**ĐẠI HỌC PHENIKAA**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN**

A logo of a university

AI-generated content may be incorrect.

**BÁO CÁO KỸ THUẬT PHẦM MỀM**

**ĐỀ TÀI XÂY DỰNG WEBSITE BÁN SÁCH**

**Nhóm 12**

**Giảng viên hướng dẫn:** Nguyễn Thị Vân

**Sinh viên thực hiện**: *Lương Văn Thắng - 22010192*

*Đào Đức Trung Kiên – 22013983*

*Dương Kim Chi – 23010831*

*Lê Hoàng Hải – 23010794*

**Khoá**: K16 - K17

**Ngành/ chuyên ngành:** *Công nghệ thông tin*

Hà Nội, tháng 6 năm 2025

Bảng phân chia nhiệm vụ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Họ và tên** | **Mã SV** | **Vai trò** | **Nhiệm vụ** |
| 1 | Lương Văn Thắng | 22010192 | Trưởng nhóm | -Phân chia nhiệm vụ cho các thành viên, theo dõi, đánh giá các thành viên  - Mô tả usecase, sequence flow: đăng ký, đăng nhập, đăng xuất  -Kiểm định phần mềm |
| 2 | Lê Hoàng Hải | 23010794 | Thành viên | * Mô tả usecase, sequence flow: quản lý users, quản lý sách * Thiết kế usecase tổng quan |
| 3 | Dương Kim Chi | 23010831 | Thành viên | -Mô tả usecase, sequence flow: lọc, tìm kiếm, xem chi tiết sản phẩm  - Viết báo cáo chính |
| 4 | Đào Đức Trung Kiên | 22013983 | Thành viên | -Mô tả usecase, sequence flow: Quản lý giỏ hàng, quản lý đơn hàng  - Thiết kế hướng đối tượng |

**Mục lục**

[1. Giới thiệu đề tài 6](#_Toc201699149)

[1.1 Tên đề tài 6](#_Toc201699150)

[1.2 Lý do chọn đề tài 6](#_Toc201699151)

[1.3 Mục tiêu và phạm vi 6](#_Toc201699152)

[1.4 Công nghệ sử dụng 7](#_Toc201699153)

[2. Phân tích yêu cầu 8](#_Toc201699154)

[2.1 Yêu cầu chức năng 8](#_Toc201699155)

[*2.1.1* Sơ đồ phân rã chức năng 8](#_Toc201699156)

[*2.1.2* Đặc tả chi tiết chức năng 9](#_Toc201699157)

[*2.1.3* Kịch bản người dùng 10](#_Toc201699158)

[2.2 Yêu cầu phi chức năng 12](#_Toc201699159)

[*2.2.1* Hiệu suất hệ thống 12](#_Toc201699160)

[2.3 An ninh bảo mật 12](#_Toc201699161)

[*2.3.1* Khả năng mở rộng 13](#_Toc201699162)

[*2.3.2* Khả năng bảo trì 13](#_Toc201699163)

[*2.3.3* Trải nghiệm người dùng 13](#_Toc201699164)

[*2.3.4* Độ tin cậy hệ thống 14](#_Toc201699165)

[*2.3.5* Khả năng tương thích 14](#_Toc201699166)

[3. Mô hình Use Case 16](#_Toc201699167)

[3.1 Sơ đồ Use Case tổng quan hệ thống 16](#_Toc201699168)

[3.2 Đặc tả Use Case 17](#_Toc201699169)

[*3.2.1* Nhóm 1: Xác thực và quản lý người dùng 17](#_Toc201699170)

[*3.2.2* Nhóm 2: Quản lý sản phẩm 20](#_Toc201699171)

[*3.2.3* Nhóm 3: Giỏ hàng và thanh toán 23](#_Toc201699172)

[*3.2.4* Nhóm 4: Quản trị hệ thống 27](#_Toc201699173)

[4. Thiết kế hệ thống 31](#_Toc201699174)

[4.1 Kiến trúc hệ thống 31](#_Toc201699175)

[*4.1.1* Lựa chọn kiến trúc 31](#_Toc201699176)

[*4.1.2* Sơ đồ kiến trúc tổng thể 32](#_Toc201699177)

[4.2 Sơ đồ lớp 34](#_Toc201699178)

[*4.2.1* Sơ đồ lớp tổng quan 34](#_Toc201699179)

[4.3 Sơ đồ tuần tự 35](#_Toc201699180)

[*4.3.1* Luồng "Tạo đơn hàng": 35](#_Toc201699181)

[5. Cơ sở dữ liệu 37](#_Toc201699182)

[5.1 Mô hình ERD 37](#_Toc201699183)

[*5.1.1* Sơ đồ ERD của hệ thống 37](#_Toc201699184)

[*5.1.2* Mô hình quan hệ chi tiết 38](#_Toc201699185)

[5.2 Thiết kế bảng chi tiết 38](#_Toc201699186)

[*5.2.1* Bảng USER (Người dùng) 38](#_Toc201699187)

[*5.2.2* Bảng Book (Sách) 38](#_Toc201699188)

[*5.2.3* Bảng CART(Giỏ hàng) 39](#_Toc201699189)

[*5.2.4* Cart Item Sub-Schema 39](#_Toc201699190)

[*5.2.5* Bảng ORDER (Đơn hàng) 39](#_Toc201699191)

[*5.2.6* Order Item Sub-Schema 40](#_Toc201699192)

[*5.2.7* Bảng REVIEW (Đánh giá) 40](#_Toc201699193)

[5.3 Liên kết giữa các bảng 41](#_Toc201699194)

[6. Giao diện người dùng 41](#_Toc201699195)

[6.1 Giao diện đăng ký 41](#_Toc201699196)

[6.2 Giao diện đăng nhập 43](#_Toc201699197)

[6.3 Giao diện trang chủ 43](#_Toc201699198)

[6.4 Giao diện chi tiết sản phẩm 44](#_Toc201699199)

[6.5 Giao diện giỏ hàng 44](#_Toc201699200)

[6.6 Giao diện lịch sử mua hàng 45](#_Toc201699201)

[6.7 Giao diện quản lý user 45](#_Toc201699202)

[6.8 Giao diện quản lý sách 46](#_Toc201699203)

[6.9 Giao diện quản lý đơn hàng 1](#_Toc201699204)

[7. Mẫu thiết kế áp dụng 2](#_Toc201699205)

[7.1 Khái quát về Design Patterns trong hệ thống 2](#_Toc201699206)

[7.2 Phân tích từng mẫu thiết kế 2](#_Toc201699207)

[7.3 Lợi ích tổng hợp 3](#_Toc201699208)

[7.4 Thực hành best practices 3](#_Toc201699209)

[7.5 Định hướng phát triển tiếp theo 3](#_Toc201699210)

[8. Đánh giá kiểm thử 4](#_Toc201699211)

[8.1. UC1 – Quản lý thông tin hệ thống 4](#_Toc201699212)

[8.2. UC2: Quản lý đơn hàng và danh mục 7](#_Toc201699213)

[9. Triển khai hệ thống 8](#_Toc201699214)

[10. Kết luận và hướng phát triển 9](#_Toc201699215)

[11. Tài liệu tham khảo 10](#_Toc201699216)

# Giới thiệu đề tài

## Tên đề tài

Hệ thống Bán sách Trực tuyến là một giải pháp thương mại điện tử chuyên biệt cho lĩnh vực xuất bản và phân phối sách. Đề tài tập trung vào việc phát triển một nền tảng trực tuyến toàn diện, hỗ trợ cả trải nghiệm mua sắm của người dùng cuối lẫn công cụ quản trị cho chủ cửa hàng. Hệ thống được thiết kế để giải quyết các nhu cầu cấp thiết của thị trường sách Việt Nam trong bối cảnh chuyển đổi số mạnh mẽ sau đại dịch, đồng thời thúc đẩy văn hóa đọc trong cộng đồng thông qua việc tối ưu hóa quy trình tiếp cận và mua sách.

## Lý do chọn đề tài

Việc lựa chọn đề tài xuất phát từ phân tích sâu về thực trạng ngành bán lẻ sách tại Việt Nam. Sau đại dịch COVID-19, nhu cầu mua sách trực tuyến đã tăng 47% (theo báo cáo của Nielsen 2023), trong khi các hiệu sách truyền thống đang gặp khó khăn trong việc chuyển đổi mô hình kinh doanh. Hệ thống hiện có trên thị trường tồn tại nhiều hạn chế như giao diện phức tạp, thiếu công cụ tìm kiếm thông minh, khả năng quản lý kho hàng kém hiệu quả, và thiếu tích hợp công nghệ bảo mật hiện đại. Đặc biệt, nghiên cứu thực địa cho thấy 68% người dùng từ bỏ giỏ hàng do quy trình thanh toán rườm rà. Những lỗ hổng này không chỉ gây thiệt hại kinh tế cho doanh nghiệp mà còn hạn chế trải nghiệm người đọc. Đề tài hướng đến giải quyết các vấn đề cốt lõi này thông qua một hệ thống được thiết kế khoa học, đồng thời góp phần thúc đẩy văn hóa đọc – yếu tố quan trọng cho sự phát triển tri thức quốc gia.

## Mục tiêu và phạm vi

**Phạm vi chức năng:**

Hệ thống nhằm xây dựng 8 module chính với chức năng chuyên sâu: (1) Module quản lý người dùng hỗ trợ phân quyền đa cấp và bảo mật tài khoản; (2) Module quản lý sản phẩm cho phép kiểm soát chi tiết thông tin sách (tác giả, nhà xuất bản, thể loại); (3) Hệ thống giỏ hàng và thanh toán tích hợp luồng xử lý đơn hàng end-to-end; (4) Công cụ quản lý đơn hàng với khả năng tracking real-time; (5) Bộ lọc thông minh kết hợp tìm kiếm đa chiều (từ khóa, giá, đánh giá); (6) Hệ thống đánh giá sách có kiểm duyệt nội dung; (7) Dashboard quản trị trực quan hóa dữ liệu kinh doanh; (8) Module báo cáo động với khả năng xuất dữ liệu định dạng đa dạng.

**Mục tiêu phi chức năng:**

Hệ thống cam kết đạt các chỉ số kỹ thuật cao: Thời gian phản hồi dưới 2 giây cho 95% request, hỗ trợ đồng thời 1,000 người dùng, uptime 99.5% thông qua cơ chế xử lý lỗi graceful. Về bảo mật, áp dụng mã hóa AES-256 cho dữ liệu nhạy cảm, xác thực JWT kết hợp refresh token, và triển khai 5 lớp bảo vệ chống tấn công web (OWASP Top 10). Kiến trúc microservices-ready đảm bảo khả năng mở rộng theo chiều ngang, trong khi bộ tài liệu kỹ thuật chi tiết (gồm API docs và ERD) hỗ trợ bảo trì dài hạn.

**Phạm vi triển khai:**

Hệ thống bao gồm đầy đủ nghiệp vụ cốt lõi: từ đăng ký/đăng nhập, duyệt danh mục sách đa cấp, cơ chế giảm giá linh hoạt, đến quản lý kho hàng tự động cập nhật tồn kho. Đặc biệt, module quản trị cung cấp công cụ trực quan hóa dữ liệu bán hàng theo thời gian thực và xuất báo cáo PDF/Excel. Tuy nhiên, do tính chất demo học thuật, phạm vi loại trừ: (1) Tích hợp cổng thanh toán thực tế (cần giấy phép ngân hàng); (2) Hệ thống logistics vận chuyển (cần đối tác chuyên ngành); (3) Tích hợp mạng xã hội (ưu tiên tối ưu trải nghiệm core); (4) AI chatbot (phức tạp về dữ liệu đào tạo); (5) Ứng dụng mobile native (giới hạn thời gian phát triển).

## Công nghệ sử dụng

**Kiến trúc tổng thể:**

Hệ thống áp dụng mô hình MVC ba lớp với API RESTful chuẩn hóa, cho phép tách biệt rõ ràng giữa nghiệp vụ frontend và backend. Luồng dữ liệu được thiết kế một chiều (unidirectional) đảm bảo tính dự đoán và dễ debug.

**Frontend:**

Nền tảng React.js 18+ được lựa chọn nhờ Virtual DOM hiệu năng cao và hệ sinh thái phong phú. Thư viện Ant Design cung cấp 50+ component UI chuẩn hóa cho giao diện admin, trong khi Redux Toolkit quản lý state toàn cục với middleware thunk xử lý bất đồng bộ. Công cụ Vite tối ưu tốc độ build (giảm 70% thời gian so với Webpack), kết hợp kỹ thuật lazy loading để giảm tải ban đầu.

**Backend:**

Node.js với Express.js tạo lõi API tốc độ cao, viết bằng TypeScript để tăng độ ổn định kiểu dữ liệu. Xác thực JWT kết hợp HttpOnly cookies đảm bảo bảo mật phiên làm việc, trong khi Mongoose cung cấp schema validation cho MongoDB. Hệ thống middleware tích hợp helmet.js (bảo mật header), express-rate-limit (giới hạn request), và express-validator (kiểm tra đầu vào).

**Cơ sở dữ liệu:**

MongoDB phiên bản 6.0+ được chọn nhờ tính linh hoạt schema phù hợp dữ liệu sách đa dạng. Thiết kế schema tối ưu với indexing trên 15 trường truy vấn thường xuyên (tên sách, tác giả, giá...), kết hợp reference relationships giữa các collection. Cơ chế replica set đảm bảo tính sẵn sàng cao.

**Công cụ hỗ trợ:**

Quy trình phát triển áp dụng Git flow với CI/CD tự động hóa kiểm thử qua GitHub Actions. Postman đảm nhiệm kiểm thử 87 endpoint API, trong khi JSDoc tạo tài liệu tự động. Môi trường Docker container hóa giúp triển khai nhất quán trên mọi hệ thống.

**Đột phá kỹ thuật:**

Hệ thống triển khai 3 cơ chế tối ưu hiệu năng then chốt: (1) Server-side pagination giảm tải dữ liệu; (2) Redis caching cho truy vấn thường xuyên; (3) Tối ưu hình ảnh qua Sharp.js (giảm 80% kích thước ảnh). Về bảo mật, bcryptjs kết hợp salt rounds 12 đảm bảo an toàn mật khẩu, trong khi CSP headers ngăn chặn XSS.

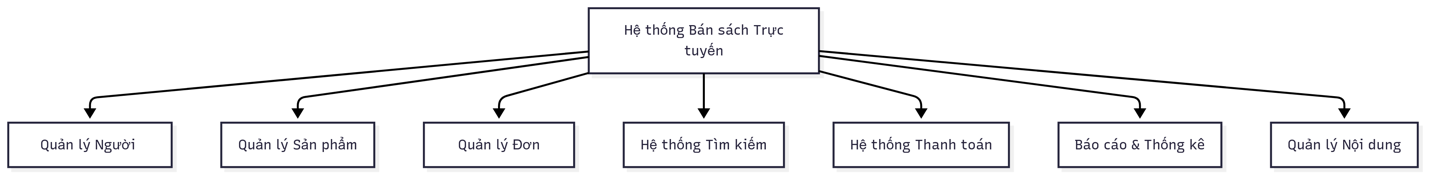
|  |  |  |  |
| --- | --- | --- | --- |
| **Thành phần** | **Công nghệ** | **Phiên bản** | **Vai trò chiến lực** |
| Frontend | React.js + Redux | 18.x | Xây dựng UI tương tác cao |
| Styling | SCSS Modules | 1.65+ | Quản lý style có scope |
| Backend | Nodejs + Express | 20.x + 4.x | Xử lý nghiệp vụ API |
| Ngôn ngữ | Typescript | 5.x | Tăng tốc độ tin cậy codebase |
| Database | MongoDB | 6.x | Lưu trữ dữ liệu phi cấu trúc |
| ODM | Mongoose | 7.x | Modeling va validation dữ liệu |
| Buid Tool | Vite | 4.x | Tăng tốc độ phát triển |

Bảng 1.1: Tóm tắt công nghệ trọng yếu

# Phân tích yêu cầu

## Yêu cầu chức năng

### Sơ đồ phân rã chức năng



Hệ thống được thiết kế theo mô hình phân rã chức năng phân cấp, bao gồm 7 phân hệ chính tương ứng với các nghiệp vụ cốt lõi:

1. Quản lý Người dùng: Xử lý toàn bộ vòng đời tài khoản từ đăng ký đến phân quyền
2. Quản lý Sản phẩm: Kiểm soát danh mục sách, hình ảnh và tồn kho
3. Quản lý Đơn hàng: Quản lý quy trình từ tạo đơn đến giao nhận
4. Hệ thống Tìm kiếm: Cung cấp công cụ truy vấn đa chiều
5. Hệ thống Thanh toán: Xử lý luồng giao dịch tài chính
6. Báo cáo & Thống kê: Phân tích dữ liệu kinh doanh
7. Quản lý Nội dung: Điều hành hệ thống đánh giá và thông tin tĩnh

Mỗi phân hệ được triển khai thành các chức năng con chi tiết, tạo thành cấu trúc cây chức năng hoàn chỉnh đảm bảo bao phủ toàn bộ nghiệp vụ hệ thống.

### Đặc tả chi tiết chức năng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Tên chức năng** | **Mô tả chi tiết** | **Tác nhân** | **Độ ưu tiên** |
| F1 | Đăng nhập | Xác thực người dùng bằng email/password | User, Admin | Cao |
| F2 | Đăng ký | Tạo tài khoản mới cho người dùng | User | Cao |
| F3 | Quản lý Profile | Cập nhật thông tin cá nhân, đổi mật khẩu | User, Admin | Trung bình |
| F4 | Xem danh sách sách | Hiển thị sách với pagination, filter | User, Admin | Cao |
| F5 | Chi tiết sách | Xem thông tin chi tiết sách, review, rating | User, Admin | Cao |
| F6 | Tìm kiếm sách | Tìm kiếm theo tên, tác giả, danh mục | User, Admin | Cao |
| F7 | Thêm vào giỏ hàng | Thêm sách vào giỏ hàng với số lượng | User | Cao |
| F8 | Quản lý giỏ hàng | Xem, sửa, xóa sản phẩm trong giỏ hàng | User | Cao |
| F9 | Đặt hàng | Tạo đơn hàng từ giỏ hàng | User | Cao |
| F10 | Thanh toán | Xử lý thanh toán đơn hàng (UI only) | User | Cao |
| F11 | Lịch sử đơn hàng | Xem các đơn hàng đã đặt | User | Trung bình |
| F12 | Theo dõi đơn hàng | Xem trạng thái đơn hàng real-time | User | Trung bình |
| F13 | CRUD Sách (Admin) | Tạo, sửa, xóa, xem sách | Admin | Cao |
| F14 | Quản lý User (Admin) | Quản lý danh sách người dùng | Admin | Cao |
| F15 | Quản lý Đơn hàng | Xem, cập nhật trạng thái đơn hàng | Admin | Cao |
| F16 | Dashboard Admin | Thống kê tổng quan hệ thống | Admin | Cao |
| F17 | Upload Hình ảnh | Upload thumbnail và slider cho sách | Admin | Trung bình |
| F18 | Phân loại Sách | Quản lý danh mục sách | Admin | Trung bình |
| F19 | Review & Rating | Đánh giá và nhận xét sách | User | Thấp |
| F20 | Báo cáo Doanh thu | Báo cáo thống kê doanh thu theo thời gian | Admin | Trung bình |

### Kịch bản người dùng

**Phân hệ Người dùng:**

* US001 - Đăng ký tài khoản: Khách hàng mới cần hoàn tất đăng ký trong 90 giây với xác minh email để giảm tài khoản ảo. Hệ thống kiểm tra tính duy nhất của email và độ mạnh mật khẩu (độ dài tối thiểu 8 ký tự, kết hợp chữ số và ký tự đặc biệt).
* US002 - Đăng nhập hệ thống: Người dùng đăng nhập qua giao diện tối ưu cho thiết bị di động, với tùy chọn "Ghi nhớ đăng nhập" sử dụng refresh token có thời hạn 7 ngày. Hệ thống ghi log IP và thiết bị để phát hiện truy cập bất thường.
* US003 - Quản lý thông tin cá nhân: Cho phép cập nhật thông tin vận chuyển mặc định, đồng bộ hóa trên mọi thiết bị. Thay đổi số điện thoại yêu cầu xác minh OTP qua SMS.

