|  |  |
| --- | --- |
|  |  |
| Phạm Phúc Lâm | **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  --------------------------------------- |
|  |
| ĐỒ ÁN TỐT NGHIỆP |
| NGÀNH HỆ THỐNG THÔNG TIN |
| **XÂY DỰNG WEBSITE BÁN LAPTOP, LINH KIỆN VÀ PHỤ KIỆN MÁY TÍNH CHO CỬA HÀNG MINH QUÂN SỬ DỤNG PHP VÀ JAVASCRIPT** |
|  |
|  |
| **CBHD: TS. Nguyễn Văn Tỉnh** |
| HỆ THỐNG THÔNG TIN | **Sinh viên: Phạm Phúc Lâm** |
| **Mã số sinh viên: 2019600262** |
|  |
|  |
|  |
| Hà Nội – Năm 2023 |

# LỜI MỞ ĐẦU

**1. Lý do chọn đề tài**

Đề tài “Xây dựng website bán laptop, linh kiện và phụ kiện máy tính cho cửa hàng Minh Quân sử dụng PHP và JavaScript” được chọn cho đồ án tốt nghiệp với mục tiêu phát triển một nền tảng mua sắm trực tuyến chuyên nghiệp và tiện lợi cho Cửa hàng Minh Quân. Trong bối cảnh sự gia tăng đáng kể của thương mại điện tử và xu hướng mua sắm trực tuyến, việc xây dựng một website bán hàng hiệu quả là cần thiết để cửa hàng có thể cạnh tranh và tiếp cận được đối tượng khách hàng rộng lớn. Đồ án tốt nghiệp này cũng mang ý nghĩa thực tiễn lớn đối với quá trình học tập và trải nghiệm thực tế của em. Việc nghiên cứu và áp dụng các ngôn ngữ lập trình như PHP và JavaScript giúp em nắm vững kỹ năng lập trình và hiểu rõ cách thức xây dựng một trang web thương mại điện tử từ đầu đến cuối.

Ngoài ra, đề tài này còn giúp Cửa hàng Minh Quân nâng cao hiệu quả kinh doanh và tăng cường mối quan hệ với khách hàng. Trang web bán hàng chất lượng sẽ tạo điều kiện thuận lợi cho khách hàng tiếp cận thông tin sản phẩm, đặt hàng và thanh toán dễ dàng. Không chỉ giúp cửa hàng thu hút được khách hàng mới, mà còn tạo sự lòng tin và đáng tin cậy trong lòng khách hàng hiện tại. Đồ án tốt nghiệp này không chỉ đóng góp vào sự phát triển của Cửa hàng Minh Quân mà còn mang ý nghĩa quan trọng đối với ngành kinh doanh thương mại điện tử trong thời đại công nghệ số. Trong bối cảnh thương mại điện tử đang ngày càng trở nên quan trọng và thay đổi cách mà người tiêu dùng tiếp cận và mua hàng, việc xây dựng một nền tảng mua sắm trực tuyến hiệu quả là vô cùng cần thiết và có ý nghĩa lớn trong việc định hình và phát triển ngành bán lẻ trong tương lai.

**2. Mục đích của đề tài**

Mục đích chính của đề tài là phát triển một website bán hàng tiện lợi, trực quan và chuyên nghiệp cho Cửa hàng Minh Quân. Em mong muốn tạo ra một nền tảng trực tuyến cho khách hàng để dễ dàng tìm kiếm và mua sắm các sản phẩm laptop, linh kiện và phụ kiện máy tính với sự hỗ trợ đáng tin cậy của PHP và JavaScript.

**3. Đối tượng và phạm vi nghiên cứu**

Đối tượng nghiên cứu của đề tài là Cửa hàng Minh Quân và khách hàng của họ. Em sẽ tập trung vào xây dựng một giao diện dễ sử dụng, tương tác tốt và hiệu quả cho việc mua sắm trực tuyến.

Phạm vi nghiên cứu của đề tài bao gồm phân tích yêu cầu, thiết kế giao diện và cơ sở dữ liệu, cùng với triển khai các chức năng chính của một website bán hàng online.

**4. Ý nghĩa khoa học và thực tiễn của đề tài**

* **Ý nghĩa khoa học**

1. Nghiên cứu đề tài này sẽ cung cấp kiến thức và kỹ năng trong lĩnh vực phát triển website và ứng dụng web, đặc biệt là sử dụng ngôn ngữ lập trình PHP và JavaScript. Điều này sẽ giúp em nắm vững các kỹ thuật cần thiết để xây dựng một website bán hàng online.
2. Phân tích yêu cầu và thiết kế giao diện trong đề tài sẽ cung cấp kinh nghiệm về quá trình lập kế hoạch và thiết kế sản phẩm công nghệ thông tin, từ đó phát triển khả năng tư duy phân tích và sáng tạo trong việc xây dựng các giải pháp trực tuyến.
3. Triển khai chức năng chính và tích hợp các công nghệ trong website giúp sinh viên rèn luyện kỹ năng thực hành và ứng dụng kiến thức lý thuyết vào việc thực tế, từ đó nâng cao kỹ năng phát triển và quản lý dự án công nghệ.

* **Ý nghĩa thực tiễn**

1. Website bán hàng online cho cửa hàng Minh Quân giúp nâng cao trải nghiệm mua sắm của khách hàng. Khách hàng có thể dễ dàng tìm kiếm, xem và lựa chọn các sản phẩm laptop, linh kiện và phụ kiện máy tính từ cửa hàng, đồng thời tiến hành thanh toán an toàn và thuận tiện.
2. Việc xây dựng website bán hàng online giúp cửa hàng Minh Quân mở rộng phạm vi kinh doanh và tiếp cận được đến một lượng khách hàng rộng hơn, đặc biệt là trong thời đại số hóa ngày càng phát triển.
3. Thông qua việc triển khai chức năng quản lý quyền truy cập, chủ cửa hàng có thể quản lý dễ dàng các tài khoản người dùng và phân quyền truy cập vào hệ thống, tăng tính bảo mật và quản lý hiệu quả.

# CHƯƠNG 1: GIỚI THIỆU VỀ CÔNG CỤ VÀ NGÔN NGỮ SỬ DỤNG

## Công cụ sử dụng

- phpMyAdmin: công cụ quản lý cơ sở dữ liệu

- CASE Studio: công cụ thiết kế mô hình thực thể liên kết

- Rational Rose: công cụ thiết kế biểu đồ use case

- Visual Studio Code: công cụ sử dụng để chỉnh sửa mã nguồn của website

### 1.1.1 phpMyAdmin

PHPMyAdmin là một công cụ quản lý cơ sở dữ liệu mã nguồn mở được viết bằng ngôn ngữ lập trình PHP. Nó cung cấp giao diện đồ họa thân thiện và dễ sử dụng, cho phép người dùng quản lý cơ sở dữ liệu MySQL một cách hiệu quả. PHPMyAdmin thường được cài đặt trên máy chủ web và truy cập thông qua trình duyệt web, giúp quản trị viên và nhà phát triển dễ dàng thực hiện các tác vụ quản lý cơ sở dữ liệu trực quan và linh hoạt.

Một số tính năng chính của PHPMyAdmin bao gồm:

1. **Quản lý cơ sở dữ liệu**: Cho phép tạo, xóa và sửa đổi các cơ sở dữ liệu, bảng, trường và chỉ mục
2. **Truy vấn SQL**: Hỗ trợ viết và thực thi các truy vấn SQL để thực hiện các tác vụ truy xuất và cập nhật dữ liệu
3. **Nhập và xuất dữ liệu**: Cho phép nhập và xuất dữ liệu từ và vào các bảng thông qua các tập tin CSV, SQL hoặc các định dạng dữ liệu khác.
4. **Tạo báo cáo**: Cung cấp khả năng tạo báo cáo và biểu đồ từ cơ sở dữ liệu.
5. **Phân quyền**: Hỗ trợ quản lý người dùng và phân quyền truy cập vào cơ sở dữ liệu, đảm bảo an toàn và bảo mật dữ liệu.
6. **Xem cấu trúc**: Hiển thị cấu trúc chi tiết của cơ sở dữ liệu, bảng và trường, giúp người dùng hiểu rõ hơn về cấu trúc dữ liệu của hệ thống.

PHPMyAdmin là một công cụ rất hữu ích cho việc quản lý cơ sở dữ liệu MySQL một cách tiện lợi và nhanh chóng. Nó được sử dụng phổ biến trong các dự án phát triển web, giúp quản lý dữ liệu một cách dễ dàng và chuyên nghiệp.

### 1.1.2 CASE Studio

CASE Studio là một công cụ thiết kế cơ sở dữ liệu chuyên nghiệp, được phát triển bởi công ty CHARONWARE s.r.o. Nó cung cấp môi trường thiết kế hướng đối tượng (OOAD) cho phép người dùng xây dựng các mô hình cơ sở dữ liệu và hệ thống phức tạp một cách chuyên nghiệp và hiệu quả. CASE Studio hỗ trợ nhiều định dạng mô hình hóa, bao gồm các biểu đồ UML (Unified Modeling Language), sơ đồ lớp, sơ đồ thực thể-liên kết và nhiều loại biểu đồ khác.

