|  |  |
| --- | --- |
|  |  |
| Nguyễn Văn Ngọc | **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **---------------------------------------** |
|  |
| ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC  NGÀNH CÔNG NGHỆ THÔNG TIN |
|  |
| **Xây dựng website bán nước hoa cho Shop NgocNguyen** |
|  |
|  |
| **CBHD: TS. Nguyễn Bá Nghiễn** |
| CÔNG NGHỆ THÔNG TIN | **Sinh viên: Nguyễn Văn Ngọc** |
| **Mã số sinh viên: 2019606788** |
|  |
|  |
|  |
| Hà Nội – Năm 2023 |
|  |

**MỤC LỤC**

[LỜI NÓI ĐẦU ii](#_Toc102342478)

[DANH MỤC CÁC CHỮ VIẾT TẮT iii](#_Toc102342479)

[DANH MỤC BẢNG BIỂU iv](#_Toc102342480)

[DANH MỤC HÌNH ẢNH v](#_Toc102342481)

[CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI 1](#_Toc102342482)

[1.1. Tên đề tài 1](#_Toc102342483)

[1.2. Lý do chọn đề tài 1](#_Toc102342484)

[1.3. Mục tiêu của đề tài 3](#_Toc102342485)

[1.4. Đối tượng và phạm vi 4](#_Toc102342486)

[1.5. Kết quả dự kiến đạt được 4](#_Toc102342487)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT VÀ CÔNG CỤ SỬ DỤNG 5](#_Toc102342488)

[2.1. Cơ sở lý thuyết 5](#_Toc102342489)

[2.2. Công cụ sử dụng 20](#_Toc102342490)

[CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 22](#_Toc102342491)

[3.1. Phân tích hệ thống 22](#_Toc102342492)

[3.2. Biểu đồ use case tổng quát 24](#_Toc102342493)

[3.3. Mô hình thực thể liên kết 25](#_Toc102342494)

[3.4. Đặc tả use case 26](#_Toc102342495)

[3.5. Thiết kế hệ thống 79](#_Toc102342496)

[KẾT LUẬN 94](#_Toc102342497)

[TÀI LIỆU THAM KHẢO 96](#_Toc102342498)

# LỜI NÓI ĐẦU

Ngày nay, việc ứng dụng công nghệ thông tin vào quản lý kinh doanh đã phát triển mạnh ở các nước tiên tiến trên thế giới. Đặc biệt trong mùa dịch, việc mua bán hàng là rất cần thiết. Trước tình hình đó, vấn đề áp dụng Tin học để quản lý, mua bán hàng dường như đã thay thế rất nhiều cho các phương thức mua hàng truyền thống, nổi trội như hàng loạt các ứng dụng mua hàng ra đời như Shopee, Lazada, Tiki.

Việc mua bán hàng online đang là một trong những nhu cầu hết sức thiết yếu hiện nay, đặc biệt là nhu cầu về mua sản phẩm, do đó hệ thống hỗ trợ mua những cuốn sản phẩm online cũng là một trong những vấn đề đang được quan tâm. Việc xây dựng website thương mại điện tử bán sản phẩm trực tuyến Bookshop góp phần giúp người dùng dễ dàng tìm hiểu và đặt mua những cuốn sản phẩm một cách dễ dàng.

Sau khi tìm hiểu và dưới sự hướng dẫn của thầy Nguyễn Bá Nghiễn, em đã phân tích và thiết kế được **Website bán nước hoa trực tuyến NgocNguyen**. Để giúp khách hàng đặt mua được những lọ nước hoa theo ý thích của mình, trước hết cần có một hệ thống tốt, một website có các dữ liệu chính xác, các đánh giá tốt của người dùng. Đó cũng là lý do mà em chọn đề tài này.

Để hoàn thành được đồ án tốt nghiệp này, em xin được gửi lời cảm ơn chân thành đến các thầy cô trong khoa Công nghệ thông tin Trường Đại học Công Nghiệp Hà Nội đã tận tình giảng dạy và trang bị kiến thức cho em trong suốt thời gian em học tập tại trường. Cô giáo hướng dẫn đề tài – **Tiến sỹ Nguyễn Bá Nghiễn**, giảng viên Khoa Công nghệ Thông tin Trường Đại học Công Nghiệp Hà Nội – đã tận tụy hết lòng giúp đỡ, hướng dẫn, chỉ dẫn tận tình để giúp em hoàn thành được đồ án. Bạn bè đã góp ý và giúp đỡ, bên cạnh động viên và khích lệ tôi trong suốt thời gian tôi làm đồ án.

***Hà Nội, Ngày 27 tháng 04 năm 2023***

# DANH MỤC CÁC CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| Từ viết tắt | Giải thích |
| API | Application Programming Interface |
| OOP | Object Oriented Programming |
| HTML | HyperText Markup Language |
| CSS | Cascading Style Sheets |
| UI | User Interface |
| GUI | Graphical User Interface |
| XML | Extensible Markup Language |
| PK | Primary key |
| PFK | Primary Foreign Key |

# DANH MỤC BẢNG BIỂU

[Bảng 3.1. Bảng Category 81](#_Toc102342540)

[Bảng 3.2. Bảng Product 81](#_Toc102342541)

[Bảng 3.3. Bảng ProductImage 82](#_Toc102342542)

[Bảng 3.4. Bảng Delivery 82](#_Toc102342543)

[Bảng 3.5. Bảng User 83](#_Toc102342544)

[Bảng 3.6. Bảng SaleOrder 83](#_Toc102342545)

[Bảng 3.7. Bảng OrderItem 84](#_Toc102342546)

# DANH MỤC HÌNH ẢNH

[Hình 2. 1 Logo C# 8](#_Toc134309629)

[Hình 2. 2 Logo Javascript 11](#_Toc134309630)

[Hình 2. 3 Logo VueJs 17](#_Toc134309631)

[Hình 2. 4 Logo MySql 21](#_Toc134309632)

[Hình 3. 1 Biểu đồ use case tổng quát 26](#_Toc134309636)

[Hình 3. 2 Mô hình thực thể liên kết 27](#_Toc134309637)

[Hình 3. 3 Biểu đồ use case đăng nhập 27](#_Toc134309638)

[Hình 3. 4 Biểu đồ trình tự đăng nhập 30](#_Toc134309639)

[Hình 3. 5 Biểu đồ hoạt động đăng nhập 31](#_Toc134309640)

[Hình 3. 6 Biểu đồ use case đăng ký 32](#_Toc134309641)

[Hình 3. 7 Biểu đồ trình tự đăng ký 33](#_Toc134309642)

[Hình 3. 8 Biểu đồ hoạt động đăng ký 34](#_Toc134309643)

[Hình 3. 9 Biểu đồ use case tìm kiếm 35](#_Toc134309644)

[Hình 3. 10 Biểu đồ trình tự tìm kiếm 36](#_Toc134309645)

# TỔNG QUAN VỀ ĐỀ TÀI

## Tên đề tài

**Xây dựng website bán nước hoa cho Shop NgocNguyen**

## Lý do chọn đề tài

Thương mại điện tử là một thuật ngữ rất phổ biến hiện nay. Nó được nhắc nhiều trên phương tiện truyền thông. Vậy thương mại điện tử là gì?

Theo tổ chức kinh tế thế giới định nghĩa: “Thương mại điện tử bao gồm việc sản xuất, quảng cáo, bán hàng và phân phối sản phẩm được mua bán và thanh toán trên mạng Internet, nhưng được giao nhận một cách hữu hình, cả các sản phẩm giao nhận cũng như những thông tin số hoá thông qua mạng Internet”.

Theo Ủy ban Thương mại điện tử của [Tổ chức Hợp tác kinh tế châu Á - Thái Bình Dương](https://vi.wikipedia.org/wiki/Di%E1%BB%85n_%C4%91%C3%A0n_H%E1%BB%A3p_t%C3%A1c_Kinh_t%E1%BA%BF_ch%C3%A2u_%C3%81_-_Th%C3%A1i_B%C3%ACnh_D%C6%B0%C6%A1ng) định nghĩa: "Thương mại điện tử liên quan đến các giao dịch thương mại trao đổi hàng hóa và dịch vụ giữa các nhóm (cá nhân) mang tính điện tử chủ yếu thông qua các hệ thống có nền tảng dựa trên Internet."

Thương mại điện tử có thể hiểu là thực hiện hoạt động thương mại bằng những phương tiện điện tử. Cụ thể hơn, đó là hoạt động giao dịch mua sắm qua internet. Đây là một phương thức kinh doanh sử dụng internet mang lại nhiều hiệu quả như giao dịch nhanh chóng, tiết kiệm chi phí mặt bằng và quản lý.

Ngày nay, các hoạt động thương mại điện tử hầu hết được thực hiện thông qua website bán hàng. Website bán hàng trở thành nơi quảng cáo, thực hiện giao dịch mua bán, thậm chí nguồn thu từ việc bán hàng qua website đã trở thành doanh số chính của nhiều công ty, doanh nghiệp. Website đã trở thành một phần không thể thiếu trong việc kinh doanh theo hình thức thương mại điện tử.

Với sự bùng nổ của cuộc cách mạng công nghiệp 4.0, internet đã có ảnh hưởng sâu sắc đến đời sống, kinh tế và xã hội. Có thể nói internet là một công cụ không thể thiếu trong cuộc sống. Để tồn tại và phát triển, con người cần phải học tập và tiếp cận nhanh chóng với những công nghệ mới, theo kịp sự phát triển của thời đại.

Trong nền kinh tế thị trường hiện nay, việc cạnh tranh giữa các cửa hàng công ty, doanh nghiệp ngày càng quyết liệt. Mục tiêu lớn nhất của cạnh tranh đó là thu hút nhiều khách hàng quan tâm đến sản phẩm của mình. Để thu hút được nhiều khách hàng, bên cạnh chất lượng sản phẩm và giá cả, các chiến lược kinh doanh, chiến lược marketing cũng rất quan trọng. Website trở thành công cụ, một phương tiện không thể thiếu để thực hiện marketing, quảng bá sản phẩm, thương hiệu của công ty, doanh nghiệp. Website trở thành một đường để doanh nghiệp giới thiệu sản phẩm ra thị trường trong nước, thậm chí ra ngoài quốc tế. Đây là nơi khách hàng tìm thấy thông tin giới thiệu về doanh nghiệp, các thông tin chi tiết về các sản phẩm. Nó cũng là một kênh giao tiếp, tư vấn trực tuyến với khách hàng, giải quyết kịp thời thắc mắc và vấn đề của khách hàng, giúp khách hàng hài lòng.

Thiết kế website là nhu cầu cần thiết đối với các công ty, doanh nghiệp. Nó là hình ảnh, bộ mặt của công ty, doanh nghiệp. Website càng tốt càng mang lại trải nghiệm tốt cho khách hàng, càng hướng khách hàng tới sản phẩm của mình. Website có ảnh hưởng rất lớn tới việc kinh doanh của công ty, doanh nghiệp như doanh thu, chi phí quản lý, sự hài lòng và đánh giá của khách.

Nước hoa là một loại hương liệu được sử dụng để tạo ra mùi thơm dành cho cơ thể. Nó đã trở thành một phần quan trọng của văn hóa và thường được sử dụng để cải thiện hình ảnh cá nhân của một người, tạo ra sự tự tin và ấn tượng tốt với người khác. Ngoài ra, nước hoa còn có khả năng kích thích các giác quan của con người, giúp giảm stress, tạo cảm giác thoải mái và tăng cường tinh thần.

Nước hoa cũng có thể được sử dụng để tạo ra một không gian thơm ngát, dễ chịu và trang trí cho nhà cửa, văn phòng hoặc cửa hàng. Đối với một số ngành công nghiệp, như công nghiệp mỹ phẩm và chăm sóc sức khỏe, nước hoa là một phần quan trọng trong sản xuất các sản phẩm liên quan đến làm đẹp và chăm sóc sức khỏe.

Với các thương hiệu nước hoa nổi tiếng, nước hoa không chỉ đơn giản là một sản phẩm mà còn là một biểu tượng của đẳng cấp và phong cách. Việc sử dụng nước hoa đã trở thành một phần không thể thiếu của nhiều người, đặc biệt là những người yêu thích thời trang và phong cách sống hiện đại.

Đối với website hàng nói chung và website bán nước hoa nói riêng, bên cạnh giao diện đẹp thì website hoạt động như thế nào là rất quan trọng. Khách hàng khi ghé thăm website thì khách hàng sẽ giao tiếp và thao tác với giao diện của website đó. Giao diện website phải đẹp, phải gây ấn tượng với khách hàng và cũng sẽ tạo nên điểm đặc biệt của thương hiệu nước hoa. Khách hàng nhìn vào website có thể dễ dàng sử dụng các công cụ có trên đó. Về hoạt động của website thì website phải chạy mượt mà để khách hàng thoải mái, không bị khó chịu khi sử dụng. Website giúp khán giả có thể xem được thông tin mình cần, có thể tìm tới sản phẩm mình cần. Nó chính là bộ mặt của thương hiệu, là phương pháp gây ấn tượng và thu hút khách hàng.

Nhận thức được sự quan trọng và cũng như để có thể áp dụng được những kiến thức đã được học và tìm hiểu, em xin được áp dụng những kiến thức đã được học và tìm hiểu đó để xây dựng một website cụ thể là: ***“*Xây dựng website bán nước hoa cho Shop NgocNguyen*”.***

## Mục tiêu của đề tài

Đề tài: **Xây dựng website bán nước hoa cho Shop NgocNguyen** đáp ứng được những mục tiêu:

* Hiểu biết những kiến thức cần thiết về xây dựng giao diện trang web, phân tích thiết kế hệ thống, cơ sở dữ liệu.
* Nắm được các kiến thức cơ bản về VueJs, Javascript, MySql, C#, ASP.net..
* Lợi ích đầu tiên khi sở hữu một website bán hàng đó chính là việc tiếp cận được những khách hàng tiềm năng. Dù khách hàng có ở bất cứ đâu trên thế giới, chỉ cần có mạng internet là có thể tìm thấy website bán hàng, biết đến sản phẩm và thương hiệu của công ty. Từ đó, chắc chắn rằng lượng khách hàng sẽ tăng lên đáng kể.
* Tiết kiệm được thời gian cho khách hàng, khách hàng không cần phải đến tận cửa hàng mới mua được hàng mà họ có thể thông qua website xem và mua hàng. Hơn nữa mặt hàng khách mua còn có thể được chuyển về địa chỉ yêu cầu đúng thời gian khách hàng mong muốn, tiết kiệm được cả chi phí đi lại cho khách hàng.
* Thông tin sản phẩm được cập nhật nhanh chính xác qua từng ngày đến khách hàng**.**
* Xây dựng và quảng bá thương hiệu cho hãng.

## Đối tượng và phạm vi

Bất cứ ai có nhu cầu mua sản phẩm trực tuyến trên nền tảng internet.

## Kết quả dự kiến đạt được

Website bao gồm những mục chính và chức năng chính:

* Đăng ký, đăng nhập tài khoản cá nhân.
* Thay đổi mật khẩu, cập nhật thông tin tài khoản.
* Quên mật khẩu của tài khoản.
* Phân quyền giữa người quản trị và khách hàng.
* Chức năng quản trị: quản lý tài khoản, danh mục sản phẩm, sản phẩm, đơn hàng, hình ảnh sản phẩm.
* Tìm kiếm sản phẩm, tìm kiếm sản phẩm theo chủ đề, xem chi tiết sản phẩm.
* Thêm sản phẩm vào giỏ hàng và lưu trữ giỏ hàng của mỗi người dùng.
* Đặt hàng, thanh toán trực tuyến.

# CƠ SỞ LÝ THUYẾT VÀ CÔNG CỤ SỬ DỤNG

## Cơ sở lý thuyết

### Ngôn ngữ UML

UML là viết tắt của “Unified Modeling Language” - là ngôn ngữ mô hình hóa thống nhất, dùng để đặc tả, hình dung, xây dựng và tư liệu hóa phần mềm hướng đối tượng.

UML dùng để:

* Trực quan hoá: Đưa ra các mô hình trực quan về hệ thống để mọi người có thể hiểu được.
* Đặc tả: UML cho phép xây dựng các mô hình đúng đắn và đầy đủ để mô tả hệ thống.
* Xây dựng: Các mô hình UML có thể liên kết trực tiếp với nhiều ngôn ngữ lập trình Java, C++, VB, ...
* Lập tài liệu: UML cho phép tạo các tài liệu như tài liệu về các yêu cầu của người dùng, kiến trúc hệ thống, thiết kế hệ thống, kiểm thử, kế hoạch dự án, ...

UML có thể được sử dụng làm công cụ giao tiếp giữa người dùng, nhà phân tích, nhà thiết kế và nhà phát triển phần mềm.

Các phần tử của UML:

View (Quan sát): Theo các phương diện khác nhau của hệ thống cần phân tích, thiết kế. Dựa vào các quan sát để thiết lập kiến trúc cho hệ thống cần phát triển.

