

ĐẠI HỌC QUỐC GIA TP.HCM
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



BÁO CÁO
LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC

XÂY DỰNG HỆ THỐNG THƯƠNG MẠI
ĐIỆN TỬ CHO THUÊ MUA BÁN SÁCH
CŨ GIÁ HỢP LÝ

Ngành: KHOA HỌC MÁY TÍNH

HỘI ĐỒNG: KHMT7CQ
GVHD: ThS MAI ĐỨC TRUNG
GVPB: ThS PHAN TRUNG HIẾU
---o0o---

SVTH 1: NGUYỄN LÊ XUÂN HÒA (2033145)

TP. HỒ CHÍ MINH, THÁNG 12/2024

LỜI CAM ĐOAN

Nhóm em nhóm thực hiện luận văn tốt nghiệp "Xây dựng hệ thống thương mại điện tử cho thuê mua bán sách cũ giá hợp lý" xin cam đoan những điều sau đây:

Nhóm em cam kết rằng toàn bộ nội dung và ý kiến trong luận văn tốt nghiệp là sự nghiên cứu và công trình của chính tác giả dưới sự hướng dẫn của thầy Mai Đức Trung.

Nhóm em khẳng định rằng mọi nguồn thông tin được sử dụng, bao gồm tài liệu, dữ liệu, và ý kiến của các tác giả khác, đã được chỉ định rõ ràng và đầy đủ theo quy định của luật pháp về bản quyền và sử dụng tài nguyên.

Nhóm em xác nhận rằng tất cả các phần trích dẫn, số liệu, hình ảnh, và các nguồn thông tin khác đều được trích dẫn đúng cách theo quy tắc tham chiếu được hướng dẫn và đã được liệt kê đầy đủ trong phần tài liệu tham khảo.

Nhóm em cam kết rằng chúng em đã tự mình thực hiện toàn bộ quá trình nghiên cứu mà không có sự sao chép từ bất kỳ nguồn nào khác mà không được ghi rõ.

Nhóm em xác nhận rằng luận văn tốt nghiệp này không có bản sao hoặc phiên bản khác tại bất kỳ trường đại học nào khác. Nếu xảy ra vi phạm, nhóm chúng em xin chịu toàn bộ trách nhiệm trước Ban giám hiệu nhà trường.

TP.HCM, Tháng 12/2024

Nguyễn Lê Xuân Hòa

LỜI CẢM ƠN

Nhóm em xin bày tỏ lòng biết ơn sâu sắc đến thầy Mai Đức Trung - người hướng dẫn tận tâm và có sự chi bảo tận tâm trong quá trình tác giả thực hiện luận văn tốt nghiệp " Xây dựng hệ thống thương mại điện tử cho thuê mua bán sách cũ giá hợp lý".

Nhóm em xin chân thành cảm ơn vì sự hỗ trợ và định hình ý kiến từ thầy Mai Đức Trung, người đã giúp chúng em xây dựng cơ sở kiến thức vững chắc, đồng thời cung cấp sự khích lệ và phản hồi tích cực từ thầy trong suốt quá trình nghiên cứu.

Cảm ơn thầy Mai Đức Trung đã dành thời gian và công sức của mình để hướng dẫn tác giả trong suốt quá trình nghiên cứu và thực hiện đồ án. Sự am hiểu, kiến thức sâu rộng, và tâm huyết của thầy đã tạo nên nguồn động viên lớn, giúp tác giả vượt qua những khó khăn và phát triển kỹ năng nghiên cứu của mình.

Nhóm em xin tỏ lòng biết ơn và sẵn lòng nhận định rằng, những kiến thức và kỹ năng đã học được từ thầy sẽ là nguồn động viên quý báu khi bước chân vào công việc và sự nghiệp tương lai.

Chân thành cảm ơn!

TÓM TẮT ĐỀ TÀI

Đề tài của tác giả là “Xây dựng hệ thống thương mại điện tử cho thuê mua bán sách cũ giá hợp lý”. Tại đây tác giả muốn xây dựng một hệ thống thương mại điện tử C2C nơi người dùng có thể thuê mướn tài liệu và cho thuê mướn tài liệu, quản lý đơn thuê và đơn cho thuê đồng thời quản lý việc đăng thuê của mình. Cũng như người dùng có thể mua bán tài liệu, quản lý đơn mua bán và đăng bài bán tài liệu. Để đảm bảo mục tiêu đó tác giả đã thực hiện được các chức năng trong hệ thống. Thực hiện chức năng đăng nhập, đăng xuất, đăng bài, quản lý bài đăng (thêm xoá bài đăng). Chức năng giỏ hàng (thêm xoá trong giỏ hàng), các chức năng đơn hàng thuê và mua bán: tạo đơn hàng, xem chi tiết đơn hàng. Chức năng quản lý đơn hàng: xem đơn hàng theo trạng thái, quản lý đơn hàng thay đổi trạng thái đơn hàng (xác nhận nhận hàng, xác nhận trả tiền, trả lại sách,...). Chức năng quản lý thông tin người dùng, sửa thông tin người dùng. Đặc biệt là chức năng quản lý voucher khuyến mãi theo người bán và voucher khuyến mãi toàn sàn. Ngoài ra hệ thống còn có giao diện để Quản trị viên hệ thống dễ dàng quản lý thông tin đơn hàng thuê, mua bán và quản lý voucher.

MỤC LỤC

LỜI CAM ĐOAN	1
LỜI CẢM ƠN	2
TÓM TẮT ĐỀ TÀI	3
MỤC LỤC	4
DANH MỤC HÌNH ẢNH	9
DANH MỤC BẢNG BIỂU	12
CHƯƠNG 1: TỔNG QUAN	14
1.1 Giới thiệu đề tài	14
1.2 Mục tiêu đề tài	14
1.3 Phạm vi đề tài	14
CHƯƠNG 2: CÁC HỆ THỐNG LIÊN QUAN	16
2.1 BooksRun	16
2.1.1 Giới thiệu	16
2.1.2 Ưu điểm	18
2.1.3 Nhược điểm	18
2.2 Chegg	19
2.2.1 Giới thiệu	19
2.2.2 Ưu điểm	20
2.2.3 Nhược điểm	20
2.3 So sánh với hệ thống của tôi	20
CHƯƠNG 3: CÔNG NGHỆ SỬ DỤNG	22
3.1 Công nghệ ở Front-end	22
3.1.1 ReactJS	22
3.1.2 TypeScript	24
3.1.3 Material UI	26
3.1.4 Zustand	27
3.2 Database sử dụng PostgreSQL	28
3.2.1 Giới thiệu về PostgreSQL	28
3.2.2 Ưu điểm	29
3.2.3 Nhược điểm	29
3.3 Công nghệ ở Back-end	29
3.3.1 Vai trò của back-end trong ứng dụng	29
3.3.2 Ngôn ngữ lập trình Java	30
3.3.3 SpringBoot	31
3.3.4 Spring Security	31
3.3.5 Hibernate	32
3.4 RESTful API	32

3.4.1	Khái niệm	32
3.4.2	Ưu điểm	33
3.4.3	Nhược điểm	34
3.5	Docker	34
3.5.1	Lịch sử và Mục đích	34
3.5.2	Ưu điểm của Docker.....	34
3.5.3	Các Thành Phần Chính của Docker.....	35
3.5.4	Nhược điểm	35
	CHƯƠNG 4: PHÂN TÍCH THIẾT KẾ	36
4.1	Nhu cầu người dùng hệ thống	36
4.1.1	Actor	36
4.1.2	Nhu cầu Quản trị viên.....	36
4.1.3	Nhu cầu người dùng	36
4.1.4	Nhu cầu khách vãng lai	36
4.2	Yêu cầu chức năng	37
4.2.1	Chức năng chung của khách và người dùng.....	37
4.2.2	Chức năng riêng dành cho người dùng có tài khoản	37
4.2.3	Các chức năng ở Quản trị viên	38
4.3	Yêu cầu phi chức năng	38
4.3.1	Thiết kế responsive đa thiết bị.....	38
4.3.2	Thiết kế đa nền tảng.....	39
4.3.3	Hệ thống có tính bảo mật.....	39
4.3.4	Hiệu suất	40
4.3.5	Khả năng bảo trì	40
4.3.6	Hệ thống có giao diện dễ nhìn, đồng đều, vừa mắt, dễ sử dụng.....	40
4.4	Use case diagram cho toàn hệ thống	41
4.5	Đặc tả Use case.....	41
4.5.1	Use cases cho Người dùng	41
4.5.2	Use Cases chung cho Quản trị viên và Người dùng	58
4.5.3	Use Cases cho Quản trị viên.....	60
4.5.4	Use Cases chung cho Người dùng và Khách.....	62
4.6	Mô hình kiến trúc hệ thống	66
4.7	Layered architecture diagram	67
4.7.1	Cho Người dùng và khách.....	67
4.7.2	Cho Quản trị viên	69
4.8	Deploy diagram	71
4.9	Sequence diagram cho các use case	72
4.9.1	Với actor Quản trị viên.....	72
4.9.2	Với actor Người cho thuê và Người thuê sách	73

4.9.3	Với actor người mua sách và bán sách	74
4.10	Activity diagram (Lược đồ hoạt động)	75
4.10.1	Quản trị viên và Người dùng với chức năng Đăng nhập	75
4.10.2	Người dùng quản lý bài đăng	77
4.10.3	Người dùng thuê tài liệu	79
4.10.4	Người dùng mua tài liệu	81
4.11	Lược đồ quan hệ cơ sở dữ liệu (ERD)	83
4.11.1	Lược đồ ERD vắn tắt	83
4.11.2	Lược đồ ERD đầy đủ	86
4.11.3	Lược đồ ERD implement	88
4.12	Class diagram	90
CHƯƠNG 5:	HIỆN THỰC BACK-END	92
5.1	Hiện thực database	92
5.2	Phân tích source-code back-end	93
5.2.1	Thư mục gốc	94
5.2.2	Thư mục chính	94
5.3	Hiện thực api	95
5.3.1	Lớp Controller và Dto	95
5.3.2	Lớp Entity	96
5.3.3	Lớp Mapper và repository	97
5.3.4	Lớp Service	98
5.4	Api xây dựng trong dự án	98
CHƯƠNG 6:	HIỆN THỰC FRONT END	101
6.1	Phân tích source code Front-end	101
6.1.1	Tổng quan thư mục Front-end	101
6.1.2	Thư mục api	102
6.1.3	Thư mục app	102
6.1.4	Thư mục components	104
6.1.5	Thư mục hooks	107
6.1.6	Thư mục types	107
6.1.7	Thư mục utils	108
6.2	Bộ nhận diện thương hiệu	108
6.2.1	Logo	108
6.2.2	Màu sắc	109
6.2.3	Font chữ	110
6.2.4	Component chung	110
6.3	Giao diện	112
6.3.1	Trang chủ	114
6.3.2	Trang tìm kiếm và hiển thị danh sách bài đăng	117

6.3.3	Trang chi tiết bài đăng	120
6.3.4	Trang giỏ hàng.....	124
6.3.5	Trang đặt hàng	127
6.3.6	Trang xuất đơn hàng.....	131
6.3.7	Trang lấy đơn hàng.....	134
6.3.8	Trang đăng nhập	135
6.3.9	Trang đăng ký	137
6.3.10	Trang quản lý thông tin cá nhân	139
6.3.11	Trang quản lý đơn thuê.....	142
6.3.12	Trang quản lý đơn hàng cho thuê	147
6.3.13	Trang quản lý đơn mua.....	149
6.3.14	Trang quản lý đơn bán.....	152
6.3.15	Trang chi tiết đơn hàng.....	152
6.3.16	Trang quản lý bài đăng	154
6.3.17	Trang tạo bài đăng	156
6.3.18	Trang quản lý voucher của tôi	159
6.3.19	Trang đăng nhập admin	162
6.3.20	Trang admin quản lý đơn hàng thuê	165
6.3.21	Trang quản lý đơn hàng mua bán admin	168
6.3.22	Trang quản lý voucher toàn sàn admin.....	168
6.3.23	Trang thêm và sửa voucher toàn sàn admin	171
6.3.24	Trang đăng ký admin	173
6.3.25	Trang chính sách bảo mật.....	176
6.3.26	Trang hướng dẫn về tình trạng sách	178
6.4	Site map giao diện hệ thống	180
CHƯƠNG 7: KIỂM THỬ	182
7.1	Công cụ kiểm thử	182
7.1.1	Công cụ hỗ trợ kiểm thử postman	182
7.1.2	Công cụ hỗ trợ kiểm thử playwright.....	183
7.1.3	Kiểm thử hiệu suất với Developer tool.....	183
7.2	Kiểm thử api	184
7.2.1	Cách thực hiện	184
7.2.2	Kết quả testcase tổng hợp	186
7.2.3	Kịch bản kiểm thử và kết quả kiểm thử.....	187
7.3	Kiểm thử tổng hợp.....	199
7.3.1	Cách thực hiện	199
7.3.2	Kịch bản kiểm thử	200
7.3.3	Báo cáo tổng hợp kết quả	210
7.3.4	Báo cáo từng testcase	211

7.4	Kiểm thử hiệu suất.....	215
7.4.1	Phân tích chi tiết từ Chrome DevTools	216
7.4.2	Đề xuất cải thiện	220
	CHƯƠNG 8: TRIỂN KHAI HỆ THỐNG	222
8.1	Triển khai Back-end	222
8.1.1	Tạo instance trên EC2	222
8.1.2	Kết nối và đưa source code back-end lên EC2.....	225
8.2	Triển khai Database	225
8.2.1	Cài đặt Docker	225
8.2.2	Thiết lập mạng Docker	226
8.2.3	Cài đặt và chạy PostgreSQL.....	226
8.2.4	Thiết lập và chạy backend	226
8.2.5	Kiểm tra hệ thống	227
8.3	Triển khai Front-end	228
	CHƯƠNG 9: KẾT LUẬN	232
9.1	Kết quả đạt được so với giai đoạn 2 – Đồ án tốt nghiệp	232
9.2	Khả năng mở rộng của dự án.....	232
9.3	Kết luận	233
	TÀI LIỆU THAM KHẢO	234
	PHỤ LỤC	235
1.	Gantt chart	235
2.	Danh mục chức năng	239

DANH MỤC HÌNH ẢNH

Hình 1: Giao diện BooksRun	17
Hình 2: Giao diện Chegg	19
Hình 3: Đồ thị phần trăm việc sử dụng tag tên trên Stackoverflow	22
Hình 4: Biểu đồ đường số lượng repository sử dụng framework.....	22
Hình 5: Biểu đồ phần trăm ngôn ngữ được yêu thích của Stackoverflow 2020.....	25
Hình 6: Thống kê Github Stars cho các state management.....	28
Hình 7: Nguyên tắc hoạt động của Restful API	33
Hình 8: Biểu đồ độ phủ sóng của trình duyệt trên toàn cầu (10/10/2023)	39
Hình 9: Use case diagram cho toàn bộ hệ thống	41
Hình 10: Module quản lý tài khoản	41
Hình 11: Module xem danh sách giỏ hàng	43
Hình 12: Module quản lý đơn hàng	49
Hình 13: Module quản lý đơn hàng của khách	55
Hình 14: Module quản lý bài đăng	56
Hình 15: Module quản lý đơn hàng của quản trị viên	60
Hình 16: Mô hình kiến trúc hệ thống MVC	66
Hình 17: Layered architecture diagram cho người dùng và khách	67
Hình 18: Layered architecture diagram cho quản trị viên	69
Hình 19: Deploy diagram	71
Hình 20: Sequence diagram cho quản trị viên	72
Hình 21: Sequence diagram cho người dùng và khách thuê sách	73
Hình 22: Sequence diagram với người mua và bán sách	74
Hình 23: Activity diagram Quản trị viên và Người dùng với chức năng Đăng nhập.....	75
Hình 24: Activity diagram người dùng quản lý bài đăng	77
Hình 25: Activity diagram thuê và trả tài liệu	79
Hình 26: Activity diagram Người dùng mua tài liệu	81
Hình 27: Lược đồ ERD vắn tắt	83
Hình 28: ERD cho toàn hệ thống	86
Hình 29: Lược đồ ERD implement	88
Hình 30: Class diagram cho hệ thống	90
Hình 32: Phiên bản logo không chữ	108
Hình 33: Logo với tên dự án The Flying Bookstore	109
Hình 34: Bảng hệ thống màu	110
Hình 35: Component list menu	111
Hình 36: Component search	111
Hình 37: Component header	111
Hình 38: Footer component	112
Hình 39: Trang chủ	114
Hình 40: Trang danh sách bài đăng và tìm kiếm tài liệu	117
Hình 41: Trang chi tiết bài đăng	120
Hình 42: Tab đánh giá	122
Hình 43: Trang giỏ hàng	124
Hình 44: Trang đặt hàng	127
Hình 45: Phần thông báo chuyển khoản trang đặt hàng	130
Hình 46: Trang xuất đơn hàng	131
Hình 47: Trang lấy đơn hàng	134
Hình 48: Trang đăng nhập	135
Hình 49: Trang đăng ký	137
Hình 50: Trang quản lý thông tin cá nhân	139
Hình 51: Trang quản lý đơn thuê	142

Hình 52: Đơn hàng có thẻ hủy.....	142
Hình 53: Đơn hàng có thẻ xác nhận thanh toán	144
Hình 54: Đơn hàng xác nhận lấy hàng	145
Hình 55: Đơn hàng xác nhận trả sách.....	145
Hình 56: Quản lý đơn hàng của khách	147
Hình 57: Chủ sách xác nhận nhận lại sách	148
Hình 58: Trang quản lý đơn mua.....	149
Hình 59: Giao diện ở người mua khi mới đặt hàng.....	150
Hình 60: Giao diện của người bán khi mới đặt hàng.....	150
Hình 61: Giao diện của người mua khi người mua đã thanh toán.....	150
Hình 62: Giao diện của người bán khi người mua đã thanh toán.....	150
Hình 63: Giao diện người mua khi người mua nhận được sách.....	150
Hình 64: Giao diện người bán khi người mua nhận được sách.....	151
Hình 65: Giao diện người mua khi quản trị viên đã trả tiền cho người bán sách.....	151
Hình 66: Giao diện người bán khi quản trị viên đã trả tiền cho người bán sách.....	151
Hình 67: Trang quản lý đơn bán.....	152
Hình 68: Trang chi tiết đơn hàng.....	152
Hình 69: Trang quản lý bài đăng	154
Hình 70: Trang tạo bài đăng	157
Hình 71: Chọn sách có sẵn	157
Hình 72: Trang quản lý voucher của tôi	160
Hình 73: Trang đăng nhập admin	163
Hình 74: Trang admin quản lý đơn thuê.....	165
Hình 75: Trang quản lý đơn hàng mua bán admin	168
Hình 76: Trang quản lý voucher toàn sàn admin	168
Hình 77: Trang thêm và sửa voucher toàn sàn admin	171
Hình 78: Trang đăng ký admin.....	173
Hình 79: Trang chính sách bảo mật.....	176
Hình 80: Trang hướng dẫn về tình trạng sách	178
Hình 81: Site map giao diện hệ thống	180
Hình 82: Cấu trúc Postman.....	182
Hình 83: Logo playwright	183
Hình 84: Viết test case đăng nhập trong postman	184
Hình 85: Kết quả run testcase	185
Hình 86: Kết quả testcase tổng hợp	186
Hình 87: Tổng hợp kết quả test case	210
Hình 88: Bảng tổng hợp kết quả test case	210
Hình 89: Kết quả testcase Playwright.....	211
Hình 90: Kết quả testcase Playwright.....	212
Hình 91: Kết quả testcase Playwright.....	213
Hình 92: Kết quả testcase Playwright.....	214
Hình 94: Phân tích chi tiết từ Chrome DevTools	215
Hình 95: Các thông số đo lường chính:.....	216
Hình 96: LCP theo giai đoạn	217
Hình 97 LCP Request Discovery	219
Hình 98: Launch một instance bên trong EC2 tên là “The flying bookstore”.....	222
Hình 99: Tạo hoặc chọn Key pair có sẵn.....	222
Hình 100: Tick chọn 2 ô Allow HTTP và HTTPS traffic	223
Hình 101: Instance tạo thành công	224
Hình 102: Hướng dẫn kết nối tới instance của EC2	225
Hình 103: Tạo mới Project Vercel	228
Hình 104: Thiết lập biến môi trường	229

Hình 105: Quản lý deployment tại trang dashboard của Vercel.....	230
Hình 106: Deploy Front-end thành công.....	231
Hình 107: Gantt chart làm tính năng chung và Frontend	235
Hình 108: Gantt chart xây dựng chức năng Mua bán sách.....	236
Hình 109: Gantt chart xây dựng chức năng Quản lý voucher	237
Hình 110: Gantt chart xây dựng chức năng Quản lý bài đăng	238

DANH MỤC BẢNG BIÊU

Bảng 1: Use case Người dùng xem thông tin cá nhân.....	42
Bảng 2: Use case Người dùng sửa thông tin cá nhân	43
Bảng 3: Use case Người dùng xem sản phẩm thuê	44
Bảng 4: Use case Người dùng xem sản phẩm mua	44
Bảng 5: Use case Người dùng xóa sản phẩm	44
Bảng 6: Use case Người dùng xem danh sách khuyến mãi.....	45
Bảng 7: Use case Người dùng áp dụng khuyến mãi.....	45
Bảng 8: Use case Người dùng tìm khuyến mãi	46
Bảng 9: Use case Người dùng đặt thuê	47
Bảng 10: Use case Người dùng đặt mua	48
Bảng 11: Use case Người dùng xem danh sách đơn hàng.....	49
Bảng 12: Use case Người dùng xem danh sách đơn đã đặt.....	50
Bảng 13: Use case Người dùng xem danh sách đơn đã nhận.....	50
Bảng 14: Use case Người dùng xem danh sách đơn đã quá hạn	51
Bảng 15: Use case Người dùng xem danh sách đơn đã trả sách	51
Bảng 16: Use case Người dùng xem danh sách đơn đã huỷ.....	52
Bảng 17: Use case Người dùng xác nhận đã trả tiền cọc	52
Bảng 18: Use case Người dùng đánh giá đơn hàng.....	53
Bảng 19: Use case Người dùng xác nhận trả tài liệu.....	53
Bảng 20: Use case Người dùng xác nhận nhận tài liệu	54
Bảng 21: Use case Người dùng xác nhận huỷ đơn hàng	54
Bảng 22: Use case Người dùng xem chi tiết đơn hàng	54
Bảng 23: Use case Người dùng xác nhận nhận lại sách	55
Bảng 24: Use case Người dùng xem danh sách bài đăng	56
Bảng 25: Use case Người dùng tạo bài đăng.....	57
Bảng 26: Use case Người dùng tìm kiếm bài đăng	57
Bảng 27: Use case Người dùng xóa bài đăng	58
Bảng 28: Use case Người dùng/Quản trị viên đăng nhập	59
Bảng 29: Use case Người dùng/Quản trị viên đăng xuất	59
Bảng 30: Use case Quản trị viên xem danh sách đơn hàng thuê	60
Bảng 31: Use case Quản trị viên xem danh sách đơn hàng mua	61
Bảng 32: Use case Quản trị viên duyệt đơn hàng.....	61
Bảng 33: Use case Quản trị viên duyệt trả tiền người thuê	62
Bảng 34: Use case Quản trị viên duyệt trả tiền người bán	62
Bảng 35: Use case Khách đăng ký tài khoản	63
Bảng 36: Use case Người dùng/Khách xem danh sách đề xuất	63
Bảng 37: Use case Người dùng/Khách xem chi tiết bài đăng	64
Bảng 38: Use case Người dùng/Khách tìm kiếm tài liệu	64
Bảng 39: Use case Người dùng/Khách thêm vào giỏ hàng	65
Bảng 40: Use case Người dùng/Khách xem giỏ hàng	65
Bảng 41: Api xây dựng trong dự án	100
Bảng 42: Mô tả chi tiết Trang chủ	115
Bảng 43: Tham số đầu vào API Tìm kiếm danh sách sách	116
Bảng 44: Mô tả chi tiết Trang danh sách bài đăng và tìm kiếm tài liệu	118
Bảng 45: Mô tả chi tiết Trang chi tiết bài đăng	122
Bảng 46: Mô tả chi tiết Tab đánh giá	123
Bảng 49: Tham số đầu vào API Lấy thông tin chi tiết tài liệu	123
Bảng 59: Tham số đầu vào API Đánh giá sách	123
Bảng 47: Mô tả chi tiết Trang giỏ hàng.....	125
Bảng 59: Tham số đầu vào API Tìm kiếm voucher session theo từ khóa.....	126
Bảng 59: Tham số đầu vào API Tìm kiếm voucher shop theo từ khóa.....	126

Bảng 48: Mô tả chi tiết Trang đặt hàng	129
Bảng 49: Tham số đầu vào API Tạo đơn hàng bán.....	129
Bảng 50: Tham số đầu vào API Cập nhật thông tin người dùng.....	130
Bảng 51: Mô tả chi tiết Trang xuất đơn hàng	133
Bảng 59: Tham số đầu vào API Lấy thông tin đơn hàng bán.....	134
Bảng 52: Mô tả chi tiết Trang đăng nhập	136
Bảng 53: Tham số đầu vào API Đăng nhập người dùng	136
Bảng 54: Mô tả chi tiết Trang đăng ký	138
Bảng 55: Tham số đầu vào API Đăng ký tài khoản mới	139
Bảng 56: Mô tả chi tiết Trang quản lý thông tin cá nhân	140
Bảng 57: Tham số đầu vào API Cập nhật thông tin người dùng.....	141
Bảng 58: Mô tả chi tiết Trang quản lý đơn thuê.....	144
Bảng 59: Tham số đầu vào API Tìm kiếm đơn thuê theo người cho thuê	145
Bảng 60: Tham số đầu vào API Tìm kiếm đơn thuê theo người thuê.....	146
Bảng 61: Tham số đầu vào API Cập nhật trạng thái đơn thuê	146
Bảng 62: Mô tả chi tiết Trang chi tiết đơn hàng	153
Bảng 63: Tham số đầu vào API Lấy chi tiết đơn thuê.....	154
Bảng 64: Mô tả chi tiết Trang quản lý bài đăng	155
Bảng 65: Tham số đầu vào API Xóa danh sách	156
Bảng 66: Mô tả chi tiết Trang tạo bài đăng	158
Bảng 67: Tham số đầu vào API Tạo danh sách mới	159
Bảng 68: Mô tả chi tiết Trang quản lý voucher của tôi	161
Bảng 69: Tham số đầu vào API Lấy danh sách voucher theo người bán.....	162
Bảng 70: Tham số đầu vào API Xóa voucher theo người bán	162
Bảng 71: Mô tả chi tiết Trang đăng nhập admin	164
Bảng 53: Tham số đầu vào API Đăng nhập admin	164
Bảng 72: Mô tả chi tiết Trang quản lý đơn thuê cho thuê admin	166
Bảng 63: Tham số đầu vào API Cập nhập đơn thuê.....	167
Bảng 73: Mô tả chi tiết Trang quản lý voucher toàn sàn admin.....	170
Bảng 75: Tham số đầu vào API Xóa voucher toàn sàn	171
Bảng 76: Mô tả chi tiết Trang thêm và sửa voucher toàn sàn admin	172
Bảng 77: Tham số đầu vào API Thêm voucher toàn sàn mới	173
Bảng 78: Mô tả chi tiết Trang đăng ký admin.....	175
Bảng 79: Tham số đầu vào API Đăng ký tài khoản mới	176
Bảng 80: Mô tả chi tiết Trang chính sách bảo mật.....	177
Bảng 81: Mô tả chi tiết Trang hướng dẫn về tình trạng sách	179
Bảng 82: Thông số quan trọng của kết quả chạy test	186
Bảng 83: Kịch bản kiểm thử và kết quả kiểm thử	199
Bảng 84: Kết quả của kiểm thử tổng hợp theo tên api unit test.....	199
Bảng 85: Kịch bản kiểm thử tự động	210
Bảng 86: Danh mục chức năng.....	240

CHƯƠNG 1: TỔNG QUAN

1.1 Giới thiệu đề tài

Hàng năm, các bạn học sinh, sinh viên mọi nơi trên thế giới nói chung và trên đất nước Việt Nam nói riêng liên tục tiến hành các công việc học tập và nghiên cứu. Do đó việc các bạn sử dụng nhiều loại tài liệu là một điều tất yếu. Trong đó tài liệu giấy là loại tài liệu chiếm tỷ lệ lớn, ngoài các tài liệu sử dụng rộng rãi như các loại sách, giáo trình, tạp chí thì nhiều tài liệu dưới dạng bản viết tay chứa nhiều kiến thức có giá trị cũng được tạo ra. Tuy nhiên sau quá trình học tập nghiên cứu chúng lại ít được sử dụng lại và có phần trở nên vô dụng và chiếm diện tích mỗi khi các bạn qua môn mới hoặc lên lớp mới. Từ đó mới phát sinh nhu cầu thuê mượn tài liệu để sử dụng trong quá trình học tập nghiên cứu, sau khi sử dụng xong thì trả lại. Việc này vừa giảm chi phí so với việc mua mới và tránh lãng phí tài nguyên, ngoài ra các bản viết tay chứa kinh nghiệm quý giá trong quá trình học tập sẽ không trở thành một bản viết vô dụng.

Internet, một môi trường rất đỗi quen thuộc với các bạn học sinh sinh viên, phải nói đúng hơn là chiếm vai trò khá là quan trọng. Việc này đã được chứng minh trong đợt Covid vừa qua khi toàn bộ công việc học hành làm việc của mọi người tiến hành trên môi trường internet. Vì vậy sẽ thuận tiện hơn nếu việc trao đổi, thuê mượn tài liệu được tiến hành trên môi trường này. Hiện trên các mạng xã hội cũng có các hội nhóm trao đổi và thuê mượn sách. Tuy nhiên nó mang tính cá nhân, chưa có một cơ chế quản lý tập trung nào, chủ yếu được phát triển bởi các thư viện hoặc các cá nhân yêu sách nhưng vẫn còn thô sơ. Một sàn thương mại điện tử tập trung, phục vụ cho việc thuê mướn, trao đổi tài liệu sẽ giúp các bạn học sinh, sinh viên thuận tiện hơn trong việc quản lý tài liệu. Vừa đỡ tốn thời gian, có thể chủ động mà không phụ thuộc vào người khác.

Ý tưởng được xuất phát và phát triển từ Quản trị viên của một page trên facebook. Page được tạo ra với mục đích trao đổi sách giữa các bạn sinh viên, học sinh. Khi các bạn sinh viên học xong một môn hoặc các bạn học sinh học xong một lớp, để tối ưu hóa chi phí thì sách đó sẽ được trao đổi giữa các bạn. Kết quả là mọi người đều đảm bảo có sách để học, các ghi chú của các bạn trước cũng được truyền lại cho các bạn học sau. Ban đầu page diễn ra trơn tru và ổn thoả, nhưng sau đó phát sinh vấn đề lớn bởi lượng công việc của Quản trị viên quá nhiều. Quản trị viên phải làm tất cả các việc đăng bài tìm bạn mượn sách, liên hệ bạn để lấy lại sách,... . Từ đó mà quyết định làm một hệ thống cho thuê mướn mua bán sách cũ ra đời.

1.2 Mục tiêu đề tài

Mục tiêu của em là xây dựng một sàn thương mại điện tử với mục đích:

Là nơi có thể giúp bạn sinh viên đem cho thuê, mượn tài liệu của mình. Đồng thời giúp quản lý quá trình cho thuê, mượn.

Là nơi các bạn sinh viên có thể mua bán tài liệu, Quản lý đơn hàng mua bán của mình.

1.3 Phạm vi đề tài

Để tập trung tốt nhất vào tối ưu source code và làm được thành phẩm cuối cùng hoàn chỉnh, dự án chỉ tập trung phát triển các chức năng cơ bản của sàn giao dịch C2C:

- Cung cấp các chức năng để người dùng đăng nhập và đăng xuất hệ thống. Đồng thời có chức năng đăng ký để người dùng có thể dễ dàng trở thành một thành viên của hệ thống. Trong hệ thống, chỉ có thể trở thành một người dùng của hệ thống thì mới có thể phát sinh đơn hàng với những người dùng khác trong hệ thống. Còn những khách ngoại lai ghé thăm hệ thống khi chưa có tài khoản vẫn có thể tìm kiếm và xem được danh sách bài đăng và chi tiết bài đăng.

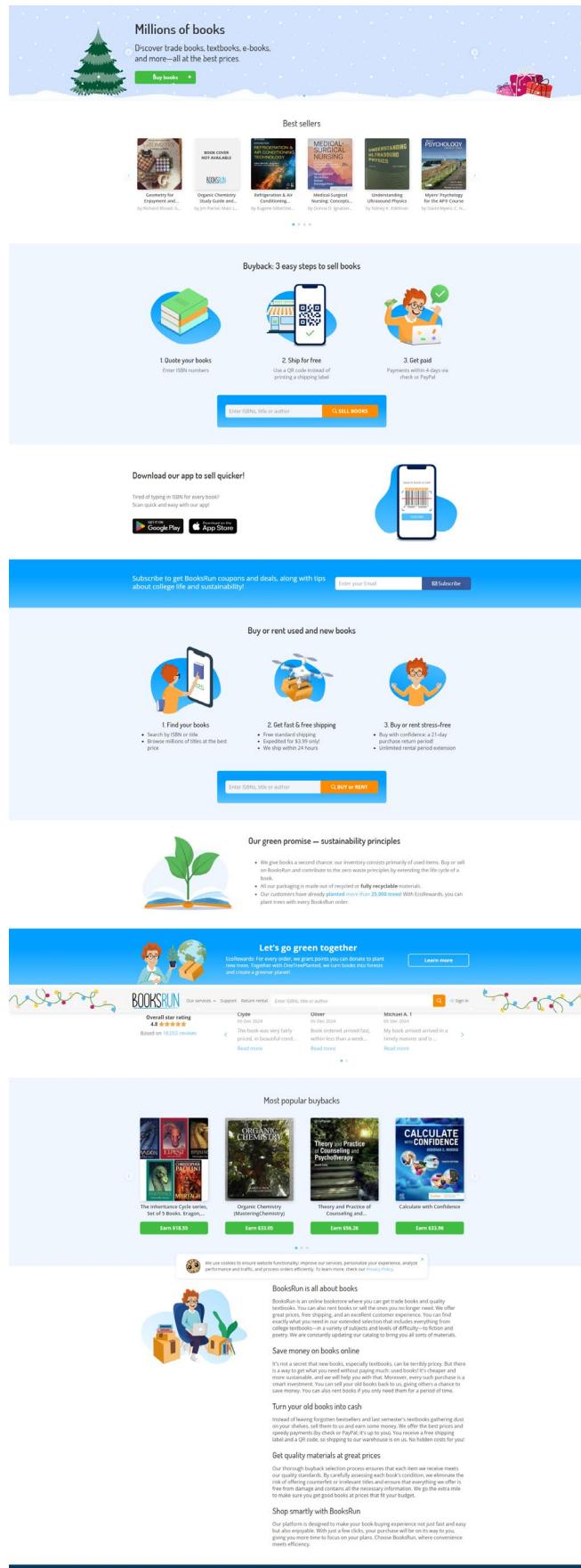
- Hệ thống chỉ cho phép phát sinh đơn hàng giữa người chủ sở hữu và một người dùng khác. Chủ sở hữu sách sẽ không thể tạo đơn hàng với sách của mình.
- Cung cấp các chức năng thể hiện thông tin bài đăng, tìm kiếm thông tin bài đăng, giúp người chủ sách và khách hàng tìm được nhau thông qua thông tin chung đó. Từ đó phát sinh giao dịch.
- Về việc thuê mướn, hệ thống giúp các bên giao tiếp bằng việc cung cấp thông tin liên lạc cơ bản của họ cho nhau từ đó thực hiện giao dịch với nhau thông qua mô giới là sàn giao dịch cho thuê sách.
- Cung cấp các chức năng đơn hàng giúp người bán và người thuê quản lý đơn hàng của mình. Thông qua những danh sách đơn hàng trực quan và phân chia theo tình trạng, trạng thái.
- Người dùng sẽ chủ động trong việc đặt thuê, huỷ đơn thuê, thanh toán, xác nhận thời điểm trả tài liệu. Việc chủ động này sẽ được hệ thống hiện thực bằng các nút xác nhận như xác nhận chuyển tiền cọc, xác nhận nhận hàng, xác nhận trả hàng, xác nhận nhận lại hàng. Điều này giúp người thuê và người cho thuê có thể chủ động hơn trong việc thuê mướn. Chủ động hơn trong việc quản lý đơn hàng của mình.
- Dự án cung cấp chức năng về thông tin người dùng, đây là một thông tin đáng quan trọng, bởi nó sẽ ảnh hưởng trực tiếp trong giao tiếp giữa người cho thuê và khách thuê. Hệ thống sẽ cung cấp thông tin này cho hai bên phát sinh giao dịch với nhau. Nên các chức năng chỉnh sửa thông tin cá nhân cũng được tích hợp vào dự án.
- Chức năng Quản trị viên đối với đơn hàng: với mục đích ban đầu là giảm bớt áp lực gánh nặng cho Quản trị viên. Các chức năng trên đã giúp Quản trị viên giảm tải việc tự đăng bài và tự liên hệ với bên thuê và bên cho thuê. Giờ đây Quản trị viên sẽ tập trung nhiều vào quản lý đơn hàng bằng giao diện duyệt đơn hàng của Quản trị viên, không bị quá tải bởi những việc ngoài luồng khác.
- Để hạn chế việc mướn không trả thì hệ thống tiến hành dựa trên phí cọc, và phí phạt. Bên trong hệ thống ngày nhận sẽ được cập nhật lại khi người thuê xác nhận nhận hàng. Ngày trả là mốc giao kèo cứng với người cho thuê. Trả sớm thì sẽ được cập nhật lại ngày và tổng tiền thuê. Quá hạn sẽ không được cập nhật lại và bị tính phí phạt. Đây là mốc để hạn chế người thuê trả trễ.
- Để hạn chế việc mua xong không trả tiền người mua cũng cần phải đưa tiền mua sách trước và khi Quản trị viên nhận tiền thì người mua sẽ nhận được tài liệu.
- Dự án còn cung cấp voucher toàn sàn theo mùa và voucher theo người bán, người bán và Quản trị viên có thể thêm xóa sửa voucher cho các đơn hàng của mình tùy vào các dịp
- Chức năng review: Giúp người thuê mới có cái nhìn và so sánh về các bài đăng đồng thời phản ánh phần nào vấn đề phục vụ, thái độ phục vụ của người cho thuê.

CHƯƠNG 2: CÁC HỆ THỐNG LIÊN QUAN

2.1 BooksRun

2.1.1 Giới thiệu

BooksRun là hệ thống cho thuê, ký gửi sách cũ, mua bán sách cũ được đầu tư giao diện đẹp mắt, lén được top SEO google với mục đích cho thuê sách và mua bán sách. Hệ thống BooksRun còn đầu tư xây dựng app mobile cho người dùng dễ dàng thuê và mua sách cũ. Hệ thống hỗ trợ tìm kiếm sách qua mã ISBN, cung cấp các tính năng vận chuyển miễn phí và quy trình hoàn tiền nhanh chóng. BooksRun cam kết với mục tiêu phát triển bền vững bằng cách sử dụng vật liệu tái chế và tham gia các hoạt động trồng cây, giúp người dùng không chỉ tiết kiệm mà còn đóng góp cho môi trường.



Hình 1: Giao diện BooksRun

2.1.2 Ưu điểm

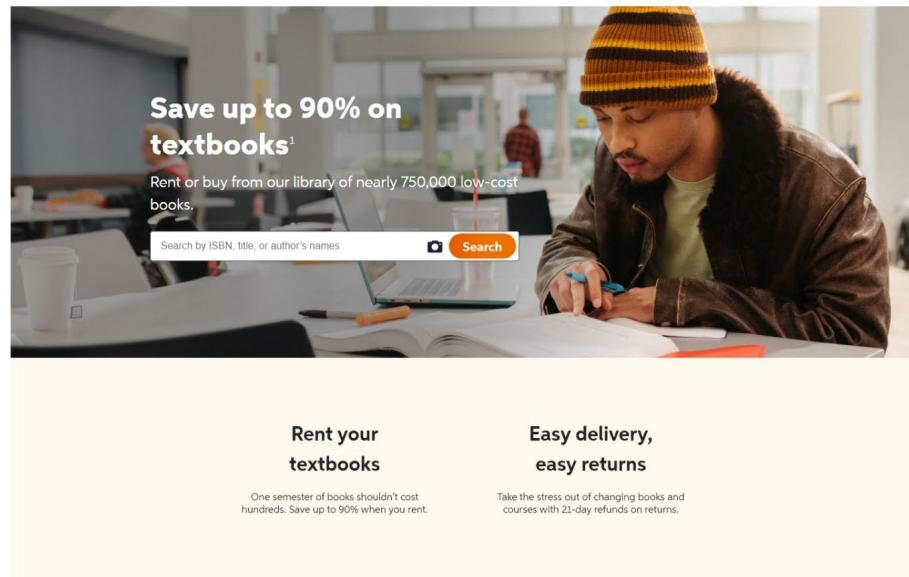
- **Chính sách thân thiện môi trường:** BooksRun hỗ trợ tái chế sách cũ, sử dụng vật liệu đóng gói thân thiện môi trường và tham gia các chương trình trồng cây, điều này nổi bật hơn trong việc bảo vệ môi trường.
- **Hỗ trợ đa dạng hình thức mua bán:** BooksRun kết hợp mua bán, ký gửi và cho thuê sách cũ, giúp tối đa hóa lợi ích sử dụng tài nguyên sách cũ.
- **Ứng dụng di động tiện lợi:** BooksRun có ứng dụng trên iOS và Android giúp người dùng dễ dàng quét mã ISBN để định giá sách hoặc tìm kiếm tài liệu nhanh chóng.
- **Giao diện hiện đại, dễ sử dụng:** Thiết kế giao diện thân thiện với trải nghiệm người dùng, phù hợp với nhiều đối tượng.
- **Chính sách vận chuyển miễn phí và hoàn tiền nhanh:** Người dùng có thể nhận tiền hoặc thanh toán nhanh chóng, tạo sự an tâm và tiện lợi khi giao dịch.

2.1.3 Nhược điểm

- **Mô hình B2C hạn chế tính cộng đồng:** BooksRun hoạt động theo mô hình doanh nghiệp-đến-người dùng, thiếu sự kết nối trực tiếp giữa các cá nhân, trong khi hệ thống của bạn khuyến khích giao dịch giữa người dùng với nhau (C2C), tối ưu hóa sự cạnh tranh giá cả.
- **Hạn chế trong quản lý đơn hàng cá nhân:** Hệ thống của bạn cung cấp tính năng quản lý đơn hàng chi tiết với nhiều trạng thái, giúp người dùng kiểm soát giao dịch hiệu quả hơn.
- **Không có cơ chế tương tác trực tiếp:** BooksRun không cung cấp nền tảng giao tiếp trực tiếp giữa người thuê và người cho thuê như hệ thống của bạn, làm giảm tính cá nhân hóa trong giao dịch.
- **Không tích hợp tính năng voucher linh hoạt:** BooksRun chưa hỗ trợ các chương trình voucher cá nhân hóa, trong khi hệ thống của bạn cho phép người dùng và quản trị viên chủ động tạo voucher theo sự kiện hoặc cá nhân.
- **Không tập trung vào mạng lưới địa phương:** Hệ thống của chúng tôi cho phép kết nối địa phương, giúp giảm chi phí vận chuyển và tăng tính linh hoạt, điều mà BooksRun không khai thác.

2.2 Chegg

2.2.1 Giới thiệu



“PSA Chegg rentals saved me 500 bucks this semester”

@alyssamtavares

This section contains two side-by-side promotional boxes. The left box features icons of a smartphone and a tablet, with the text "eBooks that plug into your life" and "Choose from over 400,000 eBooks for as low as \$4.99." It includes a "Learn more →" link. The right box features icons of a laptop, a smartphone, and a tablet, with the text "Study help made for your course" and "Pair your books with Chegg Study for step-by-step solutions & study tools." It also includes a "Learn more →" link.

This screenshot shows the "Top Textbooks" section of the Chegg website. It displays three book covers: "Delivering Health Care in America (7th edition)", "Human Anatomy & Physiology (11th Edition)", and "Organic Chemistry (3rd Edition)". Each book has a "Solutions" and an "eTextbook" link. Below this is the Chegg logo and navigation links for "Study", "Learning Lab", "For educators", "Help", and "Sign in".

Browse Textbooks by Popular Subject
Anatomy & Physiology | Art | Biology | Business & Economics | Calculus | Chemistry | Education | History | Language Arts | Literary Criticism | Mathematics | Nursing | Political Science | Psychology | Reference | Science

Hình 2: Giao diện Chegg

Chegg Books là một nền tảng cho thuê và mua bán sách giáo khoa trực tuyến, cho phép người dùng tiết kiệm lên đến 90% chi phí thông qua việc thuê sách. Nền tảng cung cấp các tính năng như tìm kiếm sách bằng ISBN, hỗ trợ eBook, và tích hợp các công cụ học tập giúp người dùng dễ dàng tiếp cận tài liệu học tập. Chegg cũng nổi bật với giao diện thân thiện và tối ưu hóa trải nghiệm người dùng, cùng với chính sách đổi trả linh hoạt.

2.2.2 Ưu điểm

- Thư viện sách đa dạng và eBook:** Chegg có sẵn hàng trăm ngàn đầu sách, bao gồm cả eBooks, giúp tiết kiệm không gian và dễ truy cập mọi lúc mọi nơi.
- Quy trình thuê sách đơn giản:** Hệ thống tự động hóa toàn bộ quá trình đặt thuê, giao nhận, và trả sách, tạo trải nghiệm trọn tru.
- Tính năng nổi bật:** Chegg hỗ trợ quét mã ISBN để tìm kiếm sách chính xác và nhanh chóng, tối ưu hóa thời gian người dùng.
- Chính sách đổi trả linh hoạt:** Với chính sách 21 ngày đổi trả, người dùng có thể dễ dàng thay đổi nếu không hài lòng.

2.2.3 Nhược điểm

- Mô hình B2C hạn chế khả năng đa dạng giá cả:** Vì Chegg hoạt động theo mô hình doanh nghiệp-đến-người dùng (B2C), giá cả không linh hoạt và không tận dụng được sự cạnh tranh từ nhiều người bán/cho thuê như trong mô hình C2C của bạn.
- Không tận dụng mạng lưới địa phương:** Hệ thống của bạn cho phép thuê sách từ các cá nhân địa phương, giảm thời gian và chi phí vận chuyển, trong khi Chegg tập trung ở mức quốc gia, hạn chế lợi ích này.
- Thiếu hỗ trợ giao tiếp giữa người dùng:** Chegg không cung cấp cơ chế giao tiếp trực tiếp giữa người mua và người bán/cho thuê, trong khi hệ thống của bạn khuyến khích giao tiếp để tối ưu hóa trải nghiệm giao dịch.
- Không hỗ trợ đa dạng quản lý đơn hàng:** Hệ thống của bạn cung cấp giao diện quản lý đơn hàng chi tiết với các trạng thái cụ thể, điều này giúp người dùng kiểm soát tốt hơn quá trình thuê/mua sách.
- Thiếu tính năng voucher cá nhân hóa:** Chegg chưa hỗ trợ voucher linh hoạt như bạn, hạn chế khuyến mãi theo từng người bán hoặc theo mùa.

2.3 So sánh với hệ thống của tôi

Tiêu chí	The Flying Bookstore (Hệ thống của tôi)	BooksRun	Chegg Books
Loại hình	Mua bán, cho thuê, kết nối giao dịch C2C	Mua bán và cho thuê sách	Mua bán, cho thuê sách, lời giải bài tập
Đối tượng sử dụng	Sinh viên	Sinh viên	Học sinh, sinh viên
Phạm vi hoạt động	Toàn quốc, hướng đến quốc tế	Mỹ, Canada, Úc, Đức, Anh	Mỹ, Ấn Độ, Đức, Anh, Israel
Phân loại tình trạng sách	Linh hoạt, người dùng tự đánh giá và quản lý và được quyền khiếu nại khi người bán đánh giá không đúng	3 loại: Nhu mô, rất tốt, chấp nhận được	Không
Shipping	Kết nối với đơn vị vận chuyển trong nước hoặc người dùng tự đến lấy	Free ship ở Mỹ	Dịch vụ bên thứ ba, có mã giảm giá ship
Chọn khoảng thời gian thuê	Cho chọn theo ngày giúp tối ưu thời gian nhất	Chỉ cho chọn theo quý	Chỉ cho chọn theo quý
Đánh giá sách	Có, người dùng trực tiếp đánh giá	Có	Không

Tính năng thêm	Voucher cá nhân hóa, tạo giao dịch C2C	Scan ISBN	Bán lời giải và tài liệu học tập, scan ISBN
-------------------	---	-----------	---

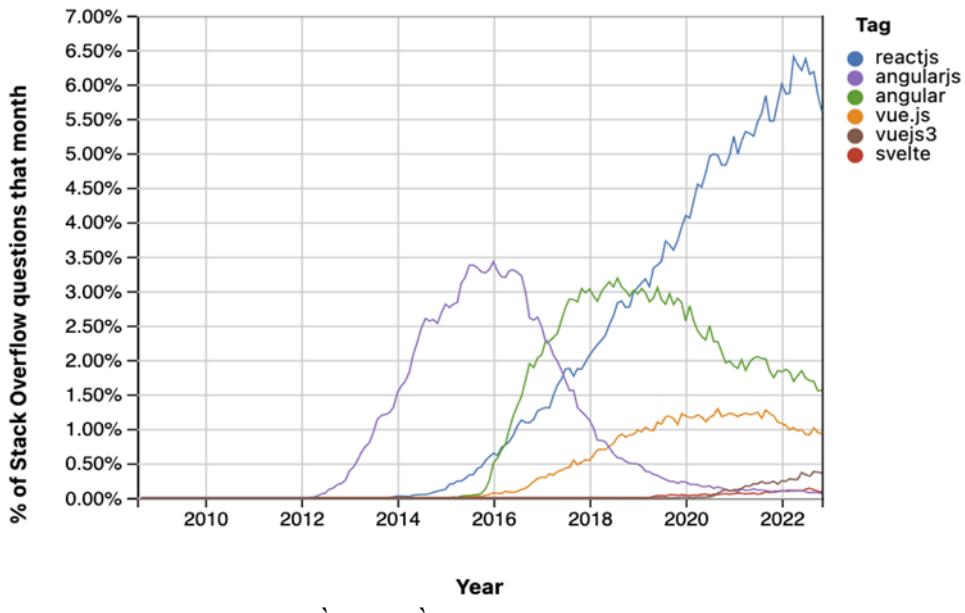
CHƯƠNG 3: CÔNG NGHỆ SỬ DỤNG

3.1 Công nghệ ở Front-end

3.1.1 ReactJS

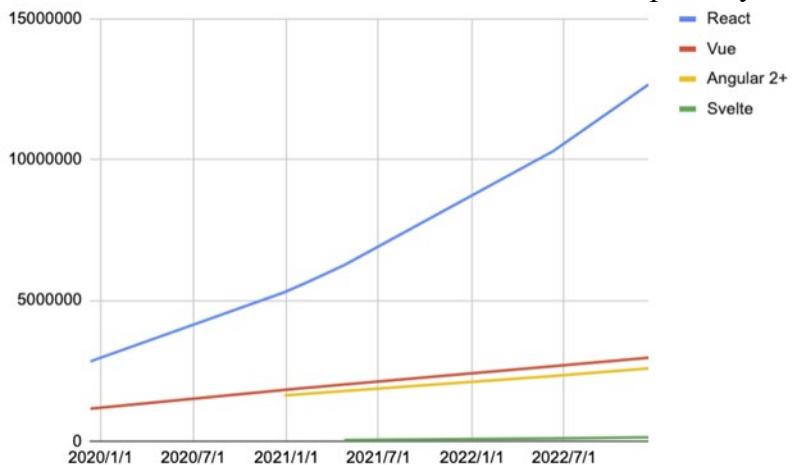
3.1.1.1 Vì sao chọn ReactJS?

Theo kết quả của Stack Overflow về xu hướng công nghệ theo thời gian. Dựa trên việc sử dụng tag trên nền tảng kể từ năm 2008 - khi Stack Overflow được thành lập - cho đến nay ReactJS vẫn đang đứng đầu.



Hình 3: Đồ thị phần trăm việc sử dụng tag tên trên Stackoverflow

Đến thời điểm hiện tại, Repository trên Github có sử dụng React lên đến gần 18 triệu repository. Cách xa đối thủ thứ hai là Vue khi Vue chỉ có 3 triệu repository.



Hình 4: Biểu đồ đường số lượng repository sử dụng framework

Và còn nhiều kết quả thống kê khác đều chứng minh React luôn đi đầu trong số các Frontend framework khác như Vue, Angular hay Svelte¹.

Ngoài ra những phiên bản mới của React đảm bảo SEO (Search Engine Optimization) có thể truy cập vào website để website có thể cạnh tranh thứ hạng so với website khác.

¹ gist.github.com. 2023. *Front-end frameworks popularity (React, Vue, Angular and Svelte)*. Truy cập từ: <https://gist.github.com/tkrotoff/b1caa4c3a185629299ec234d2314e190#github-repositories-that-depend-on>.

3.1.1.2 Giới thiệu ReactJS

Nói ngắn gọn, ReactJS (hay còn gọi tắt là React) là thư viện JavaScript để xây dựng giao diện người dùng². React ra đời từ Facebook vào năm 2013, tức là ra đời sau AngularJS. Ban đầu React sinh ra để dành cho các ứng dụng lớn dễ dàng quản lý và mở rộng. Mục tiêu chính của React là xây dựng website ở phía Frontend nhanh, đơn giản, hiệu năng cao và dễ dàng mở rộng.

ReactJS có các tính năng nổi trội như sau:

- **JSX:** là viết tắt của JavaScript extension, JSX chính là React extension, JSX giúp cho việc đổi DOM tree dễ hơn bằng HTML-style code đơn giản. JSX là một trong những tính năng tốt và dễ sử dụng, dễ đọc và dễ hiểu với những người đã quen thuộc với HTML cơ bản.
- **Components:** Website được phát triển bằng React là một sự kết hợp nhiều component lại với nhau. Chứ không phải là tất cả chung trong một Template như bình thường. Các component cũng như các hàm JavaScript giúp viết code dễ dàng vì chúng là những đoạn code độc lập và có thể tái sử dụng. Bạn có thể sử dụng component dưới dạng function hoặc class. Ngoài ra, các component còn có thể truyền state và props cho nhau.
- **Virtual DOM:** React tạo một thứ gọi là Virtual DOM (DOM ảo). Đúng như tên gọi, đó là một copy của DOM thật trên trang web đó. React dùng những DOM ảo để tìm đúng những DOM thật cần được cập nhật khi có bất kỳ sự kiện nào làm các component bên trong thay đổi.

3.1.1.3 Ưu điểm

- **Declarative (khai báo):** React giúp việc tạo giao diện người dùng trở nên dễ dàng hơn, nhanh chóng hơn. React sẽ cập nhật và hiển thị đúng các component được chỉ định một cách hiệu quả khi dữ liệu của bạn thay đổi. Declarative view làm cho code của bạn dễ dự đoán hơn và dễ debug hơn.
- **Component - Based (Xây dựng dựa trên thành phần):** Xây dựng các components được đóng gói để quản lý trạng thái của riêng chúng, sau đó kết hợp chúng để tạo các giao diện phức tạp. Vì logic thành phần được viết bằng JavaScript thay vì templates, nên bạn có thể dễ dàng truyền nhiều dạng data qua ứng dụng của mình và loại bỏ trạng thái khỏi DOM.
- **Học 1 lần, viết được mọi nơi:** React có thể render trên server dùng Node hoặc mobile app với React Native³. Khi chuyển sang xây dựng ứng dụng di động với React Native, bạn không cần phải mất thời gian học một ngôn ngữ mới hay framework mới. React mang lại sự linh hoạt khi sử dụng cùng một thư viện vào cả website và cả mobile app. React Native giúp bạn tạo một mobile app tương tác dễ dàng và hiệu suất cao cho mọi thiết bị smartphone mà không phải học hỏi một ngôn ngữ mới, điều này cũng tương tự với NodeJs khi bạn xây dựng server với Nodejs
- **React có cộng đồng người dùng rộng lớn:** React có một cộng đồng gồm hàng triệu developer như: Code of Conduct, Stack Overflow. Một số cộng đồng trao đổi khác như Cộng đồng React của DEV, Cộng đồng React của Hashnode, Reactiflux trên Discord, React trên Reddit⁴

² legacy.reactjs.org. 2023. *React – A JavaScript library for building user interfaces*. Truy cập từ: <https://legacy.reactjs.org/>.

³ legacy.reactjs.org. 2023. *React – A JavaScript library for building user interfaces*. Truy cập từ: <https://legacy.reactjs.org/>.

⁴ react.dev. 2023. *React Community – React*. Truy cập từ: <https://react.dev/community>.

- **React có thể thành full-stack khi áp dụng framework:** Đây là một trong những tính năng mạnh mẽ đáng gờm của React khi so sánh với các đối thủ như Angular hay VueJS. React là một library cho phép bạn đặt các component lại với nhau, nhưng nó không quy định cách thực hiện routing và data fetching. Vậy nên để xây dựng toàn bộ ứng dụng với React, bạn nên dùng full-stack React framework như Next.js hoặc Remix.⁵

3.1.1.4 Nhược điểm

- **React có tốc độ ra mắt những phiên bản mới khá nhanh:** Đây vừa là một ưu điểm nhưng lại là một nhược điểm gây tranh cãi khi sử dụng React. React không chỉ là library được phát triển nhanh chóng mà còn thay đổi nhanh chóng, các phiên bản cũ mau chóng trở nên lỗi thời. Điều này rõ ràng buộc developer phải tốn thêm thời gian để học và cập nhập code liên tục.
- **Tính linh hoạt và thiếu quy ước:** Vì React nhỏ gọn để có thể áp dụng một cách linh hoạt, nhưng đây cũng là một nhược điểm. Khi các library, ngôn ngữ hoặc framework khác như Angular có các tiêu chuẩn quốc tế để các developer tuân theo như style hay pattern áp dụng trong trường hợp cụ thể. Vậy nên khi các developer thay đổi dự án thì họ có thể nhanh chóng dựa vào các nguyên tắc để làm quen dự án mới nhanh hơn. Nhưng khi sử dụng React sẽ không dễ để dự đoán style hay tiêu chuẩn của dự án, khiến các thành viên mới mất nhiều thời gian hơn để làm quen dự án.
- **Không đầy đủ tính năng:** Mặc dù React là Javascript library phong phú với đầy đủ các tính năng để tạo các application lớn, nhưng chắc chắn không đầy đủ tính năng như Angular (một Javascript framework phổ biến). Trong kiến trúc MVC (Model View Controller), React chỉ xử lý phần View, đối với phần Model và Controller, bạn cần tích hợp thêm library hoặc plugin.

3.1.2 Typescript

3.1.2.1 Khái niệm

TypeScript là ngôn ngữ lập trình mạnh mẽ được xây dựng trên JavaScript, là công cụ lập trình tốt hơn Javascript⁶.

- **Hơn cả Javascript:** TypeScript thêm cú pháp bổ sung vào JavaScript để hỗ trợ tích hợp chặt chẽ hơn với trình soạn thảo của bạn. Phát hiện lỗi sớm hơn trong trình soạn thảo của bạn.
- **Có thể thực thi ở bất kì nơi nào mà JavaScript thực thi được:** Code TypeScript chuyển đổi thành code JavaScript, chạy ở mọi nơi JavaScript thực thi: Trong trình duyệt, trên Node.js hoặc Deno và trong ứng dụng của bạn.

⁵ react.dev. 2023. *React*. Truy cập từ: <https://react.dev/>.

⁶ www.typescriptlang.org. 2023. *TypeScript: JavaScript With Syntax For Types ...* Truy cập từ: <https://www.typescriptlang.org/>.

3.1.2.2 Tại sao lại chọn Typescript?

Typescript là ngôn ngữ lập trình được yêu thích và công nghệ được áp dụng nhiều nhất. Là ngôn ngữ lập trình được yêu thích thứ 2 trong cuộc khảo sát Stack Overflow 2020 về ngôn ngữ được yêu thích nhất.



Hình 5: Biểu đồ phàn trám ngôn ngữ được yêu thích của Stackoverflow 2020

Trong State of JS⁷ 2020 TypeScript đã có 78% số người sử dụng trả lời, trong đó 93% cho biết họ sẽ sử dụng lại nó.⁸

- TypeScript cho phép chỉ định các loại data: JavaScript không quy ước các loại data được truyền vào, do vậy làm cho developer phải mất thời gian xem tài liệu và dự đoán loại data. Nhưng với TypeScript, cho phép chỉ định các loại data được truyền vào trong code và có khả năng báo lỗi khi các loại data không khớp.
- Đối với ứng dụng nhỏ hoặc trang web đơn giản, TypeScript có thể sẽ hơi cồng kềnh. Nhưng đối với dự án lớn, việc phải đảm bảo không có lỗi trong code được xử lý bằng thủ công hoặc viết automation test khá mất thời gian. Ngoài ra dự án có đến hàng trăm hay hàng ngàn file, việc thay đổi một vài đoạn code nhỏ có thể gây ra sự cố lên hàng chục file khác. Vậy nên việc xác thực kết nối giữa mọi phần trong toàn bộ source code có thể tiêu tốn rất nhiều thời gian. Những ngôn ngữ kiểm tra được kiểu của data như TypeScript có thể tự động xử lý việc đó và cung cấp phản hồi tức thì trong quá trình code⁹.