Một số tính năng chính của CASE Studio bao gồm:

1. **Thiết kế cơ sở dữ liệu**: CASE Studio cho phép người dùng thiết kế các bảng dữ liệu, quan hệ giữa các bảng, các khóa chính và các quan hệ ràng buộc trong cơ sở dữ liệu. Điều này giúp xác định cấu trúc cơ sở dữ liệu một cách chính xác và rõ ràng.
2. **Mô hình hóa hướng đối tượng**: CASE Studio hỗ trợ các biểu đồ UML như biểu đồ lớp, biểu đồ Use Case, biểu đồ tuần tự và biểu đồ trạng thái. Điều này cho phép người dùng mô hình hóa các đối tượng, các tương tác và luồng công việc trong hệ thống phần mềm.
3. **Quản lý mô hình**: Các mô hình trong CASE Studio có thể được quản lý và tổ chức một cách dễ dàng, cho phép người dùng duyệt và sắp xếp các mô hình theo nhu cầu của dự án.
4. **Tích hợp công cụ**: CASE Studio cho phép tích hợp với các công cụ phát triển phần mềm khác như Visual Studio, Eclipse và các hệ thống quản lý cơ sở dữ liệu như MySQL, PostgreSQL, SQL Server.

CASE Studio là công cụ hữu ích cho các nhà phát triển phần mềm và nhà quản lý dự án trong quá trình thiết kế và quản lý cơ sở dữ liệu. Với giao diện thân thiện và các tính năng tiên tiến, nó giúp tăng hiệu suất và chất lượng của quá trình phát triển phần mềm và xây dựng hệ thống phức tạp.

### 1.1.3 Rational Rose

Rational Rose là một phần mềm CASE (Computer-Aided Software Engineering) do hãng Rational Software Corporation (hiện nay thuộc IBM) phát triển. Rational Rose được sử dụng để hỗ trợ trong quá trình phát triển và mô hình hóa phần mềm. Được ra mắt vào những năm 1990, Rational Rose nhanh chóng trở thành một trong những công cụ phổ biến và mạnh mẽ trong lĩnh vực phát triển phần mềm.

Các tính năng và chức năng chính của Rational Rose bao gồm:

1. **Mô hình UML**: Rational Rose hỗ trợ xây dựng các biểu đồ UML (Unified Modeling Language) như biểu đồ lớp, biểu đồ tuần tự, biểu đồ hoạt động, biểu đồ trạng thái, biểu đồ use case,... giúp người dùng mô hình hóa các khái niệm, quy trình và tương tác giữa các thành phần trong phần mềm.
2. **Tích hợp và tái sử dụng**: Rational Rose cho phép người dùng kết nối và tái sử dụng các thành phần mô hình đã xây dựng. Điều này giúp tạo ra các mô hình phần mềm có tính tái sử dụng cao và giảm thiểu sự trùng lắp trong phát triển.
3. **Hỗ trợ ngôn ngữ lập trình**: Rational Rose hỗ trợ nhiều ngôn ngữ lập trình như C++, Java, C#, VB.NET,... giúp người dùng xây dựng các ứng dụng đa nền tảng và tích hợp các thành phần phần mềm.
4. **Hỗ trợ tiêu chuẩn công nghiệp**: Rational Rose tuân thủ các tiêu chuẩn UML và phù hợp với các tiêu chuẩn phát triển phần mềm chuyên nghiệp, giúp tăng tính chính xác và hiệu quả trong quá trình phát triển.
5. **Quản lý phiên bản**: Rational Rose hỗ trợ quản lý phiên bản và đồng bộ hóa mô hình giữa nhiều thành viên trong nhóm phát triển.

Nhờ vào những tính năng nổi bật và sự mạnh mẽ trong việc mô hình hóa phần mềm, Rational Rose đã trở thành một công cụ phát triển phần mềm được ưa chuộng trong nhiều dự án lớn và phức tạp. Nó giúp tăng năng suất và chất lượng trong quá trình phát triển và cải thiện hiệu quả của quy trình phát triển phần mềm.

### 1.1.4 Visual Studio Code

Visual Studio Code (thường được gọi tắt là VS Code) là một môi trường phát triển tích hợp (IDE) nổi tiếng và miễn phí, được phát triển bởi Microsoft. Được ra mắt lần đầu vào năm 2015, VS Code đã nhanh chóng trở thành một trong những công cụ phát triển phần mềm phổ biến nhất và được ưa chuộng rộng rãi trong cộng đồng lập trình.

Với thiết kế tối giản nhưng mạnh mẽ, Visual Studio Code cung cấp môi trường làm việc hiệu quả cho các nhà phát triển ở mọi cấp độ. Một số điểm nổi bật của VS Code bao gồm:

1. **Mã nguồn mở và miễn phí**: Visual Studio Code là một phần mềm mã nguồn mở, cho phép mọi người dùng tải xuống, sử dụng và thậm chí đóng góp vào quá trình phát triển.
2. **Hỗ trợ đa ngôn ngữ**: VS Code hỗ trợ nhiều ngôn ngữ lập trình phổ biến, từ những ngôn ngữ cơ bản như C++, Java, Python đến các ngôn ngữ web như HTML, CSS, và JavaScript. Thậm chí, cộng đồng lập trình đã tạo ra các phần mở rộng để hỗ trợ thêm nhiều ngôn ngữ khác nhau.
3. **Tiện ích mở rộng mạnh mẽ**: Một điểm độc đáo của VS Code là khả năng mở rộng thông qua các tiện ích. Các tiện ích mở rộng cho phép bạn tùy chỉnh và mở rộng khả năng của VS Code theo nhu cầu cụ thể của dự án và ngôn ngữ lập trình.
4. **Giao diện người dùng thân thiện**: Giao diện của VS Code được thiết kế để đơn giản và dễ sử dụng. Nó cung cấp các tính năng cần thiết để tối ưu hóa quá trình phát triển mà không gây ra sự phức tạp không cần thiết.
5. **Hỗ trợ Git tích hợp**: VS Code tích hợp sẵn tích hợp quản lý phiên bản Git, cho phép bạn quản lý mã nguồn, thực hiện commit, và theo dõi sự thay đổi trong dự án của mình một cách dễ dàng.
6. **Gỡ lỗi và theo dõi mã nguồn**: VS Code cung cấp khả năng gỡ lỗi mã nguồn dễ dàng thông qua giao diện trực quan. Bạn có thể theo dõi giá trị biến, thực hiện bước từng bước và xem xét cơ cấu mã nguồn để tìm ra và sửa các lỗi.

Với những đặc điểm và tính năng đa dạng, Visual Studio Code đã trở thành một công cụ hữu ích không chỉ cho việc chỉnh sửa mã nguồn mà còn cho quá trình phát triển phần mềm toàn diện.

## 1.2 Ngôn ngữ sử dụng

### 1.2.1 PHP

#### 1.2.1.1 Giới thiệu về PHP

PHP (viết tắt của "PHP: Hypertext Preprocessor") là một ngôn ngữ lập trình phổ biến được sử dụng chủ yếu để phát triển các ứng dụng web động và tương tác trên trang web. Được phát triển vào những năm 1990, PHP đã nhanh chóng trở thành một trong những ngôn ngữ lập trình web phổ biến nhất trên thế giới.

Những điểm nổi bật của PHP bao gồm:

1. **Mã nguồn mở**: PHP là một ngôn ngữ mã nguồn mở, có nghĩa là mã nguồn của nó có thể được truy cập, sửa đổi và phân phối miễn phí. Điều này đã giúp tạo nên một cộng đồng lập trình đông đảo và tích cực, đóng góp vào sự phát triển và cải thiện của ngôn ngữ.
2. **Dễ học và sử dụng**: Với cú pháp tương tự như C và C++, PHP dễ học và sử dụng đối với người mới bắt đầu. Nó cũng hỗ trợ nhiều tính năng, cho phép lập trình viên xây dựng các ứng dụng web phức tạp và đa dạng.
3. **Hỗ trợ đa nền tảng**: PHP có thể chạy trên nhiều hệ điều hành phổ biến như Windows, macOS và Linux, làm cho việc triển khai ứng dụng web dễ dàng và linh hoạt.
4. **Tích hợp dễ dàng**: PHP hỗ trợ tích hợp tốt với nhiều cơ sở dữ liệu phổ biến như MySQL, PostgreSQL và Oracle, giúp quản lý và truy vấn dữ liệu trên trang web một cách dễ dàng.
5. **Xử lý biểu mẫu và tương tác với người dùng**: PHP cho phép xử lý dữ liệu biểu mẫu nhập vào từ người dùng, tạo ra nội dung động trên trang web và tương tác với người dùng thông qua các biểu mẫu và yêu cầu HTTP.
6. **Hỗ trợ mạnh mẽ cho mã hóa và bảo mật**: PHP cung cấp các hàm mã hóa mạnh mẽ và tích hợp tính năng bảo mật để giúp ngăn chặn các cuộc tấn công từ hacker và đảm bảo an toàn cho ứng dụng web.

Nhờ những ưu điểm nổi trội này, PHP đã trở thành một trong những công cụ ưa thích của các nhà phát triển web để xây dựng các ứng dụng web động và tương tác, từ các trang web cá nhân đơn giản đến các hệ thống quản lý nội dung phức tạp và các ứng dụng web thương mại điện tử.

