

**BAN NHÂN DÂN THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC SÀI GÒN**

A blue circle with text

Description automatically generated

**CÁC CÔNG NGHỆ LẬP TRÌNH HIỆN ĐẠI**

Chủ đề: Web bán đồ ăn cho thú cưng

Sinh viên thực hiện:

Phạm Anh Duy\_3120411006

Thành phố Hồ Chí Minh, 20 tháng 2 năm 2025

**Mục lục**

[**Chương 1 Nghiên cứu đề tài** 3](#_Toc193541865)

[**1. Lý do chọn đề tài** 3](#_Toc193541866)

[**2. Tìm hiểu công nghệ** 3](#_Toc193541867)

[Docker là gì? 3](#_Toc193541868)

[Các thành phần chính của Docker 4](#_Toc193541869)

[React 8](#_Toc193541870)

[Khái niệm 8](#_Toc193541871)

[**3. Bối cảnh kinh doanh**: 11](#_Toc193541872)

[**4. Các chức năng chung** 12](#_Toc193541873)

[**Đối tượng sử dụng** 13](#_Toc193541874)

[Tài khoản Admin (Quản lý Nhân Viên) 13](#_Toc193541875)

[Tài khoản khách Hàng 13](#_Toc193541876)

[**Phi chức năng** 14](#_Toc193541877)

[Chương 2 Thực hiện dự án 16](#_Toc193541878)

[**1. Thiết kế usecase** 16](#_Toc193541879)

[1.1 Usecase tổng quát 16](#_Toc193541880)

[1.2 UseCase quản lý của Admin 16](#_Toc193541881)

[1.3 UseCase Quản lý bình luận đánh giá của khách hàng 17](#_Toc193541882)

[1.4 Usecase quản lý giỏ hàng 17](#_Toc193541883)

[1.5 UseCase điều khiển truy cập 18](#_Toc193541884)

[1.6 UseCase Xem chi tiết sản phẩm 18](#_Toc193541885)

[1.7 Usecase thanh toán 19](#_Toc193541886)

[1.8 Usecase hàng tồn kho 20](#_Toc193541887)

[**2. User story** 20](#_Toc193541888)

[2.1 Đánh giá của khách hàng 20](#_Toc193541889)

[2.2 Danh mục sản phẩm 20](#_Toc193541890)

[2.3 Chi tiết sản phẩm 21](#_Toc193541891)

[2.4 Giỏ hàng 21](#_Toc193541892)

[2.5 Quyền truy cập 21](#_Toc193541893)

[2.6 Thanh toán 22](#_Toc193541894)

[2.7 Hàng tồn kho 22](#_Toc193541895)

[**3. Biểu đồ hoạt động** 22](#_Toc193541896)

[**3.1 Biểu đồ hoạt động đăng nhập, đăng xuất** 22](#_Toc193541897)

[**2. Biểu đồ hoạt động tìm kiếm sản phẩm** 26](#_Toc193541898)

[**3. Biểu đồ hoạt động xem chi tiết sản phẩm** 27](#_Toc193541899)

[**4. Biểu đồ hoạt động đặt hàng** 28](#_Toc193541900)

[**5. Biểu đồ hoạt động đánh giá sản phẩm** 29](#_Toc193541901)

[Biểu đồ hoạt động đánh giá sản phẩm 29](#_Toc193541902)

# **Chương 1 Nghiên cứu đề tài**

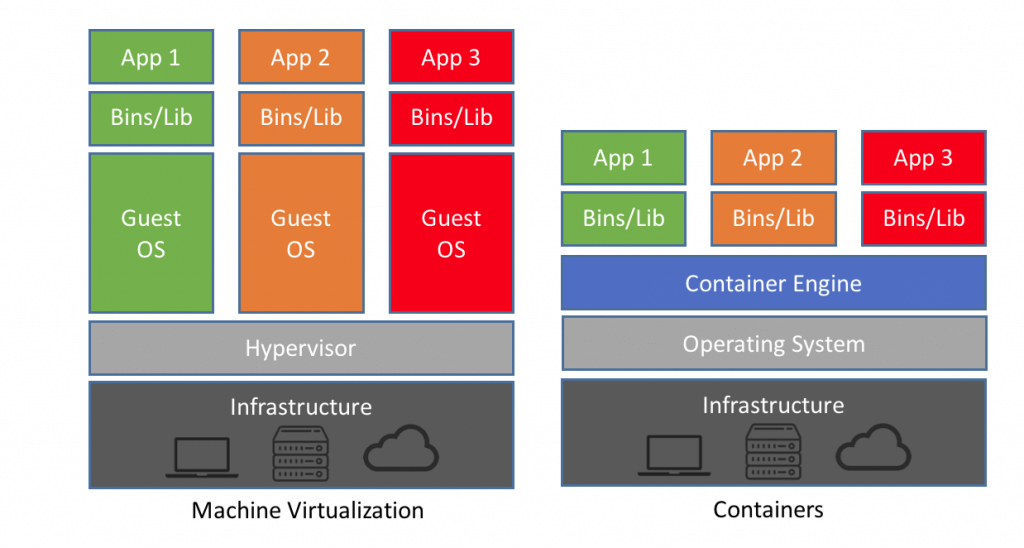
## **1. Lý do chọn đề tài**

Nhu cầu tiêu dùng trong xã luôn là lý do thúc đẩy sản xuất , như chúng cũng biết được việc thiếu thông tin cho công đoạn đáp ứng cung cầu làm cho công đoạn đáp ứng cung cầu cho người tiêu dùng trở nên khó khăn và gây nhiều lãng phí cho xã hội. Bên cạnh đó việc phổ biến sử dụng Internet đã tạo ra một bước ngoặc mới trong trong định hướng phát triển ngành Công nghệ thông tin của nước ta cùng với nhu cầu của mọi người và hỗ trợ cho việc mua bán diễn ra một cách nhanh chóng, tiết kiệm đã trở nên vô cùng cấp thiết. Mua bán qua mạng cũng là một xu thế hiện nay, hay còn gọi là thương mại điện tử, đay là một mảnh đất mới cho thị trường hàng tiêu dùng trong giai đoạn phát triển kinh tế nước ta.

## **2. Tìm hiểu công nghệ**

### 2.1Docker là gì?

**Docker** là một nền tảng mở dành cho các lập trình viên, quản trị hệ thống dùng để xây dựng, vận chuyển và chạy các ứng dụng phân tán. Docker giúp đóng gói phần mềm vào các đơn vị tiêu chuẩn hóa và được gọi là container



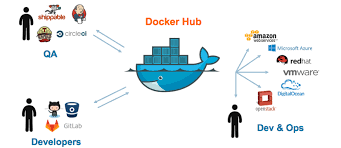
Ưu điểm của Docker

* Tiện lợi: Bình thường khi cần chạy ứng dụng chúng ta cần cài đầy đủ môi trường trên máy tính, chưa kể có sự xung đột, sự cố xảy ra với các ứng dụng. Với Docker, bạn có thể đóng gói tất cả các thành phần của ứng dụng vào một container và chạy nó trên bất kỳ máy tính nào mà không cần phải cài đặt lại môi trường.
* Dễ dàng sử dụng: Docker rất dễ cho mọi người sử dụng từ developers, systems admins, architects… nó tận dụng lợi thế của container để build, test nhanh chóng. Có thể đóng gói ứng dụng trên laptop của họ và chạy trên public cloud, private cloud… Câu thần chú là “Build once, run anywhere”.
* Tốc độ: Docker Containers tương đối nhẹ và có tốc độ rất nhanh. Bạn hoàn toàn có thể tạo và khởi chạy chỉ trong vài giây.
* Linh hoạt: Triển khai ở bất kỳ nơi đâu do sự phụ thuộc của ứng dụng vào tầng OS cũng như cơ sở hạ tầng được loại bỏ.
* Môi trường chạy và khả năng mở rộng: Bạn có thể chia nhỏ những chức năng của ứng dụng thành các container riêng lẻ. Ví dụng Database chạy trên một container và Redis cache có thể chạy trên một container khác trong khi ứng dụng Node.js lại chạy trên một cái khác nữa. Với Docker, rất dễ để liên kết các container với nhau để tạo thành một ứng dụng, làm cho nó dễ dàng scale, update các thành phần độc lập với nhau.

### 2.2 Các thành phần chính của Docker

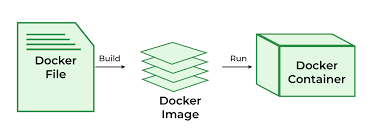
#### Docker Hub

* Docker Hub: dịch vụ lưu trữ giúp chứa các Docker image. Trên DockerHub có hàng ngàn public images được tạo bởi cộng đồng cho phép bạn dễ dàng tìm thấy những image mà bạn cần. Và chỉ cần pull về và sử dụng với một số config mà bạn mong muốn.



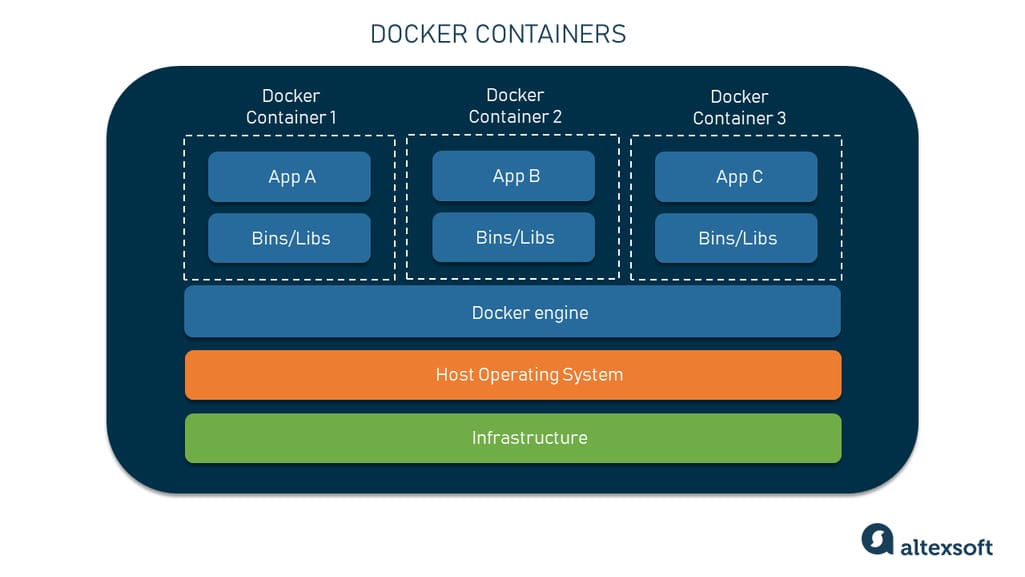
#### Docker Image

* Docker Image: một dạng tập hợp các tệp của ứng dụng, được tạo ra bởi Docker engine. Nội dung của các Docker image sẽ không bị thay đổi khi di chuyển. Docker image được dùng để chạy các Docker container. Bạn có thể tự build một image riêng cho mình hoặc sử dụng những image được chia sẽ từ cộng đồng Docker Hub. Một image sẽ được build dựa trên những chỉ dẫn của Dockerfile.