⁷ 2020.stateofjs.com. 2023. *State of JS 2020: JavaScript Flavors*. Truy cập từ: <https://2020.stateofjs.com/en-US/technologies/javascript-flavors/>.

⁸ www.typescriptlang.org. 2023. *TypeScript: JavaScript With Syntax For Types..* Truy cập từ: <https://www.typescriptlang.org/>.

⁹ www.typescriptlang.org. 2023. *TypeScript: Why does TypeScript exist?.* Truy cập từ: <https://www.typescriptlang.org/why-create-typescript>.

3.1.2.3 Ưu điểm

- **Static typing (kiểu gõ tĩnh):** Data vẫn giữ nguyên loại đã được khai báo trước, ví dụ nếu data đã được khai báo là số thì data sẽ mãi lưu trữ giá trị là số.
- **Lỗi được phát hiện sớm:** Các nhà nghiên cứu nhận thấy rằng TypeScript phát hiện ra 15% lỗi phổ biến ở giai đoạn biên dịch. Tuy còn cách 100% khá xa nhưng con số đó đủ để tiết kiệm thời gian cho developer và cho phép họ tập trung vào việc sửa các lỗi trong logic. Việc thông qua mã code vào trình biên dịch cũng làm giảm khối lượng hoạt động test và đảm bảo chất lượng.¹⁰
- **Hỗ trợ OOP:** TypeScript hỗ trợ các khái niệm từ lập trình hướng đối tượng (OOP - Object Oriented Programming) dựa trên class, như là interfaces, tính kế thừa, tính đa hình, và nhiều thứ khác.

3.1.2.4 Nhược điểm

- **Thêm bước bô sung - transpiling (dịch mã):** Vì trình duyệt không thể hiểu TypeScript, nên bạn cần dịch nó sang JavaScript trước khi chạy. Tuy nhiên, quá trình này được tự động hóa cao và không cần nhiều thời gian. Nhìn chung nhược điểm này cũng có chút lợi ích khi ở quá trình biên dịch, trình biên dịch sẽ phát hiện các sai sót và báo lỗi cho developer biết.
- **Bô sung thêm code:** TypeScript giới thiệu static typing (kiểu gõ tĩnh) như được giới thiệu ở trên, có nghĩa là các developer dễ dàng debug được lỗi nhưng cũng có nghĩa TypeScript sẽ làm tăng thời gian code khi làm việc trên các dự án nhỏ hoặc ở giai đoạn phát triển ban đầu. Nhưng điều quan trọng là nhờ đó TypeScript giúp giảm thời gian phát triển trong tương lai vì dễ debug, lỗi cũng được phát hiện sớm.
- **TypeScript đôi khi cũng khó dùng:** TypeScript đôi khi thử thách developer với những data có kiểu phức tạp hoặc kiểu data không được rõ ràng ở vài trường hợp. Trong trường hợp đó, developer đánh đổi giữa bỏ qua type declaration (định nghĩa kiểu) để kịp deadline hoặc mất thời gian tìm hiểu¹¹. Với những dự án gấp gáp việc bỏ qua type nhiều sẽ làm giảm giá trị của TypeScript, vậy nên developer nên cân nhắc về thời gian và giá trị nhận được khi sử dụng TypeScript.

3.1.3 Material UI

3.1.3.1 Khái niệm

Material UI (viết tắt là MUI) là một thư viện có chứa các React Component và được tích hợp thêm cả Google's Material Design. Material UI đẹp về mặt thiết kế do được tạo ra bởi Material Design. Material UI có một bộ tùy chọn tùy chỉnh giúp bạn dễ dàng triển khai hệ thống thiết kế tùy chỉnh của riêng mình.

3.1.3.2 Vì sao chọn Material UI?

Các doanh nghiệp nổi tiếng như Spotify, Amazon, NASA, Netflix, ... tin tưởng MUI sẽ mang lại trải nghiệm vượt trội cho cả developer và người dùng.

Ngoài ra về số lượng Starred (người yêu thích) trên Github thì Material UI vượt trội hơn cả nghìn sao so với đối thủ cùng lĩnh vực là Ant Design. Với Material UI dẫn đầu trong số các

¹⁰ www.altexsoft.com. 2023. *Pros and Cons of TypeScript: When and Why It's Better than Plain JS | AltexSoft*. Truy cập từ: <https://www.altexsoft.com/blog/typescript-pros-and-cons/>.

¹¹ itjet.io. 2023. *What is TypeScript? Benefits of TypeScript and Disadvantages | itjet*. Truy cập từ: <https://itjet.io/blog/what-is-typescript>.

thư viện UI Component xét đến thời điểm hiện tại là ngày 24/10/2023 với số lượng 89.4 nghìn sao từ người dùng trên Github¹².

3.1.3.3 Ưu điểm

- **Hơn 40 block component được build sẵn:** Không chỉ có số lượng block component build sẵn đồ sộ không thua gì các đối thủ, MUI thiết kế tỉ mỉ từng chi tiết của từng component, với mọi component của MUI đều đáp ứng các tiêu chuẩn cao nhất về hình thức và chức năng.
- **Xây dựng hệ thống thiết kế giao diện như bạn muốn:** Bạn có thể bắt đầu nhanh chóng với Material Design hoặc sử dụng tính năng tạo chủ đề nâng cao để dễ dàng điều chỉnh các component theo nhu cầu của bạn. Khi bạn chọn theme tùy chỉnh, MUI cung cấp theme chung hoặc riêng từng component để bạn có thể tùy chỉnh ở một số chỗ hoặc toàn bộ dự án. Với việc có thể điều chỉnh theme chung, đây là ưu điểm tuyệt đối mà MUI đã vượt trội hơn hẳn các đối thủ còn lại.
- **Nhanh chóng thêm và chỉnh sửa bất kỳ style nào bằng CSS Utility:** Các CSS Utility cho phép bạn code nhanh hơn và mang lại trải nghiệm mượt mà cho developer khi tạo kiểu cho bất kỳ thành phần nào. Bạn có thể style ngay tại component đó, không cần phải tách thành 2 file style và component như thông thường. Bạn có thể định nghĩa responsive style ngay ở trong props của component sao cho phù hợp với theme breakpoints.
- **Hỗ trợ tốt quy trình thiết kế của dự án:** Bộ công cụ Thiết kế chứa nhiều thành phần MUI với các trạng thái, biến thể, màu sắc, kiểu chữ và biểu tượng. MUI thường xuyên cập nhật nó để đồng bộ hóa với bản phát hành cập nhật mới nhất¹³.

3.1.3.4 Nhược điểm

- **Khó khăn với Documentation:** Developer có thể gặp khó khăn khi cập nhập một số component nhất định trong MUI và đề xuất rằng documentation nên cung cấp nhiều thông tin hơn về các thông tin phức tạp này. Một số người mới cũng mất thời gian để đọc documentation của MUI hơn so với Ant Design vì nó phức tạp và khá dài.
- **Mất thời gian để tùy chỉnh:** Một số Developer đề cập rằng các component lồng nhau trong MUI có thể gây khó khăn trong việc tùy chỉnh một số component nhất định như input. Hoặc một số component cần phải tùy chỉnh đúng theo yêu cầu của design thì MUI lại không cung cấp đủ tài liệu để điều chỉnh component.
- **Hay bị hết hạn:** Document hay bị hết hạn mặc dù giao diện vẫn được cài tiền định kỳ.

3.1.4 Zustand

3.1.4.1 Khái niệm

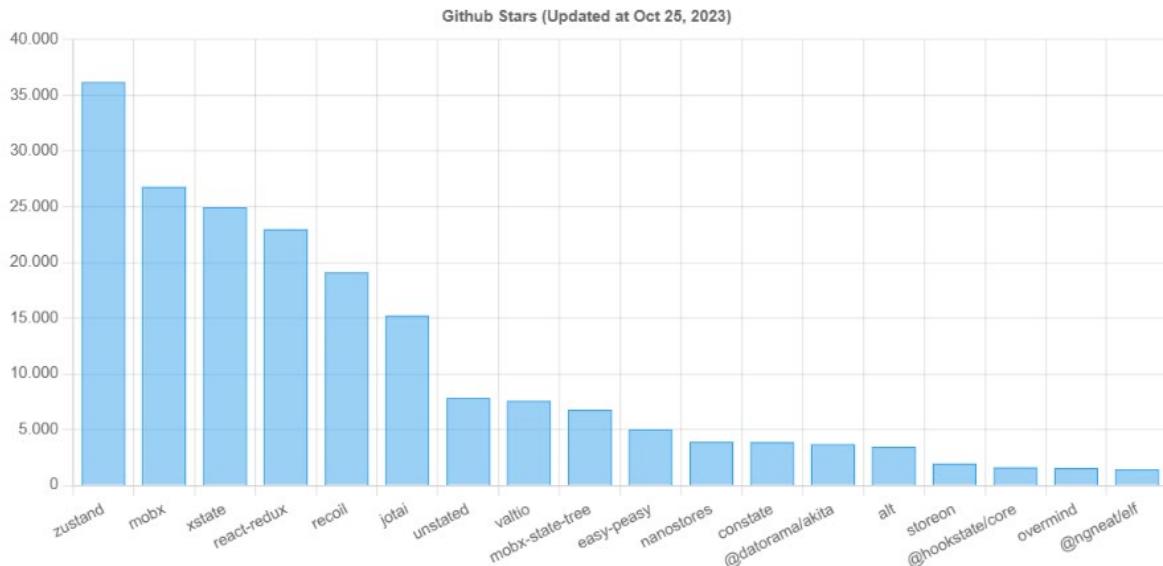
Zustand là một state-management nhỏ, nhanh và có thể mở rộng bằng cách sử dụng các nguyên tắc thông lượng đơn giản hóa. Nó được thiết kế để hoạt động hiệu quả và dễ sử dụng.

Zustand được xây dựng dựa trên React's built-in Context API. Nó làm cho hook trở thành phương tiện chính để sử dụng state và bạn không cần phải bọc ứng dụng của mình trong các context providers.

¹² [github.com. 2023. *mui/material-ui: MUI Core: Ready-to-use foundational React components, free forever. It includes Material UI, which implements Google's Material Design ..*](https://github.com/mui/material-ui) Truy cập từ: <https://github.com/mui/material-ui>.

¹³ [mui.com. 2023. *Material UI: React components based on Material Design.*](https://mui.com/) Truy cập từ: <https://mui.com/material-ui/>.

3.1.4.2 Tại sao chọn Zustand?



Hình 6: Thống kê Github Stars cho các state management

So với react-redux, Zustand đã cách một khoảng khá xa với 35k stars từ Github Stars, đây là minh chứng cho việc mọi developer yêu mến sử dụng Zustand hơn các state-management khác. Số liệu được cập nhập vào ngày 25/10/2023 theo thời gian thực.

Việc không phải bọc source code ứng dụng trong các context providers khiến cho dự án đỡ bị render nhiều lần, làm cho website trở nên nhẹ hơn, và đây cũng là một trong số lý do tại sao tác giả sử dụng Zustand thay vì Redux.

3.1.4.3 Ưu điểm

- Gọn, nhẹ và hiệu quả:** Zustand đã trở nên phổ biến trong giới React developer như một giải pháp gọn nhẹ và hiệu quả của state-management trong ứng dụng của họ. Zustand cho phép các developer tạo các store chứa data và có thể được các component trong ứng dụng truy cập và cập nhật. Zustand là một thư viện nhỏ tập trung vào hiệu suất, khiến nó phù hợp với các dự án nhỏ hơn.
- Thiết lập tối thiểu:** Zustand yêu cầu ít mã soạn sẵn hơn so với Redux, giúp thiết lập nhanh chóng và dễ dàng.
- Tích hợp dễ dàng:** Zustand có thể được sử dụng cùng với các state management khác như Redux hoặc MobX.

3.1.4.4 Nhược điểm

- Hệ sinh thái hạn chế:** Hệ sinh thái của Zustand nhỏ hơn so với Redux, nghĩa là có ít công cụ và thư viện của bên thứ ba hơn.
- Không phù hợp với các ứng dụng phức tạp:** Tính đơn giản của Zustand có thể hạn chế tính phù hợp của nó đối với các ứng dụng lớn và phức tạp. Nhược điểm này không ảnh hưởng lớn trong dự án hiện tại của tác giả vì dự án hiện tại đang không quá phức tạp, nên tác giả không yêu cầu Zustand phải có đầy đủ tính năng như React-redux.

3.2 Database sử dụng PostgreSQL

3.2.1 Giới thiệu về PostgreSQL

PostgreSQL là một hệ quản trị cơ sở dữ liệu mã nguồn mở phổ biến hiện nay dễ sử dụng và dễ dàng lưu trữ nhiều loại dữ liệu trong đó có dạng array là được đón nhận nhiều bởi người dùng. Được nhen nhóm phát triển rất sớm, từ những năm 1986 dự án Postgres lần đầu được

nghiên cứu tại Khoa Khoa học Máy tính Berkeley, Đại học California. Mãi đến năm 1996 POSTGRES mới đổi tên thành PostgreSQL. Và mục tiêu của dự án POSTGRES là thêm các tính năng tối thiểu cần thiết để hỗ trợ nhiều loại dữ liệu.

Kể từ đó, Nhóm Phát triển Toàn cầu PostgreSQL đã được hình thành, cộng đồng và các cộng tác viên tận tâm tiếp tục phát hành dự án cơ sở dữ liệu nguồn mở và miễn phí. Ban đầu, PostgreSQL được thiết kế để chạy hiệu quả trên nền tảng giống UNIX. Qua thời gian phát triển cho tới ngày nay thì PostgreSQL đã có thể chạy trên nhiều nền tảng khác nhau như Windows, macOS và Solaris.¹⁴

3.2.2 Ưu điểm

- **Mã nguồn mở và miễn phí:** PostgreSQL được phát triển dựa trên giấy phép mã nguồn mở, giúp giảm chi phí sở hữu và triển khai cơ sở dữ liệu.
- **Tiêu chuẩn tuân thủ ANSI-SQL:** PostgreSQL tuân thủ các tiêu chuẩn SQL ở các phiên bản của mình, giúp dễ dàng di chuyển giữa các hệ thống quản lý cơ sở dữ liệu (RDBMS) khác nhau.
- **Hỗ trợ cho nhiều kiểu dữ liệu:** PostgreSQL hỗ trợ nhiều loại dữ liệu, bao gồm cả các kiểu dữ liệu phức tạp như JSON, ARRAY, Geometry và XML.
- **Bảo mật mạnh mẽ:** PostgreSQL cung cấp các tính năng bảo mật như phân quyền người dùng, SSL/TLS, mã hóa dữ liệu và kiểm tra tính toàn vẹn.
- **Cộng đồng rộng lớn:** PostgreSQL được hỗ trợ bởi một cộng đồng nhiệt tình và đầy kinh nghiệm. Vì vậy khi sử dụng bạn sẽ được tận hưởng hỗ trợ bởi một cộng đồng lớn với nhiều người giàu kinh nghiệm.
- **Đối tượng hóa:** PostgreSQL không chỉ là một hệ quản trị cơ sở dữ liệu quan hệ, mà nó còn đối tượng hóa dữ liệu. Giúp linh hoạt trong việc lưu trữ dữ liệu.

3.2.3 Nhược điểm

- **Khả năng mở rộng ngang có hạn:** Mặc dù PostgreSQL có các tính năng mở rộng, nhưng trong một số trường hợp, khả năng mở rộng ngang của nó vẫn có hạn so với một số hệ thống khác.
- **Hiệu suất không nhất quán:** Trong một số tình huống, hiệu suất của PostgreSQL có thể không nhất quán so với một số hệ thống quản lý cơ sở dữ liệu khác, đặc biệt là khi xử lý các tác vụ có tính toán nặng.
- **Khó khăn trong việc quản lý và cấu hình:** PostgreSQL có thể đòi hỏi kiến thức chuyên sâu và kinh nghiệm về quản lý và cấu hình để triển khai và duy trì hiệu quả.¹⁵

3.3 Công nghệ ở Back-end

3.3.1 Vai trò của back-end trong ứng dụng

- **Xử lý logic nghiệp vụ:** Phía back-end đảm nhận việc xử lý các tác vụ liên quan đến logic nghiệp vụ của ứng dụng, bao gồm xác thực và quản lý người dùng, xử lý các yêu cầu liên quan đến dữ liệu, thực hiện các tính toán phức tạp và tương tác với các hệ thống khác và với cơ sở dữ liệu.
- **Quản lý dữ liệu:** Backend thường được sử dụng để lưu trữ và truy xuất dữ liệu từ các cơ sở dữ liệu, bao gồm việc thêm, sửa, xóa và truy vấn dữ liệu. Điều này cho phép ứng dụng lưu trữ và truy cập thông tin người dùng, sản phẩm và các dữ liệu khác.

¹⁴ postgresqltutorial.com. 2024. What is PostgreSQL?. Truy cập từ:

<https://www.postgresqltutorial.com/postgresql-getting-started/what-is-postgresql/>

¹⁵ viblo.asia. 2024. Kiến thức cơ bản về các hệ quản trị cơ sở dữ liệu quan hệ. Truy cập từ:

<https://viblo.asia/p/kien-thuc-co-ban-vecac-he-quan-tri-co-so-du-dieu-quan-he-aRBvXWMbkWE>

- **Cung cấp API:** Phía backend cung cấp các API (Application Programming Interface) cho phía client (frontend) của ứng dụng. Các API này cho phép phía client tương tác với các chức năng và dữ liệu của ứng dụng.
- **Bảo mật và xử lý lỗi:** Phía backend đảm bảo an toàn và bảo mật cho hệ thống ứng dụng bằng cách xử lý các lỗi và kiểm soát truy cập của người dùng đến các tài nguyên của hệ thống.
- **Tối ưu hóa hiệu suất:** Backend cũng đóng vai trò quan trọng trong việc tối ưu hóa hiệu suất của hệ thống, bao gồm tăng tốc độ phản hồi, tối ưu hóa tốc độ xử lý và tối ưu hóa cơ sở dữ liệu. Điều này giúp tăng trải nghiệm của người dùng và đảm bảo rằng ứng dụng hoạt động tốt.

3.3.2 Ngôn ngữ lập trình java

3.3.2.1 Giới thiệu

Nhắc đến ngôn ngữ lập trình backend thì không thể không kể đến Java. Đây là ngôn ngữ lập trình đa năng, đa nền tảng, hướng đối tượng (object-oriented) và có tính đồng thời (concurrent) được phát triển bởi Oracle. Hầu hết các kĩ sư backend đều ưu tiên sử dụng Java bởi ngôn ngữ này có thể biên dịch (compile) mã lệnh dưới dạng mã byte Java (Java bytecode) để chạy trên máy ảo Java Virtual Machine (JVM). Nhờ đó, bất kỳ thiết bị nào có thể cài đặt JVM thì có thể chạy được các ứng dụng viết bằng Java.

Java thích hợp cho cả hệ thống quy mô lớn và nhỏ. Đối với các dự án quy mô lớn, Java cung cấp các framework như Spring giúp quản lý và tổ chức mã nguồn dễ dàng. Đồng thời, với ứng dụng nhỏ, Java có thể được triển khai một cách nhẹ nhàng mà không tốn quá nhiều tài nguyên.

3.3.2.2 Ưu điểm

- **Đa nhiệm và linh hoạt:** Java hỗ trợ đa nhiệm và quản lý bộ nhớ tự động, giúp tăng cường hiệu suất và linh hoạt trong quản lý tài nguyên.
- **Độ bảo mật cao:** Java có cơ sở bảo mật mạnh mẽ với kiến trúc sandbox giúp ngăn chặn việc thực thi mã độc hại.
- **Chạy trên môi trường ảo Java (JVM):** Ứng dụng Java có thể chạy trên mọi nền tảng hỗ trợ JVM mà không cần sửa đổi.
- **Khả năng mở rộng và tích hợp dễ dàng:** Java có nhiều thư viện và framework mạnh mẽ giúp dễ dàng tích hợp với các công nghệ khác.
- **Xử lý đồng thời (Concurrency) tốt:** Java cung cấp các công cụ mạnh mẽ cho xử lý đồng thời, giúp xử lý nhiều yêu cầu đồng thời mà không gặp vấn đề.

3.3.2.3 Nhược điểm

- **Tốc độ khởi chạy chậm:** Java thường có thời gian khởi chạy lâu hơn so với một số ngôn ngữ khác.
- **Tài nguyên tiêu tốn nhiều hơn:** Ứng dụng Java có thể tiêu tốn nhiều tài nguyên hệ thống hơn so với một số ngôn ngữ khác.
- **Phản mềm khó duyệt:** JVM có thể làm tăng kích thước của ứng dụng, làm cho quá trình duyệt và tải trở nên chậm chạp hơn.¹⁶

¹⁶ javatpoint.com. 2023. *Advantages and disadvantages of Java* | TopDev. Truy cập từ: <https://www.javatpoint.com/advantages-and-disadvantages-of-java>

3.3.3 SpringBoot

3.3.3.1 Giới thiệu

Spring Boot là một framework của dự án Spring giúp việc xây dựng ứng dụng Java dễ dàng và nhanh chóng hơn. Nó cung cấp cấu hình mặc định để giảm thiểu việc cấu hình và tăng tốc quá trình phát triển. Spring Boot mạnh mẽ, chủ yếu sử dụng trong các dự án lớn và phức tạp hoặc các dự án trung bình. Một trong những lý do tác giả chọn Spring Boot là vì nó hỗ trợ cho mô hình MVC mà tác giả đã chọn.

3.3.3.2 Ưu Điểm

- **Tích hợp dễ dàng:** Spring Boot giúp tích hợp các module và thư viện của Spring một cách thuận lợi.
- **Tự cấu hình (Opinionated Defaults):** Cung cấp các giá trị mặc định cho cấu hình, giảm thiểu việc cấu hình thủ công.
- **Tích hợp embedded server:** Hỗ trợ các máy chủ như Tomcat, Jetty, và Undertow như là phần của ứng dụng.
- **Spring ecosystem:** Sử dụng được tất cả các tính năng và thư viện của Spring, giúp phát triển ứng dụng mạnh mẽ.
- **Dễ dàng kiểm soát quy mô ứng dụng:** Cho phép xây dựng ứng dụng nhỏ gọn hoặc mở rộng cho các ứng dụng lớn.

3.3.3.3 Nhược Điểm

- **Quá mạnh mẽ đối với dự án nhỏ:** Đối với các ứng dụng nhỏ, sự mạnh mẽ của Spring Boot có thể làm tăng kích thước của ứng dụng.
- **Khó khăn với người mới:** Có thể làm khó khăn đối với những người mới sử dụng nếu họ không quen với cách tự động cấu hình của Spring Boot.¹⁷

3.3.4 Spring Security

3.3.4.1 Giới thiệu

Spring Security là một module của dự án Spring cung cấp cơ sở hạ tầng bảo mật mạnh mẽ cho ứng dụng Java. Nó giúp xác định và kiểm soát quyền truy cập vào ứng dụng, bảo vệ chống lại các mối đe dọa bảo mật như tấn công CSRF, XSS, và hỗ trợ xác thực và ủy quyền.

3.3.4.2 Ưu Điểm

- **Phong phú và linh hoạt:** Hỗ trợ nhiều phương thức xác thực khác nhau như xác thực form, xác thực JWT, xác thực OAuth.
- **Tích hợp Spring Boot:** Spring Security tích hợp mạch lạc với Spring Boot, giúp tối ưu việc cấu hình và sử dụng.
- **Hỗ trợ đa nền tảng:** Hỗ trợ các kiểu xác thực phổ biến trên nền tảng web, di động và dịch vụ web.
- **Tính mở-đun và tính mở rộng:** Có thể mở rộng để đáp ứng nhu cầu bảo mật cụ thể của ứng dụng.

3.3.4.3 Nhược Điểm

- **Khó sử dụng cho người mới:** Có một số cấu hình và khái niệm phức tạp, đặc biệt đối với người mới sử dụng.

¹⁷ bambooagile.eu/. 2021. Pros and Cons of Using Spring Boot| bambooagile. Truy cập từ: <https://bambooagile.eu/insights/pros-and-cons-of-using-spring-boot/>

- **Phức tạp cho các tính năng đặc biệt:** Đôi khi, để triển khai những tính năng bảo mật đặc biệt có thể đòi hỏi kiến thức sâu rộng.

3.3.5 Hibernate

3.3.5.1 Giới thiệu

Hibernate là một framework mã nguồn mở trong hệ sinh thái của Java được thiết kế để giải quyết vấn đề ánh xạ đối tượng và cơ sở dữ liệu quan hệ. Nó cung cấp cách thuận tiện để tương tác với cơ sở dữ liệu thông qua các đối tượng Java, giảm sự phức tạp của việc viết mã SQL thủ công và hỗ trợ nhanh chóng phát triển ứng dụng.

3.3.5.2 Ưu Điểm

- **Ánh xạ đối tượng - dữ liệu:** Cho phép ánh xạ giữa các đối tượng Java và cơ sở dữ liệu quan hệ một cách tự động, giảm sự trùng lặp và tăng tính bảo trì.
- **Hỗ trợ quản lý đối tượng:** Hibernate quản lý vòng đời của đối tượng, giúp dễ dàng thực hiện các thao tác CRUD.
- **Làm giảm sự phụ thuộc vào cơ sở dữ liệu:** Ứng dụng không phụ thuộc trực tiếp vào cú pháp SQL, giảm sự phụ thuộc vào loại cơ sở dữ liệu cụ thể.
- **Hỗ trợ đa nền tảng:** Hibernate có thể hoạt động với nhiều loại cơ sở dữ liệu quan hệ khác nhau.
- **Hiệu suất tốt:** Cung cấp các cơ chế lười biếng (lazy loading) và tự động tối ưu hóa câu truy vấn.

3.3.5.3 Nhược điểm

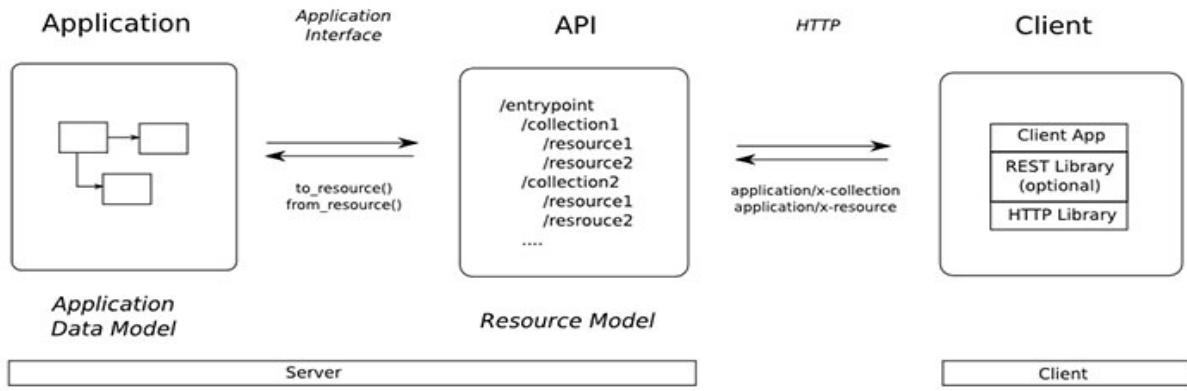
- **Khả năng tích hợp khó khăn:** Đôi khi, việc tích hợp Hibernate vào các dự án lớn và phức tạp có thể gặp khó khăn.
- **Hiện tượng N+1 Selects:** Đôi khi, hibernate có thể tạo ra quá nhiều câu truy vấn select (N+1 selects problem) nếu không được cấu hình đúng.
- **Khó kiểm soát câu truy vấn SQL:** Trong một số trường hợp, việc kiểm soát và tối ưu hóa câu truy vấn SQL có thể trở nên khó khăn.
- **Quá cứng nhắc:** Khiến việc truy cập nhiều bảng đôi khi gặp khó khăn.

3.4 RESTful API

3.4.1 Khái niệm

RESTful API là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource. Nó chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động...), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP.¹⁸

¹⁸ topdev.vn. 2019. *RESTful API là gì? Cách thiết kế RESTful API* | TopDev. Truy cập từ: <https://topdev.vn/blog/restful-api-la-gi/>.



Hình 7: Nguyên tắc hoạt động của Restful API

REST hoạt động chủ yếu dựa vào giao thức HTTP. Các hoạt động cơ bản nêu trên sẽ sử dụng những phương thức HTTP riêng.

- GET (SELECT): Trả về một Resource hoặc một danh sách Resource.
- POST (CREATE): Tạo mới một Resource.
- PUT (UPDATE): Cập nhật thông tin cho Resource.
- DELETE (DELETE): Xoá một Resource.

Những phương thức hay hoạt động này thường được gọi là CRUD tương ứng với Create, Read, Update, Delete – Tạo, Đọc, Sửa, Xóa.

API RESTful đã được chứng minh là có vai trò then chốt trong tích hợp phần mềm hiện đại do có nhiều ưu điểm. Một điểm bổ sung đáng chú ý là tính mô-đun mà chúng mang lại. Điều này có nghĩa là chúng có thể được cập nhật mà không ảnh hưởng đến hoạt động của hệ thống mà chúng được tích hợp vào. Hơn nữa, chúng thường nhẹ hơn về băng thông vì chúng sử dụng định dạng dữ liệu ít dài dòng hơn. Với những thay đổi công nghệ nhanh chóng mà chúng ta chứng kiến ngày nay, bắt buộc phải sử dụng các công cụ như API RESTful mang lại sự linh hoạt và khả năng thích ứng.

3.4.2 Ưu điểm

Tất cả những điểm mạnh của REST API tạo nên một hệ thống giao tiếp linh hoạt, bảo mật và dễ dàng mở rộng trong việc tương tác với các ứng dụng và dịch vụ khác nhau trên web. Dưới đây là một số chi tiết bổ sung về từng lợi ích:

- **Tính Tương tác:** REST APIs sử dụng các tiêu chuẩn web phổ biến, giúp REST APIs tương tác tốt với nhiều ứng dụng khác nhau trên internet. Với việc sử dụng cùng một ngôn ngữ, mã nguồn và kiến trúc, REST APIs dễ dàng kết nối và tích hợp với các hệ thống khác, từ các trang blog, mạng xã hội đến các nền tảng phát triển lớn.
- **Linh hoạt:** Sự linh hoạt của REST APIs cho phép chúng giao tiếp bằng bất kỳ định dạng dữ liệu nào, từ JSON đến XML hay các định dạng tùy chỉnh khác. Điều này giúp chúng dễ dàng tương thích và hoạt động với mọi ứng dụng trên web, bất kể định dạng, ngôn ngữ hay kiến trúc của ứng dụng đó.
- **Khả năng Mở rộng:** Với khả năng xử lý hàng nghìn, triệu và tỷ dữ liệu một cách đồng thời, REST APIs cho phép các ứng dụng mở rộng mà không làm ảnh hưởng đến hiệu suất. Điều này rất quan trọng khi cần xử lý các khối thông tin lớn mà hai trang web kết nối thông qua REST APIs vẫn hoạt động một cách mượt mà và hiệu quả.
- **Bảo mật:** REST APIs thường sử dụng các phương pháp xác thực thông qua mã thông báo truy cập (Access Token). Mã thông báo này là duy nhất và khó bị đánh cắp hoặc

phá vỡ. Hơn nữa, chúng có thể được kết hợp với các phương pháp xác thực khác để tăng cường bảo mật, giúp bảo vệ thông tin và dữ liệu của người dùng khỏi các mối đe dọa bảo mật trực tuyến.

- **Dễ sử dụng:** REST APIs được thiết kế để đơn giản và dễ tiếp cận hơn so với các loại API khác. Ngay cả khi bạn không có kinh nghiệm với phát triển web, REST API vẫn trở nên trực quan và dễ sử dụng, đặc biệt khi bạn hiểu về các ngôn ngữ như HTML, Python hoặc JavaScript.

3.4.3 Nhược điểm

- **Độ phức tạp trong thiết kế:** REST API dễ sử dụng nhưng việc thiết kế một REST API có thể phức tạp hơn so với các loại API khác, đặc biệt nếu bạn chưa quen với kiến trúc web. Để phát triển API, bạn cần hiểu về các nguyên tắc kiến trúc như RESTful để xây dựng một cấu trúc logic hợp lý và hiệu quả.
- **Kết nối qua Internet:** Tất cả thay đổi cho REST API của bạn phải được thực hiện trực tiếp trên mạng và chỉ trên mạng. Không thể chỉnh sửa API từ máy tính cá nhân mà không có kết nối internet, khác với việc chỉnh sửa các tập tin web HTML. Điều này có nghĩa là bạn phải luôn kết nối để thực hiện bất kỳ thay đổi nhỏ nào. REST API cần kết nối Internet để hoạt động, điều này có thể làm giảm tính ứng dụng của nó trong môi trường offline hoặc khi kết nối mạng không ổn định.
- **Hiệu suất và biến đổi linh hoạt:** REST APIs có thể có hiệu suất thấp hơn một chút so với các loại API khác, phụ thuộc vào máy chủ và tốc độ internet của họ. Đồng thời, cũng có thể REST API ít linh hoạt hơn so với các API khác do phải tuân theo những nguyên tắc kiến trúc cụ thể.¹⁹

3.5 Docker

Docker là một nền tảng mã nguồn mở giúp tự động hóa việc triển khai các ứng dụng bên trong các container phần mềm, cung cấp một lớp trừu tượng và tự động hóa ở mức hệ điều hành. Các container này chứa tất cả những gì cần thiết để chạy một ứng dụng: mã nguồn, runtime, thư viện hệ thống và cài đặt - giúp đảm bảo rằng ứng dụng sẽ chạy nhất quán trong mọi môi trường, từ phát triển đến sản xuất.

3.5.1 Lịch sử và Mục đích

Docker được phát triển bởi Docker, Inc. và ra mắt lần đầu vào năm 2013. Mục tiêu của Docker là tạo ra một công cụ giúp phát triển, vận chuyển và chạy các ứng dụng một cách dễ dàng và hiệu quả nhất có thể. Containerization, tức là việc đóng gói ứng dụng cùng với các phụ thuộc của nó vào một container, là trọng tâm chính của Docker, giúp giải quyết vấn đề "chạy trên máy của tôi" - khi một ứng dụng chạy tốt trên máy phát triển nhưng gặp lỗi khi chuyển sang máy sản xuất.

3.5.2 Ưu điểm của Docker

- **Khả năng di động:** Docker containers có thể chạy nhất quán trên bất kỳ môi trường nào, từ máy tính cá nhân đến các máy chủ trong trung tâm dữ liệu hoặc trên đám mây. Chỉ cần một vài dòng lệnh thì docker có thể dễ dàng tạo ra một môi trường ảo hóa cho dự án.
- **Hiệu suất:** Container nhẹ hơn và khởi động nhanh hơn so với máy ảo thông thường, vì chúng dùng một phần kernel của hệ điều hành tức chia sẻ chung kernel và chỉ sử dụng những thành phần cần thiết.

¹⁹ www.thepowermba.com. 2023. REST API: what it is, how it works, advantages and disadvantages. Truy cập từ: <https://www.thepowermba.com/en/blog/rest-api-what-it-is>.

- **Tính mở rộng và linh hoạt:** Docker hỗ trợ việc mở rộng ứng dụng dễ dàng thông qua việc sử dụng Docker Compose và Docker Swarm.
- **Cách ly tài nguyên:** Mỗi container hoạt động độc lập với các container khác, đảm bảo rằng các ứng dụng không ảnh hưởng lẫn nhau.

3.5.3 Các Thành Phần Chính của Docker

- **Docker Engine:** Là trái tim của Docker, nó bao gồm một daemon để quản lý các container (Docker daemon)
- **Docker Images:** Là những file bắt biến, chỉ đọc, chứa mọi thứ cần thiết để chạy một container. Docker Hub là một registry công cộng nơi các images có thể được lưu trữ và chia sẻ.
- **Docker Containers:** Là các phiên bản runtime của images. Containers là các thực thể chạy thực sự mà bạn có thể tạo, bắt đầu, dừng, di chuyển và xóa.
- **Docker Compose:** Một công cụ giúp định nghĩa và chạy các ứng dụng Docker đa-container, sử dụng file YAML để cấu hình các dịch vụ của ứng dụng.
- **Docker Swarm:** Một công cụ cho phép tạo một cluster của các Docker hosts, cung cấp khả năng mở rộng và quản lý dễ dàng hơn.

3.5.4 Nhược điểm

- **Hiệu suất giới hạn so với máy ảo (VM):** Dù Docker nhẹ hơn so với các máy ảo, nhưng trong một số trường hợp cụ thể, hiệu suất của các container Docker có thể không tối ưu như máy ảo, đặc biệt khi cần tài nguyên hệ thống cao hoặc khi ứng dụng yêu cầu sự cô lập tuyệt đối.
- **Bảo mật:** Các container chia sẻ kernel của hệ điều hành chủ, do đó, một lỗ hổng bảo mật trong kernel có thể ảnh hưởng đến tất cả các container. Mặc dù Docker đã cải thiện nhiều về bảo mật, nhưng vẫn còn nhiều mối lo ngại so với việc sử dụng máy ảo.

CHƯƠNG 4: PHÂN TÍCH THIẾT KẾ

4.1 Nhu cầu người dùng hệ thống

4.1.1 Actor

Hệ thống sẽ bao gồm 3 actor chính:

- **Người dùng:** Là người dùng có tài khoản, những người truy cập được vào hệ thống khi người đó đã đăng nhập vào hệ thống.
- **Khách (Guest):** Là những khách vãng lai, người truy cập vào hệ thống nhưng chưa đăng nhập vào hệ thống.
- **Quản trị viên (quản trị viên hệ thống):** Những người quản trị hệ thống, quản lý đơn hàng, duyệt đơn hàng.

4.1.2 Nhu cầu Quản trị viên

- Đăng nhập, đăng xuất.
- Quản lý được đơn hàng cho thuê: Xem và duyệt đơn hàng
- Quản lý được đơn hàng mua bán: Xem và duyệt đơn hàng
- Quản lý được voucher toàn sàn: Xem và thêm xóa sửa voucher toàn sàn

4.1.3 Nhu cầu người dùng

- Đăng nhập, đăng xuất hệ thống.
- Tìm tài liệu.
- Thấy được bài đăng các tài liệu đang đăng.
- Cập nhật thay đổi thông tin cá nhân tài khoản.
- Tạo bài đăng, xóa bài đăng.
- Quản lý tài liệu cho thuê.
- Quản lý tài liệu thuê.
- Quản lý tài liệu mua.
- Quản lý tài liệu bán.
- Quản lý voucher.
- Chức năng giỏ hàng.
- Viết review cho đơn hàng đã thuê.
- Chức năng đặt đơn hàng.

4.1.4 Nhu cầu khách vãng lai

- Đăng ký tài khoản.
- Tìm tài liệu.
- Thấy được bài đăng các tài liệu đang đăng.
- Thêm bài đăng vào giỏ hàng được nhưng muốn phát sinh đơn hàng thì phải có tài khoản.

4.2 Yêu cầu chức năng

4.2.1 Chức năng chung của khách và người dùng

- Xem danh sách đề xuất: Là xem danh sách các tài liệu được đề xuất bởi hệ thống.
- Tìm kiếm tài liệu: Khi bạn cần tài liệu nào, cung cấp từ khoá cho hệ thống hoặc thẻ loại tài liệu cho hệ thống, từ thông tin đó hệ thống sẽ cho ra danh sách những tài liệu hiện đang được đăng phù hợp với thông tin cung cấp.
- Xem thông tin chi tiết tài liệu: Xem được hình ảnh, giới thiệu về tài liệu, giá thuê, giá cọc, giá bán. Xem được review của tài liệu của những khách hàng trước.
- Chức năng đăng ký tài khoản.
- Xem giỏ hàng và thêm sản phẩm vào giỏ hàng

4.2.2 Chức năng riêng dành cho người dùng có tài khoản

- Chức năng đăng nhập, chức năng đăng xuất.
- Quản lý tài khoản: Chức năng này giúp người dùng xem hoặc thay đổi thông tin cá nhân người dùng trên hệ thống.
- Chức năng đánh giá tài liệu: Viết nhận xét và cho sao chấm điểm về những sản phẩm đã thuê, mướn thành công và đã trả tài liệu.
- Chức năng giỏ hàng: Xem giỏ hàng, thêm và xóa bài đăng thuê hoặc mua trong giỏ hàng.
- Chức năng chọn thuê: Người dùng chọn một sản phẩm trong giỏ hàng, sau đó tiến hành đặt thuê sản phẩm.
- Chức năng quản lý sách đang thuê: Xem được danh sách đang thuê. Chức năng xem chi tiết cung cấp thông tin cho người dùng về thời hạn cho thuê, giá cả, đã đến hạn trả sách hay chưa, phí phạt khi chậm trả sách, giá sách.
- Chức năng trả sách, nhận sách.
- Xem đơn hàng thuê/cho thuê theo trạng thái: người dùng có thể xem đơn hàng theo trạng thái đơn hàng. Người dùng có thể viết nhận xét cho đơn hàng đã thực hiện xong.
- Chức năng chọn mua: Người dùng chọn một sản phẩm trong giỏ hàng, sau đó tiến hành đặt mua sản phẩm.
- Chức năng quản lý sách đang mua: Xem được danh sách tài liệu đang mua. Chức năng xem chi tiết cung cấp thông tin cho người dùng về ngày mua, giá cả, được giảm giá bao nhiêu % hoặc bao nhiêu tiền.
- Xem đơn hàng mua bán theo trạng thái: người dùng có thể xem đơn hàng theo trạng thái đơn hàng, khi nhận được sách, người dùng có thể thay đổi trạng thái đơn hàng.
- Chức năng đăng bài: Người dùng cung cấp thông tin về tài liệu về hình ảnh, lời giới thiệu, giá cả, ... lên hệ thống để tạo bài đăng.
- Quản lý bài đăng: Xem được danh sách sách đang đăng của mình. Có thể xoá bài đăng. Có thể xem chi tiết bài đăng: Thông tin sách đã cho thuê như thời hạn, người thuê sách, thông tin sách, giá cả, ...
- Hệ thống thanh toán: Thanh toán tiền hành bằng nhiều hình thức online hoặc COD.
- Quản lý voucher của người bán: người bán có thể xem xóa sửa tạo voucher để áp dụng cho đơn hàng của các khách hàng mua sách của mình.

4.2.3 Các chức năng ở Quản trị viên

- Đăng nhập, đăng xuất.
- Quản lý đơn hàng mua bán: Xem danh sách đơn hàng, duyệt đơn hàng, thay đổi trạng thái đơn hàng.
- Quản lý đơn hàng thuê cho thuê: Xem danh sách đơn hàng, duyệt đơn hàng, thay đổi trạng thái đơn hàng.
- Quản lý voucher toàn sàn: Thêm xóa sửa voucher toàn sàn

4.3 Yêu cầu phi chức năng

4.3.1 Thiết kế responsive đa thiết bị

Thiết kế responsive trên laptop, ipad và mobile và hỗ trợ đa nền tảng. Thiết kế responsive trên các thiết bị sau dựa theo kích thước được cho sẵn bởi Bootstrap phiên bản 4.1.20 và trang 114 sách Responsive Web Design của Ethan Marcotte.

- Màn hình kích thước siêu lớn:
 - Chủ yếu các loại thiết bị như máy tính bàn PC và laptop có màn hình rộng.
 - Chiều dài trên 1200px.
- Màn hình kích thước lớn:
 - Máy tính bảng như iPad (1024×768) được đặt ở chế độ nằm ngang cũng như một số máy tính xách tay, netbook và máy tính để bàn với màn hình nhỏ.
 - Chiều dài từ 992px đến 1200px.
- Màn hình kích thước vừa:
 - Máy tính bảng như iPad (1024×768) được đặt ở chế độ nằm đứng.
 - Chiều dài từ 768px đến 992px.
- Màn hình kích thước nhỏ:
 - Máy tính bảng nhỏ hơn, như Amazon Kindle (600×800) và Barnes & Noble Nook (600×1024), được đặt ở chế độ nằm đứng.
 - Chiều dài từ 576px đến 768px
- Màn hình kích thước siêu nhỏ:
 - Điện thoại Iphone 4 và Iphone 5.
 - Chiều dài từ 320px đến 576px.

²⁰ getbootstrap.com. 2023. *Overview · Bootstrap*. Truy cập từ: <https://getbootstrap.com/docs/4.1/layout/overview/>.



Hình 8: Biểu đồ độ phủ sóng của trình duyệt trên toàn cầu (10/10/2023)

4.3.2 Thiết kế đa nền tảng

- Với bất kì nền tảng IOS, Android hay Window nào có thể mở được trình duyệt. Vì website sẽ được lập trình theo client-side-rendering, nên ở website tại máy người dùng sẽ yêu cầu mặc định như các website khác.
- Theo Browserslist²¹ yêu cầu mặc định của website hiện nay bao gồm:
 - Phiên bản trình duyệt được chọn có độ phủ sóng lớn hơn 0,5% theo thống kê sử dụng toàn cầu.
 - Là 2 phiên bản mới nhất cho mỗi trình duyệt, được thống kê ở bảng dưới.
 - Không bao gồm các trình duyệt đã chết: Các trình duyệt không hỗ trợ hoặc cập nhật chính thức trong 24 tháng. Hiện tại là IE 11, IE_Mob 11, BlackBerry 10, BlackBerry 7, Samsung 4, Opera Mobile 12.1 và tất cả các phiên bản của Baidu.
 - Bao gồm Firefox ESR: Là bản phát hành hỗ trợ mở rộng mới nhất của Firefox.

4.3.3 Hệ thống có tính bảo mật

- Website có chứng chỉ SSL, SSL (Secure Sockets Layer) là một tiêu chuẩn an ninh công nghệ toàn cầu. Đảm bảo an toàn dữ liệu và các giao dịch trên website thông qua việc mã hóa các thông tin nhạy cảm trong giao dịch trực tuyến.
- Website sử dụng HTTPS, giao thức của HTTPS được sử dụng để giao tiếp an toàn qua mạng máy tính và được sử dụng rộng rãi trên Internet. Thiết kế giao thức của https là để chống lại mọi sự tấn công nên bảo mật thông tin sẽ tốt hơn.
- Hệ thống yêu cầu người dùng và Quản trị viên tạo mật khẩu có độ phức tạp cao, cụ thể như sau:

²¹ browserslist. 2023. *Browserslist*. Truy cập từ: <https://browserslist/#q=defaults®ion=VN>.

- Sử dụng tối thiểu 8 ký tự, và tối đa 15 ký tự.
- Bao gồm ít nhất một chữ số, một chữ thường, một chữ in hoa và một ký tự đặc biệt.
- Không được mang ý nghĩa đi kèm như số điện thoại, ngày sinh.
- Không sử dụng các con số nổi tiếng, ví dụ như: 113, 115, 12345678
- Website được các quản trị viên sao lưu dữ liệu 6 tháng 1 lần. Sau này có thể thay đổi tùy thuộc độ lớn của dữ liệu.
- Website có trang chính sách bảo mật dành cho người dùng, người dùng phải đồng ý với chính sách bảo mật trước khi tạo tài khoản.

4.3.4 Hiệu suất

Hệ thống đảm bảo khả năng xử lý và truy cập dữ liệu nhanh chóng và hiệu quả với thời gian phản hồi tối đa 1 giây và đáp ứng truy cập của 100 người cùng lúc.

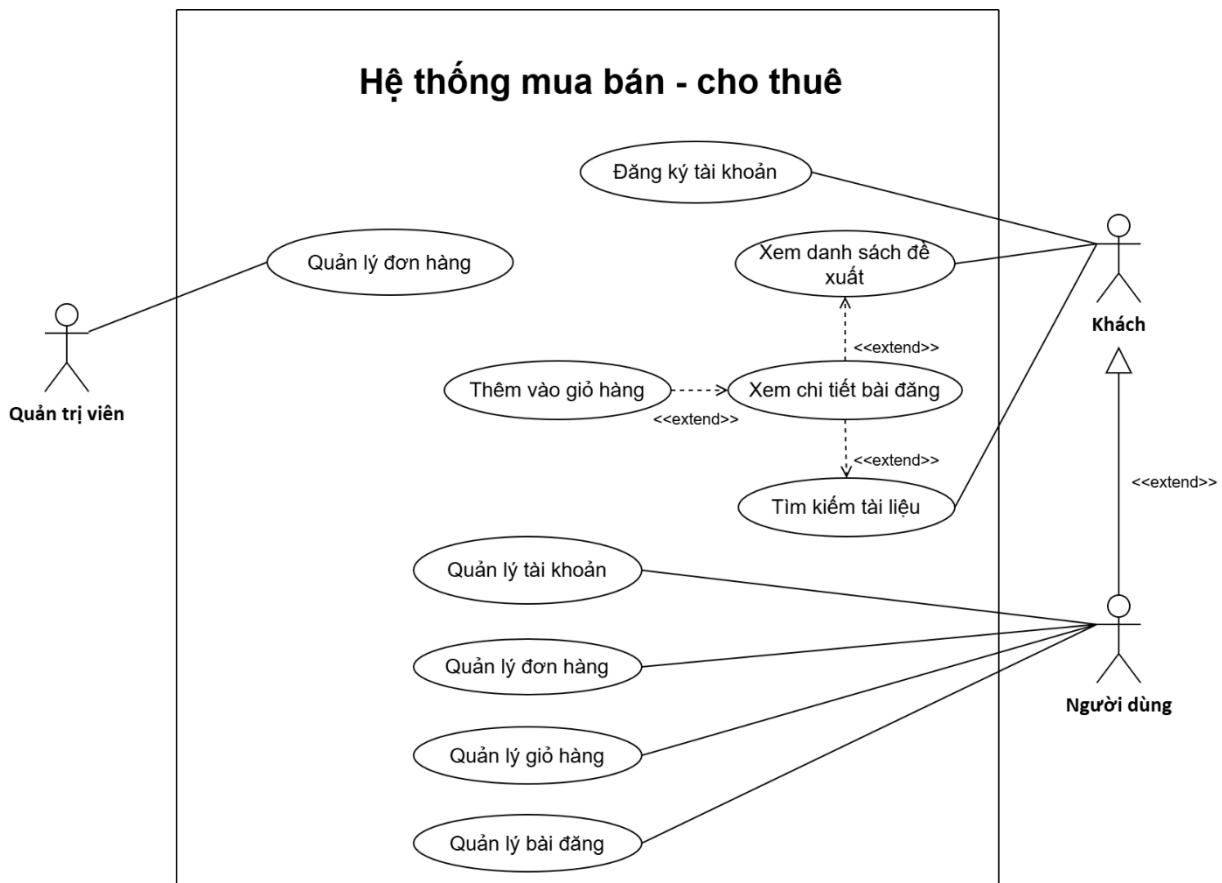
4.3.5 Khả năng bảo trì

Hệ thống thực hiện bảo trì vào ngày cuối tháng, thời gian bảo trì không vượt quá 8 tiếng.

4.3.6 Hệ thống có giao diện dễ nhìn, đồng đều, vừa mắt, dễ sử dụng

- Tuân thủ theo các quy tắc thiết kế cơ bản của Gestalt như sau:
 - Sự tương đồng (similarity)
 - Sự tiếp nối (continuation)
 - Sự gần bên (proximity)
 - Sự đóng kín (closure)
 - Chính-phụ (figure-ground)
 - Sự tối giản (prägnanz)
- Sử dụng phương pháp nghiên cứu phương thức thiết kế UX (UX Research Method):
 - Kiểm tra khả năng sử dụng (Usability Testing): Kiểm tra tính khả dụng để tạo ra một sản phẩm làm cho khách hàng mua ngay từ cái nhìn/ lần sử dụng đầu tiên.
 - Phỏng vấn (Interviews): Nhà nghiên cứu gặp gỡ trực tiếp người dùng để thảo luận sâu hơn về trải nghiệm của người dùng với những chức năng hiện có của website
 - Feedback khách hàng: Người dùng sau khi sử dụng xong một chức năng mới trên website có thể sẽ có biểu mẫu đánh giá xuất hiện để người dùng đánh giá về chức năng mới đó.
- Sử dụng hình ảnh trực quan sinh động với chất lượng cao và dung lượng nhẹ, người dùng chỉ được upload file tối đa 5MB để không bị vượt quá giới hạn lưu trữ của database.
- Dùng google analytics để phân tích Báo cáo thu nạp người dùng.

4.4 Use case diagram cho toàn hệ thống

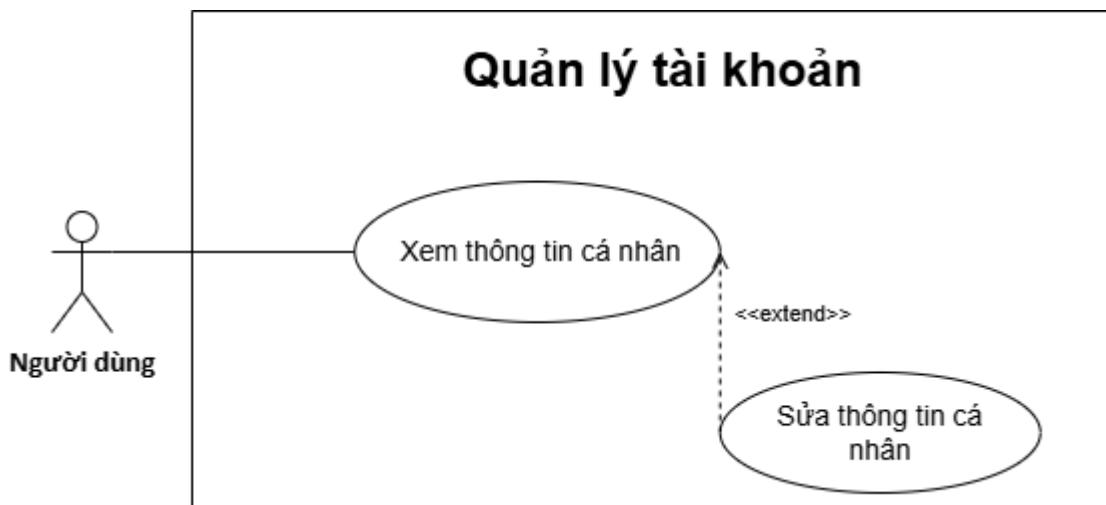


Hình 9: Use case diagram cho toàn bộ hệ thống

4.5 Đặc tả Use case

4.5.1 Use cases cho Người dùng

4.5.1.1 Người dùng Quản lý tài khoản



Hình 10: Module quản lý tài khoản

- Người dùng xem thông tin cá nhân

Use case name	Người dùng xem thông tin cá nhân
----------------------	----------------------------------

Actor	Người dùng
Description	Người dùng muốn xem thông tin cá nhân của mình
Preconditions	Người dùng đã đăng nhập thành công Người dùng đang ở trang chủ trang web
Postconditions	Người dùng xem thông tin cá nhân thành công Hệ thống hiển thị thông tin cá nhân
Normal Flow	<ol style="list-style-type: none"> 1. Người dùng bấm vào biểu tượng “Tài khoản” 2. Hệ thống hiển thị danh sách lựa chọn 3. Người dùng bấm vào cài đặt tài khoản 4. Hệ thống hiển thị thông tin cá nhân của Người dùng 5. Người dùng xem được thông tin cá nhân của mình
Alternative flow	<p>Alternative 1:</p> <ol style="list-style-type: none"> 1. Tại trang quản lý của người dùng, người dùng bấm cài đặt tài khoản ở menu bên trái 2. Hệ thống hiển thị thông tin cá nhân của Người dùng 3. Người dùng xem được thông tin cá nhân của mình
Exception	Không có

Bảng 1: Use case Người dùng xem thông tin cá nhân

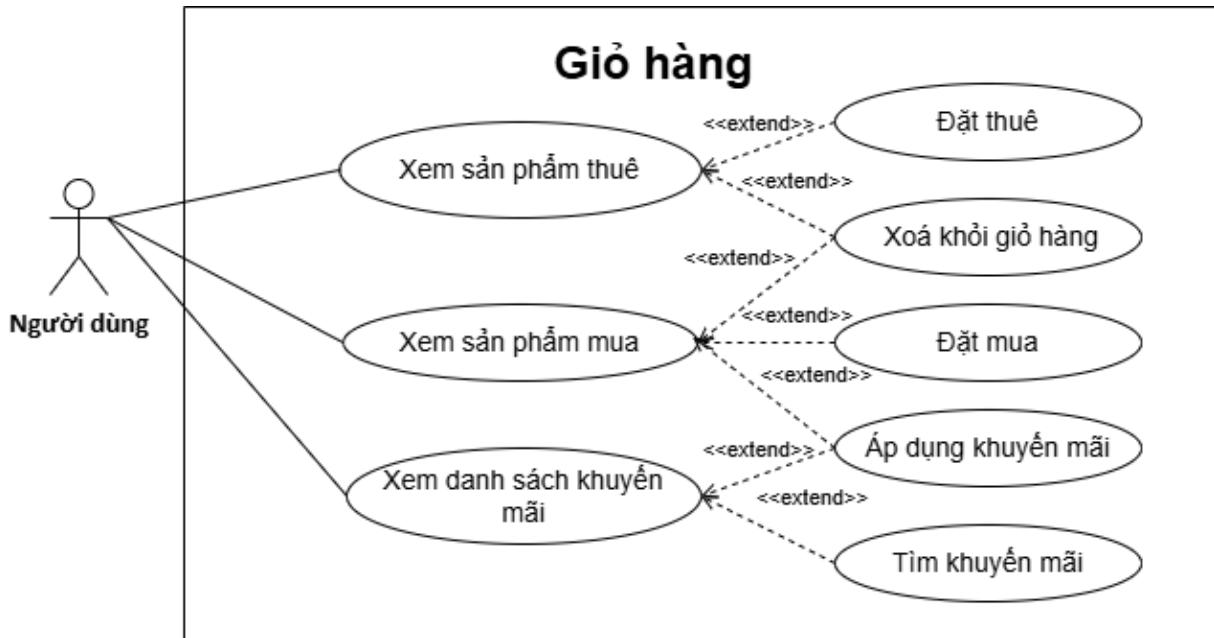
- **Người dùng sửa thông tin cá nhân**

Use case name	Người dùng sửa thông tin cá nhân
Actor	Người dùng
Description	Người dùng muốn sửa thông tin cá nhân của mình trên hệ thống
Preconditions	Người dùng đã đăng nhập thành công Người dùng đang ở trang thông tin cá nhân
Postconditions	Người dùng thay đổi thông tin cá nhân thành công Thông tin cá nhân của Người dùng đã được chỉnh sửa và cập nhật trong cơ sở dữ liệu
Normal Flow	<ol style="list-style-type: none"> 1. Hệ thống hiển thị thông tin cá nhân chứa các trường thông tin có thể chỉnh sửa 2. Người dùng sửa đổi thông tin cần thay đổi 3. Người dùng bấm nút “Xác nhận” để cập nhật thông tin cá nhân mới 4. Hệ thống xác thực thông tin chỉnh sửa của Người dùng, kiểm tra tính hợp lệ. 5. Hệ thống cập nhật thông tin cá nhân của Người dùng vào cơ sở dữ liệu. 6. Hệ thống hiển thị pop-up “Xác nhận thông tin thành công” 7. Hệ thống hiển thị thông tin cá nhân mới được chỉnh sửa

	thành công của Người dùng lên trang thông tin cá nhân
Alternative Flow	Không có
Exception	Không có

Bảng 2: Use case Người dùng sửa thông tin cá nhân

4.5.1.2 Người dùng Xem giỏ hàng



Hình 11: Module xem danh sách giỏ hàng

- **Xem sản phẩm thuê**

Use case name	Người dùng xem sản phẩm thuê
Actor	Người dùng
Description	Người dùng muốn xem sản phẩm thuê đã thêm vào giỏ hàng
Preconditions	Người dùng đang ở trang chủ website
Postconditions	Người dùng xem sản phẩm thuê trong giỏ hàng thành công Hệ thống hiển thị thông tin sản phẩm thuê trong giỏ hàng
Normal Flow	<ol style="list-style-type: none"> 1. Người dùng bấm vào icon giỏ hàng trên thanh menu 2. Hệ thống hiển thị trang giỏ hàng 3. Hệ thống hiển thị sản phẩm đã thêm vào giỏ hàng 4. Người dùng thấy sản phẩm đã thêm vào giỏ hàng
Exception	Exception tại bước 3: 3a. Không có sản phẩm thuê trong giỏ hàng 4a. Hệ thống hiển thị hình ảnh giỏ hàng trống khi không có sản phẩm trong giỏ hàng
Alternative Flow	Không có

Bảng 3: Use case Người dùng xem sản phẩm thuê

- Xem sản phẩm mua

Use case name	Người dùng xem sản phẩm mua
Actor	Người dùng
Description	Người dùng muốn xem sản phẩm mua đã thêm vào giỏ hàng
Preconditions	Người dùng đang ở trang chủ website
Postconditions	Người dùng xem sản phẩm mua trong giỏ hàng thành công Hệ thống hiển thị thông tin sản phẩm mua trong giỏ hàng
Normal Flow	<ol style="list-style-type: none"> 1. Người dùng bấm vào icon giỏ hàng trên thanh menu 2. Hệ thống hiển thị trang giỏ hàng 3. Hệ thống hiển thị sản phẩm đã thêm vào giỏ hàng 4. Người dùng thấy sản phẩm đã thêm vào giỏ hàng
Exception	Exception tại bước 3: 3a. Không có sản phẩm mua trong giỏ hàng 4a. Hệ thống hiển thị hình ảnh giỏ hàng trống khi không có sản phẩm trong giỏ hàng
Alternative Flow	Không có