#### 1.2.1.2 Cú pháp và cấu trúc PHP

Dưới đây là mô tả về cú pháp và cấu trúc cơ bản của PHP:

**Mở và đóng thẻ PHP**

PHP được bắt đầu bằng thẻ mở "<?php" và kết thúc bằng thẻ đóng "?>". Mã PHP nằm giữa hai thẻ này và sẽ được xử lý bởi máy chủ web trước khi gửi dữ liệu đến trình duyệt.

**Kết hợp PHP với HTML**

PHP có thể được kết hợp với mã HTML để tạo nội dung động trên trang web. Để chuyển từ chế độ mã PHP sang HTML, bạn không cần thêm thẻ đóng PHP và có thể viết mã HTML trực tiếp.

**Biến**

Biến trong PHP được bắt đầu bằng ký tự "$" theo sau là tên biến. Biến trong PHP không cần khai báo kiểu dữ liệu và có thể thay đổi giá trị trong quá trình thực thi chương trình.

**Comment**

PHP hỗ trợ hai kiểu comment:

* Comment trên một dòng: Sử dụng "//" để comment một dòng.
* Comment trên nhiều dòng: Sử dụng "/" để bắt đầu và "/" để kết thúc comment trên nhiều dòng.

**Câu lệnh điều kiện**

PHP hỗ trợ các câu lệnh điều kiện như if, else và elseif để thực hiện các hành động khác nhau dựa trên điều kiện đúng hay sai.

**Vòng lặp**

PHP hỗ trợ các loại vòng lặp như for, while và foreach để thực hiện một hành động lặp lại nhiều lần.

**Hàm**

Hàm trong PHP cho phép bạn viết một đoạn mã và tái sử dụng nó nhiều lần trong chương trình. Hàm có thể nhận đối số và trả về giá trị.

**Các thư viện và hàm có sẵn**

PHP đi kèm với nhiều thư viện và hàm có sẵn cho việc thực hiện các tác vụ phổ biến như xử lý chuỗi, kết nối cơ sở dữ liệu, gửi email và xử lý hình ảnh.

**Biểu thức**

PHP hỗ trợ các biểu thức số học, logic và chuỗi, cho phép bạn thực hiện các phép tính và so sánh dữ liệu.

#### 1.2.1.3 Ứng dụng của PHP

Dưới đây là một số ứng dụng phổ biến của PHP:

1. **Website động**: PHP được sử dụng để xây dựng các trang web động, cho phép tương tác với người dùng và cung cấp nội dung động dựa trên các tác động của người dùng.
2. **Hệ thống quản lý nội dung (CMS)**: PHP thường được sử dụng để phát triển các CMS như WordPress, Joomla và Drupal, giúp quản lý và cập nhật nội dung trên trang web một cách dễ dàng.
3. **Cửa hàng trực tuyến và thương mại điện tử**: PHP được sử dụng để xây dựng các cửa hàng trực tuyến và các hệ thống thương mại điện tử, cho phép người dùng mua hàng và thực hiện các giao dịch trực tuyến.
4. **Hệ thống đăng nhập và xác thực**: PHP được sử dụng để tạo các hệ thống đăng nhập và xác thực, đảm bảo tính bảo mật và quyền riêng tư cho người dùng.
5. **Diễn đàn và cộng đồng trực tuyến**: PHP được sử dụng để xây dựng các diễn đàn và cộng đồng trực tuyến, nơi người dùng có thể thảo luận và chia sẻ thông tin.
6. **Hệ thống quản lý dự án và công việc**: PHP được sử dụng để xây dựng các hệ thống quản lý dự án và công việc, giúp tổ chức công việc và theo dõi tiến độ dự án.
7. **Blog cá nhân và trang cá nhân**: PHP được sử dụng để xây dựng các blog cá nhân và trang web cá nhân, cho phép người dùng chia sẻ thông tin và ý kiến của họ với cộng đồng.
8. **Ứng dụng máy chủ**: PHP có thể được sử dụng để viết các ứng dụng máy chủ, như hệ thống gửi email, quản lý tệp tin, và xử lý dữ liệu trên máy chủ.
9. **Game trực tuyến**: PHP có thể được sử dụng để xây dựng các game đơn giản trực tuyến, cho phép người dùng chơi và tương tác với nhau qua mạng.

### 1.2.2 JavaScript

#### 1.2.2.1 Giới thiệu về JavaScript

JavaScript là một ngôn ngữ lập trình phổ biến được sử dụng phía client (trình duyệt) để xây dựng các ứng dụng web động và tương tác. Được phát triển ban đầu bởi Brendan Eich vào năm 1995, JavaScript đã trở thành một trong những ngôn ngữ lập trình phổ biến và quan trọng nhất trên thế giới.

Dưới đây là một số điểm nổi bật về JavaScript:

1. **Ngôn ngữ lập trình đa nền tảng:** JavaScript là một ngôn ngữ đa nền tảng, được hỗ trợ bởi hầu hết các trình duyệt web phổ biến như Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, và nhiều trình duyệt khác. Điều này cho phép các ứng dụng JavaScript hoạt động trên nhiều hệ điều hành và thiết bị khác nhau.
2. **Xử lý phía client**: JavaScript thường được sử dụng để tương tác trực tiếp với trình duyệt và xử lý phía client, cho phép các ứng dụng web động và linh hoạt. Các tác vụ như thay đổi nội dung của trang web, xử lý sự kiện từ người dùng, và gửi dữ liệu giữa trình duyệt và máy chủ được thực hiện bằng JavaScript.
3. **Cú pháp đơn giản**: JavaScript có cú pháp dễ hiểu và dễ học, cho phép người lập trình tạo ra mã dễ đọc và dễ bảo trì. Nó cũng hỗ trợ nhiều kiểu dữ liệu và phép tính phổ biến.
4. **DOM (Document Object Model):** JavaScript là ngôn ngữ phía client duy nhất có thể truy cập và thay đổi cấu trúc của trang web thông qua DOM. DOM là mô hình biểu diễn trang web dưới dạng cây các đối tượng, cho phép JavaScript tương tác với các phần tử HTML và CSS của trang web.
5. **AJAX (Asynchronous JavaScript and XML):** AJAX cho phép trao đổi dữ liệu giữa trình duyệt và máy chủ mà không cần tải lại trang hoàn toàn, giúp tăng trải nghiệm người dùng và làm cho các ứng dụng trở nên nhanh hơn và hiệu quả hơn.
6. **Thư viện và framework phong phú**: JavaScript có nhiều thư viện và framework mạnh mẽ như jQuery, React, Angular, Vue.js, và nhiều công cụ hỗ trợ khác giúp phát triển ứng dụng web nhanh chóng và dễ dàng.

JavaScript đã trở thành một trong những ngôn ngữ lập trình quan trọng nhất cho phát triển ứng dụng web. Nó đóng vai trò cực kỳ quan trọng trong việc tạo ra trải nghiệm tốt hơn cho người dùng và làm cho web trở nên đa dạng và phong phú hơn.

#### 1.2.2.2 Cú pháp của JavaScript

Cú pháp của JavaScript là nền tảng để viết mã lệnh trong ngôn ngữ này. Dưới đây là một số cú pháp cơ bản của JavaScript:

1. **Khai báo biến:**

* Sử dụng từ khóa "var", "let", hoặc "const" để khai báo biến.
* Ví dụ:

A black background with white text

Description automatically generated

1. **Các loại dữ liệu:**

* JavaScript hỗ trợ các loại dữ liệu như số, chuỗi, boolean, mảng, đối tượng, v.v.
* Ví dụ:

A black screen with white text

Description automatically generated

1. **Các toán tử:**

* JavaScript hỗ trợ các toán tử phổ biến như +, -, \*, /, %, ++, --, v.v.
* Ví dụ:

A black background with white text

Description automatically generated

#### 1.2.2.3 Ứng dụng của JavaScript

JavaScript có rất nhiều ứng dụng trong lĩnh vực phát triển web và nhiều lĩnh vực khác. Dưới đây là một số ứng dụng phổ biến của JavaScript:

1. **Phát triển các ứng dụng web động**: JavaScript được sử dụng chủ yếu để xây dựng các ứng dụng web động, cho phép tương tác với người dùng và cung cấp trải nghiệm tốt hơn trên trình duyệt.
2. **Tương tác với DOM**: JavaScript cho phép tương tác và thay đổi nội dung của trang web thông qua DOM (Document Object Model), giúp cải thiện tính tương tác và đáp ứng của trang.
3. **AJAX và giao tiếp với máy chủ**: JavaScript sử dụng AJAX để gửi và nhận dữ liệu từ máy chủ mà không cần tải lại trang, làm cho ứng dụng web trở nên mượt mà và hiệu quả hơn.
4. **Xây dựng ứng dụng di động và desktop**: JavaScript có thể được sử dụng để xây dựng ứng dụng di động và desktop bằng các framework như React Native và Electron.
5. **Tạo trò chơi trực tuyến**: JavaScript có thể được sử dụng để xây dựng các trò chơi trực tuyến đơn giản, đáp ứng và tương tác.
6. **Xử lý biểu mẫu**: JavaScript hỗ trợ xử lý biểu mẫu trong các trang web, giúp kiểm tra và xác thực dữ liệu nhập từ người dùng.
7. **Hỗ trợ trong phát triển game**: JavaScript được sử dụng trong phát triển game trực tuyến và trò chơi trình duyệt như HTML5 game.
8. **Tích hợp với các công nghệ khác**: JavaScript có thể tích hợp và hoạt động cùng với nhiều công nghệ khác như HTML, CSS, Node.js, và nhiều hệ thống quản lý dự án (CMS) phổ biến.

# CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## 2.1 Đặc tả bài toán

Đề tài này nhằm xây dựng một trang web bán laptop, linh kiện và phụ kiện máy tính trực tuyến cho Cửa hàng Minh Quân nhằm đáp ứng nhu cầu mua sắm của khách hàng một cách thuận tiện và nhanh chóng. Dưới đây là đặc tả bài toán của đề tài:

1. Trang web sẽ cung cấp một giao diện thân thiện và dễ sử dụng để khách hàng có thể dễ dàng tìm kiếm, xem thông tin chi tiết và giá cả của các sản phẩm laptop, linh kiện và phụ kiện máy tính.
2. Khách hàng có thể thêm các sản phẩm mà họ muốn mua vào giỏ hàng và tiến hành thanh toán khi hoàn tất việc mua sắm.
3. Để mua hàng và sử dụng các chức năng của trang web, khách hàng cần đăng nhập vào hệ thống. Nếu là khách hàng mới, họ cần đăng ký tài khoản và cung cấp đầy đủ thông tin cá nhân mà website yêu cầu.
4. Sau khi khách hàng thực hiện đặt hàng, hệ thống sẽ xác nhận thông tin đặt hàng và gửi thông báo đến khách hàng để xác nhận việc mua hàng thành công.
5. Trang web sẽ được quản lý bởi người quản trị, người có quyền thêm, sửa, xóa thông tin sản phẩm và quản lý thông tin khách hàng.
6. Để đảm bảo tính bảo mật và an toàn thông tin của khách hàng, dữ liệu cá nhân và giao dịch sẽ được mã hóa và bảo vệ.

## 2.2. Phân tích yêu cầu của website

### 2.2.1 Yêu cầu chức năng

Các yêu cầu chức năng của website bao gồm:

1. **Module tìm kiếm**:

* Cho phép người dùng tìm kiếm các sản phẩm (laptop, linh kiện, phụ kiện) trên website dựa trên các tiêu chí như tên sản phẩm, hãng sản xuất, loại sản phẩm, v.v.

1. **Module xem sản phẩm và chi tiết sản phẩm**:

* Hiển thị danh sách các sản phẩm và các thông tin liên quan như hình ảnh, giá cả, mô tả, thông số kỹ thuật, v.v.
* Cho phép người dùng xem chi tiết từng sản phẩm để biết thêm thông tin cụ thể.

3. **Module giỏ hàng và thanh toán**:

* Cho phép người dùng thêm sản phẩm vào giỏ hàng và quản lý số lượng sản phẩm trong giỏ hàng.
* Cung cấp tính năng tính tổng số tiền trong giỏ hàng và tính phí vận chuyển (nếu có).

4. **Module quản lý tài khoản người dùng**:

* Cho phép người dùng tạo tài khoản và đăng nhập để theo dõi đơn hàng, lịch sử mua hàng, và cập nhật thông tin cá nhân.
* Cung cấp tính năng quên mật khẩu và thay đổi mật khẩu để đảm bảo an toàn cho tài khoản người dùng.

5. **Module quản lý đơn hàng**:

* Cho phép nhà quản trị xem và quản lý các đơn hàng của khách hàng, bao gồm xác nhận, hủy và giao hàng.
* Cung cấp trạng thái đơn hàng để người dùng theo dõi tình trạng giao hàng của đơn hàng của mình.

6. **Module quản lý kho hàng**:

* Cho phép quản trị viên quản lý số lượng tồn kho của các sản phẩm.
* Cung cấp cơ chế tự động thông báo khi số lượng tồn kho dưới mức tối thiểu.

7. **Module quản lý giảm giá**:

* Hỗ trợ chương trình khuyến mãi, giảm giá, mã giảm giá và quản lý các chương trình khuyến mãi hiện tại

8. **Module thống kê và báo cáo**:

* Cung cấp các báo cáo thống kê về doanh số bán hàng, doanh thu, số lượng tồn kho, hoạt động của khách hàng, v.v.

### 2.2.2 Yêu cầu phi chức năng

Các yêu cầu chức năng của website bao gồm:

1. **Bảo mật và An toàn**: Website cần đảm bảo bảo mật thông tin cá nhân của người dùng và các giao dịch thanh toán.

2. **Hiệu suất và Tính sẵn sàng cao**: Website cần được tối ưu hóa để có thời gian tải trang nhanh chóng và đáng tin cậy. Điều này giúp cải thiện trải nghiệm người dùng và giảm tỷ lệ thoát khỏi trang web.

3. **Tương thích và Đáp ứng**: Website cần phải tương thích với các trình duyệt web phổ biến và các thiết bị di động khác nhau. Thiết kế đáp ứng (responsive design) giúp trang web tự điều chỉnh kích thước và bố cục để phù hợp với mọi loại màn hình.

4. **Dễ sử dụng và Thân thiện với người dùng**: Giao diện của trang web cần được thiết kế dễ sử dụng và thân thiện với người dùng. Tất cả các chức năng nên được sắp xếp logic và dễ dàng truy cập.

5. **Độ tin cậy và Sao lưu dữ liệu**: Website cần có khả năng tự động sao lưu dữ liệu thường xuyên để đảm bảo an toàn và khôi phục dữ liệu trong trường hợp có sự cố xảy ra.

6. **Phù hợp với Luật và Quy định**: Website cần tuân thủ các luật pháp và quy định về giao dịch thương mại điện tử, bảo vệ quyền riêng tư và bảo vệ người tiêu dùng.

7. **Hỗ trợ và Tích hợp với dịch vụ khác**: Website cần cung cấp hỗ trợ cho khách hàng qua các kênh liên lạc như điện thoại, v.v. . Website có tích hợp với dịch vụ hệ thống thanh toán trực tuyến và dịch vụ giao hàng.

8. **Tùy chỉnh và Mở rộng**: Website cần được thiết kế để dễ dàng tùy chỉnh và mở rộng trong tương lai để đáp ứng các yêu cầu và nhu cầu mới.

Các yêu cầu phi chức năng này đảm bảo rằng trang web hoạt động hiệu quả, cung cấp trải nghiệm tốt cho người dùng và đáp ứng được các tiêu chuẩn an toàn và chất lượng trong phát triển và vận hành.

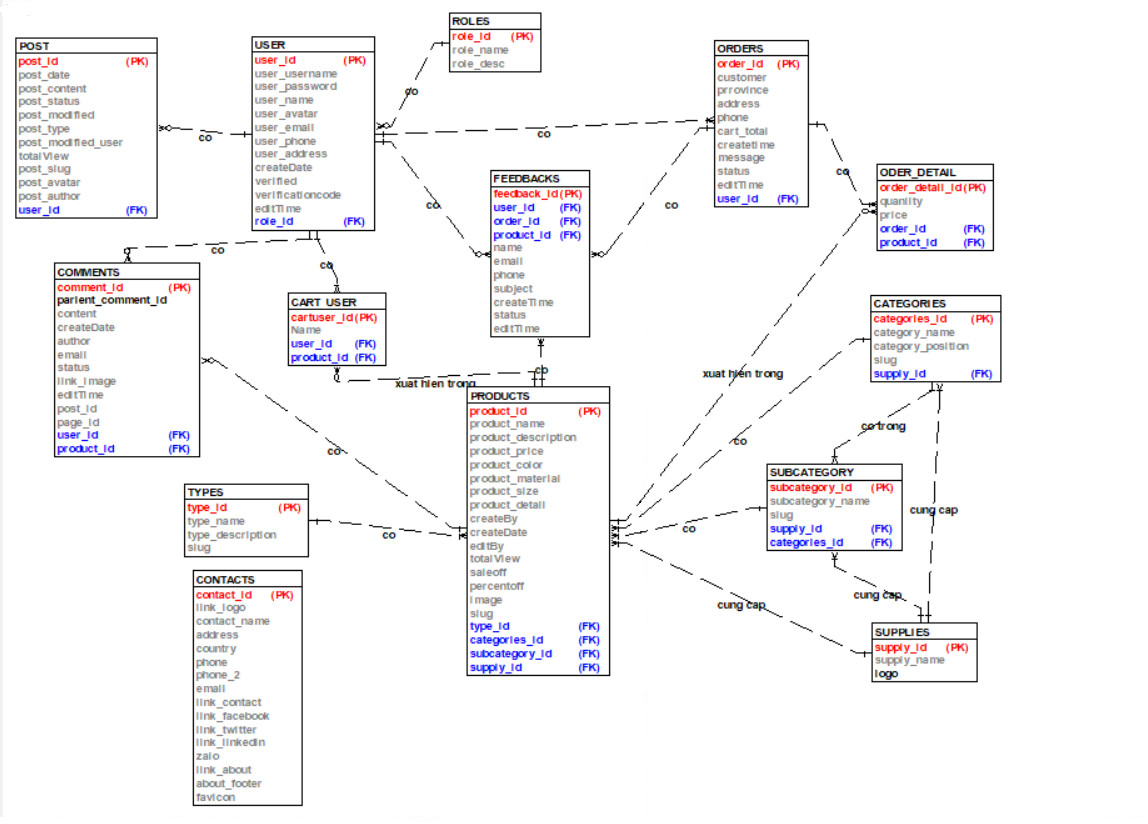
## 2.3 Thiết kế, xây dựng website hoàn chỉnh

