

BỘ GIÁO DỤC VÀ ĐÀO TẠO BỘ NÔNG NGHIỆP VÀ PTNT
TRƯỜNG ĐẠI HỌC THỦY LỢI



CAO BÁ LỢI

XÂY DỰNG TRẠNG ĐIỆN TỬ BÁN ĐỒ GIA DỤNG

ĐỒ ÁN TỐT NGHIỆP

HÀ NỘI, NĂM 2024

BỘ GIÁO DỤC VÀ ĐÀO TẠO BỘ NÔNG NGHIỆP VÀ PTNT
TRƯỜNG ĐẠI HỌC THỦY LỢI

CAO BÁ LỢI

XÂY DỰNG TRANG ĐIỆN TỬ BÁN ĐỒ GIA DỤNG

Ngành : Công nghệ thông tin

Mã số: 7480201

NGƯỜI HƯỚNG DẪN 1.TS.NGUYỄN VĂN THẨM

HÀ NỘI, NĂM 2024



TRƯỜNG ĐẠI HỌC THỦY LỢI KHOA CÔNG
NGHỆ THÔNG TIN

BẢN TÓM TẮT ĐỀ CƯƠNG ĐỒ ÁN TỐT NGHIỆP

TÊN ĐỀ TÀI: Xây dựng trang điện tử bán đồ gia dụng

Sinh viên thực hiện: Cao Bá Lợi

Lớp: 62TH3

Giáo viên hướng dẫn: Nguyễn Văn Thâm

TÓM TẮT ĐỀ TÀI

1. Bài toán cần giải quyết

Trong thời đại công nghệ phát triển mạnh mẽ như ngày nay, việc sử dụng internet để mua sắm đã trở thành một xu hướng không thể phủ nhận. Việc tìm kiếm và mua các sản phẩm gia dụng thông qua các trang web bán hàng trực tuyến không chỉ tiết kiệm thời gian mà còn mang lại sự tiện lợi và đa dạng trong lựa chọn. Với sự gia tăng của nhu cầu mua sắm trực tuyến, việc xây dựng và phát triển một trang web bán đồ gia dụng không chỉ là một cơ hội kinh doanh mà còn là một bước tiến quan trọng trong việc kết nối người tiêu dùng với các sản phẩm chất lượng và dịch vụ tốt nhất. Chính vì lý do đó nên em đã chọn đề tài: **“Xây dựng trang điện tử bán đồ gia dụng”** để có thể giúp người dùng tìm kiếm và mua hàng thuận tiện hơn.

2. Giải pháp công nghệ

- Công nghệ sử dụng: .Net Core 8, VueJs
- Ngôn ngữ lập trình: C#, JavaScript
- Hệ quản trị cơ sở dữ liệu: My SQL

CÁC MỤC TIÊU CHÍNH

- Tìm hiểu tổng quan kiến trúc và cách thức để xây dựng một website bán hàng trực tuyến.
- Tìm hiểu và áp dụng các công nghệ: .NetCore, VueJs. Các kiến thức về lập trình C#, JavaScript, làm việc với hệ quản trị cơ sở dữ liệu MySql.
- Khảo sát, phân tích thiết kế hệ thống, viết tài liệu đặc tả.
- Thiết kế website thử nghiệm để triển khai trong thực tế.
- Kiểm thử và sửa lỗi.

KẾT QUẢ DỰ KIẾN

- Các báo cáo, tài liệu đặc tả theo yêu cầu theo yêu cầu
- Thiết kế, xây dựng hệ thống đảm bảo các chức năng cơ bản như:
 - + Đăng nhập, đăng ký.
 - + Quản lý người dùng: Xem danh sách người dùng, xóa người dùng.
 - + Quản lý thông tin cá nhân: Người dùng quản lý các thông tin cá nhân như tên hiển thị, hình ảnh đại diện, đổi mật khẩu.
 - + Quản lý giỏ hàng: Người dùng có thể thêm, sửa, xóa giỏ hàng trước khi thanh toán.
 - + Quản lý đơn hàng: Người dùng và người quản trị có thể xem lịch sử đơn hàng, trạng thái đơn hàng và thông tin liên quan.
 - + Xử lý đơn hàng.
 - + Quản lý sản phẩm: Thêm mới, cập nhật thông tin, xóa sản phẩm.
 - + Tìm kiếm, lọc sản phẩm theo các tiêu chí như: thương hiệu, giá cả...
 - + Thanh toán đơn hàng.
 - + Xem chi tiết sản phẩm: hình ảnh, giá cả, mô tả.
 - + Quản lý danh mục sản phẩm.
 - + Quản lý chương trình giảm giá: thêm chương trình giảm giá, điều chỉnh mức giảm giá, điều chỉnh thời gian giảm giá.
 - + Đánh giá sản phẩm đã mua.
 - + Thống kê doanh thu.
- Các tệp dữ liệu, giao diện sử dụng.
- Thiết kế, xây dựng và cài đặt được website bán đồ gia dụng.
- Website đảm bảo chức năng cơ bản cho người quản trị website, người dùng:
 - + Dành cho người quản trị bao gồm các trang cho phép: Quản lý đơn hàng, Quản lý người dùng, Quản lý sản phẩm, Quản lý thanh toán, Quản lý chương trình giảm giá, Quản lý danh mục sản phẩm, Xử lý đơn hàng, Thống kê doanh thu.
 - + Dành cho người dùng bao gồm các trang: Trang chủ, Quản lý thông tin cá nhân, Đăng nhập, Đăng ký, Quản lý giỏ hàng, Thanh toán, Xem chi tiết sản phẩm, Tìm kiếm sản phẩm, Đánh giá sản phẩm.



CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập - Tự do - Hạnh phúc



NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên: Cao Bá Lợi

Hệ đào tạo : Đại học chính quy

Lớp: 62TH3

Ngành: Công nghệ thông tin

Khoa: Công nghệ thông tin

1. TÊN ĐỀ TÀI: Xây dựng trang điện tử bán đồ gia dụng.

2. CÁC TÀI LIỆU CƠ BẢN:

3. NỘI DUNG CÁC PHẦN THUYẾT MINH VÀ TÍNH TOÁN:

Nội dung cần thuyết minh	Tỉ lệ %
Chương 1: Cơ sở lý thuyết và công nghệ phát triển website - Bài toán quản lý - Công nghệ phát triển	25%
Chương 2: Mô hình chức năng vào dữ liệu - Phân tích hệ thống về chức năng - Thiết kế các chức năng - Thiết kế cơ sở dữ liệu	25%
Chương 3 : Mô hình phần mềm - Backend - Frontend	25%
- Mối liên hệ giữa backend và frontend	
Chương 4: Cài đặt ứng dụng - Môi trường cài đặt - Kết quả đạt được	25%

4. GIÁO VIÊN HƯỚNG DẪN TỪNG PHẦN

Phần	Họ và tên giáo viên hướng dẫn
Chương 1: Cơ sở lý thuyết và công nghệ phát triển website - Bài toán trang điện tử bán đồ gia dụng - Công nghệ phát triển	TS. Nguyễn Văn Thắm
Chương 2: Mô hình chức năng vào dữ liệu - Phân tích hệ thống về chức năng - Thiết kế các chức năng - Thiết kế cơ sở dữ liệu	TS. Nguyễn Văn Thắm
Chương 3 : Mô hình phần mềm - Backend - Frontend - Mối liên hệ giữa backend và frontend	TS. Nguyễn Văn Thắm
Chương 4: Cài đặt ứng dụng - Môi trường cài đặt - Kết quả đạt được	TS. Nguyễn Văn Thắm

5. NGÀY GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Ngày tháng năm 2024

Trưởng Bộ môn

Giáo viên hướng dẫn chính

Nhiệm vụ Đồ án tốt nghiệp đã được Hội đồng thi tốt nghiệp của Khoa thông qua

Ngày. . . tháng. . . năm 2024

Chủ tịch Hội đồng

Sinh viên đã hoàn thành và nộp bản Đồ án tốt nghiệp cho Hội đồng thi

Ngày... tháng... năm 2024

Sinh viên làm Đồ án tốt nghiệp

LỜI CAM ĐOAN

Tác giả xin cam đoan đây là Đồ án tốt nghiệp tốt nghiệp của bản thân tác giả. Các kết quả trong Đồ án tốt nghiệp tốt nghiệp này là trung thực, và không sao chép từ bất kỳ một nguồn nào và dưới bất kỳ hình thức nào. Việc tham khảo các nguồn tài liệu đã được thực hiện trích dẫn và ghi nguồn tài liệu tham khảo đúng quy định.

Tác giả ĐATN

Cao Bá Lợi

LỜI CẢM ƠN

Trong suốt quá trình học tập em luôn được sự quan tâm, hướng dẫn và giúp đỡ tận tình của các giảng viên trong khoa Công nghệ thông tin cùng với sự động viên giúp đỡ của bạn bè.

Lời đầu tiên em xin được bày tỏ lòng biết ơn sâu sắc đến Ban giám hiệu Trường Đại học Thủy Lợi, Ban chủ nhiệm khoa Công nghệ thông tin đã tận tình giúp đỡ em suốt thời gian học tại trường.

Đặc biệt em xin được bày tỏ lòng biết ơn chân thành sâu sắc tới thầy Nguyễn Văn Thảm đã trực tiếp giúp đỡ, hướng dẫn em hoàn thành đồ án tốt nghiệp này.

Em xin trân trọng cảm ơn.

MỤC LỤC

DANH MỤC CÁC HÌNH ẢNH

DANH MỤC BẢNG BIỂU

CHƯƠNG 1 CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ PHÁT TRIỂN WEBSITE

1.1 Bài toán xây dựng trang điện tử bán đồ gia dụng

1.1.1 Mô tả bài toán

Công nghệ thông tin ngày càng có tính ứng dụng cao và đi sâu vào đời sống con người. Sự ra đời của các trang thương mại điện tử làm cho việc mua sắm ngày càng trở nên thuận tiện và dễ dàng. Đặc biệt đối với ngành hàng gia dụng, nơi mà sản phẩm cần được trình bày chi tiết và thu hút, việc xây dựng một trang web bán đồ gia dụng trực tuyến trở thành một bài toán quan trọng và thú vị. Bài toán "Xây dựng trang điện tử bán đồ gia dụng" tập trung vào việc tạo ra một nền tảng thương mại điện tử tối ưu, phục vụ nhu cầu mua sắm trực tuyến các sản phẩm gia dụng.

Mục tiêu chính là thiết kế một trang web không chỉ thân thiện với người dùng mà còn tập trung vào việc thiết kế và phát triển một nền tảng trực tuyến hiện đại, có tính năng mạnh mẽ để hỗ trợ việc mua sắm các sản phẩm gia dụng. Trang web này cần phải tích hợp nhiều chức năng quan trọng như quản lý sản phẩm, xử lý đơn hàng, thanh toán trực tuyến, và chăm sóc khách hàng. Hệ thống thanh toán cần hỗ trợ nhiều phương thức khác nhau như thẻ tín dụng, ví điện tử và chuyển khoản ngân hàng. Tất cả các giao dịch phải được bảo mật chặt chẽ để bảo vệ thông tin khách hàng và đảm bảo tính an toàn trong các giao dịch trực tuyến. Hệ thống cũng cần có khả năng xử lý các yêu cầu hoàn tiền một cách nhanh chóng và chính xác.. Ngoài ra, giao diện người dùng cần phải hấp dẫn, dễ sử dụng, và tối ưu hóa cho cả máy tính và thiết bị di động.

Mục tiêu của bài toán là tạo ra một trang web không chỉ đáp ứng nhu cầu mua sắm trực tuyến mà còn mang lại trải nghiệm tốt nhất cho khách hàng, từ việc tìm kiếm sản phẩm, so sánh giá cả, đến thanh toán an toàn và theo dõi đơn hàng. Hơn nữa, trang web cần phải có hệ thống quản lý hiệu quả để giúp doanh nghiệp dễ dàng quản lý kho hàng, phân tích dữ liệu bán hàng, và thực hiện các chiến dịch marketing hiệu quả.

1.1.2 Yêu cầu thực tế

Để hoàn thiện đề tài: “Xây dựng trang điện tử bán đồ gia dụng”, các yêu cầu đặt ra không hề đơn giản. Dưới đây là một số yêu cầu khi thiết kế :

- Là website bán hàng, nên giao diện phải dễ nhìn, dễ để người dùng xem và tìm kiếm sản phẩm.
- Website có chức năng thanh toán trực tuyến, nên cần phải có tính bảo mật.
- Dữ liệu phải được lưu trữ trong cơ sở dữ liệu.
- Giúp người dùng dễ dàng đặt mua hàng trực tuyến.
- Giúp người quản lý thống kê được tình trạng bán hàng và doanh thu.

1.1.3 Giải pháp thực hiện

- Phương pháp nghiên cứu lý thuyết: Nghiên cứu các trang thương mại điện tử hiện có, nghiên cứu các đề tài đã thực hiện.
- Phương pháp phân tích: Phân tích mô hình, chức năng của trang thương mại điện tử.
- Phương pháp khảo sát: Khảo sát người dùng của các trang thương mại điện tử hiện nay.

1.2 Công nghệ phát triển

1.2.1 Hệ quản trị CSDL

Hệ quản trị cơ sở dữ liệu một phần mềm để tạo hoặc quản lý cơ sở dữ liệu. Hệ quản trị giúp người dùng cuối có thể tạo, bảo vệ, đọc, cập nhật và xóa dữ liệu trong cơ sở dữ liệu. Là loại nền tảng quản lý dữ liệu phổ biến nhất. Hệ quản trị đóng vai trò là giao diện giữa cơ sở dữ liệu và người dùng hoặc chương trình ứng dụng.

Hệ quản trị cơ sở dữ liệu MySQL là một trong những cơ sở dữ liệu hoàn thiện nhất và được sử dụng rộng rãi. MySQL được phát triển liên tục, cung cấp các chức năng phong phú và hữu ích. Khả năng kết nối, tốc độ và tính bảo mật khiến MySQL rất phù hợp để truy cập cơ sở dữ liệu trên internet. Có giao diện đơn giản, thân thiện với người dùng. Hỗ trợ cho nhiều hệ điều hành khác nhau. MySQL có thể mở rộng để đáp ứng nhu cầu của các dự án lớn. MySQL cung cấp một bộ hoàn chỉnh các công nghệ sao chép gốc, được tích hợp đầy đủ để có tính sẵn sàng cao và khắc phục sự cố.

Ưu điểm :

- Có sẵn bản miễn phí.
- Cung cấp nhiều chức năng.
- Dễ sử dụng, bảo mật và độ tin cậy cao.
- Có khả năng mở rộng, hiệu suất và tính sẵn sàng cao.

Nhược điểm :

- Cơ chế phục hồi và sao lưu kém linh hoạt hơn so với các hệ quản trị cơ sở dữ liệu khác.
- Hỗ trợ kém cho một số tính năng NoSQL.
- Tính toàn vẹn dữ liệu: MySQL cho phép chèn dữ liệu vi phạm một số ràng buộc nếu không bật chế độ kiểm tra ràng buộc.

Trong đề tài này em chọn sử dụng Hệ quản trị MySQL bởi vì MySQL có khả năng quản lý dữ liệu hiệu quả, đặc biệt là trong các ứng dụng web. Với cấu trúc quan hệ mạnh mẽ và ngôn ngữ truy vấn SQL, MySQL dễ dàng truy xuất và quản lý dữ liệu một cách nhanh chóng và chính xác. Khả năng mở rộng của MySQL giúp nó có thể xử lý từ các ứng dụng nhỏ đến các hệ thống lớn với khối lượng dữ liệu khổng lồ. MySQL cung cấp một loạt các tính năng bảo mật để bảo vệ dữ liệu, bao gồm mã hóa và quản lý quyền truy cập người dùng. MySQL cung cấp giải pháp quản lý cơ sở dữ liệu mạnh mẽ, tiết kiệm chi phí và dễ triển khai cho nhiều loại dự án khác nhau.