#### Docker Container

* Docker Container: một dạng runtime của các Docker image, dùng để làm môi trường chạy ứng dụng. Bạn có thể create, start, stop, move or delete container dựa trên Docker API hoặc Docker CLI.



#### Dockerfile

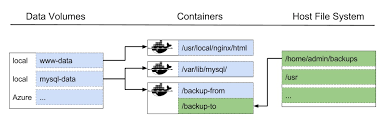
* Dockerfile là một file dạng text không có phần đuôi mở rộng, chứa các đặc tả về một trường thực thi phần mềm, cấu trúc cho Docker Image. Từ những câu lệnh đó, Docker sẽ build ra Docker image (thường có dung lượng nhỏ từ vài MB đến lớn vài GB).

A blue whale with a graduation cap

AI-generated content may be incorrect.

#### Docker volumes

* Docker volume là một volume được tạo ra cho phép các container mount volume vào trong các container hay dễ hiểu hơn là đocker sử dụng Volume đó thay thế cho 1 folder của container.



#### Docker Compose

* Docker Compose: là công cụ cho phép run app với nhiều Docker containers 1 cách dễ dàng hơn. Docker Compose cho phép bạn config các command trong file docker-compose.yml để sử dụng lại. Có sẵn khi cài Docker.

A diagram of a octopus

AI-generated content may be incorrect.

#### Docker NetWork

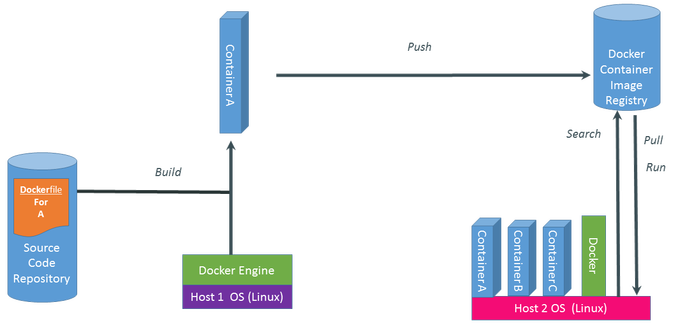
Docker network là nơi sẽ đảm nhiệm nhiệm vụ cho container kết nối vào network

* Các container cùng một network có thể liên lạc với nhau qua tên của container và cổng (port) được lắng nghe của container trên mạng đó
* Kết nối trên 1 host hoặc nhiều host,
* Kết nối giữa các cụm (swarm) docker containers.
* Kết nối container với các mạng khác nằm ngoài docker.
* Có thể cung cấp hầu hết các chức năng mà một hệ thống mạng bình thường cần có.

A diagram of a production server

AI-generated content may be incorrect.

#### Quy trình một hệ thống sử dụng Docker

[](https://devops.vinahost.vn/Image/Docker-process.png)

Như trong hình vẽ, một hệ thống Docker được thực thi với 3 bước chính :

***Build -> Push -> Pull,Run***

* Build: Đầu tiên tạo một dockerfile, trong dockerfile này chính là code của chúng ta. Dockerfile này sẽ được Build tại một máy tính đã cài đặt Docker Engine. Sau khi build ta sẽ có được Container, trong Container này chứa ứng dụng kèm bộ thư viện của chúng ta.
* Push: Sau khi có được Container, chúng ta thực hiện push Container này lên cloud và lưu tại đó.
* Pull, Run: Nếu một máy tính khác muốn sử dụng Container chúng ta thì bắt buộc máy phải thực hiện việc Pull container này về máy, tất nhiên máy này cũng phải cài Docker Engine. Sau đó thực hiện Run Container này.

### 2.3 React

**Khái niệm**

React (ReactJS) là một thư viện JavaScript mã nguồn mở, được dùng để xây dựng giao diện người dùng (frontend) cho web. React chỉ tập trung vào phần hiển thị giao diện (view), chứ không can thiệp vào cách sắp xếp logic nghiệp vụ hoặc cấu trúc ứng dụng.

A red atom model with red circles and dots

AI-generated content may be incorrect.

**Một số tính năng nổi bật của ReactJS**

Kiến trúc Component-Based

ReactJS cung cấp tính năng chia nhỏ UI thành các component nhỏ hơn và có tính độc lập. Mỗi components có trạng thái và thuộc tính (props) riêng biệt.

JSX (JavaScript Syntax Extension)

JSX là phần mở rộng cú pháp cho JavaScript cho phép lập trình viên viết mã giống HTML trong các tệp JavaScript của nó. Đồng thời, JSX làm cho components ReactJS dễ đọc và thu hút hơn.

const name="GeekforGeeks";

const ele = <h1>Welcome to {name}</h1>;



**ReactJS hoạt động như thế nào?**

ReactJS có thể cho phép truyền mã HTML với JavaScript với những lợi ích như:

* DOM là một cấu trúc cây biểu diễn tài liệu HTML, và JavaScript có thể tương tác với DOM để thay đổi nội dung và cấu trúc của trang web.
* ReactJS sử dụng một thuật toán diff hiệu quả để tìm ra những phần tử cần thay đổi và chỉ cập nhật những phần đó trên DOM thực tế, tránh việc cập nhật lại toàn bộ DOM.
* Khi có nhu cầu đọc và ghi vào DOM, JSX sẽ sử dụng DOM ảo của nó. Sau đó DOM ảo sẽ cố gắng tìm cách hiệu quả để cập nhật DOM của trình duyệt.

Trong ReactJS, bạn tạo ra các phần tử React bằng cách sử dụng các hàm JSX như <div>, <button>,… Các phần tử này không phải là DOM thực, mà là các đối tượng đơn giản được tạo ra dễ dàng.

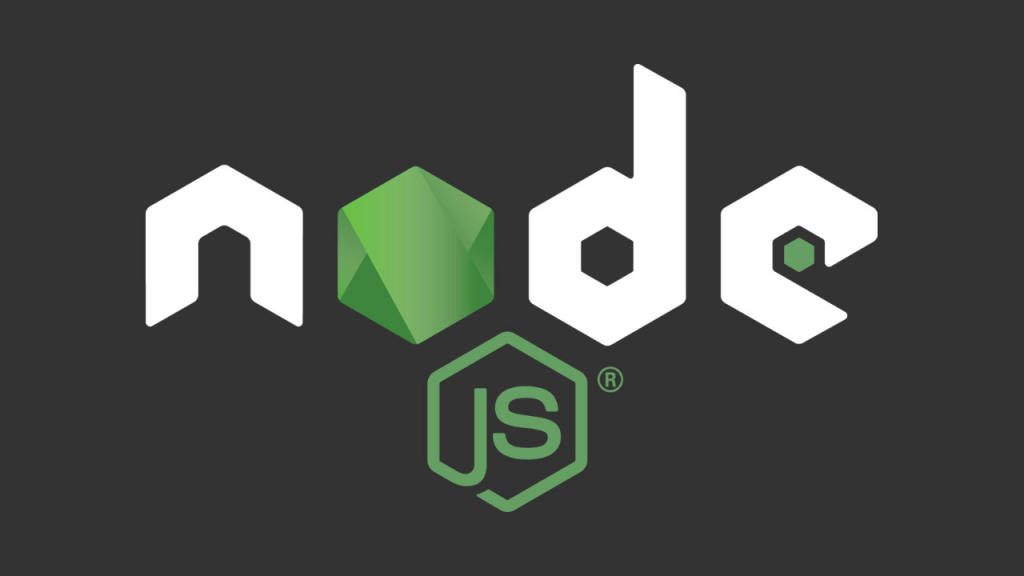
ReactJS sử dụng một DOM ảo (Virtual DOM) để tối ưu quá trình cập nhật DOM thực. Khi bạn cập nhật một phần tử React, ReactJS sẽ so sánh phần tử mới với phần tử cũ trong DOM ảo, sau đó chỉ cập nhật những phần cần thiết trong DOM thực, nhờ tốc độ xử lý nhanh của JavaScript.

### 2.4 NodeJs

**Khái niệm**

Nodejs là một môi trường runtime chạy JavaScript đa nền tảng, open-source (mã nguồn mở), được sử dụng rộng rãi bởi lập trình viên cho nhiều dự án. Nodejs cung cấp nhiều thư viện khác nhau, giúp đơn giản hóa việc lập trình.

