|  |  |
| --- | --- |
|  |  |
| NGUYỄN THỊ THU HÀ | **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **---------------------------------------** |
|  |
| ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC  NGÀNH HỆ THỐNG THÔNG TIN |
|  |
| **Xây dựng Website bán trang sức cửa hàng Thu Hà** |
|  |
|  |
| **CBHD: Th.S Nguyễn Lan Anh** |
| HỆ THỐNG THÔNG TIN | **Sinh viên: Nguyễn Thị Thu Hà** |
| **Mã số sinh viên: 2019604942** |
|  |
|  |
|  |
| Hà Nội – Năm 2023 |
|  |

**MỤC LỤC**

[LỜI NÓI ĐẦU ii](#_Toc102342478)

[DANH MỤC CÁC CHỮ VIẾT TẮT iii](#_Toc102342479)

[DANH MỤC BẢNG BIỂU iv](#_Toc102342480)

[DANH MỤC HÌNH ẢNH v](#_Toc102342481)

[CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI 1](#_Toc102342482)

[1.1. Tên đề tài 1](#_Toc102342483)

[1.2. Lý do chọn đề tài 1](#_Toc102342484)

[1.3. Mục tiêu của đề tài 3](#_Toc102342485)

[1.4. Đối tượng và phạm vi 4](#_Toc102342486)

[1.5. Kết quả dự kiến đạt được 4](#_Toc102342487)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT VÀ CÔNG CỤ SỬ DỤNG 5](#_Toc102342488)

[2.1. Cơ sở lý thuyết 5](#_Toc102342489)

[2.2. Công cụ sử dụng 20](#_Toc102342490)

[CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 22](#_Toc102342491)

[3.1. Phân tích hệ thống 22](#_Toc102342492)

[3.2. Biểu đồ use case tổng quát 24](#_Toc102342493)

[3.3. Mô hình thực thể liên kết 25](#_Toc102342494)

[3.4. Đặc tả use case 26](#_Toc102342495)

[3.5. Thiết kế hệ thống 79](#_Toc102342496)

[KẾT LUẬN 94](#_Toc102342497)

[TÀI LIỆU THAM KHẢO 96](#_Toc102342498)

# LỜI NÓI ĐẦU

Mạng Internet, là một trong những công cụ không thể thiếu trong cuoojc sống hiện đại, là nền tảng chính cho sự truyền tải và trao đổi thông tin trên toàn cầu. Cùng với sự phát triển nhảy vọt của công nghệ thông tin, Internet ngày càng giữ vai trò quan trọng trong mọi mặt của đời sống. Theo đó, việc xây dựng những website bán hàng hay phát triển lĩnh vực thương mại điện tử là một điều vô cùng cần thiết để doanh nghiệp của bạn được nhiều người biết tới hơn.

Sau đại dịch Covid-19, mua bán hàng online đang có xu hướng phát triển mạnh, con người có thể mua sắm các thiết bị và nhu yếu phẩm thông qua các sàn thương mại điện tử. Đặc biệt là nhu cầu về mua đồ trang sức, do đó hệ thống hỗ trợ mua sắm online cũng là một trong những vấn đề đang được quan tâm. Việc xây dựng website thương mại điện tử bán trang sức góp phần giúp người dùng dễ dàng tìm hiểu và đặt mua những đồ dùng một cách dễ dàng.

Sau khi tìm hiểu và dưới sự hướng dẫn của cô Nguyễn Lan Anh, em đã phân tích và thiết kế được **Website bán trang sức cửa hàng Thu Hà** . Để giúp khách hàng đặt mua được những sản phẩm theo nhu cầu và ý thích của mình, trước hết cần có một hệ thống tốt, một website có các dữ liệu chính xác, các đánh giá tốt của người dùng. Đó cũng là lý do mà em chọn đề tài này.

Để hoàn thành được đồ án tốt nghiệp này, em xin được gửi lời cảm ơn chân thành đến các thầy cô trong khoa Công nghệ thông tin Trường Đại học Công Nghiệp Hà Nội đã tận tình giảng dạy và trang bị kiến thức cho em trong suốt thời gian em học tập tại trường. Cô giáo hướng dẫn đề tài - **Thạc Sĩ Nguyễn Lan Anh**, giảng viên Khoa Công nghệ Thông tin Trường Đại học Công Nghiệp Hà Nội – đã tận tụy hết lòng giúp đỡ, hướng dẫn, chỉ dẫn tận tình để giúp em hoàn thành được đồ án. Bạn bè đã góp ý và giúp đỡ, bên cạnh động viên và khích lệ tôi trong suốt thời gian tôi làm đồ án.

***Hà Nội, Ngày 26 tháng 05 năm 2022***

# DANH MỤC CÁC CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| Từ viết tắt | Giải thích |
| JRE | Java Runtime Environment |
| API | Application Programming Interface |
| OOP | Object Oriented Programming |
| ANSI | American National Standards Institute |
| POSIX | Portable Operating System Interface |
| HTML | HyperText Markup Language |
| CSS | Cascading Style Sheets |
| JSF | JavaServer Faces |
| IBM | International Business Machines |
| AOP | Aspect Oriented Programming |
| JDBC | Java Database Connectivity |
| POJO | Plain Old Java Object |
| JMS | Java Message Service |
| JTA | Java Transaction API |
| JMX | Java Management Extensions |
| RMI | Remote Method Invocation |
| MVC | Model-View-Controller |
| UI | User Interface |
| GUI | Graphical User Interface |
| AMQP | Advanced Message Queuing Protocol |
| XML | Extensible Markup Language |
| MVCC | Multiversion concurrency control |
| PK | Primary key |
| PFK | Primary Foreign Key |

# DANH MỤC BẢNG BIỂU

[Bảng 3.1. Bảng Category 81](#_Toc102342540)

[Bảng 3.2. Bảng Product 81](#_Toc102342541)

[Bảng 3.3. Bảng ProductImage 82](#_Toc102342542)

[Bảng 3.4. Bảng Delivery 82](#_Toc102342543)

[Bảng 3.5. Bảng User 83](#_Toc102342544)

[Bảng 3.6. Bảng SaleOrder 83](#_Toc102342545)

[Bảng 3.7. Bảng OrderItem 84](#_Toc102342546)

[Bảng 3.8. Bảng ProductRating 84](#_Toc102342547)

[Bảng 3.9. Bảng Banner 85](#_Toc102342548)

# DANH MỤC HÌNH ẢNH

[Hình 2.1. Java logo 7](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035281)

[Hình 2.2. JavaScript logo 9](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035282)

[Hình 2.3. Kiến trúc của Spring Framework 13](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035283)

[Hình 2.4. React logo 16](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035284)

[Hình 2.5. PostgreSQL logo 19](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035285)

[Hình 3.1. Biểu đồ use case tổng quát 24](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035286)

[Hình 3.2. Mô hình thực thể liên kết 25](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035287)

[Hình 3.3. Biểu đồ use case đăng nhập 26](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035288)

[Hình 3.4. Biểu đồ trình tự đăng nhập 28](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035289)

[Hình 3.5. Biểu đồ hoạt động đăng nhập 29](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035290)

[Hình 3.6. Biểu đồ use case đăng ký 30](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035291)

[Hình 3.7. Biểu đồ trình tự đăng ký 31](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035292)

[Hình 3.8. Biểu đồ hoạt động đăng ký 32](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035293)

[Hình 3.9. Biểu đồ use case tìm kiếm 33](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035294)

[Hình 3.10. Biểu đồ trình tự tìm kiếm 34](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035295)

[Hình 3.11. Biểu đồ hoạt động tìm kiếm 35](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035296)

[Hình 3.12. Biểu đồ use case xem chi tiết sách 35](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035297)

[Hình 3.13. Biểu đồ trình tự xem chi tiết sách 37](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035298)

[Hình 3.14. Biểu đồ hoạt động xem chi tiết sách 38](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035299)

[Hình 3.15. Biểu đồ use case sửa thông tin cá nhân 38](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035300)

[Hình 3.16. Biểu đồ trình tự sửa thông tin cá nhân 40](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035301)

[Hình 3.17. Biểu đồ hoạt động sửa thông tin cá nhân 41](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035302)

[Hình 3.18. Biểu đồ use case quên mật khẩu 41](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035303)

[Hình 3.19. Biểu đồ trình tự quên mật khẩu 43](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035304)

[Hình 3.20. Biểu đồ hoạt động quên mật khẩu 44](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035305)

[Hình 3.21. Biểu đồ use case đặt hàng 44](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035306)

[Hình 3.22. Biều đồ trình tự đặt hàng 47](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035307)

[Hình 3.23. Biểu đồ hoạt động đặt hàng 48](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035308)

[Hình 3.24. Biểu đồ use case quản lý tài khoản 48](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035309)

[Hình 3.25. Biểu đồ trình tự quản lý tài khoản 53](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035310)

[Hình 3.26. Biểu đồ hoạt động quản lý tài khoản 54](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035311)

[Hình 3.27. Biểu đồ use case quản lý danh mục 54](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035312)

[Hình 3.28. Biều đồ trình tự quản lý danh mục 60](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035313)

[Hình 3.29. Biều đồ hoạt động quản lý danh mục 60](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035314)

[Hình 3.30. Biểu đồ use case quản lý sách 61](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035315)

[Hình 3.31. Biểu đồ trình tự quản lý sách 66](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035316)

[Hình 3.32. Biểu đồ hoạt động quản lý sách 67](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035317)

[Hình 3.33. Biểu đồ use case quản lý ảnh 67](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035318)

[Hình 3.34. Biểu đồ trình tự quản lý ảnh 73](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035319)

[Hình 3.35. Biểu đồ hoạt động quản lý ảnh 73](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035320)

[Hình 3.36. Biểu đồ use case danh sách đơn hàng 74](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035321)

[Hình 3.37. Biểu đồ trình tự danh sách đơn hàng 77](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035322)

[Hình 3.38. Biều đồ hoạt động danh sách đơn hàng 78](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035323)

[Hình 3.39. Biểu đồ triển khai hệ thống 79](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035324)

[Hình 3.40. Biều đồ mô tả cấu trúc cơ sở dữ liệu 80](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035325)

[Hình 3.41. Giao diện trang chủ 86](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035326)

[Hình 3.42. Giao diện trang đăng nhập 86](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035327)

[Hình 3.43. Giao diện trang đăng ký 87](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035328)

