|  |  |
| --- | --- |
| HOÀNG ĐẶNG HÀ ANH |  |
| **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **---------------------------------------** |
|  |
| ĐỒ ÁN TỐT NGHIỆP |
| NGÀNH HỆ THỐNG THÔNG TIN |
| **XÂY DỰNG WEBSITE BÁN VẬT TƯ, PHỤ TÙNG CƠ KHÍ CÔNG TY D&D** |
| HỆ THỐNG THÔNG TIN |  |
|  |
| **CBHD: TS. Nguyễn Văn Tỉnh** |
| Sinh viên: Hoàng Đặng Hà Anh |
| **Mã số sinh viên: 2019602030** |
|  |
|  |
|  |
| Hà Nội – Năm 2023 |

# **CHƯƠNG 1: KHẢO SÁT HỆ THỐNG**

## **1.1. Giới thiệu đề tài**

“Website bán vật tư, phụ tùng cơ khí công ty DnD” là website thương mại điện tử cung cấp đa dạng vật tư, phụ tùng cơ khí hỗ trợ thi công xây dựng hay lắp ráp thiết bị máy móc cơ khí với chất lượng tốt nhất và giá cả hợp lý đến khách hàng hay các nhà thầu khi mua sản phẩm.

Website cung cấp các chức năng bán hàng và giới thiệu sản phẩm của công ty đến khách với nhiều loại sản phẩm có kích thước khác nhau. Khách hàng có thể đặt mua hàng nhanh chóng hoăc liên hệ với công ty để được tư vấn và hỗ trợ miễn phí.

Website bán vật tư, phụ tùng cơ khí công ty DnD cung cấp các sản phẩm cho các đối tượng khách hàng, nhà thầu có nhu cầu thi công xây dựng hay lắp ráp thiết bị máy móc cơ khí.

Website này được phát triển với mong muốn hỗ trợ khách hàng có nhu cầu dễ dàng mua sắm và hiểu rõ hơn về sản phẩm qua thông tin chi tiết được đăng tải hoặc tư vấn trực tiếp.

## **1.2. Khảo sát các website bán vật tư, phụ tùng cơ khí**

Để xây dựng hệ thống “Website bán vật tư, phụ tùng cơ khí công ty DnD”, em đã tiến hành khảo sát một số hệ thống cung cấp phân phối vật tư, phụ tùng cơ khí, cụ thể như sau:

### **1.2.1. Website Thịnh Phát** [**(https://thinhphatict.com)**](https://thinhphatict.com/)

Khi khảo sát hệ thống website Thịnh Phát, em tổng hợp được một số thông tin sau:

* Quản lý sản phẩm: Sản phẩm được quản lý qua các thuộc tính như mã sản phẩm, tên sản phẩm, hình ảnh, đơn giá, mô tả chi tiết.
* Quản lý tin tức: Hệ thống cung cấp các thông tin liên quan đến sản phẩm và được quản lý qua mã tin tức, tiêu đề, hình ảnh, nội dung.
* Liên hệ: Khách hàng liên hệ với đơn vị quản lý qua các hình thức như email, số điện thoại, zalo, …

Hệ thống Thịnh Phát chủ yếu phục vụ cho khách hàng và nhà quản lý. Khách hàng tìm kiếm sản phẩm, tin tức và liên hệ công ty để được tư vấn hỗ trợ mua hàng. Đối với nhà quản lý, được cấp quyền quản trị quản lý các chức năng của hệ thống.

### **1.2.2. Website Nam Hải** [**(https://bulongnamhai.com)**](https://bulongnamhai.com/)

Qua quá trình khảo sát hệ thống website Nam Hải, những thông tin được tổng hợp gồm có:

* Quản lý sản phẩm: Sản phẩm được quản lý qua các thuộc tính như mã sản phẩm, tên sản phẩm, hình ảnh, đơn giá, mô tả chi tiết.
* Quản lý tin tức: Hệ thống cung cấp các thông tin liên quan đến sản phẩm và được quản lý qua mã tin tức, tiêu đề, hình ảnh, nội dung.
* Liên hệ: Khách hàng liên hệ với đơn vị quản lý qua các hình thức như email, số điện thoại, zalo, và các mạng xã hội khác.

Hệ thống Nam Hải chủ yếu được sử dụng bởi hai đối tượng chính là khách hàng và nhà quản lý. Khách hàng sẽ tìm kiếm sản phẩm, tin tức và liên hệ công ty để được tư vấn hỗ trợ mua hàng qua hệ thống. Nhà quản lý sẽ thực hiện các chức năng quản trị của hệ thống qua tài khoản được cung cấp.

## **1.3. Yêu cầu hệ thống**

Qua quá trình khảo sát và tổng hợp thông tin, hệ thống website cần có những yêu cầu bao gồm:

### **1.3.1. Yêu cầu chức năng**

**Đối với khách hàng**

Khi vào website, khách hàng có thể xem danh sách sản phẩm theo từng danh mục hay tìm kiếm sản phẩm cần mua một cách dễ dàng. Vì thế, hệ thống cần có các chức năng sau:

* Xem trang chủ
* Xem danh sách sản phẩm theo danh mục
* Tìm kiếm sản phẩm

Bên cạnh đó, khách hàng cũng có thể xem chi tiết thông tin sản phẩm và đặt mua các sản phẩm cần mua. Do đó, hệ thống phải có các chức năng như:

* Xem chi tiết sản phẩm, bình luận
* Giỏ hàng, thanh toán

Ngoài ra, để khách hàng có thể dễ dàng theo dõi tình trạng đơn hàng và giải đáp các thắc mắc liên quan đến đơn hàng hay sản phẩm thì website cần cung cấp các chức năng gồm:

* Xem danh sách đơn hàng
* Xem chi tiết đơn hàng
* Liên hệ

Hơn nữa, khi khách hàng muốn mua sản phẩm hay liên hệ với công ty qua website thì cần có tài khoản cá nhân. Do vậy, hệ thống cần có các chức năng:

* Đăng nhập, đăng ký tài khoản
* Cập nhạt thông tin tài khoản

**Đối với nhà quản lý**

Nhà quản lý cần có tài khoản username được cấp quyền để đăng nhập vào hệ thống và thực hiện các chức năng quản trị (xem/thêm/sửa/xóa), bao gồm:

* Đăng nhập
* Quản lý khách hàng
* Quản lý sản phẩm
* Quản lý danh mục
* Quản lý đơn hàng
* Quản lý bình luận (Xem/phản hồi)
* Báo cáo thống kê

### **1.3.2. Yêu cầu phi chức năng**

Bên cạnh những yêu cầu chức năng, hệ thống cũng cần đạt được những yêu cầu phi chức năng như sau:

**Yêu cầu giao diện**

* Giao diện website phải thân thiện, đơn giản, dễ sử dụng với màu sắc và font chữ thống nhất, bố cục mạch lạc rõ ràng.
* Các chức năng cần rõ ràng, dễ thực hiện, đúng với ý nghĩa và không nên chuyển trang quá nhiều.
* Nội dung ngắn gọn, súc tính, các tiêu đề nhãn mác cần thống nhất và dùng từ ngữ chính xác nhất.

**Yêu cầu hiệu quả**

* Tốc độ của website nhanh chóng, đảm bảo yếu tố tối ưu của web, thân thiện với các công cụ tìm kiếm (Search Engine Optimization - SEO).
* Hình ảnh, video và các yếu tố khác nên giảm thiểu tối đa dung lượng để tăng thời gian tải của web.
* Giao diện tương thích với mọi hệ điều hành, thiết bị điện tử và nội dung đảm bảo tính nhất quán thẩm mỹ với các độ phân giải khác nhau.

**Yêu cầu bảo mật**

Website cần đảm bảo tính bảo mật thông tin của khách hàng bằng cách mã hóa mật khẩu trong quá trình giao tiếp giữa máy chủ và máy khách. Đồng thời cần mã hóa mật khẩu trước khi lưu vào cơ sở dữ liệu.

## **1.4. Giới thiệu công nghệ sử dụng**

### **1.4.1. PHP**

#### **1.4.1.1. Khái niệm**

PHP khởi đầu là một dự án mã nguồn mở và ngày càng phát triển. Phiên bản đầu tiên được Rasmus Lerdorf phát hành vào năm 1994. PHP là từ viết tắt của Hypertext Preprocessor. Là một ngôn ngữ kịch bản phía máy chủ được nhúng trong HTML. Nó được sử dụng để quản lý nội dung động, cơ sở dữ liệu, theo dõi phiên, thậm chí xây dựng toàn bộ các trang thương mại điện tử.

PHP được tích hợp với một số cơ sở dữ liệu phổ biến, bao gồm MySQL, Oracle,

Microsoft SQL Server, ... Hơn nữa, do được tối ưu hóa cho các ứng dụng web, tốc độ nhanh, cú pháp giống C và Java, dễ học, dễ hiểu và thời gian xây dựng sản phẩm tương đối ngắn hơn so với các ngôn ngữ khác nên PHP đã ngày càng phổ biến trên thế giới.

