàng

Sửa CT

Xoá CT

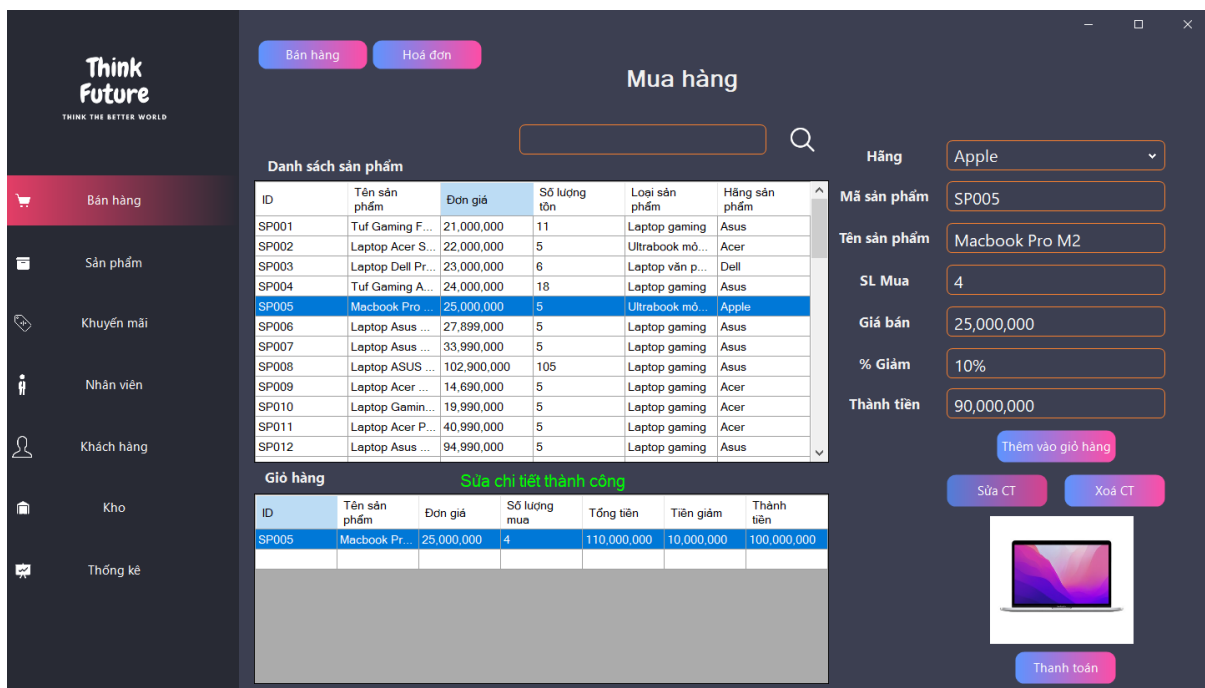
Thanh toán

Tại bảng danh sách sản phẩm người dùng có thể chọn các sản phẩm có sẵn trong bảng và nhập số lượng mua. Khi đó Thành tiền sẽ tự động được tính và người dùng có thể chọn thêm vào giỏ hàng để thêm sản phẩm vào bảng giỏ hàng bên dưới.



Khi thêm vào giỏ hàng thành công sẽ hiện lên Textfield màu xanh lá với nội dung là Thêm chi tiết thành công.

Khi người dùng đổi ý không muốn mua số lượng lúc đầu thì có thể thay đổi số lượng bằng cách chọn sản phẩm trong giỏ hàng và tại SL Mua người dùng nhập vào số lượng mong muốn và chọn sửa CT.



Hiện lên thông báo Sửa chi tiết thành công khi người dùng nhấn nút Sửa CT.

Người dùng cũng có thể nhấn Xóa CT để xóa sản phẩm khi không muốn mua sản phẩm này nữa. Và khi người dùng chọn được sản phẩm mình ưng ý rồi thì có thể nhấn vào nút Thanh toán.

Think
Future
THINK THE BETTER WORLD

Bán hàng

Sản phẩm

Khuyến mãi

Nhân viên

Khách hàng

Kho

Thống kê

Bán hàng

Hoà đơn

Mua hàng

Q

Danh sách sản phẩm

ID	Tên sản phẩm	Đơn giá	Số lượng tồn	Loại sản phẩm	Hãng sản phẩm
SP001	Tuf Gaming F...	21,000,000	11	Laptop gaming	Asus
SP002	Laptop Acer S...	22,000,000	5	Ultrabook mỗ...	Acer
SP003	Laptop Dell Pr...	23,000,000	6	Laptop văn p...	Dell
SP004	Tuf Gaming A...	24,000,000	18	Laptop gaming	Asus
SP005	Macbook Pro ...	25,000,000	5	Ultrabook mỗ...	Apple
SP006	Laptop Asus ...	27,899,000	5	Laptop gaming	Asus
SP007	Laptop Asus ...	33,990,000	5	Laptop gaming	Asus
SP008	Laptop ASUS ...	102,900,000	105	Laptop gaming	Asus
SP009	Laptop Acer ...	14,690,000	5	Laptop gaming	Acer
SP010	Laptop Gamin...	19,990,000	5	Laptop gaming	Acer
SP011	Laptop Acer P...	40,990,000	5	Laptop gaming	Acer
SP012	Laptop Asus ...	94,990,000	5	Laptop gaming	Asus

Giỏ hàng

Xóa chi tiết thành công

ID	Tên sản phẩm	Đơn giá	Số lượng mua	Tổng tiền	Tiền giảm	Thành tiền
----	--------------	---------	--------------	-----------	-----------	------------

Hãng

Apple

Mã sản phẩm

SP005

Tên sản phẩm

Macbook Pro M2

SL Mua

4

Giá bán

25,000,000

% Giảm

10%


Thành tiền

90,000,000

Thêm vào giỏ hàng

Sửa CT

Xóa CT



Thanh toán

Người dùng cũng có thể tìm kiếm sản phẩm bằng cách nhập tên sản phẩm.

Think Future
THINK THE BETTER WORLD

Bán hàng | Hoá đơn

Mua hàng

laptop acer pr

Hãng: Acer

Mã sản phẩm: SP011

Tên sản phẩm: Laptop Acer Predator Helios

SL Mua:

Giá bán: 40,990,000

% Giảm: 20%

Thành tiền: 0

Thêm vào giỏ hàng

Sửa CT | Xóa CT

Thanh toán

ID	Tên sản phẩm	Đơn giá	Số lượng tồn	Loại sản phẩm	Hãng sản phẩm
SP011	Laptop Acer P...	40,990,000	5	Laptop gaming	Acer

ID	Tên sản phẩm	Đơn giá	Số lượng mua	Tổng tiền	Tiền giảm	Thành tiền
----	--------------	---------	--------------	-----------	-----------	------------

Think Future
THINK THE BETTER WORLD

Bán hàng | Hoá đơn

Quản lý hoá đơn

1010

ID	Ngày tạo	Tổng tiền	Tiền đã giảm	Tiền phải trả	Trạng thái	Loại thanh toán	Người thanh toán
1010	2/2/2020	16,990,000	0	16,990,000	Đã thanh toán	Tiền mặt	Nguyễn Đức A...

Thanh toán

Xuất hoá đơn

Tên sản phẩm	Đơn giá	Số lượng mua	Tổng tiền	Tiền đã giảm	Thành tiền
--------------	---------	--------------	-----------	--------------	------------

Nhân viên cũng có thể tìm kiếm hoá đơn bằng cách nhập mã hoá đơn.

5.3. Chức năng thanh toán

Think
Future
THINK THE BETTER WORLD

Bán hàng

Sản phẩm

Khuyến mãi

Nhân viên

Khách hàng

Kho

Thống kê

Bán hàngHoá đơn

Chọn khách hàng cần thanh toán

Thông tin khách hàng

Mã KH

KH06

Tên KH

Phạm Phú Khánh

SĐT

11 Kinh Dương Vương, Quận Bình Tân

Địa chỉ

ppk123@gmail.com

Email

0963521875

Chọn loại thanh toán

Trực tiếp

Thanh toán

Chọn lại

Bảng khách hàng

Mã khách	Tên khách	Địa chỉ	Số điện	Email
KH01	Nguyễn H...	123 Phan ...	09635214...	trong@g...
KH02	Nguyễn H...	11 Nguyễn...	09696814...	huutoan...
KH03	Nguyễn V...	85 Nguyễn...	03596428...	Truong12...
KH04	Nguyễn H...	894 3 thá...	09698214...	hoa123@...
KH05	Nguyễn K...	123 Trươn...	03995298...	nguyenkh...
KH06	Phạm Phú...	11 Kinh D...	09635218...	ppk123@...
KH07	Lưu Gia Tĩ...	198 Điện ...	09635218...	lunh123@...
KH08	Nguyễn C...	55 Lê Lai, ...	09635219...	nct123@g...
KH09	Hà Thanh ...	63 Nguyễ...	09635219...	dien121@...

Tìm khách hàng

Sản phẩm đã mua

Tên sản phẩm	Đơn giá	Số lượng mua	Tổng tiền	Tiền đã giảm	Thành tiền
Tuf Gaming A15 2...	24,000,000	2	48000000	4,800,000	43,200,000

Loại thanh toán: Trực tiếp

43.200.000

Khi người dùng chọn thanh toán ở giao diện bán hàng hoặc các hoá đơn chưa thanh toán ở giao diện hoá đơn. Ở đây nếu người dùng là khách hàng chưa có trong dữ liệu của cửa hàng thì có thể nhấn vào nút Khách hàng để tạo khách hàng mới.

Think
Future
THINK THE BETTER WORLD

Bán hàng

Sản phẩm

Khuyến mãi

Nhân viên

Khách hàng

Kho

Thống kê

Quản lý khách hàng

Thông tin khách hàng

Mã KH

KH92

Tên KH

Hồ Sĩ Minh

SĐT

0969381267

Địa chỉ

105 An Dương Vương, Q5

Email

hsmea123@gmail.com

Thêm

Sửa

Xoá

Làm mới

BXH khách hàng

Tên khách hàng	Tổng hoá đơn	Doanh số
Nguyễn Thanh Ngân	8	502,220,000
Nguyễn Công Bách	5	449,920,000
Trần Minh Thuận	4	432,946,000
Phùng Văn Ý	5	340,230,000
Nguyễn Thị Kim Ngân	4	324,668,000
Trần Minh Trí	2	306,962,000
Trần Đình Kiệt	2	256,970,000
Nguyễn Hoàng Khang	3	222,728,000
Cao Phương Trang	4	219,560,000
Nguyễn Văn Ba	2	212,980,000

Từ khóa

Danh sách khách hàng

Đã thêm khách hàng

Mã khách hàng	Tên khách hàng	Địa chỉ	Số điện thoại	Email
KH01	Nguyễn Hữu Trọng	123 Phan Văn Trị Q5	0963521458	trong@gmail.com
KH02	Nguyễn Hữu Toàn	11 Nguyễn Trãi Q5	0969681487	huutoan@gmail.com
KH03	Nguyễn Văn Trường	85 Nguyễn Khuyến Quận 1	0359642876	Truong123@gmail.com
KH04	Nguyễn Hoà	894 3 tháng 2 Quận 3	0969821447	hoa123@gmail.com
KH05	Nguyễn Kha	123 Trường Định Quận 1	0399529874	nguyenkha@gmail.com
KH06	Phạm Phú Khánh	11 Kinh Dương Vương, Quận ...	0963521875	ppk123@gmail.com
KH07	Lưu Gia Tĩnh	198 Điện Biên Phủ, Quận 3	0963521877	lunh123@gmail.com
KH08	Nguyễn Công Trình	55 Lê Lai, Quận 1	0963521912	nct123@gmail.com
KH09	Hà Thanh Diễn	63 Nguyễn Khuyến, Quận 3	0963521985	dien121@gmail.com
KH10	Nguyễn Thị Kim Ngân	64 Nguyễn Trãi, Quận 1	0963521231	kimngan1129@gmail.com
KH11	Phùng Thiên Trang	625 Điện Biên Phủ, Quận 3	0353521981	trang123@gmail.com

Sau đó nhân viên sẽ quay lại nút Bán hàng chọn phần Hoá đơn và trong bảng danh sách hoá đơn chọn đến hoá đơn vừa rồi có trạng thái chưa thanh toán và chọn nút Thanh toán.

