|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| NGUYỄN VĂN DŨNG  Nguyễn Văn Dũng | **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **KHOA CÔNG NGHỆ THÔNG TIN**  **======\*\*\*======**  **E:\dh-cn-hanoi.png**  **BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**  **NGÀNH: CÔNG NGHỆ THÔNG TIN**  **ĐỀ TÀI: XÂY DỰNG TRANG THƯƠNG MẠI ĐIỆN TỬ TỰ ĐỘNG ĐỒNG BỘ CÁC SẢN PHẨM TỪ NỀN TẢNG THỨ BA**   |  |  | | --- | --- | | Cán bộ hướng dẫn | **: ThS. Nguyễn Xuân Hoàng** | | Sinh viên thực hiện | **: Nguyễn Văn Dũng** | | Mã số sinh viên  Lớp | **: 2020603502**  **: CNTT03-K15** |   **Hà Nội – Năm 2024** |
| CÔNG NGH Ệ THÔNG TIN |

# MỤC LỤC

[MỤC LỤC i](#_Toc165815238)

[LỜI MỞ ĐẦU 1](#_Toc165815239)

[CHƯƠNG 1. TỔNG QUAN VỀ CÔNG NGHỆ VÀ CƠ SỞ LÝ THUYẾT 3](#_Toc165815240)

[1.1. NestJS 3](#_Toc165815241)

[1.1.1. Giới thiệu về NestJS 3](#_Toc165815242)

[1.1.2. Lịch sử phát triển của NestJS 3](#_Toc165815243)

[1.1.3. Ưu điểm của NestJS 4](#_Toc165815244)

[1.2. PostgreSQL 5](#_Toc165815245)

[1.2.1. Giới thiệu về PostgreQL 5](#_Toc165815246)

[1.2.2. Lịch sử phát triển của PostgreQL 5](#_Toc165815247)

[1.2.3. Ưu điểm của PostgreQL 5](#_Toc165815248)

[1.3. ReactJS 6](#_Toc165815249)

[1.3.1. Giới thiệu ReactJs 6](#_Toc165815250)

[1.3.2. Lịch sử phát triển 7](#_Toc165815251)

[1.3.3. Ưu điểm của ReactJS 7](#_Toc165815252)

[CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG VÀ THIẾT KẾ CƠ SỞ DỮ LIỆU 8](#_Toc165815253)

[2.1. Xác định các yêu cầu chính của hệ thống 8](#_Toc165815254)

[2.1.1. Phân tích các yêu cầu 8](#_Toc165815255)

[2.1.2. Các yêu cầu phi chức năng 9](#_Toc165815256)

[2.2. Xây dựng biểu đồ use case 9](#_Toc165815257)

[2.2.1. Các tác nhân 9](#_Toc165815258)

[2.2.2. Các use case của hệ thống 9](#_Toc165815259)

[2.2.3. Biểu đồ use case 11](#_Toc165815260)

[2.2.4. Mô tả chi tiết các use case 12](#_Toc165815261)

[2.2.5. Phân tích các use case 29](#_Toc165815262)

[2.3. Xây dựng cơ cở dữ liệu 53](#_Toc165815263)

[2.3.1. Biểu đồ Entity Relationship Diagram 53](#_Toc165815264)

[2.3.2. Các bảng trong cơ sở dữ liệu 53](#_Toc165815265)

[CHƯƠNG 3. KẾT QUẢ ĐẠT ĐƯỢC 56](#_Toc165815266)

[3.1. Giao diện chương trình 56](#_Toc165815267)

[3.1.1. Giao diện phía khách hàng 56](#_Toc165815268)

[3.1.2. Giao diện phía quản lí 64](#_Toc165815269)

[TÀI LIỆU THAM KHẢO 65](#_Toc165815270)

# LỜI MỞ ĐẦU

1. **Lý do chọn đề tài**

Hiện nay công nghệ thông tin không ngừng phát triển mạnh mẽ và hiện đại, các nền tảng số ngày càng tiếp cận tới người dùng dễ dàng hơn, trong đó có các hệ thống thương mại điện tử. Nhiều cửa hàng đã dần chuyển từ hình thức bán hàng truyền thống sang thương mại điện tử nhanh chóng với mức chi phí duy trì chỉ bằng một phần nhỏ với việc thuê mặt bằng.

Với mong muốn củng cố và nâng cao kiến thức về lập trình website, áp dụng kiến thức để xây dựng một trang website thương mại điện tử tích hợp đồng bộ sản phẩm giúp người dùng có thể dễ dàng sử dụng, mua sắm, tham khảo thiết bị công nghệ chất lượng cao với mức giá chuẩn xác, sử dụng hệ thống với công nghệ mới nhanh và mượt mà hơn em đã chọn đề tài “Xây dựng trang thương mại điện tử tự động đồng bộ các sản phẩm từ nền tảng thứ ba”.

Quá trình thực hiện đề tài của em có thể còn nhiều thiếu xót. Em rất mong nhận được sự cảm thông và chia sẻ của quý thầy cô để có thể đưa đề tài phát triển thêm mạnh mẽ hơn nữa. Em xin chân thành cảm ơn!

1. **Mục tiêu của đề tài**

Xây dựng được trang thương mại hoàn thiện gồm các module website bán hàng và các module quản trị hệ thống, đồng bộ sản phẩm từ các bên thứ ba. Rèn luyện các kiến thức và kỹ năng phát triển hệ thống phần mềm và thái độ làm việc chuyên nghiệp.

1. **Nội dung nghiên cứu**

* Khảo sát và phân tích yêu cầu quy trình quản lý và mua bán hàng hoá.
* Phân tích thiết kế hệ thống.
* Thiết kế cơ sở dữ liệu.
* Lập trình backend API.
* Lập trình giao diện và kết nối tới API.
* Có kế hoạch và đánh giá kiểm thử hệ thống.

1. **Phạm vi đề tài**

Đề tài mới xây dựng được website bán hàng online mức cơ bản, hỗ trợ thanh toán trực tuyến, đồng bộ sản phẩm từ Shopify, ShopBase, báo cáo thống kê doanh thu theo tháng, năm, đồng

1. **Bố cục đề tài**

Ngoài các phần Mở đầu, Kết luận và Tài liệu tham khảo, báo cáo đồ án được bố cục thành ba chương chính sau.

* Chương 1: Trình bày tổng quan về công nghệ và cơ sở lý thuyết.
* Chương 2: Phân tích thiết kế hệ thống và thiết kế cơ sở dữ liệu.
* Chương 3: Cài đặt phần mềm và kết quả đạt được.

# TỔNG QUAN VỀ CÔNG NGHỆ VÀ CƠ SỞ LÝ THUYẾT

## NestJS

### Giới thiệu về NestJS

Là framework để xây dựng web backend dựa trên Node.js mạnh mẽ, linh hoạt và TypeScript, NestJS sử dụng các khái niệm của Angular để tạo nên một cấu trúc ứng dụng dễ dàng quản lý. NestJS chuyên về việc xây dựng các ứng dụng web server-side hiện đại. NestJS có cấu trúc dự án gọn gàng và tổ chức mã nguồn dễ dàng và hợp lý, giúp việc phát triển và bảo trì ứng dụng trở nên dễ dàng hơn.

NestJS được xây dựng trên TypeScript, giúp tăng tính linh hoạt và sự rõ ràng trong việc phát triển ứng dụng. NestJS hỗ trợ dependency injection, giúp quản lý dependencies một cách dễ dàng và làm cho ứng dụng trở nên dễ mở rộng hơn. NestJS có khả năng sử dụng middleware để xử lý các yêu cầu HTTP, giúp tạo ra các ứng dụng có khả năng linh hoạt và mạnh mẽ. NestJS có một cộng đồng lớn và tích cực, với nhiều tài liệu và hướng dẫn hữu ích, giúp người dùng dễ dàng học và sử dụng NestJS.

