**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN**

****

**BÀI TẬP LỚN LẬP TRÌNH WEB MÃ NGUỒN MỞ**

**XÂY DỰNG WEBSITE BÁN ĐÈN TRANG TRÍ**

NGÀNH: CÔNG NGHỆ THÔNG TIN

CHUYÊN NGÀNH: KỸ THUẬT PHẦN MỀM

SINH VIÊN: **LÊ XUÂN THỊNH**

MÃ LỚP: **125217**

HƯỚNG DẪN: **THS. VŨ XUÂN THẮNG**

**HƯNG YÊN – 2024**

**NHẬN XÉT**

**Nhận xét của giảng viên hướng dẫn:**

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

**GIẢNG VIÊN HƯỚNG DẪN**

*(Ký và ghi rõ họ tên)*

**LỜI CAM ĐOAN**

Em xin cam đoan bài tập lớn “Xây dựng website bán đèn trang trí” là kết quả thực hiện của bản thân em dưới sự hướng dẫn của thầy Vũ Xuân Thắng.

Những phần sử dụng tài liệu tham khảo trong bài tập lớn đã được nêu rõ trong phần tài liệu tham khảo. Các kết quả trình bày trong bài tập lớn và chương trình xây dựng được hoàn toàn là kết quả do bản thân em thực hiện.

Nếu vi phạm lời cam đoan này, em xin chịu hoàn toàn trách nhiệm trước khoa và nhà trường.

*Hưng Yên, ngày … tháng … năm…..*

SINH VIÊN

…………………………………..

**LỜI CẢM ƠN**

Để có thể hoàn thành bài tập lớn này, lời đầu tiên em xin phép gửi lời cảm ơn tới bộ môn Công nghệ phần mềm, Khoa Công nghệ thông tin – Trường Đại học Sư phạm Kỹ thuật Hưng Yên đã tạo điều kiện thuận lợi cho em thực hiện bài tập lớn môn học này.

Đặc biệt em xin chân thành cảm ơn thầy Vũ Xuân Thắng đã rất tận tình hướng dẫn, chỉ bảo em trong suốt thời gian thực hiện bài tập lớn vừa qua.

Em cũng xin chân thành cảm ơn tất cả các Thầy, các Cô trong Trường đã tận tình giảng dạy, trang bị cho em những kiến thức cần thiết, quý báu để giúp em thực hiện được bài tập lớn này.

Mặc dù em đã có cố gắng, nhưng với trình độ còn hạn chế, trong quá trình thực hiện đề tài không tránh khỏi những thiếu sót. Em hi vọng sẽ nhận được những ý kiến nhận xét, góp ý của các Thầy giáo, Cô giáo về những kết quả triển khai trong bài tập lớn.

Em xin trân trọng cảm ơn!

**MỤC LỤC**

[DANH MỤC CÁC THUẬT NGỮ 7](#_Toc50641570)

[DANH MỤC CÁC BẢNG 8](#_Toc50641571)

[DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ 9](#_Toc50641572)

[CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI 10](#_Toc50641573)

[1.1. Lý do chọn đề tài 10](#_Toc50641574)

[1.2. Mục tiêu của đề tài 10](#_Toc50641575)

[1.2.1 Mục tiêu tổng quát 10](#_Toc50641576)

[1.2.2 Mục tiêu cụ thể 10](#_Toc50641577)

[1.3. Giới hạn và phạm vi của đề tài 10](#_Toc50641578)

[1.3.1 Đối tượng nghiên cứu 10](#_Toc50641579)

[1.3.2 Phạm vi nghiên cứu 11](#_Toc50641580)

[1.4. Nội dung thực hiện 11](#_Toc50641581)

[1.5. Phương pháp tiếp cận 11](#_Toc50641582)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 12](#_Toc50641583)

[2.1. Quy trình phát triển phần mềm 12](#_Toc50641584)

[2.2. Phương pháp phân tích thiết kế hướng đối tượng 12](#_Toc50641585)

[2.3. Thiết kế và lập trình font-end 12](#_Toc50641586)

[2.4. Tổng quan về Laravel 12](#_Toc50641587)

[2.5. Công nghệ thao tác dữ liệu Entity framework (hoặc ADO.NET) 12](#_Toc50641588)

[2.6. Mô hình thao tác dữ liệu 3 layer 12](#_Toc50641589)

[CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 13](#_Toc50641590)

[3.1 Đặc tả yêu cầu phần mềm 13](#_Toc50641591)

[3.1.1 Các yêu cầu chức năng 13](#_Toc50641592)

[3.1.2 Biểu đồ lớp thực thể 13](#_Toc50641593)

[3.1.3 Các yêu cầu phi chức năng 13](#_Toc50641594)

[3.2 Thiết kế hệ thống 13](#_Toc50641595)

[3.2.1 Thiết kế cơ sở dữ liệu 13](#_Toc50641596)

[3.2.2 Thiết kế lớp đối tượng 13](#_Toc50641597)

[3.2.3 Thiết kế giao diện 13](#_Toc50641598)

[CHƯƠNG 4: TRIỂN KHAI WEBSITE 14](#_Toc50641599)

[4.1 Triển khai các chức năng cho phân hệ người dùng 14](#_Toc50641600)

[4.1.1 Trang chủ (Ví dụ) 14](#_Toc50641601)

[4.1.2 Trang Xem Sản phẩm (Ví dụ) 15](#_Toc50641602)

[4.1.3 Trang Quản lý giỏ (Ví dụ) 15](#_Toc50641603)

[4.2 Triển khai các chức năng cho phân hệ quản trị nội dung (nếu có) 15](#_Toc50641604)

[4.3 Kiểm thử và triển khai ứng dụng 15](#_Toc50641605)

[4.3.1 Kiểm thử 15](#_Toc50641606)

[4.3.2 Đóng gói ứng dụng 15](#_Toc50641607)

[4.3.3 Triển khai ứng dụng 15](#_Toc50641608)

[KẾT LUẬN 16](#_Toc50641609)

[TÀI LIỆU THAM KHẢO 17](#_Toc50641610)

DANH MỤC CÁC THUẬT NGỮ

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Từ viết tắt | Cụm từ tiếng anh | Diễn giải |
| 1 | MVC | Model View Control | Mẫu thiết kế phần mềm |
| 2 | UML | Unified Modeling Language | Ngôn ngữ mô hình hóa được tiêu chuẩn hóa |
| 3 | PHP | Hypertext Preprocessor | Ngôn ngữ lập trình kịch bản hay một loại mã lệnh dùng để phát triển các ứng dụng viết cho máy chủ, mã nguồn mở |
| 4 | HTML | **Hypertext Markup Language** | Ngôn ngữ đánh dấu siêu văn bản |
| 5 | CSS | Cascading Style Sheets | Ngôn ngữ thiết kế đơn giản, xử lý một phần giao diện của trang web |

DANH MỤC CÁC BẢNG

Bảng 3.1: Mô tả chức năng của phân hệ quản trị nội dung........................30

Bảng 3.2: Mô tả chức năng của phân hệ người dùng.................................32

Bảng 3.3: Bảng danh sách Actor................................................................32

Bảng 3.4: Bảng danh sách Use case...........................................................32

Bảng 3.5: Thông tin mô tả Categeryes.......................................................59

Bảng 3.6: Thông tin mô tả product.............................................................60

Bảng 3.7: Thông tin mô tả Colors..............................................................60

Bảng 3.8: Thông tin mô tả sizes..................................................................60

Bảng 3.9: Thông tin mô tả users..................................................................61

Bảng 3.10: Thông tin mô tả Orders.............................................................61

Bảng 3.11: Thông tin OrderDetails.............................................................62

Bảng 3.12 Bảng mô tả thuộc tính bảng ProductTypes................................65

Bảng 3.13 Bảng mô tả thuộc tính bảng Products........................................65

Bảng 3.14 Bảng mô tả thuộc tính bảng Colos.............................................65

Bảng 3.15 Bảng mô tả thuộc tính bảng Sizes..............................................66

Bảng 3.16 Bảng mô tả thuộc tính bảng chi tiết ProductDetails...................66

Bảng 3.17 Bảng mô tả thuộc tính bảng ProductImages...............................67

Bảng 3.18 Bảng mô tả thuộc tính bảng Slider..............................................67

Bảng 3.19 Bảng mô tả thuộc tính bảng Orders ...........................................67

Bảng 3.20 Bảng mô tả thuộc tính bảng OrderDetails ..................................68

Bảng 3.21: Bảng mô tả thuộc tính bảng users...............................................68

DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ

Hình 3.2: Biểu đồ use case xem đèn.......................................................35

Hình 3.3: Use case khách hàng quản lý giỏ hàng – đặt hàng.................36

Hình 3.4: Biểu đồ use case chức năng quản lý loai đèn.........................38

Hình 3.5: Biểu đồ phân rã use case quản lý đèn.....................................41

Hình 3.6: Biểu đồ phân rã use case quản lý màu sắc.............................44

Hình 3.7: Biểu đồ phân rã use case quản lý kích thước..........................47

Hình 3.8: Biểu đồ Use case Quản lý tài khoản.......................................50

Hình 3.9: Biểu đồ phân rã use case quản lý đơn hàng............................53

Hình 3.10: Use case xem báo cáo, thống kê..........................................56

Hình 3.11: Biểu đồ phân rã use case đăng nhập....................................57

Hình 3.12: Biểu đồ lớp thực thể..............................................................59

Hình 3.13: Mô hình cơ sở dữ liệu...........................................................64

Hình 3.14: Biểu đồ tuần tự chức năng nhập thông tin loại đèn..............69

Hình 3.15: Biểu đồ tuần tự chức năng sửa thông tin loại đèn.................69

Hình 3.16: Biểu đồ tuần tự chức năng xóa thông tin loại đèn..................70

Hình 3.17: Biểu đồ tuân tự chức năng tìm kiếm thông tin loại đèn........70

Hình 3.18: Biểu đồ tuần tự chức năng nhập thông tin đèn.......................71

Hình 3.19: Biểu đồ tuần tự chức năng sửa thông tin đèn........................71

Hình 3.20 : Biểu đồ tuần tự chức năng xóa thông tin đèn........................72

Hình 3.21: Biểu đồ tuần tự chức năng tìm kiếm thông tin đèn................72

Hình 3.22: Biểu đồ tuần tự chức năng nhập thông tin khách hàng...........73

Hình 3.23: Biểu đồ tuần tự chức năng sửa thông tin khách hàng..............73

Hình 3.24: Biểu đồ tuần tự chức năng xóa thông tin khách hàng............74

Hình 3.25: Biểu đồ tuần tự chức năng tìm kiếm thông tin khách hàng...74

Hình 3.26: Biểu đồ tuần tự chức năng nhập thông tin đơn hàng ............75

Hình 3.27: Biểu đồ tuần tự chức năng sửa thông tin đơn hàng ..............75

Hình 3.28: Biểu đồ tuần tự chức năng xóa thông tin đơn hàng ..............76

Hình 3.29: Biểu đồ tuần tự chức năng tìm kiếm thông tin đơn hàng......76

Hình 3.30: Biểu đồ tuần tự chức năng thống kê hàng tồn......................77

Hình 3.31: Biểu đồ tuần tự chức năng thống kê – đèn bán chạy............77

Hình 3.32: Biểu đồ tuần tự chức năng báo cáo doanh thu......................78

Hình 3.33: Biểu đồ tuần tự chức năng thống kê chi tiết doanh thu.........78

Hình 3.34: Biểu đồ tuần tự chức năng đăng nhập...................................79

Hình 3.35: Biểu đồ tuần tự chức năng thay đổi thông tin cá nhân...........79

Hình 3.36: Biểu đồ tuần tự chức năng thay đổi mật khẩu........................80

Hình 3.37: Giao diện trang đăng nhập.....................................................81

Hình 3.38: Giao diện header của trang chủ..............................................81

# TỔNG QUAN VỀ ĐỀ TÀI

1.1. Lý do chọn đề tài

Trong thời đại công nghệ hiện đại hóa hiện nay, việc xây dựng thiết kế website bán đèn trang trí là rất quan trọng đối với các cửa hàng bán khăn phối đồ và phụ kiện trang sức phục vụ trong lĩnh vực quản lý kinh doanh thời trang. Nó đóng góp 1 phần rất lớn tới hiệu quả, tiến độ công việc cho các doanh nghiệp. Dưới đây là một số vai trò và tầm quan trọng của đề tài này:

* Tạo ra điểm tiếp cận với khách hàng: Website bán đèn trang trí là một kênh tiếp thị mới giúp các cửa hàng, doanh nghiệp tiếp cận với khách hàng một cách dễ dàng và tiện lợi hơn bao giờ hết. Khách hàng có thể truy cập bất cứ lúc nào, mọi lúc mọi nơi để có thể tham khảo đèn trang trí và thanh toán.
* Tăng cường thương hiệu: Một website bán đèn trang trí được thiết kế đẹp mắt và chuyên nghiệp được nâng cao và tạo sự tin tưởng cho khách hàng. Nó còn giúp doanh nghiệp quảng bá đèn trang trí và dịch vụ của mình.
* Tiết kiệm chi phí: Không phải thuê mặt bằng, giảm số lượng nhân viên, quản lý bán đèn trang trí.
* Thuận tiện và tiết kiệm thời gian.
* Tính tương tác cao: Khách hàng có thể đưa ra câu hỏi, đánh giá, góp ý và phản hồi,...

Đề tài xây dựng thiết kế website bán đèn trang trí là rất cấp thiết trong thời đại hiện nay. Lý do là các cửa hàng, doanh nghiệp cần phải cập nhật và áp dụng công nghệ vào hoạt động kinh doanh của mình để cạnh tranh với các đối thủ cùng ngành và đánh ứng nhu cầu của khách hàng. Các lợi ích mà việc xây dựng thiết kế website bán đèn trang trí mang lại cũng đã được đề cập ở trên. Ngoài ra, với sự phát triển của internet, số lượng người mua hàng trực tuyến ngày càng nhiều và dự báo sẽ tiếp tục tăng trong tương lai. Do đó, việc xây dựng một website bán đèn trang trí chuyên nghiệp và đẹp mắt sẽ thu hút được nhiều khách hàng, tăng doanh thu và lợi nhuận. Tóm lại, đề tài này rất cấp thiết và có thể mang lại nhiều lợi ích cho các cửa hàng và doanh nghiệp trong lĩnh vực quản lí kinh doanh thời trang hiện nay.

1.2. Mục tiêu của đề tài

1.2.1 Mục tiêu tổng quát

* Thiết kế website bán đèn trang trí chuyên nghiệp và thu hút khách hàng.
* Web có thể hiển thị và tương thích với mọi loại thiết bị điện tử như laptop, máy tính bảng, điện thoại di động.
* Mang lại hiệu quả, lợi ích kinh tế: giảm chi phí hoạt động, tăng doanh thu đối với các cửa hàng sử dụng website.
* Cung cấp đa dạng, phong phú các loại mặt hàng, tăng khả năng tương tác giữa người dùng và người bán hàng, giúp cho khách hàng có trải nghiệm mua hàng một cách thuận tiện và dễ dàng nhất. Đồng thời, cũng tạo ra một thị trường thương mại điện tử đa dạng.

1.2.2 Mục tiêu cụ thể

Mục tiêu của đề tài là xây dựng một website đèn trang trí chuyên nghiệp và thu hút khách hàng. Cụ thể:

* Thiết kế giao diện đẹp mắt, chuyên nghiệp, dễ sử dụng thân thiện với người dùng.
* Xây dựng hệ thống quản lý đèn trang trí và đặt hàng hiệu quả giúp khách hàng dễ dàng tìm kiếm và mua trên web.
* Thực hiện các chức năng của 1 website như người dùng, quản lý, mua hàng,...
* Tạo ra một trải nghiệm mua hàng trực tuyến tuyệt vời cho khách hàng tăng khả năng giữ chân khách hàng cũ và thu hút khách hàng mới.
* Tạo cho khách hàng một trải nghiệm thuận tiện nhất khi mua hàng.
* Có thể để khách hàng góp ý, nhắn tin với cửa hàng để tìm hiểu thông tin khăn và phụ kiện của cửa hàng.
* Về kiến thức:
* Trình bày được các hoạt động trong quy trình phát triển một sản phẩm phần mềm.
* Trình bày được phương pháp phân tích thiết kế hướng đối tượng và các biểu đồ UML tương ứng trong các giai đoạn đặc tả yêu cầu, phân tích, thiết kế phần mềm.
* Giải thích được mô hình thao tác dữ liệu 3 layer.
* Lựa chọn được các ngôn ngữ/công nghệ/công cụ thiết kế và lập trình web phía front-end (HTML, CSS) và back-end (Laravel).
* Về kỹ năng:
* Vận dụng được các kiến thức phân tích thiết kế hướng đối tượng để xây dựng các biểu đồ ca sử dụng, lớp thực thể, lớp phân tích, biểu đồ tuần tự, lớp thiết kế chi tiết cho hệ thống website phát triển;
* Xây dựng được các trigger, store procedure để thực thi các xử lý phía cơ sở dữ liệu cho ứng dụng;
* Xây dựng website động với công nghệ front-end: HTML, CSS và back-end : Laravel kết hợp với mô hình MVC theo thiết kế;
* Kiểm thử và triển khai được Website trên Internet;
* Quản lý tài liệu, quản lý phiên bản bằng các công cụ như Github,Gitlab, Google Drive.

1.3. Giới hạn và phạm vi của đề tài

1.3.1 Đối tượng nghiên cứu

Đối tượng nghiên cứu của đề tài xây dựng website bán đèn trang trí bao gồm:

* Các cửa hàng (doanh nghiệp) kinh doanh về lĩnh vực thời trang, đặc biệt là các cửa hàng bán khăn phối đồ và phụ kiện trang sức offline muốn chuyển đổi sang hình thức kinh doanh online.
* Khách hàng tiềm năng, khách hàng hiện tại của cửa hàng, những người quan tâm đến mua sắm trực tuyến.
* Nghiên cứu về các yêu cầu và tính năng cần thiết cho website bao gồm các tính năng giới thiệu sản phẩm, quản lý đơn hàng, quản lý giỏ hàng, quản lý danh sách khách hàng và các tính năng khác để đáp ứng nhu cầu của khách hàng và cửa hàng.
* Nghiên cứu về các yếu tố ảnh hưởng đến sự thành công của website, bao gồm tính năng, giao diện, trải nghiệm người dùng, độ tin cậy và độ bảo mật.

1.3.2 Phạm vi nghiên cứu

* Phạm vi không gian: Cửa hàng bán đèn trang trí.
* Phạm vi thời gian: 1/2024 đến 5/2024.
* Ý nghĩa khoa học và thực tiễn của đề tài: Phát triển website bán hàng trực tuyến cho cửa hàng.

1.4. Nội dung thực hiện

Đề tài thiết kế website bán đèn trang trí có thể bao gồm các nội dung thực hiện sau:

* Nguyên cứu và phân tích yêu cầu: Hiểu rõ nhu cầu và yêu cầu của khách hàng về giao diện, chức năng và trải nghiệm người dùng của website bán đèn trang trí.
* Thiết kế giao diện: Tạo ra thiết kế giao diện hấp dẫn, chuyên nghiệp và phù hợp với mục tiêu kinh doanh của website. Bao gồm thiết kế logo, lựa chọn màu sắc, hình ảnh và cách bố trí các phần tử trên trang.
* Phát triển website: Lập trình Web API, Công nghệ web và ứng dụng (ReactJS) để phát triển một hệ thống Website động đáp ứng các yêu cầu thực tế của dự án. Đảm bảo giao diện thân thiện, đáp ứng (responsive) và tương thích trên các thiết bị khác nhau.
* Xây dựng cơ sở dữ liệu: Thiết kế và triển khai cơ sở dữ liệu để lưu trữ thông tin về đèn trang trí, danh mục, khách hàng, đơn hàng và các thông tin liên quan khác.
* Kiểm thử và sửa lỗi: Tiến hành kiểm thử và sửa lỗi để đảm bảo website hoạt động ổn định, tương thích trên các trình duyệt khác nhau và cung cấp trải nghiệm người dùng tốt.
* Triển khai và duy trì: Triển khai website bán đèn trang trí và tiến hành theo dõi, duy trì và cập nhật thường xuyên để đảm bảo website hoạt động ổn định và cung cấp trải nghiệm tốt cho người dùng.

1.5. Phương pháp tiếp cận

Phương pháp tiếp cận trong xây dựng website bán đèn trang trí:

* Khảo sát thực tế hệ thống quản lí và giới thiệu tại website và cửa hàng đèn trang trí. Phân tích hệ thống sau khi đã khảo sát hệ thống quản lý để đưa ra những giải pháp, những ý tưởng mới cho việc xây dựng, thiết kế hệ thống phần mềm sau này.
* Tìm hiểu về thị trường bán đèn trang trí, xu hướng, đối tượng khách hàng mục tiêu và nhu cầu của họ.
* Sử dụng phương pháp đọc tài liệu, phương pháp phân tích, phương pháp thực nghiệm, phương pháp nghiên cứu bằng cách tiếp cận trực tiếp với người sử dụng thông qua việc khảo sát các trang web bán hàng trên thị trường, đề xuất lựa chọn giao diện, chức năng phù hợp với yêu cầu của khách hàng.

# CƠ SỞ LÝ THUYẾT

* 1. Quy trình phát triển phần mềm

Quy trình phát triển website là một quá trình phức tạp, đòi hỏi sự kết hợp giữa các kỹ năng thiết kế, phát triển, kiểm thử và vận hành. Quy trình này được thực hiện một cách chặt chẽ để đảm bảo tính ổn định, tin cậy và hiệu quả của trang web. Quy trình phát triển website bán đèn trang trí là quá trình thực hiện từ khâu lên ý tưởng cho đến khi đưa trang web vào sử dụng, bao gồm các bước chính sau:

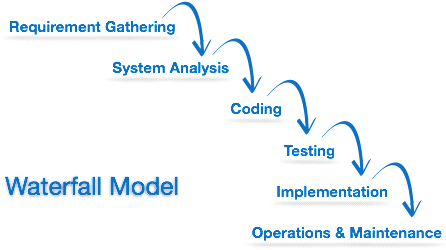
1. Khảo sát yêu cầu: Bước đầu tiên trong quy trình phát triển website bán đèn trang trí là khảo sát thu thập thông tin về yêu cầu của khách hàng, điều này bao gồm việc hiểu rõ về thương hiệu, sản phẩm khăn và phụ kiện, cũng như mục tiêu trang web giới thiệu các sản phẩm.
2. Phân tích yêu cầu: Phân tích yêu cầu để hiểu rõ các chức năng và tính năng cần thiết đưa ra phương án phát triển thích hợp cho trang web. Điều này bao gồm việc xác định trang web, cung cấp thông tin sản phẩm đèn trang trí, tương tác xã hội và nhiều yêu cầu khác.
3. Thiết kế giao diện: Sau khi đã hiểu được yêu cầu của khách hàng, đội ngũ thiết kế sẽ bắt đầu thiết kế giao diện cho website bán đèn trang trí, đảm bảo tính thẩm mỹ và trải nghiệm người dùng tốt.
4. Phát triển nội dung: Sau khi đã có thiết kế giao diện, đội ngũ phát triển nội dung sẽ bắt đầu tạo ra các trang web, nội dung cho website bán đèn trang trí.
5. Phát triển và kiểm thử: Tiếp theo, đội ngũ phát triển sẽ bắt đầu viết mã và phát triển các tính năng của website bán đèn trang trí, đồng thời tiến hành kiểm thử để đảm bảo tính ổn định và độ tin cậy của trang web.
6. Triển khai và vận hành: Sau khi đã hoàn thành phát triển và kiểm thử, trang web sẽ được triển khai và đưa vào sử dụng, đồng thời thực hiện các hoạt động vận hành để đảm bảo hoạt động ổn định và hiệu quả của website bán đèn trang trí.
7. Bảo trì và nâng cấp: Cuối cùng, đội ngũ phát triển và vận hành sẽ tiến hành bảo trì và nâng cấp website bán đèn trang trí để đảm bảo hoạt động liên tục và đáp ứng nhu cầu của khách hàng.

Các hoạt động chính trong phát triển phần mềm và vai trò của chúng

* 1. Phương pháp phân tích thiết kế hướng đối tượng

Các mô hình UML được xây dựng tương ứng trong các hoạt động phát triển phần mềm theo phương pháp hướng đối tượng.

* Waterfall model – Mô hình thác nước:



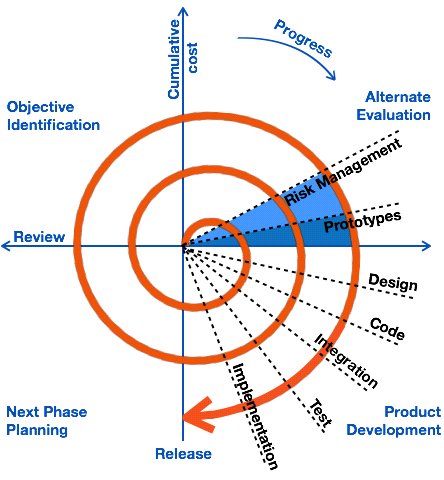
Hình 2.1 Mô hình thác nước

* Mô tả:
* Mô hình thác nước là mô hình áp dụng theo tính tuần tự của các giai đoạn phát triển phần mềm.
* Có nghĩa là: giai đoạn sau chỉ được phép thực hiện tiếp khi giai đoạn trước đã kết thúc.
* Không được quay lại giai đoạn trước để xử lý các thay đổi trong yêu cầu.
* Đây được coi mà mô hình phát triển phần mềm đầu tiên.
* Áp dụng: Thường được áp dụng cho các dự án không thường xuyên thay đổi về yêu cầu.
* Đặc điểm:
* Ưu điểm:
* Dễ sử dụng, dễ tiếp cận.
* Các giai đoạn và hoạt động được xác định rõ ràng.
* Xác nhận ở từng giai đoạn, đảm bảo phát hiện sớm lỗi.
* Nhược điểm:
* Rất khó để quay lại giai đoạn nào đó khi nó đã kết thúc.
* Ít tính linh hoạt và phạm vi điều chỉnh của nó khá là khó khăn, tốn kém.
* V-Shaped Model – Mô hình chữ V:



Hình 2.2 Mô hình chữ V

* Mô tả:
* Đây là mô hình mở rộng từ mô hình thác nước
* Thay thì di chuyển xuống theo tuần tự các bước thì quy trình sẽ đi theo hình chữ V.
* Áp dụng:
* Yêu cầu phần mềm phải xác định rõ ràng.
* Công nghệ phần mềm và các công cụ phải được tìm hiểu kỹ.
* Đặc điểm:
* Ưu điểm:
* Đơn giản, dễ sử dụng.
* Phân phối cụ thể theo mỗi giai đoạn.
* Thực hiện verification và validation sớm trong mỗi giai đoạn phát triển.
* Nhược điểm: Phạm vi điều chỉnh khá là khó khăn và tốn kém.
* Spriral Model – Mô hình xoắn ốc:



Hình 2.3 Mô hình xoắn ốc

* Mô tả:
* Là mô hình kết hợp giữa các tính năng của mô hình prototyping và mô hình thác nước.
* Mô hình xoắn ốc được ưa chuộng cho các dự án lớn, đắt tiền và phức tạp.
* Mô hình này sử dụng nhiều những giai đoạn tương tự như mô hình thác nước, về thứ tự, plan, đánh giá rủi ro, …
* Áp dụng: Thường được sử dụng cho các ứng dụng lớn và các hệ thống được xây dựng theo các giai đoạn nhỏ hoặc theo các phân đoạn
* Đặc điểm:
* Ưu điểm:
* Estimates (i.e. budget, schedule, etc.) trở nên thực tế hơn như là một quy trình làm việc, bởi vì những vấn đề quan trọng đã được phát hiện sớm hơn.
* Có sự tham gia sớm của developers
* Quản lý rủi ro và phát triển hệ thống theo phase
* Nhược điểm:
* Chi phí cao và thời gian dài để có sản phẩm cuối cùng
* Phải có kỹ năng tốt để đánh giá rủi ro và giả định.
  1. Thiết kế và lập trình font-end

HTML (HyperText Markup Language) và CSS (Cascading Style Sheets) là hai công nghệ cơ bản để thiết kế giao diện web. HTML được sử dụng để tạo cấu trúc và nội dung của trang web, trong khi CSS được sử dụng để định dạng và trang trí giao diện của trang web.