Trong bảng khách hàng chọn khách hàng vừa tạo khi đó bên thông tin sẽ xuất hiện lên thông tin của khách hàng. Lúc này chọn loại thanh toán và chọn nút Thanh toán để thanh toán.

Think
Future

THINK THE BETTER WORLD

Bán hàng

Sản phẩm

Khuyến mãi

Nhân viên

Khách hàng

Kho

Thống kê

Bán hàng

Hoá đơn

Chọn khách hàng cần thanh toán

Thông tin khách hàng

Mã KH: KH92

Tên KH: Hồ Sĩ Minh

SĐT: 105 An Dương Vương, Q5

Địa chỉ: hsmea123@gmail.com

Email: 0969381267

Chọn loại thanh toán: Trực tiếp

Thanh toán Chọn lại

Bảng khách hàng

Mã khách	Tên khách	Địa chỉ	Số điện	Email
KH85	Huỳnh Thị...	92 Võ Thị...	09386286...	diemquyn...
KH86	Nguyễn P...	109 Bạch ...	09388883...	phibang3...
KH87	Nguyễn Ý...	120 Điện ...	09397450...	ynhi23@g...
KH88	Nguyễn A...	300 Ngõ ...	09686509...	thu23@g...
KH89	Nguyễn V...	314 Huyện...	09679527...	vanson12...
KH90	Nguyễn X...	231 Điện ...	09638216...	xuantruon...
KH91	Nguyễn Q...	124 Trườn...	09635212...	quangtru...
KH92	Hồ Sĩ Minh	105 An D...	09693812...	hsmea123...

Tim khách hàng

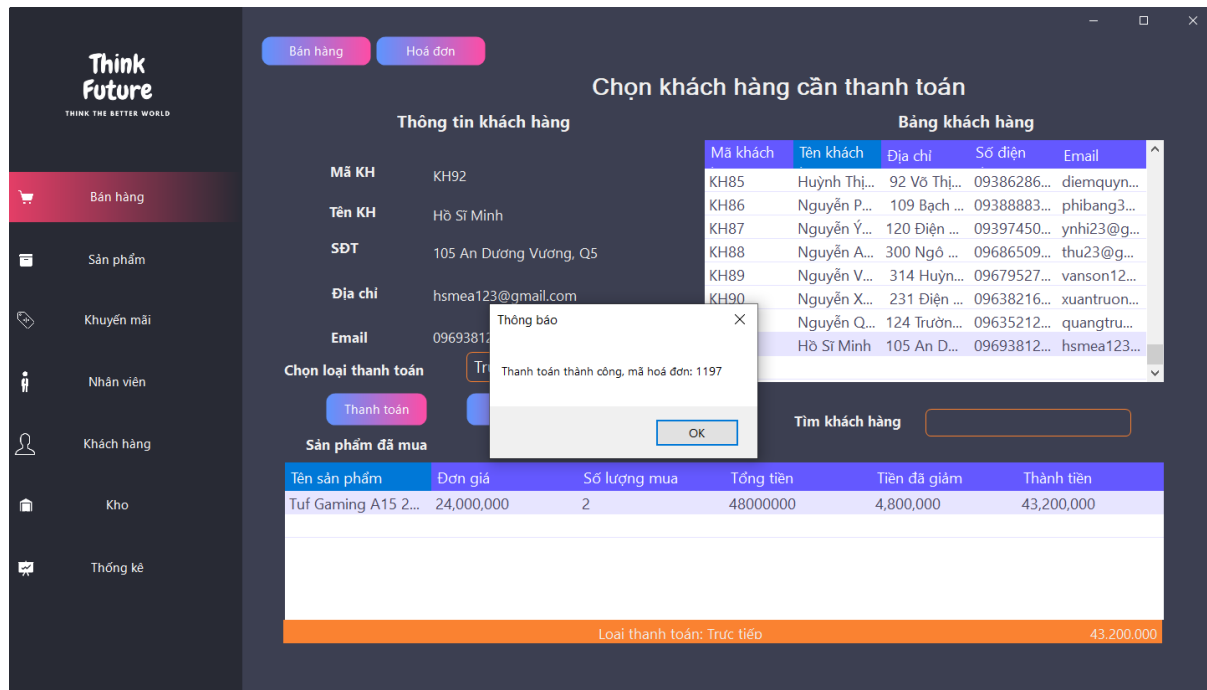
Sản phẩm đã mua

Tên sản phẩm	Đơn giá	Số lượng mua	Tổng tiền	Tiền đã giảm	Thành tiền
Tuf Gaming A15 2...	24,000,000	2	48000000	4,800,000	43,200,000

Loại thanh toán: Trực tiếp

43.200.000

Thanh toán thành công thì sẽ hiện lên dialog kèm mã hoá đơn và sẽ tự in ra một file PDF hoá đơn.



Thông tin hoá đơn 1197

Tên sản phẩm	Số lượng mua	Đơn giá	Tiền đã giảm	Thành tiền
Tuf Gaming A...	2	24,000,000	4,800,000	43,200,000

Tổng thanh toán 43,200,000

Ngày thanh toán 11/11/2023

Nhân viên thực hiện Nguyễn Hữu Trọng

5.4. Chức năng quản lý sản phẩm

Chức năng này sẽ giúp quản lý các thông tin của sản phẩm bằng các thao tác thêm, sửa, xoá, làm mới.

Form sản phẩm:

Think
Future
THINK THE BETTER WORLD

Bán hàng

Sản phẩm

Khuyến mãi

Nhân viên

Khách hàng

Kho

Thống kê

Sản phẩm

Loại sản phẩm

Hãng sản phẩm

Quản lý sản phẩm

Thông tin sản phẩm

Mã sản phẩm

SP003

Tên sản phẩm

Laptop Dell Precision 5510

Loại

Laptop văn phòng

Số lượng

3

Giá bán

23,000,000

Hãng

Dell

Thêm


Sửa

Xoá

Làm mới

Từ khoá

Chọn ảnh



Danh sách sản phẩm

ID	Tên sản phẩm	Đơn giá	Số lượng	Loại sản phẩm	Hãng sản phẩm
SP001	Tuf Gaming F15 2022	21,000,000	5	Laptop gaming	Asus
SP002	Laptop Acer Swift 3	22,000,000	5	Ultrabook mỏng nhẹ	Acer
SP003	Laptop Dell Precision 5510	23,000,000	3	Laptop văn phòng	Dell
SP004	Tuf Gaming A15 2022	24,000,000	7	Laptop gaming	Asus
SP005	Macbook Pro M2	25,000,000	5	Ultrabook mỏng nhẹ	Apple
SP006	Laptop Asus TUF Gaming ...	27,899,000	5	Laptop gaming	Asus
SP007	Laptop Asus ROG Flow X13	33,990,000	5	Laptop gaming	Asus
SP008	Laptop ASUS ROG Strix S...	102,990,000	42	Laptop gaming	Asus
SP009	Laptop Acer Gaming Aspir...	14,690,000	5	Laptop gaming	Acer
SP010	Laptop Gaming Acer Nitro ...	19,990,000	5	Laptop gaming	Acer
SP011	Laptop Acer Predator Heli...	40,990,000	5	Laptop gaming	Acer

Form loại sản phẩm:

Think
Future
THINK THE BETTER WORLD

Bán hàng

Sản phẩm

Khuyến mãi

Nhân viên

Khách hàng

Kho

Thống kê

Sản phẩm

Loại sản phẩm

Hãng sản phẩm

Quản lý loại sản phẩm

Tìm kiếm loại Laptop

Thông tin loại

Mã loại

5002

Tên loại

Ultrabook mỏng nhẹ

Thêm

Sửa

Xoá

Làm mới

Bảng loại sản phẩm

Mã loại	Tên loại
5001	Laptop gaming
5002	Ultrabook mỏng nhẹ
5003	Laptop văn phòng

Danh sách sản phẩm

ID	Tên sản phẩm	Đơn giá	Số lượng	Hãng sản phẩm
SP002	Laptop Acer Swift 3	22,000,000	5	Acer
SP005	Macbook Pro M2	25,000,000	5	Apple
SP021	Laptop Asus ZenBook 14 UX425...	18,990,000	5	Asus
SP022	Laptop Asus ExpertBook B9400...	29,990,000	5	Asus
SP023	Laptop Asus Zenbook 14 Flip O...	31,690,000	5	Asus
SP024	Laptop Acer Aspire 3 A315-56-5...	10,999,000	5	Acer
SP025	Laptop Acer Aspire 5 A514-55-5...	16,990,000	5	Acer
SP026	Laptop Acer Swift 3 SF314-512-...	22,990,000	5	Acer
SP027	Laptop Apple MacBook Pro M2 2...	29,990,000	5	Apple
SP028	Laptop Apple MacBook Air 13.6 i...	29,550,000	5	Apple
SP029	Laptop Apple MacBook Air 13.6 i...	29,550,000	5	Apple
SP030	Laptop Dell Vostro 14 3400 YX5...	16,790,000	5	Dell
SP031	Laptop Dell Inspiron 16 5608 51...	28,790,000	5	Dell

Form hãng sản phẩm:

Think
Future
THINK THE BETTER WORLD

Bán hàng

Sản phẩm

Khuyến mãi

Nhân viên

Khách hàng

Kho

Thống kê

Sản phẩmLoại sản phẩmHãng sản phẩm

Quản lý hãng sản phẩm

Tìm kiếm hãng

Thông tin hãng

Mã hãng4001

Tên hãngAsus

ThêmSửaLàm mới

Bảng hãng sản phẩm

Mã hãng	Tên hãng
4001	Asus
4002	Acer
4003	Apple
4004	Dell
4005	Lenovo

Danh sách sản phẩm

ID	Tên sản phẩm	Đơn giá	Số lượng	Loại sản phẩm
SP001	Tuf Gaming F15 2022	21,000,000	5	Laptop gaming
SP004	Tuf Gaming A15 2022	24,000,000	7	Laptop gaming
SP006	Laptop Asus TUF Gaming F15	27,899,000	5	Laptop gaming
SP007	Laptop Asus ROG Flow X13	33,990,000	5	Laptop gaming
SP008	Laptop ASUS ROG Strix SCAR 1...	102,900,000	42	Laptop gaming
SP012	Laptop Asus ROG Zephyrus M16...	94,990,000	5	Laptop gaming
SP013	Laptop ASUS ROG Zephyrus G1...	40,690,000	5	Laptop gaming
SP014	Laptop Asus ROG Strix G15 G51...	25,990,000	5	Laptop gaming
SP021	Laptop Asus ZenBook 14 UX425...	18,990,000	5	Ultrabook mỏng nhẹ
SP022	Laptop Asus ExpertBook B9400C...	29,990,000	5	Ultrabook mỏng nhẹ
SP023	Laptop Asus Zenbook 14 Flip OL...	31,690,000	5	Ultrabook mỏng nhẹ
SP036	Laptop Asus ZenBook 14 UX425...	18,999,000	5	Laptop văn phòng
SP037	Laptop Asus ExpertBook B9400C...	29,990,000	5	Laptop văn phòng

5.5. Chức năng quản lý khuyến mãi

Think Future
THINK THE BETTER WORLD

Chương trình khuyến mãi

Danh sách khuyến mãi

Mã khuyến mãi	Tên khuyến mãi
2001	Chúc mừng năm mới 2020
2002	Mừng lễ 30/4
2003	Back to School
2004	Merry Christmas
2005	Chúc mừng năm mới 2022
2006	Flash sale 2/2
2007	Flash sale 4/4
2008	Flash sale 6/6
2009	Flash sale 8/8
2010	Back to School
2011	Flash sale 11/11
2012	Merry Christmas

Chi tiết khuyến mãi

Tên sản phẩm	%Giảm giá	Trạng thái
Tuf Gaming F15 2022	10	Sắp khuyến mãi
Laptop Acer Swift 3	10	Sắp khuyến mãi
Laptop Dell Precision 5510	10	Sắp khuyến mãi
Tuf Gaming A15 2022	10	Sắp khuyến mãi
Macbook Pro M2	10	Sắp khuyến mãi
Laptop Asus TUF Gaming F15	10	Sắp khuyến mãi
Laptop Asus ROG Flow X13	10	Sắp khuyến mãi
Laptop ASUS ROG Strix SCAR 17 SE G733CX-LL678...	10	Sắp khuyến mãi
Laptop Acer Gaming Aspire 7 A715-42G-R4XX NH.Q...	10	Sắp khuyến mãi
Laptop Gaming Acer Nitro 5 Eagle AN515-57-54MV N...	10	Sắp khuyến mãi
Laptop Acer Predator Helios 300 PH315-55-78KG NH...	10	Sắp khuyến mãi

Trang điều khiển: Bán hàng, Sản phẩm, **Khuyến mãi**, Nhân viên, Khách hàng, Kho, Thống kê.

