|  |  |
| --- | --- |
|  |  |
| LÊ BÁ ĐỨC ANH | **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **---------------------------------------** |
|  |
| ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC  NGÀNH CÔNG NGHỆ THÔNG TIN |
|  |
| **Xây dựng Website bán sách sử dụng Django và ReactJS** |
|  |
|  |
| **CBHD: TS. Vũ Việt Thắng** |
| CÔNG NGHỆ THÔNG TIN | **Sinh viên: Lê Bá Đức Anh** |
| **Mã số sinh viên: 2019603016** |
|  |
|  |
|  |
| 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 21](#_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 24](#_Toc102342494)

[3.4. Đặc tả use case 25](#_Toc102342495)

[3.5. Thiết kế hệ thống 77](#_Toc102342496)

[KẾT LUẬN 84](#_Toc102342497)

[TÀI LIỆU THAM KHẢO 86](#_Toc102342498)

# LỜI NÓI ĐẦU

Ngày nay, việc ứng dụng công nghệ thông tin vào quản lý kinh doanh đã phát triển mạnh ở các nước tiên tiến trên thế giới. Đặc biệt trong mùa dịch, việc mua bán hàng là rất cần thiết. Trước tình hình đó, vấn đề áp dụng Tin học để quản lý, mua bán hàng dường như đã thay thế rất nhiều cho các phương thức mua hàng truyền thống, nổi trội như hàng loạt các ứng dụng mua hàng ra đời như Shopee, Lazada, Tiki.

Việc mua bán hàng online đang là một trong những nhu cầu hết sức thiết yếu hiện nay, đặc biệt là nhu cầu về mua sách, do đó hệ thống hỗ trợ mua những cuốn sách online cũng là một trong những vấn đề đang được quan tâm. Việc xây dựng website thương mại điện tử bán sách trực tuyến góp phần giúp người dùng dễ dàng tìm hiểu và đặt mua những cuốn sách một cách dễ dàng.

Sau khi tìm hiểu và dưới sự hướng dẫn của thầy Vũ Việt Thắng, em đã phân tích và thiết kế được **Website bán sách sử dụng Django và ReactJS**. Để giúp khách hàng đặt mua được những cuốn sách theo 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. Thầy giáo hướng dẫn đề tài – **Tiến sĩ Vũ Việt Thắng**, 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 27 tháng 03 năm 2023***

# DANH MỤC CÁC CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| Từ viết tắt | Giải thích |
| API | Application Programming Interface |
| OOP | Object Oriented Programming |
| HTML | HyperText Markup Language |
| CSS | Cascading Style Sheets |
| ORM | Object Relational Mapping |
| HTTP | Hypertext Transfer Protocol |
| CRM | Customer Relationship Management |
| MVC | Model-View-Controller |
| UI | User Interface |
| GUI | Graphical User Interface |
| B2BCRM | Business to Business Customer Relationship Management |
| CSDL | Cơ sở dữ liệu |
| XML | Extensible Markup Language |
| MVCC | Multiversion concurrency control |
| PK | Primary key |
| FK | Foreign Key |

# DANH MỤC BẢNG BIỂU

[Bảng 3.1. Bảng Category 89](#_Toc131147028)

[Bảng 3.2. Bảng Book 90](#_Toc131147029)

[Bảng 3.3. Bảng Author 90](#_Toc131147030)

[Bảng 3.4. Bảng Review 91](#_Toc131147031)

[Bảng 3.5. Bảng User 92](#_Toc131147032)

[Bảng 3.6. Bảng Order 92](#_Toc131147033)

[Bảng 3.7. Bảng OrderDetail 93](#_Toc131147034)

[Bảng 3.8. Bảng Transaction 93](#_Toc131147035)

# DANH MỤC HÌNH ẢNH

[Hình 2.1. Python logo 7](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188906)

[Hình 2.2. JavaScript logo 9](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188907)

[Hình 2.3. Kiến trúc của Django Framework 14](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188908)

[Hình 2.4. React logo 16](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188909)

[Hình 2.5. MySQL logo 19](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188910)

[Hình 3.1. Biểu đồ use case tổng quát 24](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188911)

[Hình 3.2. Mô hình thực thể liên kết 25](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188912)

[Hình 3.3. Biểu đồ use case đăng nhập 25](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188913)

[Hình 3.4. Biểu đồ trình tự đăng nhập 28](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188914)

[Hình 3.5. Biểu đồ use case đăng ký 29](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188915)

[Hình 3.6. Biểu đồ trình tự đăng ký 30](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188916)

[Hình 3.7. Biểu đồ use case tìm kiếm 31](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188917)

[Hình 3.8. Biểu đồ trình tự tìm kiếm 32](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188918)

[Hình 3.9. Biểu đồ use case xem chi tiết sách 32](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188919)

[Hình 3.10. Biểu đồ trình tự xem chi tiết sách 34](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188920)

[Hình 3.11. Biểu đồ use case sửa thông tin cá nhân 34](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188921)

[Hình 3.12. Biểu đồ trình tự sửa thông tin cá nhân 36](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188922)

[Hình 3.13. Biểu đồ use case quên mật khẩu 37](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188923)

[Hình 3.14. Biểu đồ trình tự quên mật khẩu 39](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188924)

[Hình 3.15. Biểu đồ use case đặt hàng 40](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188925)

[Hình 3.16. Biều đồ trình tự đặt hàng 42](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188926)

[Hình 3.17. Biểu đồ use case quản lý tài khoản 43](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188927)

[Hình 3.18. Biểu đồ trình tự quản lý tài khoản 48](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188928)

[Hình 3.19. Biểu đồ use case quản lý danh mục 49](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188929)

[Hình 3.20. Biều đồ trình tự quản lý danh mục 54](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188930)

[Hình 3.21. Biểu đồ use case quản lý sách 54](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188931)

[Hình 3.22. Biểu đồ trình tự quản lý sách 61](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188932)

[Hình 3.23. Biểu đồ use case quản lý tác giả 61](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188933)

[Hình 3.24. Biểu đồ trình tự quản lý tác giả 67](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188934)

[Hình 3.25. Biểu đồ use case danh sách đơn hàng 68](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188935)

[Hình 3.26. Biểu đồ trình tự danh sách đơn hàng 70](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188936)

[Hình 3.27. Biểu đồ use case quản lý bình luận 71](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188937)

[Hình 3.28. Biểu đồ trình tự quản lý bình luận 73](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188938)

[Hình 3.29. Biểu đồ use case bình luận 74](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188939)

[Hình 3.30. Biểu đồ trình tự bình luận 76](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188940)

[Hình 3.31. Biểu đồ triển khai hệ thống 77](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188941)

[Hình 3.32. Biều đồ mô tả cấu trúc cơ sở dữ liệu 78](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188942)

[Hình 3.33. Giao diện trang chủ 83](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188943)

[Hình 3.34. Giao diện chi tiết sách 83](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188944)

[Hình 3.35. Giao diện trang đăng nhập 84](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188945)

[Hình 3.36. Giao diện trang đăng ký 84](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188946)

[Hình 3.37. Giao diện giỏ hàng 85](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188947)

[Hình 3.38. Giao diện checkout 85](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188948)

[Hình 3.39. Giao diện phương thức thanh toán 86](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188949)

[Hình 3.40. Giao diện xác nhận đặt hàng 86](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188950)

[Hình 3.41. Giao diện trang chủ quản trị 87](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188951)

[Hình 3.42. Giao diện trang đăng nhập quản trị 87](file:///C:\Users\pc\Desktop\Nam%204\ĐATN\DATN_LeBaDucAnh_2019603016.docx#_Toc131188952)

# TỔNG QUAN VỀ ĐỀ TÀI

## Tên đề tài

***Xây dựng Website bán sách sử dụng Django và ReactJS***

## Lý do chọn đề tài

Thương mại điện tử là một thuật ngữ rất phổ biến hiện nay. Nó được nhắc nhiều trên phương tiện truyền thông. 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 có thể hiểu là thực hiện hoạt động thương mại bằng những phương tiện điện tử. Cụ thể hơn, đó là hoạt động giao dịch mua sắm qua internet. Đây là một phương thức kinh doanh sử dụng internet mang lại nhiều hiệu quả như giao dịch nhanh chóng, tiết kiệm 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.

Với sự bùng nổ của cuộc cách mạng công nghiệp 4.0, internet đã có ảnh hưởng sâu sắc đến đời sống, kinh tế và xã hội. Có thể nói internet là một công cụ không thể thiếu trong cuộc sống. Để tồn tại và phát triển, con người cần phải học tập và tiếp cận nhanh chóng với những công nghệ mới, theo kịp sự phát triển của thời đại.

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.

Sách là một người bạn, một người thầy đối với chúng ta. Nó vừa giúp chúng ta nâng cao hiểu biết lại vừa giúp chúng ta giải trí thư giãn. Việc đọc sách trở nên dễ dàng hơn với mạng internet. Ngày nay, mọi người có thể lên trên internet và tìm đọc những cuốn sách điện tử. Tuy nhiên, việc đọc sách trên internet thường gây ảnh hưởng đến thị lực nếu đọc trong thời gian dài. Bên cạnh đó nhiều loại sách không có sẵn trên mạng vì lý do bản quyền. Để giúp người đọc có thể mua sách tại nhà dễ dàng, tìm thấy những quyển sách phù hợp mà không tốn nhiều thời gian. Các nhà sách, nhà xuất bản cần có những website để khách hàng có thể mua sách, tìm những quyển sách họ cần. Nhu cầu thiết kế các website để bán sách là cần thiết.