1.2.2 Công nghệ web

1.2.2.1 Backend

a) Ngôn ngữ lập trình C#

Em chọn sử dụng ngôn ngữ C# để thực hiện đồ án tốt nghiệp này. Vì C# là ngôn ngữ lập trình mạnh mẽ, linh hoạt. Có cộng đồng rộng lớn và tích cực, cung cấp nhiều tài liệu, hướng dẫn, thư viện và công cụ hỗ trợ.

C# là ngôn ngữ lập trình hướng đối tượng bậc cao, linh hoạt, chạy trên nền tảng nguồn mở cũng như kiến trúc của Microsoft Windows .NET. C# đặc biệt thuận lợi cho việc

xây dựng các loại ứng dụng trên web khác nhau nhờ khả năng trong việc tạo môi trường tương tác cho người dùng. C# được phát triển bởi Microsoft và được ra mắt vào năm 2000.

Ưu điểm của C# :

- Là ngôn ngữ lập trình đa nền tảng.
- Cung cấp các tính năng quản lý bộ nhớ tự động, khả năng xử lý ngoại lệ mạnh mẽ.
- Cộng đồng lớn, có nhiều tài liệu hướng dẫn và công cụ hỗ trợ.
- cú pháp rõ ràng, dễ hiểu và được thiết kế để giảm bớt phức tạp khi lập trình.
- Được hỗ trợ liên tục và cập nhật thường xuyên.
- Cho phép lập trình bất đồng bộ.

Nhược điểm :

- Phụ thuộc vào nền tảng .NET.
- Yêu cầu tài nguyên hệ thống tương đối cao, có thể gây ảnh hưởng đến hiệu suất và khả năng mở rộng trong môi trường có tài nguyên hạn chế.
- Có thể thiếu một số tính năng và hỗ trợ cho lập trình hướng đối tượng tinh vi.

b) .NET, ASP.NET và ASP.NET Web API

.NET là nền tảng mã nguồn mở, đa nền tảng và miễn phí. Có thể chạy các chương trình được viết bằng nhiều ngôn ngữ, trong đó C# là phổ biến nhất. Được thiết kế để mang lại năng suất, hiệu suất, tính bảo mật và độ tin cậy. Cung cấp khả năng quản lý bộ nhớ tự động. Bao gồm một bộ thư viện lớn có chức năng rộng và đã được tối ưu hóa để đạt hiệu suất trên nhiều hệ điều hành và kiến trúc chip.

ASP.NET là một framework phát triển web phổ biến để xây dựng các ứng dụng web trên nền tảng .NET. ASP.NET cung cấp cho các nhà phát triển các công cụ và hệ sinh thái để xây dựng các ứng dụng web hiện đại, bảo mật và có khả năng mở rộng. ASP.NET cung cấp ba framework để tạo ứng dụng web: Web Forms, ASP.NET MVC và ASP.NET Web Pages.

ASP.NET Web API là một phần mở rộng của ASP.NET MVC. ASP.NET Web API Là một framework giúp dễ dàng xây dựng các dịch vụ HTTP tiếp cận nhiều đối tượng khách hàng, bao gồm cả trình duyệt và thiết bị di động. Là một platform lý tưởng để xây dựng các ứng dụng RESTful trên .NET.

Đồ án này em sử dụng ASP.NET Web API vì ASP.NET Web API cung cấp một cách tiện lợi để xây dựng các API cho ứng dụng, hỗ trợ các phương thức HTTP (GET, POST, PUT, DELETE và PATCH) để tương tác với dữ liệu một cách linh hoạt và hiệu quả. ASP.NET Web API đi kèm với các tính năng bảo mật mạnh mẽ như xác thực người dùng, kiểm soát truy cập và phân quyền, đảm bảo an toàn cho dữ liệu hệ thống.

c) Cấu trúc của ASP.NET Web API

Cấu trúc source code và luồng đi của ứng dụng ASP.NET Web API dựa trên mô hình MVC. Đây là mô hình rất phổ biến của các dự án web.

Các thành phần của mô hình gồm có các thành phần quan trọng sau:

- Controller: Chịu trách nhiệm xử lý các yêu cầu HTTP và thực hiện các hành động nhất định trên tài nguyên của API.
- Service: chứa các lớp được sử dụng để xử lý logic kinh doanh hoặc gọi các phương thức từ Repository.
- Repository: là nơi trực tiếp thao tác với dữ liệu. Giúp tách biệt logic truy cập dữ liệu khỏi logic nghiệp vụ.
- Model : chứa các định nghĩa lớp mô hình biểu diễn các đối tượng dữ liệu của ứng dụng.
- DTO: sử dụng để chuyển dữ liệu giữa các lớp, các thành phần của ứng dụng hoặc giữa các hệ thống khác nhau.
- Data: là nơi chứa lớp dữ liệu dùng để kết nối đến cơ sở dữ liệu.

(1) Controller

Ý nghĩa :

Controller trong ứng dụng ASP.NET Web API là nơi tiếp nhận và xử lý các yêu cầu HTTP và trả về dữ liệu cho client.

Cú pháp:

- Controller là một class C# được đánh dấu [ApiController]. Có thể đi kèm theo sau đó là route của API theo cấu hình tùy chọn. Controller sẽ được kế thừa từ ControllerBase trong ASP.NET.
- ApiController có thể trả về một String hoặc data dưới dạng JSON (nếu được chỉ định). Trong ApiController cần định rõ phương thức HTTP được sử dụng ứng với mỗi hành động.

```
[ApiController]
[Route("api/[controller]")]
0 references
public class ProductController(IProductService productService) : ControllerBase
{
    1 reference
    private readonly IProductService _productService = productService;

    [HttpGet]
    0 references
    public async Task<IActionResult> GetAll()
    {
        var products = await _productService.GetAll();
        return Ok(products);
    }
}
```

Hình 1.1 Ví dụ Controller trong ASP.NET Web API

(2) Service

Ý nghĩa:

Service là nơi xử lý logic kinh doanh hoặc gọi tới các phương thức trong Repository.

Cú pháp:

- Service trong ASP.NET Web API là một lớp chứa các phương thức tương gọi tới các phương thức có trong Repository. Thông thường, Service sẽ được sử dụng cùng với một interface. Sử dụng interface sẽ giúp tăng tính linh hoạt và dễ dàng thay đổi cũng như hỗ trợ việc Dependency Injection.

```

public interface ITagService
{
    2 references
    Task<IEnumerable<Tag>> GetAll();
    2 references
    Task<Tag?> GetById(int id);
    2 references
    Task Create(Tag tag);
    2 references
    Task Update(Tag tag);
    2 references
    Task Delete(int id);
    1 reference
    Task<Tag?> GetByName(string tagName);
    3 references
    Task<int> GetOrCreateTagId(string tagName);
}

```

Hình 1.2 Ví dụ interface service trong ASP.NET Web API

```

public class TagService(ITagRepository tagRepository) : ITagService
{
    1 reference
    private readonly ITagRepository _tagRepository = tagRepository;
    2 references
    public async Task<IEnumerable<Tag>> GetAll()
    {
        return await _tagRepository.GetAll();
    }
}

```

Hình 1.3 Ví dụ service trong ASP.NET Web API

(3) Repository

Ý nghĩa :

Chứa các phương thức dùng để thao tác với dữ liệu.

Cú pháp:

Repository trong ASP.NET Web API có cấu trúc giống với Service. Thông thường cũng được sử dụng cùng với interface.

```

public interface ITagRepository
{
    2 references
    Task<IEnumerable<Tag>> GetAll();
    2 references
    Task<Tag?> GetById(int id);
    3 references
    Task Create(Tag tag);
    2 references
    Task Update(Tag tag);
    2 references
    Task Delete(int id);
    4 references
    Task<Tag?> GetByName(string tagName);
    2 references
    Task<int> GetOrCreateTagId(string tagName);
}

```

Hình 1.4 Ví dụ interface Repository

```

public class TagRepository(DataContext context) : ITagRepository
{
    1 reference
    private readonly DataContext _context = context;

    2 references
    public async Task<IEnumerable<Tag>> GetAll()
    {
        using var connection = _context.CreateConnection();
        var sql = @"
            SELECT * FROM Tags
        ";
        return await connection.QueryAsync<Tag>(sql);
    }
}

```

Hình 1.5 Ví dụ Repository

(4) Model

Ý nghĩa:

Model là các lớp chứa định nghĩa về các đối tượng dữ liệu của ứng dụng ánh xạ đến cơ sở dữ liệu.

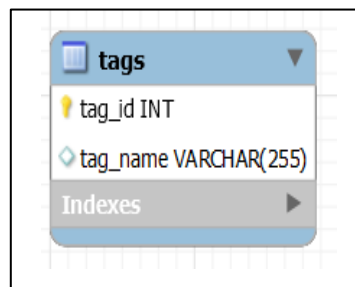
Cú pháp:

Là một lớp của C# có chứa đối tượng ánh xạ đến các đối tượng có trong cơ sở dữ liệu.

Có chứa accessors get và set.

```
public class Tag{  
    [Required]  
    2 references  
    public int tag_id {get;set;}  
    [Required]  
    1 reference  
    public string? tag_name {get;set;}  
}
```

Hình 1.6 Ví dụ model trong ASP.NET Web API



Hình 1.7 Ví dụ table trong cơ sở dữ liệu

(5) DTO

Ý nghĩa:

DTO là các lớp chứa định nghĩa về các đối tượng dữ liệu của ứng dụng. Được sử dụng để chuyển dữ liệu giữa các thành phần của ứng dụng. Mục đích nhằm giảm lượng thông tin không cần thiết phải chuyển đi, tăng tốc độ và tính bảo mật cho dự án.

Cú pháp:

Là một lớp của C#, có cú pháp giống với model.

```

public class UserDataDTO
{
    1 reference
    public string? Id { get; set; }
    1 reference
    public string? Username { get; set; }
    1 reference
    public List<string>? Roles { get; set; }
}

```

Hình 1.8 Ví dụ DTO trong ASP.NET Web API

(6) Data

Ý nghĩa:

Data là lớp quản lý kết nối tới cơ sở dữ liệu thông qua các tùy chọn cung cấp.

Cú pháp:

Là một lớp của ASP.NET, kế thừa từ DbContext hoặc IdentityDbContext (nếu có sử dụng Identity).

```

public class AppDbContext : IdentityDbContext<User>{
    0 references
    public AppDbContext(DbContextOptions<AppDbContext> options)
    : base(options)
    {
    }
}

```

Hình 1.8 Ví dụ Data trong ASP.NET Web API

1.2.2.2 Frontend

a) Framework VueJs

VueJs là một framework VueJs là một JavaScript framework miễn phí để xây dựng giao diện người dùng động. Được duy trì bởi các nhà phát triển từ nhiều cộng đồng khác nhau, bao gồm cả chính người tạo ra là Evan You và được cập nhật liên tục. VueJs tạo điều kiện thuận lợi cho việc tái sử dụng lại các thành phần khác nhau của trang web.

VueJs có thể tạo ra các thành phần (component). Trong mỗi thành phần có thể bao gồm HTML, CSS và JavaScript. Các thành phần được hợp lại với nhau để tạo nên giao diện người dùng. Có Vue Router cho việc quản lý định tuyến cho ứng dụng, cho phép dễ dàng điều hướng giữa các trang của ứng dụng mà không cần tải lại toàn bộ trang.

Đây là một component đơn giản trong VueJs

```
<template>  
  <h1>Đây là component</h1>  
</template>  
<script>  
</script>  
<style>  
</style>
```

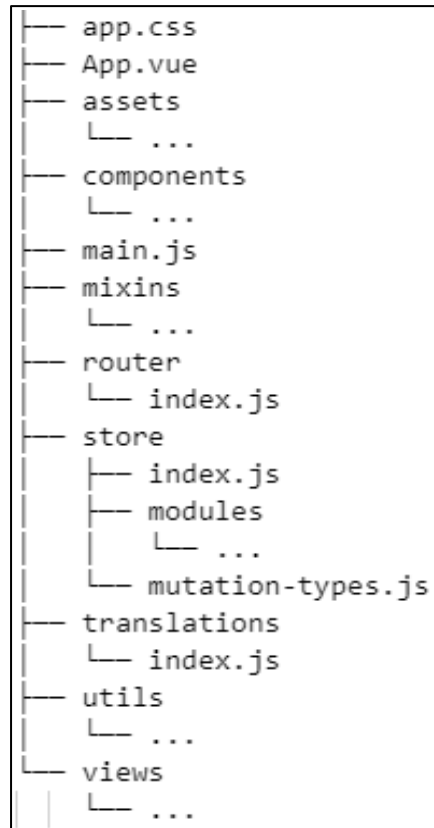
Hình 1.8 Ví dụ component trong VueJs

b) Tạo một VueJs App với Vue CLI

Vue CLI công cụ dòng lệnh mạnh mẽ dùng để tạo và quản lý các dự án VueJs. Vue CLI tích hợp sẵn các lệnh để tự động hóa các công việc phát triển.

```
npm create vue@latest  
cd vue-project  
npm install  
npm run dev
```

Hình 1.16 Vue CLI



Hình 1.17 Cấu trúc ứng dụng Vue khi dùng Vue CLI

1.2.2.3 RESTful API – Cách giao tiếp giữa Client và Server

API hay Application Programming Interface xác định các quy tắc phải tuân theo để giao tiếp với các hệ thống phần mềm khác. Các nhà phát triển hiển thị hoặc tạo API để các ứng dụng khác có thể giao tiếp với ứng dụng của họ theo chương trình. Có thể coi API như một cổng kết nối giữa máy khách và tài nguyên trên web.

REST (REpresentational State Transfer) là một kiến trúc phần mềm áp đặt các điều kiện về cách hoạt động của API. Có thể sử dụng kiến trúc REST để hỗ trợ giao tiếp hiệu suất cao và đáng tin cậy trên quy mô lớn. Có thể dễ dàng triển khai và sửa đổi, mang lại khả năng hiển thị và tính di động đa nền tảng.

RESTful API là giao diện mà hai hệ thống máy tính sử dụng để trao đổi thông tin một cách an toàn qua internet.

Chức năng API RESTful giống như duyệt internet. Máy khách liên hệ với máy chủ bằng cách sử dụng API khi yêu cầu tài nguyên. Máy chủ sẽ xác thực ứng dụng khách

đó, xác nhận quyền thực hiện của máy khách, xử lý yêu cầu nội bộ sau đó trả về phản hồi cho máy khách. Đây là các phương thức HTTP thường được sử dụng trong API RESTful:

- GET (SELECT): truy cập tài nguyên trên máy chủ.
- POST (CREATE): tạo tài nguyên trên máy chủ.
- PUT (UPDATE): cập nhật tài nguyên hiện có trên máy chủ.
- DELETE (DELETE): xóa tài nguyên khỏi máy chủ.

Dữ liệu sẽ được API sẽ trả về gồm các thành phần chính sau:

- Trạng thái: chứa mã trạng thái gồm ba chữ số cho biết yêu cầu thành công hay thất bại.
- Nội dung: chứa tài nguyên trả về. Máy chủ chọn định dạng trình bày phù hợp dựa trên nội dung trong tiêu đề yêu cầu, có thể là XML hoặc JSON.
- Tiêu đề: cung cấp thêm ngữ cảnh về phản hồi bao gồm các thông tin như máy chủ, mã hóa, ngày tháng và loại nội dung.