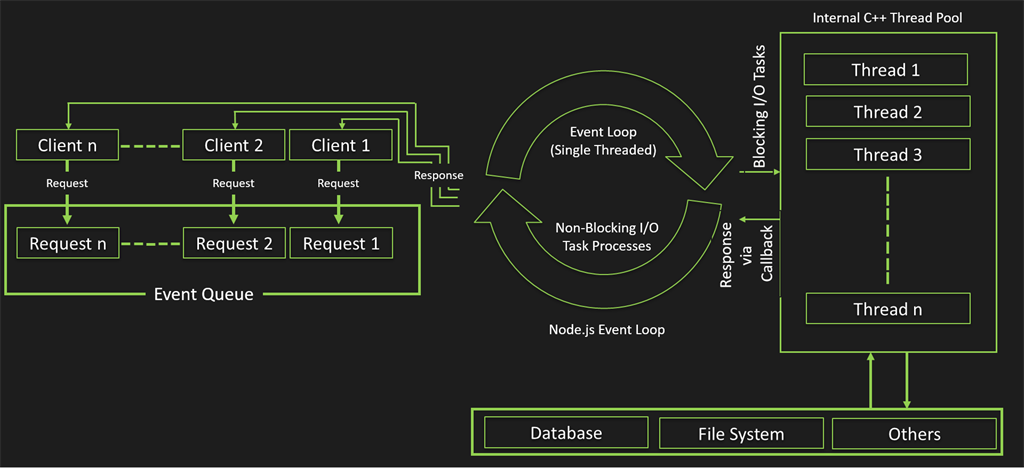
* **Nodejs là một môi trường JavaScript runtime**: Nodejs cung cấp môi trường runtime ngoài trình duyệt, cho phép bạn chạy mã JS nhanh chóng.
* **Nodejs đa nền tảng**: Nodejs có thể chạy trên nhiều nền tảng hệ điều hành khác nhau: Windows, Linux, Unix, Mac OS X,...
* **Nodejs là open-source**: source Nodejs được cung cấp công khai, cho phép bạn tạo ra module, đóng góp và duy trì.
* Một trong những tính năng chính của Nodejs là, cho phép xử lý đồng thời nhiều yêu cầu, mà không chặn các yêu cầu khác.



**Các tính năng của NodeJS**

* **Lập trình hướng sự kiện và không đồng bộ**: Toàn bộ API trong thư viện NodeJS đều không đồng bộ, hay không bị chặn. Về cơ bản điều này có nghĩa là một server sử dụng NodeJS sẽ không bao giờ chờ một API trả về dữ liệu. Server sẽ chuyển sang API kế tiếp sau khi gọi API đó và cơ chế thông báo của Events trong NodeJS giúp server nhận được phản hồi từ lần gọi API trước.
* **Cực kỳ nhanh chóng**: Được xây dựng trên Công cụ JavaScript V8 của Google Chrome, thư viện NodeJS có khả năng xử lý mã vô cùng nhanh.
* **Đơn luồng/Single thread nhưng có khả năng mở rộng cao**: NodeJS sử dụng một mô hình luồng đơn với vòng lặp sự kiện/event. Cơ chế event cho phép máy chủ phản hồi non-blocking và cũng cho phép khả năng mở rộng cao hơn so với các server truyền thống hỗ trợ giới hạn các thread để xử lý yêu cầu. NodeJS sử dụng một chương trình đơn luồng, cùng một chương trình có thể cung cấp dịch vụ cho một số lượng yêu cầu lớn hơn so với các máy chủ truyền thống như Apache HTTP Server.
* **Không có buffer** - Các ứng dụng NodeJS không có vùng nhớ tạm thời (buffer) cho bất kỳ dữ liệu nào. Các ứng dụng này chỉ đơn giản xuất dữ liệu theo khối.
* **License** - NodeJS được phát hành theo giấy phép MIT.

**Cơ chế hoạt động của Nodejs**



* Nhiều client sẽ gửi request đến server, sẽ được đưa vào Event Queue.
* Event Loop sẽ lấy lần lượt các request từ hàng đợi và xử lý chúng.
* Nếu request cần thực hiện các tác vụ non-blocking I/O, Event Loop sẽ tự động xử lý rồi trả về response.
* Đối với những tác vụ blocking I/O, Event Loop sẽ đưa tasks qua Thread Pool (thông qua thư viện Libuv).
* Khi thread đã thực hiện xong, kết quả sẽ được Nodejs đẩy callback của tasks đó về Event Queue để xử lý.
* Cuối cùng, response sẽ được phản hồi lại cho các client.

Qua sơ đồ này, Nodejs sử dụng Event Loop và mô hình I/O bất đồng bộ để xử lý một số lượng lớn request đồng thời một cách hiệu quả.

## **3. Bối cảnh kinh doanh**:

Trang web EgaPets cung cấp kịch bản kinh doanh chi tiết cho Danh mục sản phẩm, Giỏ hàng, Quy trình thanh toán, Hàng tồn kho, Đánh giá sản phẩm và Kiểm soát truy cập.

Với Danh mục sản phẩm, khách hàng (Guest) có thể duyệt và xem chi tiết sản phẩm bao gồm tên sản phẩm, số lượng tồn kho, loại sản phẩm, mô tả và xếp hạng sản phẩm (1-5 sao). Khách hàng cũng có thể lọc và tìm kiếm sản phẩm theo loại sản phẩm và giá. Khi xem chi tiết sản phẩm, khách hàng có thể thêm sản phẩm vào giỏ hàng và lựa chọn mua ngay. Nếu khách hàng chọn mua sản phẩm ngay lập tức, hệ thống sẽ yêu cầu khách hàng đăng nhập để xác định tài khoản khách hàng (Customer).

Với Giỏ hàng, khách hàng có thể thêm sản phẩm vào giỏ hàng từ trang danh mục hoặc trang chi tiết sản phẩm. Giỏ hàng sẽ hiển thị thông tin tổng quát bao gồm tổng chi phí, chi phí vận chuyển, và tổng số tiền đặt hàng. Khách hàng có thể cập nhật giỏ hàng bằng cách thêm hoặc xóa sản phẩm.

Với Quy trình thanh toán, khách hàng cần điền thông tin cá nhân gồm tên, địa chỉ, số điện thoại và lựa chọn phương thức thanh toán (COD, ví Momo hoặc chuyển khoản ngân hàng). Đặc biệt, với hình thức thanh toán quét mã QR, hệ thống EgaPets tích hợp sẵn mã QR với các thông tin thanh toán tự động như giá tiền, mã hóa đơn, và nội dung chuyển khoản, giúp khách hàng dễ dàng kiểm tra trước khi xác nhận thanh toán. Sau khi hoàn tất thanh toán, hệ thống sẽ gửi email thông báo trạng thái đơn hàng tới khách hàng.

Với tính năng Đánh giá, khách hàng có thể đánh giá sản phẩm từ 1 đến 5 sao dựa trên trải nghiệm của họ.

Với Kiểm soát truy cập, trang web EgaPets xác định 3 vai trò quản lý chính:

* Admin: Có toàn quyền quản lý và truy cập vào tất cả các bảng dữ liệu trong hệ thống.
* Nhân viên: Được phép truy cập vào các bảng dữ liệu SanPham, NhanVien, HoaDon.
* Khách hàng: Được phép truy cập vào các bảng dữ liệu SanPham, KhachHang, HoaDon.

Khách hàng và Admin đều có thể đăng nhập hoặc đăng xuất khỏi hệ thống. Khách hàng sẽ được chuyển hướng đến trang danh mục sản phẩm, còn Admin sẽ được chuyển hướng đến trang quản trị hệ thống.

Ngoài ra, khi trang web EgaPets được khởi tạo, cần có các nhiệm vụ thiết lập ban đầu như tạo người dùng Admin, hai người dùng Khách hàng mẫu và nhập liệu mẫu cho sản phẩm, hàng tồn kho và một số đánh giá sản phẩm

## **4. Các chức năng chung**

**Trang chủ (Homepage)**

* Hiển thị sản phẩm nổi bật, danh mục sản phẩm.
* Banner quảng cáo, chương trình giảm giá.

**Thanh tìm kiếm & Bộ lọc**

* Tìm kiếm sản phẩm theo tên, mã, thương hiệu.
* Lọc theo giá, danh mục, đánh giá, thương hiệu.

**Trang chi tiết sản phẩm**

* Hình ảnh sản phẩm, mô tả, giá cả, đánh giá.
* Thông tin về tồn kho, thông số kỹ thuật.

**Giỏ hàng & Thanh toán**

* Thêm, xóa, cập nhật số lượng sản phẩm trong giỏ.
* Chọn phương thức thanh toán (COD, online banking, ví điện tử).
* Chọn phương thức vận chuyển (giao hàng nhanh, tiết kiệm).

**Theo dõi đơn hàng**

* Xem lịch sử đơn hàng, trạng thái giao hàng.
* Nhận thông báo khi đơn hàng thay đổi trạng thái.

**Đăng ký & Đăng nhập**

* Đăng ký bằng email/số điện thoại.
* Đăng nhập bằng mật khẩu hoặc social login (Google, Facebook).

**Quản lý tài khoản**

* Cập nhật thông tin cá nhân (tên, số điện thoại, địa chỉ).
* Đổi mật khẩu, quản lý bảo mật tài khoản.

**Đánh giá sản phẩm**

* Viết nhận xét, đánh giá sao cho sản phẩm đã mua.

**Yêu cầu hỗ trợ**

* Liên hệ chăm sóc khách hàng, gửi yêu cầu đổi trả.

**Quản lý sản phẩm**

* Thêm, sửa, xóa sản phẩm.
* Quản lý danh mục sản phẩm.

**Quản lý đơn hàng**

* Xác nhận đơn hàng, cập nhật trạng thái (đang xử lý, đã giao).
* Hủy đơn hàng nếu cần.

**Quản lý khách hàng**

* Xem danh sách khách hàng, khóa/mở tài khoản.

**Quản lý khuyến mãi**

* Tạo mã giảm giá, chương trình khuyến mãi.

**Thống kê & Báo cáo**

* Xem doanh số bán hàng theo ngày/tháng.
* Xem số lượng đơn hàng, sản phẩm bán chạy.

## **5. Đối tượng sử dụng**

**Tài khoản Admin (Quản lý Nhân Viên)**

* Quản lý nhân viên: Thêm, sửa, xóa tài khoản nhân viên.
* Phân quyền nhân viên: Quyết định nhân viên có thể thao tác những gì trên hệ thống.
* Quản lý đơn hàng: Theo dõi, kiểm duyệt, cập nhật trạng thái đơn hàng.
* Quản lý sản phẩm: Thêm, sửa, xóa sản phẩm nhưng có thể bị giới hạn theo quyền của super admin.
* Quản lý khách hàng: Xem danh sách khách hàng, hỗ trợ xử lý vấn đề liên quan.
* Xử lý khiếu nại, đổi trả: Quản lý yêu cầu đổi trả hàng, xử lý khiếu nại từ khách.