Đối với website hàng nói chung và website bán sách 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 nhà sách, của nhà xuất bản 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 nhà sách. Khách hàng nhìn vào website có thể dễ dàng sử dụng các công cụ có trên đó. Về hoạt động của website thì website phải chạy mượt mà để khách hàng thoải mái, không bị khó chịu khi sử dụng. Website giúp khán giả có thể xem được thông tin mình cần, có thể tìm tới sản phẩm mình cần. Nó chính là bộ mặt của thương hiệu, là phương pháp gây ấn tượng và thu hút khách hàng của các nhà sách, các nhà xuất bản.

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 sách sử dụng Django và ReactJS”.***

## Mục tiêu của đề tài

Đề tài: ***Xây dựng Website bán sách sử dụng Django và ReactJS*** 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 Python, Django, JavaScript, ReactJS.
* Lợi ích đầu tiên khi sở hữu một website bán hàng đó chính là việc tiếp cận được những khách hàng tiềm năng. Dù khách hàng có ở bất cứ đâu trên thế giới, chỉ cần có mạng internet là có thể tìm thấy website bán hàng, biết đến sản phẩm và thương hiệu của công ty. Từ đó, chắc chắn rằng lượng khách hàng sẽ tăng lên đáng kể.
* Tiết kiệm được thời gian 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. Hơn nữa mặt hàng khách mua còn có thể được chuyển về địa chỉ yêu cầu đúng thời gian khách hàng mong muốn, tiết kiệm được cả chi phí đi lại cho khách hàng.
* Thông tin sản phẩm được cập nhật nhanh chính xác qua từng ngày đến khách hàng**.**
* Xây dựng và quảng bá thương hiệu cho hãng.

## Đối tượng và phạm vi

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

## Kết quả dự kiến đạt được

Website bao gồm những mục chính và chức năng chính:

* 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, tác giả.
* Tìm kiếm sản phẩm, tìm kiếm sản phẩm theo chủ đề, xem chi tiết sản phẩm.
* Thêm sản phẩm vào giỏ hàng và lưu trữ giỏ hàng của mỗi người dùng.
* Đặt hàng, thanh toán.

# 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.

UML dùng để:

* Trực quan hoá: Đưa ra các mô hình trực quan về hệ thống để mọi người có thể hiểu được.
* Đặc tả: UML cho phép xây dựng các mô hình đúng đắn và đầy đủ để mô tả hệ thống.
* Xây dựng: Các mô hình UML có thể liên kết trực tiếp với nhiều ngôn ngữ lập trình Java, C++, VB, ...
* Lập tài liệu: UML cho phép tạo các tài liệu như tài liệu về các yêu cầu của người dùng, kiến trúc hệ thống, thiết kế hệ thống, kiểm thử, kế hoạch dự án, ...

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.

Các phần tử của UML:

View (Quan sát): Theo các phương diện khác nhau của hệ thống cần phân tích, thiết kế. Dựa vào các quan sát để thiết lập kiến trúc cho hệ thống cần phát triển.

***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.

### Ngôn ngữ Python

Hình .. Python logo

Logo, company name

Description automatically generated

**Python** là một ngôn ngữ lập trình hướng đối tượng, thông dịch và cấp phát bộ nhớ động. Với các ưu điểm chính được các developer đi trước đánh giá như:

* Chu trình chỉnh sửa, kiểm tra lỗi nhanh, gỡ lỗi dễ dàng với một trình debugger được viết bằng chính python.
* Cú pháp đơn giản, dễ học dễ đọc, cấu trúc rõ ràng, sáng sủa giúp làm giảm chi phí bảo trì. Code cũng ngắn gọn giúp lập trình một cách nhanh chóng và đạt hiệu quả cao.
* Độ tin cậy cao với rất nhiều module, thư viện hỗ trợ để làm từng công việc, đặc biệt phục vụ cho xử lý dữ liệu lớn giúp xây dựng các hệ thống Recommend System với Deep Learning - một cụm từ đang hot hiện nay.
* Dễ dàng để kết hợp với các ngôn ngữ lập trình như: C, C++, Java

Python là một ngôn ngữ hướng đối tượng. Vì vậy hầu hết mọi thứ trong Python đều là những đối tượng với những thuộc tính (properties) và phương thức (methods).

OOP là một kĩ thuật lập trình cho phép tạo ra các đối tượng để trừu tượng hóa 1 đối tượng thực tế (đưa các đối tượng trong thực tế vào trong code). Cho phép lập trình viên tương tác với các đối tượng này.

Một đối tượng bao gồm: thuộc tính (attributes) và phương thức (methods)

* Thuộc tính (attributes) chính là những thông tin, đặc điểm của đối tượng. VD: Con mèo (màu sắc, có đuôi, 2 tai, bốn chân, ...)
* Phương thức (methods) là những thao tác, hành động mà đối tượng đó có thể thực hiện. VD: Con mèo (ăn, chạy nhảy, cắn, gào, rên, ....)

Lớp có thể hiểu là một bản thiết kế để tạo ra một thực thể nào đó, là tập hợp nhiều thuộc tính đặc trưng cho đối tượng được tạo ra từ lớp này.

OOP tuân theo một số nguyên lý cơ bản, gồm có 4 nguyên lý: Tính đóng gói, Tính kế thừa, Tính bao đóng, Tính đa hình.

* Tính đóng gói (Encapsulation): Các dữ liệu và phương thức có liên quan với nhau được đóng gói thành 1 lớp. Tức là mỗi lớp được xây dựng để thực hiện một nhóm chức năng đặc trưng của riêng lớp đó. Che giấu các thông tin của lớp đó đối với bên ngoài thể hiện ở public, protected, private đối với từng thuộc tính và phương thức.
* Tính kế thừa (Inheritance): Nguyên tắc này cho phép xây dựng một lớp mới dựa trên 1 lớp đã khai báo từ trước. Lớp con có thể sử dụng lại các thuộc tính và phương thức của lớp cha mà không cần khai báo lại. Tùy thuộc vào từng ngôn ngữ cho phép việc kế thừa 1 hoặc nhiều class cha.
* Tính trừu tượng (Abstraction): tổng quát hóa phương thức của đối tượng không quan tâm phương thức thực hiện như thế nào, được thể hiện bởi interface (có các tên phương thức nhưng ko có body của phương thức, khi class nào impliment interface thì thực hiện nó).
* Tính đa hình (Polymorphism): Tính đa hình được thể hiện bởi một phương thức, hành động có thể thực hiện theo nhiều cách khác nhau.

### 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.

### Django Framework

#### Django Framework là gì?

Framework là các đoạn code đã được viết sẵn, cấu thành nên một bộ khung và các thư viện lập trình được đóng gói. Chúng cung cấp các tính năng có sẵn như mô hình, API và các yếu tố khác để tối giản cho việc phát triển các ứng dụng web phong phú, năng động.

Django Framework là một Framework có vai trò phát triển cho các ứng dụng Python Web.

Đặc điểm nổi bật của Django trong Python là tính linh hoạt, độ tin cậy và khả năng mở rộng lớn. Django sở hữu các bảng admin hay còn gọi là admin panel để giúp lập trình viên quản trị dễ hơn so với các framework như Laravel hay Yii, ngoài ra Django còn có một số tính năng kỹ thuật khác như:

* Có web server riêng.
* Sử dụng cấu trục US MVC.
* Có thư viện HTTP.
* Có ORM.
* Bao gồm Batteries (tất cả những yếu tố cần để giải quyết các case thường thấy)..
* Sở hữu Python Unit Test Framework

Thêm vào đó, Django còn cung cấp giao diện CRUD (create, read, update, delete)  được định cấu hình với các mô hình quản trị và được tạo thông qua phần nội quan. CRUD được sử dụng để mô tả các lệnh cơ sở dữ liệu cơ bản, có nghĩa là giao diện tạo điều kiện thuận lợi cho việc xem, thay đổi và tìm kiếm thông tin.

#### Lịch sử phát triển của Django Framework

Django là một framework phát triển web được phát triển bởi Adrian Holovaty và Simon Willison vào năm 2003. Django được phát triển để giúp các nhà phát triển tạo ra các ứng dụng web bằng Python một cách nhanh chóng và dễ dàng.

Ban đầu, Django được phát triển bởi World Online, một công ty Internet ở Lawrence, Kansas. Tuy nhiên, sau đó Django được phát triển và duy trì bởi Django Software Foundation, một tổ chức phi lợi nhuận được thành lập vào năm 2008.

Đến năm 2005, Django đã được phát hành dưới giấy phép BSD open-source và trở thành một trong những framework phát triển web phổ biến nhất trên thế giới. Django đã được sử dụng để phát triển nhiều ứng dụng web nổi tiếng, bao gồm Instagram, Mozilla, Pinterest và The Washington Post.

Hiện nay, Django vẫn được phát triển và cải tiến liên tục, với phiên bản mới nhất là Django 3.2. Django đã trở thành một trong những framework phát triển web phổ biến nhất trên thế giới và được sử dụng rộng rãi trong các ứng dụng web lớn và nhỏ.