[Hình 3.44. Giao diện chức năng quên mật khẩu 87](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035329)

[Hình 3.45. Giao diện chức năng tìm kiếm sản phẩm 88](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035330)

[Hình 3.46. Giao diện trang chi tiết sản phẩm 88](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035331)

[Hình 3.47. Giao diện trang giỏ hàng 89](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035332)

[Hình 3.48. Giao diện trang thông tin tài khoản 89](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035333)

[Hình 3.49. Giao diện trang thông tin đơn hàng 90](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035334)

[Hình 3.50. Giao diện trang chủ admin 90](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035335)

[Hình 3.51. Giao diện trang quản lý tài khoản 91](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035336)

[Hình 3.52. Giao diện trang quản lý danh mục 91](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035337)

[Hình 3.53. Giao diện trang quản lý sản phẩm 92](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035338)

[Hình 3.54. Giao diện trang quản lý hình ảnh sản phẩm 92](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035339)

[Hình 3.55. Giao diện trang quản lý đơn hàng 93](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035340)

[Hình 3.56. Hộp thoại chi tiết đơn hàng 93](file:///D:\Study\DATN\DATN_BuiTatTrung_2018602729.docx#_Toc103035341)

# TỔNG QUAN VỀ ĐỀ TÀI

## Tên đề tài

***Xây dựng Website bán trang sức cửa hàng Thu Hà***

## Lý do chọn đề tài

Mô hình kinh doanh trên toàn cầu vẫn đang tiếp tục thay đổi dáng kể với sự ra đời của thương mại điện tử, có thể kể điến các sàn thương mại điện tử lớn hiện nay: Amazon, Alibaba, Shoppe,...Vậy thương mại điện tử là gì?

Theo tổ chức kinh tế thế giới định nghĩa: “Thương mại điện tử bao gồm việc sản xuất, quảng cáo, bán hàng và phân phối sản phẩm được mua bán và thanh toán trên mạng Internet, nhưng được giao nhận một cách hữu hình, cả các sản phẩm giao nhận cũng như những thông tin số hoá thông qua mạng Internet”.

Theo Ủy ban Thương mại điện tử của [Tổ chức Hợp tác kinh tế châu Á - Thái Bình Dương](https://vi.wikipedia.org/wiki/Di%E1%BB%85n_%C4%91%C3%A0n_H%E1%BB%A3p_t%C3%A1c_Kinh_t%E1%BA%BF_ch%C3%A2u_%C3%81_-_Th%C3%A1i_B%C3%ACnh_D%C6%B0%C6%A1ng) định nghĩa: "Thương mại điện tử liên quan đến các giao dịch thương mại trao đổi hàng hóa và dịch vụ giữa các nhóm (cá nhân) mang tính điện tử chủ yếu thông qua các hệ thống có nền tảng dựa trên Internet."

Thương mại điện tử giúp nười tiêu dùng thu thập nhanh chóng và dễ dàng các thông tin đa dạng về sản phẩm. Hơn nữa, có thể dưa ra các đánh giá về sản phẩm để lựa chọn các dịch vụ tót nhất. Đây là một phương thức kinh doanh sử dụng internet mang lại nhiều lợi ích cho các cá nhân, doanh nghiệp: giao dịch nhanh chóng, giảm chi phí mặt bằng và quản lý.

Ngày nay, các hoạt động thương mại điện tử hầu hết được thực hiện thông qua website bán hàng. Website bán hàng trở thành nơi quảng cáo, thực hiện giao dịch mua bán, thậm chí nguồn thu từ việc bán hàng qua website đã trở thành doanh số chính của nhiều công ty, doanh nghiệp. Website đã trở thành một phần không thể thiếu trong việc kinh doanh theo hình thức thương mại điện tử.

Trong nền kinh tế thị trường hiện nay, việc cạnh tranh giữa các cửa hàng công ty, doanh nghiệp ngày càng quyết liệt. Mục tiêu lớn nhất của cạnh tranh đó là thu hút nhiều khách hàng quan tâm đến sản phẩm của mình. Để thu hút được nhiều khách hàng, bên cạnh chất lượng sản phẩm và giá cả, các chiến lược kinh doanh, chiến lược marketing cũng rất quan trọng. Website trở thành công cụ, một phương tiện không thể thiếu để thực hiện marketing, quảng bá sản phẩm, thương hiệu của công ty, doanh nghiệp. Website trở thành một đường để doanh nghiệp giới thiệu sản phẩm ra thị trường trong nước, thậm chí ra ngoài quốc tế. Đây là nơi khách hàng tìm thấy thông tin giới thiệu về doanh nghiệp, các thông tin chi tiết về các sản phẩm. Nó cũng là một kênh giao tiếp, tư vấn trực tuyến với khách hàng, giải quyết kịp thời thắc mắc và vấn đề của khách hàng, giúp khách hàng hài lòng.

Thiết kế website là nhu cầu cần thiết đối với các công ty, doanh nghiệp. Nó là hình ảnh, bộ mặt của công ty, doanh nghiệp. Website càng tốt càng mang lại trải nghiệm tốt cho khách hàng, càng hướng khách hàng tới sản phẩm của mình. Website có ảnh hưởng rất lớn tới việc kinh doanh của công ty, doanh nghiệp như doanh thu, chi phí quản lý, sự hài lòng và đánh giá của khách.

Cùng với sự phát triển mạnh của nền kinh tế, những năm gần đây cũng đánh dấu bước thay đổi đời sống của người dân khi nhu cầu sử dụng trang sức tăng lên. Kinh doanh trang sức đang dần chiếm một tỷ trọng nhất định trong nền kinh tế. Chính vị vậy các hãng trang sức lớn trên thế giới đang đẩy mạng việc mua bán trên các nền tảng thương mại điện tử cũng như tạo dựng một website riêng hoặc phân phối bán lẻ cho các cửa hàng.

Đối với website hàng nói chung và website bán trang sức nói riêng, bên cạnh giao diện đẹp thì website hoạt động như thế nào là rất quan trọng. Khách hàng khi ghé thăm website của cửa hàng thì khách hàng sẽ giao tiếp và thao tác với giao diện của website đó. Giao diện website phải đẹp, phải gây ấn tượng với khách hàng và cũng sẽ tạo nên điểm đặc biệt của thương hiệu của cửa hàng. Khách hàng nhìn vào website có thể dễ dàng sử dụng các chức năng có trên đó. Về hoạt động của website thì website phải chạy nhanh, mượt mà để khách hàng thoải mái, không bị khó chịu khi sử dụng. Website giúp khách hàng có thể xem được thông tin mình cần, có thể tìm tới sản phẩm mình muốn. Nó chính là phương pháp gây ấn tượng và thu hút khách hàng của các cửa hàng.

Nhận thức được sự quan trọng và cũng như để có thể áp dụng được những kiến thức đã được học và tìm hiểu, em xin được áp dụng những kiến thức đã được học và tìm hiểu đó để xây dựng một website cụ thể là: ***“Xây dựng Website bán trang sức cửa hàng Thu Hà”.***

## Mục tiêu của đề tài

Đề tài: ***Xây dựng Website bán trang sức cửa hàng Thu Hà*** đáp ứng được những mục tiêu:

* Hiểu biết những kiến thức cần thiết về xây dựng giao diện trang web, phân tích thiết kế hệ thống, cơ sở dữ liệu.
* Nắm được các kiến thức cơ bản về ExpressJS - Javascript, VueJS.
* Dễ dàng tiếp cận với mọi khách hàng ở bất kì nơi đâu, chỉ cần kết nới mạng internet có thể tìm thấy website và thông tin sản phẩm của công ty nhanh chóng. thương hiệu của đại lý.
* Tiết kiệm được thời gian,chi phí đi lại cho khách hàng, khách hàng không cần phải đến tận cửa hàng mới mua được hàng mà họ có thể thông qua website xem và mua hàng.
* Thông tin sản phẩm được cập nhật nhanh chính xác đến khách hàng**.**
* Xây dựng và quảng bá thương hiệu cho nhãn hàng.

## Đối tượng và phạm vi

Bất cứ ai có nhu cầu mua trang sức trực tuyến trên nền tảng internet.

## Kết quả dự kiến đạt được

Website bao gồm những mục chính và chức năng chính:

* QR code trong việc mua hàng
* Share giỏ hàng, cho phép người được share chỉnh sửa giỏ hàng.
* Chức năng tiếp thị liên kết ăn % hoa hồng.
* Tích hợp thanh toán ngân hàng lên đồ án.
* Đăng ký, đăng nhập tài khoản cá nhân.
* Thay đổi mật khẩu, cập nhật thông tin tài khoản.
* Quên mật khẩu của tài khoản.
* Phân quyền giữa người quản trị và khách hàng.
* Chức năng quản trị: quản lý tài khoản, danh mục sản phẩm, sản phẩm, đơn hàng, hình ảnh sản phẩm, bài viết, chiến dịch.
* Tìm kiếm sản phẩm, xem chi tiết sản phẩm, tạo bài viết
* Đặt hàng, thanh toán trực tuyến.

# CƠ SỞ LÝ THUYẾT VÀ CÔNG CỤ SỬ DỤNG

## Cơ sở lý thuyết

### Ngôn ngữ UML

UML là viết tắt của “Unified Modeling Language” - là ngôn ngữ mô hình hóa thống nhất, dùng để đặc tả, hình dung, xây dựng và tư liệu hóa phần mềm hướng đối tượng.

Mục tiêu của UML :

UML phân phối cho người dùng một ngôn ngữ từ mô hình hóa trực quan chuẩn bị sẵn sàng để dùng và có ý nghĩa:

* Cho phép phát triển và trao đổi những mô hình mang nhiều ý nghĩa.
* Cung cấp khả năng mở rộng và chuyên môn hóa để mở rộng những khái niệm cốt lõi.
* Độc lập với ngôn ngữ lập trình chuyên biệt và các tiến trình phát triển.
* Cung cấp nền tảng về sự hiểu biết ngôn ngữ mô hình hóa.
* Khuyến khích và hỗ trợ sự phát triển của các công cụ hướng đối tượng.
* Tích hợp một cách tốt nhất với thực tiễn...

UML có thể được sử dụng làm công cụ giao tiếp giữa người dùng, nhà phân tích, nhà thiết kế và nhà phát triển phần mềm.