### Lịch sử phát triển của NestJS

NestJS là một framework phát triển ứng dụng web server-side được xây dựng trên nền tảng Node.js. Nó ra đời vào năm 2017 và từ đó đã trở thành một trong những lựa chọn phổ biến cho việc xây dựng các ứng dụng web chất lượng cao.

Lịch sử phát triển của NestJS có các cột mốc quan trọng như sau:

Ra mắt lần đầu (2017): NestJS được tạo ra bởi Kamil Myśliwiec, một lập trình viên người Ba Lan. Ông cảm thấy rằng cộng đồng Node.js cần một framework có cấu trúc rõ ràng và quy mô lớn hơn để xây dựng các ứng dụng web phức tạp. NestJS được phát triển dựa trên các nguyên tắc của Angular, một framework phát triển ứng dụng web front-end phổ biến.

Phiên bản 2 (2018): Phiên bản NestJS 2 được phát hành vào năm 2018 với nhiều cải tiến và tính năng mới. Phiên bản này đã cải thiện hiệu suất và tăng tính năng của framework.

NestJS 5 (2018): Phiên bản NestJS 5 tiếp tục nâng cao khả năng và tích hợp của framework. Nó bao gồm sự hỗ trợ tốt hơn cho GraphQL và WebSocket, cũng như các tính năng mới như Middleware, Pipes và Guards.

NestJS 6 (2019): Phiên bản NestJS 6 tiếp tục cải thiện và mở rộng các tính năng của framework. Nó giới thiệu Dependency Injection Container và Caching module, cung cấp khả năng quản lý phụ thuộc tốt hơn và tăng cường hiệu suất.

NestJS 7 (2020): Phiên bản NestJS 7 tập trung vào việc cải thiện trải nghiệm phát triển và tăng tính tương thích. Nó cải thiện hệ thống validation, hỗ trợ TypeScript 3.7 và Node.js 14, và cung cấp các tính năng mới như Code Generation.

NestJS 8 (2021): Phiên bản NestJS 8 tiếp tục mang đến nhiều cải tiến và tính năng mới. Nó hỗ trợ Fastify, một framework web độc lập với hiệu suất cao, cung cấp cơ chế interceptor mới và cải thiện hệ thống ghi nhật ký.

NestJS đã thu hút sự quan tâm và sự ủng hộ từ cộng đồng phát triển Node.js. Nó đạt được sự phát triển nhanh chóng và trở thành một trong những framework phát triển ứng dụng web server-side phổ biến và được ưa chuộng. Với cấu trúc rõ ràng, tích hợp tốt và khả năng mở rộng, NestJS giúp các nhà phát triển xây dựng các ứng dụng web chất lượng cao một cách dễ dàng và hiệu quả.

### Ưu điểm của NestJS

Kiến trúc module và cấu trúc tổ chức rõ ràng: NestJS cung cấp một kiến trúc module để phân chia và tổ chức mã nguồn ứng dụng. Kiến trúc module cho phép bạn xác định rõ ràng các thành phần, phụ thuộc và lưu trữ logic liên quan trong một vùng cụ thể. Điều này giúp tạo ra mã nguồn dễ đọc, dễ bảo trì và dễ mở rộng.

TypeScript và Angular-like: NestJS sử dụng TypeScript làm ngôn ngữ chính, cho phép bạn sử dụng kiểu tĩnh, trình biên dịch và các tính năng của TypeScript. Điều này giúp tăng tính chính xác, khả năng tái sử dụng và sự dễ dàng trong quá trình phát triển. Ngoài ra, NestJS lấy cảm hứng từ cú pháp và các khái niệm của Angular, giúp cho việc chia sẻ kiến thức và mã nguồn giữa phía back-end và front-end trở nên dễ dàng hơn.

Dependency Injection (DI): NestJS sử dụng mô hình Dependency Injection để quản lý các phụ thuộc và tạo ra các thành phần có thể tái sử dụng. DI giúp giảm sự phụ thuộc giữa các thành phần, tạo ra mã nguồn linh hoạt, dễ kiểm thử và dễ bảo trì.

Hỗ trợ tốt cho Microservices: NestJS cung cấp các công cụ và thư viện tích hợp sẵn để phát triển và quản lý kiến trúc Microservices. Việc phát triển các dịch vụ nhỏ, độc lập và có khả năng mở rộng trở nên dễ dàng hơn với NestJS.

## PostgreSQL

### Giới thiệu về PostgreQL

PostgreSQL là một hệ thống quản trị cơ sở dữ liệu quan hệ và đối tượng (object-relational database management system) miễn phí và nguồn mở (RDBMS) tiên tiến nhất hiện nay. khả năng mở rộng cao và tuân thủ các tiêu chuẩn kỹ thuật. Nó được thiết kế để xử lý một loạt các khối lượng công việc lớn, từ các máy tính cá nhân đến kho dữ liệu hoặc dịch vụ Web có nhiều người dùng đồng thời.

### Lịch sử phát triển của PostgreQL

Những nguồn gốc (1986-1995): PostgreSQL bắt đầu dưới dạng dự án nghiên cứu tại Đại học California, Berkeley vào năm 1986. Ban đầu, nó được gọi là POSTGRES, một tên viết tắt của "Post Ingres". Sau đó, phiên bản đầu tiên của PostgreSQL đã được phát hành vào năm 1995.

Phát triển mã nguồn mở (1996-2000): Sau khi phát hành lần đầu, dự án PostgreSQL đã chuyển từ một dự án nghiên cứu sang một dự án mã nguồn mở được cộng đồng hỗ trợ rộng rãi. Trong giai đoạn này, sự phát triển chủ yếu dựa vào đóng góp từ cộng đồng.

Phiên bản 7.x và sự phát triển (2000-2005): Trong giai đoạn này, PostgreSQL đã trải qua sự phát triển đáng kể, với sự giới thiệu của các tính năng mới như hỗ trợ OLAP, XML, và các tính năng nâng cao về hiệu suất và an ninh.

Cải tiến về hiệu suất và tính năng (2005-2010): PostgreSQL 8.x đánh dấu giai đoạn mà PostgreSQL trở nên phổ biến hơn, đặc biệt là trong các ứng dụng web và doanh nghiệp. Cải tiến về hiệu suất và tính năng tiếp tục được thực hiện.

Phiên bản 9.x và tính năng đa dạng (2010-2015): PostgreSQL 9.x đưa vào nhiều tính năng mới quan trọng như JSONB, hỗ trợ index GIN, cải tiến về phân vùng và hiệu suất.

Sự phát triển toàn cầu (2015-nay): PostgreSQL ngày càng trở thành một trong những hệ quản trị cơ sở dữ liệu mã nguồn mở phổ biến nhất trên toàn thế giới, với sự hỗ trợ từ cộng đồng lớn và các tổ chức công nghệ hàng đầu. Các phiên bản mới tiếp tục mang lại nhiều tính năng mới và cải tiến về hiệu suất, an ninh và tính ổn định.

### Ưu điểm của PostgreQL

Mã nguồn mở: PostgreSQL là một dự án mã nguồn mở, điều này có nghĩa là mã nguồn của nó được công bố công khai và có thể được sửa đổi, phát triển bởi cộng đồng lập trình viên trên khắp thế giới. Điều này tạo điều kiện cho sự phát triển nhanh chóng, sự đa dạng tính năng và sự tin cậy thông qua sự kiểm tra và đóng góp của cộng đồng.