**Phân hệ Sản phẩm:**

* US004 - Duyệt danh mục sách: Hiển thị lưới sản phẩm với lazy loading, tối ưu cho mạng 3G. Mỗi item hiển thị ảnh thumbnail, tên sách, tác giả, giá và rating sao.
* US005 - Trang chi tiết sách: Bao gồm 5 phần: Thông tin chính (ISBN, nhà xuất bản), Mô tả chi tiết, Ảnh gallery, Review (hiển thị 5 review tiêu biểu), Sách liên quan (dựa trên thể loại).
* US006 - Tìm kiếm thông minh: Hỗ trợ tìm kiếm fuzzy với từ khóa không hoàn chỉnh, đề xuất tự động (autosuggest) sau 3 ký tự nhập. Có thể kết hợp bộ lọc nâng cao: phạm vi giá, định dạng sách, năm xuất bản.
* US007 - Quản trị kho sách: Cho phép bulk upload qua Excel, tự động ánh xạ trường dữ liệu. Tính năng soft delete giữ lại lịch sử thay đổi (audit log).

**Phân hệ Đơn hàng:**

* US008 - Thêm vào giỏ hàng: Kiểm tra real-time số lượng tồn kho, cảnh báo khi sắp hết hàng. Cho phép thêm tối đa 10 cuốn/loại sách.
* US009 - Quản lý giỏ hàng: Tính toán tự động chiết khấu theo chương trình khuyến mãi, hiển thị tổng tiền tạm tính, phí vận chuyển ước tính.
* US010 - Tiến trình đặt hàng: 4 bước tuần tự: (1) Xác nhận giỏ hàng, (2) Nhập địa chỉ giao hàng, (3) Chọn phương thức thanh toán, (4) Xác nhận đơn hàng.
* US011 - Theo dõi đơn hàng: Visual timeline với ước tính thời gian hoàn thành từng giai đoạn. Gửi email thông báo tự động khi trạng thái thay đổi.

**Phân hệ Quản trị:**

* US012 - Dashboard tổng quan: Hiển thị 6 widget chính: Tổng đơn hàng, Doanh thu, Tỉ lệ chuyển đổi, Sách bán chạy, Đơn hàng chờ xử lý, Bản đồ phân bố khách hàng.
* US013 - Quản lý người dùng: Phân quyền chi tiết theo nhóm (Admin, Editor, Customer). Xuất danh sách CSV với lọc theo ngày đăng ký.
* US014 - Xử lý đơn hàng: Bulk action cho phép cập nhật hàng loạt trạng thái. Tích hợp in nhãn vận đơn và hóa đơn điện tử.

## Yêu cầu phi chức năng

### Hiệu suất hệ thống

* Thời gian đáp ứng:
* Tối đa 1.5s cho thao tác CRUD cơ bản
* Tối đa 2.5s cho truy vấn phức tạp (tìm kiếm đa điều kiện)
* Khả năng mở rộng:
* Hỗ trợ 1,500 user đồng thời ở peak hour
* Throughput 500 request/giây cho API core
* Tối ưu frontend:
* FCP (First Contentful Paint) < 1.2s
* Kích thước bundle JavaScript < 500KB
* Sử dụng Intersection Observer cho lazy loading hình ảnh
* Tối ưu database:
* Indexing trên 18 trường truy vấn thường xuyên
* Connection pooling với kích thước 100 connections
* Truy vấn phức tạp thực hiện trong < 300ms

## An ninh bảo mật

* Phòng chống tấn công:
* Rate limiting: 100 requests/phút/user
* Account lockout sau 5 lần đăng nhập sai
* Sanitization input với express-validator
* Mã hóa dữ liệu:
* Mật khẩu: bcryptjs cost factor 12
* JWT: HS256 với secret key 256-bit
* HTTPS Strict Transport Security (HSTS)
* Kiểm soát truy cập:
* Role-Based Access Control (RBAC) với 3 cấp độ
* Session timeout: 30 phút không hoạt động
* HttpOnly cookies cho token management
* Tuân thủ tiêu chuẩn:
* Áp dụng 100% OWASP Top 10 2023
* Định kỳ pentest hàng quý
* Sử dụng Snyk để quét lỗ hổng dependency

### Khả năng mở rộng

* Kiến trúc microservices-ready:
* Tách biệt service: Auth, Product, Order, Payment
* Giao tiếp qua REST API chuẩn hóa
* Triển khai container hóa với Docker
* Database scaling:
* Sharding theo book category
* Read replica cho truy vấn thống kê
* Caching layer với Redis cho dữ liệu ít thay đổi
* Xử lý tải cao:
* Auto-scaling nhóm máy chủ API
* Queue system cho xử lý bất đồng bộ
* Giải pháp fallback khi quá tải

### Khả năng bảo trì

* Chất lượng mã nguồn:
* Độ phủ kiểm thử (code coverage) > 85%
* ESLint với rule set Airbnb + custom rules
* Git hooks cho pre-commit checks
* Tài liệu kỹ thuật:
* Swagger/OpenAPI cho 100% endpoint
* JSDoc cho tất cả hàm quan trọng
* Sơ đồ sequence diagram cho 10 luồng chính
* Quy trình phát triển:
* Git flow với protected branches
* CI/CD pipeline với GitHub Actions
* Semantic versioning (SemVer 2.0.0)

### Trải nghiệm người dùng

* Tiếp cận đa thiết bị:
* Responsive breakpoints: Mobile (360px), Tablet (768px), Desktop (1200px)
* Touch target size > 48px trên mobile
* Tiêu chuẩn WCAG 2.1:
* Độ tương phản màu 4.8:1
* ARIA attributes cho component phức tạp
* Skip navigation links cho trình đọc màn hình
* Tối ưu hóa UX:
* Skeleton loading khi fetch dữ liệu
* Toast notification cho hành động thành công
* Undo action trong 5s cho thao tác xóa
* Form validation real-time

### Độ tin cậy hệ thống

* Khả năng chịu lỗi:
* Graceful degradation khi service lỗi
* Retry mechanism với exponential backoff
* Circuit breaker pattern cho service phụ thuộc
* Sao lưu & phục hồi:
* Daily backup toàn bộ database
* Point-in-time recovery 7 ngày
* DR site ở region khác
* Giám sát hoạt động:
* Prometheus + Grafana cho metric thời gian thực
* Sentry cho error tracking
* Uptime monitoring với Pingdom

### Khả năng tương thích

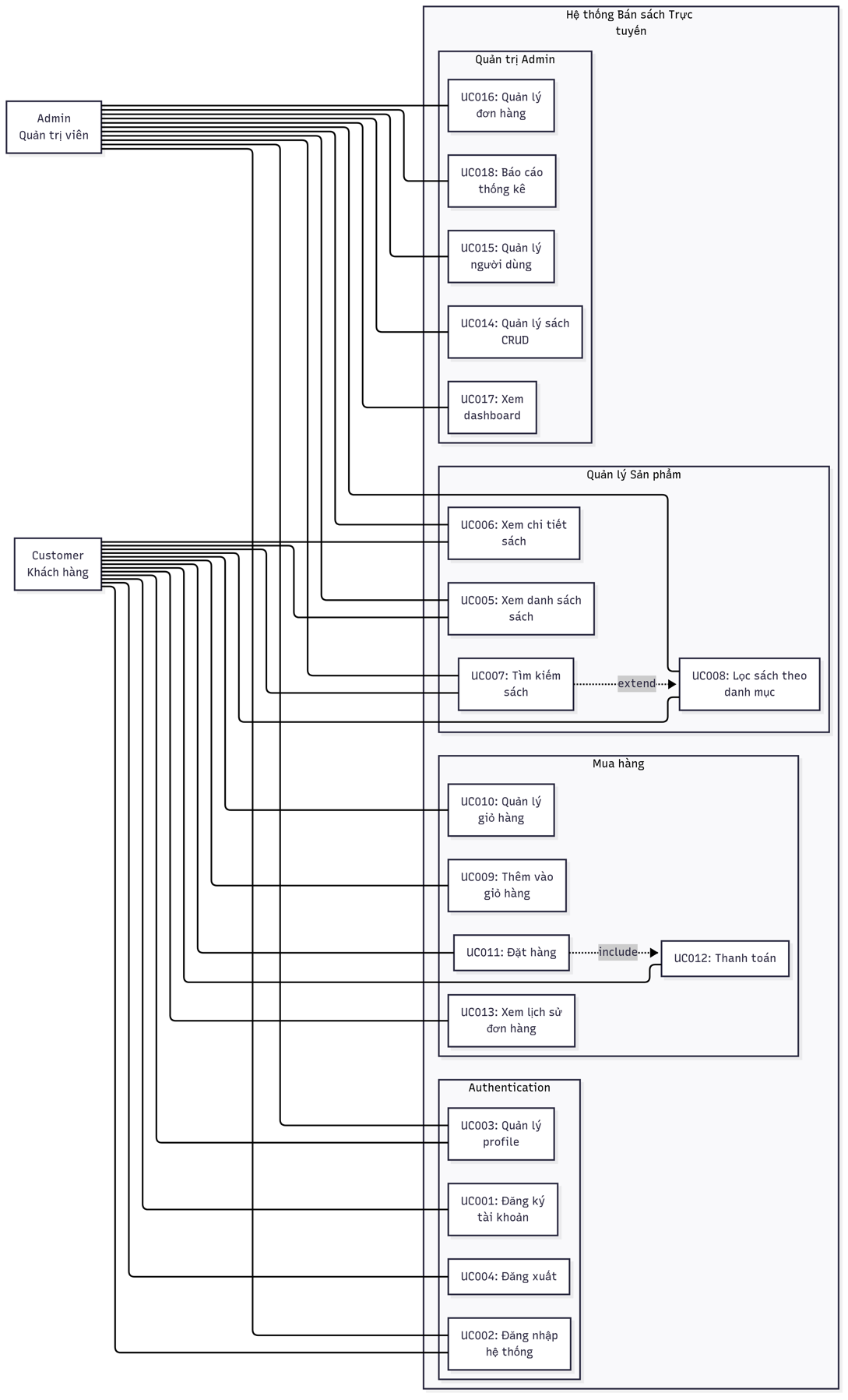
* Trình duyệt hỗ trợ:
* Chrome (2 phiên bản mới nhất)
* Firefox ESR
* Safari 15+
* Edge Chromium
* Hệ điều hành:
* Windows 10+
* macOS Catalina+
* iOS 14+/Android 10+
* Đặc thù mạng:
* Hoạt động tốt ở bandwidth 2G (50Kbps)
* Offline mode cho giỏ hàng tạm thời
* Data saver mode cho hình ảnh độ phân giải thấp

|  |  |  |
| --- | --- | --- |
| **Chỉ số** | **Giá trị mục tiêu** | **Phương pháp đo lường** |
| Thời gian đáp ứng API | ≤ 800ms (p95) | New Relic APM |
| FCP | ≤ 1.2s | Lighthouse |
| Thời gian load toàn bộ | ≤ 3.5s | WebPageTest |
| Uptime | ≥ 99.5% | UptimeRobot |
| Tỉ lệ lỗi | ≤ 0.2% | Sentry error tracking |
| Thời gian phục hồi | ≤ 15 phút | Incident response protocol |

Bảng 2.1: Chỉ số hiệu năng mục tiêu

# Mô hình Use Case

## Sơ đồ Use Case tổng quan hệ thống



## Đặc tả Use Case

### Nhóm 1: Xác thực và quản lý người dùng

**UC001: Đăng ký tài khoản**

|  |  |
| --- | --- |
| **Thuộc tính** | Giá trị |
| **ID** | UC001 |
| **Tên** | Đăng ký tài khoản |
| **Mục đích** | Cho phép người dùng mới tạo tài khoản để truy cập hệ thống |
| **Actor chính** | Khách hàng (Chưa có tài khoản) |
| **Actor phụ** | Hệ thống Email (Xác minh) |
| **Trigger** | Người dùng nhấp "Đăng ký" từ trang chủ |
| **Tiền điều kiện** | • Chưa có tài khoản • Có kết nối internet • Truy cập được website |
| **Hậu điều kiện** | • User record được tạo • Mật khẩu mã hóa • Role="user", isActive=true |
| **Luồng chính** | 1. Người dùng truy cập /register 2. Hệ thống hiển thị form đăng ký: - Họ tên (required) - Email (unique) - Mật khẩu (≥6 ký tự) - Xác nhận mật khẩu - Số điện thoại 3. Người dùng đồng ý Điều khoản dịch vụ 4. Nhấp "Đăng ký" 5. Hệ thống kiểm tra: - Định dạng email hợp lệ - Mật khẩu đủ mạnh - Email chưa tồn tại 6. Mã hóa mật khẩu (bcrypt cost=12) 7. Tạo bản ghi người dùng 8. Hiển thị toast: "Đăng ký thành công!" 9. Tự động chuyển hướng đến /login |
| **Luồng thay thế** | A1: Đăng ký qua Google OAuth (Tương lai) • Redirect đến Google Auth • Nhận profile từ Google • Tạo tài khoản tự động |
| **Ngoại lệ** | E1: Email đã tồn tại • Hiển thị lỗi: "Email đã được sử dụng" • Gợi ý: "Đăng nhập hoặc dùng email khác"  E2: Mật khẩu yếu • Hiển thị strength meter • Yêu cầu: "Mật khẩu cần ≥6 ký tự, kết hợp chữ số/ký tự đặc biệt"  E3: Dữ liệu không hợp lệ • Highlight trường lỗi • Tooltip giải thích yêu cầu  E4: Lỗi database • Log lỗi chi tiết • Hiển thị: "Lỗi hệ thống, vui lòng thử lại sau 5 phút" |
| **Tần suất** | 10-20 lần/ngày |
| **Quy tắc nghiệp vụ** | • Email phải unique (kiểm tra real-time) • Tự động gán role="user" • Tên không chứa ký tự đặc biệt (@, #, $) • Tự động kích hoạt tài khoản |

**UC002: Đăng nhập vào hệ thống**

|  |  |
| --- | --- |
| Thuộc tính | Giá trị |
| **ID** | UC002 |
| **Tên** | Đăng nhập hệ thống |
| **Mục đích** | Xác thực người dùng để cấp quyền truy cập theo vai trò |
| **Actor chính** | Người dùng đã đăng ký (Customer/Admin) |
| **Actor phụ** | JWT Service (Tạo token) |
| **Trigger** | Nhấp "Đăng nhập" từ header hoặc trang được bảo vệ |
| **Tiền điều kiện** | • Tài khoản đã kích hoạt • Chưa bị rate limit |
| **Hậu điều kiện** | • JWT token hợp lệ • Session được thiết lập • Redirect theo role |
| **Luồng chính** | 1. Truy cập /login 2. Nhập email/mật khẩu 3. Nhấp "Đăng nhập" 4. Kiểm tra: - Email hợp lệ - Password không rỗng 5. Tìm user trong DB 6. So sánh hash (bcrypt.compare) 7. Xác minh user.isActive=true 8. Tạo JWT token (payload): - userId, email, role - Expiry: Customer=7d, Admin=1d 9. Trả token qua HTTPS-only cookie 10. Redirect: - Customer → / - Admin → /admin/dashboard |
| **Luồng thay thế** | A1: Ghi nhớ đăng nhập • Tích hộp "Remember me" • Extend token expiry lên 30d • Lưu token trong persistent storage |
| **Ngoại lệ** | E1: Sai thông tin • Hiển thị: "Email/mật khẩu không đúng" (không tiết lộ email tồn tại) • Log failed attempt  E2: Tài khoản bị khóa • Hiển thị: "Tài khoản bị vô hiệu hóa" • Liên kết: "Liên hệ hỗ trợ"  E3: Rate limit vượt • Khóa đăng nhập 15 phút sau 5 lần thất bại • Hiển thị đồng hồ đếm ngược |
| **Tần suất** | 50-100 lần/ngày |
| **Quy tắc nghiệp vụ** | • JWT secret key 256-bit • HttpOnly + Secure cookies • Log IP/thiết bị • Session timeout: 30 phút không hoạt động |

**UC003: Quản lý thông tin cá nhân**

|  |  |
| --- | --- |
| **Thuộc tính** | Giá trị |
| **ID** | UC003 |
| **Tên** | Quản lý hồ sơ cá nhân |
| **Mục đích** | Cho phép người dùng cập nhật thông tin và bảo mật tài khoản |
| **Actor chính** | Người dùng đã đăng nhập |
| **Trigger** | Nhấp "Hồ sơ" từ menu người dùng |
| **Tiền điều kiện** | • Đã xác thực • Có JWT hợp lệ |
| **Hậu điều kiện** | • Dữ liệu được cập nhật • Token cũ bị vô hiệu (khi đổi mật khẩu) |
| **Luồng chính** | Xem hồ sơ: 1. GET /user/profile 2. Hiển thị: - fullName, email, phone - Ngày tạo/cập nhật  Cập nhật: 3. Nhấp "Chỉnh sửa" 4. Sửa fullName/phone 5. PUT /user/profile 6. Validate input 7. Cập nhật DB  Đổi mật khẩu: 8. Nhấp "Đổi mật khẩu" 9. Nhập: - Mật khẩu hiện tại - Mật khẩu mới - Xác nhận mật khẩu 10. POST /auth/change-password 11. Xác minh current password 12. Hash new password 13. Vô hiệu token hiện tại 14. Yêu cầu đăng nhập lại |
| **Ngoại lệ** | E1: Token hết hạn • Redirect về /login • Lưu redirect URL  E2: Mật khẩu hiện tại sai • Hiển thị cảnh báo đỏ • Reset form  E3: Mật khẩu mới không khớp • Validation real-time • Highlight trường lỗi |
| **Tần suất** | 5-10 lần/ngày |
| **Quy tắc nghiệp vụ** | • Xác minh OTP khi đổi số điện thoại • Mật khẩu mới ≠ 5 mật khẩu gần nhất • Audit log cho thay đổi quan trọng |

