

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG



ĐỒ ÁN TỐT NGHIỆP

NGÀNH CÔNG NGHỆ THÔNG TIN

Sinh viên : Hoàng Duy Thành

Giảng viên hướng dẫn: Ths.Vũ Phạm Minh Hiền

Hải Phòng - 2025

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG

XÂY DỰNG TRANG QUẢN LÝ THƯ VIỆN SÁCH HPU

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
NGÀNH CÔNG NGHỆ THÔNG TIN

Sinh viên thực hiện : Hoàng Duy Thành

Giảng viên hướng dẫn: Ths.Vũ Phạm Minh Hiễn

Hải Phòng - 2025

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG

NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP

Sinh viên : Hoàng Duy Thành - Mã SV: 2212111027
Lớp : CT2601
Ngành : Công nghệ thông tin
Tên đề tài: Xây dựng trang Quản lý Thư viện Sách HPU

NHIỆM VỤ ĐỀ TÀI

1. Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp

a. Mô tả tóm tắt đề tài

- Xây dựng website quản lý hoạt động của một thư viện sách, bao gồm quản lý sách, danh mục, thành viên, quá trình mượn-trả sách và các thông kê báo cáo liên quan.

b. Nội dung hướng dẫn

- Khảo sát và Phân tích yêu cầu:

- Xác định rõ các nhóm người dùng (Thủ thư, học sinh) và các chức năng cần thiết cho từng nhóm.
- Nghiên cứu các công nghệ phù hợp để phát triển hệ thống (VD: Ngôn ngữ lập trình, Framework, Cơ sở dữ liệu).

- Thiết kế hệ thống:

- Thiết kế kiến trúc tổng thể cho website.
- Thiết kế cơ sở dữ liệu
- Thiết kế giao diện người dùng cho các trang chức năng chính.

c. Kết quả cần đạt được

- Nắm được kiến thức về xây dựng website cũng như ngôn ngữ lập trình web.

- Xây dựng một trang web mang lại thuận tiện cho người dùng, tiết kiệm thời gian cũng như công sức.

2. Địa điểm thực tập tốt nghiệp

- Công ty TNHH máy tính Đông Phong

CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Họ và tên : Vũ Phạm Minh Hiền

Học hàm, học vị : Thạc sĩ

Cơ quan công tác : Trường Đại học Quản lý và Công nghệ Hải Phòng

Nội dung hướng dẫn:

Khảo sát và Phân tích yêu cầu:

- Xác định rõ các nhóm người dùng (Thủ thư, học sinh) và các chức năng cần thiết cho từng nhóm.
- Nghiên cứu các công nghệ phù hợp để phát triển hệ thống (VD: Ngôn ngữ lập trình, Framework, Cơ sở dữ liệu).

- Thiết kế hệ thống:

- Thiết kế kiến trúc tổng thể cho website.
- Thiết kế cơ sở dữ liệu
- Thiết kế giao diện người dùng cho các trang chức năng chính.

Đề tài tốt nghiệp được giao ngày 06 tháng 10 năm 2025

Yêu cầu phải hoàn thành xong trước ngày 27 tháng 12 năm 2025

Đã nhận nhiệm vụ ĐTTN

Sinh viên

Đã giao nhiệm vụ ĐTTN

Giảng viên hướng dẫn

Hải Phòng, ngày tháng 12 năm 2025

TRƯỞNG KHOA

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập - Tự do - Hạnh phúc

PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN TỐT NGHIỆP

Họ và tên giảng viên: Vũ Phạm Minh Hiền

Đơn vị công tác: Khoa Công nghệ thông tin - Trường Đại học Quản Lý và Công Nghệ Hải Phòng.

Họ và tên sinh viên : Hoàng Duy Thành

Ngành: Công nghệ Thông tin

Nội dung hướng dẫn : Toàn bộ đề tài

1. Tinh thần thái độ của sinh viên trong quá trình làm đề tài tốt nghiệp

2. Đánh Giá mức độ chất lượng của đề án/khóa luận (so với nội dung yêu cầu đã đề ra trong nhiệm vụ Đ.T.T.N, trên các mặt lý luận, thực tiễn, tính toán số liệu...)

3. Ý kiến của giảng viên hướng dẫn tốt nghiệp

Được bảo vệ

Không được bảo vệ

Điểm hướng dẫn

Hải Phòng, ngày tháng năm 2025

Giảng viên hướng dẫn

(ký và ghi rõ họ tên)

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập - Tự do - Hạnh phúc

PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN CHĂM PHẢN BIỆN

Họ và tên giảng viên: Vũ Trọng Chiến

Đơn vị công tác: Trường Đại học Quản lý và Công nghệ Hải Phòng

Họ và tên sinh viên: Hoàng Duy Thành

Ngành: Công nghệ Thông tin

Đề tài tốt nghiệp: Xây dựng trang Quản lý thư viện Sách HPU.

Phần nhận xét của giảng viên chăm phản biện

- Đồ án đáp ứng yêu cầu với phân tích rõ ràng về thiết kế website quản lý thư viện với các chức năng chính: đăng nhập, quản lý sách, danh mục, đơn mượn, thống kê.
- Thiết kế cơ sở dữ liệu và giao diện cơ bản.
- Báo cáo đã thể hiện vận dụng được kiến thức đã học vào các bước phân tích thiết kế.
- Nhiệt tình và có thái độ nghiên cứu khoa học tốt, đã tham gia đề tài nghiên cứu khoa học cấp Khoa và bảo vệ thành công; có báo cáo tại Chương trình “Hải Phòng – Café Khởi nghiệp Sáng tạo – Trạm số 2” do Trường Đại học Quản lý và Công nghệ Hải Phòng phối hợp với Sở Khoa học và Công nghệ thành phố Hải Phòng tổ chức.

Những mặt còn hạn chế

- Giao diện chưa chuyên nghiệp, cần cải tiến UI/UX để tăng trải nghiệm người dùng.
- Chưa khảo sát đầy đủ về nghiệp vụ thư viện.

Ý kiến của giảng viên chăm phản biện

Được bảo vệ Không được bảo vệ Điểm phản biện

Hải Phòng, ngày.....tháng năm 2025

Giảng viên chăm phản biện

(ký và ghi rõ họ tên)

LỜI CẢM ƠN

Lời đầu tiên, em xin được bày tỏ lòng biết ơn chân thành đến Quý thầy cô trong khoa Công nghệ Thông tin của trường Đại học Quản lý và Công nghệ Hải Phòng, không chỉ là những người đã tuyển đạt những kiến thức quý báu mà còn là những người thắp lên ngọn lửa đam mê, truyền cảm hứng và hướng dẫn chúng em trong suốt hành trình khám phá tri thức, ngoài lĩnh vực của công nghệ thông tin mà còn có những kỹ năng mềm và phương pháp tư duy hiệu quả.

Bên cạnh đó, em xin gửi lời cảm ơn sâu sắc đến thầy và thầy Hiền, người đã trực tiếp hướng dẫn và hỗ trợ em trong suốt quá trình thực hiện khóa luận này. Sự tận tâm, kiên nhẫn và còn có những góp ý quý báu của thầy đã giải đáp những thắc mắc, khó khăn để có thể hoàn thiện khóa luận với một cách tốt nhất.

Đặc biệt, em muốn gửi lời cảm ơn đến gia đình, mọi người luôn là hậu phương vững chắc. Sự yêu thương và hi sinh thầm lặng của bố mẹ, anh chị em đã luôn là một nguồn sức mạnh mãnh liệt để chúng em vượt qua những thử thách những hoài nghi, những vấp ngã trong suốt quá trình học tập.

Tuy đã cố gắng hết sức và dành nhiều thời gian cho việc nghiên cứu và hoàn thiện bài luận, nhưng vẫn không thể tránh khỏi những thiếu sót và hạn chế trong kiến thức và kinh nghiệm thực tế. Em rất mong có thể nhận được những góp ý chân thành từ quý thầy cô để được hoàn thiện hơn, đồng thời giúp chúng em nâng cao kiến thức và kỹ năng của mình.

Cuối cùng em xin kính gửi lời chúc tới toàn thể các thầy cô trong khoa Công nghệ thông tin cũng như các thầy cô đang giảng dạy trong trường Đại học Quản lý và Công nghệ Hải Phòng luôn luôn mạnh khỏe và tràn đầy nhiệt huyết để truyền đạt kiến thức cho các thế hệ mai sau.

Hải Phòng, ngày ... tháng năm 2025

Sinh viên thực hiện

LỜI CAM ĐOAN

Em cam đoan về nội dung, kết quả bài nghiên cứu “**Xây dựng Trang Quản lý Thư viện Sách HPU**” trong đề án là sách của em tự nghiên cứu, không sao chép của người khác, các nội dung trích dẫn đều ghi đầy đủ thông tin nguồn, tuân thủ Quy định của Luật Bản quyền.

Chúng em xin chịu trách nhiệm trước Nhà trường nếu có bất kì hành vi phạm Pháp luật nào được phát hiện.

Hải Phòng, ngày tháng năm 2025

SINH VIÊN

MỤC LỤC

Nội dung

LỜI CAM ĐOAN	9
A. Lý do chọn đề tài	10
B. Mục tiêu và ý nghĩa nghiên cứu	10
C. Phạm vi và phương pháp nghiên cứu	11
D. Cấu trúc đồ án.....	11
CHƯƠNG 1. CƠ SỞ LÝ THUYẾT	13
1.1. Mô hình Client – Server trong phát triển web	13
1.1.1. Khái niệm về mô hình Client – Server	13
1.1.2. Vai trò và chức năng của Client	13
1.1.3. Vai trò và chức năng của Server.....	14
1.1.4. Nguyên lý hoạt động	14
1.1.5. Ưu điểm của mô hình Client – Server.....	15
1.2. Front – End	15
1.2.1. React.js.....	15
1.2.2. Vite	16
1.2.3. React Router.....	18
1.2.4. SCSS Modules	19
1.2.5. Context API.....	20
1.3. Back – End.....	21
1.3.1. Node.js.....	21
1.3.2. Express.js	22
1.3.3. Mô hình phát triển dự án MVC.....	23
1.3.4. Middleware	24
1.3.5. Kết nối Frontend – Backend	25
1.4. Cơ sở dữ liệu MongoDB.....	26

1.4.1. Giới thiệu MongoDB.....	26
1.4.2. Đặc điểm nổi bật.....	26
1.4.3. Vai trò của MongoDB trong đồ án	27
1.5. Kiểm thử	27
1.5.1. Giới thiệu.....	27
1.5.2. Kiểm thử hộp đen	28
CHƯƠNG 2. PHÂN TÍCH ĐẶC TẢ CHỨC NĂNG CỦA HỆ THỐNG.....	30
2.1. Kiến trúc tổng thể	30
2.2. Yêu cầu hệ thống	30
2.2.1. Yêu cầu chức năng	30
2.2.2. Yêu cầu phi chức năng.....	31
2.3. Usecase	32
2.3.1. Tác nhân hệ thống.....	32
2.3.2. Biểu đồ usecase tổng quát.....	33
2.3.3. Biểu đồ usecase phân rã	34
2.4. Đặc tả chức năng	46
2.4.1. Use Case: Đăng nhập	46
2.4.2. Use Case: Đăng ký.....	47
2.4.3. Use Case: Đăng xuất	49
2.4.4. Use Case: Tìm kiếm sách.....	50
2.4.5. Use Case: Chi tiết sách.....	51
2.4.6. Use Case: Giỏ sách.....	53
2.4.7. Use Case: Đặt mượn và Thanh toán phí mượn	54
2.4.8. Use Case: Chi tiết đơn mượn.....	56
2.4.9. Use Case: Cập nhật tài khoản	57
2.4.10. Use Case: Thống kê	59
2.4.11. Use Case: Quản lý sách.....	60
2.4.12. Use Case: Quản lý danh mục sách	62

2.4.13.	<i>Use Case: Quản lý đơn mượn</i>	64
2.4.14.	<i>Use Case: Quản lý User</i>	65
CHƯƠNG 3.	THIẾT KẾ CHỨC NĂNG VÀ CƠ SỞ DỮ LIỆU	67
3.1.	Biểu đồ tuần tự.....	67
3.1.1.	<i>Biểu đồ tuần tự chức năng Đăng nhập</i>	67
3.1.2.	<i>Biểu đồ tuần tự chức năng Đăng ký</i>	68
3.1.3.	<i>Biểu đồ tuần tự chức năng Đăng xuất</i>	69
3.1.4.	<i>Biểu đồ tuần tự chức năng Tìm kiếm sách</i>	69
3.1.5.	<i>Biểu đồ tuần tự chức năng Chi tiết sách và Thêm vào giỏ sách</i>	70
3.1.6.	<i>Biểu đồ tuần tự chức năng Giỏ sách</i>	72
3.1.7.	<i>Biểu đồ tuần tự chức năng Đặt mượn</i>	74
3.1.8.	<i>Biểu đồ tuần tự chức năng Thanh toán phí mượn</i>	75
3.1.9.	<i>Biểu đồ tuần tự chức năng Xem chi tiết đơn mượn</i>	76
3.1.10.	<i>Biểu đồ tuần tự chức năng Cập nhật thông tin cá nhân</i>	77
3.1.11.	<i>Biểu đồ tuần tự chức năng Quản lý sách</i>	78
3.1.12.	<i>Biểu đồ tuần tự chức năng Quản lý danh mục</i>	81
3.1.13.	<i>Biểu đồ tuần tự chức năng Quản lý đơn mượn</i>	82
3.1.14.	<i>Biểu đồ tuần tự chức năng Quản lý độc giả</i>	83
3.1.15.	<i>Biểu đồ tuần tự chức năng Quản lý thông tin liên hệ</i>	84
3.1.16.	<i>Biểu đồ tuần tự chức năng Thống kê</i>	85
3.2.	Thiết kế cơ sở dữ liệu	86
3.2.1.	<i>Lược đồ cơ sở dữ liệu</i>	86
3.2.2.	<i>Danh sách các bảng</i>	86
3.2.3.	<i>Chi tiết các bảng</i>	87
CHƯƠNG 4.	KIỂM THỬ VÀ TRIỂN KHAI WEBSITE	91
4.1.	Kiểm thử	91
4.1.1.	<i>Phân tích các trường hợp kiểm thử</i>	91
4.1.2.	<i>Kết quả thực thi Test</i>	93

4.2. Yêu cầu về cài đặt.....	94
4.2.1. Yêu cầu phần cứng:.....	94
4.2.2. Yêu cầu mạng Internet:	94
4.3. Kết quả cài đặt môi trường	95
4.3.1. Data Access Layer (MongoDB)	95
4.3.2. Bussiness Logic Layer (Express)	96
4.3.3. Presentation Layer (ReactJS)	97
4.4. Kết quả thực nghiệm.....	98
4.4.1. Giao diện đăng nhập Website	98
4.4.2. Giao diện trang chủ Website.....	98
4.4.3. Giao diện chi tiết sách	99
4.4.4. Giao diện giỏ sách	99
4.4.5. Giao diện chi tiết đơn mượn	100
Giao diện cập nhật tài khoản.....	101
4.4.6. Giao diện thống kê	101
4.4.7. Giao diện quản lý sách.....	102
4.4.8. Giao diện quản lý danh mục	103
4.4.9. Giao diện quản lý đơn mượn.....	104
4.4.10. Giao diện quản lý User	105
4.4.11. Giao diện quản lý thẻ mượn.....	105
CHƯƠNG 5. KẾT LUẬN.....	106
5.1. Kết quả đạt được.....	106
5.2. Hạn chế	106
5.3. Hướng phát triển.....	106
TÀI LIỆU THAM KHẢO.....	108

DANH MỤC TỪ VIẾT TẮT VÀ THUẬT NGỮ

STT	Ký hiệu chữ viết tắt	Chữ viết đầy đủ	Nghĩa Tiếng Việt
1	HPU	Hai Phong Private University	Trường Đại học Quản lý và Công nghệ Hải Phòng
2	HTTP	Hypertext Transfer Protocol	Giao thức truyền tải siêu văn bản
3	HTTPS	Hypertext Transfer Protocol Secure	Giao thức truyền tải siêu văn bản bảo mật
4	JSON	JavaScript Object Notation	Định dạng dữ liệu dựa trên JavaScript
5	UI	User Interface	Giao diện độc giả
6	UX	User Experience	Trải nghiệm độc giả
7	UC	Use Case	Trường hợp sử dụng
8	SPA	Single Page Application	Ứng dụng đơn trang
9	MVC	Model – View – Controller	Dữ liệu – Giao Diện – Điều Khiển
10	DOM	Document Object Model	Mô hình đối tượng tài liệu
11	JSX	JavaScript XML	Cú pháp viết tắt HTML
12	SCSS	Sassy CSS	Phân mở rộng CSS, dùng để viết style
13	API	Application Programming Interface	Giao diện lập trình ứng dụng
14	JWT	Json Web Token	Chuỗi mã hóa để xác thực, bảo mật thông tin độc giả
15	HMR	Hot Module Replacement	Tính năng thay thế mô đun nóng, dùng để cập

			nhật giao diện mà ko cần tải lại trang
16	ESM	ECMAScript Module	Hệ thống module tiêu chuẩn của JavaScript
17	RDBMS	Relational Database Management System	Hệ quản trị cơ sở dữ liệu
18	SQL	Structured Query Language	Ngôn ngữ truy vấn cấu trúc, dùng để quản lý cơ sở dữ liệu trong RDBMS
19	CRUD	Create, Read, Update, Delete	Tạo, đọc, cập nhật, xóa
20	Vscode	Visual Studio Code	Phần mềm lập trình
21	IDE	Intergrated Development Enviroment	Môi trường

DANH MỤC HÌNH ẢNH

Hình 1.1: Mô hình Client – Server	13
Hình 1.2: Mô hình MVC	23
Hình 2.1: Sơ đồ kiến trúc hệ thống.....	30
Hình 2.2: Biểu đồ Use Case tổng quát	33
Hình 2.3 Biểu đồ phân rã Use Case Giỏ sách	34
Hình 2.4: Biểu đồ phân rã Use Case Đặt mượn và Thanh toán phí mượn phí mượn	35
Hình 2.5: Biểu đồ phân rã Use Case Chi tiết đơn mượn	36
Hình 2.6: Biểu đồ phân rã Use Case Cập nhật thông tin.....	37
Hình 2.7: Biểu đồ phân rã Use Case Tìm kiếm sách.....	38
Hình 2.8: Biểu đồ phân rã Use Case Chi tiết sách	39
Hình 2.9: Biểu đồ phân rã Use Case Quản lý sách	40
Hình 2.10: Biểu đồ phân rã Use Case Quản lý đơn mượn	41
Hình 2.11: Biểu đồ phân rã Use Case Quản lý độc giả.....	42
Hình 2.12: Biểu đồ phân rã Use Case Quản lý danh mục sách.....	43
Hình 2.13: Biểu đồ phân rã Use Case Quản lý liên hệ.....	44
Hình 2.14: Biểu đồ phân rã Use Case Thống kê	45
Hình 3.1: Biểu đồ tuần tự chức năng Đăng nhập	67
Hình 3.2: Biểu đồ tuần tự chức năng Đăng ký	68
Hình 3.3: Biểu đồ tuần tự chức năng Tìm kiếm sách.....	69
Hình 3.4: Biểu đồ tuần tự chức năng Chi tiết sách.....	71
Hình 3.5: Biểu đồ tuần tự chức năng Thêm sách vào giỏ sách	72
Hình 3.6: Biểu đồ tuần tự chức năng Xóa sách khỏi giỏ sách	73
Hình 3.7: Biểu đồ chức năng Đặt mượn.....	74
Hình 3.8: Biểu đồ tuần tự chức năng Thanh toán phí mượn	75
Hình 3.9: Biểu đồ tuần tự chức năng Xem chi tiết đơn mượn	76
Hình 3.10: Biểu đồ tuần tự chức năng Cập nhật thông tin cá nhân.....	77
Hình 3.11: Biểu đồ tuần tự chức năng Thêm mới sách.....	78

Hình 3.12: Biểu đồ tuần tự chức năng Chính sửa sách	79
Hình 3.13: Biểu đồ tuần tự chức năng Xóa sách.....	80
Hình 3.14: Biểu đồ tuần tự chức năng Quản lý danh mục	81
Hình 3.15: Biểu đồ tuần tự chức năng Quản lý đơn mượn	82
Hình 3.17: Biểu đồ tuần tự chức năng Quản lý thông tin liên hệ.....	84
Hình 3.18: Biểu đồ tuần tự chức năng Thống kê	85
Hình 3.19: Lược đồ cơ sở dữ liệu.....	86
Hình 4.1: Cài đặt cơ sở dữ liệu.....	95
Hình 4.2: Cài đặt Bussiness Logic Layer (Express).....	96
Hình 4.3: Cài đặt Presentation Layer (ReactJS).....	97
Hình 4.4: Giao diện Chức năng đăng nhập	98
Hình 4.5: Giao diện Trang chủ màn hình khi đã đăng nhập	98
Hình 4.6: Giao diện Chi tiết sách	99
Hình 4.7: Giao diện Giỏ sách	99
Hình 4.8: Giao diện Danh sách đơn mượn đã đặt gần đây.....	100
Hình 4.9: Giao diện Chi tiết từng đơn mượn	100
Hình 4.10: Giao diện Cập nhật tài khoản	101
Hình 4.11: Giao diện Thống kê	101
Hình 4.12: Giao diện Quản lý sách	102
Hình 4.13: Giao diện Thêm mới sách	102
Hình 4.14: Giao diện Danh sách danh mục.....	103
Hình 4.15: Giao diện Thêm mới danh mục	103
Hình 4.16: Giao diện Quản lý danh sách đơn mượn	104
Hình 4.17. Giao diện cập nhật đơn mượn	104
Hình 4.18: Giao diện Quản lý User	105
Hình 4.19: Giao diện Quản lý thẻ mượn	105

DANH MỤC BẢNG

Bảng 1.1: Ưu nhược điểm của kiểm thử hộp đen.....	28
Bảng 2.1: Danh sách tác nhân	32
Bảng 2.2: UC Đăng nhập	47
Bảng 2.3: UC Đăng ký	49
Bảng 2.4: UC Đăng xuất	50
Bảng 2.5: UC Tìm kiếm sách	51
Bảng 2.6: UC Chi tiết sách.....	53
Bảng 2.7: UC Giỏ sách.....	54
Bảng 2.8: UC Đặt mượn và Thanh toán phí mượn	56
Bảng 2.9: UC Chi tiết đơn mượn	57
Bảng 2.10: UC Cập nhật tài khoản.....	59
Bảng 2.11: UC Thống kê.....	59
Bảng 2.12: UC Quản lý sách.....	62
Bảng 2.13: UC Quản lý danh mục	63
Bảng 2.14: UC Quản lý đơn mượn.....	65
Bảng 2.15: UC Quản lý User.....	66
Bảng 3.1. Danh sách các bảng dữ liệu	86
Bảng 3.2. Bảng User.....	87
Bảng 3.3. Bảng Products	88
Bảng 3.4. Bảng ApiKey	88
Bảng 3.5. Bảng Category	88
Bảng 3.6. Bảng Cart	89
Bảng 3.7. Bảng Otp	89
Bảng 3.8. Bảng Payments.....	90
Bảng 4.1. Phân tích các trường hợp kiểm thử.....	93
Bảng 4.2. Bảng kết quả thực thi Test	93

Lời mở đầu

A. Lý do chọn đề tài

Trong bối cảnh công nghệ thông tin phát triển mạnh mẽ như hiện nay, việc ứng dụng công nghệ vào công tác quản lý là xu thế tất yếu, đặc biệt đối với các cơ sở Phí mượn sách như trường Đại học. Thư viện Trường Đại học Quản lý và Công nghệ Hải Phòng là trung tâm tri thức nơi cung cấp nguồn tài liệu học tập, nghiên cứu phong phú. Tuy nhiên, các phương pháp quản lý thủ công truyền thống (sổ sách, thẻ mượn...) đã bộc lộ nhiều hạn chế về các vấn đề quản lý, tìm kiếm, thống kê sách do số lượng sách nhiều dễ gây ra sai sót, nhầm lẫn trong quá trình nhập liệu.

Xuất phát từ những thực tế trên, việc xây dựng một hệ thống Quản lý thư viện sách dưới dạng website là giải pháp tối ưu. Một website quản lý sách sẽ cho phép cán bộ thư viện và độc giả tương tác với hệ thống một cách linh hoạt qua trình duyệt, mang lại sự tiện lợi, minh bạch và hiệu quả cao trong công tác quản lý. Công nghệ web hiện đại, với khả năng tích hợp các tính năng như tìm kiếm thông minh, Thanh toán phí mượn trực tuyến và quản lý sách sẽ đem lại sự tiện lợi cho việc quản lý cũng như tìm kiếm thông tin. Việc ứng dụng các công nghệ này vào phát triển website quản lý hứa hẹn mang lại sự tiện lợi vượt trội, giúp độc giả dễ dàng tra cứu thông tin sách, đặt mượn và Thanh toán phí mượn chỉ bằng vài thao tác đơn giản.

Chính vì lý do này, em quyết định lựa chọn đề tài "Xây dựng trang Quản lý Thư viện Sách HPU" làm đề án tốt nghiệp, nhằm nâng cao hiệu quả quản lý thư viện, cải thiện chất lượng hỗ trợ sinh viên, góp phần vào sự phát triển chung của Nhà trường .

B. Mục tiêu và ý nghĩa nghiên cứu

Mục tiêu của đề án là thiết kế, xây dựng và triển khai thành công một hệ thống quản lý thư viện dưới dạng website, có khả năng:

- Tự động hóa các nghiệp vụ chính của thư viện sách bao gồm quản lý đầu sách, thành viên, quá trình mượn trả sách và các thống kê báo cáo liên quan.
- Cung cấp giao diện quản trị thân thiện, đẹp mắt, dễ sử dụng cho nhân viên thư viện và sinh viên.
- Xây dựng giao diện tra cứu tiện lợi cho Độc giả, cho phép ra cứu trạng thái sách theo thời gian thực.
- Đảm bảo tính bảo mật, toàn vẹn dữ liệu.

Ý nghĩa nghiên cứu:

- Đối với Nhà trường/Thư viện HPU: Cung cấp công cụ quản lý chuyên nghiệp, giúp tiết kiệm thời gian, nhân lực, tăng cường độ chính xác trong kiểm soát tài sản và nâng cao chất lượng phục vụ sinh viên.
- Đối với cá nhân em: Việc làm đồ án giúp em vận dụng được kiến thức lý thuyết về Phân tích thiết kế Hệ thống Thông tin và các công nghệ lập trình web hiện đại (Client-Server, Node.js, Express, MongoDB...).

C. Phạm vi và phương pháp nghiên cứu

Phạm vi nghiên cứu:

- Phần mềm được phát triển trên nền tảng web, cung cấp giải pháp toàn diện cho việc quản lý các hoạt động kinh doanh trong lĩnh vực mượn hàng. Với giao diện trực quan và dễ sử dụng, phần mềm cho phép độc giả truy cập dễ dàng và thuận tiện.
- Phạm vi của phần mềm bao gồm các tính năng chính như quản lý sách, quản lý đơn mượn, quản lý tài khoản tìm kiếm và lọc sách, cùng với tích hợp Thanh toán phí mượn an toàn. Ngoài ra, phần mềm còn cung cấp các công cụ báo cáo và thống kê, hỗ trợ Admin trong việc nắm bắt tình hình kinh doanh và đưa ra các quyết định chiến lược một cách chính xác và kịp thời.

Phương pháp nghiên cứu:

- Khảo sát thực tế: Quan sát quy trình nghiệp vụ hiện tại của Thư viện HPU.
- Phân tích hướng đối tượng: Sử dụng các biểu đồ như Use Case, biểu đồ Trình tự, biểu đồ Lớp để xây dựng mô hình một cách chi tiết.
- Lập trình và thử nghiệm: sử dụng các công nghệ đã chọn để lập trình website và tiến hành kiểm thử đơn vị, kiểm thử tích hợp để đảm bảo chất lượng hệ thống.

D. Cấu trúc đồ án

Khóa luận tốt nghiệp xây dựng trang quản lý thư viện HPU với bộ cục 5 chương:

- **Chương 1: Cơ sở lý thuyết:** Đồ án giới thiệu về những công nghệ đã sử dụng liên quan đến các chủ đề client, server, framework, server, database, và một số công nghệ khác.
- **Chương 2: Phân tích, đặc tả chức năng hệ thống:** Đồ án đi sâu vào việc mô tả bài toán, về yêu cầu nghiệp vụ, xác định các tác nhân và các use case.
- **Chương 3: Thiết kế cơ sở dữ liệu và chức năng:** Đồ án đưa ra các lược đồ cơ sở dữ liệu, quan hệ giữa các lớp trong biểu đồ và một số biểu đồ tuần tự.

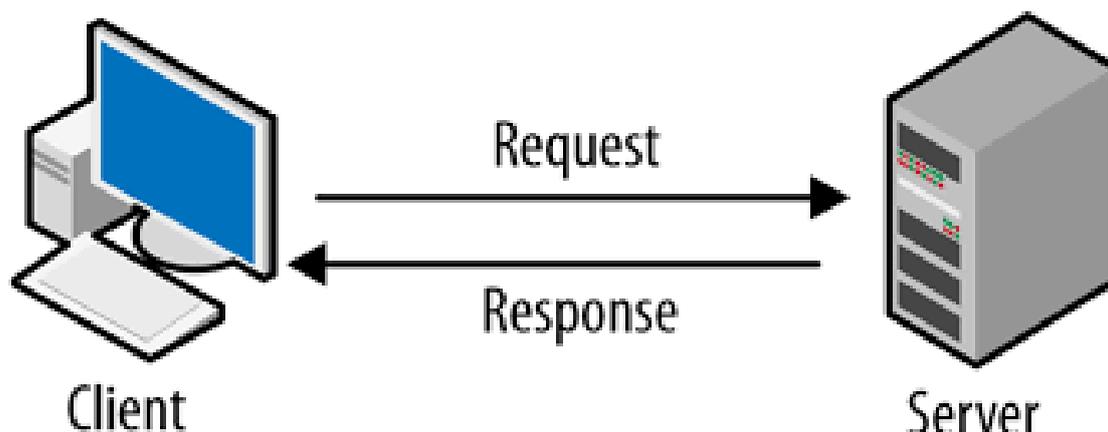
- **Chương 4: Kiểm thử và triển khai hệ thống:** Đồ án đưa ra các trường hợp kiểm thử, kết quả sau khi kiểm thử, các hướng dẫn cài đặt và kết quả thực nghiệm của phần mềm.
- **Chương 5: Kết luận:** Đưa ra các kết quả đạt được, kết quả chưa đạt được và hướng phát triển trong tương lai của phần mềm.

E. Tình hình nghiên cứu trong và ngoài nước

- Tình hình nghiên cứu trong nước: Tại Việt Nam, các hệ thống quản lý thư viện đã được triển khai rộng rãi ở nhiều cấp độ, từ các trường đại học lớn (như VNU, HUT...) đến các thư viện tỉnh/thành phố. Hầu hết các đơn vị đã chuyển đổi sang phần mềm, điển hình là các sách thương mại như Libol hoặc các giải pháp mã nguồn mở. Tuy nhiên, việc phát triển một hệ thống riêng biệt, phù hợp với đặc thù và yêu cầu công nghệ cụ thể của Thư viện HPU, tận dụng các framework công nghệ mới nhất (mô hình Client-Server với ReactJS và Node.js) vẫn là cần thiết để tối ưu hóa hiệu suất và khả năng mở rộng.
- Tình hình nghiên cứu ngoài nước: Trên thế giới, các hệ thống quản lý thư viện đã phát triển với mức độ rất cao với các tính năng tích hợp AI, Machine Learning để tìm kiếm sách và quản lý kho thông minh. Các hệ thống lớn như Koha, Evergreen hay các phần mềm thương mại tiên iên đã chứng minh tính hiệu quả. Việc tham khảo các kiến trúc phần mềm và các chuẩn quản lý thư viện quốc tế (như MARC 21) giúp đề tài có cơ sở lý luận vững chắc trong việc thiết kế các module chức năng đạt chuẩn.

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

1.1. Mô hình Client – Server trong phát triển web



Hình 1.1: Mô hình Client – Server

Trong lĩnh vực phát triển phần mềm nói chung và lập trình web nói riêng, mô hình Client – Server là một kiến trúc cơ bản và phổ biến nhất được sử dụng để xây dựng các ứng dụng web hiện đại. Kiến trúc này phân chia rõ ràng vai trò giữa hai thành phần chính: phía Client (máy khách) và Server (máy chủ), giúp hệ thống hoạt động hiệu quả, linh hoạt và dễ dàng mở rộng. Client (máy khách): Là phía độc giả tương tác trực tiếp, thường là trình duyệt web. Client gửi yêu cầu (request) tới Server thông qua các giao thức như HTTP hoặc HTTPS. Giao diện này có thể được phát triển bằng các công nghệ như HTML, CSS, JavaScript, và các thư viện/framework như React, Vue hoặc Angular.

1.1.1. Khái niệm về mô hình Client – Server

Mô hình Client – Server là một mô hình phân tán, trong đó Client là bên gửi yêu cầu dịch vụ và Server là bên cung cấp dịch vụ đó. Khi độc giả thao tác trên một website (ví dụ như xem sách, đăng nhập, đặt mượn), các hành động đó sẽ được gửi dưới dạng yêu cầu HTTP tới Server. Sau khi xử lý yêu cầu, Server sẽ gửi phản hồi về cho Client, thường dưới dạng dữ liệu JSON hoặc HTML, và dữ liệu này sẽ được Client hiển thị cho độc giả.

Đặc điểm của mô hình này là Client không trực tiếp truy cập vào cơ sở dữ liệu, mà toàn bộ tương tác với dữ liệu đều phải thông qua Server, đảm bảo tính bảo mật và quản lý dữ liệu tập trung.

1.1.2. Vai trò và chức năng của Client

Client là phần giao diện độc giả (UI – User Interface), nơi độc giả tương tác trực tiếp với hệ thống. Trong đồ án này, phần Client được xây dựng bằng React.js – một thư

viện JavaScript mạnh mẽ chuyên dùng để xây dựng các giao diện độc giả tương tác nhanh và mượt mà.

Các chức năng chính của Client bao gồm:

- Hiện thị giao diện thân thiện, dễ sử dụng.
- Gửi các yêu cầu HTTP đến Server thông qua các phương thức như GET, POST, PUT, DELETE.
- Nhận dữ liệu phản hồi từ Server và cập nhật giao diện tương ứng.
- Thực hiện điều hướng trang, quản lý trạng thái giao diện (state), và xử lý sự kiện độc giả.

Với việc sử dụng công nghệ Single Page Application (SPA), ứng dụng React chỉ tải một lần duy nhất và sau đó cập nhật nội dung động mà không cần tải lại toàn bộ trang, mang đến trải nghiệm mượt mà cho độc giả.

1.1.3. Vai trò và chức năng của Server

Server là nơi tiếp nhận, xử lý các yêu cầu từ Client và gửi phản hồi tương ứng. Trong đồ án này, Server được xây dựng bằng Node.js kết hợp với Express.js – một framework mạnh mẽ và linh hoạt cho ứng dụng web.

Các chức năng chính của Server bao gồm:

- Xử lý logic nghiệp vụ (business logic) như đăng nhập, xử lý giỏ sách, đặt mượn...
- Quản lý và tương tác với cơ sở dữ liệu (thêm, sửa, xóa, truy vấn dữ liệu).
- Cung cấp API RESTful để các client có thể dễ dàng giao tiếp.
- Bảo mật hệ thống bằng các phương pháp xác thực (authentication) và phân quyền (authorization).
- Xử lý các tệp tin (uploads), mã hóa thông tin, gửi email, v.v.
- Server thường hoạt động trên một cổng (port) riêng, lắng nghe các yêu cầu đến và phản hồi dữ liệu phù hợp với định dạng JSON hoặc HTML.

1.1.4. Nguyên lý hoạt động

Quy trình hoạt động của mô hình Client – Server có thể mô tả như sau:

- Độc giả thực hiện thao tác (ví dụ: nhấn nút "Thêm vào giỏ sách").
- Giao diện (Client) sẽ tạo một yêu cầu (HTTP request) và gửi đến Server.

- Server nhận yêu cầu, thực hiện xử lý (ví dụ: cập nhật giỏ sách trong cơ sở dữ liệu).
- Sau khi xử lý xong, Server gửi lại một phản hồi (HTTP response).
- Client nhận phản hồi và cập nhật giao diện độc giả.

1.1.5. Ưu điểm của mô hình Client – Server

Tính phân tách rõ ràng: Tách biệt giao diện và xử lý logic, dễ bảo trì và nâng cấp

Tái sử dụng cao: Backend có thể dùng chung cho nhiều loại client (web, mobile, desktop).

Dễ mở rộng: Có thể tăng hiệu năng Server hoặc triển khai thêm nhiều Client dễ dàng.

Bảo mật: Dữ liệu được quản lý tập trung tại Server, dễ kiểm soát và bảo vệ.

