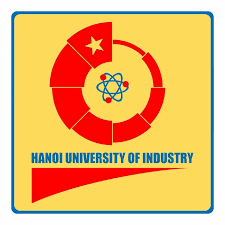
**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**-------------------------**



**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**

**XÂY DỰNG WEBSITE BÁN LAPTOP**

NGÀNH CÔNG NGHỆ THÔNG TIN

|  |  |
| --- | --- |
| **GVHD:** | **ThS. Nguyễn Tuấn Tú** |
| **Sinh viên:** | **Phạm Quý Dương** |
| **Lớp: CNTT05** | **Khoá: 15** |
|  |  |
|  |  |
|  |  |
|  |  |

**Hà nội - Năm 2024**

MỤC LỤC

[**MỤC LỤC** ii](#_Toc166526136)

[**LỜI MỞ ĐẦU** v](#_Toc166526137)

[**DANH MỤC HÌNH ẢNH** vi](#_Toc166526138)

[CHƯƠNG 1: TỔNG QUAN VỀ NỘI DUNG NGHIÊN CỨU 1](#_Toc166526139)

[1.1 Giới thiệu chung 1](#_Toc166526140)

[1.1.1 Lý do chọn đề tài 1](#_Toc166526141)

[1.1.2 Mục tiêu của đề tài 2](#_Toc166526142)

[1.1.3 Kết quả dự kiến đạt được 3](#_Toc166526143)

[1.1.4 Cấu trúc báo cáo 3](#_Toc166526144)

[1.2 Cơ sở lý thuyết 4](#_Toc166526145)

[1.2.1 Ngôn ngữ lập trình JavaScript 4](#_Toc166526146)

[1.2.2 Nodejs Platform 7](#_Toc166526147)

[1.2.3 Express Framework 10](#_Toc166526148)

[1.2.4 React Framework 13](#_Toc166526149)

[1.2.5 Cơ sở dữ liệu MongoDB 16](#_Toc166526150)

[1.3 Công cụ sử dụng 18](#_Toc166526151)

[CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 19](#_Toc166526152)

[2.1 Vai trò của người dùng 19](#_Toc166526154)

[2.2 Yêu cầu chức năng 19](#_Toc166526155)

[2.3 Yêu cầu phi chức năng 21](#_Toc166526156)

[2.3.1 Giao diện người dùng 21](#_Toc166526157)

[2.3.2 Tính bảo mật và ràng buộc 22](#_Toc166526158)

[2.4 Biểu đồ use case tổng quát 23](#_Toc166526159)

[2.5 Mô hình thực thể 24](#_Toc166526160)

[2.6 Đặc tả usecase 24](#_Toc166526161)

[2.6.1 Use case đăng nhập 24](#_Toc166526162)

[2.6.2 Use case đăng ký 27](#_Toc166526163)

[2.6.3 Use case tìm kiếm sản phẩm 28](#_Toc166526164)

[2.6.4 Use case Xem chi tiết sản phẩm 30](#_Toc166526165)

[2.6.5 Use case thêm quản lý giỏ hàng 32](#_Toc166526166)

[2.6.6 Use case xem giỏ hàng 35](#_Toc166526167)

[2.6.7 Use case Đặt hàng 35](#_Toc166526168)

[2.6.8 Use case Quản lý sản phẩm 38](#_Toc166526169)

[2.6.9 Use case Quản lý đơn hàng 41](#_Toc166526170)

[2.6.10 Use case quản lý tài khoản 42](#_Toc166526171)

[2.6.11 Use case Quản lý danh mục 45](#_Toc166526172)

[2.6.12 Use case Thanh toán 48](#_Toc166526173)

[2.6.13 Use case Thống kê 50](#_Toc166526174)

[2.6.14 Use case Quên mật khẩu 52](#_Toc166526175)

[2.6.15 Use case Liên hệ 54](#_Toc166526176)

[2.6.16 Use case Quản lý bài viết 56](#_Toc166526177)

[2.6.17 Use case xem bài viết 60](#_Toc166526178)

[CHƯƠNG 3: CÀI ĐẶT VÀ KẾT QUẢ ĐẠT ĐƯỢC 62](#_Toc166526179)

[3.1 Công cụ sử dụng 62](#_Toc166526181)

[3.2 Kết quả đạt được 62](#_Toc166526182)

[3.2.1 Giao diện chức năng chính của khách hàng 62](#_Toc166526183)

[3.2.2 Giao diện chức năng của người quản trị 65](#_Toc166526184)

[3.3 Kiểm thử hệ thống 68](#_Toc166526185)

[3.3.1 Kế hoạch kiểm thử 68](#_Toc166526186)

[3.3.2 Kết quả kiểm thử 68](#_Toc166526187)

[**KẾT LUẬN** 70](#_Toc166526188)

[**TÀI LIỆU THAM KHẢO** 71](#_Toc166526189)

**LỜI MỞ ĐẦU**

*Lời đầu tiên cho phép em gửi lời cảm ơn tới các thầy cô trong* ***khoa Công nghệ thông tin - Trường Đại học Công Nghiệp Hà Nội****. Những người đã hết mình truyền đạt và chỉ dẫn cho chúng em những kiến thức, những bài học quý báu và bổ ích trong suốt kỳ học vừa qua. Để hoàn thành được kì thực tập này, em xin được bày tỏ sự tri ân và xin chân thành cảm ơn giảng viên* ***ThS. Nguyễn Tuấn Tú****. Trong quá trình thực tập, em đã nhận được sự quan tâm giúp đỡ, hướng dẫn tận tình, tâm huyết của thầy. Những gì em nhận được không chỉ dừng lại ở các hướng dẫn thực tập mà nhiều hơn thế đó là những lời khuyên, chia sẻ thực tế từ thầy. Chính nhờ phương pháp truyền đạt của thầy mà em có thêm cách nhìn tổng quan về kì thực tập và phát huy khả năng của bản thân. Qua kì thực tập này là cơ hội tuyệt vời giúp em rèn luyện sự tự tin, kỹ năng giao tiếp, làm việc nhóm,... Đây cũng chính là hành trang quan trọng giúp chúng em tự tin bước chân vào môi trường làm việc thực tế.*

**DANH MỤC HÌNH ẢNH**

[Hình 1.1. Kiến trúc của Nodejs 9](#_Toc159768808)

[Hình 1.2. Logo Expressjs 11](#_Toc159768809)

[Hình 1.3. Logo Reactjs Framework 14](#_Toc159768810)

[Hình 2.1: Biểu đồ use case tổng quát 23](#_Toc159768811)

[Hình 2.2: Biểu đồ thực thế 24](#_Toc159768812)

[Hình 2.3: Biều đồ trình tự use case Đăng nhập 26](#_Toc159768813)

[Hình 2.4: Biều đồ trình tự use case Đăng ký 28](#_Toc159768814)

[Hình 2.5: Biều đồ trình tự use case Tìm kiếm sản phẩm 30](#_Toc159768815)

[Hình 2.6: Biều đồ trình tự use case Xem chi tiết sản phẩm 31](#_Toc159768816)

[Hình 2.7: Biều đồ trình tự use case Quản lý giỏ hàng 34](#_Toc159768817)

[Hình 2.8: Biều đồ trình tự use case Đặt hàng 37](#_Toc159768818)

[Hình 2.9: Biều đồ trình tự use case Quản lý sản phẩm 40](#_Toc159768819)

[Hình 2.10: Biều đồ trình tự use case Đơn hàng 42](#_Toc159768820)

[Hình 2.11: Biều đồ trình tự use case Tài khoản 45](#_Toc159768821)

[Hình 2.12: Biều đồ trình tự use case Danh mục 48](#_Toc159768822)

[Hình 2.13: Biều đồ trình tự use case Thanh toán 50](#_Toc159768823)

[Hình 2.14: Biều đồ trình tự use case Thống kê 52](#_Toc159768824)

[Hình 2.15: Biểu đồ trình tự use case quên mật khẩu 54](#_Toc159768825)

[Hình 2.16: Biều đồ trình tự use case Liên hệ 56](#_Toc159768826)

[Hình 2.17: Biều đồ trình tự use case Quản lý bài viết 60](#_Toc159768827)

[Hình 2.18: Biều đồ trình tự use case Quản lý bài viết 61](#_Toc159768828)

[Hình 3.1: Giao diện trang chủ website 62](#_Toc159768829)

[Hình 3.2: Giao diện danh sách sản phẩm 63](#_Toc159768830)

[Hình 3.3: Giao diện chi tiết sản phẩm 63](#_Toc159768831)

[Hình 3.4: Giao diện giỏ hàng 64](#_Toc159768832)

[Hình 3.5: Giao diện mua hàng 64](#_Toc159768833)

[Hình 3.6: Giao diện danh sách bài viết 65](#_Toc159768834)

[Hình 3.7: Giao diện trang chủ trang quản trị 65](#_Toc159768835)

[Hình 3.8: Giao diện quản lý tài khoản 66](#_Toc159768836)

[Hình 3.9: Giao diện quản lý danh mục 66](#_Toc159768837)

[Hình 3.10: Giao diện quản lý sản phẩm 67](#_Toc159768838)

[Hình 3.11: Giao diện quản lý bài viết 67](#_Toc159768839)

[Hình 3.12: Giao diện quản lý đơn hàng 68](#_Toc159768840)

[Hình 3.13: Thống kê kết quả kiểm thử 69](#_Toc159768841)

# TỔNG QUAN VỀ NỘI DUNG NGHIÊN CỨU

## Giới thiệu chung

Cửa hàng bán laptop chuyên cung cấp các loại laptop phù hợp với nhu cầu của mọi đối tượng khách hàng, từ sinh viên, người đi làm, đến các game thủ hay chuyên gia đồ họa. Với tiêu chí khách hàng là số một, chúng tôi không chỉ đảm bảoề chất lượng sản phẩm mà còn coi trọng dị vụ hỗ trợ khách hàng và sau bán hàng.

Chúng tôi hiểu rằng laptop không chỉ là một công cụ làm việc, mà còn là một phần không thể thiếu trong cuộc sống hiện đại. Do đó, cửa hàng bán laptop luôn cập nhật những mẫu laptop mới nhất, đa dạng về thương hiệu và phân khúc giá để phục vụ đa dạng nhu cầu của khách hàng.

Tiêu chí hoạt động của chúng tôi là cung cấp sản phẩm chất lượng, giá cả cạnh tranh và dịch vụ hậu mãi tốt nhất. Chúng tôi luôn lắng nghe, thấu hiểu và đáp ứng nhu cầu của khách hàng, từ đó xây dựng niềm tin và sự hài lòng cho khách hàng khi mua sắm tại cửa hàng bán laptop.

### Lý do chọn đề tài

Thương mại điện tử hiện nay đang ngày càng phát triển mạnh mẽ, trở thành một hình thức kinh doanh phổ biến và hiệu quả. Đặc biệt, trong lĩnh vực bán lẻ các sản phẩm công nghệ như laptop, việc kinh doanh trực tuyến đã mang lại nhiều cơ hội và lợi ích cho các doanh nghiệp.

Laptop là một sản phẩm công nghệ không thể thiếu trong cuộc sống hiện đại, từ công việc, học tập cho đến giải trí. Ngày càng nhiều người lựa chọn mua laptop trực tuyến để tiết kiệm thời gian, công sức và có thể so sánh giá cả, chất lượng dễ dàng. Vì vậy, việc xây dựng một website bán laptop chuyên nghiệp, đáng tin cậy là vô cùng cần thiết.

Đối với website bán laptop, bên cạnh giao diện thân thiện và dễ sử dụng, thông tin về sản phẩm cũng cần được cung cấp đầy đủ và chính xác, giúp người mua có thể hiểu rõ về sản phẩm và lựa chọn một cách dễ dàng. Ngoài ra, website còn cần có chức năng hỗ trợ khách hàng, giải đáp thắc mắc kịp thời, từ đó nâng cao sự hài lòng của khách hàng và tạo niềm tin với thương hiệu.

Nhận thức được sự quan trọng của việc này, em quyết định chọn đề tài ***"Xây dựng website bán laptop"*** để áp dụng những kiến thức đã học, đồng thời tìm hiểu thêm về thực tế kinh doanh trực tuyến, từ đó hoàn thiện kỹ năng và kiến thức của mình.

### Mục tiêu của đề tài

Đề tài ***"Xây dựng website bán laptop"*** 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 MongoDB, JavaScript, EJS, Nodejs, ReactJS.
* 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.

### 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.

- Verify email của khách hàng.

- 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, flashsale, voucher, doanh thu.

- Tìm kiếm sản phẩm, tìm kiếm sản phẩm theo tên laptop, 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, thanh toán khi nhận hàng.

### Cấu trúc báo cáo

Ngoài phần Mở đầu và Kết luận, nội dung quyển Báo cáo bài tập lớn còn bao gồm 3 chương

+Mở đầu: Trình bày về lý do chọn đề tài, mục tiêu nghiên cứu, đối tượng nghiên cứu, phạm vi nghiên cứu, kết quả dự kiến đạt được, …

+Chương 1: Tổng quan về nội dung nghiên cứu

+Chương 2: Phân tích thiết kế hệ thống.

+Chương 3: Triển khai và kiểm thử hệ thống.

## Cơ sở lý thuyết

### Ngôn ngữ lập trình JavaScript

#### JavaScript là gì?

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.

#### Ưu nhược điểm của ngôn ngữ lập trình Javascript

Là một ngôn ngữ lập trình rất phổ biến hiện nay, ***Javascript***có cho mình rất nhiều ưu điểm nổi bật. Tuy nhiên, bên cạnh những ưu điểm đó thì nhược điểm của ngôn ngữ lập trình này cũng cần được lưu ý.

**Ưu điểm:**

* Những lỗi Javascript rất dễ để phát hiện, từ đó giúp bạn sửa lỗi một cách nhanh chóng hơn.
* Những trình duyệt web có thể dịch thông qua HTML mà không cần sử dụng đến một compiler.
* JS có thể hoạt động ở trên nhiều nền tảng và các trình duyệt web khác nhau.
* Được các chuyên gia đánh giá là một loại ngôn ngữ lập trình nhẹ và nhanh hơn nhiều so với các ngôn ngữ lập trình khác.
* JS còn có thể được gắn trên một số các element hoặc những events của các trang web.
* Những website có sử dụng JS thì chúng sẽ giúp cho trang web đó có sự tương tác cũng như tăng thêm nhiều trải nghiệm mới cho người dùng.
* Người dùng cũng có thể tận dụng JS với mục đích là để kiểm tra những input thay vì cách kiểm tra thủ công thông qua hoạt động truy xuất database.
* Chương trình rất dễ học.
* Giao diện của ứng dụng phong phú với nhiều thành phần như Drag and Drop, Slider để cung cấp đến cho người dùng một Rich Interface (giao diện giàu tính năng).
* Giúp thao tác với người dùng phía Client và tách biệt giữa các Client với nhau.

**Nhược điểm:**

* JS Code Snippet khá lớn.
* JS dễ bị các hacker và scammer khai thác hơn.
* JS cũng không có khả năng đa luồng hoặc đa dạng xử lý.
* Có thể được dùng để thực thi những mã độc ở trên máy tính của người sử dụng.
* Những thiết bị khác nhau có thể sẽ thực hiện JS khác nhau, từ đó dẫn đến sự không đồng nhất.
* Vì tính bảo mật và an toàn nên các Client-Side *Javascript* sẽ không cho phép đọc hoặc ghi các file.
* JS không được hỗ trợ khi bạn sử dụng ở trong tình trạng thiết bị được kết nối mạng.

### Nodejs Platform

#### Nodejs Platfrom là gì?

Nodejs là một nền tảng (Platform) phát triển độc lập được xây dựng ở trên Javascript Runtime của Chrome mà chúng ta có thể xây dựng được các ứng dụng mạng một cách nhanh chóng và dễ dàng mở rộng.

Nodejs được xây dựng và phát triển từ năm 2009, bảo trợ bởi công ty Joyent, trụ sở tại California, Hoa Kỳ.

Phần Core bên dưới của Nodejs được viết hầu hết bằng C++ nên cho tốc độ xử lý và hiệu năng khá cao.

Nodejs tạo ra được các ứng dụng có tốc độ xử lý nhanh, realtime thời gian thực.

Nodejs áp dụng cho các sản phẩm có lượng truy cập lớn, cần mở rộng nhanh, cần đổi mới công nghệ, hoặc tạo ra các dự án Startup nhanh nhất có thể.

#### Những ứng dụng nên viết bằng Nodejs

* Websocket server: Các máy chủ web socket như là Online Chat, Game Server…
* Fast File Upload Client: là các chương trình upload file tốc độ cao.
* Ad Server: Các máy chủ quảng cáo.
* Cloud Services: Các dịch vụ đám mây.
* RESTful API: đây là những ứng dụng mà được sử dụng cho các ứng dụng khác thông qua API.
* Any Real-time Data Application: bất kỳ một ứng dụng nào có yêu cầu về tốc độ thời gian thực. Micro Services: Ý tưởng của micro services là chia nhỏ một ứng dụng lớn thành các dịch vụ nhỏ và kết nối chúng lại với nhau. Nodejs có thể làm tốt điều này.