Nút hành động: Tạo CTKM, Xuất CTKM.

Ở đây người dùng có thể xem danh sách khuyến mãi và khi chọn vào một khuyến mãi nào đó thì ở bảng chi tiết khuyến mãi sẽ xuất hiện ra những sản phẩm áp dụng khuyến mãi đó. Nếu người dùng muốn tạo thêm khuyến mãi thì nhấn vào nút Tạo CTKM.

Think Future
THINK THE BETTER WORLD

Chương trình khuyến mãi

Tên chương trình: Siêu Sale 11/11

SẢN PHẨM

ID	Tên sản phẩm	Đơn giá	Loại sản phẩm	Hãng sản phẩm
SP001	Tuf Gaming F15 ...	21,000,000	Laptop gaming	Asus
SP002	Laptop Acer Swif...	22,000,000	Ultrabook mỏng ...	Acer
SP003	Laptop Dell Prec...	23,000,000	Laptop văn phòng	Dell
SP004	Tuf Gaming A15...	24,000,000	Laptop gaming	Asus
SP005	Macbook Pro M2	25,000,000	Ultrabook mỏng ...	Apple
SP006	Laptop Asus TU...	27,899,000	Laptop gaming	Asus
SP007	Laptop Asus RO...	33,990,000	Laptop gaming	Asus
SP008	Laptop ASUS R...	102,900,000	Laptop gaming	Asus
SP009	Laptop Acer Ga...	14,690,000	Laptop gaming	Acer
SP010	Laptop Gaming ...	19,990,000	Laptop gaming	Acer
SP011	Laptop Acer Pre...	40,990,000	Laptop gaming	Acer

Chi tiết CTKM

Mã sản phẩm	Tên sản phẩm	%Giảm
SP007	Laptop Asus ROG Flow X13	10
SP005	Macbook Pro M2	3
SP009	Laptop Acer Gaming Aspire 7 ...	7

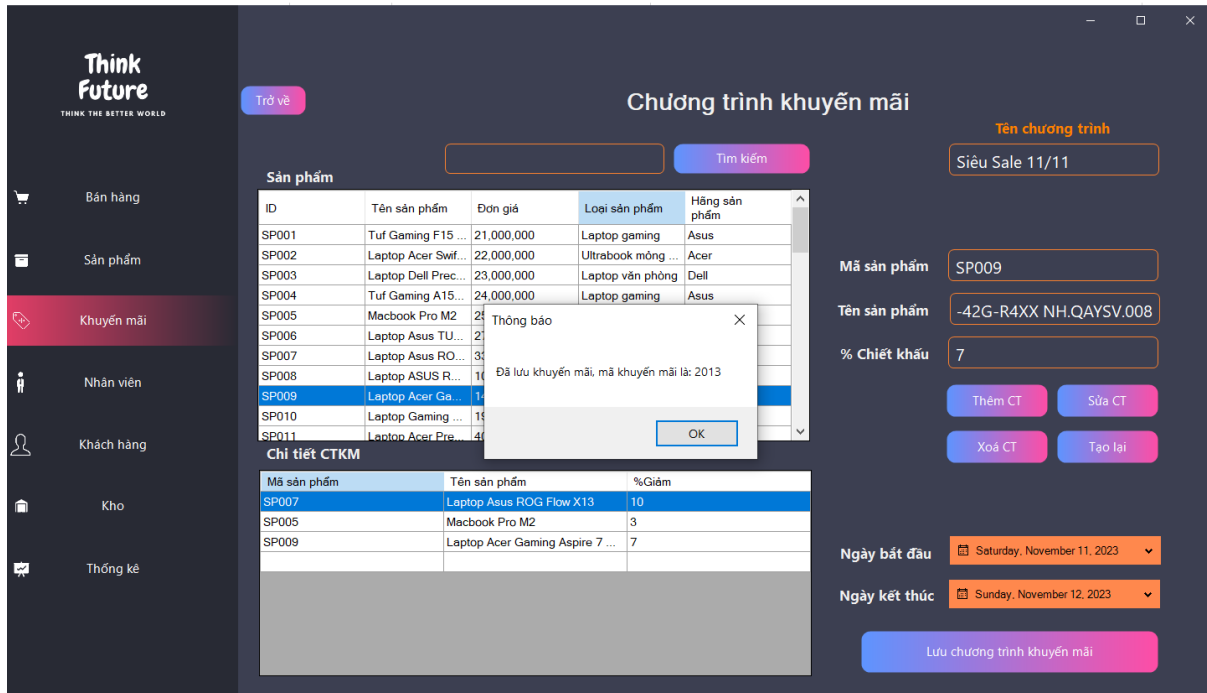
Trang điều khiển: Bán hàng, Sản phẩm, **Khuyến mãi**, Nhân viên, Khách hàng, Kho, Thống kê.

Nút hành động: Thêm CT, Sửa CT, Xóa CT, Tạo lại.

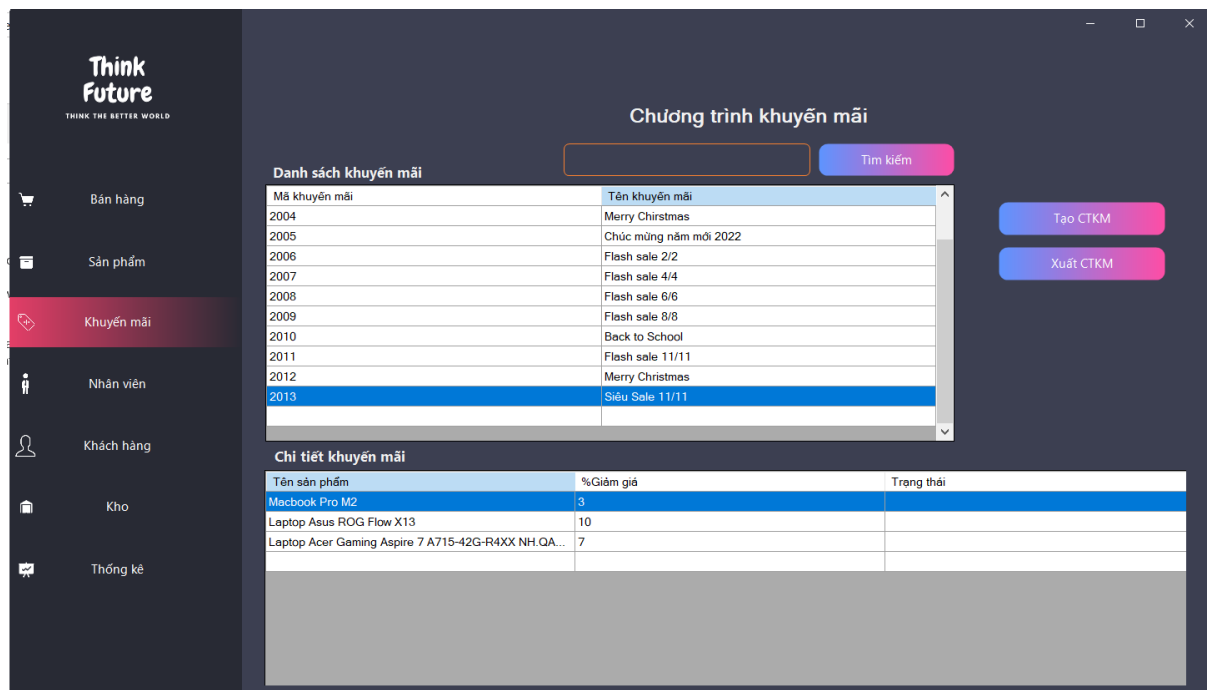
Ngày bắt đầu: Saturday, November 11, 2023
Ngày kết thúc: Sunday, November 12, 2023

Nút hành động: Lưu chương trình khuyến mãi.

Sau khi đã nhập các thông tin cần thiết người dùng nhấn thêm CT và nhấn lưu chương trình khuyến mãi.



Xuất hiện dialog kèm mã khuyến mãi thông báo tạo chương trình khuyến mãi thành công.



5.6. Chức năng quản lý nhân viên

Tên nhân viên	Tổng hoá đơn	Doanh thu
Nguyễn Hữu Trọng	8	6,941,700,000
Thanh Trúc	67	3,213,550,000
Nguyễn Hoàng Quân	42	2,841,371,000
Nguyễn Tiến	42	2,430,568,000
Nguyễn Đức Anh	42	2,037,332,000

Mã nhân viên	Tên nhân viên	Chức vụ	Sinh nhật	Số điện thoại	Lương
NV01	Nguyễn Tiến	Quản lý	6/6/2002 12:00:00 AM	0456788945	20,000,000
NV02	Nguyễn Đức Anh	Quản lý	7/7/2002 12:00:00 AM	0489784545	20,000,000
NV03	Nguyễn Hoàng Quân	Nhân viên	8/8/2002 12:00:00 AM	0489748546	15,000,000
NV04	Thanh Trúc	Nhân viên	9/9/2002 12:00:00 AM	0887484645	20,000,000

Đây là form quản lý nhân viên với các chức năng thêm, sửa, xoá, làm mới. Khi nhấn một lần vào thông tin nhân viên ở danh sách nhân viên thì thông tin của nhân viên đó sẽ được hiện lên trên các textfield để dễ dàng theo dõi, cũng như sửa đổi thông tin.

Bên phải là BXH nhân viên để có thể theo dõi tình hình bán hàng của nhân viên từ đó có thể xem xét tăng lương hoặc thăng chức...

ID	Ngày tạo	Tổng tiền	Tiền đã giảm	Tiền phải trả	Trạng thái	Loại thanh toán
1004	14/1/2020	125,000,000	0	125,000,000	Đã thanh...	Chuyển...
1008	27/1/2020	125,000,000	0	125,000,000	Đã thanh...	Chuyển...
1012	14/2/2020	16,790,000	0	16,790,000	Đã thanh...	Chuyển...
1016	27/2/2020	17,690,000	0	17,690,000	Đã thanh...	Chuyển...
1020	14/3/2020	101,970,000	0	101,970,000	Đã thanh...	Chuyển...
1024	27/4/2020	32,290,000	0	32,290,000	Đã thanh...	Chuyển...
1028	14/5/2020	91,370,000	0	91,370,000	Đã thanh...	Chuyển...
1032	27/6/2020	31,690,000	0	31,690,000	Đã thanh...	Chuyển...
1033	27/6/2020	32,997,000	0	32,997,000	Đã thanh...	Chuyển...
1034	27/6/2020	16,000,000	0	16,000,000	Đã thanh...	Chuyển...