#### Mục tiêu của Django Framework

Mục tiêu của Django là giúp các nhà phát triển tạo ra các ứng dụng web bằng Python một cách nhanh chóng và dễ dàng. Để đạt được mục tiêu này, Django cung cấp cho người dùng một số tính năng và lợi ích sau:

* Hiệu suất: Django được thiết kế để cải thiện hiệu suất của các ứng dụng web. Django sử dụng một loạt các kỹ thuật bộ nhớ đệm và tối ưu hóa để tăng tốc độ truy cập cơ sở dữ liệu và giảm thời gian phản hồi.
* Bảo mật: Django cung cấp một loạt các tính năng bảo mật để giảm thiểu các lỗ hổng bảo mật trong ứng dụng web. Django hỗ trợ cho việc xử lý các cuộc tấn công CSRF (Cross-Site Request Forgery) và XSS (Cross-Site Scripting) thông qua các tính năng như middleware và bộ lọc.
* Dễ sử dụng: Django được thiết kế để dễ sử dụng và có tính linh hoạt cao để đáp ứng nhiều nhu cầu khác nhau của các ứng dụng web. Django cung cấp một loạt các tính năng như mô hình ORM (Object-Relational Mapping), quản lý URL và template để giúp người dùng phát triển ứng dụng web nhanh chóng và dễ dàng.
* Cộng đồng: Django có cộng đồng lớn và phát triển liên tục, với rất nhiều tài liệu và hỗ trợ từ các nhà phát triển khác. Điều này giúp người dùng dễ dàng tìm kiếm giải pháp cho các vấn đề phát triển web của mình.

Với những tính năng và lợi ích trên, Django đã trở thành một trong những framework phát triển web phổ biến nhất trên thế giới và được sử dụng rộng rãi trong các ứng dụng web lớn và nhỏ.

#### Kiến trúc của Django Framework

Hình .. Kiến trúc của Django Framework

Diagram

Description automatically generated

Kiến trúc của Django framework được chia thành 3 phần chính:

* Mô hình MTV (Model-Template-View): Đây là mô hình kiến trúc của Django, nơi mà dữ liệu được lưu trữ trong mô hình (Model), giao diện người dùng được định dạng và hiển thị bằng các template (Template), và các truy vấn và xử lý dữ liệu được thực hiện trong các view (View). Qua đó, Django giúp tách biệt các thành phần của ứng dụng và giúp dễ dàng quản lý và phát triển ứng dụng.
* Các ứng dụng (Apps): Django cho phép phát triển các ứng dụng độc lập và có tính tái sử dụng cao. Mỗi ứng dụng cung cấp các chức năng cụ thể và có thể được sử dụng lại trong các ứng dụng khác. Các ứng dụng được phát triển bằng các thư mục riêng biệt và được kết nối với nhau thông qua ứng dụng chính.
* Middleware: Middleware là một phần quan trọng trong kiến trúc của Django, cho phép xử lý các yêu cầu và phản hồi trước và sau khi chúng đi qua các view. Middleware có thể được sử dụng để thêm các tính năng bảo mật, xử lý lỗi, tối ưu hóa hiệu suất và nhiều tính năng khác.

Ngoài ra, Django còn cung cấp một số thành phần khác như Form, URL dispatcher, và các công cụ hỗ trợ cho việc phát triển ứng dụng web như Django ORM, Django Template Engine, và Django Admin. Tất cả các thành phần này đóng vai trò quan trọng trong kiến trúc của Django và giúp người dùng phát triển các ứng dụng web nhanh chóng và dễ dàng.

#### Các dự án của Django Framework

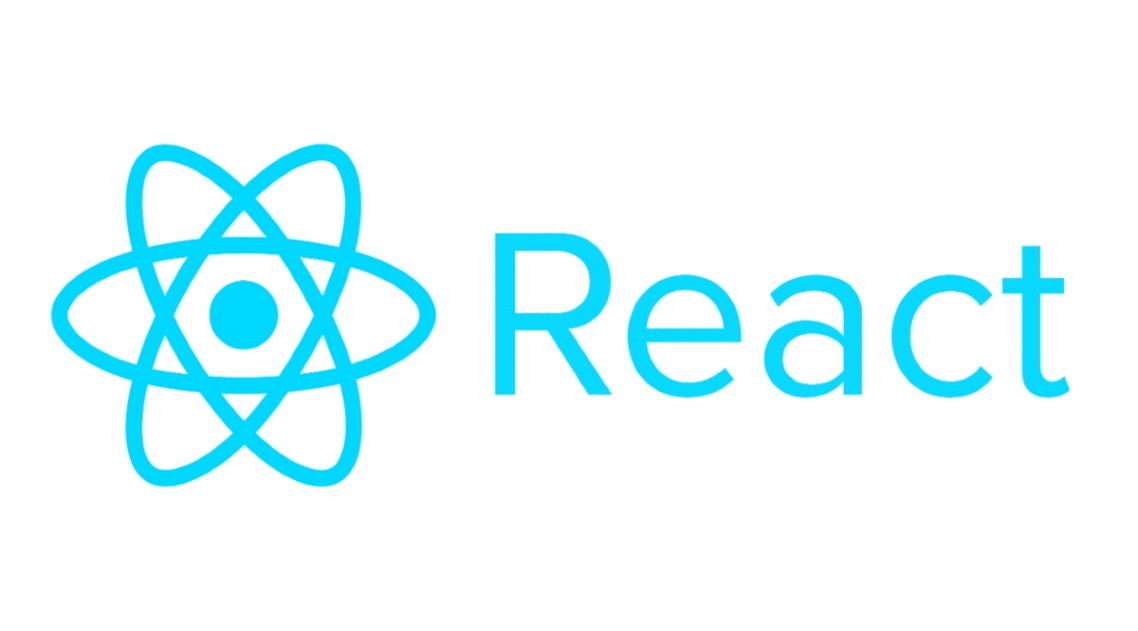
* Chính nền tảng: phân tích chức năng và tính toán kết quả gần đúng dựa trên cá nhân dữ liệu, khả năng chấp nhận rủi ro hoặc xác thực đạt được mục tiêu
* Tích hợp hệ thống CRM chỉnh sửa cho nội bộ dữ liệu
* Hệ thống B2B CRM để xử lý giao tiếp giữa các doanh nghiệp
* Nền tảng hỗ trợ giao tiếp giữa hai bên, chẳng hạn như giữa doanh nghiệp và người dùng
* Tools set the room or background background buy shopping has important to download
* Di động ứng dụng Android và iOS hỗ trợ web ứng dụng
* Hệ thống đánh giá tài sản bất động sản
* Tài liệu quản lý hệ thống
* Nền tảng xử lý các vấn đề pháp lý: như thế chấp điều kiện minh họa hoặc tình trạng cho thuê
* Dự án được thực hiện trên nhiều khuôn khổ

### ReactJS

#### ReactJS là gì?

ReactJS là một thư viện JavaScript mã nguồn mở dùng để xây dựng giao diện người dùng hoặc các thành phần UI.

ReactJS là một thư viện JavaScript với xu hướng Single Page Application.



Hình .. React logo

Hiện tại, nó đã đi trước các đối thủ chính như Angular và [Bootstrap](https://getbootstrap.com/). Được đánh giá cao và trở nên rất phổ biến với hơn 1.300 developer và hơn 94.000 trang web đang sử dụng.

#### Lịch sử phát triển của ReactJS

Trước thời điểm năm 2013 việc xây dựng UI cho Facebook hoàn toàn dựa trên framework AngularJS của Google, tuy nhiên sau thời điểm năm 2013 khi lượng người dùng trở nên quá lớn và dữ liệu Facebook lưu trữ ngày càng nhiều, kéo theo đó việc xử lý dữ liệu ở ngoài UI trở nên quá khó khăn vì AngularJS chậm và khá nặng nề. Facebook không thể tìm kiếm được một framework nào khác đủ khả năng thay thế cho AngularJS, nên đã quyết định tự tạo ra ra một công nghệ mới là “React” gồm 2 phần: ReactJS (dùng cho web app) và React Native (dùng cho mobile app).

React được tạo bởi Jordan Walke, một kỹ sư phần mềm làm việc tại Facebook, người đã tại ra bản mẫu đầu tiền của ReactJS gọi là “FaxJS”. Jordan Walke bị ảnh hưởng bới XHP (một nền tảng thành phần HTML cho PHP). React lần đầu tiên được triển khai cho ứng dụng Newsfeed của Facebook năm 2011, rồi sau đó được triển khai cho Instagram vào năm 2012. Tháng 5 năm 2013, nó được mở mã nguồn (open-sourced) tại JSConfUS nhằm mục đích có nhiều người góp phần phát triển thư viện này và sớm trở thành một framework.

Ngày 26/04/2022, React đã cho ra phiên bản mới nhất là phiên bản 18.1.0.

#### Tính khả dụng

* 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.