***Có 9 loại biểu đồ:***

* Use Case Diagram (Biểu đồ ca sử dụng): Mô tả sự tương tác giữa các tác nhân và hệ thống thông qua các ca sử dụng.
* Class Diagram (Biểu đồ lớp): Biểu đồ lớp chỉ ra sự tồn tại giữa các lớp và mối quan hệ giữa chúng trong thiết kế logic trong hệ thống. Khung nhìn tĩnh của hệ thống chủ yếu hỗ trợ các chức năng của hệ thống.
* Object Diagram (Biểu đồ đối tượng): Biểu đồ đối tượng đại diện cho một thể hiện của biểu đồ lớp, cho nên các khái niệm cơ bản là giống biểu đồ lớp. Biểu đồ đối tượng được sử dụng để hiện thị một tập hợp các đối tượng và các mối quan hệ của chúng như một thể hiện.
* Sequence Diagram (Biểu đồ trình tự): Là biểu đồ để xác định và chỉ rõ vai trò của đối tượng tham gia vào luồng sự kiện của use case.
* Collaboration Diagram (Biểu đồ cộng tác): Tương tự như biểu đồ trình tự nhưng nhấn mạnh vào sự tương tác của các đối tượng trên cơ sở cộng tác với nhau bằng cách trao đổi các thông điệp để thực hiện các yêu cầu theo ngữ cảnh công việc.
* State Diagram (Biểu đồ trạng thái): Thể hiện chu kỳ hoạt động của các đối tượng, của các hệ thống con và của cả hệ thống.
* Activity Diagram (Biểu đồ hành động): Được sử dụng để mô tả các hoạt động và các hành động được thực hiện trong một use case.
* Component Diagram (Biểu đồ thành phần): Chỉ ra cấu trúc vật lý của các thành phần trong hệ thống.
* Deployment Diagram (Biểu đồ triển khai): Chỉ ra cách bố trí vật lý các thành phần theo kiến trúc được thiết kế của hệ thống.

Relationship (Quan hệ):

* Realization (Hiện thực hoá):Là quan hệ ngữ nghĩa giữa giao diện và lớp.
* Dependency (Phụ thuộc):Là quan hệ ngữ nghĩa giữa hai phần tử, trong đó sự thay đổi của một tử sẽ tác động đến ngữ nghĩa của phần tử phụ thuộc.
* Generalization (Kế thừa): Là quan hệ mô tả sự khái quát hoá mà trong đó một số đối tượng cụ thể (của lớp con) sẽ được kế thừa các thuộc tính, các phương thức của các đối tượng tổng quát (lớp cơ sở).
* Association (Kết hợp): Là quan hệ cấu trúc xác định mối liên kết giữa các lớp đối tượng.

### File:Unofficial JavaScript logo 2.svg - Wikimedia CommonsNgôn ngữ JavaScript

Hình .. JavaScript logo

JavaScript là một ngôn ngữ lập trình đa nền tảng (cross-platform), ngôn ngữ lập trình kịch bản, hướng đối tượng. JavaScript là một ngôn ngữ nhỏ và nhẹ (small and lightweight). JavaScript chứa các thư viện tiêu chuẩn cho các đối tượng (object), ví dụ như: Array, Date, Math, và các yếu tố cốt lõi của ngôn ngữ lập trình như: toán tử (operators), cấu trúc điều khiển (control structures), và câu lệnh. JavaScript có thể được mở rộng cho nhiều mục đích bằng việc bổ sung thêm các object; ví dụ:

* Client-side JavaScript - JavaScript phía máy khách, JavaScript được mở rộng bằng cách cung cấp các object để quản lý trình duyệt và Document Object Model (DOM) của nó. Ví dụ, phần mở rộng phía máy khách cho phép một ứng dụng tác động tới các yếu tố trên một trang HTML và phản hồi giống các tác động của người dùng như click chuột, nhập form, và chuyển trang.
* Server-side JavaScript - JavaScript phía máy chủ, JavaScript được mở rộng bằng cách cung cấp thêm các đối tượng cần thiết để để chạy JavaScript trên máy chủ. Ví dụ, phần mở rộng phía server này cho phép ứng dụng kết nối với cơ sở dữ liệu (database), cung cấp thông tin một cách liên tục từ một yêu cầu tới phần khác của ứng dụng, hoặc thực hiện thao tác với các tập tin trên máy chủ.

JavaScript được phát triển bởi Brendan Eich, một nhân viên của Netscape, vào tháng 9 năm 1995. Được biết đến lần đầu tiên là Mocha, rồi sau đó được đổi tên thành LiveScript và cuối cùng là JavaScript nổi tiếng như hiện giờ. Phiên bản đầu tiên của ngôn ngữ này bị giới hạn độc quyền bởi Netscape và chỉ có các tính năng hạn chế, nhưng nó tiếp tục phát triển theo thời gian.

Năm 1996, JavaScript được chính thức đặt tên là ECMAScript. ECMAScript 2 phát hành năm 1998 và ECMAScript 3 tiếp tục ra mắt vào năm 1999. Nó liên tục phát triển thành JavaScript ngày nay, giờ đã hoạt động trên khắp mọi trình duyệt và trên khắp các thiết bị từ di động đến máy tính bàn.

JavaScript liên tục phát triển kể từ đó, có lúc đạt đến 92% website đang sử dụng JavaScript vào năm 2016. Chỉ trong 20 năm, nó từ một ngôn ngữ lập trình riêng trở thành công cụ quan trọng nhất trên bộ công cụ của các chuyên viên lập trình web.

JavaScript cùng với HTML, CSS đã trở thành ngôn ngữ không thể thiếu trong lập trình web hiện nay:

* Với việc phát triển mạnh mẽ của công nghệ, JavaScript đã dần trở thành một ngôn ngữ đa nền tảng với nhiều ứng dụng khác nhau.
* Các Framework, Library cho lập trình front-end mạnh mẽ hỗ trợ xây dựng các trang Single-page application (SPA) như: ReactJS, Angular, VueJS.
* Hỗ trợ lập trình mobile bằng thư viện React Native.
* Hỗ trợ lập trình Window bằng Electron.
* Không chỉ phát triển ở khía cạnh lập trình phía client-side, JavaScript cũng hỗ trợ viết server-side rất mạnh mẽ với NodeJS.

Với hệ sinh thái mã nguồn mở khổng lồ cùng sự phát triển như hiện nay, JavaScript được dự đoán sẽ phát triển mạnh mẽ hơn nữa trong những năm tới.

### Framework ExpressJS của ngôn ngữ JavaScript

#### ExpressJS Framework là gì?



* Express js là một Framework nhỏ, nhưng linh hoạt được xây dựng trên nền tảng của Nodejs. Nó cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile
* Về các package hỗ trợ: Expressjs có vô số các package hỗ trợ nên các bạn không phải lo lắng khi làm việc với Framework này.
* Về performance: Express cung cấp thêm về các tính năng (feature) để dev lập trình tốt hơn. Chứ không làm giảm tốc độ của NodeJS.
* Và hơn hết, các Framework nổi tiếng của NodeJS hiện nay đều sử dụng ExpressJS như một core function, chẳng hạn: SailsJS, MEAN,....

#### Lịch sử phát triển của Framework ExpressJS

Express.js được thành lập bởi TJ Holowaychuk. Bản phát hành đầu tiên, theo kho lưu trữ GitHub của Express.js, là vào ngày 22 tháng 5 năm 2010 với phiên bản 0.12

Vào tháng 6 năm 2014, StrongLoop đã mua lại quyền quản lý dự án. StrongLoop đã được IBM mua lại vào tháng 9 năm 2015. Tháng 1 năm 2016, IBM đã thông báo rằng họ sẽ đặt Express.js dưới sự quản lý của Node.js Foundation..

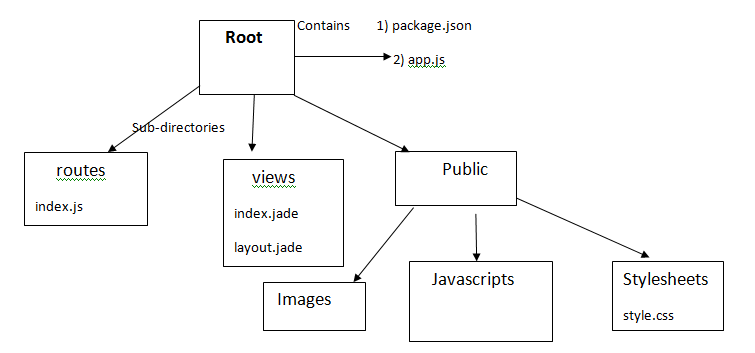
#### Tính năng của Framework ExpressJS

* Cho phép thiết lập các lớp trung gian để trả về các HTTP request.
* Định nghĩ bảng routing có thể được sử dụng với các hành động khác nhau dựa trên phương thức HTTP và URL.
* Cho phép trả về các trang HTML dựa vào các tham số truyền vào đến template.

#### Ưu điểm khi sử dụng Framework ExpressJS

* Express.js được sử dụng trong thiết kế và xây dựng các ứng dụng web một cách đơn giản và nhanh chóng.
* Vì Express js chỉ yêu cầu ngôn ngữ lập trình Javascript nên việc xây dựng các ứng dụng web và API trở nên đơn giản hơn với các lập trình viên và nhà phát triển.Expressjs cũng là một khuôn khổ của Node.js do đó hầu hết các mã code đã được viết sẵn cho các lập trình viên có thể làm việc.
* Nhờ có Expressjs mà các nhà lập trình có thể dễ dàng tạo các ứng dụng 1 web, nhiều web hoặc kết hợp. Do có dung lượng khá nhẹ, Expressjs giúp cho việc tổ chức các ứng dụng web thành một kiến trúc MVC có tổ chức hơn.Để có thể sử dụng được mã nguồn này, chúng ta cần phải biết về Javascript và HTML.

#### Cấu trúc của Framework ExpressJS



Hình .. Kiến trúc của Spring Framework

Cấu trúc của express js vô cùng đơn giản.