***Có 9 loại biểu đồ:***

* Use Case Diagram (Biểu đồ ca sử dụng): Mô tả sự tương tác giữa các tác nhân và hệ thống thông qua các ca sử dụng.
* Class Diagram (Biểu đồ lớp): Biểu đồ lớp chỉ ra sự tồn tại giữa các lớp và mối quan hệ giữa chúng trong thiết kế logic trong hệ thống. Khung nhìn tĩnh của hệ thống chủ yếu hỗ trợ các chức năng của hệ thống.
* Object Diagram (Biểu đồ đối tượng): Biểu đồ đối tượng đại diện cho một thể hiện của biểu đồ lớp, cho nên các khái niệm cơ bản là giống biểu đồ lớp. Biểu đồ đối tượng được sử dụng để hiện thị một tập hợp các đối tượng và các mối quan hệ của chúng như một thể hiện.
* Sequence Diagram (Biểu đồ trình tự): Là biểu đồ để xác định và chỉ rõ vai trò của đối tượng tham gia vào luồng sự kiện của use case.
* Collaboration Diagram (Biểu đồ cộng tác): Tương tự như biểu đồ trình tự nhưng nhấn mạnh vào sự tương tác của các đối tượng trên cơ sở cộng tác với nhau bằng cách trao đổi các thông điệp để thực hiện các yêu cầu theo ngữ cảnh công việc.
* State Diagram (Biểu đồ trạng thái): Thể hiện chu kỳ hoạt động của các đối tượng, của các hệ thống con và của cả hệ thống.
* Activity Diagram (Biểu đồ hành động): Được sử dụng để mô tả các hoạt động và các hành động được thực hiện trong một use case.
* Component Diagram (Biểu đồ thành phần): Chỉ ra cấu trúc vật lý của các thành phần trong hệ thống.
* Deployment Diagram (Biểu đồ triển khai): Chỉ ra cách bố trí vật lý các thành phần theo kiến trúc được thiết kế của hệ thống.

Relationship (Quan hệ):

* Realization (Hiện thực hoá):Là quan hệ ngữ nghĩa giữa giao diện và lớp.
* Dependency (Phụ thuộc):Là quan hệ ngữ nghĩa giữa hai phần tử, trong đó sự thay đổi của một tử sẽ tác động đến ngữ nghĩa của phần tử phụ thuộc.
* Generalization (Kế thừa): Là quan hệ mô tả sự khái quát hoá mà trong đó một số đối tượng cụ thể (của lớp con) sẽ được kế thừa các thuộc tính, các phương thức của các đối tượng tổng quát (lớp cơ sở).
* Association (Kết hợp): Là quan hệ cấu trúc xác định mối liên kết giữa các lớp đối tượng.

### Ngôn ngữ C#



Hình 2. 1 Logo C#

C# là ngôn ngữ lập trình hướng đối tượng, hướng thành phần. C# cung cấp các cấu trúc ngôn ngữ để hỗ trợ trực tiếp các khái niệm này, làm cho C# trở thành ngôn ngữ tự nhiên để tạo và sử dụng các thành phần phần mềm. Kể từ khi ra đời, C# đã bổ sung các tính năng để hỗ trợ khối lượng công việc mới và các phương pháp thiết kế phần mềm mới nổi. Về cốt lõi, C# là một ngôn ngữ hướng đối tượng. Bạn xác định các loại và hành vi của họ.

Một số tính năng của C# giúp tạo ra các ứng dụng mạnh mẽ và lâu bền. Bộ sưu tập rác tự động lấy lại bộ nhớ bị chiếm bởi các đối tượng không sử dụng không thể truy cập. Các loại nullable bảo vệ chống lại các biến không tham chiếu đến các đối tượng được phân bổ. Xử lý ngoại lệ cung cấp một cách tiếp cận có cấu trúc và có thể mở rộng để phát hiện và khôi phục lỗi. Biểu thức lambda hỗ trợ các kỹ thuật lập trình chức năng. Cú pháp Truy vấn Tích hợp Ngôn ngữ (LINQ) tạo ra một mẫu chung để làm việc với dữ liệu từ bất kỳ nguồn nào. Hỗ trợ ngôn ngữ cho các hoạt động không đồng bộ cung cấp cú pháp để xây dựng các hệ thống phân tán. C# có một hệ thống kiểu thống nhất. Tất cả các kiểu C#, bao gồm các kiểu nguyên thủy như int và double, kế thừa từ một kiểu đối tượng gốc duy nhất. Tất cả các loại chia sẻ một tập hợp các hoạt động chung. Các giá trị thuộc bất kỳ loại nào đều có thể được lưu trữ, vận chuyển và vận hành theo một cách nhất quán. Hơn nữa, C# hỗ trợ cả kiểu tham chiếu và kiểu giá trị do người dùng định nghĩa. C# cho phép phân bổ động các đối tượng và lưu trữ trực tuyến các cấu trúc nhẹ. C# hỗ trợ các phương thức và kiểu chung, giúp tăng hiệu suất và độ an toàn của kiểu. C# cung cấp các trình vòng lặp, cho phép những người triển khai các lớp tập hợp xác định các hành vi tùy chỉnh cho mã máy khách.

C# nhấn mạnh việc lập phiên bản để đảm bảo các chương trình và thư viện có thể phát triển theo thời gian theo cách tương thích. Các khía cạnh của thiết kế C# bị ảnh hưởng trực tiếp bởi các cân nhắc về phiên bản bao gồm các công cụ sửa đổi ghi đè và ảo riêng biệt, các quy tắc để giải quyết quá tải phương thức và hỗ trợ cho các khai báo thành viên giao diện rõ ràng.

**Các đặc điểm của C#:**

**Hướng đối tượng**: C# hỗ trợ lập trình hướng đối tượng, cho phép lập trình viên xây dựng các ứng dụng theo kiểu đối tượng, tạo ra các đối tượng độc lập và sử dụng chúng để tạo ra các ứng dụng phức tạp.

**Thông dịch**: C# được biên dịch thành một ngôn ngữ trung gian gọi là MSIL (Microsoft Intermediate Language), sau đó sử dụng trình thông dịch .NET Framework để chạy ứng dụng. Điều này cho phép các ứng dụng C# chạy trên nhiều nền tảng khác nhau, bao gồm Windows, Linux và Mac OS.

**An toàn kiểu dữ liệu**: C# là một ngôn ngữ kiểm tra kiểu dữ liệu tại thời điểm biên dịch và thời gian chạy. Điều này đảm bảo rằng các lỗi kiểu dữ liệu được phát hiện và xử lý trước khi ứng dụng được thực thi.

**Tính năng xử lý ngoại lệ**: C# hỗ trợ xử lý ngoại lệ (exception handling), cho phép lập trình viên xử lý các tình huống bất thường và xử lý các lỗi trong quá trình thực thi của ứng dụng.

**Sử dụng dễ dàng**: C# được thiết kế để dễ hiểu và sử dụng, đồng thời cũng cung cấp nhiều công cụ hỗ trợ phát triển, bao gồm Visual Studio và .NET Framework.

Cú pháp tương tự với C++ và Java: Cú pháp của C# rất giống với cú pháp của C++ và Java, điều này giúp cho lập trình viên dễ dàng chuyển đổi giữa các ngôn ngữ này và học C# một cách nhanh chóng.

### File:Unofficial JavaScript logo 2.svg - Wikimedia CommonsNgôn ngữ JavaScript

Hình 2. 2 Logo Javascript

JavaScript là một ngôn ngữ lập trình đa nền tảng (cross-platform), ngôn ngữ lập trình kịch bản, hướng đối tượng. JavaScript là một ngôn ngữ nhỏ và nhẹ (small and lightweight). JavaScript chứa các thư viện tiêu chuẩn cho các đối tượng (object), ví dụ như: Array, Date, Math, và các yếu tố cốt lõi của ngôn ngữ lập trình như: toán tử (operators), cấu trúc điều khiển (control structures), và câu lệnh. JavaScript có thể được mở rộng cho nhiều mục đích bằng việc bổ sung thêm các object; ví dụ:

* Client-side JavaScript - JavaScript phía máy khách, JavaScript được mở rộng bằng cách cung cấp các object để quản lý trình duyệt và Document Object Model (DOM) của nó. Ví dụ, phần mở rộng phía máy khách cho phép một ứng dụng tác động tới các yếu tố trên một trang HTML và phản hồi giống các tác động của người dùng như click chuột, nhập form, và chuyển trang.
* Server-side JavaScript - JavaScript phía máy chủ, JavaScript được mở rộng bằng cách cung cấp thêm các đối tượng cần thiết để để chạy JavaScript trên máy chủ. Ví dụ, phần mở rộng phía server này cho phép ứng dụng kết nối với cơ sở dữ liệu (database), cung cấp thông tin một cách liên tục từ một yêu cầu tới phần khác của ứng dụng, hoặc thực hiện thao tác với các tập tin trên máy chủ.

JavaScript được phát triển bởi Brendan Eich, một nhân viên của Netscape, vào tháng 9 năm 1995. Được biết đến lần đầu tiên là Mocha, rồi sau đó được đổi tên thành LiveScript và cuối cùng là JavaScript nổi tiếng như hiện giờ. Phiên bản đầu tiên của ngôn ngữ này bị giới hạn độc quyền bởi Netscape và chỉ có các tính năng hạn chế, nhưng nó tiếp tục phát triển theo thời gian.

Năm 1996, JavaScript được chính thức đặt tên là ECMAScript. ECMAScript 2 phát hành năm 1998 và ECMAScript 3 tiếp tục ra mắt vào năm 1999. Nó liên tục phát triển thành JavaScript ngày nay, giờ đã hoạt động trên khắp mọi trình duyệt và trên khắp các thiết bị từ di động đến máy tính bàn.

JavaScript liên tục phát triển kể từ đó, có lúc đạt đến 92% website đang sử dụng JavaScript vào năm 2016. Chỉ trong 20 năm, nó từ một ngôn ngữ lập trình riêng trở thành công cụ quan trọng nhất trên bộ công cụ của các chuyên viên lập trình web.

JavaScript cùng với HTML, CSS đã trở thành ngôn ngữ không thể thiếu trong lập trình web hiện nay:

* Với việc phát triển mạnh mẽ của công nghệ, JavaScript đã dần trở thành một ngôn ngữ đa nền tảng với nhiều ứng dụng khác nhau.
* Các Framework, Library cho lập trình front-end mạnh mẽ hỗ trợ xây dựng các trang Single-page application (SPA) như: ReactJS, Angular, VueJS.
* Hỗ trợ lập trình mobile bằng thư viện React Native.
* Hỗ trợ lập trình Window bằng Electron.
* Không chỉ phát triển ở khía cạnh lập trình phía client-side, JavaScript cũng hỗ trợ viết server-side rất mạnh mẽ với NodeJS.

Với hệ sinh thái mã nguồn mở khổng lồ cùng sự phát triển như hiện nay, JavaScript được dự đoán sẽ phát triển mạnh mẽ hơn nữa trong những năm tới.

### ASP.NET Web API

#### ASP.NET Web API là gì?

ASP.NET Web API là một framework phát triển dịch vụ web (web service) được xây dựng trên nền tảng ASP.NET, được sử dụng để xây dựng các dịch vụ web RESTful. REST (Representational State Transfer) là một kiểu kiến trúc phần mềm được sử dụng rộng rãi để phát triển các dịch vụ web.

#### Lịch sử phát triển của ASP.NET Web API

ASP.NET Web API được giới thiệu lần đầu tiên vào năm 2012, như một phần của phiên bản ASP.NET 4.0. Ban đầu, nó được phát triển như là một framework để xây dựng các dịch vụ web RESTful, nhằm cung cấp một cách tiếp cận linh hoạt và dễ dàng để phát triển các ứng dụng web dựa trên các dịch vụ web.

Với sự phát triển nhanh chóng của công nghệ web và sự phổ biến của các thiết bị di động, ASP.NET Web API đã được cải tiến để hỗ trợ các tính năng như hỗ trợ cho các định dạng dữ liệu khác nhau như JSON và XML, chuyển đổi các đối tượng .NET sang định dạng dữ liệu phổ biến, ràng buộc dữ liệu và định tuyến URL.

ASP.NET Web API cũng được tích hợp với ASP.NET MVC (Model-View-Controller), cho phép lập trình viên xây dựng các dịch vụ web RESTful bằng cách sử dụng kiến trúc MVC.

Với sự phát triển và cải tiến liên tục, hiện tại ASP.NET Web API đã trở thành một phần quan trọng trong các ứng dụng web hiện đại, đặc biệt là các ứng dụng web đa nền tảng (cross-platform) và các ứng dụng di động.

#### Mục tiêu của ASP.NET Web API

Mục tiêu chính của ASP.NET Web API là cung cấp một cách tiếp cận dễ dàng và linh hoạt để xây dựng các dịch vụ web RESTful. Các dịch vụ này có thể được sử dụng bởi các ứng dụng web, ứng dụng di động và các ứng dụng khác để truy cập và thao tác với các tài nguyên khác nhau trên máy chủ.

Cụ thể, một số mục tiêu của ASP.NET Web API bao gồm:

Hỗ trợ cho các dịch vụ web RESTful: ASP.NET Web API cung cấp các tiện ích và tính năng để xây dựng các dịch vụ web theo kiến trúc RESTful, giúp lập trình viên dễ dàng thiết kế và triển khai các dịch vụ web.

Hỗ trợ đa nền tảng: Các dịch vụ web xây dựng trên ASP.NET Web API có thể được sử dụng trên nhiều nền tảng khác nhau, bao gồm cả các ứng dụng web và các ứng dụng di động.

Tích hợp với công nghệ ASP.NET: ASP.NET Web API được tích hợp một cách tốt nhất với công nghệ ASP.NET, giúp lập trình viên xây dựng các ứng dụng web đa nền tảng dễ dàng hơn.

Hỗ trợ các định dạng dữ liệu khác nhau: ASP.NET Web API hỗ trợ chuyển đổi các đối tượng .NET sang các định dạng dữ liệu phổ biến như JSON, XML và các định dạng khác, giúp lập trình viên dễ dàng trao đổi dữ liệu giữa các ứng dụng khác nhau.

Đảm bảo an ninh và quản lý: ASP.NET Web API hỗ trợ các tính năng bảo mật như xác thực, phân quyền và quản lý lỗi để đảm bảo an toàn và bảo mật cho các dịch vụ web.

Top of Form

Bottom of Form

#### Ưu điểm khi sử dụng ASP.NET Web API

Đảm bảo an ninh và quản lý: ASP.NET Web API hỗ trợ các tính năng bảo mật như xác thực, phân quyền và quản lý lỗi để đảm bảo an toàn và bảo mật cho các dịch vụ web.

Tính đa nền tảng: ASP.NET Web API cho phép bạn xây dựng các dịch vụ web RESTful có thể được sử dụng trên nhiều nền tảng khác nhau, bao gồm cả các ứng dụng web và các ứng dụng di động.

Dễ dàng tiếp cận: ASP.NET Web API cung cấp các tiện ích và tính năng để xây dựng các dịch vụ web theo kiến trúc RESTful, giúp lập trình viên dễ dàng thiết kế và triển khai các dịch vụ web.

Tích hợp với công nghệ ASP.NET: ASP.NET Web API được tích hợp tốt với công nghệ ASP.NET, giúp lập trình viên xây dựng các ứng dụng web đa nền tảng dễ dàng hơn.

Hỗ trợ các định dạng dữ liệu khác nhau: ASP.NET Web API hỗ trợ chuyển đổi các đối tượng .NET sang các định dạng dữ liệu phổ biến như JSON, XML và các định dạng khác, giúp lập trình viên dễ dàng trao đổi dữ liệu giữa các ứng dụng khác nhau.

Bảo mật: ASP.NET Web API cung cấp các tính năng bảo mật như xác thực, phân quyền và quản lý lỗi để đảm bảo an toàn và bảo mật cho các dịch vụ web.

Tính linh hoạt: ASP.NET Web API cho phép bạn tuỳ chỉnh các hành vi của dịch vụ web và cung cấp tính linh hoạt cho các ứng dụng của bạn.

Hiệu suất cao: ASP.NET Web API được tối ưu hóa cho hiệu suất cao, giúp các dịch vụ web của bạn hoạt động nhanh chóng và mượt mà hơn.Top of Form

#### Kiến trúc của ASP.NET Web API

Kiến trúc của ASP.NET Web API dựa trên mô hình kiến trúc RESTful (Representational State Transfer). RESTful là một kiến trúc phần mềm cho phép các dịch vụ web tương tác với nhau thông qua các giao thức HTTP cơ bản như GET, POST, PUT, DELETE. Kiến trúc RESTful phù hợp với các ứng dụng web có tính chất phân tán và đa nền tảng.

ASP.NET Web API hỗ trợ các định dạng dữ liệu như XML và JSON để truyền tải dữ liệu giữa các ứng dụng khác nhau. Nó cung cấp các đối tượng và phương thức để xử lý các yêu cầu HTTP như GET, POST, PUT, DELETE và PATCH.

Kiến trúc của ASP.NET Web API bao gồm:

Mô hình Route: Mô hình Route cho phép lập trình viên chỉ định các định tuyến (routes) cho các yêu cầu HTTP. Các định tuyến giúp ứng dụng xác định các yêu cầu HTTP được gửi đến đúng phương thức xử lý.

Đối tượng HttpRequestMessage: Đối tượng HttpRequestMessage là đối tượng chứa thông tin về yêu cầu HTTP. Nó cung cấp thông tin như URL, headers và body của yêu cầu.

Đối tượng HttpResponseMessage: Đối tượng HttpResponseMessage là đối tượng chứa thông tin về phản hồi HTTP. Nó cung cấp thông tin như status code, headers và body của phản hồi.

Các phương thức HTTP: ASP.NET Web API hỗ trợ các phương thức HTTP như GET, POST, PUT, DELETE và PATCH để xử lý các yêu cầu HTTP.

Bộ mã hoá và giải mã: ASP.NET Web API cung cấp bộ mã hoá và giải mã để chuyển đổi dữ liệu giữa định dạng JSON hoặc XML và các đối tượng .NET.

Quản lý lỗi: ASP.NET Web API cung cấp các tính năng để quản lý lỗi trong quá trình xử lý các yêu cầu HTTP, giúp giảm thiểu các vấn đề phát sinh trong quá trình phát triển và triển khai.

### VueJs

#### VueJs là gì?

#### 

Hình 2. 3 Logo VueJs

Vue.js là một framework JavaScript mã nguồn mở được sử dụng để xây dựng các giao diện người dùng và các ứng dụng web đơn trang (single-page applications). Vue.js được tạo ra bởi Evan You vào năm 2014 và hiện đang được sử dụng rộng rãi trên khắp thế giới.