**Tài khoản khách Hàng**

* Đăng ký, đăng nhập: Tạo tài khoản để mua sắm.
* Tìm kiếm & xem sản phẩm: Duyệt danh mục, lọc sản phẩm theo giá, thương hiệu.
* Đặt hàng & thanh toán: Chọn sản phẩm, thanh toán qua nhiều phương thức.
* Theo dõi đơn hàng: Xem trạng thái đơn hàng & cập nhật vận chuyển.
* Đánh giá & nhận xét: Viết review về sản phẩm đã mua.
* Yêu cầu hoàn trả & hỗ trợ: Gửi yêu cầu đổi trả hàng.

## **6. Phi chức năng**

Hiệu suất (Performance)

* Ứng dụng phải tải trang nhanh chóng, thời gian phản hồi không quá 3 giây.
* Hỗ trợ tải trang động để giảm thời gian chờ đợi của người dùng.
* Tối ưu cơ sở dữ liệu để xử lý đơn hàng với tốc độ cao và tránh truy vấn dư thừa.

Bảo mật (Security)

* Mã hóa mật khẩu bằng thuật toán an toàn
* Xác thực 2 yếu tố (2FA) cho tài khoản quản trị cấp cao.
* Phân quyền chặt chẽ giữa các vai trò (Super Admin, Admin, Nhân viên, Khách hàng).
* Ghi log hoạt động người dùng để theo dõi thay đổi quan trọng trên hệ thống.

**Khả năng mở rộng (Scalability)**

* Hệ thống có thể mở rộng horizontally (nhiều máy chủ) khi số lượng người dùng tăng cao.
* Hỗ trợ caching (Redis, Memcached) để giảm tải truy vấn vào database.

**Tính ổn định & Khả dụng (Availability & Reliability)**

* Đảm bảo uptime ≥ 99.9%, tránh downtime ảnh hưởng đến doanh thu.
* Cơ chế tự động khôi phục khi có lỗi hệ thống
* Backup dữ liệu định kỳ để tránh mất thông tin quan trọng.
* Dễ bảo trì & Quản lý (Maintainability & Manageability)
* Mã nguồn được viết theo cấu trúc chuẩn (MVC, Clean Architecture, Microservices nếu cần).
* Tài liệu API đầy đủ nếu có tích hợp với bên thứ ba.
* Log lỗi rõ ràng, dễ truy vết khi có vấn đề.

**Trải nghiệm người dùng (User Experience - UX)**

* Giao diện thân thiện, dễ sử dụng, hỗ trợ cả desktop và mobile.
* Tìm kiếm nhanh với gợi ý sản phẩm.
* Thanh toán mượt mà, hạn chế tối đa số bước rườm rà.
* Hỗ trợ đa ngôn ngữ nếu cần.

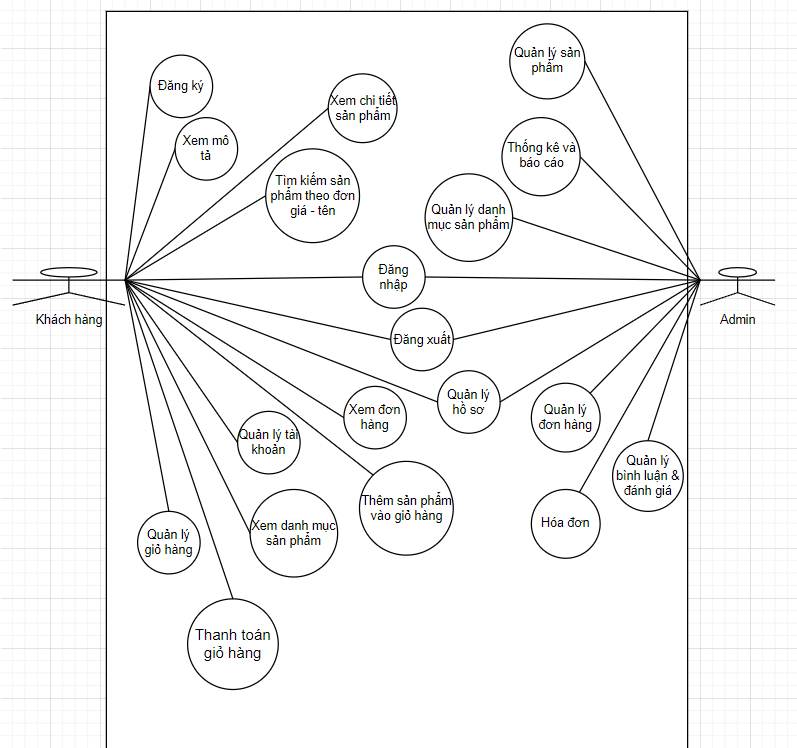
**Tích hợp & Hỗ trợ nền tảng khác**

* Hỗ trợ tích hợp cổng thanh toán (Momo, ZaloPay, PayPal, Stripe...).
* Hỗ trợ các dịch vụ vận chuyển (GHN, GHTK, Viettel Post...).
* Tích hợp social login (Google, Facebook, Apple ID...).
* API mở cho các ứng dụng khác có thể kết nối.

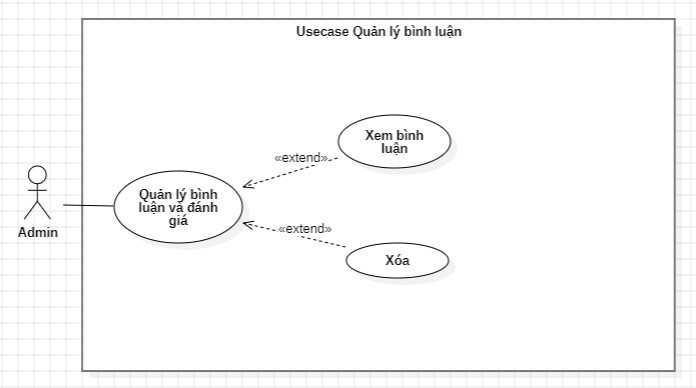
# Chương 2 Thực hiện dự án

## **1. Thiết kế usecase**

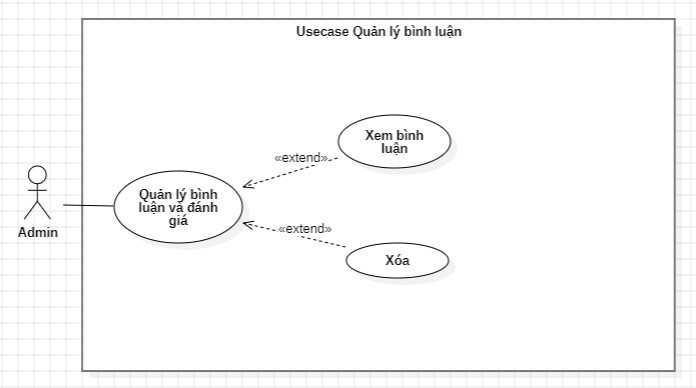
### 1.1 Usecase tổng quát



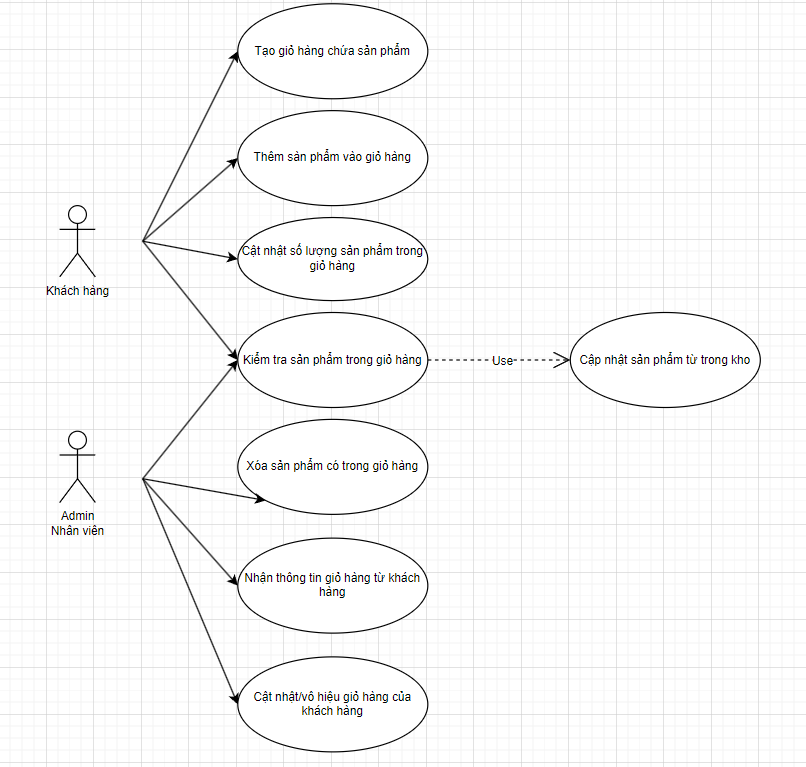
### 1.2 UseCase quản lý của Admin



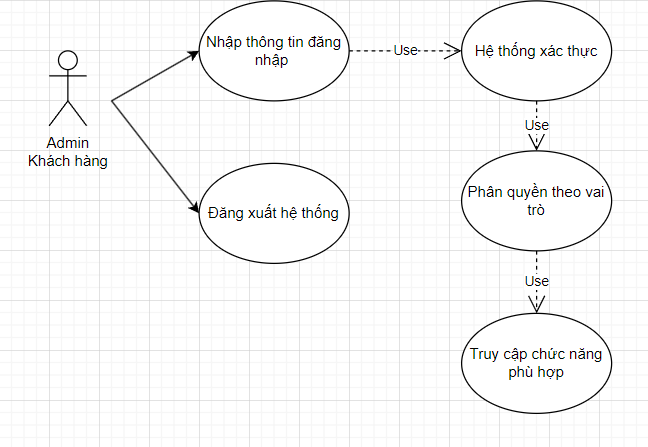
### 1.3 UseCase Quản lý bình luận đánh giá của khách hàng