1.2. Front – End

1.2.1. React.js

Khái niệm

React.js (gọi tắt là React) là một thư viện JavaScript mã nguồn mở được phát triển bởi Facebook, lần đầu được ra mắt vào năm 2013. React được sử dụng để xây dựng giao diện độc giả (UI – User Interface) trong các ứng dụng web và mobile. Thay vì thao tác trực tiếp với DOM truyền thống, React sử dụng một cơ chế gọi là Virtual DOM để tối ưu hóa hiệu suất cập nhật giao diện.

React không phải là một framework toàn diện như Angular hay Vue, mà chỉ tập trung vào “V” (View) trong mô hình MVC (Model – View – Controller), giúp lập trình viên dễ dàng tạo ra các thành phần giao diện tái sử dụng.

Đặc điểm nổi bật của React.js

Virtual DOM: React sử dụng Virtual DOM để theo dõi sự thay đổi giao diện và chỉ cập nhật những phần cần thiết trên DOM thật, giúp tăng hiệu suất và tốc độ phản hồi của ứng dụng.

Component-Based Architecture (Kiến trúc thành phần): Giao diện trong React được chia nhỏ thành các thành phần (component) riêng biệt và độc lập, giúp tái sử dụng và dễ bảo trì.

JSX (JavaScript XML): Cho phép viết HTML bên trong JavaScript, giúp code dễ đọc, dễ viết và dễ hiểu hơn.

One-Way Data Binding (Ràng buộc dữ liệu một chiều): Dữ liệu được truyền từ component cha xuống component con, tạo ra luồng dữ liệu rõ ràng và dễ kiểm soát.

Hooks (Từ phiên bản 16.8): React cung cấp các hook như useState, useEffect, useContext... cho phép sử dụng state và các tính năng khác mà không cần viết class component.

Ưu điểm của React

Giao diện mượt mà, tốc độ cao nhờ Virtual DOM.

Dễ bảo trì và mở rộng do thiết kế theo thành phần.

Thích hợp cho các ứng dụng lớn, có nhiều tương tác độc giả.

Cộng đồng phát triển lớn, tài liệu đầy đủ.

Dễ tích hợp với các thư viện và công nghệ khác như Redux, Tailwind CSS, Axios, React Router...

Ứng dụng của React trong đồ án

Trong đồ án Xây dựng trang Quản lý Thư viện Sách HPU, React.js được sử dụng để:

- Thiết kế các giao diện độc giả như: trang chủ, trang chi tiết sách, giỏ sách, đăng nhập, đăng ký, trang quản trị.
- Tạo các component tái sử dụng như: Header, Footer, ProductCard, FormInput, Modal...
- Gọi dữ liệu từ backend thông qua Axios và hiển thị lên giao diện.
- Điều hướng trang bằng thư viện react-router-dom.
- Quản lý trạng thái của giỏ sách và độc giả bằng useState, useReducer hoặc Context API.

1.2.2. Vite

Khái niệm

Vite là một công cụ build (build tool) thế hệ mới được phát triển bởi Evan You – tác giả của Vue.js. Vite được thiết kế để giúp quá trình phát triển ứng dụng frontend trở nên nhanh hơn và nhẹ hơn, đặc biệt là khi làm việc với các framework hiện đại như React, Vue, hoặc Svelte.

Từ "Vite" trong tiếng Pháp nghĩa là "nhanh" – phản ánh đúng mục tiêu mà công cụ này hướng tới: tăng tốc độ khởi động dự án và rút ngắn thời gian build sách.

Đặc điểm nổi bật của Vite

Khởi động siêu nhanh: Vite sử dụng ES Modules (ESM) để phục vụ mã nguồn trực tiếp từ trình duyệt trong môi trường phát triển, thay vì phải bundle toàn bộ project như Webpack, giúp khởi động gần như tức thì.

Hot Module Replacement (HMR) hiệu suất cao: Vite hỗ trợ HMR cực kỳ nhanh và chính xác, giúp nhà phát triển thấy ngay thay đổi khi cập nhật mã mà không cần reload trang.

Tối ưu khi build: Khi build cho production, Vite sử dụng Rollup – một bundler tối ưu, giúp tạo ra các file JavaScript gọn nhẹ, hiệu quả cao và dễ dàng cấu hình.

Hỗ trợ sẵn TypeScript, JSX, CSS Modules, PostCSS mà không cần cấu hình phức tạp.

Tương thích tốt với React, Vue và các thư viện frontend hiện đại.

Cách hoạt động của Vite

Vite chia quy trình làm việc thành hai pha chính:

- Pha phát triển (Development):
 - + Vite sử dụng native ESM và một server nhẹ được viết bằng Node.js.
 - + Các tệp JavaScript, CSS, TS... được phân phối trực tiếp đến trình duyệt mà không cần bundle, giúp thời gian phản hồi cực nhanh.
 - + Khi một tệp thay đổi, chỉ riêng tệp đó được xử lý lại thay vì rebuild toàn bộ dự án.
- Pha build (Production):
 - + Vite sử dụng Rollup để build toàn bộ project thành các file tối ưu.
 - + Tách code thành nhiều chunk, hỗ trợ lazy loading và tree shaking để giảm kích thước file.

Ứng dụng của Vite trong đồ án

Trong đồ án Xây dựng trang Quản lý Thư viện Sách HPU, Vite được sử dụng để:

- Khởi tạo dự án React nhanh chóng với lệnh `npm create vite@latest`.
- Tăng tốc độ khởi động và thay đổi giao diện trong quá trình phát triển frontend.
- Kết hợp với Tailwind CSS để build giao diện nhanh chóng, nhẹ và tối ưu.
- Tạo ra bản build nhỏ gọn, dễ deploy cho trang web mượn hàng khi triển khai thực tế.

1.2.3. React Router

Khái niệm

React Router là một thư viện định tuyến (routing library) được sử dụng phổ biến trong các ứng dụng React để quản lý việc điều hướng giữa các trang (route) một cách mượt mà mà không cần tải lại toàn bộ trang web. Nó cho phép các ứng dụng đơn trang (SPA – Single Page Application) hoạt động như các ứng dụng đa trang, bằng cách điều hướng giữa các component dựa trên URL trình duyệt.

React Router được phát triển bởi Remix Software và hiện nay phiên bản phổ biến là React Router v6, cung cấp cú pháp đơn giản, tối ưu hiệu suất, dễ sử dụng hơn các phiên bản trước.

Cách hoạt động

React Router hoạt động bằng cách:

- Theo dõi sự thay đổi của URL trong trình duyệt mà không làm reload trang.
- Gắn kết các URL tương ứng với component React cần hiển thị.
- Quản lý lịch sử điều hướng thông qua History API.

Các thành phần chính trong React Router:

- `<BrowserRouter>`: Bao bọc toàn bộ ứng dụng, quản lý lịch sử điều hướng.
- `<Routes>`: Container cho các route.
- `<Route>`: Định nghĩa từng tuyến đường, ánh xạ giữa đường dẫn và component.
- `<Link>`: Thay thế cho thẻ `<a>` để chuyển trang mà không reload.
- `useNavigate()`, `useParams()`, `useLocation()`: Các hook hỗ trợ điều hướng, lấy tham số, và vị trí hiện tại của trang.

Ưu điểm của React Router

- Tối ưu cho SPA: Điều hướng nhanh chóng, không cần tải lại trang.
- Tích hợp sâu với React: Giao diện được cập nhật theo component tương ứng.
- Hỗ trợ nested routes: Có thể lồng các route bên trong nhau.
- Route động: Hỗ trợ đường dẫn có tham số, ví dụ `/product/:id`.
- Điều hướng lập trình (programmatic navigation): Có thể điều hướng bằng hàm mà không cần click thủ công.

Ứng dụng trong đồ án Xây dựng trang Quản lý Thư viện Sách HPU

Trong đồ án Xây dựng trang Quản lý Thư viện Sách HPU, React Router được sử dụng để:

- Tạo hệ thống định tuyến gồm các trang: / (Trang chủ), /products, /product/:id, /cart, /login, /register, /admin, v.v.
- Cho phép chuyển trang mượt mà giữa các giao diện mà không tải lại toàn bộ ứng dụng.
- Hiển thị thông tin sách cụ thể thông qua route động như /product/123.
- Sử dụng useNavigate() để điều hướng độc giả sau khi đăng nhập hoặc mượn hàng thành công.

1.2.4. SCSS Modules

Khái niệm

SCSS Modules là sự kết hợp giữa SCSS (Sassy CSS) – phần mở rộng của CSS với cú pháp nâng cao, và CSS Modules – kỹ thuật viết CSS với phạm vi riêng biệt cho từng component. Việc kết hợp này giúp quản lý style trong các dự án frontend, đặc biệt là khi sử dụng React, một cách khoa học, rõ ràng và tránh trùng lặp.

SCSS cung cấp các tính năng như biến, mixins, nesting và import, giúp người phát triển có thể viết mã CSS một cách rõ ràng, dễ đọc và dễ bảo trì. Trong khi đó, CSS Modules đảm bảo rằng các lớp (class) chỉ áp dụng trong phạm vi component cụ thể, tránh xung đột toàn cục.

Ưu điểm của SCSS Modules

Phạm vi riêng biệt (local scope): Tránh xung đột tên class giữa các component trong toàn bộ ứng dụng.

Cấu trúc rõ ràng, dễ bảo trì: Mỗi component có thể gắn liền với file style riêng biệt.

Tận dụng toàn bộ sức mạnh của SCSS như biến, mixin, nesting, kế thừa...

Tương thích tốt với các dự án React + Vite, không cần cấu hình phức tạp.

Hỗ trợ modular hóa và tái sử dụng: Giúp tách nhỏ các phần giao diện mà không lo trùng lặp style.

Ứng dụng trong đồ án

Trong đồ án Xây dựng trang Quản lý Thư viện Sách HPU, SCSS Modules được sử dụng để thiết kế giao diện cho từng thành phần (component) như phần hiển thị sách, menu điều hướng, form đăng nhập, v.v. Mỗi phần có thể có file .module.scss riêng để đảm bảo

tính độc lập và dễ bảo trì. Điều này giúp quá trình phát triển frontend diễn ra thuận tiện, linh hoạt và đảm bảo tính mở rộng cao cho hệ thống.

1.2.5. Context API

Khái niệm

Context API là một cơ chế được tích hợp sẵn trong thư viện React từ phiên bản 16.3 trở lên, cho phép truyền dữ liệu giữa các component mà không cần phải sử dụng phương pháp truyền props theo từng cấp (prop drilling). Thay vì truyền dữ liệu qua nhiều lớp component trung gian, Context API cho phép chia sẻ dữ liệu trực tiếp từ một nơi trung tâm đến các component con có nhu cầu sử dụng.

Context API phù hợp để quản lý những dữ liệu cần dùng ở nhiều nơi trong ứng dụng như: thông tin độc giả, ngôn ngữ hiển thị, chủ đề giao diện (theme), trạng thái đăng nhập, giỏ sách...

Ưu điểm của Context API

Tránh lặp đi lặp lại việc truyền props qua nhiều cấp component.

Quản lý trạng thái đơn giản, đặc biệt đối với ứng dụng nhỏ hoặc vừa.

Tích hợp sẵn trong React, không cần cài thêm thư viện ngoài như Redux.

Tăng tính mô-đun và dễ mở rộng của ứng dụng.

Kết hợp tốt với React Hooks (như useContext) để làm cho mã ngắn gọn và dễ hiểu.

Ứng dụng trong đồ án

Trong đồ án Xây dựng trang Quản lý Sách HPU, Context API được sử dụng để:

Quản lý trạng thái độc giả đăng nhập trong toàn bộ ứng dụng (login/logout).

Chia sẻ giỏ sách giữa các component như header, trang chi tiết sách và trang Thanh toán phí mượn.

Quản lý giao diện độc giả như dark/light mode, lựa chọn ngôn ngữ,...

Giúp các component khác nhau truy cập dữ liệu chung mà không cần truyền props phức tạp.

Việc áp dụng Context API giúp giảm độ phức tạp trong mã nguồn và tăng hiệu quả trong quá trình phát triển frontend.

1.3. Back – End

1.3.1. Node.js

Khái niệm

Node.js là một nền tảng chạy JavaScript phía máy chủ (server-side) được xây dựng trên V8 JavaScript Engine của Google. Không giống như JavaScript trong trình duyệt (chỉ chạy ở client), Node.js cho phép lập trình viên viết các ứng dụng máy chủ bằng JavaScript, đặc biệt phù hợp cho các hệ thống web hiện đại như RESTful API, real-time application và microservices.

Node.js sử dụng mô hình non-blocking I/O (nhập/xuất không chặn) và kiến trúc event-driven (hướng sự kiện), giúp xử lý đồng thời nhiều kết nối một cách hiệu quả, đặc biệt thích hợp với các ứng dụng web cần hiệu suất cao.

Ưu điểm của Node.js

Tốc độ cao nhờ sử dụng V8 engine – biên dịch mã JavaScript sang mã máy.

Non-blocking I/O giúp xử lý nhiều request cùng lúc mà không bị chặn.

Sử dụng JavaScript toàn bộ từ frontend đến backend, giúp đồng bộ ngôn ngữ giữa client và server.

Thư viện phong phú nhờ hệ sinh thái NPM (Node Package Manager).

Thích hợp cho API hoặc microservice nhờ khả năng mở rộng linh hoạt.

Ứng dụng trong đồ án

Trong đồ án Xây dựng trang Quản lý Sách HPU, Node.js đóng vai trò là nền tảng backend chính để:

- Xây dựng RESTful API cho frontend giao tiếp.
- Xử lý các chức năng như đăng ký, đăng nhập, quản lý độc giả, sách, giỏ sách, và đơn mượn.
- Tương tác với cơ sở dữ liệu (MongoDB) thông qua các middleware.
- Quản lý các route API một cách hiệu quả và rõ ràng.
- Node.js giúp hệ thống backend xử lý nhanh, hiệu quả và đồng thời cho phép phát triển nhanh chóng với cùng một ngôn ngữ là JavaScript.

1.3.2. Express.js

Khái niệm

Express.js (thường gọi là Express) là một framework web tối giản và linh hoạt dành cho Node.js, được sử dụng để xây dựng các ứng dụng web và API nhanh chóng, hiệu quả. Đây là một trong những framework phổ biến nhất trong hệ sinh thái Node.js và là nền tảng quan trọng trong mô hình phát triển ME(N)N Stack (MongoDB, Express, Node, và thường kết hợp với React/Angular/Vue ở phía frontend).

Express cung cấp các công cụ mạnh mẽ để định nghĩa route, xử lý middleware, tương tác với HTTP request và response, cũng như tích hợp dễ dàng với các cơ sở dữ liệu và công cụ xác thực.

Ưu điểm của Express.js

Cấu trúc đơn giản, rõ ràng: Dễ tiếp cận đối với cả người mới và lập trình viên có kinh nghiệm.

Xử lý route linh hoạt: Hỗ trợ các phương thức HTTP (GET, POST, PUT, DELETE...) và route động.

Middleware mạnh mẽ: Cho phép xử lý tuần tự các yêu cầu qua nhiều bước, như kiểm tra xác thực, ghi log, xử lý lỗi...

Tương thích tốt với Node.js: Tận dụng triệt để ưu điểm non-blocking và event-driven của Node.js.

Cộng đồng lớn, tài liệu phong phú: Dễ tìm kiếm giải pháp và mở rộng chức năng.

Ứng dụng trong đồ án

Trong đồ án Xây dựng trang Quản lý Sách HPU, Express.js được sử dụng để phát triển hệ thống RESTful API kết nối giữa frontend và backend. Một số chức năng cụ thể:

Định nghĩa các API để quản lý độc giả, sách, danh mục, đơn mượn,...

Kiểm tra quyền truy cập của độc giả (qua middleware xác thực JWT).

Kết nối cơ sở dữ liệu MongoDB để truy xuất và cập nhật dữ liệu.

Xử lý upload ảnh sách, phân trang danh sách,...

Quản lý các lỗi phát sinh trong quá trình độc giả thao tác trên website.

Express giúp backend trở nên dễ tổ chức, dễ mở rộng và đáp ứng nhanh với các yêu cầu từ client.

1.3.3. Mô hình phát triển dự án MVC

Định nghĩa

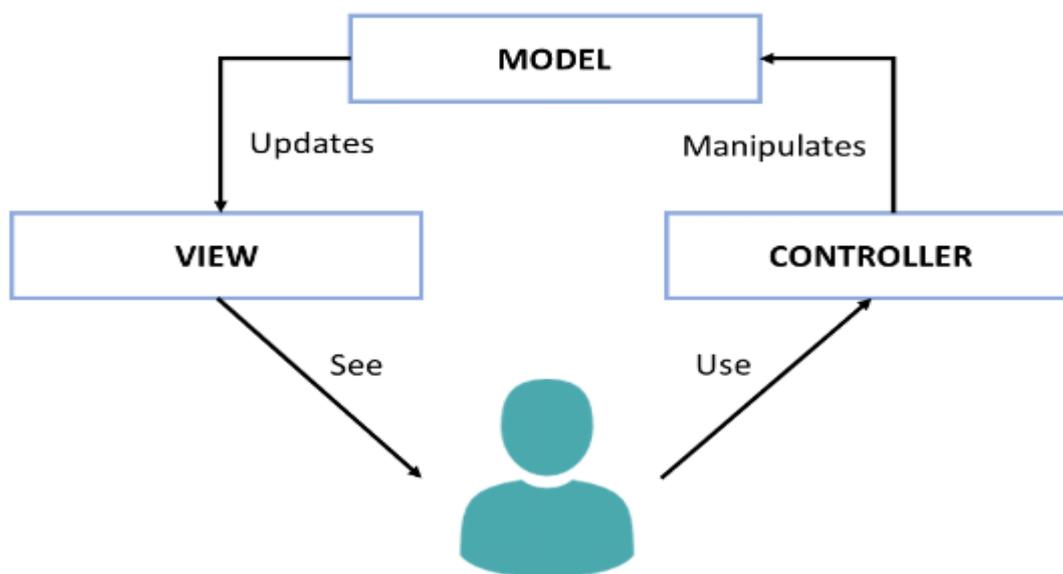
MVC là viết tắt của cụm từ “Model-View-Controller”. Đây là mô hình thiết kế được sử dụng trong kỹ thuật phần mềm, là một mẫu kiến trúc phần mềm để tạo lập giao diện độc giả trên máy tính. MVC chia thành 3 phần được kết nối với nhau và mỗi thành phần đều có nhiệm vụ riêng của nó và độc lập với các thành phần khác. Được sử dụng rộng rãi trong phát triển web, sự khác biệt được tùy chỉnh liên quan đến sự có mặt của server-client.

Các thành phần trong MVC

Đầu tiên là Model. Nó có nhiệm vụ thao tác với Database. Nó chứa tất cả các hàm, các phương thức truy vấn trực tiếp với dữ liệu. Controller sẽ thông qua các hàm, phương thức đó để lấy dữ liệu rồi gửi qua View.

Tiếp theo là View. Nó là giao diện độc giả (User Interface). Nó chứa các thành phần tương tác với độc giả như menu, button, image, text, ... View nhận dữ liệu từ Controller và hiển thị.

Cuối cùng là Controller. Nó là thành phần trung gian giữa Model và View. Controller đảm nhận vai trò tiếp nhận yêu cầu từ độc giả, thông qua Model để lấy dữ liệu sau đó thông qua View để hiển thị độc giả.



Hình 1.2: Mô hình MVC

1.3.4. Middleware

Khái niệm

Middleware là một khái niệm phổ biến trong lập trình backend, đặc biệt trong các framework như Express.js. Middleware có thể hiểu là một hàm trung gian được gọi trong quá trình xử lý HTTP request và response. Nó có nhiệm vụ can thiệp, kiểm tra, xử lý dữ liệu trước khi request được chuyển tiếp đến route chính hoặc khi response được trả về client.

Trong Express.js, middleware có thể thực hiện nhiều chức năng như:

- Kiểm tra xác thực độc giả (authentication).
- Ghi log thông tin yêu cầu.
- Phân tích body của request (body parsing).
- Xử lý lỗi.
- Hạn chế số lượng truy cập (rate limiting), kiểm tra quyền (authorization), hoặc xác minh token (JWT).

Cơ chế hoạt động

Middleware hoạt động theo chuỗi: Mỗi middleware sau khi xử lý xong có thể gọi next() để chuyển sang middleware tiếp theo. Nếu không gọi next(), quá trình sẽ dừng lại tại middleware đó. Điều này giúp kiểm soát linh hoạt luồng xử lý của ứng dụng.

Ưu điểm của Middleware

Tái sử dụng cao: Một middleware có thể áp dụng cho nhiều route.

Tăng bảo mật và kiểm soát: Nhờ có thể kiểm tra token, quyền truy cập, và trạng thái request trước khi xử lý chính.

Giúp mã nguồn rõ ràng, dễ tổ chức: Tách biệt các bước xử lý request thành từng phần riêng biệt.

Dễ mở rộng và bảo trì: Có thể thêm mới chức năng mà không ảnh hưởng logic cũ.

Ứng dụng trong đồ án

Trong đồ án Xây dựng trang Quản lý Sách HPU, Middleware đóng vai trò quan trọng trong:

- Xác thực độc giả thông qua JWT để đảm bảo chỉ người đã đăng nhập mới được thực hiện các thao tác như đặt mượn, xem thông tin tài khoản.
- Phân quyền độc giả, ví dụ: chỉ admin mới được thêm/sửa sách.

- Kiểm tra dữ liệu đầu vào (validate) như kiểm tra email, mật khẩu, tên sách...
- Xử lý lỗi chung, trả về thông báo lỗi rõ ràng và có định dạng cho frontend.
- Bảo vệ API khỏi các cuộc tấn công như spam hoặc gửi nhiều request bất hợp pháp.

Middleware giúp tăng độ bảo mật, hiệu quả và khả năng mở rộng của hệ thống backend.

1.3.5. Kết nối Frontend – Backend

Khái niệm

Trong mô hình phát triển ứng dụng web hiện đại, Frontend và Backend là hai phần hoạt động độc lập nhưng liên kết chặt chẽ với nhau. Frontend là giao diện độc giả (UI) được xây dựng bằng các công nghệ như HTML, CSS, JavaScript (thường là React, Angular, hoặc Vue). Backend là nơi xử lý logic nghiệp vụ, kết nối với cơ sở dữ liệu và cung cấp API cho frontend tương tác.

Kết nối Frontend – Backend là quá trình giao tiếp giữa client (trình duyệt độc giả) và server (máy chủ backend), thường thông qua các API (Application Programming Interface), đặc biệt là RESTful API hoặc GraphQL API.

Cơ chế kết nối phổ biến

Giao tiếp qua HTTP request: Frontend gửi yêu cầu HTTP (GET, POST, PUT, DELETE) đến backend để truy xuất hoặc thao tác dữ liệu.

Dữ liệu trao đổi theo định dạng JSON: Backend trả dữ liệu dưới dạng JSON để frontend dễ dàng xử lý.

Axios hoặc Fetch API thường được dùng ở frontend để gửi request.

CORS (Cross-Origin Resource Sharing) phải được cấu hình ở backend nếu frontend và backend chạy ở hai domain/port khác nhau.

Vai trò trong đồ án

Trong đồ án Xây dựng trang Quản lý Sách HPU, việc kết nối frontend và backend đóng vai trò thiết yếu:

- Giao tiếp động: Khi độc giả thực hiện thao tác như đăng nhập, tìm kiếm sách hoặc đặt mượn, frontend sẽ gọi API tương ứng để lấy/gửi dữ liệu.
- Xác thực độc giả: Frontend nhận token từ backend sau khi sinh viên/độc giả đăng nhập và lưu trữ để sử dụng cho các yêu cầu như xem thông tin cá nhân.

- Đồng bộ dữ liệu: Ví dụ, danh sách sách trong giao diện được lấy trực tiếp từ cơ sở dữ liệu thông qua API.
- Hiện thị lỗi và thông báo: Backend gửi phản hồi thành công/thất bại, giúp frontend thông báo rõ ràng cho độc giả.

Công cụ và công nghệ sử dụng

Frontend: React.js kết hợp với thư viện Axios để gửi HTTP requests.

Backend: Express.js xây dựng hệ thống API RESTful.

Token (JWT): Dùng để xác thực và phân quyền giữa frontend và backend.

CORS Middleware: Được sử dụng tại backend để cho phép frontend truy cập từ domain khác.

1.4. Cơ sở dữ liệu MongoDB

1.4.1. Giới thiệu MongoDB

MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở thuộc dòng NoSQL (Not Only SQL), được thiết kế theo hướng hướng tài liệu (Document-oriented). Thay vì lưu trữ dữ liệu dưới dạng bảng (Table) và hàng (Row) như các hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) truyền thống, MongoDB lưu trữ dữ liệu dưới dạng các Document (tài liệu) với định dạng JSON (JavaScript Object Notation) hoặc BSON (Binary JSON).

Được phát triển bởi MongoDB Inc., hệ quản trị này đặc biệt phù hợp với các ứng dụng web hiện đại, các hệ thống yêu cầu xử lý dữ liệu lớn (Big Data) và các ứng dụng thời gian thực nhờ khả năng mở rộng linh hoạt và hiệu suất cao. MongoDB là một thành phần quan trọng trong các bộ công nghệ (stack) phổ biến như MERN (MongoDB, Express, React, Node.js)...

1.4.2. Đặc điểm nổi bật

Cấu trúc dữ liệu linh hoạt (Schema-less): MongoDB không yêu cầu định nghĩa cấu trúc bảng cứng nhắc trước khi lưu trữ. Các Document trong cùng một Collection (Bộ sưu tập) có thể có các trường (Field) khác nhau, giúp dễ dàng thay đổi cấu trúc dữ liệu trong quá trình phát triển dự án mà không gây Phí mượn sáchn đoạn hệ thống.

Lưu trữ dạng Document (JSON/BSON): Dữ liệu được lưu trữ dưới dạng các cặp khóa-Phí mượn sách trị (key-value) phân cấp, rất trực quan và tương thích tự nhiên với các đối tượng trong ngôn ngữ JavaScript (Node.js).

Hiệu suất cao: MongoDB hỗ trợ đánh chỉ mục (Indexing) mạnh mẽ, giúp tối ưu hóa tốc độ truy vấn dữ liệu. Khả năng đọc/ghi dữ liệu của MongoDB thường nhanh hơn RDBMS trong các tác vụ xử lý dữ liệu phi cấu trúc hoặc mượn cấu trúc.

Khả năng mở rộng (Scalability): Hỗ trợ tốt khả năng mở rộng theo chiều ngang (Sharding), cho phép phân tán dữ liệu ra nhiều máy chủ để xử lý lượng truy cập lớn.

Công cụ hỗ trợ: Cung cấp các công cụ quản lý trực quan như MongoDB Compass hoặc nền tảng đám mây MongoDB Atlas giúp dễ dàng thao tác, Phí mượn sáchm sát và bảo mật dữ liệu.

1.4.3. Vai trò của MongoDB trong đồ án

- Trong đồ án Xây dựng Trang Quản lý Thư viện Sách HPU, MongoDB đóng vai trò là kho lưu trữ dữ liệu chính (Database Layer), kết hợp chặt chẽ với Node.js và Express.js ở phía Backend. Nhiệm vụ cụ thể bao gồm:
- Lưu trữ đối tượng phức tạp: Lưu trữ thông tin sách và sách với nhiều thuộc tính đa dạng (như danh sách tác giả, thể loại, hình ảnh, đánh Phí mượn sách) dưới dạng một Document duy nhất, giúp giảm thiểu việc phải thực hiện các phép nối (JOIN) phức tạp như trong SQL.
- Quản lý độc giả và xác thực: Lưu trữ thông tin tài khoản, mật khẩu (đã mã hóa), và các token xác thực phiên làm việc.
- Xử lý nghiệp vụ đặt mượn: Lưu trữ thông tin giỏ sách, đơn mượn và trạng thái mượn trả một cách nhất quán.
- Tương tác qua Mongoose: Dự án sử dụng thư viện Mongoose (ODM - Object Data Modeling) để mô hình hóa dữ liệu, xác thực (validation) và tương tác với MongoDB một cách dễ dàng và an toàn hơn.

1.5. Kiểm thử

1.5.1. Giới thiệu

Định nghĩa: Kiểm thử phần mềm (Software Testing) là hoạt động nhằm tìm kiếm và phát hiện lỗi của phần mềm. Kiểm thử phần mềm là một tiến trình hay một tập hợp các tiến trình được thiết kế đảm bảo chương trình máy tính thực hiện theo đúng yêu cầu của Độc giả. Kiểm thử phần mềm là quá trình thực thi một chương trình với mục đích tìm lỗi (theo The Art of Software Testing).

Mục tiêu: Phát hiện sớm các lỗi, và tìm được càng nhiều lỗi càng tốt để tránh sai sót, sửa đi sửa lại nhiều lần. Ngăn chặn các lỗi phát sinh, hạn chế rủi ro cho hệ thống và chứng minh rằng sách phù hợp, mang tính thực hiện, hiệu quả và đúng với đặc tả của nó. Từ đó, chứng tỏ được rằng các testcase và các báo cáo tạo ra đúng và hữu dụng.

1.5.2. Kiểm thử hộp đen

Định nghĩa: Black box testing (Kiểm thử hộp đen) là một phương pháp kiểm thử phần mềm tập trung vào việc kiểm tra hành vi bên ngoài của một ứng dụng mà không quan tâm đến cấu trúc nội bộ hay chi tiết cài đặt của nó. Trong Black box testing, người kiểm thử chỉ xem ứng dụng như một "hộp đen" và tập trung vào đầu vào và đầu ra của ứng dụng, cũng như các tình huống và điều kiện biên.

Ưu điểm	Nhược điểm
<ul style="list-style-type: none">– Phát hiện lỗi không mong muốn.– Đảm bảo tính tương tác mượt mà và đáp ứng nhanh chóng trải nghiệm độc giả tốt.– Độc lập với cấu trúc nội bộ.– Một nhóm riêng biệt chịu trách nhiệm kiểm thử ứng dụng theo phương pháp Black box testing giúp đảm bảo tính chuyên nghiệp và độc lập của quá trình kiểm thử.– Tiết kiệm thời gian và nguồn lực.	<ul style="list-style-type: none">– Không thể đảm bảo độ phủ kiểm thử cao như White box testing.– Có thể bỏ qua các lỗi nằm sâu trong hệ thống như lỗi logic hoặc lỗi thiết kế.– Không thể kiểm tra các tình huống phức tạp.– Phụ thuộc vào khả năng và kinh nghiệm của người kiểm thử nếu người kiểm thử thiếu kỹ năng có thể dẫn đến bỏ qua lỗi.

Bảng 1.1: Ưu nhược điểm của kiểm thử hộp đen

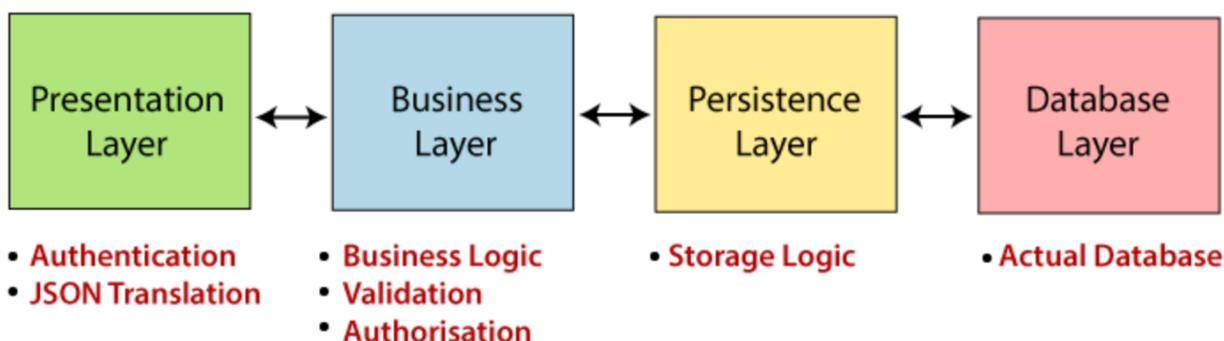
Các phương pháp được sử dụng:

- Equivalence Partitioning (Phân loại tương đương chia các Phí mượn sách trị đầu vào thành các nhóm tương đương và chỉ kiểm tra một Phí mượn sách trị trong mỗi nhóm. Điều này giúp giảm số lượng ca kiểm thử cần thực hiện mà vẫn đảm bảo kiểm thử các trường hợp quan trọng.
- Boundary Value Analysis (Phân tích Phí mượn sách trị biên tập trung vào việc kiểm tra các Phí mượn sách trị biên và điều kiện biên của ứng dụng. Kiểm thử được thực hiện với các Phí mượn sách trị gần biên và trên biên để phát hiện lỗi có thể xảy ra.
- Decision Table Testing (Kiểm thử bảng quyết định) sử dụng các bảng quyết định để xác định các điều kiện và hành động tương ứng của ứng dụng. Các ca kiểm thử được tạo dựa trên các kết hợp của các điều kiện và hành động trong bảng quyết định.

- State Transition Testing (Kiểm thử chuyển trạng thái) tập trung vào việc kiểm thử các chuyển đổi trạng thái của ứng dụng. Các ca kiểm thử được tạo dựa trên các trạng thái và các sự kiện chuyển trạng thái.
- Error Guessing (Đoán lỗi) dựa trên kinh nghiệm và sự sáng tạo của người kiểm thử để đoán các lỗi có thể xảy ra trong ứng dụng. Các ca kiểm thử được tạo dựa trên các giả định và dự đoán về các điểm yếu tiềm năng của hệ thống.
- Exploratory Testing (Kiểm thử khám phá) tập trung vào việc khám phá và kiểm thử ngẫu nhiên mà không dựa trên kịch bản hay kế hoạch cụ thể. Người kiểm thử sử dụng sự sáng tạo và trực Phí mượn sách để tìm kiếm lỗi và điều kiện biên không dự đoán trước.
- Regression Testing (Kiểm thử hồi quy đảm bảo rằng các thay đổi trong ứng dụng không gây ảnh hưởng đến các chức năng và tính năng đã được kiểm thử trước đó. Các ca kiểm thử được chọn để tái kiểm tra và đảm bảo tính ổn định của ứng dụng.

CHƯƠNG 2. PHÂN TÍCH ĐẶC TẢ CHỨC NĂNG CỦA HỆ THỐNG

2.1. Kiến trúc tổng thể



Hình 2.1: Sơ đồ kiến trúc hệ thống

Hệ thống “Xây dựng Trang Quản lý Sách HPU” ứng dụng kiến trúc hệ thống Spring Boot. Spring Boot chia ứng dụng ra làm 4 tầng như sau:

- Presentation Layer: Đây là tầng trên cùng của Spring Boot bao gồm có View và một phần front-end của ứng dụng. Ở tầng này Spring Boot sẽ nhận các request từ độc giả rồi tiến hành xử lý xác thực. Nó sẽ xử lý chuyển đổi từ đối tượng JSON sang đối tượng Java và ngược lại. Khi xác thực được thực hiện thành công nó sẽ được chuyển sang tầng tiếp theo là Business Layer.
- Business Layer: Tầng này sẽ xử lý tất cả các yêu cầu nghiệp vụ (Business Logic) bao gồm các lớp Service và sử dụng các service được cung cấp bởi tầng DAO. Nó chịu trách nhiệm xác nhận/kiểm duyệt dữ liệu và ủy quyền cho user.
- Persistence Layer: Bao gồm tất cả các logic liên quan đến database. Tầng này chịu trách nhiệm chuyển đổi đối tượng Java sang đối tượng database và ngược lại.
- Database Layer: Bao gồm tất cả các database RDBMS, NoSQL. Nó chịu trách nhiệm thực hiện CRUD dữ liệu vào database.

2.2. Yêu cầu hệ thống

2.2.1. Yêu cầu chức năng

Chức năng chung:

- Người sử dụng trước khi chưa đăng nhập có thể tìm kiếm sách xem được danh sách sách, tìm kiếm sách và thông tin chi tiết của sách.

- Phần mềm yêu cầu tất cả người sử dụng đăng nhập để thực hiện các chức năng khác.

Phía Admin:

- Phần mềm cho phép Admin quản lý sách: Xem, thêm mới, chỉnh sửa, xóa các loại sách trên website.
- Phần mềm cho phép Admin quản lý danh mục: Xem, thêm mới, chỉnh sửa, xóa các danh mục sách.
- Phần mềm cho phép Admin quản lý độc giả: Xem và quản lý độc giả: chỉnh quyền tài khoản.
- Phần mềm cho phép Admin xem và quản lý đơn mượn: Xem và quản lý các đơn mượn từ Độc giả, bao gồm xác nhận, xử lý và hủy đơn mượn khi cần thiết.
- Phần mềm cho phép Admin xem báo cáo thống kê xem tình trạng sách mượn, số tiền mượn thu được giúp Admin nắm được tình hình.