### 2.3.1 Xây dựng cơ sở dữ liệu

#### 2.3.1.1 Xác định thuộc tính

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1. USERS(id, user\_username, user\_password, user\_nam, user\_avatar, user\_email, user\_phone, user\_address, createDate, verified, verificationCode, editTime, role\_id) | |  |  | | --- | --- | | 1. USERS | | | id  user\_username  user\_password  user\_name  user\_avatar  user\_email  user\_phone  user\_address  createDate  verified  verificationCode  editTime  role\_id | int  varchar(50)  varchar(100)  varchar(255)  varchar(550)  varchar(255)  varchar(20)  varchar(200)  datetime  int  varchar(500)  datetime  int | |
| 2. USERS\_ONLINE(session, time, ip, browser, dateonline) | |  |  | | --- | --- | | 2. USERS\_ONLINE | | | session  time  ip  browser  dateonline | varchar(100)  int  varchar(34)  varchar(550)  datetime | |
| 3. CART\_USER(id, number, user\_id, product\_id) | |  |  | | --- | --- | | 3. CART\_USER | | | id  name  user\_id  product\_id | int  int  int  int | |
| 4. CATEGORIES(id, category\_name,  category\_position, slug, supply\_id) | |  |  | | --- | --- | | 4. CATEGORIES | | | id  category\_name  category\_position  slug  supply\_id | int  varchar(255)  int  varchar(255)  int | |
| 5. PRODUCTS(id, product\_name, product\_description, product\_price, product\_color, product\_material, product\_size, product\_detail, createBy, createDate, editBy, editDate, totalView, saleoff, percentoff, image, slug, product\_typeid, category\_id, sub\_category\_id, supply\_id) | |  |  | | --- | --- | | 5. PRODUCTS | | | id  product\_name  product\_description  product\_price  product\_color  product\_material  product\_size  product\_detail  createBy  createDate  editBy  editDate  totalView  saleoff  percentoff  image  slug  product\_typeid  category\_id  sub\_category\_id  supply\_id | int  varchar(500)  longtext  int  varchar(255)  varchar(255)  varchar(100)  longtext  varchar(100)  date  varchar(100)  datetime  int  tinyint  int  varchar(255)  varchar(255)  int  int  int  int | |
| 6. ROLES(id, role\_name, role\_desc) | |  |  | | --- | --- | | 6. ROLES | | | id  role\_name  role\_desc | int  varchar(255)  varchar(500) | |
| 7. TYPES(id, type\_name, type\_description, slug) | |  |  | | --- | --- | | 7. TYPES | | | id  type\_name  type\_description  slug | int  varchar(255)  varchar(255)  varchar(255) | |
| 8. ORDERS(id, customer, province, address, phone, cart\_total, createtime, message, status, editTime, user\_id) | |  |  | | --- | --- | | 8. ORDERS | | | id  customer  province  address  phone  cart\_total  createtime  message  status  editTime  user\_id | int  varchar(255)  varchar(50)  varchar(500)  varchar(20)  double  datetime  longtext  int  datetime  int | |
| 9. ORDER\_DETAIL(id, quantity, price, order\_id, product\_id) | |  |  | | --- | --- | | 9. ORDER\_DETAIL | | | id  quantity  price  order\_id  product\_id | int  int  float  int  int | |
| 10. COMMENTS(id, parent\_comment\_id, content, createDate, author, email, status, link\_image, editTime, post\_id, page\_id, user\_id, product\_id) | |  |  | | --- | --- | | 10. COMMENTS | | | id  parent\_comment\_id  content  createDate  author  email  status  link\_image  editTime  post\_id  page\_id  user\_id  product\_id | int  int  longtext  datetime  varchar(100)  varchar(100)  int  varchar(255)  datetime  int  int  int  int | |
| 11. CONTACTS(id, link\_Logo, contact\_name, address, country, phone, phone\_2, email, link\_Contact, link\_Facebook, link\_Twitter, link\_linkedin, zalo, link\_about, about\_footer, favicon) | |  |  | | --- | --- | | 11. CONTACTS | | | id  link\_Logo  contact\_name  address  country  phone  phone\_2  email  link\_Contact  link\_Facebook  link\_Twitter  link\_linkedin  zalo  link\_about  about\_footer  favicon | int  varchar(500)  varchar(255)  varchar(500)  varchar(50)  varchar(20)  varchar(20)  varchar(255)  varchar(550)  varchar(500)  varchar(255)  varchar(255)  varchar(20)  varchar(255)  longtext  varchar(255) | |
| 12. FEEDBACKS(id, name, email, phone, subject, createTime, status, editTime, user\_id, order\_id, product\_id) | |  |  | | --- | --- | | 12. FEEDBACKS | | | id  name  email  phone  subject  createTime  status  editTime  user\_id  order\_id  product\_id | int  varchar(250)  varchar(250)  int  mediumtext  datetime  int  datetime  int  int  int | |
| 13. POSTS(id, post\_date, post\_content, post\_title, post\_status, post\_modified, post\_type, post\_modified\_user, totalView, post\_slug, post\_avatar, post\_author) | |  |  | | --- | --- | | 13. POSTS | | | id  post\_date  post\_content  post\_title  post\_status  post\_modified  post\_type  post\_modified\_user  totalView  post\_slug  post\_avatar  post\_author | int  datetime  longtext  text  varchar(20)  datetime  int  varchar(50)  int  varchar(255)  varchar(500)  int | |
| 14. SUBCATEGORY(id, subcategory\_name, slug, supply\_id, category\_id) | |  |  | | --- | --- | | 14. SUBCATEGORY | | | id  subcategory\_name  slug  supply\_id  category\_id | int  varchar(255)  varchar(100)  int  int | |
| 15. SUPPLIES(id, supply\_name, logo) | |  |  | | --- | --- | | 15. SUPPLIES | | | id  supply\_name  logo | int  varchar(500)  varchar(255) | |

#### 2.3.1.2 Mô hình thực thể liên kết



### 2.3.2 Mô hình hóa chức năng

**Biểu đồ use case phía front-end:**

A diagram of a person's relationship

Description automatically generated

**Biểu đồ use case phía back-end:**

A diagram of a person with red lines and white text

Description automatically generated

### 2.3.3 Mô tả chi tiết các use case phía front-end

#### Use case Đăng kí

* Mô tả use case

|  |  |  |
| --- | --- | --- |
| **Tên Usecase:** *Đăng ký* | | |
| **Tác nhân chính** | Khách hàng | |
| **Tác nhân phụ** | Không có | |
| **Mô tả Usecase** | Cho phép khách hàng tạo tài khoản | |
| **Điều kiện để bắt đầu Usecase:** | Không có | |
| **Điều kiện để kết thúc Usecase** | Không có | |
| **Trình tự các sự kiện luồng cơ bản trong quá trình hoạt động của Usecase** | 1. Use case này được bắt đầu khi khách hàng kích vào nút “Đăng ký” trên thanh menu. Hệ thống sẽ hiển thị bảng đăng ký và checkbox “Tôi đọc và đồng ý với các điều khoản sử dụng” lên màn hình, khách hàng sẽ điền thông tin đăng ký vào bảng, đánh dấu checkbox sau đó click vào button ‘SIGN UP”, hệ thống sẽ kiểm tra username và email được nhập có trùng với bất kì email nào tồn tại trong bảng users hay không, nếu không trùng, thông tin của một tài khoản mới sẽ được thêm vào bảng users. Use case kết thúc. | |
| **Trình tự các sự kiện luồng rẽ nhánh của Usecase** | * + 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. Nếu checkbox không được đánh dấu, hệ thống sẽ hiển thị thông báo “Bạn cần xác nhận đã đọc và đồng ý với các điều khoản sử dụng”. Use case kết thúc.     3. Nếu username/email được nhập đã tồn tại trong bảng users, hệ thống sẽ hiển thị thông báo “Username/Email đã tồn tại”. Use case kết thúc. | |
| **Hoàn cảnh sử dụng thành công cơ bản** | Khách hàng tạo tài khoản thành công | |
| **Các yêu cầu đặc biệt** | | Không có |
| **Hoàn cảnh sử dụng phụ (thay thế) trong trường hợp không thành công** | | Không có |
| **Hành động liên quan sẽ xảy ra khi Usecase kết thúc** | | Không có |