Bảng 4: Use case Người dùng xem sản phẩm mua

- Xóa sản phẩm khỏi giỏ hàng

Use case name	Người dùng Xóa sản phẩm
Actor	Người dùng
Description	Người dùng muốn xóa sản phẩm khỏi giỏ hàng
Preconditions	Người dùng đang ở trang giỏ hàng Có sản phẩm trong giỏ hàng
Postconditions	Người dùng đã xóa thành công sản phẩm khỏi giỏ hàng Sản phẩm bị xoá thành công khỏi giỏ hàng
Normal Flow	<ol style="list-style-type: none"> 1. Người dùng chọn nút “thùng rác” để xóa sản phẩm đó khỏi giỏ hàng 2. Hệ thống xoá sản phẩm khỏi giỏ hàng của người dùng
Alternative Flows	Không có
Exception	Không có

Bảng 5: Use case Người dùng xóa sản phẩm

- Xem danh sách khuyến mãi

Use case name	Người dùng xem danh sách khuyến mãi
----------------------	-------------------------------------

Actor	Người dùng
Description	Người dùng muốn xem danh sách khuyến mãi
Preconditions	Người dùng đang ở trang giỏ hàng Người dùng có sản phẩm trong giỏ hàng
Postconditions	Người dùng xem danh sách khuyến mãi trong giỏ hàng thành công Hệ thống hiển thị modal danh sách khuyến mãi trên trang giỏ hàng
Normal Flow	<ol style="list-style-type: none"> 1. Hệ thống tính toán và chọn khuyến mãi giảm nhiều nhất cho người dùng ở ngay trang giỏ hàng. 2. Người dùng bấm vào “Chọn hoặc nhập mã khác” 3. Hệ thống hiển thị modal danh sách khuyến mãi 4. Hệ thống tự động sắp xếp danh sách khuyến mãi theo khuyến mãi giảm sâu nhất cho khách hàng trước, rồi đến khuyến mãi còn hạn sử dụng 5. Hệ thống tự động chọn cho người dùng khuyến mãi tốt nhất và khuyến mãi đó luôn xếp ở trên cùng
Exception	Exception tại bước 3: 3a. Không có khuyến mãi trong giỏ hàng 4a. Không có khuyến mãi phù hợp với đơn hàng, tất cả khuyến mãi đều không thỏa điều kiện
Alternative Flow	Không có

Bảng 6: Use case Người dùng xem danh sách khuyến mãi

- Áp dụng khuyến mãi

Use case name	Người dùng áp dụng khuyến mãi
Actor	Người dùng
Description	Người dùng muốn áp dụng khuyến mãi vào sản phẩm đang có trong giỏ hàng
Preconditions	Người dùng đang ở trang giỏ hàng Người dùng có sản phẩm trong giỏ hàng Sản phẩm thỏa điều kiện để áp dụng khuyến mãi Người dùng đang ở modal danh sách khuyến mãi
Postconditions	Người dùng áp dụng khuyến mãi vào giỏ hàng thành công Hệ thống hiển thị khuyến mãi trên trang giỏ hàng
Normal Flow	<ol style="list-style-type: none"> 1. Người dùng chọn khuyến mãi khác với khuyến mãi hiện tại hệ thống đã tự động chọn cho người dùng 2. Hệ thống tính lại giá tiền sau khi đơn hàng được áp dụng khuyến mãi, và hiển thị tổng tiền ở trang đơn hàng lại
Exception	Không có
Alternative Flow	Không có

Bảng 7: Use case Người dùng áp dụng khuyến mãi

- **Tìm khuyến mãi**

Use case name	Người dùng tìm khuyến mãi
Actor	Người dùng
Description	Người dùng muốn tìm khuyến mãi theo tên hoặc theo mã
Preconditions	Người dùng đang ở trang giỏ hàng Người dùng có sản phẩm trong giỏ hàng Người dùng đang ở modal danh sách khuyến mãi
Postconditions	Người dùng tìm khuyến mãi thành công Hệ thống hiển thị khuyến mãi tìm được trong modal danh sách khuyến mãi
Normal Flow	<ol style="list-style-type: none"> 1. Người dùng tìm khuyến mãi theo tên hoặc theo mã 2. Hệ thống hiển thị danh sách khuyến mãi khớp với từ khóa người dùng đã gõ, không phân biệt khuyến mãi còn hạn sử dụng hay không 3. Hệ thống sẽ tự động sắp xếp khuyến mãi theo khuyến mãi giảm giá tốt nhất đầu tiên
Exception	Exception 1: tại bước 2 2a. Nếu hệ thống không tìm thấy tài liệu của khách sẽ hiển thị hình ảnh “Không tìm thấy”
Alternative Flow	Không có

Bảng 8: Use case Người dùng tìm khuyến mãi

- **Đặt thuê**

Use case name	Người dùng đặt thuê
Actor	Người dùng
Description	Người dùng đặt thuê sản phẩm
Preconditions	Người dùng đang ở trang giỏ hàng Có sản phẩm người dùng muốn thuê trong giỏ hàng
Postconditions	Người dùng đặt thuê thành công cho đơn hàng Hệ thống cập nhật đơn hàng vào cơ sở dữ liệu
Normal Flow	<ol style="list-style-type: none"> 1. Tại trang giỏ hàng >> Người dùng bấm nút “Đặt thuê hàng” 2. Hệ thống hiển thị trang “Đặt hàng” với toàn bộ thông tin của đơn hàng gồm: thông tin người đặt thuê, thông tin đặt hàng và thông tin thanh toán 3. Người dùng nhập thông tin đặt thuê >> bấm xác nhận 4. Hệ thống cập nhật thông tin người đặt hàng. Hệ thống hiển thị thông báo nhắc “chọn thanh toán rồi tạo đơn hàng” 5. Người dùng kiểm tra thông tin đặt hàng >> người dùng chọn phương thức thanh toán vnpay

	<p>6. Hệ thống loại bỏ hidden của nút tạo đơn hàng</p> <p>7. Người dùng bấm “tạo đơn hàng”</p> <p>8. Hệ thống hiển thị trang thanh toán đơn hàng</p> <p>9. Người dùng chọn phương thức thanh toán >> tiến hành thanh toán</p> <p>10. Hệ thống xử lý thanh toán thành công</p> <p>11. Hệ thống tiến hành lưu đơn hàng vào hệ thống trạng thái đã thanh toán.</p> <p>12. Hệ thống hiển thị trang check-out với đơn hàng đã được tạo thành công</p>
Exception	Không có
Alternative Flows	<p>Alternative 1: tiếp tục với bước 5:</p> <p>5. Người dùng kiểm tra thông tin đặt hàng >> người dùng chọn phương thức thanh toán cod</p> <p>6. Hệ thống loại bỏ hidden của nút tạo đơn hàng</p> <p>7. Người dùng bấm “tạo đơn hàng”</p> <p>8. Hệ thống tiến hành lưu đơn hàng vào hệ thống trạng thái đã đặt hàng.</p> <p>9. Hệ thống hiển thị trang check-out với đơn hàng đã được tạo thành công với thông báo nhắc “Bạn hãy nhanh chóng thanh toán theo phương thức COD để lấy hàng”</p> <p>Alternative 2: tiếp tục với bước 5:</p> <p>5. Người dùng kiểm tra thông tin đặt hàng >> người dùng chọn phương thức thanh toán chuyển khoản</p> <p>6. Hệ thống hiển thị thông tin, QR code để người dùng chuyển khoản. Hệ thống loại bỏ hidden của nút tạo đơn hàng</p> <p>7. Người dùng bấm “tạo đơn hàng”</p> <p>8. Hệ thống tiến hành lưu đơn hàng vào hệ thống trạng thái đã đặt hàng.</p> <p>9. Hệ thống hiển thị trang check-out với đơn hàng đã được tạo thành công với thông báo nhắc “Bạn hãy nhanh chóng thanh toán theo phương thức chuyển khoản để lấy hàng”</p>

Bảng 9: Use case Người dùng đặt thuê

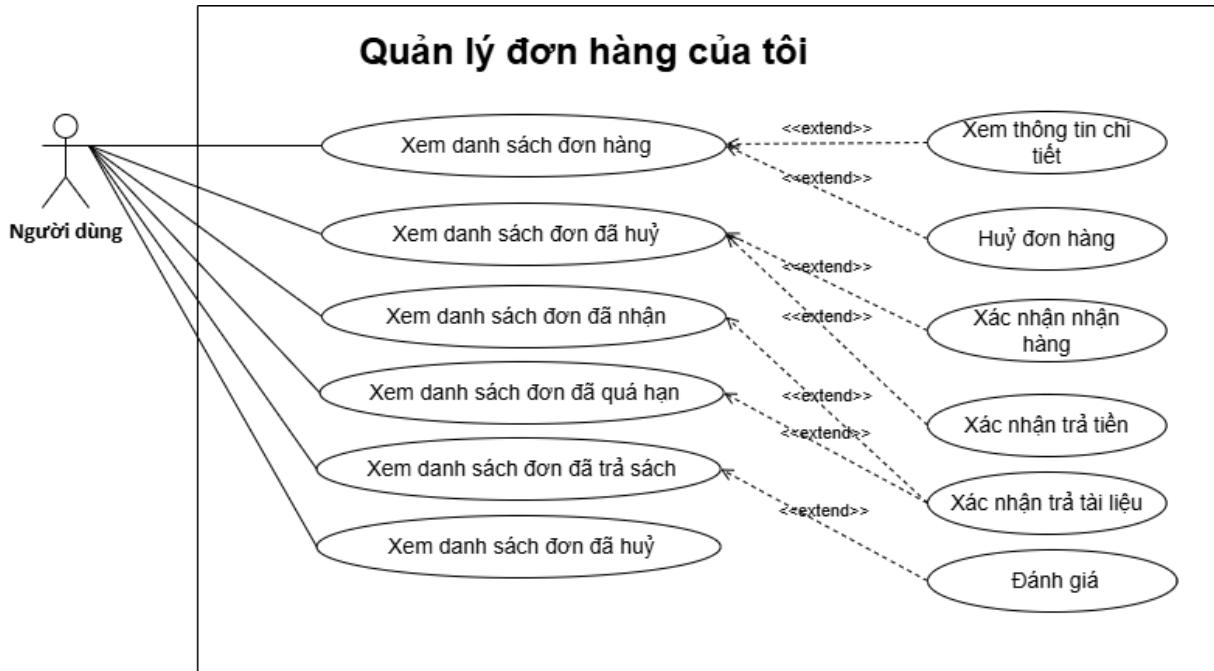
- **Đặt mua**

Use case name	Người dùng đặt mua
Actor	Người dùng
Description	Người dùng đặt mua sản phẩm
Preconditions	Người dùng đang ở trang giỏ hàng Có sản phẩm người dùng muốn mua trong giỏ hàng

Postconditions	Người dùng đặt mua thành công cho đơn hàng Hệ thống cập nhật đơn hàng vào cơ sở dữ liệu
Normal Flow	<ol style="list-style-type: none"> 1. Tại trang giỏ hàng >> Người dùng bấm nút “Đặt mua hàng” 2. Hệ thống hiển thị trang “Đặt hàng” với toàn bộ thông tin của đơn hàng gồm: thông tin người đặt, thông tin đặt hàng và thông tin thanh toán 3. Người dùng nhập thông tin đặt mua >> bấm xác nhận 4. Hệ thống cập nhật thông tin người đặt hàng. Hệ thống hiển thị thông báo nhắc “Chọn thanh toán rồi tạo đơn hàng” 5. Người dùng kiểm tra thông tin đặt hàng >> người dùng chọn phương thức thanh toán cod 6. Hệ thống loại bỏ hidden của nút tạo đơn hàng 7. Người dùng bấm “tạo đơn hàng” 8. Hệ thống tiến hành lưu đơn hàng vào hệ thống trạng thái đã đặt hàng. 9. Hệ thống hiển thị trang check-out với đơn hàng đã được tạo thành công với thông báo nhắc “Bạn hãy nhanh chóng thanh toán theo phương thức COD để lấy hàng”
Exception	Không có
Alternative Flows	<p>Alternative 1: Tiếp tục với bước 5:</p> <ol style="list-style-type: none"> 5. Người dùng kiểm tra thông tin đặt hàng >> người dùng chọn phương thức thanh toán chuyển khoản 6. Hệ thống hiển thị thông tin, QR code để người dùng dùng chuyển khoản. Hệ thống loại bỏ hidden của nút tạo đơn hàng 7. Người dùng bấm “tạo đơn hàng” 8. Hệ thống tiến hành lưu đơn hàng vào hệ thống trạng thái đã đặt hàng. 9. Hệ thống hiển thị trang check-out với đơn hàng đã được tạo thành công với thông báo nhắc “Bạn hãy nhanh chóng thanh toán theo phương thức chuyển khoản để lấy hàng”

Bảng 10: Use case Người dùng đặt mua

4.5.1.3 Người dùng quản lý đơn hàng



Hình 12: Module quản lý đơn hàng

- Xem danh sách đơn hàng

Use case name	Người dùng xem danh sách đơn hàng
Actor	Người dùng
Description	Người dùng muốn xem danh sách các đơn hàng
Preconditions	Người dùng đã đăng nhập thành công Người dùng đang ở trang chủ
Postconditions	Người dùng xem danh sách đơn hàng thành công Hệ thống hiển thị thông tin danh sách đơn hàng thành công
Normal Flow	<ol style="list-style-type: none"> 1. Người dùng bấm vào biểu tượng icon quản lý >> Quản lý đơn hàng 2. Hệ thống hiển thị trang quản lý đơn hàng ở tab tất cả 3. Hệ thống hiển thị danh sách các đơn hàng của user
Alternative Flows	Không có
Exception	Không có

Bảng 11: Use case Người dùng xem danh sách đơn hàng

- Xem danh sách đơn hàng đã đặt

Use case name	Người dùng xem danh sách đơn đã đặt
Actor	Người dùng

Description	Người dùng muốn xem danh sách các đơn hàng đã đặt
Preconditions	Người dùng đã đăng nhập thành công Người dùng đang ở trang quản lý đơn hàng của tôi
Postconditions	Người dùng xem danh sách đơn hàng đã đặt thành công Hệ thống hiển thị thông tin danh sách đơn hàng đã đặt thành công
Normal Flow	<ol style="list-style-type: none"> 1. Người dùng bấm vào tab “ĐÃ ĐẶT HÀNG” 2. Hệ thống hiển thị trang quản lý đơn hàng ở tab đã đặt hàng 3. Hệ thống hiển thị danh sách các đơn hàng đã đặt của user
Alternative Flows	Không có
Exception	Không có

Bảng 12: Use case Người dùng xem danh sách đơn đã đặt

- Xem danh sách đơn hàng đã nhận

Use case name	Người dùng xem danh sách đơn đã nhận
Actor	Người dùng
Description	Người dùng muốn xem danh sách các đơn hàng đã nhận
Preconditions	Người dùng đã đăng nhập thành công Người dùng đang ở trang quản lý đơn hàng của tôi
Postconditions	Người dùng xem danh sách đơn hàng đã nhận Hệ thống hiển thị thông tin danh sách đơn hàng đã nhận thành công
Normal Flow	<ol style="list-style-type: none"> 1. Người dùng bấm vào tab “ĐÃ NHẬN” 2. Hệ thống hiển thị trang quản lý đơn hàng ở tab đã nhận 3. Hệ thống hiển thị danh sách các đơn hàng đã nhận
Alternative Flows	Không có
Exception	Không có

Bảng 13: Use case Người dùng xem danh sách đơn đã nhận

- Xem danh sách đơn hàng đã quá hạn

Use case name	Người dùng xem danh sách đơn đã quá hạn
Actor	Người dùng
Description	Người dùng muốn xem danh sách các đơn hàng đã quá hạn
Preconditions	Người dùng đã đăng nhập thành công Người dùng đang ở trang quản lý đơn hàng của tôi
Postconditions	Người dùng xem danh sách đơn hàng đã quá hạn thành công Hệ thống hiển thị thông tin danh sách đơn hàng đã quá hạn thành

	công
Normal Flow	<ol style="list-style-type: none"> Người dùng bấm vào tab “ĐÃ QUÁ HẠN” Hệ thống hiển thị trang quản lý đơn hàng ở tab đã quá hạn Hệ thống hiển thị danh sách các đơn hàng đã quá hạn
Alternative Flows	Không có
Exception	Không có

Bảng 14: Use case Người dùng xem danh sách đơn đã quá hạn

- Xem danh sách đơn hàng đã trả sách

Use case name	Người dùng xem danh sách đơn đã trả sách
Actor	Người dùng
Description	Người dùng muốn xem danh sách các đơn hàng đã trả sách
Preconditions	Người dùng đã đăng nhập thành công Người dùng đang ở trang quản lý đơn hàng của tôi
Postconditions	Người dùng xem danh sách đơn hàng đã trả sách thành công Hệ thống hiển thị thông tin danh sách đơn hàng đã trả sách thành công
Normal Flow	<ol style="list-style-type: none"> Người dùng bấm vào tab “ĐÃ TRẢ SÁCH” Hệ thống hiển thị trang quản lý đơn hàng ở tab đã trả sách Hệ thống hiển thị danh sách các đơn hàng đã trả sách
Alternative Flows	Không có
Exception	Không có

Bảng 15: Use case Người dùng xem danh sách đơn đã trả sách

- Xem danh sách đơn hàng đã huỷ

Use case name	Người dùng xem danh sách đơn đã huỷ
Actor	Người dùng
Description	Người dùng muốn xem danh sách các đơn hàng đã huỷ
Preconditions	Người dùng đã đăng nhập thành công Người dùng đang ở trang quản lý đơn hàng của tôi
Postconditions	Người dùng xem danh sách đơn hàng đã trả huỷ Hệ thống hiển thị thông tin danh sách đơn hàng đã huỷ thành công
Normal Flow	<ol style="list-style-type: none"> Người dùng bấm vào tab “ĐÃ HUỶ” Hệ thống hiển thị trang quản lý đơn hàng ở tab đã huỷ Hệ thống hiển thị danh sách các đơn đã huỷ

Alternative Flows	Không có
Exception	Không có

Bảng 16: Use case Người dùng xem danh sách đơn đã huỷ

- Xác nhận đã trả tiền

Use case name	Người dùng xác nhận đã trả tiền cọc
Actor	Người dùng
Description	Người dùng trả tiền cọc cho một đơn hàng chuyên khoán
Preconditions	Người dùng đã đăng nhập thành công Người dùng đang ở trang quản lý đơn hàng của tôi >> tab đã đặt hàng
Postconditions	Người dùng xác nhận đã trả cọc thành công Hệ thống lưu trạng thái xác nhận
Normal Flow	1. Người dùng bấm vào nút “đã trả tiền” ở đơn hàng chuyên khoán trong danh sách đã đặt 2. Hệ thống lưu thông tin xác nhận và ẩn nút “đã trả tiền” ở đơn hàng
Alternative Flows	Không có
Exception	Không có

Bảng 17: Use case Người dùng xác nhận đã trả tiền cọc

- Đánh giá đơn hàng

Use case name	Người dùng đánh giá đơn hàng
Actor	Người dùng
Description	Người dùng đánh giá đơn hàng đã trả sách
Preconditions	Người dùng đã đăng nhập thành công Người dùng đang ở trang quản lý đơn hàng của tôi tab đã trả hoặc tab tất cả
Postconditions	Người dùng xem đánh giá đơn hàng thành công Hệ thống lưu thông tin đánh giá đơn hàng thành công
Normal Flow	1. Người dùng bấm vào nút đánh giá đơn hàng ở đơn hàng đã trả 2. Hệ thống hiển thị pop up đánh giá đơn hàng 3. Người dùng nhập thông tin số điểm đánh giá và mô tả >> Đánh giá 4. Hệ thống cập nhật đánh giá đơn hàng thành công. Hệ thống quay lại tab trang quản lý
Alternative Flows	Không có

Exception	Exception 1: Tiếp tục bước 3: 3a. Người dùng bấm nút “Huỷ” 4b. Hệ thống quay lại tab trang quản lý đơn hàng
------------------	--

Bảng 18: Use case Người dùng đánh giá đơn hàng

- Xác nhận trả tài liệu

Use case name	Người dùng xác nhận trả tài liệu
Actor	Người dùng
Description	Người dùng muốn trả tài liệu đang thuê
Preconditions	Người dùng đã đăng nhập thành công Đơn hàng đang ở trạng thái đã nhận sách thuê hoặc quá hạn Người dùng đang ở trang quản lý đơn hàng của tôi >> tab đã nhận, đã quá hạn, hoặc tất cả
Postconditions	Người dùng xác nhận trả tài liệu cho thuê thành công Hệ thống tiến hành lưu dữ liệu trên cơ sở dữ liệu thành công
Normal Flow	1. Người dùng bấm nút “trả tài liệu” ở đơn hàng 2. Hệ thống tiến hành cập nhật thông tin vào cơ sở dữ liệu >> hiển thị thay đổi lên trang web
Exception	Không có
Alternative Flows	Không có

Bảng 19: Use case Người dùng xác nhận trả tài liệu

- Xác nhận nhận hàng

Use case name	Người dùng xác nhận nhận tài liệu
Actor	Người dùng
Description	Người dùng xác nhận nhận tài liệu từ chủ thuê
Preconditions	Người dùng đã đăng nhập thành công Đơn hàng đang ở trạng thái đã thanh toán Người dùng đang ở trang quản lý đơn hàng của tôi >> tab đã đặt hàng hoặc tất cả
Postconditions	Người dùng xác nhận nhận tài liệu thành công Hệ thống tiến hành lưu dữ liệu trên cơ sở dữ liệu thành công
Normal Flow	1. Người dùng bấm nút “đã nhận được hàng” ở đơn hàng cụ thể 2. Hệ thống tiến hành cập nhật thông tin vào cơ sở dữ liệu >> hiển thị thay đổi lên trang web
Exception	Không có
Alternative Flows	Không có

Bảng 20: Use case Người dùng xác nhận nhận tài liệu

- **Huỷ đơn hàng**

Use case name	Người dùng xác nhận huỷ đơn hàng
Actor	Người dùng
Description	Người dùng xác nhận huỷ đơn hàng
Preconditions	Người dùng đã đăng nhập thành công Người dùng đang ở trang quản lý đơn hàng của tôi Đơn huỷ là đơn đang ở trạng thái chờ thanh toán
Postconditions	Người dùng huỷ đơn thành công Hệ thống tiến hành lưu dữ liệu trên cơ sở dữ liệu thành công
Normal Flow	1. Người dùng bấm icon thùng rác ở đơn ở đơn hàng có thể huỷ đơn 2. Hệ thống tiến hành cập nhật thông tin vào cơ sở dữ liệu >> hiển thị thay đổi lên trang web
Exception	Không có
Alternative Flows	Không có

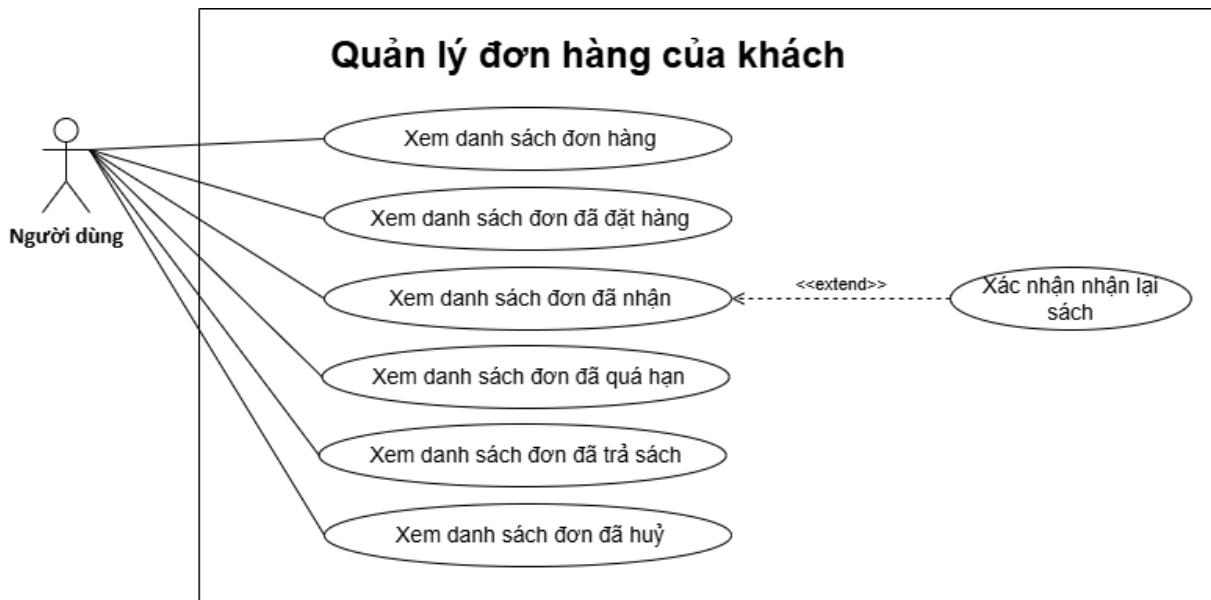
Bảng 21: Use case Người dùng xác nhận huỷ đơn hàng

- **Xem thông tin chi tiết đơn hàng**

Use case name	Người dùng xem chi tiết đơn hàng
Actor	Người dùng
Description	Người dùng xem chi tiết đơn hàng
Preconditions	Người dùng đã đăng nhập thành công Người dùng đang ở trang quản lý đơn hàng của tôi
Postconditions	Người dùng xem chi tiết đơn hàng thành công Hệ thống hiển thị chi tiết đơn hàng thành công
Normal Flow	1. Người dùng bấm icon xem chi tiết đơn hàng bên cạnh đơn hàng cụ thể 2. Hệ thống hiển thị trang chi tiết đơn hàng với thông tin chi tiết đơn hàng
Exception	Không có
Alternative Flows	Không có

Bảng 22: Use case Người dùng xem chi tiết đơn hàng

4.5.1.4 Người dùng quản lý đơn hàng của khách



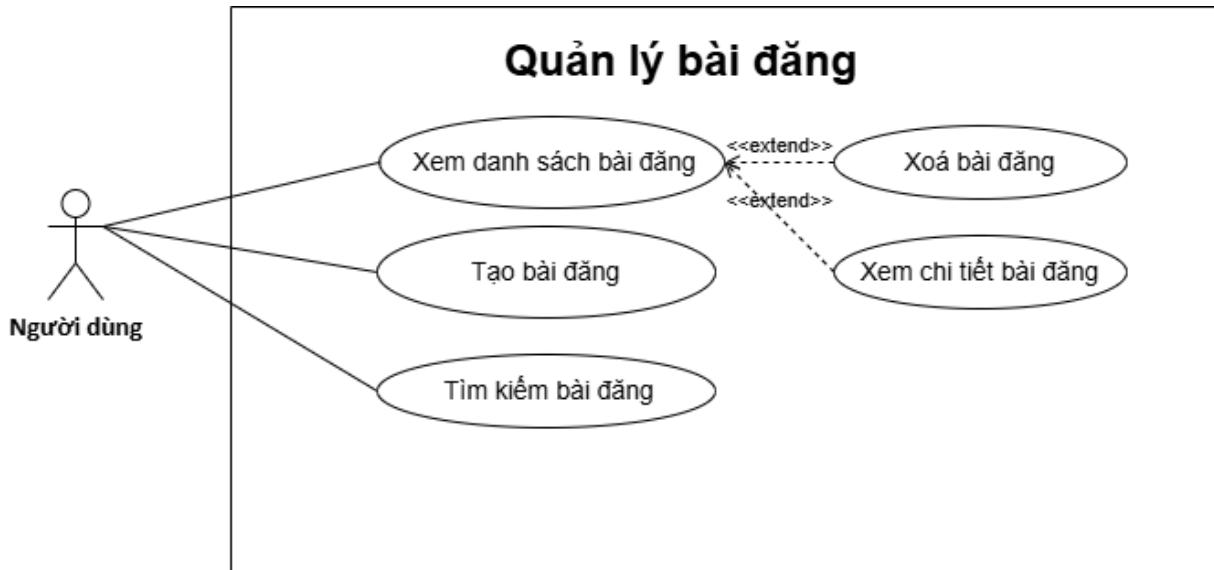
Hình 13: Module quản lý đơn hàng của khách

- Use case Xác nhận nhận lại sách

Use case name	Người dùng xác nhận lại sách
Actor	Người dùng
Description	Người dùng xác nhận lại sách
Preconditions	Người dùng đã đăng nhập thành công Người dùng đang ở trang quản lý đơn hàng của khách tab đã đặt hàng hoặc tab đã nhận
Postconditions	Người dùng xác nhận nhận lại sách thành công Hệ thống tiến hành lưu dữ liệu trên cơ sở dữ liệu thành công
Normal Flow	1. Người dùng bấm nút “đã nhận lại sách” ở đơn hàng cụ thể. 2. Hệ thống tiến hành cập nhật thông tin vào cơ sở dữ liệu >> hiển thị thay đổi lên trang web
Exception	Không có
Alternative Flows	Không có

Bảng 23: Use case Người dùng xác nhận nhận lại sách

4.5.1.5 Người dùng quản lý bài đăng



Hình 14: Module quản lý bài đăng

- Xem danh sách bài đăng

Use case name	Người dùng xem danh sách bài đăng
Actor	Người dùng
Description	Người dùng muốn xem danh sách các bài đăng của mình
Preconditions	Người dùng đã đăng nhập thành công Người dùng ở trang chủ
Postconditions	Người dùng xem được danh sách bài đăng của mình Hệ thống thành công hiển thị danh sách bài đăng của cá nhân user
Normal Flow	1. Người dùng bấm vào icon quản lý bài đăng 2. Hệ thống hiển thị trang bài đăng có danh sách bài đăng mà người dùng đã đăng
Exception	Exception 1: tại bước 2: 2a. Hệ thống hiển thị thông báo hiện không có bài đăng nào ở đây nếu người dùng chưa có bài đăng nào
Exception	Không có
Alternative Flows	Không có

Bảng 24: Use case Người dùng xem danh sách bài đăng

- Tạo bài đăng

Use case name	Người dùng tạo bài đăng
Actor	Người dùng
Description	Người dùng tạo bài đăng

Preconditions	Người dùng đã đăng nhập thành công Người dùng đang ở trang quản lý bài đăng
Postconditions	Người dùng đăng bài đăng thành công Hệ thống lưu bài đăng vào cơ sở dữ liệu và hiển thị trên trang web
Normal Flow	<ol style="list-style-type: none"> Tại trang bài đăng của website, Người dùng bấm vào nút thêm bài đăng Hệ thống hiển thị form đăng bài Ở mục tìm sách Người dùng chọn thêm mới sách >> điền thông tin thêm mới >> bấm nút “tạo sách” Hệ thống hiển thị tạo sách thành công Ở mục tài liệu người dùng thêm thông tin >> bấm “Tạo tài liệu” Hệ thống hiển thị tạo tài liệu thành công Ở mục bài đăng người dùng thêm thông tin >> bấm “Tạo bài đăng” Hệ thống hiển thị tạo bài đăng thành công. Hệ thống hiển thị trang chi tiết bài đăng
Exception	Không có
Alternative Flows	<p>Alternative 1: Tiếp tục bước 3 3a. Ở mục tìm sách Người dùng chọn sách >> bấm nút “Chọn sách” 4a. Hệ thống hiển thị chọn sách thành công Tiếp tục bước 5</p>

Bảng 25: Use case Người dùng tạo bài đăng

- Tìm kiếm bài đăng**

Use case name	Người dùng tìm kiếm bài đăng
Actor	Người dùng
Description	Người dùng muốn tìm kiếm bài đăng của mình
Preconditions	Người dùng đã đăng nhập thành công Người dùng đang ở trang quản lý bài đăng
Postconditions	Người dùng tìm kiếm được bài đăng muốn tìm Hệ thống hiển thị bài đăng nếu tìm kiếm được bài đăng
Normal Flow	<ol style="list-style-type: none"> Tại trang tìm kiếm người dùng input tìm kiếm thêm từ khoá tìm Hệ thống tìm kiếm và hiển thị các bài đăng phù hợp với nhu cầu
Exception	<p>Exception 1: tại bước 2: 4a. Người dùng không có bài đăng nào phù hợp với tình trạng tìm kiếm, hệ thống hiển thị không có bài đăng nào được tìm thấy</p>
Alternative Flows	Không có

Bảng 26: Use case Người dùng tìm kiếm bài đăng

- Xoá bài đăng**

Use case name	Người dùng xóa bài đăng
Actor	Người dùng
Description	Người dùng muốn xoá một bài đăng
Preconditions	Người dùng đã đăng nhập thành công Người dùng đang ở trang quản lý bài đăng
Postconditions	Người dùng xoá bài đăng thành công Hệ thống xoá bài đăng khỏi cơ sở dữ liệu
Normal Flow	<ol style="list-style-type: none"> 1. Người dùng bấm nút thùng rác cuối bài đăng muốn xoá 2. Hệ thống hiển thị pop-up bạn có muốn xoá bài đăng không 3. Người dùng bấm “Đồng ý” 4. Hệ thống hiển thị xoá bài đăng thành công
Exception	<p>Exception 1: Tại bước 3:</p> <p>3a. Người dùng bấm “Hủy” 4b. Hệ thống quay lại trang quản lý bài đăng</p>
Alternative Flows	Không có

Bảng 27: Use case Người dùng xóa bài đăng

4.5.2 Use Cases chung cho Quản trị viên và Người dùng

4.5.2.1 Đăng nhập

Use case name	Người dùng/Quản trị viên đăng nhập
Actor	Người dùng/Quản trị viên
Description	Người dùng/Quản trị viên cần đăng nhập vào tài khoản hiện có để sử dụng hệ thống
Preconditions	Người dùng/Quản trị viên đã đăng ký tài khoản trước đó trong hệ thống Với tài khoản Quản trị viên, thì cần được hệ thống phân quyền Quản trị viên
Postconditions	Người dùng/Quản trị viên đăng nhập thành công Hệ thống ghi nhận hoạt động đăng nhập của Người dùng/Quản trị viên vào database
Normal Flow	<ol style="list-style-type: none"> 1. Tại trang chủ của website, Người dùng/Quản trị viên nhấn nút “Đăng nhập/Đăng ký” 2. Hệ thống hiển thị form đăng nhập 3. Người dùng/Quản trị viên nhập tài khoản, mật khẩu 4. Người dùng/Quản trị viên bấm nút “đăng nhập” 5. Hệ thống xác thực thông tin đăng nhập của Người dùng/Quản trị viên

	<p>6. Hệ thống ghi nhận hoạt động đăng nhập thành công</p> <p>7. Hệ thống hiển thị trang chủ sau khi đăng nhập</p>
Exception	<p>Exception 1:</p> <p>Tại bước 5</p> <p>5a. Kiểm tra thông tin đăng nhập chưa đúng</p> <p>6a. Hệ thống hiển thị lại form đăng nhập, và thông báo mật khẩu hoặc tên đăng nhập chưa đúng</p> <p>Quay lại và tiếp tục với bước 3</p>
Alternative Flows	Không có

Bảng 28: Use case Người dùng/Quản trị viên đăng nhập

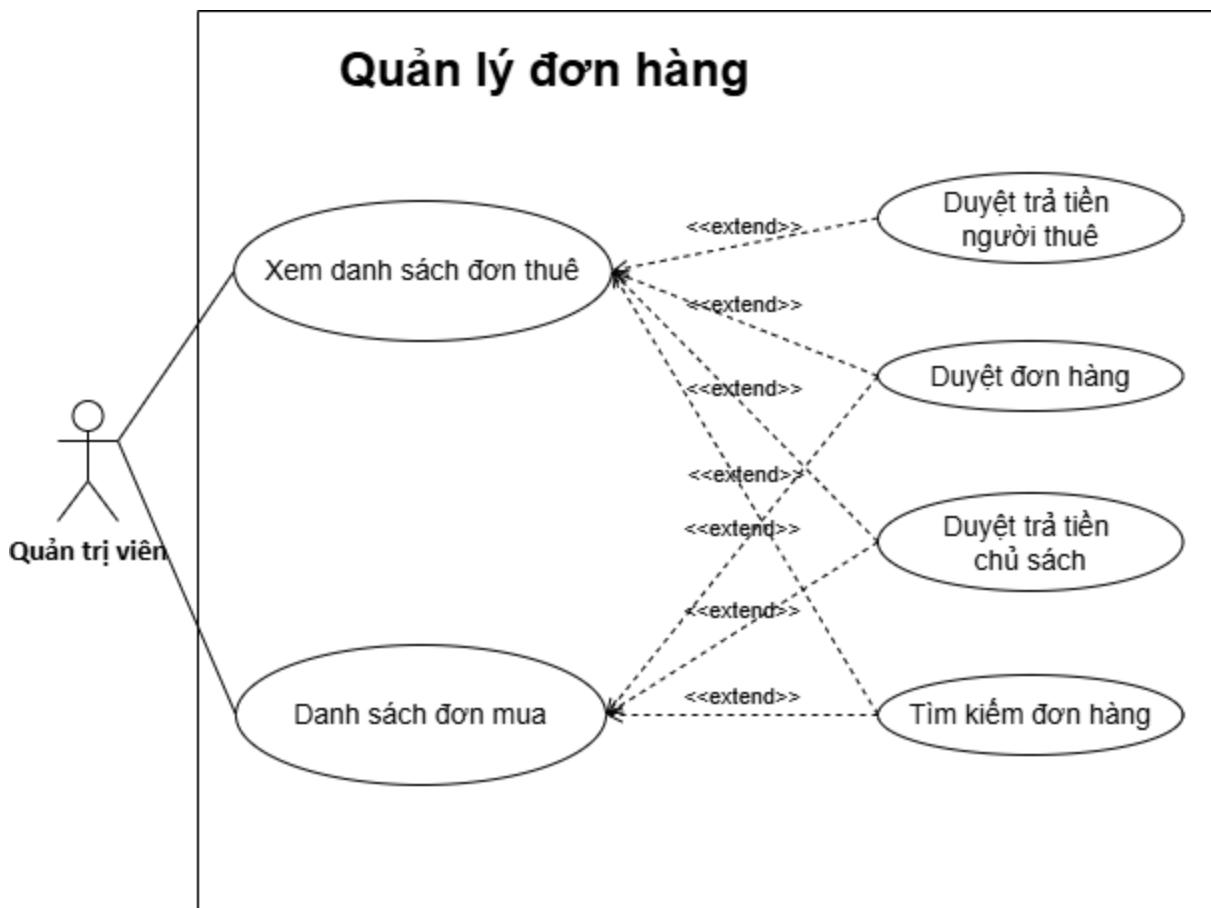
4.5.2.2 Đăng xuất

Use case name	Người dùng/Quản trị viên đăng xuất
Actor	Người dùng/Quản trị viên
Description	Người dùng/Quản trị viên cần đăng xuất khỏi tài khoản hiện có
Preconditions	Người dùng/Quản trị viên thì đã đăng nhập vào tài khoản hiện có
Postconditions	Người dùng/Quản trị viên đăng xuất thành công Hệ thống ghi nhận hoạt động đăng xuất của Người dùng/Quản trị viên vào database
Normal Flow	<ol style="list-style-type: none"> Người dùng/Quản trị viên nhấn vào icon “Tài khoản” Hệ thống hiển thị danh sách các nút bấm Người dùng/Quản trị viên nhấn nút “Đăng xuất” Hệ thống xác thực thông tin đăng xuất của Người dùng/Quản trị viên Hệ thống ghi nhận hoạt động đăng xuất thành công Hệ thống chuyển đến giao diện trang chủ
Alternative Flows	Không có
Exception	Không có

Bảng 29: Use case Người dùng/Quản trị viên đăng xuất

4.5.3 Use Cases cho Quản trị viên

4.5.3.1 Quản trị viên quản lý đơn hàng



Hình 15: Module quản lý đơn hàng của quản trị viên

- Xem danh sách đơn thuê

Use case name	Quản trị viên xem danh sách đơn hàng thuê
Actor	Quản trị viên
Description	Quản trị viên xem danh sách các đơn hàng thuê
Preconditions	Quản trị viên đã đăng nhập bằng tài khoản quản trị viên
Postconditions	Quản trị viên xem được danh sách đơn hàng thuê Hệ thống hiển thị danh sách đơn hàng thuê
Normal Flow	1. Quản trị viên truy cập vào trang quản lý đơn hàng thuê 2. Hệ thống hiển thị trang quản lý đơn hàng thuê, danh sách đơn hàng thuê
Exception	Không có
Alternative Flows	Không có

Bảng 30: Use case Quản trị viên xem danh sách đơn hàng thuê

- Xem danh sách đơn hàng mua

Use case name	Quản trị viên xem danh sách đơn hàng mua
Actor	Quản trị viên
Description	Quản trị viên xem danh sách các đơn hàng mua
Preconditions	Quản trị viên đã đăng nhập bằng tài khoản quản trị viên
Postconditions	Quản trị viên xem được danh sách đơn hàng mua Hệ thống hiển thị danh sách đơn hàng mua
Normal Flow	1. Quản trị viên truy cập vào trang quản lý đơn hàng mua 2. Hệ thống hiển thị trang quản lý đơn hàng mua, danh sách đơn hàng mua
Exception	Không có
Alternative Flows	Không có

Bảng 31: Use case Quản trị viên xem danh sách đơn hàng mua

- **Duyệt đơn hàng**

Use case name	Quản trị viên duyệt đơn hàng
Actor	Quản trị viên
Description	Quản trị viên tìm kiếm những đơn hàng muốn tìm
Preconditions	Quản trị viên đã đăng nhập bằng tài khoản quản trị viên
Postconditions	Quản trị viên duyệt được đơn hàng Thay đổi đơn hàng được cập nhật lưu trữ trong cơ sở dữ liệu
Normal Flow	1. Quản trị viên truy cập vào trang quản lý đơn hàng 2. Hệ thống hiển thị trang quản lý đơn hàng 3. Quản trị viên bấm vào nút duyệt đơn 4. Hệ thống tiến hành thay đổi trạng thái đơn hàng
Exception	Không có
Alternative Flows	Không có

Bảng 32: Use case Quản trị viên duyệt đơn hàng

- **Duyệt trả tiền người thuê**

Use case name	Quản trị viên duyệt trả tiền người thuê
Actor	Quản trị viên
Description	Quản trị viên duyệt trả tiền người thuê
Preconditions	Quản trị viên đã đăng nhập bằng tài khoản quản trị viên

Postconditions	Quản trị viên duyệt được đơn hàng Thay đổi đơn hàng được cập nhật lưu trữ trong cơ sở dữ liệu
Normal Flow	<ol style="list-style-type: none"> 1. Quản trị viên truy cập vào trang quản lý đơn hàng 2. Hệ thống hiển thị trang quản lý đơn hàng 3. Quản trị viên bấm vào nút duyệt trả tiền người thuê 4. Hệ thống tiến hành thay đổi trạng thái đơn hàng
Exception	Không có
Alternative Flows	Không có

Bảng 33: Use case Quản trị viên duyệt trả tiền người thuê

- Duyệt trả tiền người bán

Use case name	Quản trị viên duyệt trả tiền người bán
Actor	Quản trị viên
Description	Quản trị viên duyệt trả tiền người bán
Preconditions	Quản trị viên đã đăng nhập bằng tài khoản quản trị viên
Postconditions	Quản trị viên duyệt được đơn hàng Thay đổi đơn hàng được cập nhật lưu trữ trong cơ sở dữ liệu
Normal Flow	<ol style="list-style-type: none"> 1. Quản trị viên truy cập vào trang quản lý đơn hàng 2. Hệ thống hiển thị trang quản lý đơn hàng 3. Quản trị viên bấm vào nút duyệt trả tiền người bán 4. Hệ thống tiến hành thay đổi trạng thái đơn hàng
Exception	Không có
Alternative Flows	Không có

Bảng 34: Use case Quản trị viên duyệt trả tiền người bán

4.5.4 Use Cases chung cho Người dùng và Khách

4.5.4.1 Đăng ký tài khoản

Use case name	Khách đăng ký tài khoản
Actor	Khách
Description	Khách vãng lai cần đăng ký một tài khoản mới để đăng nhập hệ thống
Preconditions	Khách vãng lai muốn đăng ký tài khoản trên hệ thống
Postconditions	Khách đăng ký tài khoản mới thành công Hệ thống ghi nhận và lưu trữ tài khoản vào cơ sở dữ liệu Khách đang ở trang chủ

Normal Flow	<ol style="list-style-type: none"> 1. Khách nhấn nút “Đăng ký” 2. Hệ thống hiển thị trang đăng ký chưa form đăng ký 3. Khách điền thông tin cá nhân được yêu cầu 4. Khách bấm nút “Tạo tài khoản” 5. Hệ thống xác thực thông tin tài khoản mới, kiểm tra tài khoản nếu phù hợp để đăng ký và lưu trữ tài khoản mới vào cơ sở dữ liệu 6. Hệ thống quay lại đăng nhập
Exception	Không có
Alternative Flows	Không có

Bảng 35: Use case Khách đăng ký tài khoản

4.5.4.2 Xem danh sách đề xuất

Use case name	Người dùng/Khách xem danh sách đề xuất
Actor	Người dùng/Khách
Description	Người dùng/Khách xem danh sách bài đăng
Preconditions	Người dùng/Khách ở trang chủ của trang web
Postconditions	Người dùng/Khách xem được danh sách đề xuất thành công Hệ thống hiển thị danh sách đề xuất
Normal Flow	<ol style="list-style-type: none"> 1. Tại trang chủ của website, Người dùng/Khách thấy danh sách đề xuất
Exception	Không có
Alternative Flows	Không có

Bảng 36: Use case Người dùng/Khách xem danh sách đề xuất

4.5.4.3 Chi tiết bài đăng

Use case name	Người dùng/Khách xem chi tiết bài đăng
Actor	Người dùng/Khách
Description	Người dùng/Khách xem chi tiết bài đăng
Preconditions	Người dùng/Khách đang ở trang chủ hoặc trang danh sách tài liệu, hoặc danh sách tài liệu từ tìm kiếm tài liệu
Postconditions	Người dùng/Khách xem được chi tiết thông tin về tài liệu thành công Hệ thống hiển thị thông tin tài liệu
Normal Flow	<ol style="list-style-type: none"> 1. Tại danh sách tài liệu, Người dùng/Khách nhấn vào bài đăng muốn xem thông tin chi tiết

	2. Hệ thống hiển thị thông tin chi tiết bài đăng
Exception	Không có
Alternative Flows	Không có

Bảng 37: Use case Người dùng/Khách xem chi tiết bài đăng

4.5.4.4 Tìm kiếm tài liệu

Use case name	Người dùng/Khách tìm kiếm tài liệu
Actor	Người dùng/Khách
Description	Người dùng/Khách tìm kiếm tài liệu với từ khóa
Preconditions	Người dùng/Khách cần có từ khóa về tài liệu để dễ kiếm Người dùng/Khách đang ở trang chủ
Postconditions	Người dùng/Khách tìm được danh sách tài liệu cần tìm theo đúng từ khóa thành công Hệ thống hiển thị danh sách các tài liệu tìm thấy
Normal Flow	<ol style="list-style-type: none"> Người dùng/Khách thấy nút tìm kiếm Hệ thống hiển thị ô input để người dùng/khách nhập từ khóa Người dùng/Khách được nhập từ khóa Người dùng/Khách nhấn nút tìm kiếm Hệ thống hiển thị danh sách tài liệu có chứa từ khóa tìm kiếm của Người dùng/Khách
Exception	Exception 1: tại bước 3 3a. Nếu hệ thống không tìm thấy tài liệu của khách sẽ hiển thị là “Hiển thị 0 cuốn sách hiện có”
Alternative Flows	Alternative1 : <ol style="list-style-type: none"> Người dùng/khách nhấn icon thẻ loại bên trái input tìm kiếm Hệ thống hiển thị danh sách thẻ loại Người dùng nhấn chọn thẻ loại Hệ thống hiển thị danh sách tài liệu thuộc thẻ loại tìm kiếm

Bảng 38: Use case Người dùng/Khách tìm kiếm tài liệu

4.5.4.5 Thêm vào giỏ hàng

Use case name	Người dùng/Khách thêm vào giỏ hàng
Actor	Người dùng/Khách
Description	Người dùng/Khách thêm vào giỏ hàng tài liệu mà mình muốn
Preconditions	Người dùng/Khách cần ở trang chi tiết bài đăng của tài liệu mà Người dùng/Khách muốn thêm vào

Postconditions	Người dùng/Khách thêm tài liệu vào giỏ hàng thành công Hệ thống hiển thị bài đăng có trong giỏ hàng
Normal Flow	<ol style="list-style-type: none"> 1. Người dùng/Khách đang ở trang chi tiết bài đăng của tài liệu 2. Người dùng/Khách thêm ngày nhận ngày trả 3. Hệ thống active nút “Đặt thuê ngay” >> người dùng/khách bấm nút đó 4. Hệ thống thêm bài đăng vào giỏ hàng thành công
Exception	Không có
Alternative Flows	Không có

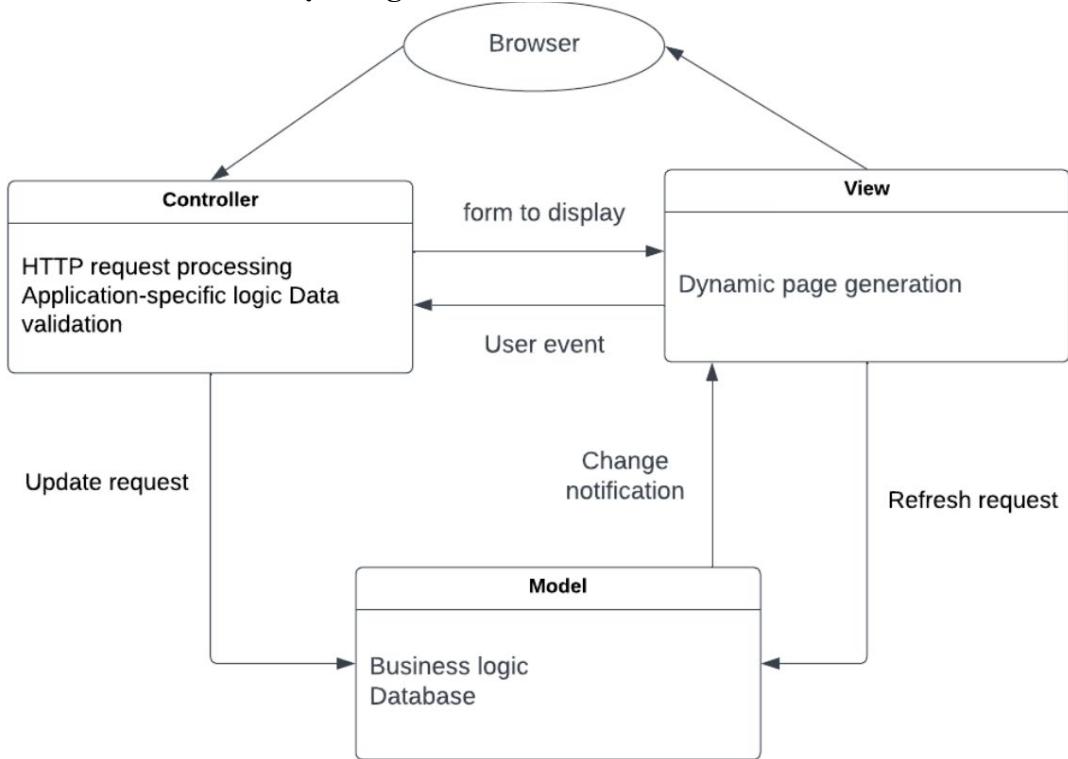
Bảng 39: Use case Người dùng/Khách thêm vào giỏ hàng

4.5.4.6 Xem giỏ hàng

Use case name	Người dùng/Khách xem giỏ hàng
Actor	Người dùng/Khách
Description	Người dùng/Khách xem được giỏ hàng
Preconditions	Người dùng/Khách ở trang chủ
Postconditions	Người dùng/Khách xem giỏ hàng thành công Hệ thống hiển thị bài đăng trong giỏ hàng
Normal Flow	<ol style="list-style-type: none"> 1. Người dùng/Khách bấm icon giỏ hàng 2. Hệ thống hiển thị trang giỏ hàng và đơn hàng trong giỏ hàng
Exception	Exception 1: tại bước 2: 2b. Không có bài đăng trong giỏ hàng. Hệ thống hiển thị “giỏ hàng trống”
Alternative Flows	Không có

Bảng 40: Use case Người dùng/Khách xem giỏ hàng

4.6 Mô hình kiến trúc hệ thống

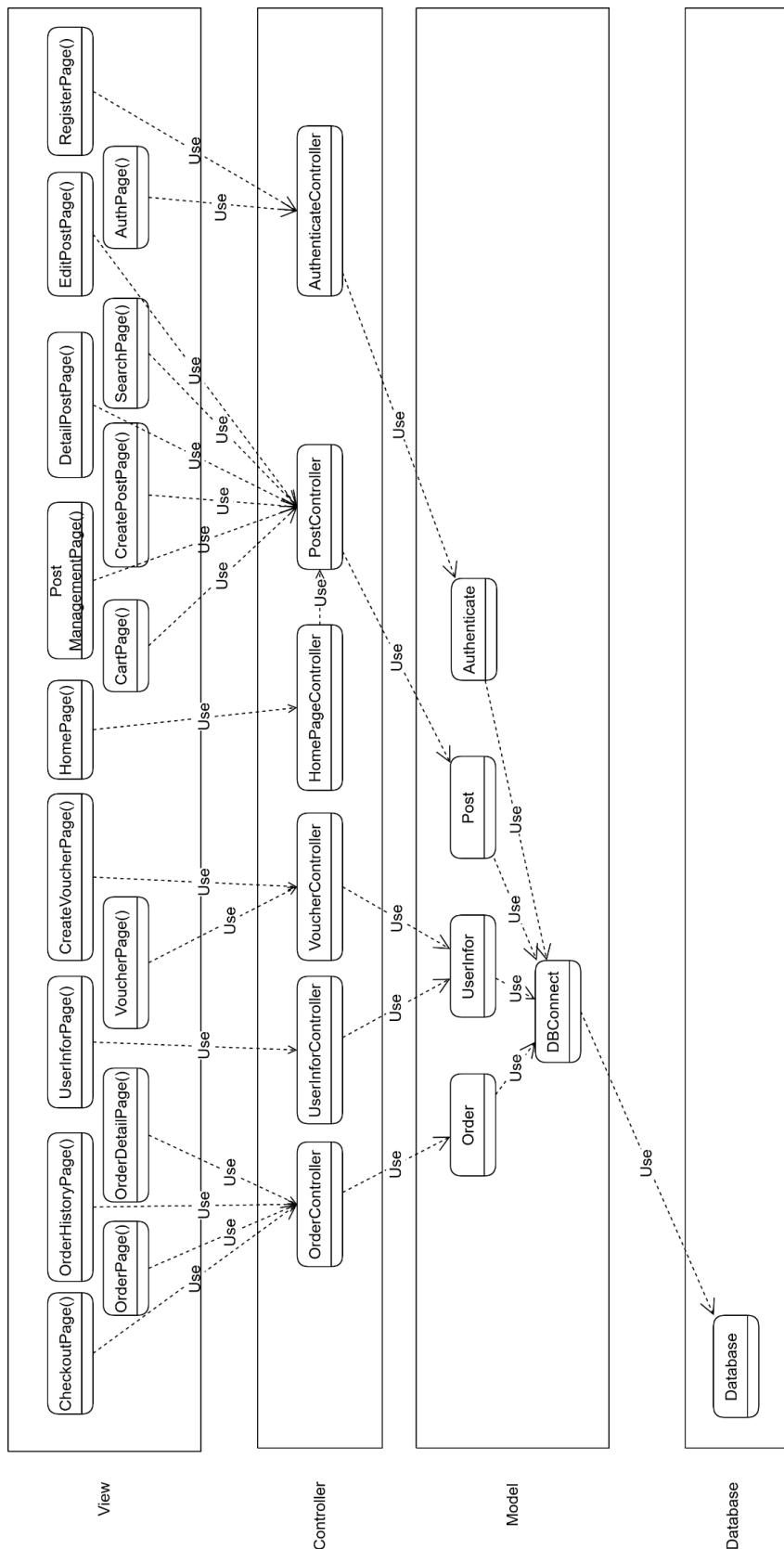


Hình 16: Mô hình kiến trúc hệ thống MVC

- **Model:** thực hiện các logic cho việc quản lý dữ liệu của hệ thống.
 - Thực hiện các chức năng truy vấn trên database.
 - Cung cấp api xử lý, trả về các dữ liệu theo business logic.
 - Phát đi thông báo sự kiện về việc dữ liệu được cập nhật để View có thể đáp ứng tức thì với sự thay đổi của dữ liệu theo thời gian thực.
- **View:** Trung gian tương tác giữa người dùng và hệ thống. Có nhiệm vụ quản lý các UI và các logic tương tác của người dùng với hệ thống thông qua UI
 - Render thành phần của UI của từng page.
 - Gửi request đến controller dựa trên người dùng event
 - Nhận các notify từ Model để cập nhật những thay đổi về dữ liệu hiển thị.
- **Controller:** Là đối tượng trung gian điều hành giao tiếp dạng request/response (yêu cầu/phản hồi) giữa Model và View, đồng thời có thể điều hành việc thực hiện một số thao tác tiền xử lý/hậu xử lý đối với các request và response
 - Lấy dữ liệu từ Model gửi cho View.
 - Yêu cầu cập nhật dữ liệu đến Model từ dữ liệu của View.
 - Thực hiện các chức năng xác thực dữ liệu.
 - Thực hiện các thao tác xử lý trên dữ liệu gửi đến model và view.

4.7 Layered architecture diagram

4.7.1 Cho Người dùng và khách



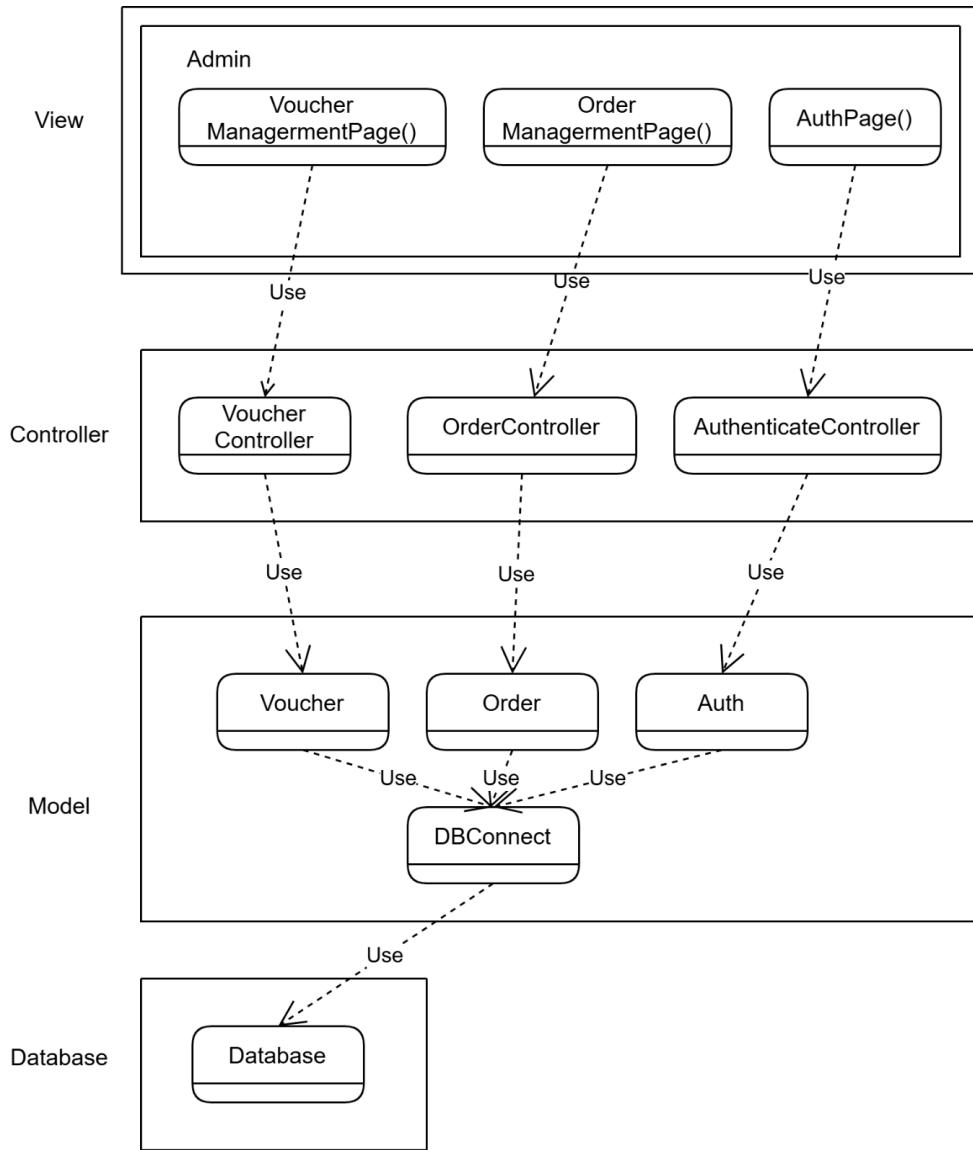
Hình 17: Layered architecture diagram cho người dùng và khách

Layered architecture diagram được team em sử dụng theo mô hình MVC.

Mô tả chi tiết các lớp như sau:

- **Lớp View:** là nền tảng website hiển thị giao diện người dùng trực quan, mỗi đối tượng trong View đại diện cho một đến nhiều trang trong website thực hiện chức năng hiển thị thông tin và các chức năng người dùng tương tác với hệ thống.
- **Lớp Controller:** cung cấp các chức năng về hiển thị và quản lý cho UI mà nó được truyền tải qua trình duyệt website
 - OrderController quản lý các chức năng UI liên quan đến đơn hàng: như tạo đơn hàng, thông tin đơn hàng, chi tiết đơn hàng, review và các chức năng thao tác khác trên đơn hàng.
 - UserInfoController quản lý các chức năng UI liên qua đến user: đăng nhập, đăng ký, lấy dữ liệu liên quan đến người dùng.
 - BookController quản lý các chức năng liên quan đến listing, book, copy: tạo listing, lấy thông tin listing.
 - VoucherController quản lý chức năng liên quan đến hiển thị danh sách tất cả voucher của tôi, tạo voucher.
- **Lớp Model:** Thực hiện chức năng của service tính toán, cập nhật database, query database và các chức năng logic nghiệp vụ khác. DBconnect (repository) là một class tạo kết nối để thực hiện thao tác trên cơ sở dữ liệu.
- **Lớp Database:** sẽ lưu trữ dữ liệu của hệ thống.