Vue.js được thiết kế để dễ học và dễ sử dụng, với cú pháp đơn giản và dễ hiểu, đặc biệt là đối với các lập trình viên JavaScript mới bắt đầu. Vue.js cũng hỗ trợ nhiều tính năng như reactivity (tự động cập nhật giao diện khi dữ liệu thay đổi), components (thành phần tái sử dụng), directives (hướng dẫn sử dụng), và routing (điều hướng đơn trang).

Vue.js được sử dụng để xây dựng các ứng dụng web đơn trang (single-page applications), các giao diện người dùng động và tương tác, và các ứng dụng web có tính năng phong phú. Vue.js cũng được sử dụng phổ biến trong các dự án web của các công ty lớn như Alibaba, Xiaomi và Xiaomi.

Một trong những ưu điểm của Vue.js là khả năng tích hợp với các thư viện và framework khác như React, Angular, hoặc Node.js. Ngoài ra, Vue.js cũng có cộng đồng phát triển đông đảo, nhiều tài liệu học tập và hỗ trợ từ cộng đồng, giúp cho việc học và sử dụng Vue.js trở nên dễ dàng hơn.

#### Lịch sử phát triển của VueJS

Vue.js được tạo ra bởi Evan You vào năm 2014. Trước khi tạo ra Vue.js, Evan đã làm việc cho Google và cũng đã tham gia vào việc phát triển AngularJS - một framework JavaScript phổ biến.

Ban đầu, Vue.js được phát triển như là một thư viện nhỏ để giúp Evan tạo ra các giao diện người dùng cho các ứng dụng web của mình một cách dễ dàng hơn. Tuy nhiên, khi được công bố công khai, Vue.js đã nhanh chóng trở thành một trong những framework JavaScript phổ biến nhất, được sử dụng rộng rãi trên toàn thế giới.

Vue.js đã được phát triển và cải tiến liên tục trong suốt những năm qua. Hiện tại, Vue.js đã đến phiên bản 3.0, được cải tiến và cung cấp nhiều tính năng mới, bao gồm hỗ trợ TypeScript và tính năng mới để tối ưu hóa hiệu suất.

Vue.js đã được sử dụng để phát triển các ứng dụng web phổ biến như Alibaba, Xiaomi, Xiaomi, Xiaomi, Grammarly và Adobe.

#### Tính khả dụng

Dễ học và sử dụng: Vue.js được thiết kế để dễ học và sử dụng, với cú pháp đơn giản và dễ hiểu.

Tính linh hoạt: Vue.js cho phép lập trình viên lựa chọn cách sử dụng framework, có thể dùng như một thư viện (library) hoặc một framework đầy đủ (full framework).

Tính tương thích: Vue.js tương thích với các thư viện và framework khác như React, Angular, hoặc Node.js.

Tính tái sử dụng cao: Vue.js hỗ trợ thành phần (component) tái sử dụng, giúp tiết kiệm thời gian và công sức trong quá trình phát triển ứng dụng.

Tính năng phong phú: Vue.js cung cấp nhiều tính năng như reactivity (tự động cập nhật giao diện khi dữ liệu thay đổi), directives (hướng dẫn sử dụng), routing (điều hướng đơn trang), và nhiều tính năng khác.

Hiệu suất cao: Vue.js được tối ưu hóa để hoạt động nhanh và mượt trên các ứng dụng web đơn trang (single-page applications).

Cộng đồng phát triển đông đảo: Vue.js có một cộng đồng phát triển lớn và nhiều tài liệu học tập và hỗ trợ từ cộng đồng, giúp cho việc học và sử dụng Vue.js trở nên dễ dàng hơn.

#### Các đặc trưng của VueJS

Cú pháp đơn giản: Vue.js sử dụng cú pháp rõ ràng và đơn giản, giúp lập trình viên dễ dàng học và sử dụng framework.

Tự động cập nhật giao diện: Vue.js có tính năng reactivity, giúp tự động cập nhật giao diện khi dữ liệu thay đổi.

Template syntax: Vue.js hỗ trợ template syntax, giúp xây dựng giao diện dễ dàng và đơn giản hơn.

Component-based architecture: Vue.js hỗ trợ kiến trúc component-based, giúp tách ứng dụng thành các thành phần nhỏ hơn, dễ dàng quản lý và tái sử dụng.

Directives: Vue.js hỗ trợ directives, giúp lập trình viên thao tác dễ dàng hơn với các thẻ HTML, ví dụ như v-bind, v-if, v-for, v-on, ...

Virtual DOM: Vue.js sử dụng Virtual DOM, giúp tối ưu hóa hiệu suất và tăng tốc độ ứng dụng.

Routing: Vue.js hỗ trợ điều hướng đơn trang (single-page application routing) thông qua Vue Router, giúp tạo ra các ứng dụng web mượt mà và thân thiện với người dùng.

State management: Vue.js hỗ trợ state management thông qua Vuex, giúp quản lý trạng thái ứng dụng một cách dễ dàng và hiệu quả.Bottom of Form

#### Lợi ích khi sử dụng VueJS

Dễ học và sử dụng: Vue.js có cú pháp đơn giản và dễ hiểu, giúp cho các lập trình viên có thể học và sử dụng nó nhanh chóng.

Được hỗ trợ bởi cộng đồng mạnh mẽ: Vue.js được phát triển bởi Evan You và có cộng đồng người dùng rộng lớn. Điều này đảm bảo rằng có rất nhiều tài liệu hướng dẫn, các thư viện và các nguồn tài nguyên có thể giúp cho các lập trình viên dễ dàng giải quyết các vấn đề phát triển.

Hiệu suất tốt: Vue.js sử dụng Virtual DOM, giúp tối ưu hóa hiệu suất và giảm thiểu việc render lại trang, giúp tăng tốc độ của ứng dụng.

Khả năng mở rộng: Vue.js hỗ trợ kiến trúc component-based, giúp tách ứng dụng thành các thành phần nhỏ hơn, dễ dàng quản lý và tái sử dụng. Điều này giúp cho các lập trình viên có thể mở rộng ứng dụng dễ dàng hơn.

Thân thiện với SEO: Vue.js có tính năng SSR (Server Side Rendering), giúp tối ưu hóa trang web cho công cụ tìm kiếm và tăng khả năng tìm kiếm của trang web.

Hỗ trợ điều hướng đơn trang: Vue.js hỗ trợ điều hướng đơn trang (single-page application routing) thông qua Vue Router, giúp tạo ra các ứng dụng web mượt mà và thân thiện với người dùng.

Quản lý trạng thái ứng dụng dễ dàng: Vue.js hỗ trợ state management thông qua Vuex, giúp quản lý trạng thái ứng dụng một cách dễ dàng và hiệu quả.

### Cơ sở dữ liệu MySql

#### MySql là gì?



Hình 2. 4 Logo MySql

MySQL là một hệ quản trị cơ sở dữ liệu mã nguồn mở, được phát triển bởi Oracle Corporation. MySQL sử dụng ngôn ngữ SQL để quản lý cơ sở dữ liệu và cung cấp một số tính năng quản lý dữ liệu như tạo, xóa, sửa đổi cơ sở dữ liệu, các bảng, thao tác với dữ liệu, phân quyền và bảo mật.

MySQL rất phổ biến và được sử dụng rộng rãi trên nhiều nền tảng và ứng dụng khác nhau. MySQL được sử dụng trong các ứng dụng web để lưu trữ dữ liệu, quản lý người dùng, quản lý hệ thống đăng nhập và các thông tin khác. MySQL cũng được sử dụng trong các hệ thống quản lý nội dung và các ứng dụng thương mại điện tử.

MySQL cung cấp nhiều phiên bản khác nhau, bao gồm MySQL Community Server (miễn phí) và MySQL Enterprise Edition (có phí). MySQL Community Server là phiên bản miễn phí, được phát triển và hỗ trợ bởi cộng đồng người dùng. MySQL Enterprise Edition cung cấp thêm tính năng, hỗ trợ và bảo mật, được hỗ trợ và cung cấp bởi Oracle Corporation.

#### Vì sao sử dụng MySql?

Miễn phí và mã nguồn mở: MySQL là một hệ quản trị cơ sở dữ liệu mã nguồn mở và hoàn toàn miễn phí để sử dụng và phân phối.

Dễ sử dụng: MySQL cung cấp một giao diện đơn giản và dễ sử dụng cho các tác vụ quản lý cơ sở dữ liệu. Nó cũng hỗ trợ nhiều ngôn ngữ lập trình và API, giúp các nhà phát triển dễ dàng tích hợp cơ sở dữ liệu vào các ứng dụng của họ.

Hiệu suất cao: MySQL có khả năng xử lý các truy vấn và ghi dữ liệu với tốc độ nhanh và hiệu quả, đặc biệt là với các ứng dụng web có tải trọng cao.

Độ tin cậy và bảo mật: MySQL được thiết kế để đảm bảo tính toàn vẹn và bảo mật của cơ sở dữ liệu. Nó có khả năng sao lưu và khôi phục dữ liệu, cũng như cung cấp các tính năng bảo mật như mã hóa dữ liệu và phân quyền truy cập.

Hỗ trợ đa nền tảng: MySQL có thể chạy trên nhiều nền tảng khác nhau, bao gồm Windows, Linux và macOS. Nó cũng có thể tích hợp với nhiều ngôn ngữ lập trình và công nghệ khác nhau.

## Công cụ sử dụng

* Visual Studio Code: phục vụ mục đích code VueJS.
* DbForge studio: cơ sở dữ liệu dùng để lưu trữ dữ liệu.
* Postman: sử dụng để thử nghiệm API.
* Chrome DevTools: gỡ lỗi khi phát triển web.
* Case Studio: thiết kế mô hình thực thể liên kết.
* Rational Rose: thiết kế hệ thống.

# PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## Phân tích hệ thống

### Vai trò của người dùng

* Hệ thống chia làm 3 nhóm người dùng chính:
  + Khách truy cập vãng lai
  + Khách hàng
  + Người quản trị
* Mô tả nhóm người dùng.
* Khách truy cập vãng lai: những người dùng truy cập trang web và chưa có tài khoản đăng nhập trên trang web. Khách truy cập vãng lai có thể xem thông tin sản phẩm, nếu không đăng ký tài khoản của trang web thì người dùng không thể đặt hàng.
* Khách hàng (thành viên của trang web): những khách hàng đã có tài khoản đăng nhập trên trang web. Khách hàng khi mua hàng trên trang web thì thông tin của khách hàng đã được lưu lại trong cơ sở dữ liệu và khi người dùng muốn mua hàng chỉ cần đăng nhập bằng tài khoản thông tin của khách hàng sẽ hiển thị và khách hàng chỉ cần tiến hành mua.
* Người quản trị: là người quản lý trang web và được truy cập vào các chức năng quản lý của hệ thống.

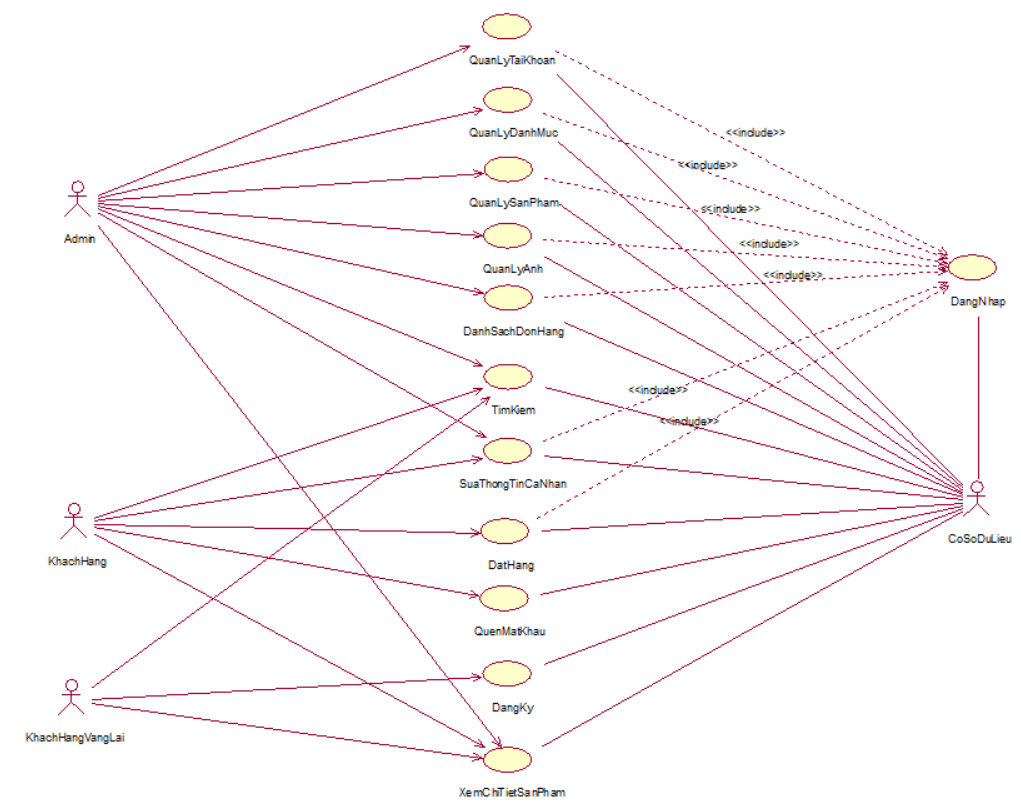
### Yêu cầu chức năng

* Khách truy cập vãng lai:
  + Đăng ký tài khoản
  + Xem chi tiết sản phẩm
  + Tìm kiếm sản phẩm
* Thành viên của trang web:
  + Đăng nhập
  + Đăng xuất
  + Xem chi tiết sản phẩm
  + Tìm kiếm sản phẩm
  + Đặt hàng
  + Sửa thông tin cá nhân
  + Quên mật khẩu
* Quản trị viên:
  + Sửa thông tin cá nhân
  + Quản lý tài khoản
  + Quản lý danh mục sản phẩm
  + Quản lý sản phẩm
  + Quản lý hình ảnh sản phẩm
  + Danh sản phẩm đơn hàng
  + Xem chi tiết sản phẩm
  + Tìm kiếm sản phẩm

### Yêu cầu phi chức năng

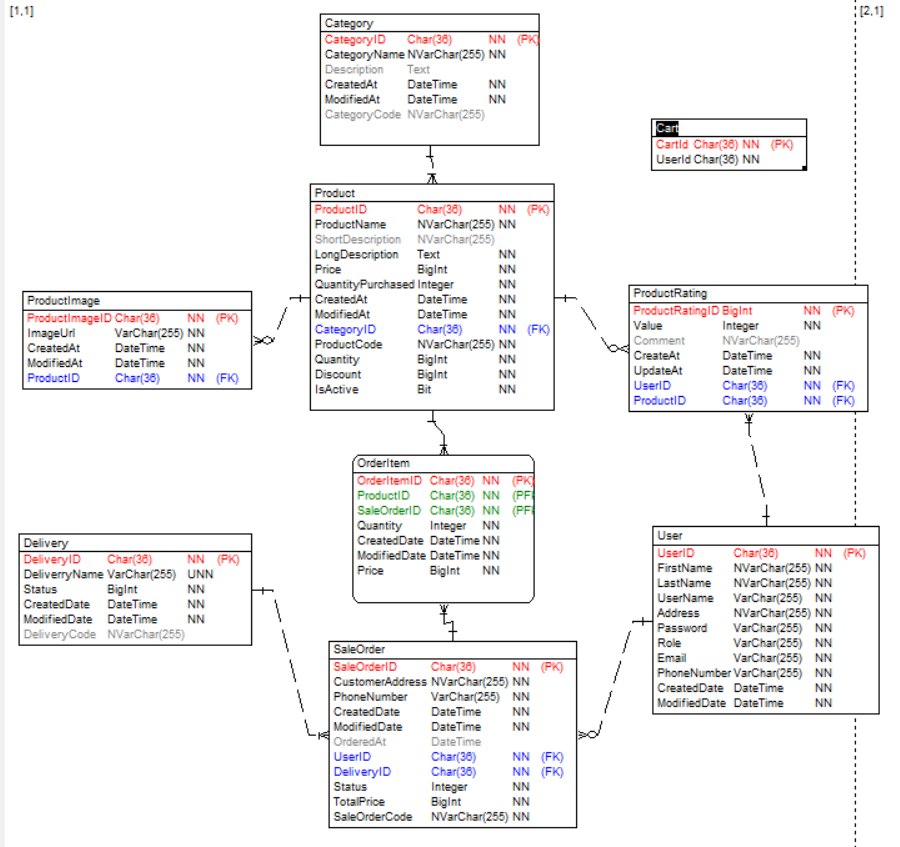
* Giao diện trang web được thiết kế dễ sử dụng và có tính thẩm mỹ cao.
* Được viết trên nền tảng web.
* Hiệu năng: tốc độ, khả năng và độ tin cậy của hệ thống.
* Website bảo mật phân quyền để thực hiện các chức năng của hệ thống.

## Biểu đồ use case tổng quát



Hình 3. 1 Biểu đồ use case tổng quát

## Mô hình thực thể liên kết



Hình 3. 2 Mô hình thực thể liên kết

## Đặc tả use case

### Use case đăng nhập

1. Biểu đồ use case:

Hình 3. 3 Biểu đồ use case đăng nhập

1. Đặc tả use case:

Use case này cho phép người dùng đăng nhập để xác định quyền truy cập vào hệ thống.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi người dùng bấm vào biểu tượng người dùng sau đó bấm vào nút “Đăng nhập” trên góc trái màn hình.
2. Hệ thống yêu cầu người dùng nhập các thông tin bao gồm: tên đăng nhập và mật khẩu.
3. Người dùng nhập các thông tin: tên đăng nhập và mật khẩu và bấm nút “Đăng nhập”.
4. Hệ thống kiểm tra tên đăng nhập và mật khẩu đã nhập và chuyển hướng sang trang chủ.
5. Use case kết thúc.
   * Luồng rẽ nhánh:
6. Sai tên đăng nhập hoặc mật khẩu: tại bước 3 trong luồng cơ bản nếu người dùng nhập tên đăng nhập hoặc mật khẩu sai, hệ thống hiển thị thông báo lỗi. Người dùng có thể nhập lại hoặc bỏ qua thao tác, khi đó use case kết thúc.
7. Đăng ký: tại bước 3 trong luồng cơ bản, nếu người dùng bấm nút “Đăng ký” use case kết thúc.
8. Quên mật khẩu: tại bước 3 trong luồng cơ bản, nếu người dùng bấm nút “Quên mật khẩu” use case kết thúc.
9. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Không có.