#### Các đặc trưng của ReactJS

* Component: là thành phần cơ bản của ReactJS. Nó là một đoạn mã ngắn có ý nghĩa thể hiện một chức năng hay đối tượng nhất định, có thể tái sử dụng nhiều lần với các thuộc tính, trạng thái khác nhau. Mọi thứ trong ReactJS đều là “component” và giúp bảo trì code khi làm việc với dự án lớn.
* Virtual DOM: công nghệ DOM ảo giúp tăng hiệu năng cho ứng dụng. Việc chỉ node gốc mới có trạng thái và khi nó thay đổi sẽ tái cấu trúc lại toàn bộ, đồng nghĩa với việc DOM tree cũng sẽ phải thay đổi một phần, điều này sẽ ảnh hưởng đến tốc độ xử lý. Virtual DOM là một object JavaScript, mỗi object chứa đầy đủ thông tin cần thiết để tạo ra một DOM, khi dữ liệu thay đổi nó sẽ tính toán sự thay đổi giữa object và tree thật, điều này sẽ giúp tối ưu hoá việc re-render DOM tree thật.
* JSX: là thành phần quan trọng trong ReactJS. JSX là một dạng ngôn ngữ cho phép viết các mã HTML trong JavaScript. Đây là cú pháp mở rộng JavaScript, là sự kết hợp của XML và JavaScript, giúp chuyển đổi cú pháp dạng XML thành dạng JavaScript, sử dụng cú pháp của XML để code. Đặc điểm khi sử dụng JSX là nhanh hơn, an toàn hơn và dễ sử dụng hơn.
* Props và State: props, viết tắt của properties, là các thuộc tính của một component. Nếu component là một đối tượng thì props là những tính chất, đặc trưng của một đối tượng. Props là một giá trị bất biến. State được dùng để biểu diễn trạng thái của component. Tức là giá trị của state có thể thay đổi được. Mỗi khi state của một component thay đổi thì component đồng thời render lại để cập nhật UI.
* Redux: là một thư viện giúp quản lý các state một cách tốt hơn. Thay vì phải truyền state qua từng component, redux tạo ra một store duy nhất dùng để thay đổi dữ liệu.
* Luồng dữ liệu một chiều (one-way data binding): dữ liệu được truyền từ cha đến con thông qua props. Luồng dữ liệu đơn giản giúp chúng ta dễ dàng kiểm soát cũng như sửa lỗi. Thích hợp xây dựng ứng dụng lớn mà dữ liệu của chúng thay đổi liên tục theo thời gian. Dữ liệu thay đổi thì hầu hết kèm theo sự thay đổi về giao diện.

#### Lợi ích khi sử dụng ReactJS

* Dễ sử dụng: ReactJS hoạt động dựa trên cơ chế kết hợp giữa [HTML](https://bizfly.vn/techblog/html-la-gi.html) và JavaScript nên khá đơn giản và dễ sử dụng. ReactJS là một thư viện GUI mã nguồn mở JavaScript tập trung vào một điều cụ thể để hoàn thành nhiệm vụ UI hiệu quả.
* Hỗ trợ Reusable Component: ReactJS mang lại khả năng tái sử dụng các components đã phát triển thành những ứng dụng khác với chức năng không đổi.
* Viết component dễ dàng hơn: component dễ viết hơn vì nó sử dụng JSX vì JSX hiệu quả trong việc phát triển components đặc biệt hoặc các ứng dụng có khối lượng lớn, giúp làm rõ toàn bộ quá trình viết cấu trúc trang web. Component có thể mở rộng cú pháp tùy chọn cho JavaScript, cho phép kết hợp HTML với JavaScript.
* Hiệu suất tốt hơn với Virtual DOM: React sẽ cập nhật hiệu quả quá trình DOM (Document Object Model – Mô hình đối tượng tài liệu) sử dụng Virtual DOM. React cập nhật quá trình DOM và tự tạo cho nó các Virtual DOM, sau đó đưa chúng vào bộ nhớ. Hoạt động này thúc đẩy việc tự động cập nhật tự động của Virtual DOM ngay khi DOM thực tế có sự thay đổi, giúp tốc độ của các ứng dụng không bị gián đoạn.
* Thân thiện với SEO: ReactJS cho phép bạn tạo giao diện người dùng có thể được truy cập trên các công cụ tìm kiếm khác nhau.

### Cơ sở dữ liệu MySQL

#### MySQL là gì?



Hình .. MySQL logo

MySQL là 1 hệ thống quản trị về cơ sở dữ liệu với mã nguồn mở (được gọi tắt là RDBMS) và đang hoạt động theo mô hình dạng client-server.

Được phát hành chính thức từ thập niên 90s, MySQL hiện đang quản lý dữ liệu qua những cơ sở dữ liệu, với mỗi một cơ sở dữ liệu hoàn toàn có thể có rất nhiều những bản quan hệ có chứa dữ liệu. Ngoài ra, MySQL cũng có cùng 1 cách thức truy xuất cũng như mã lệnh tương tự cùng với ngôn ngữ SQL.

#### Vì sao sử dụng MySQL?

MySQL hiện đang là CSDL có tốc độ ổn định, khá cao và đặc biệt là rất dễ sử dụng, hoạt động được trên nhiều những hệ điều hành.

Không những có thể sử dụng trên nhiều ứng dụng được, tính bảo mật mạnh mà nó còn được sử dụng miễn phí hoàn toàn.

MySQL không những bổ trợ cho Perl và PHP, mà còn bổ trợ cho rất nhiều những ngôn ngữ khác.

#### Các yếu tố giúp MySQL nổi bật

* **Nhanh chóng:**Nhờ vào việc đưa ra một số những tiêu chuẩn và cho phép MySQL làm việc hiệu quả cũng như tiết kiệm chi phí, giúp gia tăng tốc độ thực thi.
* **Mạnh mẽ và khả năng mở rộng:** MySQL hoàn toàn có thể xử lý số lượng lớn dữ liệu và đặc biệt hơn thế nữa thì nó còn có thể mở rộng nếu như cần thiết.
* **Đa tính năng:** Ưu điểm MySQL là gì? MySQL hiện đang hỗ trợ nhiều những chức năng SQL rất được mong chờ từ 1 hệ quản trị CSDL quan hệ cả gián tiếp cũng như trực tiếp.
* **Độ bảo mật cao:** MySQL là gì? Hiện tại nó đang rất thích hợp cho những ứng dụng truy cập CSDL thông qua internet khi sở hữu rất nhiều những tính năng về bảo mật và thậm chí là đang ở cấp cao.
* **Dễ dàng sử dụng:**MySQL đang là cơ sở dữ liệu dễ sử dụng, ổn định, tốc độ cao và hoạt động trên rất nhiều những hệ điều hành đang cung cấp 1 hệ thống lớn những hàm tiện ích rất mạnh.

## Công cụ sử dụng

* Visual Studio Code: phục vụ mục đích code Django và ReactJS.
* MySQL: cơ sở dữ liệu dùng để lưu trữ dữ liệu.
* MySQL Workbench: 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.

# PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## Phân tích hệ thống

### Vai trò của người dùng

* Hệ thống chia làm 3 nhóm người dùng chính:
  + Khách truy cập vãng lai
  + Khách hàng
  + Người quản trị
* Mô tả nhóm người dùng.
* Khách truy cập vãng lai: những người dùng truy cập trang web và chưa có tài khoản đăng nhập trên trang web. Khách truy cập vãng lai có thể xem thông tin sản phẩm, nếu không đăng ký tài khoản của trang web thì người dùng không thể đặt hàng.
* Khách hàng (thành viên của trang web): những khách hàng đã có tài khoản 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 và khi người dùng muốn mua hàng chỉ cần đăng nhập bằng tài khoản thông tin của khách hàng sẽ hiển thị và khách hàng chỉ cần tiến hành mua.
* 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 truy cập vãng lai:
  + Đăng ký tài khoản
  + Xem chi tiết sản phẩm
  + Tìm kiếm sản phẩm
* Thành viên của trang web:
  + Đăng nhập
  + Đăng xuất
  + Xem chi tiết sản phẩm
  + Tìm kiếm sản phẩm
  + Đặt hàng
  + Sửa thông tin cá nhân
  + Quên mật khẩu
* Quản trị viên:
  + Sửa thông tin cá nhân
  + Quản lý tài khoản
  + Quản lý danh mục sản phẩm
  + Quản lý sản phẩm
  + Quản lý hình ảnh sản phẩm
  + Danh sách đơn hàng
  + Xem chi tiết sản phẩm
  + Tìm kiếm sản phẩm

### 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.

## Biểu đồ use case tổng quát

Hình .. Biểu đồ use case tổng quát



## Mô hình thực thể liên kết

Diagram

Description automatically generated

Hình .. Mô hình thực thể liên kết

## Đặc tả use case

### Use case đăng nhập

1. Biểu đồ use case:

Hình .. Biểu đồ use case đăng nhập

1. 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 chủ.
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 ký” 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:

Hình .. Biểu đồ trình tự đăng nhập

### Use case đăng ký

1. Biểu đồ use case:



Hình .. Biểu đồ use case đăng ký

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 rồi kích nút đăng ký. Hệ thống thêm một bản ghi mới vào trong bảng USER 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:

Hình .. Biểu đồ trình tự đăng ký

### Use case tìm kiếm

1. Biểu đồ use case:

Hình .. Biểu đồ use case tìm kiếm

1. 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 sách vào ô tìm kiếm và kích vào biểu tượng kính lúp trên thanh tìm kiếm. Hệ thống sẽ lấy các thông tin về các sách có tiêu đề chứa từ khoá từ bảng BOOK 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:



Hình .. Biểu đồ trình tự tìm kiếm

### Use case xem chi tiết sách

1. Biểu đồ use case:

Hình .. Biểu đồ use case xem chi tiết sách

1. Đặc tả use case:

Use case cho phép khách hàng xem thông tin chi tiết về sách.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi khách hàng kích vào tên sách trong danh sách sách . Hệ thống sẽ lấy các thông tin về sách bao gồm Category\_name từ bảng CATEGORY, Author\_name từ bảng AUTHOR và Book\_name, Unit\_Price, Description, Dimensions , Page, Qty từ bảng BOOK 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:



Hình .. Biểu đồ trình tự xem chi tiết sách

### Use case sửa thông tin cá nhân

1. Biểu đồ use case:

Hình .. Biểu đồ use case sửa thông tin cá nhân

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 nút profile 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:

Hình .. Biểu đồ trình tự sửa thông tin cá nhân

### Use case quên mật khẩu

1. Biểu đồ use case:

Hình .. Biểu đồ use case quên mật khẩu

1. 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 tài khoản rồi kích nút Submit. Hệ thống kiểm tra tên tài khoản trong cơ sở dữ liệu và cập nhật lại mật khẩu mới.
3. Use case kết thúc.
   * Luồng rẽ nhánh:
4. Tại bước 2 ở luồng cơ bản, nếu tên tài khoản 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:

Hình .. Biểu đồ trình tự quên mật khẩu

### Use case đặt hàng

1. Biểu đồ use case:

Hình .. Biểu đồ use case đặt hàng

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 “Thanh toán” tại trang giỏ hàng. Hệ thống hiển thị màn hình thông tin khách hàng gồm FirstName, LastName từ bảng User và lấy thông tin các sách gồm BookName, UnitPrice từ bảng BOOK và Qty từ bảng ORDER hiển thị lên khung “Thông tin đơn hàng”.
2. Khách hàng kích vào nút “Đặt mua”. Hệ thống tạo một đơn hàng mới trong bảng ORDER, tạo các chi tiết đơn hàng mới trong bảng ORDER\_DETAIL và tạo các thông tin vận chuyển của bảng TRANSACTION và thông báo đặt hàng thành công, thông tin đơn hàng vừa đặt.
3. Use case kết thúc.
   * Luồng rẽ nhánh:
4. 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.
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:



Hình .. Biều đồ trình tự đặt hàng

### Use case quản lý tài khoản

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, Email) 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, Username, Password, Confirm Password, IsSuperuser).
   2. Quản trị viên nhập các thông tin (First Name, Last Name, Email, Username, Password, Confirm Password, IsSuperuser) và kích vào nút “Submit”. Hệ thống tạo ra UserID và lưu các thông tin (UserID, First Name, Last Name, Username, Password, Email, IsSuperuser, 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, Email, IsSuperuser, 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, Email, IsSuperuser, Password).
   2. Quản trị viên sửa thông tin (First Name, Last Name, Email, IsSuperuser, Password) 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:



Hình .. Biểu đồ trình tự quản lý tài khoản

### Use case quản lý danh mục 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 (CategoryName) 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 (CategoryName).
   2. Quản trị viên nhập các thông tin (CategoryName) và kích vào nút “Submit”. Hệ thống tạo ra CategoryID từ Name và lưu các thông tin (CategoryID, CategoryName, CreatedAt, UpdatedAt) 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 (CateogryName, 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).
   2. Quản trị viên sửa thông tin (Name) 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:





Hình .. Biều đồ trình tự quản lý danh mục

### Use case quản lý sách

1. Biểu đồ use case:

Hình .. Biểu đồ use case quản lý sách

1. Đặc tả use case:

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

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi quản trị viên kích vào nút “Book management” trên thanh Menu quản trị. Hệ thống lấy tất cả thông tin chi tiết của các sách từ bảng BOOK trong cơ sở dữ liệu và hiển thị lên màn hình.
2. Thêm mới sách:
   1. Quản trị viên kích vào nút “ADD BOOK”. Hệ thống hiển thị “Book Form” yêu cầu quản trị viên nhập các thông tin đầy đủ vào form.
   2. Quản trị viên nhập các thông tin và kích vào nút “Submit”. Hệ thống tạo ra BookID từ BookName và lưu các thông tin vừa nhập vào vào bảng BOOK 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 từ bảng BOOK trong cơ sở dữ liệu và hiển thị lên màn hình.
   2. Quản trị viên kích vào biểu tượng “X” trên cửa sổ “Book 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.
   2. Quản trị viên sửa thông tin của sách và kích vào nút “Submit”. Hệ thống sẽ cập nhật thông tin sách trong bảng BOOK 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 Book” yêu cầu xác nhận xóa.
   2. Quản trị viên kích vào nút “Yes”. Hệ thống sẽ xóa sách khỏi bảng BOOK 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.

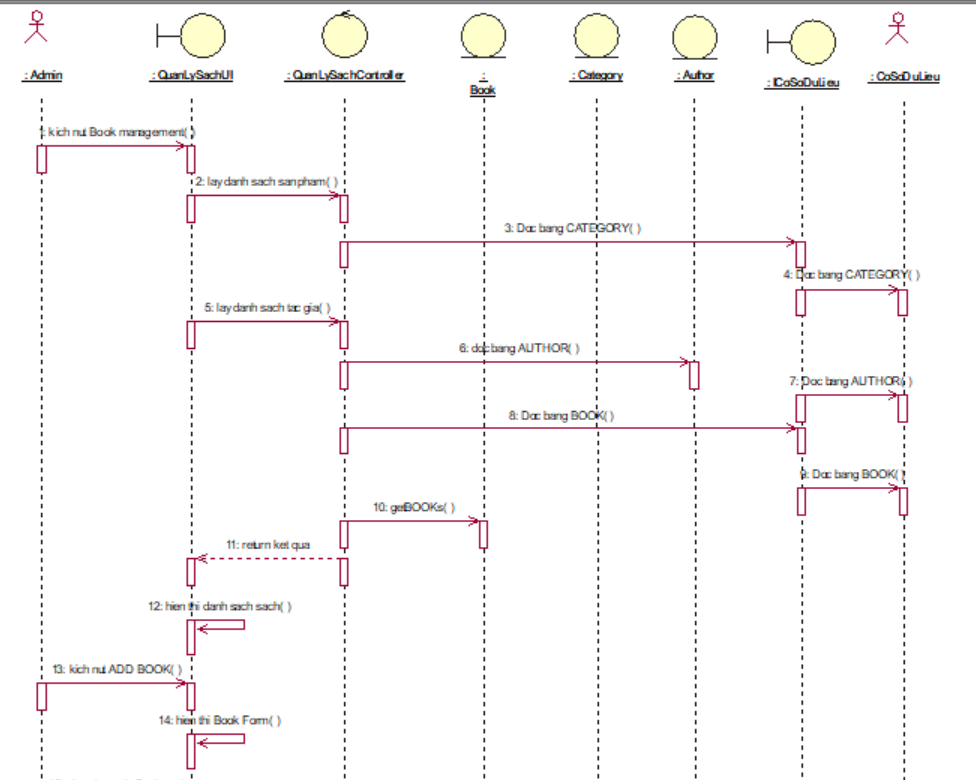
* Hậu điều kiện:

Không có.

* Điểm mở rộng:

Không có.

1. Biểu đồ trình tự:



Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Line chart

Description automatically generated with medium confidence

Hình .. Biểu đồ trình tự quản lý sách

### Use case quản lý tác giả

1. Biểu đồ use case:

Diagram

Description automatically generated

Hình .. Biểu đồ use case quản lý tác giả

1. Đặc tả use case:

Use case này cho phép quản trị viên xem, thêm, sửa thông tin, xóa các tác giả trong bảng AUTHOR.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi quản trị viên kích vào nút “Author management” trên thanh Menu quản trị. Hệ thống lấy thông tin của các Author (AuthorName, Description,Image,CreatedAt, UpdatedAt) từ bảng AUTHOR trong cơ sở dữ liệu và hiển thị lên màn hình.
2. Thêm mới tác giả:
   1. Quản trị viên kích vào nút “ADD AUTHOR”. Hệ thống hiển thị “Author Form” yêu cầu quản trị viên nhập thông tin.
   2. Quản trị viên nhập thông tin và kích vào nút “Submit”. Hệ thống tạo ra AuthorID và lưu các thông tin (AuthorID, AuthorName, Description, Image, CreatedAt, UpdatedAt) vào bảng AUHTOR và hiển thị danh sách thông tin lên màn hình.
3. Xem thông tin tác giả:
   1. Quản trị viên kích vào nút “View” trên dòng thông tin ảnh. Hệ thống lấy thông tin của tác giả từ bảng AUTHOR trong cơ sở dữ liệu và hiển thị thông tin lên màn hình.
   2. Quản trị viên kích vào biểu tượng “X” trên cửa sổ “Author Form”. Hệ thống sẽ hiển thị lại danh sách thông tin ảnh trên màn hình.
4. Sửa thông tin tác giả:
   1. Quản trị viên kích vào nút “Edit” trên dòng thông tin tác giả. Hệ thống hiển thị một màn hình với các thông tin tác giả cũ.
   2. Quản trị viên sửa thông tin của tác giả và kích vào nút “Submit”. Hệ thống sẽ cập nhật thông tin ảnh trong bảng AUTHOR sau đó hiển thị danh sách thông tin tác giả đã cập nhật.
5. Xóa thông tin tác giả:
   1. Quản trị viên kích vào nút “Delete” trên một dòng thông tin tác giả. Hệ thống sẽ hiển thị màn hình “Delete Author” yêu cầu xác nhận xóa.
   2. Quản trị viên kích vào nút “Yes”. Hệ thống sẽ xóa thông tin ảnh khỏi bảng AUTHOR và hiển thị danh sách thông tin ảnh 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 tác giả.