* + Ngôn ngữ HTML:

HTML (HyperText Markup Language) – Ngôn ngữ đánh dấu siêu văn bản được sử dụng để tạo các tài liệu có thể truy cập trên mạng. Tài liệu HTML được tạo nhờ dùng các thẻ và các phần tử của HTML. File được lưu trên máy chủ dịch vụ web với phần mở rộng “htm" hoặc ".html". Các trình duyệt sẽ đọc tập tin HTML và hiển thị chúng dưới dạng trang web. Các thẻ HTML sẽ được ẩn đi, chỉ hiển thị nội dung văn bản và các đối tượng khác: hình ảnh, media. Với các trình duyệt khác nhau đều hiển thị một tập HTML với 1 kết quả nhất định. Các trang HTML được gửi đi qua mạng internet theo giao thức HTTP. HTML không những cho phép nhưng thêm các đối tượng hình ảnh, âm thanh mà còn cho phép nhưng các kịch bản vào trong đó như các ngôn ngữ kịch bản như Javascript để tạo hiệu ứng động cho trang web. Để trình bày trang web hiệu quả hơn thi HTML cho phép sử dụng kết hợp với CSS. HTML không những cho phép nhưng thêm các đối tượng hình ảnh, âm thanh mà còn cho phép những các kịch bản vào trong đó như các ngôn ngữ kịch bản như Javascript để tạo hiệu ứng động cho trang web.

HTML là một chuẩn ngôn ngữ internet được tạo ra và phát triển bởi tổ chức World Wide Web Consortium còn được viết tắc là W3C. Trước đó thì HTML xuất bản theo chuẩn của RFC. HTML được tương thích với mọi hệ điều hành cùng các trình duyệt của nó. Khả năng dễ học, dễ viết là một ưu điểm của HTML không những vậy việc soạn thảo đòi hỏi hết sức đơn giản và thông thường sử dụng notepad là đã đủ. Hiện nay, phiên bản mới nhất của HTML là HTML5 với nhiều tính năng ưu việt so với các phiên bản cũ HTML cải tiến khá nhiều đặc biệt hỗ trợ mạnh mẽ các phần tử multimedia mà không cần các plugin. Một tập tin HTML bao gồm trong đó là các đoạn văn bản HTML, được tạo lên bởi các thẻ HTML. HTML5 nói chung mạnh mẽ hơn nhiều không chỉ về tốc độ và độ thích ứng cao mà chính là khả năng hỗ trợ API (Application Programming Interface - giao diện lập trình ứng dụng) và DOM (Document Object Model – các đối tượng thao tác văn bản).

* + Ngôn ngữ CSS:

CSS (Cascading Style Sheets) là một ngôn ngữ quy định cách trình bày cho các tài liệu viết bằng HTML, XHTML, XML, SVG, hay UML,... CSS quy định cách hiển thị của các thẻ HTML bằng cách quy định các thuộc tính của các thẻ đó (font chữ, kích thước, màu sắc...). Các đặc điểm kĩ thuật của CSS được duy trì bởi tổ chức W3C.

CSS có cấu trúc đơn giản và sử dụng các từ tiếng anh để đặt tên cho các thuộc tính. CSS khi sử dụng có thể viết trực tiếp xen lẫn vào mã HTML hoặc tham chiếu từ một file css riêng biệt. Và hiện nay CSS thường được viết riêng thành một tập tin với mở rộng là “.css”. Chính vì vậy mà các trang web có sử dụng CSS thì mã HTML sẽ trở nên ngắn gọn hơn. Ngoài ra có thể sử dụng 1 tập tin CSS đó cho nhiều website tiết kiệm rất nhiều thời gian và công sức. Một đặc điểm quan trọng đó là tính kế thừa của CSS do đó sẽ giảm được số lượng dòng code mà vẫn đạt được yêu cầu.

Tuy nhiên, đối với CSS thì các trình duyệt hiểu theo kiểu riêng của nó. Do vậy, việc trình bày 1 nội dung trên các trình duyệt khác nhau là không thống nhất.

CSS cung cấp hàng trăm thuộc tính trình bày dành cho các đối tượng với sự sáng tạo cao trong kết hợp các thuộc tính giúp mang lại hiệu quả.

* + Ngôn ngữ Javascrip:

JavaScript là ngôn ngữ lập trình được nhà phát triển sử dụng để tạo trang web tương tác. Từ làm mới bảng tin trên trang mạng xã hội đến hiển thị hình ảnh động và bản đồ tương tác, các chức năng của JavaScript có thể cải thiện trải nghiệm người dùng của trang web. Là ngôn ngữ kịch bản phía máy khách, JavaScript là một trong những công nghệ cốt lõi của World Wide Web. Ví dụ: khi duyệt internet, bất cứ khi nào bạn thấy quảng cáo quay vòng dạng hình ảnh, menu thả xuống nhấp để hiển thị hoặc màu sắc phần tử thay đổi động trên trang web cũng chính là lúc bạn thấy các hiệu ứng của JavaScript.

Quá trình thiết kế giao diện web bằng HTML, CSS bao gồm các bước sau:

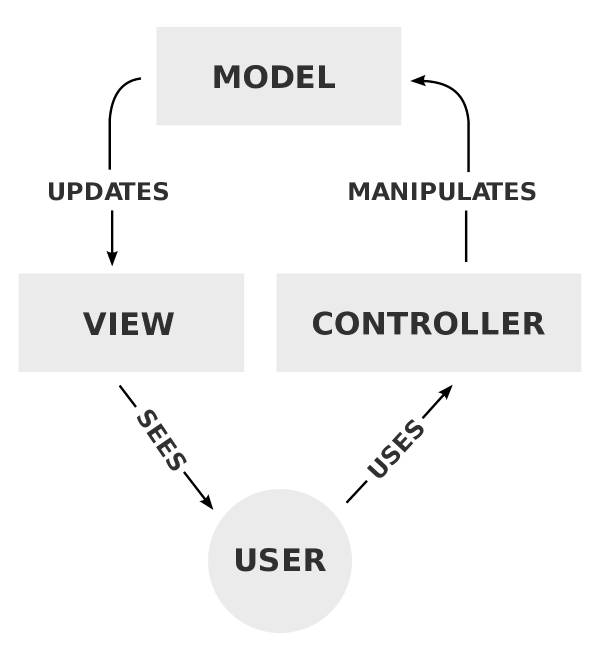
* Tạo cấu trúc HTML: Bước đầu tiên là tạo cấu trúc của trang web bằng HTML. Các thành phần cơ bản như tiêu đề, nội dung, chân trang, menu, đường dẫn và hình ảnh đều được tạo ra bằng HTML.
* Tạo stylesheet CSS: Sau khi đã tạo cấu trúc HTML, ta sẽ sử dụng CSS để định dạng và trang trí giao diện của trang web. Ta có thể sử dụng các thuộc tính CSS để thay đổi màu sắc, kích thước, kiểu chữ và vị trí của các thành phần trên trang web.
* Tạo layout của trang web: Bước này liên quan đến việc thiết kế bố cục của trang web. Ta sẽ sử dụng CSS để xác định kích thước và vị trí của các thành phần trên trang web, đồng thời thực hiện các kiểu định dạng như đối tượng, phần tử nằm trên cùng, các cột và vùng chứa.
* Tạo hiệu ứng và động thái: Sau khi đã hoàn tất bố cục của trang web, ta có thể sử dụng CSS để tạo các hiệu ứng và động thái cho trang web. Các hiệu ứng như chuyển động, mờ, nhấp nháy, và đổi màu sắc có thể giúp trang web trở nên sống động hơn.
* Kiểm tra và tối ưu: Sau khi đã hoàn tất thiết kế giao diện web, ta cần kiểm tra và tối ưu trang web để đảm bảo nó hoạt động tốt trên các trình duyệt khác nhau và trên các thiết bị khác nhau. Ta có thể sử dụng các công cụ như Google PageSpeed Insights để đánh giá tốc độ tải trang web và các yếu tố khác có thể ảnh hưởng đến trải nghiệm người dùng.
  1. Tổng quan về Laravel

Laravel được hiểu đơn giản làm một framework PHP dùng để xây dựng các ứng dụng web trên nhiều nền tảng khác nhau. Laravel cung cấp cho nhà phát triển một thư viện khổng lồ chứa các tính năng đã được lập trình sẵn. Nhờ vào đó mà người dùng xây dựng trang web trở nên nhanh chóng và đơn giản hơn do giảm số lượng và thời gian code. Các chức năng mà Laravel cung cấp có thể dễ dàng mở rộng và bảo trì. Đồng thời, các nhà phát triển còn có thể thêm thắt chức năng vào ứng dụng phần mềm của mình một cách liền mạch nhờ vào hệ thống đóng gói module và quản lý thắt chặt.

**Laravel không phải là ngôn ngữ lập trình, nó dùng ngôn ngữ kịch bản (scripting language)**. Dù hai ngôn ngữ này có mối liên quan với nhau, nhưng nó vẫn có những đặc điểm riêng biệt về hiệu suất và tính dễ sử dụng. Ngôn ngữ kịch bản là ngôn ngữ thông dịch, dành riêng cho nền tảng. Ngược lại, [các ngôn ngữ lập trình](https://bkhost.vn/blog/cac-ngon-ngu-lap-trinh-pho-bien-nhat-hien-nay/) là các ngôn ngữ biên dịch, không khả thi cho nền tảng. Tùy vào ứng dụng mà người dùng làm việc thì ngôn ngữ kịch bản sẽ được thông dịch ra ngôn ngữ phù hợp bằng trình biên dịch.

[Framework](https://bkhost.vn/blog/framework/) PHP là các thư viện chứa các tính năng được cấu hình sẵn, các nhà phát triển có thể tích hợp các tính năng mà [Framework PHP](https://bkhost.vn/blog/cac-frameworks-tot-nhat-cho-php-hien-nay/) cũng cấp vào ứng dụng của mình, từ đó giảm tải được số lượng code, giảm thời gian và tăng độ bảo mật. Ngoài ra, nó ít khi yêu cầu bảo trì hơn các ứng dụng được xây dựng từ đầu khác.

Ngoài các Framework PHP cơ bản, nó còn cung cấp framework thuộc dạng hướng đối tượng. Nếu bạn có kiến thức cơ bản về các khái niệm như lớp, đối tượng và kế thừa thì việc tìm hiểu và sử dụng framework này là hoàn toàn có thể.



Nó hoạt động theo mô hình **MVC ( Model-View-Controller**). Đây là mô hình [dữ liệu](https://bkhost.vn/blog/data-du-lieu/) ứng dụng và các logic liên quan.

## Chức năng của Laravel:

### Xử lý đường dẫn (route)

Laravel có khả năng xử lý tuyến đơn giản nhờ vào việc sử dụng tên thay vì đường dẫn. Đồng thời nó giúp duy trì ứng dụng dễ dàng bởi tên tuyến có thể thay đổi tại vị trí nhất định thay vì thay đổi trong cả quá trình xây dựng. Tất cả các tuyến giao diện web trong một ứng dụng Laravel được đăng ký trong tệp **ROUTES/Web.php.**

### Bảo mật

Laravel có một số tính năng bảo mật bao gồm xác thực người dùng, ủy quyền vai trò người dùng, xác minh email, dịch vụ [mã hóa](https://bkhost.vn/blog/ma-hoa/), băm [mật khẩu](https://bkhost.vn/blog/password/) và tính năng đặt lại mật khẩu.

### Migration

Laravel cung cấp kiểm soát phiên bản cho cơ sở dữ liệu ứng dụng bằng cách sử dụng Migration. Migration theo dõi các cơ sở dữ liệu đã được sửa đổi theo thời gian, giúp phá hủy hoặc tạo lại cơ sở dữ liệu dễ dàng hơn khi cần thiết.

### Tạo khuôn mẫu

Laravel sử dụng Blade làm công cụ tạo khuôn mẫu PHP, giúp phân chia nghiệp vụ từ khuôn mẫu HTML, giúp dễ bảo trì hơn. So với các công cụ tạo khuôn mẫu khác thì Blade chiếm ưu thế hơn bởi nó nhiều tính năng đồng thời cho phép sử dụng mã PHP thuần túy, các công cụ khác thì không.

### Sessions (Phiên)

Laravel lưu trữ thông tin người dùng theo yêu cầu ở dạng phiên. Cookie là một ví dụ điển hình.

* Xác thực dữ liệu

Laravel giúp việc xác thực dữ liệu người dùng mới trở nên đơn giản . Laravel bao gồm một số quy tắc xác thực dữ liệu, với các thông báo lỗi có thể tùy chỉnh.

### Xử lý bộ nhớ cache

Nhằm tối ưu hóa thời gian xử lý của ứng dụng, Laravel đã tích hợp bộ nhớ đệm. Trong đó là [API](https://bkhost.vn/blog/api/), ngoài ra còn có bên thứ ba như [Memcached](https://bkhost.vn/blog/memcached-la-gi/) và [Redis](https://bkhost.vn/blog/redis/).

### Xử lý lỗi

Laravel hỗ trợ thiết lập xử lý lỗi tự động, khi các ứng dụng chạy nếu có lỗi, chế độ gỡ lỗi sẽ gửi thông báo chi tiết về các lỗi xảy ra.

### Kiểm thử

Để đảm bảo chất lượng sản phẩm, Laravel hỗ trợ các loại kiểm thử như: kiểm thử đơn vị, kiểm tra các phần nhỏ, tách biệt của mã ứng dụng, cũng như kiểm tra tính năng, kiểm tra các phần mã lớn hơn và chức năng cấp cao hơn.

### Quản lý tệp và lưu trữ

Laravel sử dụng gói Flysystem PHP, có thể làm việc với nhiều hệ thống tệp khác nhau từ hệ thống tệp cục bộ đến lưu trữ dựa trên đám mây như [Amazon](https://bkhost.vn/blog/amazon-la-gi/) S3. Laravel cũng cung cấp tính năng truyền tệp với Giao thức truyền tệp [SSH](https://bkhost.vn/blog/giao-thuc-ssh-la-gi/) ([SFTP](https://bkhost.vn/blog/sftp/)).

### E-mail

Laravel bao gồm một API email dựa trên thư viện SwiftMailer, cho phép gửi email thông qua một dịch vụ được lựa chọn. Laravel hỗ trợ tệp đính kèm email và email chờ.

### Thông báo

Laravel hỗ trợ gửi thông báo qua một số kênh, cho dù là các kênh nổi tiếng như SMS hay Slack, hoặc sử dụng các kênh do cộng đồng Laravel phát triển.

* 1. Entity framework

Entity Framework (EF) được giới thiệu lần đầu tiên bởi Microsoft vào năm 2008. Ban đầu, EF được phát triển làm một phần của dự án ADO.NET Entity Framework, nhằm cung cấp một giải pháp ORM (Object-Relational Mapping) tiên tiến cho .NET Framework.

Entity Framework là một công cụ phát triển ứng dụng dành cho .NET Framework, cho phép lập trình viên làm việc với cơ sở dữ liệu quan hệ bằng cách sử dụng các đối tượng và truy vấn LINQ (Language Integrated Query). Với Entity Framework, việc tạo, truy vấn và cập nhật cơ sở dữ liệu trở nên đơn giản hơn bằng cách sử dụng các đối tượng trong mã C# hoặc VB.NET thay vì viết các truy vấn SQL trực tiếp. Điều này giúp giảm thiểu mã lập trình và tăng tính bảo mật, cùng với khả năng mở rộng và quản lý dữ liệu dễ dàng. Entity Framework hiện nay là một framework mạnh mẽ cho việc phát triển ứng dụng web, được hưởng lợi từ sự hỗ trợ đông đảo của cộng đồng người dùng.

Entity Framework (EF) có một số tính năng nổi bật và lợi ích đáng kể, bao gồm:

1. Object-Relational Mapping (ORM): EF cung cấp ORM, giúp ánh xạ dữ liệu từ cơ sở dữ liệu quan hệ thành các đối tượng trong mã lập trình. Điều này giúp lập trình viên làm việc với dữ liệu dễ dàng hơn và tránh việc phải viết truy vấn SQL trực tiếp.
2. Linh hoạt trong cấu hình: EF cho phép lập trình viên tuỳ chỉnh và cấu hình các khía cạnh của quá trình ánh xạ đối tượng-từ cơ sở dữ liệu. Lập trình viên có thể tùy chỉnh tên bảng, tên cột, quan hệ và quy tắc liên quan đến cơ sở dữ liệu.
3. Truy vấn LINQ: EF hỗ trợ việc sử dụng ngôn ngữ truy vấn tích hợp (LINQ) để truy vấn dữ liệu. LINQ cung cấp cú pháp gọn gàng và mạnh mẽ để viết truy vấn dữ liệu trong mã lập trình, giúp làm việc với dữ liệu trở nên dễ dàng và rõ ràng.
4. Migrations: EF cung cấp tính năng migrations, cho phép quản lý và duy trì phiên bản cơ sở dữ liệu trong quá trình phát triển ứng dụng. Migrations tự động tạo và áp dụng các thay đổi cơ sở dữ liệu, giúp đảm bảo sự đồng bộ giữa mô hình dữ liệu và cơ sở dữ liệu thực tế.
5. Hỗ trợ nhiều nhà cung cấp cơ sở dữ liệu: EF hỗ trợ nhiều hệ quản trị cơ sở dữ liệu như SQL Server, MySQL, SQLite, PostgreSQL và Oracle. Điều này cho phép lập trình viên lựa chọn và chuyển đổi giữa các cơ sở dữ liệu một cách linh hoạt mà không cần thay đổi mã lập trình chính.
6. Cache dữ liệu: EF cung cấp khả năng cache dữ liệu, giúp cải thiện hiệu suất và giảm tải cho cơ sở dữ liệu. Bằng cách lưu trữ và truy xuất dữ liệu từ bộ nhớ cache, EF giúp giảm số lượng truy cập cơ sở dữ liệu và tăng tốc độ thực thi của ứng dụng.
   1. Mô hình thao tác dữ liệu MVC

Mô hình MVC (viết tắt của cụm từ Model-View-Controller) là phân bổ source code thành 3 phần Model, View, Controller, mỗi thành phần có một nhiệm vụ riêng biệt và xử lý độc lập với các thành phần khác, khả năng áp dụng linh hoạt để tạo ra các ứng dụng mạnh mẽ và linh hoạt, được sử dụng rộng rãi trong phát triển ứng dụng.

MVC gồm có 3 thành phần chính là Model, View và Controller. Mặc dù đảm nhận nhiệm vụ độc lập, sự thiếu vắng hoặc sai sót từ một trong ba bộ phận trên sẽ khiến cho toàn bộ quá trình phân tích bị ảnh hưởng. Cho nên, vai trò của chúng là như nhau, và vô cùng quan trọng. Cụ thể là:

* Model (Dữ liệu)

Có khả năng lưu trữ toàn bộ dữ liệu, là cầu nối của hai thành phần còn lại là View và Controller. Một số loại dữ liệu thông dụng có thể kể đến như là file XML, cơ sở dữ liệu hoặc các đối tượng cơ bản khác (nhân vật anime trong game, ảnh, video,..).

* View (Giao diện)

Là giao diện (theme) cho người lập trình, là phương tiện cho phép hiển thị thước đo đối với các đối tượng được phát triển trong ứng dụng. Chẳng hạn, cửa sổ, nút văn bản, hình ảnh,...

* Controller (Bộ điều khiển)

Có chức năng xử lý yêu cầu của người dùng thông qua View. Một controller cơ bản bao gồm cả View lẫn Model, và nhận diện các input để điều chỉnh liên tục.

* Ưu điểm của MVC:
* Tái sử dụng và mở rộng dễ dàng

Với cấu trúc phân tách rõ ràng của MVC, việc tái sử dụng và mở rộng code trở nên dễ dàng. Bạn có thể thay đổi một thành phần (Model, View, hoặc Controller) mà không ảnh hưởng đến các thành phần khác. Điều này giúp giảm khối lượng công việc và thời gian cần thiết để thực hiện các thay đổi và bảo trì ứng dụng.

* Mượt

Sử dụng ít băng thông (bandwidth), đảm bảo website vận hành mượt mà và tăng tốc độ truy cập.

* Dễ bảo trì trong điều kiện cần thiết

Được phân chia rõ ràng và tách biệt, MVC tạo điều kiện thuận lợi để tìm và sửa các lỗi. Khi một lỗi xảy ra, bạn có thể xác định nhanh chóng thành phần gây ra lỗi và tìm hiểu vấn đề một cách tập trung. Từ đó mà bạn dễ dàng cải thiện quá trình bảo trì và giảm thời gian khắc phục sự cố.

* Độc lập dữ liệu và giao diện người dùng

Với MVC, dữ liệu và giao diện người dùng được phân tách hoàn toàn. Người phát triển có thể thay đổi giao diện người dùng mà không ảnh hưởng đến dữ liệu và ngược lại. Điều này cho phép việc phát triển giao diện phù hợp với yêu cầu thay đổi mà không cần thay đổi cấu trúc dữ liệu.

* Quản lý luồng điều khiển hiệu quả

Mô hình giúp quản lý luồng điều khiển và logic ứng dụng. Controller xử lý các yêu cầu từ người dùng và điều hướng công việc tương ứng đến Model và View. Công cụ MVC giữ cho code dễ hiểu, dễ bảo trì và quản lý.

* Quy trình phát triển nhanh hơn

MVC hỗ trợ phát việc phát triển nhanh chóng và song song. Nếu một mô hình MVC được dùng để phát triển bất kỳ ứng dụng web cụ thể nào, một lập trình viên có thể làm việc trên View và một developer khác có thể làm việc với Controller để tạo logic nghiệp vụ cho ứng dụng web đó. Do đó, ứng dụng mô hình MVC có thể được hoàn thành nhanh hơn ba lần so với các ứng dụng mô hình khác.

* Khả năng cung cấp nhiều chế độ view

Trong mô hình MVC, bạn có thể tạo nhiều View cho chỉ một mô hình. Ngày nay, nhu cầu có thêm nhiều cách mới để truy cập ứng dụng và đang ngày càng tăng. Do đó, việc sử dụng MVC để phát triển chắc chắn là một giải pháp tuyệt vời. Hơn nữa, với phương pháp này, việc nhân bản code rất hạn chế. Vì nó tách biệt dữ liệu và logic nghiệp vụ khỏi màn hình.

* Các sửa đổi không ảnh hưởng đến toàn bộ mô hình

Đối với bất kỳ ứng dụng web nào, người dùng có xu hướng thay đổi thường xuyên. Bạn có thể quan sát thông qua những thay đổi thường xuyên về màu sắc, font chữ, bố cục màn hình. Hay là thêm hỗ trợ thiết bị mới cho điện thoại hay máy tính bảng…Việc thêm một kiểu view mới trong MVC rất đơn giản. Vì phần Model không phụ thuộc vào phần View. Do đó, bất kỳ thay đổi nào trong Model sẽ không ảnh hưởng đến toàn bộ kiến trúc.

* MVC Model trả về dữ liệu mà không cần định dạng

MVC pattern có thể trả về dữ liệu mà không cần áp dụng bất kỳ định dạng nào. Do đó, các thành phần giống nhau có thể được sử dụng với bất kỳ giao diện nào.

Ví dụ: tất cả loại dữ liệu đều có thể được định dạng bằng HTML. Ngoài ra, nó cũng có thể được định dạng bằng Macromedia Flash hay Dream Viewer.

* Hỗ trợ kỹ thuật Asynchronous

Kiến trúc MVC có thể được tích hợp với cả JavaScript Framework. Có nghĩa là, các ứng dụng MVC có thể hoạt động ngay cả với các file PDF, trình duyệt riêng cho web hay các widget trên desktop. Ngoài ra, MVC cũng hỗ trợ kỹ thuật Asynchronous, giúp các developer phát triển các ứng dụng có thể load rất nhanh.

* Nền tảng MVC thân thiện với SEO

Nền tảng MVC hỗ trợ phát triển các trang web thân thiện với SEO. Bằng nền tảng này, bạn có thể dễ dàng phát triển các URL thân thiện với SEO để tạo ra nhiều lượt truy cập hơn.

# PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

* 1. Đặc tả yêu cầu phần mềm
     1. Các yêu cầu chức năng

Xây dựng Website cho phép giới thiệu và đặt hàng online cho cửa hàng giới thiệu quần áo. Website cho phép khách hàng có thể xem và đặt hàng online, nhân viên của cửa hàng có thể quản lý thông tin của Website và quản lý hoạt động đặt và giao hàng. Các yêu cầu của hệ thống được mô tả như sau:

* + - 1. Chức năng của phân hệ quản trị nội dung (nếu có)

Bảng 3. Mô tả chức năng của phân hệ quản trị nội dung

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên yêu cầu** | **Mô tả yêu cầu** |
| 1 | Đăng nhập của người quản trị | * Xác thực danh tính người quản trị * Người quản trị thực hiện đăng nhập vào hệ thống (điền user, password). Nếu user, password khớp với dữ liệu được lưu thì thông báo đăng nhập thành công, ngược lại đưa ra thông báo yêu cầu đăng nhập lại |
| 2 | Quản lí loại đèn trang trí | * Hiển thị các thông tin của loại đèn trang trí đèn trang trí, người quản lí có thể cập nhật thêm sửa xóa loại đèn trang trí trên website |
| 3 | Quản lí đèn trang trí | * Hiển thị các thông tin của đèn trang trí đèn trang trí, người quản lí có thể cập nhật thêm sửa xóa loại đèn trang trí trên website |
| 4 | Quản lý màu sắc | - Nhập thông tin màu sắc  - Cập nhật thông tin màu sắc  - Xóa thông tin màu sắc |
| 5 | Quản lý kích thước | - Nhập khích thước  - Cập nhật kích thước  - Xóa kích thước |
| 6 | Quản lý đơn hàng | - Nhập thêm thông tin đươn hàng  - Sửa thông tin đơn hàng  - Xóa thông tin đơn hàng  - Tìm kiếm đơn hàng |
| 7 | Quản lý bán hàng | - Xem thông tin các đơn đặt hàng  -Xác thực đơn hàng  -Đơn hàng đã xác thực  - Đơn hàng đã bán  -Hủy đơn hàng |
| 8 | Thống kê, báo cáo | * Hiển thị tổng quan về các chỉ số quan trọng như doanh thu, đơn hàng, đèn trang trí, đơn hoàn trả theo ngày, tuần tháng. Giúp người quản lý có thể biết được hiệu suất kinh doanh và đưa ra chiến lược hiệu quả |

* + - 1. Chức năng của phân hệ người dùng.