Tên sản phẩm	Đơn giá	Số lượng mua	Tổng tiền	Tiền đã giảm	Thành tiền
Laptop Dell Vostro 14 34...	16,790,000	1	16,790,000	0	16,790,000

Khi nhấp đúp vào một nhân viên trong bảng danh sách nhân viên sẽ chuyển qua form chi tiết nhân viên và hiện ra thông tin các hoá đơn mà nhân viên đó đã thực hiện.

Think Future
THINK THE BETTER WORLD

Nhân viên | Phân quyền

Phân quyền

Nhập tên NV để tìm kiếm

Mã nhân viên	Tên nhân viên	Chức vụ
NV01	Nguyễn Tiến	Quản lý
NV02	Nguyễn Đức Anh	Quản lý
NV03	Nguyễn Hoàng Quân	Nhân viên
NV04	Thanh Trúc	Nhân viên

Vai trò: ☒ Nhân viên ☐ Quản lý

Ngoài form Nhân viên còn có form Phân quyền để thay đổi quyền của nhân viên.

Think Future
THINK THE BETTER WORLD

Nhân viên | Phân quyền

Phân quyền

Nhập tên NV để tìm kiếm

Mã nhân viên	Tên nhân viên	Chức vụ
NV01	Nguyễn Tiến	Quản lý
NV02	Nguyễn Đức Anh	Quản lý
NV03	Nguyễn Hoàng Quân	Nhân viên
NV04	Thanh Trúc	Nhân viên

Thông báo

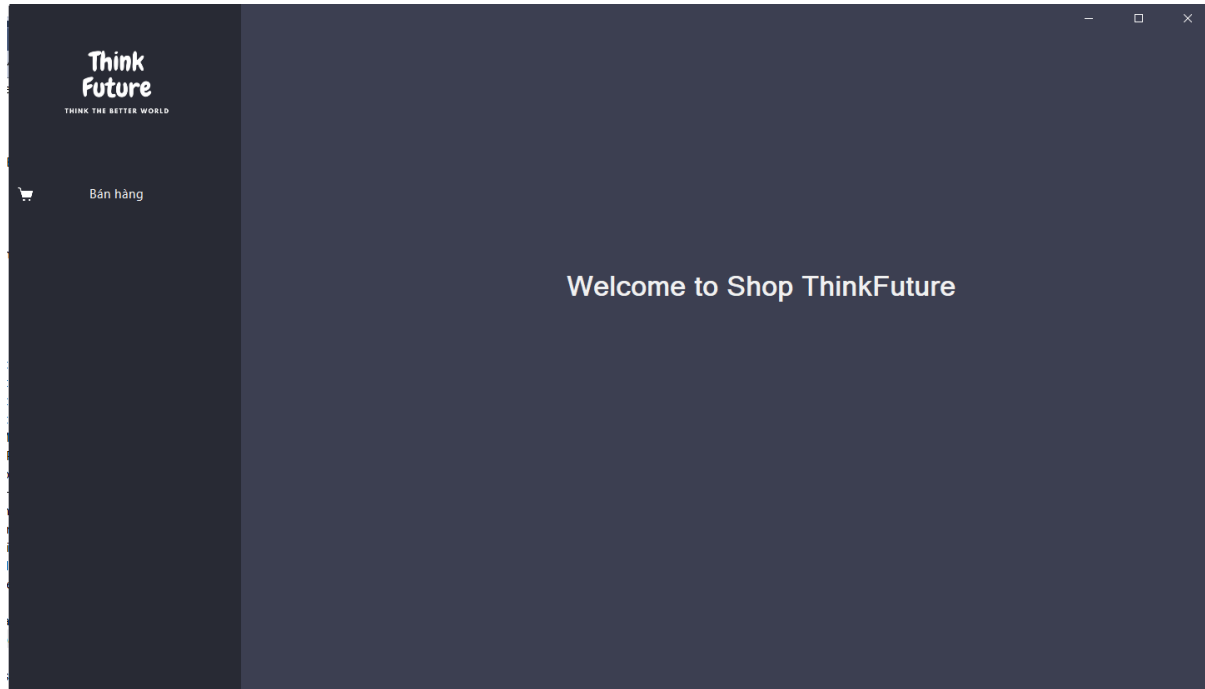
Đã sửa đổi quyền

OK

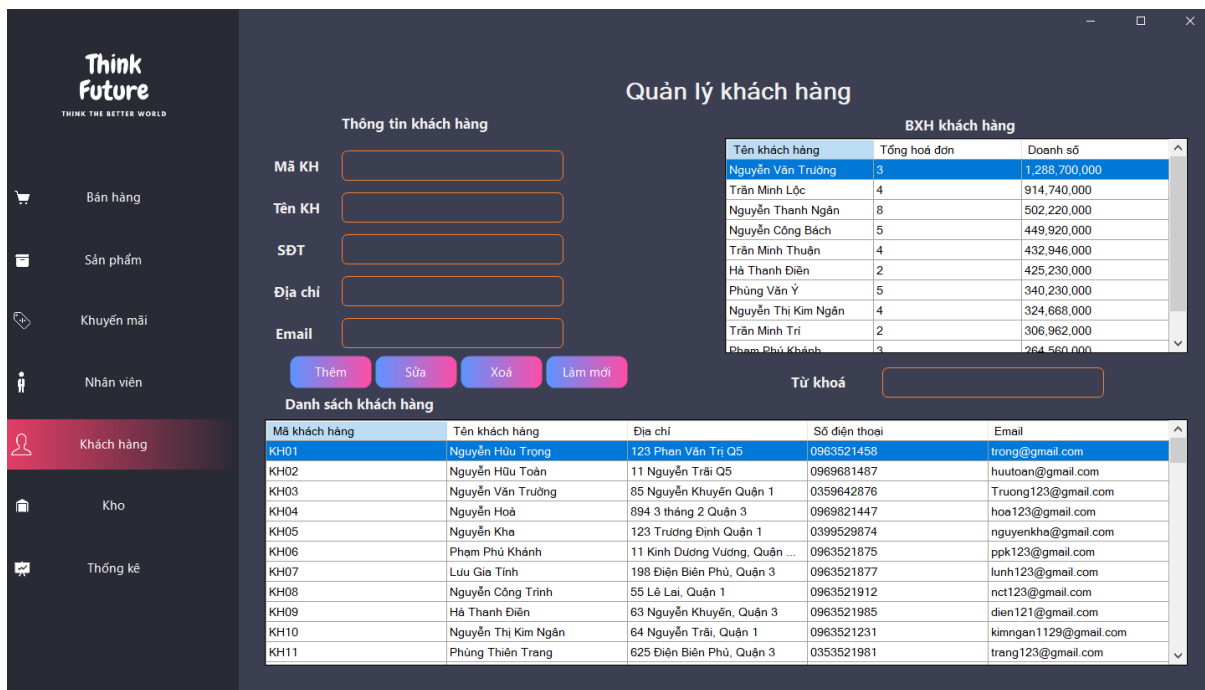
Vai trò: ☐ Nhân viên ☒ Quản lý

Ví dụ muốn thay đổi Thanh Trúc từ nhân viên thành quản lý thì ta chọn vào radio button Quản lý và nhấn nút lưu.

Ở trong chương trình tài khoản có chức vụ Quản lý thì sẽ được phép truy cập tất cả chức năng của chương trình còn nhân viên thì chỉ có thể dùng được chức năng bán hàng.



5.7. Chức năng quản lý khách hàng



Cũng tương tự như form Nhân viên người dùng có thể thêm, sửa, xoá, làm mới trong form Khách hàng. Khi nhấn một lần vào thông tin khách hàng ở danh sách khách hàng thì thông tin của khách hàng đó sẽ được hiện lên trên các textfield để dễ dàng theo dõi cũng như sửa đổi thông tin.

Think Future
THINK THE BETTER WORLD

[Trở về](#)

Thông tin khách hàng

Mã KH: KH10
Tên KH: Nguyễn Thị Kim Ngân
SĐT: 0963521231
Địa chỉ: 64 Nguyễn Trãi, Quận 1
Email: kimngan1129@gmail.com

Hoà đơn đã mua

ID	Ngày tạo	Tổng tiền	Tiền đã giảm	Tiền phải trả	Trạng thái	Loại thanh toán
1011	4/2/2020	10,990.0...	0	10,990.0...	Đã thanh...	Chuyển
1119	20/8/2021	94,950.0...	0	94,950.0...	Đã thanh...	Tiền mặt
1153	1/3/2022	123,778,...	0	123,778,...	Đã thanh...	Tiền mặt
1183	1/9/2022	94,950.0...	0	94,950.0...	Đã thanh...	Tiền mặt

Chi tiết hoá đơn

Tên sản phẩm	Đơn giá	Số lượng mua	Tổng tiền	Tiền đã giảm	Thành tiền

Từ khoá: [Tìm](#)

Khi người dùng nhấp đúp vào khách hàng bất kì trong bảng danh sách khách hàng thì sẽ hiển thị ra thông tin của khách hàng đó gồm hoá đơn khách đã mua và chi tiết hoá đơn.

5.8. Chức năng quản lý phiếu nhập kho

Form này cung cấp cho người dùng thông tin các phiếu nhập kho cũng như chi tiết của các phiếu nhập kho.

Khi người dùng muốn nhập một sản phẩm mới thì nhấn vào nút tạo PNK.

Think Future

THINK THE BETTER WORLD

Bán hàng

Sản phẩm

Khuyến mãi

Nhân viên

Khách hàng

Kho

Thống kê

Quản lý phiếu nhập kho

Tìm kiếm

Tạo PNK

Xuất PNK

Thông tin phiếu nhập kho

Mã nhập kho	Thời gian nhập	Tổng tiền	Nhà cung cấp
1	11/30/2022 12:00:00 AM	0	Lenovo
2	11/30/2022 12:00:00 AM	312,000,000	Lenovo
3	11/30/2022 12:00:00 AM	10,290,000,000	Lenovo

Chi tiết phiếu nhập kho

Tên sản phẩm	Số lượng nhập	Đơn giá	Tổng tiền
--------------	---------------	---------	-----------

Khi nhân tạo phiếu nhập kho thì hệ thống hiển thị form tạo phiếu nhập kho, người dùng có thể lựa chọn các sản phẩm cần nhập và chọn thanh toán thì phiếu nhập kho sẽ được lưu lại.

Think Future

THINK THE BETTER WORLD

Bán hàng

Sản phẩm

Khuyến mãi

Nhân viên

Khách hàng

Kho

Thống kê

Phiếu nhập kho

Tìm kiếm

Nhà cung cấp

Mã sản phẩm

Tên sản phẩm

Số lượng nhập

Số lượng có sẵn

Đơn giá

Thành tiền

Sản phẩm

ID	Tên sản phẩm	Đơn giá	Loại sản phẩm	Hãng sản phẩm
SP001	Tuf Gaming ...	21,000,000	Laptop gami...	Asus
SP002	Laptop Acer ...	22,000,000	Ultrabook m...	Acer
SP003	Laptop Dell ...	23,000,000	Laptop văn p...	Dell
SP004	Tuf Gaming ...	24,000,000	Laptop gami...	Asus
SP005	Macbook Pr...	25,000,000	Ultrabook m...	Apple
SP006	Laptop Asus ...	27,899,000	Laptop gami...	Asus
SP007	Laptop Asus ...	33,990,000	Laptop gami...	Asus

Chi tiết nhập

Thêm CT

Sửa CT

Xoá CT

Tạo lại

Mã sản phẩm	Tên sản phẩm	Số lượng nhập	Đơn giá	Tổng tiền
-------------	--------------	---------------	---------	-----------

Thanh toán

5.9. Chức năng thống kê

Chọn vào thống kê hệ thống sẽ hiển thị form thống kê như hình



Khi nhấn vào nút thống kê sẽ hiển thị thống kê tổng quát nhất về những thông tin liên quan và bảng doanh thu theo quý của từng năm.