Tính mở rộng và linh hoạt: PostgreSQL hỗ trợ nhiều loại dữ liệu và các tính năng mở rộng, từ dữ liệu cơ bản như số, văn bản đến dữ liệu phức tạp như JSON, XML, và dữ liệu địa lý (GIS). Nó cung cấp khả năng linh hoạt cho việc mở rộng cấu trúc cơ sở dữ liệu để phù hợp với các nhu cầu cụ thể của ứng dụng.

Hiệu suất cao: PostgreSQL được tối ưu hóa để cung cấp hiệu suất cao, đặc biệt là trong các tác vụ xử lý dữ liệu lớn và phức tạp. Nó có các cơ chế như bộ chỉ mục thông minh, tối ưu hóa truy vấn và quản lý bộ nhớ để giúp tối ưu hóa hiệu suất.

An ninh mạnh mẽ: PostgreSQL cung cấp các tính năng an ninh mạnh mẽ để bảo vệ dữ liệu, bao gồm kiểm soát truy cập, mã hóa dữ liệu, xác thực và kiểm tra. Nó hỗ trợ các tiêu chuẩn bảo mật như SSL và GSSAPI, cùng với các tính năng bảo mật mở rộng như mã hóa dữ liệu cột và kiểm soát bởi cơ sở dữ liệu.

Hỗ trợ đa nền tảng: PostgreSQL có sẵn trên nhiều nền tảng hệ điều hành như Linux, Windows, macOS và các hệ điều hành Unix khác. Điều này tạo điều kiện cho tính di động và linh hoạt trong việc triển khai và vận hành hệ thống.

Cộng đồng lớn và hỗ trợ chuyên môn: PostgreSQL có một cộng đồng lớn và đa dạng với các chuyên gia và nhà phát triển trên toàn thế giới. Điều này cung cấp nguồn tài nguyên vô tận cho việc học hỏi, hỗ trợ và giải quyết vấn đề khi sử dụng hoặc triển khai PostgreSQL.

## ReactJS

### Giới thiệu ReactJs

ReactJS là thư viện mã nguồn mở của Facebook được viết bằng JavaScript với mục đích xây dựng giao diện người dùng.

Trong ReactJS, giao diện được chia thành các phần nhỏ gọi là component và có thể dễ dàng tái sử dụng.

ReactJS giúp giao diện của ứng dụng web có thể phản ứng nhanh nhẹn và mượt mà hơn.

### Lịch sử phát triển

ReactJS được phát triển bởi Jordan Walke – một kỹ sư phần mềm tại Facebook vào năm 2012. Lúc đầu Facebook chỉ áp dụng vào các sản phẩm của họ nhưng sau đó không lâu, vào tháng 5 năm 2013 ReactJS đã được mã nguồn mở tại JSConf US.

### Ưu điểm của ReactJS

* Tốc độ phát triển ứng dụng nhanh.
* Dễ dàng bảo trì và mở rộng.
* ReactJS được thiết kế cho các ứng dụng yêu cầu hiệu suất cao nhờ việc cung cấp DOM ảo và kết xuất phía máy chủ.
* Dễ dàng sử dụng cho người có kiến thức cơ bản về JavaScript.
* Bên cạnh đó ReactNative là một phiên bản khác của ReactJS giúp xây dựng ứng dụng trên cả nền tảng Android và iOS.

# PHÂN TÍCH THIẾT KẾ HỆ THỐNG VÀ THIẾT KẾ CƠ SỞ DỮ LIỆU

## Xác định các yêu cầu chính của hệ thống

### Phân tích các yêu cầu

**Những quyền lợi đối với khách hàng**

* Xem trang tĩnh: Khách hàng có quyền xem nội dung trang tĩnh trên website.
* Đăng nhập: Khách hàng có thể đăng nhập hệ thống.
* Đăng ký: Khách hàng đăng kí vào hệ thống.
* Cập nhật thông tin: Khách hàng có quyền cập nhật thông tin cá nhân của bản thân.
* Xem sản phẩm: Khách hàng có quyền xem danh sách sản phẩm và chi tiết sản phẩm.
* Tìm kiếm sản phẩm: Khách hàng có thể tìm kiếm sản phẩm theo từ khoá là tên sản phẩm.
* Thêm vào giỏ hàng: Khách hàng có thể thêm bất kỳ sản phẩm nào vào giỏ hàng.
* Đặt hàng: Khách hàng có quyền đặt hàng trên website khi đăng nhập hoặc không cần đăng nhập.
* Tra cứu đơn hàng: Khách hàng xem được những đơn hàng mà mình đã đặt trước đó.
* Quản lý giỏ hàng: Khách hàng có quyền xem, thêm, sửa, xoá các sản phẩm trong giỏ hàng.

**Những quyền lợi đối với quản trị viên**

* Đăng nhập: Quản trị viên cần đăng nhập vào trang quản lý để thực hiện các chứ năng quản trị hệ thống.
* Đồng bộ sản phẩm từ Shopify, ShopBase: hệ thống tự đồng đồng bộ hoặc có thể tự đồng bộ bằng tay.
* Bảo trì danh mục: xem, thêm, sửa, xoá danh mục.
* Bảo trì sản phẩm: xem, thêm sửa xoá sản phẩm.
* Quản lý người dùng: xem, cập nhật quyền, cập nhật trạng thái tài khoản.
* Quản lý đơn hàng: xem, cập nhật trạng thái đơn hàng.
* Xem thống kê

### Các yêu cầu phi chức năng

* Hiệu năng cao, trải nghiệm mượt mà, ổn định.
* Giao diện thân thiện.
* Đáp ứng được lượng lớn truy cập đồng thời.
* Tối ưu hoá SEO.
* Dễ dàng bảo trì và mở rộng.

## Xây dựng biểu đồ use case

### Các tác nhân

* Người quản trị: Quản lý nội dung website, xử lý các yêu cầu từ khách hàng.
* Khách hàng: Tìm kiếm, xem thông sản phẩm, đặt hàng.

### Các use case của hệ thống

Bảng 2.1. Các use case hệ thống

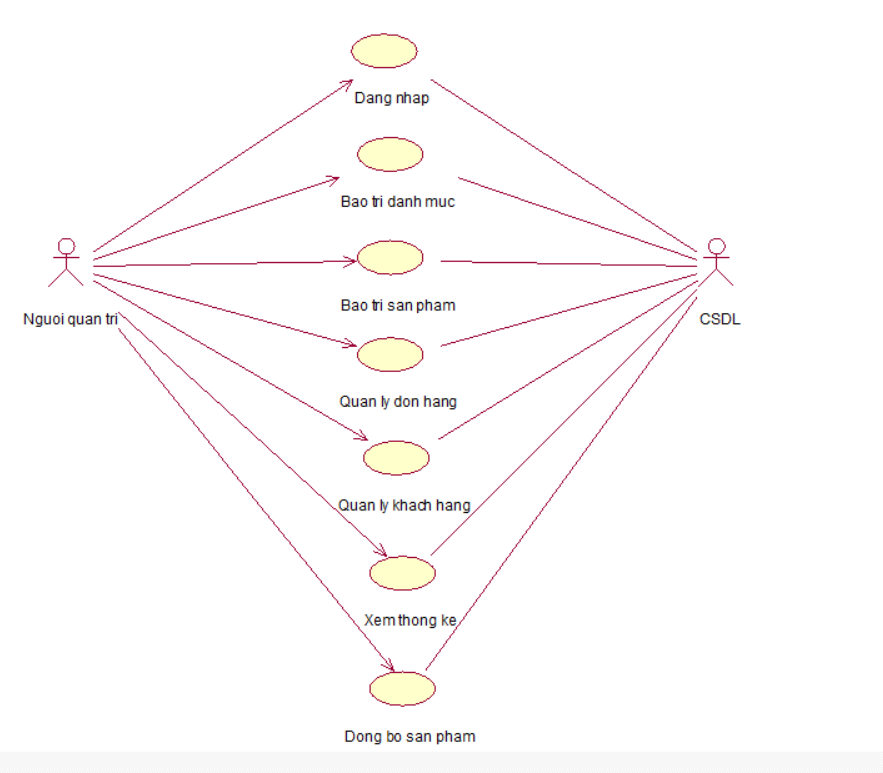
|  |  |
| --- | --- |
| **Tác nhân** | **Các ca sử dụng** |
| Khách hàng | Đăng nhập  Đăng ký  Cập nhật thông tin  Xem sản phẩm  Tìm kiếm sản phẩm  Thêm vào giỏ hàng  Đặt hàng  Quản lý giỏ hàng  Tra cứu đơn hàng |
| Người quản trị | Bảo trì danh mục  Bảo trì sản phẩm  Quản lý đơn hàng  Quản lý người dung  Đồng bộ sản phẩm  Xem thống kê |