Bảng 3. Mô tả chức năng phân hệ người dùng

|  |  |  |
| --- | --- | --- |
| **STT** | **Chức năng** | **Mô tả** |
| 1 | Xem đèn | * Người dùng xem thông tin về đèn trên website. Thông tin đèn bao gồm: tên đèn, loại đèn, mô tả, giá, ảnh… |
| 2 | Tìm kiếm | * Người dùng có thể tìm kiếm đèn mình muốn theo mã, tên, … |
| 3 | Mua hàng | * Khi người dùng xem thông tin của sách hoặc xem thông tin chi tiết của sách trên website. Nếu người dùng muốn đặt mua thì có thể thực hiện chức năng thêm vào giỏ hàng. |
| 4 | Quản lý giỏ hàng | * Người dùng quản lý thông tin của các mặt hàng đã chọn mua như sửa xoá các thông tin trong giỏ hàng đã chọn sẽ chọn các mặt hàng mình muốn mua cho vào giỏ hàng của mình. |
| 5 | Đặt hàng | * Khi người dùng muốn mua hàng sẽ vào phần giỏ hàng chọn đặt mua, người dùng nhập thông tin của mình và xác nhận đặt hàng. |

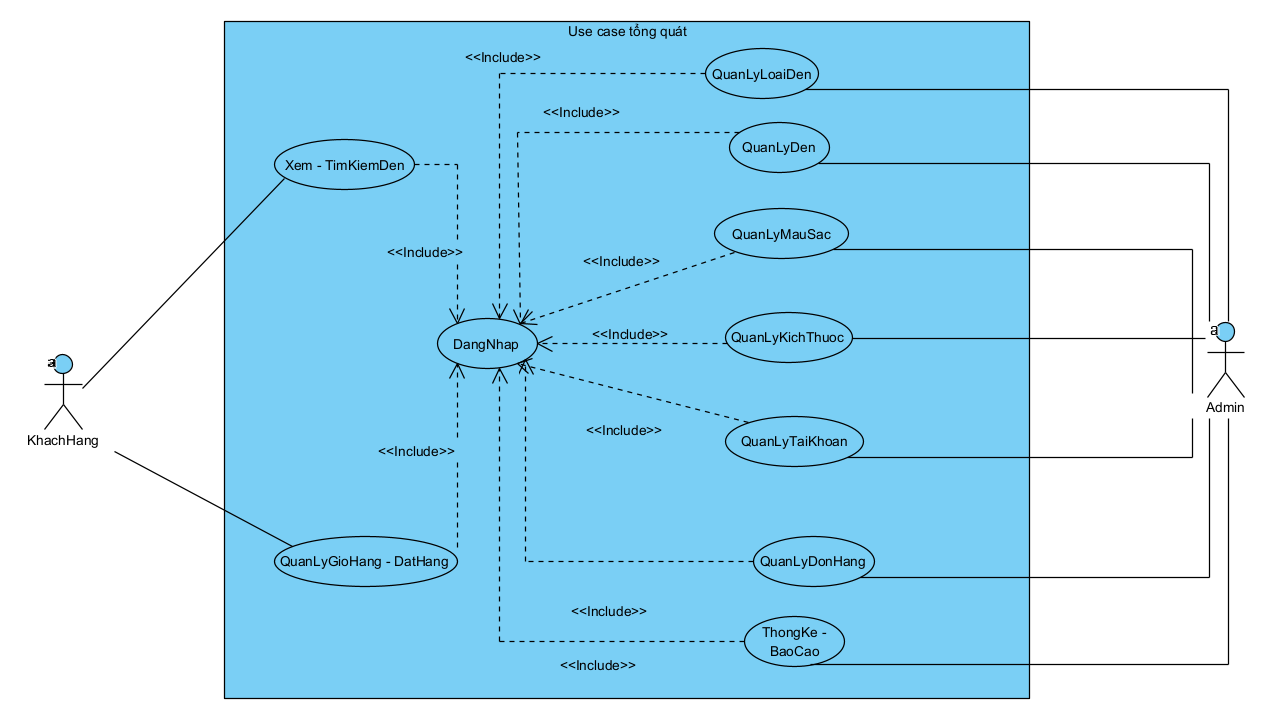
* + - 1. Biểu đồ use case tổng quát

Bảng 3. : Bảng danh sách Actor

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên Actor** | **Ý nghĩa/Ghi chú** |
| 1 | Khách hàng | Xem thông tin đèn trang trí, quản lý giỏ hàng, thực hiện được các chức năng đăng kí và đăng nhập. |
| 2 | Admin | Quản lý tất cả thông tin và dữ liệu của website, quản lý thông tin tài khoản trong ứng dụng. |

Bảng 3.: Bảng danh sách Use case

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên Use case** | **Ý nghĩa/Ghi chú** |
| 1 | Xem – tìm kiếm thông tin đèn trang trí | Cho phép khách hàng xem hình ảnh, giá và mô tả chi tiết về đèn trang trí. |
| 2 | Quản lý giỏ hàng – Đặt hàng | Cho phép lễ khách hàng thêm, sửa, xóa đèn trang trí trong giỏ hàng. |
| 3 | Quản lý loại đèn trang trí | Quản lý thông tin các loại sđèn trang trí hiện có trên website. |
| 4 | Quản lý đèn trang trí | Quản lý thông tin các đèn trang trí hiện có trên website. |
| 5 | Quản lý màu sắc | Quản lý thông tin màu sắc sản phẩm. |
| 6 | Quản lý kích thước | Quản lý thông tin màu kích thước phẩm. |
| 7 | Quản lý tài khoản | Quản lý thông tin tài khoản của người dùng |
| 7 | Quản lý đơn hàng | Quản lý thông tin đươn hàng của khách đã đặt. |
| 8 | Thống kê, báo cáo | Xem thông kê, báo cáo doanh thu |

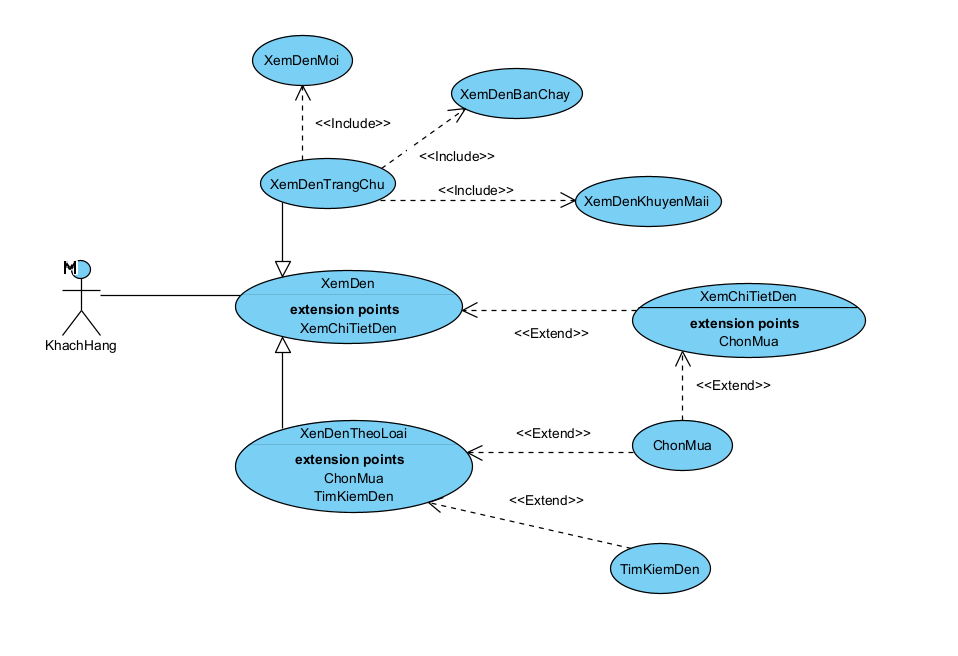


Hình 3.: Use case thổng quát hệ thống giới thiệu đền trang trí

* **Đặc tả biểu đồ ca sử dụng**

**\*) Tác nhân khách hàng**

* Use case quản lý xem – tìm kiến thông tin đèn:



Hình 3.: Biểu đồ use case xem đèn

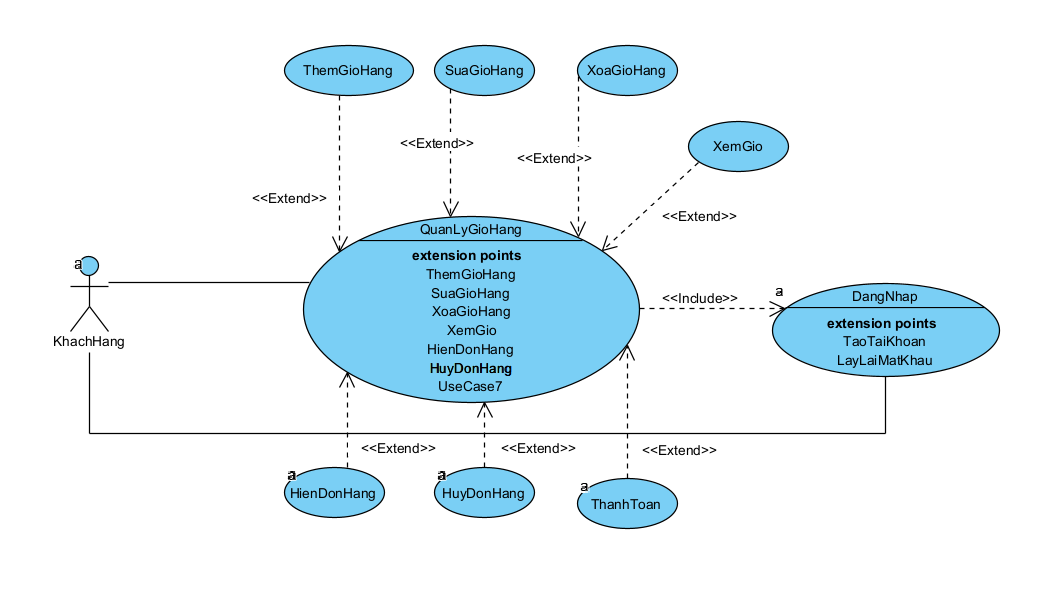
* Mô tả luồng sự kiện của ca sử dụng xem đèn trang trí theo loại:

|  |
| --- |
| **Use case name:** Xem đèn theo loại  **Brief description:** Người dùng chọn chức năng xem đèn theo loại  **Actors:** Người dùng  **Precondition:** Không có yêu cầu đặc biệt nào  **Post – condition:** Hệ thống hiển thị danh sách các đèn trang trí theo loại cần xem  **Trigger:** Người dùng yêu cầu xem đèn theo loại |
| **Basic flow** |
| 1. Người dùng yêu cầu xem đèn theo loại từ giao diện của trang Web  3. SYSTEM Hệ thống lấy về đèn theo loại hiển thị lên trang danh sách các đèn trang trí theo loại. |
| **Alternate flows** |

* Mô tả luồng sự kiện của ca sử dụng xem chi tiết đèn:

|  |
| --- |
| **Use case name:** Xem chi tiết đèn trang trí  **Brief description:** Người dùng chọn chức năng xem chi tiết đèn trang trí  **Actors:** Người dùng  **Precondition:** Không có yêu cầu đặc biệt nào  **Post – condition:** Hệ thống hiển thị thông tin chi tiết của đèn trang trí cần xem  **Trigger:** Người dùng yêu cầu xem thông tin chi tiết đèn trang trí |
| **Basic flow** |
| 1. Người dùng yêu cầu xem thông tin chi tiết đèn từ giao diện của trang Web  2. SYSTEM Hệ thống lấy về thông tin chi tiết đèn hiển thị lên trang chi tiết đèn |
| **Alternate flows** |

* Use case quản lý giỏ hàng – đặt hàng:



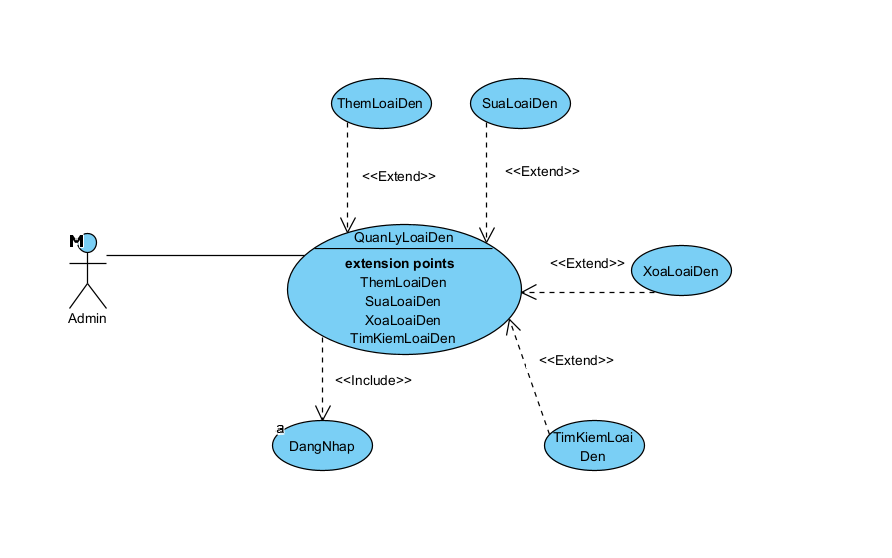
Hình 3.: Use case khách hàng quản lý giỏ hàng – đặt hàng

* Mô tả luồng sự kiện của ca sử dụng đặt hàng:

|  |
| --- |
| **Use case name:** Đặt hàng  **Brief description:** Người dùng yêu cầu đặt hàng  **Actors:** Khách hàng  **Precondition:** Không có yêu cầu đặc biệt nào  **Post – condition:** Nếu đặt hàng thành công, một đơn hàng sẽ được lưu vào hệ thống. Trong trường hợp khác, hệ thống yêu cầu hiển thị lỗi  **Trigger:** *Người dùng yêu cầu đặt hàng khi thực hiện Use Case xem giỏ hàng* |
| **Basic flow** |
| 1. Người dùng yêu cầu đặt hàng  2. SYSTEM Hệ thống hiển thị giao diện đặt hàng  3. Người dùng nhập thông tin mua hàng gồm thông tin về người dùng, địa chỉ giao hàng, số điện thoại nhận hàng  4. Người dùng yêu cầu đặt hàng, nếu không thì chuyển sang luồng phụ 4a  5. SYSTEM Hệ thống kiểm tra thông tin đặt hàng  6. SYSTEM Hệ thống lưu thông tin khách hàng  7. SYSTEM Hệ thống sinh mã đơn hàng tự động[[1]](#footnote-1)  8. SYSTEM Hệ thống lưu thông tin đơn hàng và danh sách các mặt hàng đã đặt vào chi tiết đơn hàng  9. SYSTEM Hệ thống hiển thị đặt hàng thành công, nếu không thành công thì chuyển sang ngoại lệ 9a. |
| **Alternate flows** |
| 4a. Người dùng không đặt hàng  1. Người dùng muốn ngưng giao dịch, chọn nút hủy  2. SYSTEM Hệ thống hiển thị giao diện giỏ hàng  5a. Thông tin đặt hàng không hợp lệ  1. SYSTEM Hệ thống hiển thị thông báo việc tạo đơn hàng không thành công  2. SYSTEM Hệ thống hiển thị form nhập thông tin và các trường nhập sai, quay lại bước 4  3. Khách hàng nhập lại thông tin |

**\*) Tác nhân admin**

* Use case quản lý loại đèn:



Hình 3.: Biểu đồ use case chức năng quản lý loai đèn

* Tác nhân: Admin.
* Mô tả: Đây là chức năng dành cho người quản lý và nhân viên quản lý loại đèn trang trí.
* Điều kiện cần: Người dùng cần đăng nhập vào hệ thống.
* Điều kiện đủ: Có đủ quyền hạn để thực hiện.
* Dòng sự kiện chính

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| 1. Admin yêu cầu chức năng quản lý loại đèn trang trí | 2. Hiện thị giao diện lựa chọn các chức năng quản lý loại đèn trang trí |
| 3.Admin yêu cầu chức năng thêm loại đèn trang trí | 4.Hệ thống hiển thị giao diện trang quản lý thêm loại đèn trang trí |
| 5.Admin yêu cầu chức năng sửa loại đèn trang trí |  |
| 7. Admin yêu cầu chức năng xóa loại đèn trang trí | 8. Hệ thống hiện thị giao diện quản lý chức năng xóa thông tin loại đèn trang trí |
| 9. Admin yêu cầu chức năng tìm kiếm loại đèn trang trí | 10. Hệ thống hiện thị giao diện quản lý chức năng tìm kiếm thông tin loại đèn trang trí |

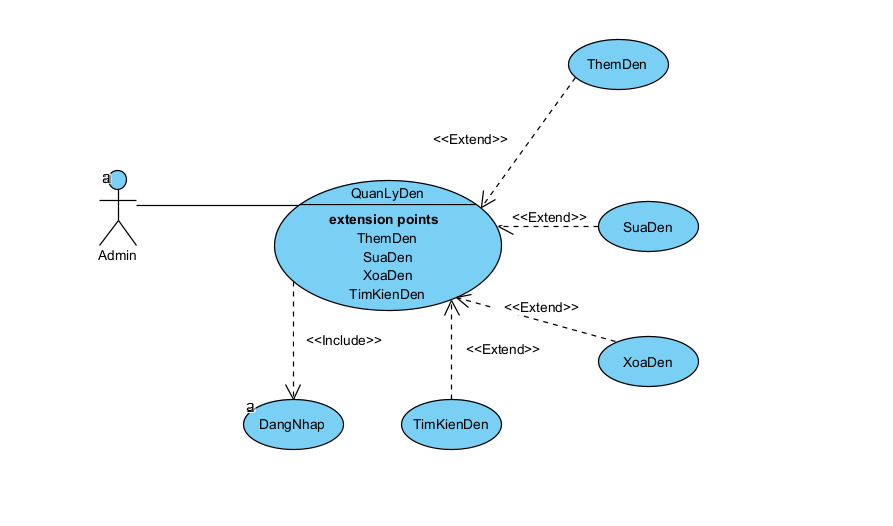
* Dòng sự kiện phụ
* Admin không muốn thêm loại đèn mới
* Admin chọn nút bỏ qua.
* Admin không muốn thêm loại đèn mới nữa, kết thúc UC thêm loại đèn.
* Thông tin đèn không hợp lệ:
* Hệ thống thông báo thông tin loại đèn không hợp lệ, yêu cầu người quản lý nhập lại thông tin.
* Hệ thống trở lại bước 3 để nhập lại thông tin.
* Sửa:
* **Dòng sự kiện chính:**

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống. |
| 1. Admin chọn chức năng sửa loại đèn | 2. Hệ thống hiển thị thông tin loại đèn |
| 1. Admin nhập thông tin loại đèn cần sửa |  |
| 1. Admin lưu thông tin loại đèn mới | 1. Kiểm tra thông tin loại đèn mới được cập nhật |
| 1. Cập nhật thành công lưu thông tin loại đèn vào hệ thống |  |

* **Dòng sự kiện phụ:**
* Admin không muốn cập nhật loại đèn mới:
* Admin nhất nút quay lại.
* Kết thúc use case cập nhật loại đèn.
* Thông tin loại đèn không hợp lệ:
* Hệ thống kiểm tra thông tin loại đèn vừa mới cập nhật, nếu thiếu hệ thống sẽ báo lỗi.
* Hệ thống sẽ quay lại bước 3 để người quản lý cập nhật lại thông tin.
* Xoá:
* **Dòng sự kiện chính:**

|  |  |
| --- | --- |
| **Hành động của tác nhân** | **Phản ứng của hệ thống.** |
| 1. Admin chọn loại đèn cần xoá. | 2. Hệ thống xác thực lại người quản lý có muốn xoá hay không. |
| 1. Admin xác nhận xoá. | 1. Hệ thống kiểm tra ràng buộc. |
| 1. Xoá thành công |  |

* **Dòng sự kiện phụ:**
* Phát sinh 1: Admin không muốn xoá đèn:
* Admin nhất nút không.
* Kết thúc use case xoá đèn.
* Phát sinh 2: Hệ thống kiểm tra ràng buộc
* Hệ thống báo cáo có ràng buộc, không thể xoá.
* Hệ thống sẽ thoát, kết thúc use case.
* Use case quản lý đèn trang trí:



Hình 3.: Biểu đồ phân rã use case quản lý đèn

* Tác nhân: Admin.
* Mô tả: Đây là chức năng dành cho người quản lý và nhân viên quản lý loại đèn.
* Điều kiện cần: Người dùng cần đăng nhập vào hệ thống.
* Điều kiện đủ: Có đủ quyền hạn để thực hiện.
* Thêm đèn:
* **Dòng sự kiện chính:**

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống. |
| 1. Admin chọn nút thêm mới trên giao diện quản lý đèn. | 2. Hệ thống hiển thị thông tin đèn cần thêm. |
| 3. Người quản lý nhập đầy đủ thông tin đèn cần thêm. |  |
| 4. Admin lưu thông tin đèn mới [phát sinh 1] | 5. Kiểm tra thông tin đèn trang trí mới được thêm [phát sinh 2] |
| 6. Thêm thành công lưu thông tin đèn trang trí. |  |

* **Dòng sự kiện phụ:**
* Phát sinh 1: Admin không muốn thêm đèn trang trí mới:
* Admin chọn nút bỏ qua.
* Người quản lý không muốn thêm đèn trang trí mới nữa, kết thúc use case thêm đèn trang trí.
* Phát sinh 2: Thông tin đèn trang trí không hợp lê:
* Hệ thống thông báo thông tin đèn trang trí không hợp lệ, yêu cầu người quản lý nhập lại thông tin.
* Hệ thống trở lại bước 3 để nhập lại thông tin.
* Sửa thông tin đèn trang trí:
* **Dòng sự kiện chính:**

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống. |
| 1. Admin chọn chức năng sửa đèn trang trí. | 2. Hệ thống hiển thị thông tin đèn trang trí cần sửa. |
| 3. Nhập thông tin đèn trang trí cần sửa |  |
| 4. Lưu thông tin đèn trang trí [phát sinh 1] | 5. Kiểm tra thông tin đèn trang trí được cập nhật [phát sinh 2] |
| 6. Cập nhật thành công lưu thông tin đèn trang trí vào hệ thống. |  |

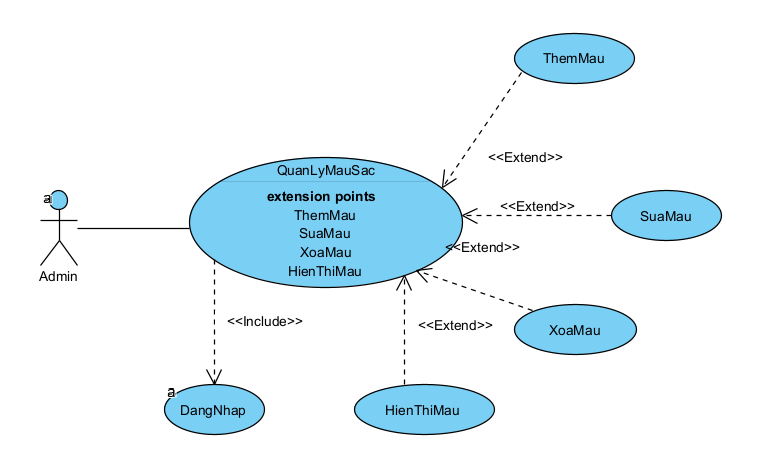
* **Dòng sự kiện phụ:**
* Phát sinh 1: Admin không muốn cập nhật đèn trang trí mới
* Admin nhấn nút quay lại.
* Kết thúc UC cập nhật đèn trang trí.
* Phát sinh 2: Thông tin đèn trang trí không hợp lệ.
* Hệ thống kiểm tra thông tin đèn trang trí vừa mới cập nhật, nếu thiếu hệ thống sẽ báo lỗi.
* Hệ thống sẽ quay lại bước 3 để người quản lý cập nhật lại thông tin đèn trang trí.
* Xoá đèn trang trí:
* **Dòng sự kiện chính:**

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống. |
| 1. Admin chọn đèn trang trí cần xoá. | 2. Hệ thống xác thực lại người quản lý có muốn xoá hay không. |
| 3. Admin xác nhận xoá [phát sinh 1] | 4. Hệ thống kiểm tra ràng buộc [phát sinh 2] |
| 5. Xoá thành công, cập nhật lại hệ thống. |  |

* **Dòng sự kiện phụ:**
* Phát sinh 1: Admin không muốn xoá đèn trang trí:
* Người quản lý nhất nút không.
* Kết thúc use case xoá đèn trang trí.
* Phát sinh 2: Hệ thống kiểm tra ràng buộc
* Hệ thống báo cáo có ràng buộc, không thể xoá.

Hệ thống sẽ thoát, kết thúc UC.

* Use case quản lý màu sắc:



Hình 3.: Biểu đồ phân rã use case quản lý màu sắc.

* **Dòng sự kiện chính:**
* Thêm:

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống. |
| 1. Admin chọn chức năng thêm màu sắc | 2. Hệ thống yêu cầu nhập thông tin màu sắc cần thêm. |
| 3. Admin nhập đầy đủ thông tin màu sắc cần thêm |  |
| 4. Admin lưu thông tin màu sắc mới [phát sinh 1] | 5. Kiểm tra thông tin màu sắc mới được thêm [phát sinh 2] |
| 6. Thêm thành công lưu thông tin KH |  |

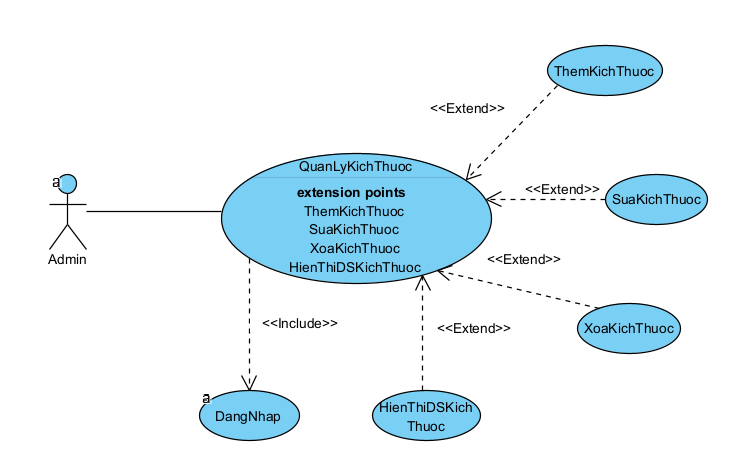
* **Dòng sự kiện phụ:**
* Phát sinh 1: Admin không muốn thêm màu sắc mới:
* Admin chọn nút bỏ qua.
* Admin không muốn thêm màu sắc mới nữa, kết thúc UC thêm màu sắc.
* Phát sinh 2: Thông tin màu sắc không hợp lê:
* Hệ thống thông báo thông tin màu sắc không hợp lệ, yêu cầu người quản lý nhập lại thông tin.
* Hệ thống trở lại bước 3 để nhập lại thông tin.
* Sửa thông tin màu sắc:
* **Dòng sự kiện chính:**

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống. |
| 1. Admin chọn chức năng sửa màu sắc | 2. Hệ thống hiển thị thông tin màu sắc cần sửa. |
| 3. Nhập thông tin màu sắc cần sửa |  |
| 4. Lưu thông tin màu sắc [phát sinh 1] | 5. Kiểm tra thông tin màu sắc được cập nhật [phát sinh 2] |
| 6. Cập nhật thành công lưu thông tin màu sắc vào hệ thống. |  |

* **Dòng sự kiện phụ:**
* Phát sinh 1: Admin không muốn cập nhật màu sắc mới
* Người quản lý nhấn nút quay lại.
* Kết thúc UC cập nhật màu sắc.
* Phát sinh 2: Thông tin khách hàng không hợp lệ.
* Hệ thống kiểm tra thông tin màu sắc vừa mới cập nhật, nếu thiếu hệ thống sẽ báo lỗi.
* Hệ thống sẽ quay lại bước 3 để người quản lý cập nhật lại thông tin màu sắc.
* Xoá:
* **Dòng sự kiện chính:**

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống. |
| 1. Admin chọn màu sắc cần xoá. | 2. Hệ thống xác thực lại người quản lý có muốn xoá hay không. |
| 3. Admin xác nhận xoá [phát sinh 1] | 4. Hệ thống kiểm tra ràng buộc [phát sinh 2] |
| 5. Xoá thành công, cập nhật lại hệ thống. |  |

* **Dòng sự kiện phụ:**
* Phát sinh 1: Admin không muốn xoá màu sắc:
* Người quản lý nhất nút không.
* Kết thúc use case xoá màu sắc.
* Phát sinh 2: Hệ thống kiểm tra ràng buộc
* Hệ thống báo cáo có ràng buộc, không thể xoá.
* Hệ thống sẽ thoát, kết thúc use case.
* Use case quản lý kích thước:



Hình 3.: Biểu đồ phân rã use case quản lý kích thước.