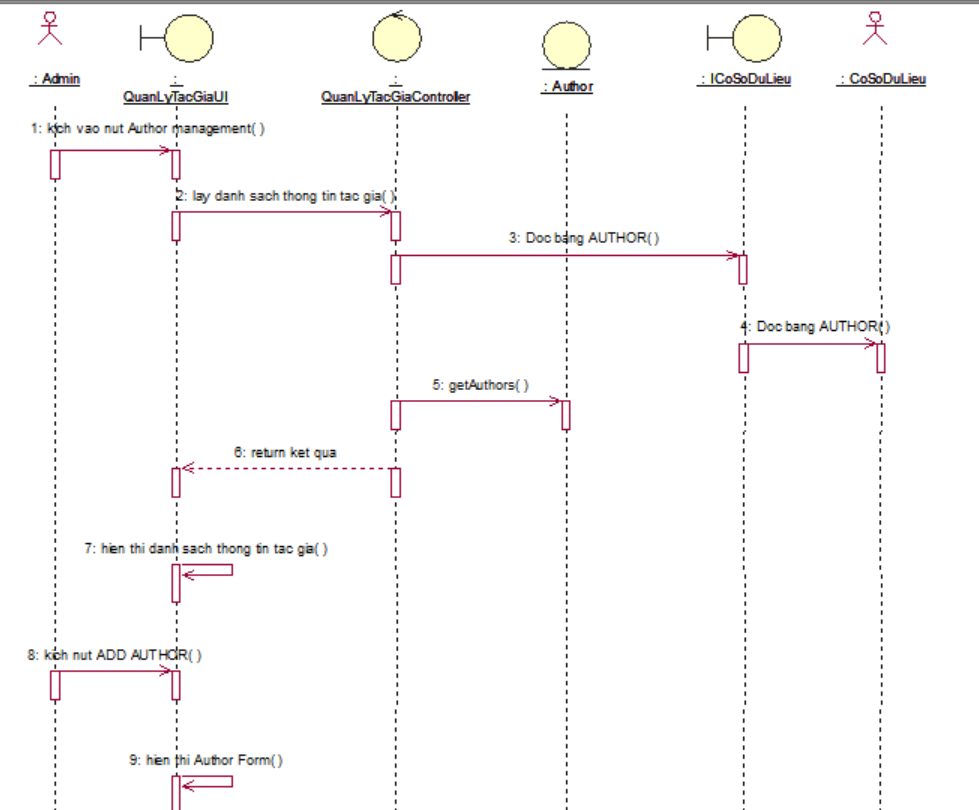
* Hậu điều kiện:

Không có.

* Điểm mở rộng:

Không có.

1. Biểu đồ trình tự:



Diagram

Description automatically generated with low confidence

Graphical user interface

Description automatically generated

Graphical user interface, diagram

Description automatically generated

Graphical user interface

Description automatically generated

Hình .. Biểu đồ trình tự quản lý tác giả

### 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 ORDER và ORDER\_DETAIL.

* 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 từ bảng ORDER và từ bảng TRANSACTION 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 (OrderID, CreatedAt, UpdatedAt, TotalPrice, Note) từ bảng ORDER, (Status, UserID, Phone, Address, amount, message) từ bảng TRANSACTION, (ProductID, Qty) từ bảng ORDER\_DETAIL và (BookName, UnitPrice) từ bảng BOOK 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 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.
   2. Quản trị viên sửa thông tin (Status) của đơn hàng và kích vào nút “Submit”. Hệ thống sẽ cập nhật thông tin đơn hàng trong bảng 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.

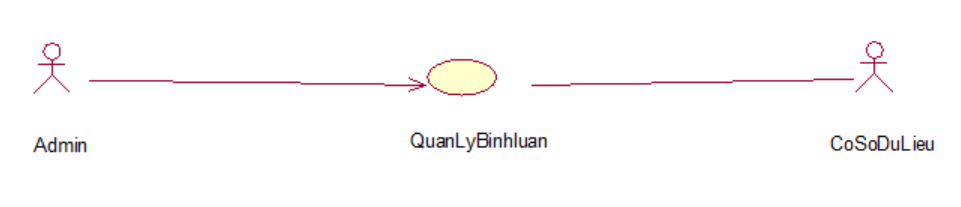
1. Biểu đồ trình tự:



Hình .. Biểu đồ trình tự danh sách đơn hàng

### Use case quản lý bình luận

1. Biểu đồ use case:



Hình .. Biểu đồ use case quản lý bình luận

1. Đặc tả use case:

Use case này cho phép quản trị viên xem, xóa các bình luận trong bảng REVIEW.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi quản trị viên kích vào nút “Review management” trên thanh Menu quản trị. Hệ thống lấy thông tin của các Review (UserID, Star, Comment, Image, BookID, CreatedAt, UpdatedAt) từ bảng REVIEW trong cơ sở dữ liệu và hiển thị lên màn hình.
2. Xem thông tin bình luận:
   1. Quản trị viên kích vào nút “View” trên dòng thông tin ảnh. Hệ thống lấy thông tin của bình luận từ bảng REVIEW và username từ bảng USER và BookName từ bảng BOOK trong cơ sở dữ liệu và hiển thị thông tin lên màn hình.
   2. Quản trị viên kích vào biểu tượng “X” trên cửa sổ “Review Form”. Hệ thống sẽ hiển thị lại danh sách thông tin ảnh trên màn hình.
3. Xóa thông tin bình luận
   1. Quản trị viên kích vào nút “Delete” trên một dòng thông tin bình luận. Hệ thống sẽ hiển thị màn hình “Delete Review” yêu cầu xác nhận xóa.
   2. Quản trị viên kích vào nút “Yes”. Hệ thống sẽ xóa thông tin bình luận khỏi bảng REVIEW và hiển thị danh sách thông tin bình luận 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 bình luận.

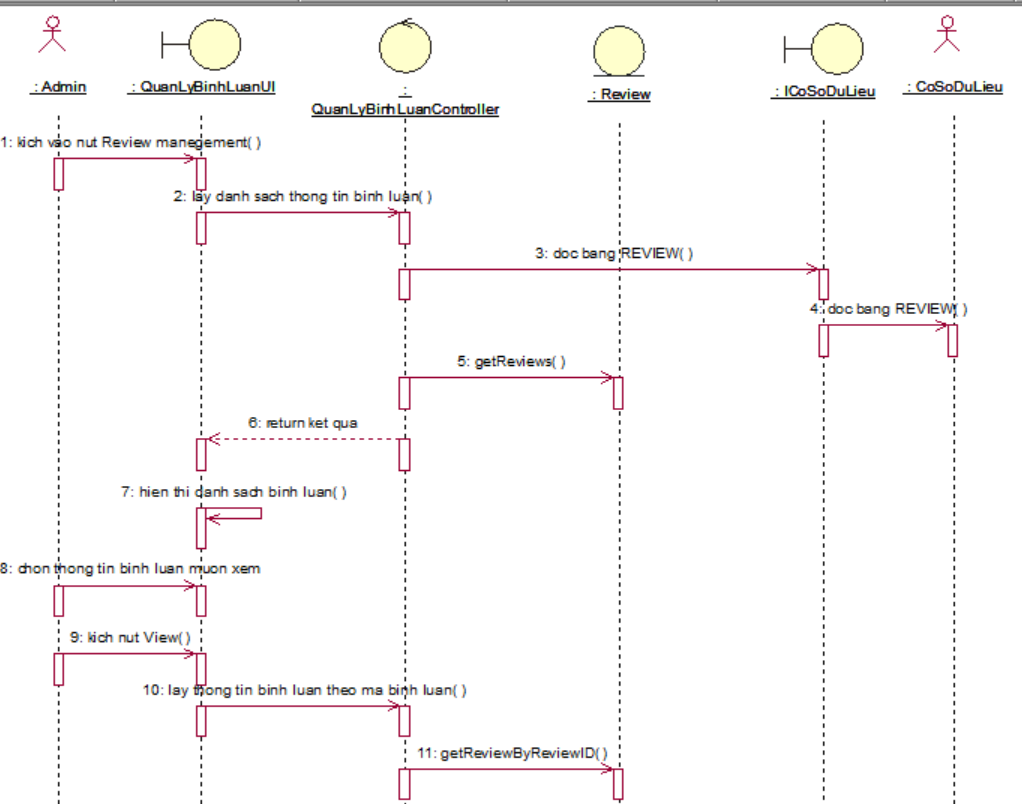
* Hậu điều kiện:

Không có.

* Điểm mở rộng:

Không có.

1. Biểu đồ trình tự



Graphical user interface

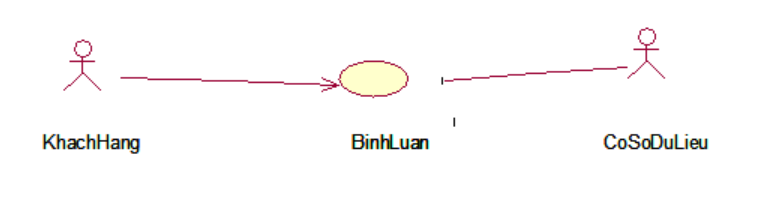
Description automatically generated

Hình .. Biểu đồ trình tự quản lý bình luận

Hình .40.