### Biểu đồ use case

A diagram of a network

Description automatically generated

Hình 2.1. Các use case phía khách hàng



Hình 2.2. Các use case phía người quản trị

### Mô tả chi tiết các use case

#### Use case Đăng nhập

* **Mô tả vắn tắt**: Use case này cho phép khách hàng đăng nhập vào hệ thống.
* **Luồng sự kiện**:
  + **Luồng cơ bản**:

1. Use ase này bắt đầu khi khách hàng nhấn vào nút đăng nhập. Hệ thống sẽ yêu cầu khách hàng nhập email và mật khẩu.
2. Khách hàng nhập email và mật khẩu sau đó nhấn vào nút “Đăng nhập”. Hệ thống sẽ kiểm tra email, mật khẩu của khách hàng trong bảng USERS, nếu có hệ thống sẽ thông báo đăng nhập thành công và chuyển hướng đến trang chủ.

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 trong luồng cơ bản, khi người dùng nhập thiếu thông tin hoặc thông tin không hợp lệ hoặc không tìm thấy bản ghi trong bảng USERS. Hệ thống sẽ hiển thị thông báo lỗi và yêu cầu nhập lại. Người dùng có thể nhập lại để tiếp tục hoặc kết thúc thao tác.

**Các yêu cầu đặc biệt**: Khách hàng cần nhập đúng thông tin để yêu cầu đăng nhập.

* **Các yêu cầu đặc biệt:**

Không có.

* **Tiền điều kiện:**

Không có.

* **Hậu điều kiện:**

Không có.

* **Điểm mở rộng:**

Không có.

#### Use case Đăng ký

* **Mô tả vắn tắt**: Use case này cho phép người dùng đăng ký tài khoản trên hệ thống.
* **Luồng sự kiện**:
  + **Luồng cơ bản**:

1. Use case này bắt đầu khi người dùng kích chuột vào nút “Đăng nhập”. Hệ thống điều hướng sang màn hình chứa form “Đăng nhập”. Hệ thống yêu cầu nhập thông tin đăng ký: name, email, password.
2. Khách hàng nhập name, email và password sau đó nhấn vào nút “Create an Accounts”. Hệ thống sẽ kiểm tra email, password của khách hàng trong bảng USERS, nếu chưa có hệ thống sẽ gửi mail xác thực tài khoản đến tài khoản vừa đăng ký. Sau khi người dùng xác thực tài khoản thành công hệ thống sẽ thông báo “Đăng ký thành công” và chuyển hướng đến trang chủ.

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 trong luồng cơ bản, khi người dùng nhập thiếu thông tin, thông tin không hợp lệ hoặc email đã được đăng ký trong bảng USERS. Hệ thống sẽ hiển thị thông báo lỗi và yêu cầu nhập lại. Người dùng có thể nhập lại để tiếp tục hoặc kết thúc thao tác.

**Các yêu cầu đặc biệt**: Khách hàng cần nhập đúng thông tin để yêu cầu đăng nhập.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Không có.

* Hậu điều kiện:

Không có.

* Điểm mở rộng:

Không có.

#### Use case Cập nhật thông tin

* **Mô tả vắn tắt:** Use case này cho phép người dùng thay đổi thông tin về họ tên, ngày sinh, giới tính, địa chỉ, số điện thoại.
* **Luồng sự kiện**
  + **Luồng cơ bản:**

1. Use case này bắt đầu khi người dùng kích chuột vào trang cá nhân và nhấn vào nút “Thông tin cá nhân” Trang sẽ điều hướng sang màn hình “Thông tin cá nhân” và hiển thị thông tin cá nhân của khách hàng bao gồm: username, email từ bảng USERS sau đó hiển thị lên màn hình.
2. Sửa thông tin cá nhân: User nhập thông tin muốn thay đổi và nhấn vào nút “Cập nhật”. Hệ thống cập nhật thông tin vừa thay đổi vào bảng USERS, sau đó lấy thông tin cá nhân của người dùng đang đăng nhập và hiển thị lên màn hình.
   * **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.

Use case kết thúc.

* **Các yêu cầu đặc biệt:**

Không có.

* **Tiền điều kiện:**

Không có.

* **Hậu điều kiện:**

Sau khi use case kết thúc thành công thì các thông tin cá nhân của người dùng sẽ được cập nhật ở bảng USERS.

* **Điểm mở rộng:**

Không có.

#### Use case Xem sản phẩm

* **Mô tả vắn tắt:** Use case này cho phép khách hàng xem thông tin sản phẩm.
* **Luồng sự kiện:**
  + **Luồng cơ bản:**

1. Use case này bắt đầu khi người dùng nhấn vào một sản phẩm. Hệ thống sẽ lấy danh sách sản phẩm, ảnh, mô tả, đánh giá, của sản phẩm từ bảng PRODUCTS, PRODUCT\_IMAGES để 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 lên 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 Tìm kiếm sản phẩm

* **Mô tả vắn tắt**: Use này cho phép khách hàng tìm kiếm sản phẩm theo từ khoá là tên sản phẩm.
* **Luồng sự kiện**:
  + **Luồng cơ bản**:

1. Use case này bắt đầu khi khách hàng nhập tên sản phẩm vào textbox tìm kiếm trên thanh input tìm kiếm. Sau 0,8s hệ thống sẽ hiển thị ra các sản phẩm có tên chứa những từ mà người dùng đã nhập và hiển thị thông tin sản phẩm ra màn hình (tên, hình ảnh, mô tả, tồn kho, giá tiền) từ bảng PRODUCTS, PRODUCT\_IMAGES.

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 lên thông báo lỗi.
2. Tại bước 1 trong luồng cơ bản, nếu hệ thống không tìm được sản phẩm nào phù hợp với text đã nhập thì hệ thống sẽ hiển thị kết là “Không tìm thấy sản phẩm nào”. 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 Thêm vào giỏ hàng

* **Mô tả vắn tắt:** Use case này cho phép khách hàng thêm bất kỳ sản phẩm nào vào giỏ hàng
* **Luồng sự kiện**
  + **Luồng cơ bản:**

1. Người dùng nhấn vào nút “Thêm vào giỏ hàng” trong chi tiết sản phẩm sau khi chọn số lượng cần mua hoặc icon giỏ hang ngay Item.
2. Hệ thống sẽ lấy ra thông tin các sản phẩm người dùng đã thêm vào giỏ hàng của mình từ bảng “ORDER\_DETALS”, nếu sản phẩm muốn thêm đã tồn tại trong giỏ hàng hệ thống sẽ cộng số lượng sản phẩm muốn thêm vào sản phẩm trong giỏ hàng, nếu sản phẩm muốn thêm chưa tồn tại hệ thống sẽ tạo mới một bản ghi và lưu vào bảng “ORDER \_DETAILS”

* **Luồng rẽ nhánh**

1. Tại thời điểm bất kỳ 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.
2. Tại bước 1 trong luồng cơ bản, nếu người dùng chưa chọn số lượng sản phẩm muốn thêm vào giỏ hàng. Hệ thống sẽ hiển thị thông báo lỗi và yêu cầu nhập. Người dùng có thể nhập lại để tiếp tục hoặc kết thúc thao tác.