CHƯƠNG 2 MÔ HÌNH CHỨC NĂNG VÀ DỮ LIỆU

2.1 Phân tích hệ thống về chức năng

2.1.1 Quy tắc nghiệp vụ hệ thống

2.1.1.1 Nhóm nghiệp vụ hệ thống

- Người dùng có thể đăng nhập, đăng xuất khỏi hệ thống.
- Các quyền cơ bản có trong hệ thống gồm quyền của người quản lý và quyền của người dùng thông thường.
- Quản lý hệ thống đăng nhập bằng tài khoản quản trị có sẵn.
- Người dùng thông thường sẽ đăng nhập bằng tài khoản đã đăng ký.
- Hệ thống cần phân biệt vai trò của tài khoản để cấp quyền truy cập tương ứng.

2.1.1.2 Nhóm nghiệp vụ quản lý

- Quản lý sản phẩm: quản lý có thể thêm, xóa, sửa thông tin của sản phẩm.
- Quản lý đơn hàng: quản lý có thể xác nhận, cập nhật tình trạng đơn hàng.

- Quản lý thanh toán: có hỗ trợ thanh toán trực tuyến, đảm bảo các giao dịch được thực hiện an toàn và chính xác.
- Quản lý báo cáo: theo dõi doanh thu, lợi nhuận và chi phí.
- Quản lý thông tin cá nhân: người dùng có thể xem thông tin, tạo hoặc sửa đổi thông tin cá nhân.
- Người dùng có xem đơn đặt hàng.
- Người dùng có thể chọn sản phẩm để tiến hành mua hàng.
- Quản lý có thể xóa người dùng khỏi hệ thống nếu cần thiết.

2.1.2 Xác định và mô tả các tác nhân

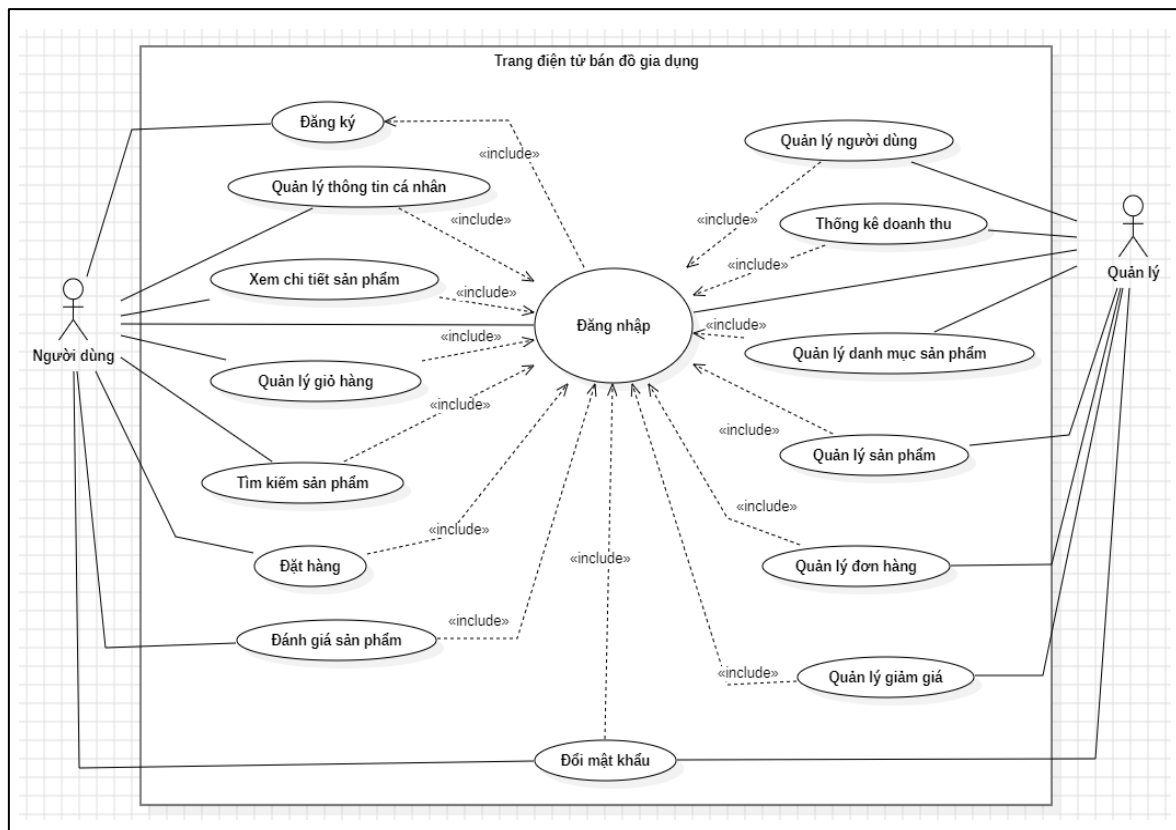
Bảng 2.1 Các tác nhân của hệ thống

Tác nhân	Mô tả
Quản lý	Chủ sở hữu của trang thương mại điện tử
Người dùng	Các khách hàng có nhu cầu xem và mua sắm sản phẩm

2.1.3 Xác định và mô tả các ca sử dụng

- Đăng nhập : Quản lý, người dùng đăng nhập hệ thống.
- Đăng ký: Người dùng đăng ký tài khoản.
- Quản lý thông tin cá nhân: Người dùng thêm, xóa, sửa thông tin cá nhân.
- Đặt hàng: Người dùng đã đăng nhập có thể đặt hàng.
- Xem chi tiết sản phẩm: Người dùng có thể xem chi tiết sản phẩm khi nhấn vào sản phẩm đó
- Quản lý đơn hàng : Quản lý xác nhận, cập nhật tình trạng đơn hàng.
- Quản lý sản phẩm: Quản lý thêm, xóa, sửa thông tin của sản phẩm.
- Quản lý tài khoản : Quản lý cấp tài khoản cho người dùng, xóa tài khoản.

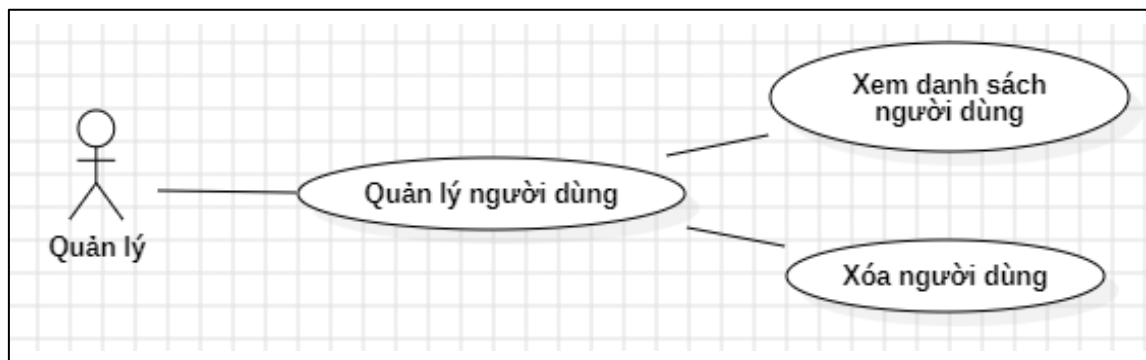
2.1.4 Xây dựng biểu đồ ca sử dụng



Hình 2.1 Biểu đồ usecase tổng quát

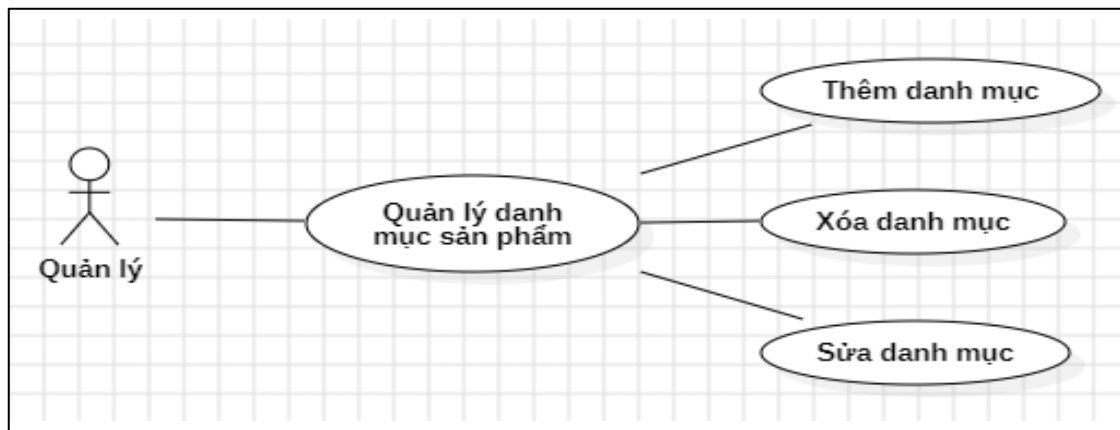
2.1.4.2 Biểu đồ phân rã usecase của tác nhân “Người quản trị”

a) Chức năng quản lý người dùng



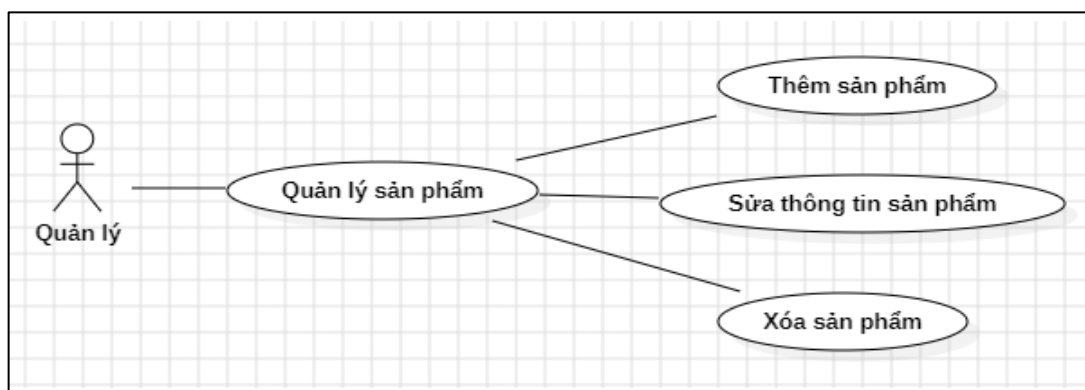
Hình 2.2 Biểu đồ phân rã usecase chức năng quản lý người dùng

b) Chức năng quản lý danh mục sản phẩm



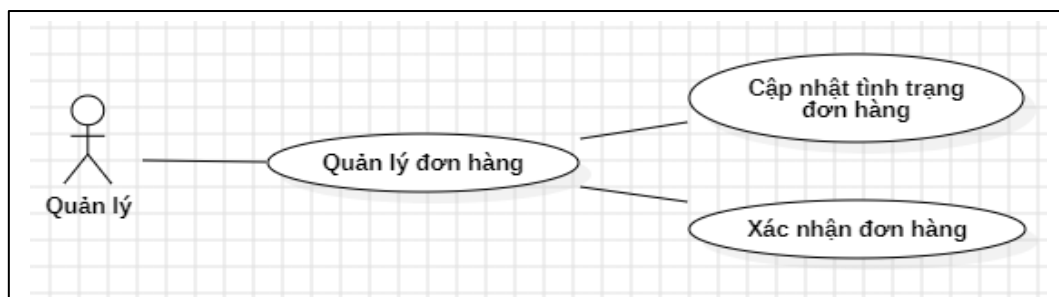
Hình 2.3 Biểu đồ phân rã usecase chức năng quản lý danh mục sản phẩm

c) Chức năng quản lý sản phẩm



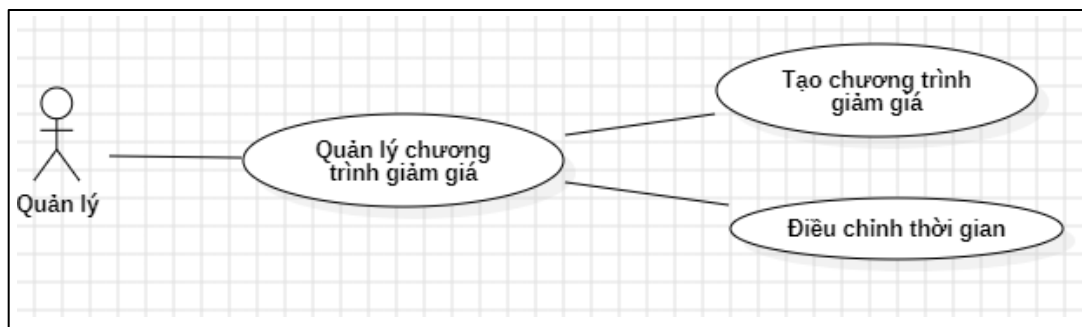
Hình 2.4 Biểu đồ phân rã usecase chức năng quản lý sản phẩm

d) Chức năng quản lý đơn hàng



Hình 2.5 Biểu đồ phân rã usecase chức năng quản lý đơn hàng

e) Chức năng thiết quản lý chương trình giảm giá



Hình 2.6 Biểu đồ phân rã usecase chức năng quản lý chương trình giảm giá

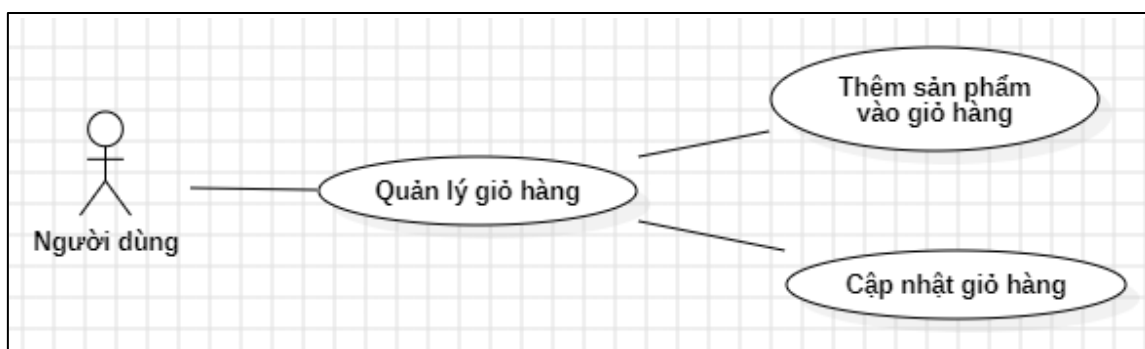
2.1.4.3 Biểu đồ phân rã usecase của tác nhân “Người dùng”

a) Chức năng quản lý thông tin cá nhân



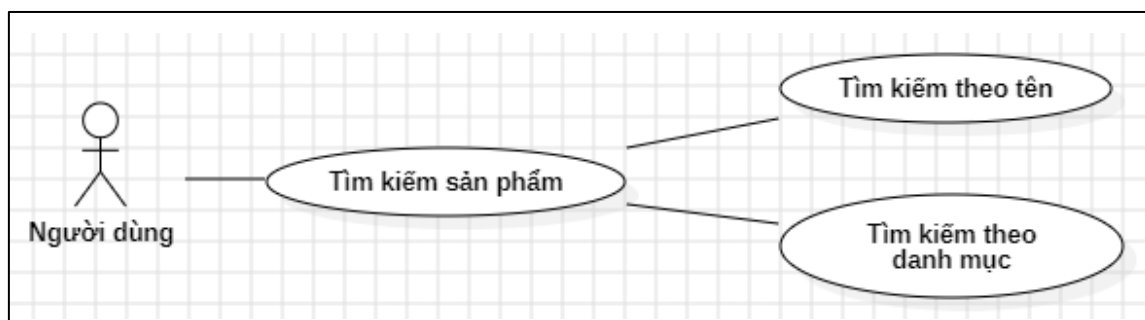
Hình 2.7 Biểu đồ phân rã usecase chức năng quản lý thông tin cá nhân