* Root:
* app.js chứa các thông tin về cấu hình, khai báo, các định nghĩa,... để ứng dụng của chúng ta chạy ok.
* package.json chứa các package cho ứng dụng chạy. Nếu bạn nào làm với PHP hoặc RoR rồi thì file này có chức năng tương tự như composer.json hoặc Gemfile
* Folder routes: chứa các route có trong ứng dụng
* Folder view: chứa view/template cho ứng dụng
* Folder public chứa các file css, js, images,...cho ứng dụng

### VueJS

#### VueJS là gì?

Vue là một framework Javascript tiên tiến trong xây dựng giao diện người dùng, không giống như các framework khác, Vue được xây dựng từ những dòng code cơ bản nhất nhằm tối ưu tốc độ. Thư viện của Vue chỉ tập trung vào lớp hiển thị, rất đơn giản để tiếp cận và dễ dàng tích hợp vào các hệ thống khác. Vue cũng có khả năng cung cấp các ứng dụng web đơn trang Single Page Application (SPA) cho phép kết hợp với nhiều các công cụ hiện đại, như Laravel chẳng hạn.



Hình .. Vue.jslogo

Vue.js được sử dụng để xây dựng giao diện người dùng giống như React (sử dụng bởi Facebook), Angular (được hậu thuẫn bởi Google), Ember… Tuy nhiên, Vue.js có tốc độ tạo trang (render) rất nhanh và chiếm khá ít bộ nhớ. Chúng ta có thể xem bảng benchmark các framework Javascript nổi tiếng nhất hiện nay, Vue có một thứ hạng không tồi chút nào.

Vue.js mới chỉ ra mắt năm 2015, nhưng Vue.js đã sớm khẳng định mình và sớm trở thành người thay thế Angular và React, đây cũng chính là lý do Laravel giới thiệu Vue.js trong thiết lập mặc định..

#### Lịch sử phát triển của VueJS

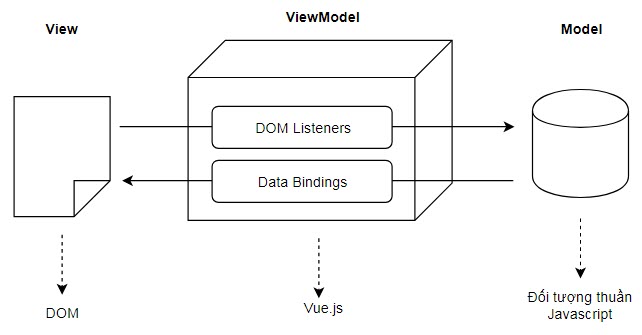
Vue được thai nghén và phát triển bởi Evan You khi tác giả đang làm kỹ sư cho ông lớn Google. Khi đang sử dụng Angular cho một số dự án khác thì Evan có ý tưởng trích một phần mà mình thích của Angular rồi từ đó xây dựng thêm một cái gì đó nhẹ nhàng hơn Angular mà không cần phải chú ý đến những khái niệm bổ sung rườm rà. Đây được xem là tiền đề giúp cho Vuejs đã, đang có mặt và phát triển mạnh mẽ tại thời điểm hiện tại.

Vue đã được phát hành đầu năm 2014 và dự án này đã được tăng tải lên HackerNew, Echo Js ngay trong ngày đầu tiên ra mắt.

Vue 2.0 được release vào đầu năm 2016 để ganh đua với ReactJS. Thực tế Vuejs được chứng minh là nhanh hơn và gọn hơn so với ReactJs và Angular 2.0. Ngoài ra, thời gian tiếp cận và làm chủ của Vue tương đối ngắn vì vậy không quá khó hiểu khi đọc xong tài liệu này bạn sẽ thích và sẽ sử dụng Vue làm framework js mặc định của mình..

#### Mô hình MVVM

MVVM là viết tắt của Model-View-ViewModel là một mô hình được áp dụng trong framework Vue.js.



Hình: Mô hình MVVM

* **Binding Data:** Binding Data trong MVVM là điều không bắt buộc, một số implement chỉ đơn giản làm ViewModel như một lớp trung gian giữa Model-View, lớp này giữ nhiệm vụ format data hoặc mapping trạng thái của View.
* **View**: Tương tự như trong mô hình MVC, View là phần giao diện của ứng dụng để hiển thị dữ liệu và nhận tương tác của người dùng. Một điểm khác biệt so với các ứng dụng truyền thống là View trong mô hình này tích cực hơn. Nó có khả năng thực hiện các hành vi và phản hồi lại người dùng thông qua tính năng binding, command.
* **Model**: Cũng tương tự như trong mô hình MVC. Model là các đối tượng giúp truy xuất và thao tác trên dữ liệu thực sự.
* **ViewModel**: Lớp trung gian giữa View và Model. ViewModel có thể được xem là thành phần thay thế cho Controller trong mô hình MVC. Nó chứa các mã lệnh cần thiết để thực hiện data binding, command.

Trong mô hình này, dữ liệu mỗi khi được "gán" vào View hoặc Model sẽ đều được Vue.js tự động gắn cho phần còn lại. Tức là khi dữ liệu thay đổi ở Model nó sẽ tự động được "cập nhật" sang View và khi người dùng thay đổi dữ liệu trên View (ví dụ nhập liệu vào ô địa chỉ email chẳng hạn) thì dữ liệu cũng được tự động cập nhật ngược lại Model. Trong cộng đồng Vue.js thường gọi mô hình này với một thuật ngữ khác là two-way data binding, tạm gọi là gán dữ liệu hai chiều.

* Tạo ra những Website chạy thật mượt mà, nhanh, có khả năng mở rộng cao và đơn giản thực hiện.
* Cho phép phá vỡ giao diện người dùng, biến nó từ phức tạp thành đơn giản hơn.

#### Ưu điểm của VueJS

* Trang web nhẹ, tốc độ xử lý cực nhanh bởi được Render, xử lý bằng Javascript.
* Đơn giản, dễ học và dễ áp dụng trong các dự án, đặc biệt là với newbie
* Có khả năng xử lý các render thuộc server thành file Js tĩnh
* Kho thư viện lớn, hỗ trợ xây dựng giao diện một cách nhanh chóng, hoàn hảo
* Dung lượng tải thấp, giúp tốc độ tải trang nhanh hơn
* Sàng lọc, tích hợp các tính năng ưu việt của nhiều framework đối thủ, giúp tối ưu hóa hiệu suất làm việc.

#### Tính năng đặc trưng của VueJS

* DOM ảo: VueJS sử dụng DOM ảo, cũng được sử dụng bởi các framework khác như React, Ember, v.v. Các thay đổi không được thực hiện đối với DOM, thay vào đó, một bản sao của DOM được tạo dưới dạng cấu trúc dữ liệu JavaScript . Khi nào có bất kỳ thay đổi nào, chúng sẽ được thực hiện đối với cấu trúc dữ liệu JavaScript. Cấu trúc sau được so sánh với cấu trúc dữ liệu ban đầu. Những thay đổi cuối cùng sau đó được cập nhật vào DOM thực, người dùng sẽ thấy thay đổi này. Điều này tốt về mặt tối ưu hóa, ít tốn kém hơn và các thay đổi có thể được thực hiện với tốc độ nhanh hơn.
* Ràng buộc dữ liệu: Tính năng ràng buộc dữ liệu giúp thao tác hoặc gán giá trị cho các thuộc tính HTML, thay đổi kiểu, gán lớp với sự trợ giúp của chỉ thị ràng buộc có tên v-bind có sẵn trong VueJS.
* Component: Component là một trong những tính năng quan trọng của VueJS giúp tạo các phần tử tùy chỉnh và có thể tái sử dụng trong HTML.
* Xử lý sự kiện: v-on là thuộc tính được thêm vào các phần tử DOM để lắng nghe các sự kiện trong VueJS.
* Hoạt ảnh/Chuyển tiếp: VueJS cung cấp nhiều cách khác nhau để áp dụng chuyển đổi sang các phần tử HTML khi chúng được thêm, cập nhật hoặc xóa khỏi DOM. VueJS có một thành phần chuyển tiếp tích hợp cần được bao quanh phần tử để tạo hiệu ứng chuyển tiếp. Người dùng có thể dễ dàng thêm các thư viện hoạt hình của bên thứ ba và cũng có thể thêm nhiều tương tác hơn vào giao diện.
* Thuộc tính được tính toán: Đây là một trong những tính năng quan trọng của VueJS. Nó giúp lắng nghe những thay đổi được thực hiện đối với các thành phần giao diện người dùng và thực hiện các tính toán cần thiết. Không cần mã hóa bổ sung cho việc này.
* Template: VueJS cung cấp các mẫu dựa trên HTML liên kết DOM với dữ liệu đối tượng Vue. VueJS biên dịch các mẫu thành các hàm Kết xuất DOM ảo. Chúng ta có thể sử dụng mẫu của các chức năng kết xuất và ngược lại.
* Chỉ thị: VueJS có các chỉ thị tích hợp như v-if, v-else, v-show, v-on, v-bind và v-model, được sử dụng để thực hiện các hành động khác nhau trên giao diện người dùng..

### Cơ sở dữ liệu MySQL

#### MySQL là gì?



MySQL là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là hệ quản trị cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet. Người dùng có thể tải về MySQL miễn phí từ trang chủ. MySQL có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OS X, Unix, FreeBSD, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS,..

MySQL là một trong những ví dụ rất cơ bản về Hệ Quản trị Cơ sở dữ liệu quan hệ sử dụng Ngôn ngữ truy vấn có cấu trúc (SQL).

#### Đặc điểm của MySQL

* Mã nguồn mở: MySQL là một phần mềm mã nguồn mở, có nghĩa là bất kỳ ai cũng có thể tải về và sử dụng nó miễn phí.
* Hiệu suất tốt: MySQL cung cấp hiệu suất tốt và tốc độ truy vấn nhanh cho các cơ sở dữ liệu lớn.
* Dễ sử dụng: MySQL cung cấp giao diện đơn giản và trực quan cho việc quản lý và truy vấn dữ liệu.
* Tính linh hoạt: MySQL cung cấp một số tùy chọn và tính năng linh hoạt để có thể tùy chỉnh cho nhu cầu của người dùng.
* Tương thích nhiều hệ điều hành: MySQL có thể chạy trên nhiều hệ điều hành, bao gồm Windows, Linux và macOS.
* Tối ưu hóa tốt: MySQL cung cấp các công cụ tối ưu hóa để giảm thời gian truy vấn và tăng hiệu suất hệ thống.
* Hỗ trợ nhiều ngôn ngữ lập trình: MySQL hỗ trợ nhiều ngôn ngữ lập trình, bao gồm PHP, Java, C++ và Python.