Phía độc giả:

- Phần mềm cho phép độc giả xem sách: Độc giả có thể xem qua danh sách sách hiện có trên trang web.
- Phần mềm cho phép độc giả tìm kiếm và lọc sách: độc giả có thể sử dụng công cụ tìm kiếm và lọc sách theo mức Phí mượn sách.
- Phần mềm cho phép độc giả thêm sách vào giỏ sách: Khi độc giả chọn sách, họ có thể thêm vào giỏ sách và tiến hành Thanh toán phí mượn.
- Phần mềm cho phép độc giả quản lý giỏ sách: Độc giả có thể xem, sửa, xóa số lượng sách muốn mượn trong giỏ sách
- Phần mềm cho phép độc giả đặt mượn và Thanh toán phí mượn hóa đơn online.
- Phần mềm cho phép độc giả quản lý thông tin cá nhân: Độc giả có thể cập nhật thông tin cá nhân như địa chỉ giao hàng, thông tin liên hệ. Họ cũng có thể thay đổi mật khẩu, đăng xuất khỏi tài khoản.

2.2.2. Yêu cầu phi chức năng

Yêu cầu phi chức năng đưa ra tiêu chí nhằm đánh giá hoạt động của hệ thống, đảm bảo hệ thống vận hành ổn định.

Hệ thống quản lý dự án cần đáp ứng được các yêu cầu phi chức năng sau:

- Yêu cầu bảo mật: Bảo vệ thông tin khỏi việc truy cập không được phép, rò rỉ thông tin và các cuộc tấn công mạng. Xác thực và cấp quyền truy cập vào hệ

thông cho độc giả dựa trên vai trò và quyền hạn tương ứng. Thực hiện các biện pháp bảo mật dữ liệu theo các tiêu chuẩn quốc tế.

- Yêu cầu sao lưu: Dữ liệu lưu trong hệ thống được sao lưu dự phòng tự động 24/24 bằng một hệ thống song hành tránh mất mát dữ liệu. Dữ liệu hệ thống có thể kết xuất ra các thiết bị lưu trữ ngoài và phục hồi khi cần thiết.
- Yêu cầu về tính sử dụng:
 - + Hệ thống đơn giản, dễ học và dễ sử dụng, đảm bảo độc giả có thể sử dụng được 80% hệ thống sau khoảng 2 tiếng hướng dẫn sử dụng.
 - + Hệ thống hỗ trợ các trình duyệt phổ biến là Chrome, Cốc cốc.
- Yêu cầu về hiệu năng: Hệ thống đáp ứng các tác vụ thực hiện tức thời trong thời gian ngừng cho phép chấp nhận dưới 60s, đảm bảo phục vụ truy cập online 50 người cùng một lúc.

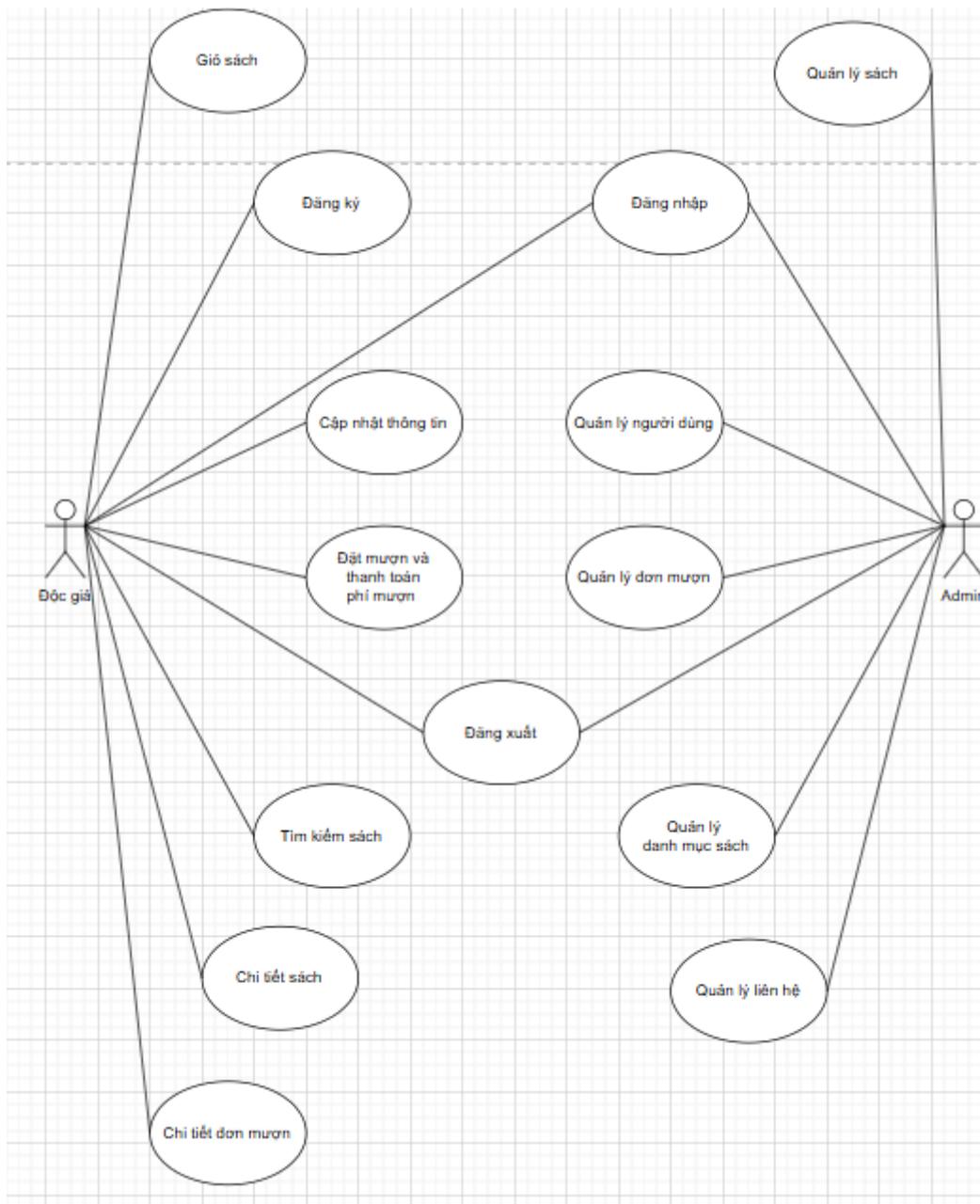
2.3. Usecase

2.3.1. Tác nhân hệ thống

STT	Tác nhân	Mô tả
1	Độc giả	Đây là tác nhân đại diện cho những độc giả/sinh viên truy cập website. Độc giả có thể xem, tìm kiếm sách, xem thông tin sách, đăng ký, đăng nhập, đặt mượn, Thanh toán phí mượn và theo dõi tình trạng đơn mượn. Ngoài ra còn có thể cập nhật thông tin cá nhân xem lịch sử mượn sách.
2	Admin	Đây là tác nhân đại diện cho người quản trị hệ thống. Khi truy cập vào trang quản trị, Admin có thể xem các báo cáo thống kê, quản lý thông tin sách, danh mục, các đơn mượn, và tài khoản độc giả.

Bảng 2.1: Danh sách tác nhân

2.3.2. Biểu đồ usecase tổng quát

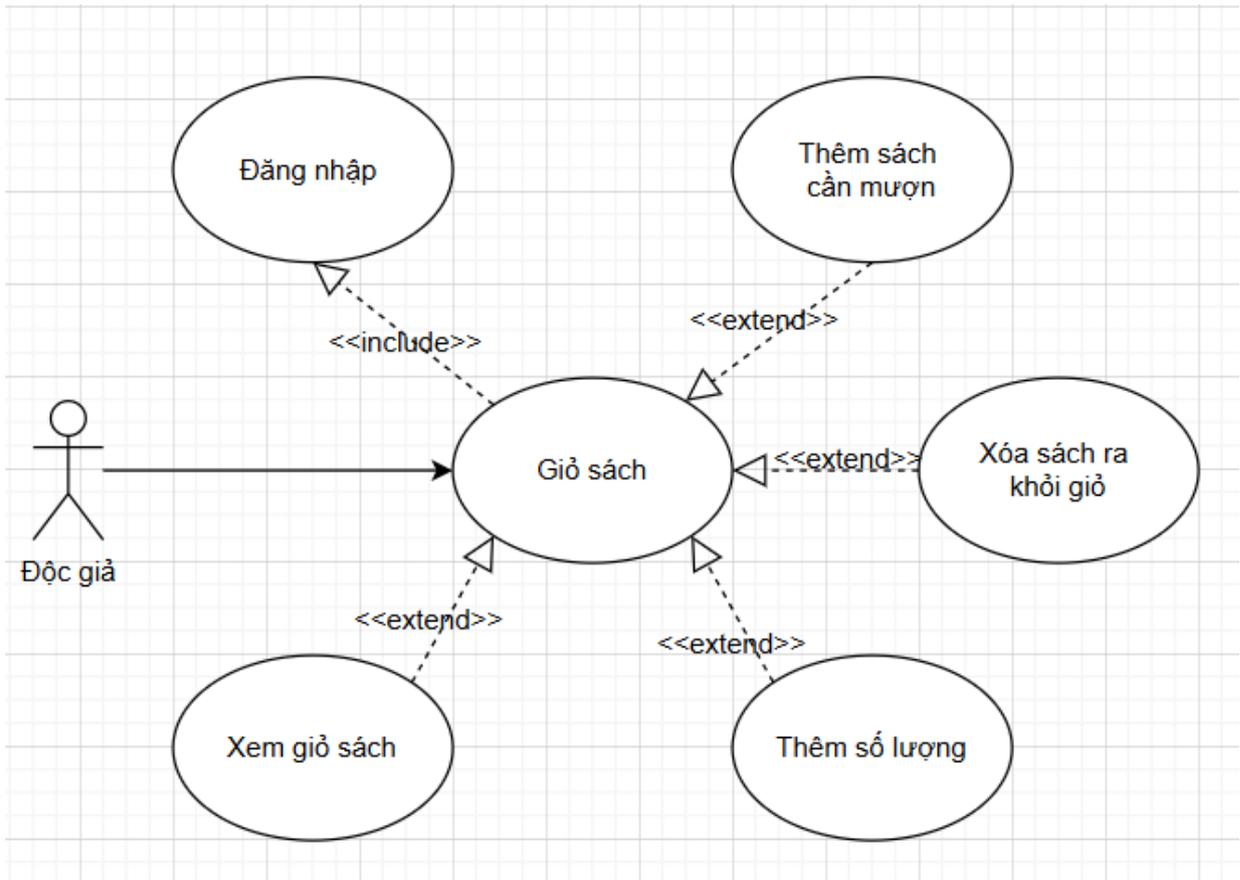


Hình 2.2: Biểu đồ Use Case tổng quát

**Biểu đồ phân rã usecase Độc giả*

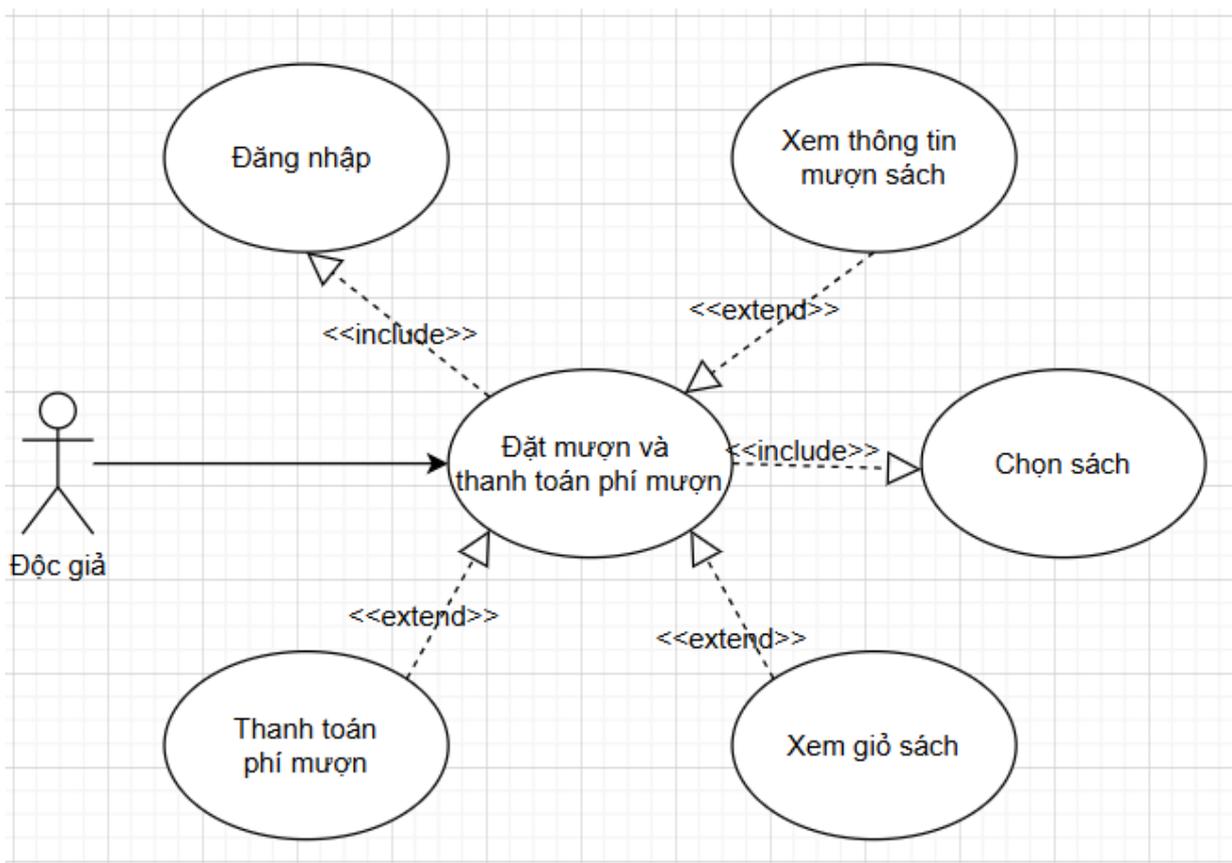
2.3.3. Biểu đồ usecase phân rã

Biểu đồ phân rã usecase Giỏ sách



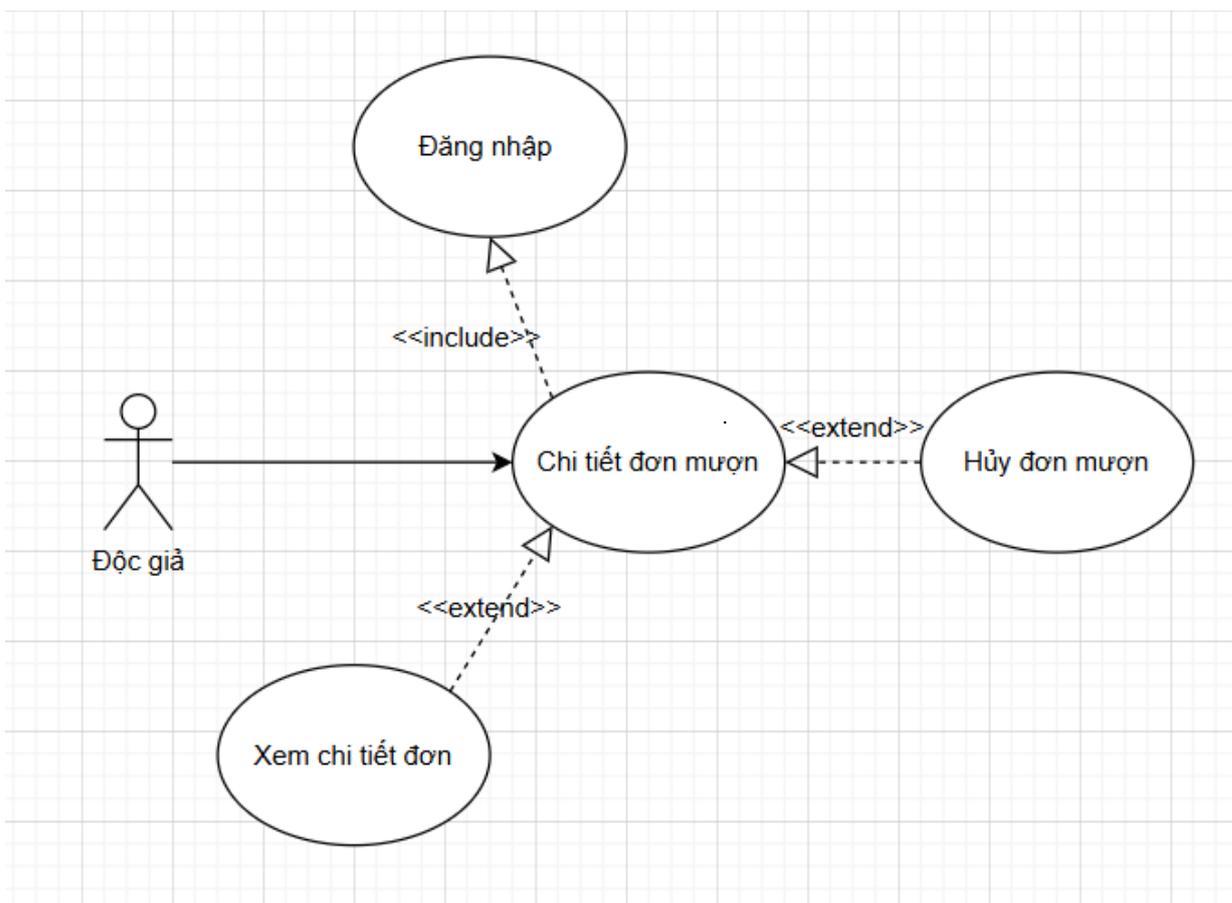
Hình 2.3 Biểu đồ phân rã Use Case Giỏ sách

Biểu đồ phân rã usecase Đặt mượn và Thanh toán phí mượn



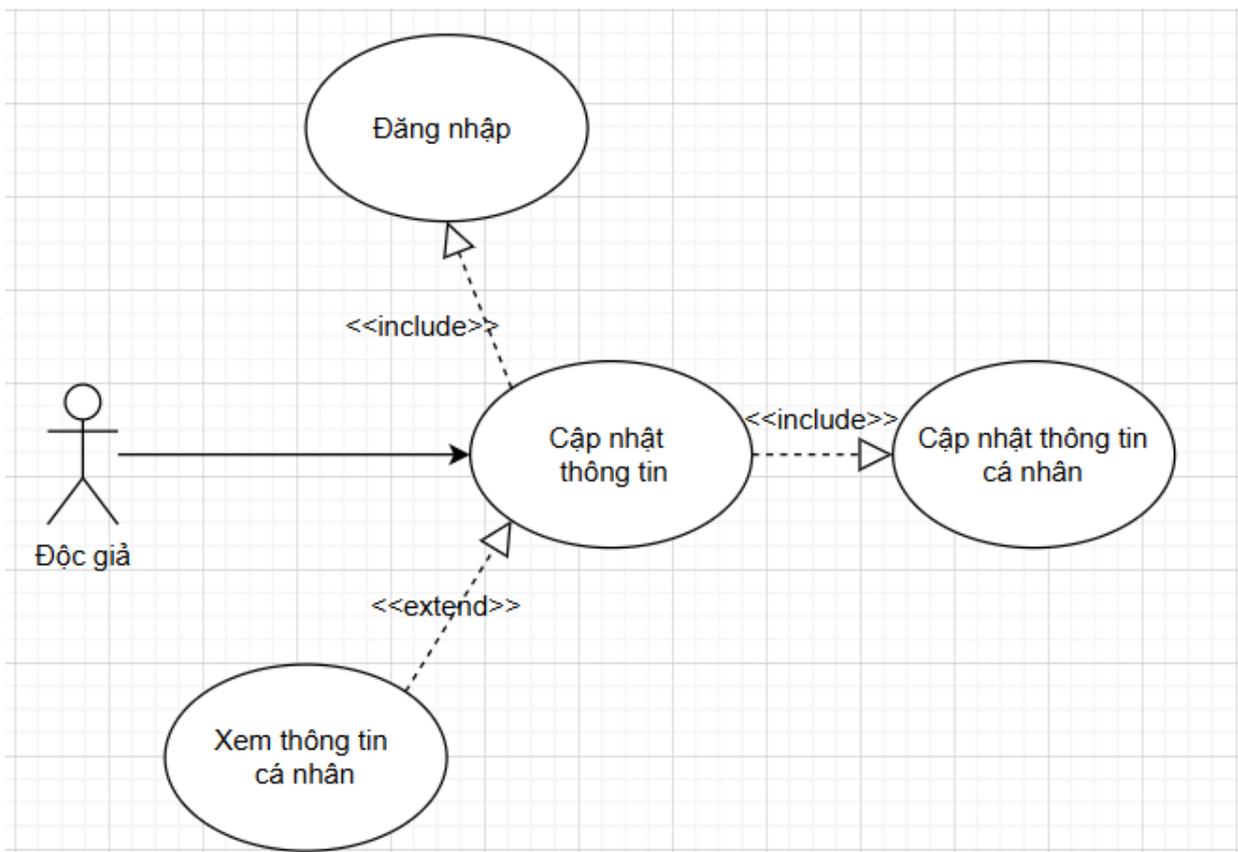
Hình 2.4: Biểu đồ phân rã Use Case Đặt mượn và Thanh toán phí mượn

Biểu đồ phân rã usecase Chi tiết đơn mượn



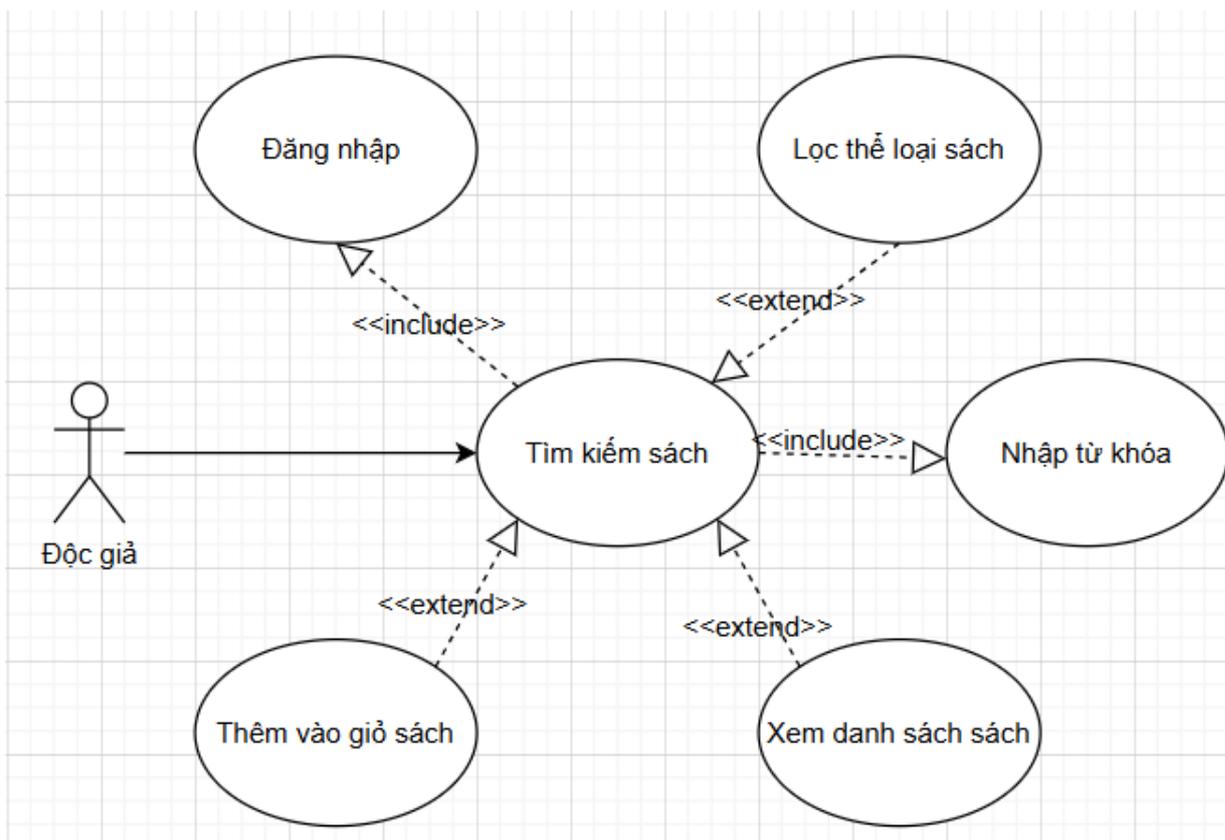
Hình 2.5: Biểu đồ phân rã Use Case Chi tiết đơn mượn

Biểu đồ phân rã usecase Cập nhật thông tin



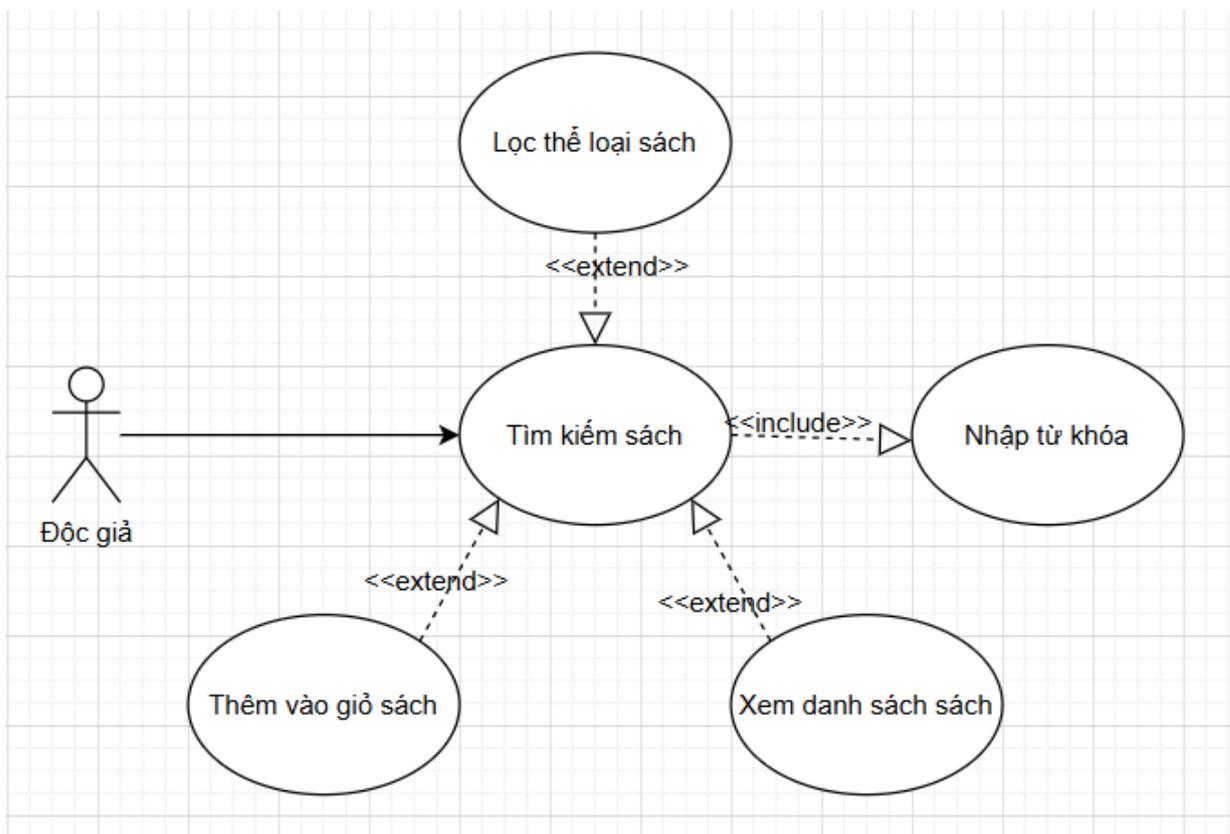
Hình 2.6: Biểu đồ phân rã Use Case Cập nhật thông tin

Biểu đồ phân rã usecase Tìm kiếm sách



Hình 2.7: Biểu đồ phân rã Use Case Tìm kiếm sách

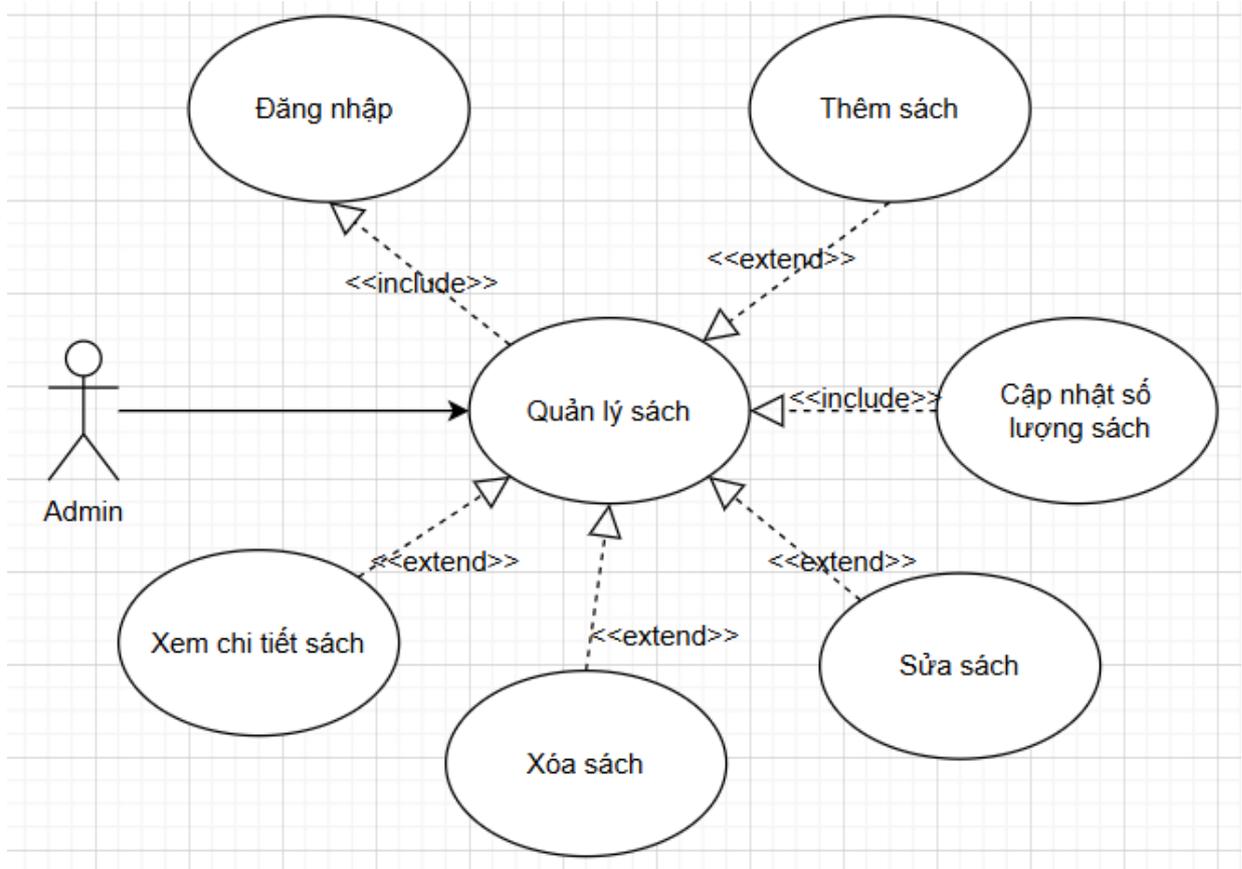
Biểu đồ phân rã usecase Chi tiết sách



Hình 2.8: Biểu đồ phân rã Use Case Chi tiết sách

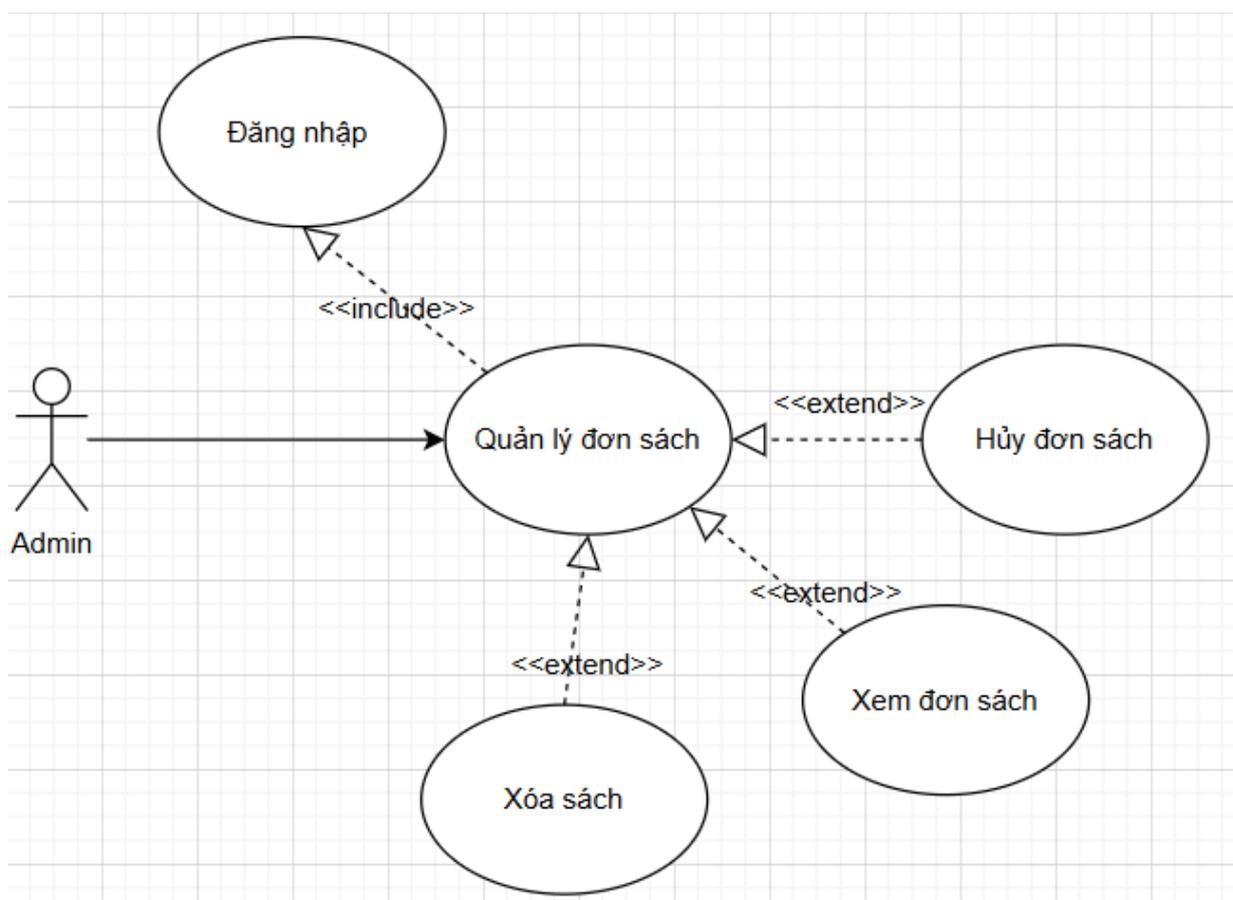
Biểu đồ phân rã usecase quản trị viên

Biểu đồ phân rã usecase Quản lý sách



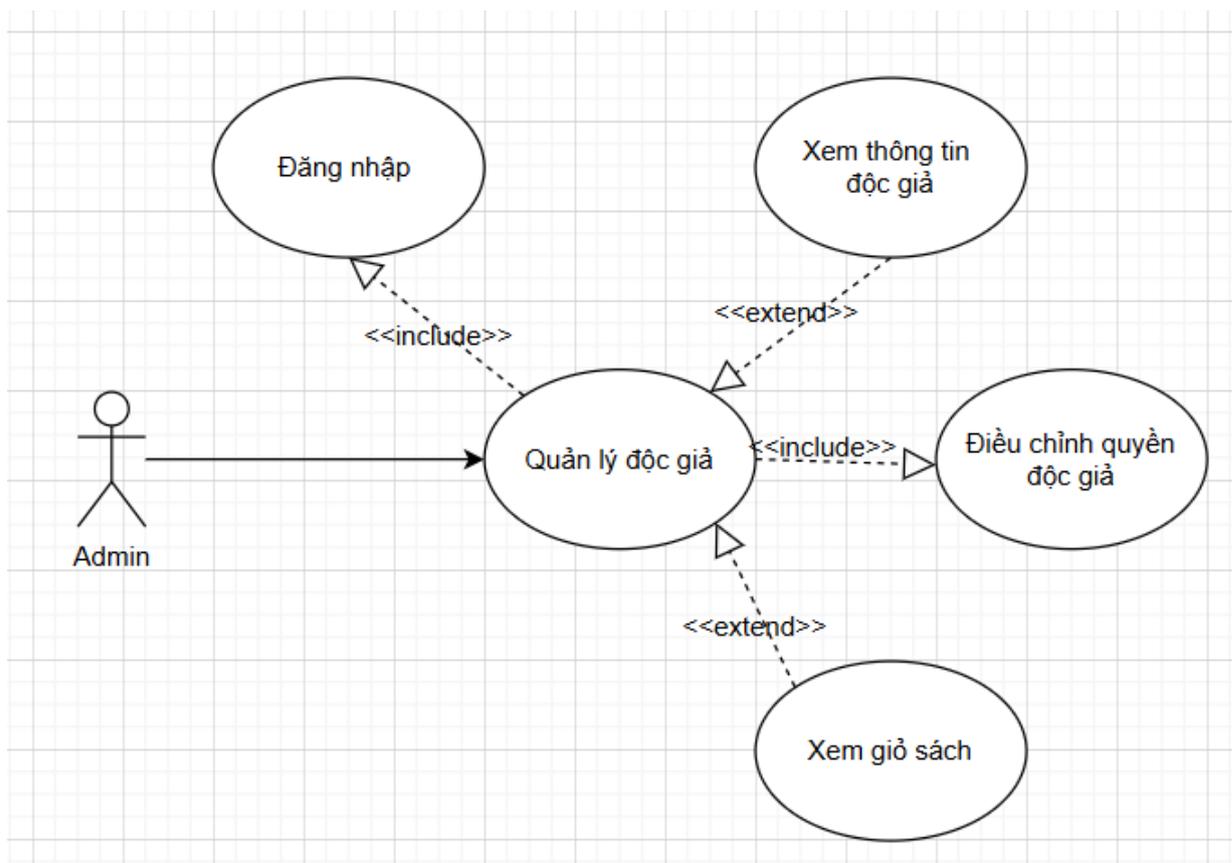
Hình 2.9: Biểu đồ phân rã Use Case Quản lý sách