#### **1.4.1.2. Sử dụng phổ biến**

Nhờ một số ưu điểm sau mà PHP đã trở thành một công cụ khá phổ biến, cụ thể như sau:

* PHP thực hiện các chức năng của hệ thống, ví dụ như từ các tệp trên hệ thống mà nó có thể tạo, mở, đọc, ghi và đóng chúng.
* PHP có thể xử lý các forms, ví dụ như thu thập dữ liệu từ tệp, lưu dữ liệu vào tệp, thông qua email, bạn có thể gửi dữ liệu, trả lại dữ liệu cho người dùng.
* Thêm, sửa, xóa các phần tử trong cơ sở dữ liệu thông qua PHP.
* Truy cập các biến cookie và đặt cookie.
* Sử dụng PHP để hạn chế lượng người truy cập vào một số trang tên trang web.
* PHP có thể mã hóa dữ liệu.

#### **1.4.1.3. Đặc điểm**

Ngoài các ưu diểm nêu trên, PHP còn có các đặc điểm nổi bật như:

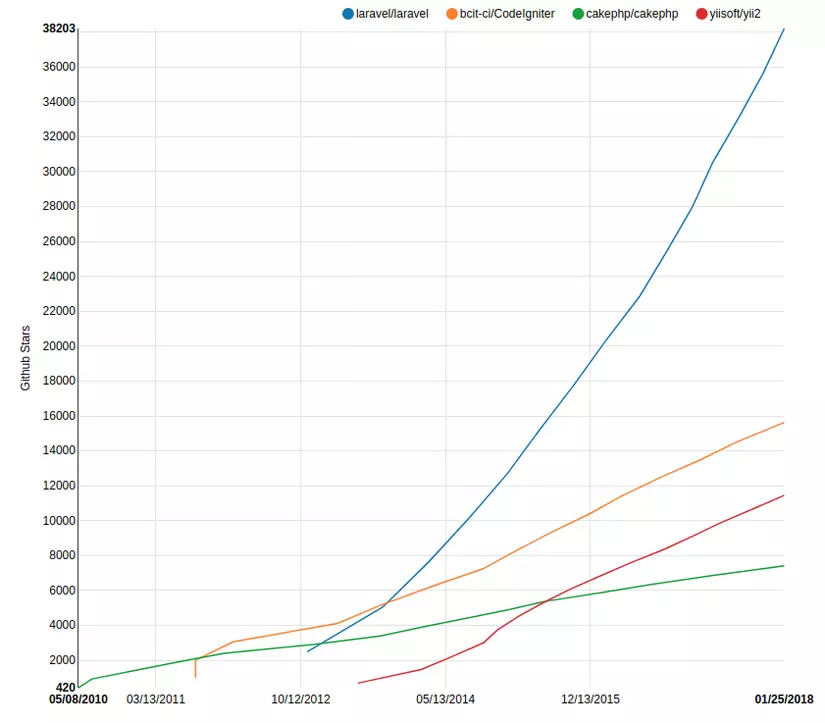
* **Tính đơn giản:** PHP cung cấp rất nhiều chức năng định nghĩa trước để bảo mật dữ liệu của bạn. Nó cũng tương thích với nhiều ứng dụng của bên thứ ba và PHP có thể dễ dàng tích hợp với các ứng dụng khác. Trong tập lệnh PHP không cần phải bao gồm các thư viện như C, các chỉ thị biên dịch đặc biệt như Java, PHP bắt đầu thực thi từ trình tự thoát (<?) và kết thúc từ trình tự thoát đóng (?>). Trong PHP script, không cần viết hàm main. Và bạn có thể làm việc với PHP mà không cần tạo lớp.
* **Tính hiệu quả:** PHP đã giới thiệu các cơ chế phân bổ tài nguyên và hỗ trợ rõ rệt cho lập trình hướng đối tượng, ngoài các tính năng quản lý phiên. Loại bỏ phân bổ bộ nhớ không cần thiết.
* **Tính bảo mật:** Một số tùy chọn mã hóa dữ liệu đáng tin cậy được hỗ trợ trong bộ chức năng được xác định trước của PHP. Bạn có thể sử dụng nhiều ứng dụng của bên thứ ba để bảo mật dữ liệu, bảo mật ứng dụng.
* **Tính quen thuộc:** Các cú pháp của PHP được kế thừa từ các ngôn ngữ lập trình cơ bản khác như C hoặc Java nên bạn có thể dễ dàng hiểu cũng như viết script PHP.
* **Tính linh hoạt:** Có thể nói rằng PHP là một ngôn ngữ rất linh hoạt vì PHP là một ngôn ngữ nhúng. Bạn có thể nhúng các tập lệnh PHP với HTML, JavaScript, XML, ... Bạn có thể chạy tập lệnh PHP của mình trên bất kỳ thiết bị nào như smartphone, PC, Laptop, ... vì tập lệnh PHP thực thi trên máy chủ sau khi gửi đến trình duyệt của thiết bị của bạn.

### **1.4.2. Laravel**

#### **1.4.2.1. Khái niệm**

Laravel là một Framework miễn phí của PHP, là mã nguồn mở, được sử dụng để phát triển website theo mô hình kiến trúc MVC. Từ đó đến nay Laravel đã phát triển mạnh mẽ, vượt qua các framework khác để vươn lên và trở thành framework PHP được ưa chuộng nhất và được cộng đồng sử dụng nhiều nhất khi phát triển website bằng PHP

Biểu đồ cho thấy sự tăng trưởng về số lượng sao trên github của Laravel so với một số framework khác.



*Hình 1.1: Biểu đồ tăng trưởng của Laravel*

Mặc dù ra đời muộn hơn so với CakePHP và CodeIgniter nhưng Laravel đã nhanh chóng được công nhận đón nhận và sử dụng nên chỉ đến khoảng giữa năm 2013 nó đã đứng đầu trong bảng xếp hạng đánh giá sao trên Github. Sở dĩ, Laravel phát triển nhanh và mạnh được đến như ngày này là nhờ vào các đặc điểm mà nó cung cấp có thể kể đến như sau:

* Dễ sử dụng, document mà Laravel cung cấp được viết rất rõ ràng và đơn giản để giúp bạn tìm thấy thứ mình muốn một cách nhanh chóng.
* Dựa trên mô hình MVC để cấu trúc và tổ chức của mã trong dự án được tổ chức tốt, dễ dàng duy trì và phát triển lâu dài.
* Được tích hợp công cụ dòng lệnh Artisan điều này làm giảm thời gian code cũng như tự động hoá một số công việc.
* Tính năng bảo mật cao. Laravel cung cấp sắn cho người dùng tính năng bảo mật cơ bản: sử dụng PDO để chống lại tấn công SQL Injection và sử dụng Field Token để chống lại tấn công CSR, các biến được đưa ra view đều được Laravel escape để tránh tấn công XSS.
* Laravel thường xuyên cập nhật các phiên bản mới, update thêm tính năng mới hỗ trợ lập trình viên.

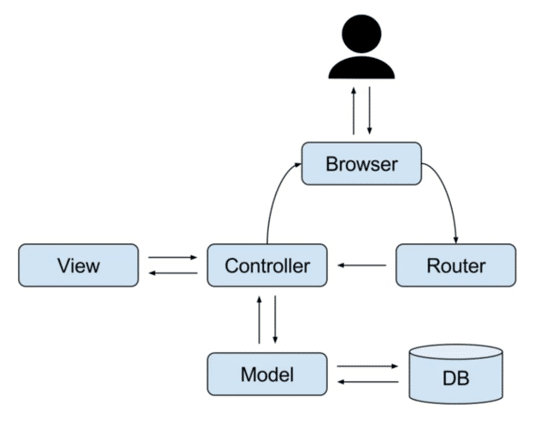
#### **1.4.2.2. Cấu trúc thư mục của Laravel**

* **Thư mục app:** chứa những code cốt lõi (core code) của ứng dụng. Hầu như tất cả các lớp (class) bạn tạo cho project sẽ nằm ở đây. Không giống các framework khác, các file model không được chứa trong một thư mục riêng biệt mà được chứa ngay tại thư mục app này.
* **Thư mục bootstrap:** chứa file app.php làm việc như một bootstrap của ứng dụng. Ngoài ra còn có thư mục cache dùng để chứa các file bộ nhớ config, route, services, ... cho việc tối ưu hiệu năng.
* **Thư mục config:** chứa tất cả file config ứng dụng, rất thuận tiện cho việc thay đổi các thiết lập.
* **Thư mục database:** chứa 2 thư mục migration (tạo và thao tác database) và seeds (tạo dữ liệu mẫu), tiện lợi để lưu trữ dữ liệu sau này.
* **Thư mục public:** chứa file index.php, .htaccess, assets (chứa các tài nguyên mà trình duyệt (browser) có thể truy cập như JS, CSS, image, ...). File khi khởi chạy ứng dụng, file .htaccess sẽ chuyển hướng mọi yêu cầu (request) tới file index.php, index.php sẽ gọi đến các thành phần tương ứng của laravel (model, view, controller, ...) để thực thi và trả về kết quả (response).
* **Thư mục resource:** chứa các tài nguyên thô chưa được biên dịch như view, LESS, SASS hoặc JavaScript, …
* **Thư mục routes:** chứa các tuyến đường (route) đã định nghĩa của ứng dụng. Mặc định các file: api.php, web.php, channels.php và console.php được kết nối với Laravel.
* **Thư mục storage:** chứa các file blade template đã được phiên dịch (complied), các file session, file cache và một số file khác được tạo bởi framework. Thư mục này gồm app, framework và logs.
* **Thư mục tests:** chứa những file tests, như PHP UnitTest dùng để test các class trong quá trình thử nghiệm trên commander.
* **File.env:** chứa các config chính của Laravel.
* **File artisan:** file thực hiện lệnh của Laravel.
* **File composer.json:** file để cấu hình việc thao tác với composer như install hay update Laravel, thêm các file hỗ trợ ...