Do có tốc độ cao và bảo mật, MySQL được xem là sự lựa chọn đầu tiên trong phát triển các ứng dụng có truy cập cơ sở dữ liệu, phù hợp làm nơi lưu trữ dữ liệu cho các trang web được viết bằng PHP hoặc NodeJS.

#### Các yếu tố giúp MySQL nổi bật

## Công cụ sử dụng

* IntelliJ IDEA: phục vụ mục đích code Java.
* Visual Studio Code: phục vụ mục đích code ReactJS.
* MySQL: cơ sở dữ liệu dùng để lưu trữ dữ liệu.
* PgAdmin: hệ quản trị cơ sở dữ liệu.
* Postman: sử dụng để thử nghiệm API.
* Chrome DevTools: gỡ lỗi khi phát triển web.
* Case Studio: thiết kế mô hình thực thể liên kết.
* Rational Rose: thiết kế hệ thống.
* Cache: Redis
* Container: Docker
* Công nghệ khác: Socket

# PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## Phân tích hệ thống

### Vai trò của người dùng

* Hệ thống chia làm 2 nhóm người dùng chính:
  + Khách hàng
  + Người quản trị
* Mô tả nhóm người dùng.
* Khách hàng: truy cập trang web, xem thông tin sản, và có thể đăng kí tài khoản và đăng nhập trên trang web. Khách hàng khi mua hàng trên trang web thì thông tin của khách hàng đã được lưu lại trong cơ sở dữ liệu.
* Người quản trị: là người quản lý trang web và được truy cập vào các chức năng quản lý của hệ thống.

### Yêu cầu chức năng

* Khách hàng:
  + Cập nhật trang cá nhân
  + Xác thực: đăng kí , đăng nhập
  + Xem danh sách thể loại
  + Xem danh sách sản phẩm
  + Xem danh sách bài viết
  + Tìm kiếm sản phẩm
  + Đặt hàng
  + Chia sẻ giỏ hàng
  + Tạo chiến dịch
* Quản trị viên:
  + Quản lý người dùng
  + Quản lý thể loại
  + Quản lý bài viết
  + Quản lý quảng cáo
  + Quản lý sản phẩm
  + Quản lý đơn hàng
  + Quản lý chiến dịch

### Yêu cầu phi chức năng

* Giao diện trang web được thiết kế dễ sử dụng và có tính thẩm mỹ cao.
* Được viết trên nền tảng web.
* Hiệu năng: tốc độ, khả năng và độ tin cậy của hệ thống.
* Website bảo mật phân quyền để thực hiện các chức năng của hệ thống.

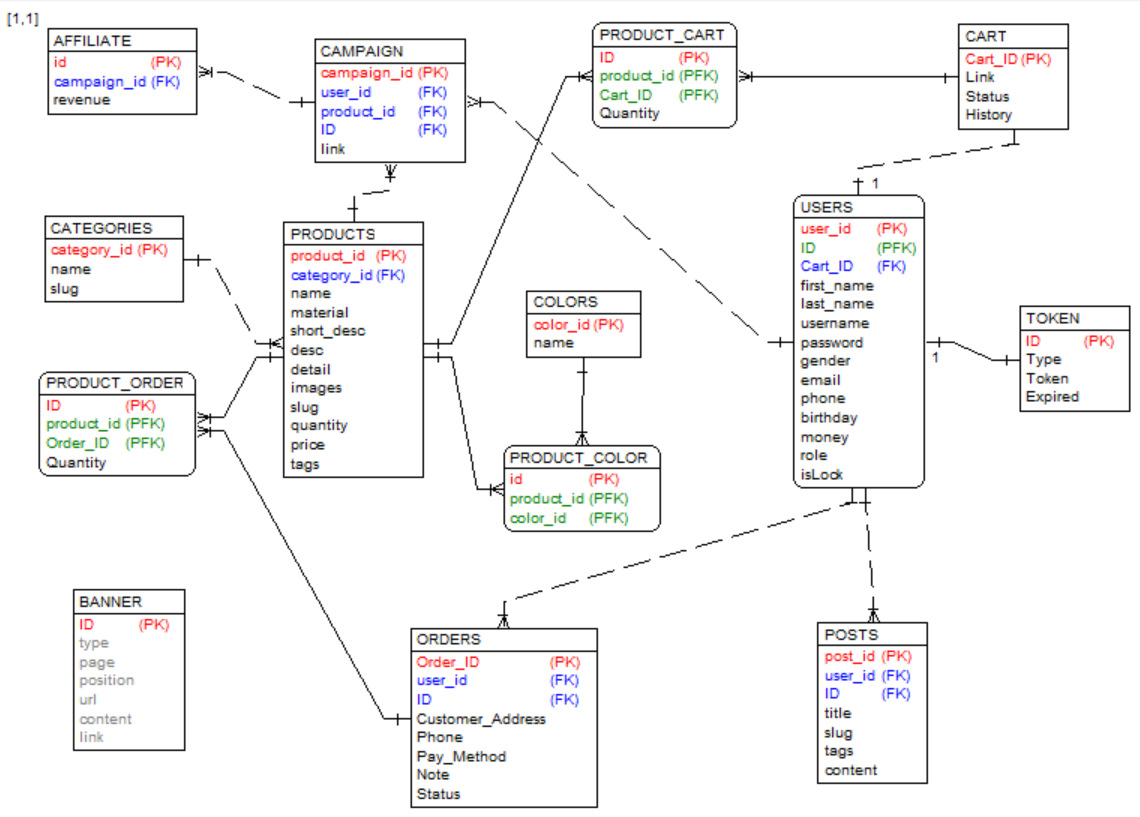
### Yêu cầu nghiệp vụ

## Biểu đồ use case tổng quát



*Hình :Biểu đồ Use case tổng quát*

## Mô hình thực thể liên kết



Hình .. Mô hình thực thể liên kết

## Đặc tả use case

### Use case đăng nhập

1. Biểu đồ use case:
2. Đặc tả use case:

Use case này cho phép người dùng đăng nhập để xác định quyền truy cập vào hệ thống.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi người dùng bấm vào biểu tượng người dùng sau đó bấm vào nút “Đăng nhập” trên góc phải màn hình.
2. Hệ thống yêu cầu người dùng nhập các thông tin bao gồm: tên đăng nhập và mật khẩu.
3. Người dùng nhập các thông tin: tên đăng nhập và mật khẩu và bấm nút “Đăng nhập”.
4. Hệ thống kiểm tra tên đăng nhập và mật khẩu đã nhập và chuyển hướng sang trang cá nhân .
5. Use case kết thúc.
   * Luồng rẽ nhánh:
6. Sai tên đăng nhập hoặc mật khẩu: tại bước 3 trong luồng cơ bản nếu người dùng nhập tên đăng nhập hoặc mật khẩu sai, hệ thống hiển thị thông báo lỗi. Người dùng có thể nhập lại hoặc bỏ qua thao tác, khi đó use case kết thúc.
7. Đăng ký: tại bước 3 trong luồng cơ bản, nếu người dùng bấm nút “Đăng kí” use case kết thúc.
8. Quên mật khẩu: tại bước 3 trong luồng cơ bản, nếu người dùng bấm nút “Quên mật khẩu” use case kết thúc.
9. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Không có.

* Hậu điều kiện:

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

* Điểm mở rộng:

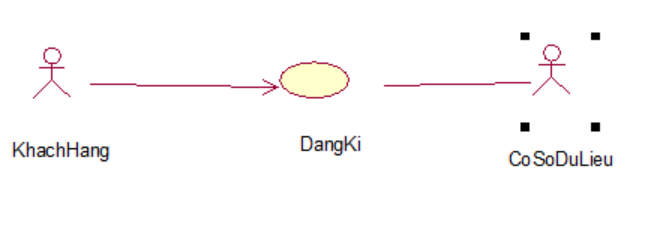
Không có.

1. Biểu đồ trình tự:



### Use case đăng ký

1. Biểu đồ use case:



1. Đặc tả use case:

Use case cho phép khách đăng ký tài khoản trên hệ thống.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi khách hàng kích vào nút Đăng ký trên màn hình. Hệ thống sẽ hiển thị màn hình đăng ký.
2. Khách hàng nhập thông tin trên form đăng ký gồm First Name, Last Name, Username, Password, Email, Phone, Address rồi kích nút đăng ký. Hệ thống thêm một bản ghi mới vào trong bảng USERS trong cơ sở dữ liệu.
3. Use case kết thúc.
   * Luồng rẽ nhánh:
4. Tại bước 2 trong luồng cơ bản, khi nhập thông tin không thoả mãn, hệ thống thông báo lỗi. Use case kết thúc.
5. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Không có.

* Hậu điều kiện:

Không có.

* Điểm mở rộng:

Không có.

1. Biểu đồ trình tự:



### Use case tìm kiếm

1. Biểu đồ use case:
2. Đặc tả use case:

Use case cho phép người dùng tìm kiếm sách theo tên.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi khách hàng nhập tên đồ dùng vào ô tìm kiếm và kích vào biểu tượng kính lúp trên thanh tìm kiếm. Hệ thống sẽ lấy các thông tin về các đồ dùng có tiêu đề chứa từ khoá và gồm name,images, price, tags từ bảng PRODUCTS và hiển thị lên màn hình.
2. Use case kết thúc.
   * Luồng rẽ nhánh:
3. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Không có.

* Hậu điều kiện:

Không có.

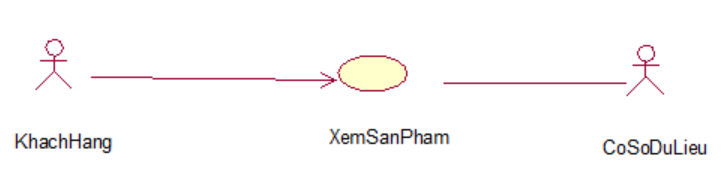
* Điểm mở rộng:

Không có.

1. Biểu đồ trình tự:

### Use case xem chi tiết sản phẩm

1. Biểu đồ use case:



1. Đặc tả use case:

Use case cho phép khách hàng xem thông tin chi tiết về sản phẩm trang sức.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi khách hàng kích vào tên sản phẩm trong danh sách sản phẩm. Hệ thống sẽ lấy các thông tin về sản phẩm bao gồm name,short\_desc, desc, metarial, images, slug, quantity, price, expiry, tags từ bảng PRODUCTS và hiển thị lên màn hình.
2. Use case kết thúc.
   * Luồng rẽ nhánh:
3. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Không có.