* Biểu đồ lớp



* Biểu đồ trình tự



#### Use case Đăng nhập

* Mô tả use case

|  |  |  |
| --- | --- | --- |
| **Tên Usecase:** *Đăng nhập* | | |
| **Tác nhân chính** | Khách hàng | |
| **Tác nhân phụ** | Không có | |
| **Mô tả Usecase** | Cho phép khách hàng đăng nhập vào tài khoản | |
| **Điều kiện để bắt đầu Usecase** | Không có | |
| **Điều kiện để kết thúc Usecase** | Không có | |
| **Trình tự các sự kiện luồng cơ bản trong quá trình hoạt động của Usecase** | 1. Use case này được bắt đầu khi khách hàng kích vào nút “Đăng nhập” trên thanh menu. Hệ thống sẽ hiển thị bảng đăng nhập lên màn hình, khách hàng sẽ điền thông tin đăng nhập vào bảng sau đó click vào button ‘LOG IN”, hệ thống sẽ kiểm tra username được nhập có trong bảng users hay không, nếu có: kiểm tra password được nhập có khớp với password của username đó không, nếu khớp: khách hàng được đăng nhập vào tài khoản. Use case kết thúc. | |
| **Trình tự các sự kiện luồng rẽ nhánh của Usecase** | 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. Nếu username được nhập không tồn tại trong bảng users, hệ thống sẽ hiển thị thông báo “Username không tồn tại”. Use case kết thúc. 3. Nếu password được nhập không khớp với password của username, hệ thống sẽ hiển thị thông báo “Mật khẩu không đúng”. Use case kết thúc. | |
| **Hoàn cảnh sử dụng thành công cơ bản** | Khách hàng đăng nhập tài khoản thành công | |
| **Các yêu cầu đặc biệt** | | Không có |
| **Hoàn cảnh sử dụng phụ (thay thế) trong trường hợp không thành công** | | Không có |
| **Hành động liên quan sẽ xảy ra khi use case kết thúc** | | Không có |

* Biểu đồ lớp



* Biểu đồ trình tự



#### Use case Tìm kiếm

* Mô tả use case

|  |  |  |
| --- | --- | --- |
| **Tên Usecase:** *Tìm kiếm* | | |
| **Tác nhân chính:** | Khách hàng | |
| **Tác nhân phụ:** | Không có | |
| **Mô tả Usecase:** | Cho phép khách hàng thực hiện tìm kiếm sản phẩm | |
| **Điều kiện để bắt đầu Usecase:** | Kh9u5ông có | |
| **Điều kiện để kết thúc usecase** | Không có | |
| **Trình tự các sự kiện luồng cơ bản trong quá trình hoạt động của usecase** | 1. Use case này được bắt đầu khi khách hàng kích vào nút “Tìm kiếm” trên thanh menu. Hệ thống sẽ trả về thông tin kết quả của việc tìm kiếm từ bảng PRODUCTS lên màn hình. Use case kết thúc. | |
| **Trình tự các sự kiện luồng rẽ nhánh của usecase** | * + 1. Tại bước 1 của luồng cơ bản, nếu không có kết quả nào nào thì hệ thống sẽ thông báo “ Không tìm thấy dữ liệu” và use case kết thúc.     2. Tại bất kỳ bước nào trong luồng cơ bản, 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. | |
| **Hoàn cảnh sử dụng thành công cơ bản** | Khách hàng tìm kiếm sản phẩm thành công | |
| **Các yêu cầu đặc biệt** | | Không có |
| **Hoàn cảnh sử dụng phụ (thay thế) trong trường hợp không thành công:** | | Không có |
| **Hành động liên quan sẽ xảy ra khi Usecase kết thúc** | | Không có |

* Biểu đồ lớp



* Biểu đồ trình tự



#### Use case Xem trang chủ

* Mô tả use case

|  |  |  |
| --- | --- | --- |
| **Tên Usecase:** *Xem trang chủ* | | |
| **Tác nhân chính** | Khách hàng | |
| **Tác nhân phụ** | Không có | |
| **Mô tả usecase** | Khách hàng xem trang chủ | |
| **Điều kiện để bắt đầu Usecase** | Không có | |
| **Điều kiện để kết thúc Usecase** | Không có | |
| **Trình tự các sự kiện luồng cơ bản trong quá trình hoạt động của Usecase** | 1. Use case này bắt đầu khi Khách hàng kích vào “Trang chủ”. Sau đó hệ thống truy vấn các bảng PRODUCTS,TYPES, và hiển thị lên màn hình. 2. Use case kết thúc. | |
| **Trình tự các sự kiện luồng rẽ nhánh của Usecase** | 1. Tại bất kỳ bước nào trong luồng cơ bản nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thống báo lỗi, và use case kết thúc | |
| **Hoàn cảnh sử dụng thành công cơ bản** | Khách hàng xem trang chủ thành công | |
| **Các yêu cầu đặc biệt** | | Không có |
| **Hoàn cảnh sử dụng phụ (thay thế) trong trường hợp không thành công** | | Không có |
| **Hành động liên quan sẽ xảy ra khi Usecase kết thúc:** | | Không có |

* Biểu đồ lớp

A screenshot of a computer

Description automatically generated

* Biểu đồ trình tự

A diagram of a project

Description automatically generated

#### Use case Xem sản phẩm

* Mô tả Use case

|  |  |  |
| --- | --- | --- |
| **Tên Usecase:** *Xem sản phẩm* | | |
| **Tác nhân chính** | Khách hàng | |
| **Tác nhân phụ** | Không có | |
| **Mô tả Usecase** | Khách hàng xem chi tiết sản phẩm | |
| **Điều kiện để bắt đầu Usecase** | Không có | |
| **Điều kiện để kết thúc Usecase** | Không có | |
| **Trình tự các sự kiện luồng cơ bản trong quá trình hoạt động của usecase** | 1. Use case này bắt đầu khi Khách hàng kích vào một sản phẩm. Hệ thống lấy thông tin của sản phẩm: Tên sản phẩm, Giá tiền, Hình ảnh minh họa từ bảng PRODUCTS sau đó hiển thị lên màn hình. 2. Use case kết thúc. | |
| **Trình tự các sự kiện luồng rẽ nhánh của usecase** | 1. Tại bất kỳ bước nào trong luồng cơ bản nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thống báo lỗi và use case kết thúc. | |
| **Hoàn cảnh sử dụng thành công cơ bản** | Khách hàng xem chi tiết sản phẩm thành công. | |
| **Các yêu cầu đặc biệt** | | Không có |
| **Hoàn cảnh sử dụng phụ (thay thế) trong trường hợp không thành công** | | Không có |
| **Hành động liên quan sẽ xảy ra khi Usecase kết thúc** | | Không có |

* Biểu đồ lớp



* Biều đồ trình tự



#### Use case Xem chi tiết sản phẩm

* Mô tả Use case

|  |  |  |
| --- | --- | --- |
| **Tên Usecase:** *Xem chi tiết sản phẩm* | | |
| **Tác nhân chính** | Khách hàng | |
| **Tác nhân phụ** | Không có | |
| **Mô tả Usecase** | Khách hàng xem chi tiết sản phẩm | |
| **Điều kiện để bắt đầu Usecase** | Không có | |
| **Điều kiện để kết thúc Usecase** | Không có | |
| **Trình tự các sự kiện luồng cơ bản trong quá trình hoạt động của Usecase** | 1. Use case này bắt đầu khi Khách hàng kích vào một sản phẩm. Hệ thống lấy thông tin của sản phẩm: Tên sản phẩm, Chi tiết sản phẩm, Giá tiền, Hình ảnh minh họa từ bảng PRODUCTS sau đó hiển thị lên màn hình. 2. Use case kết thúc. | |
| **Trình tự các sự kiện luồng rẽ nhánh của Usecase:** | 1. Tại bất kỳ bước nào trong luồng cơ bản nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thống báo lỗi và use case kết thúc. | |
| **Hoàn cảnh sử dụng thành công cơ bản** | Khách hàng xem chi tiết sản phẩm thành công | |
| **Các yêu cầu đặc biệt** | | Không có |
| **Hoàn cảnh sử dụng phụ (thay thế) trong trường hợp không thành công** | | Không có |
| **Hành động liên quan sẽ xảy ra khi Usecase kết thúc** | | Không có |