#### Tại sao nên sử dụng Nodejs

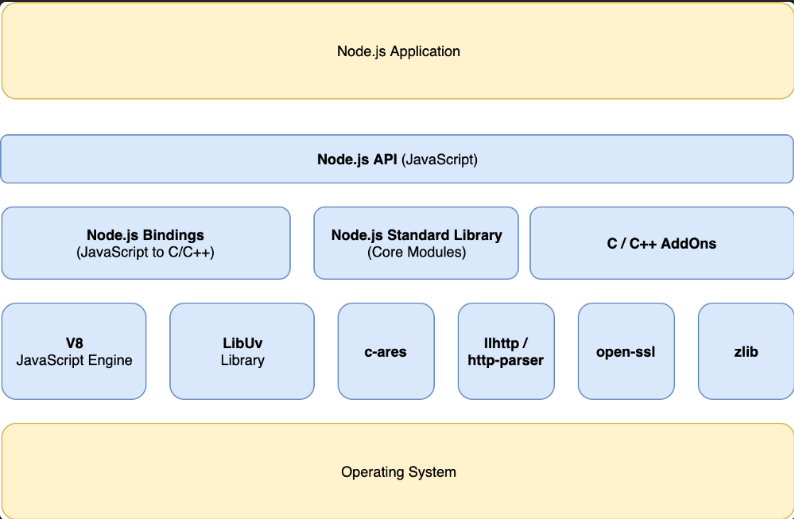
Các ứng dụng Nodejs được viết bằng **javascript**, ngôn ngữ này là một ngôn ngữ khá thông dụng. Theo tác giả của ngôn ngữ Javascript, Ryan Dahl: “Javascript có những đặc tính mà làm cho nó rất khác biệt so với các ngôn ngữ lập trình động còn lại, cụ thể là nó không có khái niệm về đa luồng, tất cả là đơn luồng và hướng sự kiện.”

Nodejs chạy đa nền tảng phía Server, sử dụng kiến trúc hướng sự kiện Event-driven, cơ chế non-blocking I/O làm cho nó nhẹ và hiệu quả.

Có thể chạy ứng dụng Nodejs ở bất kỳ đâu trên máy Mac – Window – Linux, hơn nữa cộng đồng Nodejs rất lớn và hoàn toàn miễn phí. Các bạn có thể thấy cộng đồng Nodejs lớn như thế nào, các package đều hoàn toàn free.

Các ứng dụng NodeJS đáp ứng tốt thời gian thực và chạy đa nền tảng, đa thiết bị.

#### Kiến trúc của Nodejs



Hình 1.1. Kiến trúc của Nodejs

* V8: Đây là bộ Engine V8 của chrome, bao gồm Memory Heap, Call Stack, Garbage Collector và chuyển đổi mã Javscript thành mã máy của hệ điều hành.
* Libuv: Là một thư viện quan trọng, bao gồm Thread Pool và Event Loop, Event Queue. Nó là thư viện C đa nền tảng tập trung vào các tác vụ I/O không đồng bộ.
* Node.js Standard Library: Bao gồm các thư viện, các chức năng liên quan đến hệ điều hành cho Timers như setTimeout, File system như fs, Network Calls như http.
* llhttp: Phân tích request/response HTTP (trước đây là http-parse).
* c-ares: Thư viện C cho DNS không đồng bộ được sử dụng trong dns module.
* open-ssl: Các chức năng mã hóa được sử dụng trong TLS (SSL), crypto module.
* zlib: Nén và giải nén bằng cách chạy đồng bộ, không đồng bộ và streaming.
* Node.js API: Cung cấp JavaScript API được sử dụng bởi các ứng dụng.

### Express Framework

#### Express Framework là gì?

Expressjs hay còn được viết là Express js, Express.js. Đây là một framework mã nguồn mở miễn phí cho Node.js. Express.js được sử dụng trong thiết kế và xây dựng các ứng dụng web một cách đơn giản và nhanh chóng.

Vì Express js chỉ yêu cầu ngôn ngữ lập trình Javascript nên việc xây dựng các ứng dụng web và API trở nên đơn giản hơn với các lập trình viên và nhà phát triển. Expressjs cũng là một khuôn khổ của Node.js do đó hầu hết các mã code đã được viết sẵn cho các lập trình viên có thể làm việc.

Nhờ có Expressjs mà các nhà lập trình có thể dễ dàng tạo các ứng dụng 1 web, nhiều web hoặc kết hợp. Do có dung lượng khá nhẹ, Expressjs giúp cho việc tổ chức các ứng dụng web thành một kiến trúc MVC có tổ chức hơn. Để có thể sử dụng được mã nguồn này, chúng ta cần phải biết về Javascript và HTML.

Expressjs cũng là một phần của công nghệ giúp quản lý các ứng dụng web một cách dễ dàng hơn hay còn được gọi là ngăn xếp phần mềm MEAN.Nhờ có thư viện Javascript của Express js đã giúp cho các nhà lập trình xây dựng nên các ứng dụng web hiệu quả và nhanh chóng hơn. Expressjs cũng được sử dụng để nâng cao các chức năng của Node.js.



Hình 1.2. Logo Expressjs

Hiện tại, nó đã đi trước các đối thủ chính như Angular và [Bootstrap](https://getbootstrap.com/). Được đánh giá cao và trở nên rất phổ biến với hơn 1.300 developer và hơn 94.000 trang web đang sử dụng.

#### Tại sao sử dụng ExpressJS

Javascript là một ngôn ngữ lập trình được sử dụng rộng rãi hiện nay, dễ đọc và được hỗ trợ ở mọi nơi. Mà Expressjs lại hỗ trợ Javascript, do đó nếu bạn đã biết đến Javascript thì chắc chắn việc lập trình bằng Express.js là vô cùng đơn giản. Ngay cả những người mới bắt đầu tham gia vào lĩnh vực phát triển web này cũng có thể sử dụng Expressjs.

Javascript là ngôn ngữ lập trình dễ học với cả những người không có bất kỳ kiến thức gì về ngôn ngữ lập trình khác. Chính vì tính phổ biến, dễ học và dễ sử dụng này mà Express.js cho phép các tài năng trẻ tham gia và đạt được nhiều thành công trong phát triển ứng dụng web.

**Hỗ trợ xây dựng website một cách nhanh chóng**

Thời gian chính là tài sản quý giá của bất kỳ doanh nghiệp nào. Hơn thế, nhiều lập trình viên còn phải chịu áp lực xây dựng các ứng dụng web một cách hiệu quả trong thời gian ngắn. Nhưng để thực hiện tốt công việc này cần rất nhiều thời gian và Express.js chính là công cụ cứu cánh cho các nhà lập trình.

Express.js có thể giúp làm giảm một nửa thời gian viết mã mà vẫn xây dựng lên các ứng dụng web hiệu quả. Không chỉ trợ giúp về mặt thời gian Expressjs còn làm giảm những áp lực cần thiết để xây dựng với sự trợ giúp của các tính năng khác nhau của nó. Express js còn cung cấp một phần mềm trung gian đảm nhận nhiệm vụ đưa ra các quyết định để phản hồi chính xác những yêu cầu của khách hàng.

**Express.js hoàn toàn miễn phí**

Một trong những yếu tố để tổ chức, doanh nghiệp nên sử dụng công cụ, ứng dụng nào chính là chi phí. Với một ứng dụng web với nhiều tính năng tuyệt vời như vậy mà lại không tốn một chút kinh phí nào thì không có lý do gì để không sử dụng nó.

#### Tính năng của expressjs

Phát triển máy chủ nhanh chóng: Expressjs cung cấp nhiều tính năng dưới dạng các hàm để dễ dàng sử dụng ở bất kỳ đâu trong chương trình. Điều này đã loại bỏ nhu cầu viết mã từ đó tiết kiệm được thời gian.

Phần mềm trung gian Middleware: Đây là phần mềm trung gian có quyền truy cập vào cơ sở dữ liệu, yêu cầu của khách hàng và những phần mềm trung gian khác. Phần mềm Middleware này chịu trách nhiệm chính cho việc tổ chức có hệ thống các chức năng của Express.js.

Định tuyến - Routing: Express js cung cấp cơ chế định tuyến giúp duy trì trạng thái của website với sự trợ giúp của URL.

Tạo mẫu - Templating: Các công cụ tạo khuôn mẫu được Express.js cung cấp cho phép các nhà xây dựng nội dung động trên các website bằng cách tạo dựng các mẫu HTML ở phía máy chủ.

Gỡ lỗi - Debugging: Để phát triển thành công các ứng dụng web không thể thiết đi việc gỡ lỗi. Giờ đây với Expressjs việc gỡ lỗi đã trở nên dễ dàng hơn nhờ khả năng xác định chính xác các phần ứng dụng web có lỗi.

### React Framework

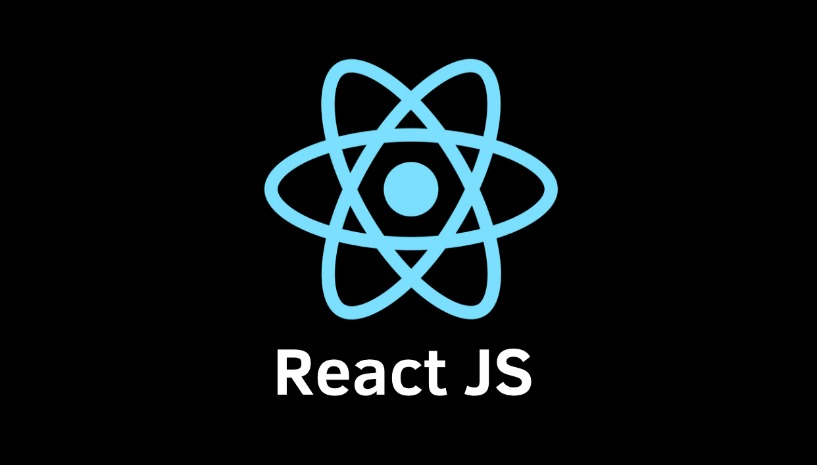
#### Tổng quan về React Framework

ReactJS là thư viện Javascript được xây dựng bởi các kỹ sư của Facebook, đang được rất nhiều công ty nổi tiếng sử dụng để phát triển các sản phẩm của họ nhưYahoo, Airbnb và tất nhiên là trong nội tại Facebook, Instagram. Nó phù hợp với các dự án lớn có tính mở rộng hơn là các dự án nhỏ.

ReactJS luôn giữ các components ở trạng thái stateless khiến ta dễ dàng quản lý bởi nó chẳng khác gì một trang HTML tĩnh. Bản thân các components này không cótrạng thái (state) nó nhận đầu vào từ bên ngoài và chỉ hiện thị ra dựa vào các đầu vào đó, điều này lý giải tại sao nó lại mang tính tái sử dụng và dễ dàng cho việc bảo trì.

#### Thế mạnh của ReactJs

ReactJS là một thư viện hiển thị view chú ý đến hiệu năng (performance-minded). Rất nhiều đối thủ nặng ký về framework MVVM (Model-View-ViewModel) mất một thời gian lớn để hiển thị những lượng dữ liệu lớn, như trong trường hợp hiển thị những danh sách và tương tự. Nhưng React đó không còn là vấn đề, vì nó chỉ hiểnthị những gì thay đổi. Một trong những điểm mạnh nữa của ReactJS là virtual DOM - thứ nằm ẩn bên trong mỗi view và là lí do khiến cho React đạt được hiệu năng tốt. Khi một view yêu cầu gọi, tất cả mọi thứ sẽ được đưa vào trong một bản sao ảo của DOM. Sau khi việcgọi hoàn thành, React tiến hành một phép so sánh giữa DOM ảo và DOM thật, và thựchiện những thay đổi được chỉ ra trong phép so sánh trên.



Hình 1.3. Logo Reactjs Framework

#### Thành phần cơ bản của Reactjs

* Thành phần cơ bản của React được gọi là components. Syntax để viết HTML sử dụng Javascript để render. Có ra một component bằng các gọi phương thức createClass của đối tượng React, điểm bắt đầu khi tiếp cận với thư viện này. Có thể lồng nhiều component vào nhau thông qua lệnh return của phương thức render.
* Trong một chương trình thì có rất nhiều các component, để đơn giản việc quản lý các component đó người ta sử dụng redux, redux giống như 1 cái kho chứa các component và khi dùng component nào thì chỉ cần gọi nó ra.
* Virtual DOM không được tạo ra bởi Reactjs nhưng lại được sử dụng rất nhiều. Đây là một chuẩn của W3C được dùng để truy xuất code HTML hoặc XML. Các Virtual DOM sẽ được tạo ra khi chạy chương trình, đó là nơi chưa các component. Sử dụng DOM sẽ tiết kiệm được hiệu suất làm việc, khi có thay đổi gì Reactjs đều tính toán trước và việc còn lại chỉ là thực hiện chúng lên DOM.

#### Ưu nhược điểm của reactjs

* **Ưu điểm:**
* Reactjs cực kì hiệu quả: Reactjs tạo ra cho chính nó DOM ảo – nơi mà các component thực sự tồn tại trên đó. Điều này sẽ giúp cải thiện hiệu suất rất nhiều. Reactjs cũng tính toán những thay đổi nào cần cập nhật lên DOM và chỉ thực hiện chúng. Điều này giúp Reactjs tránh những thao tác cần trên DOM mà nhiều chi phí.
* Reactjs giúp việc viết các đoạn code JS dễ dàng hơn: Nó dung cú pháp đặc biệt là JSX (Javascript mở rộng) cho phép ta trộn giữa code HTML và Javascript. Ta có thể thêm vào các đoạn HTML vào trong hàm Render mà không cần phải nối chuỗi. Đây là đặc tính thú vị của Reactjs. Nó sẽ chuyển đổi các đoạn HTML thành các hàm khởi tạo đối tượng HTML bằng bộ biến đổi JSX.
* Nó có nhiều công cụ phát triển: Khi bắt đầu Reactjs, đừng quên cài đặt ứng dụng mở rộng của Chrome dành cho Reactjs. Nó giúp debug code dễ dàng hơn. Sau khi bạn cài đặt ứng dụng này, bạn sẽ có cái nhìn trực tiếp vào virtual DOM như thể bạn đang xem cây DOM thông thường.
* Dễ viết các test case giao diện vì virtual DOM được cài đặt hoàn toàn bằng JS.
* Hiệu năng cao đối với các ứng dụng có dữ liệu thay đổi liên tục, dễ dàng cho bảo trì và sửa lỗi.
* **Nhược điểm:**
* Reactjs chỉ phục vụ cho tầng View. React chỉ là View Library nó không phải là một MVC framework như những framework khác. Đây chỉ là thư viện của Facebook giúp render ra phần view. Vì thế React sẽ không có phần Model và Controller, mà phải kết hợp với các thư viện khác. React cũng sẽ không có 2 way binding hay là Ajax.
* Tích hợp Reactjs vào các framework MVC truyền thống yêu cầu cần phải cấu hình lại.
* React khá nặng nếu so với các framework khác React có kých thước tương tương với Angular (Khoảng 35kb so với 39kb của Angular). Trong khi đó Angular là một framework hoàn chỉnh.
* Khó tiếp cận cho người mới học Web.

### Cơ sở dữ liệu MongoDB

#### MongoDB là gì?

MongoDB là một hệ quản trị cơ sở dữ liệu phi quan hệ (NoSQL) mã nguồn mở, được thiết kế để lưu trữ và truy xuất dữ liệu dưới dạng tài liệu(document) có cấu trúc linh hoạt và có thể mở rộng. MongoDB sử dụng ngôn ngữ truy vấn đối tượng (Object Query Language - OQL) để truy xuất dữ liệu và hỗ trợ các tính năng như replica set, sharding, index, aggregation, mapreduce, và nhiều tính năng khác. MongoDB được sử dụng rộng rãi trong các ứng dụng web, mobile, IoT và các ứng dụng có tính khả năng mở rộng cao.

#### Vì sao sử dụng MongoDB?

MongoDB là một hệ quản trị cơ sở dữ liệu phi quan hệ (NoSQL) được sử dụng để lưu trữ dữ liệu theo cấu trúc tài liệu. MongoDB được sử dụng vì nó có nhiều ưu điểm như:

- Khả năng mở rộng: MongoDB có thể mở rộng dễ dàng bằng cách thêm các node vào hệ thống.

- Tốc độ truy vấn nhanh: MongoDB có thể xử lý các truy vấn nhanh hơn so với các hệ quản trị cơ sở dữ liệu quan hệ.

- Khả năng lưu trữ dữ liệu phức tạp: MongoDB có thể lưu trữ các tài liệu phức tạp như đối tượng, mảng và các tài liệu nhúng.

- Dễ sử dụng: MongoDB có cú pháp đơn giản và dễ hiểu, giúp cho việc phát triển ứng dụng nhanh chóng và dễ dàng.

- Hỗ trợ tốt cho các ứng dụng web: MongoDB được sử dụng rộng rãi trong các ứng dụng web như các trang web thương mại điện tử, các ứng dụng xã hội và các ứng dụng di động

#### NoSQL và SQL?

NoSQL và SQL là hai loại hệ quản trị cơ sở dữ liệu khác nhau. Dưới đây là một số điểm khác nhau giữa NoSQL và SQL:

- Cấu trúc dữ liệu: SQL sử dụng cấu trúc dữ liệu bảng, trong khi NoSQL sử dụng cấu trúc dữ liệu tài liệu.

- Khả năng mở rộng: NoSQL có khả năng mở rộng dễ dàng bằng cách thêm các node vào hệ thống, trong khi SQL có thể mở rộng bằng cách thêm các máy chủ.

- Tốc độ truy vấn: NoSQL có thể xử lý các truy vấn nhanh hơn so với SQL trong một số trường hợp.

- Dữ liệu phức tạp: NoSQL có thể lưu trữ các tài liệu phức tạp như đối tượng, mảng và các tài liệu nhúng, trong khi SQL chỉ có thể lưu trữ dữ liệu có cấu trúc.