* **Dòng sự kiện chính:**

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống. |
| 1. Admin chọn chức năng thêm kích thước | 2. Hệ thống yêu cầu nhập thông tin kích thước cần thêm. |
| 3. Admin nhập đầy đủ thông tin kích thước cần thêm |  |
| 4. Admin lưu thông tin kích thước mới [phát sinh 1] | 5. Kiểm tra thông tin kích thước mới được thêm [phát sinh 2] |
| 6. Thêm thành công lưu thông tin kích thước |  |

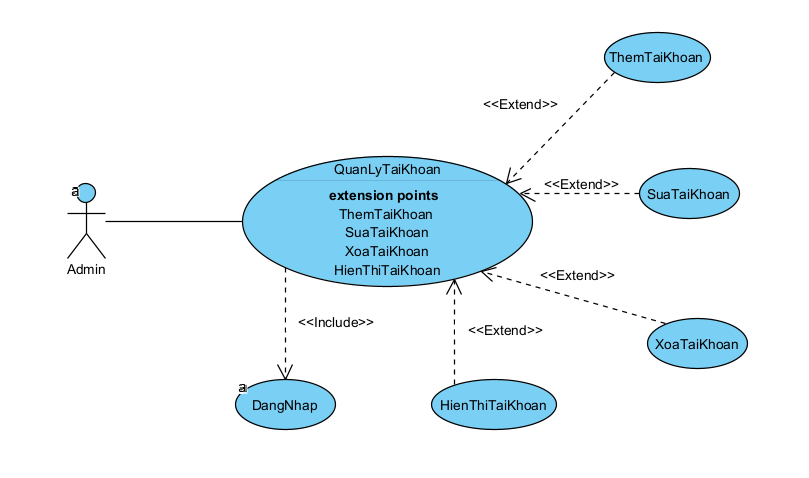
* **Dòng sự kiện phụ:**
* Phát sinh 1: Admin không muốn thêm kích thước mới:
* Admin chọn nút bỏ qua.
* Admin không muốn thêm kích thước mới nữa, kết thúc use case thêm kích thước
* Phát sinh 2: Thông tin kích thước không hợp lê:
* Hệ thống thông báo thông tin kích thước không hợp lệ, yêu cầu người quản lý nhập lại thông tin.
* Hệ thống trở lại bước 3 để nhập lại thông tin.
* Sửa thông tin kích thước:
* **Dòng sự kiện chính:**

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống. |
| 1. Admin chọn chức năng sửa kích thước | 2. Hệ thống hiển thị thông tin kích thước cần sửa. |
| 3. Nhập thông tin kích thước cần sửa |  |
| 4. Lưu thông tin kích thước [phát sinh 1] | 5. Kiểm tra thông tin kích thước được cập nhật [phát sinh 2] |
| 6. Cập nhật thành công lưu thông tin nhân viên vào hệ thống. |  |

* **Dòng sự kiện phụ:**
* Phát sinh 1: Admin không muốn cập nhật kích thước mới
* Admin nhấn nút quay lại.
* Kết thúc use case cập nhật kích thước.
* Phát sinh 2: Thông tin kích thước không hợp lệ.
* Hệ thống kiểm tra thông tin nhân viên vừa mới cập nhật, nếu thiếu hệ thống sẽ báo lỗi.
* Hệ thống sẽ quay lại bước 3 để người quản lý cập nhật lại thông tin kích thước.
* Xoá:
* **Dòng sự kiện chính:**

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống. |
| 1. Admin chọn kích thước cần xoá. | 2. Hệ thống xác thực lại người quản lý có muốn xoá hay không. |
| 3. Admin xác nhận xoá [phát sinh 1] | 4. Hệ thống kiểm tra ràng buộc [phát sinh 2] |
| 5. Xoá thành công, cập nhật lại hệ thống. |  |

* **Dòng sự kiện phụ:**
* Phát sinh 1: Admin không muốn xoá kích thước:
* Admin nhất nút không.
* Kết thúc use case xoá kích thước.
* Phát sinh 2: Hệ thống kiểm tra ràng buộc
* Hệ thống báo cáo có ràng buộc, không thể xoá.
* Hệ thống sẽ thoát, kết thúc use case.
* Use case quản lý tài khoản:



Hình 3.: Biểu đồ Use case Quản lý tài khoản

* **Dòng sự kiện chính:**
* Thêm:

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống. |
| 1. Admin chọn chức năng thêm tài khoản | 2. Hệ thống yêu cầu nhập thông tin tài khoản cần thêm. |
| 3. Admin nhập đầy đủ thông tin tài khoản cần thêm |  |
| 4. Admin lưu thông tin tài khoản mới [phát sinh 1] | 5. Kiểm tra thông tin tài khoản mới được thêm [phát sinh 2] |
| 6. Thêm thành công lưu thông tin tài khoản |  |

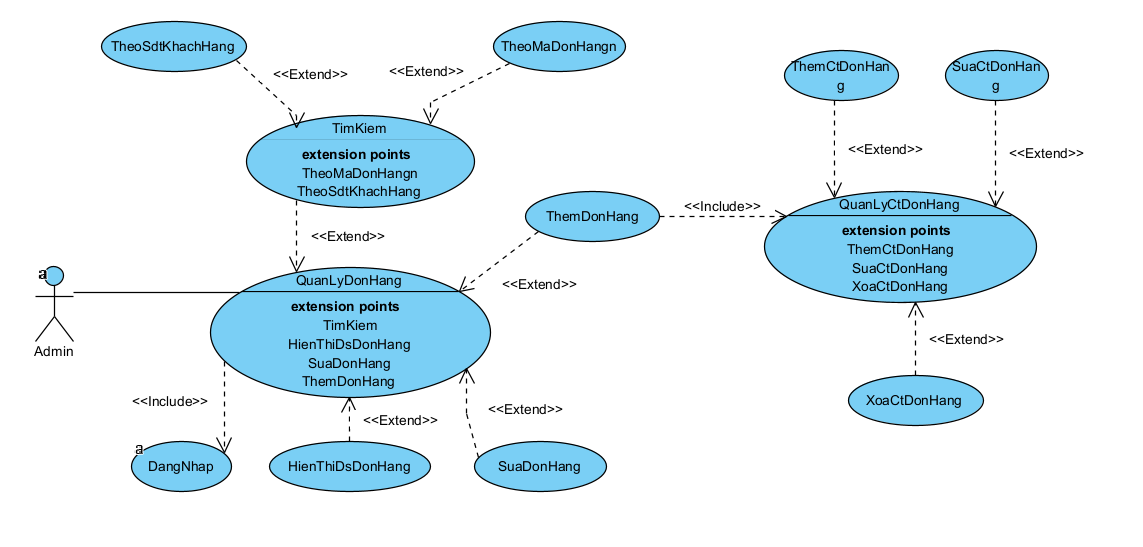
* **Dòng sự kiện phụ:**
* Phát sinh 1: Admin không muốn tài khoản thêm mới:
* Admin chọn nút bỏ qua.
* Admin không muốn thêm tài khoản mới nữa, kết thúc UC thêm tài khoản.
* Phát sinh 2: Thông tin tài khoản không hợp lê:
* Hệ thống thông báo thông tin tài khoản không hợp lệ, yêu cầu người quản lý nhập lại thông tin.
* Hệ thống trở lại bước 3 để nhập lại thông tin.
* Sửa thông tin tài khoản:
* Hệ thống sửa thông tin nhà cung cấp.
* **Dòng sự kiện chính:**

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống. |
| 1. Admin chọn chức năng sửa tài khoản | 2. Hệ thống hiển thị thông tin tài khoản cần sửa. |
| 3. Nhập thông tin tài khoản cần sửa |  |
| 4. Lưu thông tin tài khoản [phát sinh 1] | 5. Kiểm tra thông tin tài khoản được cập nhật [phát sinh 2] |
| 6. Cập nhật thành công lưu thông tin tài khoản vào hệ thống. |  |

* **Dòng sự kiện phụ:**
* Phát sinh 1: Admin không muốn cập nhật tài khoản mới
* Admin nhấn nút quay lại.
* Kết thúc UC cập nhật tài khoản.
* Phát sinh 2: Thông tin tài khoản không hợp lệ.
* Hệ thống kiểm tra thông tin tài khoản vừa mới cập nhật, nếu thiếu hệ thống sẽ báo lỗi.
* Hệ thống sẽ quay lại bước 3 để người quản lý cập nhật lại thông tin tài khoản.
* Xoá:
* **Dòng sự kiện chính:**

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống. |
| 1. Admin chọn tài khoản cần xoá. | 2. Hệ thống xác thực lại người quản lý có muốn xoá hay không. |
| 3. Người quản lý xác nhận xoá [phát sinh 1] | 4. Hệ thống kiểm tra ràng buộc [phát sinh 2] |
| 5. Xoá thành công, cập nhật lại hệ thống. |  |

* **Dòng sự kiện phụ:**
* Phát sinh 1: Admin không muốn xoá tài khoản:
* Người quản lý nhất nút không.
* Kết thúc UC xoá tài khoản.
* Phát sinh 2: Hệ thống kiểm tra ràng buộc
* Hệ thống báo cáo có ràng buộc, không thể xoá.
* Hệ thống sẽ thoát, kết thúc UC.
* Use case quản lý đơn hàng:



Hình 3.9: Biểu đồ phân rã use case quản lý đơn hàng

**+Thêm đơn hàng:**

* Dòng sự kiện chính:

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| 1. Yêu cầu chức năng sinh mã đơn hàng | 1. Tạo list các đơn hàng và đổ dữ liệu từ cơ sở dữ liệu, lấy ra mã hóa đơn bán cuối cùng trong list, tách riêng phần kí tự và số, cộng số lên 1.Nếu chưa có hóa đơn bán trong database thì chuyển xuống dòng phụ 2.a. |
| 1. Nhập thông tin đơn hàng. Thông tin đơn hàng bao gồm: Mã nhân viên, mã khách hàng, ngày đặt | 1. Kiểm tra các ràng buộc: Mã nhân viên, mã khách hàng phải tồn tại trong hệ thống, ngày bán không được lớn hơn ngày hiện tại. Nếu vi phạm ràng buộc chuyển xuống dòng phụ 4.a. |
| 1. Lưu thông tin | 1. Thêm hóa đơn bán(Dùng phương thức Add() và SaveChange() để thêm và lưu vào cơ sở dữ liệu). Chuyển tiếp đến chức năng quản lý chi tiết hóa đơn bán |

* Dòng sự kiện phụ:

***Dòng phụ 2a. chưa có đơn hàng nào trong hệ thống***

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| 1. Yêu cầu sinh mã đơn hàng mới | 1. Tạo mã mới đánh số từ HDB001. Quay lại bước 2. |

***Dòng phụ 4.a.Thông tin sản đơn hàng***

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| 1. Người dùng nhập sai thông tin đơn hàng | 1. Hiển thị giao diện báo lỗi, yêu cầu nhập lại. |
| 1. Người dùng nhập lại thông tin hóa đơn bán | 1. Quay lại bước 4. ở dòng chính. |

**Sửa đơn hàng:**

* Dòng sự kiện chính:

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| 1. Chọn hóa đơn bán cần sửa | 1. Hiển thị thông tin hóa đơn bán vừa chọn nên phần nhập thông tin. |
| 1. Nhập thông tin cần sửa vào từ bàn phím | 1. Kiểm tra các ràng buộc: Mã nhân viên, mã khách hàng phải tồn tại trong hệ thống, ngày bán không được lớn hơn ngày hiện tại.. Nếu vi phạm ràng buộc chuyển xuống dòng phụ 4.a. |
| 1. Lưu thông tin | 1. Sửa hóa đơn bán (Dùng phương thức Edit() gán lại các thông tin mới vào đối tượng hóa đơn bán mà người dùng yêu cầu và SaveChanges() để lưu thay đổi vào cơ sở dữ liệu). |

* Dòng sự kiên phụ:

***Dòng phụ 4.a: Nhập đơn hàng***

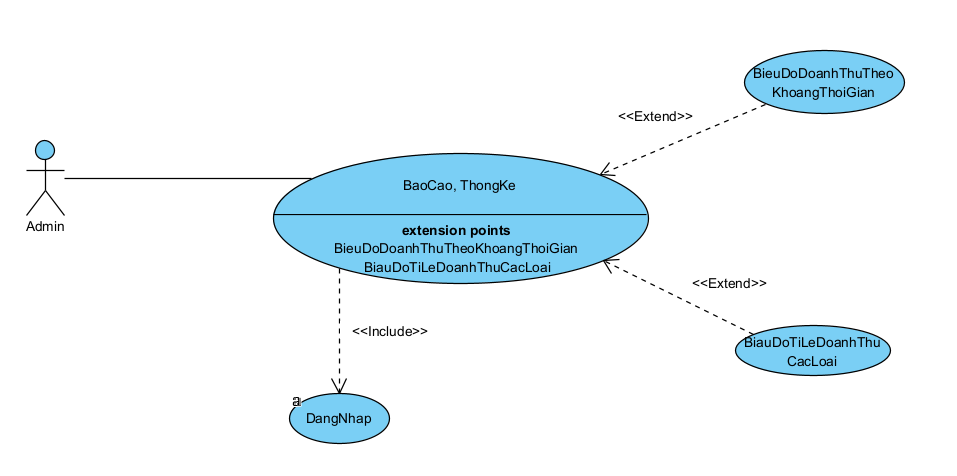
|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| 1. Người dùng nhập sai thông tin hóa đơn bán | 1. Hiển thị giao diện báo lỗi, yêu cầu nhập   lại. |
| 1. Người dùng nhập lại thông tin hóa đơn bán | 1. Quay lại bước 4 ở dòng chính. |

**Tìm kiếm đơn hàng:**

* Dòng sự kiện chính:

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| 1. Nhập thông tin đơn hàng muốn tìm kiếm.Thông tin hóa đơn bán có thể tìm kiếm gồm: Mã hóa đơn hàng, ngày nhập, số điện thoại nhà cung cấp. | 1. Truyền thông tin vừa nhập vào Stored Procedures tìm kiếm nâng cao được tạo trong hệ quản trị cơ sở dữ liệu sau đó lấy list danh sách hóa đơn bán vừa tìm thấy hiển thị lên giao diện. Nếu không tìm thấy sẽ thông báo cho người dùng |

* Use case báo cáo – thống kê:



Hình 3.0: Use case xem báo cáo, thống kê

* **Mô tả tóm tắt:**
  + Tác nhân chính: Admin
  + Chức năng: Vẽ biểu đồ doanh thu của quán trong khoảng thời gian
  + Ý nghĩa: Xem thống kê doanh thu của khách sạn
* **Dòng sự kiện chính:**

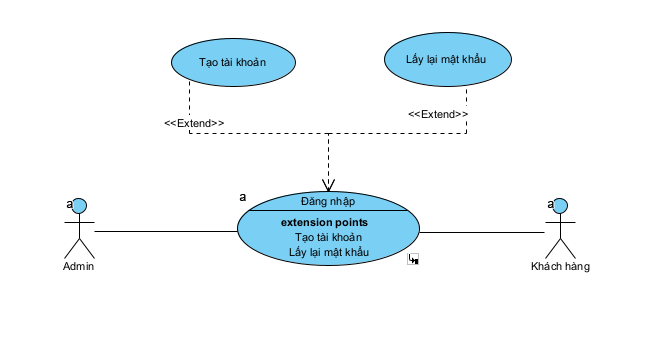
Bảng 3.14 Dòng sự kiện thống kê doanh thu

|  |  |
| --- | --- |
| **Hành động của tác nhân** | **Phản ứng của hệ thống** |
| 1. Mở chức năng thống kê doanh thu |  |
|  | 1. Hiển thị trang thống kê, báo cáo: trang hiển thị doanh thu website. |
| 1. Lựa chọn khoảng thời gian và nhấn nút xác nhận |  |
|  | 1. Kiểm tra tính hợp lệ của khoảng thời gian được chọn |
| 1. Hiện thị các thông số và biểu đồ thống kê của quán theo khoảng thời gian được chọn |  |

* *Dòng sự kiện phụ:*
  + Không có
* *Dòng sự kiện rẽ nhánh:*
  + Khoảng thời gian được chọn không hợp lệ : thông báo “khoảng thời gian được chọn không hợp lệ”
* *Dòng sự kiện ngoại lệ:*
  + Khi ứng dụng bị tắt trước khi kiểm tra tính hợp lệ của khoảng thời gian hoàn thành : huỷ tất cả thao thác trước đó
* *Trạng thái hệ thống khi bắt đầu thực hiện Use case:*
* Đăng nhập thành công với quyền Admin
* *Trạng thái hệ thống sau khi thực hiện Use case:*

Hiển thị doanh thu, hiển thị biểu đồ doanh thu theo khoảng thời gian được chọn

**\*)Use case đăng nhập:**



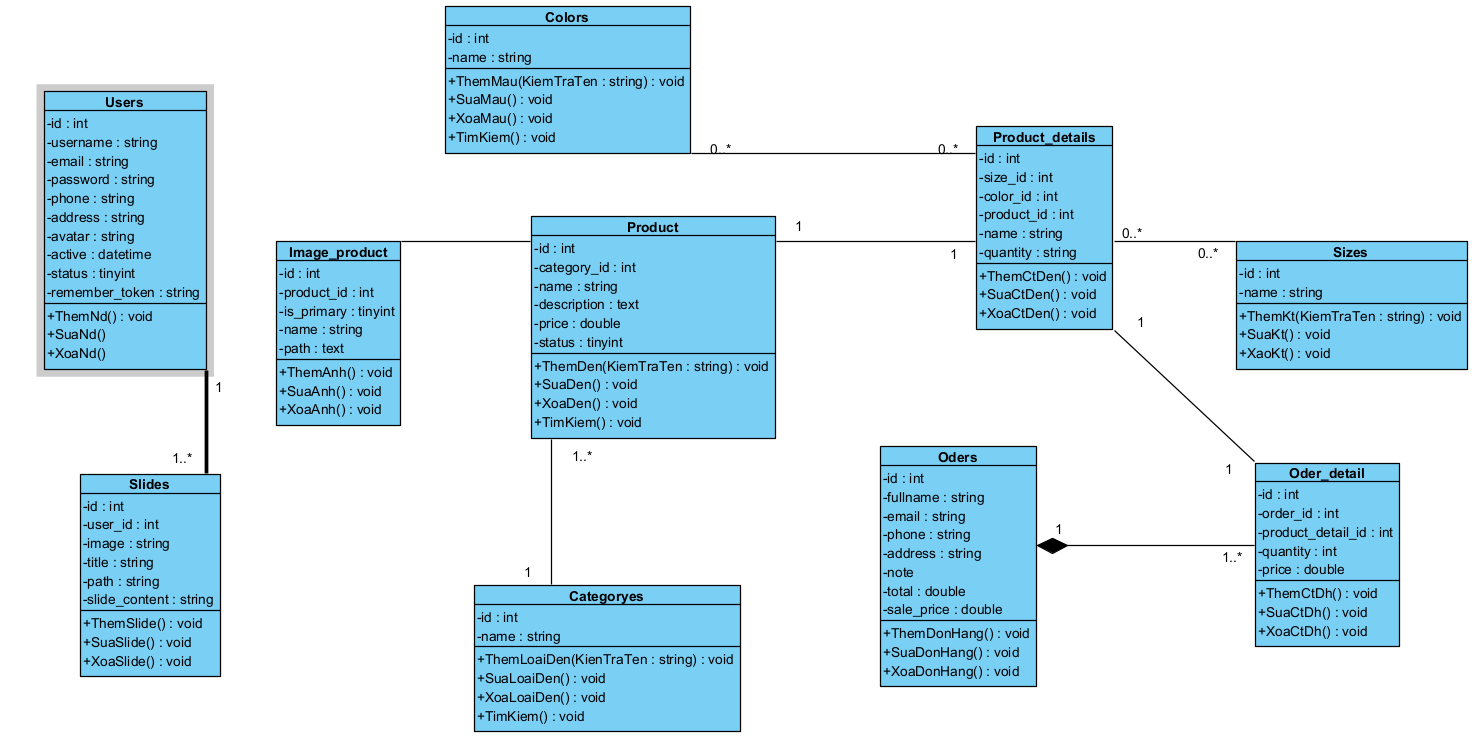
Hình 3.1: Biểu đồ phân rã use case đăng nhập

**Mô tả kịch bản của một số ca sử dụng**

* Mô tả luồng sự kiện của ca đăng nhập:

|  |
| --- |
| 1. Yêu cầu chức năng đăng nhập |
| 2. SYSTEM Hiển thị giao diện đăng nhập |
| 3. Nhập user name, nhập password, chọn chức năng ghi nhớ tài khoản |
| 4. Yêu cầu đăng nhập |
| 5. SYSTEM Lấy về thông tin người dùng , nếu người dùng không có chuyển sang luồng phụ 5a |
| 6. if Nếu chọn ghi nhớ tài khoản |
| 6.1. SYSTEM Hệ thống lưu ghi nhớ tài khoản |
| 6. end if |
| 7. SYSTEM Thiết lập trạng thái đăng nhập, đóng giao diện đăng nhập |
| Extension: |
| 5.a. Tài khoản không đúng |
| 1. SYSTEM Hệ thống hiển thịu thông báo đăng nhập không thành công, yêu cầu đăng nhập lại, quay lại bước 3 |

* + 1. Biểu đồ lớp thực thể



**Hình 3.12: Biểu đồ lớp thực thể**

Trong quá trình khảo sát xây dựng Website, em xác định được các thuật ngữ miền của hệ thống như sau:

* Categoryes

Bảng 3.: Thông tin mô tả Categeryes

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Mô tả** |
| 1 | id | Mỗi loại đèn trang trí sẽ được đánh một mã loại đèn khác nhau |
| 2 | name | Tên của loại đèn |

* Products

Bảng 3.: Thông tin mô tả product

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Mô tả** |
| 1 | id | Mỗi đèn trang trí sẽ được đánh một mã đèn khác nhau |
| 2 | category\_id | Mỗi loại đèn trang trí sẽ được đánh một mã loại đèn khác nhau |
| 3 | name | Mỗi loại đèn trang trí sẽ được đánh một mã loại đèn khác nhau Tên của đèn |
| 4 | description | Mô tả các thông tin, ảnh về đèn trang trí |
| 5 | price | Giá của đèn |
| 6 | status | Trạng thái còn hay hết của đèn |

* Colors

Bảng 3.: Thông tin mô tả Colors

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Mô tả** |
| 1 | id | Mỗi màu sắc sẽ được đánh một mã khác nhau |
| 2 | name | Tên của nhà màu |

* Sizes

Bảng 3.: Thông tin mô tả sizes

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Mô tả** |
| 1 | id | Mỗi kích thước sẽ được đánh một mã khác nhau |
| 2 | name | Tên kích thước |

* Users

Bảng 3.: Thông tin mô tả users

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Mô tả** |
| 1 | id | Mỗi tài khoản sẽ có một mã khác nhau |
| 2 | username | Tên tài khoản |
| 3 | email | Email tài khoản |
| 4 | password | Mật khẩu đăng nhập của tài khoản đó |
| 5 | phone | Số điện thoại liên kết với tài khoản |
| 6 | address | Địa chỉ người dùng tài khoản này |
| 7 | avatar | Hình ảnh đại diện của người dùng tài khoản |
| 8 | active |  |
| 9 | remember\_token | Dùng để ghi nhớ |

* Orders

Bảng 3.: Thông tin mô tả Orders

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Mô tả** |
| 1 | id | Mỗi đơn hàng sẽ được đánh một mã đơn bán hàng khác nhau |
| 2 | fullname | Tên cảu khách hàng đặt đơn |
| 2 | email | Email của tài khoản đặt đơn |
| 2 | phone | Số điện thoại liên hệ với khách đặt hàng |
| 3 | address | Địa chỉ của khách hàng |
|  | note | Ghi chú của khách hàng khi đặt hàng |
|  | total | Tổng tiền của đơn hàng |
|  | sale\_price | Giá bán của sản phẩm |

* OrderDetails

Bảng 3.: Thông tin OrderDetails

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Mô tả** |
| 1 | id | Mỗi chi tiết của đơn hàng sẽ có một mã chi tiết khác nhau |
| 2 | order\_id | Mã đơn hàng |
| 3 | product\_detail\_id | Mã chi tiết sản phẩm của đơn hàng |
| 4 | quantity | Số lượng của sản phẩm |

* + 1. Các yêu cầu phi chức năng

Bảo mật và hiệu suất:

* Yêu cầu có hệ thống xác thực và quản lý quyền truy cập cho người dùng và quản trị viên.
* Yêu cầu bảo vệ thông tin cá nhân của người dùng, bao gồm dữ liệu tài khoản và giao dịch.
* Yêu cầu bảo vệ dữ liệu quảng cáo và khuyến mãi để ngăn ngừa sự can thiệp không mong muốn.
* Yêu cầu hệ thống phải đảm bảo tốc độ truy cập nhanh chóng và thời gian phản hồi ngắn đối với người dùng.
* Yêu cầu xử lý đồng thời nhiều đơn hàng và truy vấn cơ sở dữ liệu một cách hiệu quả trong các mùa mua sắm sôi động hoặc sự kiện khuyến mãi.

Khả năng mở rộng:

* Yêu cầu hệ thống phải dễ dàng mở rộng để có thể chịu được lưu lượng truy cập gia tăng.
* Yêu cầu khả năng mở rộng dự án để thêm tính năng mới hoặc tích hợp dịch vụ bên ngoài.

Độ tin cậy:

* Yêu cầu hệ thống phải hoạt động ổn định và không gặp sự cố thường xuyên.
* Yêu cầu có kế hoạch sao lưu và phục hồi dự phòng để đảm bảo dữ liệu an toàn.

Trải nghiệm người dùng:

* Yêu cầu giao diện người dùng phải dễ sử dụng và thân thiện với người dùng.
* Yêu cầu tương tác mượt mà, thao tác mua sắm dễ dàng, và hỗ trợ nhiều thiết bị, độ phân giải.

Bảo trì và hỗ trợ:

* Yêu cầu có khả năng theo dõi và bảo trì hệ thống để đảm bảo hoạt động ổn định.
* Yêu cầu hỗ trợ khách hàng và cung cấp hướng dẫn sử dụng dễ tiếp cận.

Tuân thủ pháp luật và quy định:

* Yêu cầu phải tuân thủ tất cả các quy định pháp luật về bảo vệ dữ liệu cá nhân và giao dịch trực tuyến.
* Yêu cầu cung cấp các thông tin về quyền lợi của người tiêu dùng, chính sách vận chuyển và hoàn trả, và điều khoản sử dụng.

Bảo vệ trước tấn công và lừa đảo:

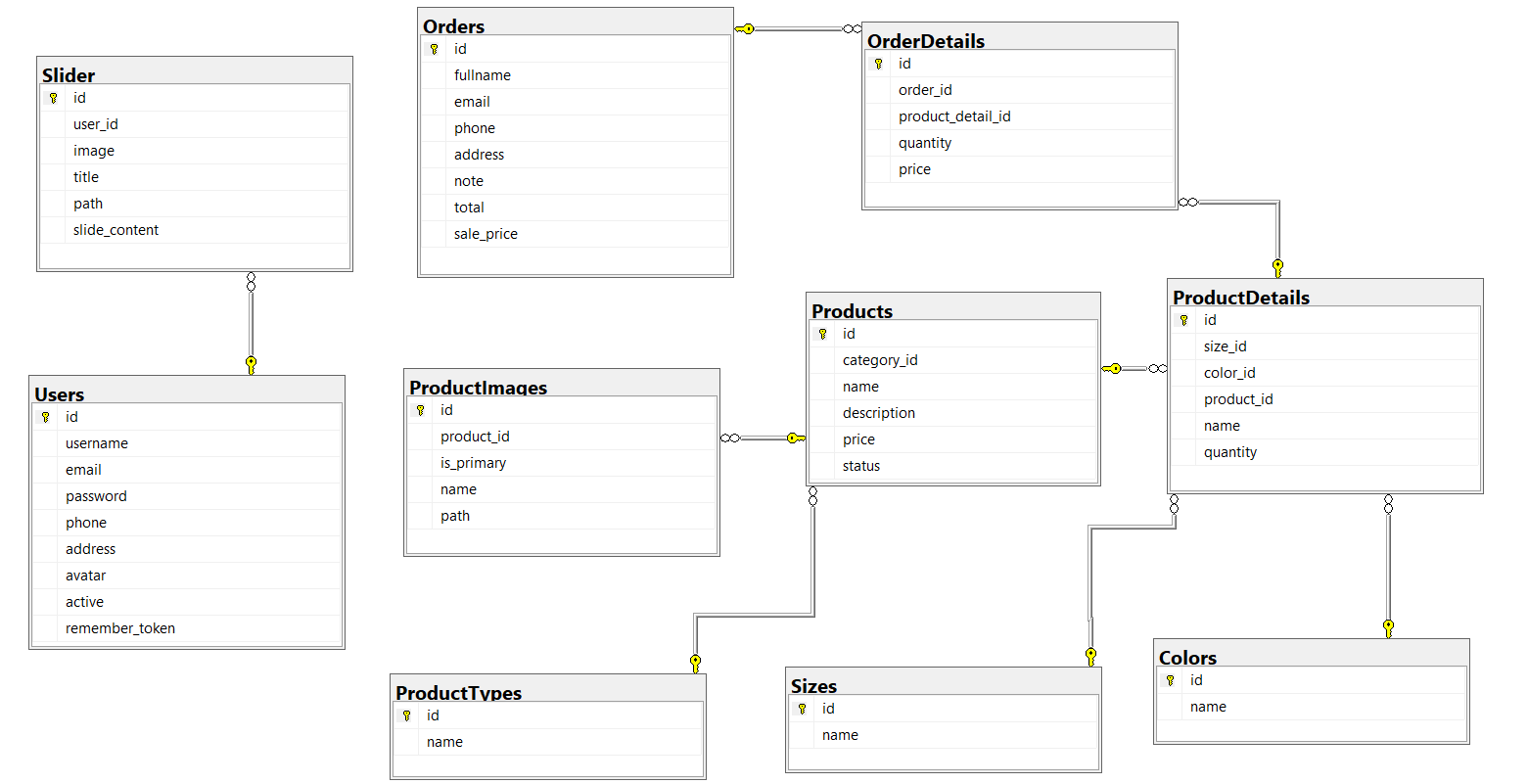
* Yêu cầu phải có biện pháp bảo vệ trước tấn công DDoS, tấn công SQL injection và các hình thức tấn công mạng khác.
* Yêu cầu theo dõi và phát hiện các hoạt động bất thường để đối phó với lừa đảo trực tuyến.

Tích hợp bên ngoài:

* Yêu cầu khả năng tích hợp với các dịch vụ bên ngoài như thanh toán trực tuyến và hệ thống giao vận.