* Biểu đồ lớp



* Biểu đồ trình tự



#### Use case Xem giỏ hàng

* Mô tả use case

|  |  |  |
| --- | --- | --- |
| **Tên Usecase:** Xem giỏ hàng | | |
| **Tác nhân chính** | Khách hàng | |
| **Tác nhân phụ** | Không có | |
| **Mô tả Usecase** | Cho phép khách hàng thêm hàng vào giỏ, xem giỏ hàng, sửa số lượng, xóa hàng khỏi giỏ và xóa toàn bộ giỏ hàng | |
| **Điều kiện để bắt đầu Usecase** | Để xem giỏ hàng khách hàng phải đăng nhập vào hệ thống | |
| **Điều kiện để kết thúc Usecase** | Không có | |
| **Trình tự các sự kiện luồng cơ bản trong quá trình hoạt động của Usecase** | * + 1. Use case này bắt đầu khi khách hàng kích vào nút “Sở hữu” trong màn hình xem chi tiết sản phẩm. Hệ thống sẽ thêm sản phẩm hiện tại với số lượng mặc định là 1 vào danh sách các sản phẩm trong bảng ORDER\_DETAIL, đồng thời lấy danh sách các sản phẩm trong giỏ với các thông tin gồm mã giỏ hàng, số lượng từ bảng ORDER\_DETAIL, ảnh chụp, tên sản phẩm từ bảng PRODUCTS, đồng thời lấy thông tin tạm tính tổng tiền từ bảng ORDER và hiển thị lên màn hình.     2. Khách hàng sửa số lượng của một sản phầm và kích vào nút “Cập nhật số lượng”. Hệ thống sẽ cập nhật lại số lượng trong bảng ORDER\_DETAIL và cập nhật tạm tính tổng tiền trong bảng ORDER và hiển thị giỏ hàng đã cập nhật.     3. Khách hàng kích vào nút “Xóa” của một sản phẩm trong giỏ hàng. Hệ thống sẽ xóa bức tranh đó khỏi bảng ORDER\_DETAIL, cập nhật tạm tính tổng tiền trong bảng ORDER và hiển thị giỏ hàng đã cập nhật.     4. Khách hàng kích vào nút “Xóa giỏ hàng”. Hệ thống sẽ xóa toàn bộ thông tin trong bảng CART\_USER, ORDER\_DETAIL và ORDER đồng thời hiển thị thông báo “Không có sản phẩm trong giỏ hàng!”     5. Khách hàng kích vào nút “Tiếp tục mua hàng”. Hệ thống sẽ lấy tên và ảnh minh họa của các loại sản phẩm từ bảng CATEGORIES và hiển thị lên màn hình. Use case kết thúc. | |
| **Trình tự các sự kiện luồng rẽ nhánh của Usecase** | 1. Tại bước 3 trong luồng cơ bản nếu trong ORDER\_DETAIL đã hết sản phẩm thì hệ thống sẽ xóa toàn bộ thông tin trong bảng ORDER và hiển thị thông báo “Không có sản phẩm trong giỏ hàng!” và chuyển đến bước 5. 2. Tại bất kỳ bước nào trong luồng cơ bản 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. | |
| **Hoàn cảnh sử dụng thành công cơ bản** | Khách hàng xem giỏ hàng thành công | |
| **Các yêu cầu đặc biệt** | | Không có |
| **Hoàn cảnh sử dụng phụ (thay thế) trong trường hợp không thành công** | | Không có |
| **Hành động liên quan sẽ xảy ra khi Usecase kết thúc** | | Không có |

* Biểu đồ lớp



* Biểu đồ trình tự

A diagram of a project

Description automatically generated

A diagram of a project

Description automatically generated



A screen shot of a computer

Description automatically generated

#### Use case Xem bình luận

* Mô tả use case

|  |  |  |
| --- | --- | --- |
| **Tên Usecase:** Xem bình luận | | |
| **Tác nhân chính:** | Khách hàng | |
| **Tác nhân phụ:** | Không có | |
| **Mô tả Usecase:** | Khách hàng xem các bình luận về sản phẩm. | |
| **Điều kiện để bắt đầu Usecase:** | Không có | |
| **Điều kiện để kết thúc Usecase:** | Không có | |
| **Trình tự các sự kiện luồng cơ bản trong quá trình hoạt động của Usecase** | 1. Use case bắt đầu khi khách hàng kích từng sản phẩm và nút bình luận. Hệ thống sẽ lấy ra thông tin gồm có tên khách hàng, nội dung bình luận trong bảng COMMENTS và hiển thị lên màn hình. 2. Để tạo bình luận, khách hàng sẽ nhập thông tin: họ tên, email và nội dung bình luận. Sau khi nhập xong, khách hàng sẽ kích vào nút “Xác nhận gửi”. Hệ thống sẽ nhận thông tin và lưu vào bảng COMMENTS hiển thị bình luận lên màn hình. | |
| **Trình tự các sự kiện luồng rẽ nhánh của Usecase** | 1. Tại bước 1 luồng cơ bản, nếu khách hàng xem bình luận mà không có dữ liệu thì thông báo “Không có dữ liệu”. Use case kết thúc. 2. Tại bước 2 luồng cơ bản, nếu khách hàng nhập đầy đủ thì thêm thành công. Ngược lại, không nhập đầy đủ thì không cho phép nhập. Use case kết thúc. 3. Tại bất kỳ bước nào trong luồng cơ bản 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 | |
| **Hoàn cảnh sử dụng thành công cơ bản** | Khách hàng xem các bình luận thành công. | |
| **Các yêu cầu đặc biệt** | | Không có |
| **Hoàn cảnh sử dụng phụ (thay thế) trong trường hợp không thành công** | | Không có |
| **Hành động liên quan sẽ xảy ra khi Usecase kết thúc** | | Không có |