- Cú pháp: SQL có cú pháp đơn giản và dễ hiểu, trong khi NoSQL có nhiều loại cú pháp khác nhau.

- Hỗ trợ cho các ứng dụng web: NoSQL được sử dụng rộng rãi trong các ứng dụng web như các trang web thương mại điện tử, các ứng dụng xã hội và các ứng dụng di động, trong khi SQL được sử dụng chủ yếu trong các ứng dụng doanh nghiệp.

Tóm lại, SQL và NoSQL có những ưu điểm và hạn chế riêng, tùy thuộc vào nhu cầu và mục đích sử dụng của người dùng để lựa chọn loại hệ quản trị cơ sở dữ liệu phù hợp.

## Công cụ sử dụng

* Visual Studio Code: phục vụ mục đích code Nodejs và ReactJS.
* 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.
* MongoDB: Cơ sở dữ liệu dùng để lưu trữ dữ liệu.
* Git, Github: Quản lý mã nguồn.

# PHÂN TÍCH THIẾT KẾ 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:
  + Tìm kiếm sản phẩm
  + Xem chi tiết sản phẩm
  + Sắp xếp sản phẩm
  + Tìm kiếm bài viết
  + Xem chi tiết bài viết
  + Gửi liên hệ
  + Đăng kí
  + Đăng nhập
  + Tìm lại mật khẩu
* Thành viên của trang web:
  + Tìm kiếm sản phẩm
  + Xem chi tiết sản phẩm
  + Bình luận sản phẩm
  + Sắp xếp sản phẩm
  + Tìm kiếm bài viết
  + Bình luận bài viết
  + Xem chi tiết bài viết
  + Gửi liên hệ
  + Xem hồ sơ
  + Đăng nhập
  + Đăng xuất
  + Tìm lại mật khẩu
  + Đổi mật khẩu
  + Thêm vào giỏ hàng
  + Cập nhật giỏ hàng
  + Đặt hàng
  + Thanh toán
  + Xem đơn hàng
  + Hủy đơn hàng
* Quản trị viên:
  + Đăng nhập
  + Đăng xuất
  + Xem thống kê
  + Thêm, sửa, xóa, tìm kiếm người dùng
  + Thêm, sửa, xóa, tìm kiếm danh mục sản phẩm
  + Thêm, sửa, xóa, tìm kiếm sản phẩm
  + Thêm, sửa, xóa, tìm kiếm bài viết
  + Xóa, tìm kiếm bình luận sản phẩm
  + Xóa, tìm kiếm bình luận bài viết
  + Xem chi tiết, cập nhật trạng thái, tìm kiếm đơn hàng
  + Xem chi tiết, trả lời, tìm kiếm, xóa phản hồi

## Yêu cầu phi chức năng

### Giao diện người dùng

*- Tính trực quan:* Giao diện người dùng của trang web đảm bảo tính trực quan cao để giúp người dùng dễ dàng tìm kiếm sản phẩm, xem thông tin chi tiết về sản phẩm, tìm hiểu về các khuyến mãi và đặt hàng.

*- Tính thân thiện với người dùng:* Giao diện người dùng đảm bảo tính thân thiện với người dùng để giúp họ dễ dàng điều hướng trang web, tìm kiếm sản phẩm và đặt hàng.

*- Độ phản hồi nhanh:* Trang web có giao diện người dùng phản hồi nhanh để giảm thời gian chờ đợi của người dùng.

- *Tính tương thích*: Giao diện người dùng đảm bảo tính tương thích với các trình duyệt web khác nhau và các thiết bị di động để đảm bảo người dùng có thể truy cập trang web từ bất kỳ thiết bị nào mà không gặp bất kỳ vấn đề nào.

*- Tính hài hòa:* Giao diện người dùng đảm bảo tính hài hòa giữa các thành phần trên trang web, bao gồm cả bố cục, font chữ, màu sắc và hình ảnh để tạo ra một trang web đẹp mắt và thu hút người dùng.

### Tính bảo mật và ràng buộc

*- Xác thực người dùng:* Trang web có tính năng xác thực người dùng để đảm bảo rằng chỉ có người dùng đã đăng ký và xác thực mới có thể đăng nhập và truy cập vào các tính năng của trang web.

*- Mã hóa dữ liệu:* Trang web sử dụng mã hóa token để bảo vệ thông tin cá nhân của người dùng khi họ thực hiện các giao dịch trên trang web.

*- Kiểm soát truy cập:* Trang web có các chính sách kiểm soát truy cập để đảm bảo rằng chỉ có những người có quyền truy cập mới có thể truy cập vào các tài nguyên của trang web.

*- Quản lý lỗi và bảo mật:* Trang web quản lý lỗi và bảo mật để đảm bảo rằng các lỗ hổng bảo mật sẽ được khắc phục và các vấn đề về bảo mật sẽ được giải quyết một cách nhanh chóng và hiệu quả.

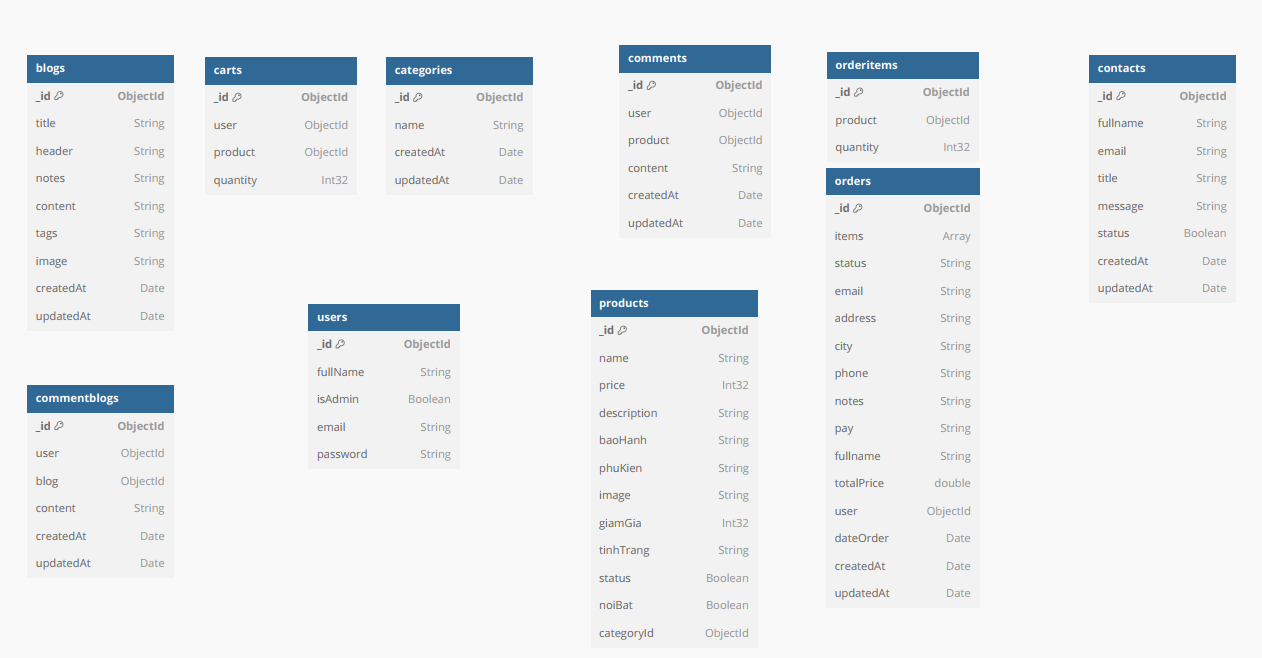
*- Tuân thủ quy định về bảo vệ thông tin cá nhân:* Trang web tuân thủ các quy định về bảo vệ thông tin cá nhân để đảm bảo rằng thông tin cá nhân của người dùng sẽ được bảo vệ và không bị lộ ra ngoài.

## Biểu đồ use case tổng quát



Hình 2.1: Biểu đồ use case tổng quát

## Mô hình thực thể



Hình 2.2: Biểu đồ thực thế

## Đặc tả usecase

### 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 phải màn hình.
2. Hệ thống yêu cầu người dùng nhập các thông tin bao gồm: tên đăng nhập(email) và mật khẩu.
3. Người dùng nhập các thông tin: tên đăng nhập(email) 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 2.3: Biều đồ trình tự use case Đăng nhập

### Use case đăng ký

1. Đặc tả use case:

Use case cho phép khách vãng lai 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 Email, tên, địa chỉ, số điện thoại, mật khẩu 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 2.4: Biều đồ trình tự use case Đăng ký

### Use case tìm kiếm sản phẩm

1. Đặc tả use case

Use case cho phép người dùng tìm kiếm sản phẩm trên bảng products.

* Luồng sự kiện:
* Luồng cơ bản:

1. Ca này bắt đầu khi người dùng nhấn vào ô “Tìm Kiếm” trên thanh menu chính.
2. Người dùng nhập từ khóa muốn tìm kiếm vào ô tìm kiếm và nhập từ khóa cần tìm kiếm. Hệ thống sẽ nhận thông tin tìm kiếm trùng với từ khóa tìm kiếm sẽ được hiển thị lên màn hình.

Use case kết thúc.

* + Luồng rẽ nhánh:

1. Tại bước 2 của luồng cơ bản nếu người dùng nhập thông tin không liên quan đến các sản phẩm trên bảng products trong CSDL thì sẽ không có sản phẩm nào được hiện lên màn hình.
2. 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:

Nếu sản phẩm tìm kiếm có trong CSDL thì các sản phẩm đó sẽ hiển thị trên màn hình.

* Điểm mở rộng:

Không có.

1. Biểu đồ trình tựDiagram

   Description automatically generated

Hình 2.5: Biều đồ trình tự use case Tìm kiếm sản phẩm

### Use case Xem chi tiết sản phẩm

1. Đặc tả use case

Use case cho phép người dùng, khách vãng lai xem thông tin chi tiết sản phẩm

* Luồng sự kiện:
  + Luồng cơ bản:

1. Ca này bắt đầu khi người dùng nhấn vào “Xem chi tiết” của bất kì sản phẩm nào. Hệ thống sẽ lấy thông tin chi tiết sản phẩm (tên sản phẩm, giá bán, bảo hành, phụ kiện, hình ảnh ,giảm giá, tình trạng) hiển thị ra màn hình

Use case kết thúc.

* + Luồng rẽ nhánh:

1. 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 2.6: Biều đồ trình tự use case Xem chi tiết sản phẩm

### Use case thêm quản lý giỏ hàng

1. Đặc tả use case

Use case này cho phép khách hàng xem, thêm, sửa, xóa các sản phẩm trong giỏ hàng.

* Luồng sự kiện:
  + Luồng cơ bản:
    1. Use case này bắt đầu khi khách hàng bấm vào biểu tượng giỏ hàng  
       trên thanh menu. Hệ thống sẽ lấy thông tin các sản phẩm có trong  
       giỏ hàng và hiển thị lên màn hình
    2. Thêm sản phẩm vào giỏ hàng: khách hàng kích vào nút “Thêm vào  
       giỏ hàng” trên màn hình chi tiết sản phẩm, hệ thống sẽ thêm sản  
       phẩm đó vào giỏ hàng với số lượng mặc định là 1.
    3. Cập nhật sản phẩm trong giỏ

a. Khách hàng nhấn nút tăng số lượng, hệ thống tự động tăng số thêm 1 đơn vị

b. Khách hàng nhấn nút giảm số lượng, hệ thống tự động giảm số lượng xuống 1 đơn vị

Khách hàng bấm vào nút “Cập nhật” để cập nhật số lượng sản phẩm trong giỏ. Hệ thống lưu thông tin vào bảng orders

* + 1. Xóa sản phẩm trong giỏ: Khách hàng bấm nút xóa trên một sản phẩm, hệ thống sẽ thực hiện loại bỏ sản phẩm đó ra khỏi giỏ hàng và cập nhật thông tin trong bảng orders.  
       Use case kết thúc
  + Luồng rẽ nhánh:
    1. 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

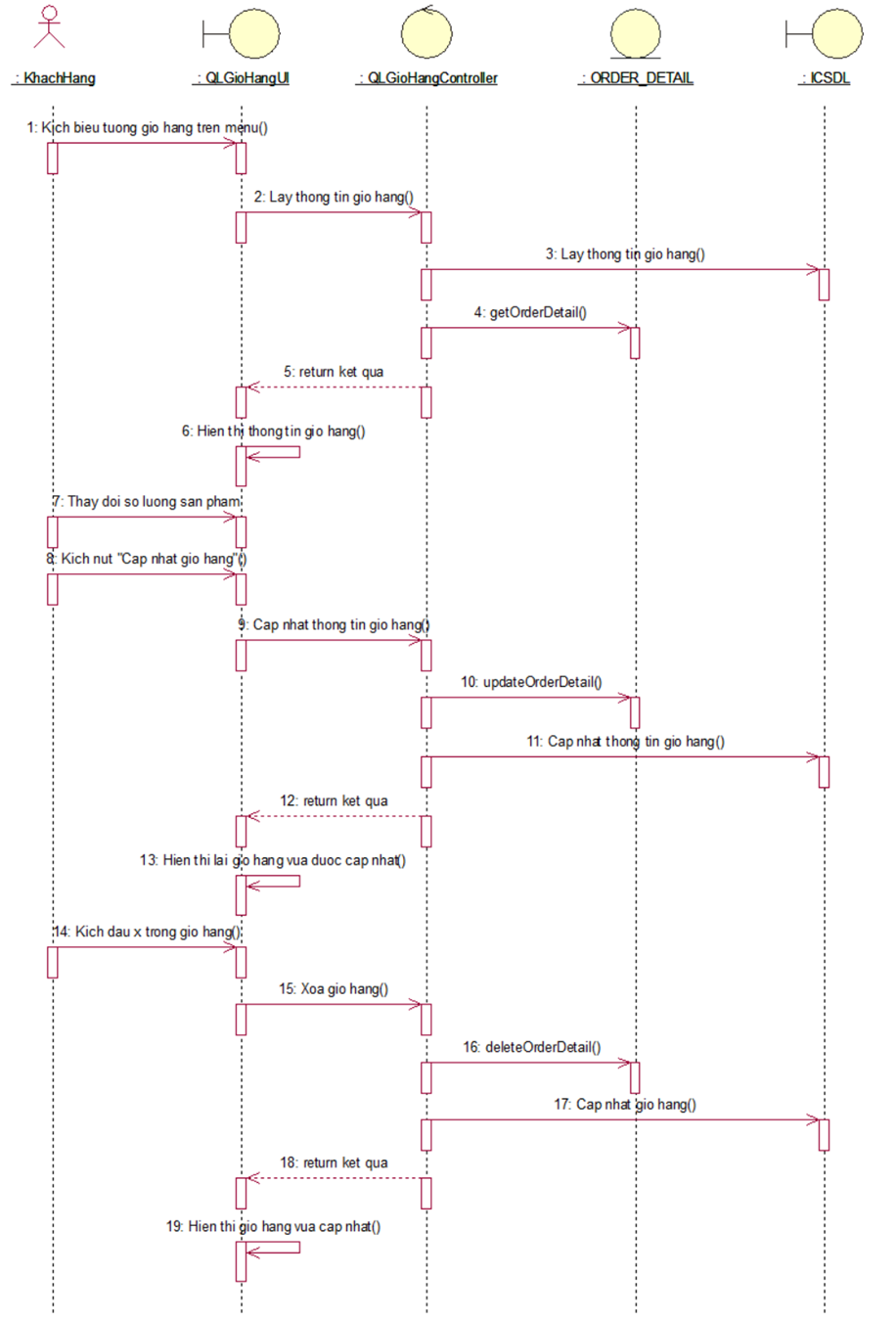
* Hậu điều kiện:

Không có

* Điểm mở rộng:

Không có.

1. Biểu đồ trình tự



Hình 2.7: Biều đồ trình tự use case Quản lý giỏ hàng

### Use case xem giỏ hàng

1. Đặc tả use case

Use case cho phép người dùng xem giỏ hàng

* Luồng sự kiện:
  + Luồng cơ bản:
    1. Use case bắt đầu khi người dùng nhấn vào biểu tượng giỏ hàng trên màn hình. Hệ thống sẽ lấy thông tin chi tiết giỏ hàng (tên sản phẩm, giá bán, bảo hành, phụ kiện, hình ảnh ,giảm giá, tình trạng, thành tiền, số lượng) hiển thị ra màn hình.

Use case kết thúc.

* + Luồng rẽ nhánh:
    1. 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.

### Use case Đặt hàng

1. Đặc tả use case

Use case cho phép người dùng đặt hàng

* Luồng sự kiện:
  + Luồng cơ bản:
    1. Use case bắt đầu khi người dùng nhấn vào “Đặt hàng” tại màn hình giỏ hàng. Hệ thống sẽ chuyển tới màn hình yêu cầu người dùng nhập thông tin để thanh toán
    2. Người dùng nhập thông tin cần thiết và bấm nút “Đặt hàng”. Hệ thống sẽ hiển thị thông báo thành công

Use case kết thúc.