* Hậu điều kiện:

Nếu use case thành công, người dùng sẽ đăng nhập được vào hệ thống. Nếu không trạng thái của hệ thống không thay đổi.

* Điểm mở rộng:

Không có.

1. Biểu đồ trình tự:

Hình 3. 4 Biểu đồ trình tự đăng nhập

1. Biểu đồ hoạt động:

Hình 3. 5 Biểu đồ hoạt động đăng nhập

### Use case đăng ký

1. Biểu đồ use case:

Hình 3. 6 Biểu đồ use case đăng ký

1. Đặc tả use case:

Use case cho phép khách đăng ký tài khoản trên hệ thống.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi khách hàng kích vào nút đăng ký trên màn hình. Hệ thống sẽ hiển thị màn hình đăng ký.
2. Khách hàng nhập thông tin trên form đăng ký gồm First Name, Last Name, Username, Password, Email, Phone, Address rồi kích nút đăng ký. Hệ thống thêm một bản ghi mới vào trong bảng USER trong cơ sở dữ liệu.
3. Use case kết thúc.
   * Luồng rẽ nhánh:
4. Tại bước 2 trong luồng cơ bản, khi nhập thông tin không thoả mãn, hệ thống thông báo lỗi. Use case kết thúc.
5. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Không có.

* Hậu điều kiện:

Không có.

* Điểm mở rộng:

Không có.

1. Biểu đồ trình tự:

Hình 3. 7 Biểu đồ trình tự đăng ký

1. Biểu đồ hoạt động:

Hình 3. 8 Biểu đồ hoạt động đăng ký

### Use case tìm kiếm

1. Biểu đồ use case:

Hình 3. 9 Biểu đồ use case tìm kiếm

1. Đặc tả use case:

Use case cho phép người dùng tìm kiếm sản phẩm theo tên.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi khách hàng nhập tên sản phẩm vào ô tìm kiếm và kích vào biểu tượng kính lúp trên thanh tìm kiếm. Hệ thống sẽ lấy các thông tin về các sản phẩm có tiêu đề chứa từ khoá và gồm Path từ bảng PRODUCT\_IMAGE và Title, Price, Author từ bảng PRODUCT và hiển thị lên màn hình.
2. Use case kết thúc.
   * Luồng rẽ nhánh:
3. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Không có.

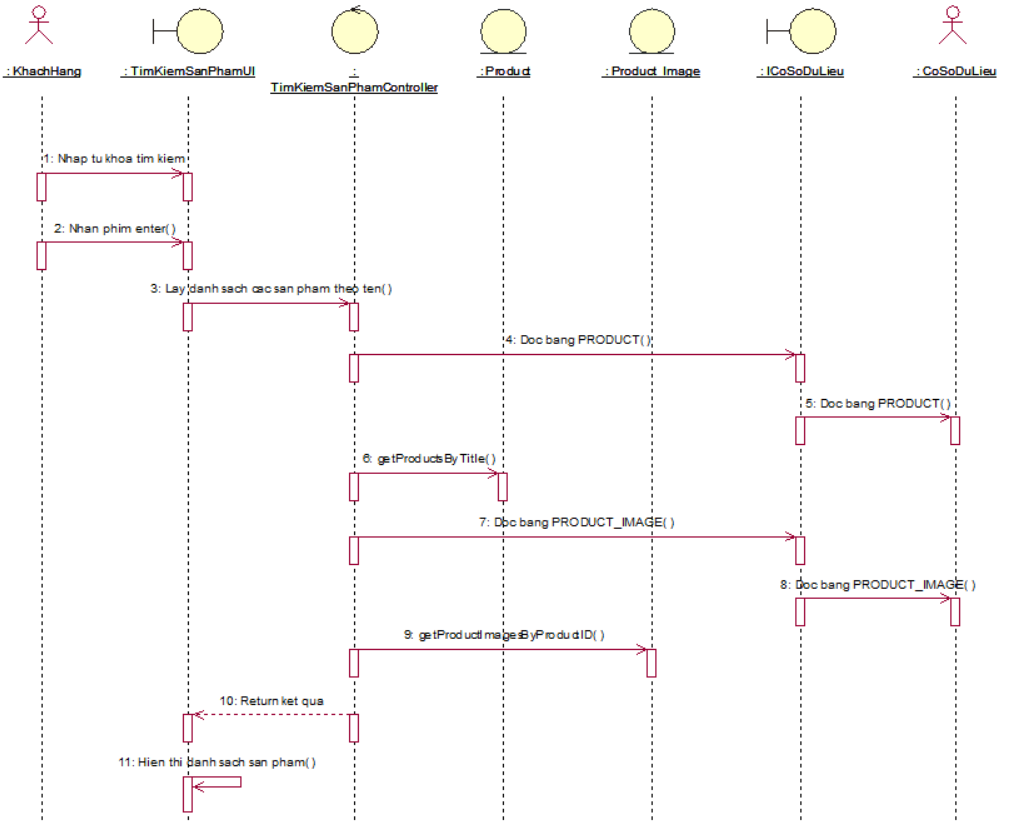
* Hậu điều kiện:

Không có.

* Điểm mở rộng:

Không có.

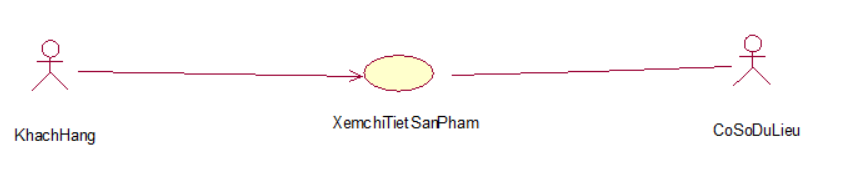
1. Biểu đồ trình tự:



Hình 3. 10 Biểu đồ trình tự tìm kiếm

### Use case xem chi tiết sản phẩm

1. Biểu đồ use case:



Hình 3. 11 Biểu đồ use case xem chi tiết sản phẩm

1. Đặc tả use case:

Use case cho phép khách hàng xem thông tin chi tiết về sản phẩm.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi khách hàng kích vào tên sản phẩm trong danh sản phẩm sản phẩm . Hệ thống sẽ lấy các thông tin về sản phẩm bao gồm Path từ bảng PRODUCT\_IMAGE và Title, Price, Author, CurrentNumber, NumberOfPage, LongDescription từ bảng PRODUCT và hiển thị lên màn hình.
2. Use case kết thúc.
   * Luồng rẽ nhánh:
3. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Không có.

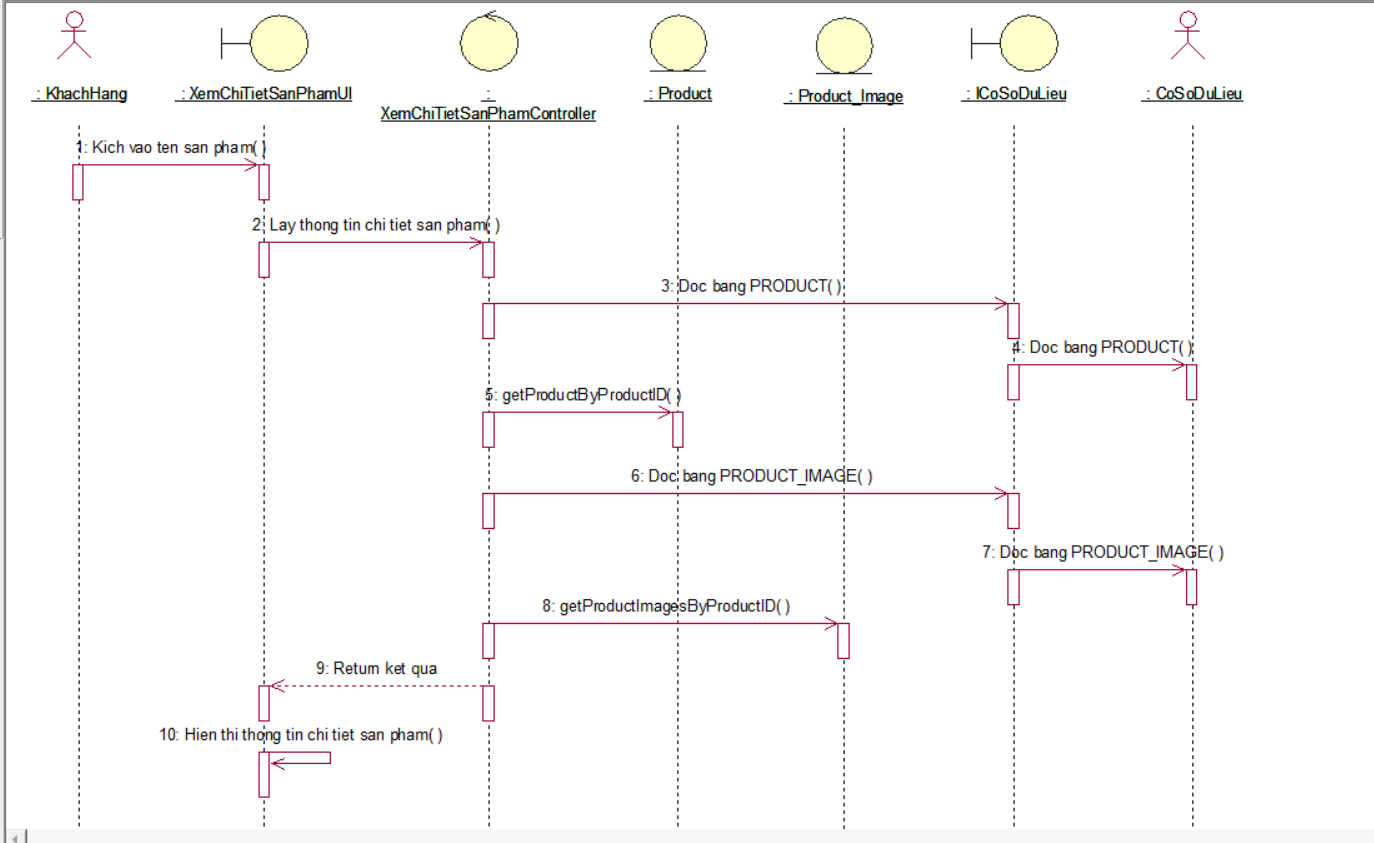
* Hậu điều kiện:

Không có.

* Điểm mở rộng:

Không có.

1. Biểu đồ trình tự:



Hình 3. 12 Biểu đồ trình tự xem chi tiết sản phẩm

### Use case sửa thông tin cá nhân

1. Biểu đồ use case:

Hình 3. 13 Biểu đồ use case sửa thông tin cá nhân

1. Đặc tả use case:

Use case cho phép khách hàng sửa thông tin cá nhân trong tài khoản.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi khách hàng kích vào nút profile trên màn hình. Hệ thống sẽ hiển thị màn hình thông tin cá nhân.
2. Khách hàng nhập thông tin cần chỉnh sửa rồi kích nút cập nhật. Hệ thống cập nhật lại thông tin trên bảng USER trong cơ sở dữ liệu.
3. Use case kết thúc.
   * Luồng rẽ nhánh:
4. Tại bước 2 ở luồng cơ bản, nếu khách hàng nhập sai định dạng, hệ thống thông báo lỗi. Use case kết thúc.
5. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Khách hàng đăng nhập thành công.

* Hậu điều kiện:

Không có.

* Điểm mở rộng:

Không có.

1. Biểu đồ trình tự:

Hình 3. 14 Biểu đồ trình tự sửa thông tin cá nhân

1. Biểu đồ hoạt động:

Hình 3. 15 Biểu đồ hoạt động sửa thông tin cá nhân

### Use case quên mật khẩu

1. Biểu đồ use case:

Hình 3. 16 Biểu đồ use case quên mật khẩu

1. Đặc tả use case:

Use case này cho phép khách hàng yêu cầu hệ thống cấp lại mật khẩu mới khi quên mật khẩu.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi khách hàng kích vào nút quên mật khẩu trên màn hình. Hệ thống sẽ hiển thị form quên mật khẩu.
2. Khách hàng nhập tên tài khoản rồi kích nút Submit. Hệ thống kiểm tra tên tài khoản trong cơ sở dữ liệu và cập nhật lại mật khẩu mới.
3. Use case kết thúc.
   * Luồng rẽ nhánh:
4. Tại bước 2 ở luồng cơ bản, nếu tên tài khoản của khách hàng không tồn tại trong cơ sở dữ liệu, hệ thống báo lỗi. Use case kết thúc.
5. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Không có.

* Hậu điều kiện:

Không có.

* Điểm mở rộng:

Không có.

1. Biểu đồ trình tự:

Hình 3. 17 Biểu đồ trình tự quên mật khẩu

1. Biểu đồ hoạt động:

Hình 3. 18 Biểu đồ hoạt động quên mật khẩu

### Use case đặt hàng

1. Biểu đồ use case:

Hình 3. 19 Biểu đồ use case đặt hàng

1. Đặc tả use case:

Use case cho phép khách hàng đặt mua các sản phẩm đã được thêm vào giỏ hàng.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi khách hàng bấm “Thanh toán” tại trang giỏ hàng. Hệ thống hiển thị màn hình thông tin khách hàng gồm FirstName, LastName, Address, Phone từ bảng User và lấy thông tin các sản phẩm gồm Title, Price từ bảng PRODUCT và Quantity từ bảng ORDER\_ITEM hiển thị lên khung “Thông tin đơn hàng”.
2. Khách hàng kích vào nút “Đặt mua”. Hệ thống tạo một đơn hàng mới trong bảng SALE\_ORDER, tạo các chi tiết đơn hàng mới trong bảng ORDER\_ITEM và thông báo đặt hàng thành công, thông tin đơn hàng vừa đặt.
3. Use case kết thúc.
   * Luồng rẽ nhánh:
4. Tại bước 1 trong luồng cơ bản, khi khách hàng bấm mua hàng mà chưa chọn số lượng hệ thống sẽ tự động đặt số lượng sản phẩm đó là 1 và thêm vào giỏ hàng.
5. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Khách hàng đăng nhập thành công.

* Hậu điều kiện:

Không có.

* Điểm mở rộng:

Không có.

1. Biểu đồ trình tự:

Hình 3. 20 Biểu đồ trình tự đặt hàng

1. Biểu đồ hoạt động:

Hình 3. 21 Biểu đồ hoạt động đặt hàng

### Use case quản lý tài khoản

1. Biểu đồ use case:

Hình 3. 22 Biểu đồ use case quản lí tài khoản

1. Đặc tả use case:

Use case này cho phép quản trị viên xem, thêm, sửa thông tin, xóa các tài khoản trong bảng USER.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi quản trị viên kích vào nút “User management” trên thanh Menu quản trị. Hệ thống lấy thông tin chi tiết của các tài khoản (FirstName, LastName, Username, Phone, Email, Address) từ bảng USER trong cơ sở dữ liệu và hiển thị lên màn hình.
2. Thêm mới tài khoản:
   1. Quản trị viên kích vào nút “ADD USER”. Hệ thống hiển thị “User Form” yêu cầu quản trị viên nhập các thông tin (First Name, Last Name, Email, Mobile, Address, Username, Password, Confirm Password, Role, Amount).
   2. Quản trị viên nhập các thông tin (First Name, Last Name, Email, Mobile, Address, Username, Password, Confirm Password, Role, Amount) và kích vào nút “Submit”. Hệ thống tạo ra UserID và lưu các thông tin (UserID, First Name, Last Name, Username, Password, Phone, Email, Address, Role, Amount, CreatedAt, UpdatedAt) vào bảng USER và hiển thị danh sản phẩm các tài khoản lên màn hình.
3. Xem chi tiết tài khoản:
   1. Quản trị viên kích vào nút “View” trên dòng tài khoản. Hệ thống lấy thông tin chi tiết của tài khoản (First Name, Last Name, Username, Phone, Email, Address, Role, Amount, CreatedAt, UpdatedAt) từ bảng USER trong cơ sở dữ liệu và hiển thị lên màn hình.
   2. Quản trị viên kích vào biểu tượng “X” trên cửa sổ “User Form”. Hệ thống sẽ hiển thị lại danh sản phẩm tài khoản trên màn hình.
4. Sửa thông tin tài khoản:
   1. Quản trị viên kích vào nút “Edit” trên dòng tài khoản. Hệ thống hiển thị một màn hình với các thông tin cũ của tài khoản (First Name, Last Name, Username, Phone, Email, Address, Role, Amount, Password).
   2. Quản trị viên sửa thông tin (First Name, Last Name, Phone, Email, Address, Mobile, Role, Amount) của tài khoản và kích vào nút “Submit”. Hệ thống sẽ cập nhật thông tin tài khoản trong bảng USER và hiển thị danh sản phẩm tài khoản đã cập nhật.
5. Xóa tài khoản:
   1. Quản trị viên kích vào nút “Delete” trên một dòng tài khoản. Hệ thống sẽ hiển thị màn hình “Delete User” yêu cầu xác nhận xóa.
   2. Quản trị viên kích vào nút “Yes”. Hệ thống sẽ xóa tài khoản khỏi bảng USER và hiển thị danh sản phẩm các tài khoản đã cập nhật.
6. Use case kết thúc.
   * Luồng rẽ nhánh:
7. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì website sẽ hiển thị một thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Quản trị viên phải đăng nhập vào hệ thống trước thực hiện quản lý tài khoản.