* Biểu đồ lớp



* Biều đồ trình tự

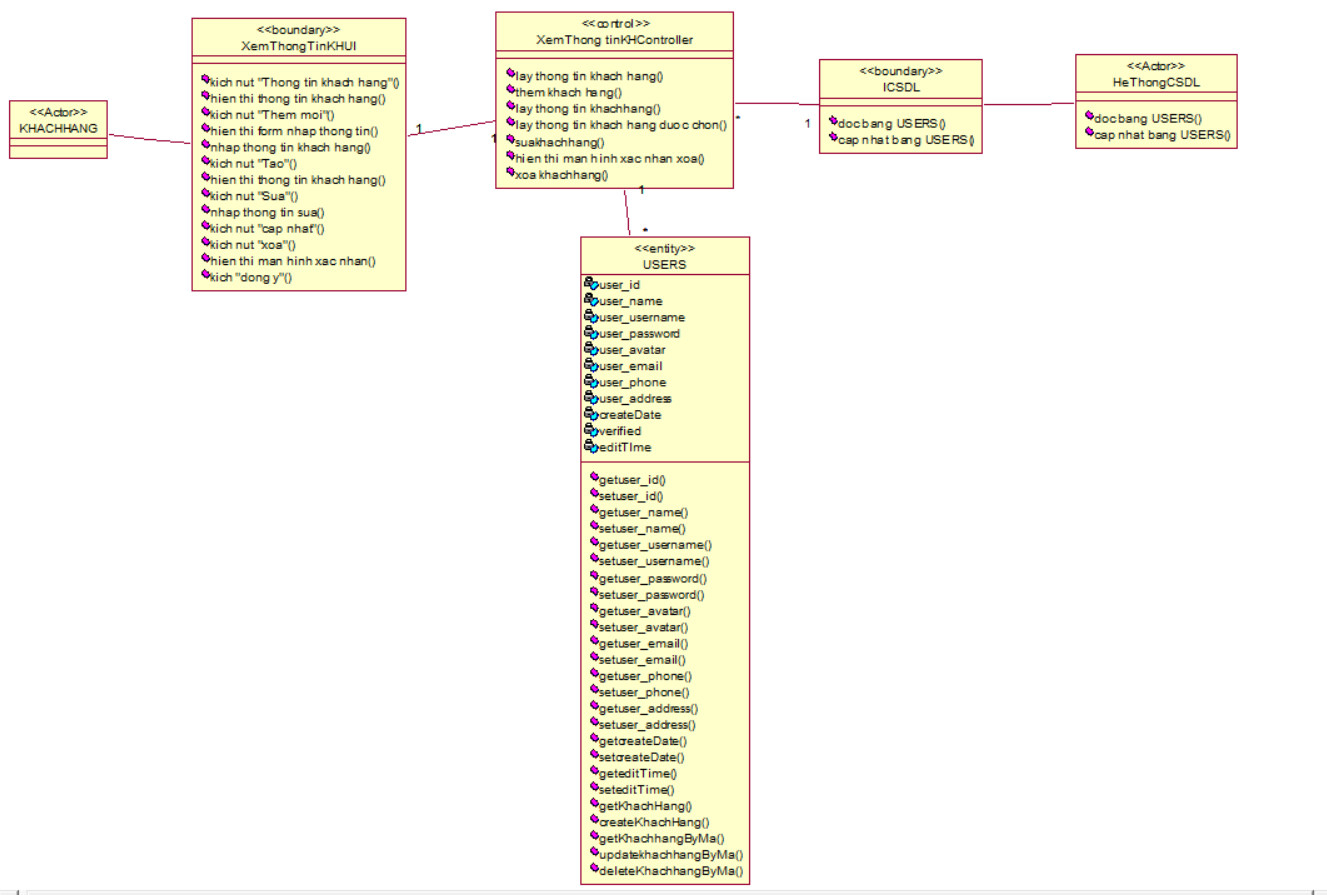


#### Use case Xem thông tin khách hàng

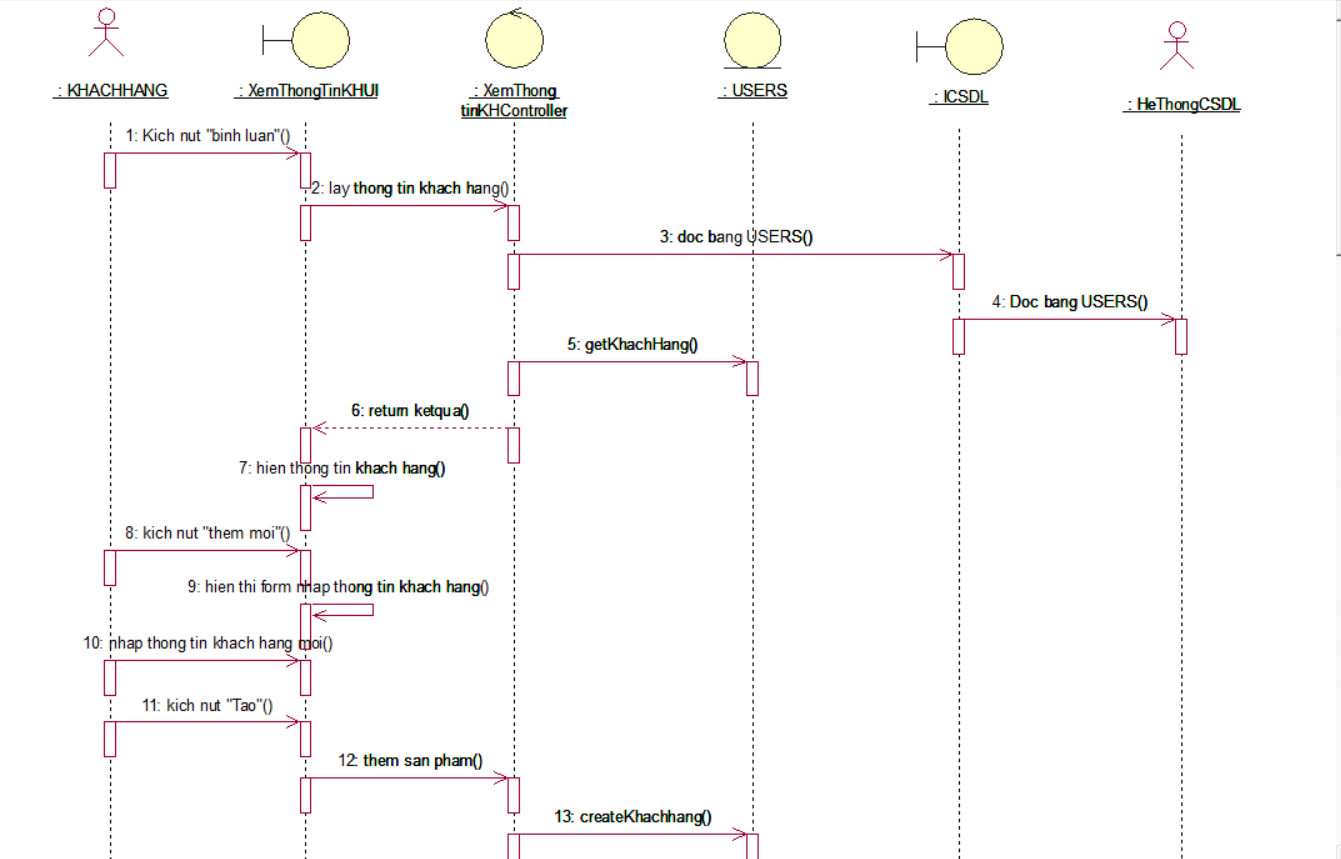
* Mô tả use case

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên Usecase:** *Xem thông tin khách hàng* | | | |
| **Tác nhân chính** | Khách hàng | | |
| **Tác nhân phụ** | Không có | | |
| **Mô tả Usecase** | Use case này cho phép khách hàng xem, thêm, sửa và xóa các thông tin cá nhân trong bảng USERS | | |
| **Điều kiện để bắt đầu Usecase:** | Không có | | |
| **Điều kiện để kết thúc Usecase:** | Không có | | |
| **Trình tự các sự kiện luồng cơ bản trong quá trình hoạt động của Usecase** | 1. Use case này bắt đầu khi khách hàng kích vào nút “Thông tin cá nhân” trên menu quản trị. Hệ thống lấy thông tin chi tiết của các khách hàng gồm: mã khách hàng, hình ảnh, tên khách hàng, địa chỉ, email, số điện thoại từ bảng USERS trong cơ sở dữ liệu và hiển thị danh sách các khách hàng lên màn hình   2. Thêm thông tin cá nhân:   1. Khách hàng kích vào nút “Thêm mới” trên cửa sổ thông tin cá nhân. Hệ thống hiển thị màn hình yêu cầu nhập thông tin chi tiết cho khách hàng gồm tên khách hàng, địa chỉ, hình ảnh, email, số điện thoại. 2. Khách hàng nhập thông tin của tên khách hàng, địa chỉ, số điện thoại tự sinh một mã khách hàng mới và kích vào nút “Tạo mới”. Hệ thống sẽ tạo một khách hàng mới trong bảng USERS và hiển thị danh sách các khách hàng đã được cập nhật.   3. Sửa thông tin cá nhân:   1. Khách hàng kích vào nút “Sửa” trên thông tin cá nhân. Hệ thống sẽ lấy thông tin cũ của khách hàng được chọn gồm có mã khách hàng, tên khách hàng, địa chỉ, số điện thoại,email từ bảng USERS và hiển thị lên màn hình. 2. Khách hàng nhập thông tin mới và kích vào nút “Cập nhật”. Hệ thống sẽ sửa thông tin của khách hàng được chọn trong bảng USERS và hiển thị thông tin cá nhân đã cập nhật.   4. Xóa thông tin cá nhân:   1. Khách hàng kích vào nút “Xóa” trên mục thông tin cá nhân. Hệ thống sẽ hiển thị một màn hình yêu cầu xác nhận xóa. 2. Khách hàng kích vào nút “Đồng ý”. Hệ thống sẽ xóa khách hàng khỏi bảng USERS và hiển thị danh sách các khách hàng đã cập nhật.Use case kết thúc. | | |
| **Trình tự các sự kiện luồng rẽ nhánh của usecase** | 1. Tại bước 2a và 3b trong luồng cơ bản nếu Khách hàng nhập thông tin cá nhân không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi yêu cầu nhập lại. Khách hàng có thể nhập lại để tiếp tục hoặc kích vào nút “Hủy bỏ” để kết thúc. 2. Tại bước 2b và 3b trong luồng cơ bản nếu Khách hàng kích vào nút “Hủy bỏ” hệ thống sẽ bỏ qua thao tác thêm mới hoặc sửa chữa tương ứng và hiển thị danh sách các khách hàng trong bảng USERS. 3. Tại bước 4b trong luồng cơ bản nếu Khách hàng kích vào nút “Không đồng ý” hệ thống sẽ bỏ qua thao tác xóa và hiển thị danh sách các khách hàng 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. | | |
| **Hoàn cảnh sử dụng thành công cơ bản** | Khách hàng bảo trì thông tin cá nhân thành công | | |
| **Các yêu cầu đặc biệt** | | Không có | |
| **Hoàn cảnh sử dụng phụ (thay thế) trong trường hợp không thành công** | | Không có |
| **Hành động liên quan sẽ xảy ra khi Usecase kết thúc** | | Không có | |

* Biểu đồ lớp



* Biều đồ trình tự



A screen shot of a computer

Description automatically generated

A diagram of a project

Description automatically generated

A diagram of a diagram

Description automatically generated

#### Use case Mua hàng

* Mô tả use case

|  |  |  |
| --- | --- | --- |
| **Tên Usecase:** *Mua hàng* | | |
| **Tác nhân chính:** | Khách hàng | |
| **Tác nhân phụ:** | Không có | |
| **Mô tả Usecase:** | Khách hàng mua sản phẩm. | |
| **Điều kiện để bắt đầu Usecase:** | Khách hàng cần phải đăng nhập thành công vào hệ thống. | |
| **Điều kiện để kết thúc Usecase** | Không có | |
| **Trình tự các sự kiện luồng cơ bản trong quá trình hoạt động của usecase** | * + 1. Use case này bắt đầu khi khách hàng chọn sản phẩm cần mua kích vào nút “Thêm vào giỏ hàng” trên từng sản phẩm. Hệ thống sẽ lấy thông tin sản phẩm trong bảng PRODUCTS gồm mã sản phẩm, tên sản phẩm, hình ảnh, giá tiền đưa vào giỏ hàng và lưu thông tin gồm mã sản phẩm, mã khách hàng và số lượng vào CART\_USER và hiển thị lên màn hình.  1. Khách hàng kích nút “Tiếp tục mua hàng” và nhập thông tin cá nhân, sau đó nhấn nút “thành toán” và hệ thống sẽ lưu thông tin của khách hàng gồm có mã đơn hàng, mã khách hàng, tên khách hàng, địa chỉ, số điện thoại, lời nhắn vào bảng ORDERS và các thông tin chi tiết về các mặt hàng bán gồm: mã đặt hàng, mã sản phẩm, số lượng và tổng tiền vào bảng ORDER\_DETAIL và xóa thông tin sản phẩm trong giỏ hàng từ bảng CART\_USER. | |
| **Trình tự các sự kiện luồng rẽ nhánh của usecase** | 1. Tại bước 1 luồng cơ bản, nếu khách hàng nhập hơn số lượng hiện có trong bảng PRODUCTS, thì hệ thống thông báo “Số lượng không đủ”. Use case kết thúc. 2. Tại bước 2 luồng cơ bản, nếu khách hàng không đăng nhập và không điền đẩy đủ thông tin thì kích nút “Thanh toán” thì thông báo “Không thực hiện được”. Use case kết thúc. 3. Tại bất kỳ bước nào trong luồng cơ bản nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi. Use case kết thúc. | |
| **Hoàn cảnh sử dụng thành công cơ bản** | Khách hàng mua sản phẩm thành công | |
| **Các yêu cầu đặc biệt** | | Không có |
| **Hoàn cảnh sử dụng phụ (thay thế) trong trường hợp không thành công** | | Không có |
| **Hành động liên quan sẽ xảy ra khi Usecase kết thúc** | | Không có |

* Biểu đồ lớp



* Biểu đồ trình tự



### 2.3.4 Mô tả chi tiết các use case phía back-end

### Giao diện

# CHƯƠNG 3: KẾT LUẬN