#### **1.4.2.3. Các đặc điểm nổi bật**

**Mô hình MVC:** là viết tắt của 3 từ Model, View, Controller. Mô hình này tách một ứng dụng web ra làm 3 thành phần đảm nhiệm chức năng tách biệt, thuận tiện cho việc xử lý và bảo trì. MVC là mô hình chuẩn cho ứng dụng web được sử dụng nhiều nhất ngày nay.

* **Model:** tương tác với DB và truy xuất các thông tin từ đối tượng của bạn.
* **View:** hiển thị dữ liệu lên cho người dùng xem.
* **Controller:** xử lý các request của người dùng và truy vấn dữ liệu bằng cách sử dụng các Model.



*Hình 1.2: Sơ đồ họat động của MVC*

**Luồng hoạt động:** Khi client gửi một yêu cầu đến server thông qua Router, Controller có nhiệm vụ nhận yêu cầu, xử lý yêu cầu đó. Và nếu cần, nó sẽ gọi đến Model để làm việc với DB. Sau khi xử lý xong, toàn bộ kết quả được đẩy về View. Tại View, sẽ gen ra mã HTML tạo nên giao diện, và trả toàn bộ HTML về trình duyệt để hiển thị.

### **1.4.3. HTML**

HTML (HyperText Markup Language) là một ngôn ngữ đánh dấu được sử dụng để tạo và định dạng các trang web. Nó là một phần chính của môi trường World Wide Web (WWW) và được sử dụng để xác định cấu trúc và nội dung của một trang web. HTML sử dụng các "thẻ" để đánh dấu và định dạng các phần tử trên trang web, chẳng hạn như tiêu đề, đoạn văn bản, hình ảnh, liên kết, bảng, biểu đồ, video, …

Một số khái niệm cơ bản trong HTML bao gồm:

* + **Thẻ (Tag):** Là các từ khoá được viết trong dấu ngoặc nhọn "<>" để đánh dấu các phần tử trên trang web. Thẻ thường đi thành cặp và bao gồm thẻ mở và thẻ đóng.
  + **Phần tử (Element):** Là sự kết hợp của một thẻ mở, nội dung và thẻ đóng.
  + **Thuộc tính (Attribute):** Là thông tin bổ sung cho một thẻ, được đặt trong cặp dấu ngoặc kép " ". Thuộc tính giúp thêm thông tin, điều chỉnh và quy định hành vi của các phần tử.
  + **Điểm gắn kết (Anchor):** Sử dụng thẻ <a> để tạo liên kết đến một trang web hoặc một vị trí cụ thể trên trang hiện tại hoặc trang khác.
  + **Hình ảnh:** Sử dụng thẻ <img> để chèn hình ảnh vào trang web. Thuộc tính "src" xác định đường dẫn tới hình ảnh và "alt" cung cấp mô tả cho hình ảnh (hiển thị khi hình ảnh không tải được).
  + **Tiêu đề:** Sử dụng các thẻ tiêu đề <h1>, <h2>, <h3>, ..., <h6> để xác định cấu trúc tiêu đề của trang web, từ tiêu đề lớn nhất đến tiêu đề nhỏ nhất.
  + **Đoạn văn bản:** Sử dụng thẻ <p> để đánh dấu các đoạn văn bản.

HTML không đảm nhiệm vai trò điều khiển giao diện hoặc trải nghiệm người dùng của trang web. Để làm điều này, CSS (Cascading Style Sheets) và JavaScript thường được sử dụng kết hợp với HTML để định dạng và tạo hiệu ứng đáng chú ý cho trang web.

### **1.4.4. CSS**

CSS (Cascading Style Sheets) là một ngôn ngữ định dạng được sử dụng để tạo kiểu và trình bày các phần tử HTML trên trang web. Nó cho phép bạn xác định cách mà các phần tử HTML sẽ được hiển thị, bao gồm màu sắc, kích thước, font chữ, độ căn chỉnh, khoảng cách, hiệu ứng chuyển động và nhiều thuộc tính khác. Sử dụng CSS, bạn có thể tách biệt nội dung và kiểu dáng, giúp quản lý và duy trì trang web dễ dàng hơn.

Một số khái niệm cơ bản trong CSS bao gồm:

* + **Lựa chọn phần tử HTML:** Để áp dụng kiểu cho một phần tử HTML, bạn cần xác định lựa chọn phần tử đó trong CSS. Có thể sử dụng tên phần tử, lớp (class), ID hoặc các lựa chọn phức tạp hơn để xác định vị trí của phần tử.
  + **Thuộc tính và giá trị:** CSS sử dụng cặp thuộc tính và giá trị để xác định các kiểu dáng.
  + **Kết hợp các thuộc tính:** Bạn có thể kết hợp nhiều thuộc tính để tạo kiểu phức tạp cho các phần tử.
  + **Kế thừa:** Một số thuộc tính sẽ được kế thừa từ phần tử cha sang các phần tử con. Ví dụ, nếu bạn đặt màu chữ cho thẻ <body>, thì các đoạn văn bản trong <p> sẽ mặc định sử dụng màu đó cho đến khi bạn định nghĩa lại màu chữ trong các phần tử con.
  + **Đơn vị đo lường:** Trong CSS, bạn sử dụng đơn vị đo lường như px (pixel), % (phần trăm), em, rem, v.v. để xác định kích thước và khoảng cách của các phần tử.
  + **CSS nội tuyến và ngoại tuyến:** CSS có thể được viết trực tiếp trong thẻ HTML (CSS nội tuyến) hoặc được viết trong các tập tin riêng biệt và liên kết đến trang HTML (CSS ngoại tuyến).

### **1.4.5. JavaScript**

JavaScript là một ngôn ngữ lập trình phổ biến và mạnh mẽ được sử dụng chủ yếu trong lĩnh vực phát triển web. Nó cho phép bạn thêm tính năng tương tác và động vào các trang web, giúp làm cho trang web trở nên sống động và tương tác hơn với người dùng.

Một số khái niệm cơ bản về JavaScript:

* + **Sự kiện và xử lý sự kiện:** JavaScript cho phép bạn xử lý các sự kiện trên trang web, chẳng hạn như khi người dùng nhấp chuột, nhập liệu vào form, hoặc cuộn trang. Bạn có thể gắn các xử lý sự kiện vào các phần tử HTML bằng cách sử dụng các hàm hoặc đối tượng có sẵn.
  + **Biến và kiểu dữ liệu:** Trong JavaScript, bạn có thể sử dụng các biến để lưu trữ dữ liệu. JavaScript là ngôn ngữ kiểu động, điều đó có nghĩa là bạn không cần xác định kiểu dữ liệu cho biến trước khi sử dụng nó. Các kiểu dữ liệu trong JavaScript bao gồm số (number), chuỗi (string), đối tượng (object), mảng (array), boolean, null và undefined.
  + **Điều kiện và vòng lặp:** JavaScript hỗ trợ các cấu trúc điều kiện như "if", "else if" và "switch" để thực thi mã theo điều kiện nhất định. Ngoài ra, bạn có thể sử dụng các vòng lặp như "for", "while" và "do-while" để lặp lại các hành động nhiều lần.
  + **Hàm:** Trong JavaScript, bạn có thể định nghĩa các hàm để thực hiện các công việc cụ thể và tái sử dụng mã. Hàm có thể nhận tham số và trả về kết quả sau khi thực hiện.
  + **DOM (Document Object Model):** DOM là một giao diện lập trình ứng dụng (API) cho HTML và XML, cho phép bạn tương tác và thay đổi cấu trúc và nội dung của trang web. JavaScript được sử dụng để thao tác với DOM, giúp thay đổi nội dung của trang, thêm hoặc xóa các phần tử HTML và thực hiện các thay đổi khác trực tiếp trên trang web.
  + **AJAX (Asynchronous JavaScript and XML):** AJAX là một kỹ thuật cho phép giao tiếp không đồng bộ với máy chủ từ trang web, giúp tải lại một phần của trang mà không cần phải tải lại toàn bộ trang.

### **1.4.6. Bootstrap**

Bootstrap là một framework HTML, CSS và JavaScript cho phép người dùng dễ dàng thiết kế các trang web theo một tiêu chuẩn nhất định và tạo các trang web thân thiện với người dùng cho các thiết bị di động như di động, iPad, máy tính bảng, ...