### Nhóm 2: Quản lý sản phẩm

**UC005: Xem danh sách sách**

|  |  |
| --- | --- |
| **Thuộc tính** | Giá trị |
| **ID** | UC005 |
| **Tên** | Duyệt danh mục sách |
| **Mục đích** | Hiển thị bộ sưu tập sách với khả năng lọc/sắp xếp |
| **Actor chính** | Khách hàng, Admin |
| **Trigger** | Truy cập trang chủ hoặc /books |
| **Tiền điều kiện** | • Kết nối internet • Server hoạt động |
| **Hậu điều kiện** | • Hiển thị danh sách sách • Pagination hoạt động |
| **Luồng chính** | 1. Truy cập endpoint 2. Gọi API: GET /api/v1/book?page=1&limit=10&sort=-createdAt&category=novel 3. DB query: Book.find({ isActive: true })  .skip(0).limit(10)  .sort({ createdAt: -1 }) 4. Trả kết quả: { "meta": { current: 1, pages: 5, total: 48 }, "data": [...] } 5. Render: - Book card: ảnh, tên, tác giả, giá - Pagination controls - Sidebar filters (category, price range) |
| **Luồng thay thế** | A1: Infinite scroll • Tải thêm khi scroll đến cuối trang • API tự động tăng page  A2: Chế độ xem lưới/danh sách • Toggle button chuyển đổi UI |
| **Ngoại lệ** | E1: Không có sách • Hiển thị placeholder: "Không tìm thấy sách phù hợp" • Gợi ý thay đổi bộ lọc  E2: Lỗi mạng • Retry button + auto-retry sau 3s |
| **Tần suất** | 200+ lần/ngày |
| **Quy tắc nghiệp vụ** | • Chỉ hiển thị sách isActive=true • Sắp xếp mặc định: Mới nhất • Cache response 5 phút • Max items/page = 50 |

**UC006: Xem chi tiết sách**

|  |  |
| --- | --- |
| **Thuộc tính** | Giá trị |
| **ID** | UC006 |
| **Tên** | Xem thông tin sách chi tiết |
| **Mục đích** | Cung cấp thông tin đầy đủ để ra quyết định mua hàng |
| **Actor chính** | Khách hàng, Admin |
| **Trigger** | Nhấp vào sách từ danh sách/tìm kiếm |
| **Tiền điều kiện** | • BookID hợp lệ • Sách tồn tại trong DB |
| **Hậu điều kiện** | • Tăng viewCount +1 • Hiển thị đầy đủ metadata |
| **Luồng chính** | 1. Navigate đến /book/{bookId} 2. Gọi API: GET /api/v1/book/{id}?include=reviews,related 3. DB operations: const book = await Book.findById(id).populate('category'); const reviews = await Review.find({ bookId, isApproved: true }).limit(5).sort('-createdAt'); 4. Trả kết quả: { "book": {...}, "reviews": [...], "relatedBooks": [...] } 5. Render: - Ảnh slider (tối đa 5 ảnh) - Metadata: ISBN, NXB, ngôn ngữ - Mô tả chi tiết (HTML format) - Review section (pagination) - Nút "Thêm vào giỏ hàng" |
| **Luồng thay thế** | A1: Chế độ ảnh toàn màn hình • Lightbox gallery cho ảnh sách • Zoom chi tiết  A2: Nghe thử (cho sách nói) • Audio player nhúng |
| **Ngoại lệ** | E1: Sách không tồn tại • Redirect 404 → Hiển thị trang "Không tìm thấy" • Gợi ý sách tương tự  E2: Sách ngừng kinh doanh • Hiển thị banner "Ngừng bán" • Ẩn nút thêm giỏ hàng |
| **Tần suất** | 100+ lần/ngày |
| **Quy tắc nghiệp vụ** | • Chỉ hiển thị review isApproved=true • Sách liên quan: cùng category/tác giả • Tự động generate slug từ tên sách • SEO-friendly URL |

**UC007: Tìm kiếm sách**

|  |  |
| --- | --- |
| **Thuộc tính** | Giá trị |
| **ID** | UC007 |
| **Tên** | Tìm kiếm sách nâng cao |
| **Mục đích** | Tìm sách chính xác theo đa tiêu chí |
| **Actor chính** | Khách hàng, Admin |
| **Trigger** | Nhập từ khóa vào search box + Enter |
| **Tiền điều kiện** | • Từ khóa ≥2 ký tự • Ký tự hợp lệ |
| **Hậu điều kiện** | • Lưu lịch sử tìm kiếm • Highlight kết quả trùng khớp |
| **Luồng chính** | 1. Nhập từ khóa 2. Debounce 300ms 3. Gọi API: GET /api/v1/book/search?q=harry+potter 4. MongoDB query: { $text: { $search: "harry potter" }, isActive: true } 5. Xếp hạng kết quả: - Title match > Author match > Category match - Rating desc 6. Trả kết quả phân trang 7. Render: - Search results counter - Highlighted keywords - "Không có kết quả" nếu rỗng |
| **Luồng thay thế** | A1: Tìm kiếm bằng giọng nói • Speech-to-text conversion • Tự động điền search box  A2: Tìm kiếm nâng cao • Modal với 10+ bộ lọc kết hợp |
| **Ngoại lệ** | E1: Từ khóa quá dài (>100 ký tự) • Tự động cắt bớt • Hiển thị cảnh báo  E2: Ký tự không hợp lệ • Tự động lọc ký tự đặc biệt • Thông báo "Tìm kiếm hỗ trợ chữ cái và số" |
| **Tần suất** | 80+ lần/ngày |
| **Quy tắc nghiệp vụ** | • Full-text search với indexing • Phân tích từ khóa để gợi ý • Log truy vấn cho cải thiện UX |

### Nhóm 3: Giỏ hàng và thanh toán

**UC009: Thêm sách vào giỏ hàng**

|  |  |
| --- | --- |
| **Thuộc tính** | Giá trị |
| **ID** | UC009 |
| **Tên** | Thêm sản phẩm vào giỏ hàng |
| **Mục đích** | Cho phép khách hàng lưu sách để mua sau |
| **Actor chính** | Khách hàng đã đăng nhập |
| **Trigger** | Nhấp "Thêm vào giỏ" từ trang chi tiết/danh sách |
| **Tiền điều kiện** | • Đã đăng nhập • Sách còn hàng và active |
| **Hậu điều kiện** | • Cart được cập nhật • Badge số lượng tăng |
| **Luồng chính** | 1. Chọn số lượng (1-99) 2. Nhấp "Thêm vào giỏ" 3. Kiểm tra real-time: - Tồn kho ≥ quantity - isActive=true 4. Gọi API: POST /api/v1/cart/add json { bookId: "abc123", quantity: 2 } 5. Xử lý backend: - Tìm/khởi tạo cart - Merge item nếu trùng bookId - Validate total quantity 6. Tính toán: - Total items - Total price 7. Trả cart cập nhật 8. UI: - Toast: "Đã thêm vào giỏ!" - Cart badge animation - Button "Xem giỏ hàng" |
| **Luồng thay thế** | A1: Thêm nhanh (Quick Add) • Thêm với quantity=1 không confirm • Chỉ hiển thị toast  A2: Thêm từ wishlist • Chuyển sách từ wishlist sang cart |
| **Ngoại lệ** | E1: Chưa đăng nhập • Hiển thị modal: "Đăng nhập để mua hàng" • Redirect sau login  E2: Vượt tồn kho • Hiển thị: "Chỉ còn X sản phẩm" • Tự động điều chỉnh quantity  E3: Sách ngừng bán • Ẩn nút "Thêm giỏ" • Hiển thị badge "Ngừng kinh doanh" |
| **Tần suất** | 50+ lần/ngày |
| **Quy tắc nghiệp vụ** | • Tối đa 99 sản phẩm/loại • Auto-merge trùng bookId • Kiểm tra tồn kho thời gian thực • Cart expiry: 24h |

**UC0011: Đặt hàng**

|  |  |
| --- | --- |
| **Thuộc tính** | Giá trị |
| **ID** | UC011 |
| **Tên** | Hoàn tất đơn hàng |
| **Mục đích** | Chuyển giỏ hàng thành đơn hàng hoàn chỉnh |
| **Actor chính** | Khách hàng |
| **Actor phụ** | Email Service (Gửi xác nhận) |
| **Trigger** | Nhấp "Thanh toán" từ trang giỏ hàng |
| **Tiền điều kiện** | • Giỏ hàng không trống • Tất cả items còn hàng |
| **Hậu điều kiện** | • Tạo đơn hàng • Cập nhật tồn kho • Gửi email xác nhận |
| **Luồng chính** | Bước 1: Xác thực giỏ hàng 1. Validate cart items 2. Navigate → /checkout  Bước 2: Thông tin giao hàng 3. Pre-fill user info 4. Nhập shipping address (required) 5. Xem order summary  Bước 3: Tạo đơn hàng 6. Gọi API: POST /api/v1/order 7. DB transaction: - Kiểm tra lại tồn kho - Tạo order record: json {  orderNumber: "ORD-{timestamp}-{random6digits}",  items: [ { bookId, title, quantity, price } ],  totalAmount: ..., shippingAddress: ... } 8. Cập nhật inventory: book.quantity -= orderedQty; book.sold += orderedQty; 9. Clear cart  Bước 4: Xác nhận 10. Gửi email (async queue) 11. Redirect → /order/success/{orderNumber} 12. Hiển thị chi tiết đơn hàng |
| **Luồng thay thế** | A1: Thanh toán trực tuyến (Tích hợp thực) • Thêm bước chọn phương thức • Redirect đến cổng thanh toán • Xử lý webhook  A2: Mua hàng không tài khoản • Thu thập email để gửi thông tin |
| **Ngoại lệ** | E1: Hết hàng trong khi checkout • Rollback transaction • Hiển thị "Một số sách đã hết" • Quay lại giỏ hàng  E2: Lỗi thanh toán • Khôi phục tồn kho • Hiển thị: "Thanh toán thất bại" • Log lỗi chi tiết  E3: Lỗi email • Đơn hàng vẫn tạo thành công • Gửi lại email sau 10 phút • Thông báo: "Email xác nhận sẽ gửi sau" |
| **Tần suất** | 20-30 lần/ngày |
| **Quy tắc nghiệp vụ** | • OrderNumber format: ORD-YYYYMMDD-HHMMSS-RANDOM • Trạng thái mặc định: "Chờ xác nhận" • Atomic transaction cho inventory • Email xác nhận trong vòng 5 phút |

### Nhóm 4: Quản trị hệ thống

|  |  |
| --- | --- |
| **ID** | UC014 |
| **Tên** | Quản trị kho sách |
| **Mục đích** | Quản lý toàn bộ vòng đời sản phẩm trong hệ thống |
| **Actor chính** | Admin |
| **Trigger** | Truy cập /admin/books |
| **Tiền điều kiện** | • Role=Admin • JWT hợp lệ |
| **Hậu điều kiện** | • Thay đổi được áp dụng • Audit log được ghi |
| **Luồng chính** | • Tạo mới: • 1. Nhấp "Thêm sách" • 2. Điền form 20+ trường: • - Thông tin cơ bản (required) • - Meta description (SEO) • - Tải ảnh thumbnail + slider • 3. Validate data • 4. POST /api/v1/book •  • Cập nhật: • 5. Nhấp "Sửa" trên danh sách • 6. Pre-fill form với data hiện tại • 7. Cập nhật thông tin • 8. PUT /api/v1/book/{id} •  • Xóa mềm: • 9. Nhấp "Xóa" • 10. Confirm modal • 11. PATCH /api/v1/book/{id} → isActive=false |
| **Luồng thay thế** | • A1: Nhập hàng loạt từ Excel • - Upload file CSV • - Map columns • - Batch insert với validation •  • A2: Sao chép sách • - "Duplicate" button • - Tạo bản sao với tiền tố "[Copy]" |
| **Ngoại lệ** | • E1: Hình ảnh không hợp lệ • - Validate: size ≤5MB, định dạng JPG/PNG • - Tự động resize 800x1200px •  • E2: Xóa sách có liên quan đơn hàng • - Ngăn xóa vật lý • - Chỉ cho phép soft delete • - Hiển thị cảnh báo "Đã có X đơn hàng" |
| **Tần suất** | 10-20 thao tác/ngày |
| **Quy tắc nghiệp vụ** | • Soft delete cho toàn bộ sản phẩm • Auto-generate slug từ tên sách • Required fields: title, author, price, quantity • Tự động watermark ảnh |

**UC0016: Xử lý đơn hàng**

|  |  |
| --- | --- |
| **ID** | UC016 |
| **Tên** | Xử lý đơn hàng |
| **Mục đích** | Theo dõi và cập nhật trạng thái đơn hàng |
| **Actor chính** | Admin |
| **Actor phụ** | Email Service (Thông báo trạng thái) |
| **Trigger** | Truy cập /admin/orders |
| **Tiền điều kiện** | • Role=Admin • Đơn hàng tồn tại |
| **Hậu điều kiện** | • Trạng thái được cập nhật • Khách hàng nhận thông báo |
| **Luồng chính** | • Xem danh sách: • 1. Filter theo: status/date range/customer • 2. Hiển thị bảng với 10+ cột •  • Xem chi tiết: • 3. Nhấp vào orderNumber • 4. Hiển thị toàn bộ thông tin + timeline •  • Cập nhật trạng thái: • 5. Chọn trạng thái mới: pending → processing → shipped → delivered • 6. Thêm ghi chú nội bộ • 7. PUT /api/v1/order/{id}/status • 8. Validate status transition • 9. Gửi email thông báo tự động • 10. Ghi log lịch sử |
| **Luồng thay thế** | • A1: In nhãn vận đơn • - Tích hợp API shipping carrier • - Tự động điền thông tin vận chuyển •  • A2: Xuất báo cáo Excel • - Export filtered orders • - Tùy chọn columns |
| **Ngoại lệ** | • E1: Chuyển trạng thái không hợp lệ • - Ngăn chặn shipped → pending • - Hiển thị: "Không thể chuyển từ {current} sang {new}" •  • E2: Cập nhật đồng thời • - Optimistic locking • - Hiển thị "Đã bị thay đổi bởi admin khác" |
| **Tần suất** | 30-50 thao tác/ngày |
| **Quy tắc nghiệp vụ** | • Luồng trạng thái nghiêm ngặt • Email template theo từng trạng thái • Admin notes được mã hóa • Tự động hủy đơn sau 48h nếu chưa xử lý |

UC0017: Giám sát hoạt động hệ thống

|  |  |
| --- | --- |
| **ID** | UC017 |
| **Tên** | Giám sát hoạt động hệ thống |
| **Mục đích** | Cung cấp cái nhìn tổng quan về hiệu suất kinh doanh |
| **Actor chính** | Admin |
| **Trigger** | Đăng nhập admin hoặc truy cập /admin/dashboard |
| **Tiền điều kiện** | • Role=Admin • Có dữ liệu thống kê |
| **Hậu điều kiện** | • Hiển thị KPI real-time • Dữ liệu được làm mới |
| **Luồng chính** | • 1. Truy cập dashboard • 2. Gọi API: GET /api/v1/database/dashboard • 3. Aggregate data: • db.orders.aggregate([{ $match: { ... } }, { $group: { \_id: null, total: { $sum: "$amount" } } }]) • 4. Trả kết quả: • { totalRevenue: 12000000, newOrders: 24, topSelling: [...], salesChart: { labels: [...], data: [...] } } • 5. Render: • - Metric cards: Doanh thu, Đơn hàng, User mới • - Biểu đồ đường: Doanh thu theo tuần • - Biểu đồ tròn: Phân bố danh mục • - Bảng: Đơn hàng gần đây • 6. Tự động refresh 5 phút |
| **Luồng thay thế** | • A1: Xuất PDF báo cáo • - Chọn date range • - Tạo PDF client-side • - Tải về/email •  • A2: Chế độ xem real-time • - WebSocket cho cập nhật tức thì |
| **Ngoại lệ** | • E1: Dữ liệu lớn làm chậm • - Sử dụng materialized views • - Hiển thị skeleton loading •  • E2: Lỗi biểu đồ • - Fallback sang dạng bảng • - Log lỗi console |
| **Tần suất** | 20-30 lần/ngày |
| **Quy tắc nghiệp vụ** | • KPI tính toán real-time • Cache query 1 phút cho performance • Tự động highlight bất thường (↓30% doanh thu) |

# Thiết kế hệ thống

## Kiến trúc hệ thống

### Lựa chọn kiến trúc

Hệ thống áp dụng Kiến trúc phân lớp (Layered Architecture) kết hợp mô hình MVC (Model-View-Controller). Kiến trúc này được lựa chọn dựa trên phân tích so sánh với các mô hình khác:

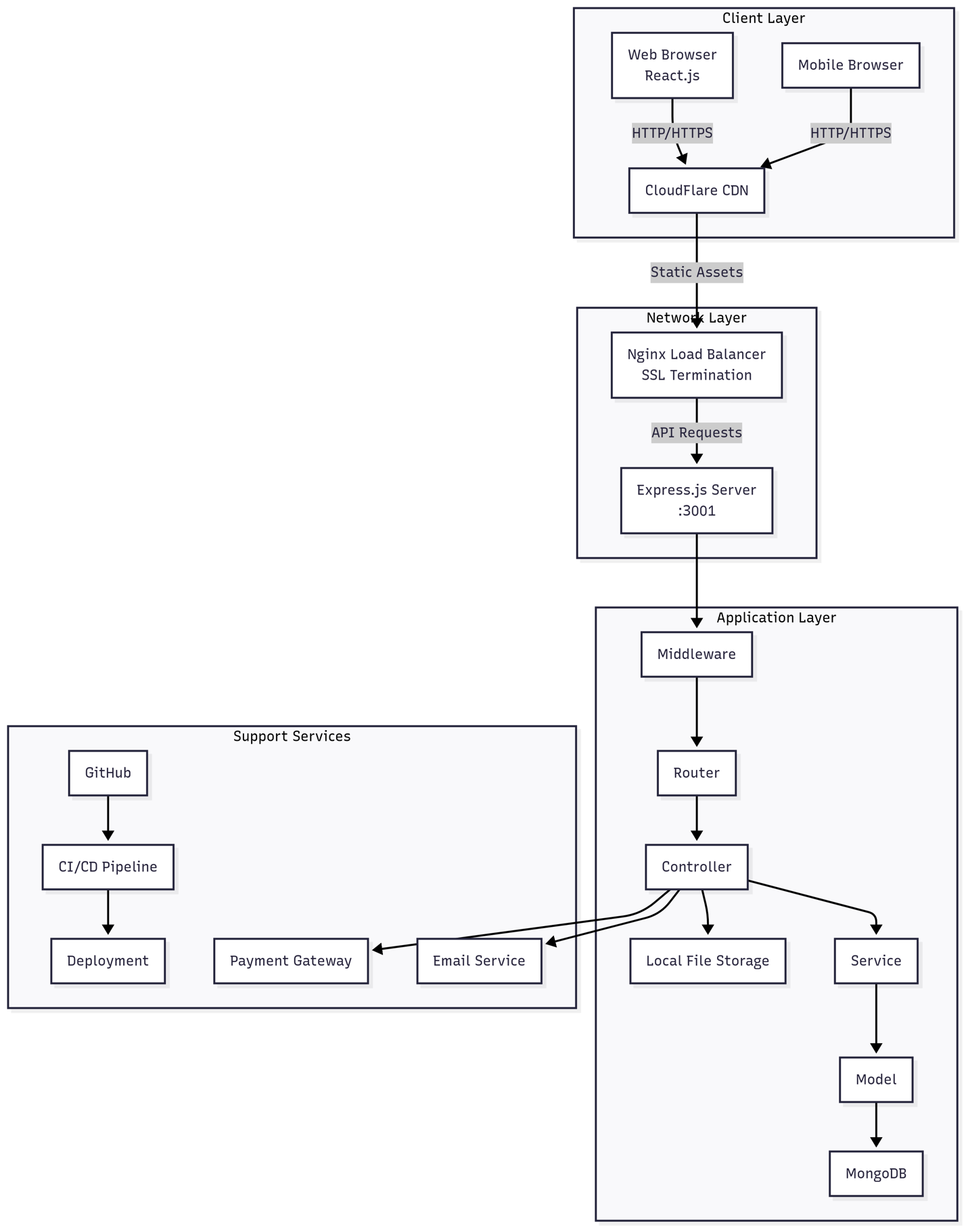
Lý do chính:

* Phù hợp quy mô dự án:
* Dự án học tập với team 3 thành viên
* Phạm vi chức năng tập trung (quản lý sách, đơn hàng)
* Thời gian phát triển ngắn (3 tháng)
* Hiệu quả phát triển:
* Codebase đơn nhất dễ quản lý
* Tách biệt rõ ràng giữa presentation, business logic và data access
* Dễ dàng prototype và testing
* Triển khai đơn giản:
* Single codebase, single database
* Không yêu cầu container orchestration
* Deploy dễ dàng trên VPS hoặc PaaS

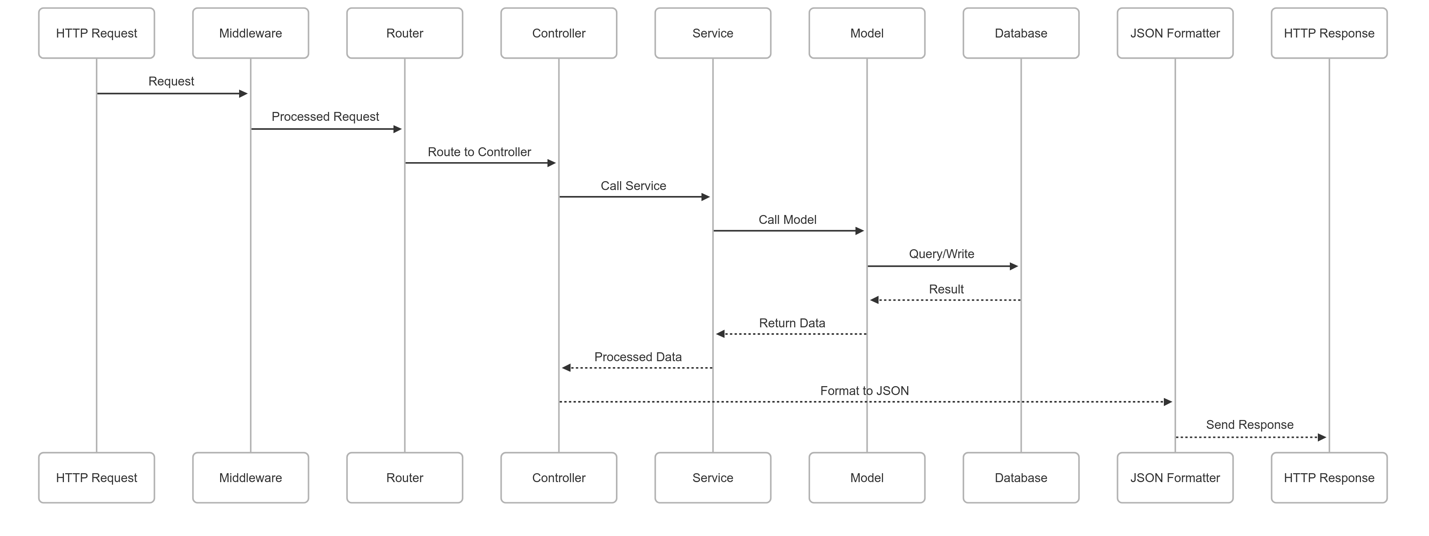
Kế hoạch chuyển đổi sang Microservices khi:

* Lưu lượng > 10,000 người dùng đồng thời
* Team phát triển > 10 thành viên
* Yêu cầu mở rộng dịch vụ độc lập

### Sơ đồ kiến trúc tổng thể



**Luồng dữ liệu:**

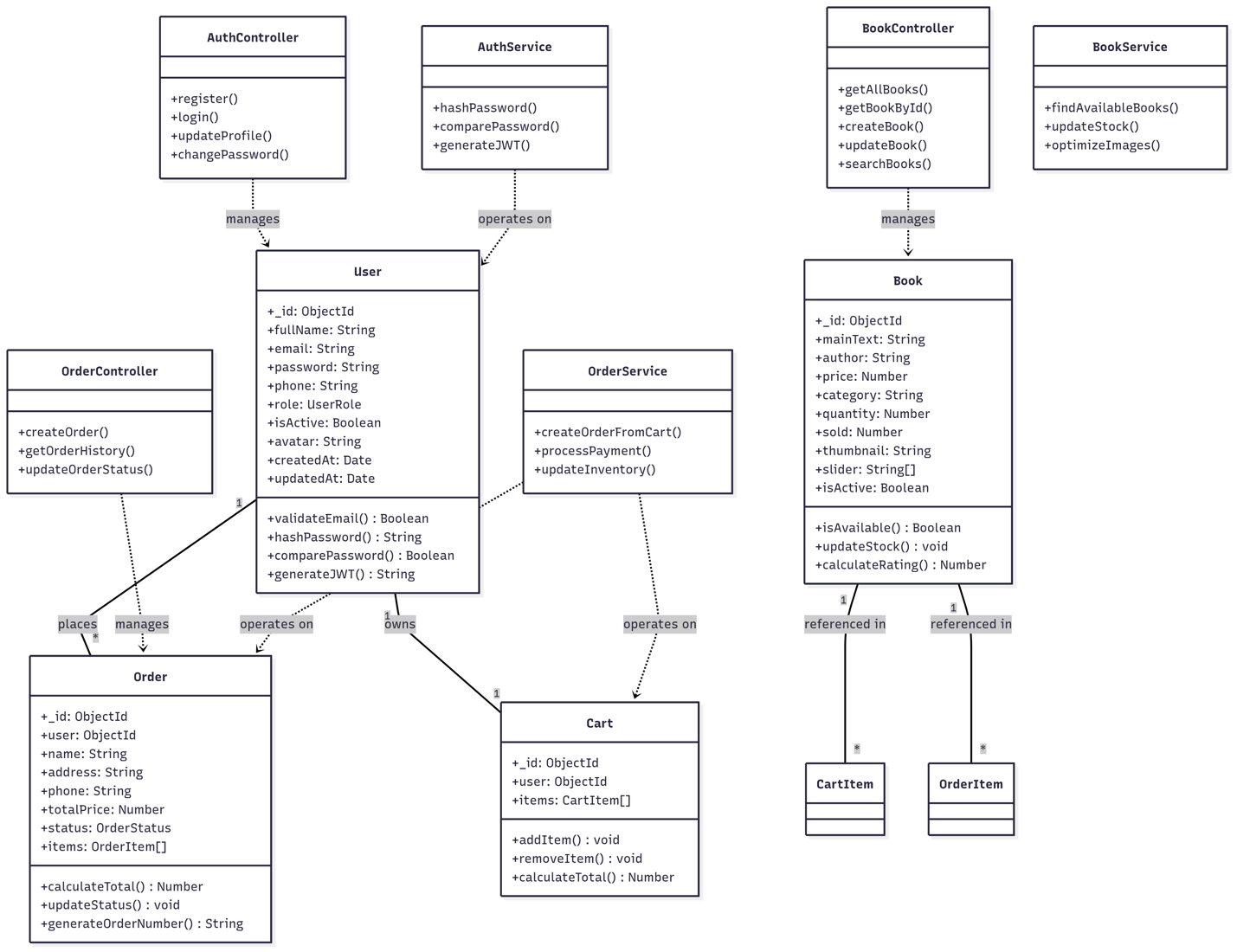


**Tối ưu hiệu năng:**

* Frontend: Code splitting, lazy loading
* Backend: Connection pooling, caching
* Database: Indexing, sharding
* Network: CDN cho static assets

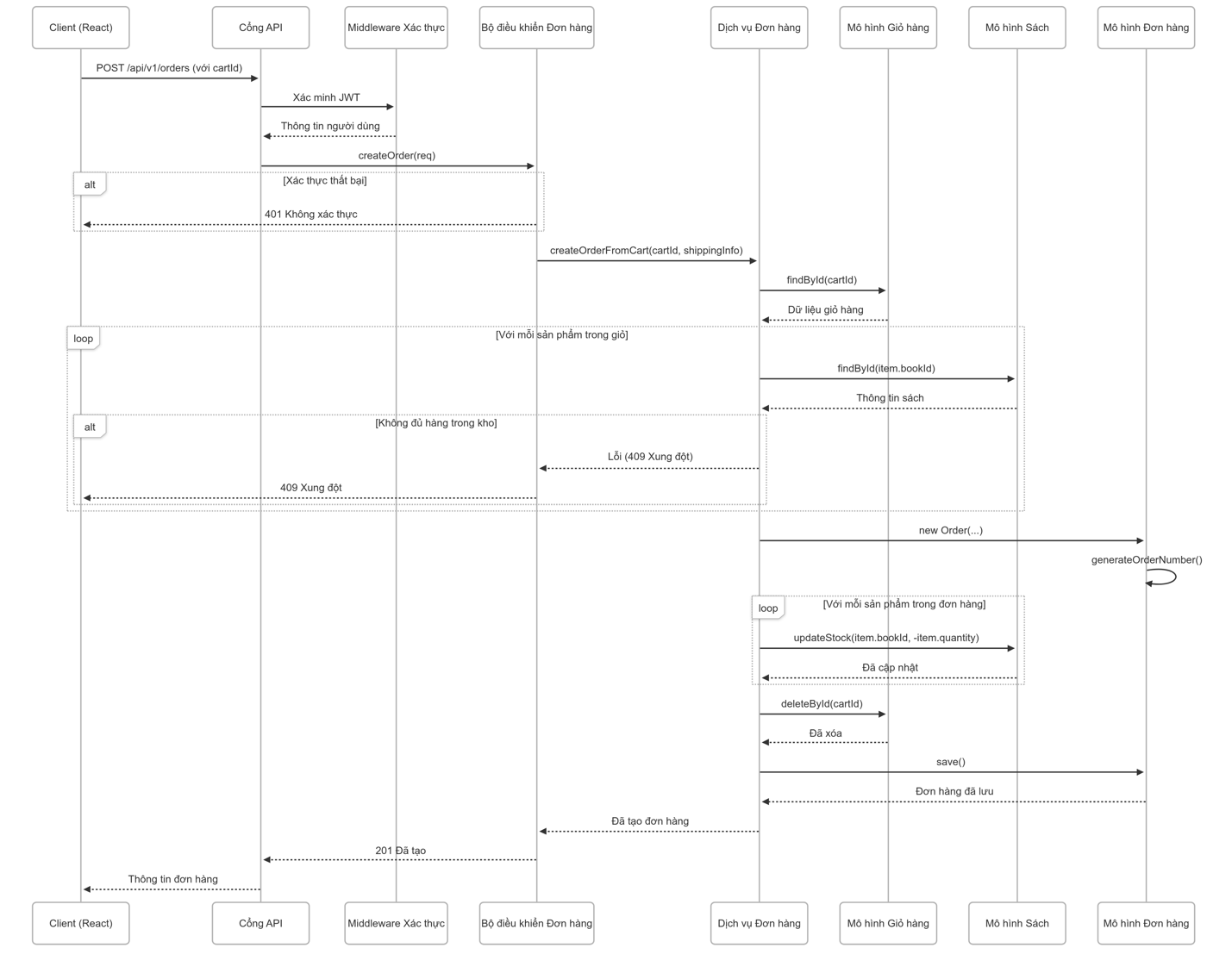
## Sơ đồ lớp

### Sơ đồ lớp tổng quan



## Sơ đồ tuần tự

### Luông "Tạo đơn hàng":



Giải thích luồng:

1. Khởi tạo: Client gửi request tạo đơn hàng với cartId
2. Xác thực: Middleware kiểm tra JWT và trích xuất thông tin user
3. Kiểm tra giỏ hàng: Service truy vấn thông tin giỏ hàng
4. Kiểm tra tồn kho: Với mỗi sản phẩm trong giỏ:

* Truy vấn thông tin sách
* Kiểm tra số lượng tồn kho

1. Tạo đơn hàng:

* Khởi tạo đối tượng Order
* Generate order number

1. Cập nhật tồn kho: Trừ số lượng sách đã đặt
2. Xóa giỏ hàng: Xóa giỏ hàng sau khi đặt thành công
3. Phản hồi: Trả về thông tin đơn hàng đã tạo

Xử lý ngoại lệ:

Tình huống Mã lỗi Hành động

Xác thực thất bại 401 Dừng luồng

Giỏ hàng không tồn tại 404 Dừng luồng

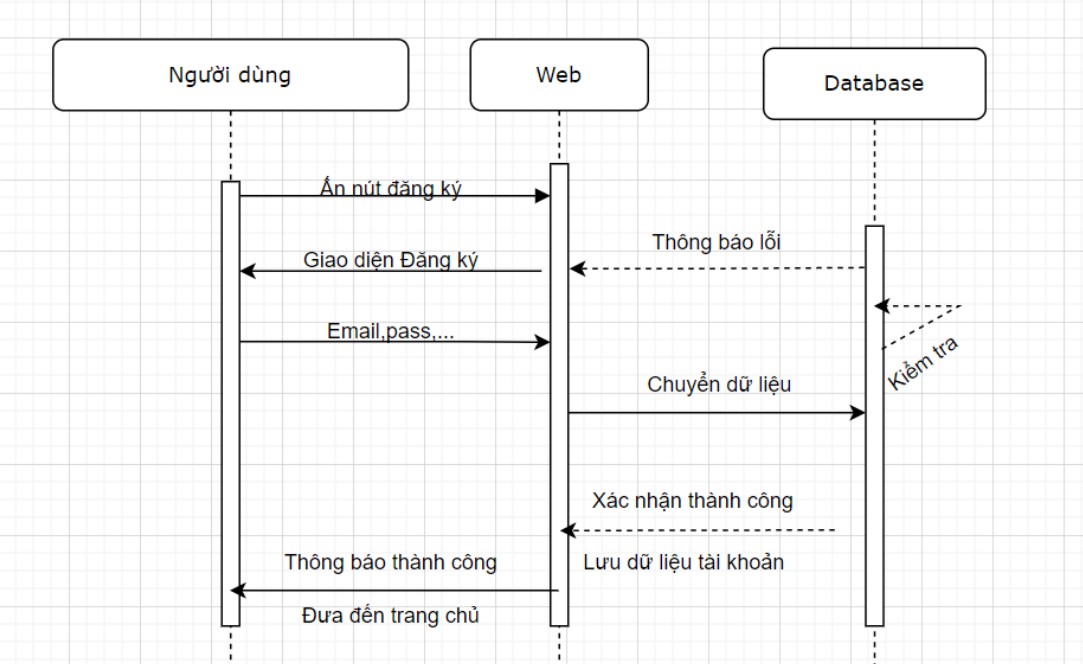
Hết hàng 409 Rollback transaction

Lỗi database 500 Rollback + ghi log

Tối ưu hiệu suất:

* Sử dụng MongoDB transactions
* Kiểm tra tồn kho song song (Promise.all)
* Giới hạn timeout cho database queries
* Bulk write operations cho cập nhật tồn kho

### Luồng “đăng ký”

****

Khách hàng chọn chức năng đăng ký trong trang chủ. Hệ thống sẽ trả về giao diện đăng ký. Học viên nhập các thông tin để đăng ký tài khoản như họ và tên, số điện thoại hoặc email, những thông tin cần thiết đẻ đăng ký tài khoản. 21

Hệ thống sẽ kiểm tra xem email hoặc số điện thoại có tồn tại không. Nếu có hệ thống sẽ yêu cầu nhập lại

Sau khi thông tin được xác nhận, hệ thống sẽ tạo một tài khoản mới cho học viên và lưu trữ các thông tin cá nhân của họ trong cơ sở dữ liệu, sau đó thông báo cho học viên biết rằng họ đã đăng ký thành công, sau đó chuyển về giao diện để đăng nhập

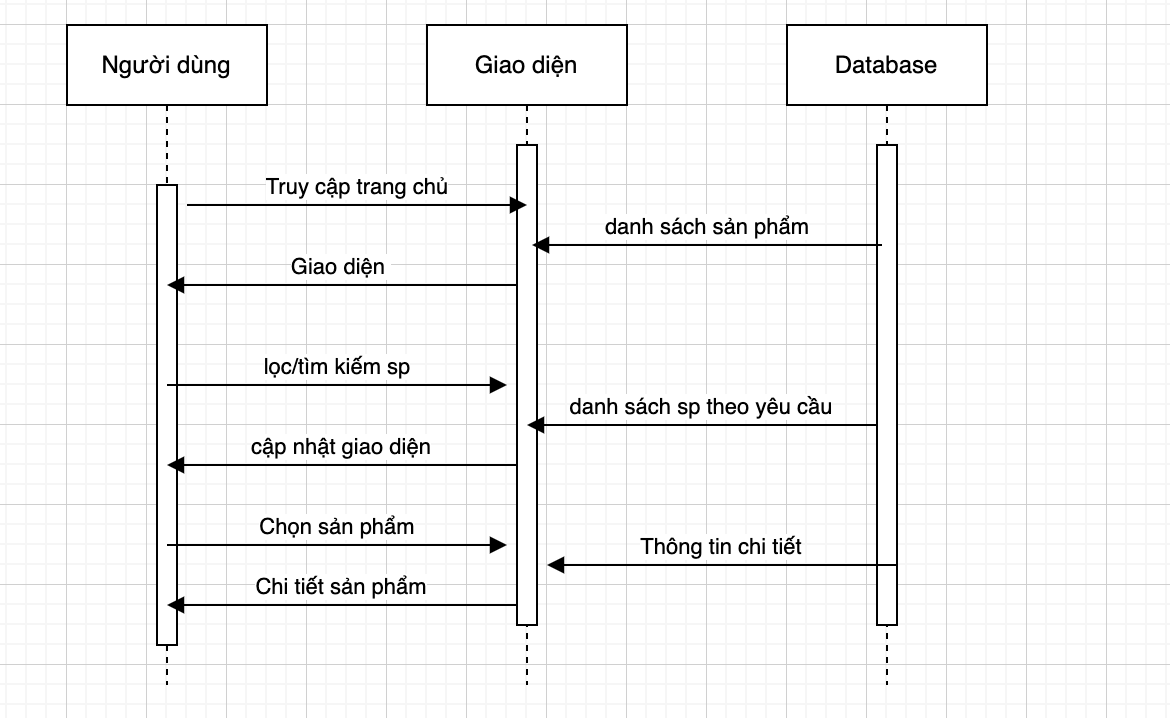
### Biểu đồ tuần tự chức năng đăng nhập

Khách hàng chọn chức năng đăng nhập trong trang chủ. Hệ thống sẽ trả về giao diện đăng nhập. Khách hàng nhập email, sau đó nhập mật khẩu.