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

* **Mô tả vắn tắt:** Use case này cho phép khách hàng đặt những sản phẩm trên trang web.
* **Luồng sự kiện:**
  + **Luồng cơ bản:**

1. Tại màn hình giỏ hàng, người dùng bấm vào nút “Thanh toán”, hệ thống sẽ chuyển sang màn hình thông tin khách hàng và thông tin của đơn đặt hàng.
2. Người dùng tiến hành nhập đầy đủ thông tin: Name, Phone number, Address, Delivery Methob.
3. Người dùng nhấn nút “Place an order” hệ thống sẽ thực hiện tạo thêm bản ghi mới vào bảng ORDERS, ORDER\_DETAILS và cập nhật lại thông tin bảng PRODUCTS sau đó hiển thị “Đặt hàng thành công”.
4. Hệ thống gửi mail đến người dùng thông tin đơn hàng vừa được người dùng đặt.

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 hiển thị thông báo lỗi.
2. Tại bước 2 trong luồng cơ bản, nếu người dùng đã đăng nhập thì sẽ hiển thị thông tin của người dùng đã đăng nhập lên màn hình. Nếu người dùng không nhập đủ thông tin thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.
3. Tại bước 3 trong luồng cơ bản, nếu số lượng sản phẩm tồn kho nhỏ hơn số lượng đặt hàng thì sẽ hiển thị thông báo lỗi “Số lượng không hợp lệ” và use case kết thúc.

* **Các yêu cầu đặc biệt**

Không có

* **Tiền điều kiện**

Khách hàng cần cung cấp đầy đủ thông tin để có thể đặt hàng.

* **Hậu điều kiện**

Khi use case kết thúc thành công, các thông tin sẽ được lưu trữ vào bảng ORDERS, ORDER\_DETAILS

* **Điểm mở rộng:**

Không có

#### Use case Tra cứu đơn hàng

* **Mô tả vắn tắt:** Use case này cho phép người dùng xem được những đơn hàng mà mình đã đặt trước đó.
* **Luồng sự kiện:**
  + **Luồng cơ bản:**

1. Use case này bắt đầu khi người dùng nhấn vào nút “Trang cá nhân” trên thanh Navigation. Hệ thống sẽ thực hiện lấy thông tin về các đơn hàng của người dùng đang đăng nhập từ bảng ORDERS và ORDER\_DETAILS sau đó hiển thị lên màn hình.
2. Người dùng nhấn vào đơn hàng, hệ thống sẽ hiển thị thông tin chi tiết của đơn hàng đó bao gồm thông tin người mua, danh sách sản phẩm trong đơn hàng từ bảng ORDERS và ORDER\_DETAILS.

* **Luồng rẽ nhánh**

1. Tại thời điểm bất kỳ trong quá trình thực hiện use case, nếu không kết nối được tới cơ sở dữ liệu thì sẽ hiển thị thông báo lỗi.
2. Tại bước 1 trong luồng rẽ nhánh, nếu người dùng chưa đăng nhập hệ thống sẽ chuyển hướng sang trang đăng nhập.

Use case kết thúc.

* **Các yêu cầu đặc biệt**

Không có

* **Tiền điều kiện**

Người dùng đăng nhập hệ thống

* **Hậu điều kiện**

Không có

* **Điểm mở rộng:**

Không có

#### Use case Quản lý giỏ hàng

* **Mô tả vắn tắt:** Use case này cho phép khách hàng xem, thêm, sửa, xoá sản phẩm ra khỏi giỏ hàng.
* **Luồng sự kiện**
  + **Luồng cơ bản:**

1. Use case này bắt đầu khi người dùng kích vào nút “Giỏ hàng” trên menu. Hệ thống sẽ  lấy thông tin chi tiết giỏ hàng của người dùng đó gồm: mã giỏ hàng từ bảng “ORDER” và danh sách các sản phẩm thuộc giỏ hàng này từ bảng “ORDER\_DETAILS” và bảng “PRODUCTS”, hiển thị danh sách các sản phẩm lên màn hình.
2. Xoá sản phẩm trong giỏ hàng: Người dùng nhấn vào biểu tượng xoá trên màn hình, hệ thống sẽ thực hiện loại bỏ sản phẩm đó ra khỏi giỏ hàng.

* **Luồng rẽ nhánh**

1. Tại thời điểm bất kỳ 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.
2. Tại bước 1 trong luồng cơ bản, nếu chưa có sản phẩm nào trong giỏ hàng sẽ hiển thị “Giỏ hàng trống”.

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 Bảo trì danh mục

* **Mô tả vắn tắt:** Use case này cho phép quản trị viên xem, thêm, sửa, xoá danh mục.
* **Luồng sự kiện**
  + **Luồng cơ bản:**

1. Use case này bắt đầu khi người dùng nhấn vào nút “Danh mục” trên thanh menu quản trị. Hệ thống sẽ lấy trong bảng CATEGORIES gồm các thông tin: id, tên, mô tả và hiển thị lên màn hình.
2. Thêm danh mục mới:
   1. Người quản trị nhấn vào nút “Thêm danh mục” trong màn hình danh sách danh mục. Hệ thống sẽ hiển thị modal yêu cầu nhập thông tin cho danh mục gồm tên danh mục và mô tả cho danh mục.
   2. Người quản trị nhập thông tin cho danh mục mới và nhấn vào nút “Gửi”. Hệ thống sẽ thêm một bản ghi mới vào bảng CATEGORIES sau đó lấy thông tin về danh sách danh mục mới gồm id, tên và hiển thị lên màn hình.
3. Sửa danh mục:
   1. Người quản trị nhấn vào biểu tưởng chỉnh sửa trên một dòng danh mục. Hệ thống sẽ lấy ảnh, tên, mô tả của danh mục được chọn từ bảng CATEGORIES.
   2. Người quản trị sẽ sửa tên, mô tả của danh mục sau đó nhấn vào nút “Cập nhật”. Hệ thống cập nhật thông tin danh mục vào bảng CATEGORIES sau đó lấy thông tin danh sách danh mục gồm id, tên và hiển thị lên màn hình.
4. Xoá danh mục:
   1. Người quản trị nhấn vào biểu tượng xoá trên một dòng danh mục. Hệ thống sẽ hiển thị lên popup để xác nhận xoá.
   2. Người quản trị nhấn vào nút “Gửi”, hệ thống sẽ thực hiện xoá danh mục đã chọn khỏi bảng CATEGORIES sau đó lấy thông tin danh sách danh mục gồm id, tên và hiển thị lên màn hình.

Use case kết thúc.

* **Luồng rẽ nhánh:**

1. Tại thời điểm bất kỳ 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 1 trong luồng cơ bản, nếu không có bản ghi nào trong bảng CATEGORIES thì hệ thống sẽ hiển thị “Không có danh mục sản phẩm” và use case kết thúc.
3. Tại bước 2b và 3b trong luồng cơ bản, khi quản trị viên nhập thông tin danh mục không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi yêu cầu người quản trị nhập lại, quản trị viên có thể tiếp tục nhập lại hoặc nhấn đóng modal để kết thúc.
4. Tại bước 4b trong luồng cơ bản, nếu người quản trị nhấn vào nút không thì hệ thống sẽ bỏ qua thao tác xoá và hiển thị danh sách các danh mục trong bảng CATEGORIES lên màn hình và use case kết thúc.