**Đặc điểm**

Bootstrap là một trong những framework được sử dụng để xây dựng một trang web hoạt động trên nhiều thiết bị. Bootstrap xây dựng chuẩn của riêng mình và rất được người dùng ưa chuộng. Vì vậy, chúng ta thường nghe một cụm từ rất phổ biến "thiết kế theo tiêu chuẩn Bootstrap". Từ “chuẩn” này, chúng ta có thể thấy rõ lợi ích của việc sử dụng Bootstrap.

* **Dễ sử dụng:** chỉ cần có kiến thức cơ bản về HTML, CSS và Javascript là có thể sử dụng tốt Bootstrap.
* **Responsive:** Bootstrap có CSS đáp ứng được tích hợp trên điện thoại di động, máy tính bảng và thiết bị máy tính để bàn. Tính năng này giúp người dùng tiết kiệm rất nhiều thời gian trong việc tạo một trang web tiện ích và thiết bị điện tử thân thiện với người dùng.
* Có thể sử dụng trên nhiều trình duyệt phổ biến hiện nay Chrome, Firefox, Safari, ...

### **1.4.7. MySQL**

MySQL là một hệ thống quản trị cơ sở dữ liệu mã nguồn mở (Relational Database Management System - RDBMS) hoạt động theo mô hình client – server. RDBMS là một phần mềm hay dịch vụ dùng để tạo và quản lý các cơ sở dữ liệu (Database) theo hình thức quản lý các mối quan hệ giữa chúng.

**Đặc điểm**

* **Dễ sử dụng:** MySQL là một cơ sở dữ liệu nhanh, dễ sử dụng, ổn định, mở rộng, cung cấp một loạt các chức năng tiện ích mạnh mẽ.
* **Tính bảo mật cao:** MySQL rất thích hợp cho các ứng dụng truy cập cơ sở dữ liệu trên Internet nếu nó có nhiều tính năng bảo mật, kể cả ở mức cao.
* **Đa tính năng:** MySQL hỗ trợ hầu hết các tính năng SQL đối với hệ quản trị cơ sở dữ liệu quan hệ.
* **Có thể mở rộng và mạnh mẽ:** MySQL có thể xử lý lượng lớn dữ liệu và cũng có thể được mở rộng nếu được yêu cầu.
* **Nhanh:** Sự ra đời của các tiêu chuẩn khác nhau cho phép MySQL hoạt động rất hiệu quả và rẻ, giúp tăng tốc độ thực thi.

### **1.4.8. XamPP**

XamPP là phần mềm cho phép bạn giả lập môi trường server hosting ngay trên máy tính, cho phép bạn chạy trang web demo mà không cần phải mua hosting hay VPS.

XamPP được viết tắt của X + Apache + MySQL + PHP + Perl vì nó được tích hợp sẵn Apache, MySQL, PHP, FTP server, Mail Server. Còn X thể hiện cho sự đa nền tảng của XamPP vì nó có thể dùng được cho 4 hệ điều hành khác nhau: Windows, MacOS, Linux và Solaris.

**Đặc điểm**

* Có thể chạy được linh hoạt được trên mọi hệ điều hành: Cross-platform, Windows, MacOS, Linux.
* Có cấu hình đơn giản nhưng mang đến nhiều chức năng hữu ích như: Lập Server giả định, lập Mail Server giả định và hỗ trợ SSL trên localhost.
* Tích hợp được nhiều tính năng với các thành phần quan trọng như: Apache, PHP, MySQL.
* Tạo mã nguồn mở: Giao diện quản lý dễ dàng và tiện lợi giúp người dùng luôn chủ động được trong chế độ khởi động lại hay bật/ tắt đối với máy chủ theo thời gian phù hợp nhất. Đây cũng chính là nguyên nhân em lựa chọn XamPP để phát triển đồ án.

# **CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

## **2.1. Biểu đồ use case**

### **2.1.1. Biểu đồ use case tổng quát**



*Hình 2.1. Biểu đồ use case toàn bộ hệ thống*

### **2.1.2. Biểu đồ use case chính**

**Front end**

**

*Hình 2.2. Biểu đồ use case front end*

**Back end**



*Hình 2.3. Biểu đồ use case toàn bộ hệ thống*

## **2.2. Mô tả chi tiết use case**

### **2.2.1. Use case Đăng nhập**

**Tên use case:** Đăng nhập

**Mô tả vắn tắt:** Use case này cho phép khách hàng đăng nhập để xác nhận quyền truy cập hệ thống.

**Luồng sự kiện**

* + **Luồng cơ bản**

1. Use case này bắt đầu khi khách hàng kích vào nút “Đăng nhập” trên giao diện của hệ thống. Hệ thống sẽ yêu cầu khách hàng nhập username và password.
2. Khách hàng nhập tên khách hàng và mật khẩu sau đó kích vào nút “Đăng nhập”. Hệ thống kiểm tra username và password đã nhập có tồn tại trong bảng TAIKHOAN và hiển thị menu chính.
3. Use case kết thúc.

* **Luồng rẽ nhánh**

1. Tại bước 2 trong luồng cơ bản, nếu khách hàng nhập username và password sai, hệ thống sẽ hiển thị một thông báo lỗi “Đăng nhập không thành công”. Người dùng có thể chọn quay về luồng cơ bản để nhập lại, hoặc bỏ qua thao tác khi và use case sẽ kết thúc.
2. Tại bước 1 trong luồng cơ bản, nếu người dùng kích vào nút “Bỏ qua” thì 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.

**Các yêu cầu đặc biệt:** Không có

**Tiền điều kiện:** Không có

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

**Điểm mở rộng:** Không có

### **2.2.2. Use case Đăng ký**

**Tên use case:** Đăng ký

**Mô tả vắn tắt:** Use case này cho phép khách hàng đăng ký tài khoản để sử dụng hệ thống.

**Luồng sự kiện**

* + **Luồng cơ bản**

1. Use case này bắt đầu khi khách hàng kích vào nút “Đăng ký tài khoản” trên giao diện của hệ thống. Hệ thống sẽ hiển thị giao diện đăng ký gồm có thông tin như tên khách hàng, địa chỉ, số điện thoại, email, username, password. Khách hàng sẽ nhập đầy đủ thông tin vào giao diện đăng ký và kích vào nút “Đăng ký”.
2. Hệ thống kiểm chứng thông tin nếu trùng khớp với dữ liệu đã lưu thì sẽ hiển thị lại giao diện đăng ký. Nếu thông tin không trùng sẽ thêm dữ liệu vào bảng KHACHHANG và TAIKHOAN trong cơ sở dữ liệu và thông báo "Đăng ký tài khoản thành công".
3. Use case kết thúc.

* **Luồng rẽ nhánh**

1. Tại bước 1 tại luồng cơ bản, nếu khách hàng nhập thông tin không khớp với định dạng yêu cầu của hệ thống thì hệ thống sẽ thông báo lỗi “Bạn nhập thông tin chưa khớp với định dạng” và yêu cầu nhập lại thông tin.
2. Tại bước 2 luồng cơ bản, nếu thông tin khách hàng đăng ký mà trùng với dữ liệu trong bảng KHACHHANG, TAIKHOAN thì hệ thống sẽ thông báo lỗi “Thông tin đã tồn tại”. Ngược lại, nếu không trùng dữ liệu thì sẽ thông báo “Đăng ký tài khoản thành công” và 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.

**Các yêu cầu đặc biệt:** Không có

**Tiền điều kiện:** Không có

**Hậu điều kiện:** Khách hàng có thể đăng nhập vào hệ thống sau khi đăng ký tài khoản.

**Điểm mở rộng:** Không có

### **2.2.3. Use case Tìm kiếm**

**Tên use case:** Tìm kiếm

**Mô tả vắn tắt:** Use case này cho phép khách hàng thực hiện tìm kiếm sản phẩm, phụ kiện.

**Luồng sự kiện**

* + **Luồng cơ bản**

1. Use case này được bắt đầu khi khách hàng kích vào nút “Tìm kiếm” trên thanh menu.
2. Khách hành nhập từ khóa cần tìm kiếm vào ấn “Enter” hoặc kích vào biểu tượng tìm kiếm. Sau đó hệ thống sẽ lấy thông tin từ bảng DANHMUC, SANPHAM và hiển thị lên màn hình.
3. Use case kết thúc.
   * **Luồng rẽ nhánh**
     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.

**Các yêu cầu đặc biệt:** Không có

**Tiền điều kiện:** Không có

**Hậu điều kiện:** Không có

**Điểm mở rộng:** Không có

### **2.2.4. Use case Xem danh mục**

**Tên use case:** Xem danh mục

**Mô tả vắn tắt:** Use case này cho phép khách hàng xem danh mục của các sản phẩm

**Luồng sự kiện**

* + **Luồng cơ bản**

1. Use case này bắt đầu khi khách hàng kích vào “Sản phẩm”. Màn hình sẽ hiển thị một danh mục các sản phẩm của website.
2. Khách hành chọn một danh mục muốn xem và kích vào danh mục đó. Sau đó hệ thống sẽ lấy thông tin của danh mục: tên danh mục, danh sách các sản phẩm từ bảng DANHMUC và tên sản phẩm, giá, hình ảnh trong bảng SANPHAM sau đó hiển thị lên màn hình.
3. Use case kết thúc.
   * **Luồng rẽ nhánh:**
4. 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.