* Hậu điều kiện:

Không có.

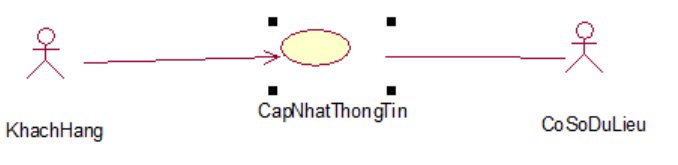
* Điểm mở rộng:

Không có.

1. `Biểu đồ trình tự:

### Use case cập nhật thông tin cá nhân

1. Biểu đồ use case:



1. Đặc tả use case:

Use case cho phép khách hàng sửa thông tin cá nhân trong tài khoản.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi khách hàng kích vào avatar trên màn hình. Hệ thống sẽ hiển thị màn hình thông tin cá nhân.
2. Khách hàng nhập thông tin cần chỉnh sửa rồi kích nút cập nhật. Hệ thống cập nhật lại thông tin trên bảng USER trong cơ sở dữ liệu.
3. Use case kết thúc.
   * Luồng rẽ nhánh:
4. Tại bước 2 ở luồng cơ bản, nếu khách hàng nhập sai định dạng, hệ thống thông báo lỗi. Use case kết thúc.
5. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Khách hàng đăng nhập thành công.

* Hậu điều kiện:

Không có.

* Điểm mở rộng:

Không có.

1. Biểu đồ trình tự:

### Use case quên mật khẩu

1. Biểu đồ use case:
2. Đặc tả use case:

Use case này cho phép khách hàng yêu cầu hệ thống cấp lại mật khẩu mới khi quên mật khẩu.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi khách hàng kích vào nút quên mật khẩu trên màn hình. Hệ thống sẽ hiển thị form quên mật khẩu.
2. Khách hàng nhập tên email đăng kí rồi kích nút Khôi phục mật khẩu. Hệ thống kiểm tra tài khoản người dùng trong cơ sở dữ liệu và gửi mã bí mật để khôi phục qua email.
3. Khách hàng nhập mã bí mật và nhập mật khẩu mới. Use case kết thúc.
   * Luồng rẽ nhánh:
4. Tại bước 2 ở luồng cơ bản, nếu tên email của khách hàng không tồn tại trong cơ sở dữ liệu, hệ thống báo lỗi. Use case kết thúc.
5. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Không có.

* Hậu điều kiện:

Không có.

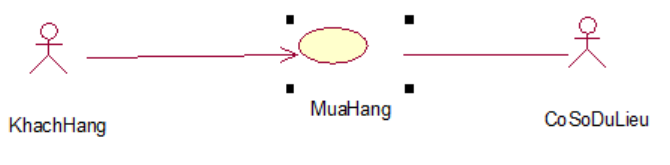
* Điểm mở rộng:

Không có.

1. Biểu đồ trình tự:

### Use case mua hàng

1. Biểu đồ use case:



1. Đặc tả use case:

Use case cho phép khách hàng đặt mua các sách đã được thêm vào giỏ hàng.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi khách hàng bấm vào nút giỏ hàng trên màn hình. Hệ thống hiển thị màn hình thông tin các sách gồm: name, image, price từ bảng PRODUCTS và quantity từ bảng PRODUCT\_ ORDER hiển thị lên màn hình
2. Khách hành nhập các thông tin được hiện thị trên khung thông tin đặt hàng: customer\_address, phone, pay\_method, email, note, status từ bảng ORDERS.
3. Khách hàng kích vào nút “Đặt hàng”. Hệ thống tạo một đơn hàng mới trong bảng PRODUCT\_ ORDER và gửi email thông báo đặt hàng thành công, thông tin đơn hàng vừa đặt.
4. Use case kết thúc.
   * Luồng rẽ nhánh:
5. Tại bước 1 trong luồng cơ bản, khi khách hàng bấm mua hàng mà chưa chọn số lượng hệ thống sẽ tự động đặt số lượng sản phẩm đó là 1 và thêm vào giỏ hàng.
6. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Khách hàng đăng nhập thành công.

* Hậu điều kiện:

Không có.

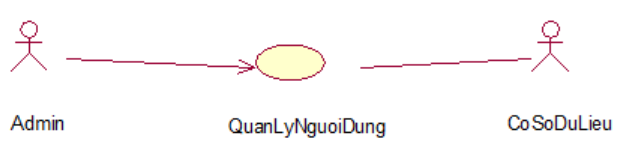
* Điểm mở rộng:

Không có.

1. Biểu đồ trình tự:

### Use case quản lý người dùng

1. Biểu đồ use case:



Hình .. Biểu đồ use case quản lý tài khoản

1. Đặc tả use case:

Use case này cho phép quản trị viên xem, thêm, sửa thông tin, xóa các tài khoản trong bảng USER.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi quản trị viên kích vào nút “User management” trên thanh Menu quản trị. Hệ thống lấy thông tin chi tiết của các tài khoản (FirstName, LastName, Username, Phone, Email, Address) từ bảng USER trong cơ sở dữ liệu và hiển thị lên màn hình.
2. Thêm mới tài khoản:
   1. Quản trị viên kích vào nút “ADD USER”. Hệ thống hiển thị “User Form” yêu cầu quản trị viên nhập các thông tin (First Name, Last Name, Email, Mobile, Address, Username, Password, Confirm Password, Role, Amount).
   2. Quản trị viên nhập các thông tin (First Name, Last Name, Email, Mobile, Address, Username, Password, Confirm Password, Role, Amount) và kích vào nút “Submit”. Hệ thống tạo ra UserID và lưu các thông tin (UserID, First Name, Last Name, Username, Password, Phone, Email, Address, Role, Amount, CreatedAt, UpdatedAt) vào bảng USER và hiển thị danh sách các tài khoản lên màn hình.
3. Xem chi tiết tài khoản:
   1. Quản trị viên kích vào nút “View” trên dòng tài khoản. Hệ thống lấy thông tin chi tiết của tài khoản (First Name, Last Name, Username, Phone, Email, Address, Role, Amount, CreatedAt, UpdatedAt) từ bảng USER trong cơ sở dữ liệu và hiển thị lên màn hình.
   2. Quản trị viên kích vào biểu tượng “X” trên cửa sổ “User Form”. Hệ thống sẽ hiển thị lại danh sách tài khoản trên màn hình.
4. Sửa thông tin tài khoản:
   1. Quản trị viên kích vào nút “Edit” trên dòng tài khoản. Hệ thống hiển thị một màn hình với các thông tin cũ của tài khoản (First Name, Last Name, Username, Phone, Email, Address, Role, Amount, Password).
   2. Quản trị viên sửa thông tin (First Name, Last Name, Phone, Email, Address, Mobile, Role, Amount) của tài khoản và kích vào nút “Submit”. Hệ thống sẽ cập nhật thông tin tài khoản trong bảng USER và hiển thị danh sách tài khoản đã cập nhật.
5. Xóa tài khoản:
   1. Quản trị viên kích vào nút “Delete” trên một dòng tài khoản. Hệ thống sẽ hiển thị màn hình “Delete User” yêu cầu xác nhận xóa.
   2. Quản trị viên kích vào nút “Yes”. Hệ thống sẽ xóa tài khoản khỏi bảng USER và hiển thị danh sách các tài khoản đã cập nhật.
6. Use case kết thúc.
   * Luồng rẽ nhánh:
7. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì website sẽ hiển thị một thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Quản trị viên phải đăng nhập vào hệ thống trước thực hiện quản lý tài khoản.

* Hậu điều kiện:

Không có.

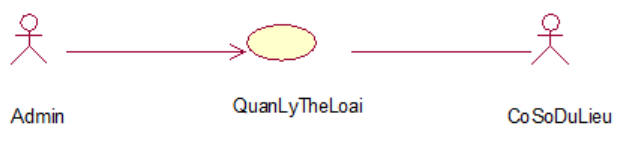
* Điểm mở rộng:

Không có.

1. Biểu đồ trình tự:

### Use case quản lý thể loại sản phẩm

1. Biểu đồ use case:



Hình .. Biểu đồ use case quản lý danh mục

1. Đặc tả use case:

Use case này cho phép quản trị viên xem, thêm, sửa thông tin, xóa các danh mục trong bảng CATEGORY.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi quản trị viên kích vào nút “Category management” trên thanh Menu quản trị. Hệ thống lấy thông tin chi tiết của các danh mục (Name, Description) từ bảng CATEGORY trong cơ sở dữ liệu và hiển thị lên màn hình.
2. Thêm mới danh mục:
   1. Quản trị viên kích vào nút “ADD CATEGORY”. Hệ thống hiển thị “Category Form” yêu cầu quản trị viên nhập các thông tin (Name, Description).
   2. Quản trị viên nhập các thông tin (Name, Description) và kích vào nút “Submit”. Hệ thống tạo ra CategoryID, Slug từ Name và lưu các thông tin (CategoryID, Name, Description, CreatedAt, UpdatedAt, Slug) vào bảng CATEGORY và hiển thị danh sách các danh mục lên màn hình.
3. Xem chi tiết danh mục:
   1. Quản trị viên kích vào nút “View” trên dòng danh mục. Hệ thống lấy thông tin chi tiết của danh mục (Name, Description, CreatedAt, UpdatedAt) từ bảng CATEGORY trong cơ sở dữ liệu và hiển thị lên màn hình.
   2. Quản trị viên kích vào biểu tượng “X” trên cửa sổ “Category Form”. Hệ thống sẽ hiển thị lại danh sách danh mục trên màn hình.
4. Sửa thông tin danh mục:
   1. Quản trị viên kích vào nút “Edit” trên dòng danh mục. Hệ thống hiển thị một màn hình với các thông tin cũ của danh mục (Name, Description).
   2. Quản trị viên sửa thông tin (Name, Description) của danh mục và kích vào nút “Submit”. Hệ thống sẽ cập nhật thông tin danh mục trong bảng CATEGORY và hiển thị danh sách danh mục đã cập nhật.
5. Xóa danh mục:
   1. Quản trị viên kích vào nút “Delete” trên một dòng danh mục. Hệ thống sẽ hiển thị màn hình “Delete Category” yêu cầu xác nhận xóa.
   2. Quản trị viên kích vào nút “Yes”. Hệ thống sẽ xóa danh mục khỏi bảng CATEGORY và hiển thị danh sách các danh mục đã cập nhật.
6. Use case kết thúc.
   * Luồng rẽ nhánh:
7. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì website sẽ hiển thị một thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Quản trị viên phải đăng nhập vào hệ thống trước thực hiện quản lý danh mục.