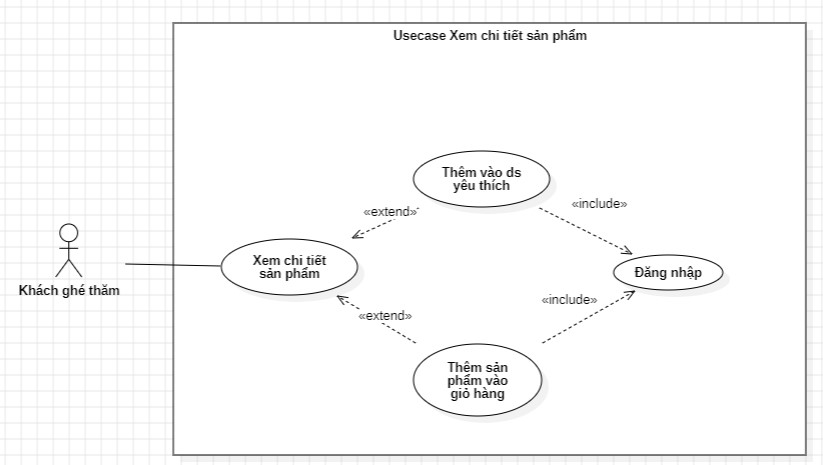
### 1.4 Usecase quản lý giỏ hàng



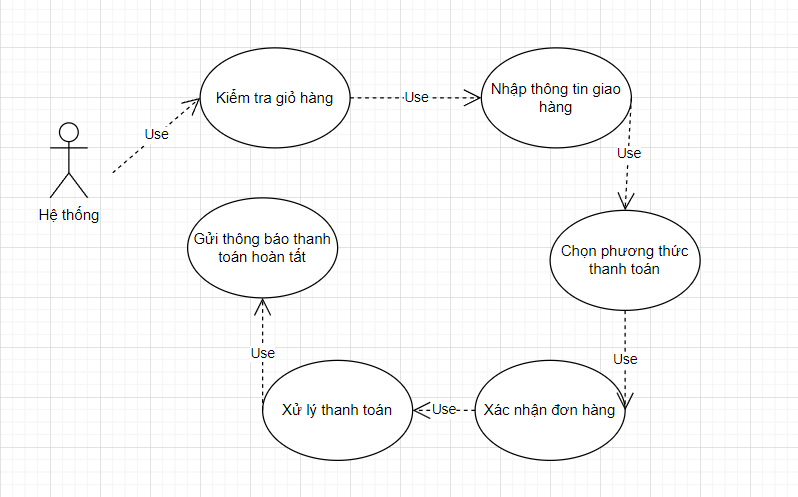
### 1.5 UseCase điều khiển truy cập



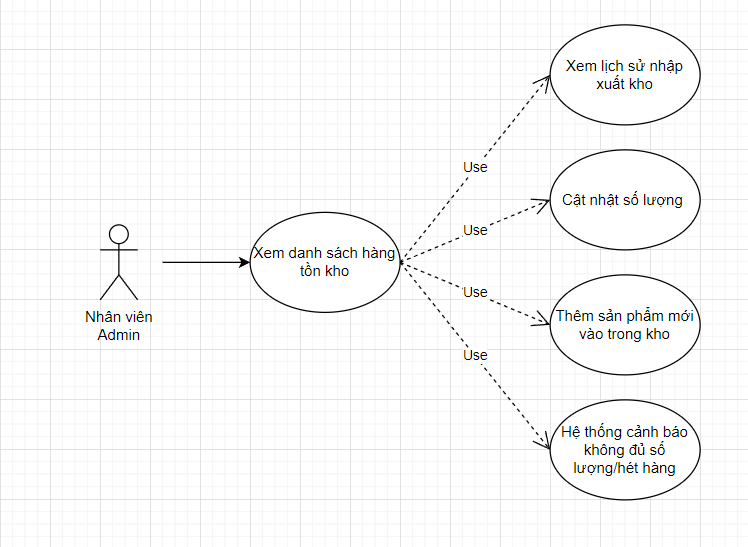
### 1.6 UseCase Xem chi tiết sản phẩm



### 1.7 Usecase thanh toán



### 1.8 Usecase hàng tồn kho



## **2. User story**

### ****2.1 Đánh giá của khách hàng****

1. **Là Người mua**, tôi muốn xem đánh giá của những khách hàng trước về sản phẩm để có thêm thông tin trước khi mua hàng.
2. **Là Người mua**, tôi muốn có thể đánh giá sản phẩm sau khi mua để chia sẻ trải nghiệm của mình với người khác.
3. **Là Người mua**, tôi muốn có thể chỉnh sửa hoặc xóa đánh giá của mình nếu tôi thay đổi ý kiến.
4. **Là Người mua**, tôi muốn có thể lọc sản phẩm theo đánh giá cao nhất/thấp nhất để tìm sản phẩm phù hợp.
5. **Là Admin**, tôi muốn có thể duyệt, ẩn hoặc xóa các đánh giá không phù hợp để đảm bảo chất lượng nội dung trên website.

### ****2.2 Danh mục sản phẩm****

1. **Là Người mua**, tôi muốn xem danh sách sản phẩm có chức năng phân loại theo danh mục để dễ dàng tìm kiếm sản phẩm mong muốn.
2. **Là Người mua**, tôi muốn có thể lọc sản phẩm theo danh mục, thương hiệu, giá cả, đánh giá để thu hẹp phạm vi lựa chọn.
3. **Là Admin**, tôi muốn có thể thêm, sửa hoặc xóa danh mục sản phẩm để quản lý hệ thống danh mục hiệu quả.
4. **Là Admin**, tôi muốn có thể đặt thứ tự ưu tiên hiển thị của các danh mục để tối ưu hóa trải nghiệm người dùng.

### 2.3 ****Chi tiết sản phẩm****

1. **Là Người mua**, tôi muốn xem chi tiết sản phẩm bao gồm hình ảnh, mô tả, giá, thông số kỹ thuật, đánh giá để đưa ra quyết định mua hàng.
2. **Là Người mua**, tôi muốn xem các sản phẩm liên quan hoặc sản phẩm cùng danh mục để có thêm lựa chọn.
3. **Là Người mua**, tôi muốn biết tình trạng hàng tồn kho của sản phẩm để đảm bảo tôi có thể đặt hàng.
4. **Là Admin**, tôi muốn có thể cập nhật thông tin sản phẩm như giá, mô tả, hình ảnh, số lượng tồn kho để đảm bảo thông tin chính xác.

### ****2.4 Giỏ hàng****

1. **Là Người mua**, tôi muốn thêm sản phẩm vào giỏ hàng để tiếp tục mua sắm trước khi thanh toán.
2. **Là Người mua**, tôi muốn có thể cập nhật số lượng sản phẩm trong giỏ hàng để điều chỉnh theo nhu cầu.
3. **Là Người mua**, tôi muốn có thể xóa sản phẩm khỏi giỏ hàng nếu tôi không muốn mua nữa.
4. **Là Người mua**, tôi muốn giỏ hàng của tôi được lưu ngay cả khi tôi thoát trang để không bị mất sản phẩm đã chọn.
5. **Là Người mua**, tôi muốn có thể xem tổng giá trị đơn hàng trước khi tiến hành thanh toán.

### ****2.5 Quyền truy cập****

1. **Là Người mua**, tôi muốn có thể đăng ký tài khoản để lưu thông tin cá nhân và lịch sử mua hàng.
2. **Là Người mua**, tôi muốn có thể đăng nhập bằng email/mật khẩu hoặc tài khoản Google/Facebook để truy cập nhanh chóng.
3. **Là Người mua**, tôi muốn có thể đặt hàng mà không cần tạo tài khoản để mua sắm thuận tiện hơn.
4. **Là Admin**, tôi muốn có thể phân quyền cho các tài khoản nhân viên với các vai trò khác nhau để kiểm soát quyền hạn truy cập.
5. **Là Admin**, tôi muốn có thể khóa hoặc mở khóa tài khoản người dùng nếu phát hiện hành vi vi phạm.

### ****2.6 Thanh toán****

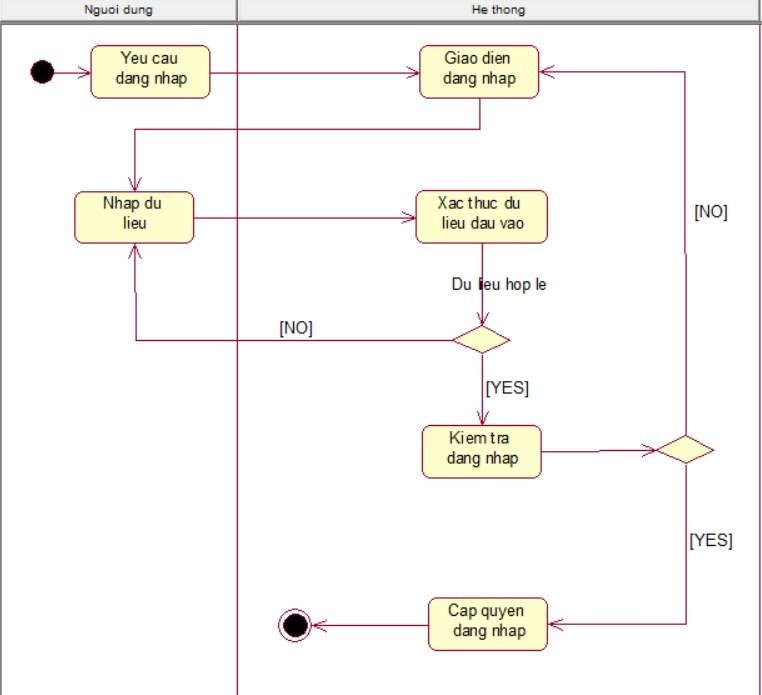
1. **Là Người mua**, tôi muốn có thể chọn nhiều phương thức thanh toán như thẻ tín dụng, ví điện tử hoặc thanh toán khi nhận hàng để thuận tiện hơn.
2. **Là Người mua**, tôi muốn nhận được email xác nhận sau khi thanh toán thành công để kiểm tra thông tin đơn hàng.
3. **Là Người mua**, tôi muốn có thể áp dụng mã giảm giá hoặc voucher khi thanh toán để nhận ưu đãi.
4. **Là Người mua**, tôi muốn thấy chi tiết phí vận chuyển và thời gian giao hàng ước tính trước khi thanh toán.
5. **Là Admin**, tôi muốn có thể quản lý các giao dịch thanh toán để theo dõi doanh thu và phát hiện giao dịch bất thường.