* + Luồng rẽ nhánh:
    1. Tại bước 2 luồng cơ bản nếu người dùng nhập không đủ thông tin hệ thống sẽ yêu cầu người dùng nhập đầy đủ mới có thể đặt hàng
    2. 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

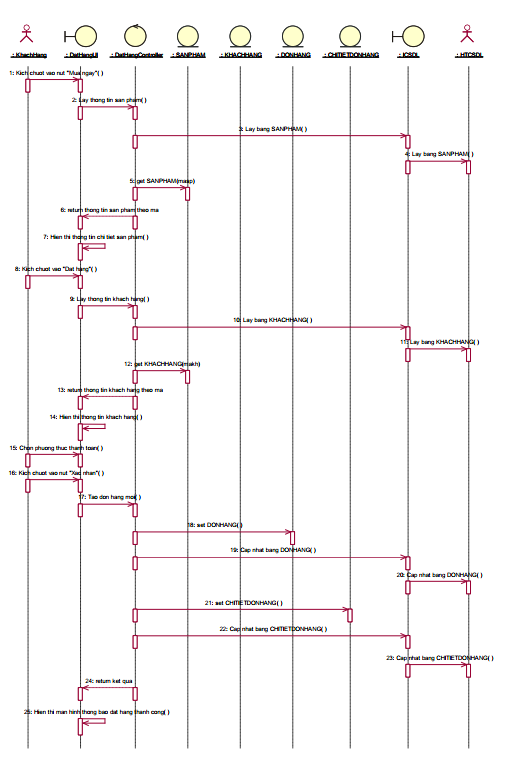
* Hậu điều kiện:

Không có

* Điểm mở rộng:

Không có.

1. Biểu đồ trình tự



Hình 2.8: Biều đồ trình tự use case Đặt hàng

### Use case Quản lý sản phẩm

1. Đặc tả use case

Use case cho phép quản trị viên quản lý sản phẩm

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case bắt đầu khi quản trị viên nhấn vào “Quản lý sản phẩm” menu quản trị. Hệ thống sẽ lấy thông tin từ bảng products và hiển thị thông tin gồm mã sản phẩm, tên sản phẩm, mô tả, giá, trạng thái,… ra màn hình
2. Thêm sản phẩm:
   1. Quản trị viên kích vào nút “Thêm mới sản phẩm” trên cửa sổ danh sách sản phẩm. Hệ thống hiển thị màn hình yêu cầu nhập thông tin chi tiết cho sản phẩm gồm: Mã danh mục, Mã sản phẩm, tên sản phẩm, hình ảnh, mô tả, số lượng, đơn giá,….
   2. Quản trị viên nhập thông tin của Mã danh mục, mã sản phẩm, tên sản phẩm, hình ảnh, mô tả, số lượng, đơn giá và kích vào nút “Save”. Hệ thống sẽ tạo một Sản phẩm mới trong bảng products và hiển thị danh sách các Sản phẩm đã được cập nhật.
3. Sửa sản phẩm:
   1. Quản trị viên kích nút “Sửa” trên một dòng sản phẩm. Hệ thống hiển thị màn hình thông tin cũ của sản phẩm gồm: Mã danh mục, Mã sản phẩm, tên sản phẩm, hình ảnh, mô tả, số lượng, đơn giá.
   2. Quản trị viên nhập thông tin cần sửa và kích “Save”. Hệ thống sẽ sửa thông tin của sản phẩm được chọn trong bảng products và hiển thị danh sách sản phẩm đã cập nhật.
4. Xóa sản phẩm:
   1. Người quản trị kích vào nút “Xóa” trên một dòng Sản phẩm. Hệ thống sẽ hiển thị một màn hình yêu cầu xác nhận xóa.
   2. Người quản trị kích vào nút “Đồng ý”. Hệ thống sẽ xóa Sản phẩm được chọn khỏi bảng products và hiển thị danh sách các Sản phẩm đã cập nhật.

Use case kết thúc.

* + Luồng rẽ nhánh:

1. Tại bước 2b hoặc 3b trong luồng cơ bản nếu người quản trị nhập thông tin Sản phẩm không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi yêu cầu nhập lại. Người quản trị có thể nhập lại để tiếp tục hoặc kích vào nút “Hủy bỏ” để kết thúc.
2. Tại bước 2b hoặc 3b trong luồng cơ bản nếu người quản trị kích vào nút “Quay lại” hệ thống sẽ bỏ qua thao tác thêm mới hoặc sửa chữa tương ứng và hiển thị danh sách các Sản phẩm trong bảng products.
3. Tại bước 4b trong luồng cơ bản nếu người quản trị kích vào nút “Không đồng ý” hệ thống sẽ bỏ qua thao tác xóa và hiển thị danh sách các Sản phẩm trong bảng products.
4. 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:

Người dùng và quản trị viên phải đăng nhập trước đó

* Hậu điều kiện:

Không có

* Điểm mở rộng:

Không có

1. Biều đồ trình tự:



Hình 2.9: Biều đồ trình tự use case Quản lý sản phẩm

### Use case Quản lý đơn hàng

1. Đặc tả use case

Use case cho quản trị viên quản lý đơn hàng của hệ thống

* Luồng sự kiện:
  + Luồng cơ bản:
    1. Use case bắt đầu khi quản trị viên chọn “Quản lý đơn hàng” trên menu quản trị. Hệ thống lấy dữ liệu và hiển thị danh sách đơn hàng gồm: Mã đơn hàng, khách hàng, tổng tiền, tình trạng, ngày bán.
    2. Quản trị viên kích nút “Xem” ứng với đơn hàng. Hệ thống sẽ lấy thông tin từ bảng tbl\_order\_products và hiển thị (Tên khách hàng, số điện thoại, địa chỉ, email, thông tin sản phẩm) ra màn hình.
    3. Quản trị viên chọn tình trạng đơn hàng và chọn “Lưu”. Hệ thống sẽ cập nhật tình trạng đơn hàng

Use case kết thúc.

* + Luồng rẽ nhánh:
    1. Tại bước 3 trong luồng cơ bản nếu quản trị viên chọn “Xuất đơn”, hệ thống sẽ gửi email cho người nhận và thông báo tình trạng hàng.
    2. 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:

Người dùng và quản trị viên phải đăng nhập trước đó

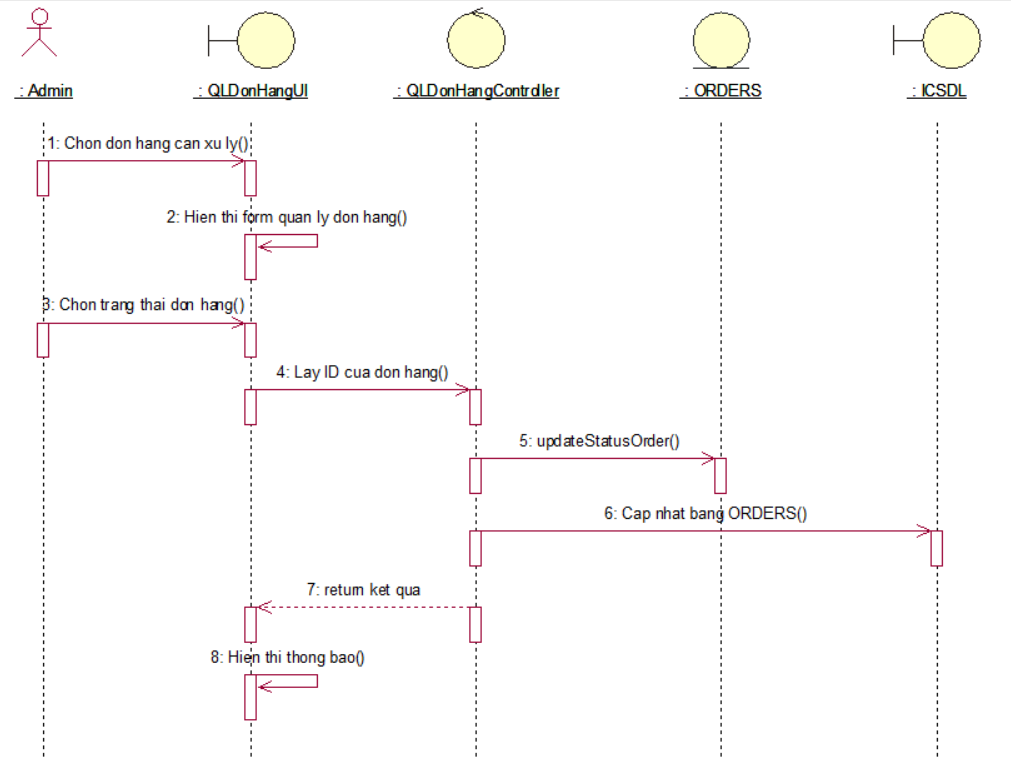
* Hậu điều kiện:

Không có

* Điểm mở rộng:

Không có

1. Biểu đồ trình tự



Hình 2.10: Biều đồ trình tự use case Đơn hàng

### Use case quản lý tài khoản

1. Đặc tả use case

Use case cho phép quản trị viên quản lý tài khoản

* Luồng sự kiện:
  + Luồng cơ bản:
    1. Use case bắt đầu khi quản trị viên nhấn vào “Quản lý tài khoản” menu quản trị. Hệ thống sẽ lấy thông tin từ bảng users và hiển thị thông tin gồm mã tài khoản, tên người dùng, email, địa chỉ, số điện thoại,… ra màn hình
    2. Sửa tài khoản:
  1. Quản trị viên kích nút “Sửa” trên một dòng tài khoản. Hệ thống hiển thị màn hình thông tin cũ của tài khoản
  2. Quản trị viên chọn quyền cần sửa và kích “Save”. Hệ thống sẽ sửa thông tin của sản phẩm được chọn trong bảng users và hiển thị danh sách tài khoản đã cập nhật.
     1. Xóa tài khoản:
  3. Người quản trị kích vào nút “Xóa” trên một dòng tài khoản. Hệ thống sẽ hiển thị một màn hình yêu cầu xác nhận xóa.
  4. Người quản trị kích vào nút “Đồng ý”. Hệ thống sẽ xóa tài khoản được chọn khỏi bảng userss và hiển thị danh sách các tài khoản đã cập nhật.

Use case kết thúc.

* + Luồng rẽ nhánh:
    - 1. Tại bước 2b hoặc 3b trong luồng cơ bản nếu người quản trị nhập thông tin Sản phẩm không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi yêu cầu nhập lại. Người quản trị có thể nhập lại để tiếp tục hoặc kích vào nút “Hủy bỏ” để kết thúc.
      2. Tại bước 2b hoặc 3b trong luồng cơ bản nếu người quản trị kích vào nút “Quay lại” hệ thống sẽ bỏ qua thao tác thêm mới hoặc sửa chữa tương ứng và hiển thị danh sách các user trong bảng users.
      3. Tại bước 4b trong luồng cơ bản nếu người quản trị kích vào nút “Không đồng ý” hệ thống sẽ bỏ qua thao tác xóa và hiển thị danh sách các tài khoản trong bảng users.
      4. 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:

Người dùng và quản trị viên phải đăng nhập trước đó

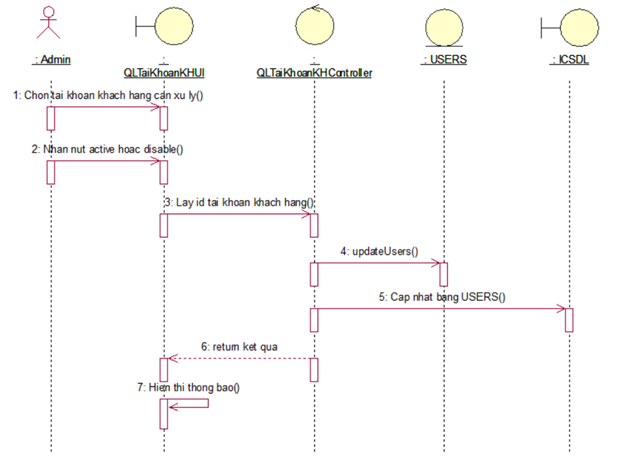
* Hậu điều kiện:

Không có

* Điểm mở rộng:

Không có

1. Biểu đồ trình tự



Hình 2.11: Biều đồ trình tự use case Tài khoản

### Use case Quản lý danh mục

1. Đặc tả use case

Use case cho phép quản trị viên quản lý các danh mục trong hệ thống

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case bắt đầu khi quản trị viên nhấn vào “Quản lý danh mục” menu quản trị. Hệ thống sẽ lấy thông tin từ bảng categories và hiển thị thông tin gồm mã danh mục, tên danh mục, mô tả, trạng thái,… ra màn hình
2. Thêm danh mục:
   1. Quản trị viên kích vào nút “Thêm mới danh mục” trên cửa sổ danh sách danh mục. Hệ thống hiển thị màn hình yêu cầu nhập thông tin chi tiết cho quyền gồm: Tên danh mục, mô tả, trạng thái.
   2. Quản trị viên nhập thông tin của Tên quyền, mô tả, trạng thái và kích vào nút “Save”. Hệ thống sẽ tạo một danh mục mới trong bảng categories và hiển thị danh sách các danh mục đã được cập nhật.
3. Sửa danh mục:
   1. Quản trị viên kích nút “Sửa” trên một dòng danh mục. Hệ thống hiển thị màn hình thông tin cũ của quyền gồm: Tên danh mục, mô tả, trạng thái
   2. Quản trị viên nhập thông tin cần sửa và kích “Save”. Hệ thống sẽ sửa thông tin của danh mục được chọn trong bảng categories và hiển thị danh sách danh mục đã cập nhật.
4. Xóa danh mục:
   1. Người quản trị kích vào nút “Xóa” trên một dòng danh mục. Hệ thống sẽ hiển thị một màn hình yêu cầu xác nhận xóa.
   2. Người quản trị kích vào nút “Đồng ý”. Hệ thống sẽ xóa danh mục được chọn khỏi bảng categories và hiển thị danh sách các danh mục đã cập nhật.

Use case kết thúc.

* + Luồng rẽ nhánh:

1. Tại bước 2b hoặc 3b trong luồng cơ bản nếu người quản trị nhập thông tin không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi yêu cầu nhập lại. Người quản trị có thể nhập lại để tiếp tục hoặc kích vào nút “Hủy bỏ” để kết thúc.
2. Tại bước 2b hoặc 3b trong luồng cơ bản nếu người quản trị kích vào nút “Quay lại” hệ thống sẽ bỏ qua thao tác thêm mới hoặc sửa chữa tương ứng và hiển thị danh sách các danh mục trong bảng categories.
3. Tại bước 4b trong luồng cơ bản nếu người quản trị kích vào nút “Không đồng ý” hệ thống sẽ bỏ qua thao tác xóa và hiển thị danh sách các danh mục trong bảng categories.
4. 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:

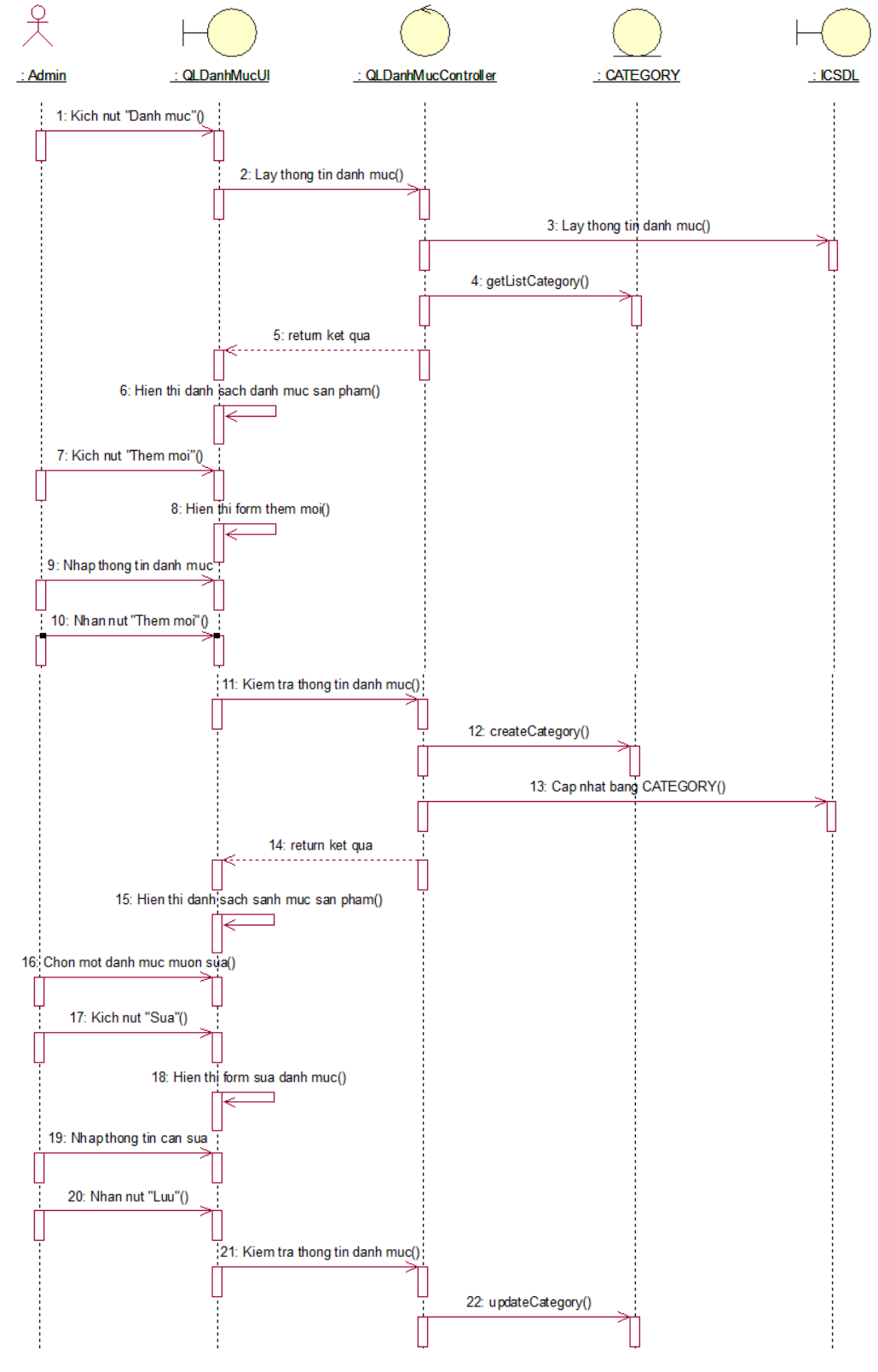
Người dùng và quản trị viên phải đăng nhập trước đó