Tương thích đa nền tảng:

* 1. Yêu cầu phải đảm bảo rằng trang web hoạt động tốt trên nhiều loại trình duyệt và thiết bị, bao gồm cả máy tính, điện thoại di động và máy tính bảng.Thiết kế hệ thống
     1. Thiết kế cơ sở dữ liệu



**Hình 3.14: Mô hình cơ sở dữ liệu**

* + - 1. ProductTypes

Bảng 3.12 Bảng mô tả thuộc tính bảng ProductTypes

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Rằng buộc** |
| **1** | id | int | Khóa chính |
| **2** | name | Nvarchar(50) | Not null |

* + - 1. Products

Bảng 3.13 Bảng mô tả thuộc tính bảng Products

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Rằng buộc** |
| **1** | id | int | Khóa chính |
| **2** | category\_id | int | Khoá ngoại |
| 3 | name | varchar(50) | Not null |
| **4** | description | Text | Not null |
| **5** | price | Int | Not null |
| **6** | status | tinyint | Not null |

* + - 1. Colos

Bảng 3.14 Bảng mô tả thuộc tính bảng Colos

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Rằng buộc** |
| **1** | id | int | Khóa chính |
| **2** | name | Nvarchar(50) | Not Null |

* + - 1. Sizes

Bảng 3.15 Bảng mô tả thuộc tính bảng Sizes

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Rằng buộc** |
| **1** | id | int | Khóa chính |
| **2** | name | Nvarchar(50) | Not Null |

* + - 1. Chi tiết ProductDetails

Bảng 3.16 Bảng mô tả thuộc tính bảng chi tiết ProductDetails

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Rằng buộc** |
| **1** | id | int | Khóa chính |
| **2** | size\_id | Int | Khoá ngoại |
| **3** | color\_id | Int | Khoá ngoại |
| **4** | product\_id | int | Khoá ngoại |
| **5** | name | Nvarchar(50) | Not null |
| **6** | quantity | int | Khoá ngoại |

* + - 1. ProductImages

Bảng 3.17 Bảng mô tả thuộc tính bảng ProductImages

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Rằng buộc** |
| **1** | id | int | Khóa chính |
| **2** | product\_id | Int | Khoá ngoại |
| **3** | is\_primary | tinyint | Not null |
| **4** | name | Nvarchar(50) | Not null |
| **5** | path | Text | Not null |

* + - 1. Slider

Bảng 3.18 Bảng mô tả thuộc tính bảng Slider

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Rằng buộc** |
| **1** | id | int | Khóa chính |
| **2** | user\_id | Int | Khoá ngoại |
| **3** | image | Nvarchar(50) | Not null |
| **4** | title | Nvarchar(50) | Not null |
| **5** | path | Text | Not null |
| **6** | slide\_content | Nvarchar(50) | Not null |

* + - 1. Orders

Bảng 3.19 Bảng mô tả thuộc tính bảng Orders

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Rằng buộc** |
| **1** | id | int | Khóa chính |
| **2** | fullname | Nvarchar(50) | Khoá ngoại |
| **3** | email | Nvarchar(50) | Not null |
| **4** | phone | Nvarchar(50) | Not null |
| **5** | address | Nvarchar(50) | Not Null |
| **6** | note | Nvarchar(50) | Not null |
| **7** | total | double | Not Null |
| **8** | sale\_price | double | Not null |

* + - 1. OrderDetails

Bảng 3.20 Bảng mô tả thuộc tính bảng OrderDetails

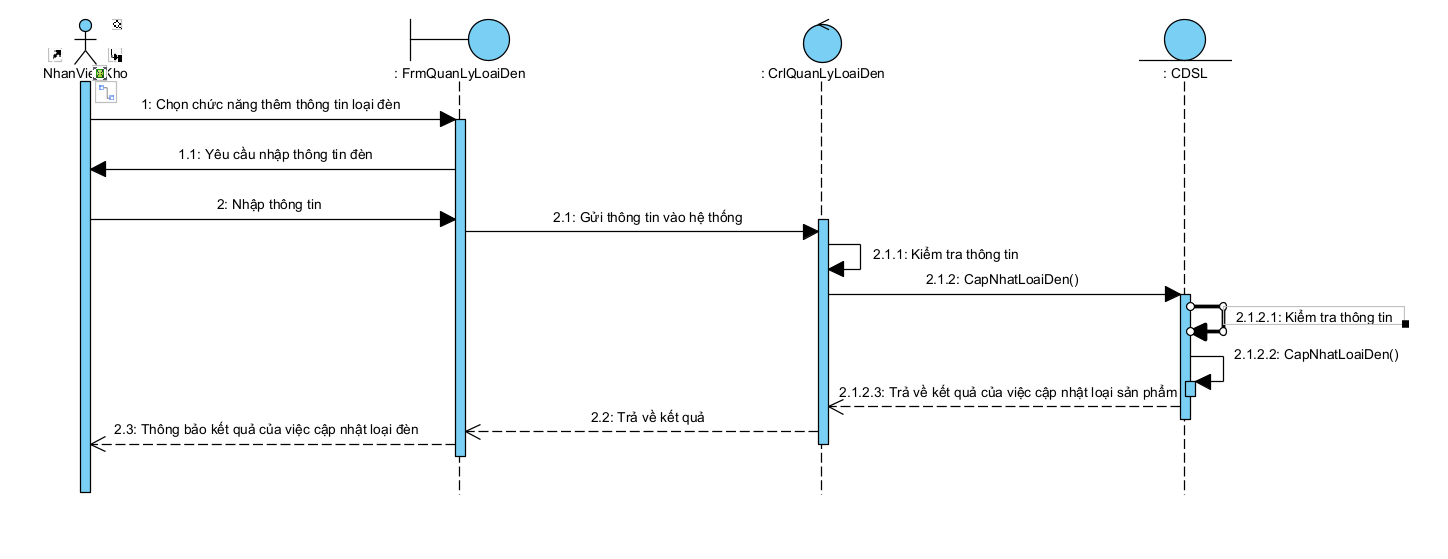
|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Rằng buộc** |
| **1** | id | int | Khoá chính |
| **2** | order\_id | int | Khoá ngoại |
| **3** | product\_detail\_id | int | Khoá ngoại |
| **4** | quantity | Int | Not null |
| **5** | price | Float | Not Null |

* + - 1. Users

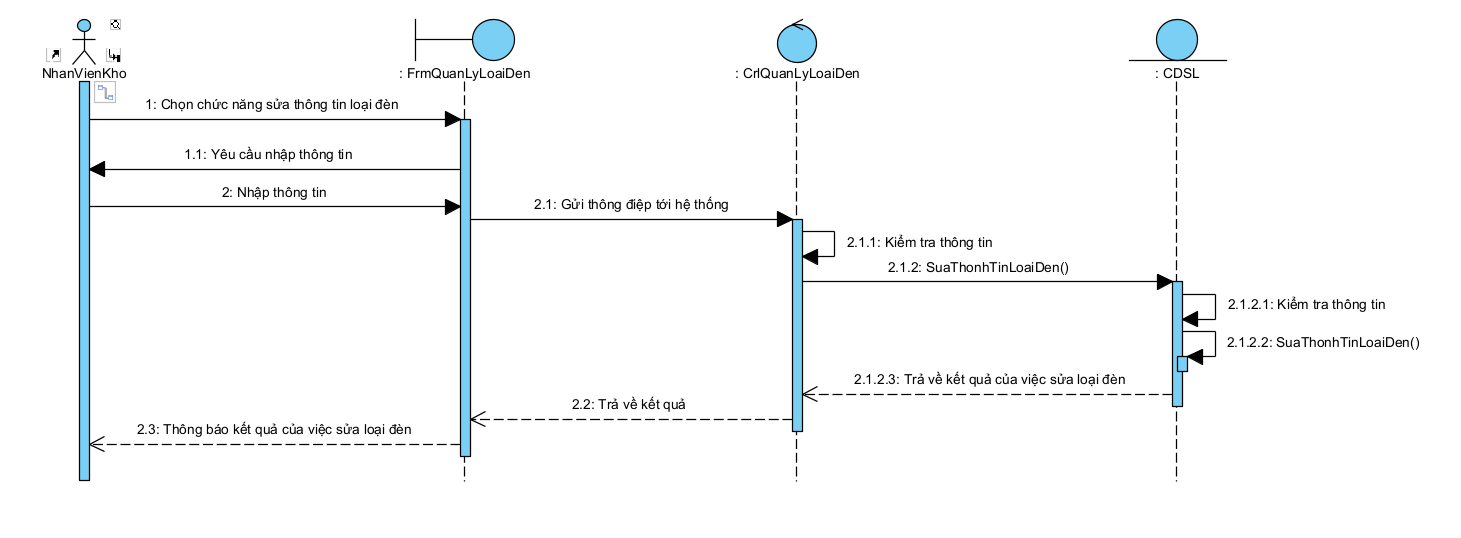
Bảng 3.21: Bảng mô tả thuộc tính bảng users

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Rằng buộc** |
| **1** | id | int | Khóa chính |
| **2** | username | Nvarchar(50) | Not Null |
| **3** | email | Nvarchar(50) | Not Null |
| **4** | password | Nvarchar(50) | Not Null |
| **5** | phone | Nvarchar(50) | Not Null |
| **6** | address | Nvarchar(50) | Not Null |
| **7** | avatar | Nvarchar(50) | Not Null |
| **8** | active | tinyint | Not Null |
| **9** | remember\_token | Nvarchar(50) | Not Null |

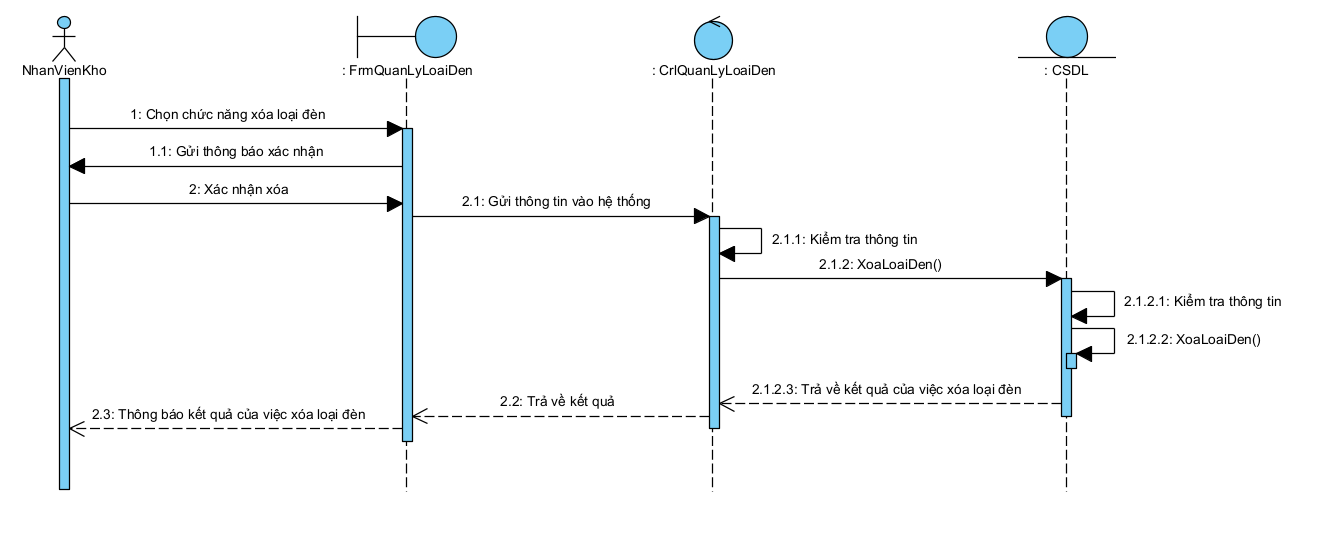
* + 1. Thiết kế lớp đối tượng (biểu đồ tuần tự)



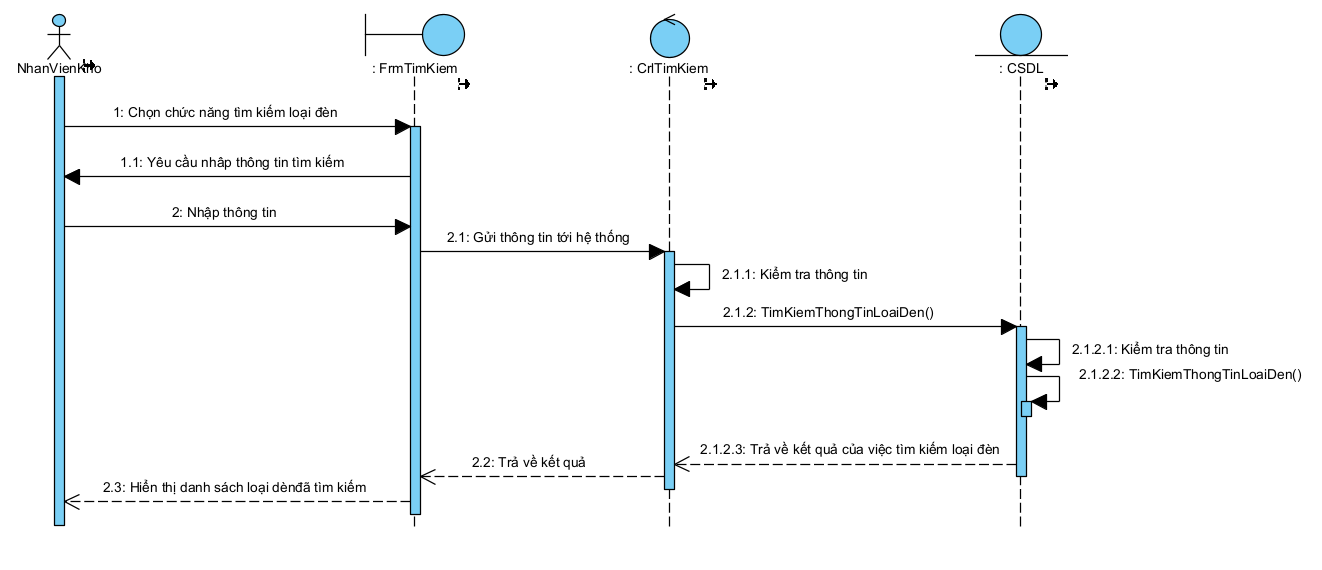
Hình 3.14: Biểu đồ tuần tự chức năng nhập thông tin loại đèn



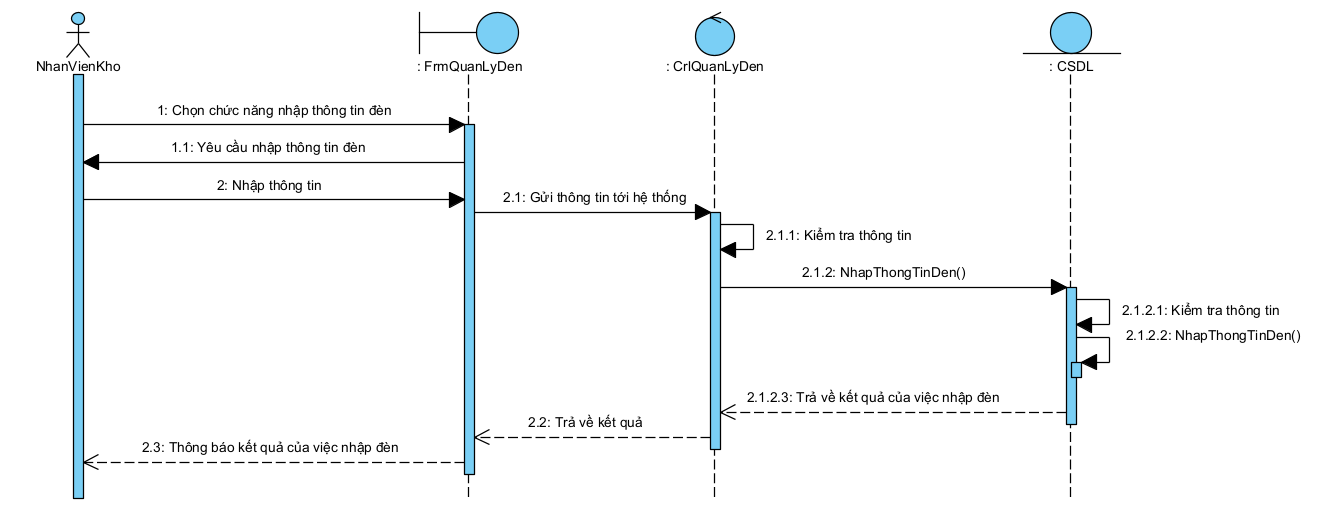
Hình 3.15: Biểu đồ tuần tự chức năng sửa thông tin loại đèn



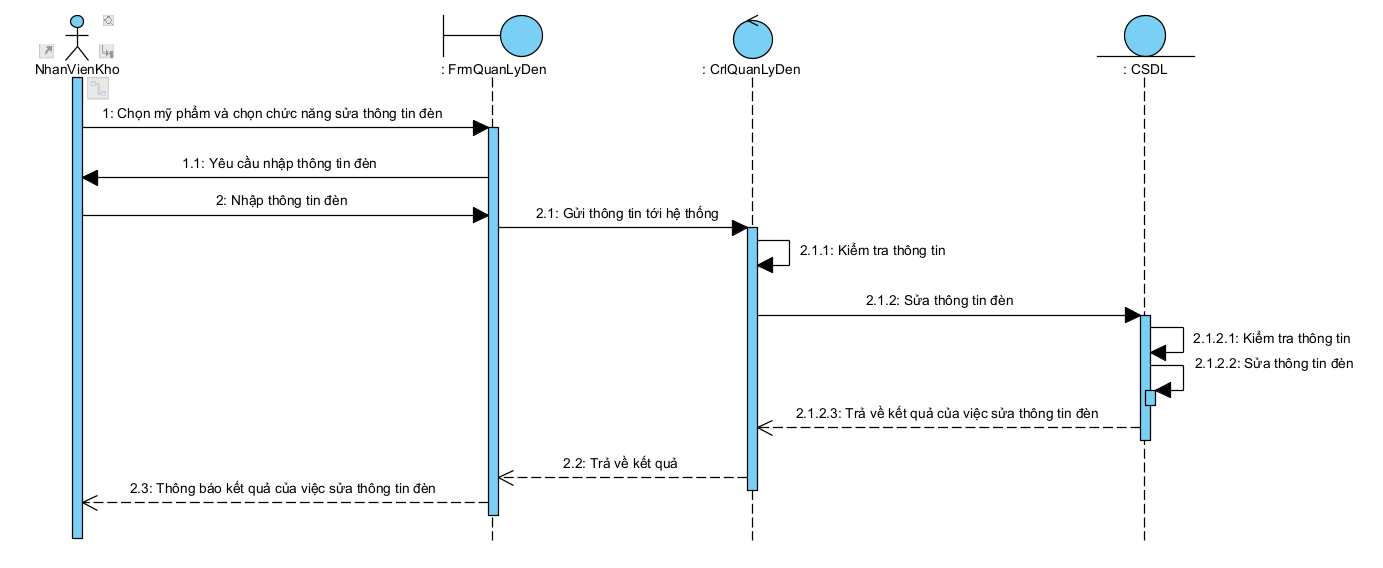
Hình 3.16: Biểu đồ tuần tự chức năng xóa thông tin loại đèn



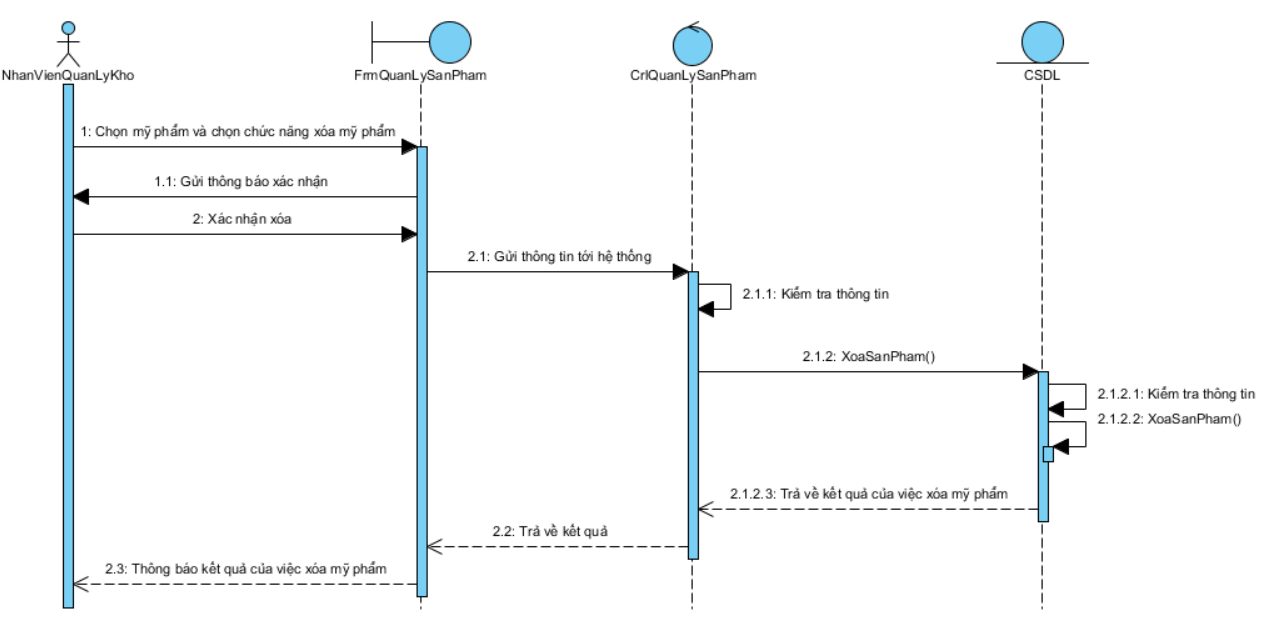
Hình 3.15: Biểu đồ tuân tự chức năng tìm kiếm thông tin loại đèn



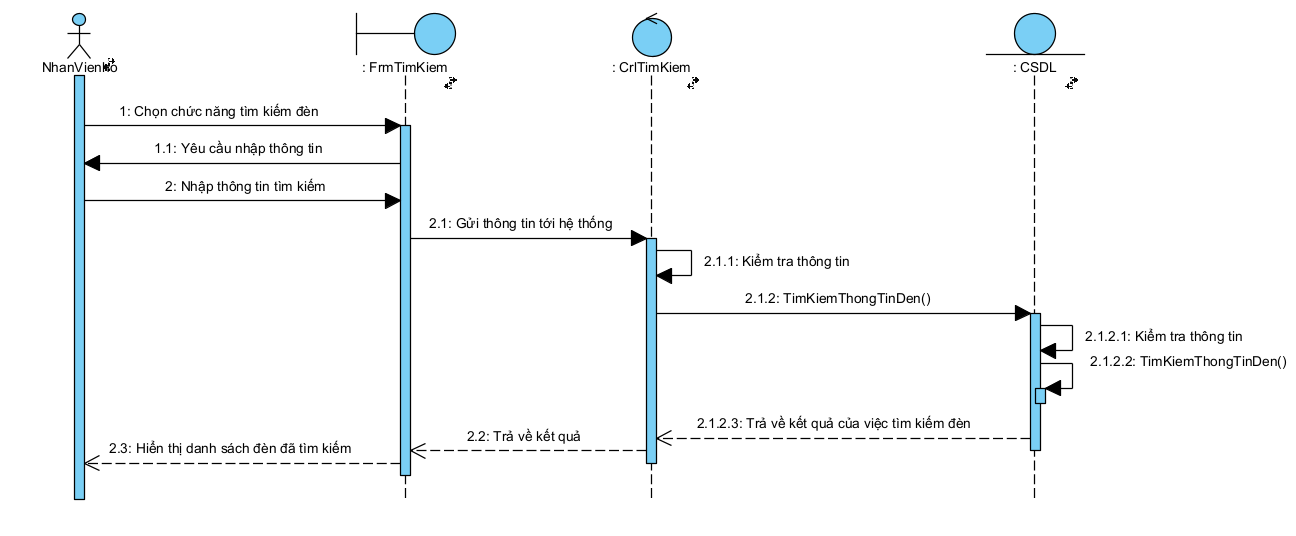
Hình 3.18: Biểu đồ tuần tự chức năng nhập thông tin đèn



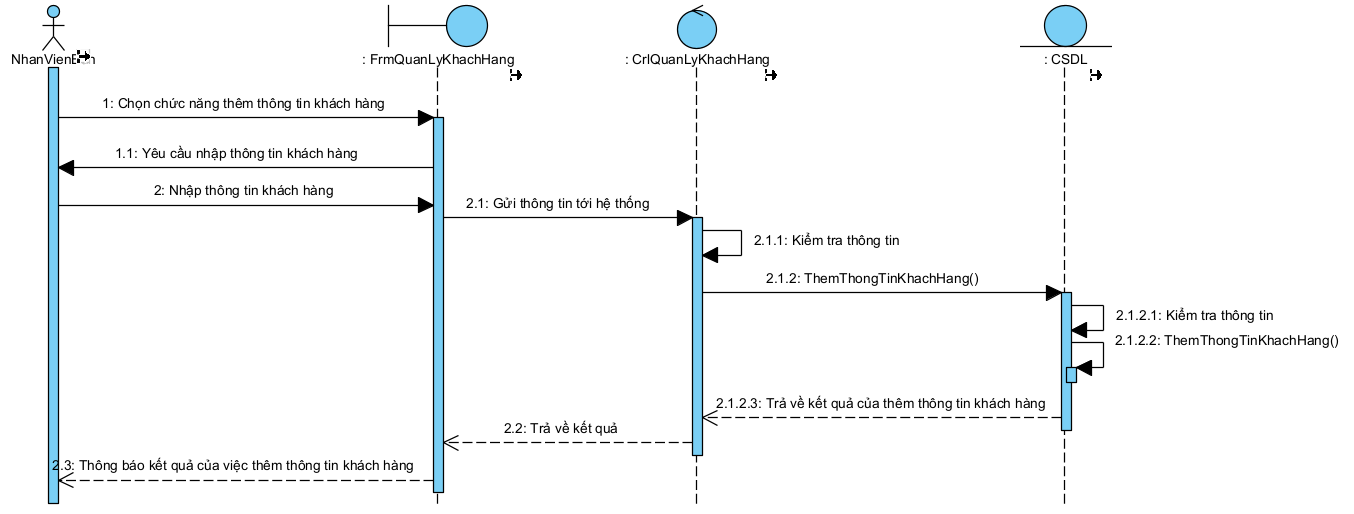
Hình 3.19: Biểu đồ tuần tự chức năng sửa thông tin đèn



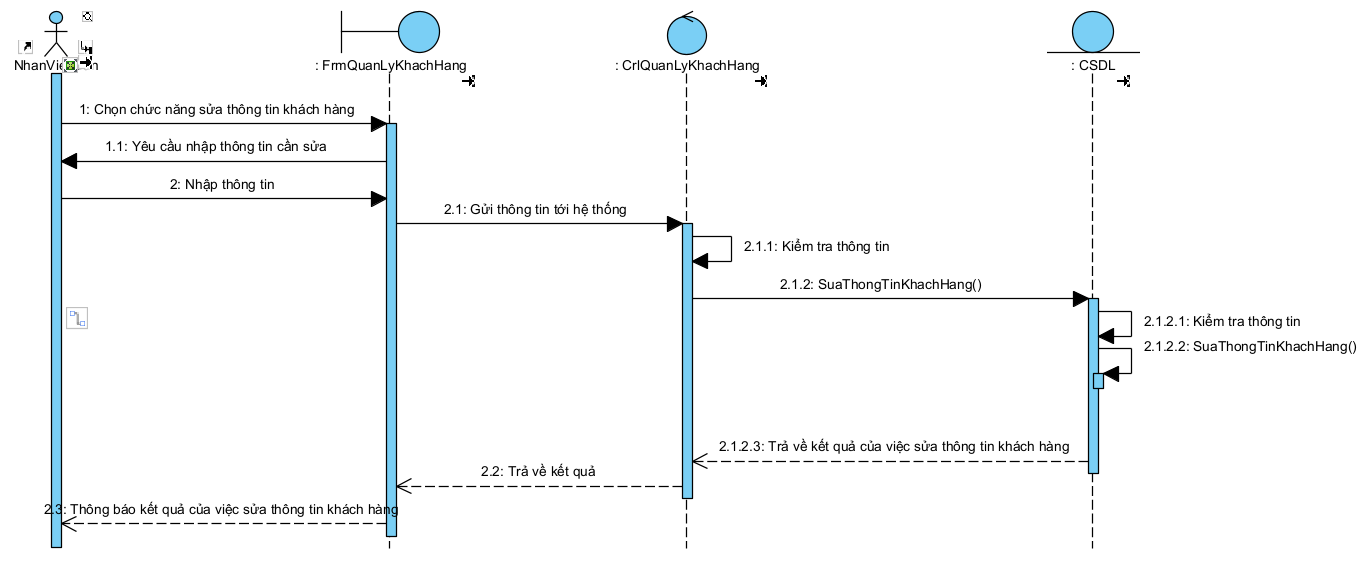
Hình 3.20 : Biểu đồ tuần tự chức năng xóa thông tin đèn