Ta có thể nhấn vào nút chuyển sang màn hình thống kê chi tiết với biểu đồ và bảng thống kê sản phẩm bán chạy.



Link Github đồ án: <https://github.com/ThanhTrucOwO/QuanLyBanLaptop>

TỔNG KẾT

Đề tài “Quản lý cửa hàng bán Laptop” xuất phát từ nhu cầu thực tiễn nhằm đáp ứng nhu cầu của người dùng. Vận dụng các kiến thức đã học cùng với việc tra cứu tài liệu chuyên ngành nhưng do kỹ năng, kinh nghiệm, hạn chế về mặt thời gian nên sai sót là việc không thể tránh khỏi nên đề tài này em đã hoàn thành ở mức độ như sau:

Các kết quả đã thực hiện:

- Giao diện phần mềm đẹp, thu hút khách hàng và người sử dụng.
- Tiết kiệm thời gian, chi phí
- Làm quen với môi trường và áp lực công việc
- Phân tích thiết kế có tính tương đối

Hạn chế

- Chức năng còn thiếu sót nhiều
- Phần mềm chưa có tính bảo mật cao
- Các thao tác chưa được tối ưu

Hướng phát triển

- Bảo trì và nâng cấp phần mềm
- Chú trọng, tiếp thu phản hồi của khách hàng

Phụ lục: Code xử lý

Chức năng đăng nhập:

```
public string login(string userName, string pass) {
    if (userName == "")
        return "requeid_user";
    if (pass == "")
        return "requeid_pass";
    TaiKhoanDAO us = new TaiKhoanDAO();
    UserDTO userDTO = new UserDTO(userName, pass);
    UserDTO user = us.searchUser(userDTO);

    if (user != null) {
        currentAcc = user;
        return "success";
    } else {
        return "notFound";
    }
}

public void addUser(string user, string pass, string service) {

    UserDTO userDTO = new UserDTO(user, pass);
    userDTO.Id = user;
    if (service == "Quản lý")
        userDTO.IdQuyen = 7002;
    else
        userDTO.IdQuyen = 7003;
    new TaiKhoanDAO().addUser(userDTO);
}

public static List < UserDTO > listUser() {

    return new TaiKhoanDAO().listUser();
}
```

Chức năng bán hàng

```
public string saveOrder()
{
    NhanVienDao dao = new NhanVienDao();
    if(idBill == -1)
    {
        return "error_notFoundBill";
    }
    NhanVien nv = dao.searchEmployeeIdUser(currentAcc.Id);
    this.hoaDon.IdEmployee = nv.Id;

    dao.updateTotalSalesEmployee(this.hoaDon.IdEmployee,
this.hoaDon.SumPay);
    this.hoaDonDAO.addHoaDon(this.hoaDon);
    foreach (ChiTietHoaDonDTO value in this.listChiTietHoaDonDTO)
    {
        chiTietHoaDonDAO.addChiTietHoaDon(value);
        deleteQuantity(value.QuantityProduct, value.IdProduct);
    }
    restartBill();

    return null;
}

public string updateDetailsProduct(string quantity, string idProduct)
{
    if (idProduct == null)
        return "error_idNull";
    string result = "";
    if (!isNumber(quantity))
        result += "error_quantity";
    if (!checkDuplicateDetails(idProduct))
        result += "error_duplicate";
    result += checkQuantityProduct(quantity, idProduct);
    if (result.Contains("error"))
    {
        return result;
    }
    else
    {
        SanPham p = sanPhamDAO.searchId(idProduct);

        float intoMoney = this.intoMoney(p.UnitPrice, quantity);
        float sumMoney = this.sumMoney(p.IdProduct, quantity,
p.UnitPrice);
        updateDetailsProduct(int.Parse(quantity), idProduct,
this.idBill, intoMoney, intoMoney- sumMoney, p.UnitPrice);
        result += "success";
    }
    return result;
}
```

```

public string deleteDetailsProduct(string idProduct)
{
    ChiTietHoaDonDTO ct = null;
    if (idProduct == null)
        return "error_idNull";
    foreach (ChiTietHoaDonDTO value in this.listChiTietHoaDonDTO)
    {
        if (value.IdProduct == idProduct && value.IdBill == idBill)
        {
            this.deleteDetailsToBill(value);
            ct = value;
        }
    }
    if(ct != null)
    {
        this.listChiTietHoaDonDTO.Remove(ct);
    }
    return "success";
}

public bool checkDuplicateDetails(string id)
{
    List<ChiTietHoaDonDTO> list = this.listChiTietHoaDonDTO;
    foreach (ChiTietHoaDonDTO detailsBill in list)
    {
        if(detailsBill.IdBill == this.idBill && detailsBill.IdProduct
== id)
            return true;
    }
    return false;
}

public string checkQuantityProduct(string qtt, string idProduct)
{
    string result = "";
    try
    {
        int quantity = int.Parse(qtt);

        if (quantity <= 0)
            result = "error_quantity";
        List<SanPham> listSp = sanPhamDAO.listProduct();
        foreach (SanPham value in listSp)
        {
            if(value.IdProduct == idProduct)
            {
                if(value.QuantityProduct < quantity)
                {
                    result = "error_quantity";
                }
            }
        }
    }
}

```

```

        }
    }
    catch (Exception ex)
    {
        return "error_quantity";
    }
    return result;
}

public List<SanPham> searchP(string s)
{
    return searchProduct(s, filterProductOtherZero());
}

public List<SanPham> filterProduct(string nameManufacture, string nameType)
{
    if(nameManufacture == "Tất cả" && nameType == "Tất cả"){
        return sanPhamDAO.listProduct();
    }
    else if(nameManufacture == "Tất cả")
    {
        return searchInType(nameType, listProduct());
    }
    else if(nameType == "Tất cả")
    {
        return searchInManu(nameManufacture, listProduct());
    }
    else
    {
        List<SanPham> listOfManu = filterInManu(nameManufacture);
        List<SanPham> listOfType = filterInType(nameType);
        List<SanPham> list = new List<SanPham>();
        foreach(SanPham valueType in listOfType)
        {
            foreach(SanPham valueManu in listOfManu)
            {
                if(valueManu.IdProduct == valueType.IdProduct)
                    list.Add(valueManu);
            }
        }
        return list;
    }
}

public static List<SanPham> filterProductOtherZero()
{
    List<SanPham> listFilter = new List<SanPham>();
    List<SanPham> listAll = listProduct();
    foreach(SanPham value in listAll)
    {
        if(value.QuantityProduct != 0)
        {
            listFilter.Add(value);
        }
    }
}

```

```

        }
    }
    return listFilter;
}

public static List<SanPham> filterInType(string idType)
{
    return new SanPhamDAO().searchType(int.Parse(idType));
}
public List<SanPham> filterInManu(string name)
{
    if (name == "Tất cả")
        return sanPhamDAO.listProduct();
    return sanPhamDAO.searchManufacturer(this.idManuProduct(name));
}
public static List<SanPham> filterInManuToId(string idManu)
{
    return new SanPhamDAO().searchManufacturer(int.Parse(idManu));
}

```

Chức năng thanh toán

```
public static List<ChiTietHoaDonDTO> listDetailsBill(int idBill)
{
    ChiTietHoaDonDAO chiTietHoaDonDAO = new ChiTietHoaDonDAO();
    List<ChiTietHoaDonDTO> list =
chiTietHoaDonDAO.listDetailsBill(idBill);
    return list;
}
public static List<HoaDon> listBill()
{
    List<HoaDon> list = new HoaDonDAO().listBill();

    return list;
}
public List<HoaDon> searchBill(string s, List<HoaDon> list)
{
    List<HoaDon> result = new List<HoaDon>();
    while (result.Count == 0)
    {
        result = searchIdBill(s, list);
        if (result.Count == 0)
        {
            result = searchDateBill(s, list);
        }
        if (s.Length > 1)
        {
            s = s.Substring(0, s.Length - 1);
        }
        else
        {
            break;
        }
    }
    return result;
}
public List<HoaDon> searchIdBill(string s, List<HoaDon> result) {
    int length = s.Length;
    List<HoaDon> list = new List<HoaDon>();
    foreach (HoaDon value in result)
    {
        string id = value.IdBill+"";
        if (id.Length >= length)
        {
            id = id.Substring(0, length);
            if (id.CompareTo(s) == 0)
            {
                list.Add(value);
            }
        }
    }
    return list;
}
```



```

    }
    public List<HoaDon> searchDateBill(string s, List<HoaDon> list)
    {
        List<HoaDon> result = new List<HoaDon>();
        result = searchDateDayBill(s, list);
        if(result.Count == 0)
            result = searchDateMonthBill(s, list);
        if(result.Count == 0)
            result = searchDateYearsBill(s, list);
        return result;
    }
    public List<HoaDon> searchDateDayBill(string s, List<HoaDon> result)
    {
        List<HoaDon> list = new List<HoaDon>();
        foreach (HoaDon value in result)
        {
            string day = value.DateBill.Day + "";
            if (day == s)
            {
                list.Add(value);
            }
        }
        return list;
    }
    public List<HoaDon> searchDateMonthBill(string s, List<HoaDon>
result)
    {
        List<HoaDon> list = new List<HoaDon>();
        foreach (HoaDon value in result)
        {
            string month = value.DateBill.Month + "";
            if (month == s)
            {
                list.Add(value);
            }
        }
        return list;
    }
    public List<HoaDon> searchDateYearsBill(string s, List<HoaDon>
result)
    {
        List<HoaDon> list = new List<HoaDon>();
        foreach (HoaDon value in result)
        {
            string year = value.DateBill.Year + "";
            if (year == s)
            {
                list.Add(value);
            }
        }
    }

```

```

    }
    return list;
}
public static HoaDon bill(int idBill)
{
    return new HoaDonDAO().searchIdBill(idBill);
}
public void updateBill(HoaDon bill)
{
    new HoaDonDAO().updateHoaDon(bill);
}

```

Chức năng quản lý sản phẩm

```

public string addProduct(string idProduct, string nameProduct, string
unitPrice, string quantityProduct, string statusProduct, string typeProduct,
string idManufacture, string image)
{
    string result = "";
    if (idProduct == "")
        return "error_idNull";
    if (nameProduct == "")
        return "error_nameNull";

    if (checkIdProduct(idProduct))
        result += "error_duplicateId";
    if (!isNumber(quantityProduct))
        result += "error_quantity";
    if (!isFloat(unitPrice))
        result += "error_unitPrice";
    if (result.Contains("error"))
        return result;
    if (int.Parse(quantityProduct) < 0)
    {
        return "error_quantity";
    }
    if (float.Parse(unitPrice) < 0)
    {
        return "error_unitPrice";
    }
    result = "success";
    if (int.Parse(quantityProduct) != 0)
        statusProduct = "Còn hàng";
    else
        statusProduct = "Hết hàng";
    SanPham sp = new SanPham(idProduct, nameProduct,
float.Parse(unitPrice),
        int.Parse(quantityProduct), statusProduct,
changeIndexToType(typeProduct), changeIndexToManu(idManufacture), image);
    this.sanPhamDAO.addProduct(sp);
    return result;
}