* **Các yêu cầu đặc biệt**

Cần kiểm soát quyền thực hiện use case này để đảm bảo tính an toàn và bảo mật.

* **Tiền điều kiện**

Người quản trị cần đăng nhập với quyền quản trị trước khi thực hiện use case.

* **Hậu điều kiện**

Sau khi use case kết thúc thành công thì lưu các thông tin vào bảng CATEGORIES

* **Điểm mở rộng**

Không có

#### Use case Bảo trì sản phẩm

* **Mô tả vắn tắt:** Use case này cho phép quản trị viên xem, thêm, sửa, xoá sản phẩm.
* **Luồng sự kiện**
  + **Luồng cơ bản:**

1. Use case này bắt đầu khi người dùng nhấn vào nút “Sản phẩm” trên thanh menu quản trị. Hệ thống sẽ lấy trong bảng PRODUCTS, CATEGORIES gồm các thông tin: id, name, stock, price, brand, category, thumbnail và hiển thị lên màn hình.
2. Thêm sản phẩm mới:
   1. Người quản trị nhấn vào nút “Add product” trong màn hình danh sách sản phẩm. Hệ thống sẽ hiển thị modal yêu cầu nhập thông tin cho sản phẩm gồm tên sản phẩm, tồn kho, giá, hãng, danh mục, ảnh.
   2. Người quản trị nhập thông tin cho sản phẩm mới và nhấn vào nút “Gửi”. Hệ thống sẽ thêm một bản ghi mới vào bảng PRODUCTS sau đó lấy thông tin về danh sách sản phẩm mới gồm id, name, stock, price, brand, category, thumbnail và hiển thị lên màn hình.
3. Sửa sản phẩm:
   1. Người quản trị nhấn vào icon “Sửa” trên một dòng sản phẩm. Hệ thống sẽ tên sản phẩm, tồn kho, giá, hãng, danh mục, ảnh từ bảng PRODUCTS.
   2. Người quản trị sẽ sửa các thông tin của sản phẩm sau đó nhấn vào nút “Gửi”. Hệ thống cập nhật thông tin sản phẩm vào bảng PRODUCTS sau đó lấy thông tin danh sách sản phẩm gồm tên sản phẩm, tồn kho, giá, hãng, danh mục, ảnh và hiển thị lên màn hình.
4. Xoá sản phẩm:
   1. Người quản trị nhấn vào biểu tượng xoá trên một dòng sản phẩm. Hệ thống sẽ hiển thị lên popup để xác nhận xoá.
   2. Người quản trị nhấn vào nút “Đồng ý”, hệ thống sẽ thực hiện xoá thương hiệu đã chọn khỏi bảng PRODUCTS sau đó lấy thông tin danh sách sản phẩm gồm tên sản phẩm, tồn kho, giá, hãng, danh mục, ảnh và hiển thị lên màn hình.

Use case kết thúc.

* **Luồng rẽ nhánh:**

1. Tại thời điểm bất kỳ 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 1 trong luồng cơ bản, nếu không có bản ghi nào trong bảng PRODUCTS thì hệ thống sẽ hiển thị “Không có sản phẩm” và use case kết thúc.
3. Tại bước 2b và 3b trong luồng cơ bản, khi quản trị viên nhập thông tin sản phẩm không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi yêu cầu người quản trị nhập lại, quản trị viên có thể tiếp tục nhập lại hoặc nhấn đóng modal để kết thúc.
4. Tại bước 4b trong luồng cơ bản, nếu người quản trị nhấn vào nút không thì hệ thống sẽ bỏ qua thao tác xoá và hiển thị danh sách các sản phẩm trong bảng PRODUCTS lên màn hình và use case kết thúc.

* **Các yêu cầu đặc biệt**

Cần kiểm soát quyền thực hiện use case này để đảm bảo tính an toàn và bảo mật.

* **Tiền điều kiện**

Người quản trị cần đăng nhập với quyền quản trị trước khi thực hiện use case.

* **Hậu điều kiện**

Sau khi use case kết thúc thành công thì lưu các thông tin vào bảng PRODUCTS.

* **Điểm mở rộng**

Không có

#### Use case Quản lý đơn hàng

* **Mô tả vắn tắt:** Use case này cho phép quản trị viên xem, thay đổi trạng thái đơn hàng.
* **Luồng sự kiện**
  + **Luồng cơ bản:**

1. Use case này bắt đầu khi người dùng nhấn vào nút “Quản lý đơn hàng” trên thanh menu quản trị. Hệ thống sẽ lấy trong bảng ORDERS, ORDER\_DETAILS gồm các thông tin: id, người mua, số điện thoại, địa chỉ, trạng thái, tổng giá trị, tổng số lượng sản phẩm, ngày tạo và hiển thị lên màn hình.
2. Xem chi tiết đơn hàng:
   1. Người quản trị nhấn vào biểu tượng chi tiết trên một dòng của đơn hàng. Hệ thống sẽ lấy ra thông tin đơn hàng gồm mã, trạng thái, người mua, số điện thoại, địa chỉ nhận hàng, danh sách các sản phẩm trong giỏ hàng từ bảng ORDER, ORDER\_DETAIL, PRODUCTS và hiển thị lên màn hình.
3. Cập nhật trạng thái đơn hàng:
   1. Người quản trị nhấn vào biểu tượng cập nhật trên một dòng đơn đặt hàng. Hệ thống sẽ thay đổi trạng thái của đơn hàng theo lựa chọn của người quản trị. Các trạng thái là:
4. Huỷ đơn hàng:
   1. Người quản trị nhấn vào biểu tượng huỷ trên một dòng thương hiệu. Hệ thống sẽ thay đổi trạng thái của đơn hàng sang đã huỷ và hiển thị lên màn hình.

Use case kết thúc.

* **Luồng rẽ nhánh:**

1. Tại thời điểm bất kỳ 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**

Cần kiểm soát quyền thực hiện use case này để đảm bảo tính an toàn và bảo mật.

* **Tiền điều kiện**

Người quản trị cần đăng nhập với quyền quản trị trước khi thực hiện use case.

* **Hậu điều kiện**

Sau khi use case kết thúc thành công thì lưu các thông tin vào bảng ORDERS.

* **Điểm mở rộng**

Không có

#### Use case Xem thống kê

* **Mô tả vắn tắt:** Use case này cho phép quản trị viên xem thống kê về sản phẩm đang thịnh hành theo thời gian, theo độ tuổi.
* **Luồng sự kiện**
  + **Luồng cơ bản:**

1. Use case này bắt đầu khi người Quản trị nhấn vào nút “Xem thống kê” trên thanh menu. Hệ thống sẽ hiển thị ra form nhập mức tuổi, khoảng thời gian để lọc ra những sản phẩm đang được quan tâm nhất.

Use case kết thúc.

* **Luồng rẽ nhánh:**

1. Tại thời điểm bất kỳ 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**

Cần kiểm soát quyền thực hiện use case này để đảm bảo tính an toàn và bảo mật.

* **Tiền điều kiện**

Người quản trị cần đăng nhập với quyền quản trị trước khi thực hiện use case.