* Hậu điều kiện:

Không có.

* Điểm mở rộng:

Không có.

1. Biểu đồ trình tự:

Hình 3. 23 Biểu đồ trình tự quản lí tài khoản

1. Biểu đồ hoạt động:

Hình 3. 24 Biểu đồ hoạt động quản lí tài khoản

### Use case quản lý danh mục sản phẩm

1. Biểu đồ use case:

Hình 3. 25 Biểu đồ use case quản lí danh mục

1. Đặc tả use case:

Use case này cho phép quản trị viên xem, thêm, sửa thông tin, xóa các danh mục trong bảng CATEGORY.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi quản trị viên kích vào nút “Category management” trên thanh Menu quản trị. Hệ thống lấy thông tin chi tiết của các danh mục (Name, Description) từ bảng CATEGORY trong cơ sở dữ liệu và hiển thị lên màn hình.
2. Thêm mới danh mục:
   1. Quản trị viên kích vào nút “ADD CATEGORY”. Hệ thống hiển thị “Category Form” yêu cầu quản trị viên nhập các thông tin (Name, Description).
   2. Quản trị viên nhập các thông tin (Name, Description) và kích vào nút “Submit”. Hệ thống tạo ra CategoryID, Slug từ Name và lưu các thông tin (CategoryID, Name, Description, CreatedAt, UpdatedAt, Slug) vào bảng CATEGORY và hiển thị danh sản phẩm các danh mục lên màn hình.
3. Xem chi tiết danh mục:
   1. Quản trị viên kích vào nút “View” trên dòng danh mục. Hệ thống lấy thông tin chi tiết của danh mục (Name, Description, CreatedAt, UpdatedAt) từ bảng CATEGORY trong cơ sở dữ liệu và hiển thị lên màn hình.
   2. Quản trị viên kích vào biểu tượng “X” trên cửa sổ “Category Form”. Hệ thống sẽ hiển thị lại danh sản phẩm danh mục trên màn hình.
4. Sửa thông tin danh mục:
   1. Quản trị viên kích vào nút “Edit” trên dòng danh mục. Hệ thống hiển thị một màn hình với các thông tin cũ của danh mục (Name, Description).
   2. Quản trị viên sửa thông tin (Name, Description) của danh mục và kích vào nút “Submit”. Hệ thống sẽ cập nhật thông tin danh mục trong bảng CATEGORY và hiển thị danh sản phẩm danh mục đã cập nhật.
5. Xóa danh mục:
   1. Quản trị viên kích vào nút “Delete” trên một dòng danh mục. Hệ thống sẽ hiển thị màn hình “Delete Category” yêu cầu xác nhận xóa.
   2. Quản trị viên kích vào nút “Yes”. Hệ thống sẽ xóa danh mục khỏi bảng CATEGORY và hiển thị danh sản phẩm các danh mục đã cập nhật.
6. Use case kết thúc.
   * Luồng rẽ nhánh:
7. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì website sẽ hiển thị một thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Quản trị viên phải đăng nhập vào hệ thống trước thực hiện quản lý danh mục.

* Hậu điều kiện:

Không có.

* Điểm mở rộng:

Không có.

1. Biểu đồ trình tự:



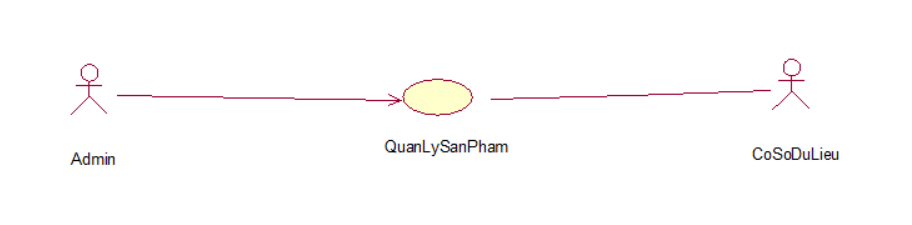
Hình 3. 26 Biểu đồ trình tự quản lí danh mục

1. Biểu đồ hoạt động:

Hình 3. 27 Biểu đồ hoạt động quản lí danh mục

### Use case quản lý sản phẩm

1. Biểu đồ use case:



Hình 3. 28 Biểu đồ use case quản lí sản phẩm

1. Đặc tả use case:

Use case này cho phép quản trị viên xem, thêm, sửa thông tin, xóa các sản phẩm trong bảng PRODUCT.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi quản trị viên kích vào nút “Product management” trên thanh Menu quản trị. Hệ thống lấy thông tin chi tiết của các sản phẩm (Title, Price, Author) từ bảng PRODUCT trong cơ sở dữ liệu và hiển thị lên màn hình.
2. Thêm mới sản phẩm:
   1. Quản trị viên kích vào nút “ADD PRODUCT”. Hệ thống hiển thị “Product Form” yêu cầu quản trị viên nhập các thông tin (Title, Long Description, Price, Author, Current Number, Number Of Page, Category).
   2. Quản trị viên nhập các thông tin (Title, Long Description, Price, Author, Current Number, Number Of Page, Category) và kích vào nút “Submit”. Hệ thống tạo ra ProductID, Slug từ Title và lưu các thông tin (ProductID, Title, Long Description, Price, Author, Current Number, Number Of Page, CategoryID, CreatedAt, UpdatedAt, Slug) vào bảng PRODUCT và hiển thị danh sản phẩm các sản phẩm lên màn hình.
3. Xem chi tiết sản phẩm:
   1. Quản trị viên kích vào nút “View” trên dòng sản phẩm. Hệ thống lấy thông tin chi tiết của sản phẩm (Title, Long Description, Price, Author, Current Number, Number Of Page, Quantity Purchased, Category, CreatedAt, UpdatedAt) từ bảng PRODUCT trong cơ sở dữ liệu và hiển thị lên màn hình.
   2. Quản trị viên kích vào biểu tượng “X” trên cửa sổ “Product Form”. Hệ thống sẽ hiển thị lại danh sản phẩm sản phẩm trên màn hình.
4. Sửa thông tin tài khoản:
   1. Quản trị viên kích vào nút “Edit” trên dòng sản phẩm. Hệ thống hiển thị một màn hình với các thông tin cũ của sản phẩm (Title, Long Description, Price, Author, Current Number, Number Of Page, Quantity Purchased, Category).
   2. Quản trị viên sửa thông tin (Title, Long Description, Price, Author, Current Number, Number Of Page, Quantity Purchased, Category) của sản phẩm và kích vào nút “Submit”. Hệ thống sẽ cập nhật thông tin sản phẩm trong bảng PRODUCT và hiển thị danh sản phẩm sản phẩm đã cập nhật.
5. Xóa sản phẩm:
   1. Quản trị viên kích vào nút “Delete” trên một dòng sản phẩm. Hệ thống sẽ hiển thị màn hình “Delete Product” yêu cầu xác nhận xóa.
   2. Quản trị viên kích vào nút “Yes”. Hệ thống sẽ xóa sản phẩm khỏi bảng Product và hiển thị danh sản phẩm các sản phẩm đã cập nhật.
6. Use case kết thúc.
   * Luồng rẽ nhánh:
7. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì website sẽ hiển thị một thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Quản trị viên phải đăng nhập vào hệ thống trước thực hiện quản lý sản phẩm.

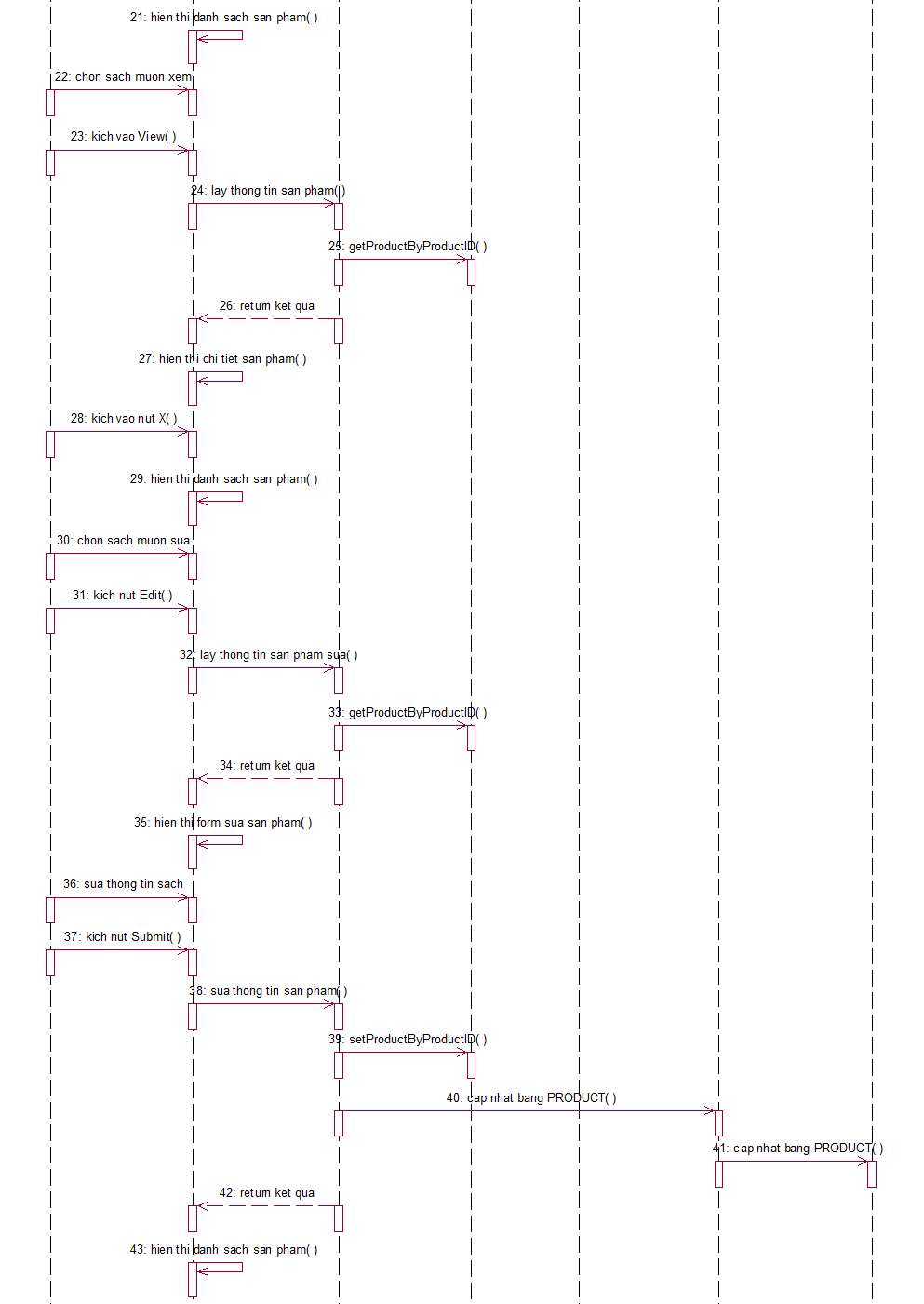
* Hậu điều kiện:

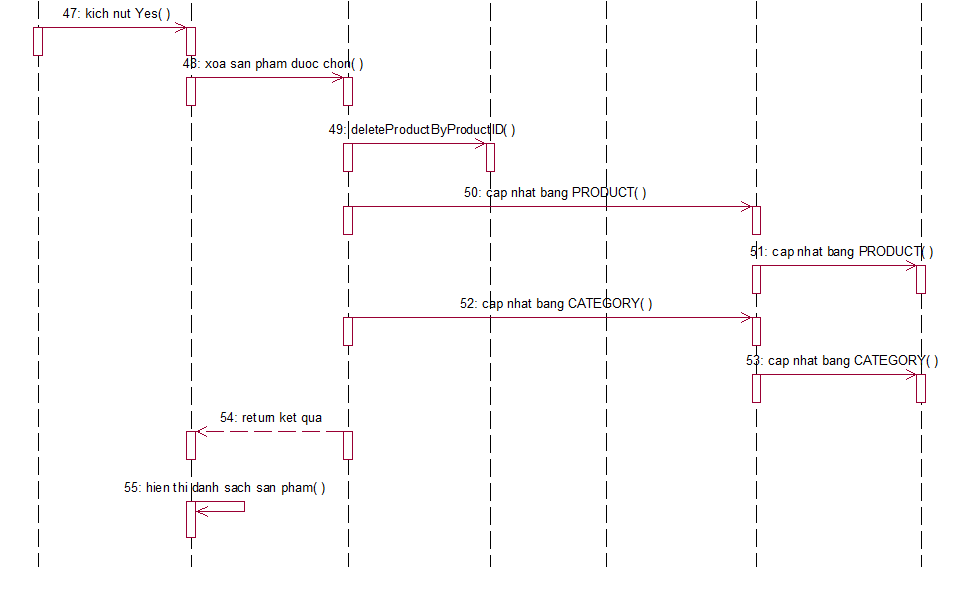
Không có.

* Điểm mở rộng:

Không có.

1. Biểu đồ trình tự:



Hình 3. 29 Biểu đồ trình tự quản lí sản phẩm

### Use case quản lý hình ảnh sản phẩm

1. Biểu đồ use case:

Hình 3. 30 Biểu đồ use case quản lí ảnh

1. Đặc tả use case:

Use case này cho phép quản trị viên xem, thêm, sửa thông tin, xóa các hình ảnh sản phẩm trong bảng PRODUCT\_IMAGE.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi quản trị viên kích vào nút “Product Images management” trên thanh Menu quản trị. Hệ thống lấy thông tin của các sản phẩm (Title) từ bảng PRODUCT và số lượng ảnh thuộc sản phẩm đó trong cơ sở dữ liệu và hiển thị lên màn hình.
2. Thêm mới sản phẩm:
   1. Quản trị viên kích vào nút “ADD PRODUCT IMAGES”. Hệ thống hiển thị “Product Image Form” yêu cầu quản trị viên nhập thông tin (Title, Path).
   2. Quản trị viên nhập thông tin (Title, Path) và kích vào nút “Submit”. Hệ thống tạo ra ProductImageID và lưu các thông tin (ProductImageID, Path, CreatedAt, UpdatedAt, ProductID) vào bảng PRODUCT\_IMAGE và hiển thị danh sản phẩm thông tin ảnh lên màn hình.
3. Xem thông tin ảnh:
   1. Quản trị viên kích vào nút “View” trên dòng thông tin ảnh. Hệ thống lấy thông tin của sản phẩm (Title) từ bảng PRODUCT và (Path) từ bảng PRODUCT\_IMAGE trong cơ sở dữ liệu và hiển thị thông tin ảnh lên màn hình.
   2. Quản trị viên kích vào biểu tượng “X” trên cửa sổ “Product Image Form”. Hệ thống sẽ hiển thị lại danh sản phẩm thông tin ảnh trên màn hình.
4. Sửa thông tin ảnh:
   1. Quản trị viên kích vào nút “Edit” trên dòng thông tin ảnh. Hệ thống hiển thị một màn hình với các thông tin ảnh cũ (Title, Long Description, Path).
   2. Quản trị viên sửa thông tin (Title, Long Description, Path) của sản phẩm và kích vào nút “Submit”. Hệ thống sẽ cập nhật thông tin ảnh trong bảng PRODUCT và bảng PRODUCT\_IMAGE sau đó hiển thị danh sản phẩm thông tin ảnh đã cập nhật.
5. Xóa thông tin ảnh:
   1. Quản trị viên kích vào nút “Delete” trên một dòng thông tin ảnh. Hệ thống sẽ hiển thị màn hình “Delete Product Image” yêu cầu xác nhận xóa.
   2. Quản trị viên kích vào nút “Yes”. Hệ thống sẽ xóa thông tin ảnh khỏi bảng PRODUCT\_IMAGE và hiển thị danh sản phẩm thông tin ảnh đã cập nhật.
6. Use case kết thúc.
   * Luồng rẽ nhánh:
7. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì website sẽ hiển thị một thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Quản trị viên phải đăng nhập vào hệ thống trước thực hiện quản lý hình ảnh sản phẩm.

* Hậu điều kiện:

Không có.

* Điểm mở rộng:

Không có.

1. Biểu đồ trình tự:



Hình 3. 31 Biểu đồ trình tự quản lí ảnh

1. Biểu đồ hoạt động:

Hình 3. 32 Biểu đồ hoạt động quản lí ảnh

### Use case danh sản phẩm đơn hàng

1. Biểu đồ use case:

Hình 3. 33 Biểu đồ use case quản lí đơn hàng

1. Đặc tả use case:

Use case này cho phép quản trị viên xem, sửa trạng thái đơn hàng trong bảng SALE\_ORDER và ORDER\_ITEM.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi quản trị viên kích vào nút “Sale orders management” trên thanh Menu quản trị. Hệ thống lấy thông tin của đơn hàng (SaleOrderID, CustomerName, CustomerPhone) từ bảng SALE\_ORDER và (Value) từ bảng DELIVERY trong cơ sở dữ liệu và hiển thị lên màn hình.
2. Xem thông tin đơn hàng:
   1. Quản trị viên kích vào nút “View” trên dòng đơn hàng. Hệ thống lấy thông tin của đơn hàng (SaleOrderID, CreatedAt, Customer Name, Customer Phone) từ bảng SALE\_ORDER, (Value) từ bảng DELIVERY, (ProductID, Quantity) từ bảng ORDER\_ITEM và (Title, Price) từ bảng PRODUCT trong cơ sở dữ liệu và hiển thị thông tin đơn hàng lên màn hình.
   2. Quản trị viên kích vào biểu tượng “X” trên cửa sổ “View Sale Order”. Hệ thống sẽ hiển thị lại danh sản phẩm thông tin ảnh trên màn hình.
3. Sửa trạng thái đơn hàng:
   1. Quản trị viên kích vào nút “Edit” trên dòng đơn hàng. Hệ thống hiển thị một màn hình với các thông tin cũ của đơn hàng (SaleOrderID, Value).
   2. Quản trị viên sửa thông tin (Value) của đơn hàng và kích vào nút “Submit”. Hệ thống sẽ cập nhật thông tin đơn hàng trong bảng SALE\_ORDER sau đó hiển thị danh sản phẩm đơn hàng đã cập nhật.
4. Use case kết thúc.
   * Luồng rẽ nhánh:
5. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì website sẽ hiển thị một thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Quản trị viên phải đăng nhập vào hệ thống trước thực hiện quản lý đơn hàng.