Biểu đồ phân rã usecase Quản lý đơn mượn



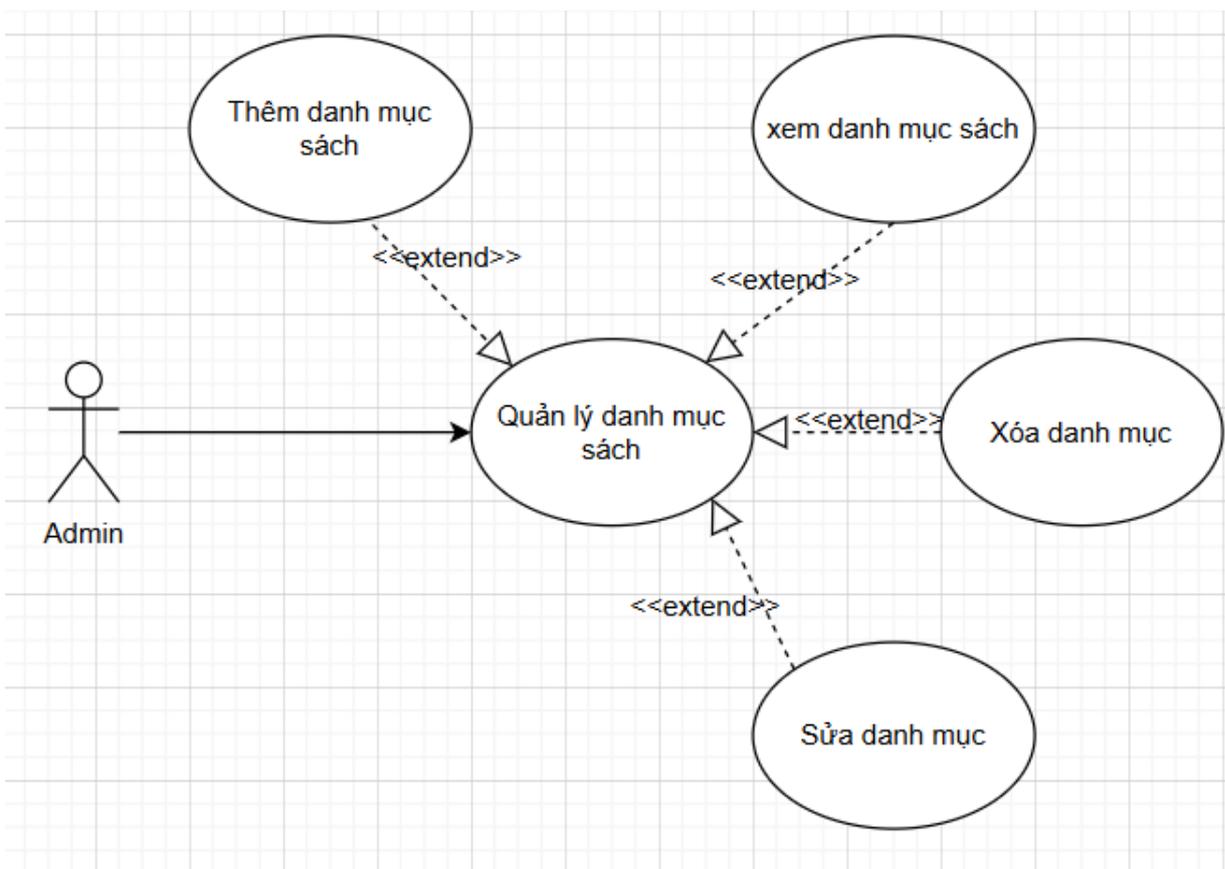
Hình 2.10: Biểu đồ phân rã Use Case Quản lý đơn mượn

Biểu đồ phân rã usecase Quản lý độc giả



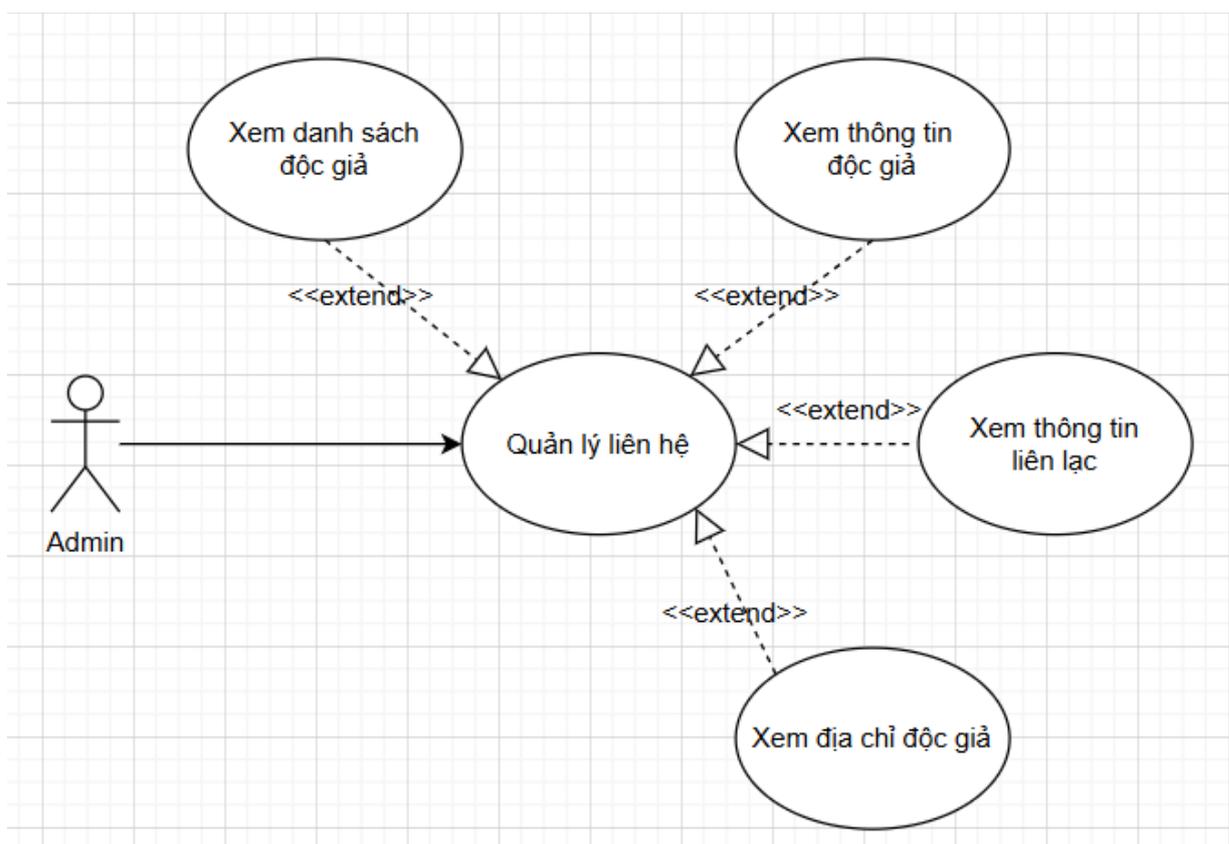
Hình 2.11: Biểu đồ phân rã Use Case Quản lý độc giả

Biểu đồ phân rã usecase Quản lý danh mục sách



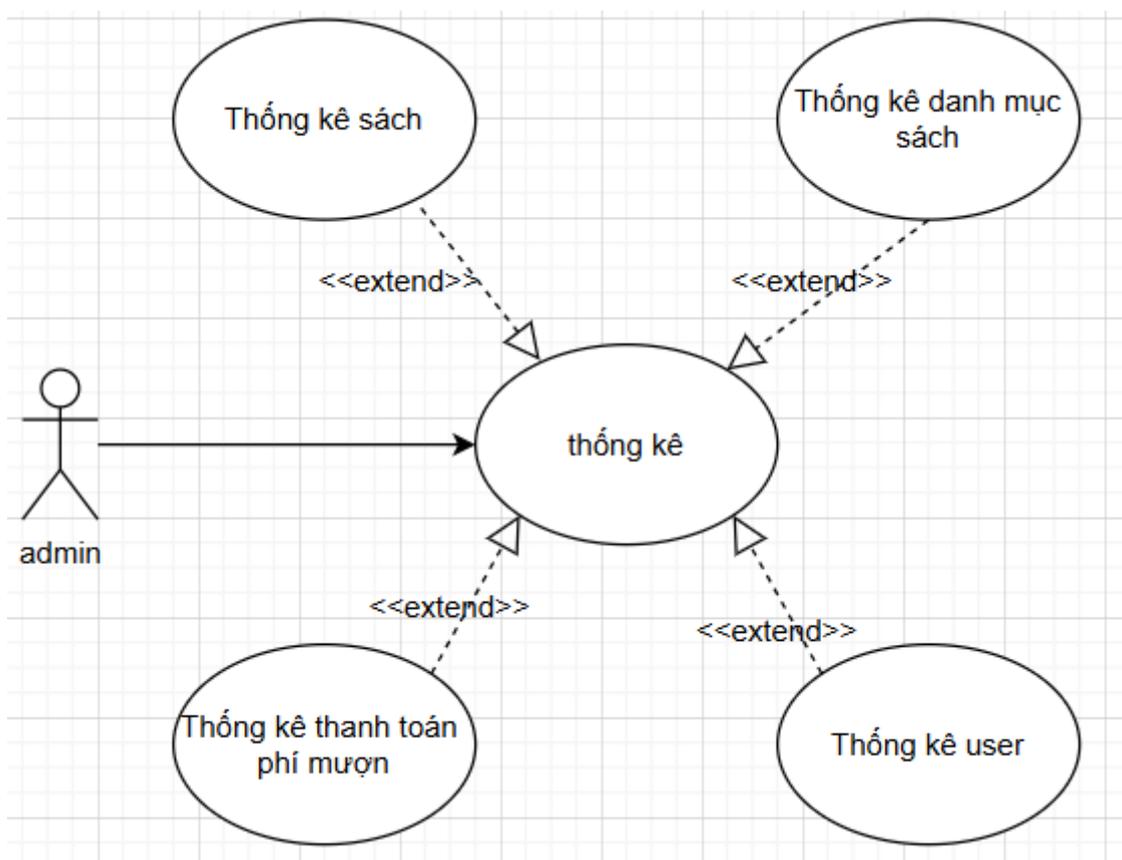
Hình 2.12: Biểu đồ phân rã Use Case Quản lý danh mục sách

Biểu đồ phân rã usecase Quản lý liên hệ



Hình 2.13: Biểu đồ phân rã Use Case Quản lý liên hệ

Biểu đồ phân rã usecase Thống kê



Hình 2.14: Biểu đồ phân rã Use Case Thống kê

2.4. Đặc tả chức năng

2.4.1. Use Case: Đăng nhập

Tên Use Case		Đăng nhập	Độ phức tạp: Cao
Mô tả		Cho phép độc giả đăng nhập và sử dụng dịch vụ	
Tác nhân		Độc giả	
Tiền điều kiện		Tác nhân đã có tài khoản trên hệ thống	
Hậu điều kiện	Thành công	Đăng nhập thành công vào trang chủ	
	Lỗi	Đăng nhập thất bại	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính/Kịch bản chính			
Usecase này bắt đầu khi user cần đăng nhập vào trang web. <ul style="list-style-type: none">- Trang web yêu cầu độc giả bắt buộc nhập email và mật khẩu đăng nhập.- User nhập tài khoản và mật khẩu đã đăng ký trước đó vào các trường thông tin- Hệ thống kiểm tra xem user đã nhập đủ thông tin trong các trường bắt buộc chưa: Nếu chưa nhập đủ, thực hiện Luồng A.- Hệ thống kiểm tra số điện thoại và mật khẩu:<ul style="list-style-type: none">+ Hệ thống hiển thị thông báo “Đăng nhập thành công” và hiển thị trang chủ+ Nếu không hợp lệ, thực hiện Luồng B.			
Luồng sự kiện phát sinh/Kịch bản phát sinh			
Luồng A: Không nhập trường bắt buộc			
User bỏ trống các trường bắt buộc: <ul style="list-style-type: none">- User bỏ trống trường email, hệ thống hiển thị thông báo “Email là bắt buộc”.- User bỏ trống mật khẩu, hệ thống hiển thị thông báo “Mật khẩu là bắt buộc”.			

Luồng B: Số điện thoại hoặc mật khẩu không đúng đăng nhập
User nhập sai thông tin: <ul style="list-style-type: none"> - Hệ thống hiển thị thông báo lỗi: “Tài khoản hoặc mật khẩu không chính xác” nếu không tìm thấy thông tin tài khoản và mật khẩu tương ứng trong Database

Bảng 2.2: UC Đăng nhập

2.4.2. Use Case: Đăng ký

Tên Use Case		Đăng ký	Độ phức tạp: Cao
Mô tả		Cho phép độc giả đăng ký tài khoản mới trong website để có thể truy cập và sử dụng dịch vụ.	
Tác nhân		Độc giả	
Tiền điều kiện		User truy cập vào website.	
Hậu điều kiện	Thành công	Đăng ký thành công, tài khoản được tạo và dữ liệu tài khoản được lưu vào database, hệ thống điều hướng sang giao diện đăng nhập.	
	Lỗi	Đăng ký không thành công.	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính/ Kịch bản chính			
Use case bắt đầu khi User muốn đăng ký tài khoản vào website: <ul style="list-style-type: none"> - Sau khi truy cập website, User chọn đăng ký tài khoản. - Website hiển thị giao diện đăng ký tài khoản yêu cầu nhập những trường: <ul style="list-style-type: none"> + Số điện thoại (bắt buộc) + Họ và tên (bắt buộc) + Địa chỉ của bạn (bắt buộc) + Email (bắt buộc) + Mật khẩu (bắt buộc) 			

- + Xác nhận mật khẩu (bắt buộc)
- Chi tiết yêu cầu:
 - + Số điện thoại phải là số điện thoại hợp lệ (bắt đầu bằng 0 hoặc +84 và có 10 chữ số).
 - + Họ và tên nhập tối đa 50 kí tự.
 - + Địa chỉ của bạn nhập tối đa 243 kí tự.
 - + Email nhập tối đa 50 kí tự, ở cuối phải có đuôi @gmail.
 - + Mật khẩu phải trên 6 kí tự
 - + Xác nhận mật khẩu phải trùng với mật khẩu đã nhập ở trên.
- Website kiểm tra User đã nhập các trường bắt buộc hay chưa, nếu chưa nhập thì thực thi luồng A.
- Website xác nhận các trường đăng ký có hợp lệ hay ko nếu không hợp lệ thì thực thi luồng B.
- User nhập đầy đủ và hợp lệ tất cả các thông tin yêu cầu, sau đó nhấn nút “Đăng ký” để hoàn thành việc đăng ký tài khoản.
- Website tạo tài khoản mới cho User, thông báo đăng ký thành công.

Luồng sự kiện phát sinh/ Kịch bản phát sinh

Luồng A: Không nhập các trường bắt buộc

User bỏ trống các trường bắt buộc

- Website hiện thị inline message tương ứng:
 - + Số điện thoại là bắt buộc
 - + Họ và tên là bắt buộc
 - + Email là bắt buộc
 - + Mật khẩu là bắt buộc
 - + Xác nhận mật khẩu là bắt buộc

Luồng B: Sai điều kiện của các trường

User nhập sai yêu cầu:

- Website hiển thị inline message tương ứng:
 - + Số điện thoại không hợp lệ
 - + Email không hợp lệ
 - + Họ và tên không hợp lệ
 - + Mật khẩu không hợp lệ
 - + Mật khẩu không trùng khớp

Bảng 2.3: UC Đăng ký

2.4.3. Use Case: Đăng xuất

Tên Use Case		Đăng xuất	Độ phức tạp: trung bình
Mô tả		Cho phép User đăng xuất khỏi Website, kết thúc phiên làm việc	
Tác nhân		Độc giả, Admin	
Tiền điều kiện		User đã đăng nhập vào Website	
Hậu điều kiện	Thành công	Đăng xuất thành công khỏi Website	
	Lỗi	Không có	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính/Kịch bản chính			
<p>Usecase này bắt đầu khi User muốn đăng xuất khỏi Website.</p> <ul style="list-style-type: none"> - Tại màn hình trang chủ, User chọn Tên tài khoản ở góc phải màn hình. - Menu tài khoản được mở ra, User chọn "Đăng xuất". - Hệ thống đăng xuất User ra khỏi Website. - Sau khi đăng xuất, Website sẽ chuyển hướng User về trang chủ với trạng thái chưa đăng nhập vào tài khoản. - User có thể tiếp tục sử dụng trang web với các tính năng cơ bản dành cho Độc giả chưa đăng nhập hoặc có thể đăng nhập lại tài khoản nếu muốn. 			

Luồng sự kiện phát sinh/Kịch bản phát sinh*Bảng 2.4: UC Đăng xuất***2.4.4. Use Case: Tìm kiếm sách**

Tên Use Case		Tìm kiếm sách	Độ phức tạp: trung bình
Mô tả		Cho phép User tìm kiếm các sách.	
Tác nhân		Độc giả	
Tiền điều kiện		User đã truy cập vào Website.	
Hậu điều kiện	Thành công	Hiển thị danh sách các sách phù hợp với từ khóa tìm kiếm.	
	Lỗi	Không có	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính/ Kịch bản chính			
<p>Use case bắt đầu khi User muốn tìm kiếm sách.</p> <ul style="list-style-type: none"> - User nhập từ khóa vào ô tìm kiếm “Tìm kiếm”. - Website truy cập cơ sở dữ liệu để tìm kiếm các sách phù hợp với từ khóa mà User nhập. - Nếu tìm thấy sách ,Website hiển thị danh sách sách phù hợp với keyword, bao gồm thông tin cơ bản như: <ul style="list-style-type: none"> + Tên sách + Phí mượn sách sách + Hình ảnh sách - Nếu không tìm thấy sách trùng khớp thực hiện luồng A. - Luồng con User có thể sắp xếp các sách theo mức Phí mượn sách: <ul style="list-style-type: none"> + User sắp xếp sách theo Phí mượn sách bằng cách chọn loại mong muốn từ giao diện lọc: sắp xếp từ cao đến thấp hoặc sắp xếp từ thấp đến cao. 			

<ul style="list-style-type: none"> + Website cập nhật kết quả tìm kiếm và hiển thị danh sách các sách phù hợp với điều kiện lọc đã chọn.
Luồng sự kiện phát sinh/ Kịch bản phát sinh
Luồng A: Không tìm thấy sách
<p>Nếu không có sách nào phù hợp với từ khóa tìm kiếm:</p> <ul style="list-style-type: none"> - Website hiển thị thông báo: “Không tìm thấy sách nào”. - User có thể sửa từ khóa tìm kiếm và thử lại.

Bảng 2.5: UC Tìm kiếm sách

2.4.5. Use Case: Chi tiết sách

Tên Use Case		Chi tiết sách	Độ phức tạp: Cao
Mô tả		Cho phép User có thể xem đầy đủ thông tin về một sách trước khi quyết định mượn hàng.	
Tác nhân		Độc giả	
Tiền điều kiện		User truy cập vào Website.	
Hậu điều kiện	Thành công	User có thể xem đầy đủ thông tin về sách.	
	Lỗi	Thêm sách vào giỏ sách thất bại.	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính/ Kịch bản chính			
<p>Usecase bắt đầu khi User muốn xem chi tiết sách.</p> <ul style="list-style-type: none"> - User chọn một sách cụ thể từ danh sách sách bằng cách nhấn vào tên hoặc hình ảnh sách. - Website hiển thị chi tiết sách, bao gồm các thông tin về: <ul style="list-style-type: none"> + Tên sách. 			

- + Phí mượn sách sách.
- + Mô tả sách
- + Hình ảnh sách.
- + Số lượng sách thêm vào giỏ sách hoặc mượn ngay.
- + Số lượng hàng tồn kho.
- + Số lượng đã đánh Phí mượn sách và số sao của sách đó.
- + Đánh Phí mượn sách của độc giả.
- Luồng con User thực hiện thêm sách vào giỏ sách hoặc mượn ngay:
 - + Nhập số lượng sách muốn mượn.
 - + User có thể nhấn vào nút "Thêm vào giỏ sách" để thêm sách vào giỏ sách hoặc "Mượn ngay" để tiến hành đặt mượn.
 - + Nếu User chưa đăng nhập vào Website thì thực thi luồng A, nếu User đã đăng nhập vào Website thì thực hiện bước tiếp theo.
 - + Website kiểm tra số lượng sách trong kho:
 - Nếu số lượng sách trong giỏ sách không vượt quá số lượng tồn kho, Website tiến hành thêm mới sách vào giỏ sách hoặc chuyển sang giao diện đặt mượn.
 - Nếu số lượng sách trong giỏ sách vượt quá số lượng tồn kho, thì thực thi luồng B.
- User có thể tiếp tục duyệt sách khác hoặc chuyển sang trang giỏ sách/Thanh toán phí mượn.

Luồng sự kiện phát sinh/ Kịch bản phát sinh

Luồng A: User nhấn thêm sản phẩm vào giỏ sách hoặc mượn ngay khi chưa đăng nhập vào thống.

Nếu User chưa đăng nhập:

- Website hiển thị thông báo lỗi với nội dung: “Bạn cần phải đăng nhập để mượn ngay” khi nhấn “Mượn ngay”.

<ul style="list-style-type: none"> – Website hiển thị thông báo lỗi với nội dung: “Bạn cần phải đăng nhập để thêm vào giỏ sách” khi nhấn “Thêm vào giỏ sách”.
Luồng B: Số lượng sách vượt quá số lượng hàng tồn kho.
Khi Website kiểm tra và phát hiện rằng số lượng yêu cầu vượt quá khả năng cung cấp: <ul style="list-style-type: none"> – Website hiển thị thông báo lỗi với nội dung: "Số lượng mượn vượt quá số lượng trong kho."

Bảng 2.6: UC Chi tiết sách

2.4.6. Use Case: Giỏ sách

Tên Use Case		Giỏ sách	Độ phức tạp: Cao
Mô tả		Cho phép User quản lý giỏ sách của mình.	
Tác nhân		Độc giả	
Tiền điều kiện		User đã đăng nhập vào Website và đã có sách trong giỏ sách.	
Hậu điều kiện	Thành công	User có thể thực hiện các thao tác quản lý sách trong giỏ sách.	
	Lỗi	Thay đổi thất bại.	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính/ Kịch bản chính			
Usecase bắt đầu khi User truy cập chức năng giỏ sách. <ul style="list-style-type: none"> – User chọn nút giỏ sách trên giao diện Website. – Website hiển thị giao diện giỏ sách với danh sách các sách mà User đã thêm vào từ các trang sách, bao gồm các thông tin về: <ul style="list-style-type: none"> + Tên sách + Số lượng sách + Phí mượn sách mượn 			

<ul style="list-style-type: none"> + Tổng tiền - User có thể thực hiện các thao tác với từng sách riêng lẻ trong giỏ sách: <ul style="list-style-type: none"> + Chọn checkbox trước sách muốn chọn. + Chọn nút “+” để tăng số lượng sách. + Chọn nút “-” để giảm số lượng sách. + Chọn nút “Xóa” để xóa sách khỏi giỏ sách. - Độc giả có thể thao tác với tất cả sách trong giỏ sách: <ul style="list-style-type: none"> + Chọn checkbox "Chọn tất cả sách" để áp dụng thao tác cho toàn bộ sách. + Chọn button xóa để xóa toàn bộ sách trong giỏ sách. - Khi Độc giả nhấn “Thanh toán phí mượn ” Website kiểm tra số lượng sách trong kho: <ul style="list-style-type: none"> + Nếu số lượng sách trong giỏ sách không vượt quá số lượng tồn kho, Website cập thay đổi về số lượng sách và Phí mượn sách trị Tổng tiền tương ứng với thay đổi số lượng trong giỏ sách. + Nếu số lượng sách trong giỏ sách vượt quá số lượng tồn kho, thì Website thực thi luồng A. - Sau khi thực hiện thành công, Website chuyển sang giao diện đặt mượn.
Luồng sự kiện phát sinh/ Kịch bản phát sinh
Luồng A: Số lượng sách vượt quá số lượng hàng tồn kho
<p>Khi Website kiểm tra và phát hiện rằng số lượng yêu cầu vượt quá khả năng cung cấp:</p> <ul style="list-style-type: none"> - Website hiển thị thông báo lỗi với nội dung: "Số lượng sách yêu cầu vượt quá số lượng tồn kho."

Bảng 2.7: UC Giỏ sách

2.4.7. Use Case: Đặt mượn và Thanh toán phí mượn

Tên Use Case	Đặt mượn	Độ phức tạp: Cao
Mô tả	Cho phép User đặt mượn sách trong giỏ sách của mình	

Tác nhân		Độc giả
Tiền điều kiện		Tác nhân đã đăng nhập vào Website và đã có sách trong giỏ sách
Hậu điều kiện	Thành công	Đặt mượn hàng thành công.
	Lỗi	Đặt mượn và Thanh toán phí mượn không thành công
ĐẶC TẢ CHỨC NĂNG		
Luồng sự kiện chính/ Kịch bản chính		
<p>Use case bắt đầu khi User muốn đặt mượn sách trong giỏ sách của mình.</p> <ul style="list-style-type: none"> – User click chọn “Giỏ sách” trên giao diện Website. – Website hiển thị giao diện giỏ sách với đầy đủ thông tin về các sách trong giỏ sách (Tên sách kèm hình ảnh, Phí mượn sách, số lượng, tổng tiền). – User chọn sách muốn đặt mượn và nhấn vào nút “Thanh toán phí mượn”. – Website hiển thị thông tin Độc giả, Website tiến hành thực hiện Thanh toán phí mượn theo lựa chọn hình thức Thanh toán phí mượn của Độc giả, mặc định là hình thức Thanh toán phí mượn khi mượn sách. – Nếu Độc giả chọn hình thức Thanh toán phí mượn “Thanh toán phí mượn bằng thẻ ngân hàng”, Website chuyển sang giao diện Thanh toán phí mượn của VnPay hoặc Momo. <ul style="list-style-type: none"> + Nếu Thanh toán phí mượn thành công, Website hiển thị thông báo “Đặt mượn thành công”. + Nếu Thanh toán phí mượn thất bại, Website thực hiện thất bại thì thực hiện theo luồng A – Website tiến hành lưu thông tin đặt mượn vào cơ sở dữ liệu, hiển thị thông báo “Đặt mượn thành công”. 		
Luồng sự kiện phát sinh/ Kịch bản phát sinh		
Luồng A: Thanh toán phí mượn thất bại		
Nếu Thanh toán phí mượn thất bại:		

- Website hiển thị thông báo "Thanh toán phí mượn thất bại" và chuyển về màn hình giỏ sách để Độc giả có thể thực hiện Thanh toán phí mượn lại hoặc đổi phương thức Thanh toán phí mượn.

Bảng 2.8: UC Đặt mượn và Thanh toán phí mượn

2.4.8. Use Case: Chi tiết đơn mượn

Tên Use Case		Chi tiết đơn mượn	Độ phức tạp: Cao
Mô tả		Cho phép các User theo dõi đơn mượn đã đặt trước đó.	
Tác nhân		Độc giả	
Tiền điều kiện		User đã đăng nhập vào Website	
Hậu điều kiện	Thành công	Hiển thị chi tiết đơn mượn đã đặt trước đó	
	Lỗi	Thao tác thất bại	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính/ Kịch bản chính			
<p>Use case bắt đầu khi tác nhân xem thông tin chi tiết về các đơn mượn đã được đặt.</p> <ul style="list-style-type: none"> – User có thể xem được chi tiết các thông tin của đơn mượn như: tên người nhận, địa chỉ mượn sách, số điện thoại, email, trạng thái đơn mượn. – User có thể thực hiện các hành động như hủy đơn mượn, đánh Phí mượn sách. Website cung cấp luồng con cho các thao tác này: <ul style="list-style-type: none"> + User chọn button “Hủy đơn mượn”, luồng con Hủy đơn mượn sẽ được thực hiện. + Nếu sách ở trạng thái “Đã nhận được hàng” User có thể chọn button “Đánh Phí mượn sách”, luồng con “Đánh Phí mượn sách sách” sẽ được thực hiện. – Luồng con “Hủy đơn mượn”: 			

<ul style="list-style-type: none"> + Website tiến hành kiểm tra nếu sách đã được giao hàng thì thực thi luồng A, nếu chưa được giao thì Website sẽ hủy đơn mượn, cập nhật trạng thái đơn mượn thành “Đã Hủy” và hiển thị “Đã hủy đơn mượn”. - Luồng con “Đánh Phí mượn sách sách”: + Website hiển thị pop up Đánh Phí mượn sách sách yêu cầu nhập những thông tin: chất lượng sách, nội dung đánh Phí mượn sách, ảnh. + User điền thông tin vào form. + Website kiểm tra User đã nhập các trường bắt buộc hay chưa, nếu chưa nhập thì thực thi luồng B.
Luồng sự kiện phát sinh/ Kịch bản phát sinh
Luồng A: Hủy đơn không thành công
<p>Nếu sách đã được giao:</p> <ul style="list-style-type: none"> - User không thể tiến hành hủy đơn mượn. - Website hiển thị thông báo "Hủy đơn mượn thất bại" và không thực hiện hành động hủy.
Luồng B: Không nhập các trường bắt buộc
User bỏ trống các trường bắt buộc. Website hiển thị thông báo “Vui lòng điền đầy đủ thông tin đánh Phí mượn sách và xếp hạng sách.”

Bảng 2.9: UC Chi tiết đơn mượn

2.4.9. Use Case: Cập nhật tài khoản

Tên Use Case	Cập nhật tài khoản	Độ phức tạp: Cao
Mô tả	User có thể xem và chỉnh sửa thông tin cá nhân hoặc thay đổi mật khẩu.	
Tác nhân	Độc giả	
Tiền điều kiện	User đăng nhập vào Website.	

Hậu điều kiện	Thành công	Xem, thay đổi được một số thông tin và đổi mật khẩu.
	Lỗi	Không thay đổi được thông tin hoặc mật khẩu.

ĐẶC TẢ CHỨC NĂNG

Luồng sự kiện chính/Kịch bản chính

Usecase bắt đầu khi User muốn cập nhật thông tin tài khoản.

- Tại màn hình trang chủ, tác nhân chọn vào tên tài khoản ở góc phải màn hình
- Một menu tài khoản được mở ra, User chọn tùy chọn “Cập nhật tài khoản”.
- Website hiển thị giao diện thông tin cá nhân bao gồm các trường thông tin như:
 - + Số điện thoại
 - + Họ tên
 - + Địa chỉ
- Khách có thể thực hiện các thao tác chỉnh sửa thông tin cá nhân hoặc thay đổi mật khẩu.
- Luồng con chỉnh sửa thông tin tài khoản:
 - + User chỉnh sửa các thông tin gồm: địa chỉ, ngày sinh, giới tính, ảnh đại diện.
 - + Sau khi chỉnh sửa xong User bấm Lưu, Website sẽ lưu lại các thông tin của Độc giả về cơ sở dữ liệu và hiển thị các thông tin mới ra lại màn hình.
 - + Website hiển thị thông báo "Cập nhật thành công".

Luồng sự kiện phát sinh/Kịch bản phát sinh

Luồng A: Không nhập các trường bắt buộc

User bỏ trống các trường bắt buộc

- Website hiển thị inline message tương ứng:
 - + Email hiện tại là bắt buộc

- + Địa chỉ là bắt buộc
- + Số điện thoại là bắt buộc

Bảng 2.10: UC Cập nhật tài khoản

2.4.10. Use Case: Thống kê

Tên Use Case		Thống kê	Độ phức tạp: Trung bình
Mô tả		Cho phép admin xem Phí mượn mượn hàng của ngày, của tháng, của năm	
Tác nhân		Admin	
Tiền điều kiện		Admin truy cập vào Website	
Hậu điều kiện	Thành công	Hiện thị được các thông tin thống kê	
	Lỗi	Không có	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính/Kịch bản chính			
<p>Usecase bắt đầu khi người quản trị muốn xem thống kê của website</p> <ul style="list-style-type: none"> - Website hiển thị danh sách sách hiện có trên Website, bao gồm các thông tin: <ul style="list-style-type: none"> + Website sẽ hiển thị tổng User, số đơn, số sách mượn được, Phí mượn của tháng. + Website hiển thị thêm biểu đồ đường thể hiện số sách mượn và Phí mượn được theo tháng. <ul style="list-style-type: none"> o User có thể ẩn biểu đồ đường một trong hai trường dữ liệu trên. 			

Bảng 2.11: UC Thống kê

2.4.11. Use Case: Quản lý sách

Tên Use Case		Quản lý sách	Độ phức tạp: Cao
Mô tả		Cho phép Admin có thể thêm, sửa hoặc xóa sách trên Website.	
Tác nhân		Admin	
Tiền điều kiện		Admin đăng nhập vào Website quản lý và chọn mục “Sách”.	
Hậu điều kiện	Thành công	Thêm, sửa hoặc xóa sách thành công.	
	Lỗi	Thêm, sửa hoặc xóa sách thất bại.	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính/Kịch bản chính			
<p>Usecase bắt đầu khi User muốn thêm, sửa, xoá sách.</p> <ul style="list-style-type: none"> – Website hiển thị danh sách các sách hiện có, bao gồm các thông tin: <ul style="list-style-type: none"> + Tên và ảnh sách + Phí mượn sách sách + Danh mục + Tình trạng + Thao tác: thêm mới, cập nhật và xóa – Admin có thể thực hiện thêm sách mới, sửa sách hoặc xóa sách. Website cung cấp 3 luồng con cho các thao tác này: <ul style="list-style-type: none"> + User chọn hành động “Thêm mới sách”, luồng con thêm sách sẽ được thực hiện. + User chọn hành động “Sửa sách”, luồng con Sửa sách sẽ được thực hiện. + User chọn hành động “Xoá sách”, luồng con Xoá sách sẽ được thực hiện. – Luồng con “Thêm sách”: <ul style="list-style-type: none"> + Admin chọn button “Thêm sách” trên giao diện quản lý sách. + Website hiển thị form yêu cầu nhập các thông tin thay đổi cho sách mới, bao gồm: <ul style="list-style-type: none"> ○ Tên sách ○ Phí mượn sách sách ○ Số lượng sách ○ Mô tả sách ○ Hình ảnh sách 			

- Danh mục
- Thuộc tính sách
- + User điền thông tin vào form.
- + Website kiểm tra User đã nhập các trường bắt buộc hay chưa, nếu chưa nhập thì thực thi luồng A.
- + Website kiểm tra tính hợp lệ của thông tin:
 - Phí mượn sách và số lượng phải là số dương.
 - Hình ảnh sách phải là định dạng ảnh hợp lệ (JPEG, PNG).
 - Nếu có bất kỳ thông tin nào không hợp lệ, Website thực thi luồng B.
- + Khi tất cả thông tin hợp lệ, User chọn "Lưu".
- + Sau khi hoàn tất sẽ tạo ra sách mới và lưu lại vào cơ sở dữ liệu.
- + Website hiển thị thông báo “Thêm sách thành công”.
- + Website hiển thị thông tin sách mới trong danh sách sách.
- Luồng con Cập nhật sách:
 - + Admin chọn button “Cập nhật” trên giao diện quản lý sách
 - + Website sẽ hiển thị dữ liệu sách mà trước đó User đã nhập, cho phép người quản trị chỉnh sửa các trường thông tin của sách:
 - Tên sách
 - Phí mượn sách sách
 - Số lượng tồn kho
 - Mô tả sách
 - Hình ảnh sách
 - + Website kiểm tra tính hợp lệ của các thông tin mới tương tự như trong luồng con “Thêm sách”
 - + Sau khi chỉnh sửa, người quản trị bấm “Lưu”.
 - + Nếu thông tin hợp lệ, Website cập nhật sách với các thay đổi vào cơ sở dữ liệu sau đó hiển thị thông báo “Sửa đổi sách thành công”.
 - + Website hiển thị thông tin sách đã cập nhật trong danh sách.
- Luồng con Xoá sách:
 - + Admin chọn button “Xóa” trên giao diện quản lý sách.
 - + Website hiển thị thông báo xác nhận với nội dung: "Bạn có muốn xóa sách ‘Tên sách’?"
 - User chọn “Không”, Website quay lại giao diện danh sách sách.
 - User chọn “Xóa”, Website tiến hành xóa sách khỏi cơ sở dữ liệu.