b) Chức năng tra cứu biểu mẫu



Hình 2.8 Biểu đồ phân rã usecase chức năng quản lý giỏ hàng

c) Chức năng tìm kiếm sản phẩm



Hình 2.9 Biểu đồ phân ra usecase chức năng tìm kiếm sản phẩm

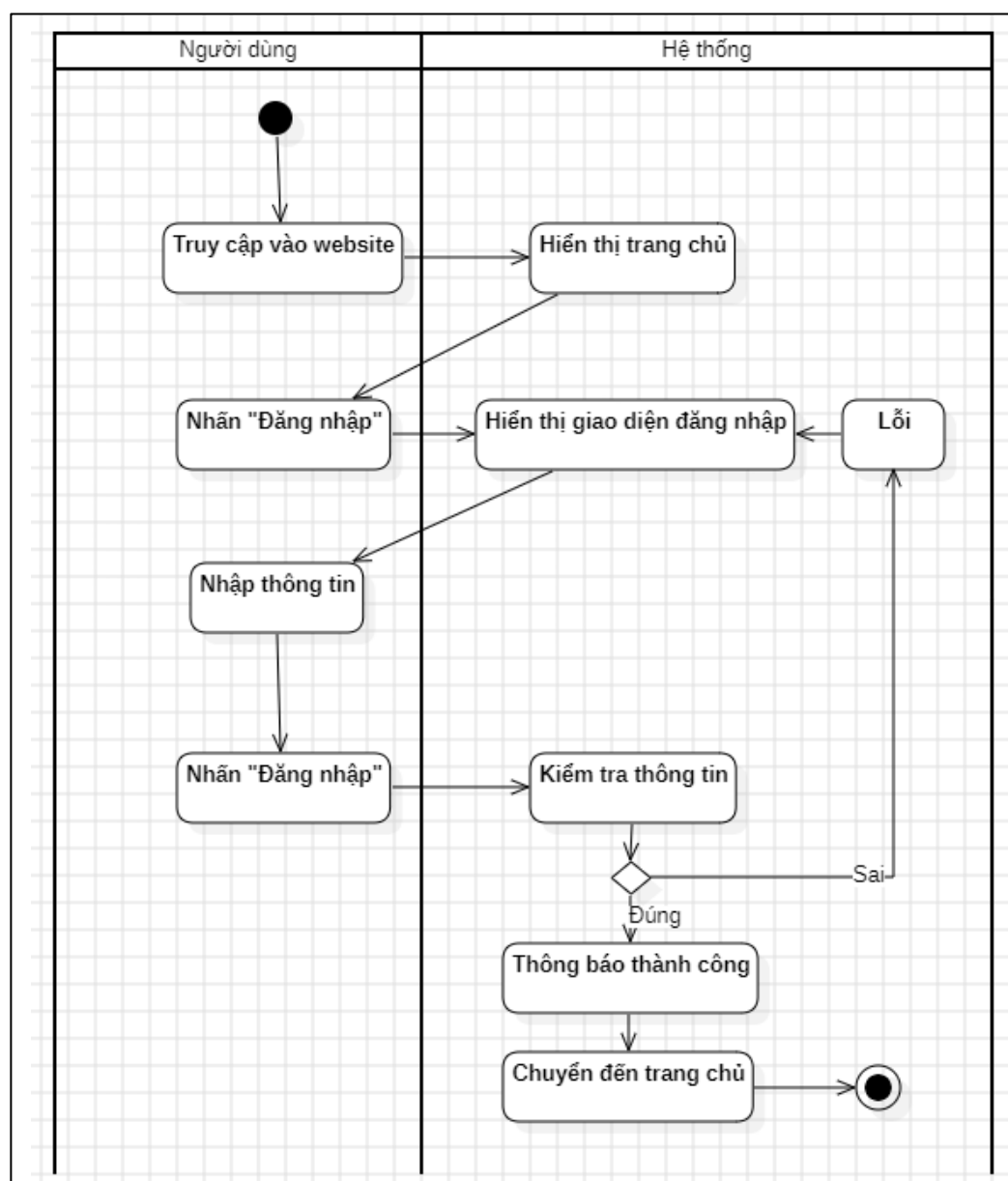
2.1.5 Xây dựng kịch bản

2.1.5.1 Kịch bản đăng nhập tài khoản

Bảng 2.2 Bảng kịch bản đăng nhập tài khoản

Tên usecase	Đăng nhập		
Tác nhân	Người sử dụng		
Mô tả	Tại giao diện trang chủ, người dùng nhấn vào nút “Đăng nhập”		
Điều kiện tiên quyết	Người dùng đã có tài khoản		
Luồng sự kiện		Tác nhân	Hệ thống phản hồi
	1	Truy cập vào website	Hiển thị trang chủ
	2	Nhấn “Đăng nhập”	Hiển thị giao diện đăng nhập
	3	Nhập thông tin	Thông tin bắt buộc: Tên tài khoản và mật khẩu
	4	Nhấn “Đăng nhập”	Hệ thống kiểm tra thông tin, định dạng: - Nếu chính xác chuyển sang bước 5 - Nếu không chính xác: + Tên tài khoản bỏ trống (Luồng ngoại lệ EF1) + Tên tài không tồn tại (Luồng ngoại lệ EF2) + Mật khẩu bỏ trống (Luồng ngoại lệ EF3) + Mật khẩu không chính xác (Luồng ngoại lệ EF4)

	5		Thông báo đăng nhập thành công và chuyển đến trang của người dùng.
Luồng ngoại lệ	EF1		Hiển thị thông báo: Tên tài khoản không được để trống
	EF2		Hiển thị thông báo: Tài khoản không tồn tại
	EF3		Hiển thị thông báo: Mật khẩu không được để trống
	EF4		Hiển thị thông báo: Mật khẩu không chính xác



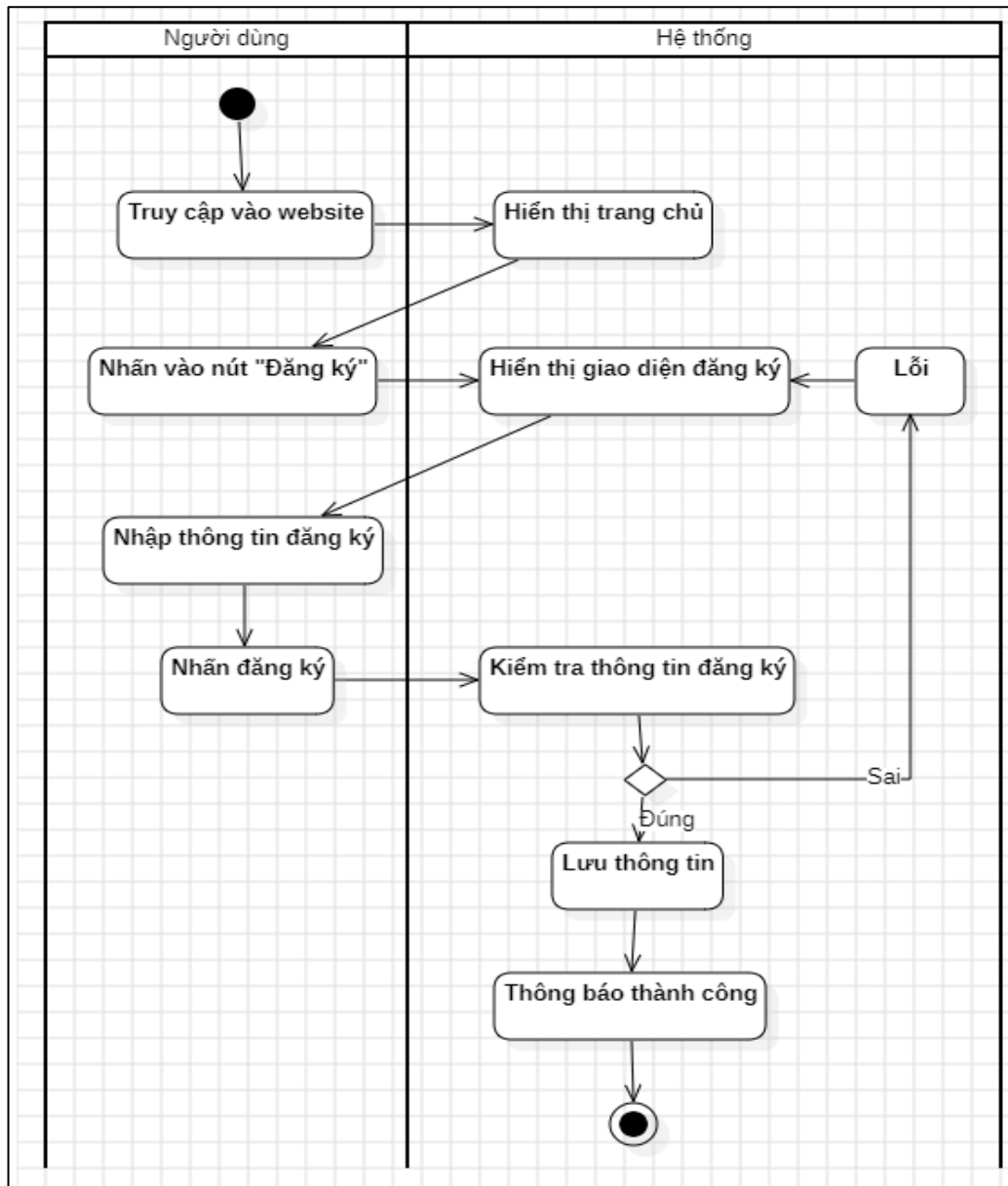
Hình 2.10 Biểu đồ hoạt động chức năng đăng nhập

2.1.5.2 Kịch bản đăng ký tài khoản

Bảng 2.3 Bảng kịch bản đăng ký tài khoản

Tên use case	Đăng ký		
Tác nhân	Người sử dụng		
Mô tả	Tại giao diện trang chủ, người dùng nhấn vào nút “Đăng ký”		
Điều kiện tiên quyết	Người dùng chưa có tài khoản		
Luồng sự kiện		Tác nhân	Hệ thống phản hồi
	1	Truy cập vào website	Hiển thị trang chủ
	2	Nhấn “Đăng ký”	Hiển thị giao diện đăng ký
	3	Nhập thông tin	Thông tin bắt buộc: Tên tài khoản, mật khẩu và email
	4	Nhấn “Đăng ký”	Hệ thống kiểm tra thông tin, định dạng: - Nếu chính xác chuyển sang bước 5 - Nếu không chính xác: + Tên tài khoản bỏ trống (Luồng ngoại lệ EF1) + Tên tài khoản ít hơn 6 ký tự (Luồng ngoại lệ EF2) + Mật khẩu bỏ trống (Luồng ngoại lệ EF3) + Mật khẩu ít hơn 8 ký tự, có chứa khoảng trắng, không chứa tối thiểu 1 ký tự đặc biệt, 1 chữ cái, 1 chữ số (Luồng ngoại lệ EF4) + Email không đúng định dạng (Luồng ngoại lệ EF5) + Email đã tồn tại (Luồng ngoại lệ EF6)
	5		Lưu trữ thông tin với vai trò là người dùng thông thường và đưa thông báo: Đăng ký thành công
Luồng ngoại lệ	EF1		Hiển thị thông báo: Tên tài khoản không được để trống
	EF2		Hiển thị thông báo: Tên tài khoản không được ít hơn 6 ký tự
	EF3		Hiển thị thông báo: Mật khẩu không được để trống
	EF4		Hiển thị thông báo: Mật khẩu không ít hơn 8 ký tự, không chứa khoảng trắng. Chứa tối thiểu 1 ký tự đặc biệt, 1 chữ cái và 1 chữ số

	EF5		Hiển thị thông báo: Email không chính xác
	EF6		Hiển thị thông báo: Email đã tồn tại



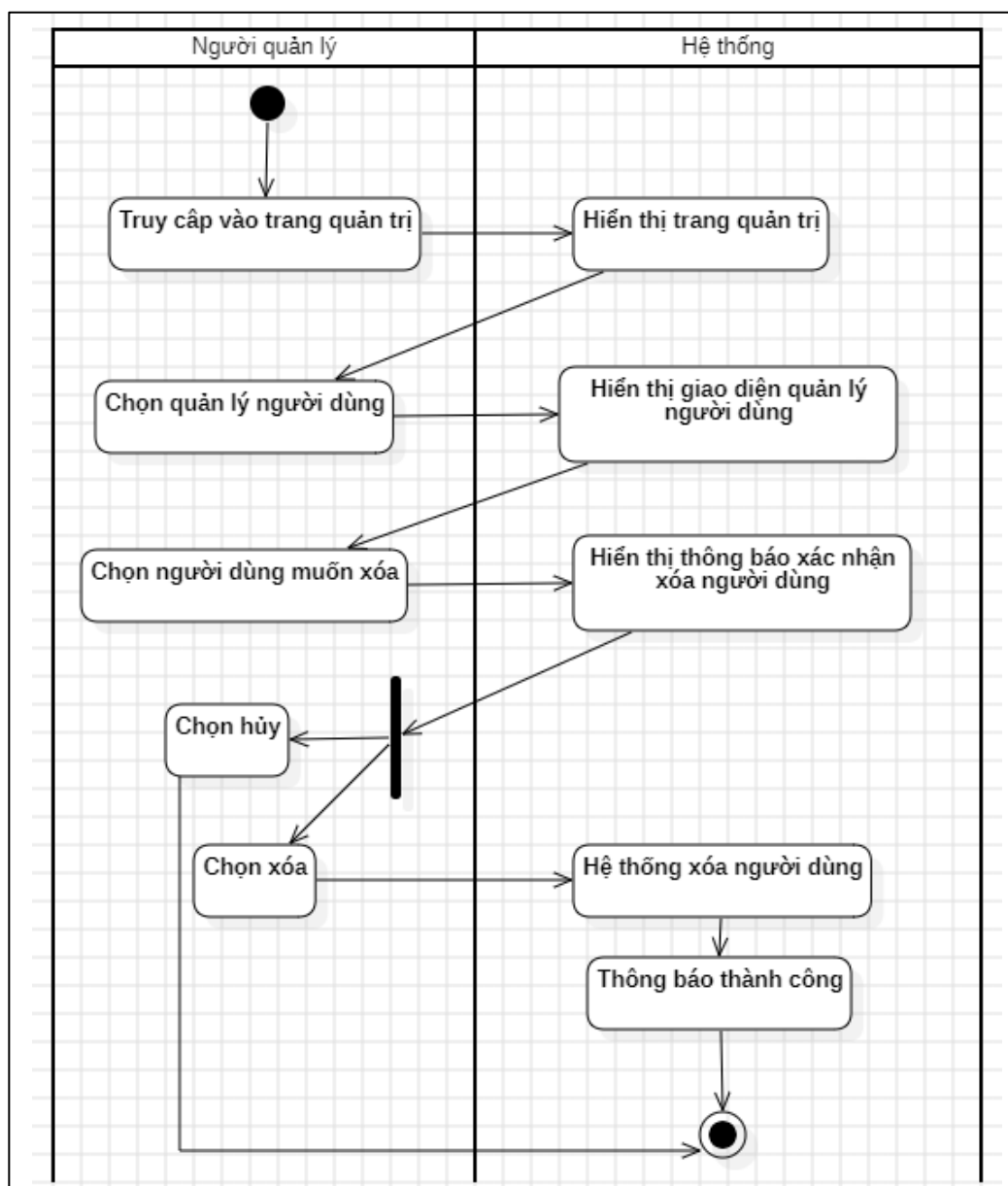
Hình 2.11 Biểu đồ hoạt động chức năng đăng ký tài khoản