* Hậu điều kiện:

Không có.

* Điểm mở rộng:

Không có.

1. Biểu đồ trình tự:

Hình 3. 34 Biểu đồ trình tự quản lí đơn hàng

1. Biểu đồ hoạt động:

Hình 3. 35 Biểu đồ hoạt động quản lí đơn hàng

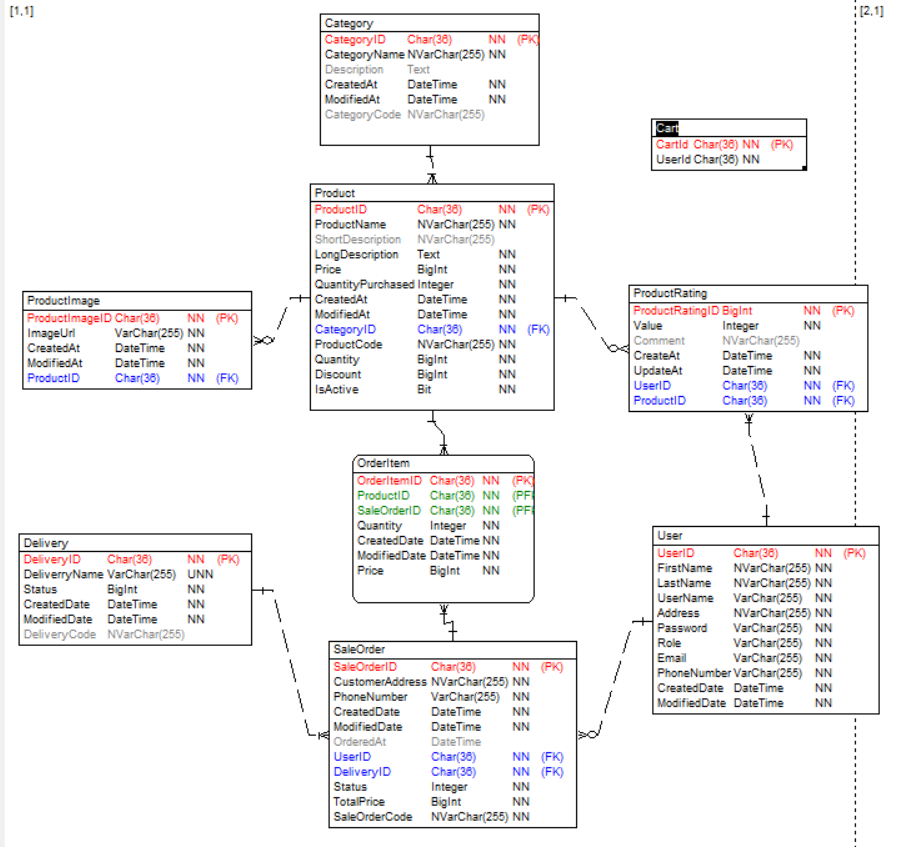
## Thiết kế hệ thống

### **Diagram Description automatically generated**Biểu đồ triển khai hệ thống

Hình 3. 36 Biểu đồ triển khai hệ thống

### Thiết kế cơ sở dữ liệu

#### Biểu đồ mô tả cấu trúc cơ sở dữ liệu



Hình 3. 37 Biểu đồ mô tả cấu trúc cơ sở dữ liệu

#### Mô tả cơ sở dữ liệu

Bảng 3.1. Bảng Category

Bảng Category để lưu thông tin danh mục sản phẩm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu | Độ dài | Khóa | Giải thích |
| CategoryID | Char | 36 | PK | Id danh mục |
| CategoryName | NVarChar | 255 |  | Tên danh mục |
| CategoryCode | VarChar | 255 |  | Mã danh mục |
| CreatedDate | DateTime |  |  | Thời gian tạo |
| ModifiedDate | DateTime |  |  | Thời gian cập nhật |

Bảng 3.2. Bảng Product

Bảng Product để lưu thông tin sản phẩm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu | Độ dài | Khóa | Giải thích |
| ProductID | Char | 36 | PK | Id sản phẩm |
| ProductName | NVarChar | 255 |  | Tên |
| ShortDescription | NVarChar | 255 |  | Mô tả ngắn |
| LongDescription | Text |  |  | Mô tả |
| Price | BigInt |  |  | Giá |
| Author | NVarChar | 255 |  | Tác giả |
| Quantity | Integer |  |  | Số lượng còn hiện tại |
| IsActive | Integer |  |  | Còn bán hay không |
| Discount | Interger |  |  | Giảm giá |
| QuantityPurchased | Integer |  |  | Số lượng đã được mua |
| CreatedDate | DateTime |  |  | Thời gian tạo |
| ModifiedDate | DateTime |  |  | Thời gian cập nhật |
| CategoryID | Char | 36 | FK | Id danh mục |

Bảng 3.3. Bảng ProductImage

Bảng ProductImage để lưu thông tin hình ảnh sản phẩm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu | Độ dài | Khóa | Giải thích |
| ProductImageID | Char | 36 | PK | Id ảnh sản phẩm |
| ImageUrl | VarChar | 255 |  | Đường dẫn ảnh |
| CreatedDate | DateTime |  |  | Thời gian tạo |
| Modified | DateTime |  |  | Thời gian cập nhật |
| ProductID | Char | 36 | FK | Id sản phẩm |

Bảng 3.4. Bảng Delivery

Bảng Delivery để lưu thông tin trạng thái giao hàng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu | Độ dài | Khóa | Giải thích |
| DeliveryID | Char | 36 | PK | Id trạng thái giao hàng |
| DeliveryCode | VarChar | 255 |  | Mã |
| Status | Interger |  |  | Giá trị trạng thái |
| CreatedDate | DateTime |  |  | Thời gian tạo |
| ModifiedDate | DateTime |  |  | Thời gian cập nhật |

Bảng 3.5. Bảng User

Bảng User để lưu thông tin người dùng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu | Độ dài | Khóa | Giải thích |
| UserID | Char | 36 | PK | Id người dùng |
| FirstName | NVarChar | 255 |  | Tên |
| LastName | NVarChar | 255 |  | Họ |
| Username | VarChar | 255 |  | Tên tài khoản |
| Address | NVarChar | 255 |  | Địa chỉ |
| Password | VarChar | 255 |  | Mật khẩu |
| Role | VarChar | 255 |  | Vai trò |
| Email | VarChar | 255 |  | Địa chỉ email |
| PhoneNumber | VarChar | 255 |  | Số điện thoại |
| CreatedDate | DateTime |  |  | Thời gian tạo |
| ModifiedDate | DateTime |  |  | Thời gian cập nhật |

Bảng 3.6. Bảng SaleOrder

Bảng SaleOrder để lưu thông tin đơn hàng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu | Độ dài | Khóa | Giải thích |
| SaleOrderID | Char | 36 | PK | Id đơn hàng |
| CustomerAddress | NVarChar | 255 |  | Địa chỉ nhận hàng |
| PhoneNumber | VarChar | 255 |  | Số điện thoại |
| UserID | Char | 36 | FK | Id người dùng |
| DeliveryID | Char | 36 | FK | Id trạng thái giao hàng |
| CreatedDate | DateTime |  |  | Thời gian tạo |
| ModifiedDate | DateTime |  |  | Thời gian cập nhật |
| TotalPrice | Interger |  |  | Tổng tiền |
| Status | Interger |  |  | Trạng thái đơn hàng |
| SaleOrderCode | NVarChar | 255 |  | Mã đơn hàng |

Bảng 3.7. Bảng OrderItem

Bảng OrderItem để lưu thông tin mặt hàng trong đơn hàng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu | Độ dài | Khóa | Giải thích |
| OrderItemID | Char | 36 | PK | Id mặt hàng |
| ProductID | Char | 36 | PFK | Id sản phẩm |
| SaleOrderID | Char | 36 | PFK | Id đơn hàng |
| Quantity | Integer |  |  | Số lượng |
| CreatedDate | DateTime |  |  | Thời gian tạo |
| ModifiedDate | DateTime |  |  | Thời gian cập nhật |
| Price | Interger |  |  | Giá |

Bảng 3.8. Bảng ProductRating

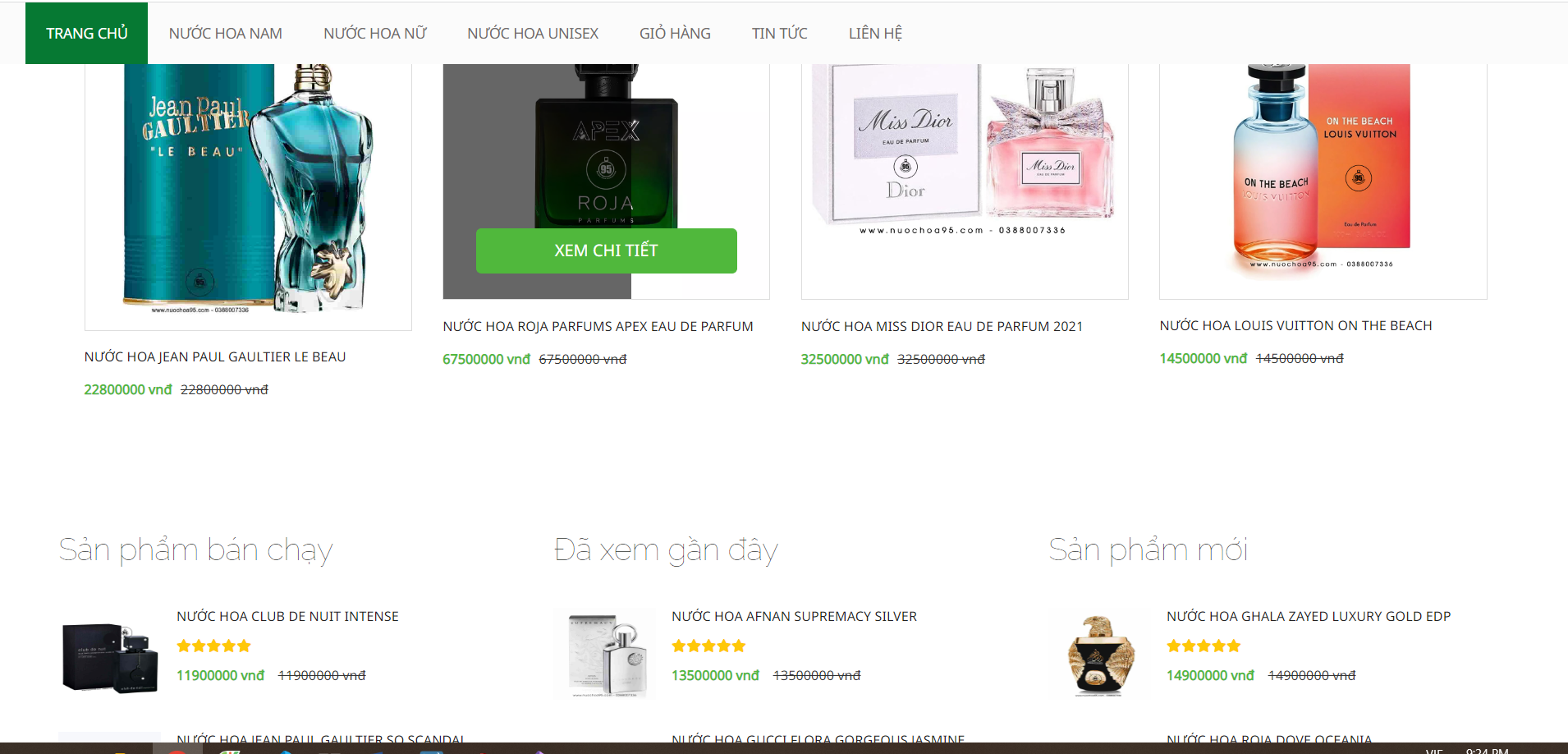
Bảng ProductRating để lưu thông tin đánh giá của sản phẩm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu | Độ dài | Khóa | Giải thích |
| ProductRatingID | Char | 36 | PK | Id đánh giá sản phẩm |
| Value | Integer |  |  | Giá trị đánh giá |
| Comment | NVarChar | 255 |  | Bình luận |
| UserID | Char | 36 | FK | Id người dùng |
| ProductID | Char | 36 | FK | Id sản phẩm |
| CreatedDate | DateTime |  |  | Thời gian tạo |
| ModifiedDate | DateTime |  |  | Thời gian cập nhật |