4.7.2 Cho Quản trị viên



Hình 18: Layered architecture diagram cho quản trị viên

Hệ thống được xây dựng dựa trên mô hình MVC (Model-View-Controller), với các lớp chức năng được phân chia rõ ràng như sau:

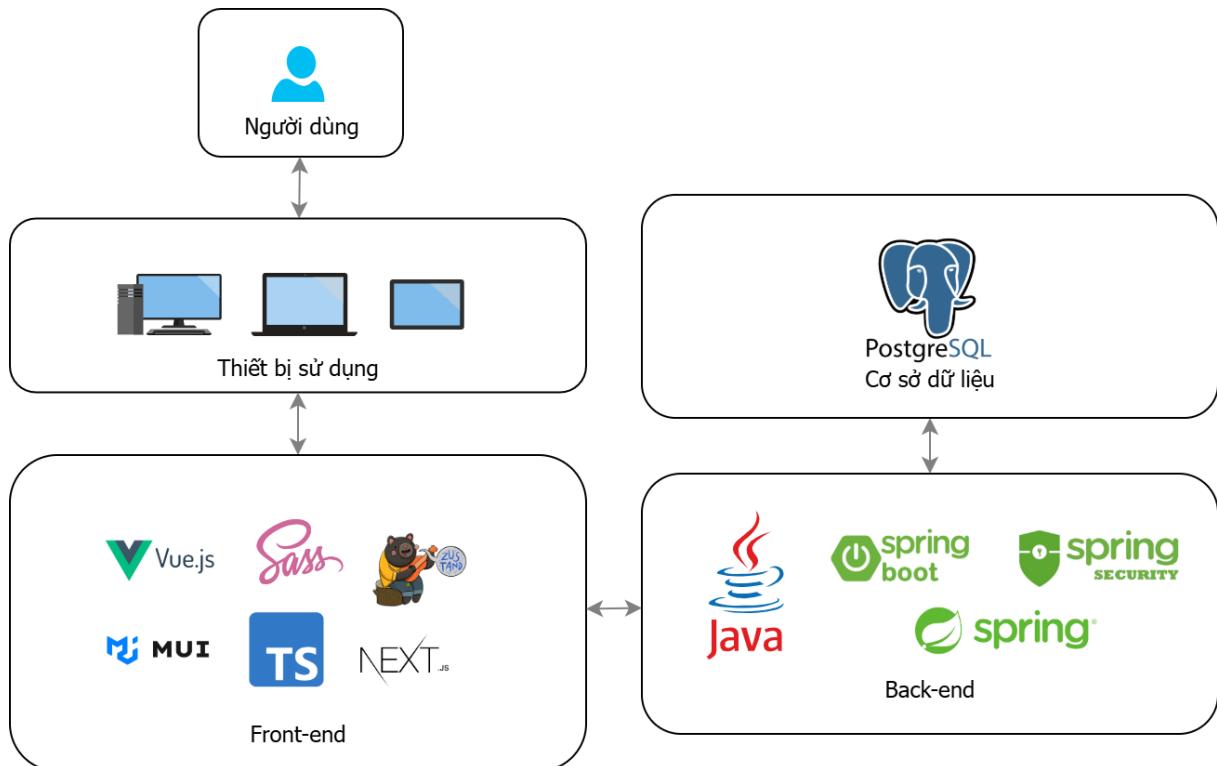
- **Lớp View:**
 - Là giao diện người dùng dành cho quản trị viên (Admin), bao gồm các trang hiển thị thông tin và cho phép tương tác với hệ thống.
 - Trong sơ đồ, lớp View gồm ba đối tượng chính:
 - **VoucherManagementPage()**: Trang quản lý voucher, hiển thị danh sách voucher và các chức năng thêm/sửa/xóa.
 - **OrderManagementPage()**: Trang quản lý đơn hàng, hiển thị thông tin đơn hàng, chi tiết đơn hàng, và hỗ trợ thao tác quản lý đơn hàng.
 - **AuthPage()**: Trang xác thực người dùng, cung cấp chức năng đăng nhập để truy cập hệ thống quản trị.
- **Lớp Controller:**

- Đảm nhận vai trò xử lý logic hiển thị, nhận yêu cầu từ View, và điều phối thông tin từ Model.
- Các Controller trong hệ thống bao gồm:
 - **VoucherController:** Xử lý các chức năng liên quan đến quản lý voucher, như lấy danh sách voucher, tạo, cập nhật hoặc xóa voucher.
 - **OrderController:** Quản lý các chức năng liên quan đến đơn hàng, như tạo mới, xem thông tin chi tiết, chỉnh sửa trạng thái đơn hàng.
 - **AuthenticateController:** Xử lý các chức năng xác thực, bao gồm đăng nhập, kiểm tra quyền truy cập, và đăng xuất.
- **Lớp Model:**
 - Là nơi thực hiện các logic nghiệp vụ, tính toán, truy vấn và cập nhật dữ liệu trong cơ sở dữ liệu.
 - Các đối tượng chính trong lớp Model:
 - **Voucher:** Xử lý logic liên quan đến quản lý voucher.
 - **Order:** Chịu trách nhiệm xử lý dữ liệu và nghiệp vụ liên quan đến đơn hàng.
 - **Auth:** Quản lý logic xác thực, lưu trữ thông tin phiên người dùng.
 - **DBConnect:** Là class repository, chịu trách nhiệm kết nối với cơ sở dữ liệu và thực hiện các thao tác như truy vấn, thêm, sửa, xóa dữ liệu.
- **Lớp Database:**
 - Là nơi lưu trữ toàn bộ dữ liệu của hệ thống, bao gồm thông tin voucher, đơn hàng, và thông tin người dùng.
 - Lớp Database được truy cập thông qua lớp Model bằng cách sử dụng **DBConnect**.

Luồng hoạt động:

1. Quản trị viên (Admin) truy cập vào một trong các trang trong lớp View (Voucher Management, Order Management, Auth Page).
2. Các yêu cầu từ View sẽ được gửi đến Controller tương ứng (VoucherController, OrderController, AuthenticateController).
3. Controller xử lý yêu cầu bằng cách gọi đến các thành phần trong lớp Model để thực hiện logic nghiệp vụ và truy xuất dữ liệu.
4. Lớp Model thông qua **DBConnect** truy vấn dữ liệu từ lớp Database và trả kết quả về Controller.
5. Controller gửi dữ liệu trở lại lớp View để hiển thị kết quả cho quản trị viên.

4.8 Deploy diagram



Hình 19: Deploy diagram

Dưới đây là mô tả Deploy Diagram

1. Người dùng: Người dùng truy cập vào hệ thống thông qua các thiết bị khác nhau như máy tính để bàn, laptop hoặc tablet.

2. Front-end: Front-end được phát triển sử dụng các công nghệ hiện đại bao gồm:

- **Vue.js**: Framework JavaScript cho xây dựng giao diện người dùng. Dùng cho giao diện admin
- **Sass**: Công cụ giúp quản lý và tạo CSS hiệu quả.
- **Zustand**: Thư viện quản lý trạng thái cực kì nhẹ, thay thế cho Redux.
- **MUI (Material-UI)**: Thư viện giao diện dựa trên Material Design.
- **TypeScript (TS)**: Ngôn ngữ lập trình mạnh mẽ, mở rộng từ JavaScript.
- **Next.js**: Framework xây dựng ứng dụng React toàn diện với hỗ trợ SSR (Server-Side Rendering). Dùng cho giao diện Người dùng chính thức.

3. Back-end: Back-end được xây dựng dựa trên:

- **Java**: Ngôn ngữ lập trình chính cho server-side.
- **Spring Framework**: Framework toàn diện cho phát triển ứng dụng doanh nghiệp.
- **Spring Boot**: Tạo ứng dụng Spring nhanh chóng và hiệu quả.
- **Spring Security**: Bảo mật ứng dụng bằng cách kiểm soát xác thực và ủy quyền.

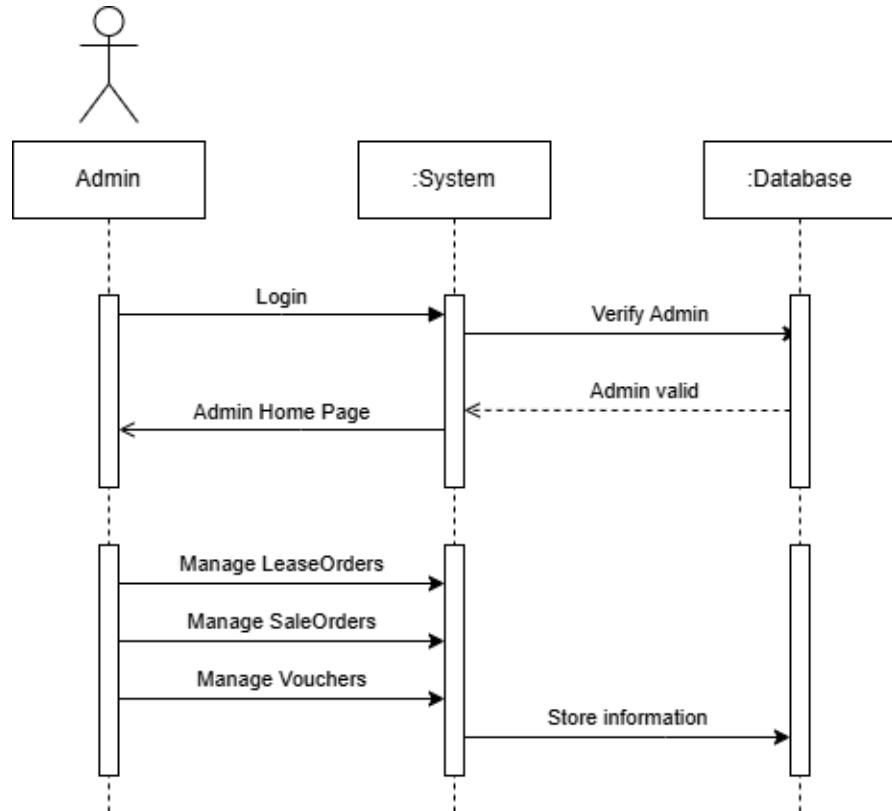
4. Cơ sở dữ liệu:

- **PostgreSQL**: Hệ quản trị cơ sở dữ liệu quan hệ được sử dụng để lưu trữ và quản lý dữ liệu.

Luồng hoạt động: Người dùng tương tác với giao diện Front-end qua thiết bị sử dụng. Front-end sẽ giao tiếp với Back-end để xử lý logic và truy xuất dữ liệu từ cơ sở dữ liệu PostgreSQL. Mọi hoạt động được đảm bảo an toàn nhờ Spring Security trong Back-end.

4.9 Sequence diagram cho các use case

4.9.1 VỚI ACTOR QUẢN TRỊ VIÊN

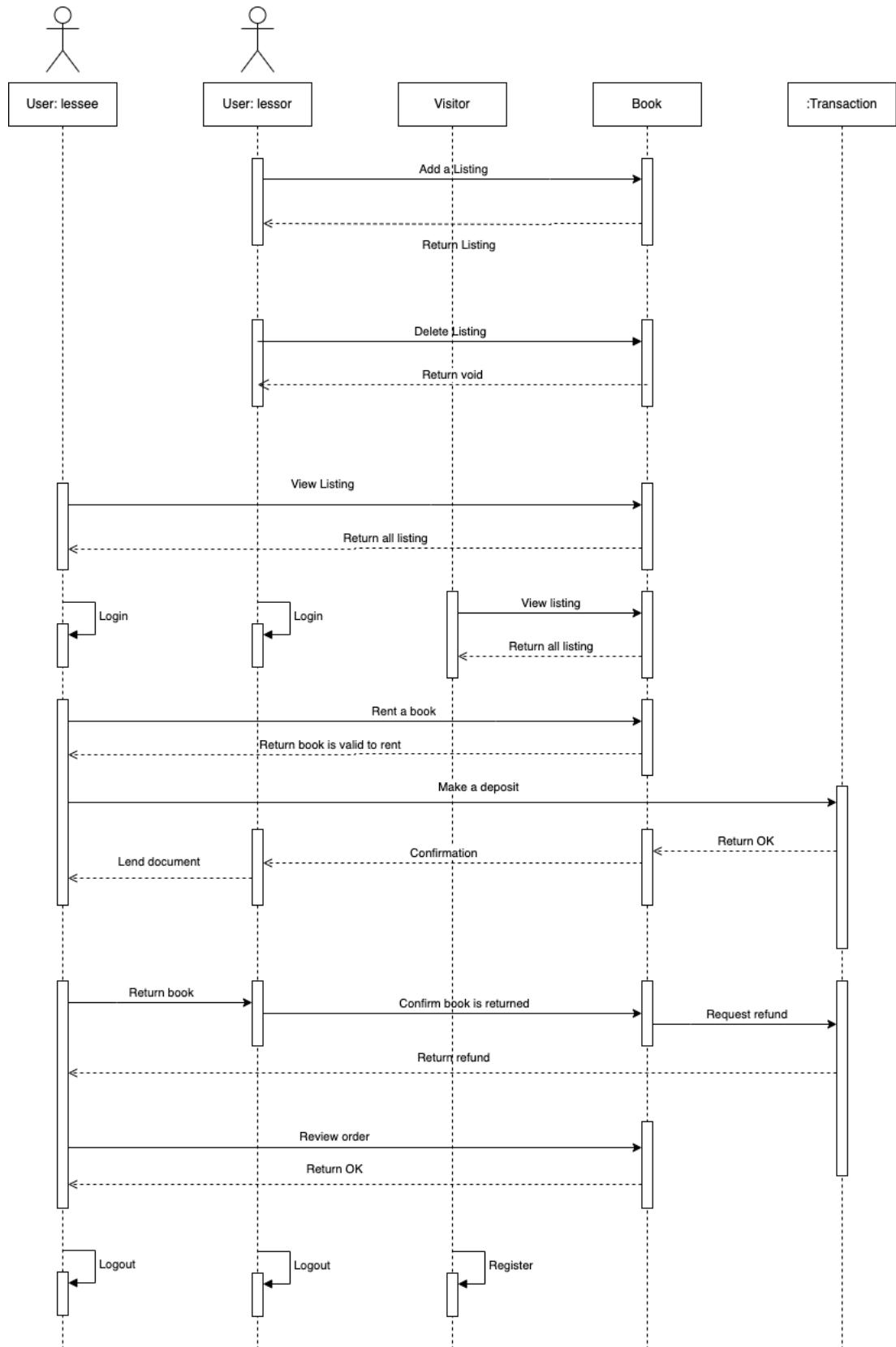


Hình 20: Sequence diagram cho quản trị viên

Lược đồ trình tự của Quản trị viên được hiểu như sau:

- Quản trị viên đăng nhập vào hệ thống, hệ thống tiến hành xác minh username và password mà Quản trị viên cung cấp.
- Sau khi hệ thống xác minh hợp lệ thì hệ thống hiển thị trang quản lý Quản trị viên.
- Quản trị viên tiến hành các thao tác quản lý đơn hàng như duyệt đơn thuê và bán, duyệt trả phí, quản lý voucher. Hệ thống lưu trữ thông tin thao tác vào cơ sở dữ liệu.

4.9.2 Với actor Người cho thuê và Người thuê sách

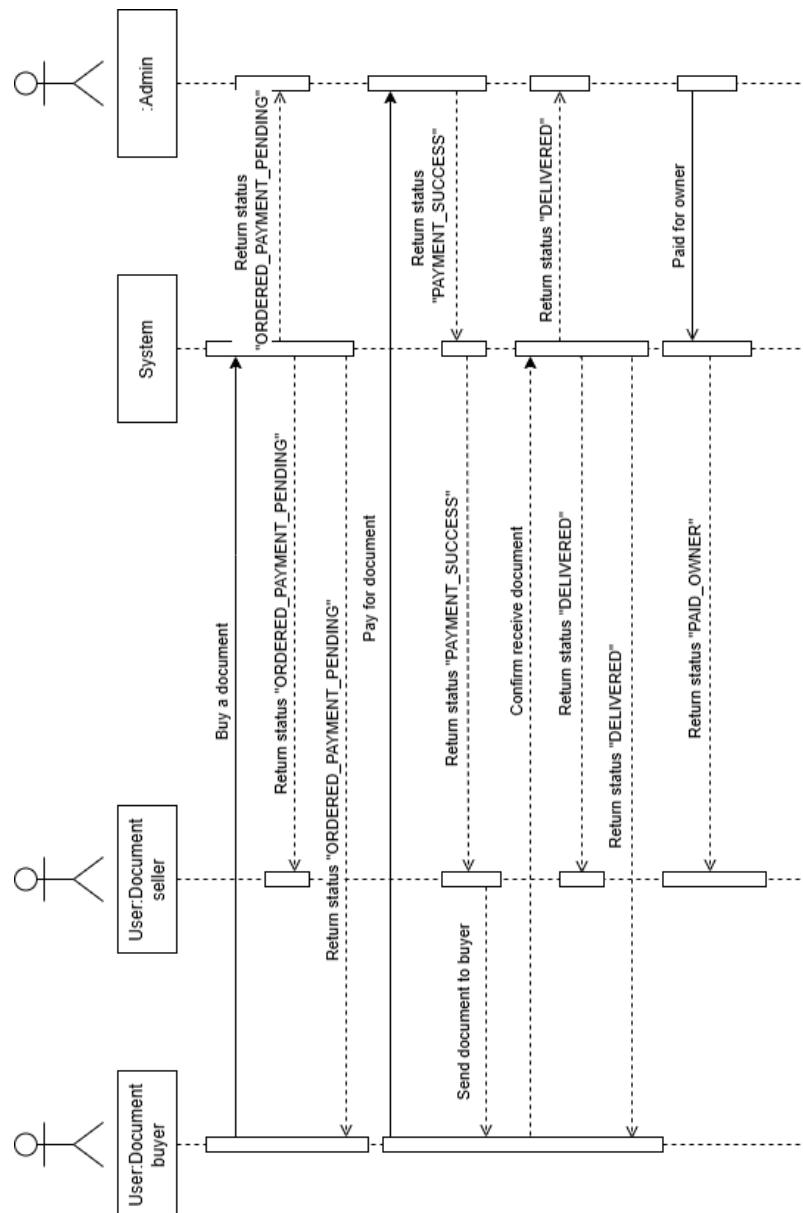


Hình 21: Sequence diagram cho người dùng và khách thuê sách

Lược đồ trình tự cho hoạt động của người cho thuê và người thuê được thể hiện như sau:

- Chủ sách thao tác tạo bài đăng trên hệ thống, hệ thống trả về bài đăng mới đã được tạo.
- Người dùng xoá bài đăng trên hệ thống, hệ thống trả về void.
- Người dùng và khách truy cập trang web và hệ thống hiển thị danh sách các bài đăng.
- Người dùng tiến hành mượn sách, hệ thống đánh giá sách có thể cho mượn. Người dùng trả phí cọc, hệ thống chuyển trạng thái của đơn hàng đã trả cọc.
- Người dùng trả sách, hệ thống chuyển trạng thái đơn hàng đã trả sách.
- Người dùng đánh giá đơn hàng đã trả sách, hệ thống trả về review đơn hàng.

4.9.3 VỚI ACTOR NGƯỜI MUA SÁCH VÀ BÁN SÁCH



Hình 22: Sequence diagram với người mua và bán sách

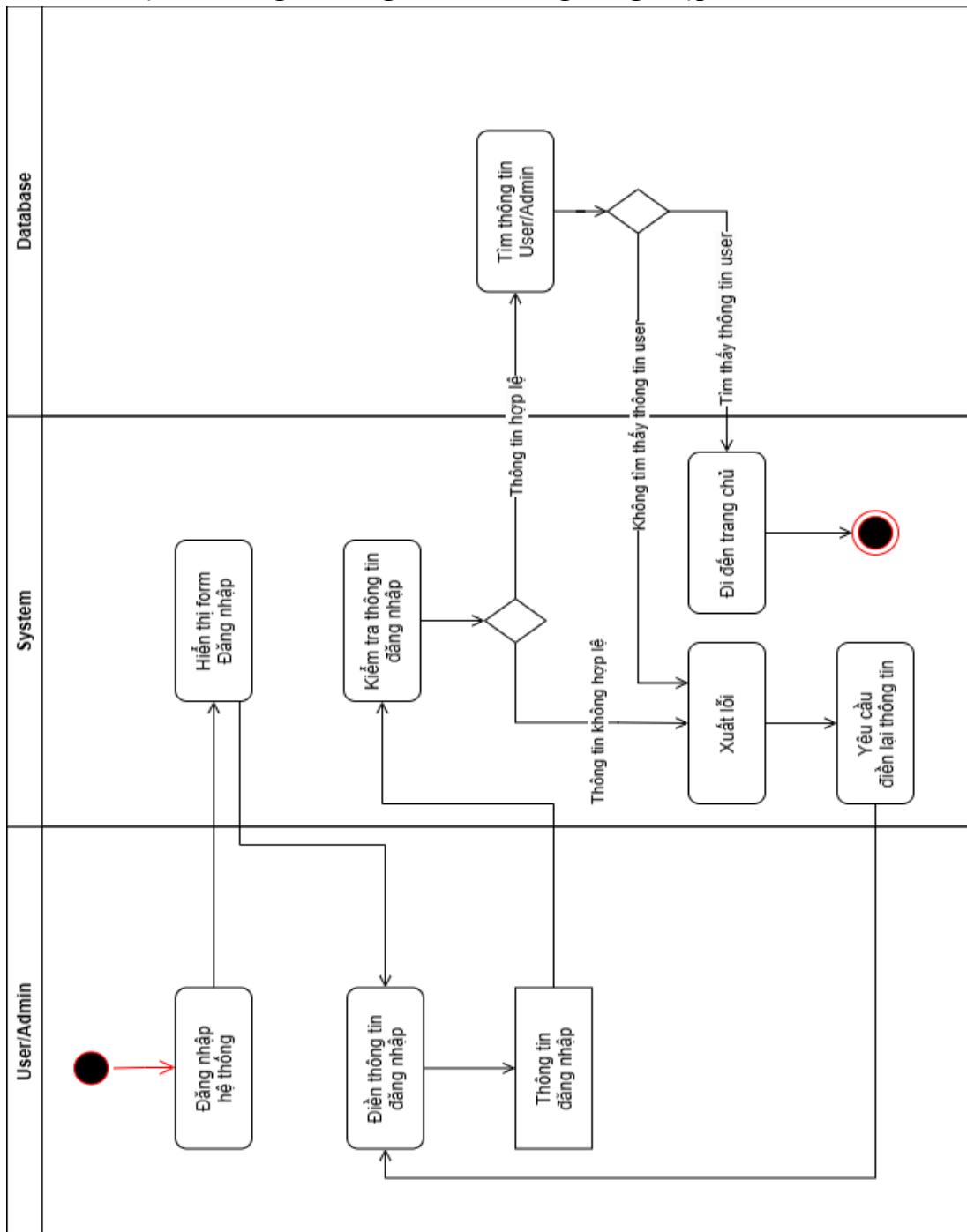
Lược đồ trình tự cho hoạt động của người mua và người bán sách được thể hiện như sau:

- Người mua sách thực hiện thao tác mua sách trên hệ thống, hệ thống trả về trạng thái "ORDERED_PAYMENT_PENDING".

- Người mua sách thanh toán cho sách, hệ thống chuyển trạng thái đơn hàng thành "PAYMENT_SUCCESS".
- Người bán sách gửi sách cho người mua, hệ thống trả về trạng thái "DELIVERED".
- Người mua sách xác nhận đã nhận sách, hệ thống cập nhật trạng thái đơn hàng thành "DELIVERED".
- Hệ thống thực hiện thanh toán cho người bán sách, hệ thống trả về trạng thái "PAID_OWNER".

4.10 Activity diagram (Lược đồ hoạt động)

4.10.1 Quản trị viên và Người dùng với chức năng Đăng nhập



Hình 23: Activity diagram Quản trị viên và Người dùng với chức năng Đăng nhập

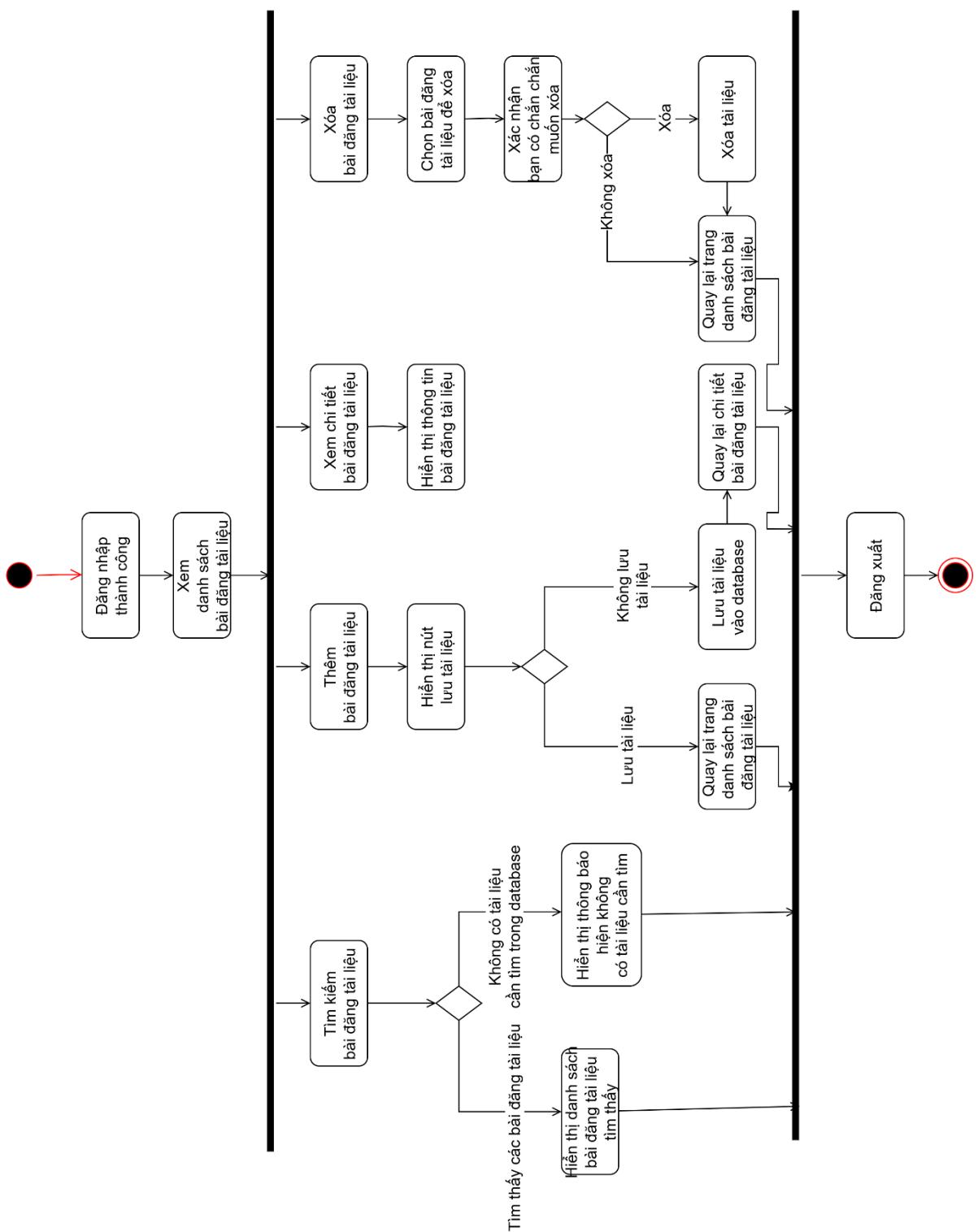
Mô tả Activity Diagram: Chức năng Đăng nhập của Quản trị viên và Người dùng

1. **Bắt đầu:** Quản trị viên hoặc người dùng bắt đầu tiến trình đăng nhập vào hệ thống.
2. **Nhập thông tin đăng nhập:** Người dùng/Quản trị viên nhập thông tin đăng nhập (tên đăng nhập và mật khẩu) vào form.
3. **Gửi thông tin đăng nhập:** Thông tin đăng nhập được gửi từ giao diện người dùng tới hệ thống.
4. **Kiểm tra thông tin đăng nhập:** Hệ thống kiểm tra tính hợp lệ của thông tin đăng nhập:
 - a. **Nếu thông tin đăng nhập không hợp lệ:** Hệ thống thông báo lỗi và yêu cầu người dùng nhập lại thông tin.
 - b. **Nếu thông tin hợp lệ:** Hệ thống tiếp tục bước tiếp theo.
5. **Tìm thông tin User/Admin trong cơ sở dữ liệu:** Hệ thống truy vấn cơ sở dữ liệu để tìm thông tin người dùng hoặc quản trị viên.
6. **Nếu không tìm thấy thông tin trong cơ sở dữ liệu:** Hệ thống trả về thông báo lỗi, yêu cầu người dùng kiểm tra lại thông tin đăng nhập.
7. **Nếu tìm thấy thông tin:** Tiến hành chuyển đến bước tiếp theo.
8. **Đi đến trang chủ:** Hệ thống xác nhận đăng nhập thành công và điều hướng người dùng/Quản trị viên đến trang chủ của hệ thống.
9. **Kết thúc:** Quy trình đăng nhập hoàn tất.

Lưu ý:

- Các trường hợp lỗi, bao gồm nhập sai thông tin đăng nhập hoặc không tìm thấy thông tin người dùng trong cơ sở dữ liệu, đều được xử lý bằng cách yêu cầu nhập lại thông tin.
- Nếu đăng nhập thành công, hệ thống sẽ cho phép truy cập vào trang chính.

4.10.2 Người dùng quản lý bài đăng



Hình 24: Activity diagram người dùng quản lý bài đăng

Mô tả Activity Diagram: Quản lý bài đăng của người dùng

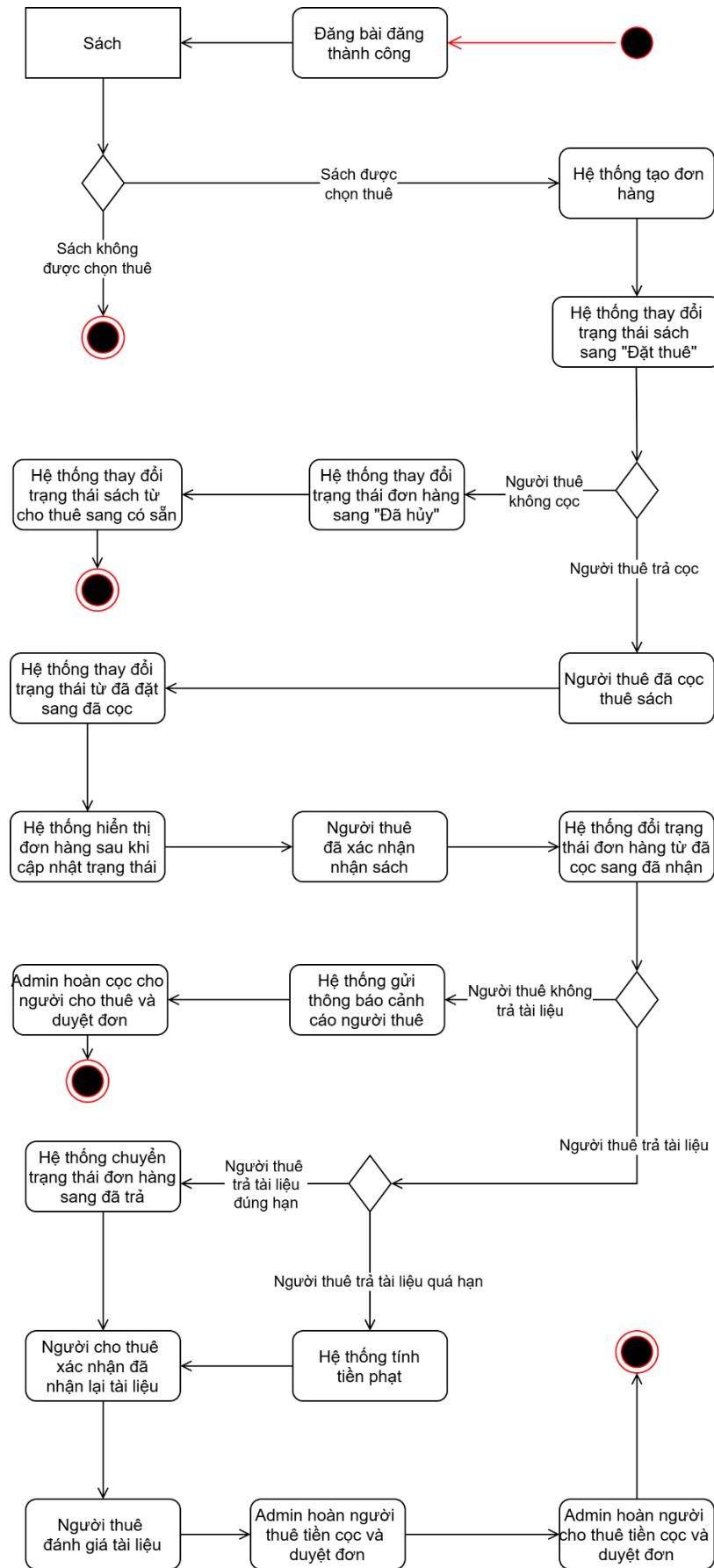
1. **Bắt đầu:** Người dùng đăng nhập thành công vào hệ thống và truy cập trang quản lý bài đăng tài liệu.
 2. **Tìm kiếm tài liệu:** Người dùng thực hiện tìm kiếm bài đăng tài liệu:

- a. **Nếu tìm thấy bài đăng:** Hệ thống hiển thị danh sách các bài đăng phù hợp với tiêu chí tìm kiếm.
 - b. **Nếu không tìm thấy bài đăng:** Hệ thống hiển thị thông báo "Không có tài liệu nào phù hợp".
3. **Thêm bài đăng tài liệu:** Người dùng chọn thao tác thêm bài đăng tài liệu:
 - a. Hệ thống hiển thị giao diện thêm bài đăng mới và nút lưu tài liệu.
 - b. Người dùng có thể quyết định:
 - i. **Không lưu tài liệu:** Hệ thống quay lại trang danh sách bài đăng tài liệu.
 - ii. **Lưu tài liệu:** Hệ thống lưu thông tin bài đăng vào cơ sở dữ liệu và quay lại trang danh sách bài đăng.
 4. **Xem chi tiết bài đăng tài liệu:** Người dùng chọn bài đăng để xem chi tiết.
 - a. Hệ thống hiển thị thông tin chi tiết bài đăng tài liệu.
 - b. Người dùng có thể quay lại trang danh sách bài đăng sau khi xem chi tiết.
 5. **Xóa bài đăng tài liệu:** Người dùng chọn bài đăng muốn xóa. Sau đó, hệ thống hiển thị yêu cầu xác nhận việc xóa.
 - a. **Nếu người dùng xác nhận xóa:** Hệ thống thực hiện xóa bài đăng khỏi cơ sở dữ liệu.
 - b. **Nếu người dùng không xác nhận:** Hệ thống quay lại trang danh sách bài đăng tài liệu.
 6. **Kết thúc:** Sau khi hoàn thành các thao tác quản lý bài đăng, người dùng có thể đăng xuất khỏi hệ thống.

Lưu ý:

- Hệ thống luôn kiểm tra và trả về thông báo phù hợp trong từng trường hợp, đảm bảo tính minh bạch và dễ sử dụng.
- Người dùng có quyền tùy chỉnh và thao tác theo nhu cầu trong phạm vi quản lý bài đăng của mình.

4.10.3 Người dùng thuê tài liệu

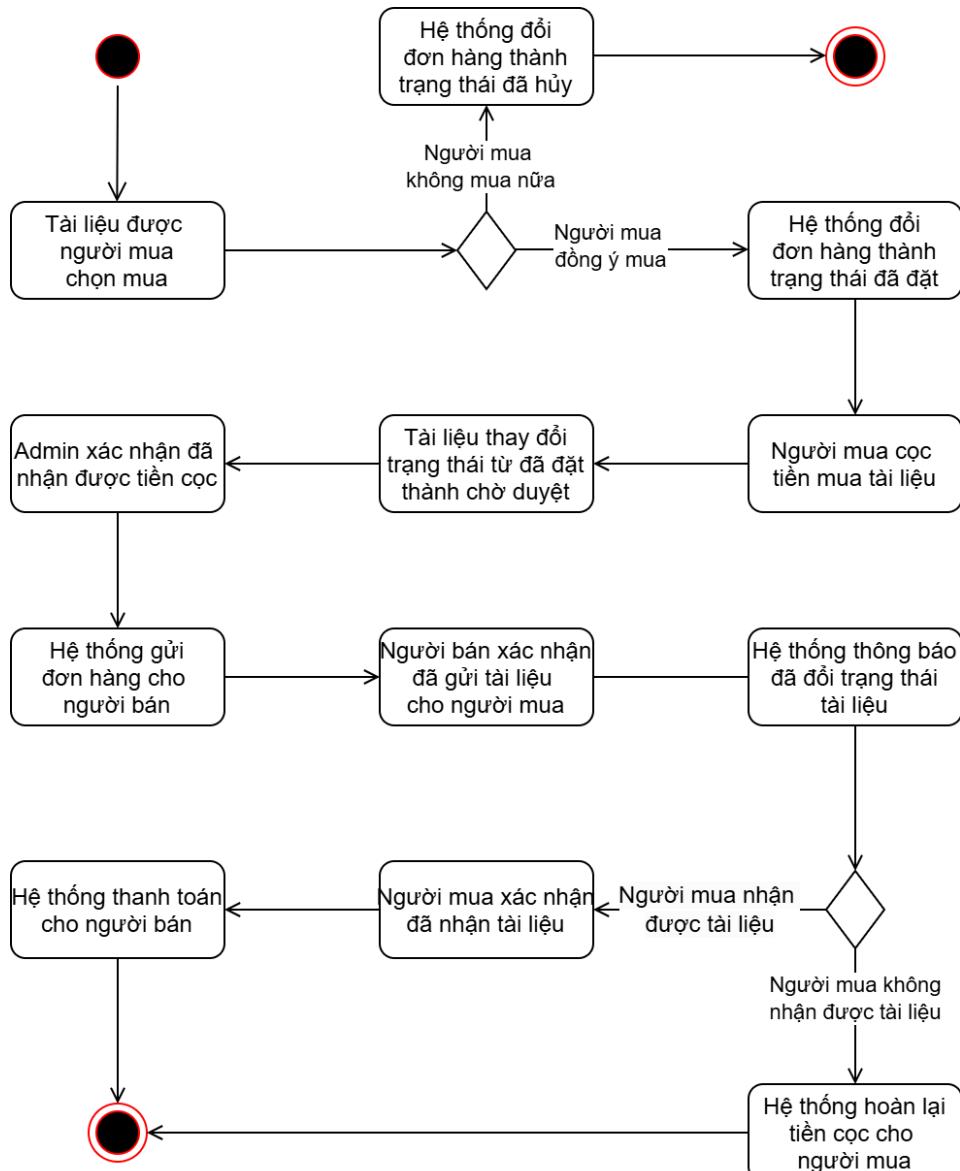


Hình 25: Activity diagram thuê và trả tài liệu

Mô tả Lược đồ Hoạt động: Thuê và Trả Tài Liệu

1. **Khởi đầu:** Chủ sách tạo bài đăng sách thành công.
2. **Kiểm tra trạng thái sách:**
 - a. **Nếu sách không được thuê:** Kết thúc hoạt động.
 - b. **Nếu sách được thuê bởi người thuê:**
 - i. Hệ thống tạo đơn hàng.
 - ii. Thay đổi trạng thái sách từ *sẵn có* sang *đã cho thuê*.
3. **Đóng tiền cọc thuê sách:** Người thuê tiến hành đóng tiền cọc:
 - a. **Nếu không đóng cọc:**
 - i. Đơn hàng bị hủy.
 - ii. Sách quay về trạng thái *sẵn có*.
 - b. **Nếu đóng cọc thành công:**
 - i. Hệ thống thay đổi trạng thái đơn hàng từ *đã đặt hàng* sang *đã cọc sách*.
4. **Nhận sách:** Người thuê nhận sách và xác nhận nhận được sách. Nếu xác nhận thành công, hệ thống thay đổi trạng thái đơn hàng từ *đã cọc sách* sang *đã nhận sách*.
5. **Trả sách:** Người thuê tiến hành trả sách:
 - a. **Nếu không trả sách:**
 - i. Quản trị viên không hoàn trả tiền cọc cho người thuê.
 - ii. Quản trị viên tiến hành thanh toán tiền thuê sách cho chủ sách.
 - b. **Nếu trả sách:**
 - i. **Trả sách đúng hạn:**
 1. Tiến hành bước đánh giá tài liệu.
 - ii. **Trả sách trễ hạn:**
 1. Hệ thống tính phí trễ hạn.
 2. Tiến hành bước đánh giá tài liệu.
6. **Thanh toán cuối cùng:** Quản trị viên thanh toán tiền thuê sách cho chủ sách và hoàn tiền (nếu có) cho người thuê.
7. **Kết thúc hoạt động.**

4.10.4 Người dùng mua tài liệu



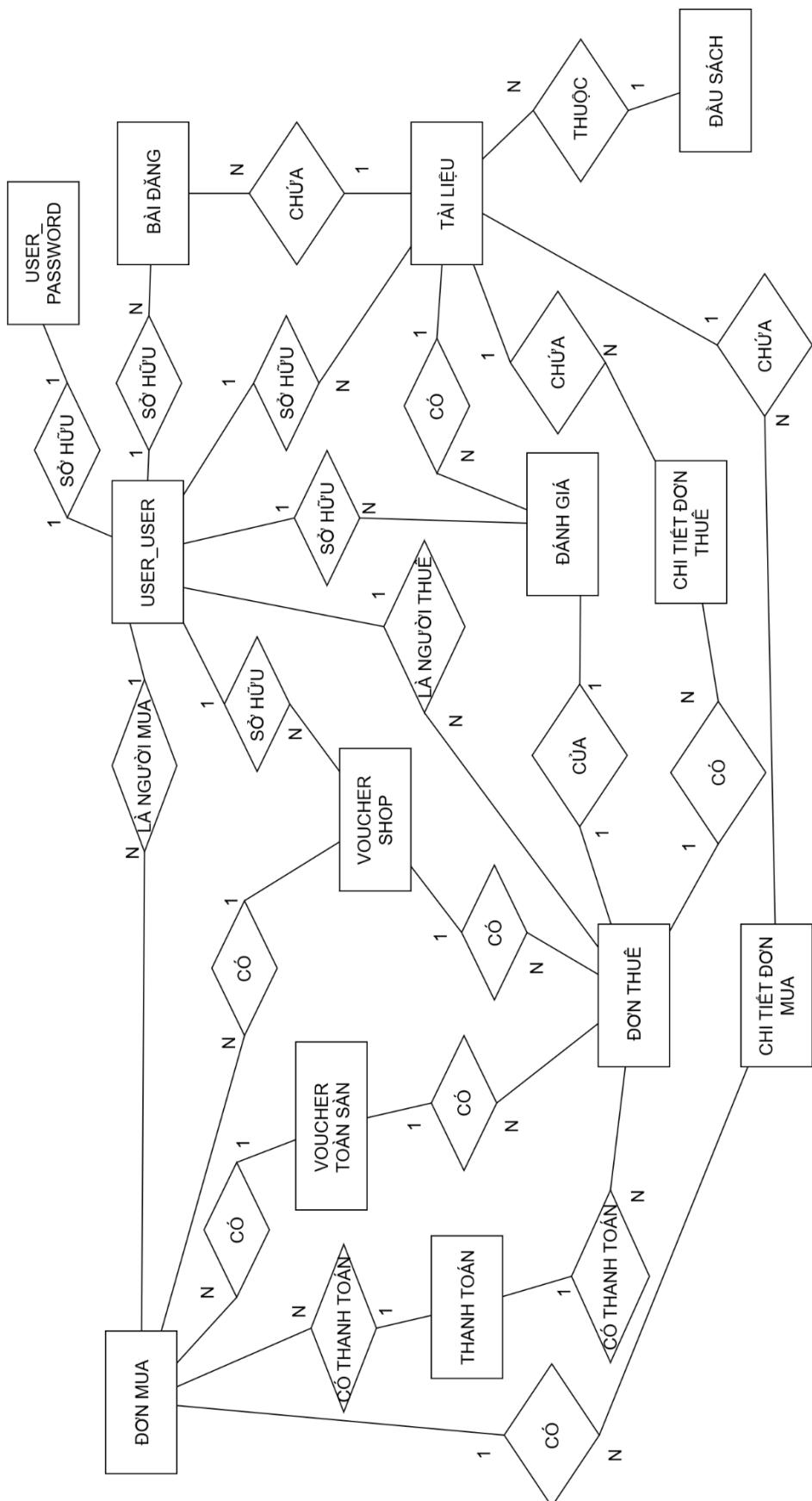
Hình 26: Activity diagram Người dùng mua tài liệu

- Khởi đầu:** Người mua chọn tài liệu muốn mua.
- Xác nhận ý định mua:** Hệ thống hỏi người mua có tiếp tục mua hay không:
 - Nếu người mua không đồng ý mua:** Hệ thống hủy đơn hàng, và quy trình kết thúc.
 - Nếu người mua đồng ý mua:** Hệ thống chuyển trạng thái đơn hàng thành "Đã đặt hàng".
- Đặt cọc mua tài liệu:** Người mua tiến hành đặt cọc cho tài liệu.
 - Nếu người mua không đặt cọc:** Hệ thống hủy đơn hàng.
 - Nếu đặt cọc thành công:** Hệ thống chuyển trạng thái tài liệu từ "Đã đặt" sang "Chờ duyệt".
- Xác nhận bởi Admin:** Admin kiểm tra và xác nhận đã nhận được tiền cọc từ người mua.
- Gửi tài liệu cho người mua:**

- a. Hệ thống gửi đơn hàng tới người bán.
 - b. Người bán xác nhận đã gửi tài liệu cho người mua.
 - c. Hệ thống thông báo rằng trạng thái tài liệu đã được thay đổi.
6. **Xác nhận nhận tài liệu:** Người mua xác nhận rằng họ đã nhận được tài liệu:
- a. **Nếu người mua nhận được tài liệu:** Quy trình tiếp tục tới bước thanh toán.
 - b. **Nếu người mua không nhận được tài liệu:** Hệ thống hoàn lại tiền cọc cho người mua.
7. **Thanh toán:** Hệ thống thực hiện thanh toán:
- a. Thanh toán tiền mua tài liệu cho người bán.
 - b. Nếu có vấn đề xảy ra, tiền cọc sẽ được hoàn lại (nếu áp dụng).
8. Kết thúc hoạt động

4.11 Lược đồ quan hệ cơ sở dữ liệu (ERD)

4.11.1 Lược đồ ERD vắn tắt



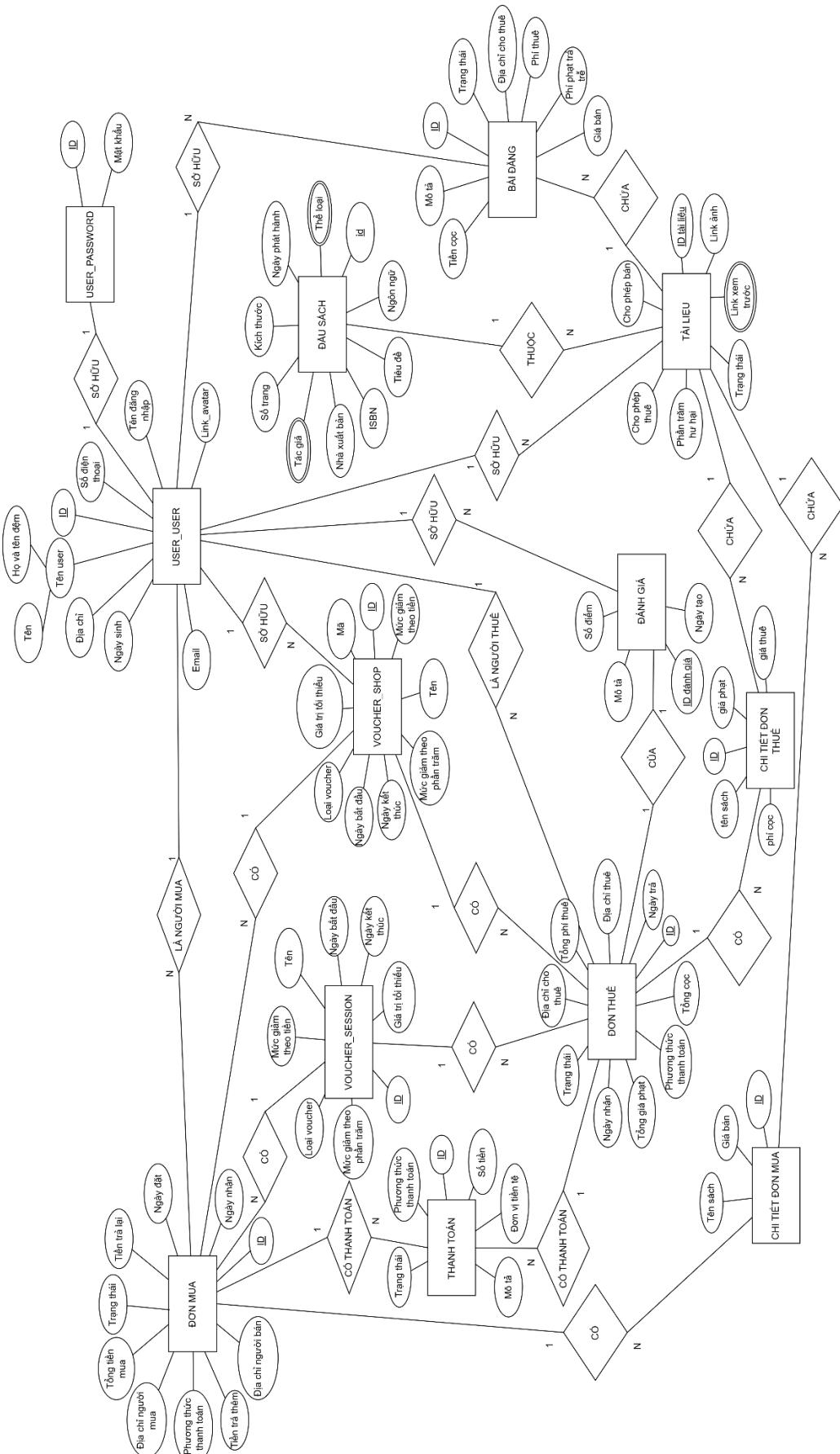
Hình 27: Lược đồ ERD vắn tắt

Lược đồ quan hệ cơ sở dữ liệu gồm các entity: user_user, user_password, bài đăng, tài liệu, đầu sách, đánh giá, đơn thuê, chi tiết đơn thuê, thanh toán, đơn mua, chi tiết đơn mua, voucher toàn sàn và voucher shop.

- **Người dùng sở hữu bài đăng:** Mỗi quan hệ này chứa thông tin về bài đăng mà người dùng có. Một người dùng có thể sở hữu nhiều bài đăng, nhưng một bài đăng chỉ thuộc về một người dùng, vì vậy mối quan hệ này là **một - nhiều**.
- **Người dùng sở hữu tài liệu:** tương tự với bài đăng, vì vậy mối quan hệ này là **một - nhiều**.
- **Tài liệu chứa bài đăng:** Mỗi bài đăng chỉ có một tài liệu và một tài liệu có thể được chứa trong nhiều bài đăng, vì vậy mối quan hệ này là **một - nhiều**.
- **Tài liệu thuộc đầu sách:** Mỗi quan hệ này chứa thông tin về tài liệu thuộc đầu sách nào. Một tài liệu chỉ thuộc một đầu sách, nhưng một đầu sách có thể chứa nhiều tài liệu, vì vậy mối quan hệ này là **nhiều - một**.
- **Người dùng viết đánh giá:** Mỗi quan hệ này chứa thông tin về các đánh giá mà người dùng viết. Một người dùng có thể viết nhiều đánh giá, nhưng một đánh giá chỉ do một người dùng viết, vì vậy mối quan hệ này là **một - nhiều**.
- **Đánh giá thuộc đơn thuê:** Mỗi quan hệ này chứa thông tin về đánh giá thuộc đơn thuê nào. Một đánh giá chỉ thuộc một đơn thuê, nhưng một đơn thuê có thể có nhiều đánh giá, vì vậy mối quan hệ này là **nhiều - một**.
- **Chi tiết đơn thuê thuộc đơn thuê:** Mỗi quan hệ này chứa thông tin chi tiết đơn thuê liên quan đến đơn thuê nào. Một chi tiết đơn thuê chỉ thuộc một đơn thuê, nhưng một đơn thuê có thể có nhiều chi tiết, vì vậy mối quan hệ này là **nhiều - một**.
- **Thanh toán liên kết với đơn thuê hoặc đơn mua:** Mỗi quan hệ này chứa thông tin thanh toán cho đơn thuê hoặc đơn mua. Một thanh toán có thể liên kết với một đơn thuê hoặc đơn mua, nhưng một đơn thuê hoặc đơn mua chỉ liên kết với một thanh toán duy nhất, vì vậy mối quan hệ này là **một - nhiều**.
- **Chi tiết đơn mua thuộc đơn mua:** Mỗi quan hệ này chứa thông tin chi tiết đơn mua thuộc về đơn mua nào. Một chi tiết đơn mua chỉ thuộc một đơn mua, nhưng một đơn mua có thể có nhiều chi tiết, vì vậy mối quan hệ này là **một - nhiều**.
- **Người dùng áp dụng voucher shop:** Mỗi quan hệ này chứa thông tin về việc người dùng sử dụng voucher shop. Một người dùng có thể sử dụng nhiều voucher shop, nhưng một voucher shop có thể được sử dụng bởi nhiều người dùng, vì vậy mối quan hệ này là **một - nhiều**.
- **Chi tiết đơn thuê chứa tài liệu:** Một đơn thuê có thể có nhiều sản phẩm, tức là một chi tiết đơn thuê chứa nhiều tài liệu. Nhưng một tài liệu cũng có thể chứa nhiều chi tiết đơn thuê. Vậy nên mối quan hệ này là **nhiều - nhiều**.
- **Chi tiết đơn mua chứa tài liệu:** Tương tự với chi tiết đơn thuê. Vậy nên mối quan hệ này là **nhiều - nhiều**.
- **Người dùng sở hữu mật khẩu:** Mỗi người dùng có một mật khẩu duy nhất, mật khẩu chỉ thuộc về một người dùng duy nhất, vì vậy mối quan hệ này là **một - một**.
- **Người dùng có đơn mua:** Mỗi người dùng có nhiều đơn mua và một đơn mua chỉ có duy nhất một người mua đơn mua đó, vậy nên mối quan hệ này là **một - nhiều**.
- **Người dùng có đơn thuê:** tương tự với đơn mua, vì vậy nên mối quan hệ này là **một - nhiều**.

- **Đơn mua có voucher toàn sàn:** Mỗi đơn mua chỉ được áp dụng một voucher toàn sàn, nhưng một voucher toàn sàn có thể được áp dụng bởi nhiều đơn mua khác. Vậy nên mối quan hệ này là **nhiều – một**.
- **Đơn mua có voucher theo shop:** Tương tự với voucher toàn sàn, vậy nên mối quan hệ này là **nhiều – một**.
- **Đơn thuê có voucher toàn sàn:** Tương tự với đơn mua, mối quan hệ này là **nhiều – một**.
- **Đơn thuê có voucher theo shop:** Tương tự với đơn mua, mối quan hệ này là **nhiều – một**.

4.11.2 Lược đồ ERD đầy đủ

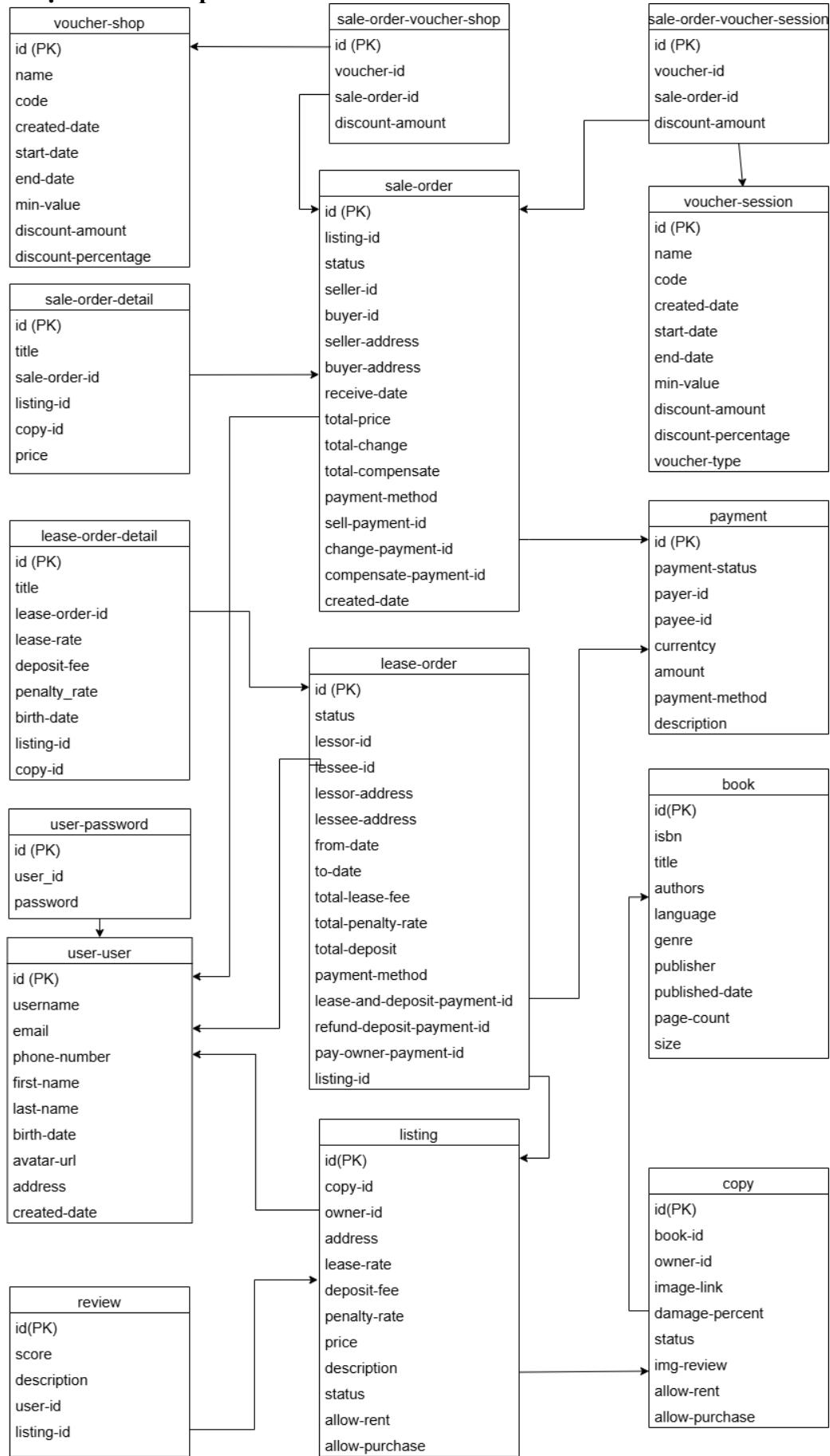


Hình 28: ERD cho toàn hệ thống

Vì ERD này hơi to nên tác giả sẽ in 1 bảng A3 kẹp vào trong bài báo cáo, dưới đây là mô tả cụ thể từng thuộc tính của ERD:

- **User_user** chứa thông tin cá nhân của người dùng như id, họ tên, địa chỉ, ngày sinh, số điện thoại, email, người dùng name.
- **User_password** chứa thông tin mật khẩu của user và id
- **Bài đăng** chứa các trường id, mô tả, phí cọc, phí thuê, mô tả, trạng thái, phí phạt trả trễ, giá bán.
- **Tài liệu** chứa các trường id, phần trăm hư hại, trạng thái, hình ảnh tài liệu, id, các hình ảnh xem trước, cho phép thuê, cho phép bán
- **Đầu sách** chứa các trường id, thể loại, ngày phát hành, kích thước, số trang, tác giả, nhà xuất bản, isbn, tiêu đề, ngôn ngữ
- **Đánh giá** chứa các trường id, ngày tạo, số điểm, mô tả
- **Đơn thuê** chứa các trường id, ngày nhận, ngày trả, địa chỉ cho thuê, tổng cọc, phương thức thanh toán, địa chỉ cho thuê và trạng thái đơn hàng.
- **Đơn mua** chứa các trường id, trạng thái, tiền trả lại, ngày đặt, ngày nhận, tổng tiền mua, địa chỉ người bán, tiền trả thêm, phương thức thanh toán, địa chỉ người bán.
- **Chi tiết đơn thuê** chứa id, tên sách, giá thuê, giá phạt và phí cọc.
- **Chi tiết đơn mua** chứa id, tên sách, giá bán.
- **Thanh toán** chứa id, trạng thái, phương thức thanh toán, mô tả, đơn vị tiền tệ, số tiền.
- **Voucher_session** tức là voucher toàn sàn chứa id, giá trị tối thiểu, mức giảm theo phần trăm, loại voucher, mức giảm theo tiền, tên, ngày bắt đầu, ngày kết thúc.
- **Voucher_shop** chứa voucher của người bán chứa id, giá trị tối thiểu, mức giảm theo phần trăm, loại voucher, mức giảm theo tiền, tên, ngày bắt đầu, ngày kết thúc.