* **Hậu điều kiện**

Không có

* **Điểm mở rộng**

Không có

#### Use case Quản lý người dùng

* **Mô tả vắn tắt:** Use case này cho phép quản trị viên xem, khóa người dùng.
* **Luồng sự kiện**
  + **Luồng cơ bản:**

1. Use case này bắt đầu khi người dùng nhấn vào nút “Quản lý người dùng” trên thanh menu quản trị. Hệ thống sẽ lấy trong bảng USERS gồm các thông tin: id, số điện thoại, email, họ tên, giới tính, quyền và hiển thị lên màn hình.
2. Khóa người dùng:
   1. Người quản trị nhấn vào nút “Khóa” trong màn hình danh sách người dùng. Hệ thống sẽ khóa thông tin người dùng đó
3. Khóa người dùng:
   1. Người quản trị nhấn vào biểu tượng khóa trên một dòng người dùng. Hệ thống sẽ hiển thị lên popup để xác nhận kh.
   2. Người quản trị nhấn vào nút “Đồng ý”, hệ thống sẽ thực hiện xoá thương hiệu đã chọn khỏi bảng USERS sau đó lấy thông tin danh sách sản phẩm gồm họ tên, email, ngày sinh, số điện thoại, địa chỉ, giới tính và hiển thị lên màn hình.

Use case kết thúc.

* **Luồng rẽ nhánh:**

1. Tại thời điểm bất kỳ 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 1 trong luồng cơ bản, nếu không có bản ghi nào trong bảng USERS thì hệ thống sẽ hiển thị “Không có người dùng” và use case kết thúc.
3. Tại bước 2b và 3b trong luồng cơ bản, khi quản trị viên nhập thông tin người dùng không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi yêu cầu người quản trị nhập lại, quản trị viên có thể tiếp tục nhập lại hoặc nhấn đóng modal để kết thúc.
4. Tại bước 4b trong luồng cơ bản, nếu người quản trị nhấn vào nút không thì hệ thống sẽ bỏ qua thao tác xoá và hiển thị danh sách các sản phẩm trong bảng USERS lên màn hình và use case kết thúc.

* **Các yêu cầu đặc biệt**

Cần kiểm soát quyền thực hiện use case này để đảm bảo tính an toàn và bảo mật.

* **Tiền điều kiện**

Người quản trị cần đăng nhập với quyền quản trị trước khi thực hiện use case.

* **Hậu điều kiện**

Sau khi use case kết thúc thành công thì lưu các thông tin vào bảng USERS

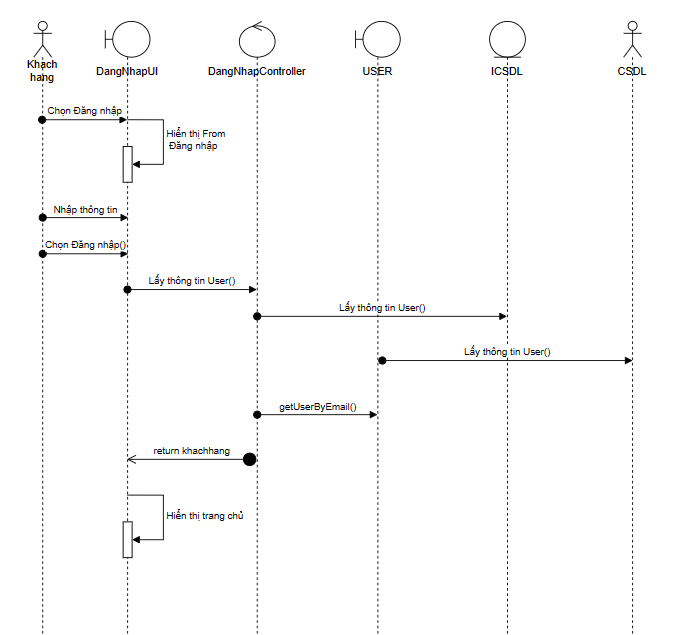
* **Điểm mở rộng**

Không có

### Phân tích các use case

#### Use case Đăng nhập

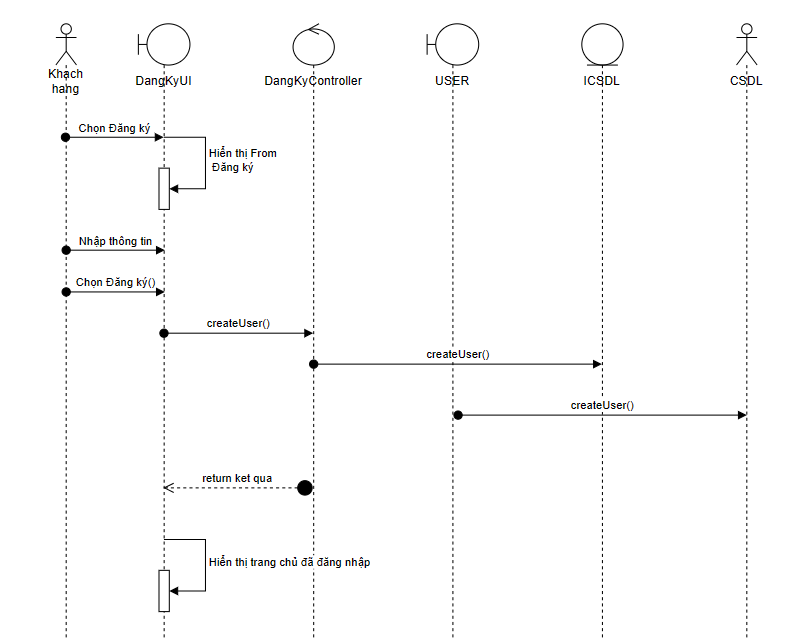
* **Biểu đồ trình tự**



Hình 2.4. Biểu đồ trình tự use case Đăng nhập

#### Use case Đăng ký

* **Biểu đồ trình tự**

****

Hình 2.6. Biểu đồ trình tự use case Quản lý địa chỉ

#### Use case Xem sản phẩm

* **Biểu đồ trình tự**

A black background with white text

Description automatically generated

Hình 2.10. Biểu đồ trình tự use case Xem sản phẩm

#### Use case Tìm kiếm sản phẩm

* **Biểu đồ trình tự**

A black screen with white text

Description automatically generated

Hình 2.12. Biểu đồ trình tự use case Tìm kiếm sản phẩm

#### Use case Thêm vào giỏ hàng

#### Biểu đồ trình tự

A black screen with white text

Description automatically generated

Hình 2.4. Biểu đồ trình tự use case Thêm vào giỏ hàng

#### Use case Đặt hàng

* **Biểu đồ trình tự**

A black background with white text

Description automatically generated

Hình 2.16. Biểu đồ trình tự use case Đặt hàng

#### Use case Tra cứu đơn hàng

* **Biểu đồ trình tự**

A black background with white text

Description automatically generated

Hình 2.18. Biểu đồ trình tự use case Tra cứu đơn hàng

#### Use case Quản lý giỏ hàng

* **Biểu đồ trình tự**

A black background with white text

Description automatically generated

Hình 2.22. Biểu đồ trình tự use case Quản lý giỏ hàng

#### Use case Bảo trì danh mục

* **Biểu đồ trình tự**

A black background with white lines

Description automatically generated

Hình 2.24. Biểu đồ trình tự use case Bảo trì danh mục

#### Use case Bảo trì sản phẩm

* **Biểu đồ trình tự**

A diagram of a product

Description automatically generated

A diagram of a product

Description automatically generated

A diagram with lines and dots

Description automatically generated

A diagram of a product

Description automatically generated

A close-up of a diagram

Description automatically generated

Hình 2.26. Biểu đồ trình tự use case Bảo trì sản phẩm

#### Use case Quản lý đơn hàng

* **Biểu đồ trình tự**

A diagram of a product

Description automatically generated

A white sheet with black lines and black dots

Description automatically generated with medium confidence

A white sheet of paper with a grid

Description automatically generated

Hình 2.28. Biểu đồ trình tự use case Quản lý đơn hàng

#### Use case Xem thống kê

* **Biểu đồ trình tự**

A diagram of a diagram

Description automatically generated

Hình 2.30. Biểu đồ trìnhh tự use case Xem thống kê

#### Use case Quản lý người dùng

* **Biểu đồ trình tự**

A black background with white text

Description automatically generated

Hình 2.32. Biểu đồ trình tự use case Quản lý người dùng

## Xây dựng cơ cở dữ liệu

### Biểu đồ Entity Relationship Diagram

A screenshot of a computer

Description automatically generated

Hình 2.42. Thiết kế cơ sở dữ liệu

### Các bảng trong cơ sở dữ liệu

* **Bảng USERS** (Lưu tài khoản)