```

```

public int changeIndexToType(string index)
{
    List<DanhMucLapTopDTO> listType = listTypeProduct();
    return listType[int.Parse(index)].IdTypeProduct;
}
public int changeIndexToManu(string index)
{
    List<HangSanPhamDTO> list = listBrand();

    return list[int.Parse(index)].IdManufacture;
}
public string updateProduct(string idProduct, string nameProduct,
string unitPrice, string quantityProduct, string statusProduct, string
typeProduct, string idManufacture, string image)
{
    if (idProduct == "")
        return "error_idNull";
    if (idProduct == "")
        return "error_nameNull";
    string result = "";
    if (!isNumber(quantityProduct))
        result += "error_quantity";
    if (!isFloat(unitPrice))
        result += "error_unitPrice";
    if (!checkIdProduct(idProduct))
        result = "error_duplicateId";
    if (result.Contains("error"))
        return result;
    if (int.Parse(quantityProduct) < 0)
    {
        return "error_quantity";
    }
    if (float.Parse(unitPrice) < 0)
    {
        return "error_unitPrice";
    }
    if (int.Parse(quantityProduct) != 0)
        statusProduct = "Còn hàng";
    else
        statusProduct = "Hết hàng";
    SanPham sp = new SanPham(idProduct, nameProduct,
float.Parse(unitPrice),
        int.Parse(quantityProduct), statusProduct,
changeIndexToType(typeProduct), changeIndexToManu(idManufacture), image);
    this.sanPhamDAO.updateProduct(sp);

    return "success";
}
public string deleteProduct(string idProduct)
{
    if (idProduct == "")
        return "error_idNull";

```

```

        string result = "";
        if (!checkIdProduct(idProduct))
            result = "error_notFoundID";

        result = "success";

        this.sanPhamDAO.deleteProdcut(idProduct);

        return result;
    }
    public bool checkIdProduct(string id)
    {
        foreach(SanPham value in listProduct())
        {
            if (value.IdProduct == id)
                return true;
        }
        return false;
    }

    public static List<SanPham> listProduct()
    {
        List<SanPham> listSp = new SanPhamDAO().listProduct();
        return listSp;
    }
    public static void addQuantity(int quantity, string idProdcut)
    {
        SanPhamDAO sp = new SanPhamDAO();
        List<SanPham> listP = sp.listProduct();
        foreach (SanPham value in listP)
        {
            if( value.IdProduct == idProdcut)
            {
                value.QuantityProduct += quantity;
                sp.updateProdcut(value);
                break;
            }
        }
    }
    public static void deleteQuantity(int quantity, string idProdcut)
    {
        SanPhamDAO sp = new SanPhamDAO();
        List<SanPham> listP = sp.listProduct();
        foreach (SanPham value in listP)
        {
            if (value.IdProduct == idProdcut)
            {
                value.QuantityProduct -= quantity;
                sp.updateProdcut(value);
                break;
            }
        }
    }
}

```

```

list)      public static List<SanPham> searchProduct(string s, List<SanPham>
           {
               List<SanPham> listP = new List<SanPham>();
               while (listP.Count == 0)
               {
                   listP = searchInId(s, list);
                   if (listP.Count == 0)
                   {
                       listP = searchInName(s, list);
                   }
                   if (listP.Count == 0)
                   {
                       listP = searchInType(s, list);
                   }
                   if (listP.Count == 0)
                   {
                       listP = searchInManu(s, list);
                   }
                   if (s.Length <= 1)
                   {
                       break;
                   }
                   else
                   {
                       s = s.Substring(0, s.Length - 1);
                   }
               }
               return listP;
           }
listP)     public static List<SanPham> searchInName(string s, List<SanPham>
           {
               int length = s.Length;
               List<SanPham> list = new List<SanPham>();

               foreach (SanPham value in listP)
               {
                   string sOb = value.NameProduct.ToString();
                   if (sOb.Length >= length)
                   {
                       sOb = sOb.Substring(0, length);
                       sOb = sOb.ToLower();
                       if (s.ToLower() == sOb)
                       {
                           list.Add(value);
                       }
                   }
               }

               return list;
           }

```

```

    }
    public static List<SanPham> searchInQuantity(string s, List<SanPham>
listP)
    {
        int length = s.Length;
        List<SanPham> list = new List<SanPham>();

        foreach (SanPham value in listP)
        {
            string sOb = value.QuantityProduct.ToString();
            if (sOb.Length >= length)
            {
                if (s.CompareTo(sOb.Substring(0, length)) == 0)
                {
                    list.Add(value);
                }
            }
        }
        return list;
    }
    public static List<SanPham> searchInId(string s, List<SanPham> listP)
    {
        int length = s.Length;
        List<SanPham> list = new List<SanPham>();
        foreach (SanPham value in listP)
        {
            string sOb = value.IdProduct.ToString();
            if (sOb.Length >= length)
            {
                sOb = sOb.Substring(0, length);
                if (s == sOb)
                {
                    list.Add(value);
                }
            }
        }
        return list;
    }
    public static List<SanPham> searchInManu(string s, List<SanPham>
listP)
    {
        int length = s.Length;
        List<SanPham> list = new List<SanPham>();
        List<HangSanPhamDTO> listHang = listBrand();

        List<HangSanPhamDTO> listManuSearch = new List<HangSanPhamDTO>();
        foreach (HangSanPhamDTO value in listHang)
        {
            string sOb = value.NameManufacturer.ToString();
            if (sOb.Length >= length)
            {

```

```

        sOb = sOb.Substring(0, length);
        sOb = sOb.ToLower();

        if (s.ToLower() == sOb)
        {
            listManuSearch.Add(value);
        }
    }
}
foreach (SanPham value in listP)
{
    foreach (HangSanPhamDTO item in listManuSearch)
    {
        if (item.IdManufacture == value.IDManuFacture)
        {
            list.Add(value);
            break;
        }
    }
}
return list;
}

public static List<SanPham> searchInType(string s, List<SanPham>
listP)
{
    int length = s.Length;
    List<SanPham> list = new List<SanPham>();
    List<DanhMucLapTopDTO> listDanhMuc = listTypeProduct();
    List<DanhMucLapTopDTO> listTypeSearch = new
List<DanhMucLapTopDTO>();
    foreach (DanhMucLapTopDTO value in listDanhMuc)
    {
        string sOb = value.NameTypeProduct.ToString();
        if (sOb.Length >= length)
        {
            sOb = sOb.Substring(0, length);
            sOb = sOb.ToLower();

            if (s.ToLower() == sOb)
            {
                listTypeSearch.Add(value);
            }
        }
    }
    foreach (SanPham value in listP)
    {
        foreach (DanhMucLapTopDTO valueP in listTypeSearch)
            if (value.IdTypeProduct == valueP.IdTypeProduct)

```

```

        {
            list.Add(value);
            break;
        }

    }
    return list;
}
public SanPham searchId(string id)
{
    foreach(SanPham value in listProduct())
    {
        if (value.IdProduct == id)
            return value;
    }
    return null;
}

```

Chức năng quản lý khuyến mãi

```

public List<KhuyenMaiDTO> searchPromotion(string s)
{
    List<KhuyenMaiDTO> result = new List<KhuyenMaiDTO>();
    while (result.Count == 0)
    {
        result = searchIdPromotion(s);
        if (result.Count == 0)
        {
            result = searchNamePromotion(s);
        }
        if (s.Length > 1)
        {
            s = s.Substring(0, s.Length - 1);
        }
        else
        {
            break;
        }
    }
    return result;
}
public List<KhuyenMaiDTO> searchIdPromotion(string s)
{
    int length = s.Length;
    List<KhuyenMaiDTO> list = listPromotion();
    List<KhuyenMaiDTO> result = new List<KhuyenMaiDTO>();

    foreach (KhuyenMaiDTO item in list)
    {
        string id = item.IdPromotion + "";
        if(id == s)

```



```

        result.Add(item);
    }
    return result;
}
public List<KhuyenMaiDTO> searchNamePromotion(string s)
{
    int length = s.Length;
    List<KhuyenMaiDTO> list = listPromotion();
    List<KhuyenMaiDTO> result = new List<KhuyenMaiDTO>();

    foreach (KhuyenMaiDTO item in list)
    {
        string name = item.NamePromotion + "";
        if(name.Length >= length)
        {
            name = name.Substring(0, length);
            name = name.ToLower();
            if (name.CompareTo(s.ToLower()) == 0)
                result.Add(item);
        }
    }
    return result;
}
public List<SanPham> searchP(string s)
{
    return searchProduct(s, listProduct());
}
public void addPromotion()
{
    this.idPromotion = maxIdPromotion() + 1;
}
public int maxIdPromotion()
{
    List<int> listId = new List<int>();
    foreach (KhuyenMaiDTO item in listPromotion())
    {
        listId.Add(item.IdPromotion);
    }
    return MaxNumber(listId);
}
public string savePromotion(string name, string status, DateTime
dateStart, DateTime dateEnd)
{
    if (name == "")
        return "error_name";
    string result = "";
    result += "success";
    KhuyenMaiDTO khuyenMaiDTO = new KhuyenMaiDTO(this.idPromotion,
name, status, dateStart, dateEnd);
    khuyenMaiDAO.addKhuyenMai(khuyenMaiDTO);
    saveDetailsPromotion(this.listDt);
    return result;
}

```

```

    }
    public void saveDetailsPromotion(List<ChiTietKhuyenMaiDTO> list) {
        foreach (ChiTietKhuyenMaiDTO item in list)
        {
            chiTietKhuyenMaiDAO.addChiTietKhuyenMai(item);
        }
    }
    public string addDetailsPromotion(string idProduct, string discount)
    {
        if (idProduct == "")
            return "error_idNull";
        if(discount == "")
            return "error_discountNull";
        string result = "";
        if (!isFloat(discount))
            result += "error_discount";
        if (checkDuplicateDetailsPromotion(idProduct, this.idPromotion))
            result += "error_duplicate";
        if(result.Contains("error"))
            return result;
        result += "success";
        ChiTietKhuyenMaiDTO chiTietKhuyenMaiDTO = new
ChiTietKhuyenMaiDTO(float.Parse(discount), "", idProduct, this.idPromotion);
        this.listDt.Add(chiTietKhuyenMaiDTO);
        return result;
    }
    public string updateDetailsPromotion(string idProduct, string
discount)
    {
        if (idProduct == "")
            return "error_idNull";
        if (discount == "")
            return "error_discountNull";
        string result = "";
        if (!isFloat(discount))
            result += "error_discount";
        if (!checkDuplicateDetailsPromotion(idProduct, this.idPromotion))
            result += "error_duplicate";
        if (result.Contains("error"))
            return result;

        result += "success";
        ChiTietKhuyenMaiDTO chiTietKhuyenMaiDTO = new
ChiTietKhuyenMaiDTO(float.Parse(discount), "", idProduct, this.idPromotion);
        updateDetailsPromotion(chiTietKhuyenMaiDTO);
        return result;
    }
    public void updateDetailsPromotion(ChiTietKhuyenMaiDTO ct)
    {
        foreach(ChiTietKhuyenMaiDTO value in this.listDt)
        {

```

```

        if(value.IdProduct == ct.IdProduct && value.IdPromotion ==
ct.IdPromotion)
        {
            value.Discount = ct.Discount;
            break;
        }
    }
}

public string deleteDetailsPromotion(string idProduct)
{
    if (idProduct == "")
        return "error_idNull";
    string result = "";
    if (!checkDuplicateDetailsPromotion(idProduct, this.idPromotion))
        result += "error_duplicate";
    if (result.Contains("error"))
        return result;
    ChiTietKhuyenMaiDTO ct = null;
    foreach (ChiTietKhuyenMaiDTO value in this.listDt)
    {
        if (value.IdProduct == idProduct && value.IdPromotion ==
this.idPromotion)
        {
            ct = value;
            break;
        }
    }
    this.listDt.Remove(ct);
    return "success";
}

public bool checkDuplicateDetailsPromotion(string idProduct, int
idPromotion)
{
    foreach(ChiTietKhuyenMaiDTO value in this.listDt){
        if(value.IdProduct == idProduct && value.IdPromotion ==
idPromotion)
            return true;
    }
    return false;
}

public string exportPromotionExcel(string idPromotion)
{
    if (idPromotion == "")
        return "error";
    List<ChiTietKhuyenMaiDTO> listDtP =
listDetailsPromotion(int.Parse(idPromotion));
    List<SanPham> listP = new List<SanPham>();
    foreach(ChiTietKhuyenMaiDTO value in listDtP)
    {
        listP.Add(sanPhamDAO.searchId(value.IdProduct));
    }
}