<ul style="list-style-type: none"> ○ Sau khi xóa, Website hiển thị “Xóa sách thành công” và cập nhật lại danh sách sách, không còn hiển thị sách đã xóa.
Luồng sự kiện phát sinh/Kịch bản phát sinh
Luồng A: Không nhập các trường bắt buộc
User bỏ trống các trường bắt buộc. <ul style="list-style-type: none"> - Website hiển thị inline message tương ứng: <ul style="list-style-type: none"> + Tên sách là bắt buộc. + Danh mục là bắt buộc.
Luồng B: Sai điều kiện của các trường
User nhập sai yêu cầu. <ul style="list-style-type: none"> - Website hiển thị inline message tương ứng: <ul style="list-style-type: none"> + Phí mượn sách trị không hợp lệ. + Sai định dạng

Bảng 2.12: UC Quản lý sách

2.4.12. Use Case: Quản lý danh mục sách

Tên Use Case		Quản lý danh mục sách	Độ phức tạp: Cao
Mô tả		Cho phép Admin có thể thêm, sửa hoặc xóa các danh mục trên Website.	
Tác nhân		Admin.	
Tiền điều kiện		Admin đăng nhập vào Website quản lý và chọn mục “Danh mục”.	
Hậu điều kiện	Thành công	Thêm, sửa hoặc xóa danh mục thành công.	
	Lỗi	Thêm danh mục không thành công.	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính/ Kịch bản chính			
Use case bắt đầu khi Admin muốn Thêm, sửa hoặc xóa danh mục:			

- Website hiển thị giao diện quản lý danh mục với các thông tin bao gồm:
 - + Tên danh mục.
 - + Các nút
 - o Chi tiết: Hiển thị các danh mục này.
 - o Cập nhật: Cập nhật thông tin danh mục.
 - o Xóa: Xóa danh mục.
- Luồng con Thêm danh mục:
 - + Admin nhấn nút Thêm danh mục và thực hiện các thao tác điền tên danh mục để tạo mới một danh mục
 - + Nếu thành công Website hiển thị thông báo “Thành công” và thêm danh mục mới vào danh sách, nếu thất bại thì thực hiện luồng A.
- Luồng con: Cập nhật thông tin danh mục.
 - + Admin nhấn nút “Cập nhật” Website hiển thị tên danh mục hiện có.
 - + Admin thực hiện điền thông tin danh mục mới. Nếu thành công Website hiển thị thông báo “Thành công” và thêm mới lên danh sách, nếu thất bại thì thực hiện theo luồng A.
- Luồng con Xóa danh mục :
 - + Admin nhấn nút “Xóa” Website hiển thị thông báo “Bạn có muốn xóa danh mục?” và hai nút “Có” và “Không” .
 - o Nếu quản trị ấn nút “Có” Website tiến hành xóa danh mục khỏi cơ sở dữ liệu và cập nhật lại danh sách.
 - o Nếu quản trị ấn nút “Không” Website giữ nguyên danh mục trong danh sách.

Luồng sự kiện phát sinh/ Kịch bản phát sinh

Luồng A: Thêm danh mục không thành công

Nếu Admin bỏ trống các trường bắt buộc

- Website hiển thị thông báo “Tên danh mục không được để trống”.

Bảng 2.13: UC Quản lý danh mục

2.4.13. Use Case: Quản lý đơn mượn

Tên Use Case		Quản lý đơn mượn	Độ phức tạp: Cao
Mô tả		Cho phép Admin xem chi tiết và thay đổi trạng thái đơn mượn.	
Tác nhân		Admin.	
Tiền điều kiện		Admin đăng nhập vào Website quản lý và chọn mục “Đơn mượn”.	
Hậu điều kiện	Thành công	Admin thực hiện được chức năng quản lý đơn mượn.	
	Lỗi	Thay đổi trạng thái đơn mượn không thành công.	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính/ Kịch bản chính			
<p>Use case bắt đầu khi Admin muốn xem danh sách đơn mượn hoặc thay đổi trạng thái đơn mượn:</p> <ul style="list-style-type: none"> – Website hiển thị giao diện quản lý đơn mượn với các thông tin bao gồm: tên sách, người đặt, tổng tiền, trạng thái, thao tác xem chi tiết. – Admin chọn đơn mượn muốn xem chi tiết. <p>+ Website hiển thị giao diện chi tiết đơn mượn với các thông tin sau:</p> <ul style="list-style-type: none"> ○ Thông tin các sách: tên, hình ảnh, Phí mượn sách tiền của từng sách. ○ Tên Độc giả ○ Số điện thoại ○ Địa chỉ mượn sách ○ Phương thức Thanh toán phí mượn ○ Trạng thái ○ Tổng tiền 			

<ul style="list-style-type: none"> + Admin có thể thay đổi trạng thái đơn mượn: Chờ xác nhận, chuẩn bị hàng, đang giao hàng, đã nhận được hàng. <ul style="list-style-type: none"> ○ Website cập nhật lại trạng thái đơn mượn trong giao diện chi tiết và hiển thị thông báo “Đơn mượn đã được cập nhật”. + Admin nhấn nút “Hủy đơn mượn” để hủy đơn mượn. <ul style="list-style-type: none"> ○ Website tiến hành kiểm tra nếu sách đã được giao hàng thì thực thi luồng A, nếu chưa được giao thì Website tiến hành hủy đơn mượn, cập nhật trạng thái đơn mượn thành “Đã Hủy” và hiển thị thông báo “Đã hủy đơn mượn”.
Luồng sự kiện phát sinh/ Kịch bản phát sinh
Luồng A: Hủy đơn không thành công
<p>Nếu sách ở trạng thái “Đang giao hàng”:</p> <ul style="list-style-type: none"> – User không thể tiến hành hủy đơn mượn. – Website hiển thị thông báo "Hủy đơn mượn thất bại" và không thực hiện hủy.

Bảng 2.14: UC Quản lý đơn mượn

2.4.14. Use Case: Quản lý User

Tên Use Case		Quản lý User	Độ phức tạp: trung bình
Mô tả		Cho phép Admin có thể xem thông tin và khóa tài khoản của User.	
Tác nhân		Admin.	
Tiền điều kiện		Admin đăng nhập vào Website quản lý và chọn mục “User”.	
Hậu điều kiện	Thành công	Admin có thể xem thông tin và khóa tài khoản User.	
	Lỗi	Không có	
ĐẶC TẢ CHỨC NĂNG			

Luồng sự kiện chính/ Kịch bản chính

Use case bắt đầu khi Admin muốn xem thông tin User:

- Website hiển thị giao diện quản lý đơn mượn với các thông tin bao gồm:
 - + Tên User
 - + Email
 - + Số điện thoại
 - + Địa chỉ

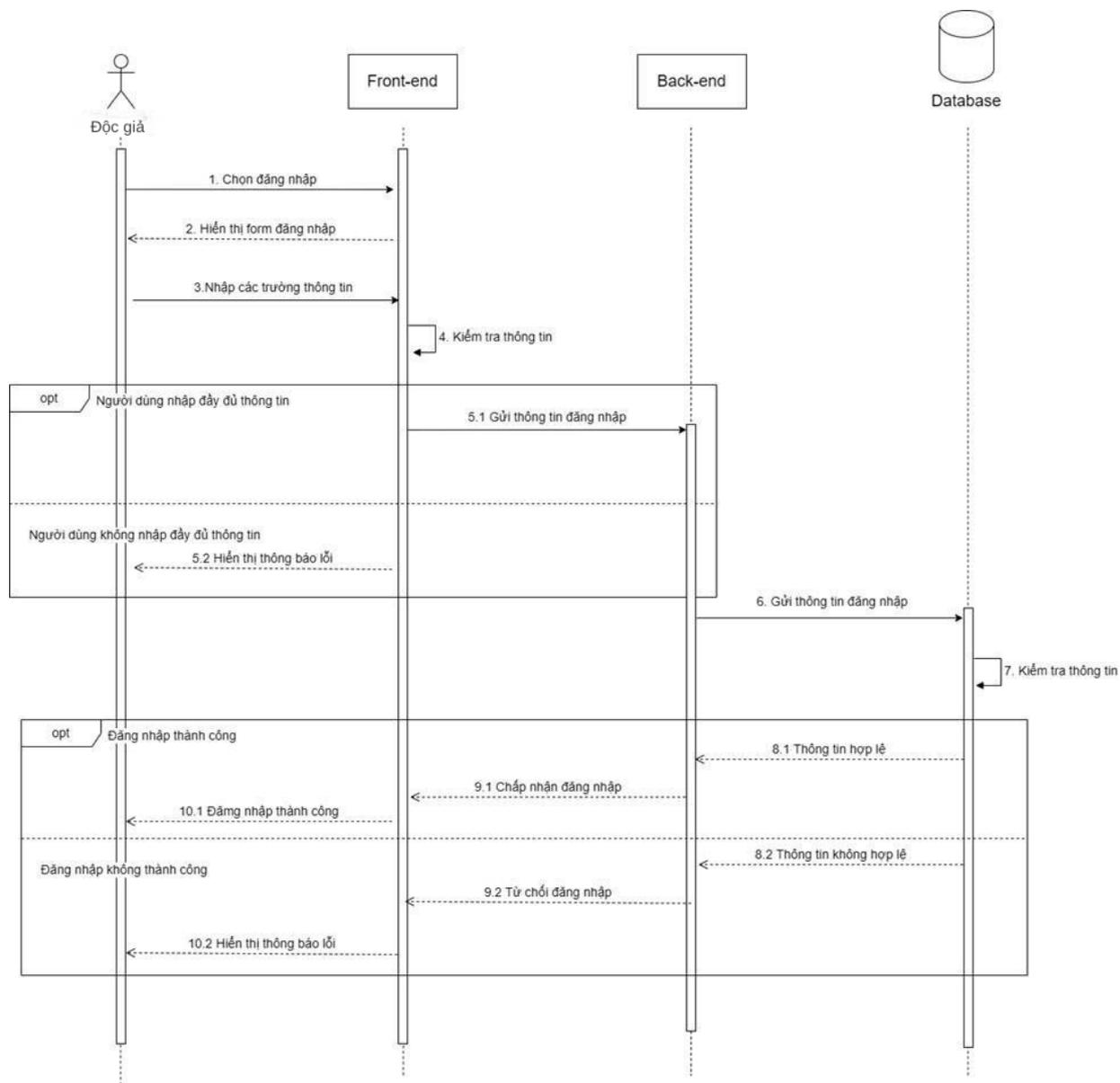
Luồng sự kiện phát sinh/ Kịch bản phát sinh

Bảng 2.15: UC Quản lý User

CHƯƠNG 3. THIẾT KẾ CHỨC NĂNG VÀ CƠ SỞ DỮ LIỆU

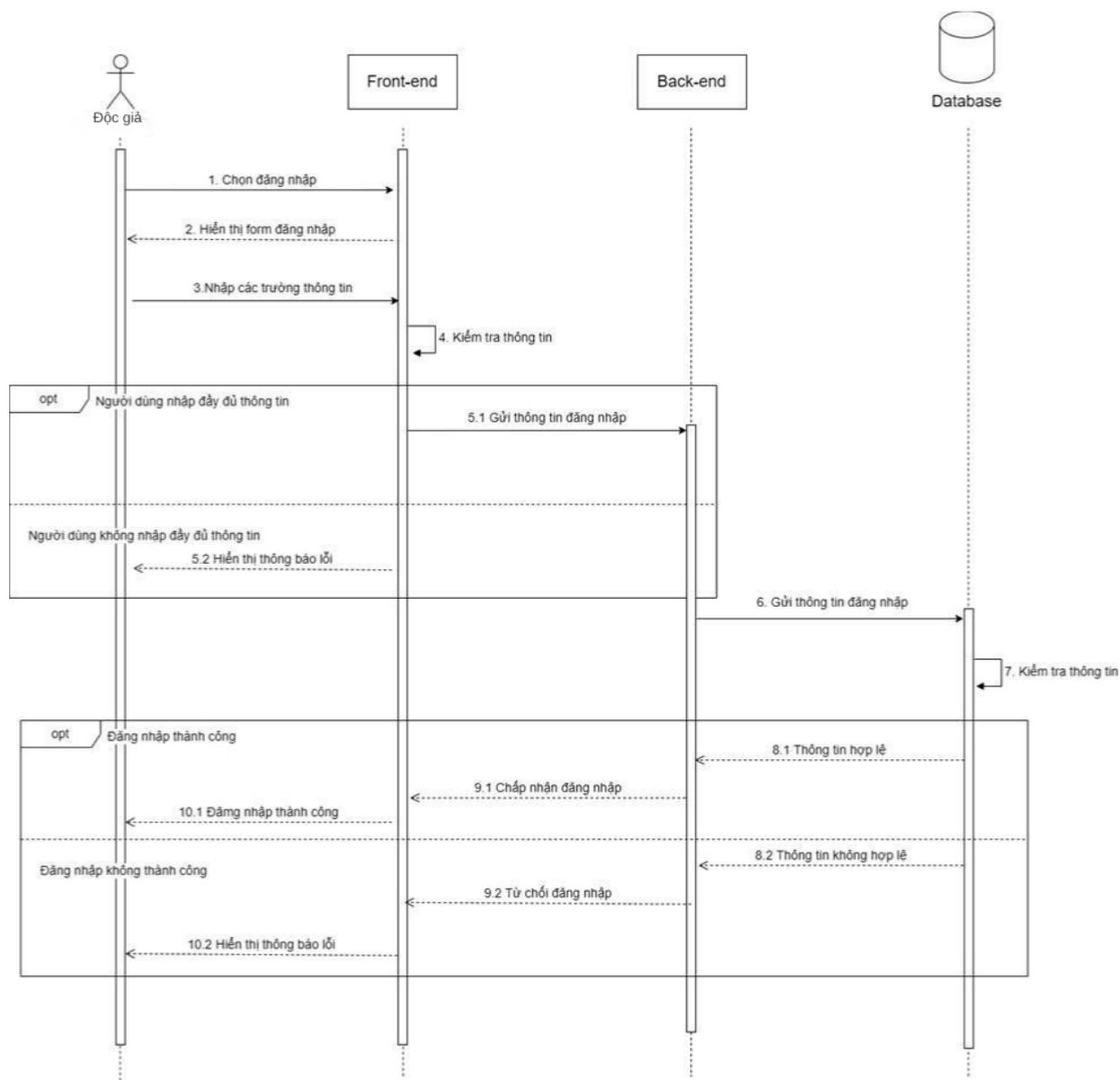
3.1. Biểu đồ tuần tự

3.1.1. Biểu đồ tuần tự chức năng Đăng nhập



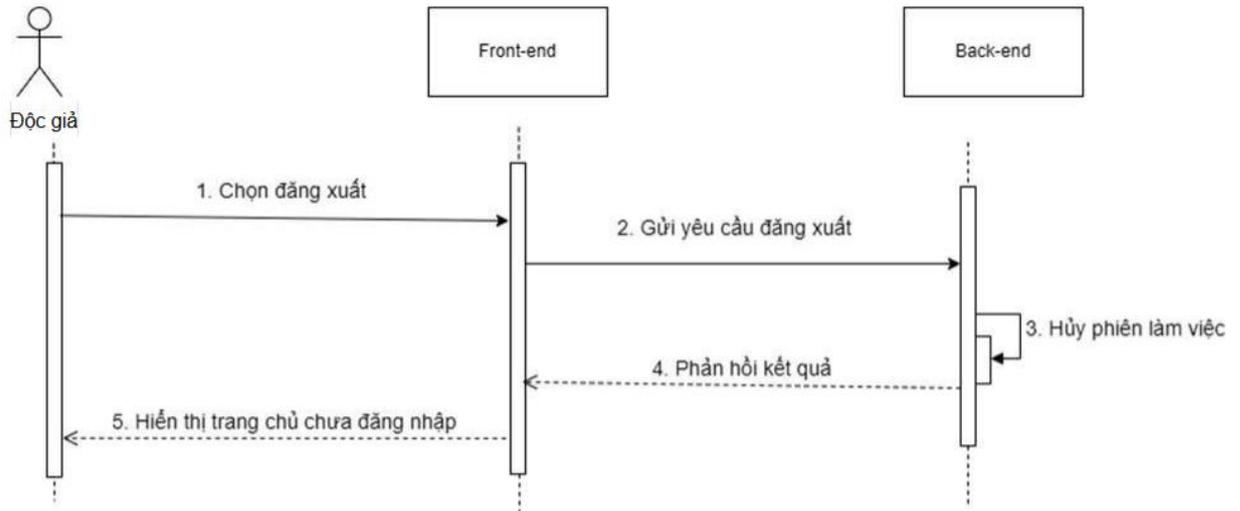
Hình 3.1: Biểu đồ tuần tự chức năng Đăng nhập

3.1.2. Biểu đồ tuần tự chức năng Đăng ký



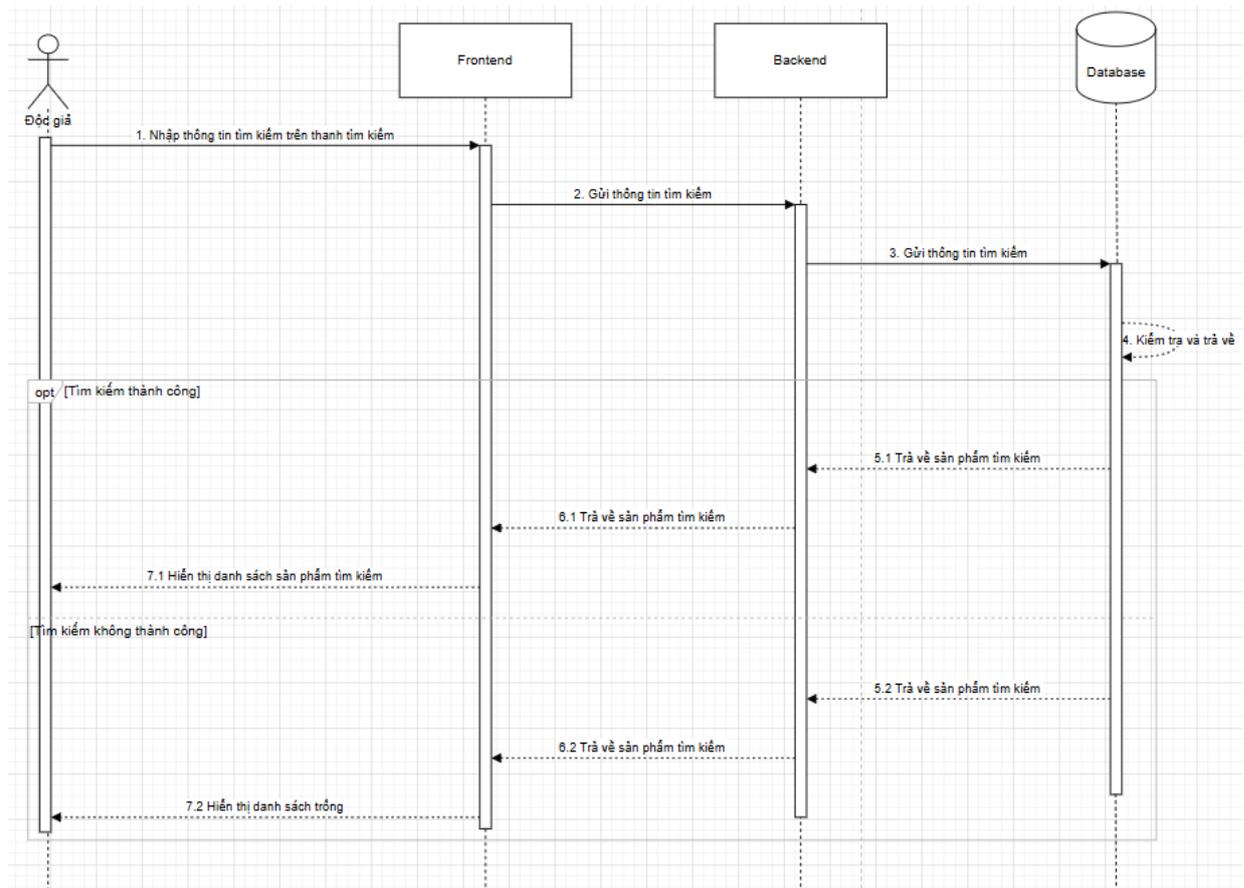
Hình 3.2: Biểu đồ tuần tự chức năng Đăng ký

3.1.3. Biểu đồ tuần tự chức năng Đăng xuất



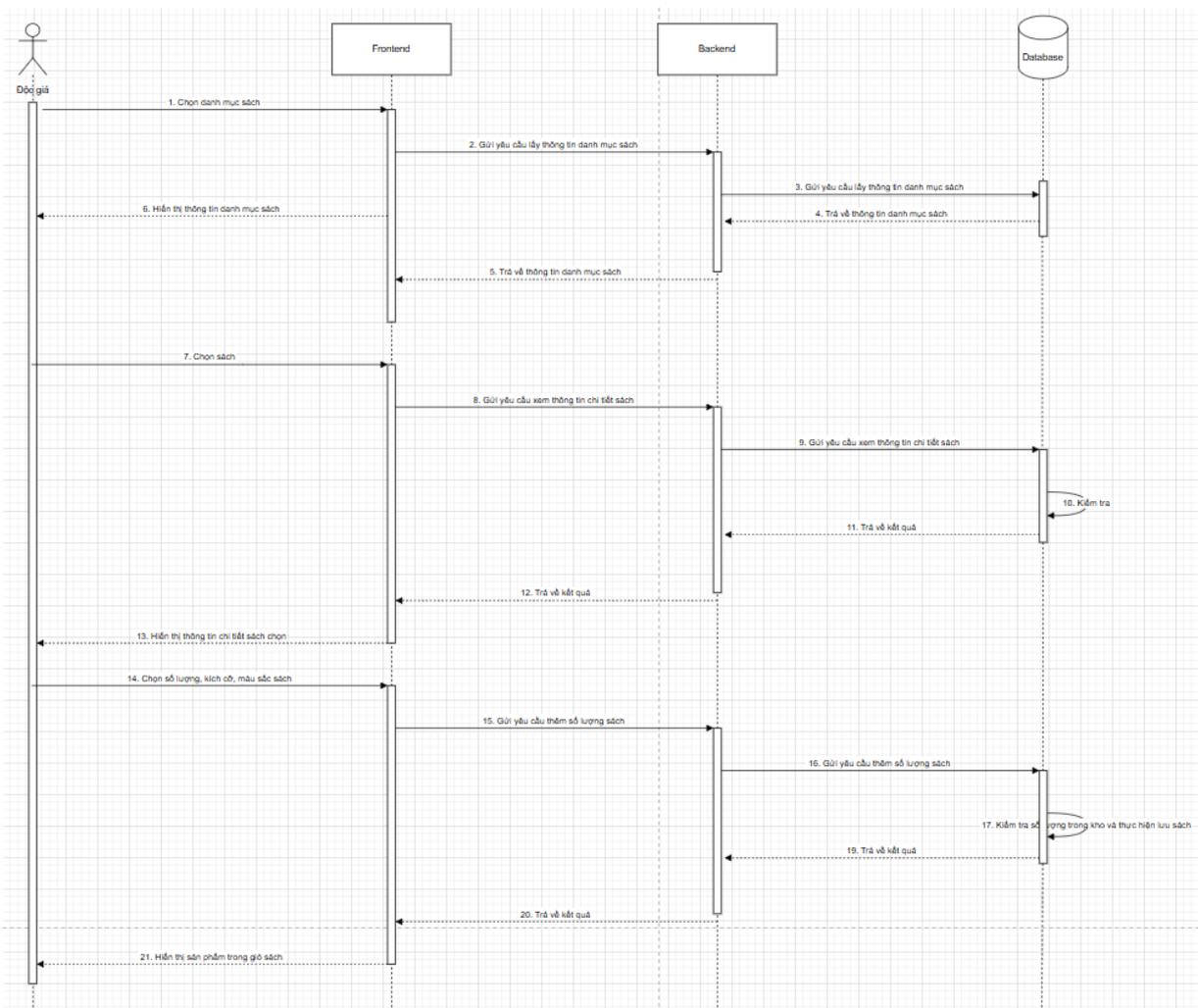
Hình 3.3: Biểu đồ tuần tự chức năng Đăng xuất

3.1.4. Biểu đồ tuần tự chức năng Tìm kiếm sách

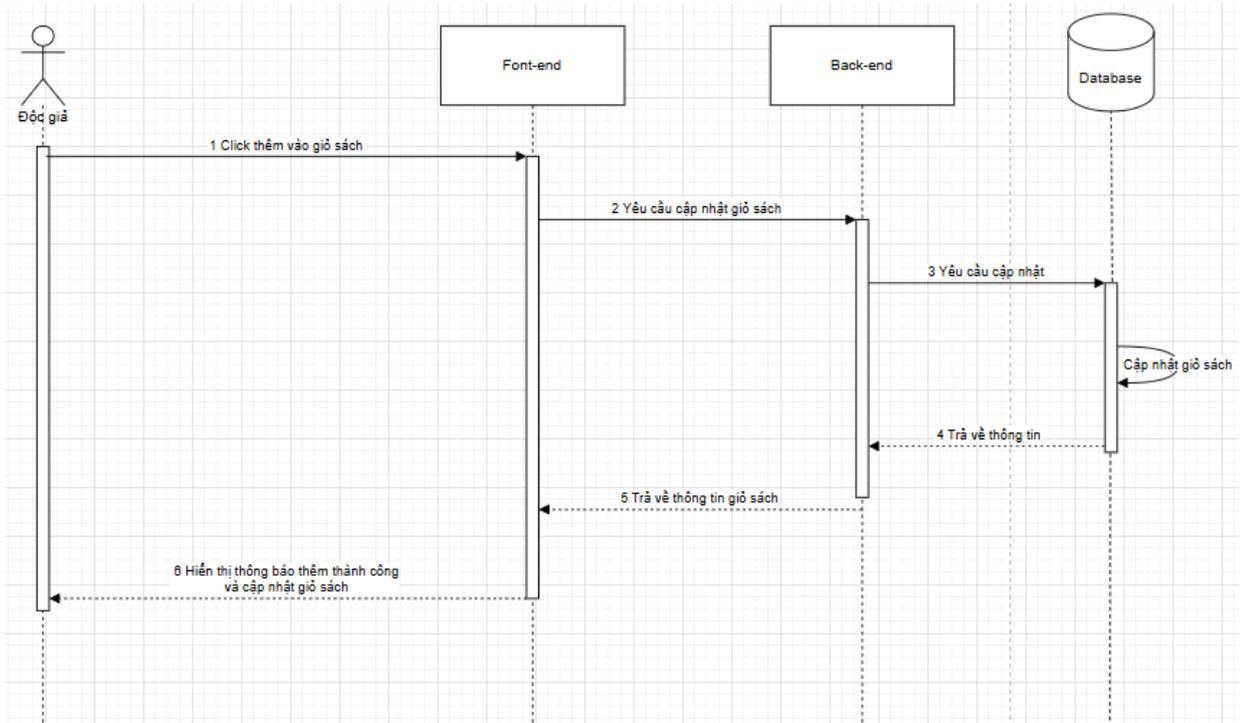


Hình 3.3: Biểu đồ tuần tự chức năng Tìm kiếm sách

3.1.5. Biểu đồ tuần tự chức năng Chi tiết sách và Thêm vào giỏ sách



Chi tiết sách

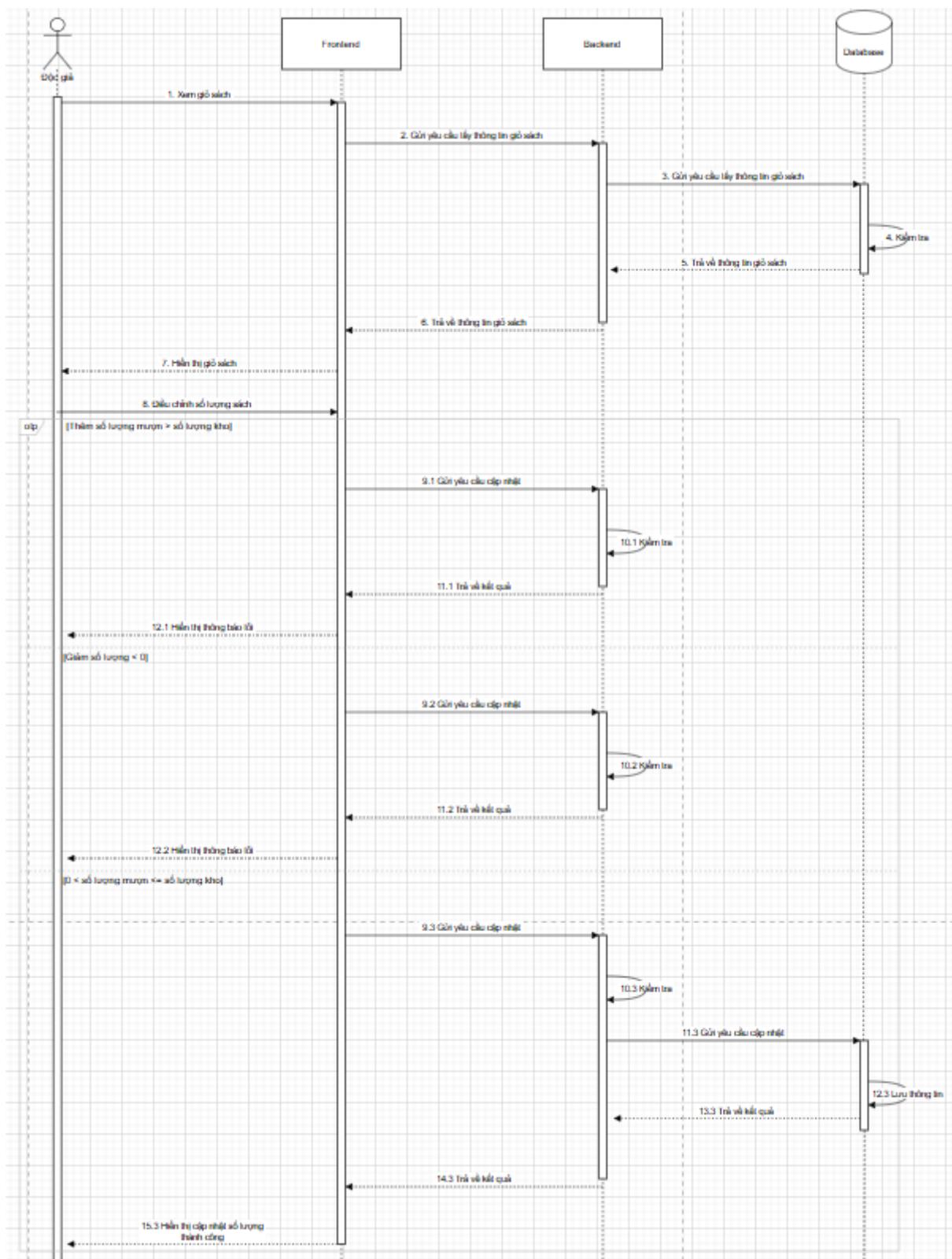


Thêm vào giỏ sách

Hình 3.4: Biểu đồ tuần tự chức năng Chi tiết sách

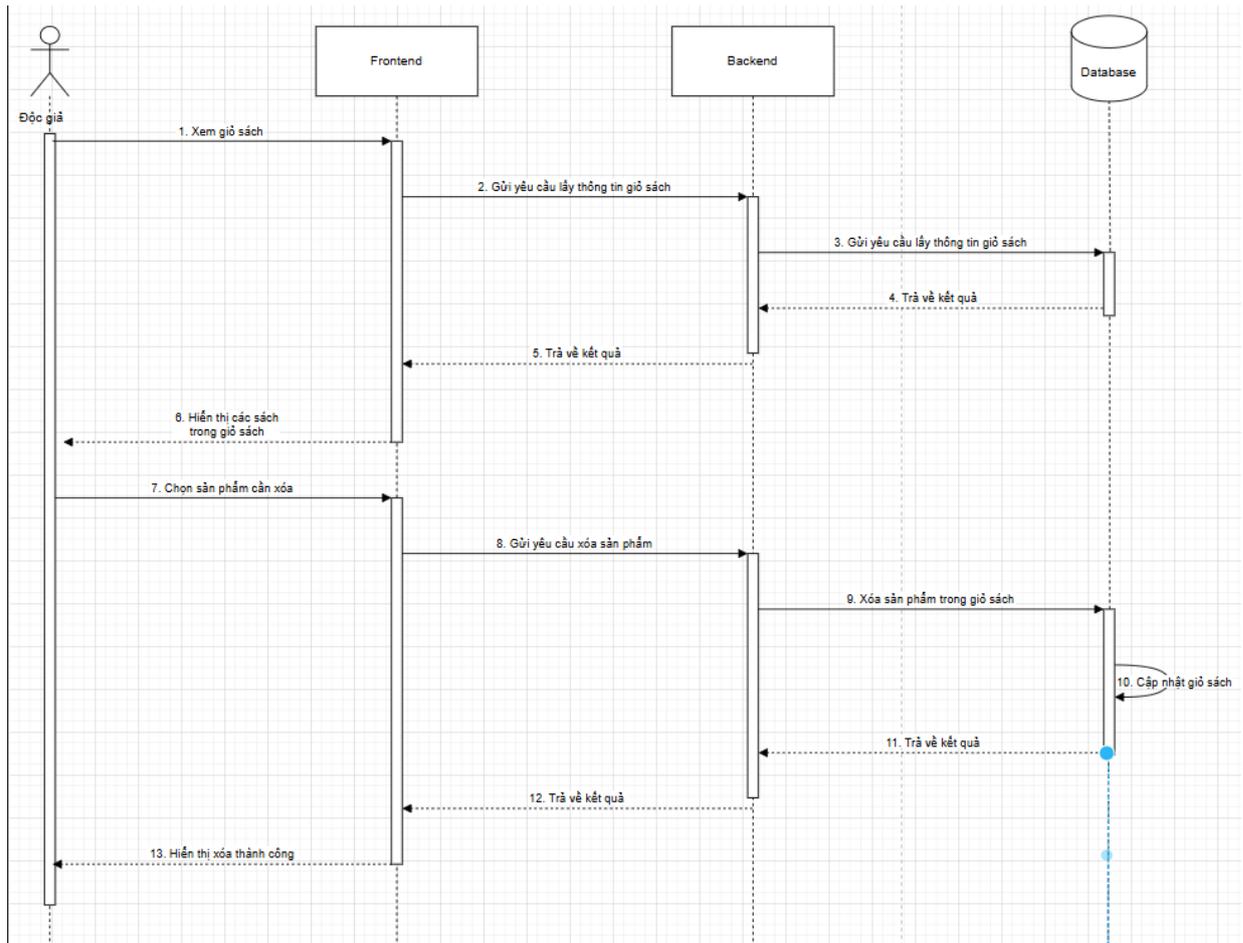
3.1.6. Biểu đồ tuần tự chức năng Giỏ sách

Chức năng Thêm sách vào giỏ sách



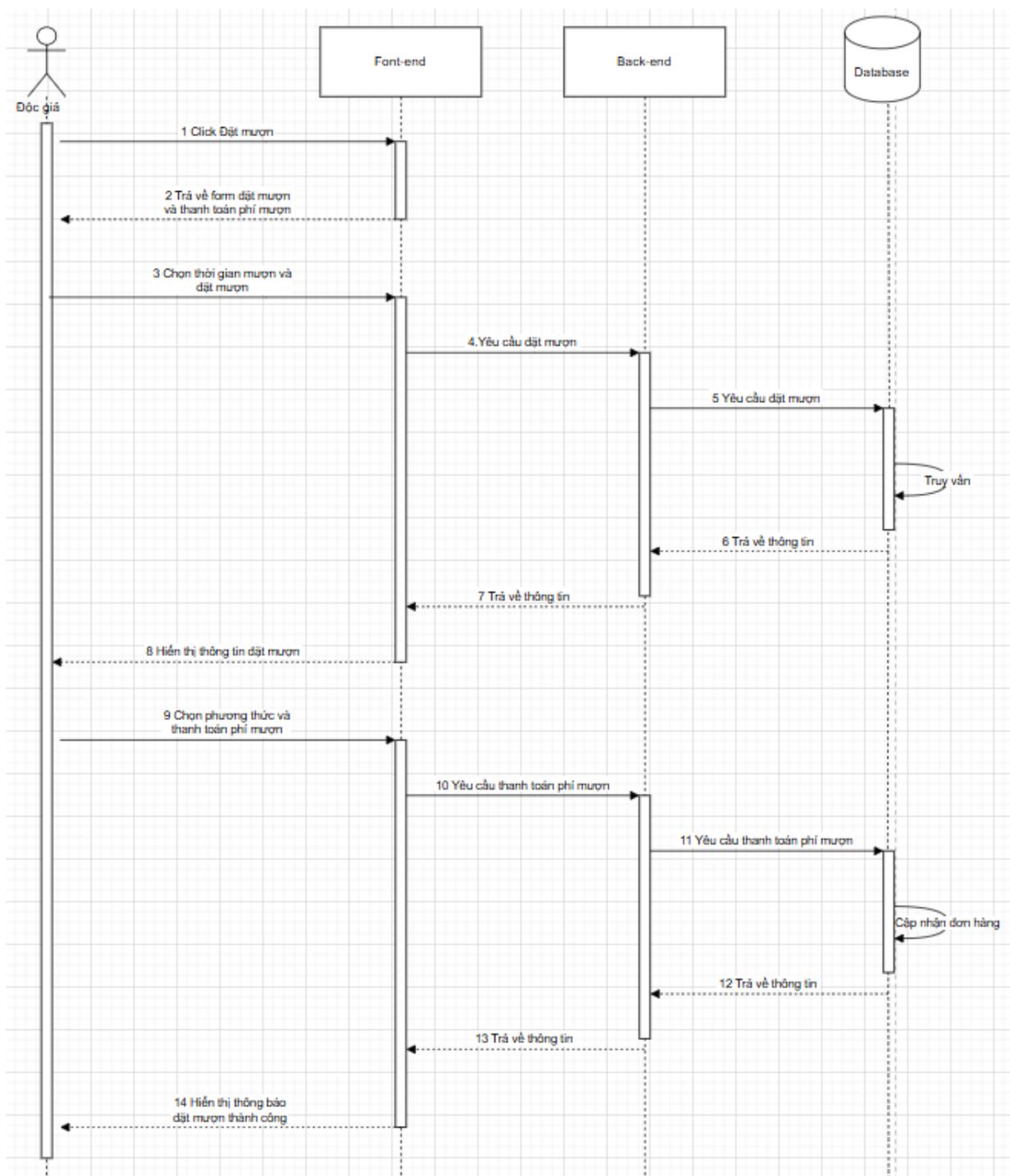
Hình 3.5: Biểu đồ tuần tự chức năng Thêm sách vào giỏ sách