A screenshot of a computer

Description automatically generated

Hình 2.43. Bảng users

**Bảng CATEGORIES** (Lưu danh mục sản phẩm)

A screenshot of a computer

Description automatically generated

Hình 2.44. Bảng categories

**Bảng BRANCHES** (Lưu hãng sản phẩm)

A screenshot of a computer

Description automatically generated

Hình 2.45. Bảng branhes

**Bảng ORDERS** (Thông tin giỏ hàng và đơn hàng)

A screenshot of a computer

Description automatically generated

Hình 2.46. Bảng orders

**Bảng ORDER\_DETAILS** (Lưu thông tin chi tiết đơn hàng)

A screenshot of a computer

Description automatically generated

Hình 2.47. Bảng order\_details

**Bảng PRODUCTS** (Lưu thông tin sản phẩm)

A screenshot of a computer

Description automatically generated

Hình 2.48. Bảng products

**Bảng PRODUCT\_IMAGES** (Lưu các hình ảnh của sản phẩm)

A screenshot of a computer

Description automatically generated

Hình 2.49. Bảng product\_images

# KẾT QUẢ ĐẠT ĐƯỢC

## Giao diện chương trình

### Giao diện phía khách hàng

#### Giao diện trang chủ

A screenshot of a smart watch

Description automatically generated

**A screenshot of a smart watch

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

Hình 3.1. Màn hình trang chủ

#### Màn hình xem sản phẩm theo danh mục

A screenshot of a computer

Description automatically generated

Hình 3.2. Màn hình xem sản phẩm theo danh mục

#### Màn hình xem chi tiết sản phẩm

A screenshot of a smart watch

Description automatically generated

Hình 3.3. Màn hình xem chi tiết sản phẩm

#### Màn hình giỏ hàng

A screenshot of a computer

Description automatically generated

3.5. Màn hình giỏ hàng

#### Màn hình thông tin thanh toán

A screenshot of a computer

Description automatically generated

3.6. Màn hình thông tin thanh toán

#### Màn hình thông tin cá nhân

A screenshot of a computer

Description automatically generated

3.7. Màn hình thông tin cá nhân

3.1.1.7. Màn hình đăng ký

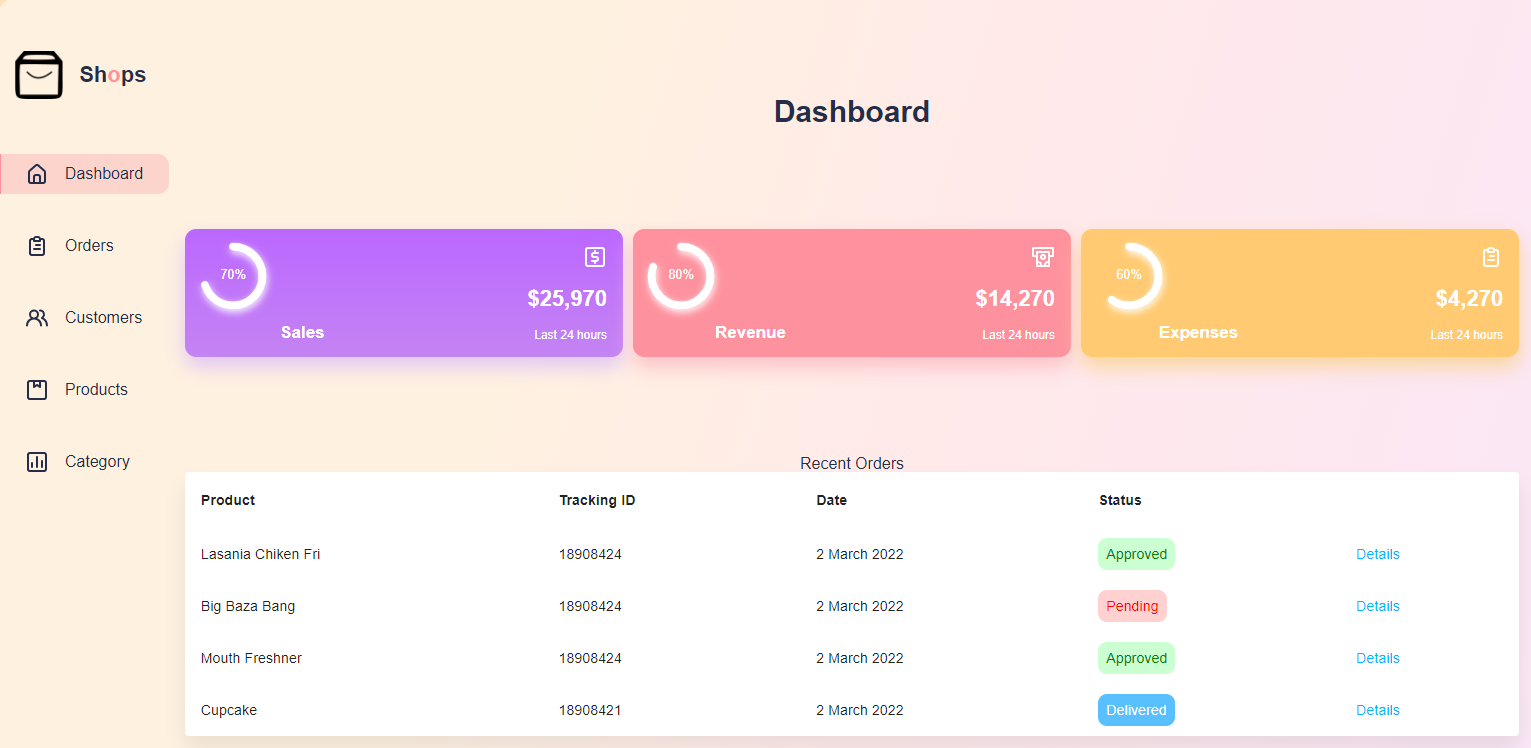
A screenshot of a login form

Description automatically generated

3.8. Màn hình đăng ký

### Giao diện phía quản lí

#### Màn hình dashboard



3.10. Màn hình dashboard

#### Màn hình danh sách khách hàng

A screenshot of a computer

Description automatically generated

3.11. Màn hình danh sách khách hàng

#### Màn hình danh sách sản phẩm

A screenshot of a computer

Description automatically generated

3.12. Màn hình danh sách sản phẩm

#### Màn hình danh sách các danh mục

A screenshot of a computer

Description automatically generated

3.16. Màn hình danh sách các danh mục

#### Màn hình danh sách các hãng

A screenshot of a computer

Description automatically generated

3.17. Màn hình danh sách các hãng

#### Màn hình quản lý đơn hàng

A screenshot of a computer

Description automatically generated

3.18. Màn hình quản lý đơn hàng

# TÀI LIỆU THAM KHẢO

[1] Giáo trình môn học Phân tích thiết kế hệ thống.

[2] Giáo trình môn học Thiết kế cơ sở dữ liệu.

[3] <https://www.postgresql.org/docs> : tài liệu hướng dẫn PostgreSQL.

[4] <https://typeorm.io> : tài liệu hướng dẫn thư viện TypeORM.

[5] <https://docs.nestjs.com> : tài liệu framework NestJS.

[6] [https://react.dev/](https://react.dev/%20) : tài liệu tham khảo ReactJS.