**Các yêu cầu đặc biệt:** Không có

**Tiền điều kiện:** Không có

**Hậu điều kiện:** Không có

**Điểm mở rộng:** Không có

### **2.2.5. Use case Xem chi tiết sản phẩm**

**Tên use case:** Xem chi tiết sản phẩm

**Mô tả vắn tắt:** Use case này cho phép khách hàng xem thông tin chi tiết của các sản phẩm

**Luồng sự kiện**

* + **Luồng cơ bản**
    1. Use case này bắt đầu khi khách hàng kích vào tên sản phẩm hoặc ảnh minh họa của một sản phẩm bất kỳ. Sau đó hệ thống sẽ lấy thông tin của sản phẩm: mã sản phẩm, tên sản phẩm, ảnh minh họa, thời gian bảo hành, giá bán, cấu hình, đánh giá, quà tặng, nhà cung cấp và số lượng sản phẩm từ bảng SANPHAM sau đó hiển thị lên màn hình.
    2. Use case kết thúc.
* **Luồng rẽ nhánh**

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 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

### **2.2.6. Use case Mua hàng**

**Tên use case:** Mua hàng

**Mô tả vắn tắt:** Use case này cho phép khách hàng chọn mua một sản

phẩm và thêm vào giỏ hàng.

**Luồng sự kiện**

* **Luồng cơ bản**

1. Use case này bắt đầu khi khách hàng nhập số lượng sản phẩm hoặc số lượng mặc định là 1 và kích vào nút “Chọn mua sản phẩm” trên màn hình chi tiết sản phẩm. Hệ thống lấy thông tin sản phẩm gồm có mã sản phẩm, số lượng và thêm vào giỏ hàng với bảng GIOHANG.
2. Use case kết thúc.

* **Luồng rẽ nhánh**

1. Tại bước 1 trong luồng cơ bản, khách hàng không nhập số lượng sản phẩm hoặc số lượng nhỏ hơn 0. Hệ thống sẽ thông báo “Vui lòng nhập số lượng mua lớn hơn hoặc bằng 1”. Use case kết thúc.
2. Tại bất kì thời điểm nào trong quá trình thực hiện ca sử dụng nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

**Các yêu cầu đặc biệt:** Không có.

**Tiền điều kiện:** Khách hàng cần đăng nhập trước khi chọn mua sản phẩm.

**Hậu điều kiện:** Sản phẩm trong giỏ hàng cần được lưu vào cơ sở dữ liệu.

**Điểm mở rộng:** Không có.

### **2.2.7. Use case Đặt hàng**

**Tên use case:** Đặt hàng

**Mô tả vắn tắt:** Use case này cho phép khách hàng chọn mua một sản

phẩm và thêm vào giỏ hàng.

**Luồng sự kiện**

* **Luồng cơ bản**

1. Use case này bắt đầu khi khách hàng kích nút “Mua hàng” trong màn hình giỏ hàng. Hệ thống hiển thị thông tin nhận hàng, hình thức thanh toán, danh sách sản phẩm mua, phí vận chuyển và tổng tiền.
2. Khách hàng điền đầy đủ thông tin nhận hàng gồm có email, họ tên, số điện thoại, địa chỉ, lời nhắn và chọn phương thức thanh toán phù hợp. Sau đó hệ thống sẽ tính tổng tiền đơn hàng và hiển thị phí vận chuyển và tổng tiền lên màn hình.
3. Khi khách hàng kích nút “Đặt hàng”, hệ thống sẽ lưu thông tin nhận hàng gồm có email, họ tên, số điện thoại, địa chỉ vào bảng KHACHHANG; lời nhắn, mã đơn hàng, hình thức thanh toán, phí vận chuyển, trạng thái đơn hàng, trạng thái thanh toán và tổng tiền vào bảng DONHANG và danh sách sản phẩm mua, số lượng lưu vào bảng CHITIETDONHANG; xóa dữ liệu đơn hàng ở bảng GIOHANG; cập nhật số lượng ở bảng SANPHAM và hiển thị thông báo “Mua hàng thành công”.
4. Use case kết thúc.

* **Luồng rẽ nhánh**

1. Tại bước 1 trong luồng cơ bản, khi giỏ hàng không có sản phẩm nào thì không hiển thị nút “Mua hàng”. Use case kết thúc.
2. Tại bước 2 trong luồng cơ bản, khi khách hàng không nhập đầy đủ thông tin thì hệ thống thông báo “Vui lòng điền vào trường này”. Use case kết thúc.
3. Tại bất kì thời điểm nào trong quá trình thực hiện ca sử dụng nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

**Các yêu cầu đặc biệt:** Không có.

**Tiền điều kiện:** Khách hàng cần đăng nhập vào hệ thống và thêm sản phẩm vào giỏ hàng.

**Hậu điều kiện:** Đơn hàng được lưu vào cơ sở dữ liệu.

**Điểm mở rộng:** Không có.

### **2.2.8. Use case Xem chi tiết đơn hàng**

**Tên use case:** Xem chi tiết đơn hàng

**Mô tả vắn tắt:** Use case này cho phép khách hàng xem chi tiết thông tin, trạng thái đơn hàng và trạng thái vận chuyển đơn hàng; cập nhật thông tin khách hàng.

**Luồng sự kiện**

* **Luồng cơ bản**

1. Use case này bắt đầu khi khách hàng chọn một đơn hàng trong danh sách đơn hàng đã mua. Hệ thống sẽ hiển thị thông tin đơn hàng gồm có mã đơn hàng, thời gian đặt hàng, trạng thái đơn hàng, đơn vị vận chuyển, mã vận chuyển, trạng thái thanh toán, tổng tiền, phí vận chuyển, lời nhắn từ bảng DONHANG; họ tên khách hàng, email, số điện thoại, địa chỉ từ bảng KHACHHANG; thời gian, trạng thái vận chuyển từ bảng VANCHUYEN; số lượng, đơn giá từ bảng CHITIETDONHANG; hình ảnh, tên sản phẩm từ bảng SANPHAM.
2. Khi đơn hàng chưa được xác nhận, khách hàng có thể kích nút “Hủy đơn hàng”. Hệ thống sẽ cập nhật trạng thái “Đã hủy” vào bảng DONHANG và cập nhật lại số lượng sản phẩm trong bảng SANPHAM.
3. Khi đơn hàng được xác nhận và đã giao hàng thành công, khách hàng có thể kích nút “Đã nhận được hàng”. Hệ thống sẽ cập nhật trạng thái “Đã nhận hàng” vào bảng DONHANG.
4. Đối với hình thức thanh toán online, khi đơn hàng đã được đặt hàng mà chưa thanh toán, khách hàng kích nút “Thanh toán”. Hệ thống sẽ hiển thị màn hình chọn hình thức thanh toán.
5. Khi khách hàng chọn phương thức thanh toán và kích nút “Thanh toán”. Hệ thống sẽ thanh toán và hiển thị thông báo thành công.

* **Luồng rẽ nhánh**

1. Tại bước 5 trong luông cơ bản, khi khách hàng không chọn hình thức thanh toán mà kích nút “Thanh toán”. Hệ thống sẽ hiển thị thông báo “Bạn chưa chọn hình thức thanh toán”. Use case kết thúc.
2. Tại bất kì thời điểm nào trong quá trình thực hiện ca sử dụng nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

**Các yêu cầu đặc biệt:** Không có.

**Tiền điều kiện:** Khách hàng cần đăng nhập để xem thông tin đơn hàng.

**Hậu điều kiện:** Không có.

**Điểm mở rộng:** Không có.

### **2.2.9. Use case Quản lý khách hàng**

**Tên use case:** Quản lý khách hàng

**Mô tả vắn tắt:** Use case này cho phép người quản trị xem, thêm, sửa và xóa các thông tin khách hàng.

**Luồng sự kiện**

* + **Luồng cơ bản**

1. Use case này bắt đầu khi người quản trị kích vào nút “Quản lý khách hàng” trên menu quản trị. Nhà quản trị chọn một dòng khách hàng, sau đó hệ thống lấy thông tin chi tiết của các khách hàng gồm: mã khách hàng, tên khách hàng, địa chỉ, số điện thoại, email từ bảng KHACHHANG trong cơ sở dữ liệu và hiển thị danh sách các khách hàng lên màn hình.
2. Thêm khách hàng.
3. Người quản trị kích vào nút “Thêm mới” trên cửa sổ danh sách khách hàng. Hệ thống hiển thị màn hình yêu cầu nhập thông tin chi tiết cho khách hàng gồm tên khách hàng, địa chỉ, số điện thoại, email.
4. Người quản trị nhập thông tin của của khách hàng (tên khách hàng, địa chỉ, số điện thoại, email) và hệ thống sẽ tự sinh một mã khách hàng mới và kích vào nút “Tạo mới”. Hệ thống sẽ tạo một khách hàng mới trong bảng KHACHHANG và hiển thị danh sách các khách hàng đã được cập nhật.
5. Sửa khách hàng.
6. Người quản trị kích vào nút “Sửa” trên một dòng khách hàng. 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 KHACHHANG và hiển thị lên màn hình.
7. Người quản trị nhập thông tin mới cho khách hàng và kích vào nút “Cập nhật”. Hệ thống sẽ sửa thông tin của khách hàng được chọn trong bảng KHACHHANG và hiển thị danh sách khách hàng đã cập nhật.
8. Xóa khách hàng
9. Người quản trị kích vào nút “Xóa” trên một dòng khách hàng. Hệ thống sẽ hiển thị một màn hình yêu cầu xác nhận xóa.
10. Người quản trị kích vào nút “Đồng ý”. Hệ thống sẽ xóa khách hàng được chọn khỏi bảng KHACHHANG và hiển thị danh sách các khách hàng đã cập nhật.
11. Use case kết thúc.