### Use case bình luận

1. Biểu đồ use case:



Hình .. Biểu đồ use case bình luận

1. Đặc tả use case:

Use case này cho phép người dùng thêm các bình luận về sách

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi quản trị viên kích vào nút “Review” trên trang chi tiết sách. Hệ thống hiển thị “review form” lên màn hình.
2. Xem thông tin bình luận:
   1. Người dùng kích vào nút “Review” trên trang xem chi tiết sách. Hệ thống lấy thông tin của các bình luận từ bảng REVIEW và username từ bảng USER trong cơ sở dữ liệu và hiển thị thông tin lên màn hình.
3. Thêm thông tin bình luận
   1. Người dùng kích vào nút “ADD COMMENT”. Hệ thống hiển thị “Comment Form” yêu cầu người dùng nhập thông tin.
   2. Người dùng nhập thông tin và kích vào nút “Submit”. Hệ thống tạo ra ReviewID và lưu các thông tin (ReviewID, Star, Comment, Image, BookID, CreatedAt, UpdatedAt) vào bảng REVIEW và hiển thị thông tin lên màn hình.
4. Xóa thông tin bình luận
   1. Người dùng kích vào nút “Delete” trên một dòng thông tin bình luận của mình. Hệ thống sẽ hiển thị màn hình “Delete Review” yêu cầu xác nhận xóa.
   2. Người dùng kích vào nút “Yes”. Hệ thống sẽ xóa thông tin bình luận khỏi bảng REVIEW và hiển thị danh sách thông tin bình luận đã cập nhật.
5. Use case kết thúc.
   * Luồng rẽ nhánh:
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 website sẽ hiển thị một thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Người dùng phải đăng nhập vào hệ thống trước thực hiện bình luận.

* Hậu điều kiện:

Không có.

* Điểm mở rộng:

Không có.

Biểu đồ trình tự



Hình .. Biểu đồ trình tự bình luận

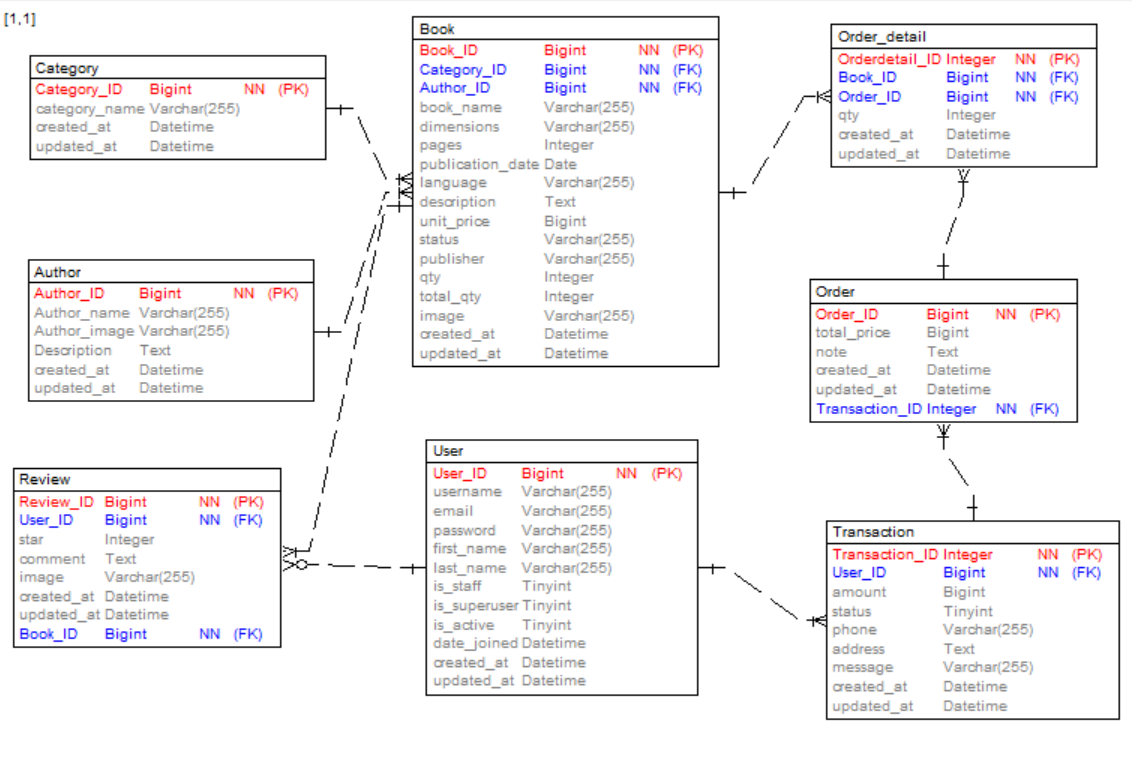
## Thiết kế hệ thống

### **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 sách

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu | Độ dài | Khóa | Giải thích |
| ID | BigInt |  | PK | Id danh mục |
| Category\_name | NVarChar | 255 |  | Tên danh mục |
| CreatedAt | DateTime |  |  | Thời gian tạo |
| UpdatedAt | DateTime |  |  | Thời gian cập nhật |

Bảng .. Bảng Book

Bảng Product để lưu thông tin sách

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu | Độ dài | Khóa | Giải thích |
| ID | BigInt |  | PK | Id sản phẩm |
| Book\_name | NVarChar | 255 |  | Tiêu đề |
| Dimension | NVarChar | 255 |  | Mô tả ngắn |
| Description | Text |  |  | Mô tả |
| Price | BigInt |  |  | Giá |
| Qty | Integer |  |  | Số lượng còn lại |
| Total\_qty | Integer |  |  | Tổng số lượng |
| Page | Int |  |  | Số trang |
| Status | Varchar | 255 |  | Trạng thái |
| CreatedAt | DateTime |  |  | Thời gian tạo |
| UpdatedAt | DateTime |  |  | Thời gian cập nhật |
| Image | Varchar | 255 |  | Hình ảnh |
| Publisher | Varchar | 255 |  | Nhà xuất bản |
| CategoryID | BigInt |  | FK | Id danh mục |
| AuthorID | BigInt |  | FK | Id tác giả |

Bảng .. Bảng Author

Bảng Author để lưu thông tin tác giả

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu | Độ dài | Khóa | Giải thích |
| ID | BigInt |  | PK | Id tác giả |
| Author\_name | VarChar | 255 |  | Họ tên tác giAả |
| Author\_image | VarChar | 255 |  | Ảnh tác giả |
| Description | Text |  |  | Mô tả |
| CreatedAt | DateTime |  |  | Thời gian tạo |
| UpdatedAt | DateTime |  |  | Thời gian cập nhật |

Bảng .. Bảng Review

Bảng Review để lưu thông tin đánh giá của người dùng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu | Độ dài | Khóa | Giải thích |
| ReviewID | BigInt |  | PK | Id đánh giá người dùng |
| Star | VarChar | 255 |  | Số sao đánh giá |
| Comment | VarChar | 255 |  | Bình luận |
| Image | VarChar | 255 |  | Hình ảnh |
| Book\_ID | BigInt |  |  | Mã sách |
| 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 | VarChar | 255 |  | Tên |
| LastName | VarChar | 255 |  | Họ |
| Username | VarChar | 255 |  | Tên tài khoản |
| Address | VarChar | 255 |  | Địa chỉ |
| Password | VarChar | 255 |  | Mật khẩu |
| Is\_staff | TinyInt |  |  | Vai trò |
| Is\_superuser | TinyInt |  |  | Quản trị viên |
| Is\_active | TinyInt |  |  | Hoạt động |
| Email | VarChar | 255 |  | Địa chỉ email |
| Date\_joined | DateTime |  |  | Ngày tham gia |
| CreatedAt | DateTime |  |  | Thời gian tạo |
| UpdatedAt | DateTime |  |  | Thời gian cập nhật |

Bảng .. Bảng Order

Bảng Order để lưu thông tin đơn hàng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu | Độ dài | Khóa | Giải thích |
| OrderID | BigInt |  | PK | Id đơn hàng |
| Total\_price | NVarChar | 255 |  | Địa chỉ nhận hàng |
| Note | VarChar | 255 |  | Số điện thoại |
| TransactionID | BigInt |  | FK | Id trạng thái giao hàng |
| CreatedAt | DateTime |  |  | Thời gian tạo |
| UpdatedAt | DateTime |  |  | Thời gian cập nhật |

Bảng .. Bảng OrderDetail

Bảng OrderDetail để lưu thông tin sách trong đơn hàng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu | Độ dài | Khóa | Giải thích |
| OrderDetailID | BigInt |  | PK | Id chi tiết đơn hàng |
| BookID | BigInt |  | PFK | Id sách |
| OrderID | BigInt |  | PFK | Id đơn hàng |
| Qty | Integer |  |  | Số lượng |
| CreatedAt | DateTime |  |  | Thời gian tạo |
| UpdatedAt | DateTime |  |  | Thời gian cập nhật |