4.11.3 Lược đồ ERD implement



Hình 29: Lược đồ ERD implement

1. Các thực thể chính và quan hệ

- **User-user:** Lưu thông tin người dùng bao gồm username, email, số điện thoại, họ tên, ngày sinh, avatar, và ngày tạo tài khoản.
- **User-password:** Lưu trữ mật khẩu gắn liền với từng tài khoản người dùng.
- **Book:** Lưu thông tin chi tiết của sách như ISBN, tiêu đề, tác giả, ngôn ngữ, thể loại, nhà xuất bản, ngày phát hành, số trang, và kích thước.
- **Copy:** Mỗi sách có thể có nhiều bản sao (copy) với các thuộc tính như hình ảnh, trạng thái hư hỏng, và các cờ để xác định bản sao đó có được cho thuê hoặc bán không.
- **Listing:** Danh sách các bản sao sách sẵn sàng để cho thuê hoặc bán với thông tin về giá thuê, giá bán, phí đặt cọc, tỷ lệ phạt, mô tả, và trạng thái.
- **Review:** Đánh giá người dùng về các bản sách đã thuê hoặc mua, bao gồm điểm số, mô tả, và thông tin người đánh giá.

2. Quản lý đơn hàng

- **Sale-order:** Đại diện cho đơn hàng mua sách, bao gồm thông tin người mua, người bán, địa chỉ giao hàng, tổng giá trị đơn hàng, phương thức thanh toán, và các khoản bù trừ.
- **Sale-order-detail:** Chi tiết từng mục trong đơn hàng, liên kết với bản sao sách và giá bán.
- **Lease-order:** Đại diện cho đơn hàng thuê sách, bao gồm thông tin người cho thuê, người thuê, thời gian thuê, phí thuê, phí phạt, tiền đặt cọc, và phương thức thanh toán.
- **Lease-order-detail:** Chi tiết từng mục thuê, bao gồm tiêu đề sách, phí thuê, phí đặt cọc, ngày sinh, và liên kết đến bản sao sách.

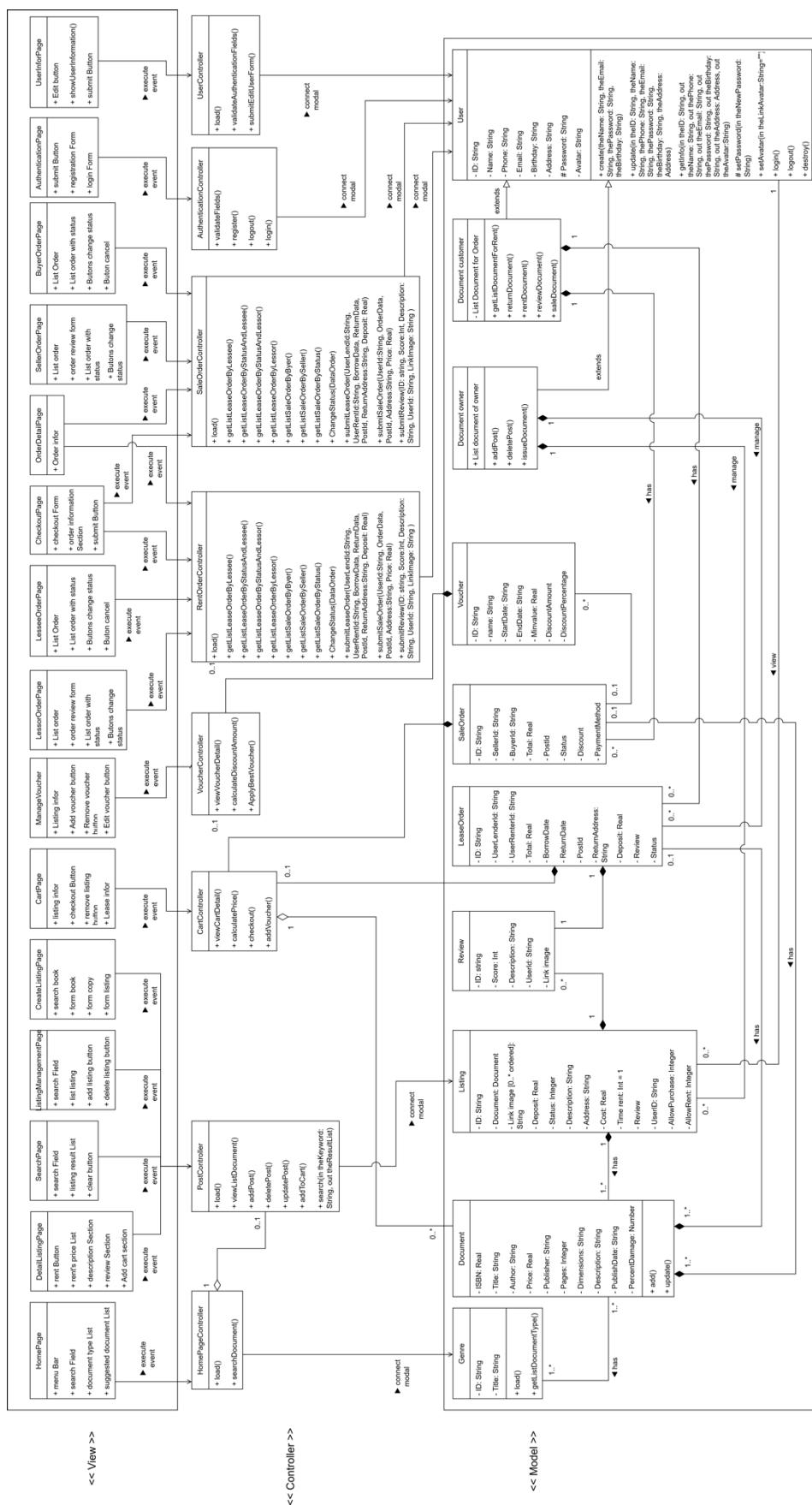
3. Voucher và Khuyến mãi

- **Voucher-shop:** Lưu thông tin voucher có sẵn trong cửa hàng như tên, mã, ngày bắt đầu, ngày kết thúc, giá trị giảm, và tỷ lệ giảm.
- **Sale-order-voucher-shop:** Áp dụng voucher vào đơn hàng mua.
- **Voucher-session:** Tương tự voucher-shop nhưng được dùng trong các chương trình khuyến mãi ngắn hạn.
- **Sale-order-voucher-session:** Áp dụng voucher session cho đơn hàng.

4. Thanh toán

- **Payment:** Quản lý các giao dịch thanh toán bao gồm trạng thái, người thanh toán, người nhận thanh toán, số tiền, loại tiền tệ, phương thức thanh toán, và mô tả chi tiết.

4.12 Class diagram



Hình 30: Class diagram cho hệ thống

Class diagram của hệ thống được triển khai qua ba lớp view, controller, model:

- Lớp view chứa các trang front-end của hệ thống: Trang homepage, thanh navbar, danh sách đề xuất, các thông tin giới thiệu khác. Trang chi tiết bài đăng chứa thông tin chi tiết bài đăng và nút “đặt thuê” để thêm bài đăng vào giỏ hàng. Trang tìm kiếm có chứa sidebar thể loại, thanh tìm kiếm và danh sách tìm kiếm, nút clear để loại bỏ lọc danh sách. Trang quản lý bài đăng có tìm bài đăng, danh sách bài đăng của người dùng, nút thêm bài đăng, nút xoá bài đăng. Trang tạo bài đăng có search book và các form cần thiết để tạo bài đăng. Trang giỏ hàng có nút xoá bài đăng và thông tin bài đăng. Trang tạo đơn hàng có chứa cá thông tin để tạo đơn hàng và nút đặt thuê. Các trang quản lý đơn thuê và đơn cho thuê chứa danh sách đơn hàng và các nút thay đổi trạng thái đơn hàng. Trang xác thực người dùng là nhóm trang chứa chức năng đăng nhập, đăng xuất, đăng ký. Trang quản lý thông tin người dùng chứa thông tin người dùng và nút edit.
- Lớp Controller là lớp giữa giao tiếp giữa view và model dựa trên các api lớp này chứa các api đưa thông tin từ view xuống model và lấy thông tin từ model lên cho view. Gồm các controller: HomePageController, PostController, CartController, OrderController, AuthenticationController, UserController
- Lớp model chứa các lớp thực hiện tác vụ logic nghiệp vụ. Bao gồm các model: thể loại, tài liệu, đánh giá đơn hàng, đơn hàng, bài đăng, người dùng.

CHƯƠNG 5: HIỆN THỰC BACK-END

5.1 Hiện thực database

Bước 1: Database được config trong thư mục application-dev.yaml

spring:

```
datasource:  
  url: jdbc:postgresql://localhost:5432/base?currentSchema=public  
  username: postgres  
  password: postgres  
  driver-class-name: org.postgresql.Driver
```

Bước 2: Xem các file sql trong thư mục init-db-scripts

```
└── 00-common_area.sql  
└── 00-common_language.sql  
└── 00-common.sql  
└── 10-user.sql  
└── book.sql  
└── lease.sql  
└── payment.sql  
└── sale.sql  
└── voucher.sql
```

Các script để init database được lưu trong thư mục init-db-scripts:

- **00-common_area.sql:** Chứa script khởi tạo bảng tĩnh thành, là bảng common_area và các câu lệnh insert dữ liệu mẫu.
- **00-common_language.sql:** Chứa script khởi tạo các bảng liên quan đến ngôn ngữ, là bảng language và các câu lệnh insert dữ liệu mẫu.
- **00-common.sql:** Chứa script thiết lập các bảng liên quan đến giới tính và khu vực địa lý, bao gồm bảng common_area_types, common_locale_classification, common_sex
- **10-user.sql:** (Đã mô tả trong yêu cầu ban đầu) Chứa script SQL khởi tạo bảng user - người dùng, bảng user-password, và các câu lệnh insert dữ liệu người dùng.
- **book.sql:** (Đã mô tả trong yêu cầu ban đầu) Chứa script SQL khởi tạo bảng và dữ liệu các bảng liên quan đến sách và bài đăng: bảng book, bảng copy, bảng listing.
- **lease.sql:** (Đã mô tả trong yêu cầu ban đầu) Chứa script khởi tạo bảng và dữ liệu ban đầu cho các bảng liên quan đến đơn hàng: lease-order, lease-order-detail, và review.
- **payment.sql:** (Đã mô tả trong yêu cầu ban đầu) Chứa script khởi tạo bảng và dữ liệu ban đầu cho bảng payment.
- **sale.sql:** Chứa script khởi tạo bảng và dữ liệu ban đầu cho các bảng liên quan đến mua bán sách, bao gồm: bảng sale, sale_order, và sale_order_detail.
- **voucher.sql:** Chứa script khởi tạo bảng và dữ liệu ban đầu cho hệ thống voucher, bao gồm bảng voucher_session, voucher_shop và các câu lệnh insert mẫu.

Bước 3: Sau khi chạy lệnh sau thì database sẽ được khởi tạo

docker compose -f local.compose.yaml up -d

5.2 Phân tích source-code back-end

Sử dụng <https://github.com/structure-codes/cli> để lấy cấu trúc source dễ nhìn đẹp mắt được như sau:

```
├── api.iml
├── build.gradle
├── dockerfile
└── gradle
    └── wrapper
        ├── gradle-wrapper.jar
        └── gradle-wrapper.properties
├── gradlew
├── gradlew.bat
├── HELP.md
├── init-db-scripts
├── local.compose.yaml
├── settings.gradle
└── src
    ├── main
    │   └── java
    │       └── base
    │           └── api
    │               ├── ApiApplication.java
    │               ├── authorization
    │               ├── book
    │               │   ├── controller
    │               │   ├── dto
    │               │   ├── entity
    │               │   ├── exception
    │               │   └── job
    │               │       └── LeaseOrderJobs.java
    │               ├── mapper
    │               ├── repository
    │               └── service
    │               └── common
    │                   ├── exception
    │                   └── internal
    │                       ├── controller
    │                       ├── entity
    │                       ├── mapper
    │                       └── repository
    │               └── image
    │                   └── controller
    │               └── payment
    │                   ├── controller
    │                   ├── dto
    │                   ├── entity
    │                   ├── mapper
    │                   ├── repository
    │                   └── service
    │               └── system
    │                   ├── cache
    │                   ├── database
    │                   ├── jackson
    │                   ├── security
    │                   └── storage
    │               └── user
    │                   └── util
    │                       └── api
    └── resources
        ├── application-aws.yaml
        └── application-dev.yaml
```

```

    └── application-prod.yaml
    └── application.yaml
  test
    └── java
      └── base
        └── api
          └── ApiApplicationTests.java

```

5.2.1 Thư mục gốc

Thư mục gốc chứa các file cấu hình và công cụ hỗ trợ build và chạy dự án:

- **api.iml**: File cấu hình IntelliJ IDEA cho dự án.
- **build.gradle**: Tập tin cấu hình Gradle để quản lý dependencies và tasks.
- **dockerfile**: Định nghĩa cách build và chạy ứng dụng trong container Docker.
- **gradle/**: Chứa các file hỗ trợ Gradle:
 - gradle-wrapper.jar: File thực thi Gradle.
 - gradle-wrapper.properties: File cấu hình Gradle wrapper.
- **gradlew, gradlew.bat**: Script để chạy Gradle trên các hệ điều hành khác nhau.
- **HELP.md**: File mô tả hoặc hướng dẫn sử dụng ứng dụng.
- **local.compose.yaml**: File cấu hình Docker Compose để chạy các container liên quan (database, service, ...).
- **settings.gradle**: File định nghĩa các module hoặc dự án con trong Gradle.
- **init-db-scripts/**: Thư mục chứa script khởi tạo cơ sở dữ liệu.

5.2.2 Thư mục chính

Chứa mã nguồn chính của dự án, được chia làm hai phần:

- **main/**: Mã nguồn chính của ứng dụng.
- **test/**: Mã nguồn kiểm thử.

5.2.2.1 Thư mục main

Tổ chức mã nguồn chính theo kiến trúc phân tầng:

- **base.api**: Namespace chính của ứng dụng.
 - **ApiApplication.java**: File entry point khởi chạy ứng dụng Spring Boot.
 - **authorization/**: Chứa mã nguồn quản lý xác thực và phân quyền.
 - **book/**: Module quản lý sách và các tính năng liên quan:
 - controller/: Xử lý yêu cầu HTTP và gửi phản hồi.
 - dto/: Chứa các lớp Data Transfer Object để truyền dữ liệu giữa client và server.
 - entity/: Mô hình dữ liệu tương ứng với cơ sở dữ liệu.
 - exception/: Xử lý các lỗi và ngoại lệ liên quan đến sách.
 - job/: Chứa các tác vụ định kỳ như LeaseOrderJobs.java.
 - mapper/: Chuyển đổi giữa entity và dto.
 - repository/: Quản lý truy vấn cơ sở dữ liệu.
 - service/: Chứa logic nghiệp vụ của module.

- **common/**: Chứa các thành phần chung cho toàn bộ hệ thống:
 - **exception/**: Xử lý lỗi chung.
 - **internal/**: Các module chỉ dùng nội bộ, gồm:
 - controller/, entity/, mapper/, repository/: Tổ chức tương tự module book.
- **image/**: Module xử lý hình ảnh:
 - controller/: Xử lý upload và tải hình ảnh.
- **payment/**: Module quản lý thanh toán:
 - Tương tự như book, có đầy đủ controller, dto, entity, mapper, repository, service.
- **system/**: Chứa các thành phần hỗ trợ hệ thống:
 - **cache/**: Quản lý cache.
 - **database/**: Cấu hình cơ sở dữ liệu.
 - **jackson/**: Cấu hình bộ chuyển đổi JSON.
 - **security/**: Cấu hình bảo mật.
 - **storage/**: Quản lý lưu trữ file.
- **user/**: Module quản lý người dùng.
- **util/**: Các tiện ích chung.
 - **api/**: Tiện ích cho các thành phần trong hệ thống.
 - **resources/**: Chứa các file cấu hình và tài nguyên của ứng dụng:
- **application.yaml, application-{env}.yaml**: Cấu hình ứng dụng cho các môi trường (AWS, dev, prod).
- **File tĩnh**: Các template, file dữ liệu tĩnh (nếu có).

5.2.2.2 Thư mục test

Chứa các bài kiểm thử cho ứng dụng. Gồm có file **ApiApplicationTests.java**: File kiểm thử ứng dụng Spring Boot.

5.3 Hiện thực api

Các api trong dự án được tiến hành xây dựng thông qua các lớp: controller, service, mapper, repository, dto. Mỗi lớp có chức năng, nhiệm vụ riêng.

5.3.1 Lớp Controller và Dto

Lớp Controller: là thành phần quan trọng của một api. Lớp này xử lý các yêu cầu HTTP, hình thành nên api mà front-end có thể dùng nó để tương tác với back-end. Lớp này sử dụng và tương tác với lớp server (ở dưới) để thực hiện các nghiệp vụ cơ bản của dự án. Một số annotation được sử dụng trong lớp này:

- **@RestController**: Đánh dấu lớp là một RESTful controller.
- **@RequestMapping**: Ánh xạ tất cả các yêu cầu HTTP tới đường dẫn cho lớp controller tương ứng.
- **@PathVariable** để nhận tham số, **@RequestBody** để nhận thông tin body, **@RequestParam** để nhận thông số đầu vào cho mỗi một đầu vào của api.

Sau đây là Cây thư mục lớp controller, dto của book:

```

— controller
  └── BookController.java
  └── BooksController.java
  └── CopyController.java
  └── GenreController.java
  └── LeaseOrderController.java
  └── ListingController.java
  └── ReviewController.java
  └── SaleOrderController.java
  └── VoucherSessionController.java
  └── VoucherShopController.java
— dto
  └── BookDto.java
  └── BooksDto.java
  └── CopyDto.java
  └── EditStatusListingRequest.java
  └── ErrorResponse.java
  └── GenreDto.java
  └── LeaseOrderAdmin.java
  └── LeaseOrderCreateRequest.java
  └── LeaseOrderDetailDto.java
  └── LeaseOrderDto.java
  └── LeaseOrderDtoDetail.java
  └── ListingDetailDto.java
  └── ListingDto.java
  └── ListingExtendedDto.java
  └── ReviewDto.java
  └── SaleOrderCreateRequest.java
  └── SaleOrderCreateRequestFromLease.java
  └── SaleOrderDetailDto.java
  └── SaleOrderDetailManagementDto.java
  └── SaleOrderDto.java
  └── SaleOrderVoucherSessionDto.java
  └── SaleOrderVoucherShopDto.java
  └── search
    └── CopyDtoUpload.java
    └── LeaseOrderUpdateRequest.java
    └── ListingSearchByOwnerAndNameDto.java
    └── ListingSearchDto.java
  └── VoucherSessionDto.java
  └── VoucherShopDto.java

```

Lớp DTO chứa các mẫu dữ liệu sẵn để truyền dữ liệu giữa các lớp, tầng của api. DTO tác giả xây dựng ở đây được sử dụng để chuyển dữ liệu từ lớp service lên lớp controller, và ngược lại. DTO được xây dựng ở đây nhằm mục đích tách biệt khỏi thực thể entity, khi cấu trúc thực thể thay đổi thì không ảnh hưởng đến giao diện api.

5.3.2 Lớp Entity

Các lớp entity đại diện cho các bảng trong cơ sở dữ liệu. Mỗi instance của lớp này sẽ tương ứng với một dòng trong bảng.

Một số annotation sử dụng trong lớp này giúp xác định cách ánh xạ và quản lý các thực thể trong cơ sở dữ liệu:

- **@Entity:** Được sử dụng để đánh dấu lớp này sẽ được ánh xạ đến một bảng trong cơ sở dữ liệu.
- **@Table:** Dùng để xác định bảng trong cơ sở dữ liệu mà lớp này sẽ ánh xạ tới.
- **@Id:** Đánh dấu primary key của bảng.
- **@GeneratedValue:** Được sử dụng để chỉ định chiến lược tạo giá trị tự động cho khóa chính.

- `@Column`: Được sử dụng để ánh xạ một thuộc tính của lớp tới một cột trong bảng. Có thể chỉ định tên cột, kiểu dữ liệu, và các thuộc tính khác.

Sau đây là Cây thư mục lớp entity của book:

```
— entity
  — Book.java
  — Copy.java
  — Genre.java
  — LeaseOrder.java
  — LeaseOrderDetail.java
  — Listing.java
  — Review.java
  — SaleOrder.java
  — SaleOrderDetail.java
  — SaleOrderVoucherSession.java
  — SaleOrderVoucherShop.java
  — VoucherSession.java
  — VoucherShop.java
```

5.3.3 Lớp Mapper và repository

Lớp mapper hay còn gọi là lớp chuyển đổi, lớp này được sử dụng để chuyển đổi dữ liệu giữa các lớp entity với các lớp DTO. Bởi vì lớp DTO đại diện cho cấu trúc dữ liệu trao đổi qua api, còn entity đại diện cho cấu trúc dữ liệu trong cơ sở dữ liệu. Nên lớp mapper này được sinh ra để nâng cao tính bảo mật và tính mềm dẻo của ứng dụng khi có sự thay đổi trong các tầng thì không phải thay đổi toàn bộ api. Đồng thời lớp này sẽ giúp lớp controller và service trở nên gọn gàng và dễ hiểu hơn.

Trong Spring Boot, lớp Repository là một thành phần quan trọng dùng để thao tác với cơ sở dữ liệu. Repository là một phần của Spring Data JPA, cung cấp các phương thức để thực hiện các thao tác CRUD (Create, Read, Update, Delete) trên các thực thể (entities). Lớp Repository cung cấp một cách tiếp cận đơn giản để sử dụng để truy cập cơ sở dữ liệu, thông qua các phương thức truy vấn đã được định nghĩa sẵn hoặc tùy chỉnh. Nó giúp tách biệt logic truy cập dữ liệu khỏi logic nghiệp vụ, làm cho mã nguồn dễ bảo trì và dễ mở rộng hơn. Trong dự án này tác giả xây dựng mỗi repository bằng cách tạo interface kế thừa từ 'JpaRepository', điều này ngầm định rằng lớp này là một lớp repository.

Sau đây là Cây cấu trúc thư mục job, mapper, repository của book:

```
— job
  — LeaseOrderJobs.java
— mapper
  — BookMapper.java
  — CopyMapper.java
  — GenreMapper.java
  — LeaseOrderDetailMapper.java
  — LeaseOrderMapper.java
  — ListingMapper.java
  — ReviewMapper.java
  — SaleOrderDetailMapper.java
  — SaleOrderMapper.java
  — SaleOrderVoucherSessionMapper.java
  — SaleOrderVoucherShopMapper.java
  — VoucherSessionMapper.java
  — VoucherShopMapper.java
— repository
  — BookRepository.java
  — CopyRepository.java
  — GenreRepository.java
  — LeaseOrderDetailRepository.java
  — LeaseOrderRepository.java
  — ListingRepository.java
  — ReviewRepository.java
```

```

├── SaleOrderDetailRepository.java
├── SaleOrderRepository.java
├── SaleOrderVoucherSessionRepository.java
└── SaleOrderVoucherShopRepository.java
├── VoucherSessionRepository.java
└── VoucherShopRepository.java

```

Lớp Job trong dự án này chứa các hàm chạy tự động trong dự án, giúp update cơ sở dữ liệu bởi những nghiệp vụ riêng.

5.3.4 Lớp Service

Lớp Service đóng vai trò trung gian giữa các lớp Controller và Repository, xử lý các logic nghiệp vụ (business logic) của dự án. Lớp Service chịu trách nhiệm thực hiện các tác vụ như xử lý dữ liệu, gọi các phương thức của Repository để tương tác với cơ sở dữ liệu, và đảm bảo rằng các quy tắc nghiệp vụ được áp dụng. Một số annotation trong service:

- `@Service`: có chức năng đánh dấu một lớp là một "Service" trong Spring. Lớp này sẽ chứa logic nghiệp vụ.
- `@Transactional` có chức năng đảm bảo rằng các phương thức được gọi bên trong lớp Service sẽ được thực thi trong một giao dịch (transaction). Nên nếu có bất kỳ lỗi nào xảy ra trong quá trình thực thi, toàn bộ giao dịch sẽ được rollback để đảm bảo tính toàn vẹn của dữ liệu.
- `@Autowired` có chức năng: Được sử dụng để tự động tiêm (inject) các dependency vào lớp. Giúp quản lý các phụ thuộc mà không cần viết mã khởi tạo phức tạp.

Sau đây là Cây cấu trúc thư mục service của book

```

└── service
    ├── BookService.java
    ├── BooksService.java
    ├── CopyService.java
    ├── GenreService.java
    ├── LeaseOrderDetailService.java
    ├── LeaseOrderService.java
    ├── ListingService.java
    ├── ReviewService.java
    ├── SaleOrderDetailService.java
    ├── SaleOrderService.java
    ├── SaleOrderVoucherSessionService.java
    ├── SaleOrderVoucherShopService.java
    ├── VoucherSessionService.java
    └── VoucherShopService.java

```

5.4 Api xây dựng trong dự án

Stt	Nhóm api	Tên api	Endpoint	Loại api
1	user	Đăng ký	api/user/register	post
2		Đăng nhập	api/user/login	post
3		Lấy thông tin user	api/user/myInfo	get
4		Lấy user theo id	api/user/{id}	get
5		Sửa thông tin user	api/user/{id}	put
6	genre	Lấy danh sách thể loại	api/genre	get
7		Tạo thể loại	api/genre	post
8		Lấy chi tiết thể loại	api/genre/{id}	get

9		Sửa thể loại	api/genre/{id}	put
10		Xóa thể loại	api/genre/{id}	delete
11	book	Lấy tất cả book	api/book	get
12		Lấy book theo id	api/book/{id}	get
13		Tìm book theo title	api/book/search/{title}	get
14		Tạo book	api/book	post
15		Sửa book	api/book/{id}	put
16		Xóa book	api/book/{id}	delete
17	listing	Tìm kiếm listing	http://api/listing/search?size={size}&genre={genre}&page={page}&title={title}&allowRent={allowRent}&allowPurchase={allowPurchase}	get
18		Lấy listing theo id	api/listing/detailListing/{id}	get
19		Tạo listing	api/listing	post
20		Lấy listing theo user	api/listing/search/byOwnerId/{id}	get
21		Xóa listing	api/listing/delete/{id}	delete
22		Lấy chi tiết listing	api/listing/detailListing/{id}	get
23		Tìm listing theo user và title	api/listing/search/byOwnerIdAndName?title={title}&ownerId={id}	get
24		Sửa listing	api/listing/{id}	put
25	copy	Tạo copy	api/copy	post
26		Lấy tất cả copy	api/copy	get
27		Lấy copy theo id	api/copy/{id}	get
28		Sửa copy	api/copy/{id}	put
29		Xóa copy	api/copy/{id}	delete
30	review	Tạo review	api/review	post
31		Lấy tất cả review theo userId	api/review/search/byUserId/{userId}	get
32		Lấy review theo listingId	api/review/search/byListingId/{listingId}	get
33	Lease order	Tạo đơn hàng	api/leaseOrder	post
34		Lấy danh sách đơn hàng theo người cho thuê	api/leaseOrder/search/lessor/{lessorId}	get
35		Lấy danh sách đơn hàng theo người thuê	api/leaseOrder/search/lessee/{lesseeId}	get
36		Lấy danh sách đơn hàng theo người cho thuê và status	api/leaseOrder/search/lessor/status/{lessorId}?status={status}	get
37		Lấy danh sách đơn hàng theo người thuê và status	api/leaseOrder/search/lessee/status/{lesseeId}?status={status}	get

38		Thay đổi trạng thái đơn hàng	api/leaseOrder/edit/status?status={status}&id={id}	get
39		Lấy đơn hàng theo id	api/leaseOrder/{id}	get
40	Sale order	Tạo đơn hàng	api/SaleOrder/createSaleOrder	post
41		Lấy danh sách đơn hàng theo người bán	/api/SaleOrder/seller/{sellerId}	get
42		Lấy danh sách đơn hàng theo người mua	/api/SaleOrder/buyer/{buyerId}	get
43		Lấy tất cả đơn hàng mua bán	/api/SaleOrder	get
44		Thay đổi trạng thái đơn hàng mua	api/SaleOrder/status?status={status}&id={id}	get
45		Lấy đơn hàng theo id	api/SaleOrder/{id}	get
46	voucher shop	Lấy danh sách voucher người bán	api/voucher-shop	get
47		Search voucher người bán	api/voucher-shop/search?keyword={keyword}	get
48		Tạo voucher người bán	api/voucher-shop	post
49		Lấy chi tiết voucher người bán	api/voucher-shop/{id}	get
50		Sửa voucher người bán	api/voucher-shop/{id}	put
51		Xóa voucher người bán	api/voucher-shop/{id}	delete
52	voucher session	Lấy danh sách voucher toàn sàn	api/voucher-session	get
53		Search voucher toàn sàn	api/voucher-session/search?keyword={keyword}	get
54		Tạo voucher toàn sàn	api/voucher-session	post
55		Lấy chi tiết voucher toàn sàn	api/voucher-session/{id}	get
56		Sửa voucher toàn sàn	api/voucher-session/{id}	put
57		Xóa voucher toàn sàn	api/voucher-session/{id}	delete

Bảng 41: Api xây dựng trong dự án

CHƯƠNG 6: HIỆN THỰC FRONT END

6.1 Phân tích source code Front-end

6.1.1 Tổng quan thư mục Front-end

Cấu trúc được thiết kế cho một dự án Next.js với các phần cụ thể được tổ chức hợp lý, hỗ trợ phát triển front-end hiện đại và hiệu quả.

```
└── .env.local
└── .eslintrc.json
└── .gitignore
└── docs
└── next-env.d.ts
└── next.config.js
└── next.config.mjs
└── package.json
└── pnpm-lock.yaml
└── postcss.config.js
└── public
└── README.md
└── src
    ├── api
    ├── app
    ├── assets
    ├── components
    ├── hooks
    ├── server
    ├── types
    └── utils
└── tailwind.config.ts
└── tsconfig.json
```

- **.env.local:** File này lưu trữ các biến môi trường cần thiết cho dự án như API keys, endpoints, secret keys.
- **.eslintrc.json:** Cấu hình ESLint để đảm bảo mã nguồn nhất quán và giảm lỗi cú pháp.
- **docs/:** Chứa tài liệu liên quan đến dự án, phục vụ việc ghi chép và chuyển giao kiến thức.
- **public/:** Chứa các file tĩnh như hình ảnh, favicon, hoặc các tài nguyên không cần xử lý qua JavaScript.
- **src/:** Thư mục chính chứa mã nguồn, được phân chia cụ thể như sau:
 - **api/:** Quản lý các API clients, kết nối với server hoặc dịch vụ bên thứ ba.
 - **app/:** Chứa các route, page hoặc các cấu trúc app-specific, tận dụng hệ thống app directory của Next.js 13.
 - **assets/:** Chứa tài nguyên (hình ảnh, fonts, CSS).
 - **components/:** Các component React được dùng lại nhiều lần như navbar, footer, button.
 - **hooks/:** Chứa các custom hooks, giúp quản lý trạng thái hoặc logic phức tạp.
 - **server/:** Xử lý logic phía server hoặc tích hợp các dịch vụ backend.
 - **types/:** Chứa các định nghĩa TypeScript (interface, types).
 - **utils/:** Chứa các hàm tiện ích như format ngày tháng, xử lý chuỗi.
- **tailwind.config.ts:** Cấu hình Tailwind CSS, tùy chỉnh theme và các tiện ích CSS.

- **tsconfig.json:** Cấu hình TypeScript, tối ưu cho mã nguồn an toàn và rõ ràng.

6.1.2 Thư mục api

```

├── auth
│   ├── loginService.tsx
│   └── registerService.tsx
├── bookListService.tsx
└── checkoutService.tsx
├── create
│   ├── createBookService.tsx
│   ├── createDocumentService.ts
│   └── createPostService.tsx
├── deletePostService.tsx
├── genreService.tsx
├── handleError.ts
├── imageService.ts
├── managerPostService.tsx
├── order.ts
├── orderBuy.ts
├── profile.tsx
└── ratingService.tsx
└── voucher
    ├── voucherSession.ts
    └── voucherShop.ts

```

Thư mục api chứa các dịch vụ gọi api tới Back-end. Chức năng chính của thư mục api:

- Tương tác với API Backend: Tất cả các file dịch vụ đều đảm nhận vai trò gửi và nhận dữ liệu từ API backend.
- Đóng gói logic xử lý API: Mỗi dịch vụ được tổ chức thành các file riêng biệt để dễ bảo trì và mở rộng.
- Hỗ trợ các module Front-end: Cung cấp dữ liệu và logic xử lý cần thiết cho các thành phần (components) hoặc trang (pages) trong ứng dụng Next.js.

6.1.3 Thư mục app

```

├── (auth)
│   ├── login
│   │   ├── [...login]
│   │   │   └── page.tsx
│   │   └── Login.css
│   └── sign-up
│       └── [...sign-up]
│           └── page.tsx
└── (manager)
    ├── buy-order
    │   └── page.tsx
    ├── customer-order
    │   └── page.tsx
    └── layout.tsx
    └── manager-post
        ├── add-post
        │   └── layout.tsx
        └── edit-post
            └── [...postId]
                └── layout.tsx
            └── layout.tsx
            └── page.tsx
    └── manager-voucher
        └── add-voucher

```

```

    └── layout.tsx
        └── page.tsx
    └── layout.tsx
        └── page.tsx
    └── my-order
        └── [...status]
            └── layout.tsx
            └── page.tsx
    └── order
        └── [...order]
            └── layout.tsx
            └── page.tsx
    └── profile
        └── page.tsx
    └── sell-order
        └── page.tsx
    └── (policy)
        └── book-condition-guide
            └── page.tsx
        └── private-policy
            └── page.tsx
    └── cart
        └── layout.tsx
        └── page.tsx
    └── checkout
        └── [[...checkout]]
            └── page.tsx
        └── layout.tsx
    └── detail
        └── [...detail]
            └── layout.tsx
            └── page.tsx
    └── error.tsx
    └── globals.css
    └── layout.tsx
    └── loading.tsx
    └── not-found.scss
    └── not-found.tsx
    └── page.tsx
    └── search
        └── layout.tsx
        └── page.tsx

```

Thư mục app chứa các thành phần đại diện cho các trang chính của ứng dụng Next.js. Đây là nơi tổ chức các tệp và thư mục liên quan đến giao diện và logic của từng trang cụ thể, bao gồm:

- Quản lý xác thực người dùng: Bao gồm các trang như đăng nhập, đăng ký, và xử lý liên quan đến xác thực.
- Chức năng quản lý: Cung cấp giao diện để quản lý đơn hàng (mua/bán), bài đăng, voucher, và thông tin cá nhân.
- Chính sách và hướng dẫn: Hiển thị các trang hướng dẫn sử dụng (như điều kiện sách) và chính sách bảo mật.
- Xử lý giỏ hàng và thanh toán: Cung cấp các trang quản lý giỏ hàng và thực hiện thanh toán.
- Chi tiết và tìm kiếm: Hiển thị chi tiết sản phẩm/bài viết và các kết quả tìm kiếm của người dùng.

- Các trang lỗi và tải trạng thái: Bao gồm các thành phần xử lý lỗi (404, not found), trang thái tải trang, và giao diện chung của ứng dụng.
- Tổ chức bộ cục (Layouts): Các tệp layout.tsx cung cấp bộ cục (header, footer, sidebar) dùng chung cho từng nhóm trang cụ thể.

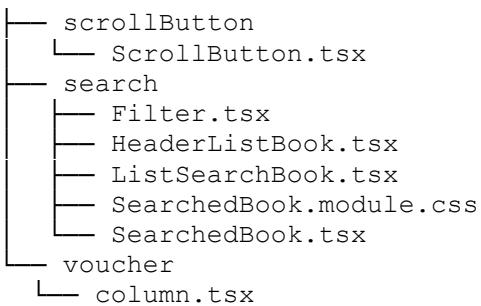
6.1.4 Thư mục components

```

├── auth
│   ├── FormLogin.tsx
│   └── FormSignup.tsx
└── cart
    ├── BookDetailInfo.tsx
    ├── Cart.css
    ├── Cart.scss
    ├── CartInfo.tsx
    ├── CartInfoItem.tsx
    ├── CartItem.tsx
    ├── InfoCheckout.tsx
    ├── voucherSession
    │   ├── calculateVoucher.ts
    │   ├── ListVoucher.tsx
    │   ├── ModalChooseVoucher.tsx
    │   └── StyleVoucher.tsx
    └── voucherShop
        └── ListVoucherShop.tsx
└── checkout
    ├── BookItem.tsx
    ├── FormInputText.tsx
    ├── HowToRent.tsx
    ├── InfoRent.tsx
    ├── Order.tsx
    ├── OrderBuy.tsx
    ├── parseUrlParams.ts
    ├── Pay.tsx
    ├── PaymentStatus.tsx
    ├── Step.css
    ├── Step.scss
    ├── Step1.tsx
    ├── Step2.tsx
    ├── Step2Buy.tsx
    └── Step3.tsx
└── common
    └── CommonAlert.tsx
└── createPost
    ├── AccordionCustom.tsx
    ├── CreateBook.tsx
    ├── CreateDocument.tsx
    ├── CreatePost.tsx
    ├── dummy.ts
    ├── FindBookAutocomplete.tsx
    ├── GenreAutocomplete.tsx
    ├── InputListing.tsx
    └── ModalSearchBook.tsx
└── customerOrder
    ├── ListOrderMain.tsx
    └── ListOrderMainBuy.tsx
└── detail
    ├── BookGallery.css
    ├── BookGallery.scss
    ├── BookGallery.tsx
    ├── BookInfo.tsx
    ├── DocumentDetail.tsx
    ├── DocumentInfo.tsx
    └── Owner.tsx

```

```
    ├── Quality.tsx
    ├── RateBook.tsx
    ├── RateComment.tsx
    ├── ReadSample.tsx
    ├── RentBook.tsx
    ├── RentCombo.tsx
    └── RentDay.tsx
  └── footer
    ├── BookCategoryList.tsx
    ├── Footer.css
    ├── Footer.scss
    ├── Footer.tsx
    ├── Info.tsx
    └── Mail.tsx
  └── home
    ├── Category
    │   ├── BookCategory.tsx
    │   ├── Category.css
    │   ├── Category.scss
    │   ├── Category.tsx
    │   └── hoangtube.jpg
    ├── HeroSection
    │   ├── HeroSection.css
    │   ├── HeroSection.jsx
    │   └── HeroSection.scss
    ├── HomeBanner.tsx
    ├── NewComing
    │   ├── NewBook.css
    │   ├── NewBook.scss
    │   ├── NewBook.tsx
    │   └── NewComingList.tsx
    ├── PromoteSection.tsx
    └── Statistic
      ├── BookCardCarousel.tsx
      ├── G13ZDV5U.jpg
      ├── Statistic.css
      ├── Statistic.jsx
      └── Statistic.scss
  └── managerPost
    ├── column.tsx
    └── DeletePostModal.tsx
  └── nav
    ├── AlertSignOut.tsx
    ├── BookMenu.tsx
    ├── Navbar.tsx
    └── SearchBar.tsx
  └── order
    ├── CancelModal.tsx
    ├── column.tsx
    ├── contentAlert.tsx
    ├── CustomTabPanel.tsx
    ├── DetailOrder.tsx
    ├── HeaderOrder.tsx
    ├── ListOrder.tsx
    ├── MenuProfile.tsx
    ├── NoData.tsx
    ├── OrderFooter.tsx
    └── RateModel.tsx
  └── orderBuy
    ├── CancelModalBuy.tsx
    ├── DetailOrderBuy.tsx
    ├── HeaderOrderBuy.tsx
    ├── ListOrderBuy.tsx
    └── OrderFooterBuy.tsx
```



- auth/: Chứa các component liên quan đến xác thực người dùng như FormLogin.tsx, FormSignup.tsx. Đây là các form đăng nhập và đăng ký, đảm bảo người dùng có thể truy cập vào hệ thống.
- cart/: Các thành phần giao diện giỏ hàng:
 - Hiển thị thông tin sách, tổng quan giỏ hàng (CartInfo.tsx, CartItem.tsx).
 - Tính toán voucher (calculateVoucher.ts, ListVoucher.tsx).
 - Thành phần liên quan đến lựa chọn voucher trong giỏ hàng (ModalChooseVoucher.tsx, ListVoucherShop.tsx).
- checkout/: Quản lý giao diện trong quy trình thanh toán:
 - Các bước thanh toán (Step1, Step2, Step3).
 - Hiển thị trạng thái thanh toán (PaymentStatus.tsx).
 - Các thành phần nhập liệu và xác nhận đơn hàng (FormInputText.tsx, Order.tsx, OrderBuy.tsx).
- common/: Chứa các component chung được dùng nhiều nơi trong hệ thống, ví dụ như CommonAlert.tsx dùng để hiển thị thông báo.
- createPost/: Các thành phần liên quan đến việc đăng bài và quản lý bài viết:
 - Tạo mới sách, tài liệu, hoặc bài viết (CreateBook.tsx, CreateDocument.tsx, CreatePost.tsx).
 - Các autocomplete hỗ trợ người dùng chọn thể loại hoặc tìm sách (FindBookAutocomplete.tsx, GenreAutocomplete.tsx).
- customerOrder/: Hiển thị danh sách đơn hàng của khách hàng. Bao gồm danh sách đơn hàng thuê (ListOrderMain.tsx) và mua (ListOrderMainBuy.tsx).
- detail/: Chi tiết bài đăng hoặc sách:
 - Hiển thị thông tin sách (BookInfo.tsx, DocumentInfo.tsx).
 - Tính năng liên quan đến thuê sách (RentBook.tsx, RentCombo.tsx, RentDay.tsx).
 - Thư viện hình ảnh (BookGallery.tsx) và đánh giá sách (RateBook.tsx, RateComment.tsx).
- footer/: Giao diện footer của ứng dụng:
 - Thông tin danh mục sách (BookCategoryList.tsx).
 - Các thành phần hiển thị thông tin liên hệ (Info.tsx, Mail.tsx).
- home/: Giao diện trang chủ:
 - Hiển thị danh mục sách, banner, sách mới ra mắt (BookCategory.tsx, HomeBanner.tsx, NewComingList.tsx).

- Thành phần thống kê và carousel sách (Statistic.tsx, BookCardCarousel.tsx).
- managerPost/: Thành phần hỗ trợ quản lý bài đăng. Ví dụ như Modal xóa bài đăng (DeletePostModal.tsx) hoặc định nghĩa các cột trong bảng quản lý (column.tsx).
- nav/: Giao diện thanh điều hướng và tìm kiếm:
 - Thanh navbar chính (Navbar.tsx).
 - Tính năng tìm kiếm sách (SearchBar.tsx).
- order/ và orderBuy/: Quản lý đơn hàng thuê và mua:
 - Hiển thị danh sách đơn hàng (ListOrder.tsx, ListOrderBuy.tsx).
 - Chi tiết đơn hàng và trạng thái (DetailOrder.tsx, DetailOrderBuy.tsx).
- scrollButton/: Chứa nút cuộn lên đầu trang (ScrollButton.tsx).
- search/: Các component liên quan đến tìm kiếm sách:
 - Hiển thị kết quả tìm kiếm (ListSearchBook.tsx, SearchedBook.tsx).
 - Bộ lọc và tiêu đề danh sách sách (Filter.tsx, HeaderListBook.tsx).
- voucher/: Thành phần quản lý và hiển thị voucher. Ví dụ: Định nghĩa cột voucher trong bảng (column.tsx).

6.1.5 Thư mục hooks

```
hooks/
├── alert.tsx
├── cart.tsx
├── choosenBook.tsx
├── genre.ts
├── listNewBook.tsx
├── order.tsx
├── search.tsx
└── step.ts
├── useApiCall.tsx
└── người dùng.tsx
└── voucher.tsx
```

Thư mục hooks chứa các file liên quan đến việc sử dụng các hooks để thao tác với các trạng thái của các component trong dự án:

- alert.tsx: hook quản lý hiển thị thông báo hoặc cảnh báo trên giao diện người dùng.
- cart.tsx: hook quản lý giỏ hàng trong ứng dụng.
- choosenBook.tsx: hook quản lý sách được chọn.
- genre.ts: hook quản lý thể loại sách.
- listNewBook.tsx: hook hiển thị danh sách sách mới.
- order.tsx: hook quản lý đơn hàng, thay đổi trạng thái đơn hàng trong ứng dụng.
- search.tsx: hook chứa chức năng xác định loại tìm kiếm trong ứng dụng.
- người dùng.tsx: hook quản lý thông tin và token của người dùng.
- voucher.tsx: hook quản lý voucher shop và voucher session lưu khi ở trong giỏ hàng

6.1.6 Thư mục types

```
types/
├── book.ts
└── category.ts
```

```
├ checkout.ts  
├ common.ts  
├ form.ts  
├ order.ts  
├ page.ts  
└ voucher.ts  
└ người dùng.ts
```

Thư mục types được sử dụng để chứa các định nghĩa về kiểu dữ liệu (types và interface) của các thực thể. Các tệp trong thư mục này giúp xác định cấu trúc của các đối tượng và đảm bảo rằng mã nguồn tuân thủ các quy tắc về kiểu dữ liệu, giúp phát hiện lỗi sớm và làm cho mã nguồn dễ hiểu hơn. Trong đó có cấu trúc của các dữ liệu liên quan đến sách, giỏ hàng, thẻ loại, thanh toán, các form, đơn hàng, người dùng, voucher.

6.1.7 Thư mục utils

```
utils/  
├ env.ts  
├ helps.ts  
└ theme.ts
```

Thư mục utils được sử dụng để chứa các hàm tiện ích và các tệp mã có thể tái sử dụng trên toàn bộ ứng dụng. Như các tệp hỗ trợ các tác vụ logic hoặc xử lý dữ liệu. Mục đích của thư mục này giúp tái sử dụng mã, làm dòng code ngắn gọn và dễ hiểu hơn đồng thời giúp cho việc quản lý tốt hơn. Trong đây em đã tiến hành lưu trữ các biến môi trường ở env. Các hàm tiện ích như chia mảng, format giá trị các số,... trong helps. Và thư mục theme được sử dụng để tạo ra một đối tượng theme tùy chỉnh cho ứng dụng sử dụng Material-UI.

6.2 Bộ nhận diện thương hiệu

6.2.1 Logo

Logo được chọn theo tiêu chí đẹp, sáng tạo và mơ mộng, thân thiện với người dùng trẻ, bao gồm có một cuốn sách và đôi cánh. Nên em đã dùng AI để tạo ra được logo sau đây:

Phiên bản logo không chữ:



Hình 31: Phiên bản logo không chữ

Sau đây là phiên bản logo có chữ:



Hình 32: Logo với tên dự án The Flying Bookstore

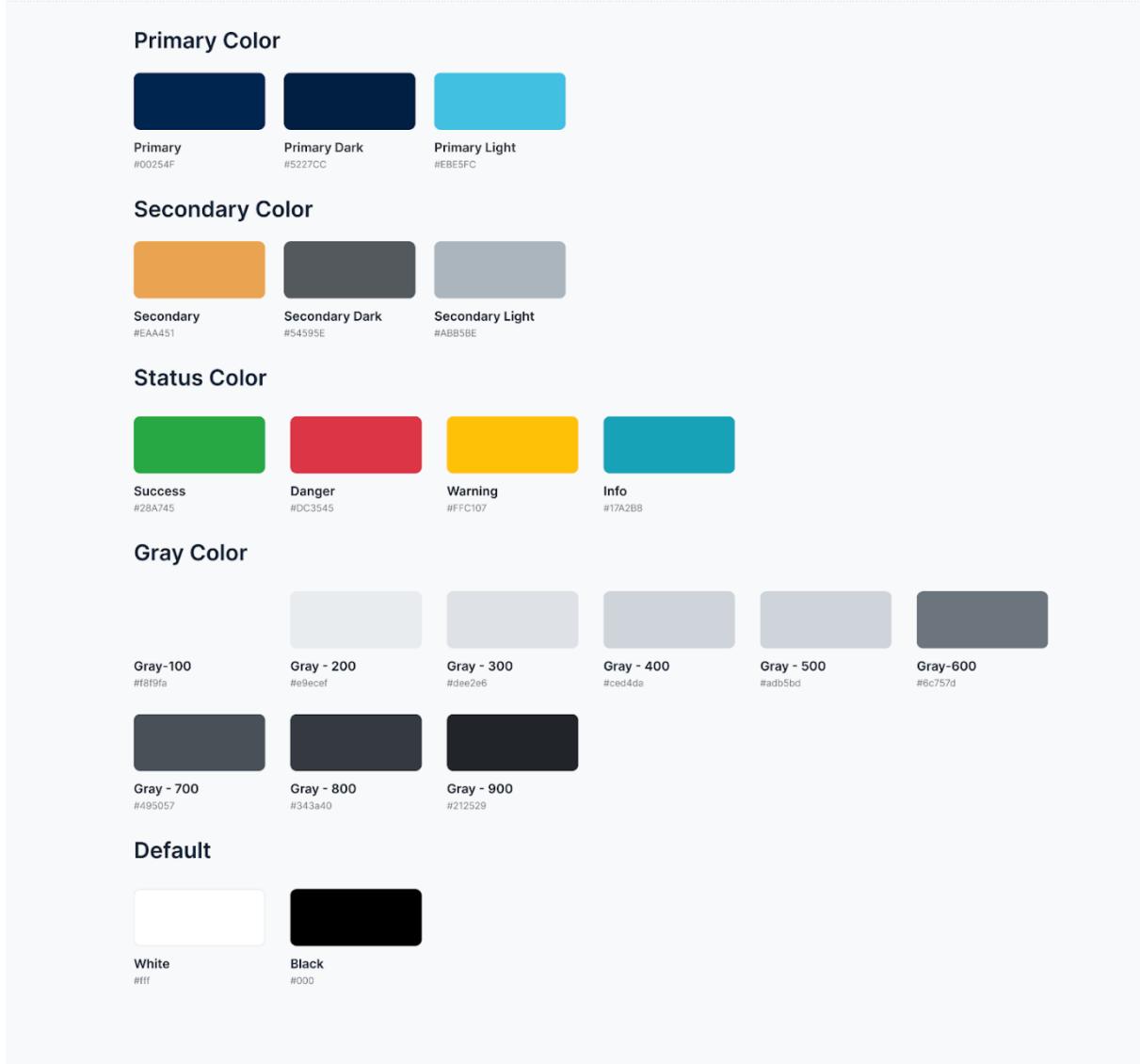
6.2.2 Màu sắc

Màu sắc áp dụng trong logo được áp dụng trên toàn bộ website và tạo thành các biến, trong đó em có bốn hệ thống bao gồm:

- Màu chính: #00254F
- Màu phụ: #EAA451
- Màu trạng thái
 - Thành công: #28A745
 - Nguy hiểm: #DC3545
 - Cảnh báo: #FFC107
 - Thông tin: #17A2B8

Các màu được sắp xếp và tổ chức theo hệ thống màu của Bootstrap

Colors



Hình 33: Bảng hệ thống màu

6.2.3 Font chữ

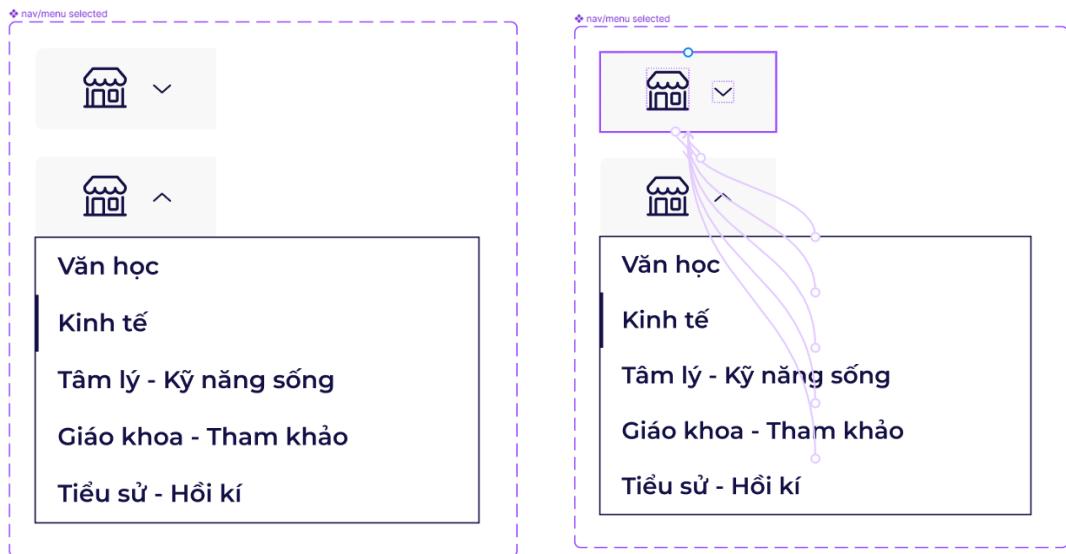
Font chữ chính: Montserrat

Font chữ phụ: Open sans

6.2.4 Component chung

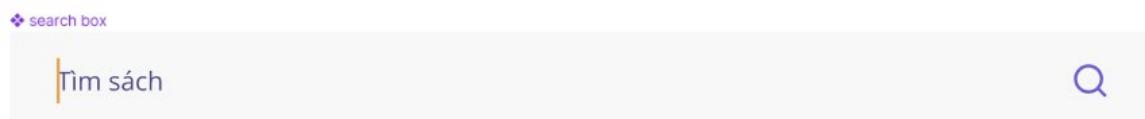
6.2.4.1 Header

Để tạo thành Header, ngoài logo có sẵn, button được lấy từ Primary button ở trên, tác giả còn tự thiết kế thêm phần list menu và gắn prototype kỹ càng cho từng đề mục



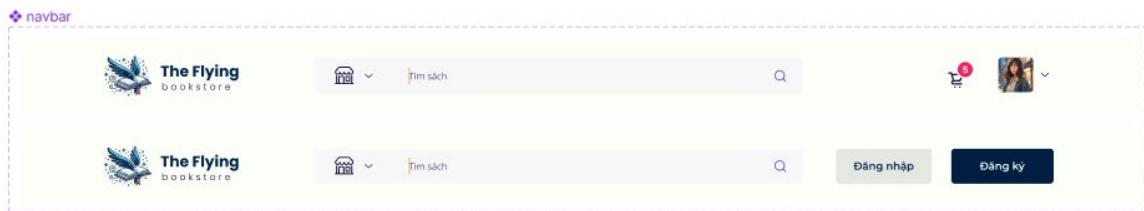
Hình 34: Component list menu

Tác giả tạo thêm search box component cho riêng ô search



Hình 35: Component search

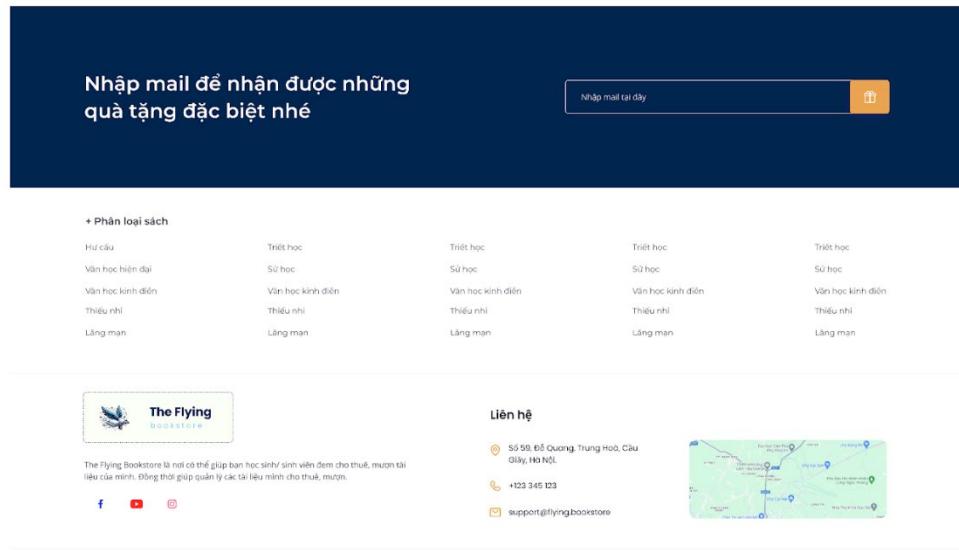
Gom tắt cả component từ trên lại, ta có một header với navbar hoàn chỉnh như sau, có 2 trạng thái cho navbar này là đăng nhập rồi (ở trên) và chưa đăng nhập (ở dưới)



Hình 36: Component header

6.2.4.2 Footer

Footer được tạo thành từ 2 component, một là component Nhập email để gửi quà, hai là footer với các thông tin liên hệ và phân loại sách



Hình 37: Footer component

6.3 Giao diện

Các trang của front-end gồm:

1. Trang chủ
2. Trang tìm kiếm và hiển thị danh sách bài đăng
3. Chi tiết bài đăng
4. Giỏ hàng
5. Đặt hàng
6. Xuất đơn hàng
7. Lấy đơn hàng
8. Trang đăng nhập
9. Trang đăng ký
10. Quản lý thông tin người dùng
11. Quản lý đơn hàng thuê
12. Quản lý đơn hàng cho thuê
13. Quản lý đơn hàng bán
14. Quản lý đơn hàng mua
15. Quản lý bài đăng
16. Chi tiết đơn hàng
17. Tạo bài đăng
18. Trang đăng ký Quản trị viên

19. Trang đăng nhập Quản trị viên
20. Trang quản lý đơn hàng thuê Quản trị viên
21. Trang quản lý đơn hàng mua bán Quản trị viên
22. Trang quản lý voucher toàn sàn Quản trị viên
23. Trang thêm voucher toàn sàn Quản trị viên
24. Trang Chính Sách Bảo Mật
25. Trang Hướng dẫn về tình trạng sách

6.3.1 Trang chủ

The Flying bookstore

BÁN CHẠY

Danh nhân Việt Nam

Doãn Kế Thiện #Văn học Việt Nam

Những câu chuyện về các danh nhân nước Việt. Đó là chuyện về Nguyễn Công Trứ ngang tàng mà suốt đời lận đận vì dân vì nước, chuyện về Trạng Bùng Phùng Khắc Khoan đấu trí với ngoại bang để không làm mất thể diện nước nhà, chuyện về Nguyễn Văn Giai đến tranh hồn để trả nghĩa người con gái năm xưa...

[Thuê ngay](#) Xem chi tiết →

5.000đ/ngày 74.000đ giảm 20%



Giao hàng nhanh

Bảo mật thanh toán

Chất lượng đỉnh nhất

Chính sách rõ ràng

Sách mới ra mắt

Từ 24/9 đến 29/11

Ice	Companions	Alchemy and Academe	Houses of Stone	Hide and Seek
Thuê: 1.600đ/ngày Mua: 263.620đ	Thuê: 2.700đ/ngày Mua: 202.650đ	Thuê: 2.000đ/ngày Mua: 84.210đ	Thuê: 1.400đ/ngày Mua: 53.500đ	Thuê: 2.200đ/ngày Mua: 161.630đ
Thêm vào giỏ hàng	Thêm vào giỏ hàng	Thêm vào giỏ hàng	Thêm vào giỏ hàng	Thêm vào giỏ hàng

Danh mục sản phẩm

- Giật gân
- Học thuật
- Hồi ký
- Kinh điển
- Hải kịch
- truyện trar

Nhập mail để nhận được những quà tặng đặc biệt nhé!

Nhập mail tại đây [Gửi](#)

Liên hệ

17/11 Trường Phước Phan, Bình Hưng Hòa, Bình Tân, TP.HCM
+123 345 123
support@flying.bookstore

Chính sách

Chính sách bảo mật
Hướng dẫn về tình trạng sách

Hình 38: Trang chủ

6.3.1.1 Tổng quan

- Mục đích:** Cho phép khách truy cập trang web xem danh sách tài liệu được đề xuất và thể loại liên quan để chọn sách phù hợp.
- Tác nhân tham gia:** Khách (người dùng chưa đăng nhập).
- Điều kiện tiên quyết: N/A.
- Kết quả mong muốn:** Hiển thị danh sách tài liệu nổi bật và các thể loại sách phổ biến để khách dễ dàng lựa chọn.

6.3.1.2 Quy tắc và Ràng buộc

- Khách không cần đăng nhập vẫn có thể xem nội dung đề xuất.
- Danh sách tài liệu đề xuất bao gồm:
 - Sách mới ra mắt.
 - Thể loại phổ biến.
- Mỗi tài liệu hiển thị thông tin cơ bản:
 - Tên sách.
 - Giá thuê/ngày hoặc giá mua.
 - Hình ảnh minh họa.
- Các tài liệu và thể loại hiển thị dựa trên mức độ ưu tiên do quản trị viên cấu hình.

6.3.1.3 Mô tả chi tiết

STT	Tên	Mô tả	Loại điều khiển	Kiểu dữ liệu	Giá trị mặc định	Bắt buộc	Quy tắc
1	Danh sách sách bán chạy	Hiển thị các sách bán chạy trên hệ thống.	Gridview	Array	N/A	Có	Bao gồm tên, giá, hình ảnh và nút thêm vào giỏ hàng.
2	Danh sách sách mới	Hiển thị sách mới ra mắt trong khoảng thời gian gần đây.	Gridview	Array	N/A	Có	Bao gồm tên, giá, hình ảnh và nút thêm vào giỏ hàng.
3	Thể loại sách	Danh sách các thể loại phổ biến được hiển thị dưới dạng biểu tượng.	Gridview	Array	N/A	Có	Hiển thị biểu tượng và tên từng thể loại.
4	Nút thêm vào giỏ hàng	Cho phép khách thêm sách vào giỏ hàng từ danh sách đề xuất.	Button	Boolean	N/A	Có	Hoạt động khi khách click vào nút.
5	Chi tiết sách	Khi nhấp vào tên hoặc hình ảnh sách, dẫn đến trang chi tiết của sách.	Hyperlink	String	N/A	Có	Link đến trang chi tiết của tài liệu được chọn.