Chức năng Xóa sách khỏi giỏ sách



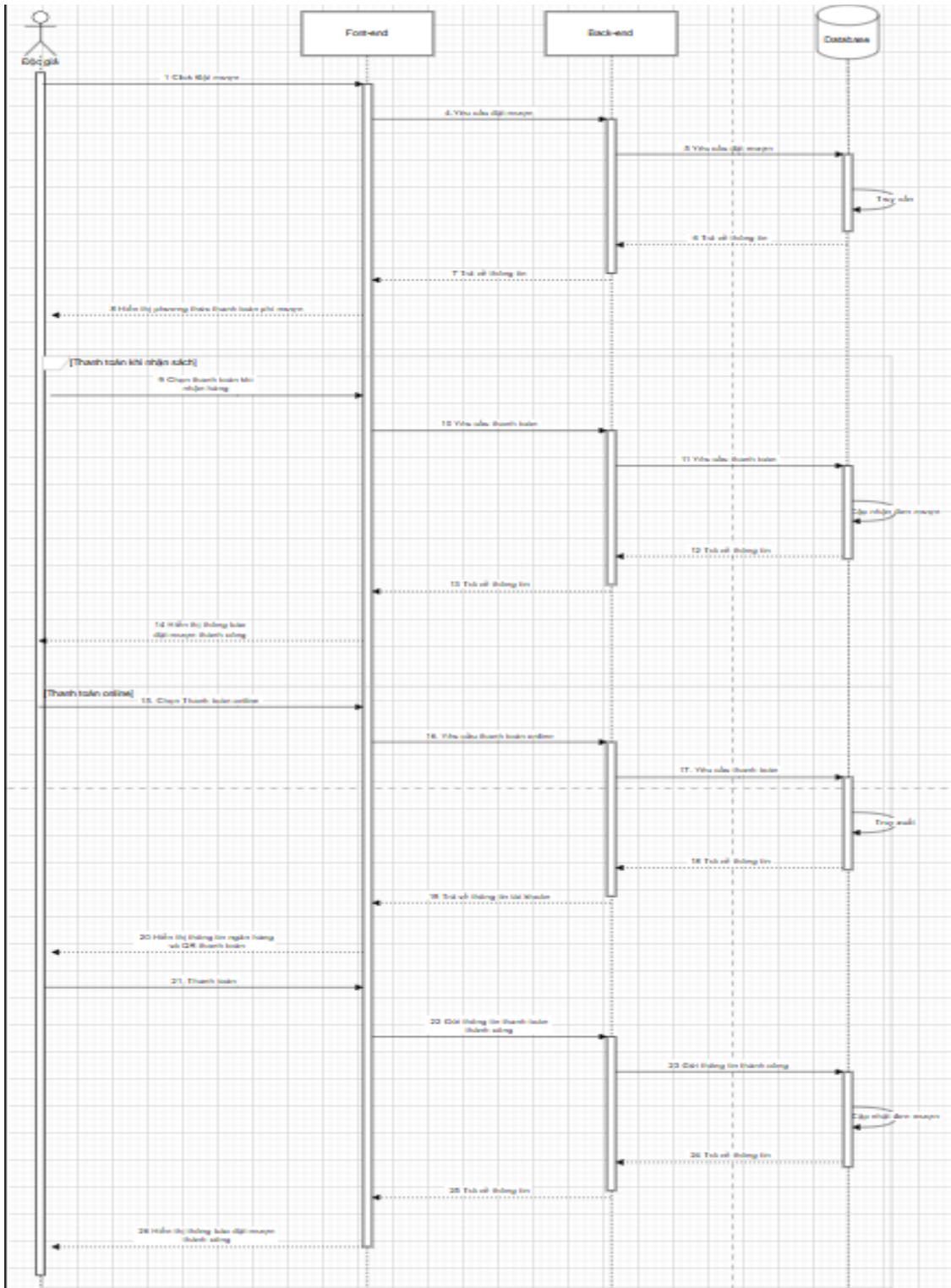
Hình 3.6: Biểu đồ tuần tự chức năng Xóa sách khỏi giỏ sách

3.1.7. Biểu đồ tuần tự chức năng Đặt mượn



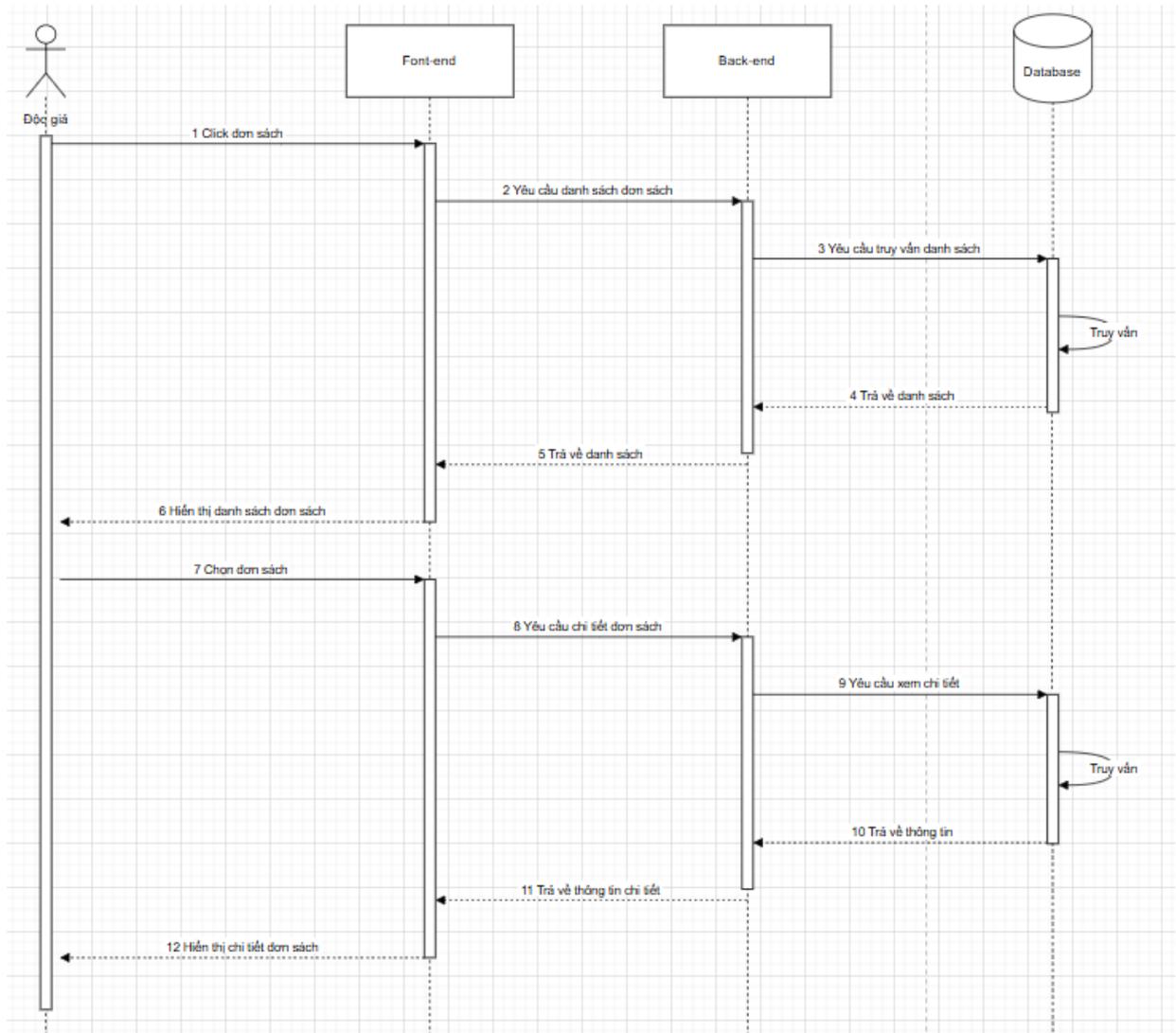
Hình 3.7: Biểu đồ chức năng Đặt mượn

3.1.8. Biểu đồ tuần tự chức năng Thanh toán phí mượn



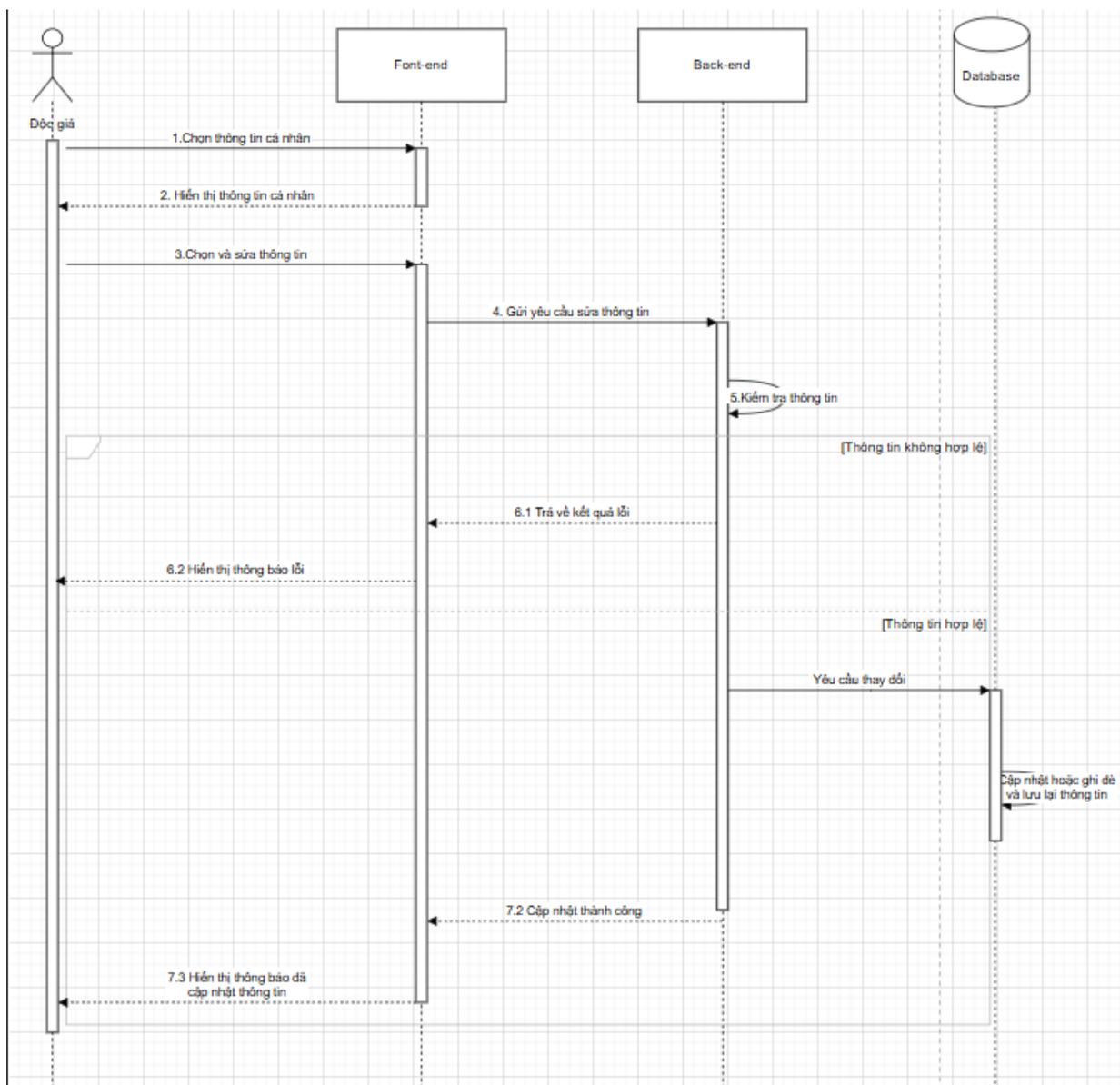
Hình 3.8: Biểu đồ tuần tự chức năng Thanh toán phí mượn

3.1.9. Biểu đồ tuần tự chức năng Xem chi tiết đơn mượn



Hình 3.9: Biểu đồ tuần tự chức năng Xem chi tiết đơn mượn

3.1.10. Biểu đồ tuần tự chức năng Cập nhật thông tin cá nhân

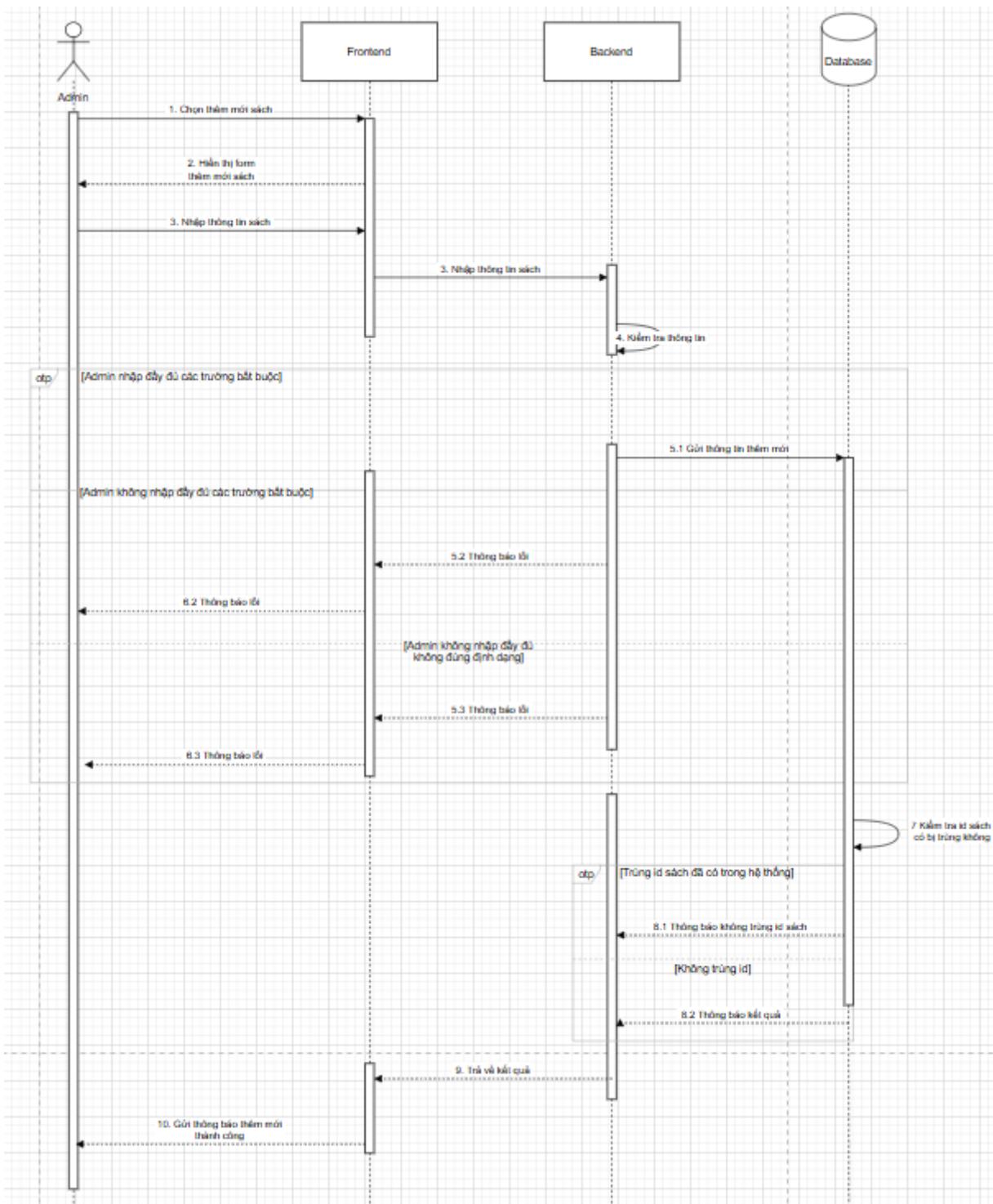


Hình 3.10: Biểu đồ tuần tự chức năng Cập nhật thông tin cá nhân

**Biểu đồ tuần tự admin*

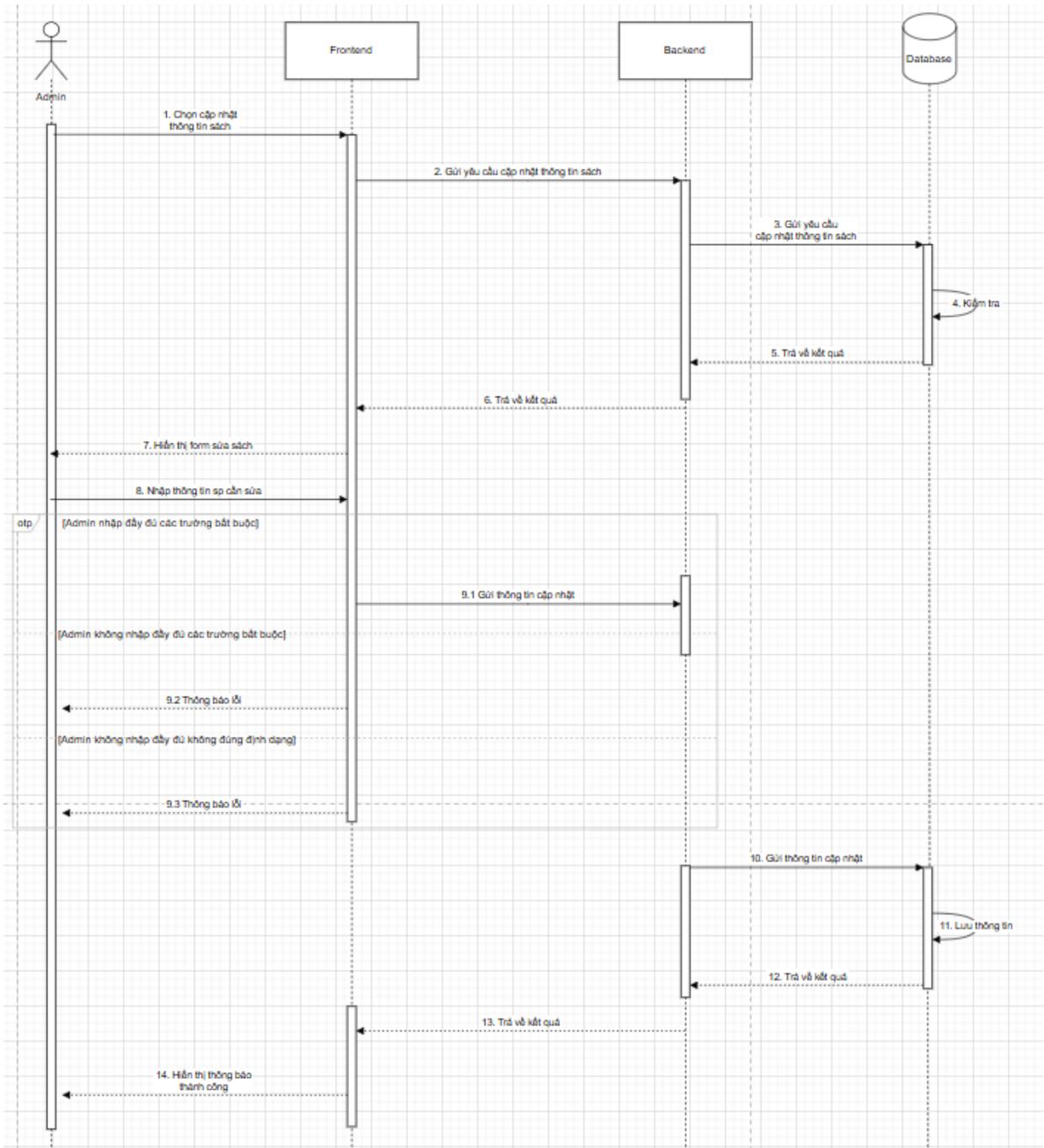
3.1.11. Biểu đồ tuần tự chức năng Quản lý sách

Chức năng thêm mới sách



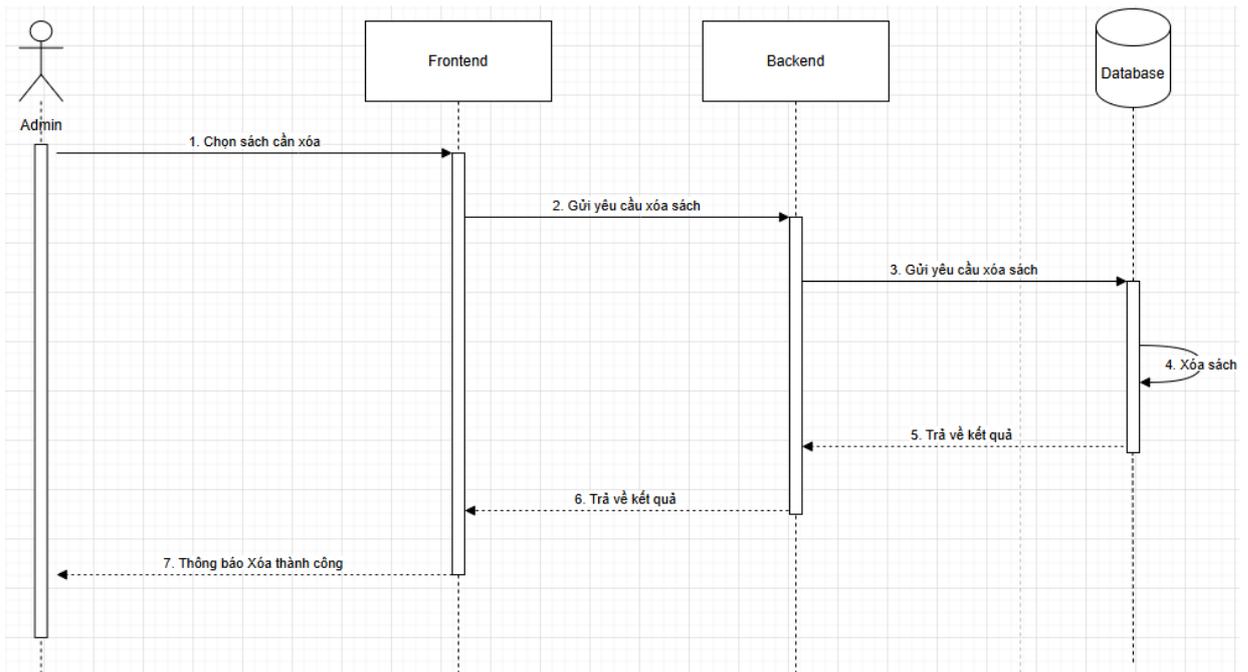
Hình 3.11: Biểu đồ tuần tự chức năng Thêm mới sách

Chức năng chỉnh sửa sách



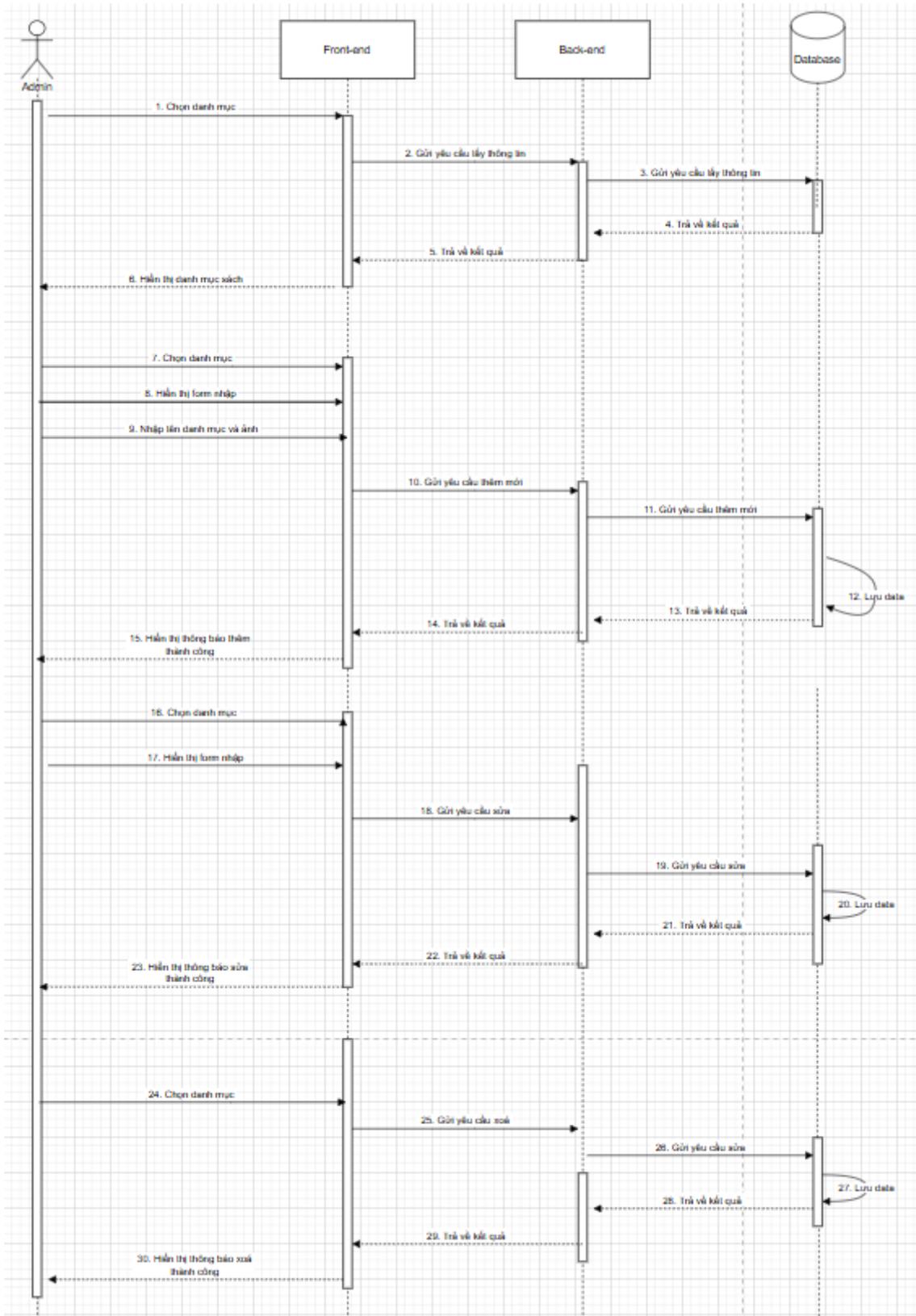
Hình 3.12: Biểu đồ tuần tự chức năng Chỉnh sửa sách

Chức năng Xóa sách



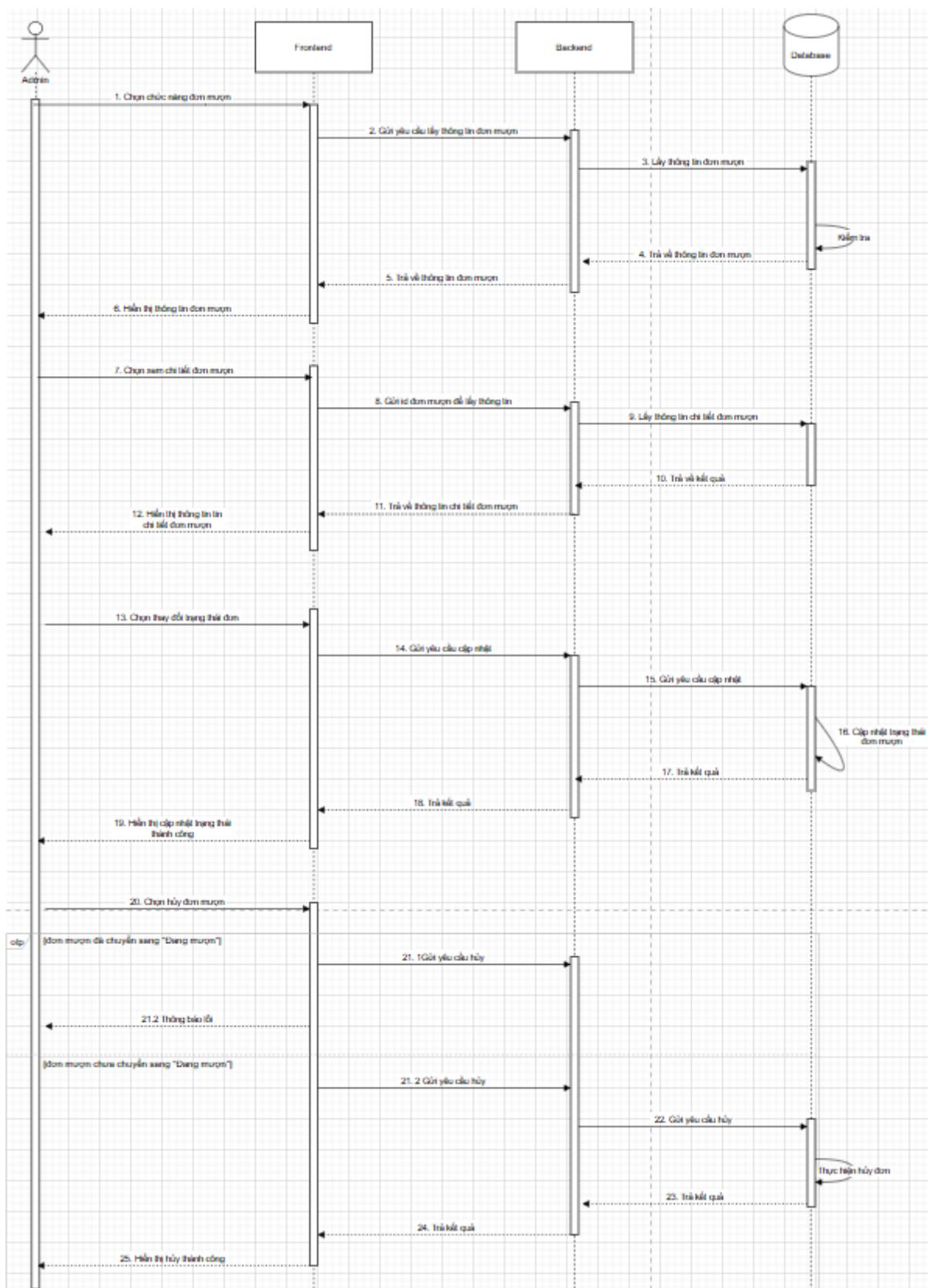
Hình 3.13: Biểu đồ tuần tự chức năng Xóa sách

3.1.12. Biểu đồ tuần tự chức năng Quản lý danh mục



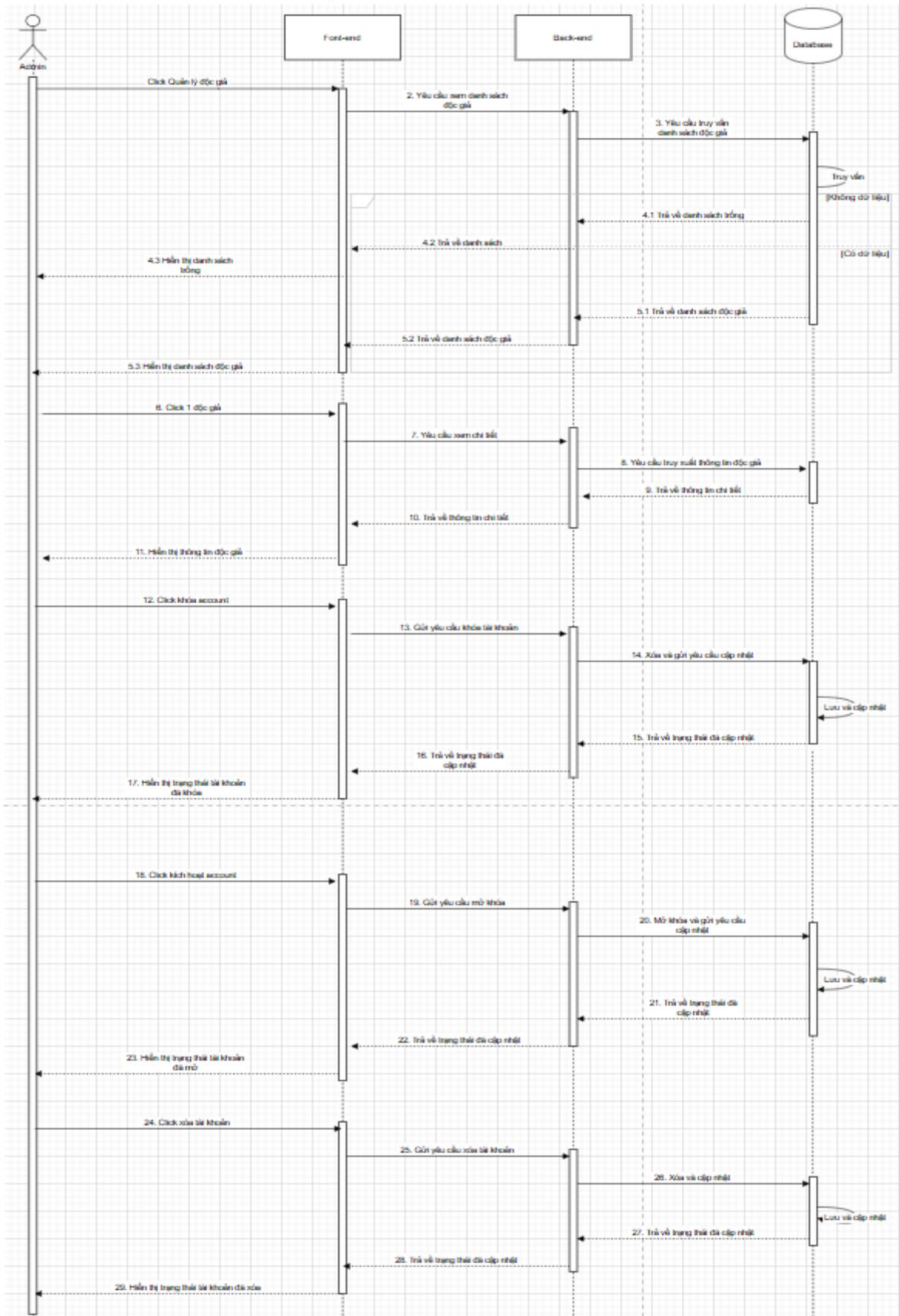
Hình 3.14: Biểu đồ tuần tự chức năng Quản lý danh mục