### ****2.7 Hàng tồn kho****

1. **Là Admin**, tôi muốn có thể theo dõi số lượng tồn kho của từng sản phẩm để tránh tình trạng hết hàng.
2. **Là Admin**, tôi muốn nhận thông báo khi một sản phẩm sắp hết hàng để nhập thêm kịp thời.
3. **Là Admin**, tôi muốn có thể cập nhật số lượng hàng tồn kho sau khi có đơn hàng mới hoặc hàng về kho.
4. **Là Admin**, tôi muốn xem báo cáo lịch sử nhập/xuất kho để theo dõi biến động hàng tồn kho.

**3. Biểu đồ hoạt động**

**3.1 Biểu đồ hoạt động đăng nhập, đăng xuất**



Biểu đồ hoạt động chức năng đăng nhập

**Mô tả hoạt động:**

Bước 1: Người dùng gửi yêu cầu đăng nhập vào hệ thống bằng cách click vào nút

“Đăng nhập”

Bước 2: Màn hình hiển thị giao diện đăng nhập

Bước 3: Người dùng nhập vào tài khoản và mật khẩu

Bước 4: Hệ thống kiểm tra tính hợp lệ của dữ liệu đầu vào:

* Dữ liệu hợp lệ ➔ Chuyển sang Bước 5
* Dữ liệu không hợp lệ ➔ Chuyển sang bước 3

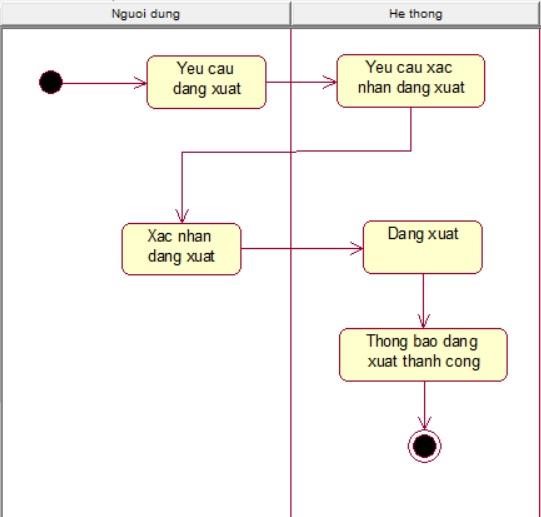
Bảng 3.2 Các trường hợp dữ liệu không hợp lệ

|  |  |
| --- | --- |
| **Dữ liệu không hợp lệ** | **Thông báo hệ thống** |
| Tài khoản chứa khoảng trắng | Tài khoản không được chứa khoảng  trắng |
| Mật khẩu chứa khoảng trắng | Mật khẩu không được chứa khoảng trắng |
| Tài khoản chứa kí tự đặc biệt | Tài khoản không được chứa kí tự đặc biệt |
| Tài khoản ít hơn 3 kí tự | Tài khoản tối thiểu 3 kí tự |
| Tài khoản nhiều hơn 50 kí tự | Tài khoản tối đa 50 kí tự |
| Mật khẩu ít hơn 6 kí tự | Mật khẩu tối thiểu 6 kí tự |
| Mật khẩu nhiều hơn 30 kí tự | Mật khẩu tối đa 30 kí tự |

Bước 5: Hệ thống kiểm tra thông tin đăng nhập trong CSDL:

* Nếu dữ liệu tồn tại trong CSDL ➔ Chuyển sang Bước 6
* Nếu dữ liệu không tồn tại trong CSDL ➔ thông báo “Sai tên tài khoản hoặc mật khẩu” và trở lại Bước 3

Bước 6: Cấp quyền đăng nhập cho người dùng



Biểu đồ hoạt động chức năng đăng xuất

**Mô tả hoạt động:**

Bước 1: Người dùng gửi yêu cầu đăng xuất khỏi hệ thống bằng cách click vào nút

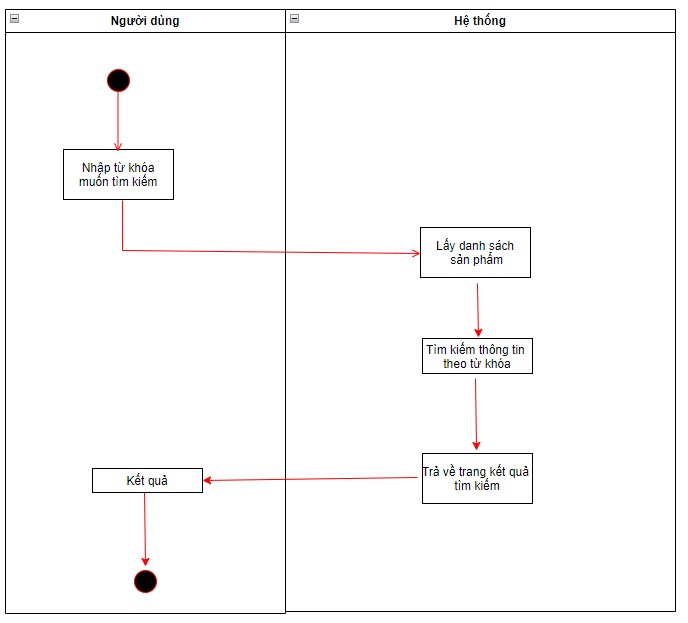
“Đăng xuất”

Bước 2: Hệ thống yêu cầu người dùng xác nhận đăng xuất

Bước 3: Người dùng xác nhận đăng xuất

Bước 4: Hệ thống thực hiện đăng xuất tài khoản người dùng ra khỏi hệ thống và thông báo “Đăng xuất thành công”

**3.2 Biểu đồ hoạt động tìm kiếm sản phẩm**



Biểu đồ hoạt động tìm kiếm sản phẩm

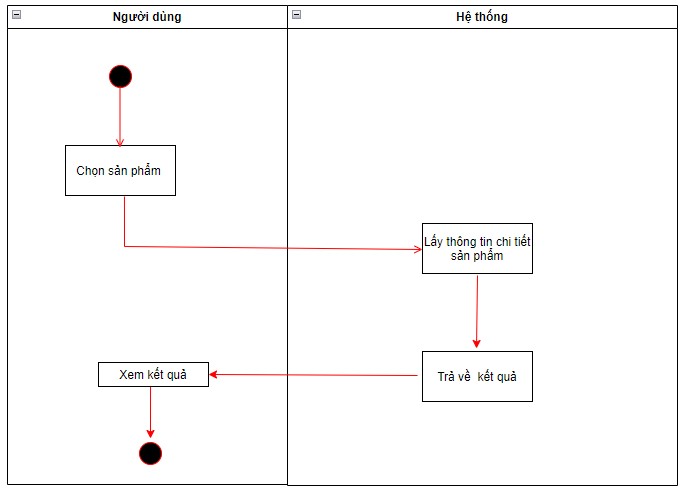
**Mô tả hoạt động:**

Bước 1: Người dùng nhập từ khóa tìm kiếm

Bước 2: Hệ thống lấy ra danh sách sản phẩm và đối chiếu với từ khóa người dùng nhập vào và đưa ra kết quả

Bước 3: Hiển thị kết quả ra màn hình

**3.3 Biểu đồ hoạt động xem chi tiết sản phẩm**



Biểu đồ hoạt động xem chi tiết sản phẩm

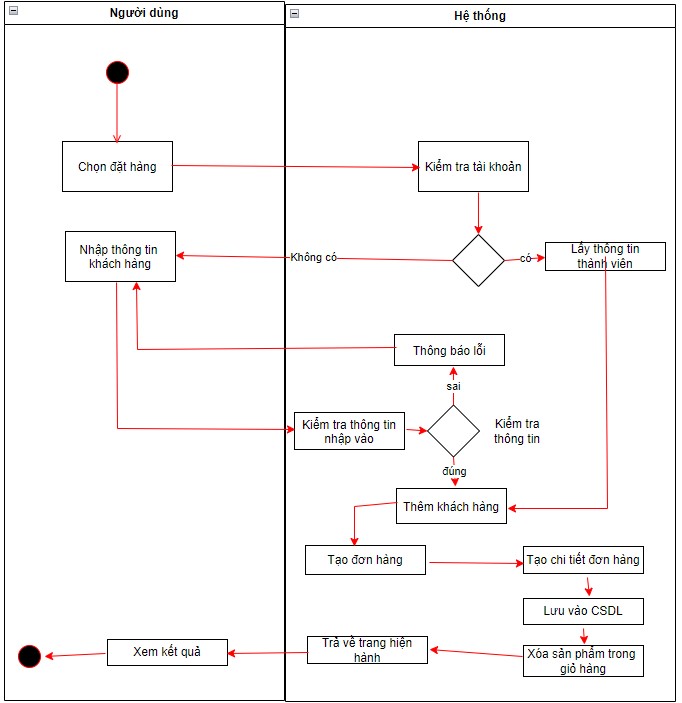
**Mô tả hoạt động:**

Bước 1: Người dùng chọn sản phẩm cần xem chi tiết

Bước 2: Hệ thống lấy dữ liệu và trả về kết quả

Bước 3: Hiển thị kết quả lên màn hình

**3.4 Biểu đồ hoạt động đặt hàng**



Biểu đồ hoạt động đặt hàng

**Mô tả hoạt động:**

Bước 1: Người dùng muốn đặt hàng thì click vào nút “Đặt hàng”