### Thiết kế giao diện

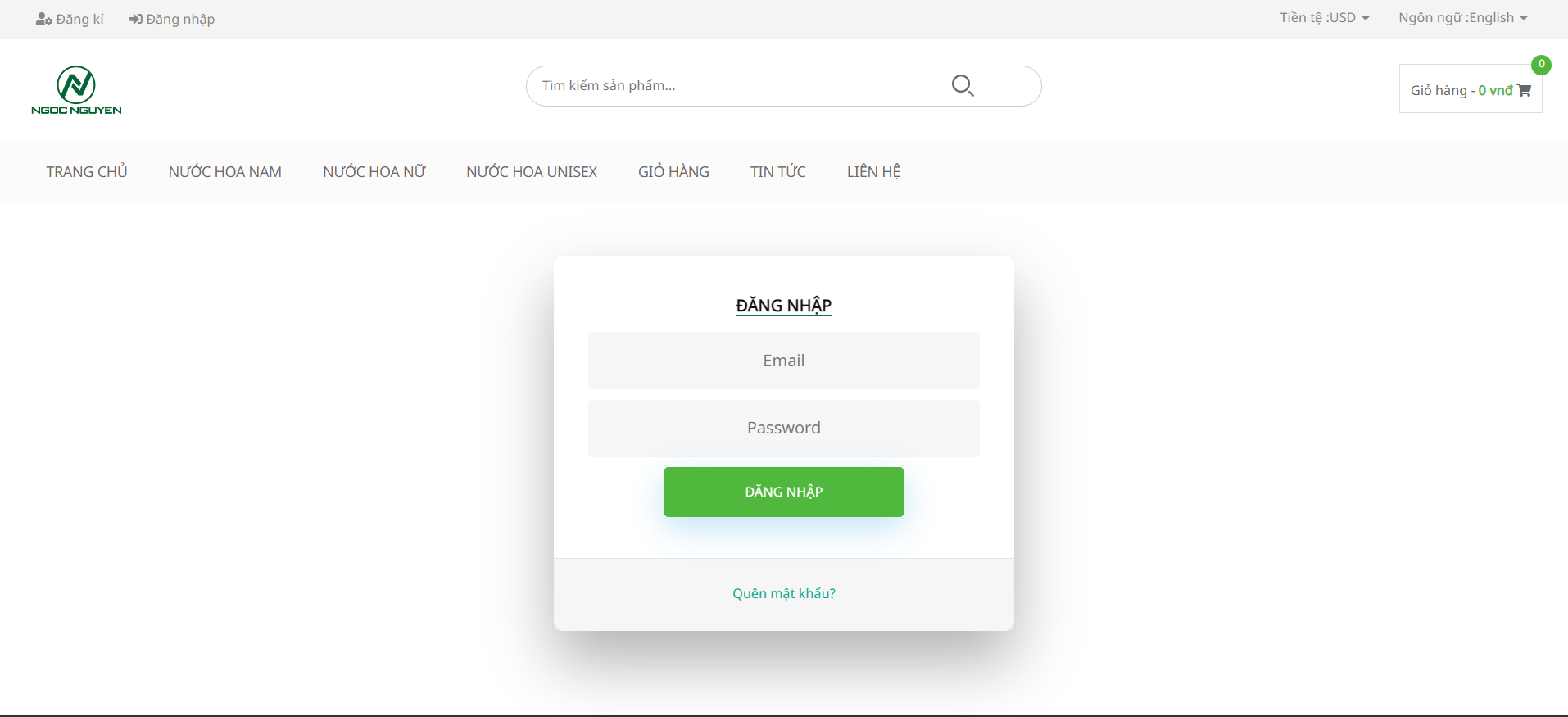
#### Giao diện chung

Giao diện trang chủ được minh họa bằng hình ảnh dưới đây



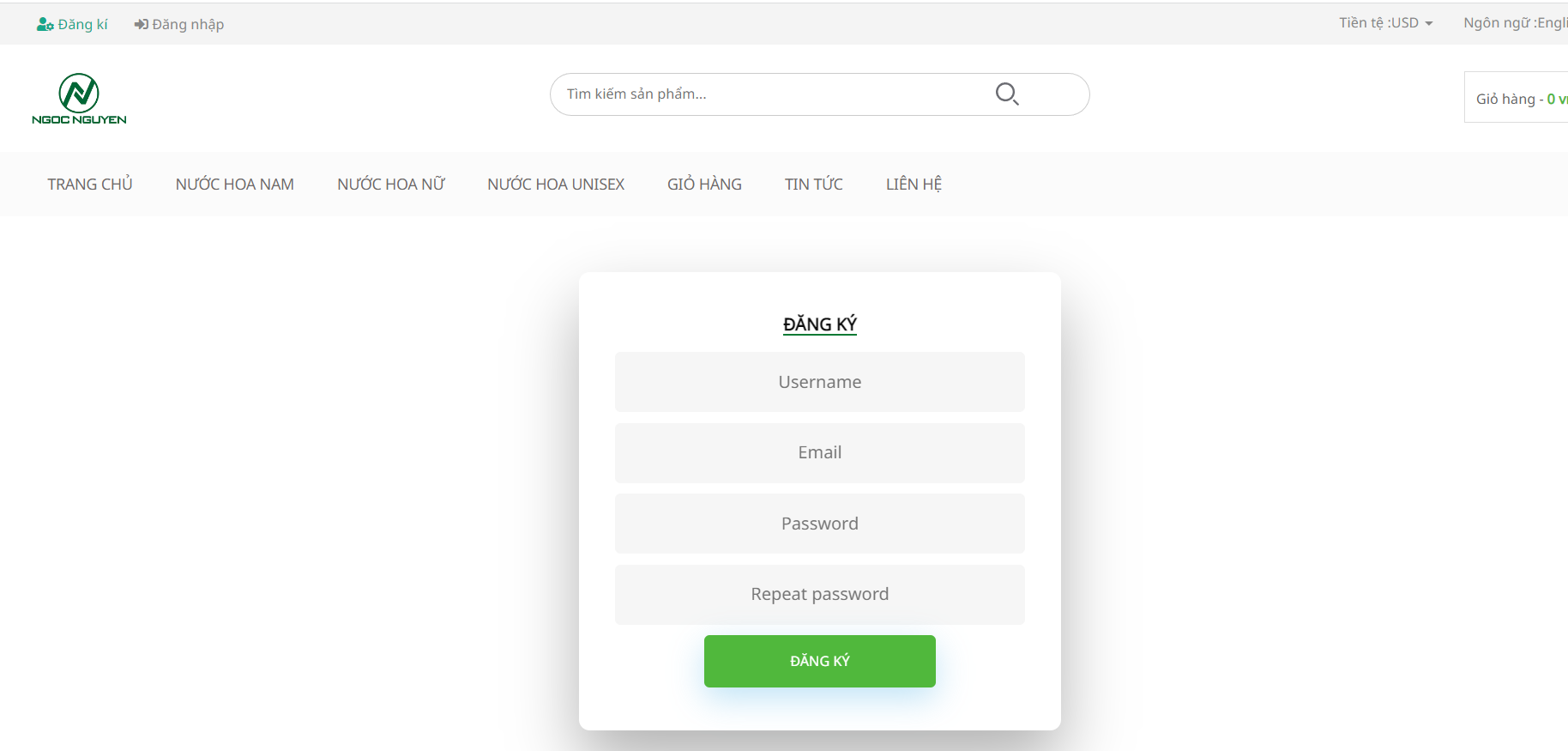
Hình 3. 38 Giao diện trang chủ

Giao diện trang đăng nhập được minh họa bằng hình ảnh dưới đây

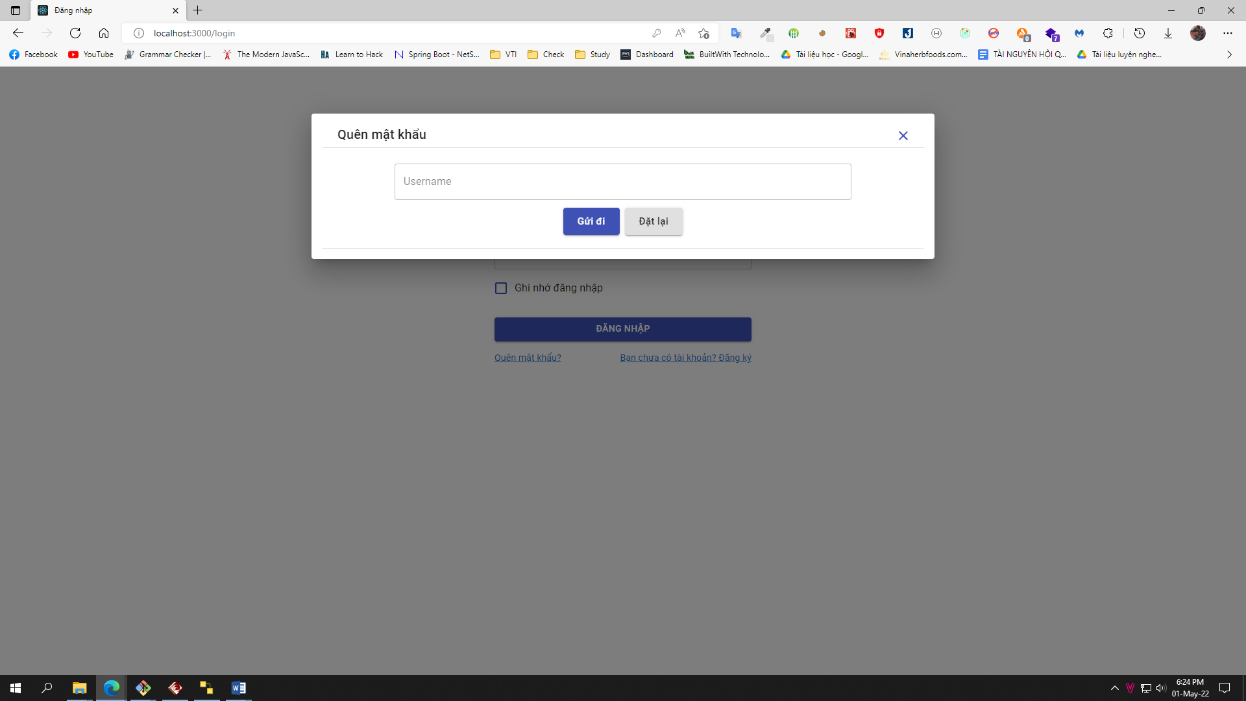


Hình 3. 39 Giao diện trang đăng nhập

Khách hàng có thể đăng ký tài khoản theo hình minh họa dưới đây

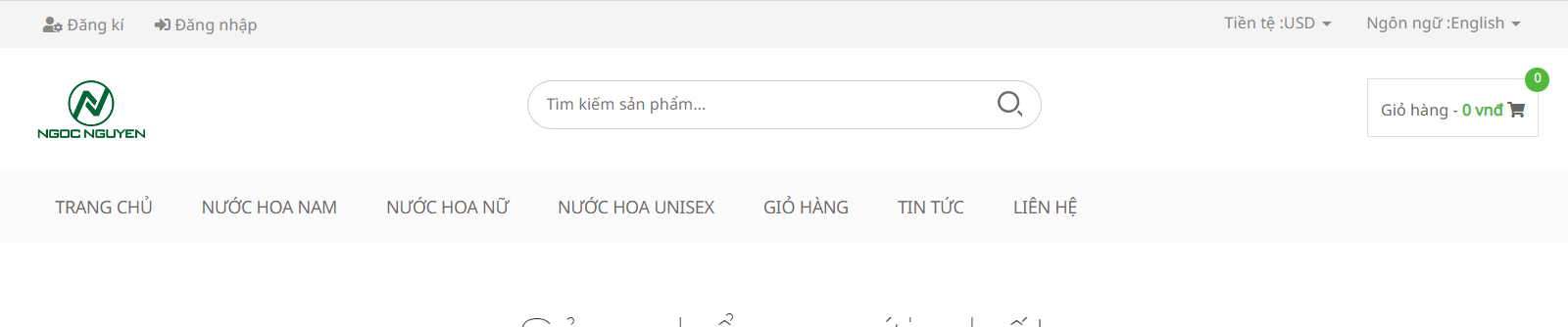


Hình 3. 40 Giao diện trang đăng kí

Khi quên mật khẩu, khách hàng có thể sử dụng chức năng quên mật khẩu

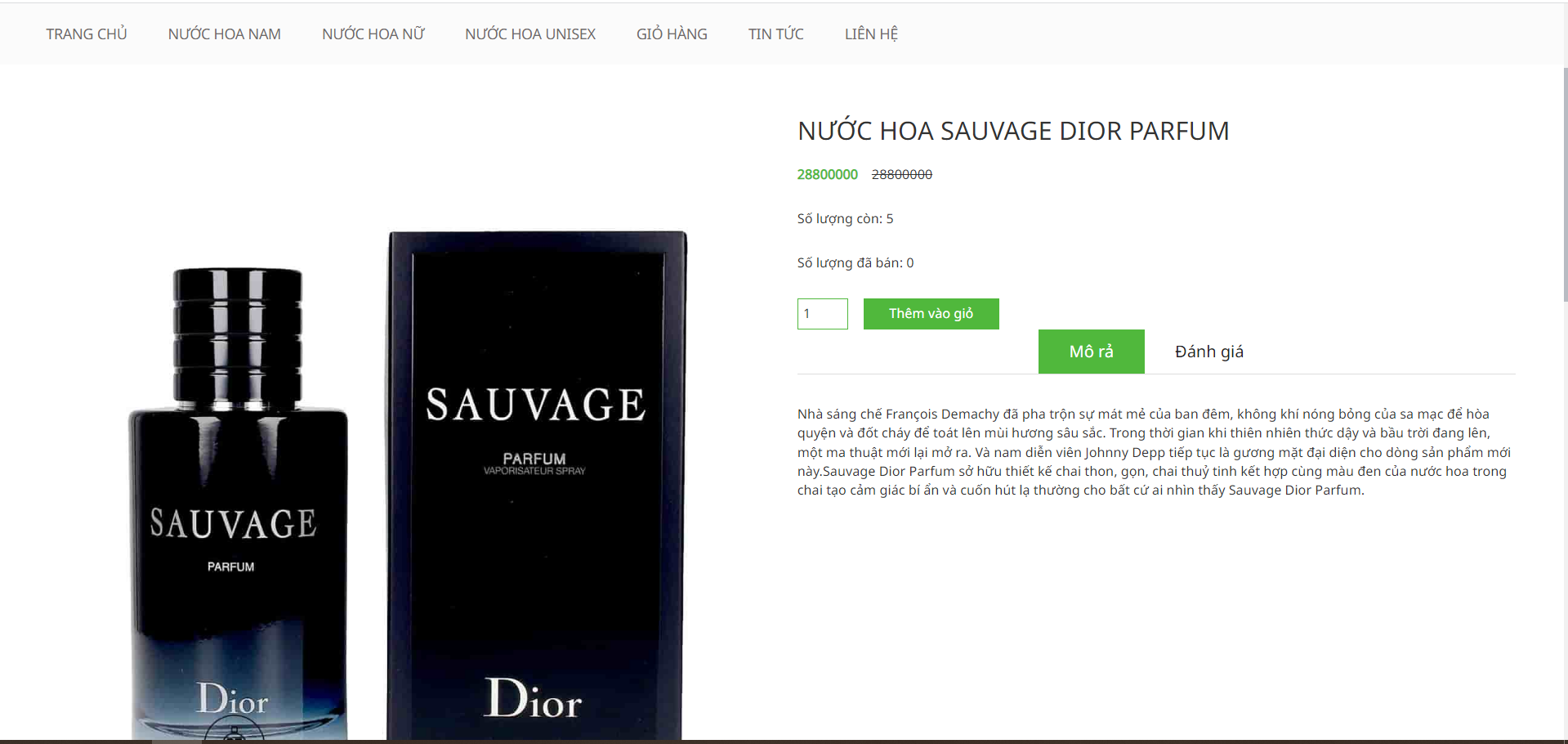
Hình 3. 41 Giao diện chức năng quên mật khẩu

Tại trang chủ, người dùng có thể thực hiện tìm kiếm sản phẩm theo tên, danh mục



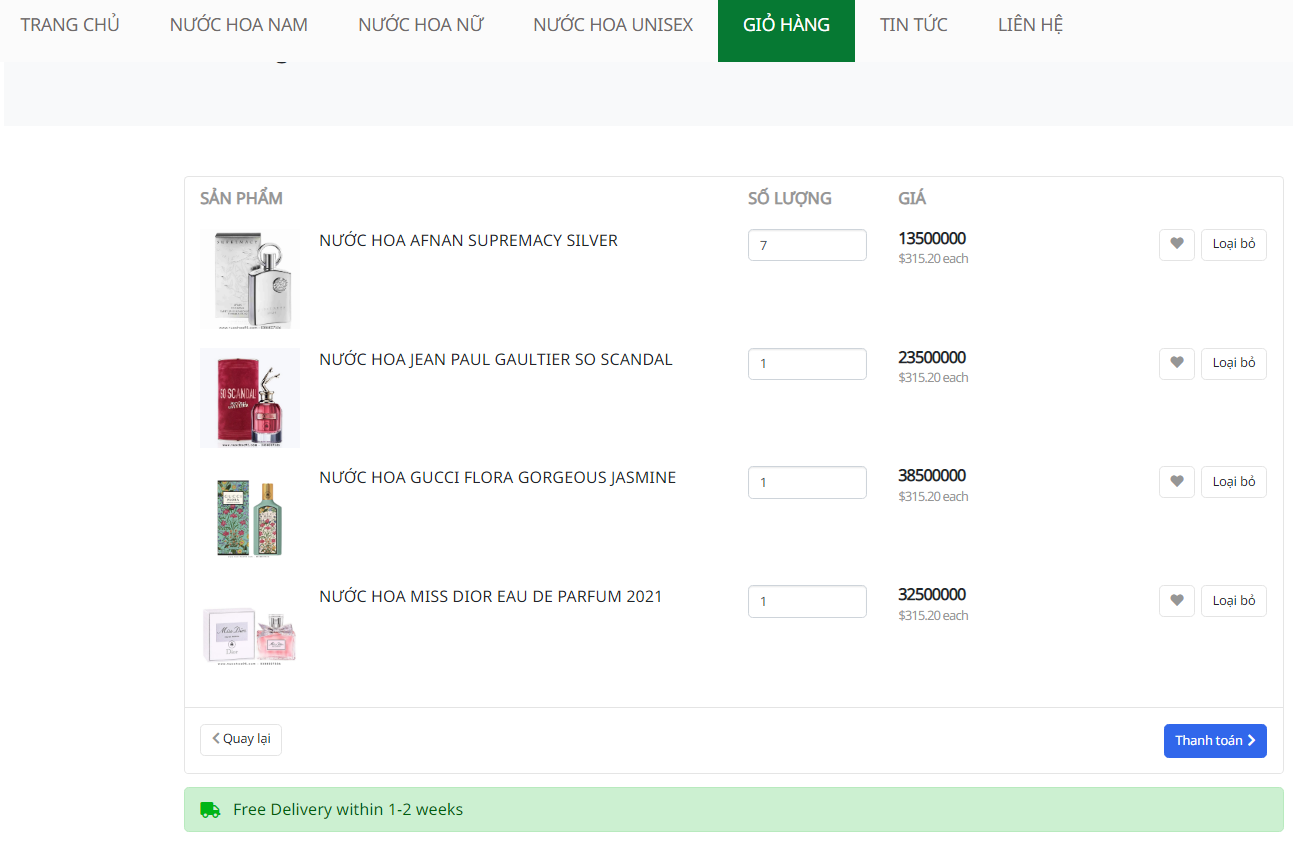
Hình 3. 42 Giao diện chức năng tìm kiếm sản phẩm

Tại màn hình chi tiết sản phẩm, khách hàng có thể chọn số lượng và thêm vào giỏ hàng



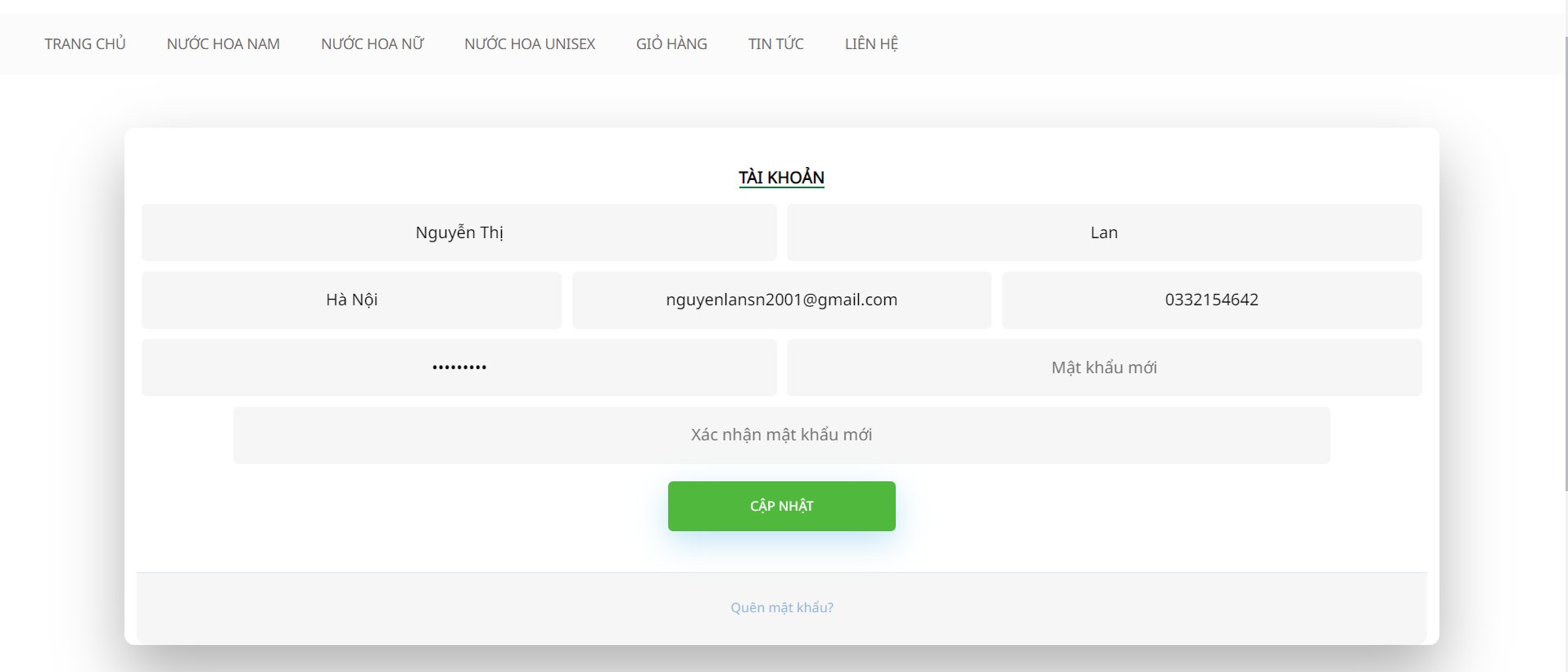
Hình 3. 43 Giao diện trang chi tiết sản phẩm