* Hậu điều kiện:

Không có.

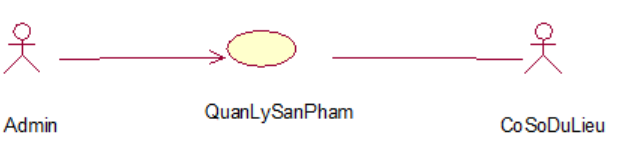
* Điểm mở rộng:

Không có.

1. Biểu đồ trình tự:

### Use case quản lý sản phẩm

1. Biểu đồ use case:



Hình .. Biểu đồ use case quản lý sản phẩm

1. Đặc tả use case:

Use case này cho phép quản trị viên xem, thêm, sửa thông tin, xóa các sách trong bảng PRODUCT.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi quản trị viên kích vào nút “Product management” trên thanh Menu quản trị. Hệ thống lấy thông tin chi tiết của các sách (Title, Price, Author) từ bảng PRODUCT trong cơ sở dữ liệu và hiển thị lên màn hình.
2. Thêm mới sách:
   1. Quản trị viên kích vào nút “ADD PRODUCT”. Hệ thống hiển thị “Product Form” yêu cầu quản trị viên nhập các thông tin (Title, Long Description, Price, Author, Current Number, Number Of Page, Category).
   2. Quản trị viên nhập các thông tin (Title, Long Description, Price, Author, Current Number, Number Of Page, Category) và kích vào nút “Submit”. Hệ thống tạo ra ProductID, Slug từ Title và lưu các thông tin (ProductID, Title, Long Description, Price, Author, Current Number, Number Of Page, CategoryID, CreatedAt, UpdatedAt, Slug) vào bảng PRODUCT và hiển thị danh sách các sách lên màn hình.
3. Xem chi tiết sách:
   1. Quản trị viên kích vào nút “View” trên dòng sách. Hệ thống lấy thông tin chi tiết của sách (Title, Long Description, Price, Author, Current Number, Number Of Page, Quantity Purchased, Category, CreatedAt, UpdatedAt) từ bảng PRODUCT trong cơ sở dữ liệu và hiển thị lên màn hình.
   2. Quản trị viên kích vào biểu tượng “X” trên cửa sổ “Product Form”. Hệ thống sẽ hiển thị lại danh sách sách trên màn hình.
4. Sửa thông tin tài khoản:
   1. Quản trị viên kích vào nút “Edit” trên dòng sách. Hệ thống hiển thị một màn hình với các thông tin cũ của sách (Title, Long Description, Price, Author, Current Number, Number Of Page, Quantity Purchased, Category).
   2. Quản trị viên sửa thông tin (Title, Long Description, Price, Author, Current Number, Number Of Page, Quantity Purchased, Category) của sách và kích vào nút “Submit”. Hệ thống sẽ cập nhật thông tin sách trong bảng PRODUCT và hiển thị danh sách sách đã cập nhật.
5. Xóa sách:
   1. Quản trị viên kích vào nút “Delete” trên một dòng sách. Hệ thống sẽ hiển thị màn hình “Delete Product” yêu cầu xác nhận xóa.
   2. Quản trị viên kích vào nút “Yes”. Hệ thống sẽ xóa sách khỏi bảng Product và hiển thị danh sách các sách đã cập nhật.
6. Use case kết thúc.
   * Luồng rẽ nhánh:
7. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì website sẽ hiển thị một thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Quản trị viên phải đăng nhập vào hệ thống trước thực hiện quản lý sách.

* Hậu điều kiện:

Không có.

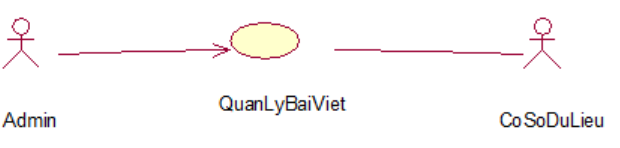
* Điểm mở rộng:

Không có.

1. Biểu đồ trình tự:

### Use case quản lý bài viết

1. Biểu đồ use case:



Hình .. Biểu đồ use case quản lý bài viết

1. Đặc tả use case:

Use case này cho phép quản trị viên xem, thêm, sửa thông tin, xóa các sách trong bảng PRODUCT.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi quản trị viên kích vào nút “Product management” trên thanh Menu quản trị. Hệ thống lấy thông tin chi tiết của các sách (Title, Price, Author) từ bảng PRODUCT trong cơ sở dữ liệu và hiển thị lên màn hình.
2. Thêm mới sách:
   1. Quản trị viên kích vào nút “ADD PRODUCT”. Hệ thống hiển thị “Product Form” yêu cầu quản trị viên nhập các thông tin (Title, Long Description, Price, Author, Current Number, Number Of Page, Category).
   2. Quản trị viên nhập các thông tin (Title, Long Description, Price, Author, Current Number, Number Of Page, Category) và kích vào nút “Submit”. Hệ thống tạo ra ProductID, Slug từ Title và lưu các thông tin (ProductID, Title, Long Description, Price, Author, Current Number, Number Of Page, CategoryID, CreatedAt, UpdatedAt, Slug) vào bảng PRODUCT và hiển thị danh sách các sách lên màn hình.
3. Xem chi tiết sách:
   1. Quản trị viên kích vào nút “View” trên dòng sách. Hệ thống lấy thông tin chi tiết của sách (Title, Long Description, Price, Author, Current Number, Number Of Page, Quantity Purchased, Category, CreatedAt, UpdatedAt) từ bảng PRODUCT trong cơ sở dữ liệu và hiển thị lên màn hình.
   2. Quản trị viên kích vào biểu tượng “X” trên cửa sổ “Product Form”. Hệ thống sẽ hiển thị lại danh sách sách trên màn hình.
4. Sửa thông tin tài khoản:
   1. Quản trị viên kích vào nút “Edit” trên dòng sách. Hệ thống hiển thị một màn hình với các thông tin cũ của sách (Title, Long Description, Price, Author, Current Number, Number Of Page, Quantity Purchased, Category).
   2. Quản trị viên sửa thông tin (Title, Long Description, Price, Author, Current Number, Number Of Page, Quantity Purchased, Category) của sách và kích vào nút “Submit”. Hệ thống sẽ cập nhật thông tin sách trong bảng PRODUCT và hiển thị danh sách sách đã cập nhật.
5. Xóa sách:
   1. Quản trị viên kích vào nút “Delete” trên một dòng sách. Hệ thống sẽ hiển thị màn hình “Delete Product” yêu cầu xác nhận xóa.
   2. Quản trị viên kích vào nút “Yes”. Hệ thống sẽ xóa sách khỏi bảng Product và hiển thị danh sách các sách đã cập nhật.
6. Use case kết thúc.
   * Luồng rẽ nhánh:
7. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì website sẽ hiển thị một thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Quản trị viên phải đăng nhập vào hệ thống trước thực hiện quản lý sách.

* Hậu điều kiện:

Không có.

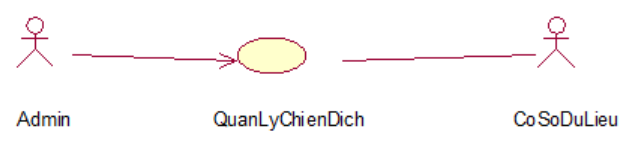
* Điểm mở rộng:

Không có.

1. Biểu đồ trình tự:

### Use case quản lý chiến dịch

1. Biểu đồ use case:

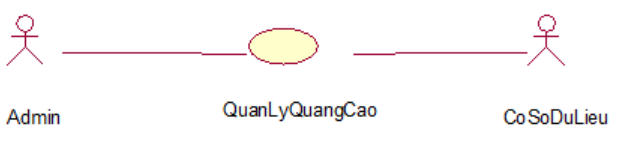


Hình .. Biểu đồ use case quản lý

1. Đặc tả use case:

### Use case quản lý quảng cáo

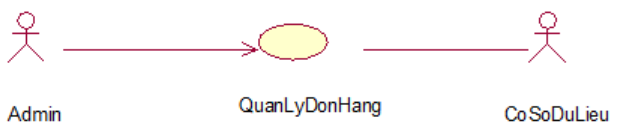
1. Biểu đồ use case:



1. Đặc tả use case:
   * Luồng rẽ nhánh:
2. Biểu đồ trình tự:

### Use case danh sách đơn hàng

1. Biểu đồ use case:



Hình .. Biểu đồ use case danh sách đơn hàng

1. Đặc tả use case:

Use case này cho phép quản trị viên xem, sửa trạng thái đơn hàng trong bảng SALE\_ORDER và ORDER\_ITEM.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi quản trị viên kích vào nút “Sale orders management” trên thanh Menu quản trị. Hệ thống lấy thông tin của đơn hàng (SaleOrderID, CustomerName, CustomerPhone) từ bảng SALE\_ORDER và (Value) từ bảng DELIVERY trong cơ sở dữ liệu và hiển thị lên màn hình.
2. Xem thông tin đơn hàng:
   1. Quản trị viên kích vào nút “View” trên dòng đơn hàng. Hệ thống lấy thông tin của đơn hàng (SaleOrderID, CreatedAt, Customer Name, Customer Phone) từ bảng SALE\_ORDER, (Value) từ bảng DELIVERY, (ProductID, Quantity) từ bảng ORDER\_ITEM và (Title, Price) từ bảng PRODUCT trong cơ sở dữ liệu và hiển thị thông tin đơn hàng lên màn hình.
   2. Quản trị viên kích vào biểu tượng “X” trên cửa sổ “View Sale Order”. Hệ thống sẽ hiển thị lại danh sách thông tin ảnh trên màn hình.
3. Sửa trạng thái đơn hàng:
   1. Quản trị viên kích vào nút “Edit” trên dòng đơn hàng. Hệ thống hiển thị một màn hình với các thông tin cũ của đơn hàng (SaleOrderID, Value).
   2. Quản trị viên sửa thông tin (Value) của đơn hàng và kích vào nút “Submit”. Hệ thống sẽ cập nhật thông tin đơn hàng trong bảng SALE\_ORDER sau đó hiển thị danh sách đơn hàng đã cập nhật.
4. Use case kết thúc.
   * Luồng rẽ nhánh:
5. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì website sẽ hiển thị một thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Quản trị viên phải đăng nhập vào hệ thống trước thực hiện quản lý đơn hàng.