Bước 2: Nhập các thông tin cần thiết như họ tên, số điện thoại, địa chỉ,…

Bước 3: Hệ thống kiểm tra thông tin nhập vào

Nếu hợp lệ chuyển sang bước 4

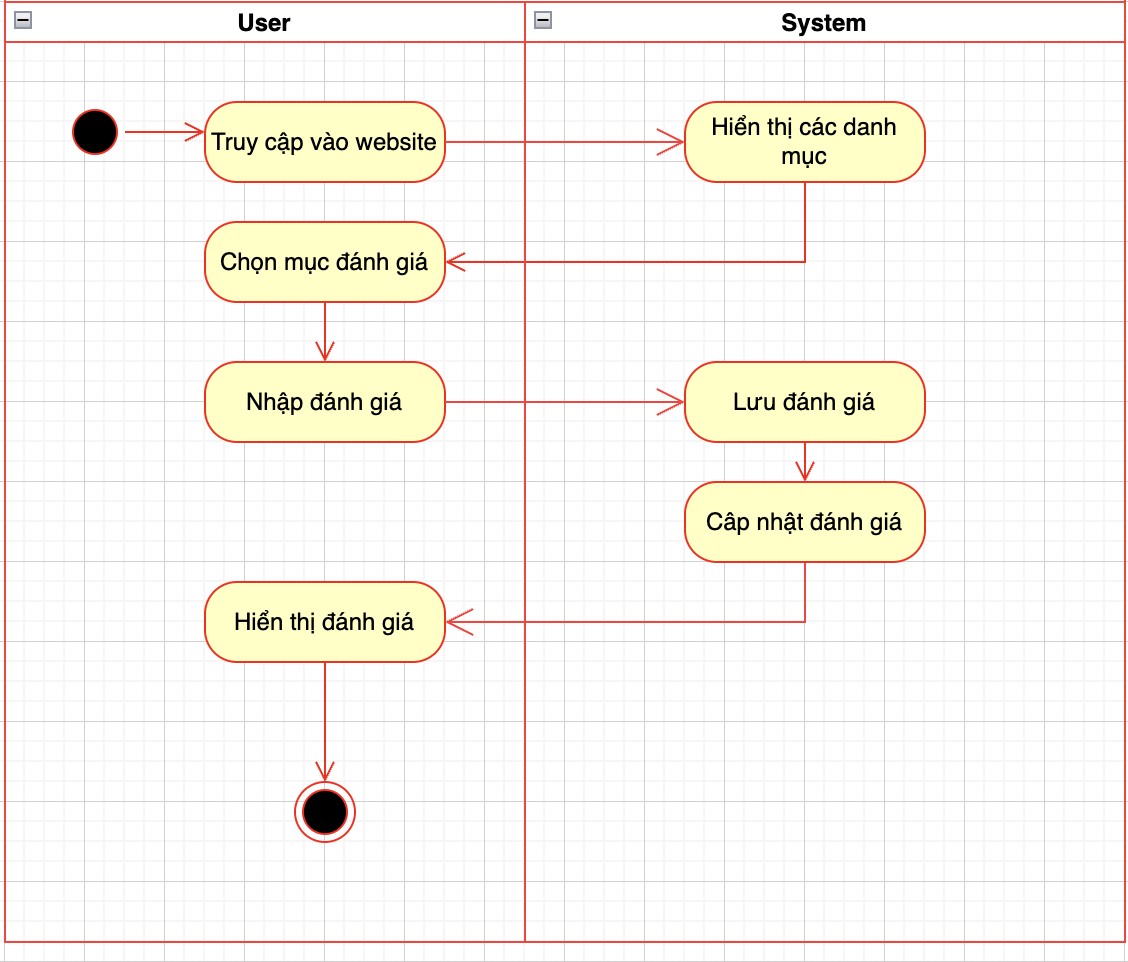
Nếu không hợp lệ chuyển về bước 2 và đưa ra thông báo

|  |  |
| --- | --- |
| **Dữ liệu không hợp lệ** | **Thông báo hệ thống** |
| Các ô dữ liệu để trống | Vui lòng không bỏ trống ô này |

Bước 4: Hệ thống tạo đơn hàng và lưu vào cơ sở dữ liệu

Bước 5: Hiển thị lên màn hình kết quả và đưa ra thông báo “Bạn đã đặt hàng thành công. Chúng tôi sẽ liên hệ trong 24h để xác nhận đơn hàng”

**3.5 Biểu đồ hoạt động đánh giá sản phẩm**



Biểu đồ hoạt động đánh giá sản phẩm

**Mô tả hoạt động:**

Use case cho phép người dùng truy cập vào website để viết bình luận đánh giá về chất lượng sản phẩm

* Dòng sự kiện khác: Không có.
* Yêu cầu đặc biệt: Không có.
* Tiền điều kiện: không có.
* Hậu điều kiện: Nếu use case thành công thì người dùng sẽ có quyền sử dụng hệ thống để viết bình luận, đánh giá
* Luồng sự kiện:

Sau khi truy cập website, hệ thống sẽ hiện ra mục đánh giá của khách hàng, những đánh giá của khách hàng về chất lượng và sản phẩm của web.