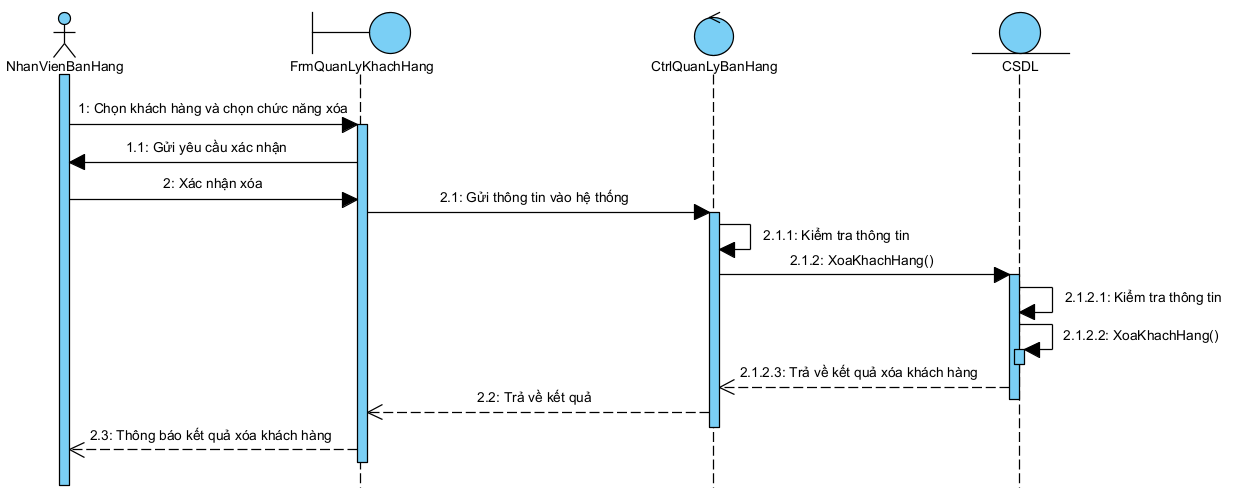
Hình 3.21: Biểu đồ tuần tự chức năng tìm kiếm thông tin đèn



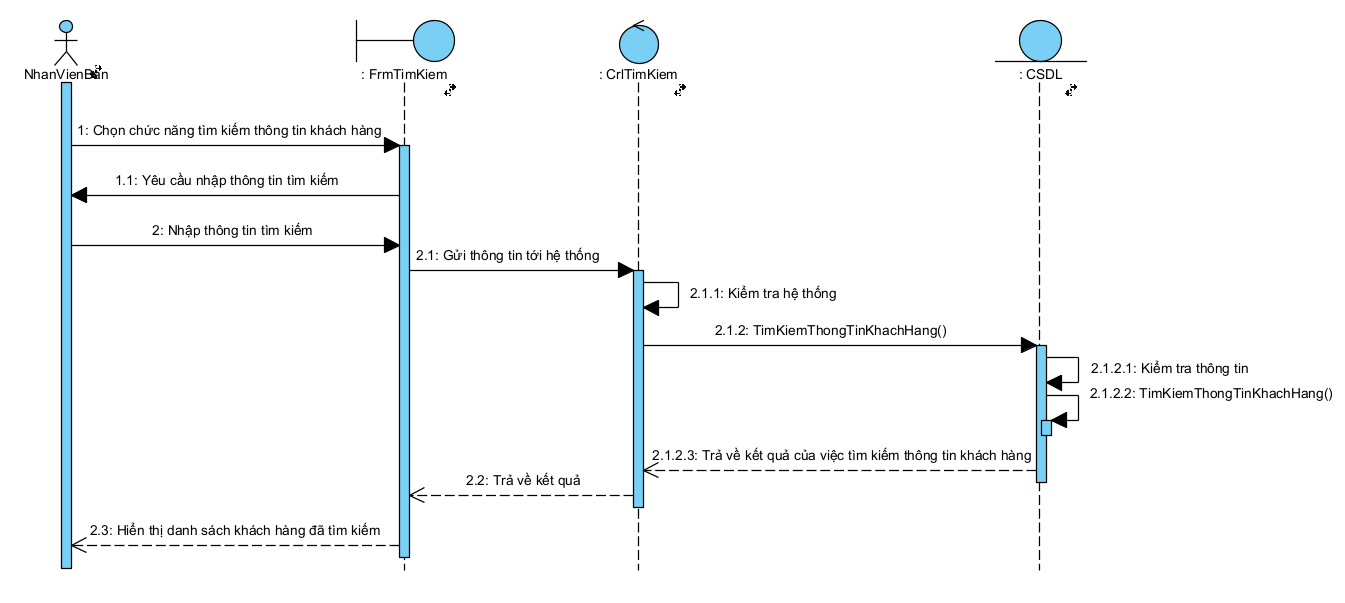
Hình 3.22: Biểu đồ tuần tự chức năng nhập thông tin khách hàng



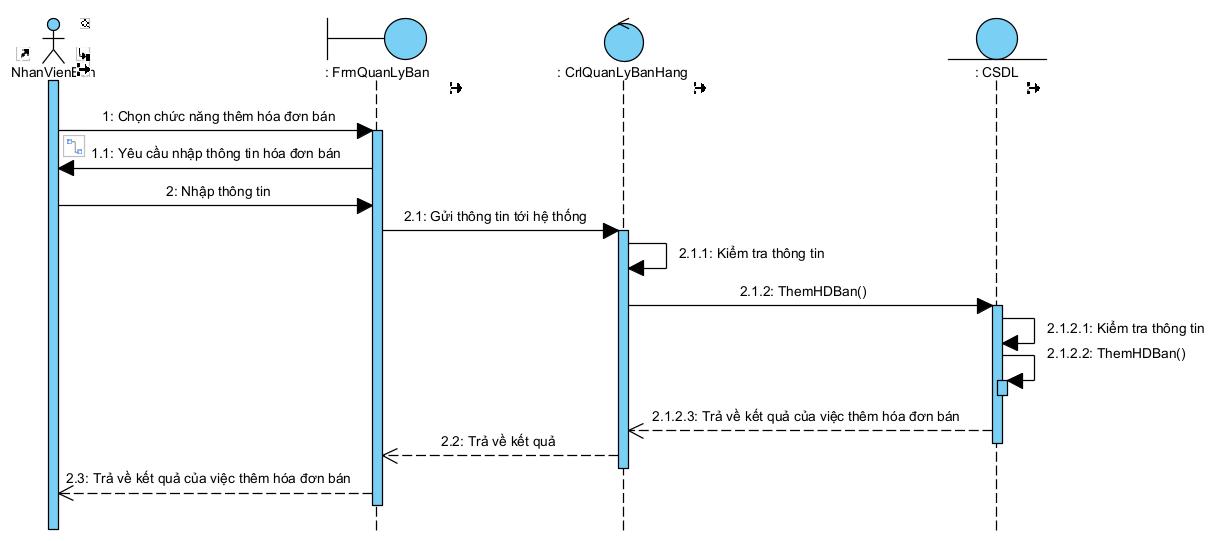
Hình 3.23: Biểu đồ tuần tự chức năng sửa thông tin khách hàng



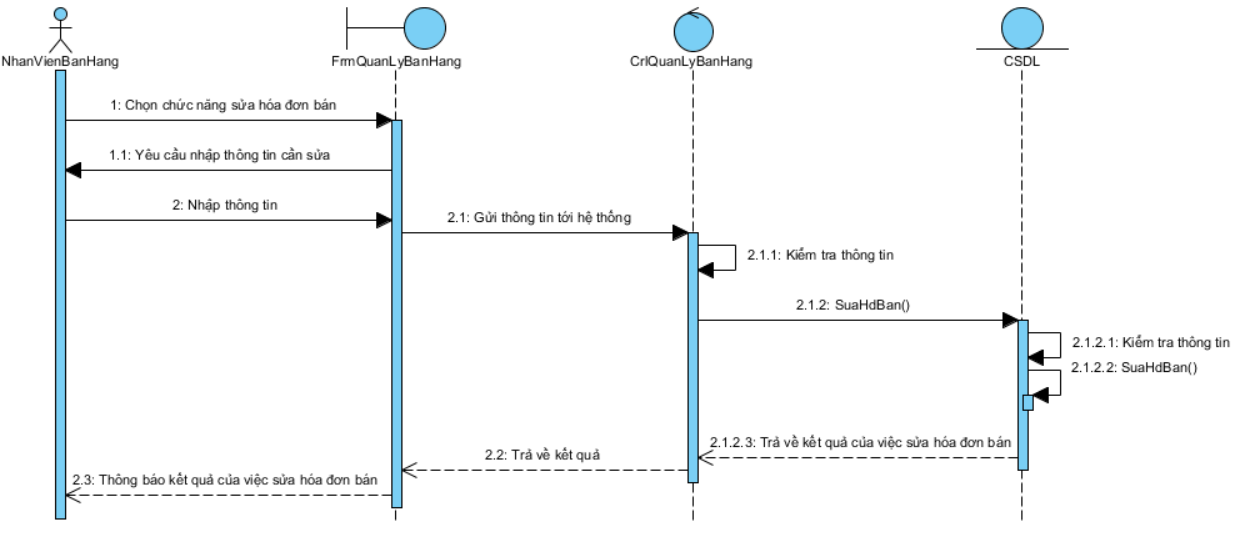
Hình 3.24: Biểu đồ tuần tự chức năng xóa thông tin khách hàng



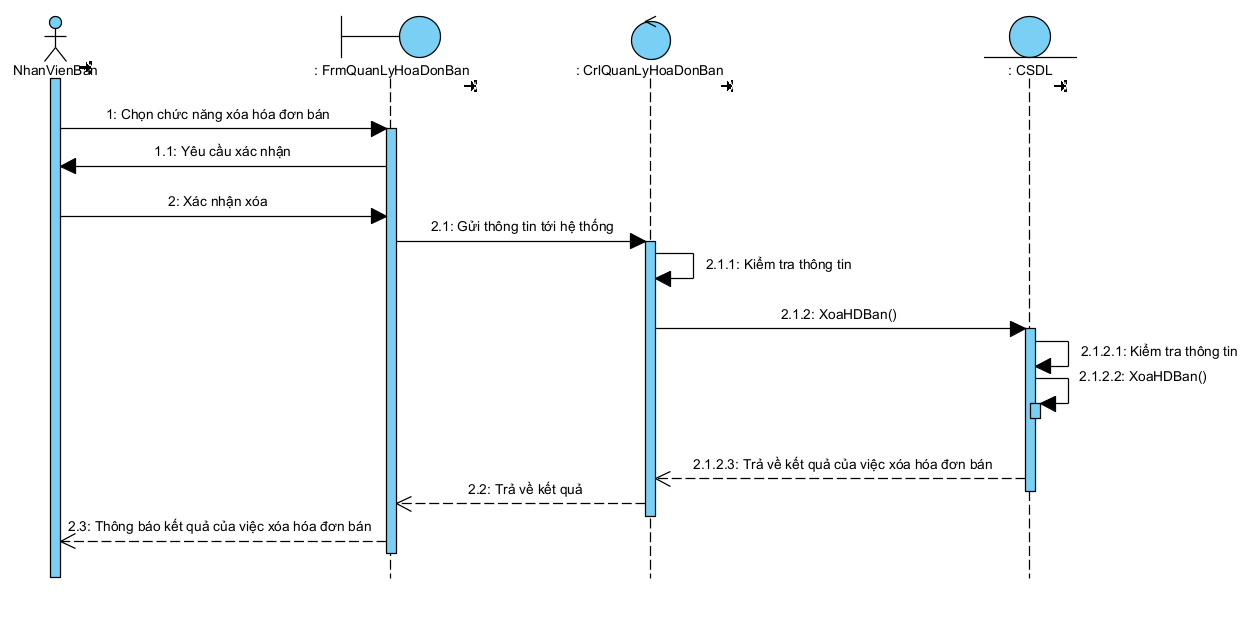
Hình 3.25: Biểu đồ tuần tự chức năng tìm kiếm thông tin khách hàng



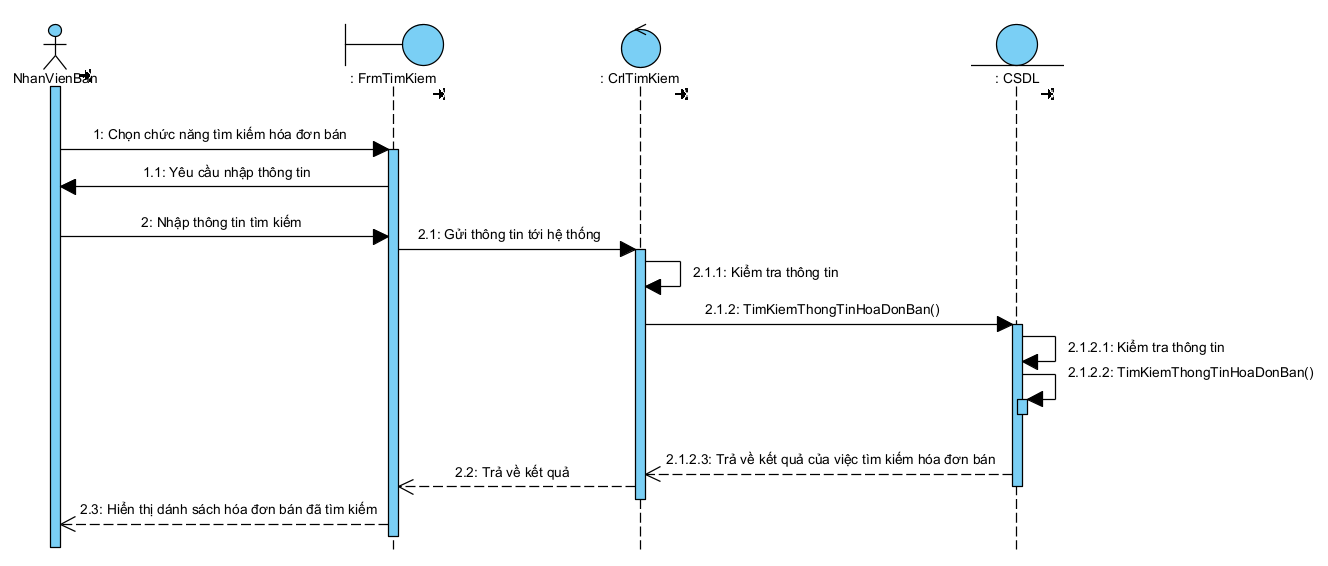
Hình 3.26: Biểu đồ tuần tự chức năng nhập thông tin đơn hàng



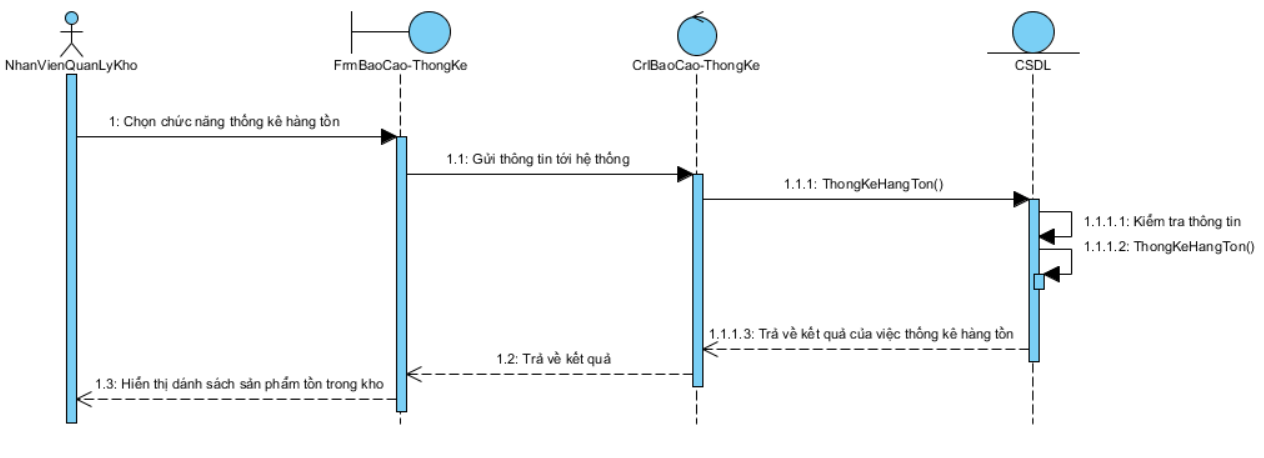
Hình 3.27: Biểu đồ tuần tự chức năng sửa thông tin đơn hàng



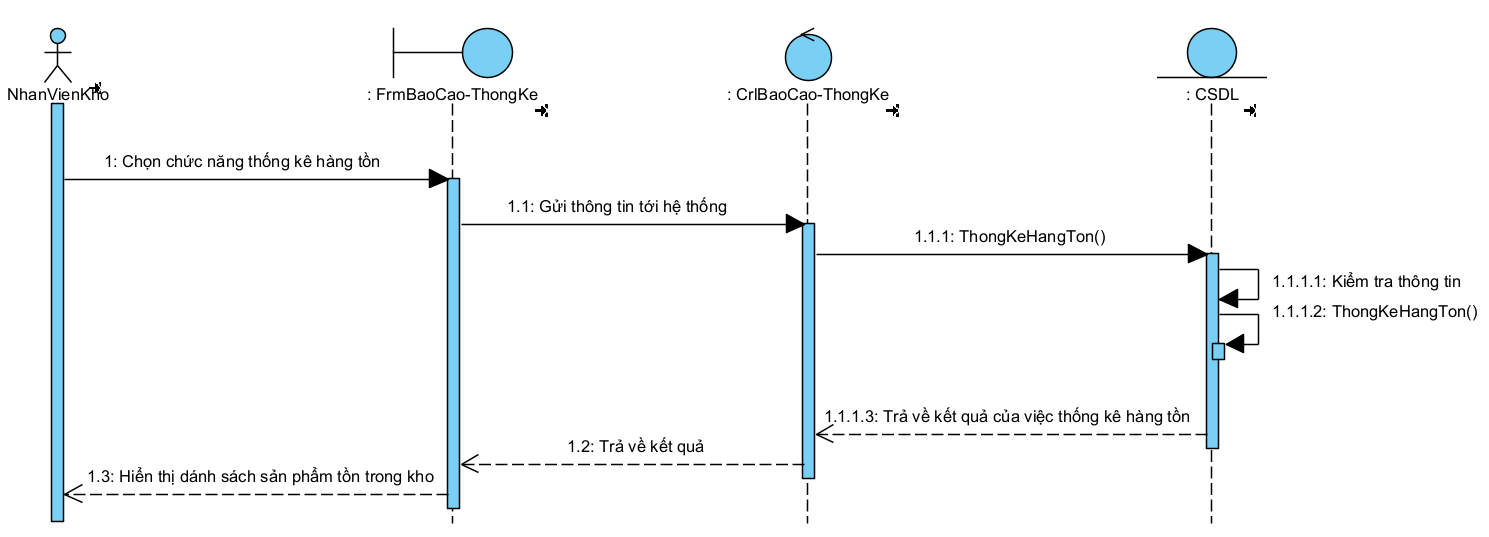
Hình 3.28: Biểu đồ tuần tự chức năng xóa thông tin đơn hàng



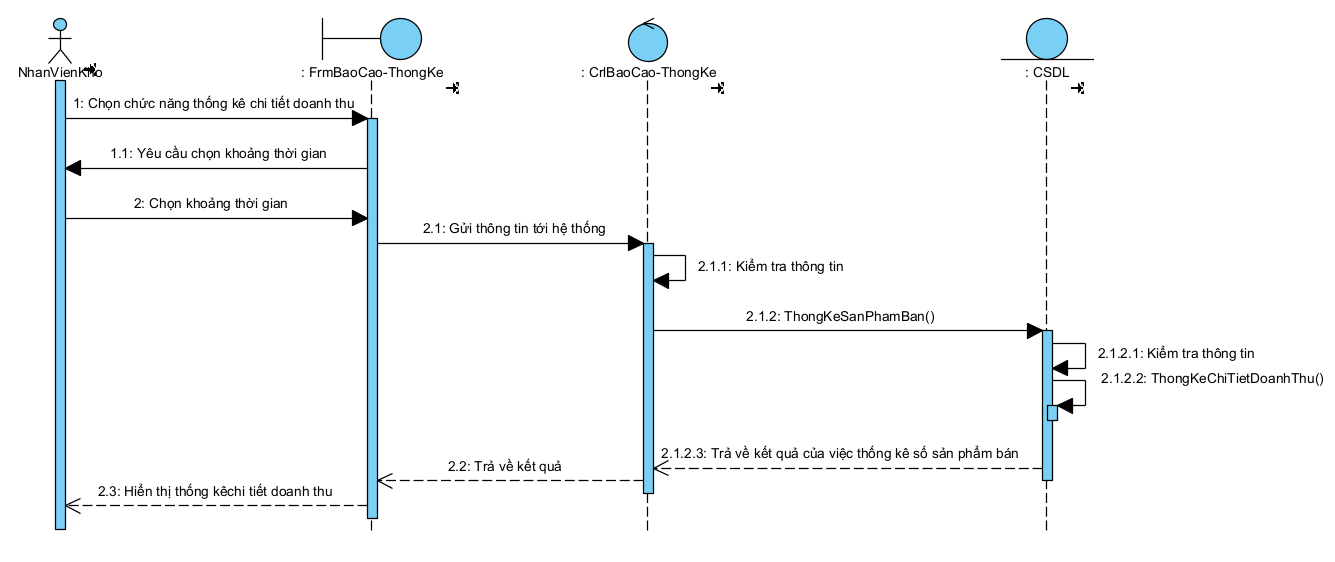
Hình 3.29: Biểu đồ tuần tự chức năng tìm kiếm thông tin đơn hàng



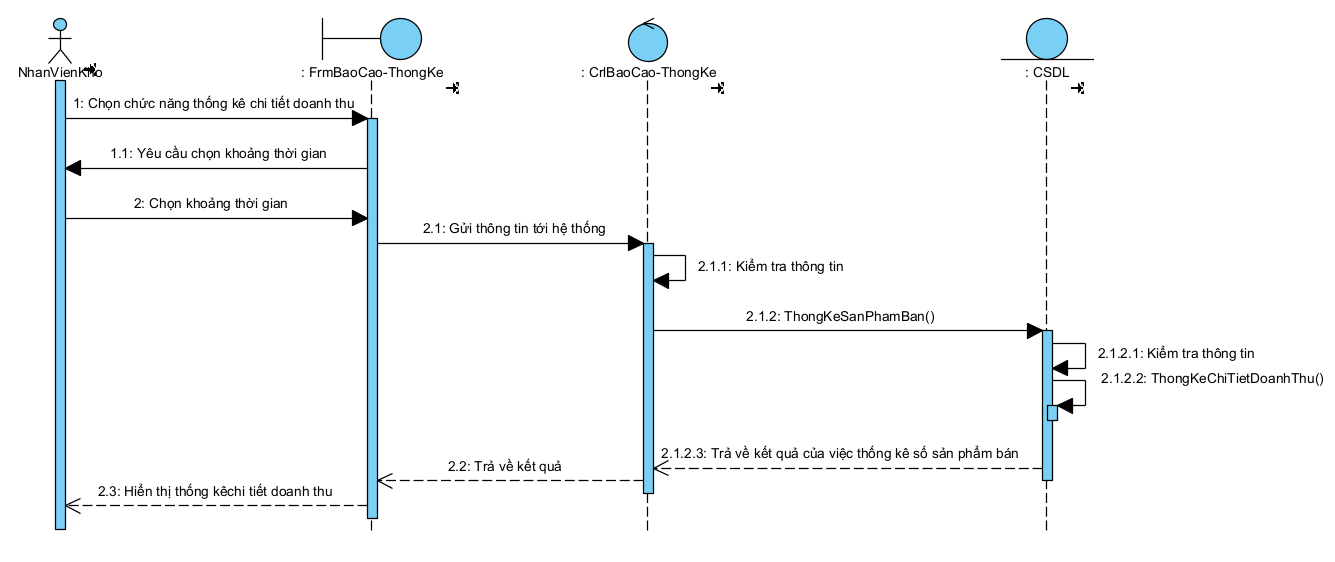
Hình 3.30: Biểu đồ tuần tự chức năng thống kê hàng tồn



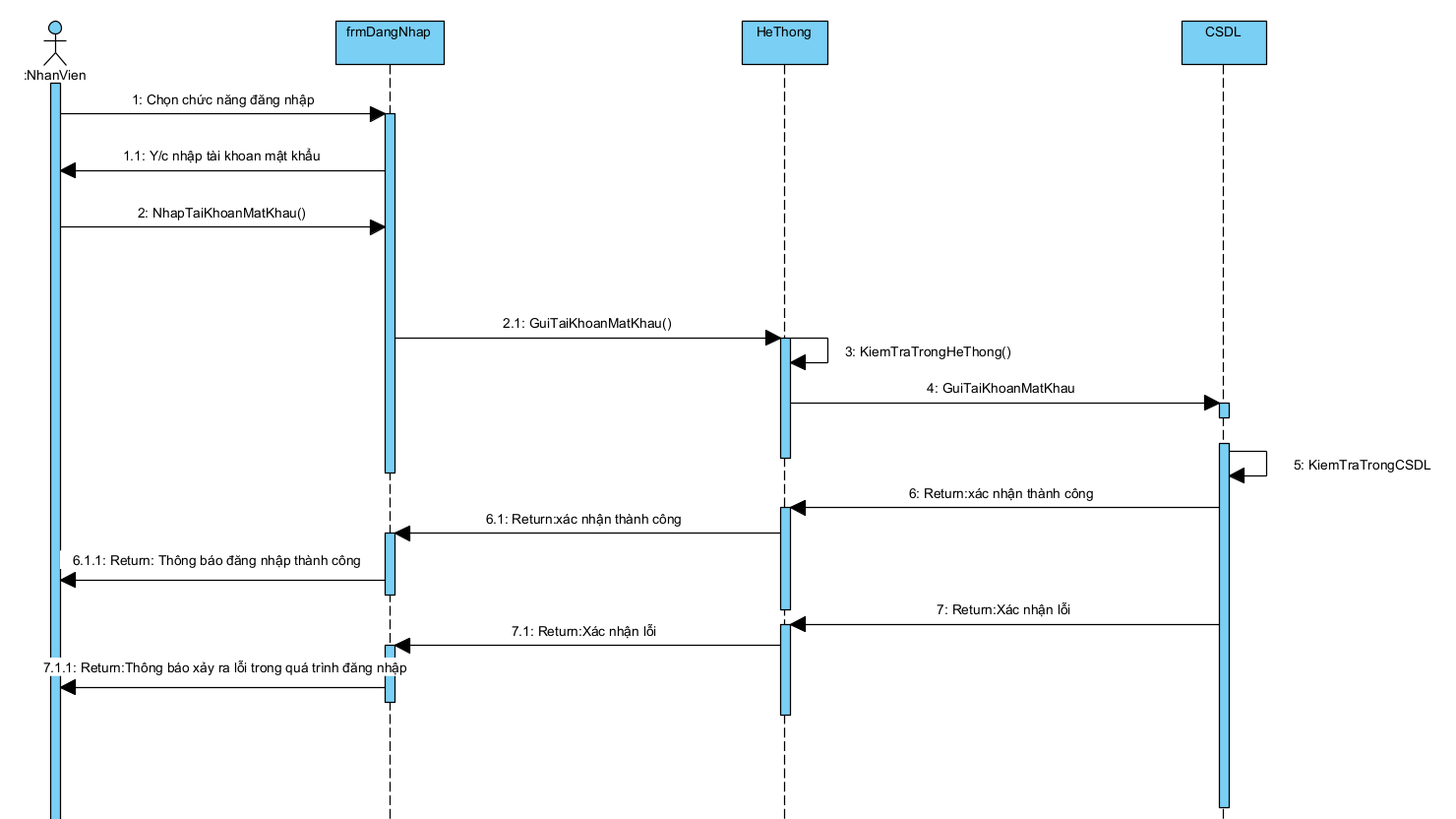
Hình 3.31: Biểu đồ tuần tự chức năng thống kê – đèn bán chạy



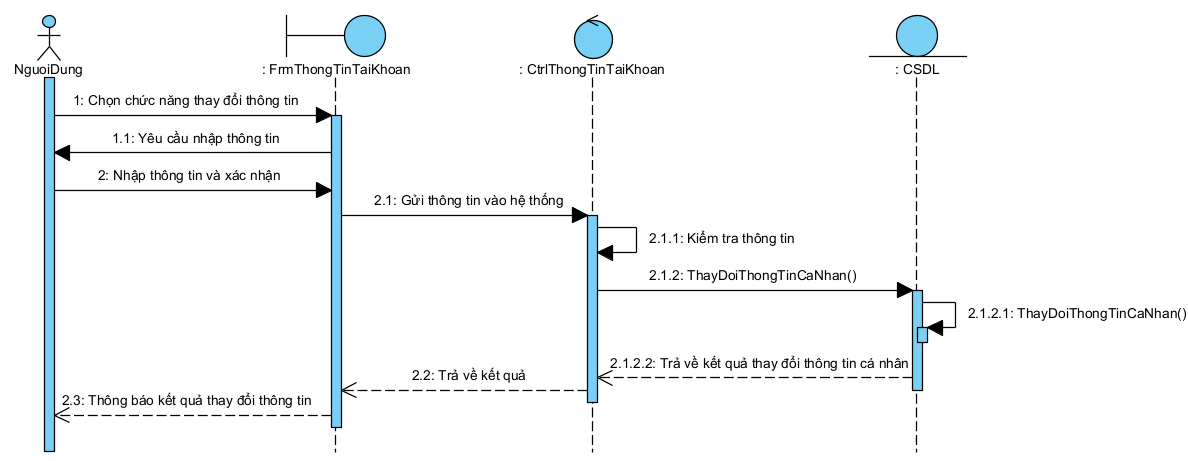
Hình 3.32: Biểu đồ tuần tự chức năng báo cáo doanh thu