* Hậu điều kiện:

Không có.

* Điểm mở rộng:

Không có.

1. Biểu đồ trình tự:

## Thiết kế hệ thống

### **Diagram Description automatically generated**Biểu đồ triển khai hệ thống

Hình .. Biểu đồ triển khai hệ thống

### Thiết kế cơ sở dữ liệu

#### Biểu đồ mô tả cấu trúc cơ sở dữ liệu

Hình .. Biều đồ mô tả cấu trúc cơ sở dữ liệu

#### Mô tả cơ sở dữ liệu

Bảng .. Bảng Category

Bảng Category để lưu thông tin danh mục sản phẩm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu | Độ dài | Khóa | Giải thích |
| CategoryID | BigInt |  | PK | Id danh mục |
| Name | NVarChar | 255 |  | Tên danh mục |
| Description | Text |  |  | Mô tả danh mục |
| Slug | VarChar | 255 |  | Mô tả URL thân thiện |
| IsAuthor | Bit |  |  | Là tác giả |
| CreatedAt | DateTime |  |  | Thời gian tạo |
| UpdatedAt | DateTime |  |  | Thời gian cập nhật |
| ParentCategoryID | BigInt |  | FK | Id danh mục cha |

Bảng .. Bảng Product

Bảng Product để lưu thông tin sản phẩm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu | Độ dài | Khóa | Giải thích |
| ProductID | BigInt |  | PK | Id sản phẩm |
| Title | NVarChar | 255 |  | Tiêu đề |
| ShortDescription | NVarChar | 255 |  | Mô tả ngắn |
| LongDescription | Text |  |  | Mô tả |
| Price | BigInt |  |  | Giá |
| Author | NVarChar | 255 |  | Tác giả |
| CurrentNumber | Integer |  |  | Số lượng còn hiện tại |
| NumberOfPage | Integer |  |  | Số trang |
| Slug | VarChar | 255 |  | Mô tả URL thân thiện |
| QuantityPurchased | Integer |  |  | Số lượng đã được mua |
| CreatedAt | DateTime |  |  | Thời gian tạo |
| UpdatedAt | DateTime |  |  | Thời gian cập nhật |
| CategoryID | BigInt |  | FK | Id danh mục |

Bảng .. Bảng ProductImage

Bảng ProductImage để lưu thông tin hình ảnh sản phẩm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu | Độ dài | Khóa | Giải thích |
| ProductImageID | BigInt |  | PK | Id ảnh sản phẩm |
| ImageUrl | VarChar | 255 |  | Đường dẫn ảnh |
| ImagePublicID | VarChar | 255 |  | Id công khai ảnh |
| CreatedAt | DateTime |  |  | Thời gian tạo |
| UpdatedAt | DateTime |  |  | Thời gian cập nhật |
| ProductID | BigInt |  | FK | Id sản phẩm |

Bảng .. Bảng Delivery

Bảng Delivery để lưu thông tin trạng thái giao hàng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu | Độ dài | Khóa | Giải thích |
| DeliveryID | BigInt |  | PK | Id trạng thái giao hàng |
| Index | VarChar | 255 |  | Chỉ mục trạng thái |
| Value | NVarChar | 255 |  | Giá trị trạng thái |
| CreatedAt | DateTime |  |  | Thời gian tạo |
| UpdatedAt | DateTime |  |  | Thời gian cập nhật |

Bảng .. Bảng User

Bảng User để lưu thông tin người dùng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu | Độ dài | Khóa | Giải thích |
| UserID | BigInt |  | PK | Id người dùng |
| FirstName | NVarChar | 255 |  | Tên |
| LastName | NVarChar | 255 |  | Họ |
| Username | VarChar | 255 |  | Tên tài khoản |
| Address | NVarChar | 255 |  | Địa chỉ |
| Password | VarChar | 255 |  | Mật khẩu |
| Role | VarChar | 255 |  | Vai trò |
| Amount | BigInt |  |  | Số tiền |
| Email | VarChar | 255 |  | Địa chỉ email |
| Phone | VarChar | 255 |  | Số điện thoại |
| CreatedAt | DateTime |  |  | Thời gian tạo |
| UpdatedAt | DateTime |  |  | Thời gian cập nhật |

Bảng .. Bảng SaleOrder

Bảng SaleOrder để lưu thông tin đơn hàng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu | Độ dài | Khóa | Giải thích |
| SaleOrderID | BigInt |  | PK | Id đơn hàng |
| CustomerAddress | NVarChar | 255 |  | Địa chỉ nhận hàng |
| Phone | VarChar | 255 |  | Số điện thoại |
| UserID | BigInt |  | FK | Id người dùng |
| DeliveryID | BigInt |  | FK | Id trạng thái giao hàng |
| CreatedAt | DateTime |  |  | Thời gian tạo |
| UpdatedAt | DateTime |  |  | Thời gian cập nhật |
| OrderedAt | DateTime |  |  | Thời gian đặt hàng |

Bảng .. Bảng OrderItem

Bảng OrderItem để lưu thông tin mặt hàng trong đơn hàng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu | Độ dài | Khóa | Giải thích |
| OrderItemID | BigInt |  | PK | Id mặt hàng |
| ProductID | BigInt |  | PFK | Id sản phẩm |
| SaleOrderID | BigInt |  | PFK | Id đơn hàng |
| Quantity | Integer |  |  | Số lượng |
| CreatedAt | DateTime |  |  | Thời gian tạo |
| UpdatedAt | DateTime |  |  | Thời gian cập nhật |

Bảng .. Bảng ProductRating

Bảng ProductRating để lưu thông tin đánh giá của sản phẩm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu | Độ dài | Khóa | Giải thích |
| ProductRatingID | BigInt |  | PK | Id đánh giá sản phẩm |
| Value | Integer |  |  | Giá trị đánh giá |
| Comment | NVarChar | 255 |  | Bình luận |
| UserID | BigInt |  | FK | Id người dùng |
| ProductID | BigInt |  | FK | Id sản phẩm |
| CreatedAt | DateTime |  |  | Thời gian tạo |
| UpdatedAt | DateTime |  |  | Thời gian cập nhật |

Bảng .. Bảng Banner

Bảng Banner để lưu thông tin ảnh bìa

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu | Độ dài | Khóa | Giải thích |
| BannerID | BigInt |  | PK | Id ảnh bìa |
| Title | NVarChar | 255 |  | Tiêu đề |
| ImageUrl | VarChar | 255 |  | Đường dẫn ảnh |
| Type | Integer |  |  | Kiểu |
| IsActive | Bit |  |  | Trạng thái kích hoạt |
| CreatedAt | DateTime |  |  | Thời gian tạo |
| UpdatedAt | DateTime |  |  | Thời gian cập nhật |

### Thiết kế giao diện

#### Giao diện chung

#### Giao diện trang admin

# KẾT LUẬN

Đề tài **“Xây dựng bán trang sức cửa hàng Thu Hà”** xuất phát từ những nhu cầu thực tế mà ngày nay trong ngành kinh doanh cần có. Những phương pháp bán hàng truyền thống dần lỗi thời và thay vào đó việc bán hàng qua website nâng cao hiệu quả hơn, khắc phục được những tồn đọng khó khăn mà phương pháp thủ công đang gặp phải.

Qua quá trình thực hiện đề tài, em đã tìm hiểu, tích lũy và học hỏi thêm được nhiều kinh nghiệm cũng như kiến thức công nghệ mới.

**\*Những kết quả đạt được:**

**Về công nghệ:**

* Hiểu được quá trình thiết kế một website thực tế đi từ bước cơ bản: khảo sát dự án, phân tích hệ thống, thiết kế, thực hiện, kiểm thử, triển khai, bảo trì.
* Xây dựng thành công website đáp ứng nhu cầu bán hàng và quảng bá thương hiệu cho hãng sách Bookshop.
* Nắm được các kiến thức xây dựng giao diện website: HTML, CSS, JavaScript.
* Nắm vững về cách xây dựng website thông qua thư viện, framework: React, Java và Spring Boot trong lập trình.
* Sử dụng thành thạo các công cụ hỗ trợ lập trình: Visual Studio Code, IntelliJ.

**Về cài đặt chương trình:**

* Cho phép khách hàng có thể xem thông tin sản phẩm và có thể mua hàng trực tuyến.
* Tiếp nhận đơn đặt hàng của khách hàng.
* Các mặt hàng được cập nhật liên tục theo nhu cầu của khách hàng.
* Trang quản trị đầy đủ chức năng để quản lý một trang web: quản lý tài khoản, danh mục. sản phẩm, đơn hàng.

**Tính năng khác:**

* Giao diện đẹp gây ấn tượng thân thiện với khách hàng.
* Tính bảo mật.

**\*Hướng phát triển**

* Tiếp tục hoàn thiện các tính năng mở rộng phần mềm nhằm cải tiến và nâng cấp chương trình.
* Hoàn thiện chương trình, sửa chữa các lỗi, cải thiện tính bảo mật.
* Xây dựng Website quy mô lớn hơn với nhiều ứng dụng như: trò chuyện, tư vấn khách hàng.
* Tích hợp thêm chức năng thanh toán với ví điện tử.

# TÀI LIỆU THAM KHẢO

[1] Hoàng Quang Huy, Phùng Đức Hòa, Trịnh Bá Quý, *Nhập môn công nghệ phần mềm*, NXB Đại học Công nghiệp Hà Nội.

[2] Nguyễn Thị Thanh Huyền, Ngô Thị Bích Thúy, Phạm Thị Kim Phượng, *Giáo trình phân tích thiết kế hệ thống*, NXB Giáo dục VN.

[3] Nguyễn Bá Nghiễn, *Lập trình java*, NXB thông tin và truyền thông.

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

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

[6] Trang tài liệu Java <https://www.w3schools.com/java/default.asp>

[7] Trang tài liệu Spring Boot <https://spring.io/projects/spring-boot>

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

[9] Trang tài liệu ReactJS <https://reactjs.org>

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