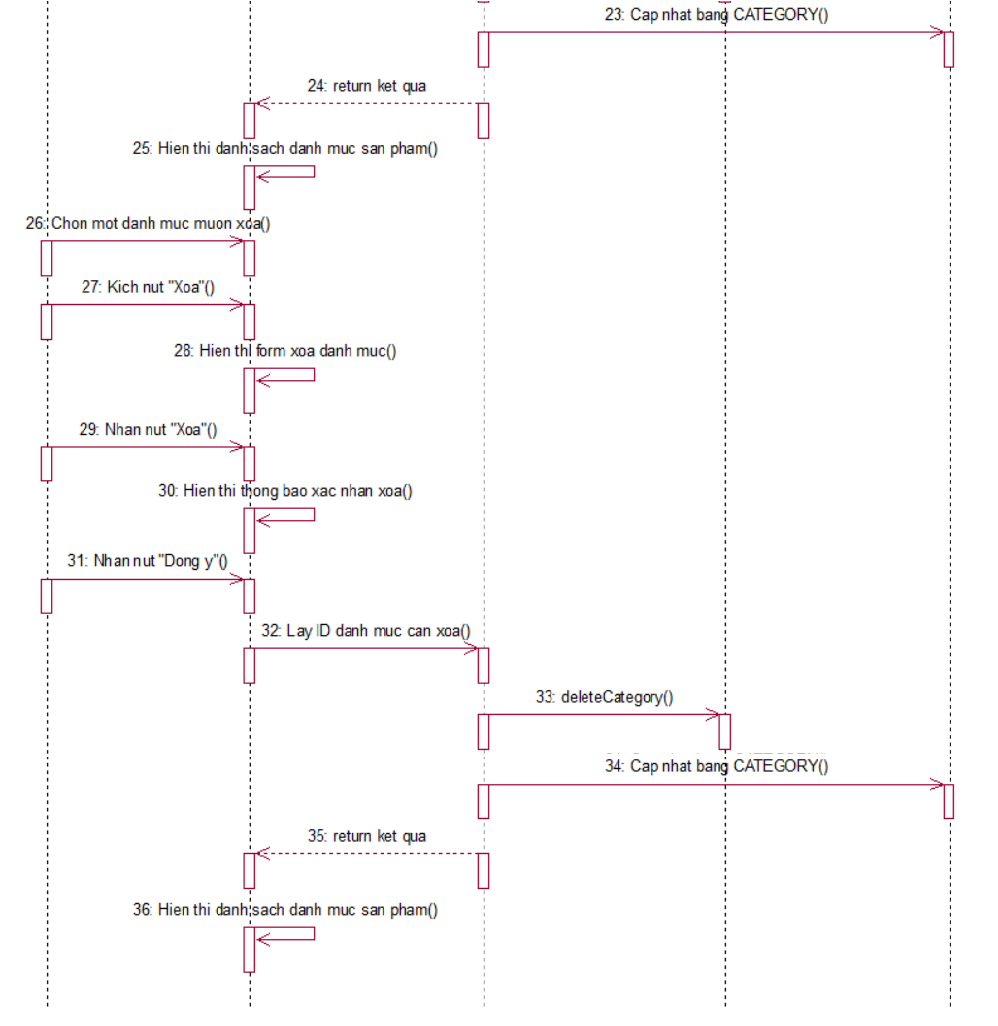
* Hậu điều kiện:

Không có

* Điểm mở rộng:

Không có

1. Biểu đồ trình tự



Hình 2.12: Biều đồ trình tự use case Danh mục

### Use case Thanh toán

1. Đặc tả use case

Use case cho phép Khách hàng sử dụng chức năng này để thanh toán sản phẩm.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case bắt đầu khi người dùng kích “Thanh toán” trên thanh menu. Hệ thống sẽ hiển thị form thanh toán và thông tin giỏ hàng từ bảng CART lên màn hình.

2. Khách hàng nhập thông tin và kích nút “Thanh toán”. Hệ thống sẽ tạo mới một đơn hàng trong bảng ORDER sẽ hiển thị một thông tin và use case kết thúc.

* + Luồng rẽ nhánh:

1. Tại bước 2 nếu người dùng nhập sai thông tin thì hệ thống sẽ hiển thị thông báo. Người dùng có thể nhập lại hoặc hủy bỏ. Use case kết thúc..
2. 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:

Người dùng phải đăng nhập trước đó

* Hậu điều kiện:

Nếu use case thành công thì thông tin về đơn hàng sẽ được cập nhật trong cơ sở dữ liệu

* Điểm mở rộng:

Không có

1. Biểu đồ trình tự Diagram

   Description automatically generated

Hình 2.13: Biều đồ trình tự use case Thanh toán

### Use case Thống kê

1. Đặc tả use case

Use case cho phép Admin sử dụng chức năng này để xem các thông kê của cửa hàng.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case bắt đầu khi người dùng kích “Dashboard” trên thanh menu. Hệ thống sẽ lấy thông tin và hiển thị thông tin thông kê gồm tổng đơn hàng , doanh thu và tổng khách hàng từ bảng ORDERS, USERS lên màn hình. Use case kết thúc
   * Luồng rẽ nhánh:
2. 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:

Quản trị viên phải đăng nhập trước đó

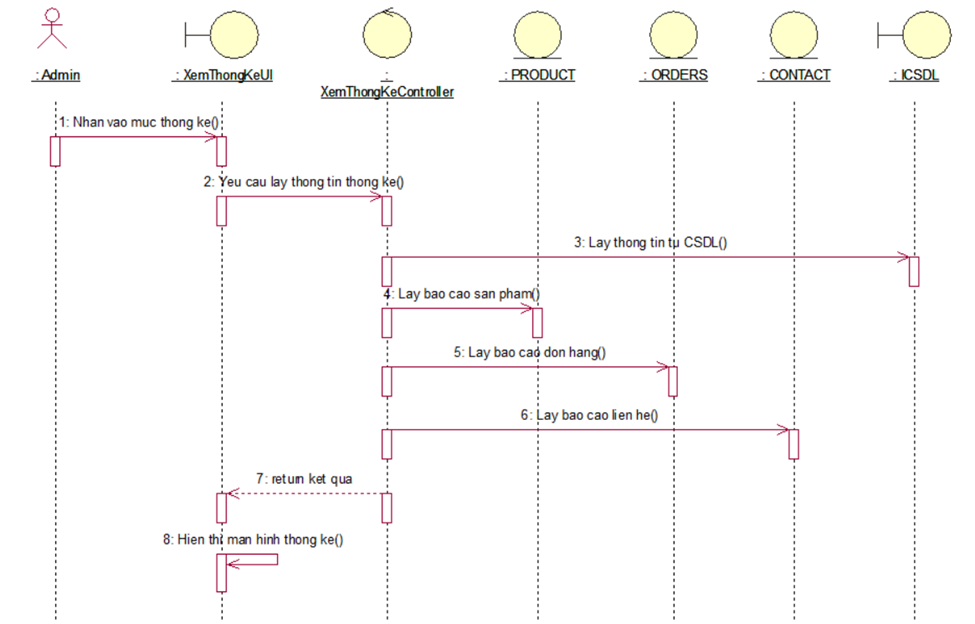
* Hậu điều kiện:

Không có

* Điểm mở rộng:

Không có

1. Biểu đồ trình tự



Hình 2.14: Biều đồ trình tự use case Thống kê

### 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 email tài khoản rồi kích nút gửi. Hệ thống kiểm tra email tài khoản trong cơ sở dữ liệu và gửi 1 đường dẫn cập nhật mật khẩu về email.
    3. Khách hàng nhập mật khẩu mới và kích nút cập nhật. Hệ thống sẽ lấy userId thông qua token và kiển tra trong cơ sở dữ liệu. Và cập nhật mật khẩu vào cơ sở dữ liệu.
    4. Use case kết thúc.
  + Luồng rẽ nhánh:
    - 1. Tại bước 2 ở luồng cơ bản, nếu email tài khoản của khách hàng không tồn tại trong cơ sở dữ liệu, hệ thống báo lỗi. Use case kết thúc.
      2. Tại bước 3 ở luông cơ bản, nếu userId của khách hàng không tồn tại trong cơ sở dữ liệu , hệ thống báo lỗi.Use case kết thúc.
      3. Tại bước 3 ở luồng cơ bản, nếu token hết hạn, hệ thống báo lỗi. Use case kết thúc
      4. 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

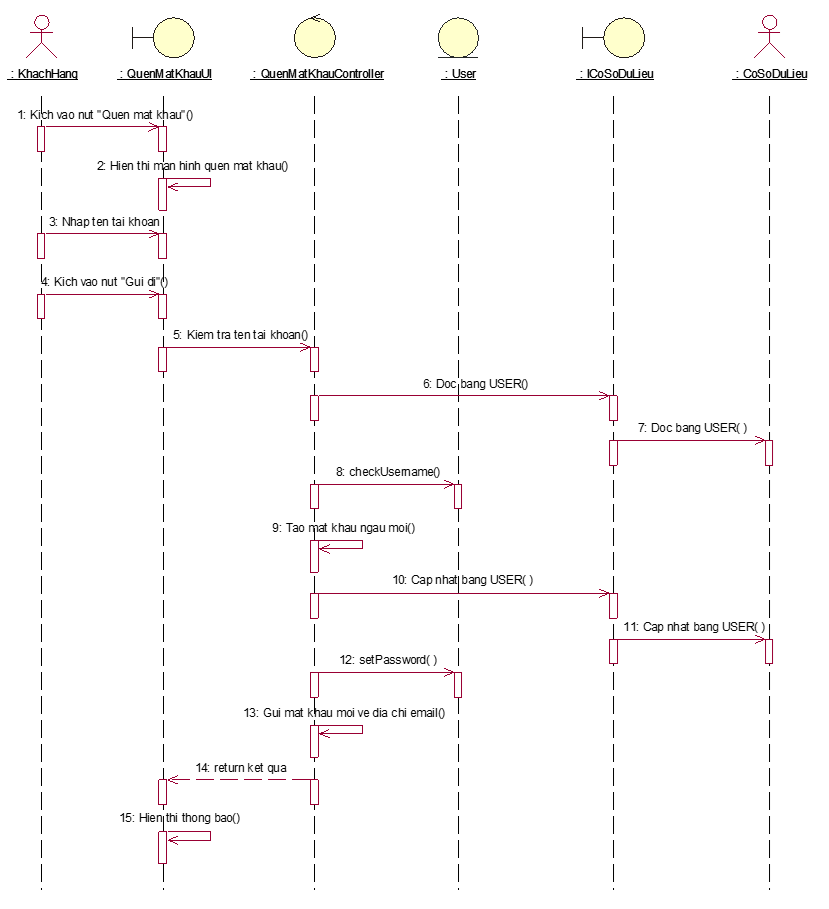
* Hậu điều kiện:

Không có

* Điểm mở rộng:

Không có

1. Biểu đồ trình tự



Hình 2.15: Biểu đồ trình tự use case quên mật khẩu

### Use case Liên hệ

1. Đặc tả use case

Use case này cho phép khách hàng có thể gửi ý kiến thắc mắc cho admin.

* Luồng sự kiện:
  + Luồng cơ bản:
    1. Use case này bắt đầu khi người dùng kích nút “Liên hệ” trên thanh menu. Hệ thống hiển thị màn hình nhập thông tin (tên khách hàng, email, số điện thoại, nội dung) để khách hàng nhập.
    2. Khách hàng nhập các thông tin yêu cầu rồi kích nút “Gửi thư”. Hệ thống sẽ tiến hành thêm bản ghi mới vào bảng CONTACT và chuyển màn đến màn hình gửi thư thành công

Use case kết thúc.

* + Luồng rẽ nhánh:
    1. 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.
    2. Tại bước 2 nếu người dùng nhập thiếu thông tin hoặc nhập thông tin không hợp lệ. Hệ thống sẽ đưa ra thông báo lỗi và yêu cầu nhập lại. Khách hàng có thể nhập lại để tiếp tục thao tác
* 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 tựDiagram

   Description automatically generated

Hình 2.16: Biều đồ trình tự use case Liên hệ

### Use case Quản lý bài viết

1. Đặc tả use case

Use case này cho phép admin thêm, cập nhật, xóa bài viết trong hệ thống.

* Luồng sự kiện:
  + Luồng cơ bản:
    1. Use case này bắt đầu khi người dùng (nhân viên, admin) nhấn vào mục bài viết trên thanh menu, hệ thống hiển thị danh sách bài viết
    2. Thêm blog: Khi người dùng ấn vào nút thêm, hệ thống hiển thị ra form (danh mục blog, tên blog, ảnh đại diện, nội dung…). Người dùng nhập thông tin danh mục rồi ấn “Lưu”. Hệ thống sẽ kiểm tra tính hợp lệ dữ liệu rồi thêm vào bảng BLOG trong database. Thêm thành công hệ thống sẽ hiển thị thông báo và hiển thị danh sách blog. Use case kết thúc
    3. Cập nhật blog: Khi người dùng ấn vào nút sửa, hệ thống hiển thị ra form (danh mục blog, tên blog, ảnh đại diện, nội dung…) và hiển thị thông tin của blog cần sửa. Người dùng nhập thông tin blog rồi ấn “Lưu”. Hệ thống sẽ kiểm tra tính hợp lệ dữ liệu rồi cập nhật trong bảng BLOG trong database. Cập nhật thành công hệ thống sẽ hiển thị thông báo và hiển thị danh sách các blog. Use case kết thúc
    4. Xóa blog: Khi người dùng chọn xóa trong cột chức năng tương ứng với blog trong danh sách blog thì hệ thống sẽ hiển thị form xác nhận xóa. Người dùng ấn xác nhận hệ thống sẽ thực hiện xóa blog trong bảng BLOG. Use case kết thúc
  + Luồng rẽ nhánh:
    1. Tại luồng cơ bản 2, 3: Nếu người dùng nhập sai thông tin của blog hoặc để trống không nhập thì hệ thống sẽ hiển thị thông báo yêu cầu người dùng nhập lại.
    2. Tại luồng cơ bản 4: Sau khi hiển thị thông báo có muốn xóa hay không nếu người dùng chọn “Không đồng ý” thông báo biến mất, use case kết thúc.
    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:

Admin phải đăng nhập trước đó

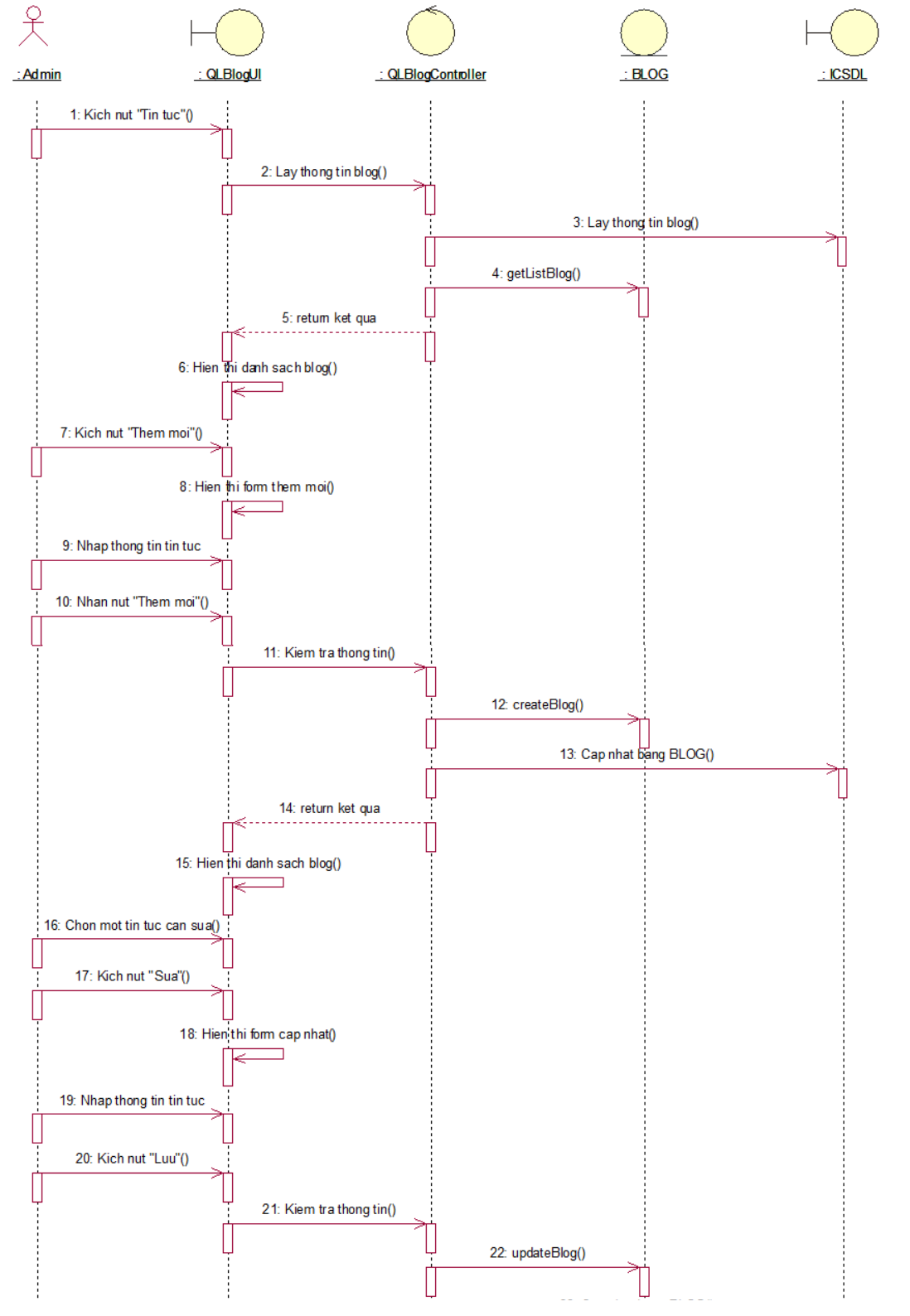
* Hậu điều kiện:

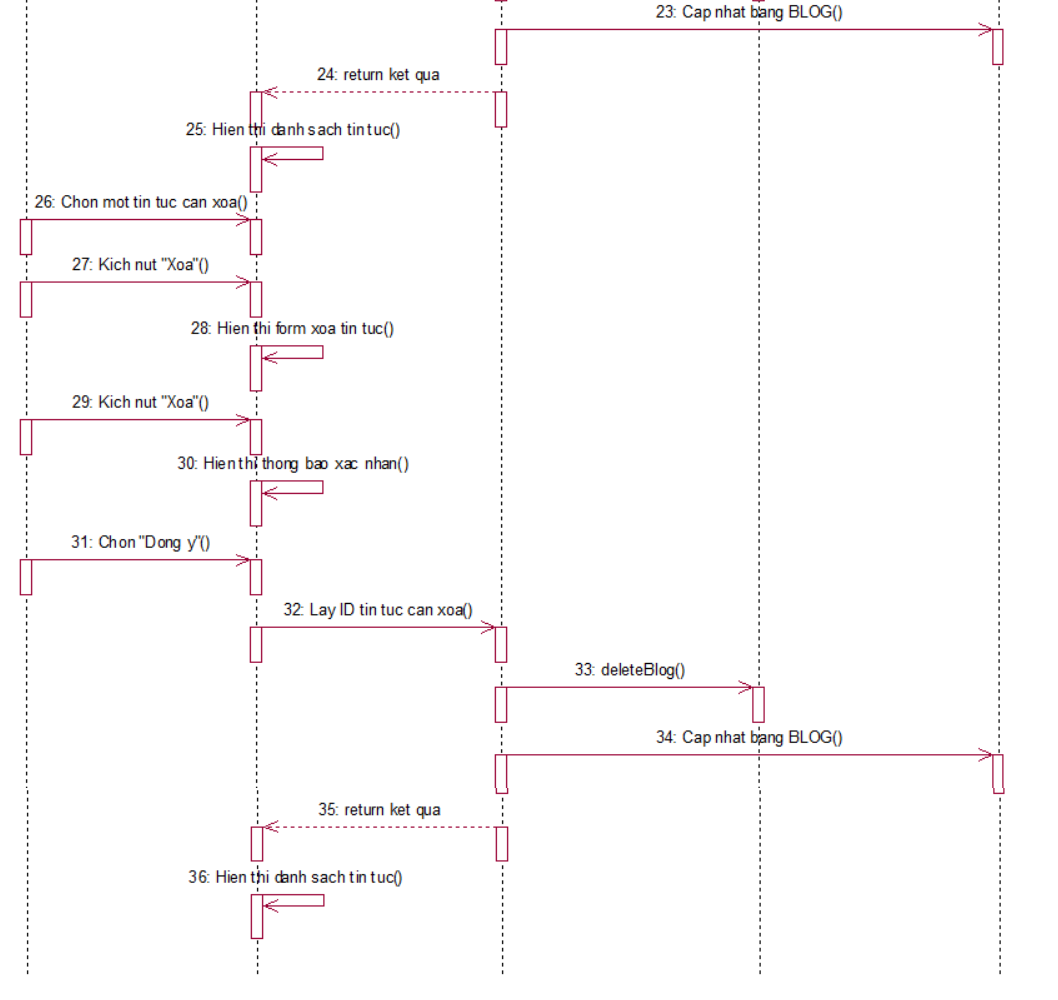
Không có

* Điểm mở rộng:

Có thể chuyển sang use case Dang nhap

1. Biểu đồ trình tự





Hình 2.17: Biều đồ trình tự use case Quản lý bài viết

### Use case xem bài viết

1. Đặc tả use case

Use case này cho phép người sử dụng có thể xem được các bài viết trong trang web.

* Luồng sự kiện:
  + Luồng cơ bản:
    1. Use case bắt đầu khi người sử dụng kích vào link tin tức bất kì trên website. Hệ thông lấy thông tin từ bảng BLOG (Mã tin tức, Tên tin tức, Tiêu đề, Nội dung, Hình ảnh, Ngày đăng bài) trong cơ sở dữ liệu và hiển thị lên màn hình. Use case kết thúc
  + Luồng rẽ nhánh:
    1. 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

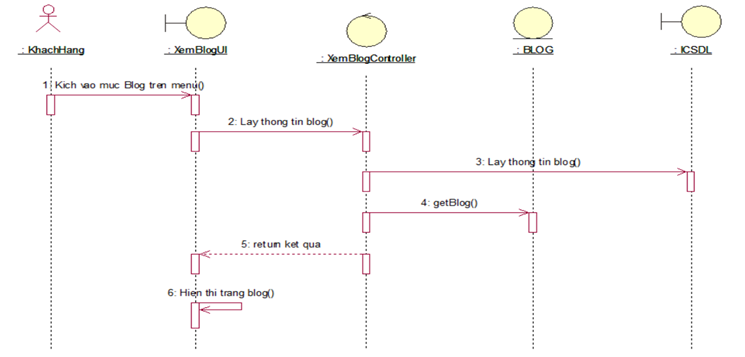
* Hậu điều kiện:

Không có

* Điểm mở rộng:

Không có

1. Biểu đồ trình tự



Hình 2.18: Biều đồ trình tự use case Quản lý bài viết

# CÀI ĐẶT VÀ KẾT QUẢ ĐẠT ĐƯỢC



## Công cụ sử dụng

* Visual Studio Code: phục vụ mục đích code Nodejs.
* MongoDB: 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.

## Kết quả đạt được

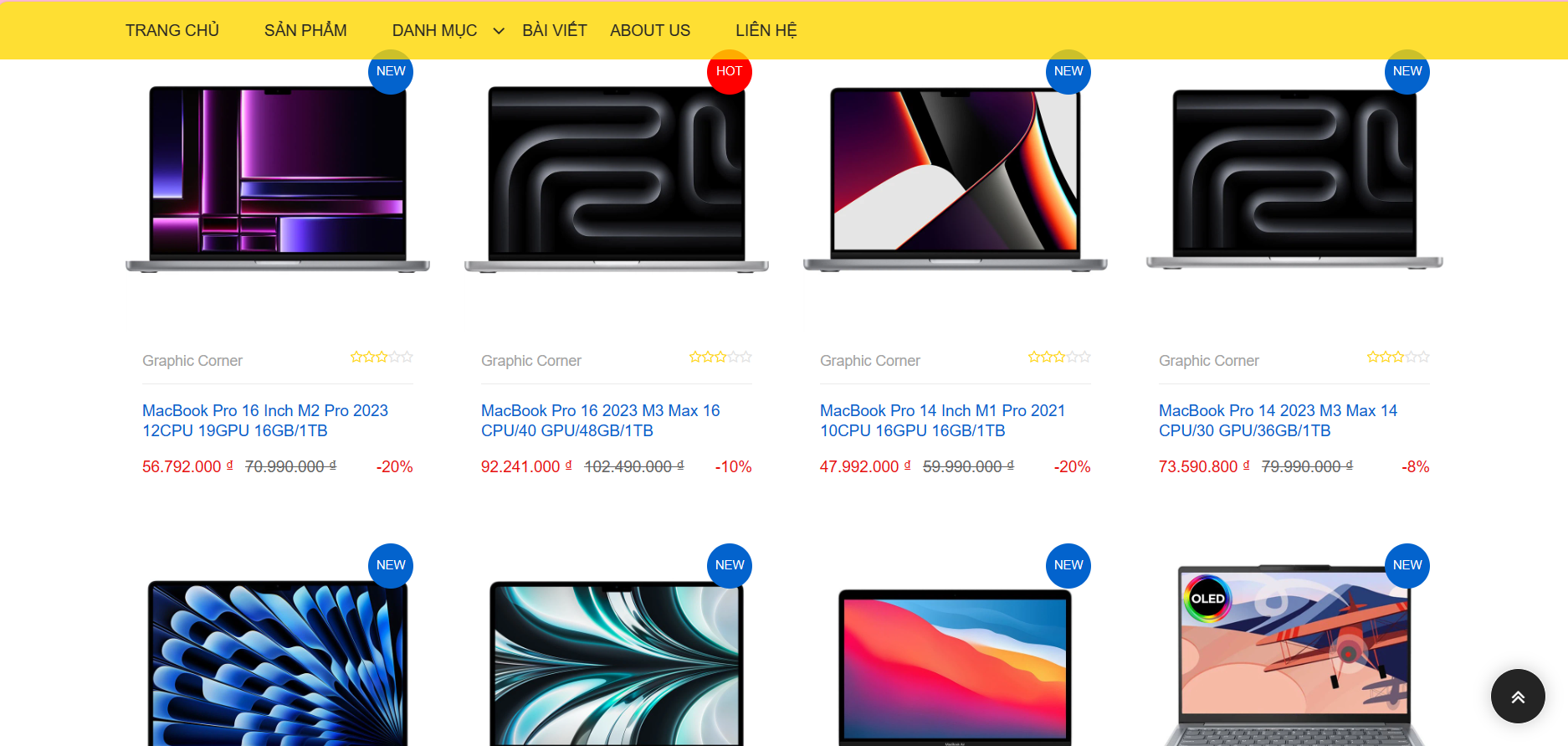
### Giao diện chức năng chính của khách hàng

* Giao diện trang chủ của website:



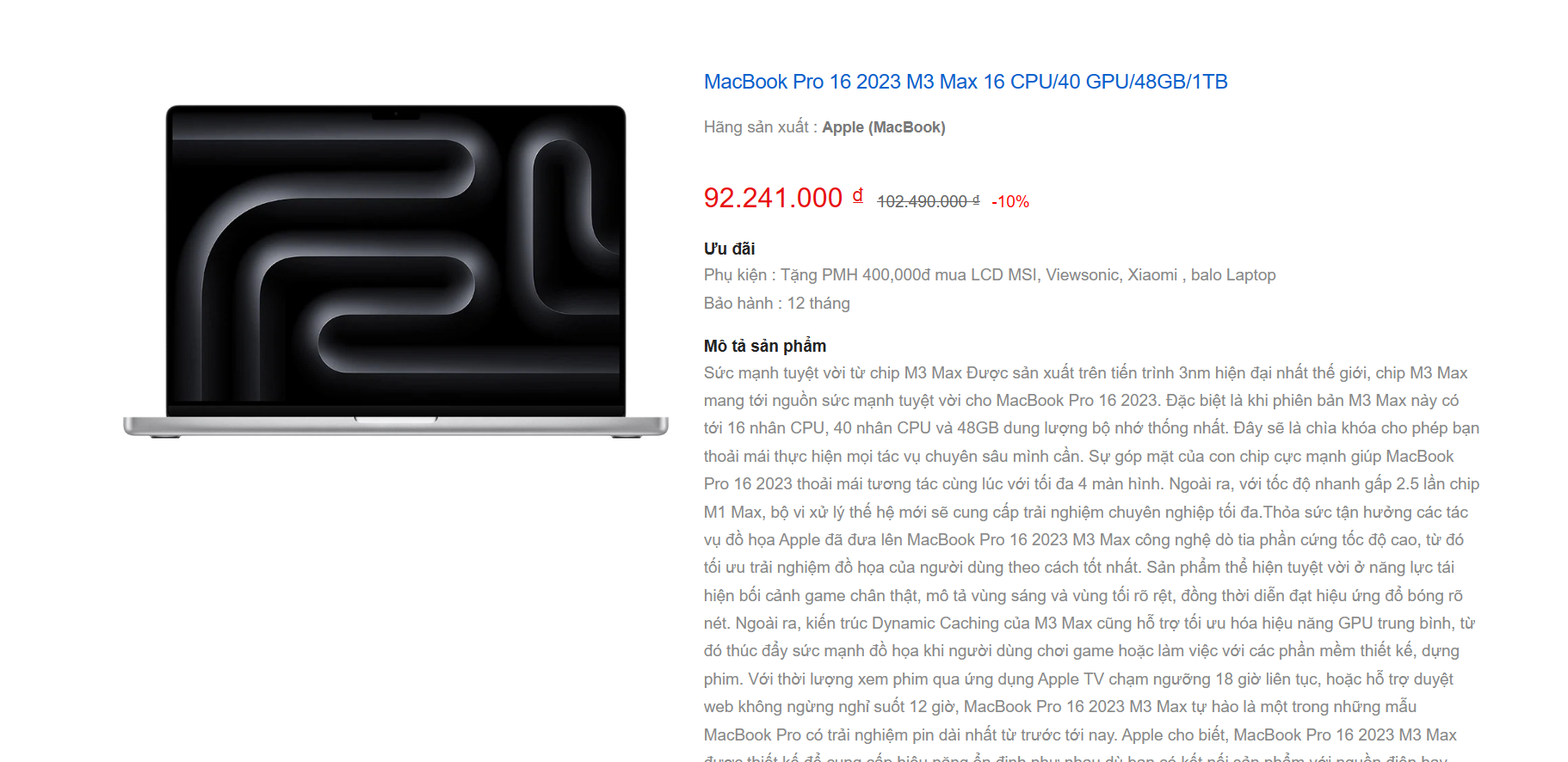
Hình 3.1: Giao diện trang chủ website

* Giao diện danh sách sản phẩm:



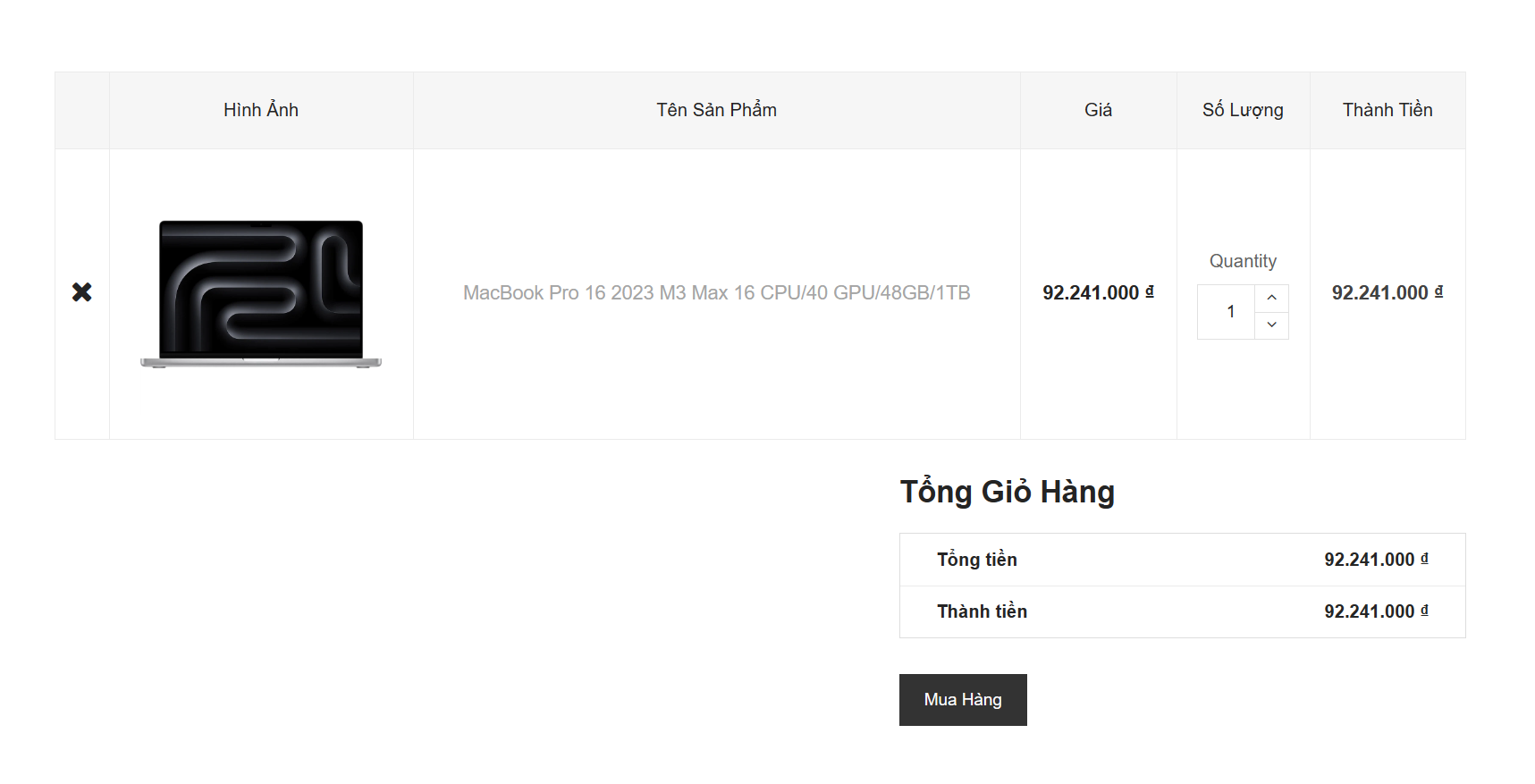
Hình 3.2: Giao diện danh sách sản phẩm

* Giao diện chi tiêt sản phẩm



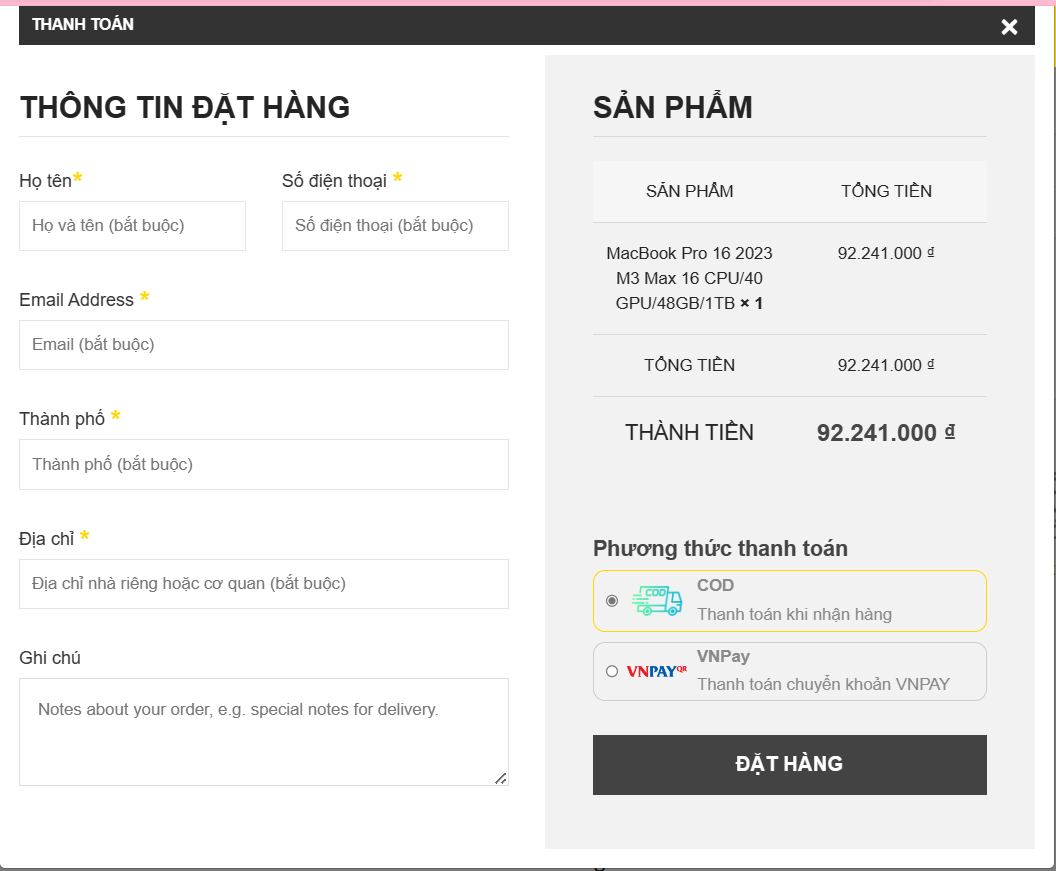
Hình 3.3: Giao diện chi tiết sản phẩm

* Giao diện giỏ hàng:



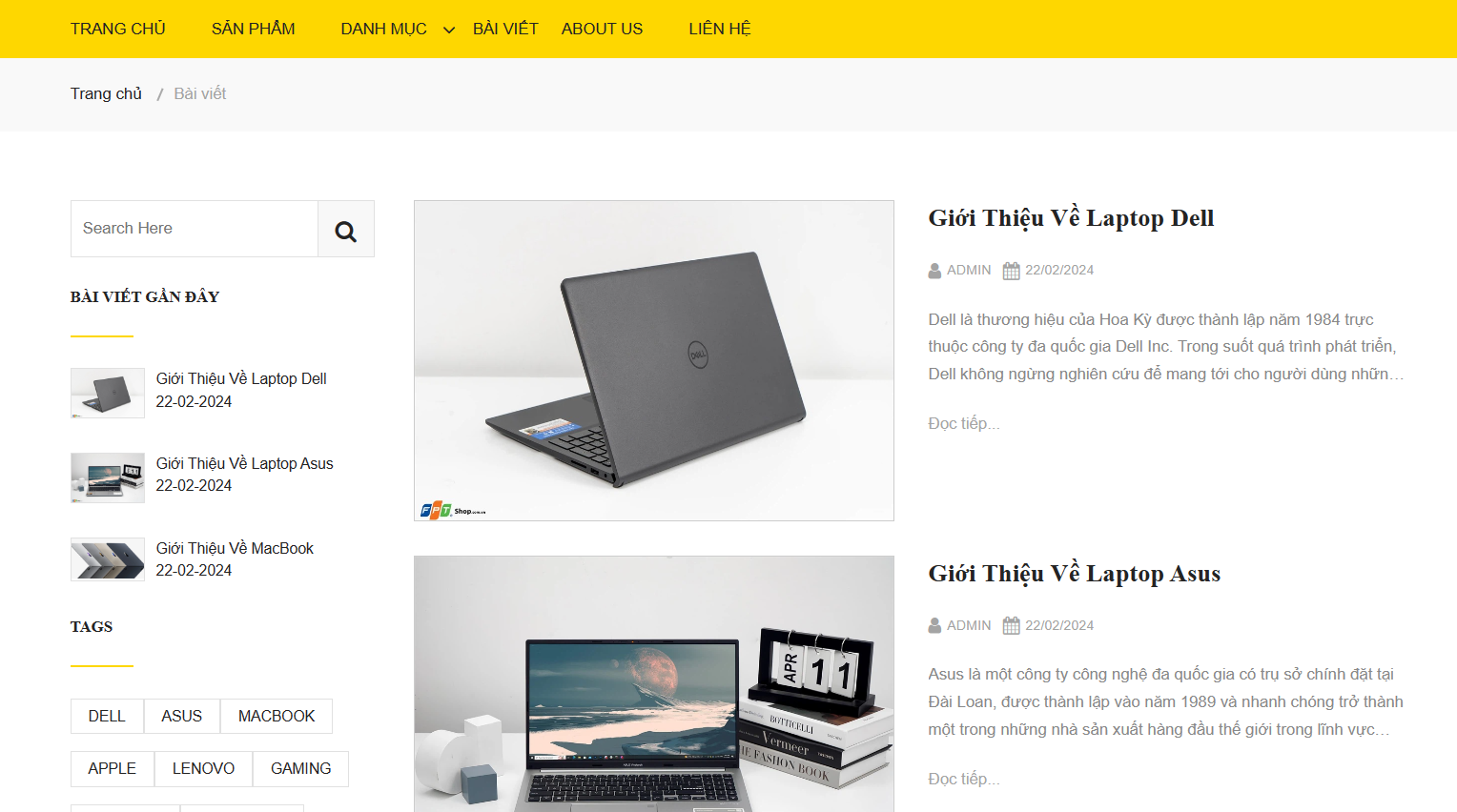
Hình 3.4: Giao diện giỏ hàng

* Giao diện mua hàng:



Hình 3.5: Giao diện mua hàng

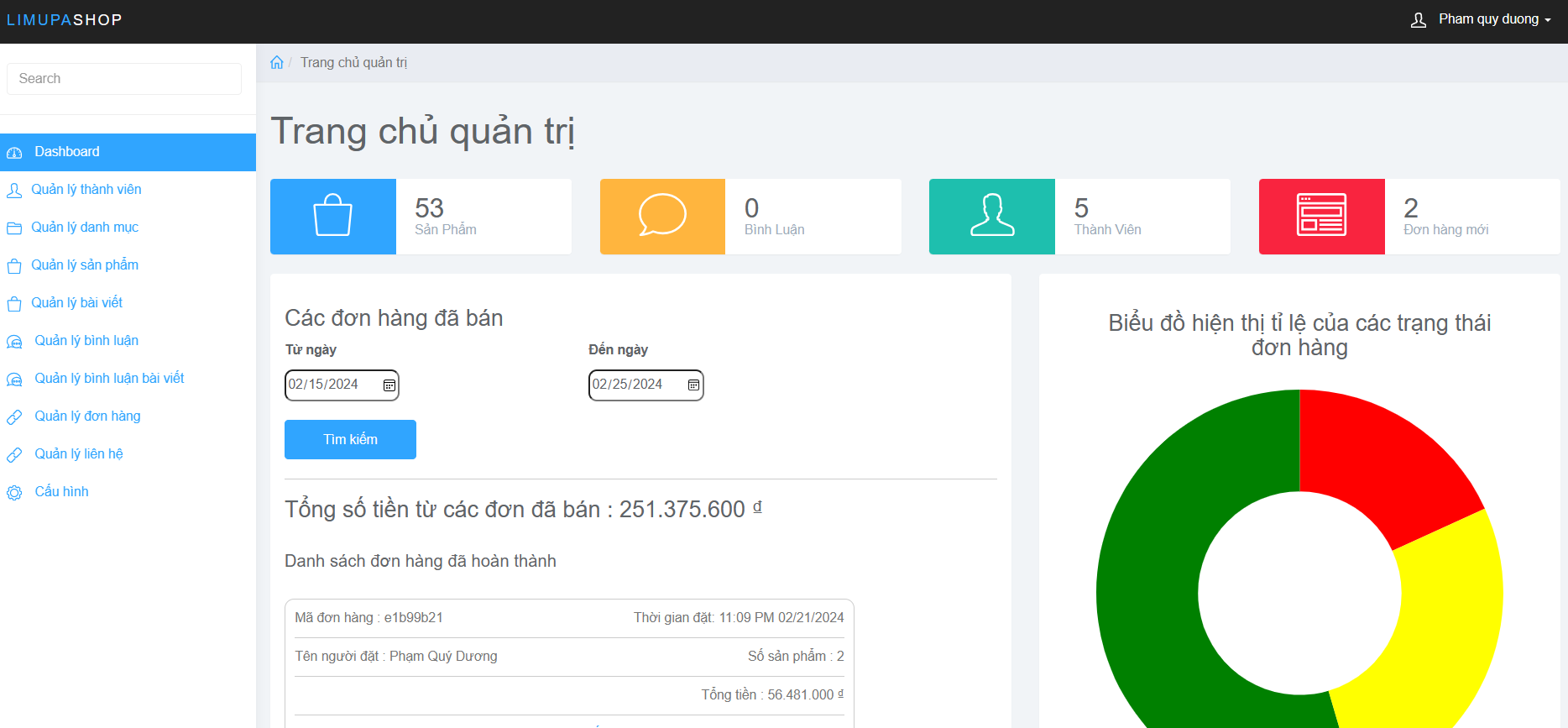
* Giao diện danh sách bài viết:



Hình 3.6: Giao diện danh sách bài viết

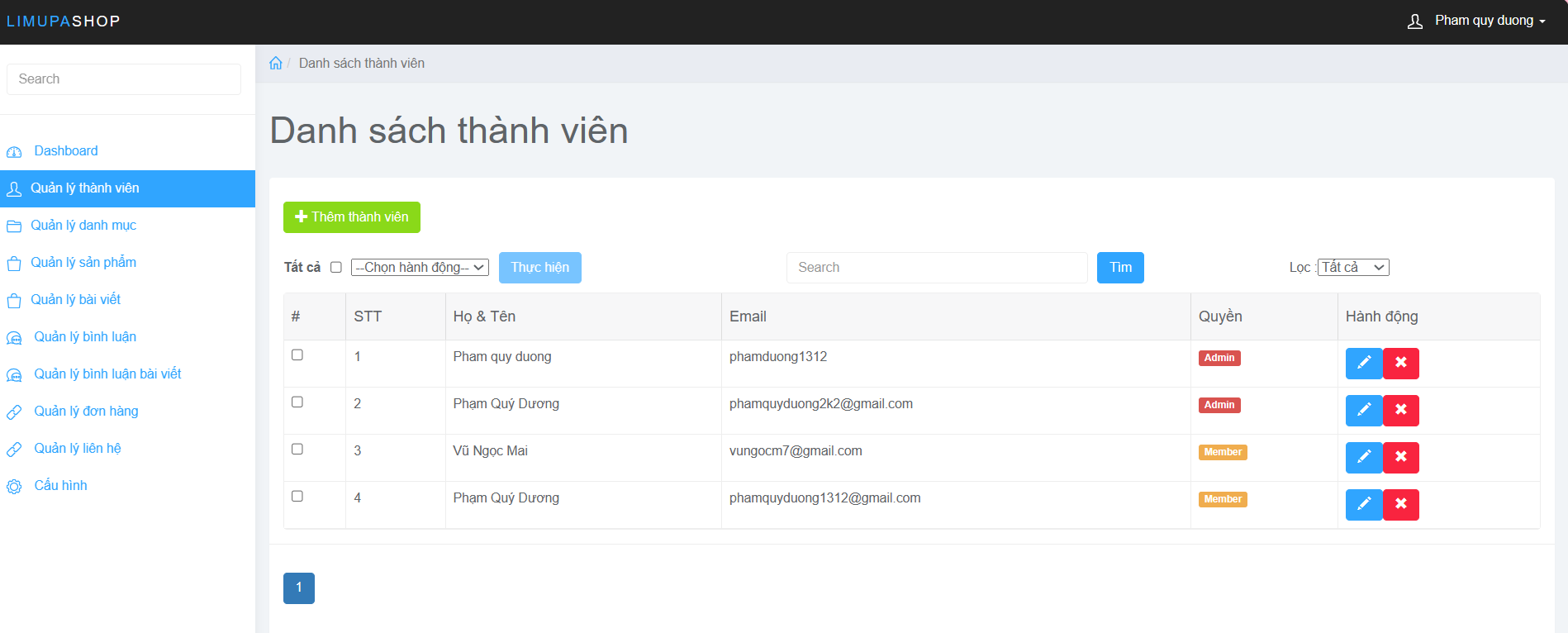
### Giao diện chức năng của người quản trị

* Giao diện trang chủ trang quản trị:



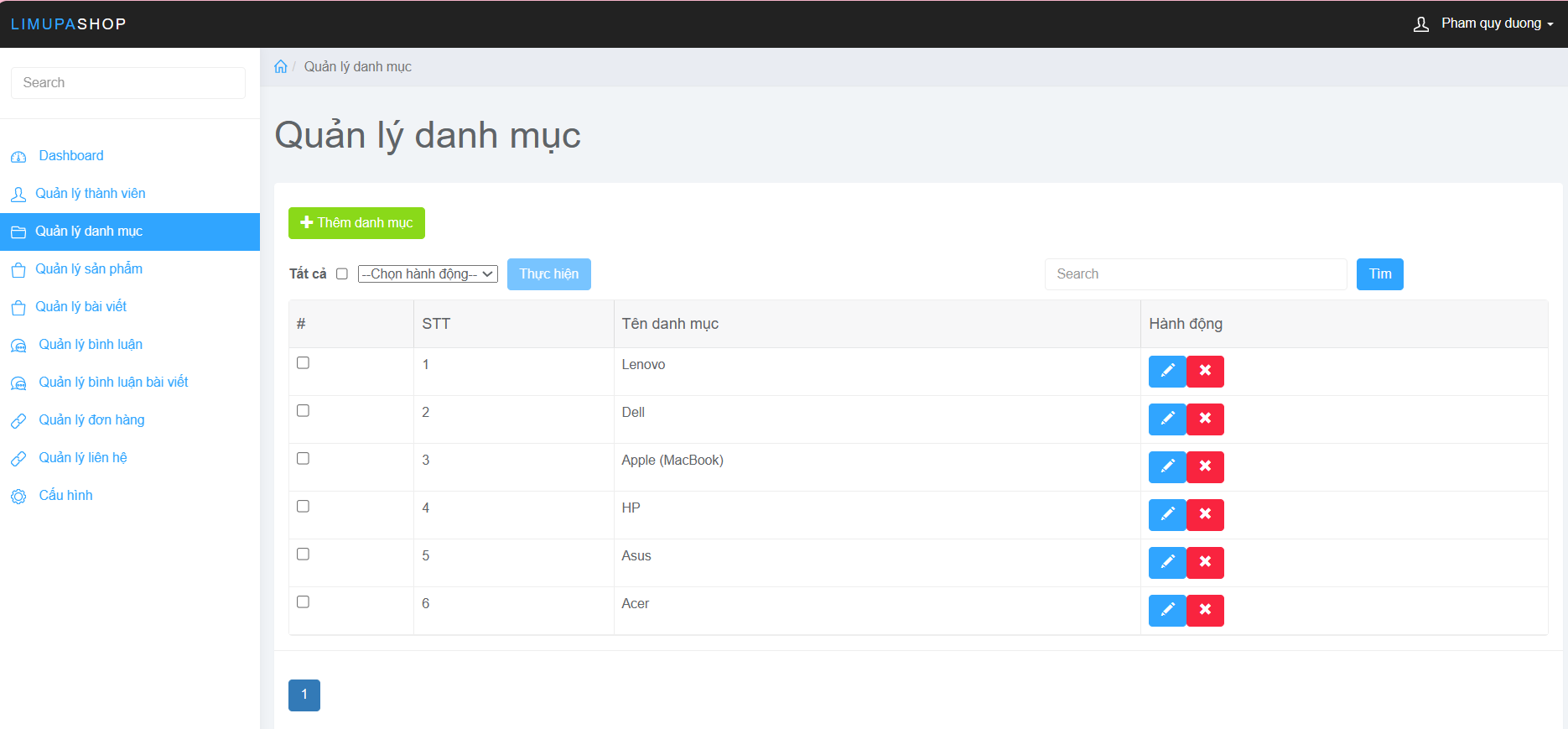
Hình 3.7: Giao diện trang chủ trang quản trị