3.1.13. Biểu đồ tuần tự chức năng Quản lý đơn mượn



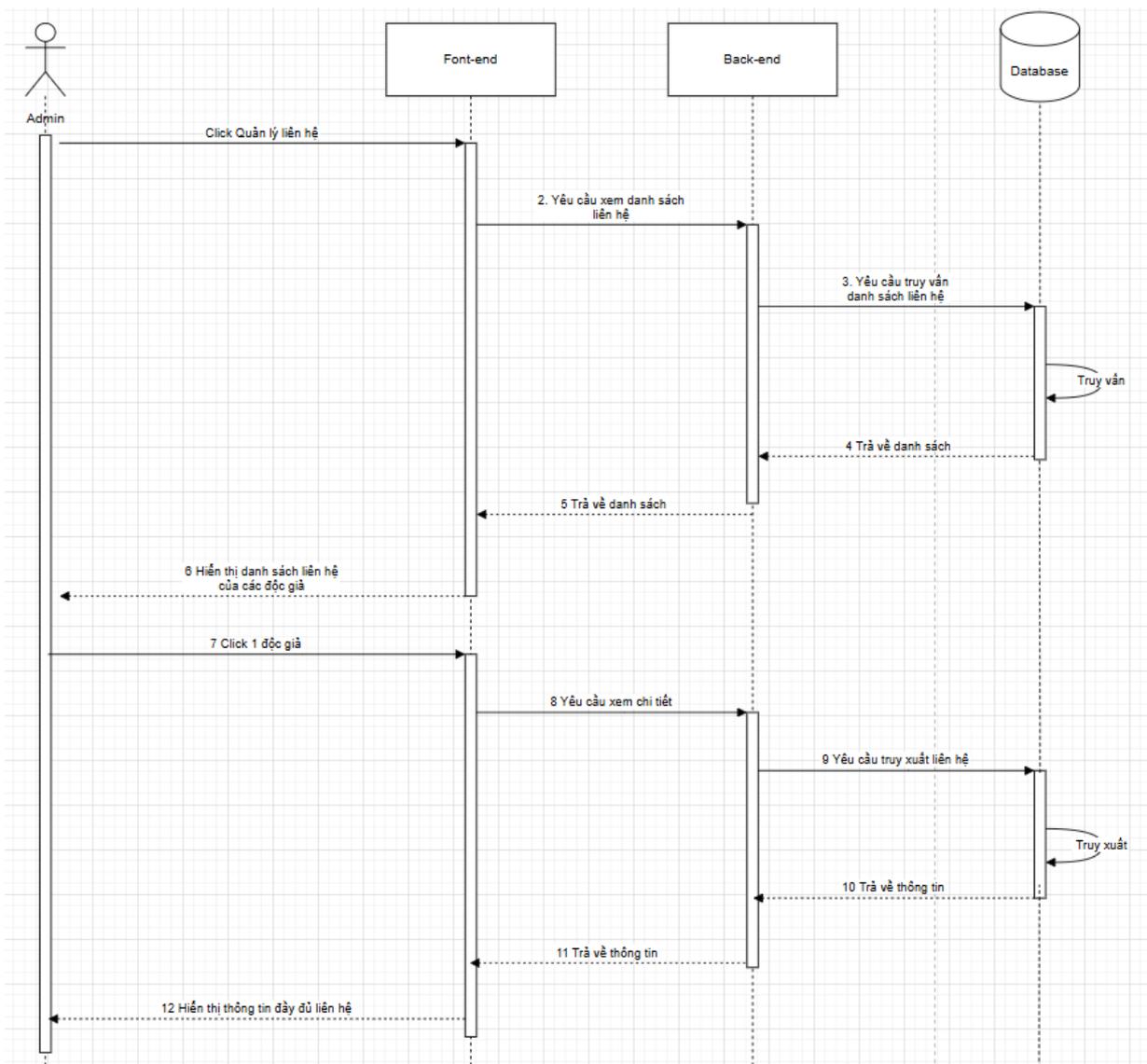
Hình 3.15: Biểu đồ tuần tự chức năng Quản lý đơn mượn

3.1.14. Biểu đồ tuần tự chức năng Quản lý độc giả



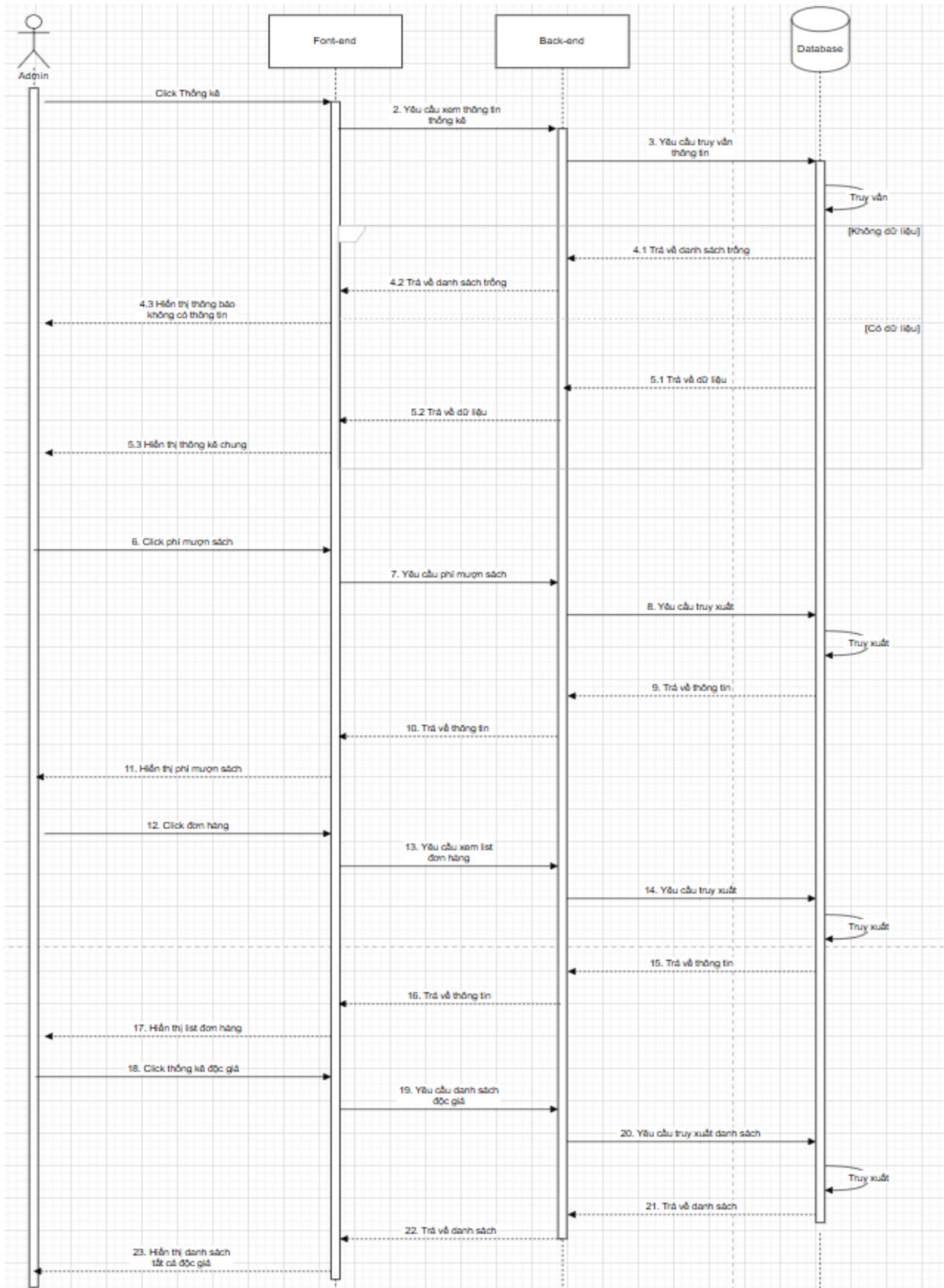
Hình 3.16: Biểu đồ tuần tự chức năng Quản lý độc giả

3.1.15. Biểu đồ tuần tự chức năng Quản lý thông tin liên hệ



Hình 3.17: Biểu đồ tuần tự chức năng Quản lý thông tin liên hệ

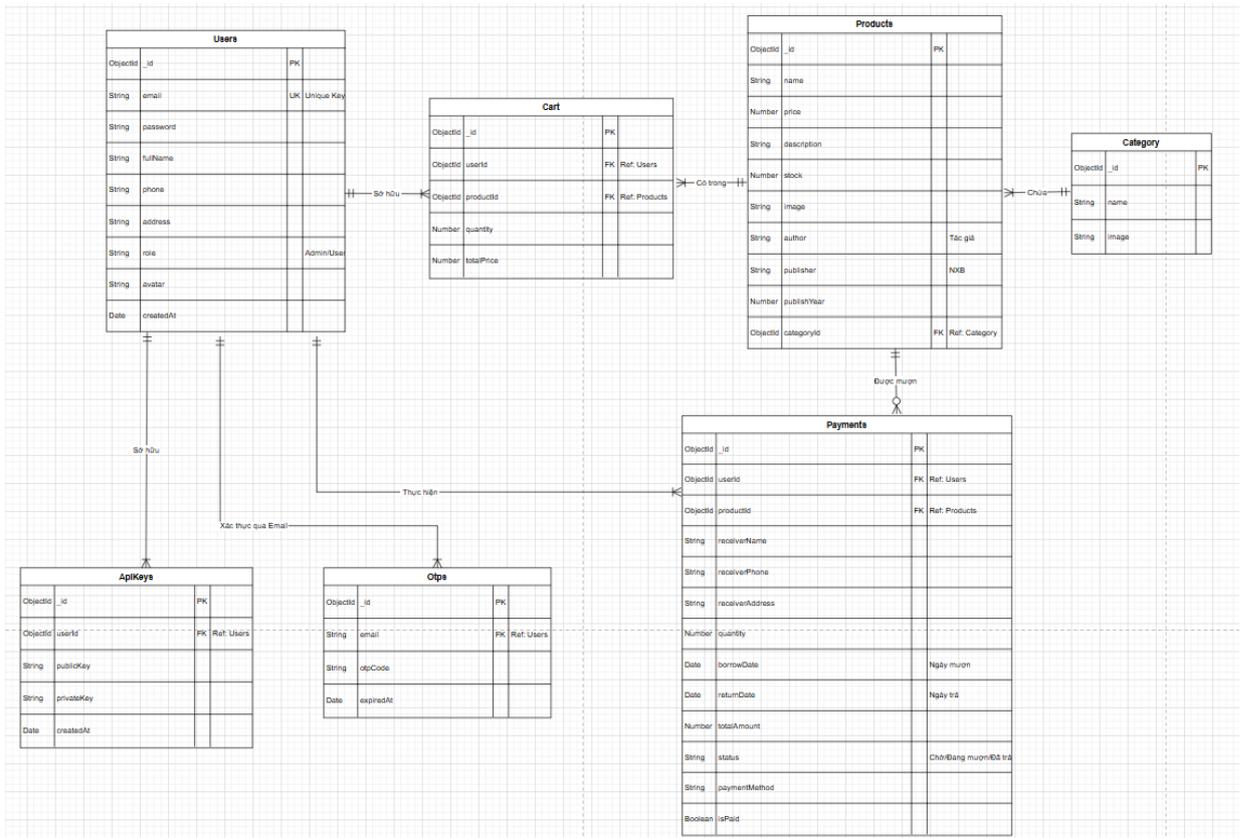
3.1.16. Biểu đồ tuần tự chức năng Thống kê



Hình 3.18: Biểu đồ tuần tự chức năng Thống kê

3.2. Thiết kế cơ sở dữ liệu

3.2.1. Lược đồ cơ sở dữ liệu



Hình 3.19: Lược đồ cơ sở dữ liệu

3.2.2. Danh sách các bảng

STT	Tên bảng	Mô tả
1	Users	Lưu thông tin và tài khoản của User.
2	Products	Lưu thông tin các sách.
3	Category	Lưu thông tin các danh mục sách.
4	Payments	Lưu thông tin đơn mượn.
5	Apikey	Lưu thông tin khoá bí mật login Độc giả
6	Carts	Lưu thông tin giỏ sách.
7	Otp	Lưu thông tin OTP xác thực tài khoản

Bảng 3.1. Danh sách các bảng dữ liệu

3.2.3. Chi tiết các bảng

Bảng Users

STT	Tên trường	Kiểu dữ liệu	Ghi chú
1	Id	UUID	Mã User
2	isAdmin	STRING	Quyền User
3	fullName	STRING	Tên User
4	phone	STRING	Số điện thoại
5	address	STRING	Địa chỉ
6	password	STRING	Mật khẩu tài khoản
7	email	STRING	Email.
8	typeLogin	ENUM	Loại tài khoản người dung (Email, Google)
9	Avatar	STRING	Ảnh đại diện
10	sex	ENUM	Giới tính

Bảng 3.2. Bảng User

Bảng Products

STT	Tên trường	Kiểu dữ liệu	Ghi chú
1	id	UUID	Mã sách
2	nameProduct	STRING	Tên sách
3	priceProduct	INTEGER	Phí mượn sách
4	description	TEXT	Mô tả

5	typeCode	STRING	Mã loại sách
6	categoryID	STRING	Mã danh mục
7	Stock	INTEGER	Số lượng sách đang có trong kho
8	stakePrice	INTEGER	Phí mượn sách hiện tại
9	images	TEXT	Ảnh sách
10	sizes	JSON	Kích cỡ

Bảng 3.3. Bảng Products

Bảng apiKey

STT	Tên trường	Kiểu dữ liệu	Ghi chú
1	id	Objectid	Mã Id comment
2	userId	STRING	Id User
3	publicKey	TEXT	Chuỗi công khai
4	privateKey	TEXT	Chuỗi riêng tư

Bảng 3.4. Bảng ApiKey

Bảng Category

STT	Tên trường	Kiểu dữ liệu	Ghi chú
1	id	UUID	Mã danh mục
2	nameCategory	STRING	Tên danh mục

Bảng 3.5. Bảng Category

Bảng Cart

STT	Tên trường	Kiểu dữ liệu	Ghi chú
1	id	UUID	Mã giỏ sách
2	userId	STRING	Id User
3	productId	STRING	Id sách
4	Quantity	INTEGER	Số lượng sách đặt mượn
5	fullName	STRING	Tên người mượn sách
6	Phone	STRING	Số điện thoại người mượn sách
7	Address	STRING	Địa chỉ người mượn
8	endDate	DATE	Thời gian kết thúc
9	size	STRING	Kích cỡ
10	startDate	DATE	Thời gian bắt đầu

Bảng 3.6. Bảng Cart

Bảng otp

STT	Tên trường	Kiểu dữ liệu	Ghi chú
1	email	STRING	Email
2	otp	STRING	Mã xác thực

Bảng 3.7. Bảng Otp

Bảng payments

STT	Tên trường	Kiểu dữ liệu	Ghi chú
1	id	UUID	Mã giỏ sách
2	paymentId	STRING	Id Thanh toán phí mượn
3	userId	STRING	Id User
4	productId	STRING	Id sách
5	quantity	INTEGER	Số lượng sách đặt mượn
6	size	STRING	Kích cỡ
7	startDate	DATE	Thời gian bắt đầu
8	endDate	DATE	Thời gian kết thúc
9	fullName	STRING	Tên độc giả
10	phone	STRING	Số điện thoại người mượn
11	address	STRING	Địa chỉ mượn
12	typePayment	STRING	Thanh toán phí mượn
13	pricePayment	STRING	Tổng Thanh toán phí mượn
14	status	STRING	Trạng thái đơn mượn

Bảng 3.8. Bảng Payments

CHƯƠNG 4. KIỂM THỬ VÀ TRIỂN KHAI WEBSITE

4.1. Kiểm thử

4.1.1. Phân tích các trường hợp kiểm thử

STT	Chức năng	Mục tiêu
1	Đăng nhập	<ul style="list-style-type: none">– Kiểm tra giao diện chung màn hình, hiển thị đầy đủ và chính xác các button, textbox tương ứng.– Kiểm tra tính hợp lệ của các Phí mượn sách trị đầu vào và hiển thị các thông báo lỗi trên form Đăng nhập theo đúng đặc tả usecase:<ul style="list-style-type: none">+ Số điện thoại.+ Mật khẩu.– Kiểm tra chức năng đăng nhập vào Website.
2	Đăng ký	<ul style="list-style-type: none">– Kiểm tra giao diện chung màn hình, hiển thị đầy đủ và chính xác các button, textbox tương ứng.– Kiểm tra tính hợp lệ của các Phí mượn sách trị đầu vào và hiển thị các thông báo lỗi của các trường trên form Đăng ký theo đúng đặc tả usecase:<ul style="list-style-type: none">+ Số điện thoại.+ Họ và tên.+ Địa chỉ của bạn.+ Email.+ Mật khẩu.+ Xác nhận mật khẩu.– Kiểm tra chức năng đăng ký vào Website.
3	Cập nhật tài khoản	<ul style="list-style-type: none">– Kiểm tra giao diện chung màn hình, hiển thị đầy đủ thông tin và chính xác các button, textbox tương ứng.

		<ul style="list-style-type: none"> - Kiểm tra tính hợp lệ của các Phí mượn sách trị đầu vào và hiển thị các thông báo lỗi của các trường trên màn hình cập tài khoản: <ul style="list-style-type: none"> + Họ và tên. + Địa chỉ của bạn. + Ngày sinh. + Giới tính. - Kiểm tra thao tác hiển thị, thay đổi và lưu thông tin tài khoản. - Kiểm tra chức năng thay đổi mật khẩu, thay ảnh đại diện.
4	Tìm kiếm	<ul style="list-style-type: none"> - Kiểm tra giao diện chung màn hình, hiển thị đầy đủ và chính xác các button, textbox tương ứng. - Kiểm tra chức năng tìm kiếm và hiển thị thông tin sách.
5	Giỏ sách	<ul style="list-style-type: none"> - Kiểm tra giao diện chung màn hình, hiển thị đầy đủ và chính xác các button, textbox tương ứng. - Kiểm tra chức năng thêm, sửa, xóa và hiển thị thông tin các sách trong giỏ sách.
6	Chi tiết sách	<ul style="list-style-type: none"> - Kiểm tra giao diện chung màn hình, hiển thị đầy đủ và chính xác các button, textbox tương ứng. - Kiểm tra các chức năng: <ul style="list-style-type: none"> + Thay đổi số lượng sách. + Thêm sách vào giỏ sách. + Mượn ngay.
7	Quản lý sách	<ul style="list-style-type: none"> - Kiểm tra giao diện chung màn hình, hiển thị đầy đủ và chính xác các button, textbox tương ứng. - Kiểm tra chức năng thêm, sửa, xóa và hiển thị thông tin các sách.

8	Quản lý danh mục	<ul style="list-style-type: none"> – Kiểm tra giao diện chung màn hình hiển thị đầy đủ và chính xác các button, textbox, tương ứng. – Kiểm tra chức năng thêm thêm sửa, xóa và hiển thị thông tin các danh mục hiện có trên website.
9	Quản lý đơn mượn	<ul style="list-style-type: none"> – Kiểm tra giao diện chung màn hình hiển thị đầy đủ và chính xác các button, textbox, tương ứng. – Kiểm tra chức năng thay đổi trạng thái đơn mượn và hiển thị thông tin các sách trong giỏ sách.

Bảng 4.1. Phân tích các trường hợp kiểm thử

4.1.2. Kết quả thực thi Test

STT	Chức năng	Kết quả	
		Pass	Fail
1	Đăng nhập	14	0
2	Đăng ký	38	4
3	Cập nhật tài khoản	30	4
4	Tìm kiếm	17	1
5	Giỏ sách	22	1
6	Quản lý sách	46	1
7	Chi tiết sách	12	0
8	Quản lý danh mục	21	0
9	Quản lý đơn mượn	11	0
Tổng		228	11

Bảng 4.2. Bảng kết quả thực thi Test

4.2. Yêu cầu về cài đặt

Yêu cầu phần mềm:

- Hệ điều hành: Hệ điều hành chạy window 10 trở lên.

Visual Studio Code (VSCode) là một **trình soạn thảo mã nguồn đa nền tảng**, được phát triển bởi **Microsoft**. Nhờ vào hệ sinh thái tiện ích mở rộng (extensions) phong phú, VSCode có thể hoạt động như một **IDE nhẹ** hỗ trợ nhiều ngôn ngữ lập trình như **JavaScript, Python, C++, Java, PHP, Go** và nhiều ngôn ngữ khác.

VSCode cung cấp các tính năng nổi bật như:

- Tự động hoàn thiện mã (IntelliSense)
- Gỡ lỗi trực tiếp trong trình soạn thảo (debugging)
- Tích hợp Git để quản lý mã nguồn
- Terminal tích hợp
- Hỗ trợ lập trình từ xa (Remote Development)
- Hàng nghìn tiện ích mở rộng để hỗ trợ framework, ngôn ngữ và công cụ phát triển

Với giao diện thân thiện, tốc độ xử lý nhanh và khả năng tùy biến cao, VSCode trở thành công cụ phổ biến trong cộng đồng lập trình viên, đặc biệt là với các dự án web, app và microservice.

4.2.1. Yêu cầu phần cứng:

- Màn hình: Để hiển thị giao diện làm việc
- Chuột và bàn phím: Để thao tác lập trình (đối với máy bàn)
- Máy tính gồm: bộ xử lý (CPU), RAM tối thiểu 4GB (khuyến nghị 8GB trở lên), ổ cứng SSD hoặc HDD, mainboard, v.v.

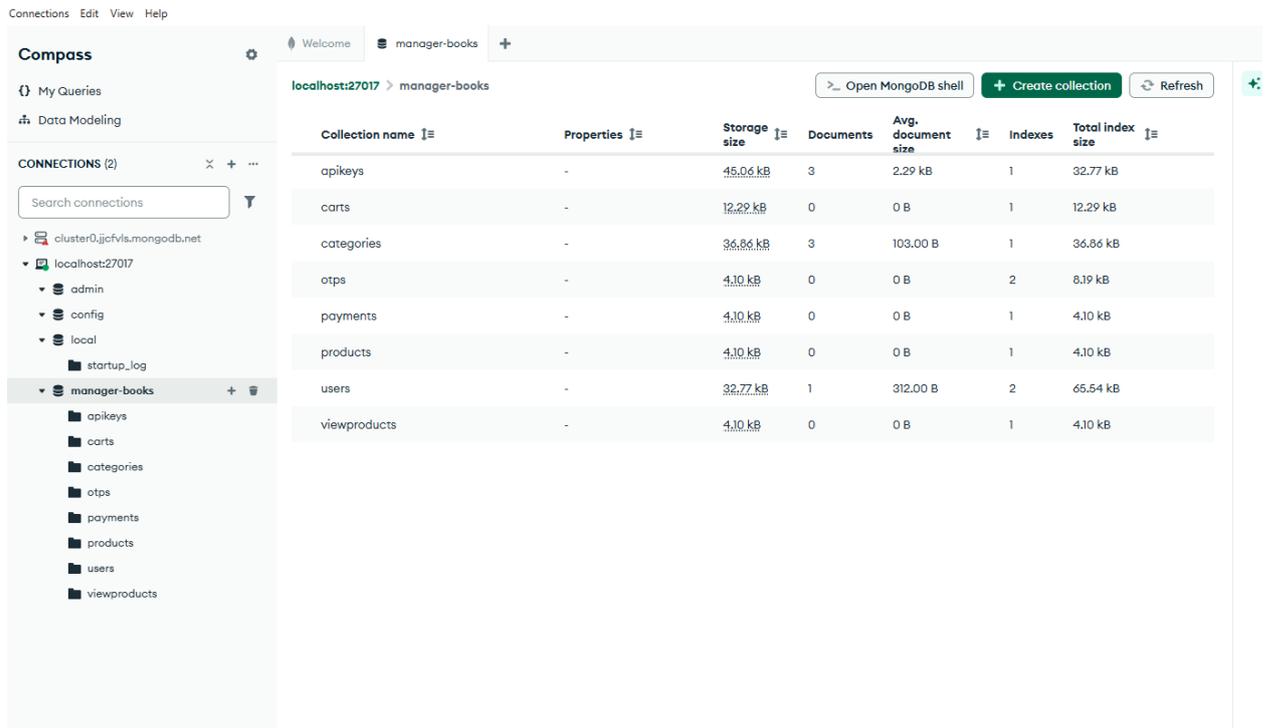
4.2.2. Yêu cầu mạng Internet:

Thiết bị cần kết nối mạng để:

- Tải và cập nhật tiện ích mở rộng (extensions)
- Cập nhật phần mềm và tính năng mới
- Đồng bộ hóa cài đặt nếu dùng tài khoản Microsoft
- Làm việc với các dịch vụ như GitHub, Live Share, Remote SSH, v.v.

4.3. Kết quả cài đặt môi trường

4.3.1. Data Access Layer (MongoDB)



Collection name	Properties	Storage size	Documents	Avg. document size	Indexes	Total index size
apikeyes	-	45.06 kB	3	2.29 kB	1	32.77 kB
carts	-	12.29 kB	0	0 B	1	12.29 kB
categories	-	36.86 kB	3	103.00 B	1	36.86 kB
otps	-	4.10 kB	0	0 B	2	8.19 kB
payments	-	4.10 kB	0	0 B	1	4.10 kB
products	-	4.10 kB	0	0 B	1	4.10 kB
users	-	32.77 kB	1	312.00 B	2	65.54 kB
viewproducts	-	4.10 kB	0	0 B	1	4.10 kB

Hình 4.1: Cài đặt cơ sở dữ liệu

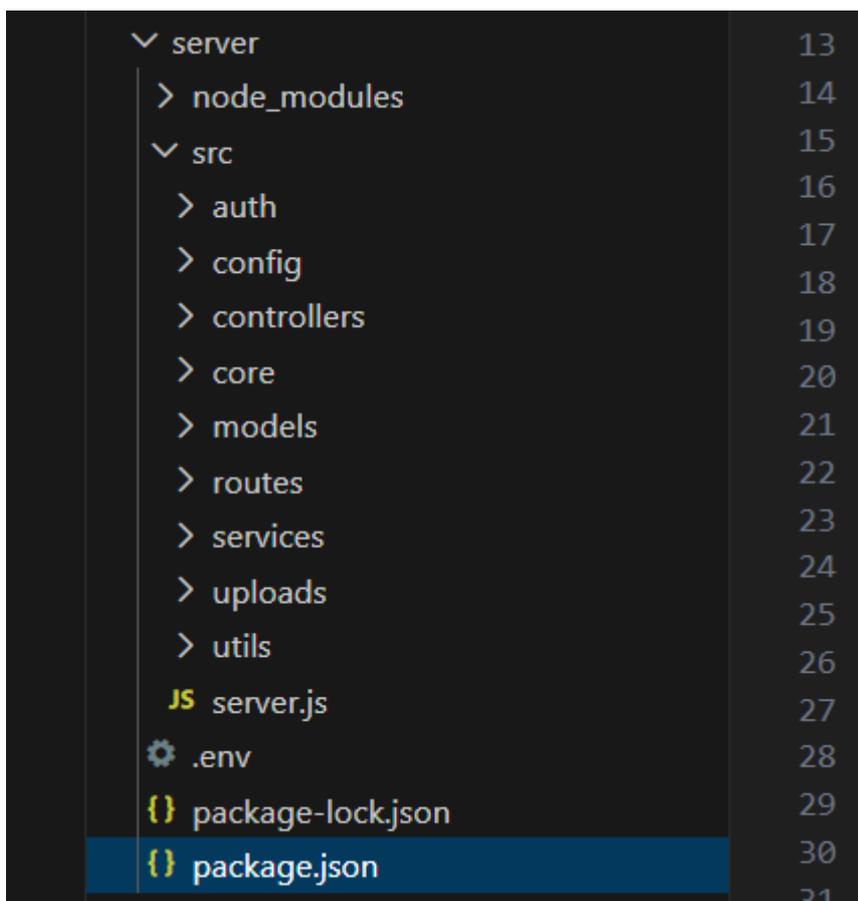
Hệ thống sử dụng MongoDB làm hệ quản trị cơ sở dữ liệu chính. Dữ liệu được lưu trữ dưới dạng các Document (tài liệu) JSON và được tổ chức thành các Collection (Bộ sưu tập). Việc tương tác với cơ sở dữ liệu được thực hiện thông qua thư viện Mongoose trong Node.js, giúp định nghĩa Schema chặt chẽ và đảm bảo tính toàn vẹn dữ liệu.

Với MongoDB, User có thể dễ dàng:

- Thiết lập và triển khai cơ sở dữ liệu chỉ trong vài bước
- Tự động sao lưu và phục hồi dữ liệu
- Quản lý hiệu năng và bảo mật với các công cụ tích hợp sẵn
- Mở rộng Website theo nhu cầu mà không cần can thiệp sâu vào hạ tầng

Giải pháp này đặc biệt phù hợp cho những doanh nghiệp và lập trình viên muốn tập trung vào phát triển ứng dụng mà không phải lo lắng về các vấn đề kỹ thuật như bảo trì máy chủ, cập nhật bảo mật hay tối ưu hiệu suất Website.

4.3.2. Business Logic Layer (Express)



server	13
node_modules	14
src	15
auth	16
config	17
controllers	18
core	19
models	20
routes	21
services	22
uploads	23
utils	24
server.js	25
.env	26
package-lock.json	27
package.json	28

Hình 4.2: Cài đặt Business Logic Layer (Express)

Config: mục này chứa các config cho dự án như là kết nối với MySQL Workbean

Controller: mục này chứa các controller kết hợp với các service để thực hiện yêu cầu từ User và trả về response.

Core: mục này có ý nghĩa là định nghĩa các status code khi trả về cho User

Auth: xử lý các yêu cầu User, nhằm kiểm tra quyền của User

Model : Mục này chứa các Model định nghĩa các trường để tạo và lưu trữ dữ liệu của User trong Database

Routes: Định nghĩa các tuyến đường giữa Front-End và Back-end nhằm truyền tải dữ liệu

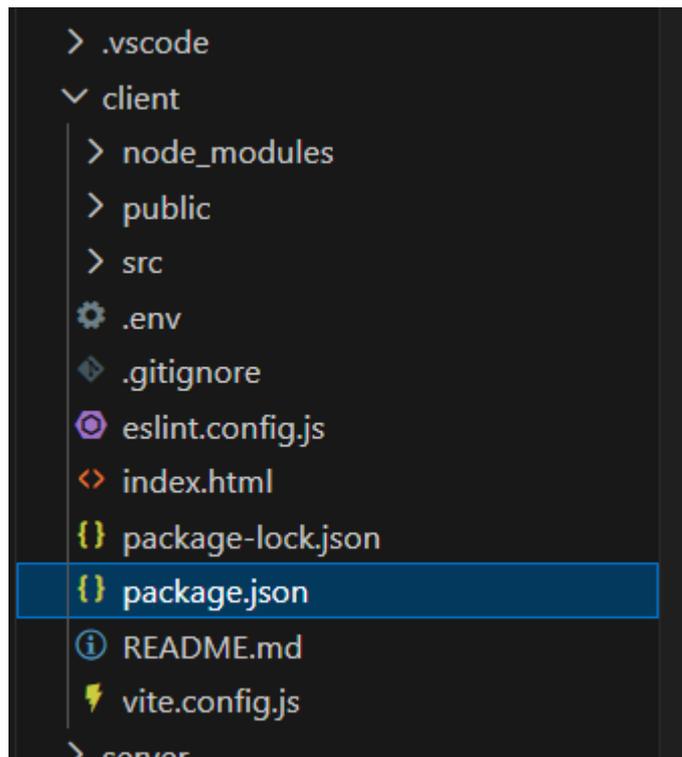
Services : Định nghĩa các logic nghiệp vụ được sử dụng ở nhiều nơi và có thể tái sử dụng lại tránh lặp lại logic quá nhiều

Uploads : chứa các file ảnh của sách

Utils : nhằm viết các chức năng mở rộng như gửi Email quên mật khẩu.

.env : lưu trữ các khoá bí mật cho Server

4.3.3. *Presentation Layer (ReactJS)*



Hình 4.3: Cài đặt *Presentation Layer (ReactJS)*

Assets : lưu trữ các ảnh tĩnh cho website UI/UX

Components : Định nghĩa các Components được tái sử dụng lại giữa các trang như Header, CardBody

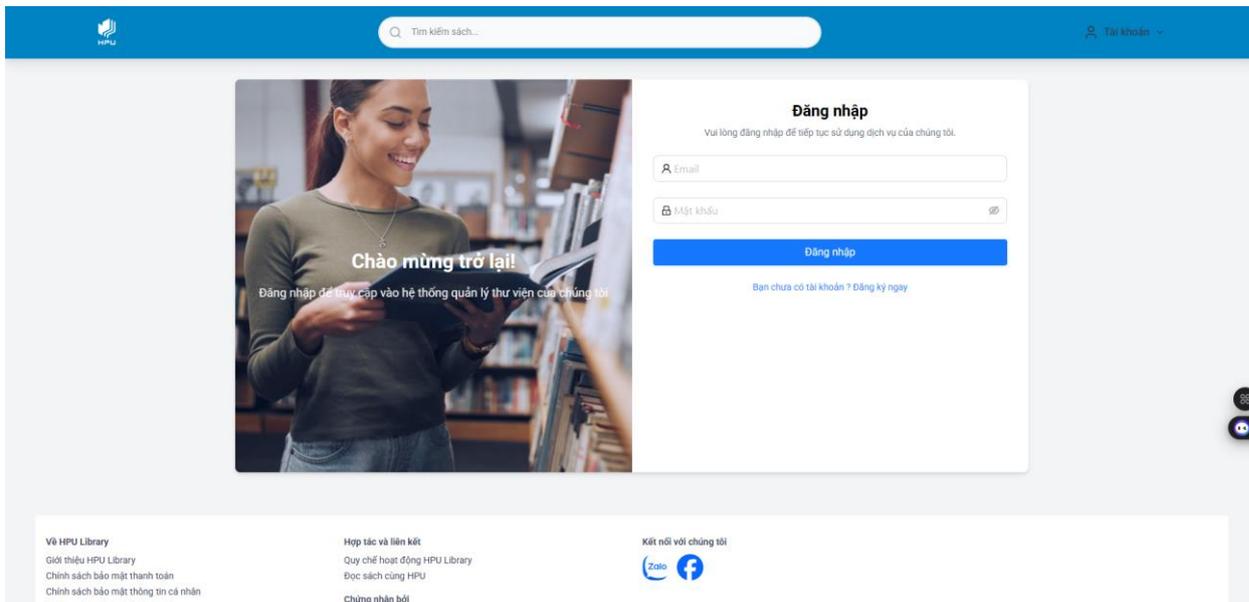
Config : là nơi để lưu trữ các router kết nối đến với Server Express

Hook : là một hàm JavaScript bắt đầu bằng use, cho phép tái sử dụng Logic có liên quan đến state hoặc side effects giữa nhiều các component, được kết hợp giữa các hooks sẵn có của ReactJS

4.4. Kết quả thực nghiệm

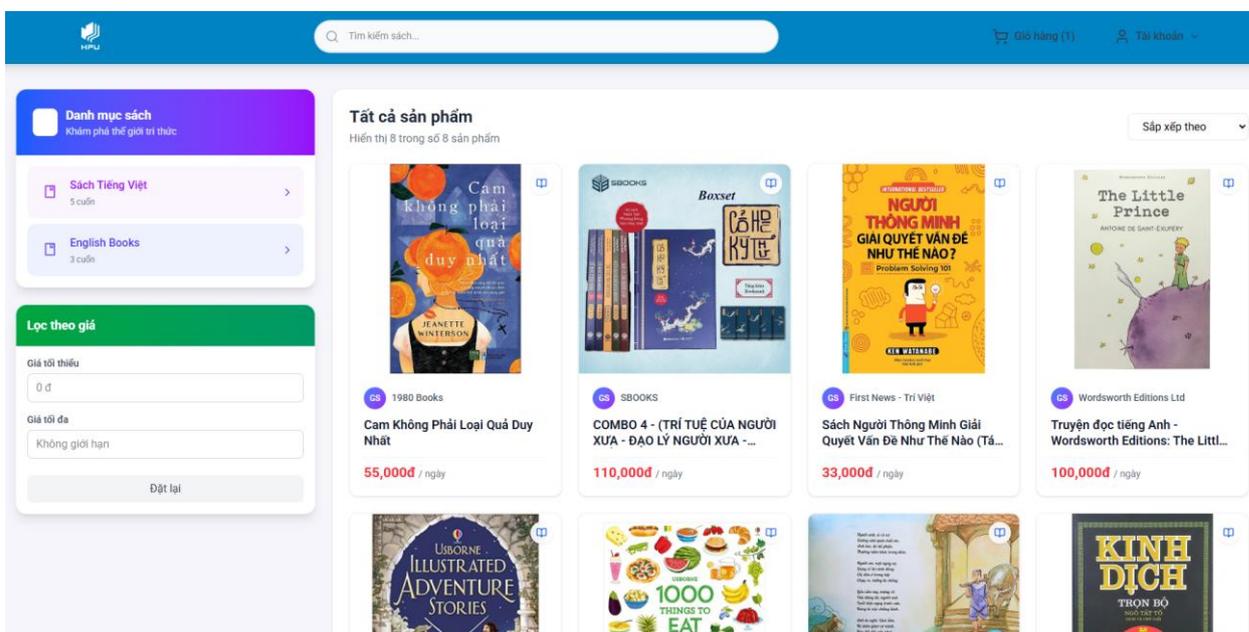
4.4.1. Giao diện đăng nhập Website

Khi User đăng nhập Website yêu cầu User nhập email và mật khẩu.