2.1.5.3 Kịch bản chức năng quản lý người dùng

Bảng 2.4 Bảng kịch bản quản lý người dùng

Tên use case	Quản lý người dùng
Tác nhân	Người quản lý
Mô tả	Tại giao diện trang quản trị, chọn “Quản lý người dùng”
Điều kiện tiên quyết	Người quản lý đang đăng nhập

Luồng sự kiện		Tác nhân	Hệ thống phản hồi
	1	Truy cập vào trang quản trị	Hiển thị trang quản trị
	2	Nhấn “Quản lý người dùng”	Hiển thị giao diện quản lý người dùng
	3	Chọn người dùng muốn xóa	Thông báo xác nhận muốn xóa người dùng (Luồng ngoại lệ EF1)
	4	Nhấn “Xóa”	Hệ thống xóa người dùng khỏi hệ thống và hiển thị thông báo thành công
Luồng ngoại lệ	EF1	Chọn “Hủy”	Hệ thống không xóa người dùng và trở lại trang quản lý người dùng

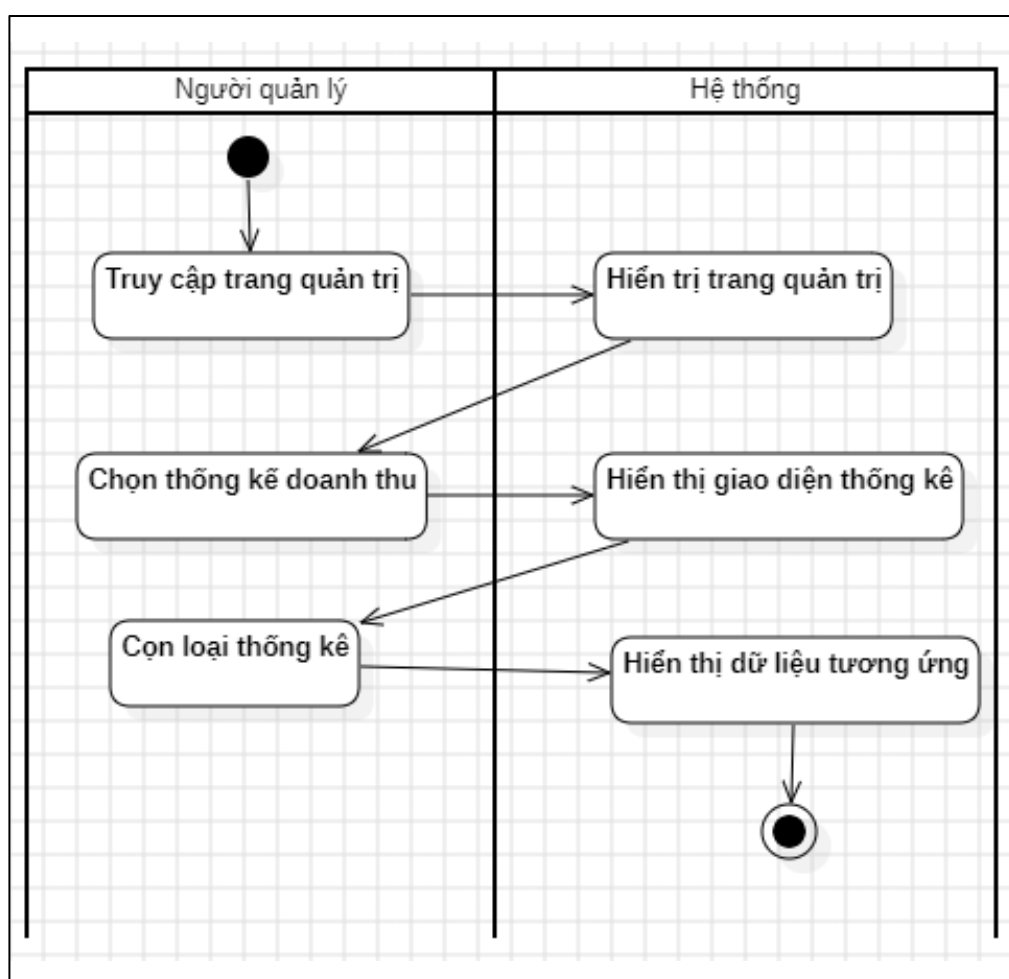


Hình 2.12 Biểu đồ hoạt động chức năng quản lý tài khoản

2.1.5.4 Kịch bản chức năng thống kê doanh thu

Bảng 2.5 Bảng kịch bản chức năng thống kê doanh thu

Tên use case	Quản lý người dùng		
Tác nhân	Người quản lý		
Mô tả	Tại giao diện trang quản trị, chọn “Thống kê doanh thu”		
Điều kiện tiên quyết	Người quản lý đang đăng nhập		
Luồng sự kiện		Tác nhân	Hệ thống phản hồi
	1	Truy cập vào trang quản trị	Hiển thị trang quản trị
	2	Nhấn “Thống kê doanh thu”	Hiển thị giao diện thống kê doanh thu
	3	Chọn loại thống kê	Hiển thị dữ liệu tương ứng. (Luồng ngoại lệ EF1)
Luồng ngoại lệ	EF1		Trả về thông báo nếu không có dữ liệu phù hợp.



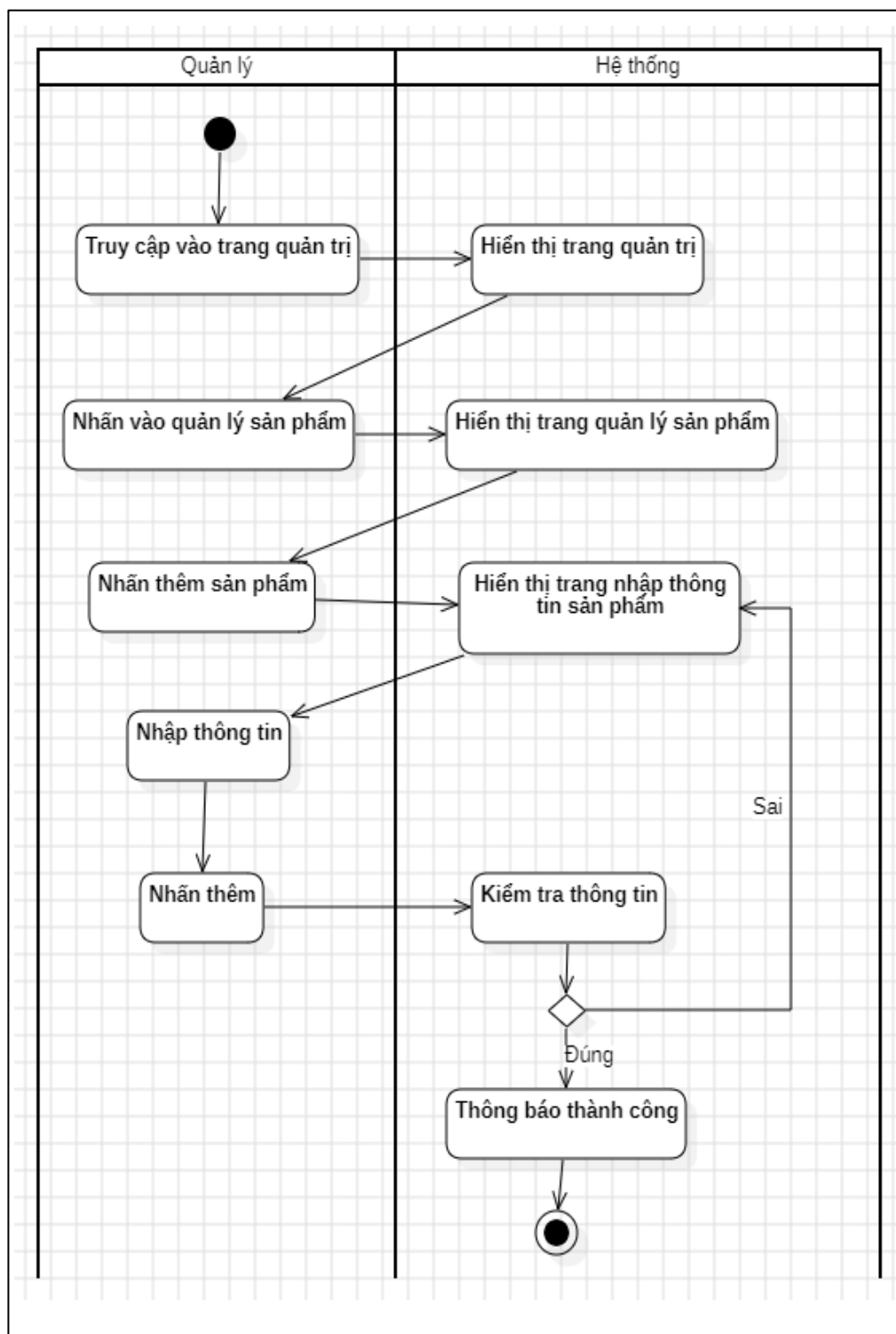
Hình 2.13 Biểu đồ hoạt động chức năng quản lý cơ quan

2.1.5.5 Kịch bản chức năng thêm sản phẩm

Bảng 2.6 Bảng kịch bản quản lý biểu mẫu

Tên use case	Thêm sản phẩm		
Tác nhân	Người quản lý		
Mô tả	Tại giao diện quản lý sản phẩm, chọn thêm sản phẩm		
Điều kiện tiên quyết	Người quản lý đã đăng nhập		
Luồng sự kiện		Tác nhân	Hệ thống phản hồi
	1	Truy cập vào trang quản trị	Hiển thị trang quản trị
	2	Nhấn vào quản lý sản phẩm	Chuyển đến trang quản lý sản phẩm
	3	Nhấn thêm sản phẩm	Hiển thị trang thêm sản phẩm với các trường: tên sản phẩm, danh mục, tag, hình ảnh, giá bán, số lượng, mô tả
	4	Nhập thông tin	
	5	Nhấn thêm	Hệ thống kiểm tra thông tin và định dạng: - Nếu chính xác chuyển sang bước 6 - Ngược lại: + Tên sản phẩm có chứa ký tự đặc biệt (Luồng ngoại lệ EF1) + Danh mục chứa ký tự đặc biệt và hoặc có chứa chữ số (Luồng ngoại lệ EF2) + Nhãn chứa ký tự đặc biệt và hoặc có chứa chữ số (Luồng ngoại lệ EF3) + Giá bán có chứa ký tự đặc biệt và hoặc có chứa chữ cái (Luồng ngoại lệ EF4) + Số lượng có chứa ký tự đặc biệt và hoặc có chứa chữ số, không phải là số nguyên (Luồng ngoại lệ EF5) + Mô tả có chứa ký tự đặc biệt (Luồng ngoại lệ EF6)
	6		Thông báo thêm sản phẩm thành công và chuyển đến trang quản lý sản phẩm
Luồng ngoại lệ	EF1		Hiển thị thông báo: Tên sản phẩm không chứa ký tự đặc biệt
	EF2		Hiển thị thông báo: Tên danh mục chỉ chứa chữ cái
	EF3		Hiển thị thông báo: Nhãn chỉ chứa chữ cái

	EF4		Hiển thị thông báo: Giá bán là kiểu số
	EF5		Hiển thị thông báo: Số lượng là kiểu số nguyên
	EF6		Hiển thị thông báo: Mô tả không chứa ký tự đặc biệt



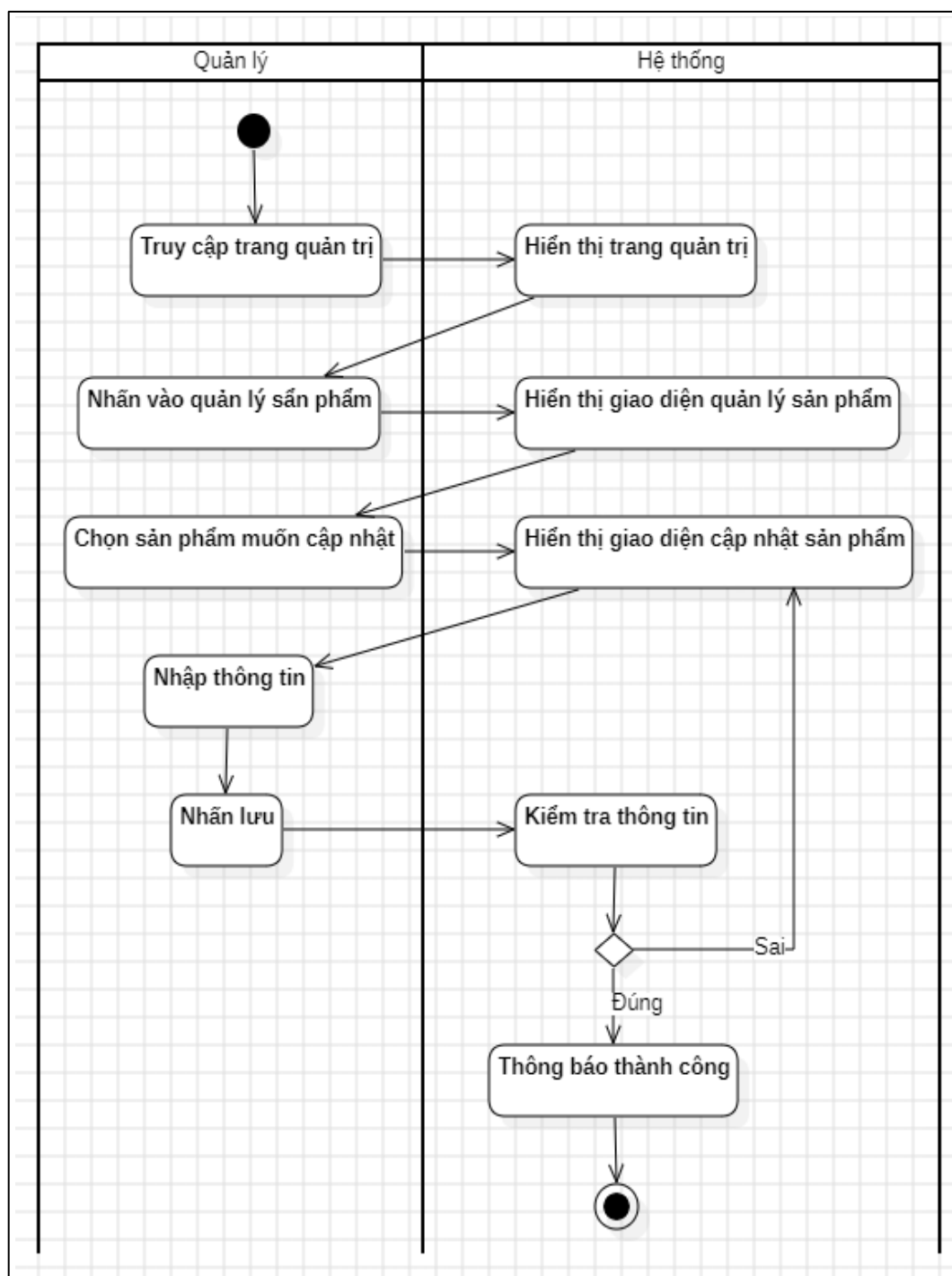
Hình 2.14 Biểu đồ hoạt động chức năng thêm sản phẩm