Bảng 42: Mô tả chi tiết Trang chủ

6.3.1.4 Quy trình xử lý

1. Khách truy cập trang chủ của hệ thống.
2. Giao diện hiển thị:
 - o **Sách mới ra mắt:** Hiển thị sách.
 - o **Thể loại phổ biến:** Hiển thị danh sách các thể loại.
3. Khách chọn tài liệu để xem chi tiết hoặc thêm vào giỏ hàng.

6.3.1.5 API Mô tả

API: Lấy danh sách thể loại (genre)

- **Endpoint:** /api/genre
- Phương thức: GET
- **Mô tả:** Lấy danh sách thể loại (genre).
- Tham số đầu vào: Không có

API: Tìm kiếm danh sách sách

- **Endpoint:** /api/listing/search
- Phương thức: GET
- **Mô tả:** Tìm kiếm danh sách sách.
- Tham số đầu vào:

Tên tham số	Loại dữ liệu	Mô tả	Bắt buộc
size	int	Số lượng kết quả trên mỗi trang	Có
page	int	Trang hiện tại (bắt đầu từ 0)	Có
title	String	Tiêu đề sách để tìm kiếm	Không
allowRent	int	Cho phép thuê (1 là true, 0 là false)	Có
allowPurchase	int	Cho phép mua (1 là true, 0 là false)	Có

Bảng 43: Tham số đầu vào API Tìm kiếm danh sách sách

6.3.2 Trang tìm kiếm và hiển thị danh sách bài đăng

Lọc theo

XÓA BỘ LỌC

Danh mục

- Hoc thuật
- Hồi ký
- Kinh điển
- Hải kịch
- truyện tranh
- Văn hóa
- Kịch
- Kinh tế học
- Giáo dục
- Môi trường
- Sứ thi
- Truyện cổ tích
- Tưởng tượng
- Viễn tưởng
- Địa lý
- ma
- Sức khỏe
- Lịch sử
- Kinh dị
- Nhân văn
- Văn học
- Bí ẩn
- Thần thoại
- tên
- Thiên nhiên
- Tiểu thuyết
- Vật lý
- Thơ
- Tâm lý
- Tôn giáo
- Lãng mạn
- Lãng man
- Khoa học
- Truyền ngắn
- Xã hội
- Xã hội học
- Các môn thể thao
- Hồi hộp
- Sách giáo khoa
- Giật gân

Sách theo thể loại Hồi ký

Hiển thị 40 cuốn sách hiện có

Sách mua và thuê

Tên sách	Giá thuê	Giá mua
Castaway	200đ/ngày	313.670đ
Red Dust	1.200đ/ngày	82.950đ
Hàng Time	500đ/ngày	343.210đ
Broken	500đ/ngày	95.480đ
Dannii	1.600đ/ngày	252.330đ
The Game - out of print	2.600đ/ngày	277.200đ
31 Songs	1.600đ/ngày	306.190đ
Jane Austen	2.300đ/ngày	240.310đ
At the Center of the Storm	300đ/ngày	138.460đ
Botham	2.800đ/ngày	64.330đ
A Parrot in the Pepper Tree	1.100đ/ngày	320.740đ
Bringing the House Down	1.300đ/ngày	147.280đ
When Heaven and Earth C...	2.600đ/ngày	261.100đ
Without Mercy	500đ/ngày	29.680đ
Broken Music	1.700đ/ngày	171.640đ
Every Secret Thing	500đ/ngày	172.200đ
It was on Fire when I Lay D...	1.000đ/ngày	192.080đ
Talking Straight	2.500đ/ngày	259.140đ
A Blessing over Ashes	100đ/ngày	32.060đ
Don Cherry's Hockey Stori...	2.500đ/ngày	174.930đ

Nhập mail để nhận được những quà tặng đặc biệt nhé !

Nhập mail tại đây

Liên hệ

- 17/11 Trương Phước Phan, Bình Hưng Hòa, Bình Tân, TP.HCM
- +123 345 123
- support@flying.bookstore

Chính sách

- Chính sách bảo mật
- Hướng dẫn về tình trạng sách

Hình 39: Trang danh sách bài đăng và tìm kiếm tài liệu

6.3.2.1 Tổng quan

- Mục đích:** Chức năng tìm kiếm tài liệu cho phép khách hàng dễ dàng tìm thấy sách mong muốn trên hệ thống thông qua thanh tìm kiếm hoặc bộ lọc.

- **Tác nhân tham gia:** Khách (không cần đăng nhập).
- **Điều kiện tiên quyết:**
 - Dữ liệu sách đã được lưu trữ đầy đủ trên hệ thống.
 - Hệ thống có tích hợp thanh tìm kiếm và các bộ lọc.
- **Kết quả mong muốn:** Khách hàng tìm thấy các sách liên quan đến từ khóa hoặc tiêu chí lọc mà họ quan tâm.

6.3.2.2 Quy tắc và Ràng buộc

Thanh tìm kiếm:

- Hỗ trợ tìm kiếm theo tên sách, tác giả, thể loại.
- Tìm kiếm không phân biệt chữ hoa/chữ thường.

Bộ lọc:

- Lọc sách theo danh mục (thể loại).
- Hỗ trợ lọc theo giá thuê, giá mua.

Phân trang:

- Hiển thị tối đa 20 sách trên một trang, với các nút điều hướng phân trang.

Kết quả hiển thị:

- Hiển thị danh sách sách có liên quan đến từ khóa hoặc tiêu chí lọc.
- Mỗi sách hiển thị hình ảnh, tiêu đề, giá thuê, giá mua.

6.3.2.3 Mô tả chi tiết

STT	Tên	Mô tả	Loại điều khiển	Kiểu dữ liệu	Giá trị mặc định	Bắt buộc	Quy tắc
1	Thanh tìm kiếm	Nhập từ khóa để tìm kiếm sách.	Textbox	varchar(255)		X	Kết quả tìm kiếm hiển thị sách có chứa từ khóa trong tên hoặc tác giả.
2	Danh mục sách	Bộ lọc sách theo thể loại (ví dụ: Hồi ký, Tiểu thuyết, Khoa học, v.v.).	Checkbox group	varchar(255)			Cho phép chọn một hoặc nhiều thể loại để lọc sách.
3	Giá thuê/mua	Hiển thị giá thuê/mua của từng sách.	Hiển thị thông tin	decimal(10,2)			Hiển thị giá dưới mỗi sách trong kết quả tìm kiếm/löc.
4	Phân trang	Điều hướng giữa các trang kết quả tìm kiếm/löc.	Button group	integer	Trang 1		Hiển thị số trang và nút điều hướng (Trước/Tiếp).

Bảng 44: Mô tả chi tiết Trang danh sách bài đăng và tìm kiếm tài liệu

6.3.2.4 Quy trình xử lý

1. Tìm kiếm bằng từ khóa:
 - a. Khách hàng nhập từ khóa (tên sách hoặc tác giả) vào thanh tìm kiếm.
 - b. Hệ thống thực hiện tìm kiếm và trả về danh sách sách có chứa từ khóa liên quan.
2. Lọc sách:
 - a. Khách hàng chọn danh mục sách từ danh sách bộ lọc (ví dụ: Hồi ký, Lịch sử, Kinh tế).
 - b. Hệ thống cập nhật kết quả dựa trên danh mục đã chọn.
3. Kết hợp tìm kiếm và lọc:
 - a. Khách hàng có thể sử dụng cả thanh tìm kiếm và bộ lọc để thu hẹp kết quả.
 - b. Hệ thống thực hiện tìm kiếm và lọc đồng thời, trả về kết quả phù hợp.
4. Phân trang:
 - a. Khi kết quả vượt quá 20 sách, hệ thống phân trang để khách hàng dễ dàng điều hướng.
 - b. Khách hàng nhấn nút "Trang trước" hoặc "Trang sau" để xem thêm.

6.3.2.5 Đặc tả API

Sử dụng lại API: Lấy danh sách thể loại (genre) của màn hình Trang chủ

6.3.3 Trang chi tiết bài đăng

The screenshot shows the product detail page for the book 'Hide and Seek' by Fern Michaels. At the top, there's a navigation bar with a logo, a search bar, and user icons. Below the header is the book cover, which features the title 'HIDE AND SEEK' and the author's name 'FERN MICHAELS'. To the right of the cover, there's a star rating of 0, publication details ('Tác giả: Fern Michaels', 'Nhà xuất bản: Oxford University Press', 'Ngày xuất bản: 01-01-2008'), and a brief description: 'Flight lieutenant Mark Hunter, one of the RAF's most ambitious pilots, discovers that it was not the Iraqis who used the most virulent strains of chemical warfare during the Gulf War. The Pentagon and Whitehall will stop at nothing to ensure that a deadly skeleton from Desert Storm remains buried.' Below the description, the price 'Giá thuê: 2.200đ/ngày' is listed, along with a note 'Hiện vẫn còn hàng'.

CHI TIẾT **ĐÁNH GIÁ**

Thông tin chung về sách

Ngày xuất bản	01/01/2008
Kích thước	15.5x20.5
ISBN	1558746218
Số trang	316

Thông tin về sách thuê

Tiền cọc	230.900đ
Phản trả hư hại	9
Địa chỉ cho thuê	123 Nguyễn Trọng Tuyển, phường 12, quận Bình Thạnh
Phi phạt trả trễ	2.500đ/ngày

Đặt thuê

Thuê theo ngày: 2.200đ/ngày

Ngày nhận: 19/05/2024

Ngày trả:

Giá thuê: 2.200đ/ngày

Số ngày thuê: 0

Tổng tiền thuê: 0

Tiền cọc: 230.900đ

Tổng tiền: 0

Bạn cần chọn ngày trả

ĐẶT THUÊ NGAY

Chủ sách

Developer Dev

Số sách hiện có: 989

Số sách đang cho thuê: 1

Sách mới ra mắt

Từ 24/2 đến 19/05

Pride and Prejudice	Emma	Sense and Sensibility	The House of Scrofa	Ice
2.300đ/ngày	200đ/ngày	2.400đ/ngày	2.000đ/ngày	1.600đ/ngày
Thêm vào giỏ hàng				

Nhập mail để nhận được những quà tặng đặc biệt nhé!

Nhập mail tại đây

Liên hệ

17/11 Trường Phước Phan, Bình Hưng Hòa, Bình Tân, TP.HCM
+123 345 123
support@flying.bookstore

The Flying Bookstore là nơi có thể giúp bạn học sinh/ sinh viên tìm kiếm cho thuê, mượn tài liệu của mình. Đồng thời giúp quản lý các tài liệu mình cho thuê, mượn.

Hình 40: Trang chi tiết bài đăng

6.3.3.1 Tổng quan

- Mục đích:** Cung cấp thông tin chi tiết về sách, bao gồm mô tả, tình trạng sẵn có, giá thuê/mua và các thông tin khác, giúp khách đưa ra quyết định trước khi đăng ký tài khoản.
- Tác nhân tham gia:** Khách (người dùng chưa đăng nhập).

- Điều kiện tiên quyết:** Khách đã chọn một tài liệu cụ thể từ danh sách.
- Kết quả mong muốn:** Giao diện hiển thị đầy đủ thông tin chi tiết về tài liệu.

6.3.3.2 Quy tắc và Ràng buộc

- Khách có thể xem thông tin tài liệu mà không cần đăng nhập.
- Thông tin chi tiết tài liệu bao gồm:
 - Thông tin cơ bản: Tên sách, tác giả, nhà xuất bản, ngày phát hành, số trang, kích thước, ISBN.
 - Thông tin về thuê: Giá thuê/ngày, phí trả chậm, phần trăm hoàn trả cọc.
 - Thông tin về mua: Giá bán, tình trạng sẵn có.
 - Địa chỉ thuê/mua.
- Hiển thị nút hành động "Đặt thuê ngay" hoặc "Mua ngay" nếu tài liệu còn khả dụng.
- Nếu tài liệu không khả dụng, hiển thị thông báo "Sách hiện không có sẵn".

6.3.3.3 Mô tả chi tiết

Cùng với sử dụng tab thông tin chi tiết

STT	Tên	Mô tả	Loại điều khiển	Kiểu dữ liệu	Giá trị mặc định	Bắt buộc	Quy tắc
1	Tên sách	Tên của tài liệu/sách cần xem chi tiết.	Textview				Hiển thị tiêu đề của sách.
2	Tác giả	Tác giả của sách.	Textview				Hiển thị tên tác giả.
3	Nhà xuất bản	Tên nhà xuất bản của sách.	Textview				Hiển thị nhà xuất bản của sách.
4	Ngày xuất bản	Ngày sách được xuất bản.	Textview				Hiển thị ngày xuất bản của sách.
5	Kích thước	Kích thước sách (dài x rộng).	Textview				Hiển thị thông tin kích thước sách.
6	ISBN	Mã ISBN của sách.	Textview				Hiển thị mã ISBN của sách.
7	Số trang	Số lượng trang của sách.	Textview				Hiển thị số trang của sách.
8	Giá cọc	Số tiền đặt cọc khi thuê sách.	Textview				Hiển thị số tiền cọc phải thanh toán khi thuê.
9	Giá thuê theo ngày	Số tiền thuê mỗi ngày.	Textview				Hiển thị giá thuê theo ngày.
10	Phần trăm phí hư hại	Tỷ lệ phần trăm tiền đền bù nếu sách bị hư hại.	Textview				Hiển thị phần trăm tiền bồi thường khi làm hư hại sách.
11	Tổng tiền thuê	Tổng chi phí thuê sách trong thời gian chọn.	Textview				Hiển thị tổng tiền thuê được tính toán theo ngày thuê.

12	Đặt thuê ngay	Nút để xác nhận thuê sách.	Button				Người dùng bấm vào nút để thực hiện đặt thuê sách.
13	Đặt mua ngay	Nút để xác nhận mua sách.	Button				Người dùng bấm vào nút để thực hiện mua sách.

Bảng 45: Mô tả chi tiết Trang chi tiết bài đăng

Cùng với sử dụng tab đánh giá hình dưới

The screenshot shows a user interface for a review section. At the top, there's a navigation bar with 'CHI TIẾT' and 'DÁNH GIÁ'. Below it, a summary shows a 5-star rating (yellow stars) and '2 bài đánh giá'. There are buttons for 'Xem tất cả' and '5 sao', with other star ratings (4, 3, 2, 1) shown in smaller circles. Two sample reviews are listed below:

- Ấn danh**
Sách đẹp, vẫn còn mới
★★★★★ 5 sao | 13/05/2024
- Ấn danh**
Sách rất tốt
★★★★★ 5 sao | 19/05/2024

Hình 41: Tab đánh giá

STT	Tên	Mô tả	Điều khiển	Kiểu dữ liệu	Giá trị mặc định	Bắt buộc	Quy tắc
1	Tổng số sao đánh giá	Điểm trung bình sao của các đánh giá về sách.	Textview				Hiển thị trung bình điểm đánh giá (1-5 sao).
2	Tổng số lượng đánh giá	Tổng số bài đánh giá đã được ghi nhận.	Textview				Hiển thị tổng số bài đánh giá hiện có.
3	Bộ lọc sao đánh giá	Bộ lọc để xem các bài đánh giá dựa theo số sao (5 sao, 4 sao, v.v.).	Button				Cho phép người dùng lọc bài đánh giá theo số sao.
4	Danh sách bài đánh giá	Hiển thị danh sách các bài đánh giá chi tiết từ người dùng.	Textview				Hiển thị các bài đánh giá bao gồm nội dung, tên người đánh giá.
5	Nội dung đánh giá	Nội dung chi tiết từng bài đánh giá từ người dùng.	Textview				Hiển thị nội dung nhận xét về sách (ví dụ: "Sách đẹp, còn mới").
6	Ngày đánh giá	Ngày bài đánh giá được đăng.	Textview				Hiển thị thời gian bài đánh giá được tạo.

7	Người đánh giá	Tên hoặc thông tin người viết đánh giá (tên danh nếu không công khai).	Textview				Hiển thị tên người đánh giá hoặc để “Tên danh” nếu không hiển thị.
---	----------------	--	----------	--	--	--	--

Bảng 46: Mô tả chi tiết Tab đánh giá

6.3.3.4 Quy trình xử lý

1. Khách chọn một tài liệu từ danh sách đề xuất.
2. Giao diện chi tiết tài liệu hiển thị các thông tin cơ bản, giá cả, và mô tả.
3. Khách có thể thực hiện các hành động sau:
 - a. Thuê: Nhấn vào "Đặt thuê ngay".
 - b. Mua: Nhấn vào "Mua ngay".
 - c. Nếu tài liệu không khả dụng, khách chỉ có thể xem thông tin.

6.3.3.5 ĐẶC TẢ API

API: Lấy thông tin chi tiết tài liệu

- **Endpoint:** /api/listing/detailListing/{listingId}
- Phương thức: GET
- **Mô tả:** Lấy thông tin chi tiết của một bài đăng.
- Tham số đầu vào:

Tên tham số	Loại dữ liệu	Mô tả	Bắt buộc
listingId	number	ID của bài đăng	Có

Bảng 47: Tham số đầu vào API Lấy thông tin chi tiết tài liệu

API: Đánh giá sách

- **API:** Lấy danh sách đánh giá sách
- **Endpoint:** /api/review/{listingId}
- Phương thức: GET
- **Mô tả:** Lấy danh sách đánh giá chi tiết của sách.
- Tham số đầu vào:

Tên tham số	Loại dữ liệu	Mô tả	Bắt buộc
listingId	number	ID của bài đăng	Có

Bảng 48: Tham số đầu vào API Đánh giá sách

6.3.4 Trang giỏ hàng

The Flying Bookstore

Tìm sách...

Giỏ hàng của bạn

SÁCH THUÊ SÁCH MUA

At the Center of the Storm
George Tenet

Số lượng: 1 138.460đ 197.800đ

Khuyến Mãi từ người bán
Giảm 10.000đ Cho đơn hàng từ 50.000đ BỎ CHỌN

Chọn hoặc nhập mã khác

Khuyến Mãi toàn sàn
Giảm 50.000đ Cho đơn hàng từ 400.000đ BỎ CHỌN

Chọn hoặc nhập mã khác

Thông tin đặt hàng

Nguôi bán	Developer Dev	Giá giảm trực tiếp	59.340đ
Địa chỉ	123 Nguyễn Trọng Tuyển, phường 12, quận Bình Thạnh	Khuyến mãi từ người bán	10.000đ
Số điện thoại	0905 907 362	Khuyến mãi từ The Flying Bookstore	50.000đ
Giá gốc	197.800đ	Tổng tiền thanh toán	78.460đ

ĐẶT MUA HÀNG

Nhập mail để nhận được những quà tặng đặc biệt nhé !

Nhập mail tại đây

Liên hệ

- 171/11 Trương Phước Phan, Bình Hưng Hòa, Bình Tân, TP.HCM
- +84 123 345 123
- support@flyingbookstore.com

Chính sách

- Chính sách bảo mật
- Hướng dẫn về tình trạng sách

Tiệm Giòm Huỳnh Cảnh

Phản hồi: Đánh giá sản phẩm

Hình 42: Trang giỏ hàng

6.3.4.1 Tổng quan

- Mục đích:** Cho phép người dùng quản lý danh sách sách muốn thuê/mua trong giỏ hàng, cập nhật số lượng hoặc xóa sách trước khi đặt hàng.
- Tác nhân tham gia:** Người dùng (đã đăng nhập hoặc không đăng nhập).
- Điều kiện tiên quyết:**
 - Người dùng đã thêm ít nhất một sản phẩm vào giỏ hàng.
 - Thông tin khuyến mãi, giá sách đã được cập nhật chính xác.
- Kết quả mong muốn:**
 - Người dùng có thể xem, chỉnh sửa (số lượng), hoặc xóa sách trong giỏ hàng.
 - Tổng tiền thanh toán được cập nhật chính xác khi thay đổi số lượng hoặc áp dụng mã giảm giá.

6.3.4.2 Quy tắc và Ràng buộc

- Xem giỏ hàng: Hiển thị đầy đủ thông tin sách trong giỏ hàng bao gồm: tên sách, số lượng, giá, tổng giá trị.
- Xóa sách khỏi giỏ hàng:
 - Người dùng có thể xóa sách không mong muốn.
 - Hệ thống tự động cập nhật lại sau khi xóa sách.

- Áp dụng mã giảm giá:
 - Tự động chọn voucher tốt nhất cho khách hàng và áp dụng mã khuyến mãi từ người bán hoặc toàn sàn.
 - Hiển thị tổng tiền sau khi áp dụng khuyến mãi.
- Đặt mua hàng:
 - Hiển thị các thông tin: người nhận, địa chỉ, số điện thoại.
 - Tổng tiền cuối cùng hiển thị rõ ràng trước khi đặt hàng.

6.3.4.3 Mô tả chi tiết

STT	Tên	Mô tả	Loại điều khiển	Kiểu dữ liệu	Giá trị mặc định	Bắt buộc	Quy tắc
1	Danh sách sách	Hiển thị sách trong giỏ hàng kèm thông tin số lượng và giá.	Table	varchar(255)		X	Mỗi hàng đại diện một sách, bao gồm hình ảnh, tên, giá, và nút xóa.
2	Xóa sách	Xóa một sách khỏi giỏ hàng.	Button				Cập nhật tổng tiền sau khi xóa sách.
3	Áp dụng mã giảm giá	Nhập mã giảm giá từ người bán hoặc toàn sàn.	Input textbox	varchar(50)			Xác nhận mã hợp lệ trước khi áp dụng.
4	Tổng tiền thanh toán	Hiển thị tổng tiền sau khi áp dụng khuyến mãi.	Hiển thị thông tin	decimal(10,2)		X	Tổng tiền = Tổng giá sách - Khuyến mãi.
5	Đặt mua hàng	Xác nhận đơn hàng và thông tin thanh toán.	Button			X	Yêu cầu thông tin người nhận và địa chỉ chính xác.

Bảng 49: Mô tả chi tiết Trang giỏ hàng

6.3.4.4 Quy trình xử lý

1. Xem giỏ hàng:
 - a. Người dùng truy cập giỏ hàng để xem danh sách sách đã thêm.
 - b. Hệ thống hiển thị các sách kèm thông tin chi tiết (tên, giá, số lượng).
2. Cập nhật giỏ hàng:
 - a. Người dùng có thể thay đổi số lượng sách bằng cách nhập trực tiếp hoặc chọn tăng/giảm.
 - b. Hệ thống tự động cập nhật tổng giá trị của giỏ hàng.
3. Xóa sách:
 - a. Khi người dùng nhấn nút Xóa, sách sẽ bị loại bỏ khỏi giỏ.
 - b. Hệ thống tính lại tổng tiền thanh toán.
4. Áp dụng mã giảm giá:

- a. Người dùng nhập mã giảm giá và nhấn Áp dụng.
 - b. Hệ thống kiểm tra mã giảm giá hợp lệ, sau đó cập nhật tổng tiền với số tiền giảm.
5. Xác nhận đặt hàng:
- a. Người dùng kiểm tra lại thông tin đặt hàng (địa chỉ, số điện thoại, tên người nhận).
 - b. Nhấn nút Đặt mua hàng để xác nhận.

6.3.4.5 ĐẶC TẢ API

Sử dụng lại Api lấy chi tiết bài đăng ở trên

API: Tìm kiếm voucher session theo từ khóa

- **Endpoint:** /api/voucher-session/search
- Phương thức: GET
- **Mô tả:** Tìm kiếm voucher session dựa trên từ khóa được cung cấp.
- Tham số đầu vào:

Tên tham số	Loại dữ liệu	Mô tả	Bắt buộc
keyword	string	Từ khóa dùng để tìm kiếm voucher session	Có

Bảng 50: Tham số đầu vào API Tìm kiếm voucher session theo từ khóa

API: Tìm kiếm voucher shop theo từ khóa

- **Endpoint:** /api/voucher-shop/search
- Phương thức: GET
- **Mô tả:** Tìm kiếm voucher shop dựa trên từ khóa được cung cấp.
- Tham số đầu vào:

Tên tham số	Loại dữ liệu	Mô tả	Bắt buộc
keyword	string	Từ khóa dùng để tìm kiếm voucher shop	Có

Bảng 51: Tham số đầu vào API Tìm kiếm voucher shop theo từ khóa

6.3.5 Trang đặt hàng

The screenshot shows the booking interface for 'The Flying Bookstore'. At the top, there is a search bar with placeholder text 'Tim sach...' and a magnifying glass icon. To the right of the search bar are icons for account, cart, and document.

Below the search bar, there are three numbered steps: 1. Điền thông tin (Fill information), 2. Xuất đơn hàng (Create order), and 3. Lấy hàng (Get the item).

Thông tin đặt mua (Purchase information) includes:

- Họ và tên lót *: Nguyễn Lê Xuân
- Tên *: Hòa
- Email *: hoa@gmail.com
- Ngày sinh *: 14/04/1999
- Số điện thoại *: 0905907362
- Địa chỉ *: 12 Lý Chính Thắng
- Chọn thanh toán rồi tạo đơn hàng
-

Thông tin sản phẩm (Product information) shows a book titled 'At the Center of the Storm' by George Tenet, with a quantity of 1, unit price of 138.460đ, and total price of 197.890đ.

Mã giảm giá (Discount codes) lists two options:

- Khuyến Mãi từ người bán: Giảm 10.000đ (For orders over 50.000đ) - BỎ CHỌN
- Khuyến Mãi toàn sàn: Giảm 50.000đ (For orders over 400.000đ) - BỎ CHỌN

Thanh toán (Payment) shows the total amount as 78.460đ and payment methods:

- Đưa tiền trực tiếp
- Chuyển khoản
- Thanh toán VNPay

Quy trình thuê sách (Rental process):

- Đến nhận sách**: Includes an icon of a map and text: 'Sau khi thanh toán thành công, bạn có thể chọn 1 trong 2 cách sau:
- Ban hãy đến địa chỉ thuê sách và đưa mã đơn hàng chủ sách để lấy sách ban nhé.
- Ban liên hệ chủ sách và đưa mã đơn hàng cho chủ sách, sau đó bạn tự đặt ship và thanh toán tiền ship để ship sách về.'
- Đọc sách**: Includes an icon of a book and text: 'Gìn giữ sách cẩn thận bạn nhé.
Hãy nhớ sau bạn còn có người khác đọc nữa, nên tránh việc viết vẽ hay làm rách sách bạn nhé.'
- Trả sách**: Includes an icon of a calendar and text: 'Vui lòng trả cuốn sách đúng hạn, bạn có thể đặt ship trả sách nhưng thời gian ship có thể không được quá thời gian trả sách. Sau khi chủ sách nhận lại được sách, số tiền còn lại sẽ được hoàn về tài khoản của bạn. Nếu bạn cần thêm thời gian, bạn có thể giá hạn thêm hoặc mua cuốn sách đó luôn chỉ với 1 click.'

Nhập mail để nhận được những quà tặng đặc biệt nhé !



The Flying Bookstore là nơi có thể giúp bạn học sinh/ sinh viên đếm cho thuê, mượn tài liệu của mình. Đồng thời giúp quản lý các tài liệu mình cho thuê, mượn.



Liên hệ

171/11 Trường Phước Phan, Bình Hưng Hòa, Bình Tân, TP.HCM
+123 345 123
support@flying bookstore

Chính sách

Chính sách bảo mật
Hướng dẫn về tình trạng sách



Hình 43: Trang đặt hàng

6.3.5.1 Tổng quan

- Mục đích:** Người dùng có thể xem lại và xác nhận các thông tin đặt thuê sách trước khi tạo đơn hàng chính thức.
- Tác nhân tham gia:** Người dùng đã chọn sách để thuê và đã đăng nhập vào hệ thống.
- Điều kiện tiên quyết:** Người dùng đã thêm sách vào giỏ hàng và đã có đầy đủ thông tin cá nhân để xác nhận đơn hàng.
- Kết quả mong muốn:** Người dùng có thể tạo thành công đơn hàng thuê sách

6.3.5.2 Quy tắc và Ràng buộc

- Thông tin người dùng:** Bao gồm Họ tên, Email, Ngày sinh, Số điện thoại, Địa chỉ (được lấy từ thông tin cá nhân của người dùng).
- Thông tin sản phẩm mua:** Tên sách, Số lượng, Giá thuê và Tiền cọc cần được hiển thị chi tiết để người dùng kiểm tra.
- Phương thức thanh toán:** Người dùng có thể chọn giữa các phương thức thanh toán như Tiền mặt, Chuyển khoản, hoặc Thanh toán qua ví điện tử.

6.3.5.3 Mô tả chi tiết

STT	Tên	Mô tả	Loại điều khiển	Kiểu dữ liệu	Giá trị mặc định	Bắt buộc	Quy tắc
1	Họ và tên lót	Phần họ và tên lót của người dùng	Textbox	varchar(255)	Lấy từ cơ sở dữ liệu	X	Hệ thống tự động điền từ thông tin cá nhân.
2	Tên	Phần tên của người dùng	Textbox	varchar(255)	Lấy từ cơ sở dữ liệu	X	Hệ thống tự động điền từ thông tin cá nhân.
3	Email	Địa chỉ email của người dùng	Textbox	varchar(255)	Lấy từ cơ sở dữ liệu	X	Email hợp lệ được tự động điền từ thông tin cá nhân.
4	Ngày sinh	Ngày tháng năm sinh của người dùng	Date picker	date	Lấy từ cơ sở dữ liệu		Phải là một ngày hợp lệ.
5	Số điện thoại	Số điện thoại liên lạc của người dùng	Textbox	varchar(11)	Lấy từ cơ sở dữ liệu	X	Phải là số điện thoại hợp lệ.
6	Địa chỉ	Địa chỉ hiện tại của người dùng	Textbox	varchar(255)	Lấy từ cơ sở dữ liệu		Người dùng có thể thay đổi nếu cần.
7	Thông tin sản phẩm	Hiển thị thông tin sách đang thuê					Bao gồm tên sách, số lượng, giá thuê và tiền cọc.
8	Phương thức thanh toán	Các lựa chọn thanh toán cho người dùng	Button group		Mặc định chọn "Tiền mặt"	X	Người dùng có thể chọn thanh toán qua tiền mặt, chuyển khoản, hoặc qua ví điện tử.
9	Tổng tiền thanh toán	Tổng tiền phải trả bao gồm tiền thuê và tiền cọc	Label	Decimal	Tính toán tự động		Tự động tính tổng dựa trên thông tin thuê sách.

10	Tạo đơn hàng	Nút để tạo đơn hàng	Button			X	Khi nhấn, hệ thống sẽ xác nhận đơn hàng và chuyển đến trang thanh toán.
----	--------------	---------------------	--------	--	--	---	---

Bảng 52: Mô tả chi tiết Trang đặt hàng

6.3.5.4 Quy trình xử lý

1. Người dùng truy cập vào trang "Thông tin đặt thuê".
2. Hệ thống hiển thị thông tin cá nhân của người dùng và chi tiết sách đang thuê, bao gồm tên sách, số lượng, giá thuê, và tiền cọc.
3. Người dùng có thể chọn phương thức thanh toán: Tiền mặt, Chuyển khoản, hoặc Thanh toán qua VNPay.
4. Sau khi kiểm tra kỹ thông tin, người dùng nhấn nút "Tạo đơn hàng" để xác nhận đơn hàng.
5. Hệ thống kiểm tra tính hợp lệ của thông tin và chuyển hướng người dùng đến trang xác nhận thanh toán.
6. Nếu có lỗi (ví dụ thông tin chưa đủ), hệ thống sẽ thông báo cho người dùng chỉnh sửa lại trước khi tiếp tục.

6.3.5.5 Đặc tả API

API: Tạo đơn hàng bán

- **Endpoint:** /api/SaleOrder/createSaleOrder
- Phương thức: POST
- **Mô tả:** API này cho phép tạo mới một đơn hàng bán.
- Tham số đầu vào:

Tên tham số	Loại dữ liệu	Bắt buộc	Mô tả
status	string	Có	Trạng thái của đơn hàng.
listingId	number	Có	ID của danh sách liên quan đến đơn hàng.
buyerAddress	string	Có	Địa chỉ của người mua.
paymentMethod	string	Có	Phương thức thanh toán cho đơn hàng.
VoucherShopId	string	Không	ID của voucher shop.
VoucherSessionId	string	Không	ID của voucher session.

Bảng 53: Tham số đầu vào API Tạo đơn hàng bán

API: Cập nhật thông tin người dùng

- **Endpoint:** /api/user/{userId}
- Phương thức: PUT
- **Mô tả:** API này cho phép cập nhật thông tin chi tiết của người dùng.
- Tham số đầu vào:

Tên tham số	Loại dữ liệu	Bắt buộc	Mô tả
id	number	Có	ID duy nhất của người dùng.

username	string	Có	Tên đăng nhập của người dùng.
email	string	Có	Địa chỉ email của người dùng.
phoneNumber	string	Có	Số điện thoại của người dùng.
firstName	string	Có	Tên của người dùng.
lastName	string	Có	Họ của người dùng.
birthDate	string	Có	Ngày sinh của người dùng.
avatarUrl	string	Không	URL ảnh đại diện của người dùng.
address	string	Có	Địa chỉ của người dùng.
password	string	Có	Mật khẩu của người dùng.

Bảng 54: Tham số đầu vào API Cập nhật thông tin người dùng

Đối với thanh toán chuyển khoản giao diện cung cấp thêm thông tin chuyển khoản cho khách đặt thuê:

Thanh toán



Bước 1: Thanh toán

Cách 1: Bạn có thể quét mã QR ở dưới đây:



Cách 2: Bạn chuyển khoản theo thông tin sau:

Số tài khoản: 0004100030828002
 Chủ tài khoản: NGUYEN LE XUAN HOA
 Ngân hàng: Thương mại cổ phần Phương Đông-OCB
 Chi nhánh: PCD LÝ THƯỜNG KIỆT
 Tỉnh/TP: TP.Hồ Chí Minh

Bước 2: Tin nhắn chuyển khoản theo cú pháp:

<Tên khách hàng>_<Tổng tiền>_Thanh toán cọc sách

Bước 3: Chờ quản trị xác nhận

Hình 44: Phần thông báo chuyển khoản trang đặt hàng

6.3.6 Trang xuất đơn hàng

The screenshot shows the website interface for The Flying Bookstore. At the top, there is a navigation bar with icons for account, cart, and search. Below the header, there are three main sections:

- Đơn hàng được tạo thành công!**: Displays the order details for "At the Center of the Storm" by George Tenet, quantity 1, at a price of 138.460đ.
- Thông tin đặt hàng**: Shows the order information including delivery method (COD), payment method (Giá bán: 138.460đ), and discounts (Khuyến mãi từ người bán: 10.000đ, Khuyến mãi từ The Flying Bookstore: 50.000đ).
- Quy trình thuê sách**: Describes the rental process with three steps: Đến nhận sách (Delivery), Đọc sách (Read), and Trả sách (Return). It includes icons and explanatory text for each step.

At the bottom, there is a call-to-action banner encouraging users to enter their email to receive special gifts, followed by footer links for contact, policies, and a map.

Hình 45: Trang xuất đơn hàng

6.3.6.1 Tổng quan

- Mục đích:** Người dùng có thể xem thông tin chi tiết của đơn hàng vừa đặt, bao gồm tên sách, số lượng, phương thức thanh toán và trạng thái đơn hàng.
- Tác nhân tham gia:** Người dùng đã đăng nhập và vừa hoàn tất đặt hàng.
- Điều kiện tiên quyết:** Người dùng đã đăng nhập vào hệ thống và hoàn tất việc đặt sách.

- **Kết quả mong muốn:** Người dùng xem được thông tin chi tiết của đơn hàng mới đặt thành công.

6.3.6.2 Quy tắc và Ràng buộc

- **Đơn hàng:** Thông tin của đơn hàng bao gồm mã đơn hàng, ngày đặt, tên sách, số lượng, giá thành, phương thức thanh toán, và trạng thái đơn hàng.
- **Thông tin giao hàng:** Bao gồm địa chỉ nhận sách, số điện thoại liên hệ của người thuê, và thời gian thuê sách.
- **Tùy chọn:** Người dùng có thể quản lý đơn hàng hoặc tiếp tục mua sách mới.

6.3.6.3 Mô tả chi tiết

STT	Tên	Mô tả	Loại điều khiển	Kiểu dữ liệu	Giá trị mặc định	Bắt buộc	Quy tắc
1	Mã đơn hàng	Mã đơn hàng hiển thị sau khi đặt hàng thành công	Textbox	varchar(255)		X	Mã đơn hàng được tạo tự động sau khi đặt hàng.
2	Tên sách	Tên sách mà người dùng vừa đặt	Textbox	varchar(255)		X	Hiển thị tên sách vừa đặt.
3	Số lượng	Số lượng sách mà người dùng đã đặt	Textbox	Integer	1	X	Hiển thị số lượng sách vừa đặt.
4	Giá thành	Giá tiền cho mỗi cuốn sách	Textbox	Float		X	Hiển thị tổng giá trị đơn hàng.
5	Phương thức thanh toán	Phương thức thanh toán mà người dùng đã chọn	Textbox	varchar(50)	COD	X	Hiển thị phương thức thanh toán đã chọn (COD).
6	Trạng thái	Trạng thái của đơn hàng (Chờ thanh toán, Đang giao, Hoàn thành, Hủy bỏ)	Textbox	varchar(50)	Chờ thanh toán	X	Hiển thị trạng thái đơn hàng hiện tại.
7	Địa chỉ giao hàng	Địa chỉ mà người dùng đã chọn để nhận sách	Textbox	varchar(255)		X	Hiển thị địa chỉ nhận sách.

8	Số điện thoại liên hệ	Số điện thoại liên hệ của người thuê sách	Textbox	varchar(20)		X	Hiển thị số điện thoại của người dùng.
9	Thời gian thuê	Thời gian người dùng thuê sách	Textbox	Integer		X	Hiển thị thời gian thuê sách.
10	Khuyến mãi người bán	Khuyến mãi từ người bán	Textbox	varchar(11)			Mã khuyến mãi từ người bán còn hiệu lực
11	Khuyến mãi toàn sàn	Khuyến mãi từ The Flying Bookstore	Textbox	varchar(11)			Mã khuyến mãi còn hiệu lực

Bảng 55: Mô tả chi tiết Trang xuất đơn hàng

6.3.6.4 Quy trình xử lý

1. Người dùng hoàn tất việc đặt sách và hệ thống hiển thị trang xác nhận đơn hàng.
2. Người dùng xem thông tin chi tiết về đơn hàng, bao gồm:
 - Mã đơn hàng.
 - Tên sách.
 - Số lượng sách.
 - Giá thành và tổng tiền cần thanh toán.
 - Phương thức thanh toán (COD).
 - Địa chỉ giao hàng và số điện thoại liên hệ.
 - Thời gian thuê sách.
 - Trạng thái đơn hàng (Chờ thanh toán).
3. Người dùng có thể chọn "Tiếp tục mua sách" hoặc "Quản lý đơn hàng" để kiểm tra lại đơn hàng của mình.

6.3.6.5 ĐẶC TẢ API

API: Lấy thông tin đơn hàng bán

- **Endpoint:** /api/SaleOrder/{saleOrderId}
- Phương thức: GET
- **Mô tả:** API này trả về thông tin chi tiết của một đơn hàng bán cụ thể dựa trên ID được cung cấp.
- Tham số đầu vào:

Tên tham số	Loại dữ liệu	Mô tả	Bắt buộc
saleOrderId	number	Id của đơn hàng bán	Có

Bảng 56: Tham số đầu vào API Lấy thông tin đơn hàng bán

6.3.7 Trang lấy đơn hàng

Lấy thành công đơn hàng!

Đọc sách vui bạn nhé!

Giờ hàng

Thông tin đặt hàng

Mã đơn hàng	20	Trạng thái	Đã lấy hàng
Ngày đặt thuê	19/05/2024	Phương thức thanh toán	Chuyển khoản
Người cho thuê	Braun Tracy	Số ngày thuê	1
Thời gian thuê	19/05/2024 - 19/05/2024	Tiền thuê	2.000đ
Địa chỉ	Đường Kim Giang, Phường Kim Giang, Quận Thanh Xuân, Hà Nội	Tiền cọc	120.300đ
Số điện thoại	0905 907 362	Tổng cộng	122.300đ

Quy trình thuê sách

- Đến nhận sách
- Đọc sách
- Trả sách

Nhập mail để nhận được những quà tặng đặc biệt nhé!

Liên hệ

- 171/11 Trường Phước Phan, Bình Hưng Hòa, Bình Tân, TP.HCM
- +123 345 123
- support@flying bookstore

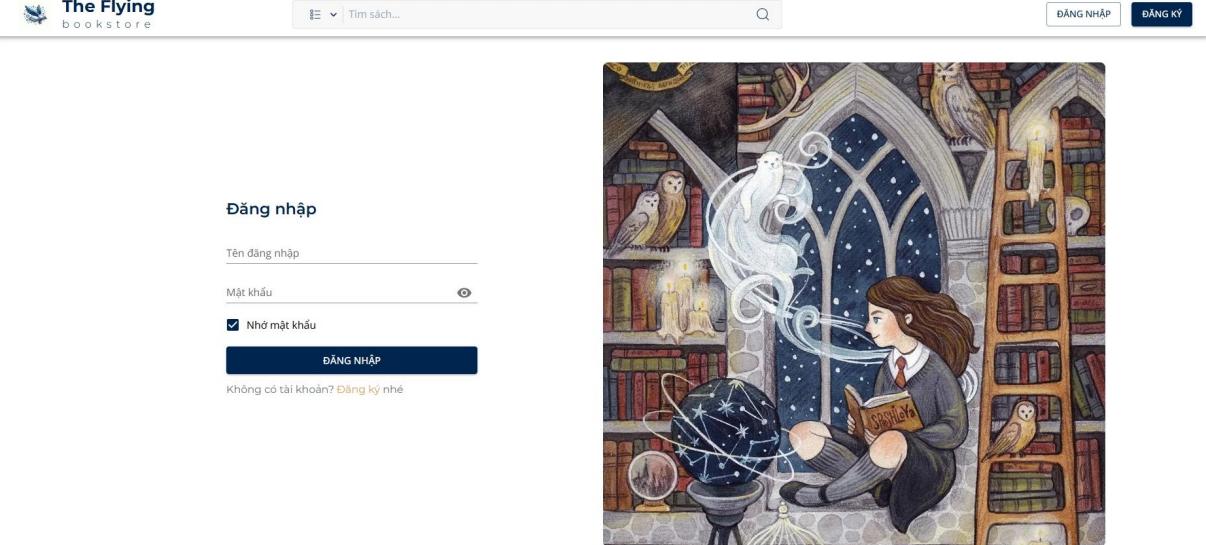
The Flying Bookstore là nơi có thể giúp bạn sinh/sinh viên tìm kiếm, mượn tài liệu của mình. Đồng thời giúp quản lý các tài liệu mượn cho thuê, mượn.

[Nhập mail tại đây](#)

Hình 46: Trang lấy đơn hàng

Trang lấy đơn hàng chứa thông tin đơn hàng và các nút tiếp tục mua sắm để bạn quay lại tiếp tục tìm sách và thuê sách, nút quản lý đơn hàng để về trang quản lý đơn hàng. Giao diện giống như trang xuất đơn hàng nhưng đổi trạng thái và status.

6.3.8 Trang đăng nhập



The screenshot shows the login page of a website called "The Flying bookstore". At the top, there is a logo, a search bar with the placeholder "Tim sách...", and two buttons: "ĐĂNG NHẬP" (Login) and "ĐĂNG KÝ" (Register). The main form is titled "Đăng nhập" (Login). It contains fields for "Tên đăng nhập" (Username) and "Mật khẩu" (Password), both with placeholder text. There is also a checked checkbox for "Nhớ mật khẩu" (Remember password) and a "ĐĂNG NHẬP" (Login) button. Below the form, a link says "Không có tài khoản? Đăng ký nhé". To the right of the form is a decorative illustration of a girl reading a book in a magical library filled with books and owls.

Hình 47: Trang đăng nhập

Ngoài form đăng ký thì trang còn trang trí bằng hình ảnh thơ mộng về tủ sách thiên thần nhằm thể hiện tinh thần của dự án.

6.3.8.1 Tổng quan

- Mục đích:** Cho phép người dùng đăng nhập để truy cập vào tài khoản cá nhân và sử dụng các tính năng như thuê sách, quản lý giỏ hàng và đơn hàng.
- Đối tượng tham gia:** Người dùng.
- Điều kiện tiên quyết:** Người dùng đã có tài khoản trong hệ thống.
- Kết quả mong muốn:** Người dùng đăng nhập thành công và được chuyển hướng đến trang chủ hoặc trang cá nhân.

6.3.8.2 Quy tắc và Ràng buộc

- Người dùng phải nhập đúng tên đăng nhập và mật khẩu.
- Mật khẩu sẽ được ẩn bằng ký tự "*".
- Hệ thống sẽ hiển thị thông báo lỗi nếu:
 - Tên đăng nhập không tồn tại.
 - Mật khẩu không đúng.
- Nếu chọn "Nhớ mật khẩu", hệ thống sẽ lưu thông tin đăng nhập trên thiết bị của người dùng.

6.3.8.3 Mô tả chi tiết

STT	Tên	Mô tả	Loại điều khiển	Kiểu dữ liệu	Giá trị mặc định	Bắt buộc	Quy tắc
1	Tên đăng nhập	Tên tài khoản người dùng	Textfield	Chuỗi	N/A	X	Phải nhập đầy đủ
2	Mật khẩu	Mật khẩu tài khoản	Password field	Chuỗi	N/A	X	Phải nhập đúng

3	Nhớ mật khẩu	Lưu thông tin đăng nhập trên trình duyệt	Checkbox	Boolean	Bỏ chọn		-
4	Đăng nhập	Thực hiện đăng nhập	Button	Nút	N/A	X	Chỉ hoạt động khi đủ thông tin

Bảng 57: Mô tả chi tiết Trang đăng nhập

6.3.8.4 Quy trình xử lý

1. Người dùng truy cập vào trang đăng nhập.
2. Nhập thông tin tên đăng nhập và mật khẩu.
3. Tùy chọn Nhớ mật khẩu nếu muốn lưu thông tin đăng nhập.
4. Nhấn nút Đăng nhập để gửi yêu cầu đăng nhập.
5. Hệ thống kiểm tra thông tin:
 - o Nếu thông tin hợp lệ:
 - Hiển thị thông báo Đăng nhập thành công.
 - Chuyển hướng đến trang chính.
 - o Nếu thông tin không hợp lệ:
 - Hiển thị thông báo lỗi chi tiết (Ví dụ: "Tên đăng nhập không tồn tại").

6.3.8.5 Đặc tả API

API: Đăng nhập người dùng

- Phương thức: POST
- Endpoint: /api/login
- Tham số đầu vào:

Tên tham số	Loại dữ liệu	Mô tả	Bắt buộc
username	varchar(255)	Tên đăng nhập	Có
password	varchar(255)	Mật khẩu	Có

Bảng 58: Tham số đầu vào API Đăng nhập người dùng

6.3.9 Trang đăng ký

Nhập mail để nhận được những quà tặng đặc biệt nhé !

hoaminzy

Liên hệ

- 171/11 Trường Phước Phan, Bình Hưng Hòa, Bình Tân, TP.HCM
- +84 123 345 123
- support@flying bookstore

Chính sách

- Chính sách bảo mật
- Hướng dẫn về tình trạng sách

Địa chỉ: Tiệm Giường Huyền Cảnh - Xem bản đồ lớn hơn

Hình 48: Trang đăng ký

6.3.9.1 Tổng quan

- Mục đích:** Cung cấp tính năng cho khách hàng tạo tài khoản mới để truy cập vào hệ thống và sử dụng các dịch vụ như thuê sách, mua sách, và quản lý tài khoản cá nhân.
- Đối tượng tham gia:** Khách hàng.
- Điều kiện tiên quyết:** Khách chưa có tài khoản trên hệ thống.
- Kết quả mong muốn:** Khách hàng đăng ký thành công tài khoản và nhận được thông tin xác nhận.

6.3.9.2 Quy tắc và Ràng buộc

- Mọi trường bắt buộc phải được điền đầy đủ và chính xác.
- Mật khẩu phải đáp ứng yêu cầu về bảo mật (ít nhất 8 ký tự, bao gồm chữ hoa, chữ thường, và số).
- Người dùng phải đồng ý với Điều khoản bảo mật để hoàn tất đăng ký.

6.3.9.3 Mô tả chi tiết

STT	Tên	Mô tả	Loại điều khiển	Kiểu dữ liệu	Giá trị mặc định	Bắt buộc	Quy tắc
-----	-----	-------	-----------------	--------------	------------------	----------	---------

1	Tên đăng nhập	Tên tài khoản của người dùng	Textfield	Chuỗi	N/A	X	Phải là duy nhất
2	Mật khẩu	Mật khẩu tài khoản	Password field	Chuỗi	N/A	X	Tối thiểu 8 ký tự
3	Nhập lại mật khẩu	Xác nhận lại mật khẩu	Password field	Chuỗi	N/A	X	Phải trùng khớp với trường Mật khẩu
4	Điều khoản bảo mật	Đồng ý với điều khoản bảo mật	Checkbox	Boolean	Bỏ chọn	X	Không thể bỏ qua
5	Đăng ký	Nút gửi yêu cầu đăng ký	Button	Nút	N/A	X	Hoạt động khi đủ thông tin

Bảng 59: Mô tả chi tiết Trang đăng ký

6.3.9.4 Quy trình xử lý

- Khách hàng truy cập vào trang đăng ký.
- Điền đầy đủ thông tin trong các trường bắt buộc:
 - Tên đăng nhập.
 - Mật khẩu và xác nhận mật khẩu.
- Đọc và chọn vào checkbox “Đồng ý với Điều khoản bảo mật”.
- Nhấn nút “Đăng ký” để gửi yêu cầu tạo tài khoản.
- Hệ thống kiểm tra thông tin:
 - Nếu thông tin hợp lệ:
 - Hiển thị thông báo “Đăng ký thành công”.
 - Lưu tài khoản vào cơ sở dữ liệu.
 - Gửi email xác nhận cho người dùng.
 - Nếu thông tin không hợp lệ:
 - Hiển thị lỗi chi tiết (Ví dụ: "Tên đăng nhập đã tồn tại").

6.3.9.5 Đặc tả API

API: Đăng ký tài khoản mới

- Phương thức: POST
- Endpoint: /api/user/register
- Tham số đầu vào:

Tên tham số	Loại dữ liệu	Mô tả	Bắt buộc
username	varchar(255)	Tên đăng nhập	Có
email	varchar(255)	Địa chỉ email	Có
phoneNumber	varchar(255)	Số điện thoại	Có
firstName	varchar(255)	Tên	Có

lastName	varchar(255)	Họ	Có
birthDate	Date	Ngày sinh	Có
avatarUrl	varchar(255)	URL ảnh đại diện	Không
address	varchar(255)	Địa chỉ	Có
password	varchar(255)	Mật khẩu	Có

Bảng 60: Tham số đầu vào API Đăng ký tài khoản mới

6.3.10 Trang quản lý thông tin cá nhân

The screenshot shows a user interface for managing personal information. On the left, there's a sidebar with icons for account setup, order management, customer management, login, and add new. The main area is titled 'Thông tin cá nhân' (Personal Information). It contains fields for 'Họ và tên lót' (Name), 'Email', 'Ngày sinh' (Date of Birth), and 'Địa chỉ' (Address), each with its own placeholder value. At the bottom is a large 'XÁC NHẬN' (Confirm) button.

Hình 49: Trang quản lý thông tin cá nhân

Vừa cung cấp thông tin cá nhân vừa có thể chỉnh sửa thông tin cá nhân. Để phục vụ cho việc đặt thuê.

6.3.10.1 Tổng quan

- Mục đích:** Người dùng có thể chỉnh sửa các thông tin cá nhân như tên, email, ngày sinh, số điện thoại, và địa chỉ trực tiếp trong trang tài khoản của mình.
- Tác nhân tham gia:** Người dùng đã đăng nhập vào hệ thống.
- Điều kiện tiên quyết:** Người dùng đã có tài khoản và đã đăng nhập thành công vào hệ thống.
- Kết quả mong muốn:** Người dùng có thể cập nhật thành công các thông tin cá nhân của mình.

6.3.10.2 Quy tắc và Ràng buộc

- Các trường bắt buộc:** Tên, Email, Số điện thoại.
- Quy tắc Email:** Phải là định dạng email hợp lệ (ví dụ: user@example.com).
- Quy tắc số điện thoại:** Phải là một chuỗi số hợp lệ có độ dài từ 10 đến 11 chữ số.
- Ngày sinh:** Phải là một ngày hợp lệ trong định dạng ngày/tháng/năm (dd/mm/yyyy).

6.3.10.3 Mô tả chi tiết

STT	Tên	Mô tả	Loại điều khiển	Kiểu dữ liệu	Giá trị mặc định	Bắt buộc	Quy tắc
1	Họ và tên lót	Phần họ và tên lót của người dùng	Textbox	varchar(255)	Lấy từ cơ sở dữ liệu	X	Hệ thống tự động điền dữ liệu hiện tại của

							người dùng từ cơ sở dữ liệu.
2	Tên	Phần tên của người dùng	Textbox	varchar(255)	Lấy từ cơ sở dữ liệu	X	Người dùng nhập tên muốn chỉnh sửa.
3	Email	Địa chỉ email của người dùng	Textbox	varchar(255)	Lấy từ cơ sở dữ liệu	X	Phải là email hợp lệ.
4	Ngày sinh	Ngày tháng năm sinh của người dùng	Date picker	date	Lấy từ cơ sở dữ liệu		Phải là một ngày hợp lệ.
5	Số điện thoại	Số điện thoại liên lạc của người dùng	Textbox	varchar(11)	Lấy từ cơ sở dữ liệu	X	Phải là số điện thoại hợp lệ (10-11 chữ số).
6	Địa chỉ	Địa chỉ hiện tại của người dùng	Textbox	varchar(255)	Lấy từ cơ sở dữ liệu		Người dùng có thể cập nhật địa chỉ hiện tại.
7	Xác nhận	Nút để xác nhận thông tin đã thay đổi	Button			X	Khi nhấn nút, hệ thống sẽ kiểm tra tính hợp lệ và cập nhật thông tin mới.

Bảng 61: Mô tả chi tiết Trang quản lý thông tin cá nhân

6.3.10.4 Quy trình xử lý

1. Người dùng truy cập vào trang "Cài đặt tài khoản".
2. Các trường thông tin cá nhân của người dùng được hiển thị với giá trị hiện tại từ cơ sở dữ liệu.
3. Người dùng có thể chỉnh sửa các trường thông tin mong muốn như Tên, Email, Ngày sinh, Số điện thoại và Địa chỉ.
4. Sau khi hoàn thành việc chỉnh sửa, người dùng nhấn nút "Xác nhận" để lưu thay đổi.
5. Hệ thống kiểm tra tính hợp lệ của các thông tin:
 - a. Nếu hợp lệ, thông tin sẽ được cập nhật và hiển thị thông báo thành công.
 - b. Nếu có lỗi, hệ thống sẽ thông báo và yêu cầu người dùng chỉnh sửa lại các trường sai sót.

6.3.10.5 Đặc tả API

API: Cập nhật thông tin người dùng

- **Endpoint:** /api/user/{id}
- Phương thức: PUT

- **Mô tả:** API này cho phép cập nhật thông tin chi tiết của người dùng.
- **Xác thực:** Yêu cầu Bearer Token trong phần **Authorization Header**.
- Tham số đầu vào:

Tên tham số	Loại dữ liệu	Mô tả	Bắt buộc
id	number	ID duy nhất của người dùng	Có
username	varchar(255)	Tên đăng nhập	Có
email	varchar(255)	Địa chỉ email	Có
phoneNumber	varchar(255)	Số điện thoại	Có
firstName	varchar(255)	Tên riêng của người dùng	Có
lastName	varchar(255)	Họ của người dùng	Có
birthDate	Date	Ngày sinh	Có
avatarUrl	varchar(255)	URL ảnh đại diện	Không
address	varchar(255)	Địa chỉ của người dùng	Có
password	varchar(255)	Mật khẩu của người dùng	Có

Bảng 62: Tham số đầu vào API Cập nhật thông tin người dùng

6.3.11 Trang quản lý đơn thuê

Nhập mail để nhận được những quà tặng đặc biệt nhé !

Hình 50: Trang quản lý đơn thuê

Chứa danh sách đơn hàng của user đi thuê sách của người khác. Mỗi đơn hàng có một trạng thái riêng và có những nút tương ứng với trạng thái đó. Ta có thể xem chi tiết đơn hàng ở nút xem chi tiết ngay đầu trên bên phải mỗi đơn hàng. Mỗi cuốn sách trong đơn hàng có thể nhấp vào nút thông tin để điều hướng đến chi tiết cuốn sách đó.

Trang này hiển thị các trạng thái chung nhất của đơn hàng, với mỗi trạng thái ta hiển thị có bao nhiêu đơn hàng và được bấm nút tải lại để làm mới đơn hàng.

Như đơn hàng ở trạng thái chờ thanh toán này có thể cancel được:

Tổng tiền thuê: 2.000đ
Tổng tiền cọc: 120.300đ
Tổng tiền: 120.300đ

Hình 51: Đơn hàng có thể hủy

6.3.11.1 Tổng quan

- Mục đích:** Người dùng có thể quản lý, xem và theo dõi trạng thái các đơn hàng thuê sách của mình.
- Tác nhân tham gia:** Người dùng đã đăng nhập và có các đơn hàng thuê sách.
- Điều kiện tiên quyết:** Người dùng đã tạo một hoặc nhiều đơn hàng thuê sách trước đó.
- Kết quả mong muốn:** Người dùng có thể dễ dàng xem chi tiết từng đơn hàng và thực hiện các hành động phù hợp với từng trạng thái đơn hàng.

6.3.11.2 Quy tắc và Ràng buộc

- Hiển thị thông tin đơn hàng:** Bao gồm các chi tiết của từng đơn thuê sách như mã đơn hàng, sách đã thuê, thời gian thuê, tiền cọc, tiền thuê, và tổng tiền.
- Trạng thái đơn hàng:** Đơn hàng sẽ có các trạng thái như “Chờ thanh toán”, “Đã trả tiền cọc”, “Đã hủy”, và “Đã trả sách”.
- Các hành động khả dụng:** Tùy vào trạng thái của đơn hàng, các nút như “Đã Trả Tiền”, “Mua Sách”, và “Đánh Giá” sẽ xuất hiện để người dùng thao tác.

6.3.11.3 Mô tả chi tiết

STT	Tên	Mô tả	Loại điều khiển	Kiểu dữ liệu	Giá trị mặc định	Bắt buộc	Quy tắc
1	Id đơn hàng	Mã định danh cho đơn hàng thuê	Label	varchar(10)	Tạo tự động	X	Được hệ thống tạo khi đơn hàng được khởi tạo.
2	Tên sách	Tên sách mà người dùng thuê	Label	varchar(255)		X	Lấy từ cơ sở dữ liệu của đơn hàng.
3	Thời gian thuê	Thời gian thuê sách	Label	Date range		X	Hiển thị khoảng thời gian thuê sách.
4	Tiền cọc	Tiền cọc người dùng phải trả	Label	Decimal		X	Lấy từ dữ liệu đơn hàng.
5	Tiền thuê	Phí thuê sách	Label	Decimal		X	Lấy từ dữ liệu đơn hàng.
6	Tổng cộng	Tổng số tiền của đơn hàng	Label	Decimal		X	Tính tự động từ tiền thuê và tiền cọc.
7	Trạng thái đơn hàng	Trạng thái hiện tại của đơn hàng thuê	Label	Enum	“Chờ thanh toán”	X	Cập nhật tự động khi đơn hàng thay đổi trạng thái.

8	Thời hạn thanh toán	Thời gian tối đa cho người dùng thanh toán	Label	DateTime			Hiển thị ngày, giờ thanh toán cuối cùng cho đơn hàng.
9	Đánh giá	Người dùng có thể đánh giá đơn hàng sau khi hoàn tất	Button				Chỉ hiện nếu đơn hàng đã hoàn thành.
10	Đã trả tiền	Xác nhận người dùng đã thanh toán tiền cọc	Button				Chỉ hiện khi đơn hàng ở trạng thái "Chờ thanh toán".
11	Xem chi tiết	Xem thông tin chi tiết về đơn hàng	Button				Điều hướng người dùng đến trang chi tiết đơn hàng.