* **Luồng rẽ nhánh**

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

**Các yêu cầu đặc biệt:** Use case này chỉ cho phép một số vai trò như người quản trị, người chủ hệ thống thực hiện.

**Tiền điều kiện:** Người quản trị cần đăng nhập với vai trò quản trị hệ thống trước khi có thể thực hiện use case.

**Hậu điều kiện:** Nếu use case kết thúc thành công thì thông tin về khách hàng sẽ được cập nhật trong cơ sở dữ liệu.

**Điểm mở rộng:** Không có

### **2.2.10. Use case Quản lý sản phẩm**

**Tên use case:** Quản lý sản phẩm

**Mô tả vắn tắt:** Use case này cho phép người quản trị xem, thêm, sửa và xóa các thông tin sản phẩm.

**Luồng sự kiện**

* + **Luồng cơ bản:**

1. Use case này bắt đầu khi người quản trị kích vào nút “Quản lý sản phẩm” trên menu quản trị. Nhà quản trị chọn một dòng sản phẩm, sau đó hệ thống lấy thông tin chi tiết của các sản phẩm gồm: mã sản phẩm, tên sản phẩm, ảnh minh họa, thời gian bảo hành, giá bán, cấu hình, đánh giá, quà tặng, nhà cung cấp và số lượng sản phẩm từ bảng SANPHAM trong cơ sở dữ liệu và hiển thị danh sách các sản phẩm lên màn hình.
2. Thêm sản phẩm
3. Người quản trị kích vào nút “Thêm mới” trên cửa sổ danh sách sản phẩm. Hệ thống hiển thị màn hình yêu cầu nhập thông tin chi tiết cho sản phẩm gồm: mã sản phẩm, tên sản phẩm, ảnh minh họa, thời gian bảo hành, giá bán, cấu hình, đánh giá, quà tặng, nhà cung cấp và số lượng sản phẩm.
4. Người quản trị nhập thông tin của của sản phẩm (tên sản phẩm, ảnh minh họa, thời gian bảo hành, giá bán, cấu hình, đánh giá, quà tặng, nhà cung cấp và số lượng sản phẩm) và hệ thống sẽ tự sinh một mã sản phẩm mới và kích vào nút “Tạo mới”. Hệ thống sẽ tạo một sản phẩm mới trong bảng SANPHAM và hiển thị danh sách các sản phẩm đã được cập nhật.
5. Sửa sản phẩm
6. Người quản trị kích vào nút “Sửa” trên một dòng sản phầm. Hệ thống sẽ lấy thông tin cũ của sản phẩm được chọn gồm mã sản phẩm, tên sản phẩm, ảnh minh họa, thời gian bảo hành, giá bán, cấu hình, đánh giá, quà tặng, nhà cung cấp và số lượng sản phẩm từ bảng SANPHAM và hiển thị lên màn hình.
7. Người quản trị nhập thông tin mới cho sản phẩm và kích vào nút “Cập nhật”. Hệ thống sẽ sửa thông tin của sản phẩm được chọn trong bảng SANPHAM và hiển thị danh sách sản phẩm đã cập nhật.
8. Xóa sản phẩm
9. Người quản trị kích vào nút “Xóa” trên một dòng sản phẩm. Hệ thống sẽ hiển thị một màn hình yêu cầu xác nhận xóa.
10. Người quản trị kích vào nút “Đồng ý”. Hệ thống sẽ xóa sản phẩm được chọn khỏi bảng SANPHAM và hiển thị danh sách các sản phẩm đã cập nhật.
11. Use case kết thúc.

* **Luồng rẽ nhánh:**

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

**Các yêu cầu đặc biệt:** Use case này chỉ cho phép một số vai trò như người quản trị, người chủ hệ thống thực hiện.

**Tiền điều kiện:** Người quản trị cần đăng nhập với vai trò quản trị hệ thống trước khi có thể thực hiện use case.

**Hậu điều kiện:** Nếu use case kết thúc thành công thì thông tin về sản phẩm sẽ được cập nhật trong cơ sở dữ liệu.

**Điểm mở rộng:** Không có

### **2.2.11. Use case Quản lý đơn hàng**

**Tên của use case:** Quản lý đơn hàng

**Mô tả vắn tắt:** Use case này cho phép người quản trị xem chi tiết, cập nhật và xóa các đơn hàng trong bảng DONHANG và CHITIETDONHANG.

**Luồng sự kiện**

* **Luồng cơ bản**

1. Use case này bắt đầu khi người quản trị kích vào “Danh sách đơn hàng” trên menu quản trị. Hệ thống lấy thông tin chi tiết đơn hàng gồm: mã đơn hàng, họ tên khách hàng, ngày tạo, hình thức thanh toán, trạng thái đơn hàng, trạng thái thanh toán, tổng tiền, vận chuyển, người xử lý từ bảng DONHANG trong cơ sở dữ liệu và hiển thị danh sách đơn hàng lên màn hình.
2. Cập nhật trạng thái đơn hàng
3. Người quản trị kích vào nút trạng thái trên từng dòng đơn hàng. Hệ thống sẽ hiển thị màn hình cập nhật trạng thái đơn hàng.
4. Người quản trị chọn trạng thái đơn hàng và kích nút “Cập nhật”. Hệ thống sẽ sửa trạng thái đơn hàng trong bảng DONHANG và hiển thị danh sách đơn hàng đã cập nhật.
5. Cập nhật đơn vị vận chuyển
6. Người quản trị kích vào trạng thái vận chuyển trên từng dòng đơn hàng. Hệ thống hiển thị màn hình cập nhật đơn vị vận chuyển và mã vận chuyển. Người quản trị điền thông tin và kích nút “Cập nhật”. Hệ thống cập nhật vào bảng DONHANG vào cơ sở dữ liệu và hiển thị lên màn hình.
7. Người quản trị kích vào tên nhà vận chuyển. Hệ thống hiển thị màn hình cập nhật trạng thái vận chuyển. Người quản trị kích chọn trạng thái vận chuyển và kích nút “Cập nhật”. Hệ thống tạo mới vào VANCHUYEN và hiển thị lên màn hình.
8. Cập nhật đơn hàng
9. Người quản trị kích chọn “Cập nhật” trên từng dòng đơn hàng. Hệ thống hiển thị thông tin đơn hàng gồm có mã đơn hàng, ngày tạo, hình thức thanh toán, trạng thái đơn hàng và thanh toán, đơn vị vận chuyển, mã vận chuyển, trạng thái vận chuyển, ghi chú, người xử lý, tổng tiền, phí vận chuyển từ bảng DONHANG; hình ảnh, tên sản phẩm từ bảng SANPHAM; số lượng và đơn giá từ bảng CHITIETDONHANG; mã khách hàng, tên khách hàng, điện thoại, email, địa chỉ từ bảng KHACHHANG và hiển thị lên màn hình.
10. Người quản trị cập nhật trạng thái đơn hàng, trạng thái vận chuyển ở bước 2, 3 trong luồng cơ bản; Người quản trị cập nhật người xử lý đơn hàng ở bước 4 trong luồng cơ bản ở trang chi tiết đơn hàng.
11. Xóa sản phẩm
12. Người quản trị kích vào nút “Xóa” trên một dòng sản phẩm. Hệ thống sẽ hiển thị một màn hình yêu cầu xác nhận xóa.
13. Người quản trị kích vào nút “Đồng ý”. Hệ thống sẽ xóa đơn hàng được chọn khỏi bảng DONHANG và CHITIETDONHANG và hiển thị danh sách đơn hàng đã cập nhật.
14. Use case kết thúc.