Hình 3.33: Biểu đồ tuần tự chức năng thống kê chi tiết doanh thu



Hình 3.34: Biểu đồ tuần tự chức năng đăng nhập



Hình 3.35: Biểu đồ tuần tự chức năng thay đổi thông tin cá nhân



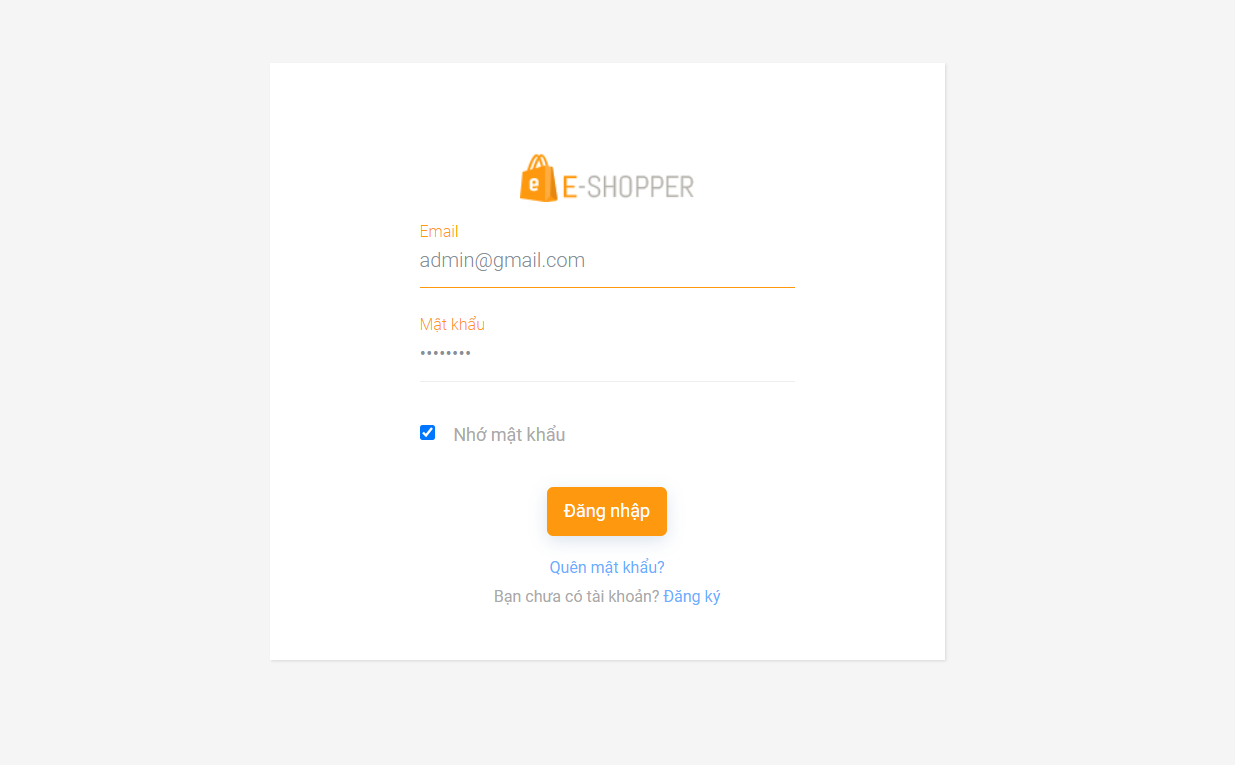
**Hình 3.36: Biểu đồ tuần tự chức năng thay đổi mật khẩu**

* + 1. Thiết kế giao diện

Ví dụ mô tả giao diện hệ thống:

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên giao diện** | **Chức năng** |
| **1** | Giao diện trang đăng nhập | Kiểm tra tính đúng đắn của tài khoản trước khi cho phép sử dụng các chức năng của chương trình. |
| **2** | Giao diện trang chủ | Hiển thị tất cả các chức năng chính của hệ thống. |
| **3** | Giao diện các trang chi tiết | Hiển thị tất cả các thông tin liên quan đến đèn trang trí và các lựa chọn theo nhu cầu của khách hàng . |
| **4** | Giao diện trang giỏ hàng | Hiển thị các đèn trang trí mà khách hàng đã chọn tùy theo nhu cầu của khách hàng có thể thêm, sửa, xóa. |
| **5** | Giao diện trang quản trị | Hiển thị các tính năng của người quản trị hệ thống. |
| **6** | Giao diện trang thanh toán | Hiển thị tất cả các chức năng mua hàng tiến hành thanh toán các sản phẩm. |

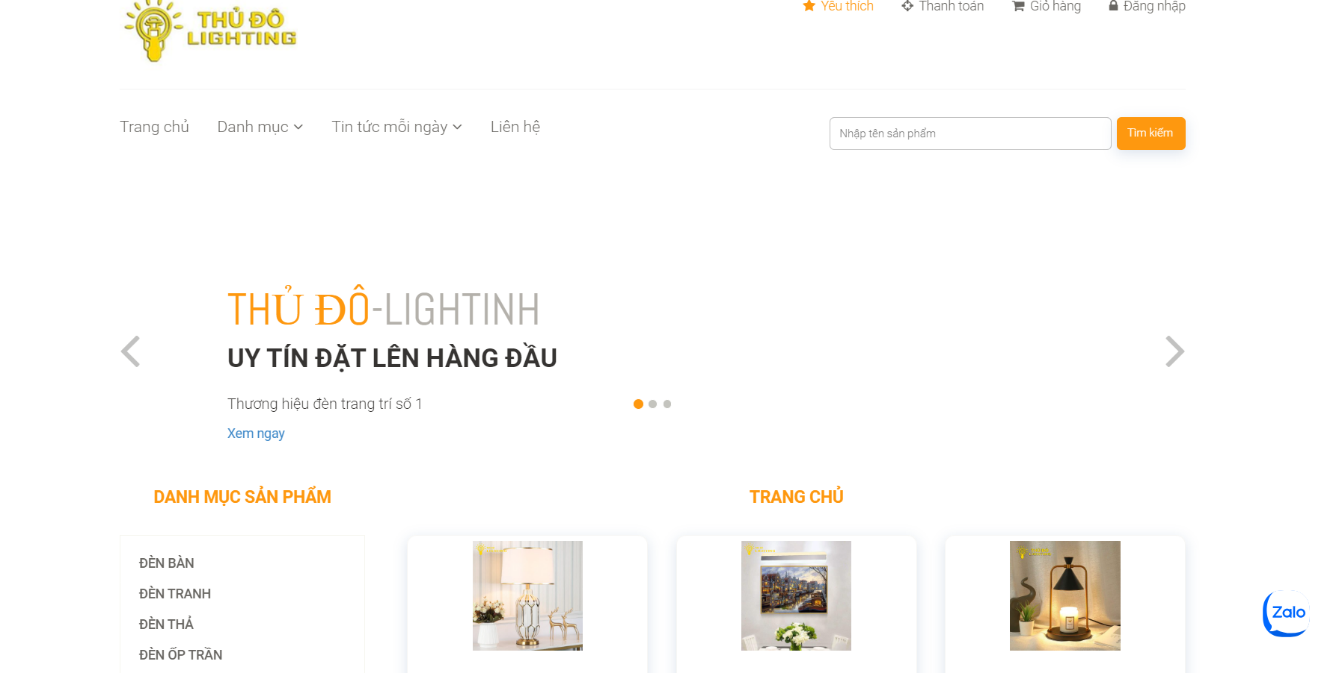
Giao diện trang đăng nhập



Hình 3.37 Giao diện trang đăng nhập

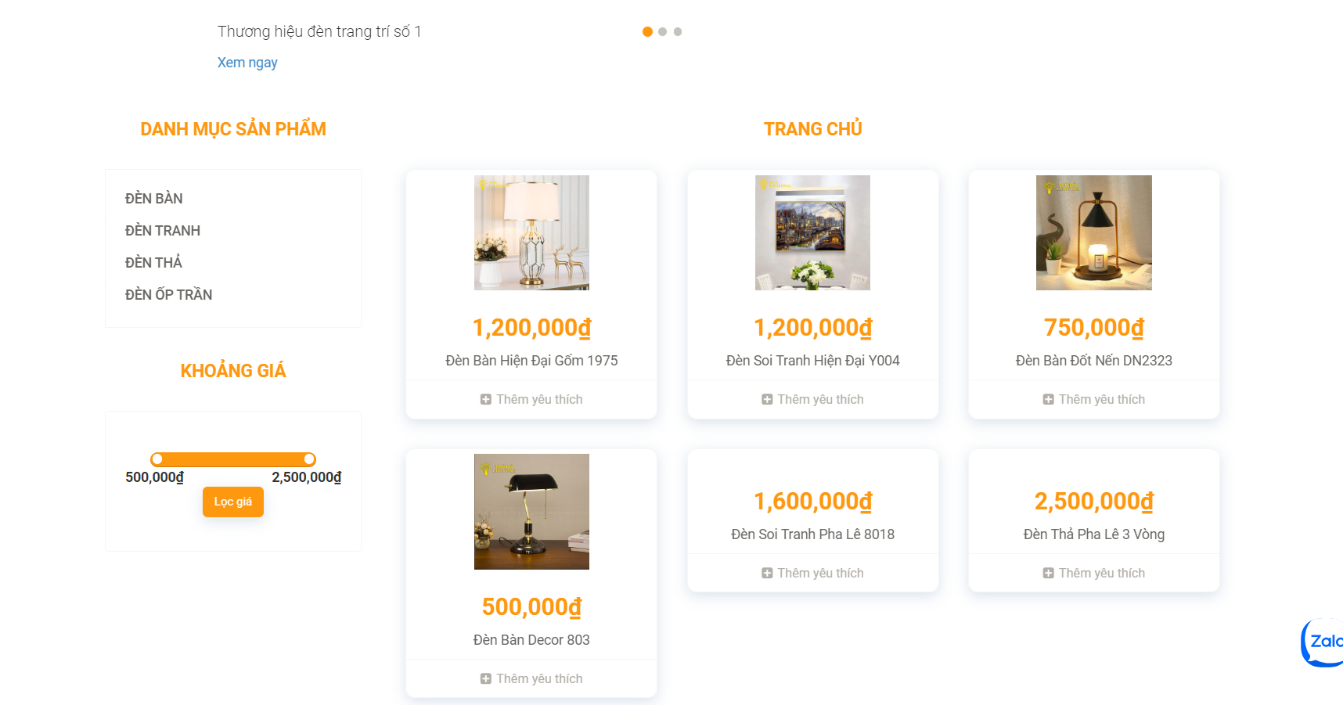
* Mô tả: Bao gồm các chức năng đăng nhập, ghi nhớ mật khẩu, quên mật khẩu.

Giao diện trang chủ



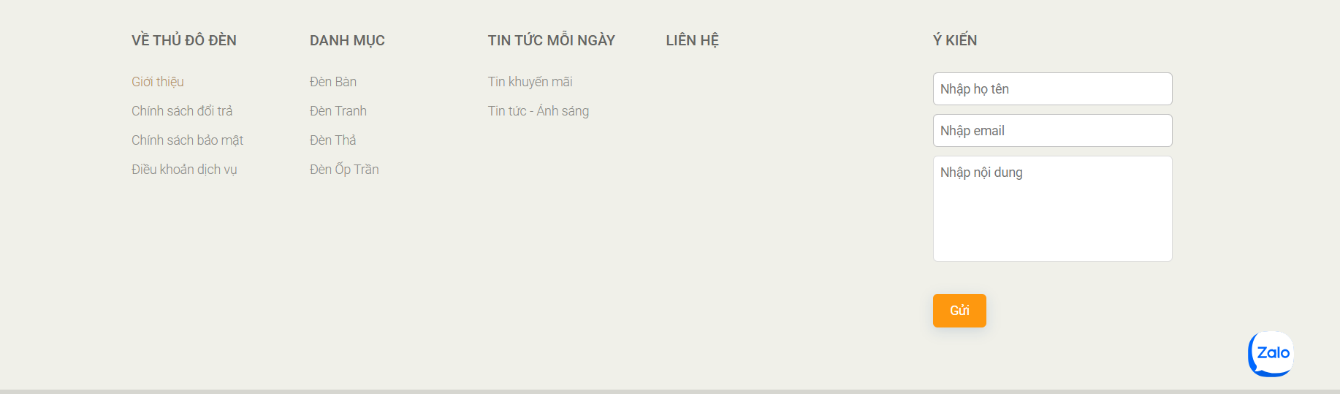
Hình 3.38 Giao diện header của trang chủ

* Mô tả: Header bao gồm logo, tìm kiếm, giỏ hàng, tài khoản(đăng kí hoặc đăng nhập).



Hình 3.39 Giao diện body của trang chủ

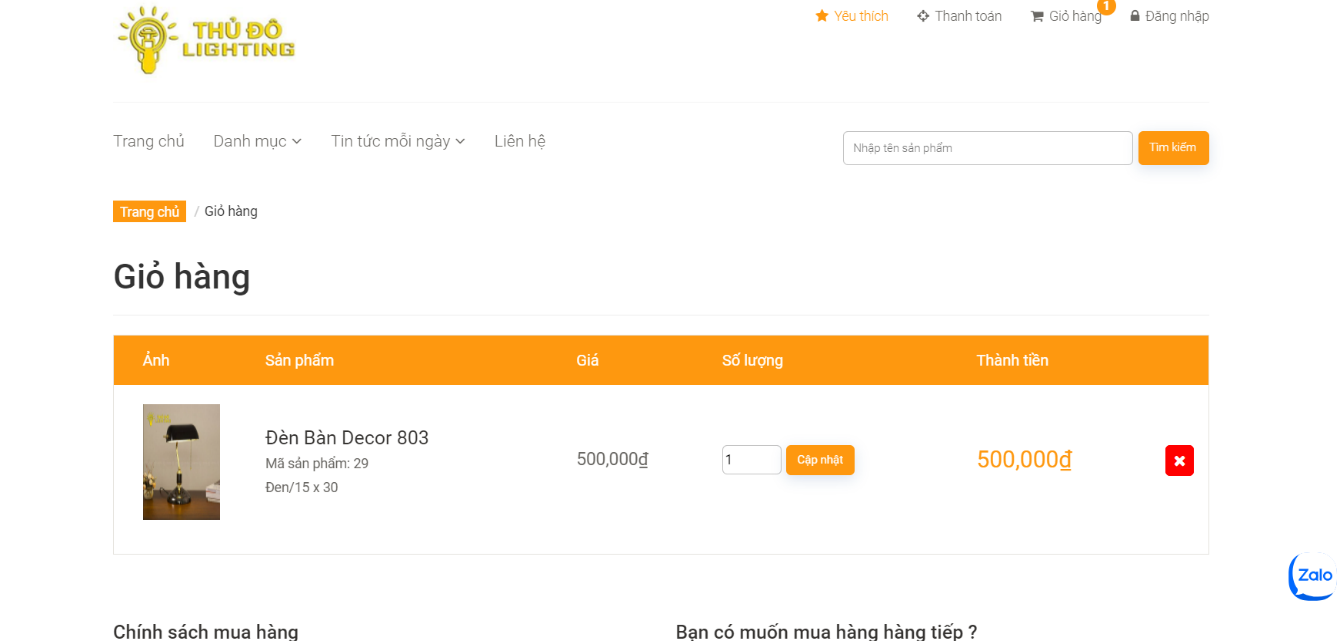
* Mô tả: Body bao gồm menu và hình ảnh các sản phẩm, tên sản phẩm, giá bán, các ưu đãi của sản phẩm.



Hình 3.40 Giao diện footer của trang chủ

* Mô tả: Footer bao gồm các dịch vụ liên quan như chăm sóc khách hàng, thanh toán, kết nối với các nền tảng mạng xã hội khác, tải ứng dụng trên các thiết bị và các thông tin liên quan đến cửa hàng.

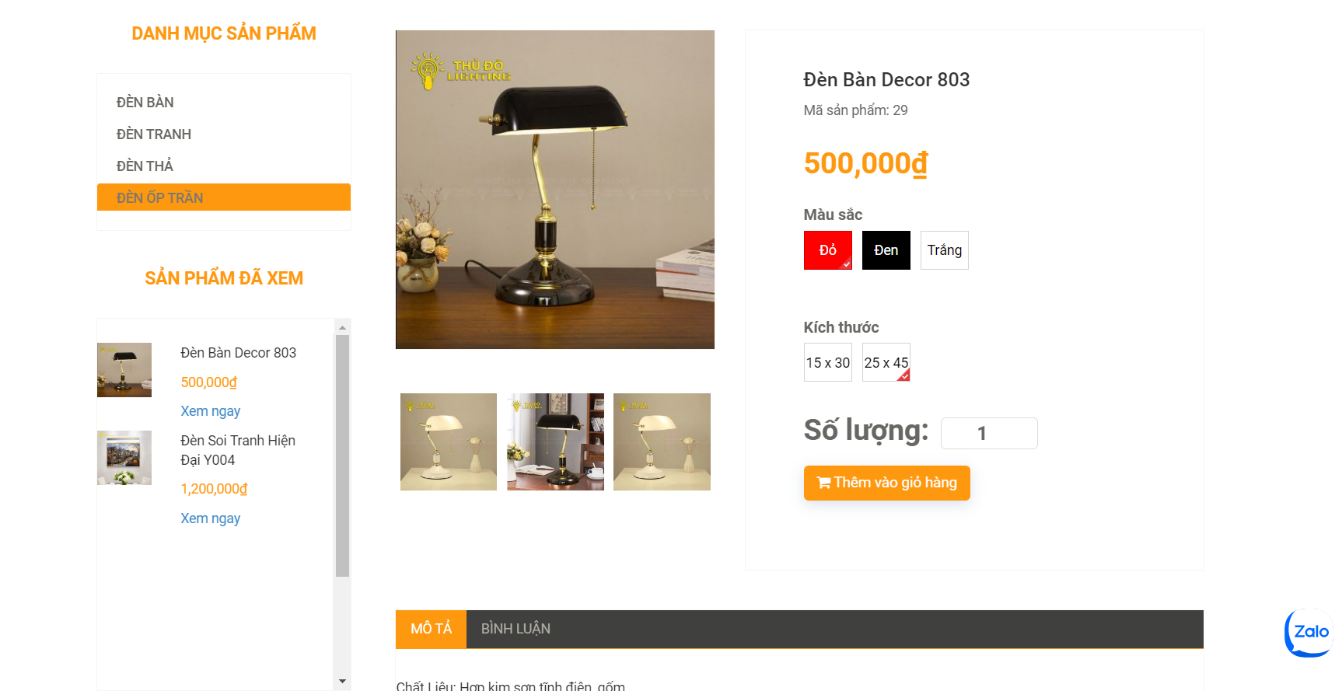
Giao diện trang giỏ hàng



Hình 3.41 Giao diện trang giỏ hàng

* Mô tả: Bao gồm danh sách các sản phẩm sau khi đã thêm vào giỏ hàng và có thể thực hiện thêm, cập nhật số lượng sản phẩm giỏ hàng và khách hàng có thể tiến hành mua hàng.

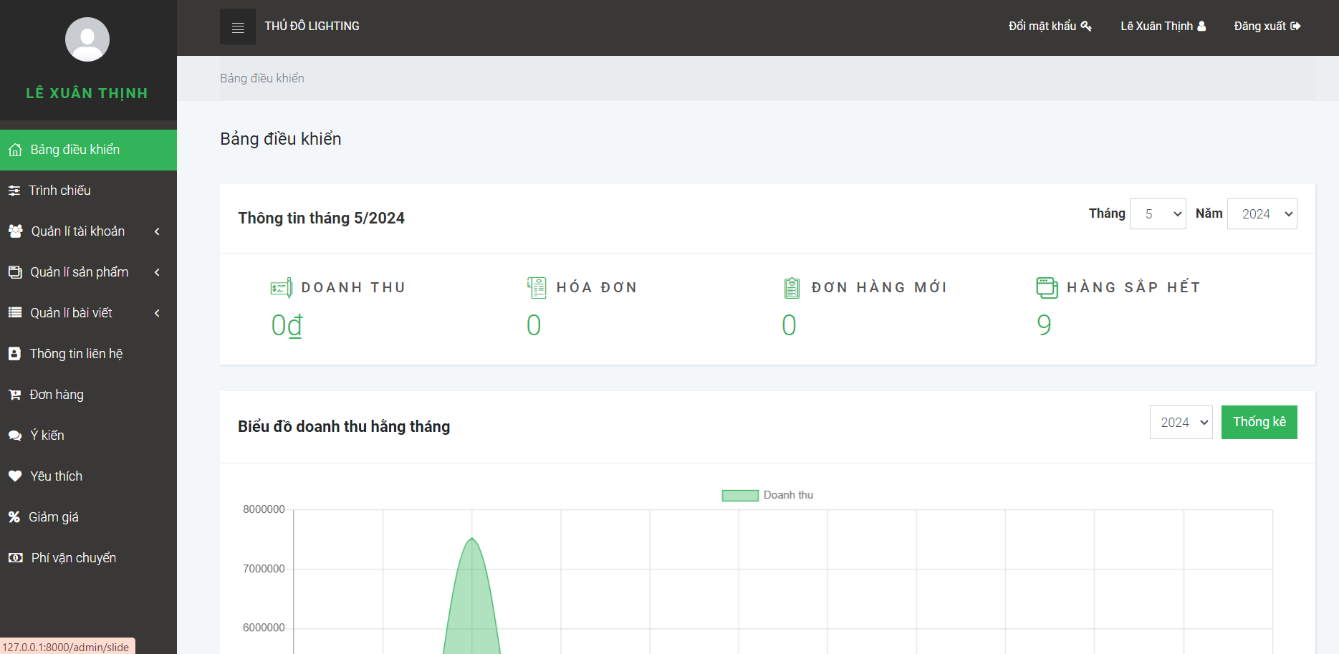
Giao diện trang chi tiết sản phẩm

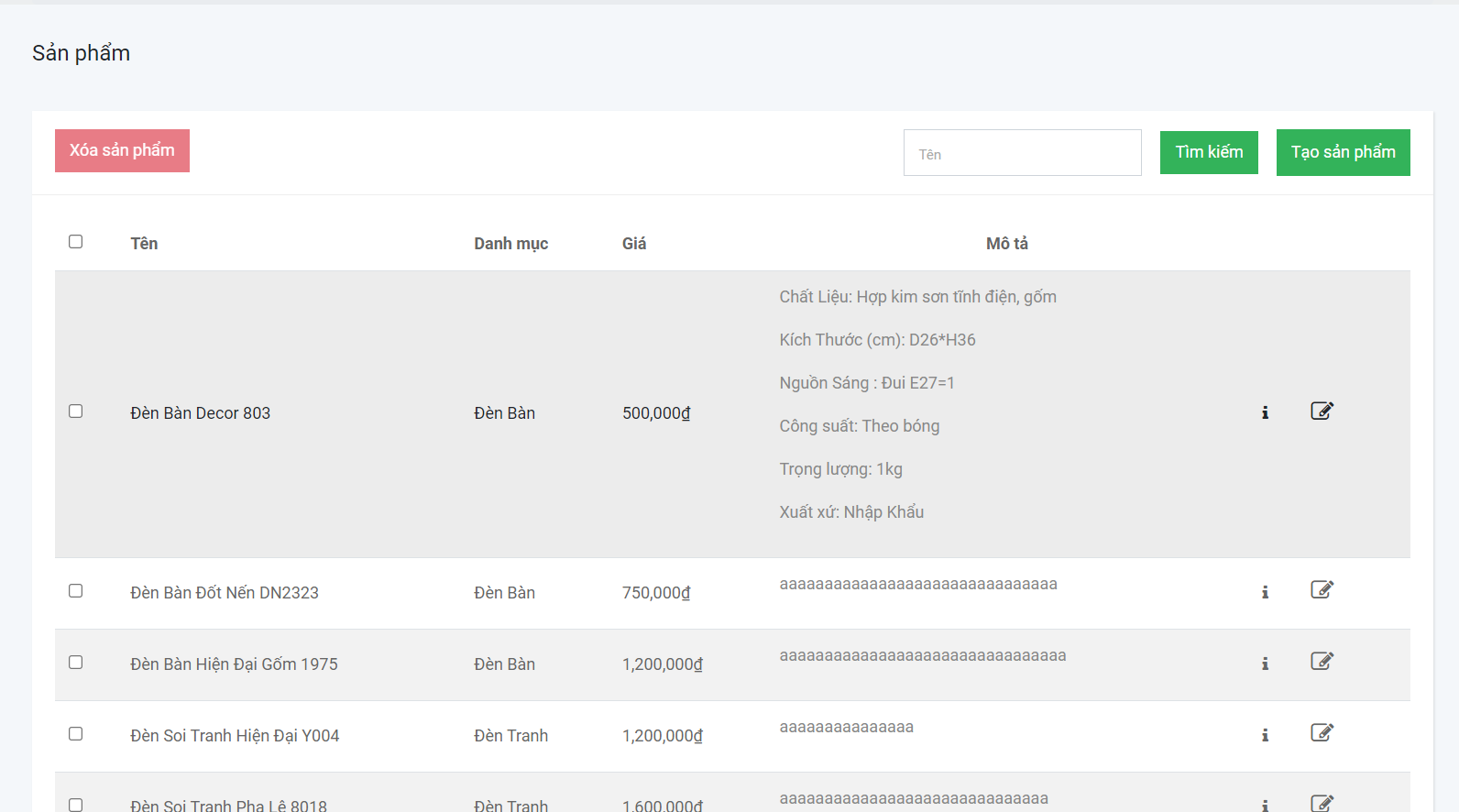


*Hình 3.42 Giao diện trang chi tiết sản phẩm*

* + Mô tả: Giúp khách hàng xem chi tiết thông tin về sản phẩm đèn trang trí và có thể thêm vào giỏ hàng các sản phẩm muốn mua.

Giao diện trang quản trị

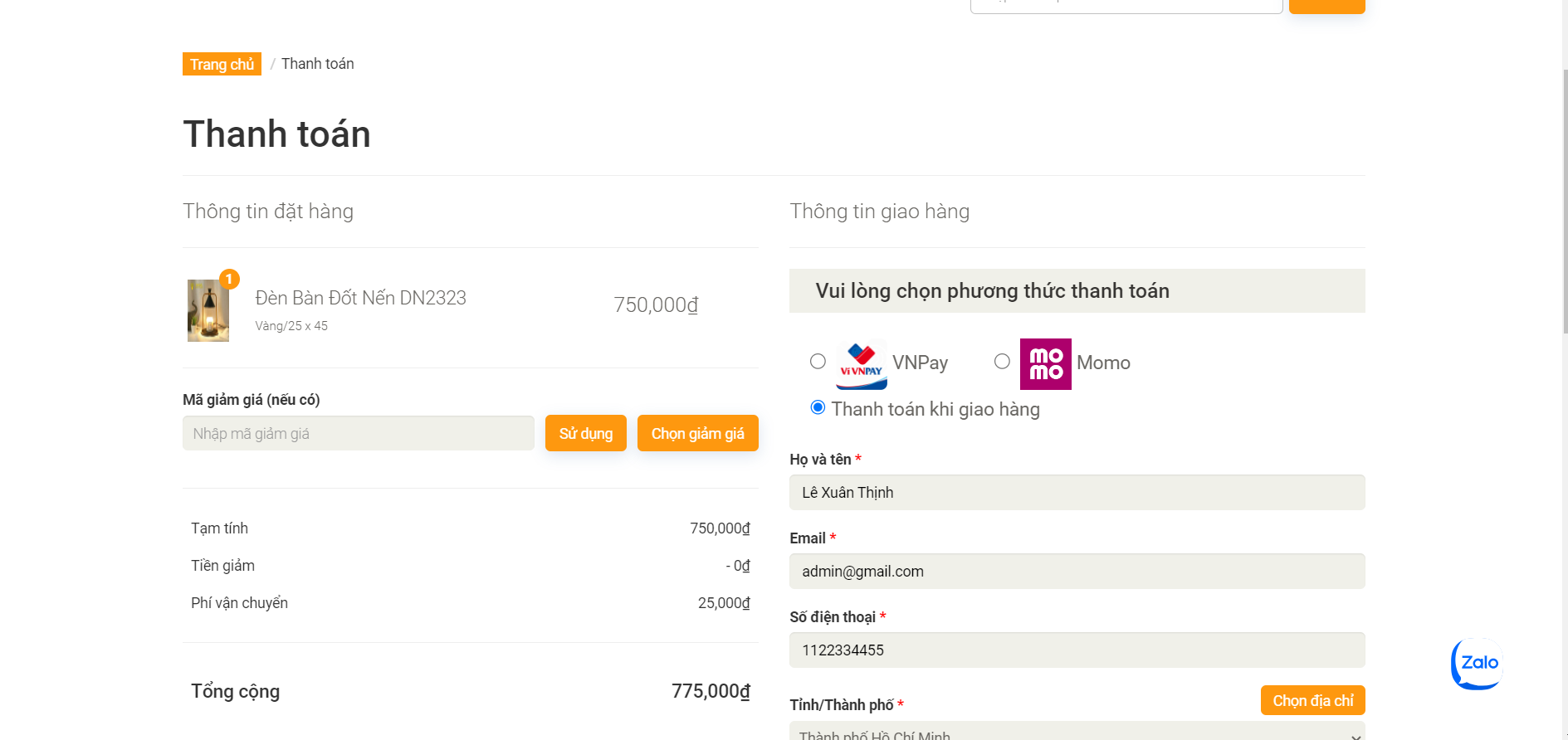




Hình 3.43 Giao diện trang quản trị

* Mô tả: Bao gồm những hoạt động riêng biệt của quản trị, thực hiện các chức năng quản lý website.