2.1.5.6 Kịch bản chức năng cập nhật sản phẩm

Bảng 2.7 Bảng kịch bản chức năng cập nhật sản phẩm

Tên use case	Cập nhật sản phẩm		
Tác nhân	Người quản trị		
Mô tả	Tại giao diện quản lý sản phẩm, chọn sản phẩm muốn cập nhật và chọn sửa		
Điều kiện tiên quyết	Người quản lý đã đăng nhập		
Luồng sự kiện		Tác nhân	Hệ thống phản hồi
	1	Truy cập vào trang quản trị	Hiển thị trang quản trị
	2	Truy cập vào trang quản lý sản phẩm	Hiển thị trang quản lý sản phẩm
	3	Nhấn vào button sửa ứng với sản phẩm muốn cập nhật	Hiển thị popup cập nhật sản phẩm với các trường: tên sản phẩm, danh mục, tag, hình ảnh, giá bán, số lượng, mô tả
	4	Nhập thông tin	
	5	Nhấn lưu	Hệ thống kiểm tra thông tin và định dạng: - Nếu chính xác chuyển sang bước 6 - Ngược lại: + Tên sản phẩm có chứa ký tự đặc biệt (Luồng ngoại lệ EF1) + Danh mục chứa ký tự đặc biệt và hoặc có chứa chữ số (Luồng ngoại lệ EF2) + Tag chứa ký tự đặc biệt và hoặc có chứa chữ số (Luồng ngoại lệ EF3) + Giá bán có chứa ký tự đặc biệt và hoặc có chứa chữ cái (Luồng ngoại lệ EF4) + Số lượng có chứa ký tự đặc biệt và hoặc có chứa chữ số, không phải là số nguyên (Luồng ngoại lệ EF5) + Mô tả có chứa ký tự đặc biệt (Luồng ngoại lệ EF6)
Luồng ngoại lệ	6		Thông báo cập nhật sản phẩm thành công và chuyển đến trang quản lý sản phẩm
	EF1		Hiển thị thông báo: Tên sản phẩm không chứa ký tự đặc biệt
	EF2		Hiển thị thông báo: Tên danh mục chỉ chứa chữ cái

	EF3		Hiển thị thông báo: Tag chỉ chứa chữ cái
	EF4		Hiển thị thông báo: Giá bán là kiểu số
	EF5		Hiển thị thông báo: Số lượng là kiểu số nguyên
	EF6		Hiển thị thông báo: Mô tả không chứa ký tự đặc biệt

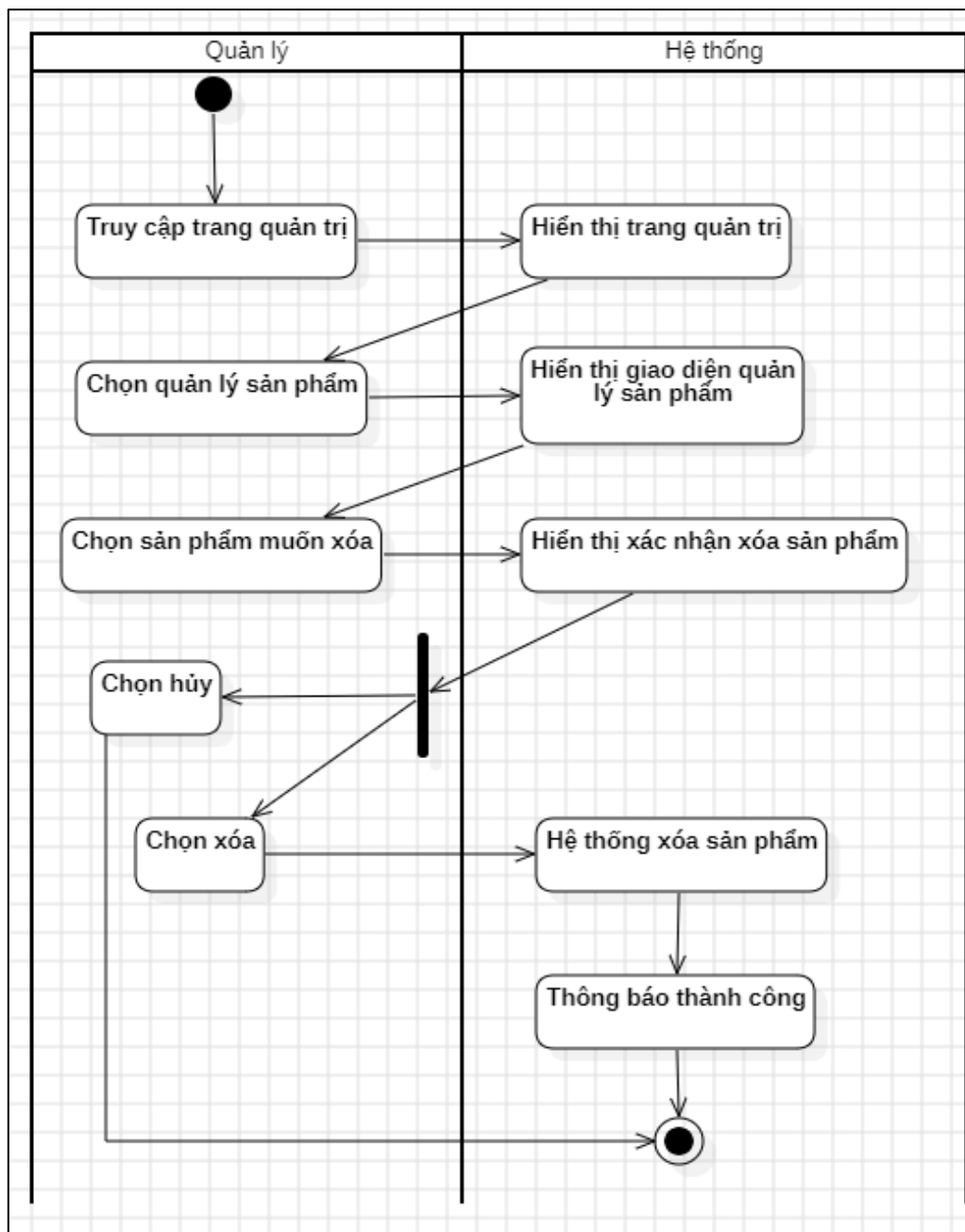


Hình 2.16 Biểu đồ hoạt động chức năng cập nhật sản phẩm

2.1.5.7 Kịch bản chức năng xóa sản phẩm

Bảng 2.8 Bảng kịch bản chức năng xóa sản phẩm

Tên use case	Xóa sản phẩm		
Tác nhân	Người quản trị		
Mô tả	Tại giao diện quản lý sản phẩm, chọn xóa sản phẩm		
Điều kiện tiên quyết	Người quản lý đã đăng nhập		
Luồng sự kiện		Tác nhân	Hệ thống phản hồi
	1	Truy cập vào trang quản trị	Hiển thị trang quản trị
	2	Truy cập vào trang quản lý sản phẩm	Chuyển đến trang quản lý sản phẩm
	3	Nhấn vào button xóa ứng với sản phẩm muốn xóa	Thông báo xác nhận xóa sản phẩm
	4	Nhấn xóa	Thông báo xóa sản phẩm thành công. (Luồng ngoại lệ EF1)
Luồng ngoại lệ	EF1	Nhấn hủy	Trở về trang quản lý sản phẩm



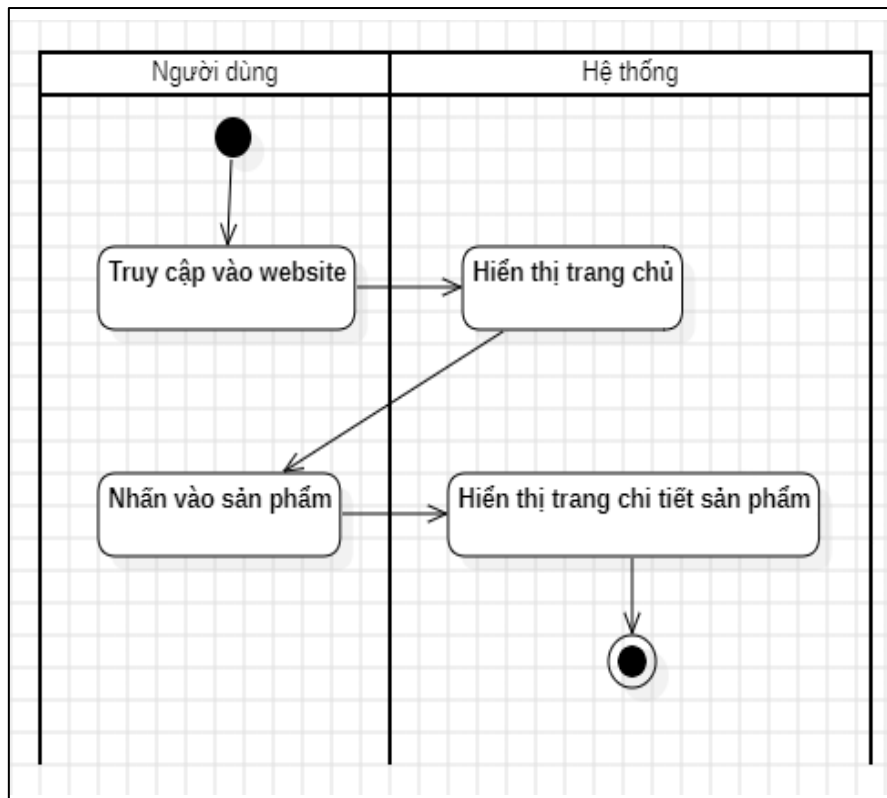
Hình 2.17 Biểu đồ hoạt động chức năng xóa sản phẩm

2.1.5.8 Kịch bản chức năng xem chi tiết sản phẩm

Bảng 2.9 Bảng kịch bản chức năng xem chi tiết sản phẩm

Tên use case	Xem chi tiết sản phẩm		
Tác nhân	Người sử dụng		
Mô tả	Tại giao diện hiển thị người dùng nhấn vào sản phẩm muốn xem thông tin		
Điều kiện tiên quyết			
		Tác nhân	Hệ thống phản hồi

Luồng sự kiện	1	Truy cập vào website	Hiển thị trang chủ
	2	Nhấn vào sản phẩm	Chuyển đến trang hiển thị chi tiết thông tin về sản phẩm được chọn
Luồng ngoại lệ			



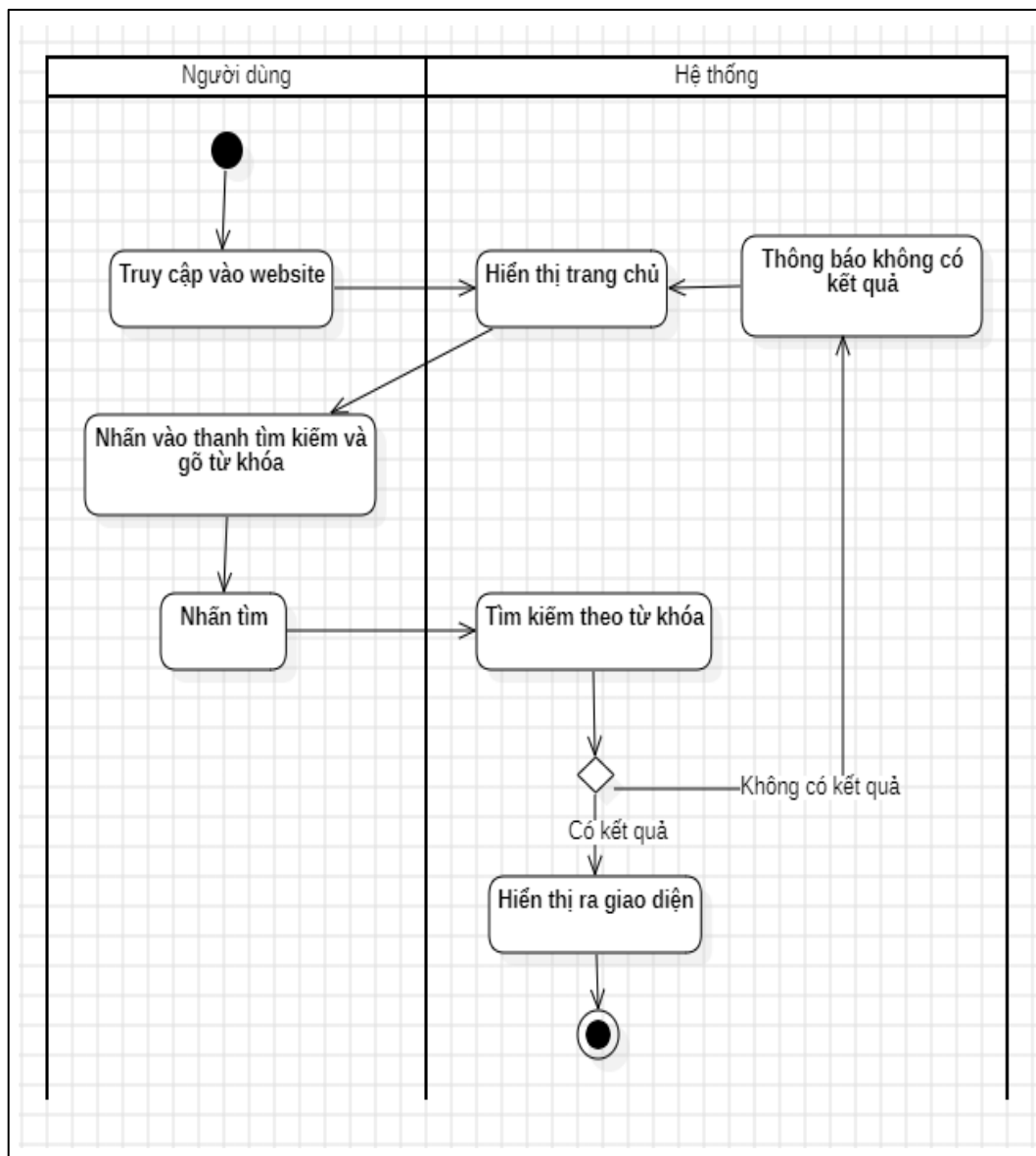
Hình 2.18 Biểu đồ hoạt động chức năng xem chi tiết sản phẩm

2.1.5.8 Kịch bản chức năng tìm kiếm sản phẩm

Bảng 2.10 Bảng kịch bản chức năng tìm kiếm sản phẩm

Tên use case	Xem chi tiết sản phẩm		
Tác nhân	Người sử dụng		
Mô tả	Tại giao diện hiển thị người dùng nhấn vào sản phẩm muốn xem thông tin		
Điều kiện tiên quyết			
Luồng sự kiện		Tác nhân	Hệ thống phản hồi
	1	Truy cập vào website	Hiển thị trang chủ