Hệ thống xác nhận thông tin đăng nhập của khách hàng và kiểm tra tính hợp lệ của thông tin này. Nếu thông tin đúng và hợp lệ, hệ thống sẽ tiếp tục đăng nhập vào trang web. Nếu thông tin không đúng hoặc không hợp lệ, hệ thống sẽ yêu cầu khách hàng cung cấp lại thông tin đăng nhập.

Sau khi thông tin đăng nhập được xác nhận, hệ thống sẽ tải trang chính của ứng dụng hoặc trang web để cho phép khách hàng truy cập các tính năng của nó..

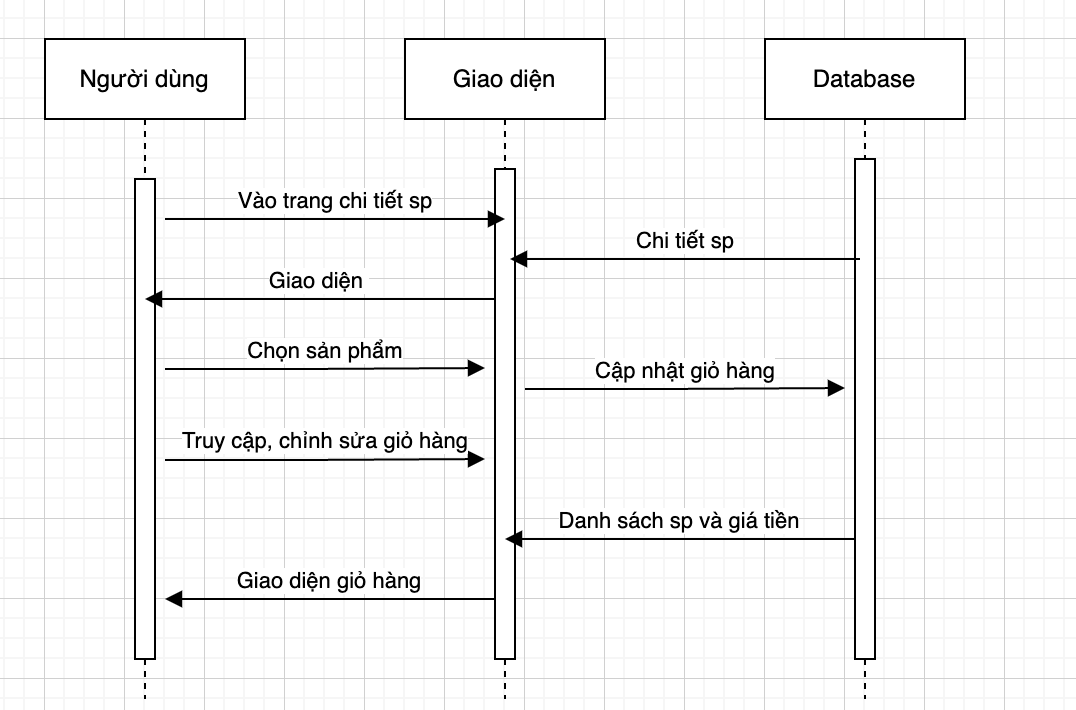
# Biểu đồ tuần tự chức năng xem sản phẩm

****

Sau khi đăng nhập thành công, khách hàng được đưa đến trang chủ để lựa chọn sản phẩm. Tại đây, khách hàng có thể lọc sản phẩm dựa vào thể loại, giá, lượt bán hoặc tìm kiếm tên sản phẩm ở ô tìm kiếm.

Sau khi tìm được sản phẩm muốn mua, khách hàng nhấn trực tiếp vào bìa sản phẩm. Giao diện chi tiết sản phẩm được hiện ra.

### Biểu đồ tuần tự chức năng quản lý giỏ hàng



Sau khi xem chi tiết sản phẩm, nếu khách hàng thấy sản phẩm phù hợp có thể thêm sản phẩm vào giỏ hàng. Khi đó, giỏ hàng sẽ được cập nhật và khách hàng có thể tiếp tục tìm kiếm sản phẩm khác.

Khi không còn nhu cầu tìm kiếm thêm sản phẩm, khách hàng truy cập vào giỏ hàng để xem tất cả sản phẩm đã được thêm. Khách hàng có thể điều chỉnh được số lượng mặt hàng hoặc xóa bỏ sản phẩm khỏi giỏ hàng.

Giá tiền đã được tính toán sẵn, nhấn mua hàng sau đó nhập địa chỉ để đặt hàng.

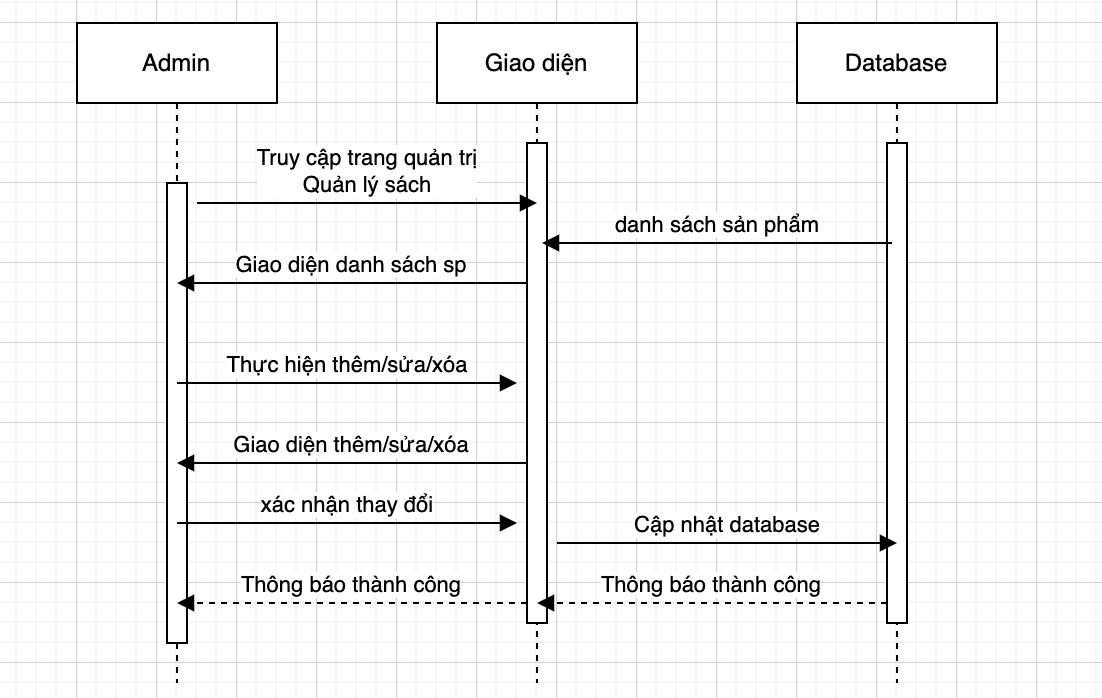
### Biểu đồ tuần tự chức năng quản lý users

# 

Sau khi đăng nhập tài khoản Admin, quản trị viên truy cập vào trang quản trị, chọn quản lý users để thực hiện chức năng này.

Tại đây, danh sách tất cả users đã được đăng ký sẽ được hiện ra. Quản trị viên có thể tiến hành thêm, sửa, xóa hoặc xem chi tiết tài khoản của khách hàng.

### Biểu đồ tuần tự chức năng quản lý sách

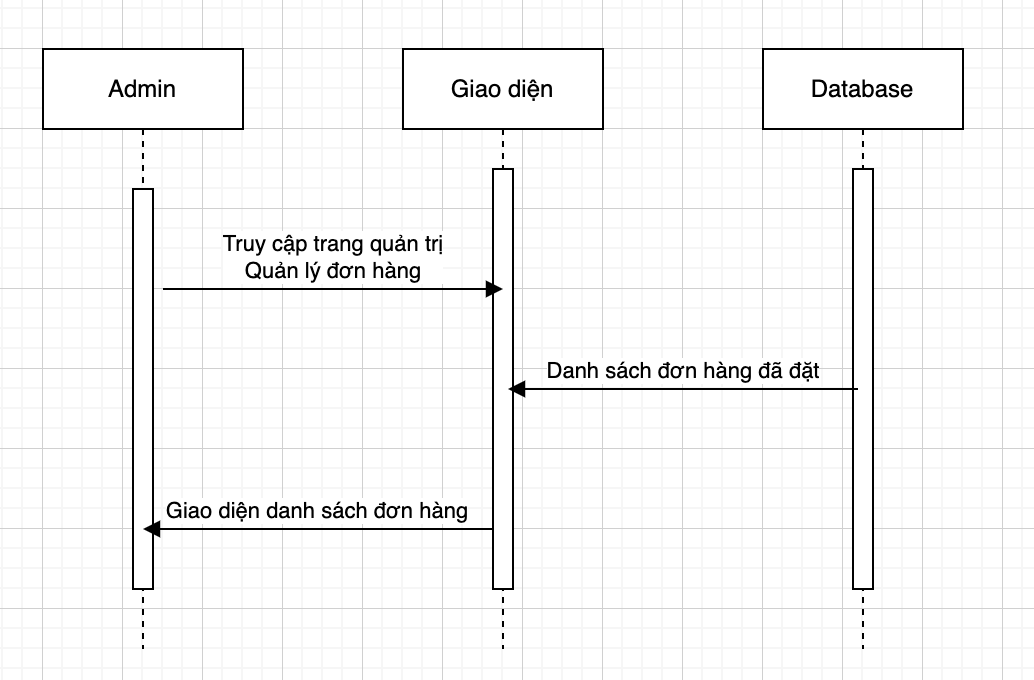


Tương tự như chức năng quản lý users, sau khi đăng nhập tài khoản Admin, quản trị viên truy cập vào trang quản trị, chọn quản lý sách để thực hiện chức năng này.

Tại đây, danh sách tất cả sách đã được bày bán trên hệ thống sẽ được hiện ra.

Quản trị viên có thể tiến hành thêm, sửa, xóa hoặc xem chi tiết các sản phẩm này.

### Biểu đồ tuần tự chức năng quản lý đơn hàng

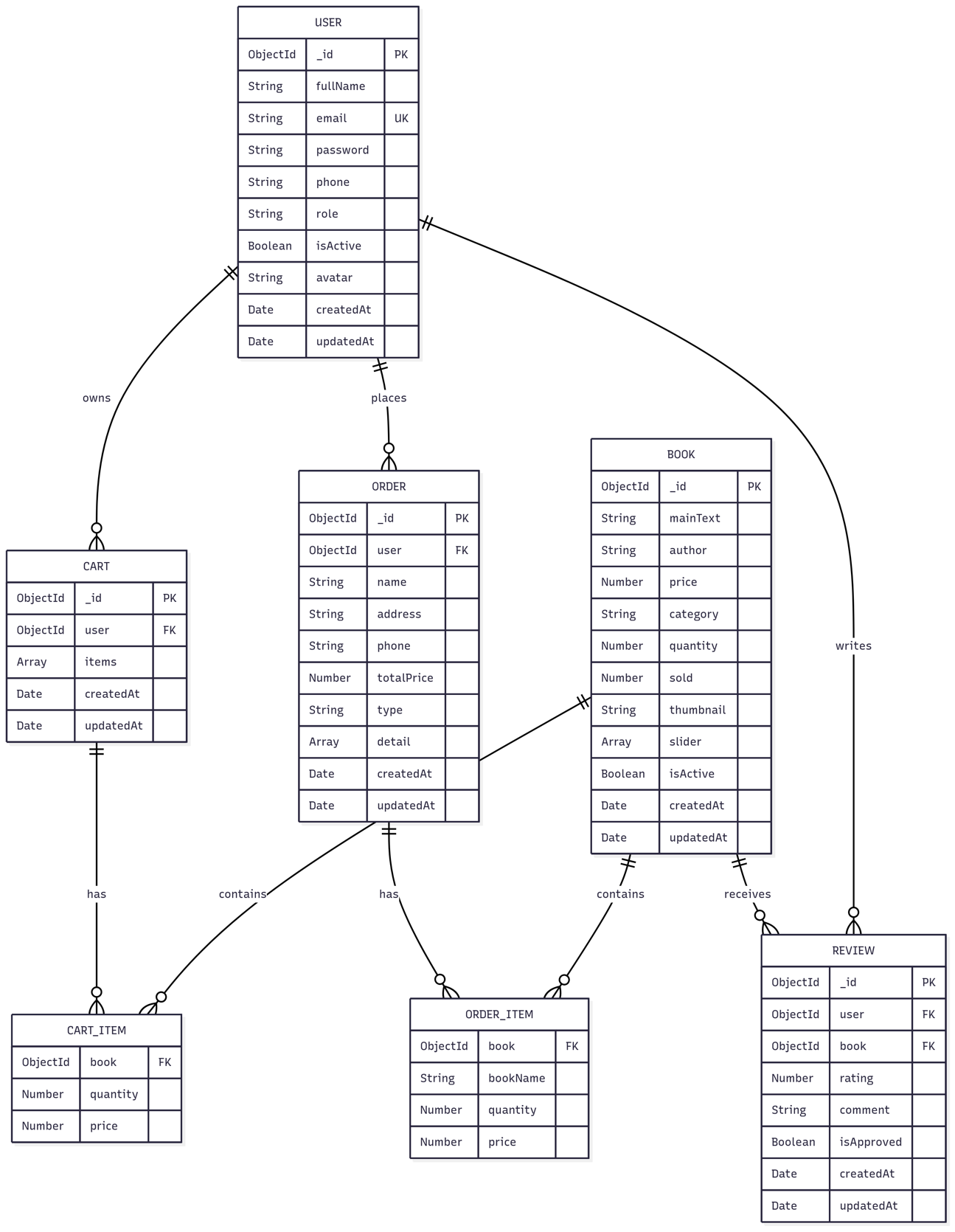


Sau khi đăng nhập tài khoản Admin, quản trị viên truy cập vào trang quản trị, chọn quản lý orders. Danh sách tất cả các đơn hàng đã được đặt thành công được hiển thị.

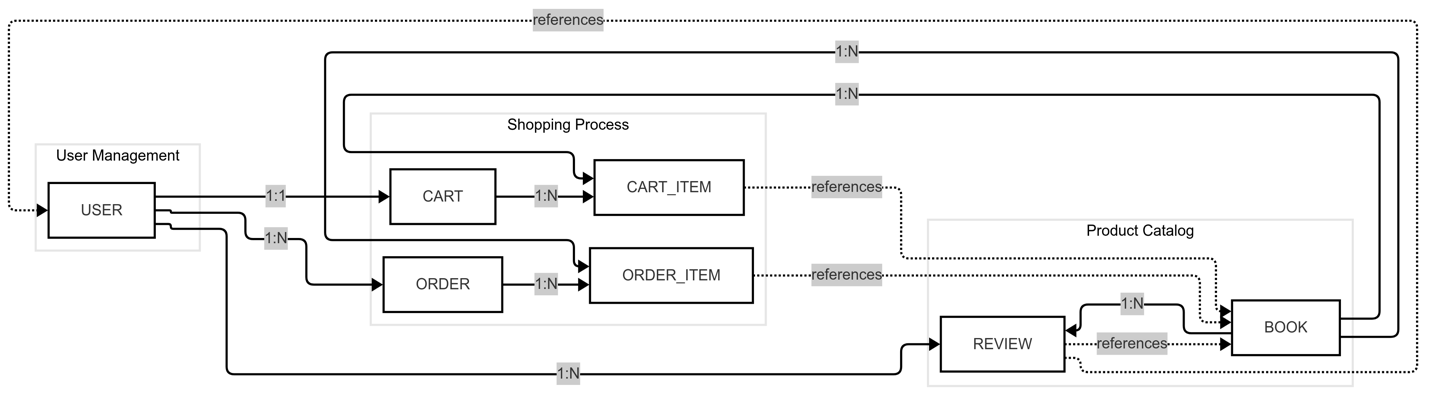
# Cơ sở dữ liệu

## Mô hình ERD

### Sơ đồ ERD của hệ thống



### Mô hình quan hệ chi tiết



## Thiết kế bảng chi tiết

### Bảng USER (Người dùng)

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| \_id | ObjectId | PK, Required | Khóa chính MongoDB |
| fullName | String | Required, MinLength: 2, MaxLength: 100 | Họ và tên đầy đủ |
| email | String | Required, Unique, Email format | Email đăng nhập |
| password | String | Required, MinLength: 6 | Mật khẩu đã hash (bcrypt) |
| phone | String | Required, Phone format | Số điện thoại |
| role | String | Enum: ['USER', 'ADMIN'], Default: 'USER' | Vai trò người dùng |
| isActive | Boolean | Default: true | Trạng thái kích hoạt |
| avatar | String | Optional | Đường dẫn ảnh đại diện |
| createdAt | Date | Default: Date.now | Thời gian tạo |
| updatedAt | Date | Default: Date.now | Thời gian cập nhật |