Giao diện trang thanh toán



Hình 3.44 Giao diện trang thanh toán

* Mô tả: Bao gồm các hoạt động nhập thông tin của khách hàng và hệ thống thanh toán các sản phẩm qua đó hiện thị đơn đặt hàng.

# TRIỂN KHAI WEBSITE

* 1. Triển khai các chức năng cho phân hệ người dùng
* Đăng nhập tài khoản: Người dùng cần có thể đăng nhập vào hệ thống để sử dụng các chức năng khác. Chức năng này cần cung cấp các trường thông tin tên đăng nhập và mật khẩu.
* Xem sản phẩm: Trang web cần hiển thị danh sách sản phẩm cho người dùng xem. Để làm được điều này, ta có thể sử dụng một danh sách các sản phẩm với ảnh minh họa, tên sản phẩm, giá cả và mô tả ngắn.
* Thêm sản phẩm vào giỏ hàng: Khi người dùng muốn mua một sản phẩm, họ cần có thể thêm sản phẩm đó vào giỏ hàng. Ta có thể tạo một nút "Thêm vào giỏ hàng" trên mỗi sản phẩm.
* Thanh toán sản phẩm: Người dùng nhập thông tin thanh toán, xem tổng thanh toán hóa đơn và nhấn nút “Đặt hàng” để tạo đơn hàng.

Để xây dựng được các chức năng của trang người dùng theo thiết kế đã được trình bày ở chương 3, bài tập lớn đã sử dụng HTML, CSS, và JavaScrip để thiết kế giao diện và thao tác dữ liệu các trang theo yêu cầu. Tiếp theo, bài tập lớn sẽ trình bày các kỹ thuật được sử dụng để xây dựng các trang.

* + 1. Trang chủ
* Tầng views



* + 1. Trang Đăng nhập
* Tầng views



4.1.3 Trang Giỏ hàng

* Tầng views



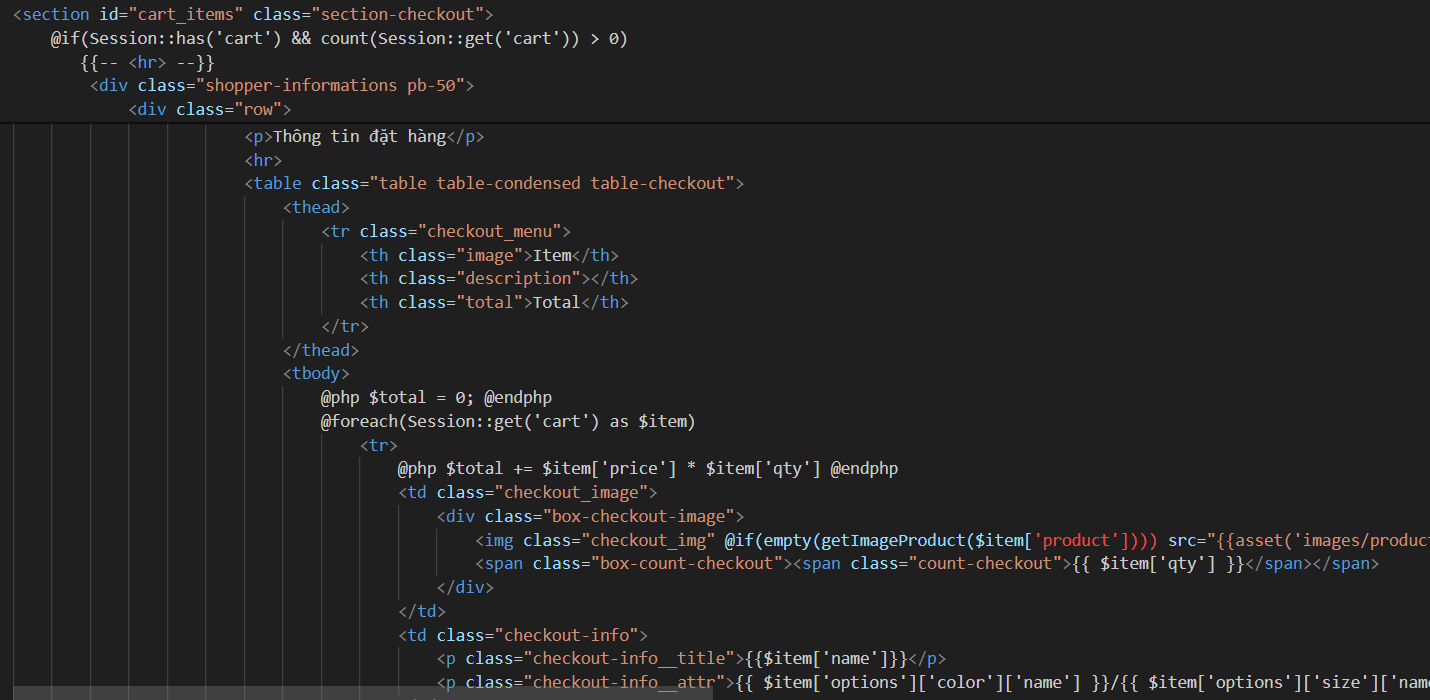
4.1.4 Trang Chi tiết sản phẩm

* Tầng views

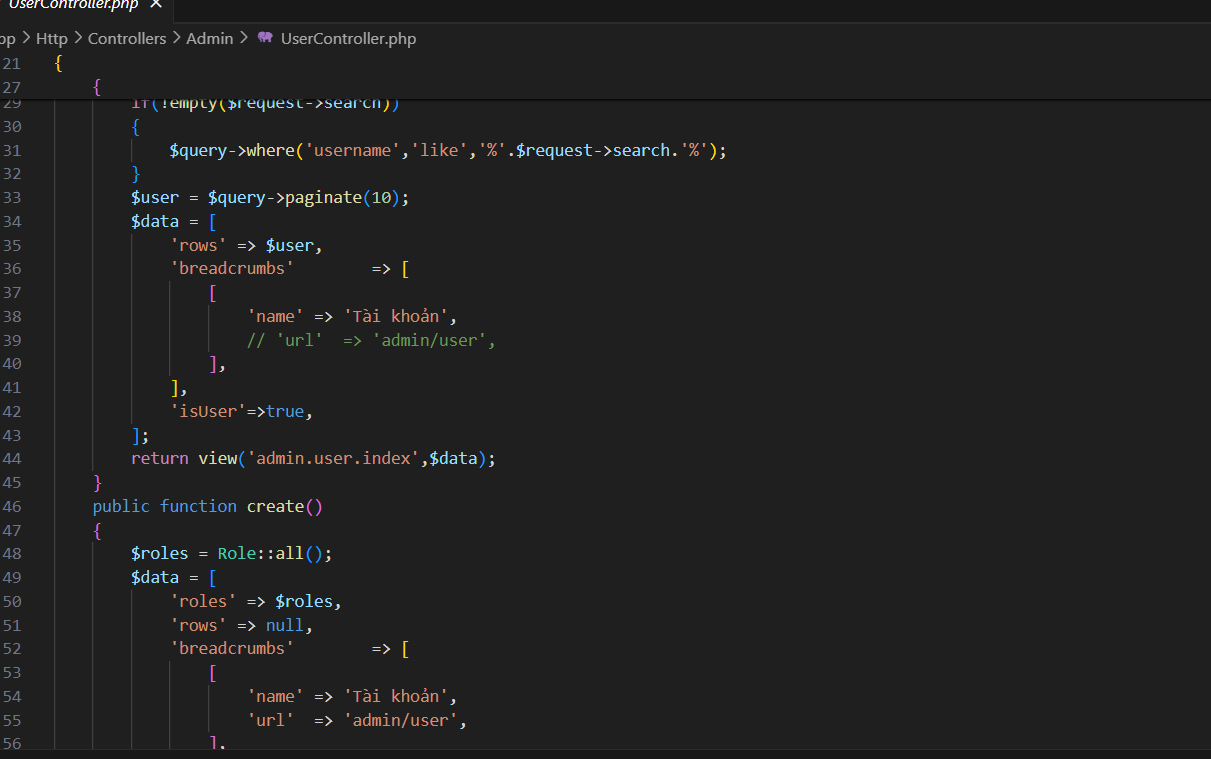


4.1.5 Trang Thanh toán

* Tầng views



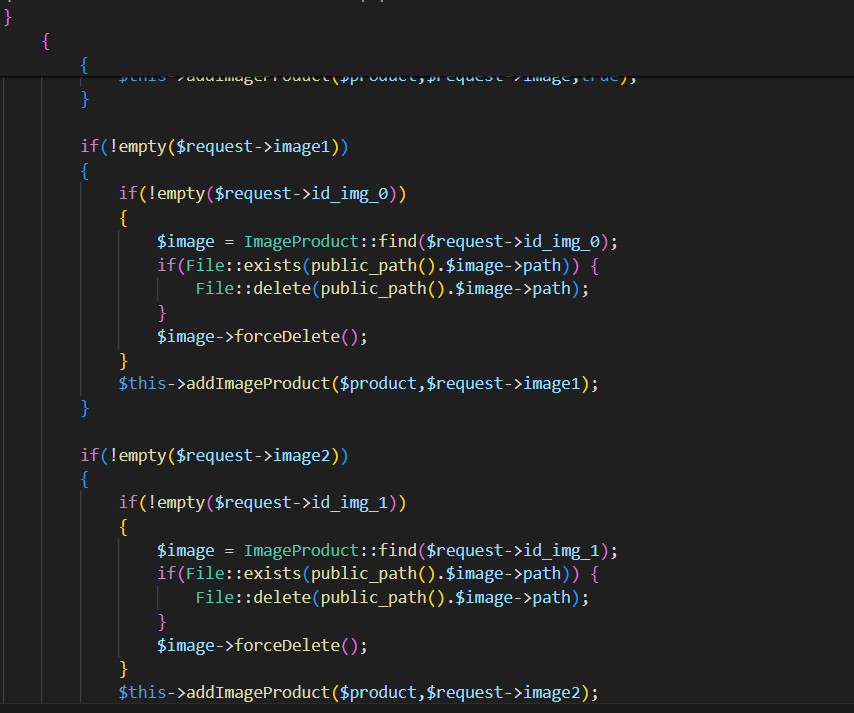
* 1. Triển khai các chức năng cho phân hệ quản trị nội dung
* Triển khai các lớp Controllers
* Quản lý tài khoản



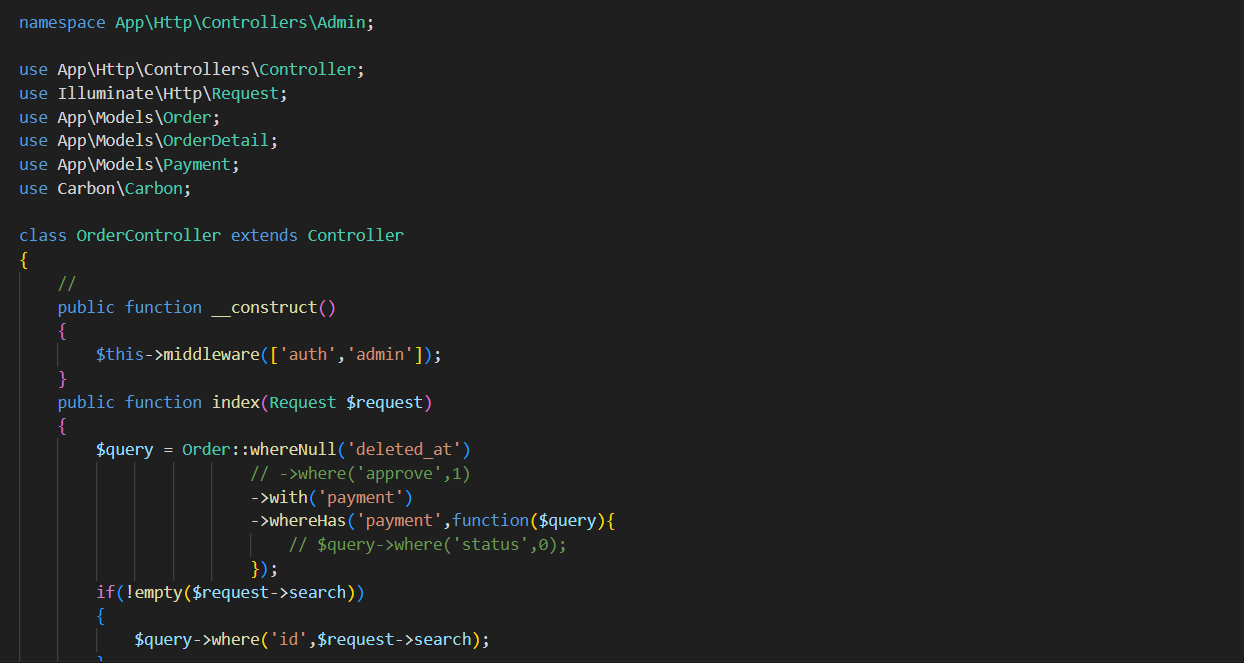
* Quản lý loại sản phẩm



* Quản lý sản phẩm



* Quản lý đơn hàng



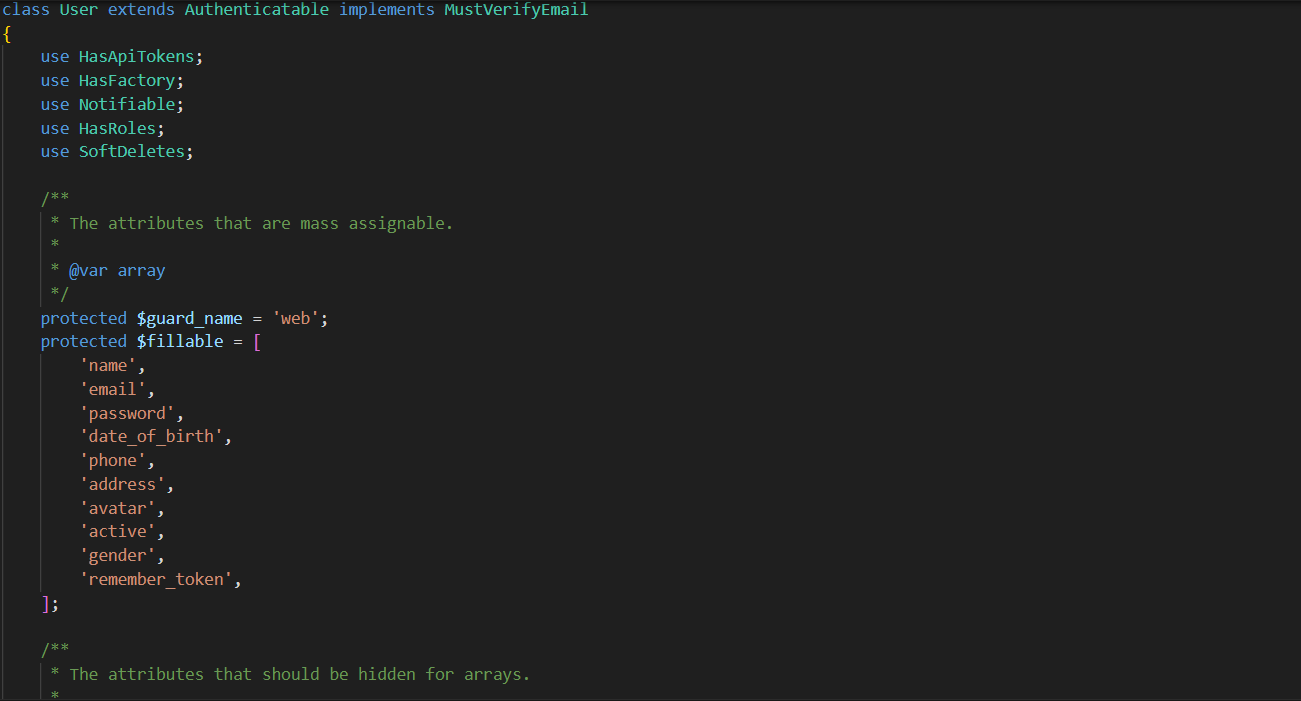
* Quản lý đơn hàng



* Quản lý chi tiết đơn hàng



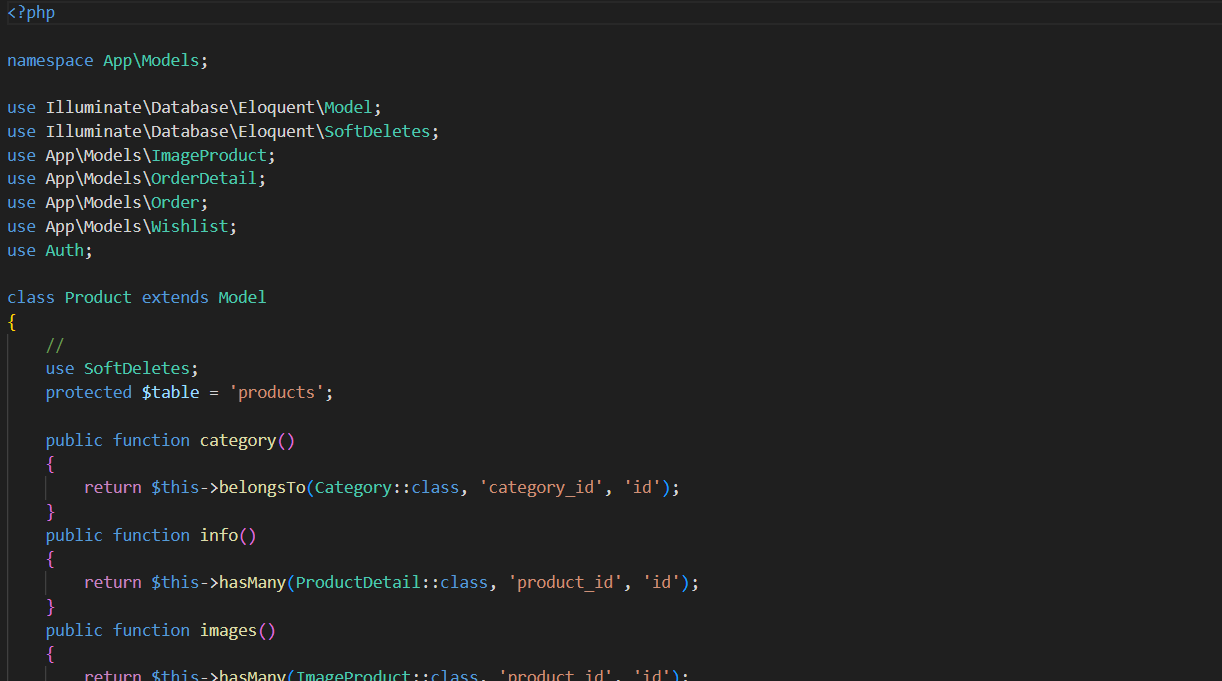
* Triển khai các lớp Models
* Quản lý tài khoản



* Quản lý loại sản phẩm



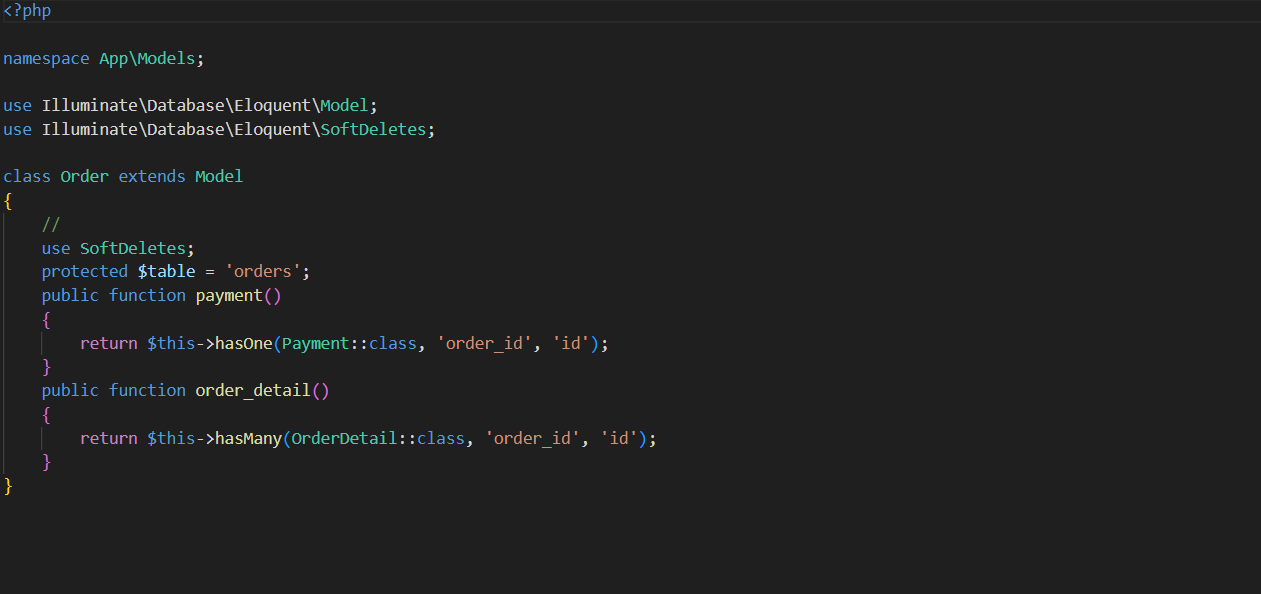
* Quản lý sản phẩm



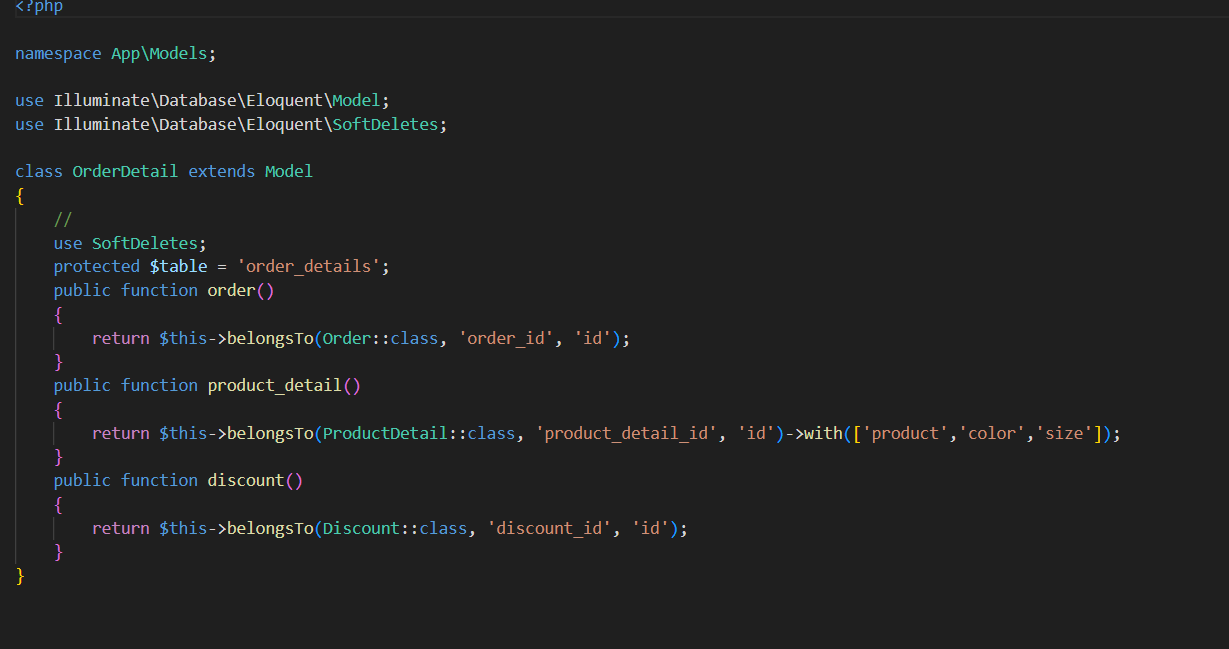
* Quản lý chi tiết sản phẩm



* Quản lý đơn hàng



* Quản lý chi tiết đơn hàng



* 1. Kiểm thử và triển khai ứng dụng
     1. Kiểm thử
* Kiểm thử tính khả dụng
* Giao diện hợp lí, ổn định.
* Kiểm thử chức năng
* Đăng nhập

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Input | Không tìm thấy tài khoản và mật khẩu trong Database | Tìm thấy tài khoản và mật khẩu trong Database | Tài khoản hoặc mật khẩu trống | Output |
| Tài khoản và mật khẩu | Xảy ra |  |  | Sai tài khoản hoặc mật khẩu |
| Tài khoản và mật khẩu |  | Xảy ra |  | Đăng nhập thành công |
| Tài khoản và mật khẩu |  |  | Xảy ra | Vui lòng nhập lại |

* Tìm kiếm sản phẩm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Input | Thông tin sản phẩm đã tồn tại | Thông tin sản phẩm chưa tồn tại | Thông tin sản phẩm rỗng | Out put |
| Nhập thông tin tìm kiếm | Xảy ra |  |  | Hiển thị thông tin vừa tìm |
| Nhập thông tin tìm kiếm |  | Xảy ra |  | Không tìm thấy thông tin muốn tìm |
| Nhập thông tin tìm kiếm |  |  | Xảy ra | Vui lòng nhập lại |

* Thêm sản phẩm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Input | Đối tượng đã tồn tại | Đối tượng chưa tồn tại | Thông tin bắt buộc bỏ trống | Output |
| Thông tin đối tượng | Xảy ra |  |  | Sản phẩm đã tồn tại |
| Thông tin đối tượng |  | Xảy ra |  | Thêm sản phẩm thành công |
| Thông tin đối tượng |  |  | Xảy ra | Các mục không được bỏ trống |

* Sửa sản phẩm

|  |  |  |  |
| --- | --- | --- | --- |
| Input | Đối tượng đã tồn tại | Thông tin bắt buộc bỏ trống | Output |
| Thông tin đối tượng | Xảy ra |  | Sửa thành công |
| Thông tin đối tượng |  | Xảy ra | Các mục không được bỏ trống |

4.3.2 Đóng gói ứng dụng

* Nén project source (.zip)
* Xuất file datable (.mysql)

4.3.3 Triển khai ứng dụng

* Hệ điều hành: Windows 11
* Web hosting control panel: Aapanel
* Công cụ phát triển: Visual Studio 2010 trở lên

# KẾT LUẬN

* Kết quả đạt được
* Giao diện thân thiện, bắt mắt, bố cục cân đối, màu sắc hài hòa.
* Xử lý được nhiều dữ liệu đầu vào, đầu ra.
* Bố trí được file tệp khoa học, dễ dàng phát hiện cũng như sửa lỗi khi hệ thống gặp vấn đề.
* Tương tác tốt giữa người dùng và hệ thống.
* Xử lý tốt các điều khiển thao tác đến dữ liệu trong cơ sở dữ liệu, đảm bảo tính logic.
* Hạn chế của đề tài
* Chưa hoàn thành được toàn bộ các chức năng phân hệ quản trị nội dung (về cơ bản đã hoàn thiện được một số chức năng chính).
* Chưa làm được chức năng thanh toán.
* Hướng phát triển của đề tài.
* Tiếp tục hoàn thiện đầy đủ các chức năng phái quản trị nội dung.
* Bổ sung các sản phẩm mới để hệ thống thêm sinh động, phong phú.
* Đưa hệ thống lên Internet để có nhiều người dùng tiếp cận và trải nghiệm.

TÀI LIỆU THAM KHẢO

x

1. Nguyễn Văn Quyết, Giáo trình Công nghệ Web và ứng dụng. Hưng Yên, Việt Nam: Khoa Công nghệ Thông tin - UTEHY, 2010.
2. Giáo trình Thiết kế web với HTML& CSS, Khoa Công nghệ thông tin, Trường Đại học Sư phạm Kỹ thuật Hưng Yên.
3. https://www.youtube.com/
4. <https://www.w3schools.com/>
5. https://www.laravel.com/

1. [↑](#footnote-ref-1)