```

```

Dictionary<string, string> listExcel = new Dictionary<string,
string>();
for(int i = 0; i < listDtP.Count; i++)
{
    listExcel.Add(listP[i].NameProduct, listDtP[i].Discount+"");
}
return exportExcelPromotion(listExcel, idPromotion);
}

```

Chức năng quản lý nhân viên

```

public static List<NhanVien> removeNullEmployee()
{
    List<NhanVien> list = new NhanVienDao().listEmployee();
    int index = 0;
    foreach (NhanVien v in list)
    {
        if (v.Id == "")
            break;
        index++;
    }
    list.RemoveAt(index);
    return list;
}

public string addEmployee(string idEmployee, string nameEmployee,
string serviceEmployee, DateTime birthday, string phoneNumber, string
salaryEmployee)
{
    if(idEmployee == "")
        return "error_isNull";
    if (nameEmployee == "")
        return "error_nameNull";
    string result = "";
    if (checkIdEmployee(idEmployee))
        result = "error_duplicateId";
    phoneNumber = changePhoneNumber(phoneNumber);
    if (phoneNumber.Contains("error"))
        result += "error_phone";
    if (!isFloat(salaryEmployee))
        result += "error_float";
    if (result.Contains("error"))
        return result;
    result = "success";
    NhanVien employee = new NhanVien(idEmployee, nameEmployee,
serviceEmployee, birthday, phoneNumber, 0, 0,idEmployee,
float.Parse(salaryEmployee));
    new Login().addUser(idEmployee, "1234", serviceEmployee);
    nvDAO.addNhanVien(employee);
    return result;
}

```

```

        public string updateEmployee(string idEmployee, string nameEmployee,
string serviceEmployee, DateTime birthday, string phoneNumber, string
salaryEmployee)
        {
            if (idEmployee == "")
                return "error_isNull";
            if (nameEmployee == "")
                return "error_nameNull";
            string result = "";
            if (!checkIdEmployee(idEmployee))
                result = "error_duplicateId";
            phoneNumber = changePhoneNumber(phoneNumber);
            if (phoneNumber.Contains("error"))
                result += "error_phone";
            if (!isFloat(salaryEmployee))
                result += "error_float";
            if (result.Contains("error"))
                return result;
            result = "success";
            NhanVien employee = new NhanVien(idEmployee, nameEmployee,
serviceEmployee, birthday, phoneNumber, 0, 0, idEmployee,
float.Parse(salaryEmployee));
            nvDAO.updateNhanVien(employee);
            return result;
        }

        public string deleteEmployee(string idEmployee)
        {
            if (idEmployee == "")
                return "error_isNull";
            if (!checkIdEmployee(idEmployee))
                return "error_duplicateId";
            nvDAO.deleteEmployee(idEmployee);
            new HoaDonDAO().updateBillToEmployee(idEmployee);
            return "success";
        }

        public bool checkIdEmployee(string id)
        {
            foreach(NhanVien value in listEmployee())
            {
                if (value.Id == id)
                    return true;
            }
            return false;
        }

        public static List<NhanVien> searchEmployee(string s)
        {
            List<NhanVien> result = new List<NhanVien> ();
            while(result.Count == 0)
            {
                result = searchId(s);
            }
        }
    }

```

```

        if(result.Count == 0)
        {
            result = searchName(s);
        }
        if (result.Count == 0)
        {
            result = searchService(s);
        }
        if (result.Count == 0)
        {
            result = searchPhone(s);
        }
        if (s.Length <= 1)
            break;
        else
            s = s.Substring(0, s.Length - 1);
    }
    return result;
}

public static List<NhanVien> searchId(string s)
{
    List<NhanVien> list = listEmployee();
    List<NhanVien> result = new List<NhanVien> ();
    foreach(NhanVien value in list)
    {
        string id = value.Id;
        if(id == s)
        {
            result.Add(value);
            break;
        }
    }
    return result;
}

public static List<NhanVien> searchName(string s)
{
    List<NhanVien> list = listEmployee();
    List<NhanVien> result = new List<NhanVien>();
    foreach (NhanVien value in list)
    {
        string name = value.Name;
        if(name.Length >= s.Length)
        {
            if(name.Substring(0,
s.Length).ToLower().CompareTo(s.ToLower()) == 0)
                result.Add(value);
        }
    }
    return result;
}

```

```

    }
    public static List<NhanVien> searchService(string s)
    {
        List<NhanVien> list = listEmployee();
        List<NhanVien> result = new List<NhanVien>();
        foreach (NhanVien value in list)
        {
            string service = value.ServiceEmployee;
            if (service.Length >= s.Length)
            {
                if (service.Substring(0,
s.Length).ToLower().CompareTo(s.ToLower()) == 0)
                    result.Add(value);
            }
        }
        return result;
    }

    public static List<NhanVien> searchPhone(string s)
    {
        List<NhanVien> list = listEmployee();
        List<NhanVien> result = new List<NhanVien>();
        foreach (NhanVien value in list)
        {
            string phone = value.Phone;
            if (phone.Length >= s.Length)
            {
                if (phone.Substring(0, s.Length).CompareTo(s) == 0)
                    result.Add(value);
            }
        }
        return result;
    }

    public List<NhanVien> topEmployee()
    {
        List<NhanVien> list = listEmployee();
        List<NhanVien> listSort = new List<NhanVien>();
        int length = list.Count;
        for (int i = 0; i < length; i++)
        {
            float max = list[0].TotalSales;
            int index = 0;
            for (int j = 1; j < list.Count; j++)
            {
                if (list[j].TotalSales > max)
                {
                    max = list[j].TotalSales;
                    index = j;
                }
            }
        }
    }

```

```

        }
        listSort.Add(list[index]);
        list.RemoveAt(index);
    }
    return listSort;
}

public NhanVien searchToId(string idEmployee)
{
    return nvDAO.searchEmployee(idEmployee);
}

```

Chức năng quản lý khách hàng

```

public static List<KhachHang> listCustomer()
{
    List<KhachHang> list = new KhachHangDAO().listCustomer();
    list.RemoveAt(0);
    return list;
}

public string addCustomer(string id, string name, string address,
string phoneNumber, string email)
{
    if (id == "")
        return "error_isNull";
    if (name == "")
        return "error_nameNull";
    string result = "";
    if (checkIdCustomer(id))
        result = "error_duplicateId";
    phoneNumber = changePhoneNumber(phoneNumber);
    if (phoneNumber.Contains("error"))
        result += "error_phone";
    if (result.Contains("error"))
        return result;
    result = "success";
    KhachHang customer = new KhachHang(id, name, address,
phoneNumber, email);
    khachHangDAO.addKhachHang(customer);
    return result;
}

public string updateCustomer(string id, string name, string address,
string phoneNumber, string email)
{
    if (id == "")
        return "error_isNull";
    if (name == "")
        return "error_nameNull";
    string result = "";
    if (!checkIdCustomer(id))
        result = "error_duplicateId";
    phoneNumber = changePhoneNumber(phoneNumber);
}

```



```

        if (phoneNumber.Contains("error"))
            result += "error_phone";
        if (result.Contains("error"))
            return result;
        result = "success";
        KhachHang customer = new KhachHang(id, name, address,
        phoneNumber, email);
        khachHangDAO.updateKhachHang(customer);
        return result;
    }
    public string deleteCustomer(string id)
    {
        if (id == "")
            return "error_isNull";

        string result = "";
        if (!checkIdCustomer(id))
            result = "error_duplicateId";

        if (result.Contains("error"))
            return result;
        result = "success";
        hoaDonDAO.updateBillToCustomer(id);
        khachHangDAO.deleteKhachHang(id);

        return result;
    }
    public bool checkIdCustomer(string id)
    {
        foreach (KhachHang value in listCustomer())
        {
            if (value.Id == id)
                return true;
        }
        return false;
    }

    public static List<KhachHang> searchCustomer(string s)
    {
        List<KhachHang> result = new List<KhachHang>();
        while (result.Count == 0)
        {
            result = searchId(s);
            if (result.Count == 0)
            {
                result = searchName(s);
            }
            if (result.Count == 0)
            {
                result = searchPhone(s);
            }
            if (s.Length <= 1)
            {

```

```

        break;
    }
    else
    {
        s = s.Substring(0, s.Length - 1);
    }
}
return result;
}
public static List<KhachHang> searchId(string s)
{
    List<KhachHang> list = listCustomer();

    List<KhachHang> result = new List<KhachHang>();
    foreach (KhachHang value in list)
    {
        string id = value.Id;
        if (id == s)
        {
            result.Add(value);
            break;
        }
    }

    return result;
}

public static List<KhachHang> searchName(string s)
{
    List<KhachHang> list = listCustomer();
    List<KhachHang> result = new List<KhachHang>();
    foreach (KhachHang value in list)
    {
        string name = value.Name;
        if (name.Length >= s.Length)
        {
            if (name.Substring(0,
s.Length).ToLower().CompareTo(s.ToLower()) == 0)
                result.Add(value);
        }
    }
    return result;
}

public static List<KhachHang> searchPhone(string s)
{
    List<KhachHang> list = listCustomer();
    List<KhachHang> result = new List<KhachHang>();
    foreach (KhachHang value in list)
    {
        string phone = value.Phone;

```

```

        if (phone.Length >= s.Length)
        {
            if (phone.Substring(0, s.Length).CompareTo(s) == 0)
                result.Add(value);
        }
    }
    return result;
}

public List<Dictionary<string, string>> topCustomer()
{
    List<KhachHang> listCu = listCustomer();
    var map = sumMoneyInCustomer(listCu);

    List<Dictionary<string, string>> result = new
List<Dictionary<string, string>>();

    foreach (var value in map)
    {
        Dictionary<string, string> dict = new Dictionary<string,
string>();
        dict.Add("name",
khachHangDAO.searchNameCustomer(value.Key).Name);
        dict.Add("sumBill",
hoaDonDAO.quantityBill(value.Key).ToString());
        dict.Add("sumMoney", value.Value.ToString());
        result.Add(dict);
    }
    return result;
}

public Dictionary<string, float> sumMoneyInCustomer(List<KhachHang>
listCu)
{
    var result = new Dictionary<string, float>();
    foreach (KhachHang value in listCu)
    {
        result.Add(value.Id,
hoaDonDAO.sumMoneyInIdCustomer(value.Id));
    }

    return sortSumMoney(result);
}

public Dictionary<string, float> sortSumMoney(Dictionary<string,
float> list)
{
    int length = list.Count;
    var result = new Dictionary<string, float>();
    for (int i = 0; i < length; i++)
    {

        float max = 0;
        string id = "";
        foreach (var value in list)
        {

```

```

        if (value.Value >= max)
        {
            max = value.Value;
            id = value.Key;
        }

    }
    result.Add(id, max);
    list.Remove(id);

}

return result;
}

public KHachHang searchCustomerToId(string idCustomer)
{
    return khachHangDAO.searchNameCustomer(idCustomer);
}

```

Chức năng quản lý phiếu nhập kho

```

public void addCoupon()
{
    phieuYeuCauNhapHang = new PhieuYeuCauNhapHang();
    int idCoupon = maxId();
    this.phieuYeuCauNhapHang.IdCoupon = idCoupon;
}

public string addDetailsCoupon(string idProduct, string quantity,
string unitPrice, string intoMoney)
{
    if (idProduct == "")
        return "error_idNull";
    if (checkDuplicateId(idProduct))
        return "error_duplicateId";
    string result = "";
    if (!isNumber(quantity))
        return "error_quantity";

    result = "success";

    ChiTietPhieuYeuCauNhapHangDTO chiTietPhieuYeuCauNhapHangDTO = new
    ChiTietPhieuYeuCauNhapHangDTO(idProduct, this.phieuYeuCauNhapHang.IdCoupon,
        int.Parse(quantity), float.Parse(unitPrice),
    float.Parse(intoMoney));
    this.listDt.Add(chiTietPhieuYeuCauNhapHangDTO);
    return result;
}

public bool checkDuplicateId(string idProduct)
{