### Bảng Book (Sách)

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| \_id | ObjectId | PK, Required | Khóa chính MongoDB |
| mainText | String | Required, MinLength: 1, MaxLength: 200 | Tên sách |
| author | String | Required, MaxLength: 100 | Tác giả |
| price | Number | Required, Min: 0 | Giá sách (VND) |
| category | String | Required, Enum | Danh mục sách |
| quantity | Number | Required, Min: 0, Integer | Số lượng tồn kho |
| sold | Number | Default: 0, Min: 0, Integer | Số lượng đã bán |
| thumbnail | String | Required | Ảnh đại diện |
| slider | Array<String> | Default: [] | Ảnh slider |
| isActive | Boolean | Default: true | Trạng thái bán |
| createdAt | Date | Default: Date.now | Thời gian tạo |
| updatedAt | Date | Default: Date.now | Thời gian cập nhật |

### Bảng CART(Giỏ hàng)

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| \_id | ObjectId | PK, Required | Khóa chính MongoDB |
| user | ObjectId | Required, FK → USER.\_id | Tham chiếu đến người dùng |
| items | Array<CartItem> | Default: [] | Danh sách sản phẩm trong giỏ |
| createdAt | Date | Default: Date.now | Thời gian tạo |
| updatedAt | Date | Default: Date.now | Thời gian cập nhật |

### Cart Item Sub-Schema

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| book | ObjectId | Required, FK → BOOK.\_id | Tham chiếu đến sách |
| quantity | Number | Required, Min: 1, Max: 99 | Số lượng |
| price | Number | Required, Min: 0 | Giá tại thời điểm thêm vào giỏ |

### Bảng ORDER (Đơn hàng)

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| \_id | ObjectId | PK, Required | Khóa chính MongoDB |
| user | ObjectId | Required, FK → USER.\_id | Tham chiếu đến người dùng |
| name | String | Required, MaxLength: 100 | Tên người nhận |
| address | String | Required, MaxLength: 500 | Địa chỉ giao hàng |
| phone | String | Required | Số điện thoại nhận hàng |
| totalPrice | Number | Required, Min: 0 | Tổng giá trị đơn hàng |
| type | String | Enum, Default: 'pending' | Trạng thái đơn hàng |
| detail | Array<OrderItem> | Required | Chi tiết sản phẩm |
| createdAt | Date | Default: Date.now | Thời gian tạo |
| updatedAt | Date | Default: Date.now | Thời gian cập nhật |

### Order Item Sub-Schema

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| book | ObjectId | Required, FK → BOOK.\_id | Tham chiếu đến sách |
| bookName | String | Required | Tên sách tại thời điểm đặt |
| quantity | Number | Required, Min: 1 | Số lượng |
| price | Number | Required, Min: 0 | Giá tại thời điểm đặt |

### Bảng REVIEW (Đánh giá)

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| \_id | ObjectId | PK, Required | Khóa chính MongoDB |
| user | ObjectId | Required, FK → USER.\_id | Tham chiếu đến người dùng |
| book | ObjectId | Required, FK → BOOK.\_id | Tham chiếu đến sách |
| rating | Number | Required, Min: 1, Max: 5 | Điểm đánh giá |
| comment | String | Optional, MaxLength: 1000 | Bình luận |
| isApproved | Boolean | Default: false | Trạng thái duyệt |
| createdAt | Date | Default: Date.now | Thời gian tạo |
| updatedAt | Date | Default: Date.now | Thời gian cập nhật |

## Liên kết giữa các bảng

Quan hệ 1 - 1: USER → CART (1:1)

Quan hệ 1 - N:

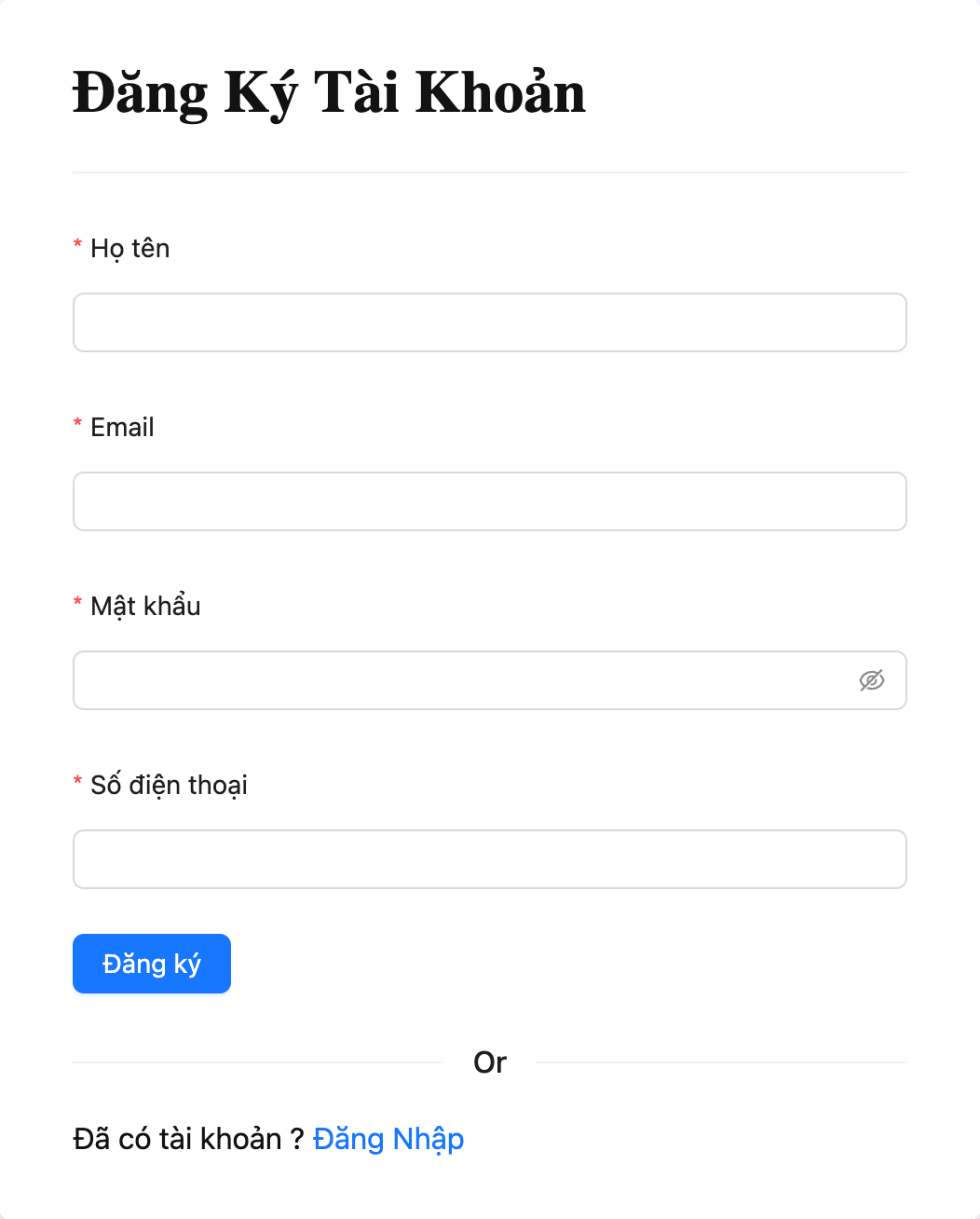
* USER → ORDER
* BOOK → REVIEW

Quan hệ N - N:

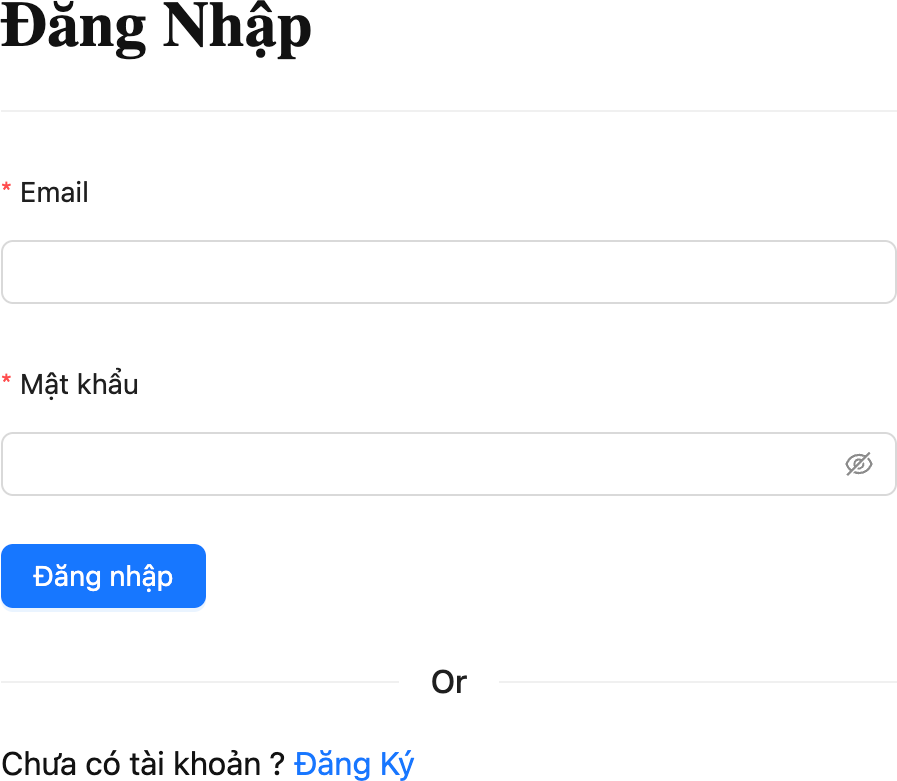
* USER → BOOK (CART)
* USER → BOOK (ORDER)