## **4. Thiết kế giao diện phần mềm:**

### 4.1 UI/UX

**4.1.1 Giao diện trang chủ**

### 

### 4.1.2 Giao diện trang sản phẩm

### 

### 4.1.3 Giao diện giỏ hàng

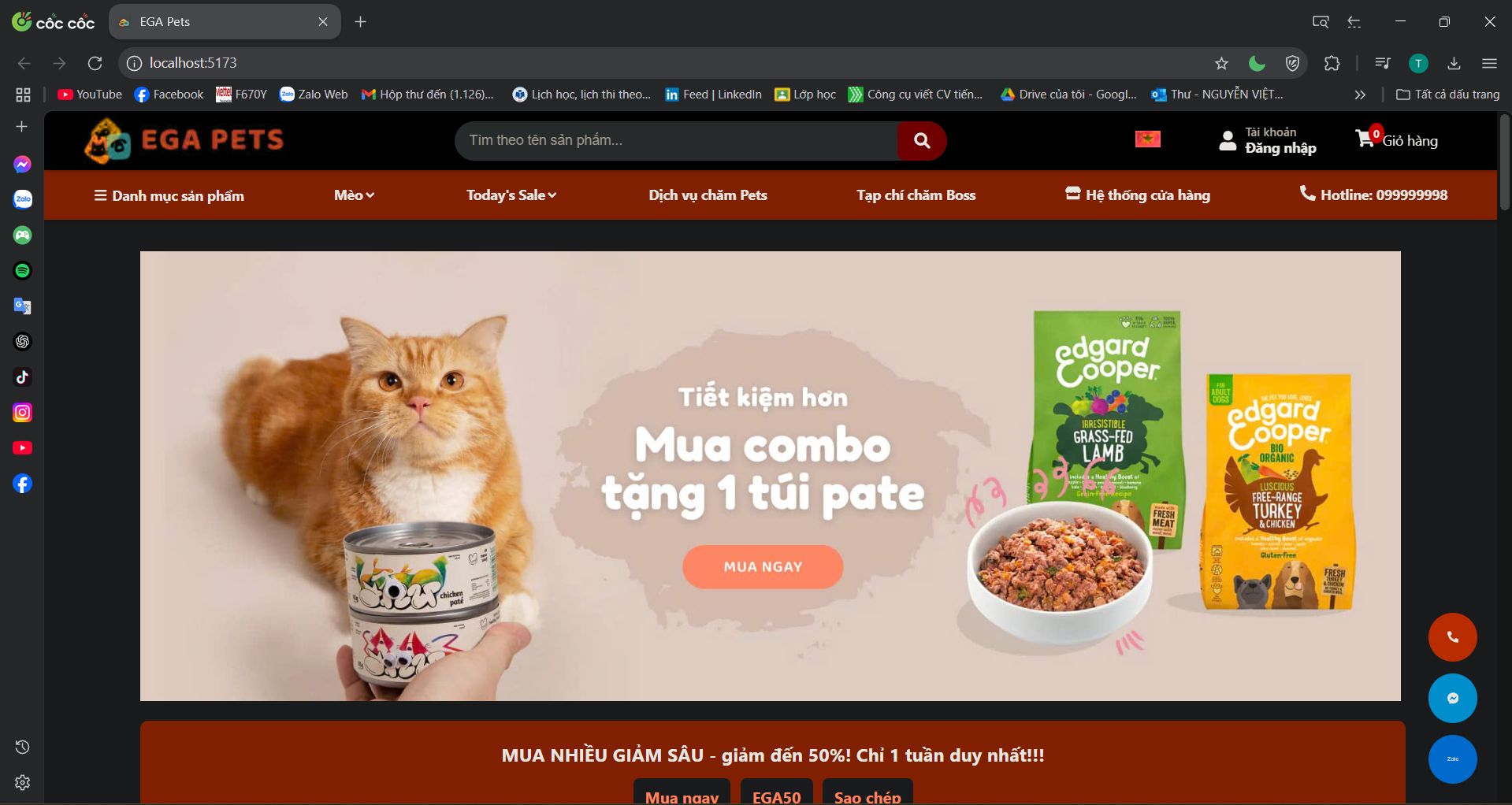
### 

### 4.1.4 Giao diện thanh toán

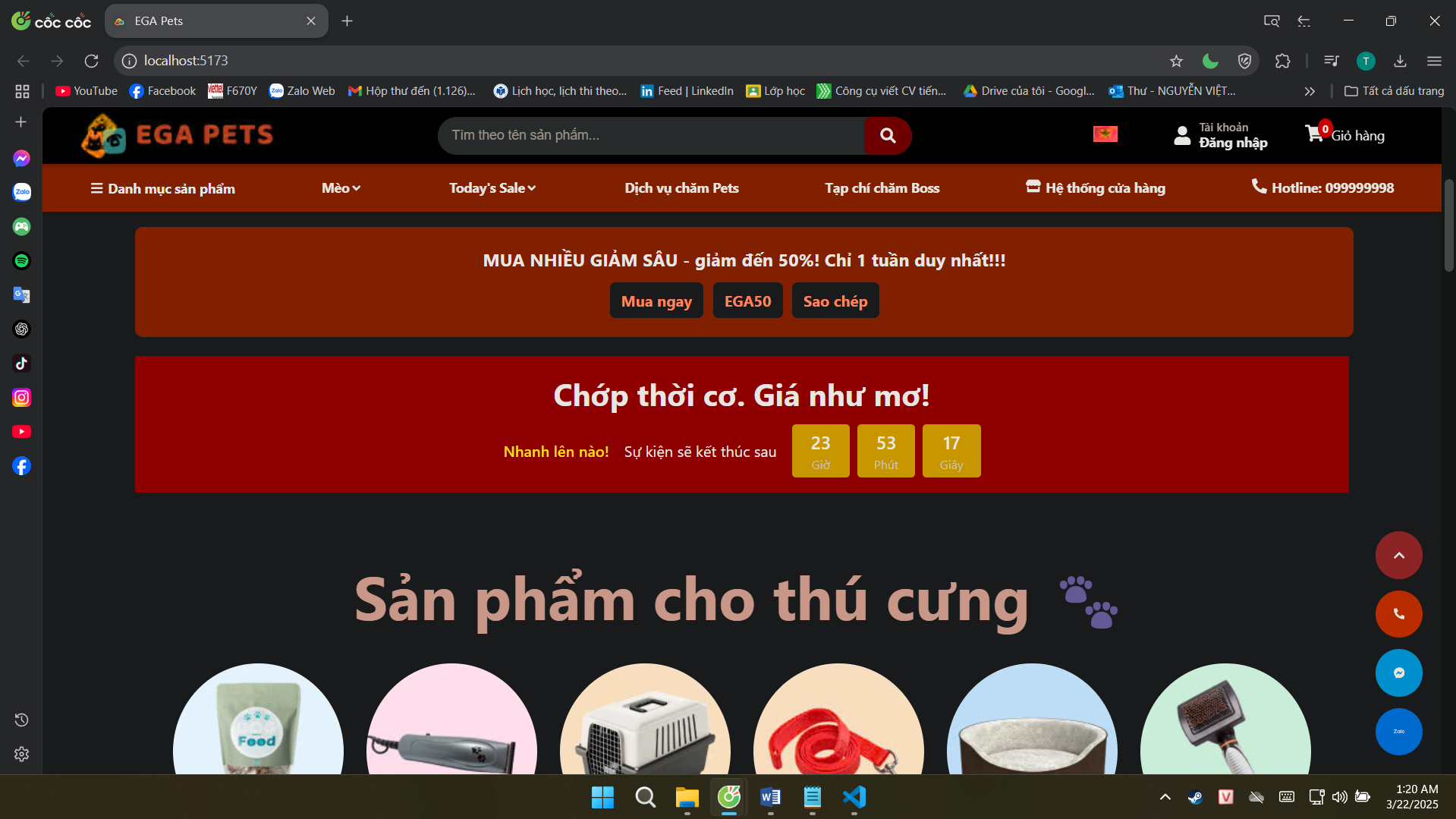
# **Chương 3 Thực hiện đồ án**

**1. Giao diện trang web**

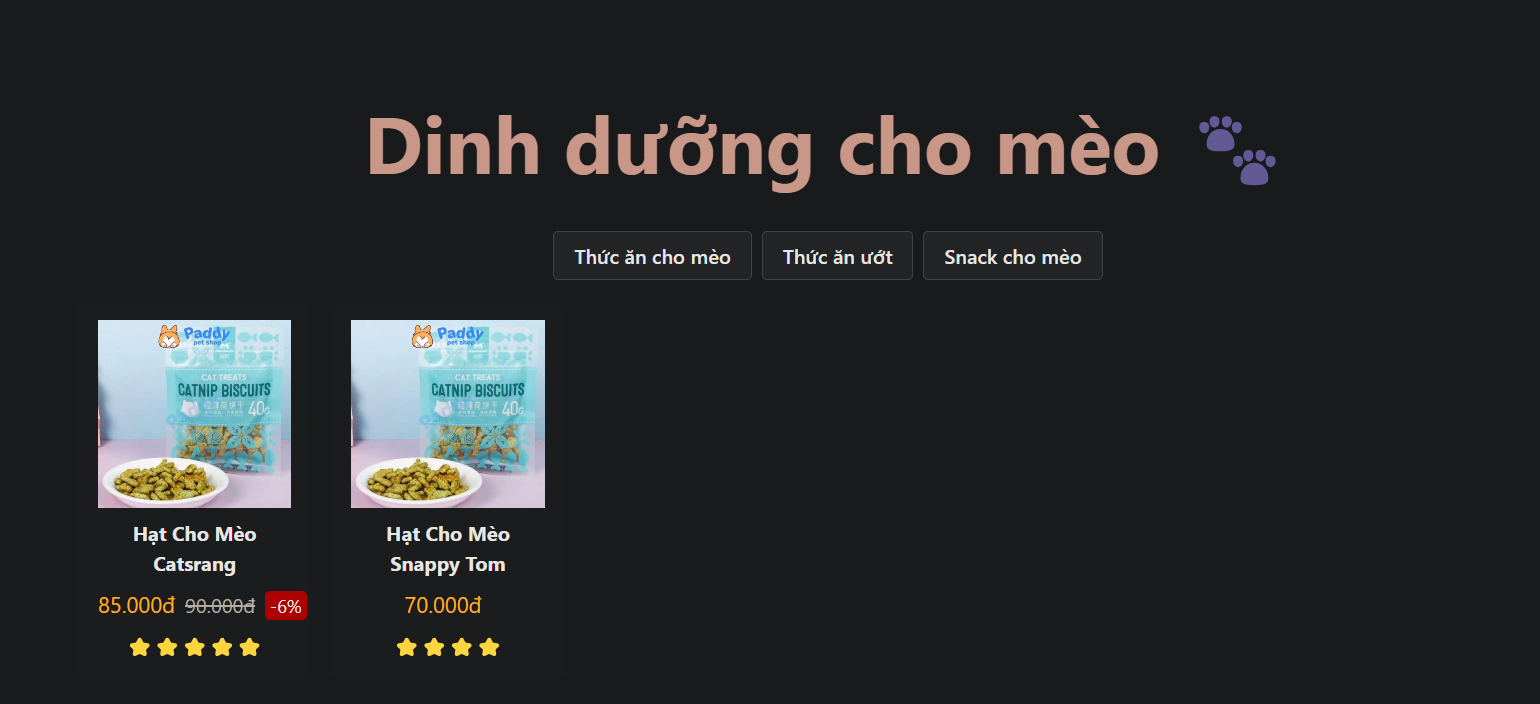
### 1.1 Giao diện phần trên

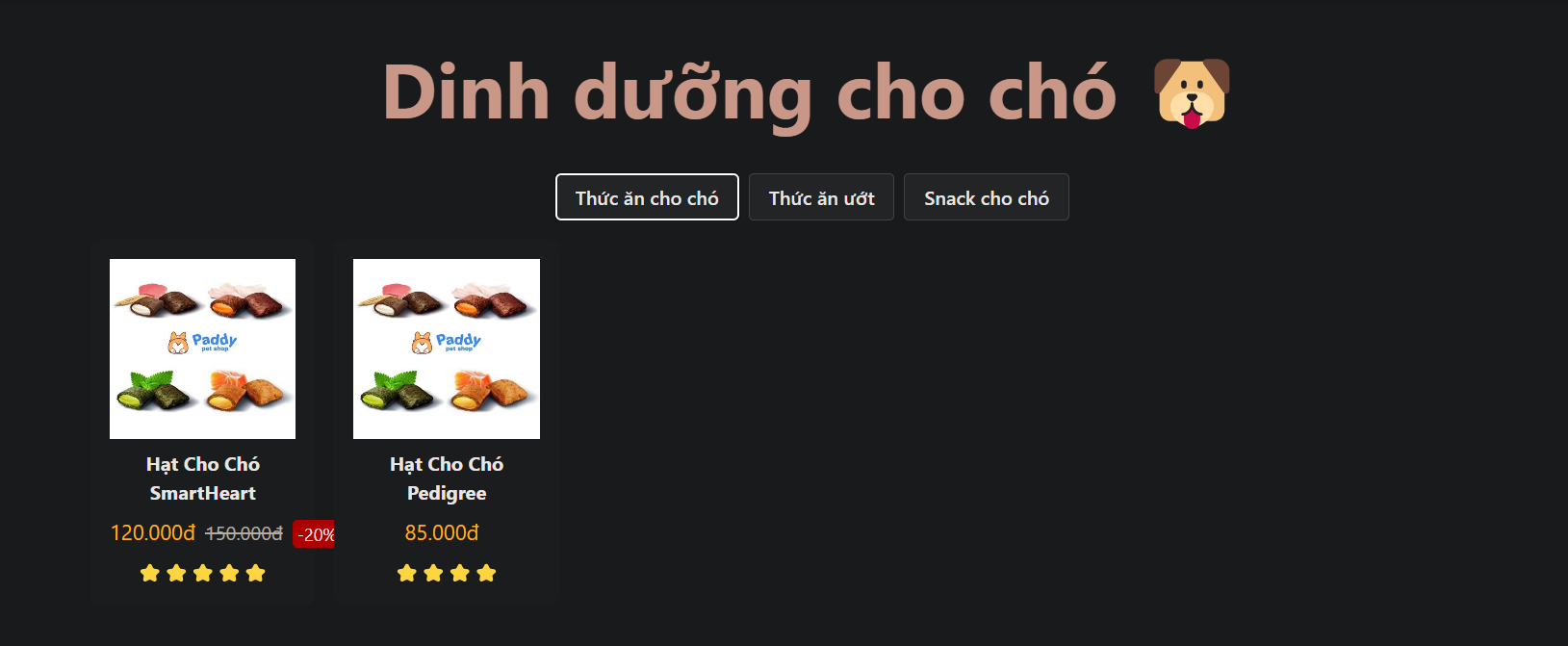


### 1.2 Giao diện phần dưới



**2. Giao diện trang sản phẩm**





**3. Giao diện trang khuyến mãi**



**4. Giao diện giỏ hàng**

## **Tài liệu tham khảo**

# **Chương 4 Báo cáo và trình bày**

## **1. Các công việc đã thực hiện**

## **2. Đánh giá mức độ hoàn thành**