	2	Nhấn vào sản phẩm	Chuyển đến trang hiển thị chi tiết thông tin về sản phẩm được chọn
Luồng ngoại lệ			



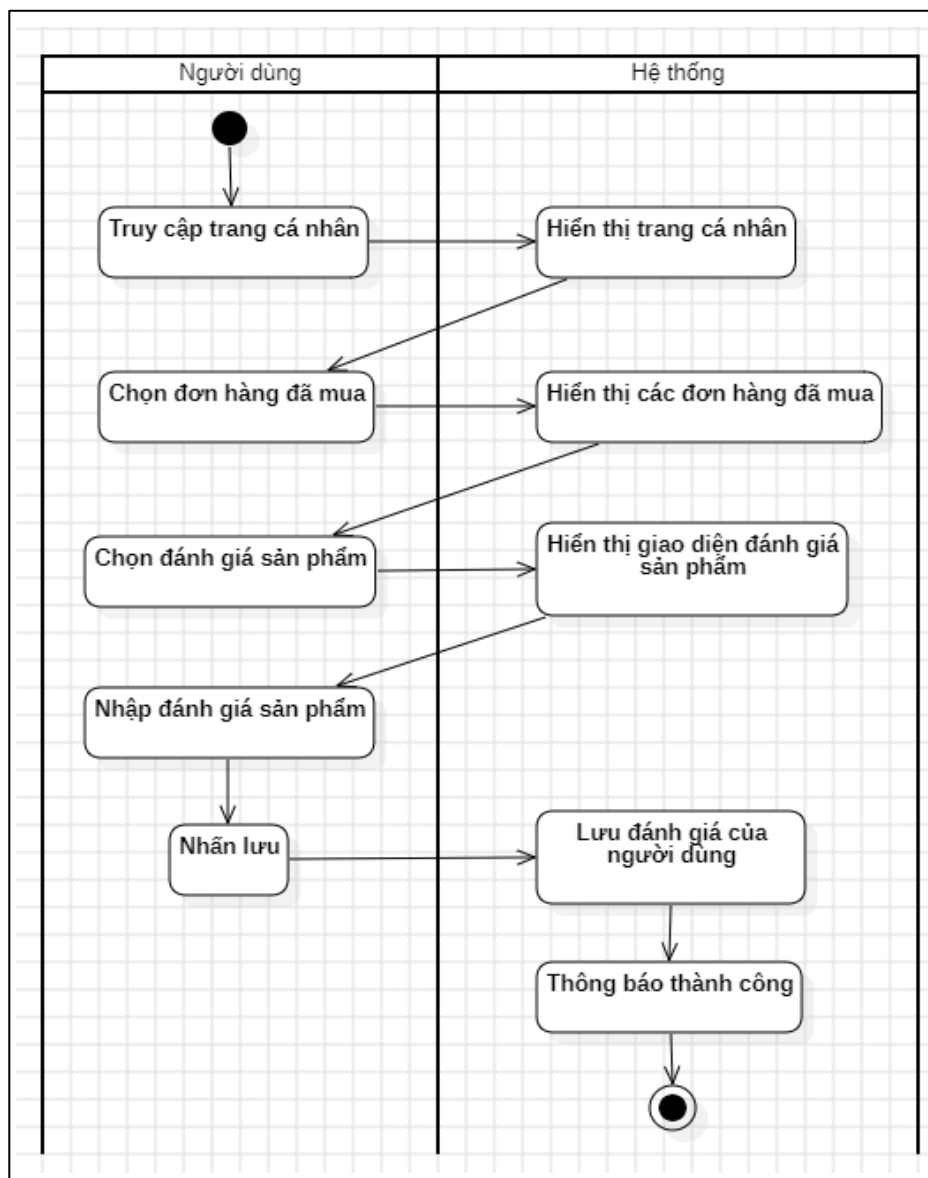
Hình 2.19 Biểu đồ hoạt động chức năng tìm kiếm sản phẩm

2.1.5.9 Kịch bản chức năng đánh giá sản phẩm

Bảng 2.11 Bảng kịch bản chức năng đánh giá sản phẩm

Tên use case	Đánh giá sản phẩm
Tác nhân	Người sử dụng
Mô tả	Người dùng có thể đánh giá sản phẩm sau khi đã nhận hàng
Điều kiện tiên quyết	Người dùng đã đăng nhập, đơn hàng đã được giao thành công

Luồng sự kiện		Tác nhân	Hệ thống phản hồi
	1	Truy cập vào trang cá nhân	Hiển thị trang cá nhân
	2	Nhấn vào đơn hàng đã mua	Hiển thị các đơn hàng đã mua
	3	Chọn đơn hàng muốn đánh giá	Hiển thị giao diện đánh giá
Luồng ngoại lệ	4	Nhập đánh giá	Lưu đánh giá và thông báo thành công

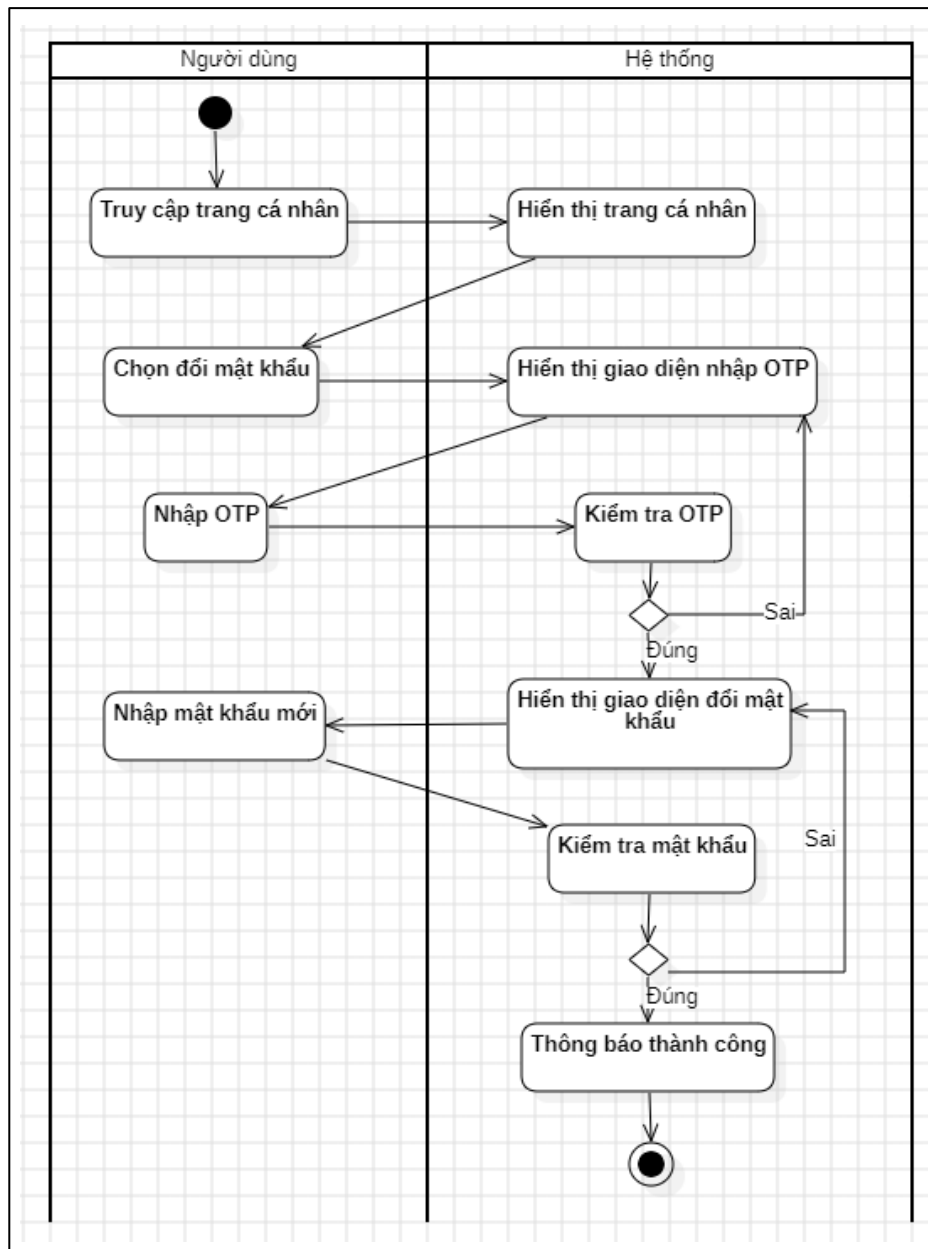


Hình 2.20 Biểu đồ hoạt động chức năng đánh giá sản phẩm

2.1.5.10 Kịch bản chức năng đổi mật khẩu

Bảng 2.12 Bảng kịch bản chức năng đổi mật khẩu

Tên use case	Đổi mật khẩu		
Tác nhân	Người sử dụng		
Mô tả	Tại trang cá nhân của người dùng, nhấn đổi mật khẩu		
Điều kiện tiên quyết	Người dùng đã đăng nhập		
Luồng sự kiện		Tác nhân	Hệ thống phản hồi
	1	Truy cập vào trang cá nhân	Hiển thị trang trang cá nhân
	2	Nhấn đổi mật khẩu	Hệ thống gửi OTP về email và hiển thị giao diện nhập OTP
	3	Nhập otp	Hệ thống kiểm tra otp: - Nếu chính xác thì chuyển qua bước 4 - Ngược lại (Luồng ngoại lệ EF1)
	4		Hệ thống hiển thị giao diện đặt lại mật khẩu gồm các trường: mật khẩu mới, nhập lại mật khẩu
	5	Nhập mật khẩu mới và nhập lại mật khẩu	Hệ thống kiểm tra thông tin, định dạng: - Nếu chính xác chuyển sang bước 6 - Ngược lại: Mật khẩu ít hơn 8 ký tự, có chứa khoảng trắng, không chứa tối thiểu 1 ký tự đặc biệt, 1 chữ cái, 1 chữ số(Luồng ngoại lệ EF2)
	6		Thông báo đổi mật khẩu thành công và chuyển đến trang cá nhân
Luồng ngoại lệ	EF1		Hiển thị thông báo: Mã otp không chính xác
	EF2		Hiển thị thông báo: Mật khẩu không hợp lệ



Hình 2.21 Biểu đồ hoạt động chức năng đổi mật khẩu

2.2 Thiết kế các chức năng

2.2.1 Giao diện đăng nhập

Chức năng: Giúp người dùng đăng nhập vào hệ thống

Hình 2.22

2.2.2 Giao diện đăng ký tài khoản

Chức năng: Giúp người dùng đăng ký tài khoản

Hình 2.23

2.2.3 Giao diện cho người dùng

2.2.3.1 Giao diện quản lý thông tin cá nhân

Chức năng: Giúp người dùng chỉnh sửa thông tin cá nhân

Hình 2.24

2.2.3.2 Giao diện đánh giá sản phẩm

Chức năng : Giúp người dùng đánh giá sản phẩm đã mua

Hình 2.25

2.2.4 Giao diện người quản trị

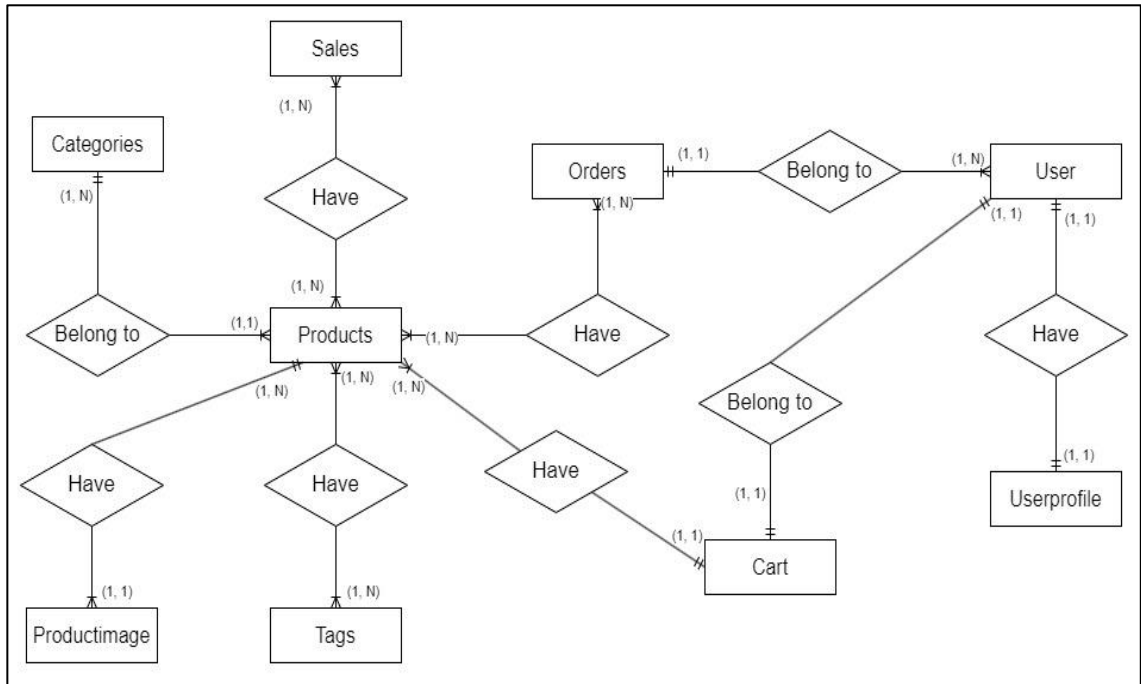
2.2.4.1 Giao diện quản lý sản phẩm

Chức năng : Giúp người quản lý tìm kiếm, thêm, xóa, sửa sản phẩm

Hình 2.26

2.3 Thiết kế cơ sở dữ liệu

2.3.1 Mô hình thực thể quan hệ



Hình 2.27 Sơ đồ thực thể quan hệ

2.3.2 Các bảng dữ liệu

2.3.2.1 Bảng categories

Dùng để lưu trữ các danh mục sản phẩm

Bảng 2.13 Bảng categories

STT	Tên cột	Kiểu dữ liệu	Kích thước	Mô tả
1	category_id	INT		Mã danh mục
2	category_name	VARCHAR	255	Tên danh mục

2.3.2.2 Bảng tags

Dùng để lưu trữ các nhãn của sản phẩm

Bảng 2.14 Bảng tags

STT	Tên cột	Kiểu dữ liệu	Kích thước	Mô tả
1	tag_id	INT		Mã của nhãn
2	tag_name	VARCHAR	255	Tên của nhãn

2.3.2.3 Bảng producttag

Dùng để ánh xạ nhãn vào sản phẩm

Bảng 2.15 Bảng producttag

STT	Tên cột	Kiểu dữ liệu	Kích thước	Mô tả
1	producttag_id	INT		Mã của bảng ánh xạ
2	product_id	INT		Mã của sản phẩm
3	tag_id	INT		Mã của nhãn

2.3.2.4 Bảng productimage

Lưu trữ hình ảnh của sản phẩm

Bảng 2.16 Bảng productimage

STT	Tên cột	Kiểu dữ liệu	Kích thước	Mô tả
1	image_id	INT		Mã hình ảnh
2	product_id	INT		Mã sản phẩm
3	image_url	VARCHAR	255	Địa chỉ hình ảnh

2.3.2.5 Bảng products

Lưu trữ thông tin sản phẩm

Bảng 2.17 Bảng products

STT	Tên cột	Kiểu dữ liệu	Kích thước	Mô tả
1	product_id	INT		Mã của sản phẩm
2	product_name	VARCHAR	255	Tên sản phẩm
3	category_id	INT		Mã của danh mục
4	product_price	INT		Giá của sản phẩm
5	product_quantity	INT		Số lượng sản phẩm
6	product_description	VARCHAR	500	Mô tả sản phẩm

2.3.2.6 Bảng users

Lưu trữ người dùng của hệ thống

Bảng 2.18 Bảng users

STT	Tên cột	Kiểu dữ liệu	Kích thước	Mô tả
1	Id	VARCHAR	255	Mã người dùng
2	OTP	LONGTEXT		Mã OTP
3	OtpExpiration	DATETIME	6	Thời hạn của mã OTP
4	Email	VARCHAR	256	Email của người dùng
5	PasswordHash	LONGTEXT		Mật khẩu mã hóa

2.3.2.7 Bảng *userprofile*

Lưu trữ thông tin cá nhân của người dùng

*Bảng 2.19 Bảng **userprofile***

STT	Tên cột	Kiểu dữ liệu	Kích thước	Mô tả
1	profile_id	INT		Mã của thông tin cá nhân
2	user_id	VARCHAR	255	Mã của người dùng
3	fullname	VARCHAR	255	Tên đầy đủ của người dùng
4	phone	VARCHAR		Số điện thoại
5	address	VARCHAR	255	Địa chỉ