Bảng .. Bảng Transaction

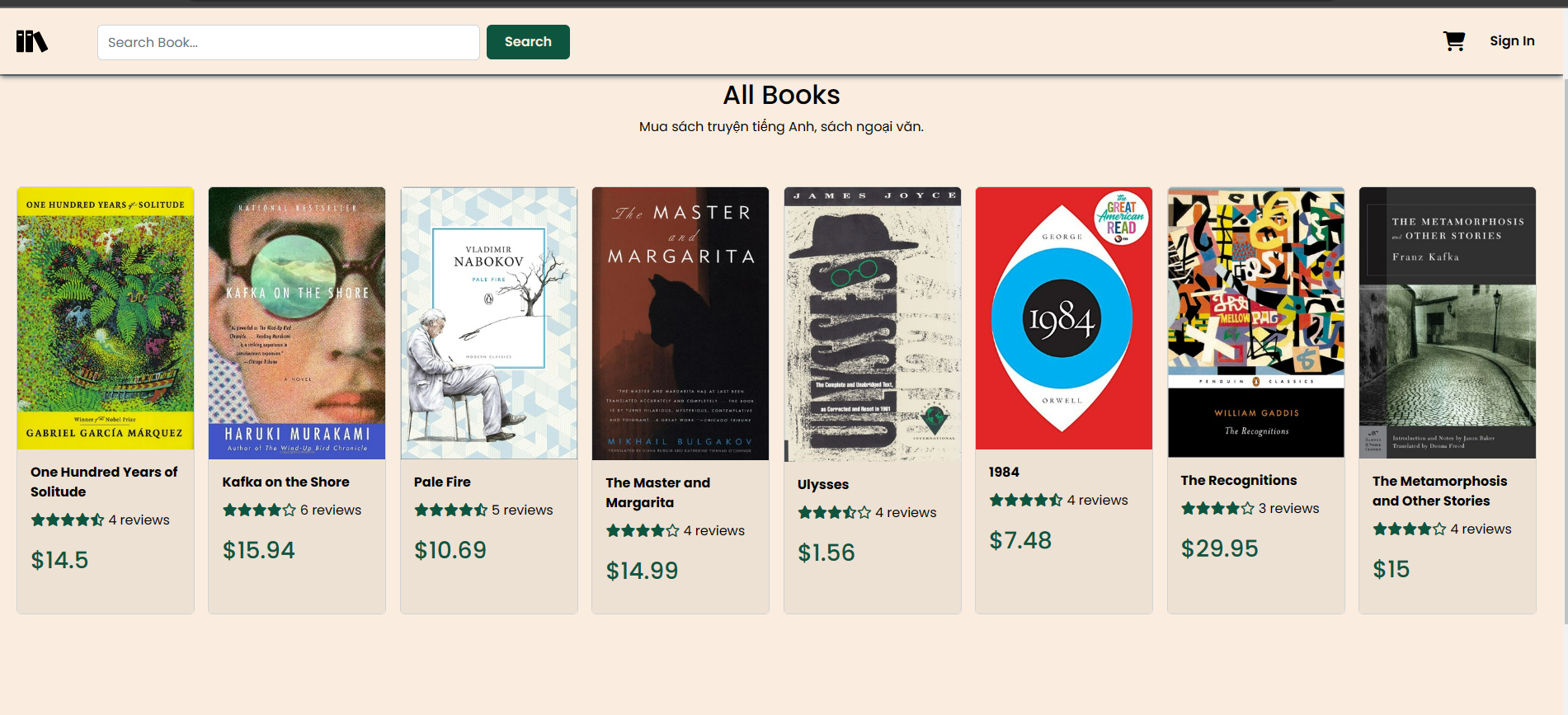
Bảng Transaction để lưu thông tin vận chuyển

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu | Độ dài | Khóa | Giải thích |
| TransactionID | BigInt |  | PK | Id vận chuyển |
| Status | VarChar | 255 |  | Tiêu đề |
| Phone | VarChar | 255 |  | Đường dẫn ảnh |
| Address | Varchar | 255 |  | Kiểu |
| Amount | Int |  |  | Số tiền |
| Message | Varchar | 255 |  | Lời nhắn |
| UserID | BigInt |  | FK | ID Người dùng |
| CreatedAt | DateTime |  |  | Thời gian tạo |
| UpdatedAt | DateTime |  |  | Thời gian cập nhật |

### Thiết kế giao diện

#### Giao diện chung

Giao diện trang chủ được minh họa bằng hình ảnh dưới đây



Hình .. Giao diện trang chủ

Tại màn hình chi tiết sản phẩm, khách hàng có thể chọn số lượng và thêm vào giỏ hàng

Graphical user interface, text, application

Description automatically generated

*Hình 3.34. Giao diện chi tiết sách*

Giao diện trang đăng nhập được minh họa bằng hình ảnh dưới đây

Graphical user interface

Description automatically generated

*Hình 3.35. Giao diện trang đăng nhập*

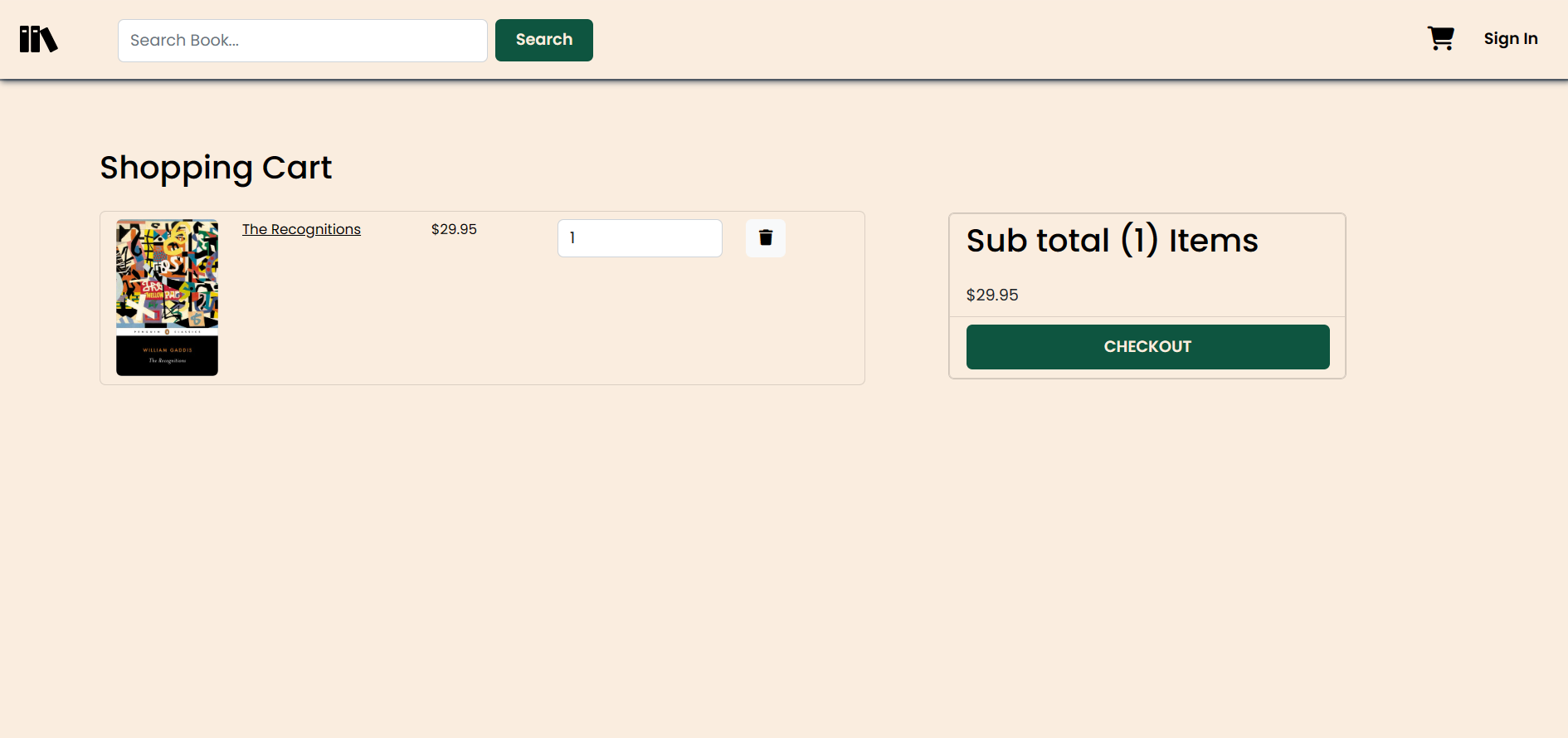
Khách hàng có thể đăng ký tài khoản theo hình minh họa dưới đây

Graphical user interface, application

Description automatically generated

*Hình 3.36. Giao diện trang đăng ký*

Sau khi khách hàng thêm các sản phẩm vào giỏ hàng, khách hàng bấm vào biểu tượng giỏ hàng để xem hoặc thực hiện đặt mua hàng



*Hình 3.37. Giao diện giỏ hàng*

Khách hàng nhập thông tin cần thiết để đặt mua hàng

Graphical user interface, application

Description automatically generated

*Hình 3.38. Giao diện checkout*

Khách hàng chọn phương thức mua hàng là tiền mặt

Graphical user interface, text, application

Description automatically generated

*Hình 3.39. Giao diện phương thức thanh toán*

Màn hình xác nhận mua hàng của khách hàng

Graphical user interface, text, application, email

Description automatically generated

*Hình 3.40. Giao diện xác nhận đặt hàng*

#### Giao diện quản trị

Graphical user interface, application, website

Description automatically generated

*Hình 3.41. Giao diện trang chủ quản trị*

Graphical user interface, application

Description automatically generated

Hình .. Giao diện trang đăng nhập quản trị

# KẾT LUẬN

Đề tài **“Xây dựng Website bán sách sử dụng Django và ReactJS”** 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.
* 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, Python và Django trong lập trình.
* Sử dụng thành thạo các công cụ hỗ trợ lập trình: Visual Studio Code, Postman.

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

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

[5] Trang tài liệu Python <https://www.python.org/>

[6] Trang tài liệu Django <https://www.djangoproject.com/>

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

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

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