* **Luồng rẽ nhánh**

1. Tại bước 2b hoặc 3b hoặc 4b trong luồng cơ bản nếu người quản trị nhập thông tin sản phẩm không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi yêu cầu chọn lại.
2. Tại bước 5b trong luồng cơ bản nếu người quản trị kích vào nút
3. “Hủy” hệ thống sẽ bỏ qua thao tác xóa và hiển thị danh sách đơn hàng trong bảng DONHANG.
4. Tại bất kì thời điểm nào trong quá trình thực hiện ca sử dụng nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

**Các yêu cầu đặc biệt:** Use case này chỉ cho phép một số vai trò như người quản trị thực hiện.

**Tiền điều kiện:** Người quản trị cần đăng nhập với vai trò quản trị hệ thống trước khi có thể thực hiện use case.

**Hậu điều kiện:** Nếu use case kết thúc thành công thì thông tin về đơn hàng sẽ được cập nhật trong cơ sở dữ liệu.

**Điểm mở rộng:** Không có.

## **2.3. Phân tích use case**

### **2.3.1. Use case Đăng nhập**



*Hình 2.4. Biểu đồ trình tự - Use case Đăng nhập*



*Hình 2.5. Biểu đồ VOPC – Use case Đăng nhập*

### **2.3.2. Use case Đăng ký**



*Hình 2.6. Biểu đồ trình tự - Use case Đăng ký*



*Hình 2.7. Biểu đồ VOPC - Use case Đăng ký*

### **2.3.3. Use case Tìm kiếm**



*Hình 2.8. Biểu đồ trình tự - Use case Tìm kiếm*

**

*Hình 2.9. Biểu đồ VOPC - Use case Tìm kiếm*

### **2.3.4. Use case Xem danh mục**

**

*Hình 2.10. Biểu đồ trình tự - Use case Xem danh mục*

**

*Hình 2.11. Biểu đồ VOPC - Use case Xem danh mục*

### **2.3.5. Use case Xem chi tiết sản phẩm**

**

*Hình 2.12. Biểu đồ trình tự - Use case Xem chi tiết sản phẩm*

**

*Hình 2.13. Biểu đồ VOPC - Use case Xem chi tiết sản phẩm*

### **2.3.6. Use case Mua hàng**



*Hình 2.14. Biểu đồ VOPC - Use case Mua hàng*



*Hình 2.15. Biểu đồ VOPC - Use case Mua hàng*

### **2.3.7. Use case Đặt hàng**



*Hình 2.16. Biểu đồ VOPC - Use case Đặt hàng*



*Hình 2.17. Biểu đồ VOPC - Use case Đặt hàng*

### **2.3.8. Use case Xem chi tiết đơn hàng**

### **2.3.9. Use case Quản lý khách hàng**

**

*Hình 2.20. Biểu đồ trình tự - Use case Quản lý khách hàng*

**

*Hình 2.21. Biểu đồ VOPC - Use case Quản lý khách hàng*

### **2.3.10. Use case Quản lý sản phẩm**

**

*Hình 2.22. Biểu đồ trình tự - Use case Quản lý sản phẩm*

**

*Hình 2.23. Biểu đồ VOPC - Use case Quản lý sản phẩm*

### **2.3.11. Use case Quản lý đơn hàng**



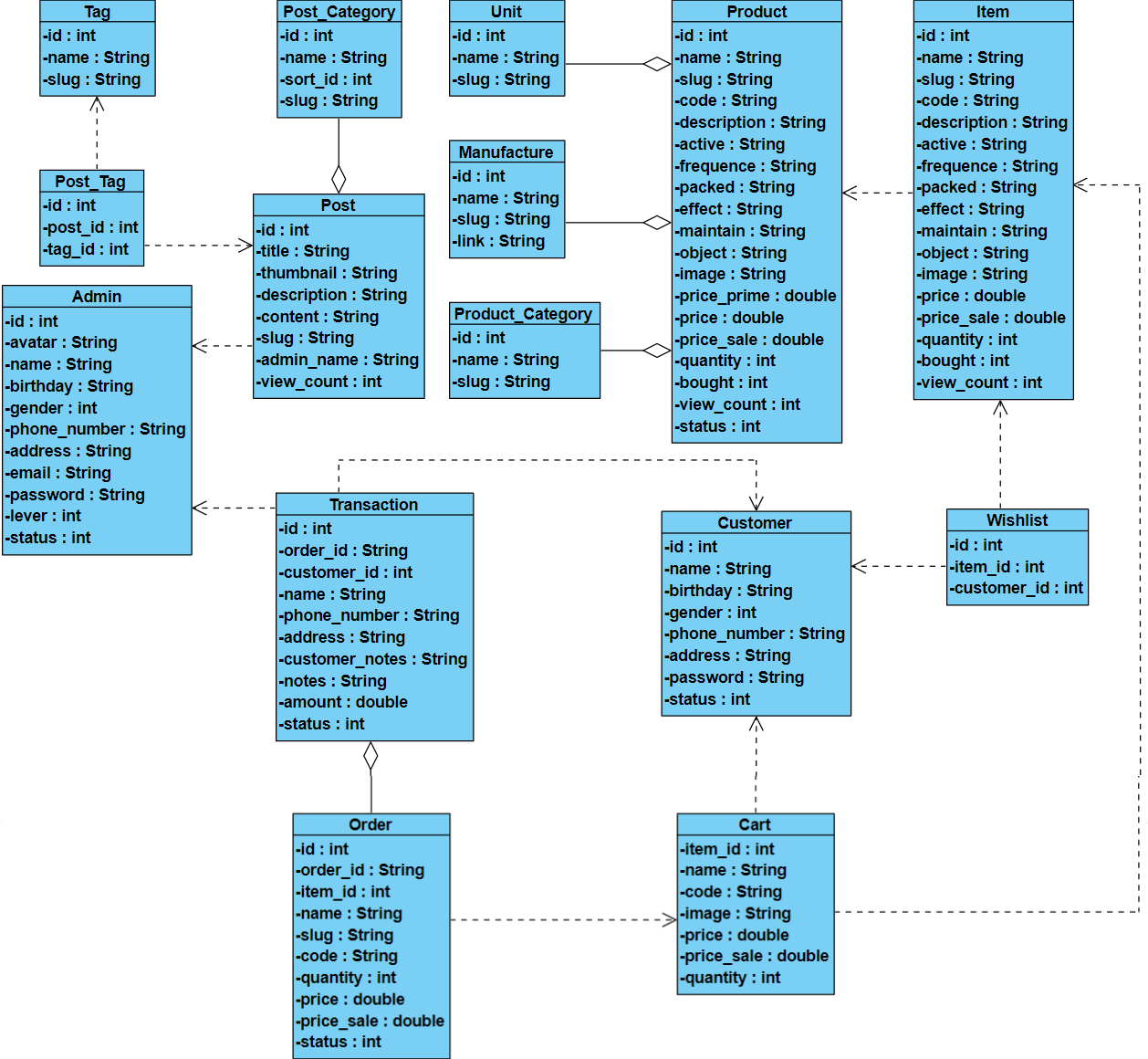
*Hình 2.24. Biểu đồ VOPC - Use case Quản lý đơn hàng*

**

*Hình 2.25. Biểu đồ VOPC - Use case Xem chi tiết sản phẩm*

## **2.4. Biểu đồ lớp**

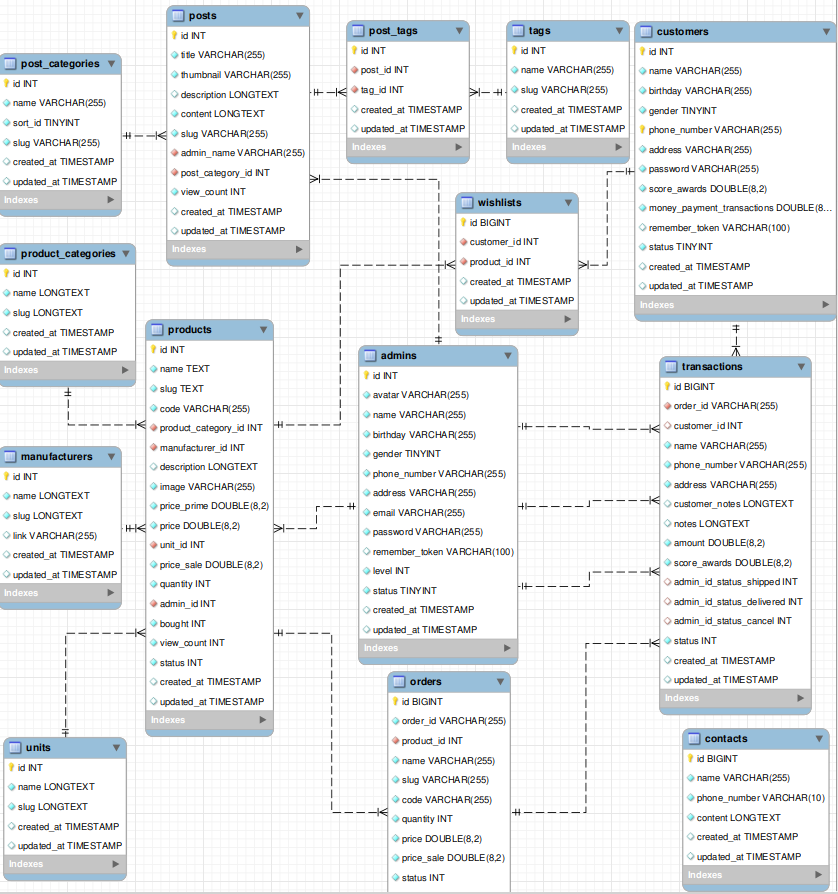
Xây dựng biểu đồ lớp phân tích là một phần của phân tích mô hình tĩnh, là cách mô tả để xây dựng và khởi tạo hệ thống, thể hiện việc phân chia các thành phần logic và vật lý của hệ thống và cách chúng tương tác với nhau.