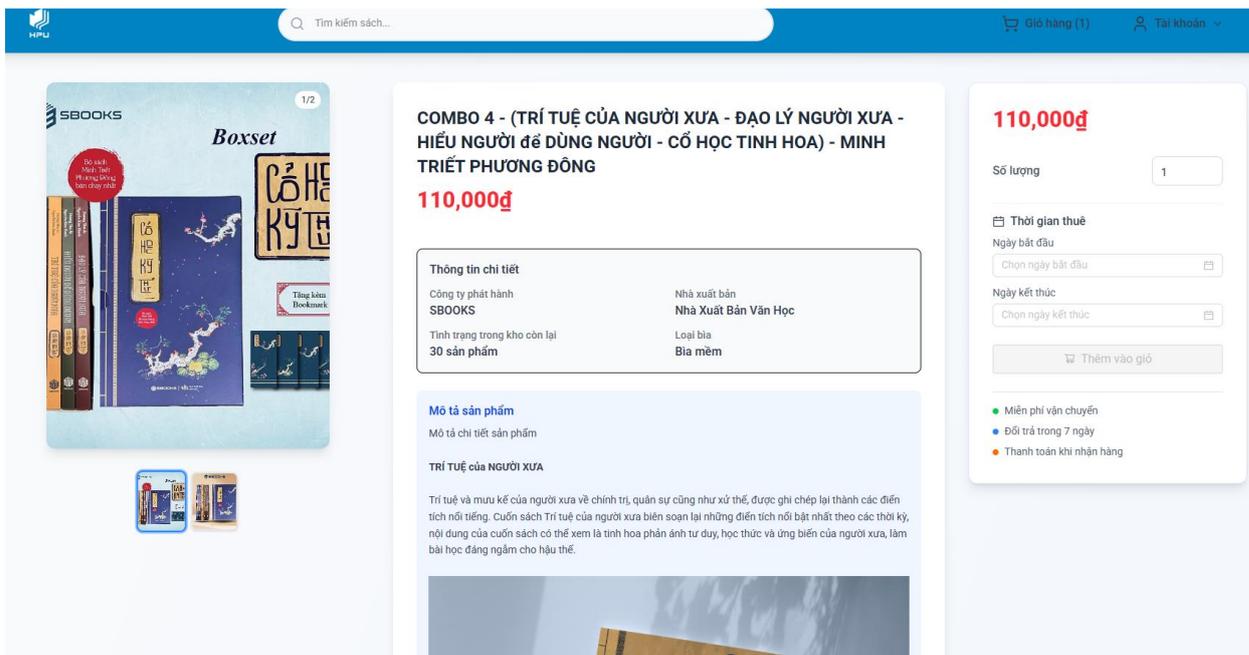
Hình 4.4: Giao diện Chức năng đăng nhập

4.4.2. Giao diện trang chủ Website



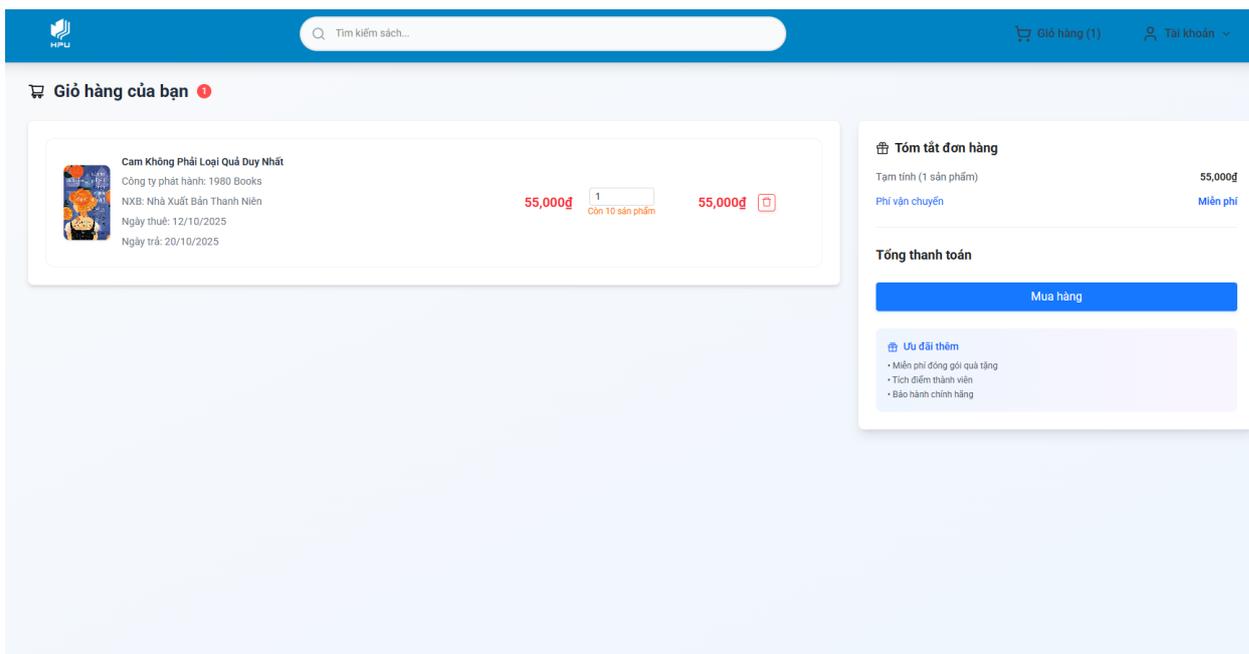
Hình 4.5: Giao diện Trang chủ màn hình khi đã đăng nhập

4.4.3. Giao diện chi tiết sách



Hình 4.6: Giao diện Chi tiết sách

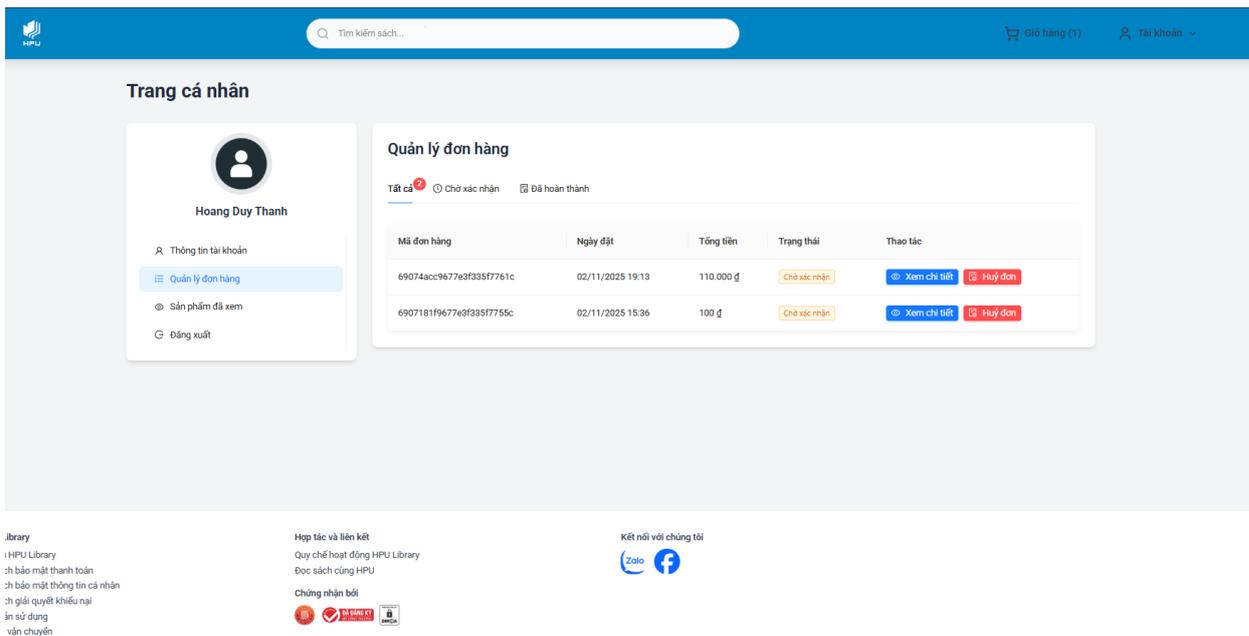
4.4.4. Giao diện giỏ sách



Hình 4.7: Giao diện Giỏ sách

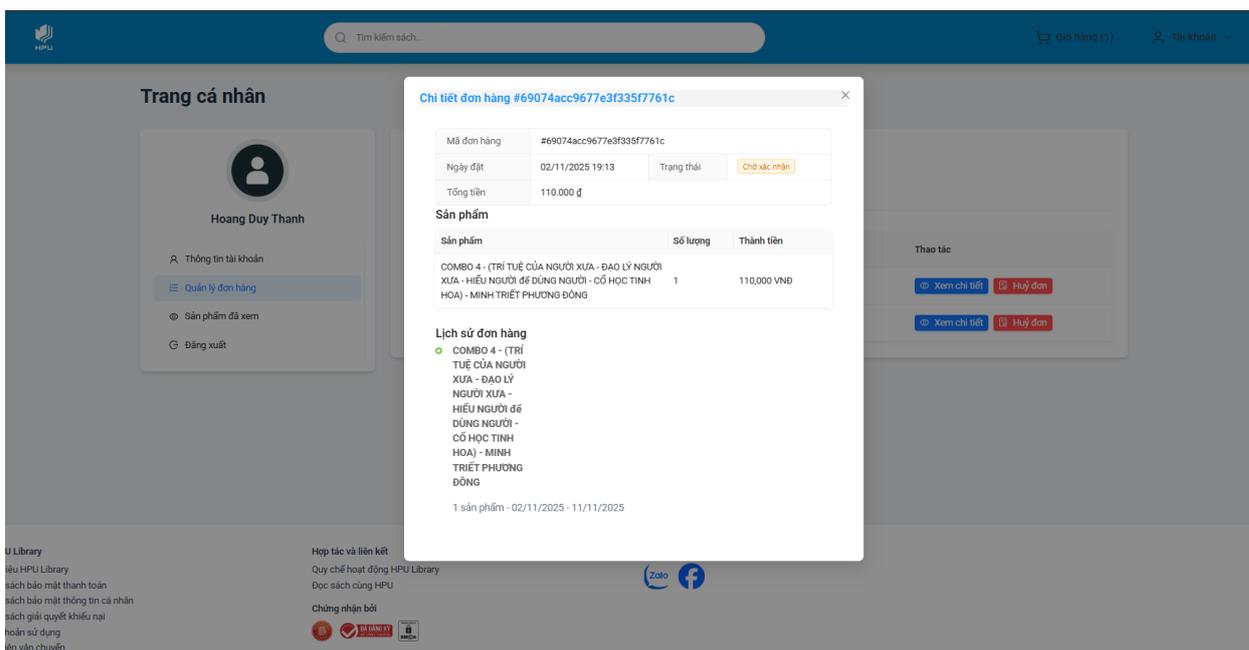
4.4.5. Giao diện chi tiết đơn mượn

Giao diện thông tin các đơn mượn đã đặt:



Hình 4.8: Giao diện Danh sách đơn mượn đã đặt gần đây

Giao diện chi tiết từng đơn mượn:

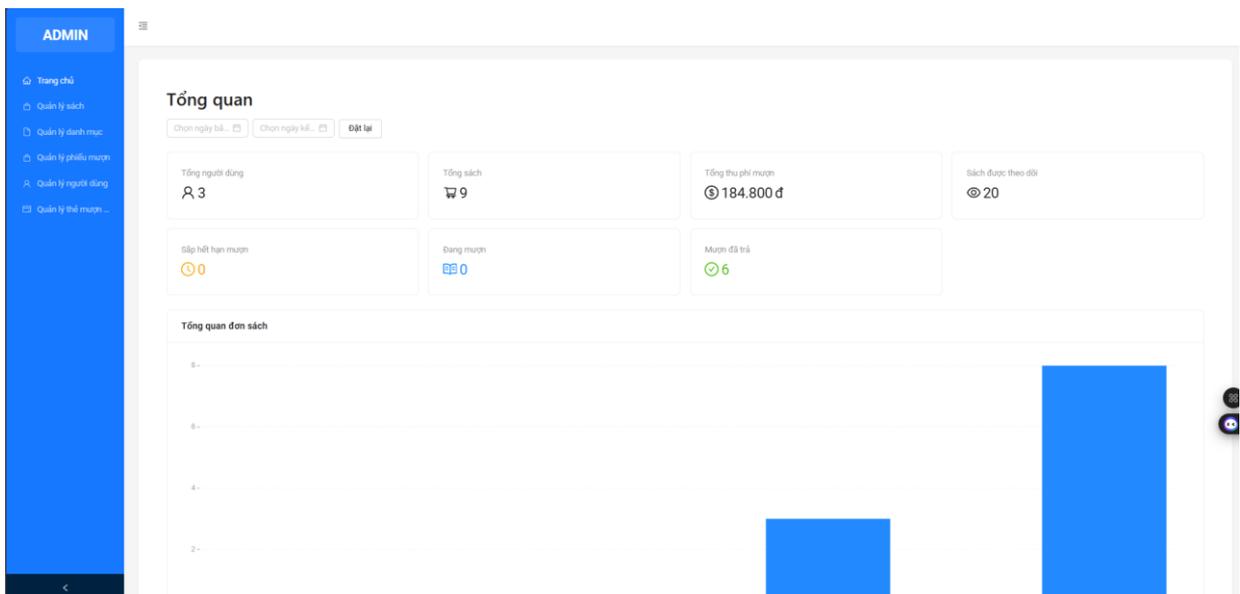


Hình 4.9: Giao diện Chi tiết từng đơn mượn

Giao diện cập nhật tài khoản

Hình 4.10: Giao diện Cập nhật tài khoản

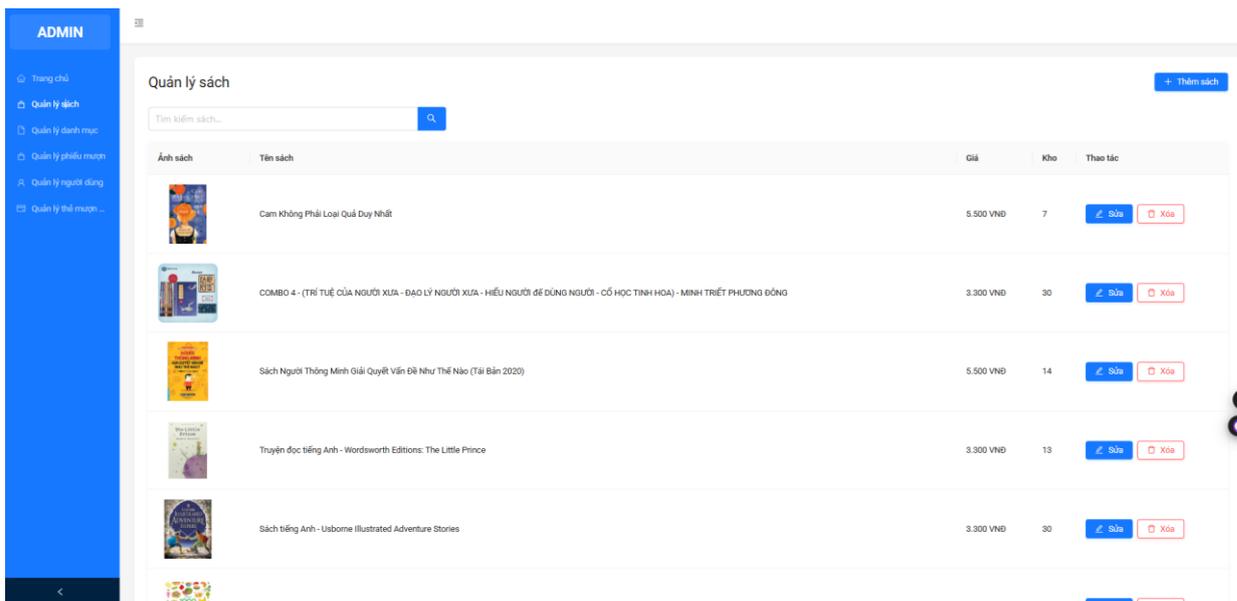
4.4.6. Giao diện thống kê



Hình 4.11: Giao diện Thống kê

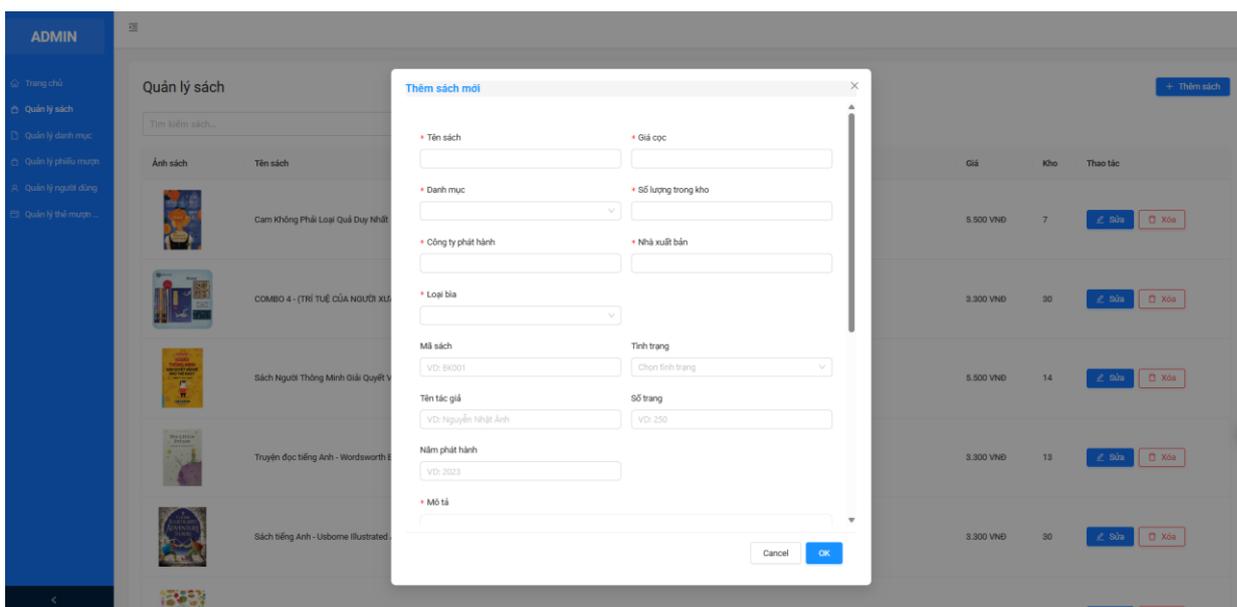
4.4.7. Giao diện quản lý sách

Giao diện quản lý sách:



Hình 4.12: Giao diện Quản lý sách

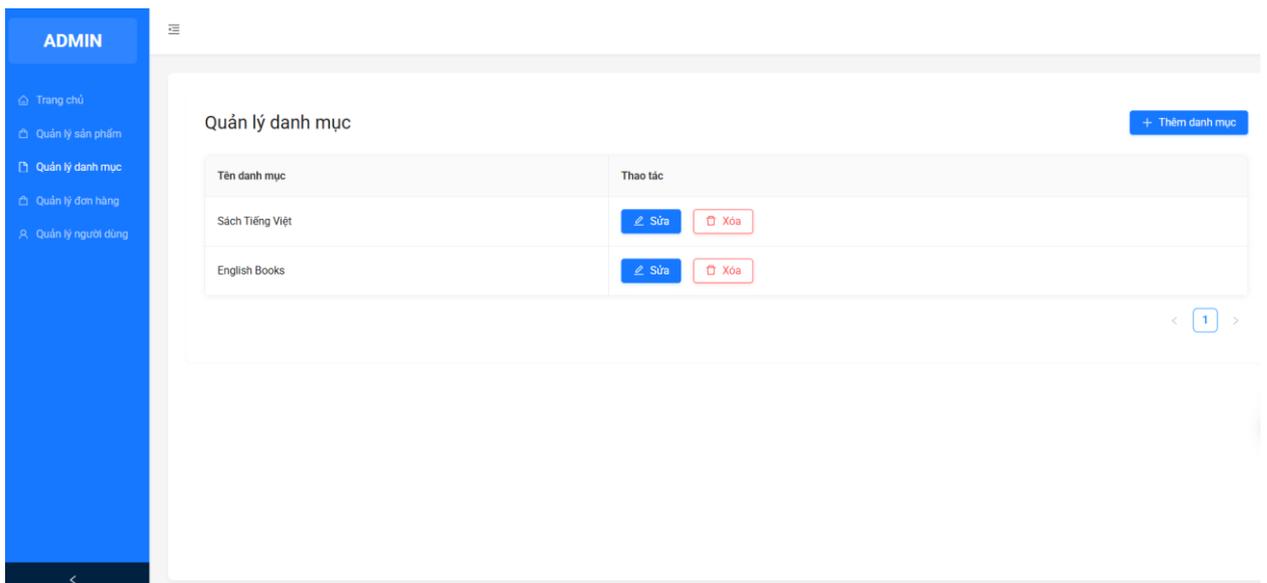
Giao diện thêm mới sách:



Hình 4.13: Giao diện Thêm mới sách

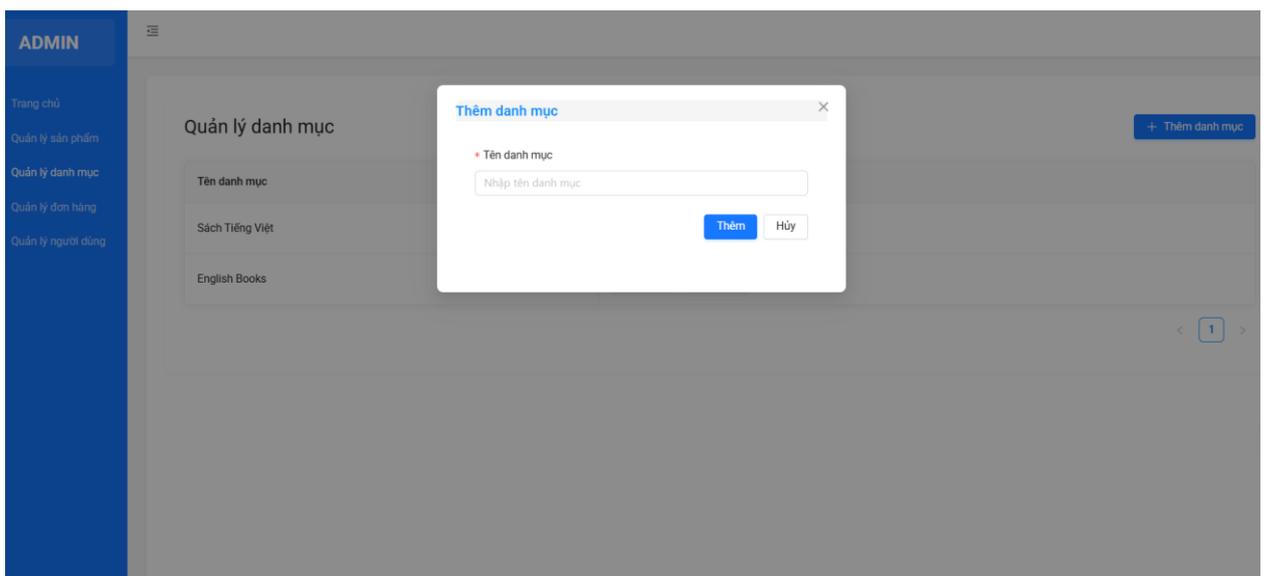
4.4.8. Giao diện quản lý danh mục

Giao diện quản lý danh mục:



Hình 4.14: Giao diện Danh sách danh mục

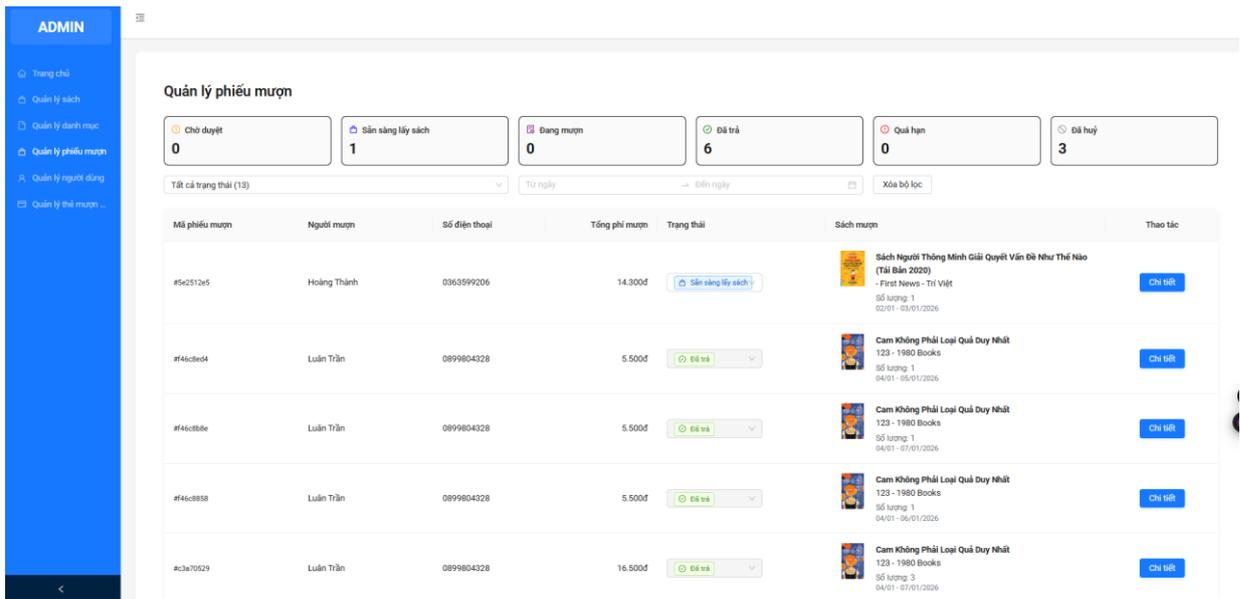
Giao diện thêm mới danh mục:



Hình 4.15: Giao diện Thêm mới danh mục

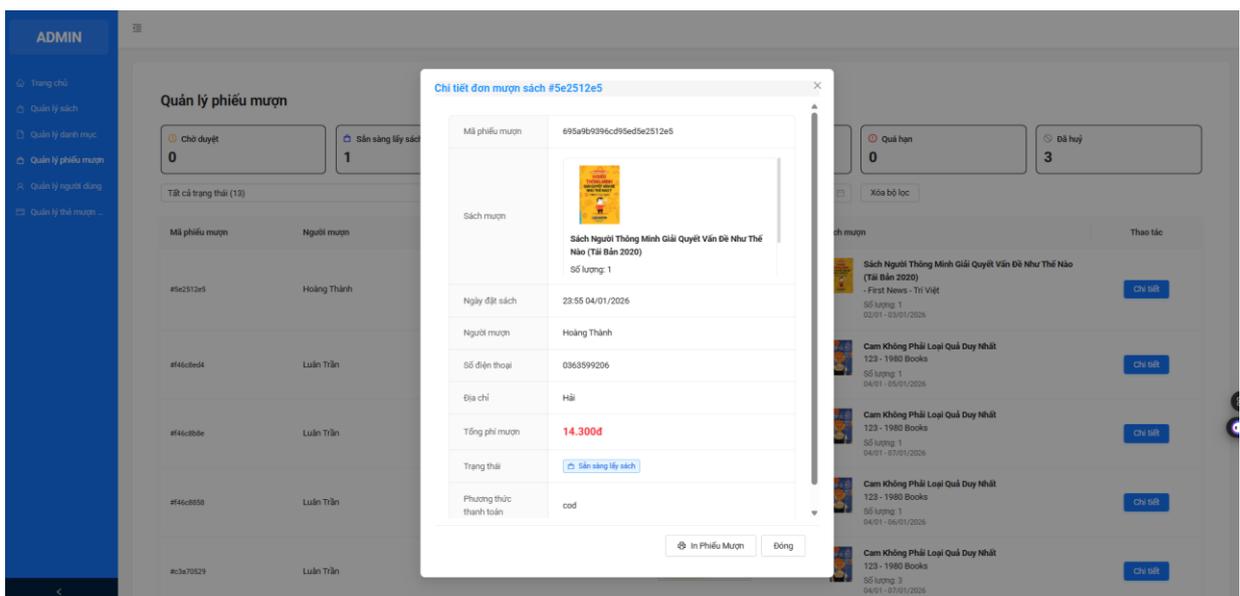
4.4.9. Giao diện quản lý đơn mượn

Giao diện quản lý danh sách đơn mượn:



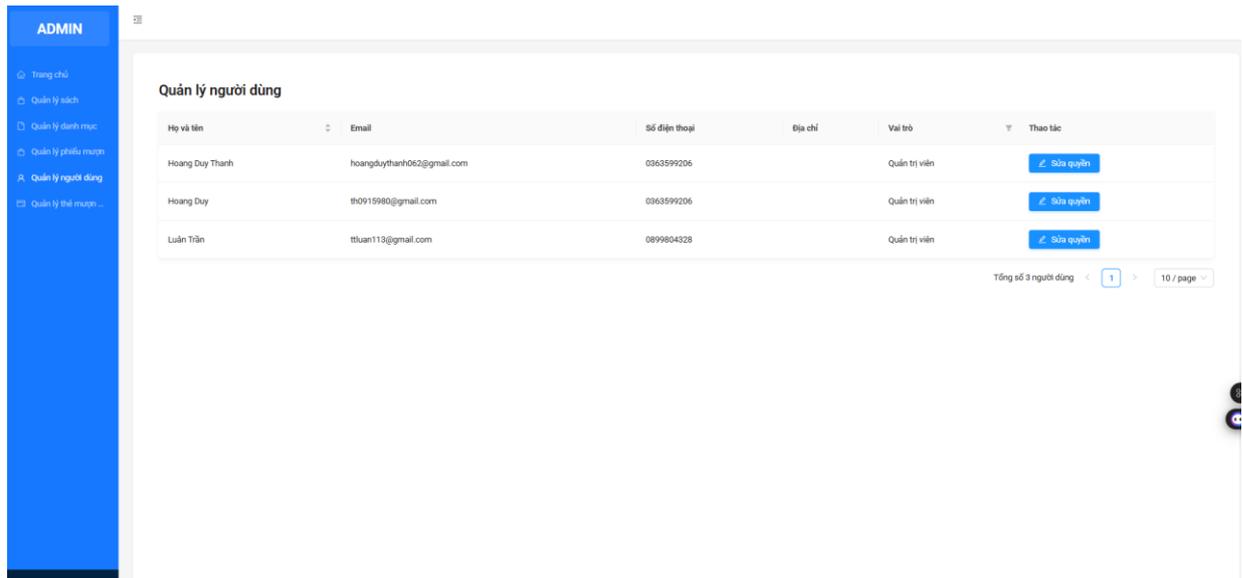
Hình 4.16: Giao diện Quản lý danh sách đơn mượn

Giao diện chi tiết đơn mượn:



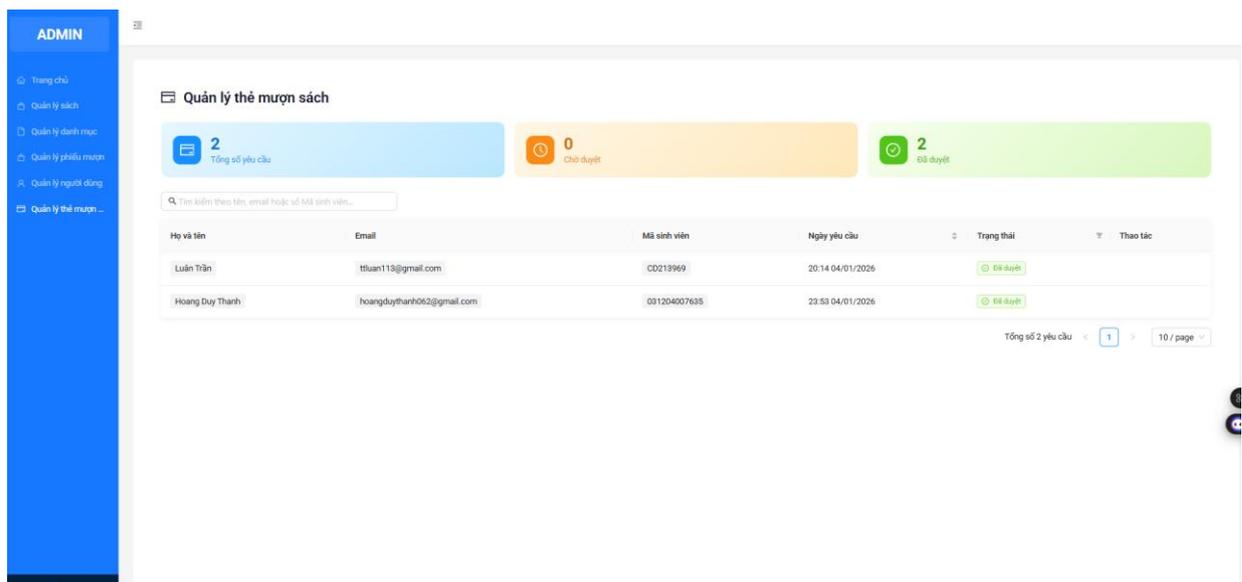
Hình 4.17. Giao diện cập nhật đơn mượn

4.4.10. Giao diện quản lý User



Hình 4.18: Giao diện Quản lý User

4.4.11. Giao diện quản lý thẻ mượn



Hình 4.19: Giao diện Quản lý thẻ mượn

CHƯƠNG 5. KẾT LUẬN

5.1. Kết quả đạt được

Trong quá trình triển khai và phát triển Website web thương mại điện tử, em đã đạt được những kết quả quan trọng, bao gồm:

- Hoàn thiện tài liệu phân tích và đặc tả các yêu cầu Website một cách chi tiết.
- Thiết kế và xây dựng thành công cơ sở dữ liệu cho Website, đảm bảo lưu trữ và quản lý dữ liệu hiệu quả.
- Tạo dựng bộ cục giao diện cho cả trang mượn hàng và trang quản trị, đáp ứng các yêu cầu cơ bản về chức năng.
- Xây dựng các tính năng tìm kiếm, lọc sách, hỗ trợ User tìm kiếm dễ dàng.
- Hoàn thành luồng đặt mượn từ bước đặt mượn, xác nhận đến theo dõi đơn mượn.
- Triển khai chức năng giỏ sách, giúp Độc giả quản lý sách chọn mượn.
- Tích hợp Thanh toán phí mượn trực tuyến qua cổng VNPAY, hỗ trợ Thanh toán phí mượn nhanh chóng và tiện lợi.
- Phát triển các chức năng quản trị như quản lý tài khoản, sách, danh mục, đơn mượn.
- Kiểm thử các chức năng cơ bản của Website.

5.2. Hạn chế

Bên cạnh những kết quả đạt được, Website vẫn còn một số hạn chế:

- Sách chưa có tính chuyên nghiệp cao, cần tiếp tục nâng cao để đáp ứng tiêu chuẩn thị trường.
- Giao diện chưa thực sự thẩm mỹ, cần cải tiến để tạo ấn tượng tốt hơn với User.
- Một số lỗi có thể vẫn xuất hiện trong quá trình sử dụng, cần kiểm thử và tối ưu hóa thêm để nâng cao độ ổn định của Website.

5.3. Hướng phát triển

Dựa trên kết quả đạt được và các hạn chế, Website có nhiều tiềm năng phát triển trong tương lai, bao gồm:

- Phát triển ứng dụng trên nền tảng di động (Android, iOS) để User dễ dàng sử dụng trên thiết bị di động, nâng cao trải nghiệm Độc giả.

- Tích hợp thêm nhiều phương thức Thanh toán phí mượn khác để tăng sự linh hoạt và tiện lợi cho Độc giả.
- Cải thiện công cụ gợi ý sách, giúp Độc giả dễ dàng tìm thấy sách phù hợp với nhu cầu.
- Thêm tính năng trò chuyện trực tiếp (live chat) để hỗ trợ Độc giả tốt hơn.
- Bổ sung chức năng đa ngôn ngữ, mở rộng khả năng tiếp cận Độc giả từ nhiều quốc gia.

Với nền tảng cơ bản đã xây dựng, em tin rằng dự án này có thể được tiếp tục phát triển để đáp ứng tốt hơn nhu cầu của doanh nghiệp và Độc giả, góp phần vào sự phát triển của lĩnh vực thương mại điện tử.

TÀI LIỆU THAM KHẢO

- [1] Mozilla Developer Network (MDN). *Client–Server overview*. https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Client-Server_overview
- [2] React Official Documentation – <https://react.dev>
- [3] Vite Official Documentation – <https://vitejs.dev>
- [4] React Router – Official Documentation: <https://reactrouter.com>
- [5] Nguyễn Thanh Hưng (2023). *Ứng dụng SCSS và CSS Modules trong phát triển giao diện React*, Trường Đại học Công nghệ Thông tin TP.HCM.
- [6] MDN Web Docs – *React Context API Overview*: <https://developer.mozilla.org/>
- [7] Node.js Official Website – <https://nodejs.org/en>
- [8] Express.js Official Website – <https://expressjs.com/>
- [9] Express.js Official Website – <https://expressjs.com/>
- [10] Phạm Quang Huy (2022). *Ứng dụng kiến trúc MVC trong phát triển web với Node.js và Express*, Trường Đại học FPT TP.HCM.
- [11] Trần Đức Tài (2023). *Sử dụng Middleware trong Express.js để bảo vệ Website API*, Đại học Công nghệ Thông tin - ĐHQG TP.HCM.
- [12] Nguyễn Văn Minh (2022). *Xây dựng và kết nối frontend-backend trong ứng dụng thương mại điện tử*, Trường Đại học Bách Khoa Hà Nội.
- [13] TopDev – MongoDB:
<https://topdev.vn/blog/mongodb-la-gi/>