* Giao diện quản lý tài khoản:



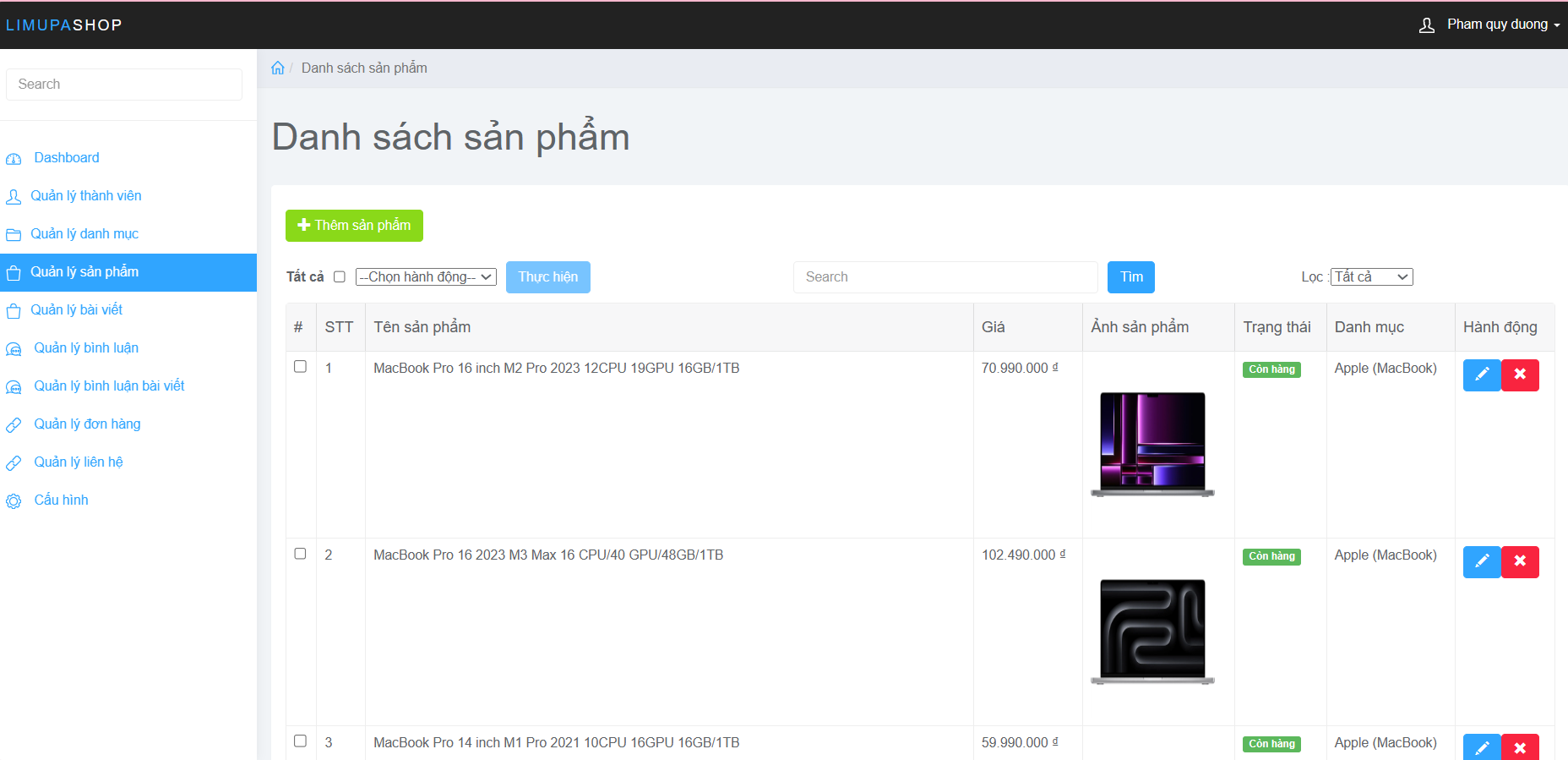
Hình 3.8: Giao diện quản lý tài khoản

* Giao diện quản lý danh mục



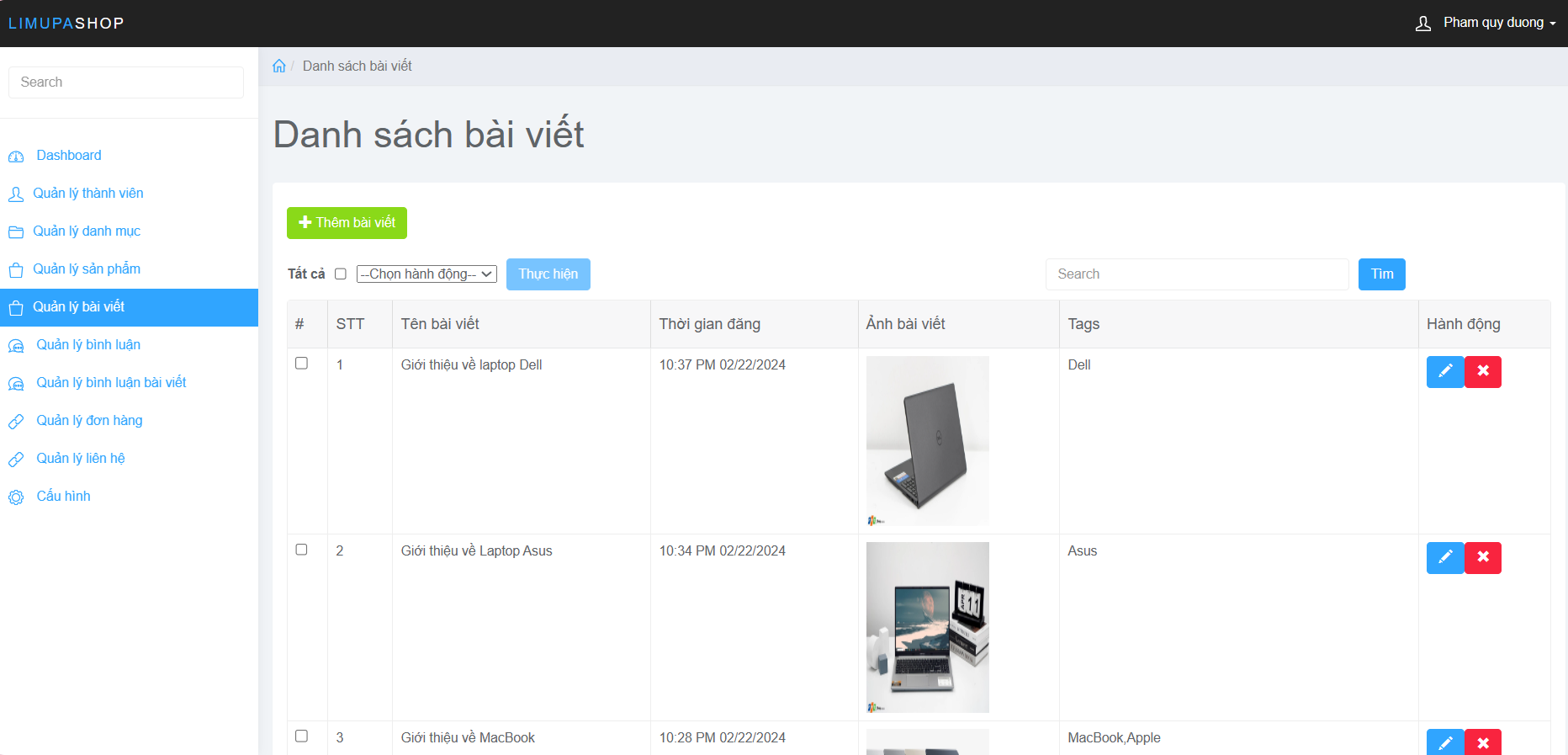
Hình 3.9: Giao diện quản lý danh mục

* Giao diện quản lý sản phẩm:



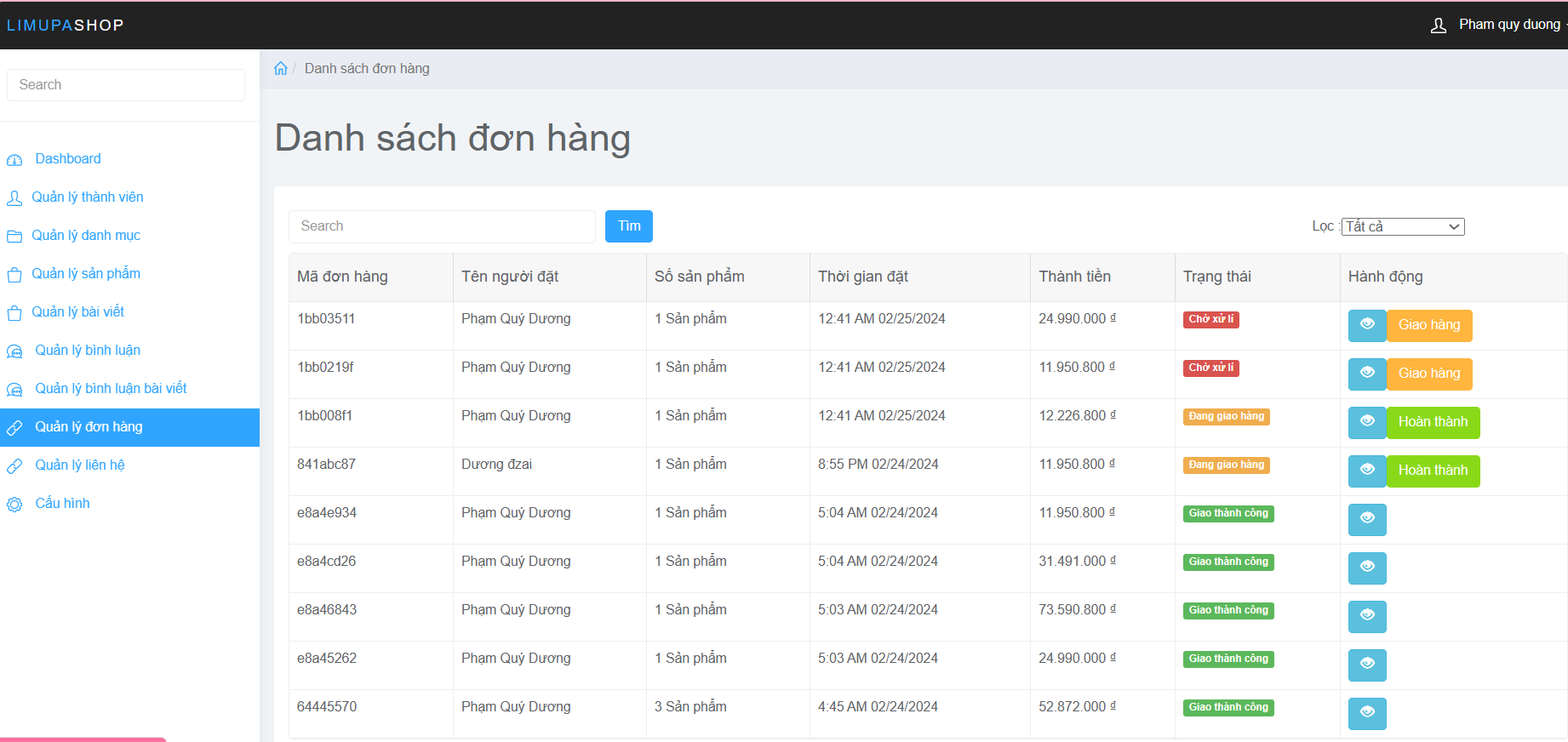
Hình 3.10: Giao diện quản lý sản phẩm

* Giao diện quản lý bài viết



Hình 3.11: Giao diện quản lý bài viết

* Giao diện quản lý đơn hàng:



Hình 3.12: Giao diện quản lý đơn hàng

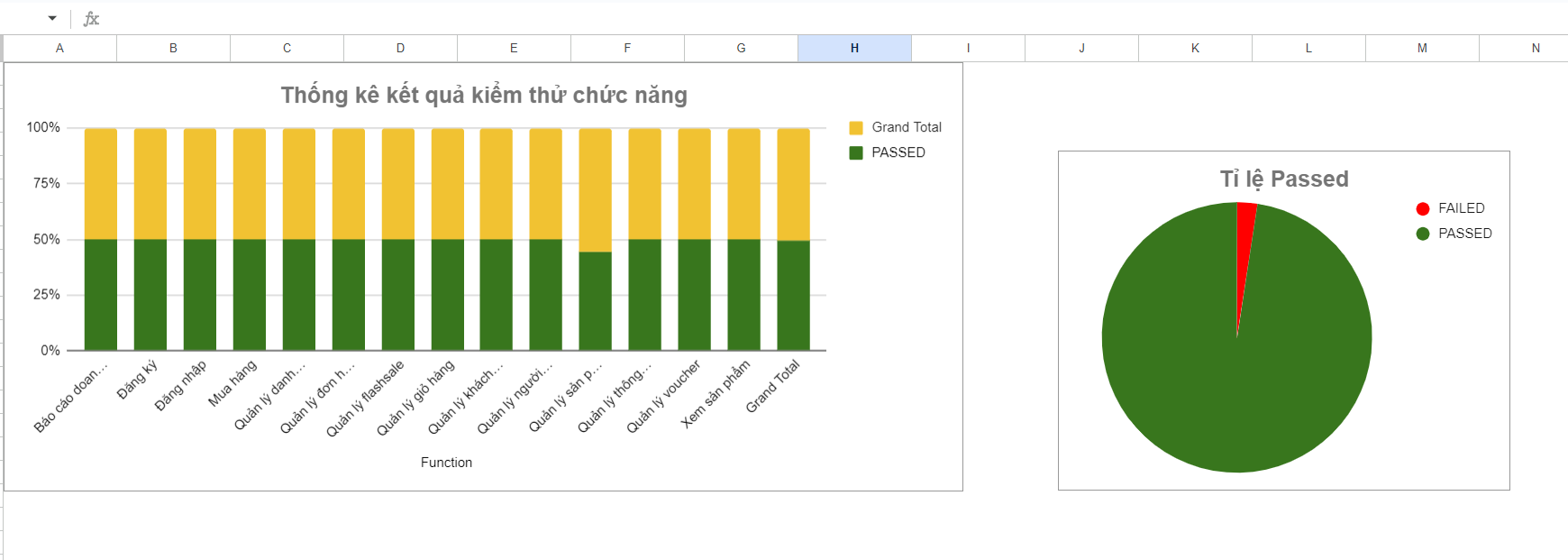
## Kiểm thử hệ thống

### Kế hoạch kiểm thử

* Mục tiêu: Rà soát, phát hiện bug tồn tại của sản phẩm phần mềm, từ đó đưa ra kế hoạch fix bug hiệu quả.
* Phạm vi: Kiểm thử các chức năng chính của website.
* Chiến lược kiểm thử: Kiểm thử thủ công ở mức System test trên cả UI và chức năng.

### Kết quả kiểm thử

* Tỉ lệ test đạt yêu cầu (passed): ~97,6%
* Tỉ lệ test chưa đạt yêu cầu (failed): ~2,4%
* Hệ thống chạy ổn định trên các trình duyệt web khác nhau (Microsoft Edge, Google chrome, cốc cốc)



Hình 3.13: Thống kê kết quả kiểm thử

**KẾT LUẬN**

Đề tài "Xây dựng Website bán laptop" là đề tài nổi bật, phản ánh nhu cầu thực tế của ngành kinh doanh hiện đại, chuyển từ phương pháp bán hàng truyền thống sang bán hàng online để nâng cao hiệu quả kinh doanh. Qua quá trình thực hiện, đã thu được kết quả đáng kể về công nghệ và cài đặt chương trình, bao gồm:

**\*Những kết quả đạt được:**

* Hiểu biết quy trình thiết kế website: từ khảo sát đến bảo trì.
* Xây dựng website bán laptop hiệu quả, quảng bá thương hiệu.
* Nắm vững kiến thức về HTML, CSS, JavaScript, EJS, ReactJs, Jquery.
* Thành thạo công cụ lập trình: Visual Studio Code, Postman.
* Website cho phép xem và mua hàng trực tuyến, cập nhật sản phẩm theo nhu cầu, quản lý tài khoản, sản phẩm, đơn hàng, khuyến mãi.
* Giao diện thân thiện, tính bảo mật cao.

**\*Hướng phát triển**

* Hoàn thiện và mở rộng tính năng, cải tiến chương trình.
* Quản lý hàng bị lỗi, kiểm soát hàng tồn.
* Tính năng đa ngôn ngữ, tích hợp AI để phân tích ý kiến khách hàng.
* Cải thiện bảo mật, xây dựng website lớn hơn với nhiều ứng dụng như tư vấn, gợi ý mua hàng.
* Tích hợp thanh toán đa dạng.

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

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

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

[6] Trang tài liệu NodeJS <https://nodejs.org/en/docs>

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

[8] Trang tài liệu Express <https://expressjs.com>

[9] Trang tài liệu EJS <https://ejs.co/#docs>

[10] Trang tài liệu Bootstrap <https://getbootstrap.com/docs/5.3/getting-started/introduction/>

[11] Trang tài liệu Jquery <https://www.w3schools.com/jquery/>