Bảng 63: Mô tả chi tiết Trang quản lý đơn thuê

6.3.11.4 Quy trình xử lý

1. Người dùng truy cập vào trang "Quản lý đơn thuê" từ menu.
2. Hệ thống hiển thị danh sách các đơn thuê sách của người dùng, phân loại thành các mục như "Tất cả", "Chờ đặt hàng", "Đã nhận", "Đã trả sách", "Đã hủy".
3. Người dùng có thể xem các chi tiết cơ bản của từng đơn hàng bao gồm mã đơn hàng, tên sách, thời gian thuê, tiền cọc, tiền thuê, tổng cộng, và trạng thái đơn hàng.
4. Người dùng có thể chọn từng đơn hàng để thực hiện các hành động phù hợp với trạng thái của nó:
5. Đơn hàng ở trạng thái chờ thanh toán có thể xác nhận đã thanh toán nếu bạn đã thanh toán:

Id đơn hàng #20	Chủ sách Braun Tracy	Thời gian thuê 19/05 - 19/05/2024	Trạng thái người thuê Chờ thanh toán	Xem chi tiết	
Id bài đăng	Tên sách	Thời gian thuê	Tiền cọc	Tiền thuê	Tổng cộng
7	Alchemy and Aca...	1 ngày	120,300	2,000	122,300 ⓘ
<input checked="" type="checkbox"/> Vui lòng thanh toán đơn hàng trong 24 giờ, nếu chuyển khoản thành công, bạn hãy nhấn nút Đã trả tiền			ĐÃ TRẢ TIỀN ⓘ		
			Tổng tiền thuê: 2.000đ Tổng tiền cọc: 120.300đ Tổng tiền: 120.300đ		

Hình 52: Đơn hàng có thể xác nhận thanh toán

Đơn hàng trạng thái đã lấy hàng có thể xác nhận lấy hàng nếu bạn đã lấy hàng từ người cho thuê:

Id đơn hàng #14	Chủ sách Developer Dev	Thời gian thuê 16/05 - 21/05/2024	Trạng thái người thuê Lấy hàng	Xem chi tiết	
Id bài đăng	Tên sách	Thời gian thuê	Tiền cọc	Tiền thuê	Tổng cọc
320	Bad Samaritans	5 ngày	247,000	13,500	260,500
<input checked="" type="checkbox"/> Vui lòng chỉ nhấn “đã nhận được hàng” khi đơn hàng đã được giao đến bạn và bạn đã nhận được hàng			ĐÃ NHẬN ĐƯỢC HÀNG		
			Tổng tiền thuê: 13.500đ Tổng tiền cọc: 247.000đ Tổng tiền: 247.000đ		

Hình 53: Đơn hàng xác nhận lấy hàng

Đơn hàng trạng thái đã lấy hàng có thể xác nhận trả sách cho chủ sách:

Id đơn hàng #14	Chủ sách Developer Dev	Thời gian thuê 16/05 - 21/05/2024	Trạng thái người thuê Đã lấy hàng	Xem chi tiết	
Id bài đăng	Tên sách	Thời gian thuê	Tiền cọc	Tiền thuê	Tổng cọc
320	Bad Samaritans	5 ngày	247,000	13,500	260,500
<input checked="" type="checkbox"/> Bạn còn 5 ngày nữa, bạn có muốn trả sách sớm?			ĐÃ TRẢ SÁCH		
			Tổng tiền thuê: 13.500đ Tổng tiền cọc: 247.000đ Tổng tiền: 247.000đ		

Hình 54: Đơn hàng xác nhận trả sách

- Nếu muốn xem chi tiết hơn về từng đơn hàng, người dùng nhấn vào nút “Xem chi tiết”.

6.3.11.5 Đặc tả API

API: Tìm kiếm đơn thuê theo người cho thuê

- Endpoint:** /api/leaseOrder/search/lessor/{lessorId}
- Phương thức: GET
- Mô tả:** Lấy danh sách đơn thuê sách chi tiết liên quan đến một người cho thuê cụ thể.
- Xác thực:** Yêu cầu Bearer Token trong phần **Authorization Header**.
- Tham số đầu vào:

Tên tham số	Loại dữ liệu	Mô tả	Bắt buộc
lessorId	number	ID của người cho thuê	Có

Bảng 64: Tham số đầu vào API Tìm kiếm đơn thuê theo người cho thuê

API: Tìm kiếm đơn thuê theo người thuê

- Endpoint:** /api/leaseOrder/search/lessee/{lesseeId}
- Phương thức: GET

- **Mô tả:** Lấy danh sách đơn thuê sách chi tiết liên quan đến một người thuê cụ thể.
- **Xác thực:** Yêu cầu Bearer Token trong phần **Authorization Header**.
- Tham số đầu vào:

Tên tham số	Loại dữ liệu	Mô tả	Bắt buộc
lessorId	number	ID của người thuê	Có

Bảng 65: Tham số đầu vào API Tìm kiếm đơn thuê theo người thuê

API: Cập nhật trạng thái đơn thuê

- **Endpoint:** /api/leaseOrder/edit/status
- Phương thức: GET
- **Mô tả:** Chính sửa trạng thái của đơn thuê dựa trên ID và trạng thái mới.
- **Xác thực:** Yêu cầu Bearer Token trong phần **Authorization Header**.
- Giá trị hợp lệ cho tham số status:
 - RETURNED: Người cho thuê đã nhận lại.
 - RETURNING: Người đi thuê đã bấm trả sách.
 - CANCELED: Người thuê sách hủy đơn hàng.
 - PAYMENT_SUCCESS: Người thuê trả tiền thành công.
 - DELIVERED: Đã giao sách đến tay người thuê.
 - USER_PAID: Người thuê xác nhận đã trả tiền.
 - PAID_OWNER: Đã trả tiền cho chủ sách.
 - DEPOSIT_RETURNED: Đã hoàn tiền đặt cọc cho người thuê.
 - ORDERED_PAYMENT_PENDING: Đơn hàng đang chờ thanh toán.
- Tham số đầu vào:

Tên tham số	Loại dữ liệu	Bắt buộc	Mô tả
id	number	Có	ID của đơn thuê cần chỉnh sửa
status	string	Có	Trạng thái mới của đơn thuê

Bảng 66: Tham số đầu vào API Cập nhật trạng thái đơn thuê

6.3.12 Trang quản lý đơn hàng cho thuê

Quản lý đơn hàng cho thuê

TẤT CẢ ĐÃ ĐẶT HÀNG ĐÃ NHÂN ĐÃ QUÁ HẠN ĐÃ TRẢ SÁCH ĐÃ HỦY

Bạn hiện có 9 đơn hàng C TÀI LẠI

Id đơn hàng	Người thuê	Thời gian thuê	Trạng thái	Xem chi tiết		
#2	Developer Dev	28/05 - 09/07/2024	Chờ khách thanh toán	Xem chi tiết		
Id bài đã...	Tên sách	Thời gian thuê	Tiền cọc	Tiền thuê	Tổng cộng	
2	From Potter's Fi...	42 ngày	414.800	9.000	414.800	
✓ Người thuê mới đặt hàng, chờ người thuê thanh toán Đã trả tiền Mua sách X						
						Tổng tiền thuê: 9.000đ Tổng tiền cọc: 414.800đ Tổng tiền: 414.800đ
Id đơn hàng	Người thuê	Thời gian thuê	Trạng thái	Xem chi tiết		
#3	Developer Dev	01/03 - 01/04/2024	Khách đã hủy	Xem chi tiết		
Id bài đã...	Tên sách	Thời gian thuê	Tiền cọc	Tiền thuê	Tổng cộng	
3	Deep Waters	31 ngày	90.000	84.000	90.000	
						Tổng tiền thuê: 84.000đ Tổng tiền cọc: 90.000đ Tổng tiền: 90.000đ
Id đơn hàng	Người thuê	Thời gian thuê	Trạng thái	Xem chi tiết		
#4	Developer Dev	26/11 - 26/12/2024	Chờ khách lấy hàng	Xem chi tiết		
Id bài đã...	Tên sách	Thời gian thuê	Tiền cọc	Tiền thuê	Tổng cộng	
3	Deep Waters	30 ngày	90.000	84.000	90.000	
✓ Người thuê đã thanh toán thành công, chủ sách cần chuẩn bị sách để người thuê lấy sách Đã nhận được sách Mua sách						
						Tổng tiền thuê: 84.000đ Tổng tiền cọc: 90.000đ Tổng tiền: 90.000đ
Id đơn hàng	Người thuê	Thời gian thuê	Trạng thái	Xem chi tiết		
#5	Romaguera Nigel	05/10 - 05/10/2024	Chủ sách đã đưa hàng	Xem chi tiết		
Id bài đã...	Tên sách	Thời gian thuê	Tiền cọc	Tiền thuê	Tổng cộng	
4	Foundling	1 ngày	383.700	252.000	383.700	
✓ Đã giao sách cho người thuê Đã Trả Sách Mua sách						
						Tổng tiền thuê: 252.000đ Tổng tiền cọc: 383.700đ Tổng tiền: 383.700đ

Hình 55: Quản lý đơn hàng của khách

Chủ sách có thể xác nhận nhận lại sách nếu sách ở trạng thái chờ chủ nhận lại sách:

Id đơn hàng #12	Người thuê Developer Dev	Thời gian thuê 16/05 - 16/05/2024	Trạng thái chủ sách Chủ sách chờ nhận sách	Xem chi tiết	
Id bài đăng	Tên sách	Thời gian thuê	Tiền cọc	Tiền thuê	Tổng cột
22	Into the Darkness	0 ngày	79,200	1,300	80,500
<input checked="" type="checkbox"/> Bạn đã được nhận lại sách chưa?			ĐÃ NHẬN LẠI SÁCH		Tổng tiền thuê: 1.300đ Tổng tiền cọc: 79.200đ Tổng tiền: 79.200đ

Hình 56: Chủ sách xác nhận nhận lại sách

Giống như mô tả ở giao diện của khách đi thuê.

6.3.13 Trang quản lý đơn mua

Id đơn hàng #1	Người bán Developer Dev	Trạng thái Người bán Chờ khách thanh toán	Xem chi tiết
37	To Die for	16/11/2024 336,400 235,480 235,480d	Vui lòng thanh toán đơn hàng trong 24 giờ, nếu chuyển khoản thành công, bạn hãy nhấn nút Đã trả tiền Đã Trả Tiền Giá bán: 235,480đ Tổng tiền: 235,480đ
Id đơn hàng #2	Người bán Developer Dev	Trạng thái Người bán Khách đã hủy	Xem chi tiết
13	Vanishing Point	12/09/2024 251,000 175,700 175,700d	Khách đã hủy Giá bán: 175,700đ Tổng tiền: 175,700đ
Id đơn hàng #3	Người bán Developer Dev	Trạng thái Người bán Chờ khách lấy hàng	Xem chi tiết
13	Vanishing Point	15/11/2024 251,000 175,700 175,700d	Vui lòng chỉ nhận 'đã nhận được sách' khi đơn hàng đã được giao đến bạn và bạn đã kiểm tra xong hàng Đã Nhận Được Sách Giá bán: 175,700đ Tổng tiền: 175,700đ
Id đơn hàng #4	Người bán Developer Dev	Trạng thái Người bán Chủ sách đã đưa hàng	Xem chi tiết
14	Hide and Seek	12/11/2024 230,900 161,630 161,630d	Đã giao sách cho người mua, chờ admin trả người bán tiền bán sách Giá bán: 161,630đ Tổng tiền: 161,630đ

Nhập mail để nhận được những quà tặng đặc biệt nhé !

hoaminzy

The Flying Bookstore

Liên hệ:

- 17/11 Trường Phước Phan, Bình Hưng Hòa, Bình Tân, TP.HCM
- +84 123 345 123
- support@flyingbookstore

Chính sách:

- Chính sách bảo mật
- Hướng dẫn về tình trạng sách

Hình 57: Trang quản lý đơn mua

Giống với giao diện của quản lý đơn thuê, nhưng khác biệt ở các trạng thái như sau:

- Khi người mua mới đặt hàng

Giao diện ở người mua:

Id đơn hàng #11	Người bán Developer Dev	Trạng thái Người bán Chờ khách thanh toán	Xem chi tiết
151	Daughter of the ...	05/12/2024 193,000 135,100 135,100d	Vui lòng thanh toán đơn hàng trong 24 giờ Giá bán: 135,100đ Tổng tiền: 135,100đ

Hình 58: Giao diện ở người mua khi mới đặt hàng

Giao diện người bán:

Id đơn hàng #11	Người mua Nguyễn Lê Xuân Hòa	Trạng thái Chờ thanh toán	Xem chi tiết ↗			
Id bài đã...	Tên sách	Thời gian đặt	Giá gốc	Giá bán	Tổng cộng	
151	Daughter of the ...	05/12/2024	193.000	135.100	135.100	ⓘ
<input checked="" type="checkbox"/> Người mua mới đặt hàng, chờ người mua thanh toán						
						Giá bán: 135.100đ Tổng tiền: 135.100đ

Hình 59: Giao diện của người bán khi mới đặt hàng

- Khi người mua đã thanh toán:

Giao diện ở người mua:

Id đơn hàng #11	Người bán Developer Dev	Trạng thái Chờ khách lấy hàng	Xem chi tiết ↗			
Id bài đã...	Tên sách	Thời gian đặt	Giá gốc	Giá bán	Tổng cộng	
151	Daughter of the ...	05/12/2024	193.000	135.100	135.100	ⓘ
<input checked="" type="checkbox"/> Vui lòng chỉ nhấn "đã nhận được sách" khi đơn hàng đã được giao đến bạn và bạn đã kiểm tra xong hàng						Đã Nhận Được Sách
						Giá bán: 135.100đ Tổng tiền: 135.100đ

Hình 60: Giao diện của người mua khi người mua đã thanh toán

Giao diện ở người bán:

Id đơn hàng #11	Người mua Nguyễn Lê Xuân Hòa	Trạng thái Lấy hàng	Xem chi tiết ↗			
Id bài đã...	Tên sách	Thời gian đặt	Giá gốc	Giá bán	Tổng cộng	
151	Daughter of the ...	05/12/2024	193.000	135.100	135.100	ⓘ
<input checked="" type="checkbox"/> Người mua đã thanh toán thành công, người bán cần chuẩn bị sách để người mua lấy sách						Giá bán: 135.100đ Tổng tiền: 135.100đ

Hình 61: Giao diện của người bán khi người mua đã thanh toán

- Khi người mua nhận được sách và đã duyệt xong sách:

Giao diện người mua:

Id đơn hàng #11	Người bán Developer Dev	Trạng thái Chủ sách đã đưa hàng	Xem chi tiết ↗			
Id bài đã...	Tên sách	Thời gian đặt	Giá gốc	Giá bán	Tổng cộng	
151	Daughter of the ...	05/12/2024	193.000	135.100	135.100	ⓘ
<input checked="" type="checkbox"/> Đã giao sách cho người mua, chờ admin trả người bán tiền bán sách						Giá bán: 135.100đ Tổng tiền: 135.100đ

Hình 62: Giao diện người mua khi người mua nhận được sách

Giao diện người bán:

Id đơn hàng #11	Người mua Nguyễn Lê Xuân Hòa	Trạng thái Đã lấy hàng	Xem chi tiết		
Id bài đã...	Tên sách	Thời gian đặt	Giá gốc	Giá bán	Tổng cộng
151	Daughter of the ...	05/12/2024	193.000	135.100	135.100 ⓘ
✓ Đã giao sách cho người mua, chờ admin trả người bán tiền bán sách				Giá bán: 135.100đ Tổng tiền: 135.100đ	

Hình 63: Giao diện người bán khi người mua nhận được sách

- Khi Quản trị viên đã trả tiền cho người bán sách:

Giao diện người mua:

Id đơn hàng #11	Người bán Developer Dev	Trạng thái Đã trả tiền cho người bán	Xem chi tiết		
Id bài đã...	Tên sách	Thời gian đặt	Giá gốc	Giá bán	Tổng cộng
151	Daughter of the ...	05/12/2024	193.000	135.100	135.100 ⓘ
✓ Đơn hàng đã hoàn thành, Admin đã trả tiền cho người bán				Giá bán: 135.100đ Tổng tiền: 135.100đ	

Hình 64: Giao diện người mua khi quản trị viên đã trả tiền cho người bán sách

Giao diện người bán:

Id đơn hàng #11	Người mua Nguyễn Lê Xuân Hòa	Trạng thái Đã trả tiền cho người bán	Xem chi tiết		
Id bài đã...	Tên sách	Thời gian đặt	Giá gốc	Giá bán	Tổng cộng
151	Daughter of the ...	05/12/2024	193.000	135.100	135.100 ⓘ
✓ Đơn hàng đã hoàn thành, Admin đã trả tiền cho người bán				Giá bán: 135.100đ Tổng tiền: 135.100đ	

Hình 65: Giao diện người bán khi quản trị viên đã trả tiền cho người bán sách

6.3.14 Trang quản lý đơn bán

Nhập mail để nhận được những quà tặng đặc biệt nhé !



The Flying Bookstore là nơi có thể giúp bạn học sinh/sinh viên mua thuê, mượn tài liệu của mình. Đồng thời giúp quản lý các tài liệu mua, mượn.



Liên hệ

④ 17/11 Trường Phước Phan, Bình Hưng Hòa, Bình Tân, TP.HCM
⑤ +123 345 123
✉ support@flying bookstore

Chính sách

Chính sách bảo mật
Hướng dẫn về tình trạng sách



Hình 66: Trang quản lý đơn bán

Tương tự như trang quản lý đơn mua và đã mô tả các trạng thái ở trang quản lý đơn mua.

6.3.15 Trang chi tiết đơn hàng

Hình 67: Trang chi tiết đơn hàng

6.3.15.1 Tổng quan

- Mục đích:** Người dùng có thể xem thông tin chi tiết về đơn hàng của mình, bao gồm thông tin sản phẩm, thời gian thuê, giá cả, và trạng thái đơn hàng để theo dõi một cách rõ ràng.
- Đối tượng tham gia:** Người dùng.

- Điều kiện tiên quyết:** Người dùng đã đăng nhập vào hệ thống và có ít nhất một đơn hàng được tạo.
- Kết quả mong muốn:** Người dùng có thể xem được đầy đủ chi tiết về đơn hàng của mình qua giao diện hiển thị trên trang.

6.3.15.2 Quy tắc và Ràng buộc

- Đơn hàng phải thuộc về người dùng hiện tại.
- Thông tin hiển thị phải chính xác và cập nhật trạng thái.
- Dữ liệu bao gồm: mã đơn hàng, ngày đặt, thông tin người cho thuê, thời gian thuê, địa chỉ, số tiền thuê, tiền cọc và tổng tiền.

6.3.15.3 Mô tả chi tiết

STT	Tên	Mô tả	Kiểm soát	Loại dữ liệu	Giá trị mặc định	Bắt buộc	Quy tắc
1	Mã đơn hàng	Hiển thị mã đơn hàng	Textview	Chuỗi	N/A	Có	Hiển thị mã đơn hàng duy nhất của người dùng
2	Ngày đặt thuê	Ngày mà đơn hàng được đặt	Textview	Ngày tháng	N/A	Có	Hiển thị theo định dạng dd/mm/yyyy
3	Người cho thuê	Tên người/đơn vị cung cấp sản phẩm thuê	Textview	Chuỗi	N/A	Có	Phải là thông tin đã đăng ký trước
4	Thời gian thuê	Thời gian thuê từ ngày nào đến ngày nào	Textview	Ngày tháng	N/A	Có	Hiển thị định dạng từ ngày dd/mm/yyyy đến dd/mm/yyyy
5	Địa chỉ người cho thuê	Địa chỉ chi tiết nơi cung cấp sản phẩm	Textview	Chuỗi	N/A	Có	Dữ liệu từ hệ thống đã lưu trữ
6	Số điện thoại người cho thuê	Hiển thị số điện thoại của người cho thuê	Textview	Chuỗi	N/A	Có	Đảm bảo tính chính xác và cập nhật
7	Số ngày thuê	Tổng số ngày thuê	Textview	Số nguyên	N/A	Có	Tính toán tự động dựa trên thời gian thuê
8	Tiền thuê	Số tiền thuê cần thanh toán	Textview	Số thực	0	Có	Hiển thị theo đơn vị VNĐ
9	Tiền cọc	Số tiền cọc đã thanh toán	Textview	Số thực	0	Có	Hiển thị theo đơn vị VNĐ
10	Tổng cộng	Tổng chi phí của đơn hàng	Textview	Số thực	0	Có	Tính toán tự động: Tiền thuê + Tiền cọc

Bảng 67: Mô tả chi tiết Trang chi tiết đơn hàng

6.3.15.4 Quy trình xử lý

- Người dùng truy cập vào mục **Đơn thuê**.
- Chọn đơn hàng muốn xem chi tiết.
- Hệ thống sẽ hiển thị đầy đủ thông tin chi tiết đơn hàng như trên.
- Người dùng có thể đổi chiều thông tin hoặc liên hệ người cho thuê qua số điện thoại hiển thị.

6.3.15.5 ĐẶC TẢ API

API: Lấy chi tiết đơn thuê

- **Endpoint:** /api/leaseOrder/{leaseOrderId}
- Phương thức: GET
- **Mô tả:** Truy xuất thông tin chi tiết của một đơn thuê cụ thể dựa trên ID.
- Xác thực:
 - Không yêu cầu body trong request.
 - Yêu cầu Bearer Token trong phần **Authorization Header**.
- Tham số đầu vào:

Tên tham số	Loại dữ liệu	Bắt buộc	Mô tả
leaseOrderId	number	Có	ID của đơn thuê cần truy xuất

Bảng 68: Tham số đầu vào API Lấy chi tiết đơn thuê

API: Lấy danh sách đơn thuê (Admin)

- **Endpoint:** /api/leaseOrder/admin
- Phương thức: GET
- **Mô tả:** Truy xuất danh sách các đơn thuê để phục vụ mục đích quản lý, kèm theo thông tin phân trang và chi tiết từng đơn thuê.
- Xác thực:
 - Không yêu cầu body trong request.
 - Yêu cầu Bearer Token trong phần **Authorization Header**.
- **Tham số đầu vào:** Không yêu cầu tham số đầu vào.

6.3.16 Trang quản lý bài đăng

The Flying bookstore

Chào Tracy

Quản lý bài đăng của tôi

Thêm bài đăng

Bài đăng bán và thuê

Bài đăng chỉ thuê

Bài đăng chỉ bán

Tìm kiếm...

ID	Tên bài đăng	Tác giả	Giá thuê	Tiền cọc	Tiền phạt	Giá bán
1	Walls	Anne Ed...	300đ/ngày	224.000đ	2.400đ/ngày	156.870đ
3	Deep Waters	Barbara N...	1.400đ/ngày	46.000đ	2.000đ/ngày	32.200đ
4	Foundling	June Old...	2.600đ/ngày	383.700đ	4.200đ/ngày	268.590đ
5	Ice	Ed McBain	1.600đ/ngày	376.600đ	4.100đ/ngày	263.620đ
6	Daring Moves	Linda Lae...	1.200đ/ngày	105.700đ	1.500đ/ngày	73.990đ

1-5 trong 8 < >

Hình 68: Trang quản lý bài đăng

Trang quản lý bài đăng có danh sách bài đăng của chủ sách và nút thêm bài đăng từ đây có thể đến được trang tạo bài đăng. Muốn tìm kiếm bài đăng bạn chỉ cần nhập từ khóa vào ô tìm kiếm thì danh sách bài đăng sẽ lọc theo từ khoá tìm kiếm.

6.3.16.1 Tổng quan

- Mục đích:** Người dùng có thể xem, chỉnh sửa, hoặc xóa bài đăng của mình để quản lý danh sách các sản phẩm cho thuê hoặc bán một cách dễ dàng.
- Đối tượng tham gia:** Người dùng.
- Điều kiện tiên quyết:** Người dùng đã đăng nhập vào hệ thống và có ít nhất một bài đăng được tạo.
- Kết quả mong muốn:** Người dùng có thể thực hiện các thao tác quản lý như xem, chỉnh sửa hoặc xóa bài đăng từ giao diện hiển thị.

6.3.16.2 Quy tắc và Ràng buộc

- Bài đăng phải thuộc về người dùng hiện tại.
- Thông tin bài đăng cần hiển thị đầy đủ: tên bài đăng, tác giả, giá thuê, tiền cọc, tiền phạt, và giá bán.
- Chỉ những bài đăng hợp lệ mới có thể chỉnh sửa hoặc xóa.

6.3.16.3 Mô tả chi tiết

STT	Tên	Mô tả	Kiểm soát	Loại dữ liệu	Giá trị mặc định	Bắt buộc	Quy tắc
1	Id bài đăng	Hiển thị Id duy nhất của bài đăng	Textview	Số nguyên	N/A	Có	Hiển thị giá trị duy nhất
2	Tên bài đăng	Tên của sản phẩm cho thuê/bán	Textview	Chuỗi	N/A	Có	Hiển thị đầy đủ tên bài đăng
3	Tác giả	Tên tác giả của sản phẩm	Textview	Chuỗi	N/A	Có	Phải khớp thông tin tác giả
4	Giá thuê	Giá cho thuê của sản phẩm	Textview	Chuỗi	0đ/ngày	Có	Hiển thị định dạng số tiền/ngày
5	Tiền cọc	Số tiền cọc cần thanh toán	Textview	Số thực	0đ	Có	Hiển thị theo đơn vị VNĐ
6	Tiền phạt	Số tiền phạt khi không trả đúng hạn	Textview	Chuỗi	0đ/ngày	Có	Hiển thị định dạng số tiền/ngày
7	Giá bán	Giá bán của sản phẩm (nếu có)	Textview	Chuỗi	0đ	Không	N/A nếu không áp dụng bán
8	Xem bài đăng	Mở giao diện xem chi tiết bài đăng	Button	Nút	N/A	Có	Hiển thị thông tin đầy đủ của bài đăng
9	Chỉnh sửa bài đăng	Mở giao diện chỉnh sửa thông tin bài đăng	Button	Nút	N/A	Có	Phải có quyền chỉnh sửa
10	Xóa bài đăng	Xóa bài đăng khỏi danh sách	Button	Nút	N/A	Có	Chỉ áp dụng nếu bài đăng không có đơn hàng liên quan

Bảng 69: Mô tả chi tiết Trang quản lý bài đăng

6.3.16.4 Quy trình xử lý

1. Người dùng truy cập vào mục **Quản lý bài đăng**.
2. Danh sách các bài đăng sẽ hiển thị bao gồm thông tin cơ bản (Id, tên, tác giả, giá thuê, tiền cọc, tiền phạt, giá bán).
3. Người dùng có thể:
 - a. Nhấn vào biểu tượng **Xem chi tiết** để xem đầy đủ thông tin bài đăng.
 - b. Nhấn vào biểu tượng **Chỉnh sửa** để cập nhật thông tin bài đăng.
 - c. Nhấn vào biểu tượng **Xóa** để xóa bài đăng (nếu không có đơn hàng liên quan).

6.3.16.5 Đặc tả API

API: Lấy danh sách bài đăng

- Phương thức: GET
- **Endpoint:** /api/listing
- Tham số đầu vào: Không có

API: Xóa danh sách

- **Endpoint:** /api/listing/{listingId}
- Phương thức: DELETE
- **Mô tả:** Dùng để xóa một danh sách cụ thể dựa trên ID.
- Tham số đầu vào:

Tên tham số	Loại dữ liệu	Mô tả	Bắt buộc
listingId	number	ID của danh sách cần xóa	Có

Bảng 70: Tham số đầu vào API Xóa danh sách

6.3.17 Trang tạo bài đăng

Hình 69: Trang tạo bài đăng

Tạo bài đăng cần cung cấp sách: sách ở đây bạn có thể chọn sách hoặc tạo sách. Nếu chọn sách thì tìm sách ở mục tìm sách. Thông tin sách sẽ được load lên các input, nếu muốn tạo sách thì điền thông tin tạo sách và bấm tạo sách. Sau đó cung cấp thông tin tài liệu bấm tài liệu để tạo tài liệu. Cuối cùng cung cấp thông tin bài đăng để tạo bài đăng.

Hình 70: Chọn sách có sẵn

6.3.17.1 Tổng quan

- Mục đích:** Người dùng có thể tạo bài đăng mới để giới thiệu sách của mình đến người thuê hoặc người mua tiềm năng.
- Đối tượng tham gia:** Người dùng.
- Điều kiện tiên quyết:** Người dùng đã đăng nhập vào hệ thống.
- Kết quả mong muốn:** Một bài đăng mới được tạo và hiển thị trên danh sách các bài đăng.

6.3.17.2 Quy tắc và Ràng buộc

- Người dùng cần nhập đầy đủ thông tin bắt buộc của sách.
- Mỗi bài đăng chỉ được gắn với một sách cụ thể.
- Chỉ được đăng bài với các mục đích cụ thể: cho thuê, bán, hoặc cả hai.
- Giá thuê và tiền cọc phải được nhập dưới dạng số hợp lệ (≥ 0).

6.3.17.3 Mô tả chi tiết

STT	Tên	Mô tả	Kiểm soát	Loại dữ liệu	Giá trị mặc định	Bắt buộc	Quy tắc
1	Tên sách	Tên của cuốn sách cần cho thuê hoặc bán	Textfield	Chuỗi	N/A	Có	Phải nhập đầy đủ
2	Tác giả	Tên tác giả của sách	Textfield	Chuỗi	N/A	Có	Phải nhập đầy đủ
3	Nhà xuất bản	Nhà xuất bản của sách	Textfield	Chuỗi	N/A	Không	-
4	Số trang	Tổng số trang của sách	Numberfield	Số nguyên	0	Không	≥ 0
5	Kích thước	Kích thước vật lý của sách (cm)	Textfield	Chuỗi	N/A	Không	-
6	Hình ảnh sách	Hình ảnh đại diện cho bài đăng	File upload	File ảnh	N/A	Có	Chỉ chấp nhận định dạng .jpg, .png
7	Dịch vụ	Lựa chọn loại dịch vụ: thuê, bán, hoặc cả hai	Radio button	Enum	N/A	Có	Ít nhất một lựa chọn
8	Giá thuê/ngày	Giá thuê sách mỗi ngày	Numberfield	Số thực	0	Không	≥ 0
9	Tiền cọc	Số tiền cọc cho sách	Numberfield	Số thực	0	Có	≥ 0
10	Phí phạt trả trễ	Phí phạt khi trả sách trễ hạn	Numberfield	Số thực	0	Không	≥ 0
11	Địa chỉ	Địa chỉ nơi cho thuê sách	Textfield	Chuỗi	N/A	Có	Phải nhập đầy đủ
12	Tạo bài đăng	Lưu bài đăng vào hệ thống	Button	Nút	N/A	Có	Lưu thành công nếu đủ thông tin

Bảng 71: Mô tả chi tiết Trang tạo bài đăng

6.3.17.4 Quy trình xử lý

- Người dùng truy cập vào mục Tạo bài đăng.
- Nhập thông tin bắt buộc bao gồm:
 - Tên sách, tác giả, hình ảnh đại diện.
 - Lựa chọn dịch vụ (thuê, bán hoặc cả hai).
 - Thông tin giá thuê, tiền cọc, và địa chỉ (nếu áp dụng).
- Người dùng nhấn Tạo bài đăng để lưu bài đăng.
- Hệ thống hiển thị thông báo thành công và chuyển hướng đến danh sách bài đăng của người dùng.

6.3.17.5 Đặc tả API

API: Tạo danh sách mới

- **Endpoint:** /api/listing
- Phương thức: POST
- **Mô tả:** Dùng để tạo một danh sách mới.
- Tham số đầu vào:

Tên tham số	Loại dữ liệu	Mô tả	Bắt buộc
copyId	number	ID của bản sao (copy)	Có
ownerId	number	ID của chủ sở hữu	Có
quantity	number	Số lượng	Có
address	string	Địa chỉ của danh sách	Có
expiryDate	string	Ngày hết hạn của danh sách	Có
leaseRate	number	Giá cho thuê	Có
depositFee	number	Phí đặt cọc	Có
penaltyRate	number	Tỷ lệ phạt	Có
price	number	Giá bán	Có
description	string	Mô tả danh sách	Có
listingStatus	string	Trạng thái của danh sách	Có
allow_rent	number	Cờ cho phép thuê (0 hoặc 1)	Có
allow_purchase	number	Cờ cho phép mua (0 hoặc 1)	Có

Bảng 72: Tham số đầu vào API Tạo danh sách mới

- **Xác thực:** Yêu cầu Bearer Token trong phần **Authorization Header**.

6.3.18 Trang quản lý voucher của tôi

ID	Tên	Mã	Thời gian hiệu lực	Giá trị tối thiểu	Loại voucher	Số tiền giảm	Phần trăm giảm
1	Shop Anniversary	ANNIV20	1/1/2024 - 30/12/2024	15.000đ	Giảm theo số tiền	20.000đ	
2	Flash Sale	FLASH5	1/4/2024 - 7/4/2024	75.000đ	Giảm theo phần trăm	5%	
3	VIP Member Dis...	VIP50	1/5/2024 - 31/5/2024	300.000đ	Giảm theo số tiền	50.000đ	
4	Golden Weekend	GOLDEN8	1/7/2024 - 3/7/2024	100.000đ	Giảm theo phần trăm	8%	
5	Festival Fever	FEST7	1/9/2024 - 10/9/2024	50.000đ	Giảm theo số tiền	7.000đ	

Hình 71: Trang quản lý voucher của tôi

6.3.18.1 Tổng quan

- Mục đích:** Cho phép người bán quản lý danh sách voucher mà họ đã tạo, bao gồm việc xem thông tin chi tiết và xóa voucher không còn cần thiết.
- Đối tượng tham gia:** Người bán đã đăng nhập vào hệ thống.
- Điều kiện tiên quyết:**
 - Người bán đã đăng nhập vào tài khoản của mình.
 - Có quyền truy cập vào trang quản lý voucher.
- Kết quả mong muốn:**
 - Người bán có thể xem toàn bộ danh sách voucher của họ.
 - Người bán có thể xóa voucher cụ thể, và danh sách được cập nhật ngay lập tức.

6.3.18.2 Quy tắc và Ràng buộc

- Người bán chỉ có thể xem và xóa các voucher do chính họ tạo ra.
- Hệ thống không cho phép người bán xóa các voucher đang được áp dụng hoặc đã hết hạn.
- Việc xóa voucher là hành động không thể hoàn tác.

6.3.18.3 Mô tả chi tiết

- Xem Voucher**
 - Người bán truy cập trang Quản lý voucher của tôi.
 - Giao diện hiển thị danh sách voucher với thông tin sau:

STT	Tên	Mô tả	Loại điều khiển	Kiểu dữ liệu	Giá trị mặc định	Bắt buộc	Quy tắc
1	ID Voucher	Mã định danh duy nhất của voucher.	Nhãn	Số nguyên	Tự động tạo	X	Không thể chỉnh sửa.
2	Tên Voucher	Tên của voucher.	Nhãn	nvarchar(255)		X	Được hiển thị rõ ràng cho

							Admin để nhận biết loại voucher.
3	Mã Code	Mã code của voucher để áp dụng khuyến mãi.	Nhãn	nvarchar(20)		X	Hiển thị định dạng đúng (Ví dụ: SPRING5, LNY2024).
4	Loại Voucher	Loại khuyến mãi (giảm theo tiền hoặc phần trăm).	Nhãn	nvarchar(50)		X	Không thể chỉnh sửa.
5	Giá trị tối thiểu	Giá trị tối thiểu của đơn hàng để áp dụng voucher.	Nhãn	Số tiền			Hiển thị đúng đơn vị tiền tệ của hệ thống (VD: VNĐ).
6	Giá trị khuyến mãi	Số tiền hoặc tỷ lệ phần trăm được giảm khi sử dụng voucher.	Nhãn	Số tiền/tỷ lệ			Hiển thị rõ ràng giá trị khuyến mãi theo đúng loại voucher.
7	Thời gian hiệu lực	Khoảng thời gian voucher có hiệu lực (từ ngày - đến ngày).	Nhãn	Ngày		X	Hiển thị định dạng ngày tháng chính xác.
8	Xóa Voucher	Nút "Xóa"	Nút bấm				Khi được nhấn, voucher sẽ bị xóa sau khi Admin xác nhận hành động.

Bảng 73: Mô tả chi tiết Trang quản lý voucher của tôi

- Xóa Voucher

1. Người bán nhấn vào biểu tượng thùng rác (Xóa) ở dòng tương ứng với voucher cần xóa.
2. Hiển thị hộp thoại xác nhận với nội dung: "Bạn có chắc chắn muốn xóa voucher này? Hành động này không thể hoàn tác."
3. Nếu chọn "Xác nhận", hệ thống:
 - a. Gửi yêu cầu xóa voucher đến server.
 - b. Hiển thị thông báo kết quả (Thành công/Thất bại).
4. Nếu chọn "Hủy", quay lại trang danh sách voucher mà không có thay đổi gì.

6.3.18.4 Quy trình xử lý

1. Xem Voucher

1. Người bán đăng nhập và truy cập trang Quản lý voucher của tôi.
2. Danh sách voucher của người bán được tải và hiển thị đầy đủ.
3. Người bán có thể:
 - a. Tìm kiếm voucher theo tên hoặc mã voucher.
 - b. Lọc danh sách theo loại voucher hoặc trạng thái hiệu lực (Còn hiệu lực, Đã hết hạn).

2. Xóa Voucher

1. Người bán nhấn vào biểu tượng thùng rác của voucher cần xóa.
2. Hệ thống hiển thị hộp thoại xác nhận.
3. Nếu người bán đồng ý:
 - a. Hệ thống gửi yêu cầu xóa đến server.
 - b. Cập nhật lại danh sách voucher (ẩn voucher vừa xóa).
4. Nếu không đồng ý, hộp thoại xác nhận sẽ đóng mà không thay đổi dữ liệu.

6.3.18.5 Đặc tả API

API: Lấy danh sách voucher theo người bán

- Phương thức: GET
- Endpoint: /api/voucher-shop
- Tham số đầu vào:

Tên tham số	Loại dữ liệu	Mô tả	Bắt buộc
sellerId	Số nguyên	ID người bán	Có

Bảng 74: Tham số đầu vào API Lấy danh sách voucher theo người bán

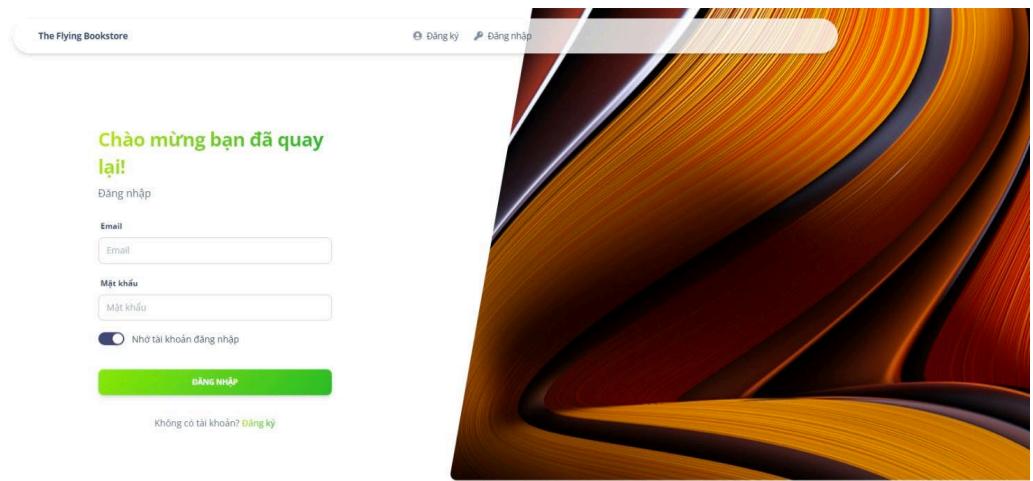
API: Xóa voucher theo người bán

- Phương thức: DELETE
- Endpoint: /api/voucher-shop/{voucherId}
- Tham số đầu vào:

Tên tham số	Loại dữ liệu	Mô tả	Bắt buộc
voucherId	Số nguyên	ID voucher cần xóa	Có

Bảng 75: Tham số đầu vào API Xóa voucher theo người bán

6.3.19 Trang đăng nhập admin



Hình 72: Trang đăng nhập admin

6.3.19.1 Tổng quan

- Mục đích:** Admin có thể đăng nhập vào hệ thống để quản lý các chức năng và thao tác cần thiết như quản lý người dùng, sách, đơn hàng, v.v.
- Tác nhân tham gia: Admin.
- Điều kiện tiên quyết:** Admin đã có tài khoản trong hệ thống.
- Kết quả mong muốn:** Admin đăng nhập thành công và được chuyển hướng tới trang quản lý.

6.3.19.2 Quy tắc và Ràng buộc

- Ràng buộc về mật khẩu: Mật khẩu phải có ít nhất 8 ký tự, bao gồm cả chữ hoa, chữ thường và số.

6.3.19.3 Mô tả chi tiết

STT	Tên	Mô tả	Loại điều khiển	Kiểu dữ liệu	Giá trị mặc định	Bắt buộc	Quy tắc
1	Tên tài khoản	Tên tài khoản admin để đăng nhập	Textbox	varchar(255)		X	Người dùng nhập tài khoản đã đăng ký.
2	Mật khẩu	Mật khẩu để truy cập hệ thống	Password textbox	varchar(255)		X	Người dùng nhập mật khẩu hợp lệ.
3	Nhớ tài khoản đăng nhập	Cho phép hệ thống ghi nhớ tài khoản đăng nhập lần sau	Toggle button	Boolean	Tắt		Người dùng có thể bật/tắt tính năng này.

4	Đăng nhập	Nút để xác nhận thông tin đăng nhập	Button			X	Khi nhấn nút, hệ thống sẽ kiểm tra thông tin và chuyển hướng tới dashboard nếu thành công.
5	Đăng ký	Liên kết tới trang đăng ký tài khoản mới	Hyperlink				Hiển thị link để người dùng có thể đăng ký tài khoản nếu chưa có.
6	Quên mật khẩu	Liên kết tới trang khôi phục mật khẩu	Hyperlink				Hiển thị link để người dùng khôi phục mật khẩu nếu quên.

Bảng 76: Mô tả chi tiết Trang đăng nhập admin

6.3.19.4 Quy trình xử lý

1. Admin truy cập vào trang đăng nhập.
2. Nhập thông tin tài khoản và mật khẩu vào các trường tương ứng.
3. (Tùy chọn) Bật công tắc "Nhớ tài khoản đăng nhập" nếu muốn hệ thống ghi nhớ tài khoản cho lần sau.
4. Nhấn nút "Đăng nhập".
5. Hệ thống kiểm tra thông tin:
 - o Nếu đúng, admin sẽ được chuyển tới trang quản lý.
 - o Nếu sai, hiển thị thông báo lỗi và yêu cầu nhập lại.
6. Admin có thể sử dụng các liên kết để đăng ký tài khoản mới hoặc khôi phục mật khẩu nếu cần thiết.

6.3.19.5 Đặc tả API

API: Đăng nhập admin

- Phương thức: POST
- Endpoint: /api/login
- Tham số đầu vào:

Tên tham số	Loại dữ liệu	Mô tả	Bắt buộc
username	varchar(255)	Tên đăng nhập	Có
password	varchar(255)	Mật khẩu	Có

Bảng 77: Tham số đầu vào API Đăng nhập admin

- **Mô tả:** Gửi yêu cầu xác thực tài khoản khi người dùng bấm nút "Đăng nhập". Hệ thống sẽ kiểm tra tính hợp lệ của thông tin tài khoản và phản hồi kết quả.

6.3.20 Trang admin quản lý đơn hàng thuê

The screenshot shows a web-based administrative interface for managing rental orders. At the top left is the logo 'The Flying bookstore' and at the top right are links for 'Tài khoản' and user icons. The main content area has a header 'Quản lý đơn hàng' and a sub-header 'Danh sách đơn hàng'. Below this is a table with columns: ID, TIÊU ĐỀ, CHỦ SÁCH, NGƯỜI THUÊ, KHOẢNG THỜI GIAN, PHƯƠNG THỨC, TIỀN CỌC, TIỀN THUÊ, TỔNG TIỀN, and TRẠNG THÁI. There are three rows of data:

ID	Tiêu đề	Chủ sách	Người thuê	Khoảng thời gian	Phương thức	Tiền cọc	Tiền thuê	Tổng tiền	Trạng thái
3	Deep Waters	Dev Developer	Tracy Braun	01/03 - 01/04/2024	Chuyển khoản	46.000đ	84.000đ	46.000đ	9. Người thuê đã hủy ✓
4	Deep Waters	Dev Developer	Tracy Braun	01/05 - 01/06/2024	Chuyển khoản	46.000đ	84.000đ	46.000đ	3. Đã thanh toán ✓
2	From Potter's Field	Dev Developer	Tracy Braun	01/05 - 01/06/2024	Chuyển khoản	414.800đ	9.000đ	414.800đ	9. Người thuê đã hủy ✓

At the bottom left is a copyright notice: © 2024. Bản quyền thuộc về đội ngũ The Flying Bookstore. On the right side of the table are navigation arrows and a page number '1'.

Hình 73: Trang admin quản lý đơn thuê

Chưa danh sách đơn hàng và trạng thái đơn hàng, ở đây bạn có thể duyệt đơn hàng bằng cách thay đổi trạng thái của đơn hàng.

6.3.20.1 Tổng quan

- Mục đích:** Quản trị viên có thể xem và quản lý tất cả các đơn thuê sách, bao gồm thông tin người thuê, khoảng thời gian thuê, phương thức thanh toán, và trạng thái từng đơn hàng.
- Tác nhân tham gia:** Quản trị viên hoặc nhân viên quản lý của hệ thống thuê sách.
- Điều kiện tiên quyết:** Có các đơn thuê sách được tạo bởi người dùng và lưu trữ trong hệ thống.
- Kết quả mong muốn:** Quản trị viên dễ dàng theo dõi chi tiết từng đơn thuê và cập nhật trạng thái đơn thuê nếu cần thiết.

6.3.20.2 Quy tắc và Ràng buộc

- Thông tin hiển thị:** Các trường dữ liệu bao gồm ID đơn hàng, tiêu đề sách, tên người thuê, khoảng thời gian thuê, phương thức thanh toán, tiền cọc, tiền thuê, tiền trả cho khách, và trạng thái đơn hàng.
- Trạng thái đơn hàng:** Đơn hàng có thể có các trạng thái như "Chờ người thuê thanh toán", "Đã trả cọc cho người thuê", "Người thuê đã trả", và các trạng thái khác liên quan đến quá trình giao nhận sách.
- Các nút hành động:** Tùy thuộc vào trạng thái hiện tại của đơn hàng, nút hành động như "Người thuê đã trả tiền" hoặc các thông báo về trạng thái sẽ xuất hiện để hỗ trợ quản trị viên.

6.3.20.3 Mô tả chi tiết

STT	Tên	Mô tả	Loại điều khiển	Kiểu dữ liệu	Giá trị mặc định	Bắt buộc	Quy tắc

1	ID đơn hàng	Mã định danh của đơn hàng thuê	Label	int	Tạo tự động	X	Được hệ thống tạo khi đơn hàng được khởi tạo.
2	Tiêu đề	Tiêu đề của sách đã thuê	Label	varchar(255)		X	Lấy từ cơ sở dữ liệu của đơn hàng.
3	Chủ sách	Người sở hữu sách	Label	varchar(255)		X	Được lấy từ dữ liệu người bán hoặc người sở hữu sách.
4	Người thuê	Tên người thuê sách	Label	varchar(255)		X	Lấy từ dữ liệu của đơn hàng.
5	Khoảng thời gian	Khoảng thời gian thuê sách	Label	Date range		X	Hiển thị khoảng thời gian thuê sách.
6	Phương thức thanh toán	Hình thức thanh toán mà người thuê sử dụng	Label	Enum (Chuyển khoản, Tiền mặt)		X	Cập nhật từ đơn hàng.
7	Tiền cọc	Số tiền cọc cho đơn thuê	Label	Decimal		X	Lấy từ dữ liệu đơn hàng.
8	Tiền thuê	Số tiền thuê sách	Label	Decimal		X	Lấy từ dữ liệu đơn hàng.
9	Tiền trả cho khách	Số tiền hoàn trả cho người thuê	Label	Decimal		X	Tính toán dựa trên các điều kiện hoàn trả.
10	Trạng thái đơn hàng	Trạng thái hiện tại của đơn hàng	Label	Enum	“Chờ thanh toán”	X	Cập nhật khi có thay đổi trạng thái đơn hàng.
11	Chuyển trạng thái	Xác nhận chuyển trạng thái tiếp theo	Button				Hiển thị khi đơn hàng đang ở trạng thái hiện tại.

Bảng 78: Mô tả chi tiết Trang quản lý đơn thuê cho thuê admin

6.3.20.4 Quy trình xử lý

1. Quản trị viên truy cập vào trang “Quản lý đơn thuê” từ menu.
2. Hệ thống hiển thị danh sách các đơn thuê sách, bao gồm tất cả các chi tiết cần thiết về từng đơn thuê.
3. Quản trị viên có thể xem trạng thái hiện tại của từng đơn thuê, bao gồm thông tin về người thuê, số tiền thuê, cọc, và các trạng thái hoàn trả.
4. Tùy vào trạng thái của từng đơn hàng, quản trị viên có thể nhấn các nút tương ứng để thay đổi hoặc cập nhật trạng thái đơn hàng:
 - a. Chờ người thuê thanh toán: Quản trị viên có thể nhấn nút “Người thuê đã trả tiền” khi nhận được thanh toán từ người thuê.
 - b. Đã trả cọc cho người thuê: Hiển thị khi cọc được trả lại cho người thuê sau khi đơn hàng hoàn tất.
5. Hệ thống có phân trang cho danh sách đơn hàng để giúp quản trị viên dễ dàng duyệt qua các đơn hàng khác nhau.

6.3.20.5 Đặc tả API

API: Lấy danh sách đơn thuê

- Phương thức: GET
- URL: /api/orders/admin
- Mô tả: Lấy danh sách tất cả các đơn hàng thuê sách cho quản trị viên.

API: Cập nhập đơn thuê

- Phương thức: POST
- URL: /api/order/{orderId}/change-status
- Mô tả: Xác nhận thanh toán cho đơn hàng thuê sách của người thuê khi quản trị viên nhấn nút “Người thuê đã trả tiền”.
- Tham số đầu vào:

Tên tham số	Loại dữ liệu	Bắt buộc	Mô tả
orderId	number	Có	ID của đơn thuê cần truy xuất
status	string	Có	Trạng thái thay đổi

Bảng 79: Tham số đầu vào API Cập nhập đơn thuê

6.3.21 Trang quản lý đơn hàng mua bán admin

ID	TÊN SÁCH	NGƯỜI BÁN	NGƯỜI MUA	NGÀY ĐẶT	PHƯƠNG THỨC	GIÁ BÁN	TIỀN TRẢ CHO NGƯỜI BÁN	TRẠNG THÁI
1	Wallis	Tracy Braun	Dev Developer	09/07/2024	Chuyển khoản	224.100đ	117.600đ	1. Chờ người mua thanh toán Người mua đã trả tiền
2	From Potter's Field	Dev Developer	Tracy Braun	09/07/2024	Chuyển khoản	414.800đ	9.000đ	1. Chờ người mua thanh toán Người mua đã trả tiền
3	Deep Waters	Dev Developer	Tracy Braun	01/04/2024	Chuyển khoản	90.000đ	84.000đ	5. Người mua đã hủy
4	Deep Waters	Dev Developer	Tracy Braun	26/12/2024	Chuyển khoản	90.000đ	84.000đ	2. Đã thanh toán
5	Foundling	Nigel Romaguera	Tracy Braun	05/10/2024	Chuyển khoản	383.700đ	252.000đ	3. Người bán đã giao sách
6	Ice	Giovani Kertmann	Tracy Braun	25/05/2024	Chuyển khoản	376.600đ	78.000đ	
7	Daring Moves	Charlotte Blick	Tracy Braun	24/05/2024	Chuyển khoản	105.700đ	48.000đ	
8	Alchemy and Academe	Adonis Labadie	Tracy Braun	27/03/2024	Chuyển khoản	120.300đ	36.000đ	
9	Houses of Stone	Toby Emmerich	Tracy Braun	01/04/2024	Chuyển khoản	76.500đ	60.000đ	4. Đã trả tiền cho người bán
10	Nemesis	Adolph Runolfsson	Tracy Braun	24/05/2024	Chuyển khoản	194.200đ	42.000đ	

Hình 74: Trang quản lý đơn hàng mua bán admin

Giống với trang quản lý đơn hàng thuê cho thuê nhưng lại về đơn hàng mua bán

6.3.22 Trang quản lý voucher toàn sàn admin

ID	TÊN VOUCHER	MÃ CODE	LOẠI VOUCHER	GIÁ TRỊ TỐI THIỂU	GIÁ TRỊ KHUYẾN MÃI	THỜI GIAN HIỆU LỰC	
1	Lunar New Year Bonus	LNY2024	Giảm theo tiền	10.000đ	20.000đ	01/11/2024 - 30/12/2024	
2	Valentine's Day Offer	VALENTINE10	Giảm theo phần trăm	150.000đ	10%	01/02/2024 - 14/02/2024	
3	Spring Festival Discount	SPRING5	Giảm theo tiền	50.000đ	5.000đ	01/03/2024 - 31/03/2024	
4	Women's Day Special	WOMEN15	Giảm theo phần trăm	200.000đ	15%	01/03/2024 - 08/03/2024	
5	Summer Bonanza	SUMMER30	Giảm theo tiền	300.000đ	30.000đ	01/06/2024 - 30/06/2024	
6	Mid-Autumn Sale	MIDAUTUMN12	Giảm theo phần trăm	100.000đ	12%	01/09/2024 - 15/09/2024	
7	Halloween Deal	HALLOWEEN5	Giảm theo tiền	50.000đ	5.000đ	01/10/2024 - 31/10/2024	
8	Black Friday Extravaganza	BLACKFRIDAY20	Giảm theo phần trăm	500.000đ	20%	29/11/2024 - 29/11/2024	
9	Cyber Monday Special	CYBERMONDAY50	Giảm theo tiền	400.000đ	50.000đ	01/12/2024 - 01/12/2024	
10	Christmas Countdown	XMAS10	Giảm theo phần trăm	250.000đ	10%	01/12/2024 - 24/12/2024	

Hình 75: Trang quản lý voucher toàn sàn admin

6.3.22.1 Tổng quan

- Mục đích:** Chức năng quản lý voucher được thiết kế để Admin có thể xem toàn bộ danh sách voucher hiện có và xóa các voucher không cần thiết. Tính năng này đảm bảo danh sách voucher được cập nhật, loại bỏ các mục không hợp lệ hoặc đã hết hạn, giúp hệ thống quản lý mã khuyến mãi hiệu quả và gọn gàng.
- Tác nhân tham gia:** Admin.

- Điều kiện tiên quyết:
 - Các voucher đã được tạo trước đó và tồn tại trong cơ sở dữ liệu của hệ thống.
 - Admin đã đăng nhập vào hệ thống với đủ quyền để truy cập tính năng quản lý voucher.
- Kết quả mong muốn:
 - Admin có thể xem danh sách voucher đầy đủ với thông tin chi tiết.
 - Admin có thể xóa thành công các voucher không cần thiết và hệ thống cập nhật danh sách mà không có lỗi phát sinh.

6.3.22.2 Quy tắc và Ràng buộc

- Admin chỉ có thể xóa các voucher không thuộc trạng thái "Đang sử dụng" hoặc đã hết hạn.
- Hệ thống phải cung cấp chức năng xác nhận trước khi Admin xóa voucher để tránh thao tác nhầm.
- Tất cả các thay đổi (xóa voucher) phải được ghi lại trong hệ thống để đảm bảo khả năng kiểm tra và khôi phục nếu cần.

6.3.22.3 Mô tả chi tiết

STT	Tên	Mô tả	Loại điều khiển	Kiểu dữ liệu	Giá trị mặc định	Bắt buộc	Quy tắc
1	ID Voucher	Mã định danh duy nhất của voucher.	Nhãn	Số nguyên	Tự động tạo	X	Không thể chỉnh sửa.
2	Tên Voucher	Tên của voucher.	Nhãn	nvarchar(255)		X	Được hiển thị rõ ràng cho Admin để nhận biết loại voucher.
3	Mã Code	Mã code của voucher để áp dụng khuyến mãi.	Nhãn	nvarchar(20)		X	Hiển thị định dạng đúng (Ví dụ: SPRING5, LNY2024).
4	Loại Voucher	Loại khuyến mãi (giảm theo tiền hoặc phần trăm).	Nhãn	nvarchar(50)		X	Không thể chỉnh sửa.
5	Giá trị tối thiểu	Giá trị tối thiểu của đơn hàng để áp dụng voucher.	Nhãn	Số tiền			Hiển thị đúng đơn vị tiền tệ của hệ thống (VD: VNĐ).

6	Giá trị khuyến mãi	Số tiền hoặc tỷ lệ phần trăm được giảm khi sử dụng voucher.	Nhấn	Số tiền/tỷ lệ			Hiển thị rõ ràng giá trị khuyến mãi theo đúng loại voucher.
7	Thời gian hiệu lực	Khoảng thời gian voucher có hiệu lực (từ ngày - đến ngày).	Nhấn	Ngày		X	Hiển thị định dạng ngày tháng chính xác.
8	Xóa Voucher	Nút "Xóa"	Nút bấm				Khi được nhấn, voucher sẽ bị xóa sau khi Admin xác nhận hành động.

Bảng 80: Mô tả chi tiết Trang quản lý voucher toàn sàn admin

6.3.22.4 Quy trình xử lý

1. Admin đăng nhập vào hệ thống với tài khoản có quyền truy cập chức năng quản lý voucher.
2. Admin vào mục "Quản lý voucher" từ menu chính.
3. Danh sách các voucher hiện có được hiển thị đầy đủ, bao gồm thông tin chi tiết như ID, tên, loại, giá trị và thời gian hiệu lực.
4. Để xóa một voucher, Admin nhấn nút "Xóa" tại dòng voucher tương ứng.
5. Hệ thống hiển thị hộp thoại xác nhận với thông tin cụ thể về voucher sẽ xóa.
6. Sau khi Admin xác nhận, hệ thống xóa voucher và cập nhật lại danh sách trên giao diện.
7. Thay đổi được ghi lại trong hệ thống để đảm bảo khả năng kiểm tra sau này.

6.3.22.5 Đặc tả API

API: Lấy danh sách voucher toàn sàn

- Phương thức: GET
- **Endpoint:** /api/voucher-session
- Tham số đầu vào: không có.

API: Xóa voucher toàn sàn

- Phương thức: DELETE
- **Endpoint:** /api/voucher-session/{voucherId}
- Tham số đầu vào:

Tên tham số	Loại dữ liệu	Mô tả	Bắt buộc
voucherId	Số nguyên	ID voucher cần xóa	Có

Bảng 81: Tham số đầu vào API Xóa voucher toàn sàn

6.3.23 Trang thêm và sửa voucher toàn sàn admin

The Flying bookstore

Tạo voucher mới

Tên voucher

Mã voucher

Ngày bắt đầu

Ngày kết thúc

Giá trị tối thiểu

Giá trị tối đa

Loại voucher

Giảm giá bằng số tiền

Số tiền giảm

XÁC NHẬN

© 2024. Bản quyền thuộc về đội ngũ The Flying Bookstore.

Hình 76: Trang thêm và sửa voucher toàn sàn admin

6.3.23.1 Tổng quan

- Mục đích:** Cung cấp khả năng cho quản trị viên quản lý các chương trình khuyến mãi bằng cách tạo mới hoặc chỉnh sửa voucher.
- Đối tượng tham gia:** Quản trị viên.
- Điều kiện tiên quyết:**
 - Admin đã đăng nhập vào hệ thống.
 - Có quyền truy cập vào trang quản lý voucher.
- Kết quả mong muốn:** Voucher được thêm hoặc sửa thành công và có thể sử dụng trên hệ thống.

6.3.23.2 Quy tắc và Ràng buộc

- Các thông tin về voucher phải được cung cấp đầy đủ và chính xác.
- Ngày bắt đầu và ngày kết thúc phải hợp lệ (Ngày bắt đầu \leq Ngày kết thúc).
- Số tiền giảm hoặc phần trăm giảm giá phải là số dương và không vượt quá giá trị tối thiểu.
- Mã voucher phải là duy nhất.

6.3.23.3 Mô tả chi tiết

1. Thêm Voucher

STT	Tên	Mô tả	Loại điều khiển	Kiểu dữ liệu	Giá trị mặc định	Bắt buộc	Quy tắc
1	Tên voucher	Tên của chương trình khuyến mãi	Textfield	Chuỗi	N/A	Có	Không trống

2	Mã voucher	Mã duy nhất để áp dụng	Textfield	Chuỗi	N/A	Có	Phải là duy nhất
3	Ngày bắt đầu	Thời gian voucher có hiệu lực	Date picker	Ngày	N/A	Có	\leq Ngày kết thúc
4	Ngày kết thúc	Thời gian hết hiệu lực của voucher	Date picker	Ngày	N/A	Có	\geq Ngày bắt đầu
5	Giá trị tối thiểu	Giá trị đơn hàng tối thiểu để áp dụng	Number field	Số	0	Có	Số dương
6	Loại voucher	Hình thức giảm giá (tiền hoặc phần trăm)	Dropdown	Enum	N/A	Có	Chỉ nhận giá trị hợp lệ
7	Số tiền giảm/phần trăm giảm	Giá trị giảm giá	Number field	Số	N/A	Có	Số dương

Bảng 82: Mô tả chi tiết Trang thêm và sửa voucher toàn sàn admin

2. Sửa Voucher

- Giao diện tương tự thêm mới voucher, với các trường được tự động điền dữ liệu từ voucher cần sửa.
- Các quy tắc kiểm tra tương tự như khi thêm mới.

6.3.23.4 Quy trình xử lý

1. Thêm Voucher

1. Admin truy cập trang Tạo voucher mới.
2. Điền đầy đủ thông tin cần thiết vào các trường bắt buộc.
3. Nhấn nút Xác nhận.
4. Hệ thống kiểm tra tính hợp lệ:
 - a. Nếu thông tin hợp lệ:
 - i. Lưu voucher vào cơ sở dữ liệu.
 - ii. Hiển thị thông báo Tạo voucher thành công.
 - b. Nếu thông tin không hợp lệ: Hiển thị thông báo lỗi chi tiết (Ví dụ: "Mã voucher đã tồn tại").

2. Sửa Voucher

1. Admin truy cập trang Quản lý voucher và chọn voucher cần chỉnh sửa.
2. Thực hiện chỉnh sửa các thông tin cần thay đổi.
3. Nhấn nút Xác nhận.
4. Hệ thống kiểm tra:
 - a. Nếu thông tin hợp lệ:
 - i. Cập nhật thông tin trong cơ sở dữ liệu.