2.3.2.8 Bảng *orders*

Lưu trữ thông tin các đơn hàng

*Bảng 2.20 Bảng **orders***

STT	Tên cột	Kiểu dữ liệu	Kích thước	Mô tả
1	order_id	INT		Mã đơn hàng
2	user_id	VARCHAR	255	Mã người dùng
3	order_total_price	INT		Tổng giá của đơn hàng
4	order_status	VARCHAR	255	Trạng thái đơn hàng
5	order_date	DATETIME		Ngày đặt hàng

2.3.2.9 Bảng orderdetail

Lưu trữ thông tin chi tiết đơn hàng

Bảng 2.21 Bảng orderdetail

STT	Tên cột	Kiểu dữ liệu	Kích thước	Mô tả
1	order_detail_id	INT		Mã chi tiết đơn hàng
2	order_id	INT		Mã đơn hàng
3	product_id	INT		Mã sản phẩm
4	order_detail_quantity	INT		Số lượng sản phẩm
5	order_detail_price	INT		Giá của sản phẩm

2.3.2.10 Bảng sales

Lưu trữ thông tin chương trình giảm giá

Bảng 2.22 Bảng sales

STT	Tên cột	Kiểu dữ liệu	Kích thước	Mô tả
1	sale_id	INT		Mã của chương trình
2	sale_begin_time	DATETIME		Thời gian bắt đầu
3	sale_end_time	DATETIME		Thời gian kết thúc

2.3.2.11 Bảng saledetail

Lưu trữ chi tiết chương trình giảm giá

Bảng 2.23 Bảng saledetail

STT	Tên cột	Kiểu dữ liệu	Kích thước	Mô tả
1	sale_detail_id	INT		Mã chi tiết chương trình
2	sale_id	INT		Mã chương trình giảm giá
3	product_id	INT		Mã sản phẩm
4	sale_percent	FLOAT		Mức giảm giá

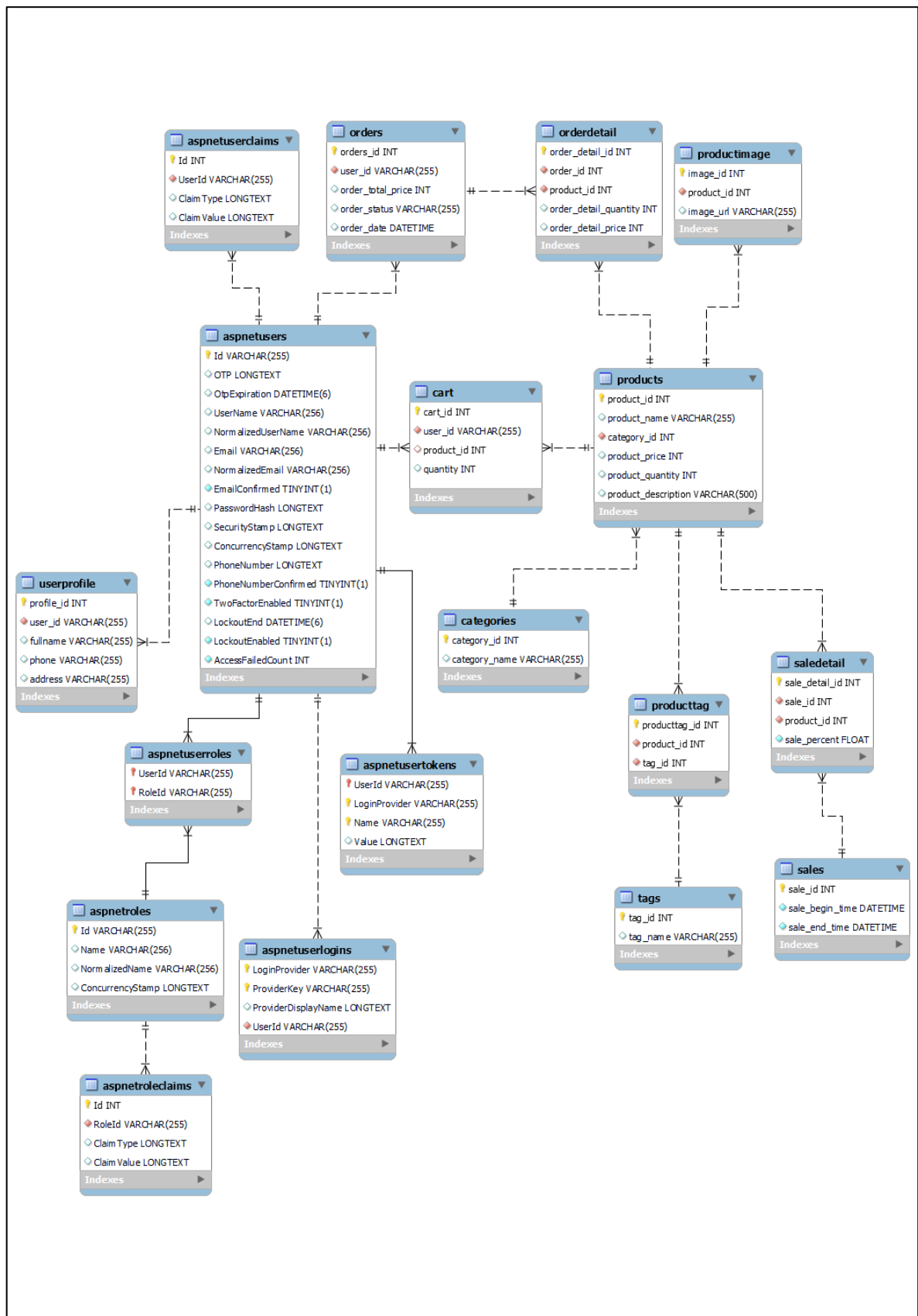
2.3.2.12 Bảng cart

Lưu trữ thông tin giỏ hàng

Bảng 2.24 Bảng cart

STT	Tên cột	Kiểu dữ liệu	Kích thước	Mô tả
1	cart_id	INT		Mã của giỏ hàng
2	user_id	VARCHAR	255	Mã người dùng
3	product_id	INT		Mã sản phẩm
4	quantity	INT		Số lượng

2.3.3 Mô hình quan hệ

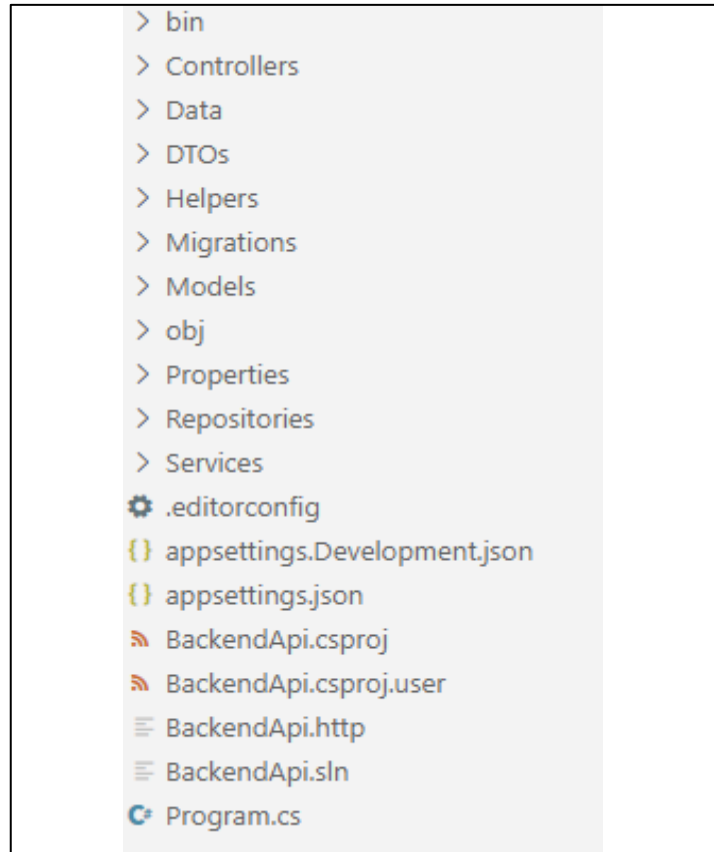


Hình 2.25 Mô hình quan hệ

CHƯƠNG 3 MÔ HÌNH PHẦN MỀM

3.1 Backend

Cấu trúc thư mục backend của dự án gồm nhiều thư mục



Hình 3.1 Cấu trúc backend

Bảng 3.1 Mô tả chi tiết backend

Tên thư mục	Chức năng
Controllers	Chứa các controller để xử lý yêu cầu tới API
Data	Chứa các class cấu hình kết nối đến cơ sở dữ liệu
DTOs	Chứa các class dữ liệu được truyền giữa các thành phần trong hệ thống
Models	Chứa các class đại diện ứng với các bảng trong cơ sở dữ liệu
Repositories	Chứa các interface và các class thực hiện thao tác trực tiếp với cơ sở dữ liệu
Services	Chứa các interface và các class gọi tới các hàm của repositories

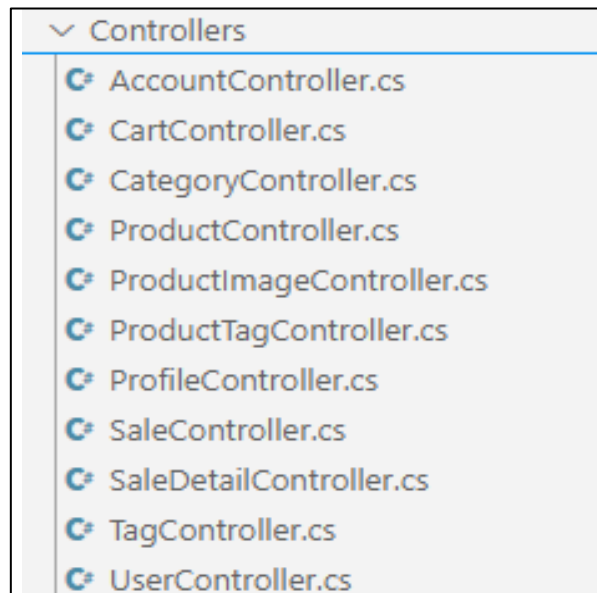
Chi tiết về cấu hình và khởi động hệ thống(khai báo, cấu hình các dịch vụ của hệ thống,...) được lưu ở trong file program.cs

```
var builder = WebApplication.CreateBuilder(args);
var configuration = builder.Configuration;
builder.Services.AddControllers();
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();

builder.Services.AddDbContext<AppDbContext>(options => {
    options.UseMySQL(builder.Configuration.GetConnectionString("DefaultConnection"),
        new MySqlServerVersion(new Version(8, 0, 36)));
});
builder.Services.AddIdentity<User, IdentityRole>(options => {
    options.Password.RequireDigit = true;
    options.Password.RequireNonAlphanumeric = true;
    options.Password.RequireLowercase = true;
    options.Password.RequiredLength = 8;
    options.User.RequireUniqueEmail = true;
})
.AddEntityFrameworkStores<AppDbContext>()
.AddDefaultTokenProviders();
```

Hình 3.2 File cấu hình hệ thống

3.1.1 Controller



Hình 3.3 Danh sách các controller

Bảng 3.2 Chi tiết các controller

Controller	Mô tả
AccountController	Xử lý yêu cầu liên quan đến tài khoản
CartController	Xử lý yêu cầu liên quan đến giỏ hàng
CategoryController	Xử lý yêu cầu liên quan đến danh mục sản phẩm
ProductController	Xử lý yêu cầu liên quan đến sản phẩm
ProductImageController	Xử lý yêu cầu liên quan đến hình ảnh của sản phẩm
ProductTagController	Xử lý yêu cầu liên quan đến gắn nhãn cho sản phẩm
ProfileController	Xử lý yêu cầu liên quan đến thông tin cá nhân của người dùng
SaleController	Xử lý yêu cầu liên quan đến chương trình giảm giá
SaleDetailController	Xử lý yêu cầu liên quan đến chi tiết chương trình giảm giá
TagController	Xử lý yêu cầu liên quan đến nhãn
UserController	Xử lý yêu cầu liên quan đến thông tin của người dùng
OrderController	Xử lý yêu cầu liên quan đến đơn hàng
OrderDetailController	Xử lý yêu cầu liên quan đến chi tiết đơn hàng

3.1.2 Models

Các models tương ứng với bảng trong cơ sở dữ liệu

Bảng 3.3 Bảng models

Model	Bảng trong cơ sở dữ liệu
Cart	Cart
Category	Categories
Product	Products
ProductImage	Productimage
ProductTag	Producttag
Profile	Userprofile

Sale	Sales
Saledetail	Saledetail
Tag	Tags
Order	Orders
Orderdetail	Orderdetail
Login, Register, User	User

3.1.3 Repositories

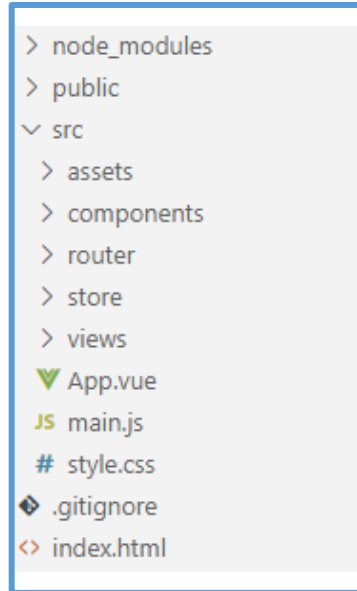
Bảng 3.4 Danh sách các repositories

Repository	Mô tả
CartRepository	Thực hiện các thao tác với bảng Cart
CategoryRepository	Thực hiện các thao tác với bảng Categories
ProductRepository	Thực hiện các thao tác với bảng Products
ProductImageRepository	Thực hiện các thao tác với bảng Productimage
ProductTagRepository	Thực hiện các thao tác với bảng Producttag
ProfileRepository	Thực hiện các thao tác với bảng Userprofile
SaleRepository	Thực hiện các thao tác với bảng Sales
SaleDetailRepository	Thực hiện các thao tác với bảng Saledetail
TagRepository	Thực hiện các thao tác với bảng Tags
OrderRepository	Thực hiện các thao tác với bảng Orders
OrderDetailRepository	Thực hiện các thao tác với bảng Orderdetail

3.2 Frontend

3.2.1 Cấu trúc chung

Cấu trúc thư mục cho phần frontend gồm các thư mục như hình :



Hình 3.4 Cấu trúc thư mục frontend

Bảng 3.5 Cấu trúc frontend chi tiết

Thư mục	Mô tả
node_modules	Nơi chứa tất cả các thư viện mà ứng dụng sử dụng
public	Nơi chứa các tài nguyên tĩnh, không được webpack xử lý
src	Nơi chứa mã nguồn của ứng dụng
assets	Nơi chứa các tài nguyên tĩnh, được webpack xử lý
components	Chứa các thành phần nhỏ của ứng dụng và có thể tái sử dụng
router	Nơi cấu hình route kết nối đến các component
store	Nơi chứa các file cấu hình để quản lý trạng thái của ứng dụng
views	Nơi chứa các trang giao diện của ứng dụng

3.2.2 Luồng hoạt động cơ bản

Mối liên hệ giữa Backend và Frontend

CHƯƠNG 4 CÀI ĐẶT ỨNG DỤNG

Môi trường cài đặt

4.1.1 MySQL Workbench 8.0 CE

4.1.2 Google Chrome

4.1.3 Visual Studio Code

Một số kết quả đạt được

4.2. Giao diện đăng nhập

4.2.2 Giao diện đăng kí tài khoản

4.2.3 Giao diện quản lý sản phẩm

4.2.4 Giao diện

4.3 Kết luận và hướng phát triển

4.3.1 Hạn chế

4.3.2 Hướng phát triển

TÀI LIỆU THAM KHẢO