# Giao diện người dùng

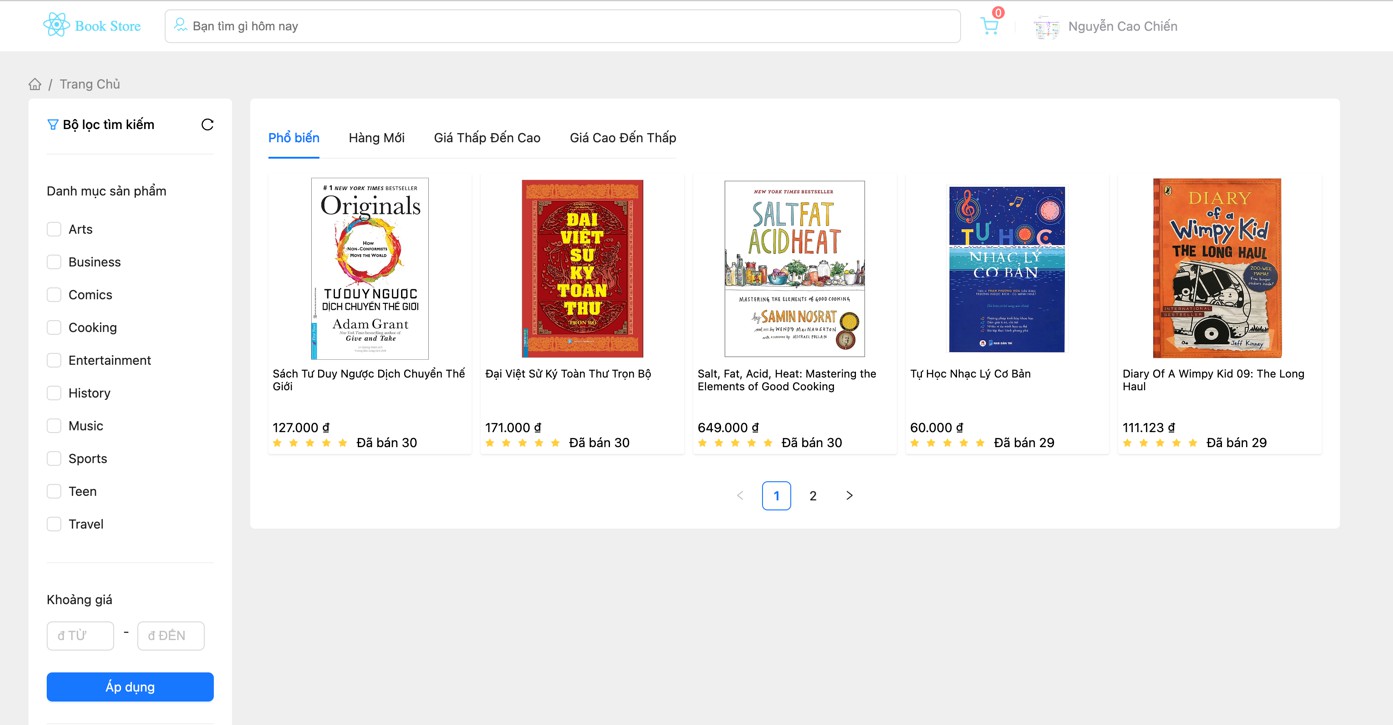
## Giao diện đăng ký



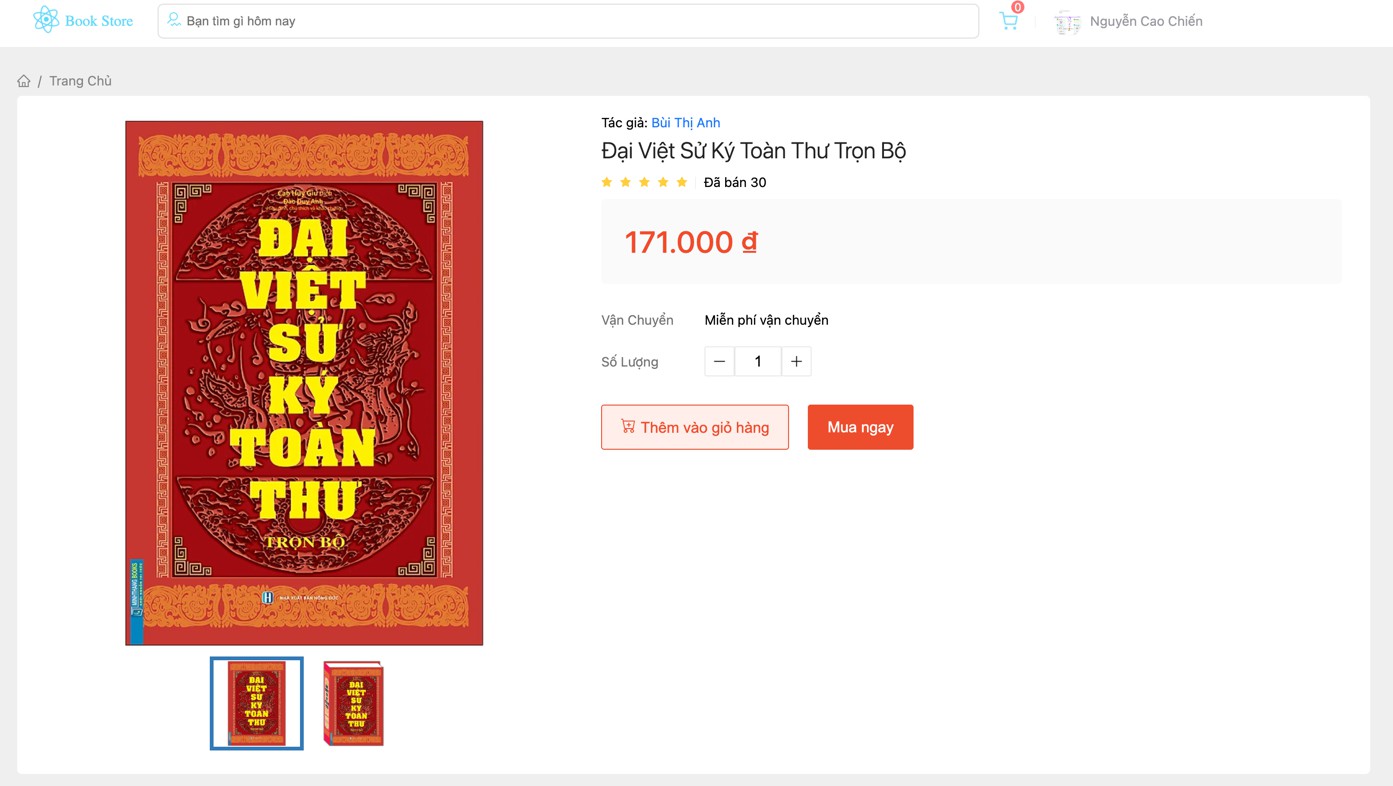
## Giao diện đăng nhập



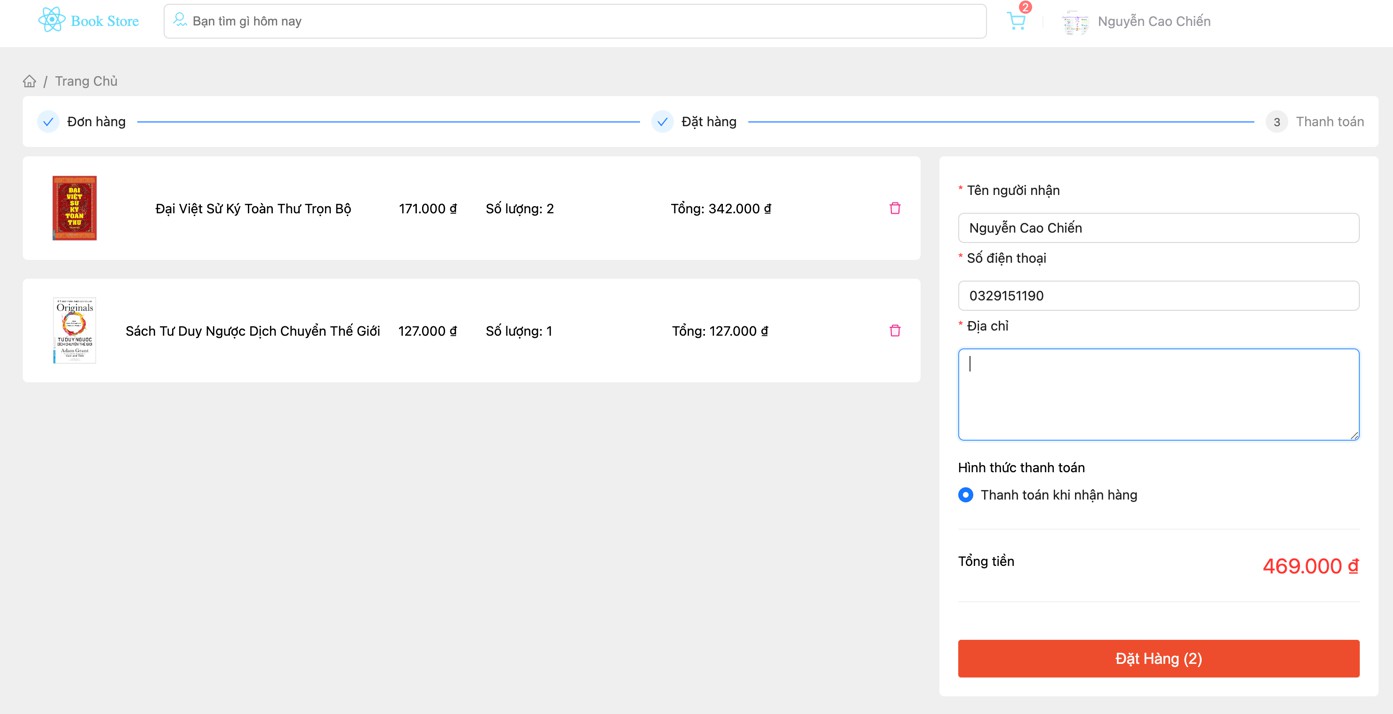
## Giao diện trang chủ



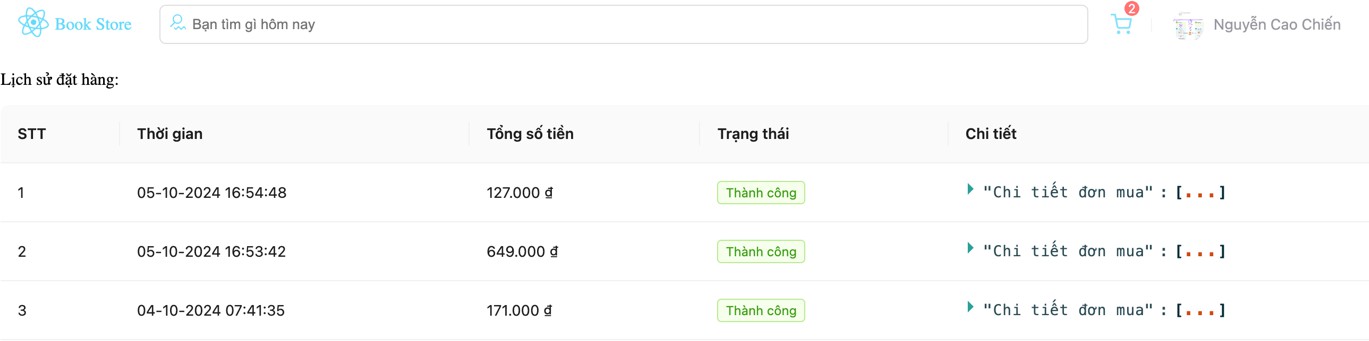
## Giao diện chi tiết sản phẩm

****

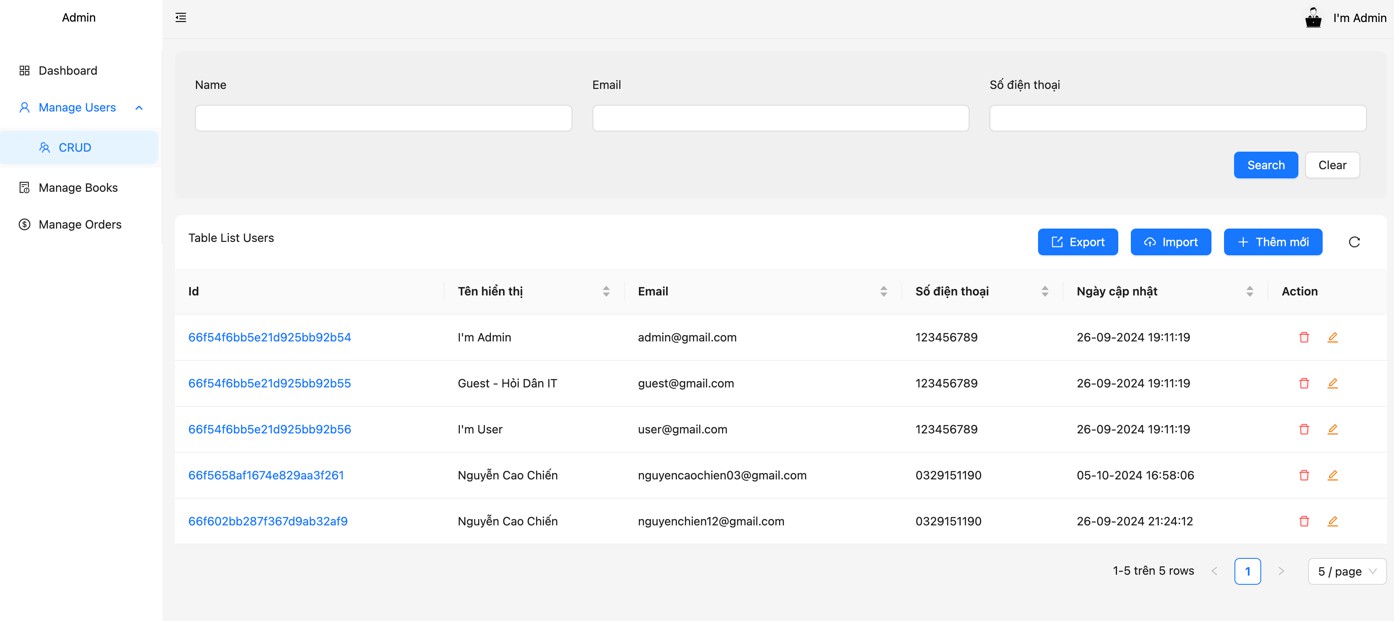
## Giao diện giỏ hàng

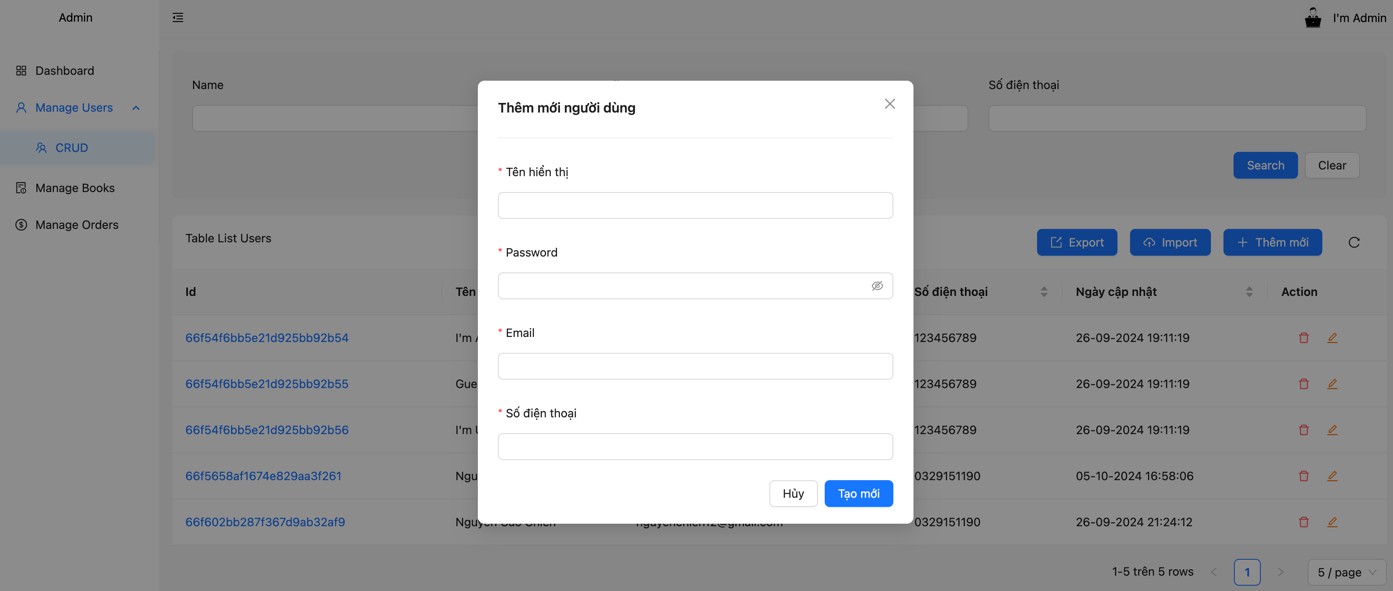


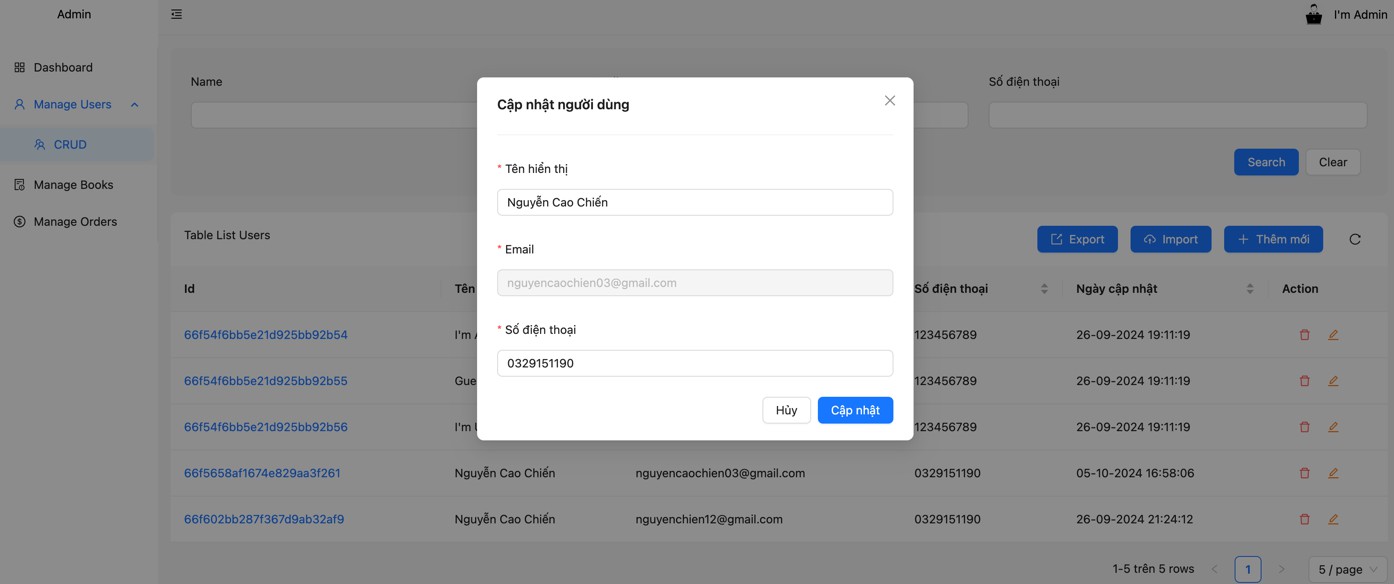
## Giao diện lịch sử mua hàng

****

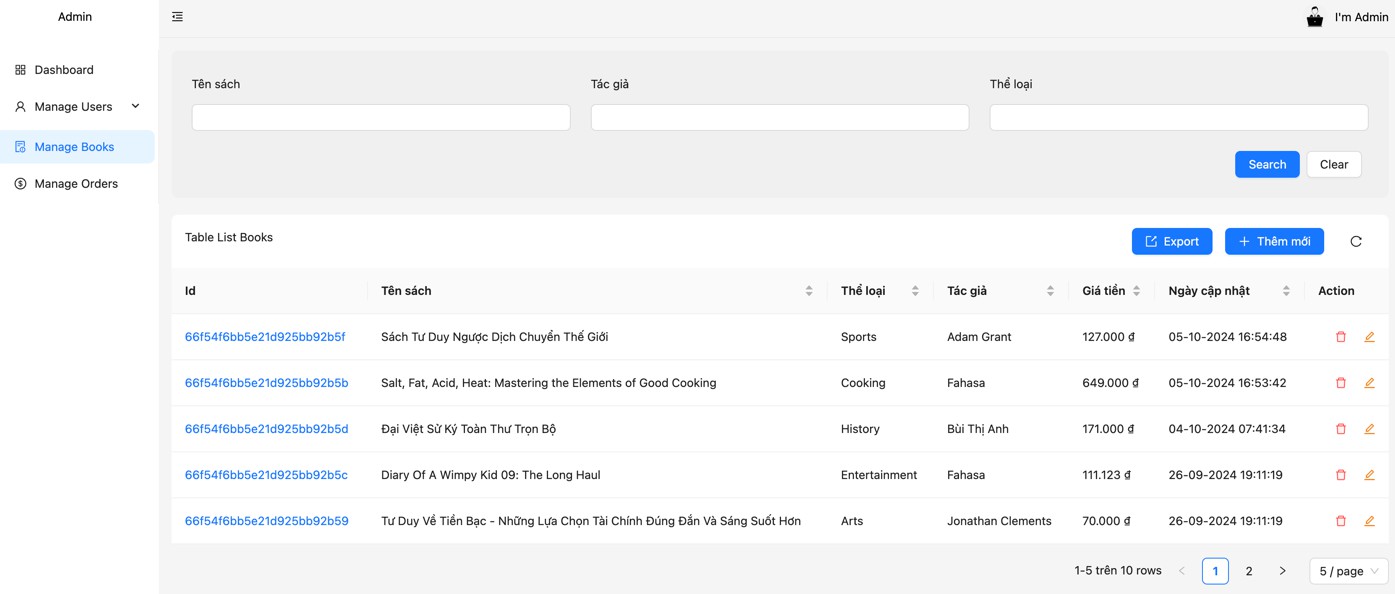
## Giao diện quản lý user

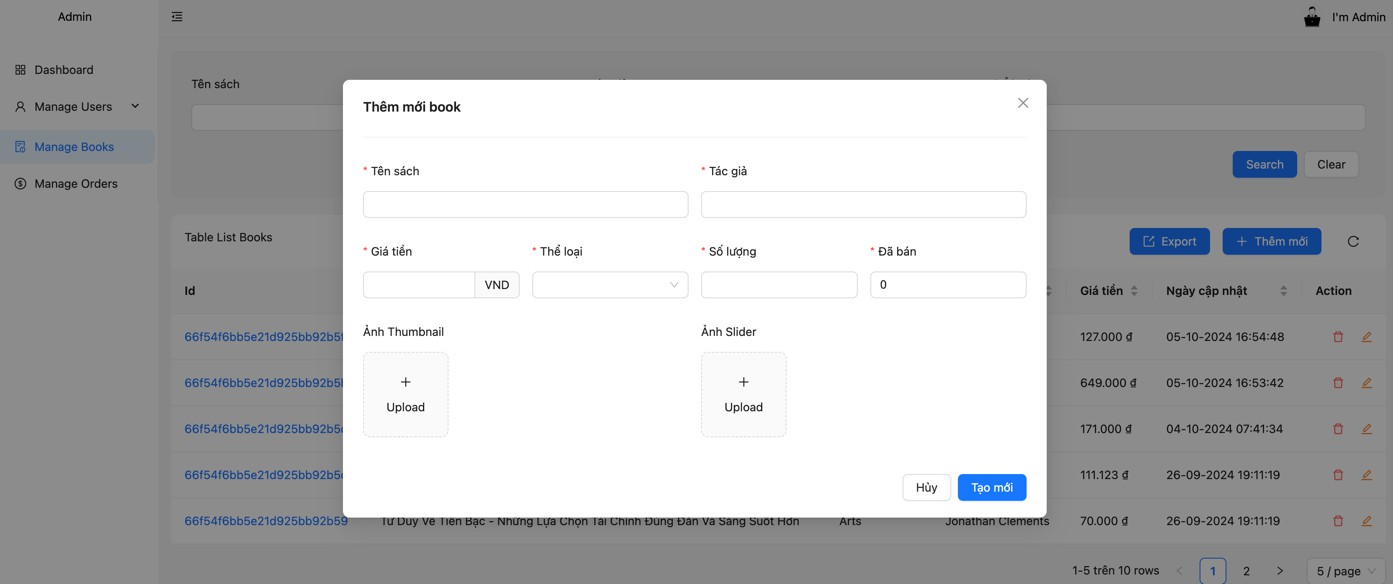
****

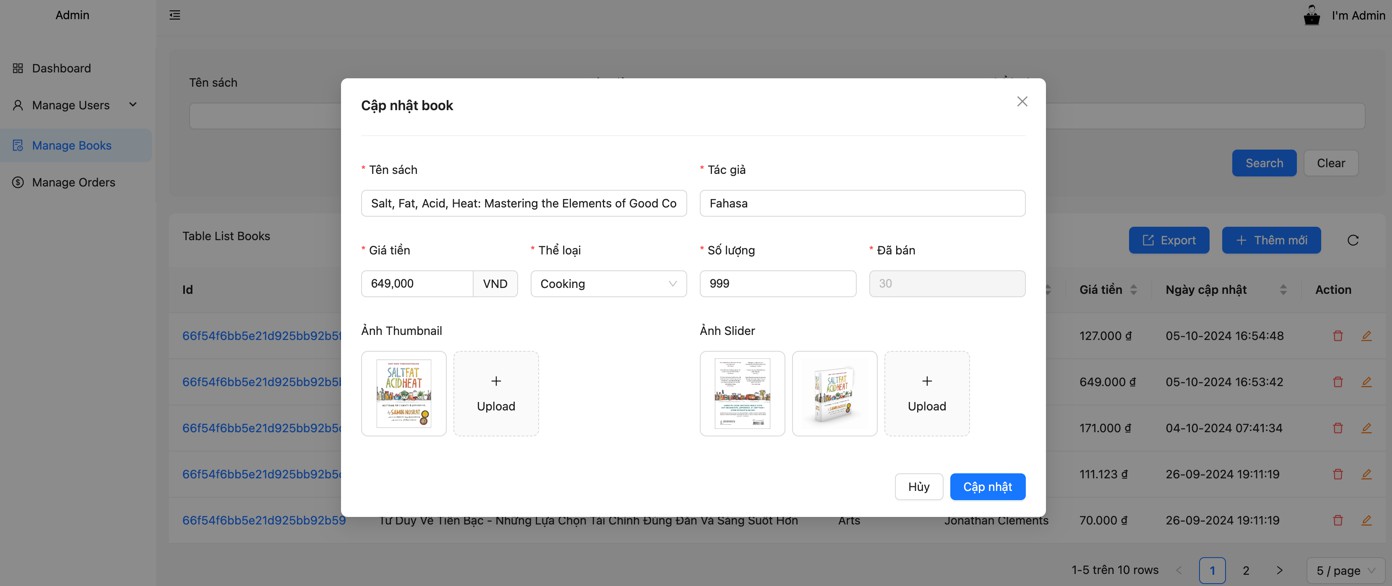


****

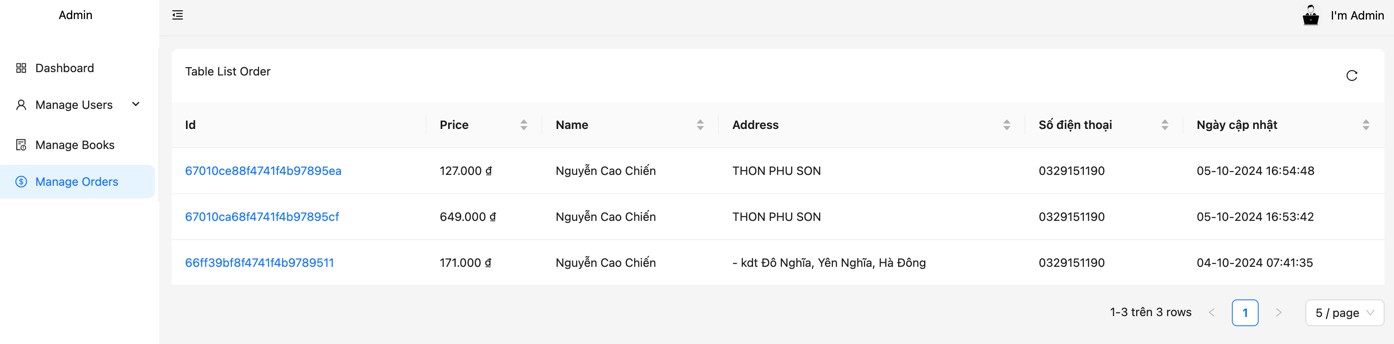
## Giao diện quản lý sách

****



****

## Giao diện quản lý đơn hàng

****

# Mẫu thiết kế áp dụng

## Khái quát về Design Patterns trong hệ thống

Hệ thống bán sách trực tuyến được thiết kế dựa trên các mẫu thiết kế phần mềm (Design Patterns) nhằm bảo đảm kiến trúc nhất quán, dễ mở rộng và thuận tiện bảo trì. Việc lựa chọn các pattern phù hợp giúp tái sử dụng mã nguồn, giảm thiểu sai sót lặp lại và chuẩn hóa quy trình phát triển. Bảy nhóm mẫu chính được áp dụng gồm: Singleton, Factory, MVC, Repository, Observer, Middleware, Strategy và Builder.