*Hình 2.26. Biểu đồ lớp*

## **2.5. Thiết kế cơ sở dữ liệu**

### **2.5.1. Mô hình thực thể liên kết**



*Hình 2.27. Mô hình thực thể liên kết*

### **2.5.2. Thiết kế bảng dữ liệu**

#### **2.5.2.1. Bảng nhân viên (admins)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Not Null** | **Mô tả** |
| id | INT | x | định danh |
| avatar | VARCHAR(255) | x | tên ảnh đại diện |
| name | VARCHAR(255) | x | tên nhân viên |
| birthday | VARCHAR(255) | x | ngày sinh |
| gender | TINYINT | x | giới tính |
| phone\_number | VARCHAR(255) | x | số điện thoại |
| address | VARCHAR(255) | x | địa chỉ |
| email | VARCHAR(255) | x | email |
| password | VARCHAR(255) | x | mật khẩu |
| remember\_token | VARCHAR(100) |  | token để nhớ thông tin |
| level | INT | x | quyền truy cập |
| status | TINYINT | x | tình trạng hoạt động |
| created\_at | TIMESTAMP |  | thời gian tạo |
| updated\_at | TIMESTAMP |  | thời gian cập nhật lần cuối |

*Hình 2.28. Bảng nhân viên (admins)*

2.5.2.2. Bảng khách hàng (customers)

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Not Null** | **Mô tả** |
| id | INT | x | định danh |
| name | VARCHAR(255) | x | tên khách hàng |
| birthday | VARCHAR(255) | x | ngày sinh |
| gender | TINYINT | x | giới tính |
| phone\_number | VARCHAR(255) | x | số điện thoại |
| address | VARCHAR(255) | x | địa chỉ |
| password | VARCHAR(255) | x | mật khẩu |
| score\_awards | DOUBLE(8, 2) | x | số điểm thưởng |
| money\_payment\_transactions | DOUBLE(8, 2) | x | số tiền đã mua hàng |
| remember\_token | VARCHAR(100) |  | token để nhớ thông tin |
| status | TINYINT | x | tình trạng hoạt động |
| created\_at | TIMESTAMP |  | thời gian tạo |
| updated\_at | TIMESTAMP |  | thời gian cập nhật lần cuối |

2.5.2.3. Bảng sản phẩm (products)

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Not Null** | **Mô tả** |
| id | INT | x | định danh |
| name | TEXT | x | tên sản phẩm |
| slug | TEXT | x | đường dẫn |
| code | VARCHAR(255) | x | mã sản phẩm |
| product\_category\_id | INT | x | id của loại sản phẩm |
| manufacturer\_id | INT | x | id của nhà sản xuất |
| description | LONGTEXT |  | mô tả ngắn |
| image | VARCHAR(255) | x | tên ảnh |
| price\_prime | DOUBLE(8, 2) | x | giá nhập |
| price | DOUBLE(8, 2) | x | giá bán |
| unit\_id | INT | x | id của đơn vị tính |
| price\_sale | DOUBLE(8, 2) | x | giá bán khuyến mại |
| quantity | INT | x | số lượng tồn kho |
| admin\_id | INT | x | id của nhân viên |
| bought | INT | x | số lượng đã bán |
| view\_count | INT | x | số lượt xem |
| status | INT | x | tình trạng hoạt động |
| created\_at | TIMESTAMP |  | thời gian tạo |
| updated\_at | TIMESTAMP |  | thời gian cập nhật lần cuối |

2.5.2.4. Bảng đơn vị tính (units)

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Not Null** | **Mô tả** |
| id | INT | x | định danh |
| name | LONGTEXT | x | tên đơn vị tính |
| slug | LONGTEXT | x | đường dẫn |
| created\_at | TIMESTAMP |  | thời gian tạo |
| updated\_at | TIMESTAMP |  | thời gian cập nhật lần cuối |

2.5.2.5. Bảng loại sản phẩm (product\_categories)

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Not Null** | **Mô tả** |
| id | INT | x | định danh |
| name | LONGTEXT | x | tên loại sản phẩm |
| slug | LONGTEXT | x | đường dẫn |
| created\_at | TIMESTAMP |  | thời gian tạo |
| updated\_at | TIMESTAMP |  | thời gian cập nhật lần cuối |

2.5.2.6. Bảng nhà sản xuất (manufacturers)

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Not Null** | **Mô tả** |
| id | INT | x | định danh |
| name | LONGTEXT | x | tên nhà sản xuất |
| slug | LONGTEXT | x | đường dẫn |
| link | VARCHAR(255) |  | đường link của nhà sản xuất |
| created\_at | TIMESTAMP |  | thời gian tạo |
| updated\_at | TIMESTAMP |  | thời gian cập nhật lần cuối |

2.5.2.7. Bảng sản phẩm yêu thích (wishlists)

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Not Null** | **Mô tả** |
| id | BIGINT | x | định danh |
| customer\_id | INT | x | id của khách hàng |
| product\_id | INT | x | id của sản phẩm |
| created\_at | TIMESTAMP |  | thời gian tạo |
| updated\_at | TIMESTAMP |  | thời gian cập nhật lần cuối |

2.5.2.8. Bảng đơn hàng (orders)

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Not Null** | **Mô tả** |
| id | BIGINT | x | định danh |
| order\_id | VARCHAR(255) | x | mã đơn hàng |
| product\_id | INT | x | id của sản phẩm |
| name | VARCHAR(255) | x | tên sản phẩm |
| slug | VARCHAR(255) | x | đường dẫn |
| code | VARCHAR(255) | x | mã sản phẩm |
| quantity | INT | x | số lượng |
| price | DOUBLE(8, 2) | x | giá bán |
| price\_sale | DOUBLE(8, 2) | x | giá bán khuyến mại |
| status | INT | x | tình trạng đơn hàng |
| created\_at | TIMESTAMP |  | thời gian tạo |
| updated\_at | TIMESTAMP |  | thời gian cập nhật lần cuối |

2.5.2.9. Bảng giao dịch (transactions)

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Not Null** | **Mô tả** |
| id | BIGINT | x | định danh |
| order\_id | VARCHAR(255) | x | mã đơn hàng |
| customer\_id | INT |  | id của khách hàng |
| name | VARCHAR(255) | x | tên khách hàng |
| phone\_number | VARCHAR(255) | x | số điện thoại |
| address | VARCHAR(255) | x | địa chỉ |
| customer\_notes | LONGTEXT |  | lưu ý của khách hàng |
| notes | LONGTEXT |  | lưu ý của nhân viên |
| amount | DOUBLE(8, 2) | x | tổng tiền |
| score\_awards | DOUBLE(8, 2) | x | số điểm thưởng |
| admin\_id\_status\_shipped | INT |  | id nhân viên phê duyệt |
| admin\_id\_status\_delivered | INT |  | id nhân viên xác nhận |
| admin\_id\_status\_cancel | INT |  | id nhân viên hủy |
| status | INT | x | tình trạng giao dịch |
| created\_at | TIMESTAMP |  | thời gian tạo |
| updated\_at | TIMESTAMP |  | thời gian cập nhật lần cuối |

2.5.2.10. Bảng tin tức - sự kiện (posts)

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Not Null** | **Mô tả** |
| id | INT | x | định danh |
| title | VARCHAR(255) | x | tiêu đề |
| thumbnail | VARCHAR(255) | x | tên ảnh |
| description | LONGTEXT |  | mô tả ngắn |
| content | LONGTEXT | x | nội dung |
| slug | VARCHAR(255) | x | đường dẫn |
| admin\_name | VARCHAR(255) | x | tên nhân viên |
| post\_category\_id | INT | x | id của chủ đề |
| view\_count | INT | x | số lượt xem |
| created\_at | TIMESTAMP |  | thời gian tạo |
| updated\_at | TIMESTAMP |  | thời gian cập nhật lần cuối |

2.5.2.11. Bảng thẻ tin tức - sự kiện (tags)

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Not Null** | **Mô tả** |
| id | INT | x | định danh |
| name | VARCHAR(255) | x | tên thẻ tin tức - sự kiện |
| slug | VARCHAR(255) | x | đường dẫn |
| created\_at | TIMESTAMP |  | thời gian tạo |
| updated\_at | TIMESTAMP |  | thời gian cập nhật lần cuối |

2.5.2.12. Bảng chủ đề tin tức - sự kiện (post\_categories)

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Not Null** | **Mô tả** |
| id | INT | x | định danh |
| name | VARCHAR(255) | x | tên chủ đề |
| sort\_id | TINYINT | x | thứ tự hiển thị |
| slug | VARCHAR(255) | x | đường dẫn |
| created\_at | TIMESTAMP |  | thời gian tạo |
| updated\_at | TIMESTAMP |  | thời gian cập nhật lần cuối |

2.5.2.13. Bảng liên kết giữa tin tức - sự kiện và thẻ tin tức - sự kiện (post\_tags)

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Not Null** | **Mô tả** |
| id | INT | x | định danh |
| post\_id | INT | x | id của tin tức - sự kiện |
| tag\_id | INT | x | id của thẻ tin tức - sự kiện |
| created\_at | TIMESTAMP |  | thời gian tạo |
| updated\_at | TIMESTAMP |  | thời gian cập nhật lần cuối |

# **CHƯƠNG 3: CÀI ĐẶT HỆ THỐNG**