Sau khi khách hàng thêm các sản phẩm vào giỏ hàng, khách hàng bấm vào biểu tượng giỏ hàng để xem hoặc thực hiện đặt mua hàng



Hình 3. 44 Giao diện trang giỏ hàng

Khi khách hàng muốn thay đổi thông tin cá nhân, khách hàng sẽ bấm vào Thông tin tài khoản trong mục profile tại góc trên bên phải giao diện



Hình 3. 45 Giao diện trang thông tin tài khoản

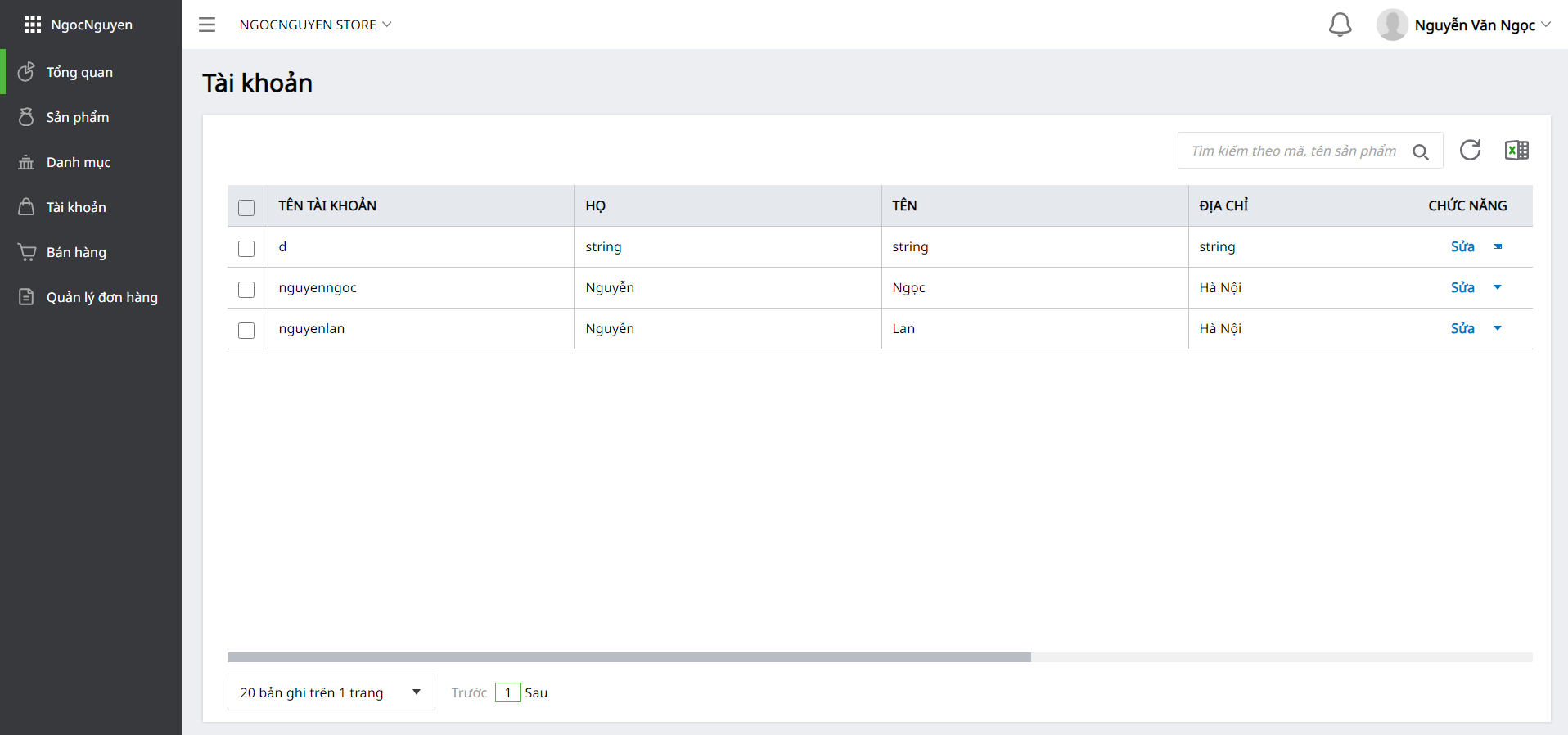
Khách hàng có thể xem các đơn hàng của mình tại mục đơn hàng trong trang profile

#### Giao diện trang admin

Màn hình trang chủ dành cho người quản trị, giúp người quản trị xem danh sản phẩm các đơn hàng, lọc các đơn hàng theo khoảng ngày, thống kê tổng tiền

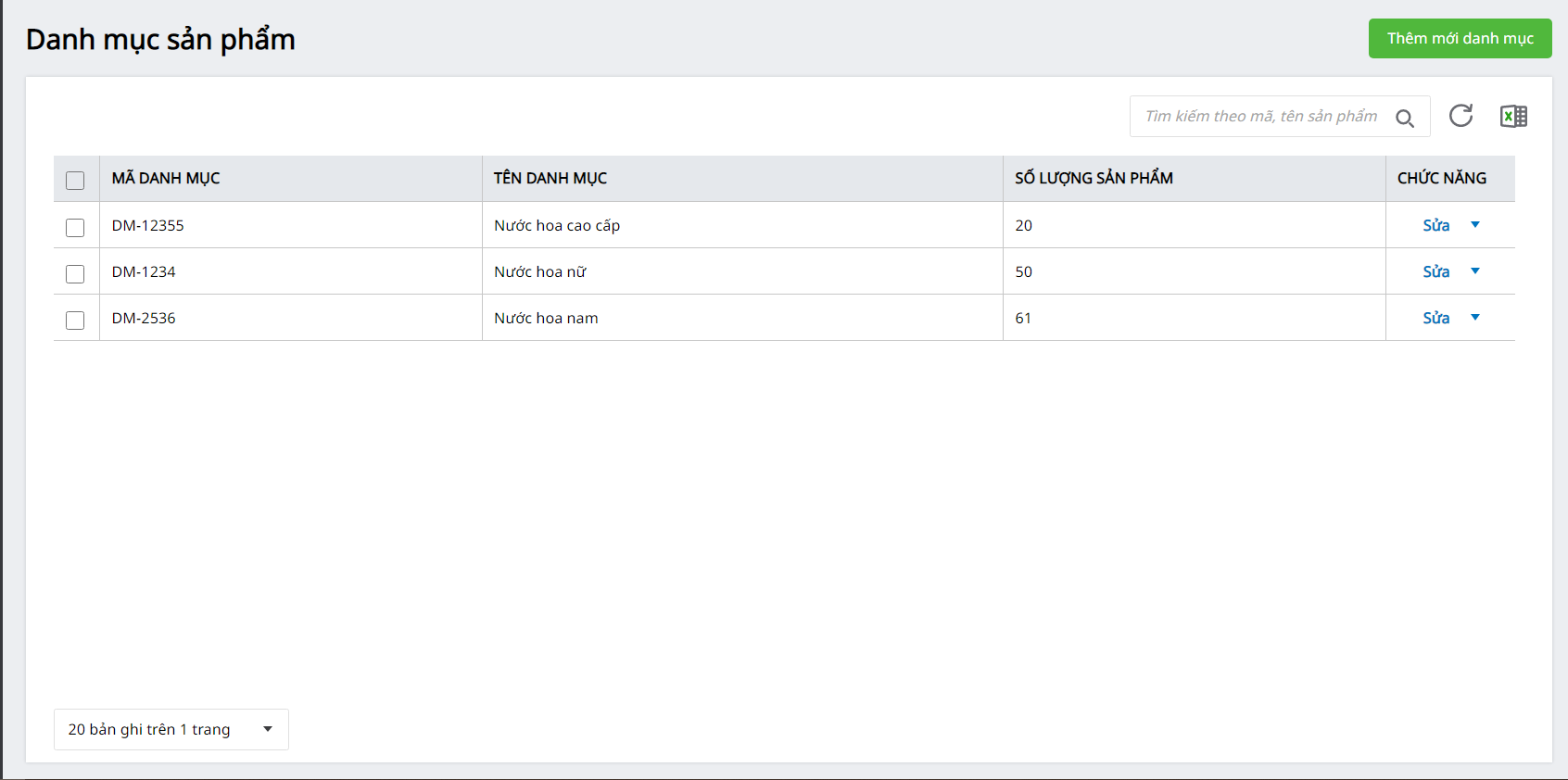
Hình 3. 46 Giao diện trang chủ Admin

Giao diện trang quản lý tài khoản, tại đây người quản trị có thể thực hiện các thao tác: xem thông tin chi tiết tài khoản, thêm, sửa, xóa tài khoản



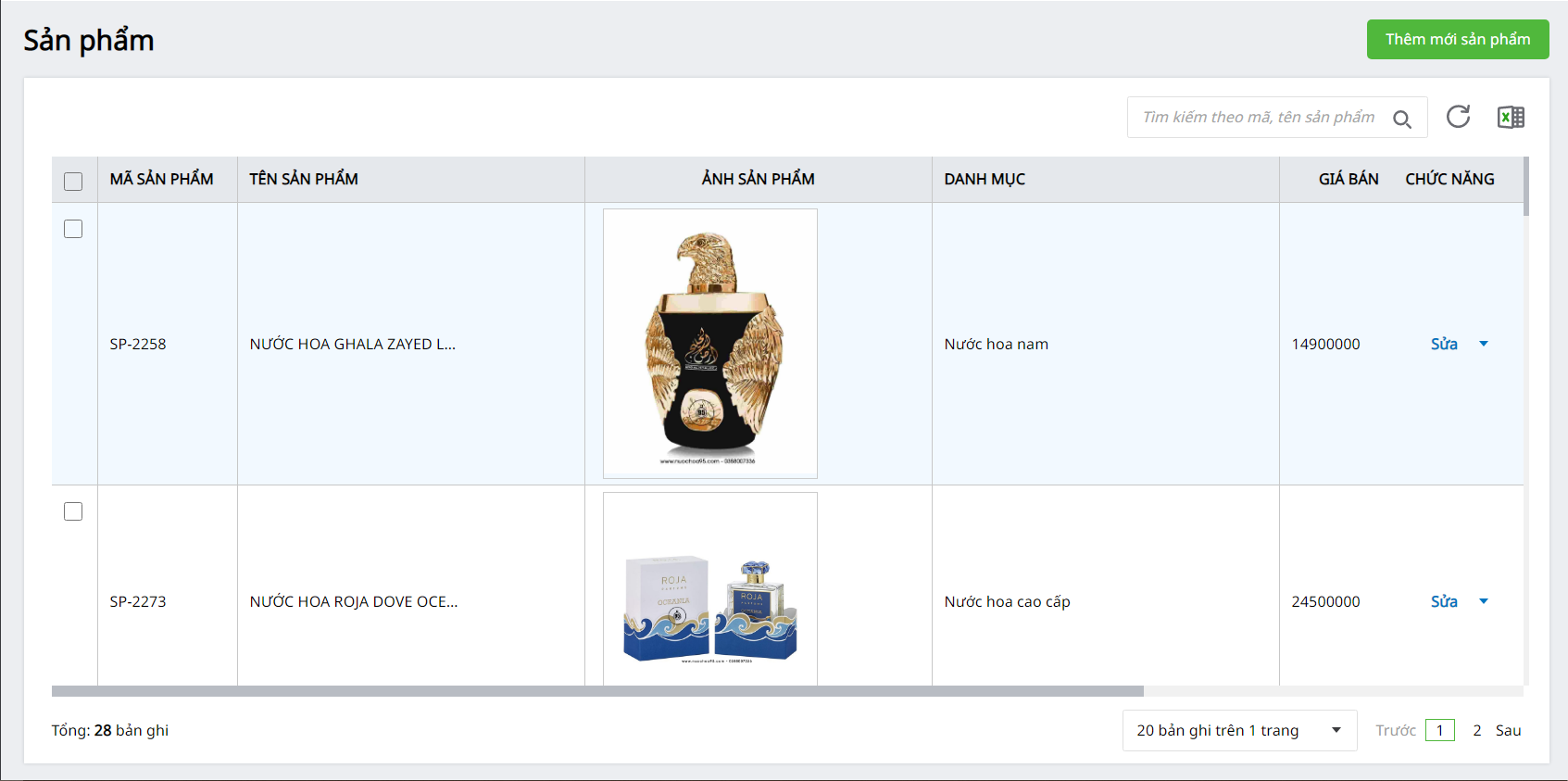
Hình 3. 47 Giao diện trang quản lí tài khoản

Giao diện trang quản lý danh mục, tại đây người quản trị có thể thực hiện các thao tác: xem thông tin chi tiết danh mục, thêm, sửa, xóa danh mục



Hình 3. 48 Giao diện trang quản lí danh mục

Dưới đây là hình ảnh của màn hình quản lý sản phẩm, tại đây người quản trị có thể thực hiện các thao tác: xem danh sản phẩm sản phẩm, xem thông tin chi tiết của sản phẩm, thêm, sửa, xóa sản phẩm



Hình 3. 49 Giao diện trang quản lí sản phẩm

Giao diện trang quản lý hình ảnh sản phẩm, tại đây người quản trị có thể thực hiện các thao tác: xem thêm, sửa, xóa hình ảnh sản phẩm

Hình 3. 50 Giao diện trang quản lí hình ảnh sản phẩm

Dưới đây là hình ảnh của màn hình quản lý đơn hàng, tại đây người quản trị có thể thực hiện các thao tác: xem danh sản phẩm đơn hàng, xem chi tiết đơn hàng, cập nhật trạng thái đơn hàng, tìm kiếm theo mã đơn hàng, lọc danh sản phẩm đơn hàng theo trạng thái



Hình 3. 51 Giao diện trang quản lí đơn hàng

Người quản trị có thể thực hiện thao tác in đơn hàng trong hộp thoại xem chi tiết đơn hàng

Hình 3. 52 Chi tiết đơn hàng

# KẾT LUẬN

Đề tài **“Xây dựng website bán nước hoa cho Shop NgocNguyen”** xuất phát từ những nhu cầu thực tế mà ngày nay trong ngành kinh doanh cần có. Những phương pháp bán hàng truyền thống dần lỗi thời và thay vào đó việc bán hàng qua website nâng cao hiệu quả hơn, khắc phục được những tồn đọng khó khăn mà phương pháp thủ công đang gặp phải.

Qua quá trình thực hiện đề tài, em đã tìm hiểu, tích lũy và học hỏi thêm được nhiều kinh nghiệm cũng như kiến thức công nghệ mới.

**\*Những kết quả đạt được:**

**Về công nghệ:**

* Hiểu được quá trình thiết kế một website thực tế đi từ bước cơ bản: khảo sát dự án, phân tích hệ thống, thiết kế, thực hiện, kiểm thử, triển khai, bảo trì.
* Xây dựng thành công website đáp ứng nhu cầu bán hàng và quảng bá thương hiệu cho hãng sản phẩm NgocNguyen.
* Nắm được các kiến thức xây dựng giao diện website: HTML, CSS, JavaScript.
* Nắm vững về cách xây dựng website thông qua thư viện, framework: VueJS, C#,ASP.NET trong lập trình.
* Sử dụng thành thạo các công cụ hỗ trợ lập trình: Visual Studio Code, Visual Studio

**Về cài đặt chương trình:**

* Cho phép khách hàng có thể xem thông tin sản phẩm và có thể mua hàng trực tuyến.
* Tiếp nhận đơn đặt hàng của khách hàng.
* Các mặt hàng được cập nhật liên tục theo nhu cầu của khách hàng.
* Trang quản trị đầy đủ chức năng để quản lý một trang web: quản lý tài khoản, danh mục. sản phẩm, đơn hàng.

**Tính năng khác:**

* Giao diện đẹp gây ấn tượng thân thiện với khách hàng.
* Tính bảo mật.

**\*Hướng phát triển**

* Tiếp tục hoàn thiện các tính năng mở rộng phần mềm nhằm cải tiến và nâng cấp chương trình.
* Hoàn thiện chương trình, sửa chữa các lỗi, cải thiện tính bảo mật.
* Xây dựng Website quy mô lớn hơn với nhiều ứng dụng như: trò chuyện, tư vấn khách hàng.
* Tích hợp thêm chức năng thanh toán với ví điện tử.

# TÀI LIỆU THAM KHẢO

[1] Hoàng Quang Huy, Phùng Đức Hòa, Trịnh Bá Quý, *Nhập môn công nghệ phần mềm*, NXB Đại học Công nghiệp Hà Nội.

[2] Nguyễn Thị Thanh Huyền, Ngô Thị Bích Thúy, Phạm Thị Kim Phượng, *Giáo trình phân tích thiết kế hệ thống*, NXB Giáo dục VN.

[3] Nguyễn Xuân Hoàng, *Lập trình ASP.NET*, NXB thông tin và truyền thông.

[4] *Giáo trình thiết kế web*, Trường đại học Công Nghiệp Hà Nội.

[5] Trang tài liệu UML <https://www.tutorialspoint.com/uml/index.htm>

[6] Trang tài liệu C# https://learn.microsoft.com/vi-vn/dotnet/csharp/

[7] Trang tài liệu ASP.NET https://dotnet.microsoft.com/en-us/apps/aspnet/apis

[8] Trang tài liệu JavaScript <https://javascript.info>

[9] Trang tài liệu VueJS https://vuejs.org/

[10] Trang tin tức về lập trình <https://viblo.asia>