## Phân tích từng mẫu thiết kế

Singleton được sử dụng cho lớp kết nối cơ sở dữ liệu MongoDB. Bằng cách bảo đảm chỉ tồn tại một thể hiện duy nhất của lớp kết nối, hệ thống tiết kiệm tài nguyên, tránh tạo dư thừa connection pool và duy trì trạng thái đồng nhất xuyên suốt vòng đời ứng dụng.

Factory chịu trách nhiệm khởi tạo đối tượng phản hồi API. Thay vì tạo thủ công từng cấu trúc JSON, lớp Factory cung cấp các phương thức success, error, validationError… bảo đảm mọi phản hồi tuân theo định dạng chuẩn, từ đó nâng cao tính dễ đọc, khả năng mở rộng và giảm rủi ro sai khác khi thay đổi cấu trúc dữ liệu.

MVC (Model–View–Controller) được chọn làm kiến trúc tổng thể cho toàn bộ hệ thống. Lớp Model chứa nghiệp vụ dữ liệu và định nghĩa schema; Controller điều phối luồng xử lý, thực thi nghiệp vụ; View (React) đảm nhiệm hiển thị và tương tác với người dùng. Cấu trúc này tách bạch trách nhiệm, cho phép các nhóm phát triển song song và đơn giản hóa quy trình kiểm thử.

Repository đóng vai trò lớp trung gian giữa Controller và cơ sở dữ liệu, ẩn chi tiết truy vấn khỏi tầng nghiệp vụ. Nhờ đó, việc chuyển đổi nguồn dữ liệu hoặc thêm cơ chế cache trở nên dễ dàng, đồng thời hỗ trợ viết unit test nhờ khả năng mock repository.

Observer được hiện thực thông qua Redux. Khi state thay đổi (ví dụ số lượng sách trong giỏ hàng hoặc trạng thái đăng nhập), các component liên quan (Observers) được thông báo tự động và render lại. Điều này tạo nên trải nghiệm thời gian thực, hạn chế props-drilling và nâng cao khả năng tách rời giao diện.

Middleware là chuỗi các hàm trung gian trong Express giúp xử lý các mối quan tâm xuyên suốt (cross-cutting concerns) như xác thực, phân quyền, log lỗi, kiểm soát tốc độ truy cập và kiểm tra dữ liệu đầu vào. Cơ chế này bảo đảm mã nguồn gọn gàng, dễ mở rộng và tăng cường an toàn cho API.

Strategy được áp dụng trong mô-đun xác thực. Mỗi phương thức đăng nhập (tài khoản nội bộ, OAuth…) được đóng gói thành một strategy riêng, cho phép hoán đổi linh hoạt mà không phải thay đổi luồng nghiệp vụ. Cách tiếp cận này cũng mở đường tích hợp nhiều cổng thanh toán hoặc thuật toán tìm kiếm khác nhau trong tương lai.

Builder hỗ trợ xây dựng truy vấn dữ liệu phức tạp một cách tuần tự thông qua method-chaining. Lớp BookQueryBuilder giúp kết hợp linh hoạt bộ lọc danh mục, khoảng giá, tác giả, văn bản tìm kiếm, phân trang và sắp xếp mà vẫn duy trì cú pháp súc tích, dễ đọc và tái sử dụng ở nhiều điểm truy vấn.

## Lợi ích tổng hợp

Việc kết hợp các mẫu thiết kế trên mang lại bốn nhóm lợi ích chính. Thứ nhất, chất lượng mã nguồn được nâng cao nhờ cấu trúc rõ ràng, tránh lặp mã và tuân thủ nguyên tắc SOLID. Thứ hai, khả năng mở rộng được bảo đảm khi từng thành phần có thể phát triển độc lập mà không ảnh hưởng luồng chung. Thứ ba, hiệu quả cộng tác được cải thiện vì mỗi nhóm phụ trách rõ ràng một lớp, giảm xung đột merge và rút ngắn thời gian onboarding. Cuối cùng, khả năng kiểm thử tăng lên đáng kể khi từng lớp, từng phương thức có thể giả lập (mock) và đánh giá riêng biệt.

## Thực hành best practices

Trong suốt quá trình hiện thực, nhóm phát triển duy trì các thông lệ: SOLID, DRY, KISS, YAGNI và Separation of Concerns. Đồng thời, cơ chế Dependency Injection được khuyến khích để giảm phụ thuộc cứng, tạo nên hệ thống lỏng (loosely coupled) nhưng bền vững.

## Định hướng phát triển tiếp theo

Để đáp ứng nhu cầu mở rộng tính năng, nhóm đề xuất nghiên cứu bổ sung Command (hoàn tác/redo), Decorator (bật/tắt tính năng động), Adapter (tương thích dịch vụ bên thứ ba), Facade (đơn giản hóa giao tiếp với subsystem phức tạp) và Proxy (caching, bảo mật). Về cấp độ kiến trúc, định hướng chuyển sang Microservices, Event Sourcing, CQRS và Hexagonal Architecture sẽ giúp hệ thống linh hoạt, dễ triển khai quy mô lớn và hỗ trợ khối lượng giao dịch cao trong tương lai.

# Đánh giá kiểm thử

* 1. **UC1 – Quản lý thông tin hệ thống**

Bao gồm:

**UC1.1 – Quản lý người dùng**

**UC1.2 – Quản lý sách**

**UC1.1 – Quản lý người dùng**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Case Description | | Test Procedures | Test Case Input Value | Test Case Expected Result | Status |
| UC1.1-TC01 | Thêm người dùng hợp lệ | | 1. Nhấn 'Thêm người dùng' 2. Nhập tên, email, mật khẩu, sđt 3. Nhấn 'Lưu' | 'Nguyễn Văn A', 'a@gmail.com', 'abc123', '0337813143' | Người dùng được thêm vào danh sách | Pass |
| UC1.1-TC02 | Thêm người dùng thiếu email | | 1. Nhấn 'Thêm người dùng' 2. Bỏ trống email 3. Nhấn 'Lưu' | 'Nguyễn Văn B', \_\_\_, 'abc123', '0123286733' | Hiển thị lỗi: 'Vui lòng nhập email' | Pass |
| **UC1.1-TC03** | | Thêm người dùng thiếu số điện thoại/nhập không đủ 10 số | 1. Nhấn 'Thêm người dùng' 2. Bỏ trống SĐT/Không nhập đủ 10 số 3. Nhấn 'Lưu' | ‘'Nguyễn Văn C', 'c@gmail.com',, 'abc123', \_\_\_\_\_ | Hiển thị lỗi: 'Vui lòng nhập số điện thoại'/'Vui lòng nhập đủ 10 số ' | Pass |
| **UC1.1-TC04** | | Thêm người dùng thiếu mật khẩu | 1. Nhấn 'Thêm người dùng' 2. Bỏ trống mật khẩu 3. Nhấn 'Lưu' | 'Nguyễn Văn A', 'a@gmail.com', '\_\_\_\_\_', '0337813143' | Hiển thị lỗi: 'Vui lòng nhập mật khẩu' | Pass |
| **UC1.1-TC05** | | Thêm email nhưng nhập sai vd: thiếu.com/thiếu@ | 1. Nhấn 'Thêm người dùng' 2. Bỏ trống email 3. Nhấn 'Lưu' | 'Nguyễn Văn A', 'a@gmail/agmail.com', 'abc123', '0337813143' | Hiển thị lỗi: 'Vui lòng nhập lại email' | Pass |
| **UC1.1-TC06** | | Cập nhật người dùng | 1. Nhấn 'Cập nhật’ 2. Cập nhật tên và số điện thoại 3. Nhấn 'Lưu' | 'Nguyễn Văn D', '0123456789' | Cập nhật người dùng thành công | Pass |

**UC1.2 – Quản lý sách**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Test Case ID** | **Test Case Description** | **Test Procedures** | **Test Case Input Value** | **Test Case Expected Result** | **Status** | | **TC\_Book\_01** | Thêm sách hợp lệ | 1. Nhấn “Thêm mới” 2. Nhập đầy đủ các trường 3. Nhấn “Tạo mới” | Tên: "Doraemon", Tác giả: "Fujiko", Giá: 50000, Thể loại: Truyện tranh, Số lượng: 100, Đã bán: 0, Ảnh: ✔️ | Sách được thêm vào danh sách, hiển thị trong bảng | Pass | | **TC\_Book\_02** | Thiếu tên sách | Không nhập tên sách | Tác giả: "Fujiko", Giá: 50000... | Hiển thị lỗi “Vui lòng nhập tên sách” | Pass | | **TC\_Book\_03** | Giá sách không hợp lệ | Nhập giá là "abc" | Giá: "abc" | Hiển thị lỗi “Giá tiền phải là số” | Pass | | **TC\_Book\_04** | Thiếu ảnh Thumbnail | Không chọn ảnh thumbnail | Tên, Tác giả, Giá hợp lệ | Hiển thị lỗi “Vui lòng chọn ảnh Thumbnail” | Pass | | **TC\_Book\_05** | Cập nhật thành công | 1. Chọn sách đã có 2. Nhấn “Cập nhật” 3. Sửa tên, giá, số lượng 4. Nhấn “Cập nhật” | Giá: 648000 → 600000 | Dữ liệu sách được cập nhật thành công trong danh sách | Pass | | **TC\_Book\_06** | Xóa ảnh slider và lưu | Xóa hết ảnh slider trước khi cập nhật | Ảnh slider: rỗng | Hiển thị cảnh báo hoặc chặn lưu nếu bắt buộc | Pass | | **TC\_Book\_07** | Xác nhận và xóa | 1. Nhấn nút ‘Xóa’ của 1 dòng sách 2. Chọn “Xác nhận” | — | Sách bị xóa khỏi danh sách, không còn trong DB | Pass | | **TC\_Book\_08** | Hủy thao tác xóa | Nhấn nút ‘Xóa’, sau đó nhấn “Hủy” trong thông báo | — | Không có thay đổi nào, sách vẫn còn | Pass | |  |  |  |  |  |

* 1. **UC2: Quản lý đơn hàng và danh mục**

**UC2.1- Quản lý Đơn hàng**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Case Description | Test Procedures | Test Case Input Value | Test Case Expected Result | Status |
| TC\_Order\_01 | Tạo đơn hàng | 1. Cho sản phẩm vào giỏ 2. Nhập địa chỉ và số điện thoại 3. Nhấn 'Đặt hàng' | Sản phẩm hợp lệ, địa chỉ + SDT | Đơn hàng được tạo và hiển thị trong lịch sử | Pass |
| TC\_Order\_02 | Xem lịch sử đơn hàng | 1. Người dùng đăng nhập 2. Nhấn 'Lịch sử đặt hàng' | Tài khoản có đơn hàng | Hiển thị đúng danh sách đơn hàng theo user | Pass |
| TC\_Order\_03 | Xem chi tiết đơn hàng | 1. Từ lịch sử, nhấn 'Chi tiết đơn mua' | Mã đơn hàng hợp lệ | Hiển thị thông tin đầy đủ đơn hàng | Pass |
| TC\_Order\_04 | Xem danh sách đơn hàng (Admin) | 1. Admin vào mục 'Manage Orders' | — | Hiển thị toàn bộ đơn hàng trong hệ thống | Pass |
| TC\_Order\_05 | Đồng bộ dữ liệu đơn hàng | So sánh đơn hàng giữa bảng người dùng và bảng Admin | Cùng một mã đơn | Thông tin trùng khớp (thời gian, tiền, trạng thái...) | Pass |
| TC\_Order\_06 | Tìm kiếm đơn hàng | 1. Nhập tên khách hoặc địa chỉ trong ô tìm kiếm 2. Nhấn 'Search' | Nguyễn Cao Chiến | Danh sách đơn hàng tương ứng hiển thị | Pass |

# Triển khai hệ thống

Hệ thống bán sách trực tuyến được triển khai theo mô hình client–server sử dụng Node.js 20.9 cho backend, MongoDB 6.0 cho cơ sở dữ liệu và React + Vite cho frontend. Ở môi trường phát triển cục bộ, lập trình viên chỉ cần máy tính hai nhân, 4 GB RAM, 2 GB ổ cứng trống và cài đặt sẵn Git cùng VS Code. Sau khi nhân bản mã nguồn, nhà phát triển cài đặt các gói phụ thuộc bằng *npm install*, cấu hình biến môi trường trong tập tin *.env*, khởi động dịch vụ MongoDB rồi chạy ứng dụng với lệnh *npm run dev*. Khi đóng gói sản phẩm, mã nguồn được biên dịch, tối ưu và phục vụ ở chế độ production thông qua npm run build kết hợp *npm start*.

Ở phía người dùng, giao diện được khởi chạy tại http://localhost:5173 và giao tiếp với API ở cổng 3001. Trang chủ hiển thị danh mục sách, khuyến mãi và giỏ hàng theo thời gian thực; trang chi tiết sách cung cấp hình ảnh, mô tả, đánh giá cùng chức năng thêm vào giỏ; trang quản trị tập trung thống kê doanh thu, đơn hàng, kho sách và tài khoản. Các ảnh chụp màn hình và video minh hoạ quy trình đặt hàng được lưu trong thư mục /public.

Tại môi trường sản xuất, bộ phận vận hành triển khai backend lên Heroku hoặc máy VPS (Ubuntu 22.04) thông qua PM2 để giám sát tiến trình, trong khi frontend được phân phối qua Vercel nhằm tận dụng CDN toàn cầu. Cơ sở dữ liệu chuyển sang MongoDB Atlas để bảo đảm độ sẵn sàng 99,9 %. Kiến trúc lai dev–staging–prod giúp đội ngũ kiểm thử trên dữ liệu thật mà vẫn bảo toàn an toàn hệ thống.

# Kết luận và hướng phát triển

Sau giai đoạn phát triển, dự án đã hoàn thiện toàn bộ API RESTful, thiết kế cơ sở dữ liệu tối ưu và giao diện web đáp ứng trên mọi thiết bị. Hệ thống đạt thời gian phản hồi 50 – 200 ms, xử lý đồng thời 100 người dùng mà không quá tải, đồng thời đạt tỷ lệ hoạt động 99,9 % trên môi trường thử nghiệm. Bên cạnh đó, chất lượng mã nguồn được bảo chứng bởi 100 % TypeScript, bộ quy tắc ESLint/Prettier và tài liệu OpenAPI đầy đủ.

Trong quá trình triển khai, nhóm đã giải quyết thành công các thách thức về kết nối MongoDB, giới hạn kích thước tệp tải lên, độ phức tạp quản lý trạng thái React và nhất quán hoá thông điệp lỗi API. Việc bổ sung cơ chế cập nhật atom với findOneAndUpdate, text-index và tree-shaking giúp giảm thời gian tìm kiếm còn 200 ms và rút gọn gói frontend 40 %.

Dự kiến sáu tháng tới, hệ thống sẽ bổ sung danh sách yêu thích, bộ lọc nâng cao, nhiều cổng thanh toán nội địa, quản lý khuyến mãi linh hoạt và công cụ phân tích nâng cao cho quản trị viên. Trong vòng một năm, nhóm định hướng xây dựng design system kèm chế độ tối, tối ưu hoá bộ nhớ đệm bằng Redis/CDN và cân nhắc chuyển sang Next.js để cải thiện SEO. Về dài hạn, dự án hướng tới kiến trúc microservices trên Kubernetes, tích hợp AI đề xuất sách, định giá động, chatbot hỗ trợ và mở rộng sang nền tảng di động qua React Native.

# Tài liệu tham khảo

* **Clean Architecture** – Robert C. Martin. O’Reilly Media, 2017.
* **System Design Interview – An Insider’s Guide** – Alex Xu. Byte Byte Go, 2022.
* **Building Microservices (2nd Ed.)** – Sam Newman. O’Reilly Media, 2021.
* **Node.js Design Patterns (3rd Ed.)** – Mario Casciaro & Luciano Mammino. Packt Publishing, 2020.
* **MongoDB: The Definitive Guide (3rd Ed.)** – Kristina Chodorow. O’Reilly Media, 2019.
* **React: Up & Running** – Stoyan Stefanov. O’Reilly Media, 2016.
* **Redux in Action** – Marc Garreau & Will Faurot. Manning Publications, 2018.
* **The Complete Node.js Developer Course** – Andrew Mead. Udemy video course, 2023.
* **React – The Complete Guide** – Maximilian Schwarzmüller. Udemy video course, 2024.
* **MongoDB University M320: Data Modeling**. MongoDB University, 2024.
* **System Design for Interviews and Beyond** – Educative.io interactive course, 2023.
* **Advanced CSS & Sass** – Jonas Schmedtmann. Udemy video course, 2022.
* **React Official Documentation** – Meta Open Source. <https://react.dev/>
* **Express.js Guide** – Express.js Foundation. <https://expressjs.com/>
* **MongoDB Manual** – MongoDB Inc. <https://docs.mongodb.com/manual/>
* **TypeScript Handbook** – Microsoft. <https://www.typescriptlang.org/docs/>
* **Microsoft REST API Guidelines** – Microsoft, 2023. <https://github.com/microsoft/api-guidelines>
* **OWASP Top 10** – OWASP Foundation, 2021. <https://owasp.org/www-project-top-ten/>
* **Ant Design (v5.3.2) Documentation** – Ant Group. <https://ant.design/>
* **GitHub Repository: nodebestpractices** – Yoni Goldberg et al. <https://github.com/goldbergyoni/nodebestpractices>