```

```

        foreach (ChiTietPhieuYeuCauNhapHangDTO value in this.listDt)
            if (value.IdProduct == idProduct)
                return true;
        return false;
    }

    public string updateDetailsCoupon(string idProduct, string quantity,
string unitPrice, string intoMoney)
    {
        if (idProduct == "")
            return "error_idNull";
        if (!checkDuplicateId(idProduct))
            return "error_duplicateId";
        if (!isNumber(quantity))
            return "error_quantity";

        string result = "success";

        ChiTietPhieuYeuCauNhapHangDTO chiTietPhieuYeuCauNhapHangDTO = new
ChiTietPhieuYeuCauNhapHangDTO(idProduct, this.phieuYeuCauNhapHang.IdCoupon,
            int.Parse(quantity), float.Parse(unitPrice),
float.Parse(intoMoney));
        updateProduct(chiTietPhieuYeuCauNhapHangDTO);
        return result;
    }

    public void updateProduct(ChiTietPhieuYeuCauNhapHangDTO p)
    {
        foreach (ChiTietPhieuYeuCauNhapHangDTO value in listDt)
        {
            if (value.IdProduct == p.IdProduct && value.IdCoupon ==
p.IdCoupon)
            {
                value.QuantityImport = p.QuantityImport;
                value.IntoMoney = p.IntoMoney;
            }
        }
    }

    public string deleteDetailsCoupon(string idProduct)
    {
        if (idProduct == "")
            return "error_idNull";
        if (!checkDuplicateId(idProduct))
            return "error_duplicateId";
        string result = "success";

        int index = 0;
        foreach (ChiTietPhieuYeuCauNhapHangDTO value in this.listDt)
        {
            if (value.IdCoupon == this.phieuYeuCauNhapHang.IdCoupon &&
value.IdProduct == idProduct)
            {
                this.deleteDetailsCoupon(value);
                break;
            }
        }
    }

```

```

        }
        index++;
    }
    listDt.RemoveAt(index);

    return result;
}

public void deleteDetailsCoupon(ChiTietPhieuYeuCauNhapHangDTO ct)
{
    this.phieuYeuCauNhapHang.IntoMoney -= ct.IntoMoney;
}

public void updateDetailsCoupon(ChiTietPhieuYeuCauNhapHangDTO ct)
{
    foreach (var item in this.listDt)
    {
        if (item.IdProduct == ct.IdProduct && item.IdCoupon ==
ct.IdCoupon)
        {
            item.QuantityImport = ct.QuantityImport;
            item.IntoMoney = ct.IntoMoney;
            updateSumCoupon(item, ct);
            break;
        }
    }
}

public int maxId()
{
    int max = 0;
    foreach (PhieuYeuCauNhapHang value in listCoupon())
    {
        if (value.IdCoupon > max) max = value.IdCoupon;
    }
    return max + 1;
}

public void sumIntoMoney(float intoMoney)
{
    this.phieuYeuCauNhapHang.IntoMoney += intoMoney;
}

public void updateSumCoupon(ChiTietPhieuYeuCauNhapHangDTO dtOld,
ChiTietPhieuYeuCauNhapHangDTO dtNew)
{
    this.phieuYeuCauNhapHang.IntoMoney -= dtOld.IntoMoney;
    this.phieuYeuCauNhapHang.IntoMoney += dtNew.IntoMoney;
}

public void saveCoupon(string idSupplier)
{
    this.phieuYeuCauNhapHang.IdSupplier = idSupplier;
    this.phieuYeuCauNhapHang.DateAdded = DateTime.Now;
    this.phieuYeuCauNhapHang.IntoMoney = sumReceipt();
    phieuYeuCauNhapHangDAO.addCoupon(this.phieuYeuCauNhapHang);
    foreach (ChiTietPhieuYeuCauNhapHangDTO value in this.listDt)
    {

```

```

        SanPham p = sanPhamDAO.searchId(value.IdProduct);
        p.QuantityProduct = p.QuantityProduct + value.QuantityImport;
        sanPhamDAO.updateProduct(p);
        ctDAO.addDetailCoupon(value);
    }
}
public float sumReceipt()
{
    float sum = 0;
    foreach (ChiTietPhieuYeuCauNhapHangDTO value in listDt)
    {
        sum += value.IntoMoney;
    }
    return sum;
}
public List<PhieuYeuCauNhapHang> searchId(string s)
{
    List<PhieuYeuCauNhapHang> list = listCoupon();
    List<PhieuYeuCauNhapHang> result = new
List<PhieuYeuCauNhapHang>();
    while (list.Count == 0)
    {
        result = searchId(s, list);
        if (s.Length > 1)
        {
            s = s.Substring(0, s.Length - 1);
        }
    }
    return result;
}
public List<PhieuYeuCauNhapHang> searchId(string s,
List<PhieuYeuCauNhapHang> list)
{
    List<PhieuYeuCauNhapHang> result = new
List<PhieuYeuCauNhapHang>();
    int length = s.Length;
    foreach (PhieuYeuCauNhapHang value in list)
    {
        if (value.IdCoupon.ToString().Length >= length)
        {
            string temp = value.IdCoupon.ToString();
            temp = temp.Substring(0, length);
            if (temp == s)
                result.Add(value);
        }
    }
    return result;
}
public static string exportReceiptExcel(int idReceipt)
{

```

```

        List<ChiTietPhieuYeuCauNhapHangDTO> listDetailsReceipt =
listDtCoupon(idReceipt.ToString());
        List<SanPham> listP = new List<SanPham>();
        foreach (ChiTietPhieuYeuCauNhapHangDTO value in
listDetailsReceipt)
        {
            listP.Add(new SanPhamDAO().searchId(value.IdProduct));
        }
        PhieuYeuCauNhapHang p = new
PhieuYeuCauNhapHangDAO().coupon(idReceipt);
        return exportExcelReceipt(p, listDetailsReceipt, listP);
    }

```

Chức năng thống kê

```

public Dictionary<string, string> overview(string year)
{
    Dictionary<string, string> map = new Dictionary<string,
string>();
    string quantityEmployee = this.quantityEmployee();
    string quantityProduct = this.quantityProduct();
    string total = this.totalRevenue(this.totalRevenueYear(year));
    string quantityCustomer = this.quantityCustomer();

    map.Add("employee", quantityEmployee);
    map.Add("product", quantityProduct);
    map.Add("total", total);
    map.Add("customer", quantityCustomer);
    return map;
}
public string quantityEmployee()
{
    List<NhanVien> listEmployee = employeeDao.listEmployee();
    return listEmployee.Count + "";
}
public List<HoaDon> totalRevenueYear(string year)
{
    List<HoaDon> listBill = billDao.listBill();
    List<HoaDon> result = new List<HoaDon>();
    foreach (HoaDon hoaDon in listBill)
    {
        if (hoaDon.DateBill.Year.ToString() == year)
            result.Add(hoaDon);
    }
    return result;
}
public List<HoaDon> totalRevenueMonth(string month, string year)
{
    List<HoaDon> listBillYear = totalRevenueYear(year);
    List<HoaDon> result = new List<HoaDon>();
    foreach (HoaDon hoaDon in listBillYear)

```



```

        {
            if (hoaDon.DateBill.Year.ToString() == year &&
hoaDon.DateBill.Month.ToString() == month)
                result.Add(hoaDon);
        }
        return result;
    }
    public string totalRevenue(List<HoaDon> listBillYear)
    {
        float sum = 0;

        foreach (HoaDon hoaDon in listBillYear)
        {
            sum += hoaDon.SumPay;
        }
        return sum + "";
    }
    public string quantityProduct()
    {
        return new SanPhamDAO().listProduct().Count + "";
    }
    public string quantityCustomer()
    {
        return new KhachHangDAO().listCustomer().Count + "";
    }
    public Dictionary<string, string> totalMonth(string year)
    {
        Dictionary<string, string> result = new Dictionary<string,
string>();

        for (int i = 1; i < 13; i++)
        {
            List<HoaDon> listBillMonth = totalRevenueMonth(i+"", year);
            string sum = totalRevenue(listBillMonth);
            result.Add(i+"", sum);
        }
        return result;
    }
    public List<Dictionary<string, string>> topProduct(string year)
    {
        List<HoaDon> listBillYear = totalRevenueYear(year);
        List<SanPham> listProduct = new SanPhamDAO().listProduct();
        List<ChiTietHoaDonDTO> listDetailsBillInYear =
this.listDeitalsBillInYear(listBillYear);
        Dictionary<string, float> totalProduct =
this.totalProductInYear(listDetailsBillInYear, listProduct);
        Dictionary<string, float> topTotalProduct =
this.topTotalProduct(totalProduct);
        List<Dictionary<string, string>> result =
topProduct(listDetailsBillInYear, topTotalProduct);
        return result;
    }

```

```

    }
    public List<ChiTietHoaDonDTO> listDeitalsBillInYear(List<HoaDon>
listBillYear)
    {
        List<ChiTietHoaDonDTO> result = new List<ChiTietHoaDonDTO>();
        foreach(HoaDon hoaDon in listBillYear)
        {
            List<ChiTietHoaDonDTO> list = new
ChiTietHoaDonDAO().listDetailsBill(hoaDon.IdBill);
            foreach(ChiTietHoaDonDTO value in list)
            {
                result.Add(value);
            }
        }
        return result;
    }
    public Dictionary<string, float>
totalProductInYear(List<ChiTietHoaDonDTO> list, List<SanPham> listProduct)
    {
        Dictionary<string, float> result = new Dictionary<string,
float>();
        foreach(SanPham sanPham in listProduct)
        {
            float sum = 0;
            foreach(ChiTietHoaDonDTO value in list)
            {
                if (value.IdProduct == sanPham.IdProduct)
                    sum += value.IntoMoney;
            }
            result.Add(sanPham.IdProduct, sum );
        }
        return result;
    }
    public Dictionary<string, float> topTotalProduct(Dictionary<string,
float> list)
    {
        var items = from pair in list
                    orderby pair.Value descending
                    select pair;
        Dictionary<string, float> result = new Dictionary<string,
float>();
        foreach (KeyValuePair<string,float> item in items)
        {
            result.Add(item.Key, item.Value);
        }
        return result;
    }
    public List<Dictionary<string, string>>
topProduct(List<ChiTietHoaDonDTO> listDetailsProduct, Dictionary<string,
float> listTopProduct)
    {
        List<Dictionary<string, string>> result = new
List<Dictionary<string, string>>();

```

```

foreach(var value in listTopProduct)
{
    int quantity = 0;
    SanPham sanPham = new SanPhamDAO().searchId(value.Key);
    foreach(ChiTietHoaDonDTO item in listDetailsProduct)
    {
        if(sanPham.IdProduct == item.IdProduct)
        {
            quantity += item.QuantityProduct;
        }
    }
    Dictionary<string, string> map = new Dictionary<string,
string>();

    map.Add("name", sanPham.NameProduct);
    map.Add("total", value.Value + "");
    map.Add("quantity", quantity + "");
    result.Add(map);
}
return result;
}

```

Tài liệu tham khảo

[1] https://vi.wikipedia.org/wiki/C_Sharp

[2] <https://200lab.io/blog/sql-server-la-gi/>

[3] <https://www.imic.edu.vn/tin-tuc-cong-nghe/30518/uu-nhuoc-diem-cua-c-net-nen-hoc-lap-trinh-c-o-dau-thi-tot-nhat-ha-noi-.html>