- ii. Hiển thị thông báo Cập nhật voucher thành công.
- b. Nếu thông tin không hợp lệ: Hiển thị thông báo lỗi chi tiết (Ví dụ: "Ngày bắt đầu không hợp lệ").

6.3.23.5 Đặc tả API

API: Thêm voucher toàn sàn mới

- Phương thức: POST
- **Endpoint:** /api/voucher-session/add
- Tham số đầu vào:

Tên tham số	Loại dữ liệu	Mô tả	Bắt buộc
name	Chuỗi	Tên voucher	Có
code	Chuỗi	Mã voucher	Có
startDate	Chuỗi	Ngày bắt đầu (YYYY-MM-DD)	Có
endDate	Chuỗi	Ngày kết thúc (YYYY-MM-DD)	Có
minValue	Số	Giá trị tối thiểu	Có
discountType	Enum	Loại giảm giá (tiền hoặc phần trăm)	Có
discountValue	Số	Số tiền/phần trăm giảm	Có

Bảng 83: Tham số đầu vào API Thêm voucher toàn sàn mới

API: Sửa voucher toàn sàn

- Phương thức: PUT
- **Endpoint:** /api/voucher-session/edit/{voucherId}
- **Tham số đầu vào:** Tương tự thêm mới, chỉ khác có thêm id.

6.3.24 Trang đăng ký admin

The Flying Bookstore

Đăng ký

Đăng nhập

Chào mừng bạn!

Đăng ký

Tên đăng nhập

Email

Mật khẩu

Xác nhận mật khẩu

Tôi đồng ý với Điều khoản và Điều kiện

ĐĂNG KÝ

Đã có tài khoản rồi? Đăng nhập

© 2024 Bản quyền thuộc về đội ngũ The Flying Bookstore.

Hình 77: Trang đăng ký admin

Giống như trang đăng nhập nhưng thêm email và xác nhận mật khẩu và dùng api đăng ký

6.3.24.1 Tổng quan

- Mục đích:** Cho phép admin đăng ký tài khoản quản trị thông qua giao diện đăng ký.
- Tác nhân tham gia:** Admin (có quyền truy cập đặc biệt).
- Điều kiện tiên quyết:** Hệ thống đã xác định vai trò admin.
- Kết quả mong muốn:** Admin đăng ký thành công và có thể đăng nhập với quyền quản trị.

6.3.24.2 Quy tắc và Ràng buộc

- Kiểm tra vai trò admin khi đăng ký: Admin cần nhập mã xác thực hoặc email đặc biệt để được cấp quyền.
- Thông tin bắt buộc khi đăng ký:
 - Tên đăng nhập, email, mật khẩu, và xác nhận mật khẩu.
 - Đồng ý với điều khoản và điều kiện.
- Phân quyền tự động: Sau khi đăng ký, tài khoản được gán vai trò admin trong cơ sở dữ liệu.

6.3.24.3 Mô tả chi tiết

STT	Tên	Mô tả	Loại điều khiển	Kiểu dữ liệu	Giá trị mặc định	Bắt buộc	Quy tắc
1	Tên đăng nhập	Tên đăng nhập của admin	Textbox	varchar(50)		X	Admin phải nhập tên đăng nhập hợp lệ. Không chứa ký tự đặc biệt, không trùng lặp với các tên đăng nhập khác trong hệ thống.
2	Email	Địa chỉ email để đăng ký tài khoản admin	Textbox	nvarchar(255)		X	Email phải đúng định dạng, duy nhất trong hệ thống.
3	Mật khẩu	Mật khẩu bảo vệ tài khoản admin	Password	nvarchar(255)		X	Mật khẩu phải chứa ít nhất 8 ký tự, bao gồm ít nhất một chữ in hoa, một số, và một ký tự đặc biệt.
4	Xác nhận mật khẩu	Nhập lại mật khẩu để xác nhận	Password	nvarchar(255)		X	Trường này phải khớp với mật khẩu đã nhập ở trên.
5	Mã xác thực admin	Mã xác thực để hệ thống nhận biết người	Textbox	nvarchar(50)		X	Mã xác thực phải đúng với mã hệ thống đã cung cấp trước đó.

		đăng ký là admin					
6	Đồng ý điều khoản	Checkbox xác nhận admin đồng ý với các điều khoản và điều kiện sử dụng hệ thống	Checkbox	Boolean	FALSE	X	Admin phải tích chọn vào ô này trước khi đăng ký.
7	Đăng ký	Nút bấm để hoàn tất quá trình đăng ký	Button				Khi nhấn, hệ thống kiểm tra thông tin hợp lệ, lưu tài khoản admin vào cơ sở dữ liệu, đồng thời gửi email xác nhận (nếu yêu cầu).

Bảng 84: Mô tả chi tiết Trang đăng ký admin

6.3.24.4 Quy trình xử lý

1. Kiểm tra thông tin nhập liệu:
 - a. Hệ thống kiểm tra tính hợp lệ của tất cả các trường bắt buộc.
 - b. Báo lỗi nếu trường nào không hợp lệ (hiển thị thông báo cụ thể cho từng lỗi).
2. Xử lý mã xác thực admin:
 - a. Hệ thống kiểm tra mã xác thực admin với cơ sở dữ liệu.
 - b. Nếu mã không đúng, từ chối đăng ký và hiển thị thông báo lỗi.
3. Gửi email xác nhận (tùy chọn): Sau khi đăng ký thành công, hệ thống có thể gửi email xác nhận đến email của admin.
4. Phân quyền: Tài khoản sau khi đăng ký sẽ tự động được gán vai trò admin.

6.3.24.5 Đặc tả API

API: Đăng ký tài khoản mới

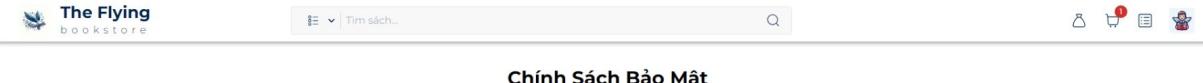
- Phương thức: POST
- **Endpoint:** /api/user/register
- Tham số đầu vào:

Tên tham số	Loại dữ liệu	Mô tả	Bắt buộc
username	varchar(255)	Tên đăng nhập	Có
email	varchar(255)	Địa chỉ email	Có
phoneNumber	varchar(255)	Số điện thoại	Có
firstName	varchar(255)	Tên	Có
lastName	varchar(255)	Họ	Có

birthDate	Date	Ngày sinh	Có
avatarUrl	varchar(255)	URL ảnh đại diện	Không
address	varchar(255)	Địa chỉ	Có
password	varchar(255)	Mật khẩu	Có

Bảng 85: Tham số đầu vào API Đăng ký tài khoản mới

6.3.25 Trang chính sách bảo mật



Chính Sách Bảo Mật

Chính sách bảo mật này quy định cách thức mà The Flying Bookstore thu thập, sử dụng, duy trì và tiết lộ thông tin thu thập từ người dùng (mỗi người là Người Dùng) của trang web The Flying Bookstore.com. Chính sách bảo mật này áp dụng cho Trang Web và tất cả các sản phẩm và dịch vụ do The Flying Bookstore cung cấp.

Thông Tin Nhận Dạng Cá Nhân

Chúng tôi có thể thu thập thông tin nhận dạng cá nhân từ Người Dùng theo nhiều cách khác nhau, bao gồm, nhưng không giới hạn, khi Người Dùng truy cập trang web của chúng tôi, đăng ký trên trang web, đặt hàng, đăng ký nhận bản tin, và liên quan đến các hoạt động, dịch vụ, tình năng hoặc tài nguyên khác mà chúng tôi cung cấp trên Trang Web của chúng tôi. Người Dùng có thể được yêu cầu, khi thich hợp, cung cấp tên, địa chỉ email, địa chỉ gửi thư, số điện thoại. Tuy nhiên, Người Dùng có thể truy cập Trang Web của chúng tôi ăn danh. Chúng tôi sẽ chỉ thu thập thông tin nhận dạng cá nhân từ Người Dùng nếu họ tự nguyện gửi thông tin đó cho chúng tôi. Người Dùng luôn có thể từ chối cung cấp thông tin nhận dạng cá nhân, ngoại trừ việc nó có thể ngăn họ tham gia vào một số hoạt động liên quan đến Trang Web.

Thông Tin Không Nhận Dạng Cá Nhân

Chúng tôi có thể thu thập thông tin không nhận dạng cá nhân như Người Dùng mỗi khi họ tương tác với Trang Web của chúng tôi. Thông tin không nhận dạng cá nhân có thể bao gồm tên trinh duyệt, loại máy tính và thông tin kỹ thuật về phương tiện kết nối của Người Dùng đến Trang Web của chúng tôi, chẳng hạn như điều hành và các nhà cung cấp dịch vụ Internet được sử dụng và các thông tin tương tự khác.

Cookie Trình Duyệt Web

Trang Web của chúng tôi có thể sử dụng cookie để cải thiện trải nghiệm của Người Dùng. Trình duyệt web của Người Dùng đặt cookie trên ổ cứng của họ cho mục đích lưu trữ hồ sơ và dõi theo để dõi thông tin về họ. Người Dùng có thể chọn đặt trình duyệt web của họ để chặn cookie, hoặc cảnh báo họ khi cookie đang được gửi. Nếu họ làm như vậy, lưu ý rằng một số phần của Trang Web có thể không hoạt động đúng.

Chắc Chắn Tôi Sử Dụng Thông Tin Thu Thập Được

The Flying Bookstore có thể thu thập và sử dụng thông tin cá nhân của Người Dùng cho các mục đích sau:

Để cải thiện dịch vụ khách hàng: Thông tin Người Dùng cung cấp giúp chúng tôi đáp ứng các yêu cầu dịch vụ khác hàng và nhu cầu hỗ trợ của họ một cách hiệu quả hơn.

Để cá nhân hóa trải nghiệm người dùng: Chúng tôi có thể sử dụng thông tin trong trình duyệt để hiểu cách Người Dùng của chúng tôi như một nhóm sử dụng các dịch vụ và tài nguyên được cung cấp trên Trang Web của chúng tôi.

Để cải thiện Trang Web của chúng tôi: Chúng tôi có thể sử dụng phản hồi mà Người Dùng cung cấp để cải thiện các sản phẩm và dịch vụ của chúng tôi.

Để xử lý thanh toán: Chúng tôi có thể sử dụng thông tin mà Người Dùng cung cấp về bán thân khi đặt hàng để cung cấp dịch vụ cho đơn hàng đó. Chúng tôi không chia sẻ thông tin này với các bên ngoài trừ khi cần thiết để cung cấp dịch vụ.

Để gửi email và SMS định kỳ: Chúng tôi có thể sử dụng địa chỉ email hoặc số điện thoại để gửi thông tin và cập nhật liên quan đến đơn hàng của Người Dùng. Nó cũng có thể được sử dụng để trả lời các yêu cầu, câu hỏi v.v./hoặc các yêu cầu khác của họ. Nếu Người Dùng quyết định đăng ký vào danh sách gửi thư của chúng tôi, họ sẽ nhận được email có thể bao gồm tin tức công ty, cập nhật, thông tin sản phẩm hoặc dịch vụ liên quan, vv. Nếu bắt đầu nhận được tin nhắn này, họ có thể hủy đăng ký nhận email trong tương lai, chúng tôi có bao gồm hướng dẫn hủy đăng ký chỉ định ở cuối mỗi email, hoặc Người Dùng có thể liên hệ với chúng tôi qua Trang Web của chúng tôi. Nếu Người Dùng đặt đơn hàng thuê với chúng tôi, một tin nhắn nhắc nhở SMS sẽ được gửi tự động để thông báo cho Người Dùng về ngày hết hạn của đơn hàng thuê và cung cấp hướng dẫn trả hàng. Các tin nhắn nhắc nhở SMS sẽ dừng tự động sau khi đóng tất cả các giao dịch thuê hiện tại với chúng tôi.

Chính Sách SMS

Yêu Cầu Đồng Ý / Đồng Ký

Chúng tôi sử dụng loại chiến dịch không tiếp thị này để gửi một lần nhắc nhở SMS trả hàng thuê cho Người Dùng về ngày hết hạn đơn hàng thuê khi họ phải trả lại mặt hàng đã thuê cho <https://The Flying Bookstore.com/> với vận chuyển trả hàng miễn phí.

Các tin nhắn nhắc nhở trả hàng thuê chỉ được gửi cho những Người Dùng có đơn hàng thuê đang hoạt động được đặt trên trang web.

Người Dùng đăng ký nhận SMS trong quá trình thanh toán đơn hàng bằng cách đánh dấu vào ô rằng họ đồng ý nhận nhắc nhở trả hàng miễn phí qua SMS. Bằng cách xác nhận đơn hàng thuê, Người Dùng đồng ý nhận một tin nhắn nhắc nhở trả hàng cho mỗi mặt hàng từ The Flying Bookstore qua SMS (tư 0905 907 362). Mỗi SMS sẽ chứa ngày hết hạn và một liên kết đến trang đổi hàng thuê. Khi Người Dùng đóng tất cả các giao dịch thuê hiện tại, họ sẽ ngừng nhận SMS từ The Flying Bookstore.

Hủy Đăng Ký

Người Dùng có thể hủy đăng ký nhận các tin nhắn nhắc nhở trả hàng bằng cách trả lời STOP đến 0905 907 362 và sẽ không còn nhận tin nhắn SMS từ The Flying Bookstore nữa.

Trợ Giúp

Bằng cách gửi tin nhắn Trợ Giúp đến 0905 907 362, Người Dùng sẽ nhận được một tin nhắn SMS với các liên hệ của đội Hỗ Trợ của chúng tôi.

Tiết Lộ Giá

Các tin nhắn nhắc nhở trả hàng thuê của chúng tôi là miễn phí. Các khoản phí có thể áp dụng nếu Người Dùng gửi cho chúng tôi một tin nhắn SMS tùy thuộc vào gói dịch vụ được sử dụng và các khoản phí được đặt ra bởi nhà cung cấp dịch vụ.

Chắc Chắn Tôi Bảo Vệ Thông Tin Người Dùng

Chúng tôi áp dụng các biện pháp thu thập dữ liệu, lưu trữ, và xử lý phù hợp và các biện pháp bảo mật để bảo vệ chống truy cập trái phép, thay đổi, tiết lộ, hoặc phá hủy thông tin cá nhân của bạn, tên đăng nhập, mật khẩu, thông tin giao dịch và dữ liệu được lưu trữ trên Trang Web của chúng tôi.

Chia Sẻ Thông Tin Cá Nhân Của Người Dùng

Chúng tôi không bán, trao đổi, hoặc thuê thông tin nhận dạng cá nhân của Người Dùng cho người khác. Chúng tôi có thể chia sẻ thông tin nhận dạng cá nhân học tông hợp chung không liên kết với bất kỳ thông tin nhận dạng cá nhân nào liên quan đến khách hàng trực tiếp và người dùng với các đối tác kinh doanh, các bên liên kết đăng tin cậy và các nhà quảng cáo cho các mục đích như trên. Chúng tôi có thể sử dụng các nhà cung cấp dịch vụ bên thứ ba để giúp chúng tôi vận hành kinh doanh (giao hàng đơn hàng, xử lý thanh toán, vv.) và Trang Web hoặc quản lý các hoạt động thay mặt chúng tôi, chẳng hạn như gửi bản tin hoặc khảo sát. Không có thông tin di động nào sẽ được chia sẻ với các bên thứ ba/dối tác cho các mục đích tiếp thị/quảng cáo. Số điện thoại của Người Dùng không được chia sẻ với bất kỳ bên thứ ba nào trừ khi Người Dùng vi phạm chính sách của chúng tôi yêu cầu các hành động pháp lý từ phía chúng tôi.

Thay Đổi Chính Sách Bảo Mật

The Flying Bookstore có quyền cập nhật chính sách bảo mật này bất cứ lúc nào. Khi chúng tôi làm vậy, chúng tôi sẽ sửa đổi ngày cập nhật tại cuối trang này. Chúng tôi khuyến khích Người Dùng thường xuyên kiểm tra trang này để có biết kỹ thay đổi nào để luôn cập nhật về cách chúng tôi bảo vệ thông tin cá nhân mà chúng tôi thu thập. Bạn thừa nhận và đồng ý rằng bạn có trách nhiệm xem lại chính sách bảo mật này định kỳ và nhận thức được các sửa đổi.

Sự Chấp Nhận Của Bạn Về Các Điều Khoản Này

Bằng cách sử dụng Trang Web này, bạn đồng ý với chính sách này. Nếu bạn không đồng ý với chính sách này, vui lòng không sử dụng Trang Web của chúng tôi. Việc bạn tiếp tục sử dụng Trang Web sau khi có thay đổi chính sách này sẽ được coi là bạn chấp nhận những thay đổi đó.

Liên Hệ Với Chúng Tôi

Nếu bạn có bất kỳ câu hỏi nào về Chính Sách Bảo Mật này, các thực tiễn của trang web này, hoặc các giao dịch của bạn với trang web này, hoặc trong trường hợp bạn muốn yêu cầu xóa dữ liệu của mình mà chúng tôi sử dụng trên nền tảng của mình, vui lòng liên hệ với chúng tôi tại:

The Flying Bookstore
173/11 Trương Phước Phan
Bình Trị Đông, quận Bình Tân
Cập nhật vào ngày 27 tháng 5 năm 2024

Hình 78: Trang chính sách bảo mật

6.3.25.1 Tổng quan

- Mục đích:** Chức năng này cho phép người dùng xem trang "Chính Sách Bảo Mật" để hiểu rõ cách hệ thống xử lý, bảo vệ và quản lý thông tin cá nhân của họ khi sử dụng dịch vụ. Trang này cung cấp thông tin minh bạch và đầy đủ để đảm bảo người dùng có thể an tâm khi sử dụng website.

- **Tác nhân tham gia:** Người dùng (bao gồm cả khách hàng và quản trị viên).
- Điều kiện tiên quyết:
 - Người dùng có thể truy cập website mà không cần đăng nhập.
 - Trang "Chính Sách Bảo Mật" đã được thiết kế và cập nhật đầy đủ nội dung.
- Kết quả mong muốn:
 - Người dùng có thể đọc được thông tin về chính sách bảo mật một cách chi tiết và đầy đủ.
 - Trang được hiển thị rõ ràng và thân thiện với người dùng trên mọi thiết bị.

6.3.25.2 Quy tắc và Ràng buộc

- Nội dung trên trang "Chính Sách Bảo Mật" chỉ mang tính chất tham khảo, không thể chỉnh sửa trực tiếp từ giao diện người dùng.
- Trang phải hiển thị đúng bô cục, phông chữ và định dạng để dễ dàng đọc hiểu.
- Chỉ hỗ trợ hiển thị nội dung tĩnh, không có API được gọi trong tính năng này.
- Nội dung được cập nhật bởi quản trị viên thông qua hệ thống quản lý nội dung (CMS).

6.3.25.3 Mô tả chi tiết

STT	Tên	Mô tả	Loại điều khiển	Kiểu dữ liệu	Giá trị mặc định	Bắt buộc	Quy tắc
1	Tiêu đề trang	Tiêu đề chính của trang (Chính Sách Bảo Mật).	Nhãn	nvarchar(255)	Chính Sách Bảo Mật	X	Hiển thị rõ ràng, nổi bật trên đầu trang.
2	Nội dung chính	Các đoạn văn bản miêu tả chính sách bảo mật của hệ thống.	Nhãn	nvarchar(MAX)		X	Nội dung hiển thị đầy đủ theo thông tin được quản trị viên nhập trong hệ thống CMS.

Bảng 86: Mô tả chi tiết Trang chính sách bảo mật

6.3.25.4 Quy trình xử lý

1. Người dùng nhấn chọn mục "Chính Sách Bảo Mật" từ trang đăng nhập hoặc footer của trang.
2. Hệ thống điều hướng người dùng đến trang "Chính Sách Bảo Mật".
3. Trang hiển thị nội dung tĩnh bao gồm:
 - a. Tiêu đề "Chính Sách Bảo Mật".
 - b. Các mục nội dung liên quan như: Thông tin cá nhân, cookie, bảo mật dữ liệu, chia sẻ thông tin.
4. Người dùng đọc nội dung và không thực hiện thêm tương tác nào khác (chỉ xem thông tin).

6.3.26 Trang hướng dẫn về tình trạng sách

The screenshot shows the website's header with the logo 'The Flying bookstore'. Below the header is a search bar with placeholder text 'Tim sách...'. To the right of the search bar are icons for user profile, cart, and other site functions. The main content area has a title 'Hướng dẫn về tình trạng sách'. Below the title is a detailed text block explaining book conditions. It includes sections for 'Giải thích về tình trạng sách', 'Mới - hư hại 0%', 'Như mới - hư hại 10%', 'Rất tốt - hư hại 20%', 'Tốt - hư hại 30%', and 'Chấp nhận được - hư hại hơn 30%'. A note at the bottom states that the guidelines help customers find suitable books. At the bottom of the page is a dark banner with the text 'Nhập mail để nhận được những quà tặng đặc biệt nhé!' (Enter your email to receive special gifts!) and a map showing the store's location at 171/11 Trương Phước Phan, Bình Hưng Hòa, Bình Tân, TP.HCM.

Hình 79: Trang hướng dẫn về tình trạng sách

6.3.26.1 Tổng quan

- Mục đích:** Chức năng này giúp người dùng hiểu rõ cách hệ thống phân loại và mô tả tình trạng sách, từ đó có thể đưa ra lựa chọn phù hợp nhất khi mua hoặc thuê sách.
- Tác nhân tham gia:** Người dùng (bao gồm cả khách hàng và quản trị viên).
- Điều kiện tiên quyết:**
 - Người dùng có thể truy cập website mà không cần đăng nhập.
 - Trang "Hướng Dẫn Về Tình Trạng Sách" đã được thiết kế và cập nhật đầy đủ nội dung.
- Kết quả mong muốn:**
 - Người dùng có thể đọc và hiểu được các tiêu chí phân loại tình trạng sách một cách rõ ràng và chính xác.
 - Nội dung hiển thị rõ ràng, dễ đọc và phù hợp với mọi thiết bị.

6.3.26.2 Quy tắc và Ràng buộc

- Nội dung trên trang "Hướng Dẫn Về Tình Trạng Sách" chỉ mang tính chất tham khảo, không thể chỉnh sửa trực tiếp từ giao diện người dùng.
- Trang hiển thị nội dung tĩnh và không yêu cầu gọi API.

6.3.26.3 Mô tả chi tiết

STT	Tên	Mô tả	Loại điều khiển	Kiểu dữ liệu	Giá trị mặc định	Bắt buộc	Quy tắc

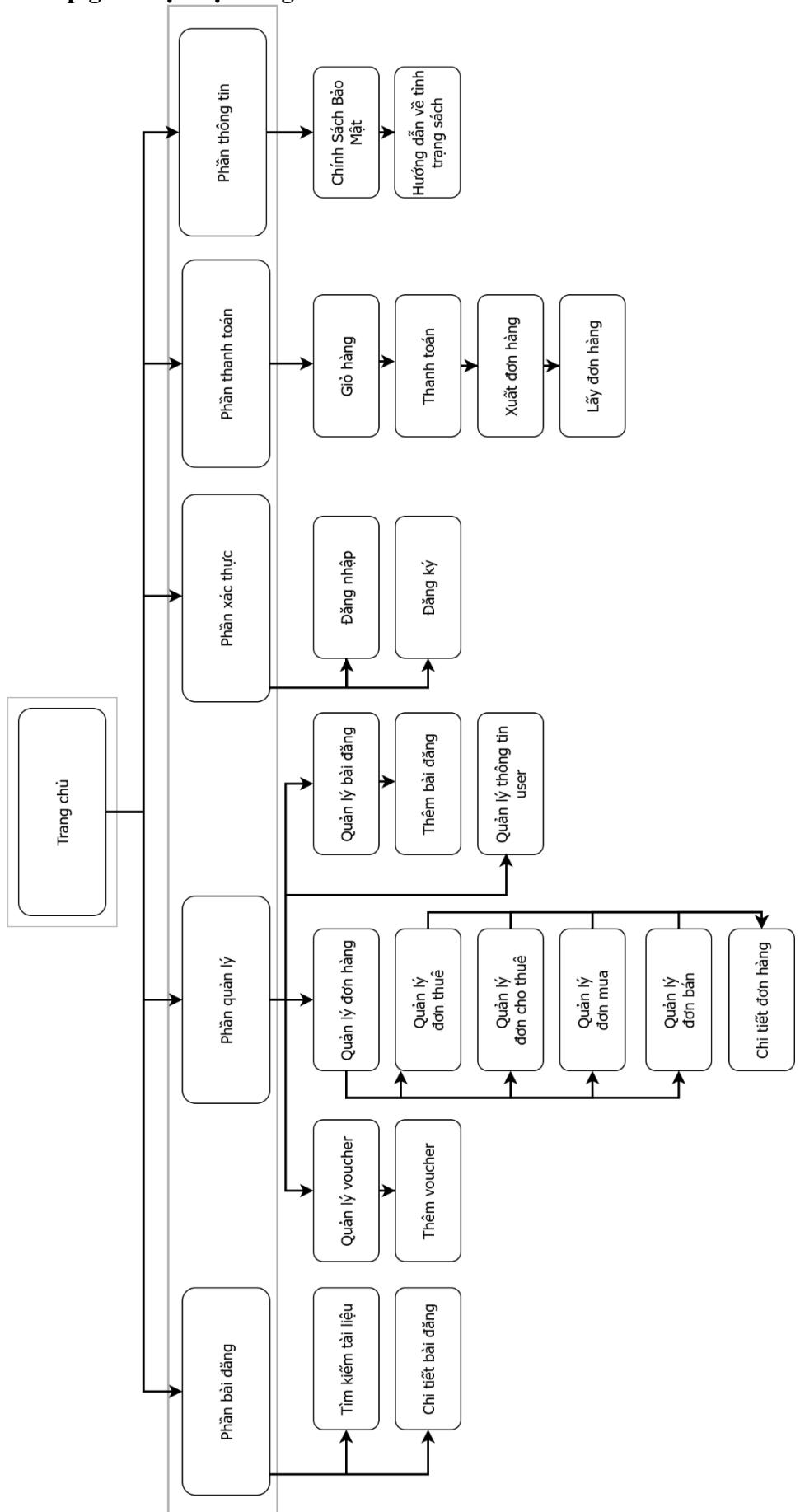
1	Tiêu đề trang	Tiêu đề chính của trang (Hướng Dẫn Về Tình Trạng Sách).	Nhãn	nvarchar(255)	Hướng Dẫn Về Tình Trạng Sách	X	Hiển thị rõ ràng, nổi bật trên đầu trang.
2	Nội dung chính	Các đoạn văn bản giải thích về các loại tình trạng sách và cách phân loại.	Nhãn	nvarchar(MAX)		X	Nội dung hiển thị đầy đủ theo thông tin được quản trị viên nhập trong hệ thống CMS.
3	Danh sách phân loại	Liệt kê các tình trạng sách như: Mới, Như mới, Rất tốt, Tốt, Chấp nhận được.	Nhãn	nvarchar(500)		X	Mỗi loại sách được mô tả chi tiết để giúp người dùng dễ dàng hiểu và lựa chọn.

Bảng 87: Mô tả chi tiết Trang hướng dẫn về tình trạng sách

6.3.26.4 Quy trình xử lý

1. Người dùng nhấn chọn mục "Hướng Dẫn Về Tình Trạng Sách" từ trang Tạo bài đăng hoặc footer của trang.
2. Hệ thống điều hướng người dùng đến trang "Hướng Dẫn Về Tình Trạng Sách".
3. Trang hiển thị nội dung tinh bao gồm:
 - a. Tiêu đề "Hướng Dẫn Về Tình Trạng Sách".
 - b. Các mục nội dung giải thích các tiêu chí phân loại tình trạng sách.
 - c. Danh sách phân loại tình trạng sách với các mô tả chi tiết tương ứng.
4. Người dùng đọc nội dung và không thực hiện thêm tương tác nào khác (chỉ xem thông tin).

6.4 Site map giao diện hệ thống



Hình 80: Site map giao diện hệ thống

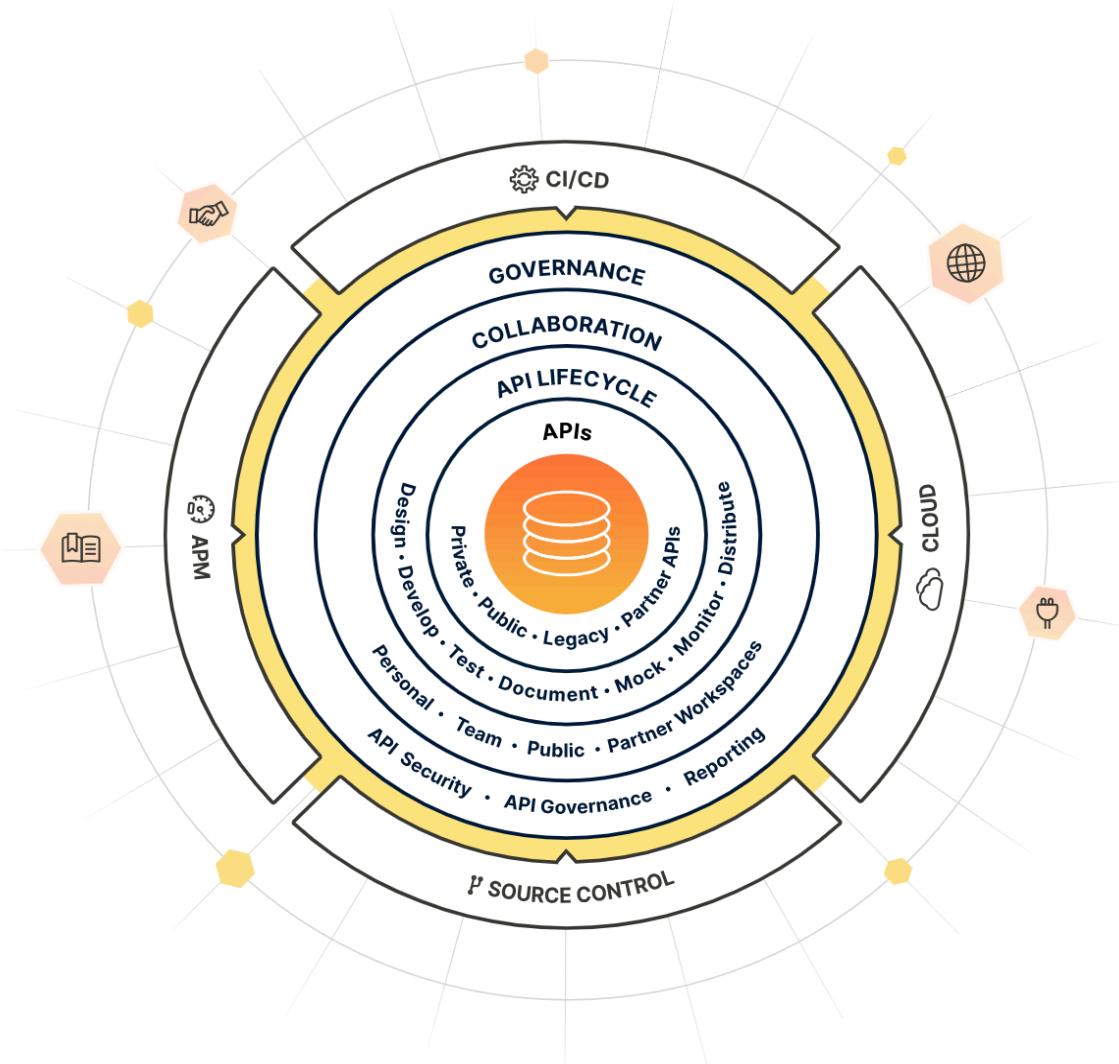
1. Trang chủ: Là điểm bắt đầu chính của người dùng khi truy cập vào hệ thống.
2. Phần thông tin: Chứa các thông tin liên quan đến:
 - a. **Chính sách bảo mật:** Trình bày cách hệ thống bảo vệ thông tin cá nhân của người dùng.
 - b. **Hướng dẫn về tình trạng sách:** Cung cấp thông tin hướng dẫn kiểm tra hoặc đánh giá tình trạng sách.
3. Phần thanh toán: Quy trình thanh toán bao gồm:
 - a. **Giỏ hàng:** Nơi người dùng xem các sách đã chọn.
 - b. **Thanh toán:** Thực hiện giao dịch mua hoặc thuê sách.
 - c. **Xuất đơn hàng:** Tạo hóa đơn cho đơn hàng.
 - d. **Lấy đơn hàng:** Quản lý việc nhận sách.
4. Phần xác thực: Hỗ trợ việc đăng nhập và đăng ký tài khoản, bao gồm:
 - a. **Đăng nhập:** Dành cho người dùng đã có tài khoản.
 - b. **Đăng ký:** Dành cho người dùng mới.
5. Phần bài đăng: Hỗ trợ quản lý và tương tác với các bài đăng sách, bao gồm:
 - a. **Tìm kiếm tài liệu:** Tìm kiếm thông tin sách cụ thể.
 - b. **Chi tiết bài đăng:** Xem thông tin chi tiết về sách.
 - c. **Thêm bài đăng:** Người dùng có thể thêm bài đăng mới về sách để bán hoặc cho thuê.
6. Phần quản lý: Chứa các chức năng dành cho quản trị viên và người dùng:
 - a. **Quản lý thông tin user:** Xem và cập nhật thông tin cá nhân của người dùng.
 - b. **Quản lý đơn hàng:** Bao gồm:
 - i. **Quản lý đơn thuê:** Theo dõi và xử lý các đơn thuê sách.
 - ii. **Quản lý đơn mua:** Theo dõi và xử lý các đơn mua sách.
 - iii. **Quản lý đơn bán:** Theo dõi các sách được bán bởi người dùng.
 - iv. **Quản lý đơn cho thuê:** Theo dõi sách mà người dùng đã cho thuê.
 - v. **Chi tiết đơn hàng:** Xem thông tin chi tiết đơn hàng.
 - c. **Quản lý voucher:**
 - i. **Thêm voucher:** Tạo mã giảm giá mới.
 - ii. **Quản lý voucher:** Theo dõi, chỉnh sửa, hoặc xóa mã giảm giá.

CHƯƠNG 7: KIỂM THỬ

7.1 Công cụ kiểm thử

7.1.1 Công cụ hỗ trợ kiểm thử postman

Postman là một ứng dụng được sử dụng rộng rãi trong quá trình phát triển và kiểm thử các API (Application Programming Interface). Nó cung cấp một giao diện người dùng đồ họa dễ sử dụng để gửi yêu cầu HTTP đến các endpoint API, hỗ trợ việc xem và phân tích phản hồi từ server.



Hình 81: Cấu trúc Postman

Một số tính năng chính của Postman phục vụ việc kiểm thử api backend bao gồm:

- Tạo và Gửi Yêu Cầu HTTP: Tạo và gửi các loại yêu cầu HTTP như GET, POST, PUT, DELETE, PATCH, v.v. đến các API endpoints.
- Xây Dựng và Tùy Chỉnh Yêu Cầu: Postman cho phép thêm các tham số, header, body và authentication vào các yêu cầu của api. Cũng có thể lập kế hoạch cho các yêu cầu hoặc sử dụng các biến để tái sử dụng thông tin.
- Tạo và Quản Lý Bộ Kiểm Tra (Test Suite): Bạn có thể tạo các bộ kiểm tra để tự động kiểm tra phản hồi từ server và xác minh tính đúng đắn của API.
- Môi Trường (Environment): Postman cho phép định nghĩa các biến môi trường, giúp dễ dàng chuyển đổi giữa các môi trường như development, staging, production, v.v.

- Xem và Phân Tích Phản Hồi: Có thể xem phản hồi từ server trong nhiều định dạng khác nhau như JSON, XML, HTML, v.v. Postman cung cấp các công cụ phân tích phản hồi và hiển thị dữ liệu một cách trực quan.

7.1.2 Công cụ hỗ trợ kiểm thử playwright

Playwright là một công cụ automation test trình duyệt web mã nguồn mở, được phát triển bởi Microsoft Edge Team. Nó được tạo ra để đáp ứng nhu cầu thử nghiệm toàn diện. Cho phép bạn viết các kịch bản tự động hóa để thực hiện các thao tác trên các trình duyệt web như Chromium (bao gồm cả Google Chrome), Firefox và WebKit.



Hình 82: Logo playwright

Một số đặc điểm chính của Playwright:

- **Kiểm thử đa trình duyệt:** Tương tác với nhiều trình duyệt phổ biến như Chrome, Firefox và WebKit chỉ với một bộ mã nguồn, giúp bạn đảm bảo ứng dụng hoạt động ổn định trên mọi nền tảng.
- **API mạnh mẽ và linh hoạt:** Playwright cung cấp một API dễ sử dụng để tương tác với trình duyệt và các thành phần của trang web như nút, biểu mẫu, đối tượng DOM, v.v. API này được thiết kế để đáp ứng các nhu cầu tự động hóa phức tạp của các ứng dụng web hiện đại.
- **Hỗ trợ các tính năng mới:** Playwright hỗ trợ các tính năng hiện đại của trình duyệt như chụp ảnh toàn bộ trang web (full page screenshot), thao tác với file, xác thực người dùng, và nhiều hơn nữa.
- **Tích hợp với các framework kiểm thử:** Playwright có thể tích hợp dễ dàng với các framework kiểm thử như Jest, Mocha, hoặc cơ sở kiểm thử tự động khác thông qua các thư viện bổ sung.
- **Hỗ trợ nền tảng đa nền tảng:** Playwright hoạt động trên nhiều hệ điều hành như Windows, macOS và Linux, giúp bạn kiểm thử ứng dụng trên nhiều môi trường khác nhau.
- **Đơn giản dễ sử dụng:** Có thể bắt đầu cài đặt và sử dụng playwright đơn giản bằng cách sử dụng npm hoặc yarn. Và có thể bắt đầu viết testcase và test bằng VS Code

7.1.3 Kiểm thử hiệu suất với Developer tool

Chrome Developer Tools (DevTools) là một công cụ mạnh mẽ để đo và phân tích hiệu suất của website hoặc ứng dụng web.

Nhưng DevTools có rất nhiều thông số và với quy mô dự án hiện tại thì ta sẽ tập trung vào các thông số sau:

- **Network:** Tối ưu API và tải tài nguyên.
- **Timings:** Tập trung giảm thời gian hiển thị nội dung chính.
- **Frames:** Giảm lag khi cuộn trang hoặc tương tác.
- **Interactions:** Đảm bảo phản hồi người dùng nhanh chóng.
- **Animations:** Tối ưu giao diện động nếu có nhiều hiệu ứng.

7.2 Kiểm thử api

7.2.1 Cách thực hiện

Dùng Postman Tests để tiến hành manual testing với từng api sử dụng trong hệ thống. Viết test case kiểm thử api bằng cách sử dụng các mã Javascript thêm vào mỗi request để giúp verify kết quả trả về thành công hay thất bại, so sánh với kết quả mong đợi, kiểm tra status trả về... . Đoạn mã test case thường bắt đầu bằng pm.test và có thể coi nó như là các assert, verify command như ở các công cụ khác.

Các testcase sẽ được viết trong tab Test của postman, sau đây là testcase cho api Đăng nhập:

The screenshot shows the Postman interface with a test script for a login API. The URL is {{url}} /api/user/login. The script checks the response code and token presence based on the status code.

```
let statusCode = pm.response.code;
if (statusCode === 200) {
    pm.test("Status code is 200", function () {
        pm.response.to.have.status(200);
    });
    pm.test("Token is present and set as variable", function () {
        let jsonResponse = pm.response.json();
        pm.expect(jsonResponse).to.have.property('token');

        let token = jsonResponse.token;
        pm.collectionVariables.set("token", token);
        console.log("Token saved to collection variable:", token);
    });
} else if (statusCode === 401) {
    pm.test("Status code is 401", function () {
        pm.response.to.have.status(401);
    });
    pm.test("Token is not present", function () {
        let jsonResponse = pm.response.json();
        pm.expect(jsonResponse).to.not.have.property('token');
    });
} else {
    pm.test("Unexpected status code", function () {
        pm.expect([200, 401]).to.include(statusCode);
    });
}
```

Test Results (2/2)

Result	Message
PASSED	Status code is 200
PASSED	Token is present and set as variable

Hình 83: Viết test case đăng nhập trong postman

Đoạn code trên để kiểm tra mã trạng thái API và xử lý token khi đăng nhập thành công.

- **Xác định mã trạng thái (statusCode)** từ phản hồi API.
- **Mã 200 (OK):**

- Kiểm tra mã phản hồi là 200.
 - Xác nhận phản hồi có thuộc tính token và lưu vào biến collection.
- **Mã 401 (Unauthorized):**
 - Kiểm tra mã phản hồi là 401.
 - Xác nhận phản hồi không có thuộc tính token.
 - **Mã khác:** Đảm bảo mã phản hồi nằm trong [200, 401].

Mỗi lần thực hiện request thì các testcase cũng được thực hiện và cho ra kết quả test results như hình bên dưới.

The screenshot shows the Postman interface with the following details:

- HTTP:** Do an tot nghiep / voucher session / add voucher
- Method:** POST
- URL:** {{url}} /api/voucher-session
- Script (Post-response):**

```

1  let statusCode = pm.response.code;
2  if (statusCode === 200) {
3      pm.test("Status code is 200", function () {
4          pm.response.to.have.status(200);
5      });
6      pm.test("Validate the response schema", function () {
7          const responseData = pm.response.json();
8          pm.expect(responseData).to.be.an('object');
9          pm.expect(responseData.id).to.exist.and.to.be.a('number');
10         pm.expect(responseData.name).to.exist.and.to.be.a('string');
11         pm.expect(responseData.code).to.exist.and.to.be.a('string');
12         pm.expect(responseData.startDate).to.exist.and.to.be.a('string');
13         pm.expect(responseData.endDate).to.exist.and.to.be.a('string');
14         pm.expect(responseData.minLength).to.exist.and.to.be.a('number');
15         if (responseData.discountAmount !== null) {
16             pm.expect(responseData.discountAmount).to.be.a('number');
17         }
18         if (responseData.discountPercentage !== null) {
19             pm.expect(responseData.discountPercentage).to.be.a('number');
20         }
21         pm.expect(responseData.voucherType).to.exist.and.to.be.a('number');
22     });
23 } else if (statusCode === 409) {
24     pm.test("Status code is 409", function () {
25         pm.response.to.have.status(409);
26     });
27     pm.test("Message in the response body is not empty", function () {
28         const responseData = pm.response.json();
29         pm.expect(responseData.message).to.exist.and.to.not.be.empty;
30     });
31 }
32 pm.test("Response time is within an acceptable range", function () {

```

- Test Results (3/3):**

 - PASSED Status code is 200
 - PASSED Validate the response schema
 - PASSED Response time is within an acceptable range

Hình 84: Kết quả run testcase

7.2.2 Kết quả testcase tổng hợp

The screenshot shows a Postman test run summary. At the top, it says "Do an tot nghiep - Run results" and "Ran on Dec 01, 2024 at 06:08:55 · View all runs". Below this is a table with columns: Source (Runner), Environment (Testing environment), Iterations (1), Duration (23s 130ms), All tests (133), and Avg. Resp. Time (214 ms). A green button "Run Again" is at the top right. To the right of the table are buttons for "Automate Run", "+ New Run", and "Export Results". Below the table, it says "All Tests Passed (133) Failed (0) Skipped (0)". On the far right, there are "Generate Tests" and "View Summary" buttons. The main area shows a list of tests under "Iteration 1":

- POST dang_ky**: localhost:8080/api/user/register. Status: 200 OK, 597 ms, 691 B. Details: PASS Status code is 200, PASS Check user data.
- POST login**: localhost:8080/api/user/login. Status: 200 OK, 296 ms, 598 B. Details: PASS Status code is 200, PASS Token is present and set as variable.
- GET myInfo**: localhost:8080/api/user/myInfo. Status: 200 OK, 88 ms, 730 B. Details: PASS Status code is 200, PASS check data.
- PUT edit user**: localhost:8080/api/user/2. Status: 200 OK, 79 ms, 676 B. Details: PASS Status code is 200, PASS check data.

At the bottom of the interface are various icons: Postbot, Runner, Start Proxy, Cookies, Vault, Trash, and a help icon.

Hình 85: Kết quả testcase tổng hợp

Dựa trên hình ảnh, các thông số quan trọng của kết quả chạy test được tổng hợp như sau:

Thông số	Giá trị
Môi trường	Testing environment
Số lần lặp	1
Thời gian chạy	23 giây 130ms
Tổng số test	133
Số test thành công	133
Số test thất bại	0
Số test bị bỏ qua	0
Thời gian phản hồi TB	214ms

Bảng 88: Thông số quan trọng của kết quả chạy test

Nhận xét:

- Số lượng test:** Toàn bộ 133 test case đã được thực hiện, và tất cả đều THÀNH CÔNG (PASS), điều này cho thấy hệ thống hiện tại đang hoạt động ổn định trong phạm vi đã được kiểm tra.
- Thời gian thực thi:** Tổng thời gian chạy là 23 giây 130ms, trung bình mỗi test case mất khoảng 214ms, đây là thời gian phản hồi khá nhanh, phù hợp với các API yêu cầu xử lý hiệu quả.
- Không có lỗi:** Không có test case nào bị lỗi hoặc bị bỏ qua (Thất bại = 0, Bỏ qua = 0), điều này chứng tỏ hệ thống đã được kiểm thử kỹ lưỡng với kết quả đáng tin cậy.

7.2.3 Kịch bản kiểm thử và kết quả kiểm thử

	Nhóm api	Tên api	Testcase thực hiện	Kết quả mong muốn	Kết quả
1	User	Register	Kiểm tra status trả về	Status trả về là 200 (thành công)	pass
2			Kiểm tra các trường data trả về	Dữ liệu trả về đúng với body post request bao gồm các trường: - username - email - phoneNumber - firstName - lastName - birthDate - address. Và các trường có giá trị đúng với dữ liệu đầu vào.	pass
3			Kiểm tra id trả về	Có tạo được id user và kết quả thành công trả về id của user	pass
4			Kiểm tra đăng ký trùng pass	Status trả về 409 (đăng ký không thành công)	pass
5			Kiểm tra message trả về khi lỗi đăng ký	Trả về message "Error creating new user because reasons"	pass
6			Kiểm tra không khai báo password	Trả về message "Error creating new user because reasons" Status trả về 409	pass
7		Login	Kiểm tra status trả về	Status trả về là 200 (thành công)	pass
8			Kiểm tra token trả về	Có trả về token	pass
9			Kiểm tra sai username	Status trả về là 401 Không có trường token nào được trả về	pass
10	Genre	GetAllGenres	Kiểm tra status trả về	Status trả về là 200 (thành công khi đăng ký thành công)	pass
11			Kiểm tra dữ liệu trả về	Dữ liệu trả về gồm một list các object trong đó mỗi object gồm các trường id, name, nameVn	pass
12	Book	GetBookByTitle	Kiểm tra status trả về	Status trả về là 200 (thành công)	pass
13			Kiểm tra kết quả trả về với các từ khoá khác nhau (tìm kiếm thành công)	Dữ liệu trả về list gồm đầy đủ các trường như mong muốn gồm: - isbn - title - authors - languageCode - genre - publisher - publishedDate - pageCount - size	pass

14		Kiểm tra kết quả trả về với từ khoá đặc biệt @, . ,#,123,... (tim kiếm thành công)	Dữ liệu trả về list gồm đầy đủ các trường như mong muốn gồm: - isbn - title - authors - languageCode - genre - publisher - publishedDate - pageCount - size	pass
15		Kiểm tra kết quả trả về với từ khoá đặc biệt @, . ,#,123,... (không có dữ liệu trùng khớp)	Dữ liệu trả về là một list rỗng	pass
16		GetBookById	Kiểm tra status trả về	Status trả về là 200 (thành công)
17		GetBookById	Kiểm tra các trường data trả về	Dữ liệu trả về list gồm đầy đủ các trường như mong muốn gồm: - isbn - title - authors - languageCode - genre - publisher - publishedDate - pageCount - size
18		GetBookById	Kiểm tra id dữ liệu trả về	Trả về đúng id truyền vào (id có trong cơ sở dữ liệu)
19		CreateBookOk	Kiểm tra status trả về	Status trả về là 200 (thành công)
20		CreateBookOk	Kiểm tra các trường data trả về	Dữ liệu trả về list gồm đầy đủ các trường như mong muốn gồm: - isbn - title - authors - languageCode - genre - publisher - publishedDate - pageCount - size
21		CreateBookOk	Kiểm tra các trường null ko phải title và isbn	Status trả về là 200 (thành công) Dữ liệu trả về list gồm đầy đủ các trường như mong muốn gồm: - isbn - title - authors - languageCode - genre

				- publisher - publishedDate - pageCount - size	
22			Kiểm tra trả về trường id khi tạo book thành công	Dữ liệu trả về có trả về trường id khi đăng ký thành công	pass
23		GetAllBooks	Kiểm tra status trả về khi thành công	Status trả về là 200 (khi thành công)	pass
24			Kiểm tra dữ liệu trả về khi thành công	Dữ liệu trả về list gồm đầy đủ các trường như mong muốn gồm: - id - isbn - title - authors - languageCode - genre - publisher - publishedDate - pageCount - size	pass
25		GetListingDetails	Kiểm tra status trả về	Status trả về là 200 (khi thành công)	pass
26			Kiểm tra dữ liệu listing trả về	Trả về đầy đủ và đúng các trường mong muốn gồm: - id - user - address - leaseRate - depositFee - penaltyRate - description - listingStatus - copy - book - review - bookOwned - bookLeasing	pass
27			Kiểm tra status nếu không tìm thấy dữ liệu	Trả về status 204 (No Content)	pass
28		SearchListings	Kiểm tra status trả về	Status trả về là 200 (thành công)	pass
29			Kiểm tra dữ liệu khi không param size phân trang	Status trả về là 200 (thành công) size của một trang bằng 20 hiển thị trang 0 là trang đầu tiên "Trả về đầy đủ và đúng các trường mong muốn gồm: - id - user - address - leaseRate	pass

			<ul style="list-style-type: none"> - depositFee - penaltyRate - description - listingStatus - copy - book - review - bookOwned - bookLeasing 	
30		Kiểm tra dữ liệu khi có param size phân trang	<p>Status trả về là 200 (thành công)</p> <p>size của một trang bằng size param truyền vào</p> <p>hiển thị trang 0 là trang đầu tiên</p> <p>"Trả về đầy đủ và đúng các trường mong muốn gồm:</p> <ul style="list-style-type: none"> - id - user - address - leaseRate - depositFee - penaltyRate - description - listingStatus - copy - book - review - bookOwned - bookLeasing 	pass
31		Kiểm tra dữ liệu khi có param page phân trang không có param page phân trang	<p>Status trả về là 200 (thành công)</p> <p>size của một trang bằng size param truyền vào</p> <p>trang hiển thị là trang của param truyền vào</p> <p>"Trả về đầy đủ và đúng các trường mong muốn trong content gồm:</p> <ul style="list-style-type: none"> - id - user - address - leaseRate - depositFee - penaltyRate - description - listingStatus - copy - book - review - bookOwned - bookLeasing 	pass
32		Kiểm tra dữ liệu khi không có param page phân trang	<p>Status trả về là 200 (thành công)</p> <p>size của một trang bằng 20</p> <p>hiển thị trang 0 là trang đầu tiên</p>	pass

				"Trả về đầy đủ và đúng các trường mong muốn trong content gồm: - id - user - address - leaseRate - depositFee - penaltyRate - description - listingStatus - copy - book - review - bookOwned - bookLeasing	
33		Kiểm tra tìm dữ liệu với param title page size và page number		Status trả về là 200 (thành công) size của một trang bằng 20 hiển thị trang 0 là trang đầu tiên Trả về đầy đủ và đúng các trường mong muốn trong content gồm: - id - user - address - leaseRate - depositFee - penaltyRate - description - listingStatus - copy - book - review - bookOwned - bookLeasing	pass
34		Kiểm tra tìm dữ liệu với param genre		Status trả về là 200 (thành công) size của một trang bằng 20 hiển thị trang 0 là trang đầu tiên Trả về đầy đủ và đúng các trường mong muốn trong content gồm: - id - user - address - leaseRate - depositFee - penaltyRate - description - listingStatus - copy - book - review - bookOwned - bookLeasing	pass

35		Kiểm tra tìm dữ liệu khi không có param title và genre	Status trả về là 200 (thành công) size của một trang bằng 20 hiển thị trang 0 là trang đầu tiên Trả về đầy đủ và đúng các trường mong muốn trong content gồm: - id - user - address - leaseRate - depositFee - penaltyRate - description - listingStatus - copy - book - review - bookOwned - bookLeasing	pass
36		Kiểm tra tìm dữ liệu khi có cả hai param title và genre	Status trả về là 200 (thành công) size của một trang bằng 20 hiển thị trang 0 là trang đầu tiên Trả về đầy đủ và đúng các trường mong muốn trong content gồm: - id - user - address - leaseRate - depositFee - penaltyRate - description - listingStatus - copy - book - review - bookOwned - bookLeasing	pass
37		Kiểm tra dữ liệu khi không tìm thấy	Status trả về là 200 (thành công) size của một trang bằng 20 hiển thị trang 0 là trang đầu tiên content là một list rỗng	pass
38	GetListin gById	Kiểm tra status trả về	Status trả về là 200 (thành công)	pass
39		Kiểm tra dữ liệu khi request title bởi nhiều từ khoá: @, . ,#,123,... (tim kiem thành công)	Status trả về là 200 (thành công) size của một trang bằng 20 hiển thị trang 0 là trang đầu tiên Trả về đầy đủ và đúng các trường mong muốn trong content gồm: - id - user - address - leaseRate - depositFee	pass

				- penaltyRate - description - listingStatus - copy - book - review - bookOwned - bookLeasing	
40			Kiểm tra dữ liệu khi không tìm thấy	Status trả về là 200 (thành công) size của một trang bằng 20 hiển thị trang 0 là trang đầu tiên content là một list rỗng	pass
41		CreateListing	Kiểm tra status trả về khi tạo thành công	Status trả về là 200 (thành công)	pass
42			Kiểm tra data trả về khi tạo thành công	Trả về đầy đủ các trường mong muốn gồm: - ownerId - address - leaseRate - depositFee - penaltyRate - description - listingStatus Dữ liệu trả về giống với dữ liệu từ body request	pass
43			Kiểm tra id trả khi tạo thành công	Có trả về trường id khi tạo thành công	pass
44		DeleteListing	Kiểm tra status trả về khi xoá thành công	Status trả về là 204 (thành công)	pass
45			Kiểm tra dữ liệu	Trả về đúng đủ các trường như mong muốn	pass
46			Kiểm tra owner id trả về	Trả về đúng id được truyền vào	pass
47		CreateCopy	Kiểm tra status trả về khi tạo thành công	Status trả về là 200 (thành công)	pass
48		Copy	Kiểm tra dữ liệu khi tạo thành công	Trả về đầy đủ các trường mong muốn gồm: - id - score - description - leaseOrderId - userId - listingId - createdDate Dữ liệu trả về giống với dữ liệu truyền từ body request	pass
49		GetCopyById	Kiểm tra status trả về khi thành công	Status trả về là 200 (thành công)	pass

50			Kiểm tra dữ liệu khi tạo thành công	Trả về đầy đủ các trường mong muốn gồm: - id - score - description - leaseOrderId - userId - listingId - createdDate Dữ liệu trả về giống với dữ liệu truyền từ body request	pass
51		GetCopiesByOwnerId	Kiểm tra status trả về khi thành công	Status trả về là 200 (thành công)	pass
52		GetCopiesByOwnerId	Kiểm tra dữ liệu khi tạo thành công	Trả về đầy đủ các trường mong muốn gồm: - id - score - description - leaseOrderId - userId - listingId - createdDate Dữ liệu trả về giống với dữ liệu truyền từ body request	pass
53		CreateReview	Kiểm tra status trả về khi tạo thành công	Status trả về là 200 (thành công)	pass
54		CreateReview	Kiểm tra id trả khi tạo thành công	Có trả về trường id khi tạo thành công	pass
55		Review	Kiểm tra dữ liệu khi tạo thành công	Trả về đầy đủ các trường mong muốn gồm: - id - score - description - leaseOrderId - userId - listingId - createdDate Dữ liệu trả về giống với dữ liệu truyền từ body request	pass
56		CreateNewLeaseOrder	Kiểm tra status trả về khi tạo thành công	Status trả về là 200 (thành công)	pass
57		CreateNewLeaseOrder	Kiểm tra id trả khi tạo thành công	Có trả về trường id khi tạo thành công	pass
58		LeaseOrder	Kiểm tra dữ liệu trả ra khi thành công	Trả về các trường sau đúng với dữ liệu body request truyền vào: - listingId - status - lesseeId - lesseeAddress - fromDate	pass

			<ul style="list-style-type: none"> - toDate - paymentMethod 	
59		Kiểm tra tạo thành công lease order details	<p>Trả về có leaseOrderdetail với các trường mong muốn đó là :</p> <ul style="list-style-type: none"> - id - title - listingId - leaseRate - depositFee - penaltyRate 	pass
60		Kiểm tra tạo thành công lease order details với các loại method payment (COD, VNPay, BANK_TRANSFER),	<p>Trả về có leaseOrderdetail với các trường mong muốn như tạo thành như tạo thành công. Và trả về method payment đúng với method payment truyền vào từ request body</p>	pass
61		Kiểm tra tạo thành công lease order details với ngày tạo là ngày hiện tại	<p>Trả về có leaseOrderdetail với các trường mong muốn như tạo thành như tạo thành công. Và trả về createdDate đúng với ngày hiện tại</p>	pass
62		Kiểm tra tạo thành công lease order details, tạo được leaseAndDepositPaymentId	<p>Trả về có leaseOrderdetail với các trường mong muốn như tạo thành như tạo thành công. Trả về có giá trị của trường leaseAndDepositPaymentId</p>	pass
63		Kiểm tra tạo thành công lease order details, kiểm tra tổng tiền phạt ban đầu	<p>Trả về có leaseOrderdetail với các trường mong muốn như tạo thành như tạo thành công. Trả về giá trị của trường totalPenaltyRate là 0.</p>	pass
64		Kiểm tra tạo thành công lease order details, kiểm tra tổng phí thuê	<p>Trả về có leaseOrderdetail với các trường mong muốn như tạo thành như tạo thành công. Trả về giá trị của trường totalLeaseFee tính đúng business logic</p>	pass
65		Kiểm tra tạo thành công lease order details, kiểm tra tiền cọc	<p>Trả về có leaseOrderdetail với các trường mong muốn như tạo thành như tạo thành công. Trả về giá trị của trường totalDeposit đúng với trường depositFee trong leaseOrderDetails</p>	pass
66	GetLeaseOrdersByLessorId	Kiểm tra status trả về khi thành công	<p>Status trả về là 200 (thành công)</p>	pass
67		Kiểm tra dữ liệu trả ra khi thành công	<p>Trả về các trường sau trong mỗi phần tử của list:</p> <ul style="list-style-type: none"> - leaseOrder - listing - lessor 	pass

68		Kiểm tra lessor id trả về khi tìm kiếm thành công có giá trị	Phản tử thứ nhất trong dãy trả về có giá trị lessor id bằng với lessor id truyền vào	pass
69		Kiểm tra khi tìm kiếm thành công không có giá trị	Trả về list rỗng	pass
70	GetLeaseOrdersByLesseeId	Kiểm tra status trả về khi thành công	Status trả về là 200 (thành công)	pass
71		Kiểm tra dữ liệu trả ra khi thành công có giá trị	Trả về các trường sau trong mỗi phản tử của list: - leaseOrder - listing - lessor	pass
72		Kiểm tra lesse id trả về khi tìm kiếm thành công có giá trị	Phản tử thứ nhất trong dãy trả về có giá trị lesse id bằng với lesse id truyền vào	pass
73		Kiểm tra khi tìm kiếm thành công không có giá trị	Trả về list rỗng	pass
74	GetLeaseOrdersByLessorIdAndStatus	Kiểm tra status trả về khi thành công	Status trả về là 200 (thành công)	pass
75		Kiểm tra dữ liệu trả ra khi thành công	Trả về các trường sau trong mỗi phản tử của list: - leaseOrder - listing - lessor	pass
76		Kiểm tra lessor id trả về khi tìm kiếm thành công có giá trị	Phản tử thứ nhất trong dãy trả về có giá trị lessor id bằng với lessor id truyền vào	pass
77		Kiểm tra khi tìm kiếm thành công không có giá trị	Trả về list rỗng	pass
78		Kiểm tra với từng status lựa chọn	Trả về đúng status theo business logic	pass
79	GetLeaseOrdersByLesseeIdAndStatus	Kiểm tra status trả về khi thành công	Status trả về là 200 (thành công)	pass
80		Kiểm tra dữ liệu trả ra khi thành công có giá trị	Trả về các trường sau trong mỗi phản tử của list: - leaseOrder - listing - lessor	pass
81		Kiểm tra lesse id trả về khi tìm kiếm thành công có giá trị	Phản tử thứ nhất trong dãy trả về có giá trị lesse id bằng với lesse id truyền vào	pass
82		Kiểm tra khi tìm kiếm thành công không có giá trị	Trả về list rỗng	pass

83		Kiểm tra với từng status lựa chọn	Trả về đúng status theo business logic	pass
84		GetLeaseOrderById	Kiểm tra status trả về khi thành công	Status trả về là 200 (thành công)
85			Kiểm tra dữ liệu trả ra khi thành công có giá trị	Trả về các trường sau trong mỗi phần tử của list: - leaseOrder - listing - lessor
86			Kiểm tra trường id trả về khi thành công	Trả về id giống với id truyền vào khi thành công
87		UpdateLeaseOrderStatus	Kiểm tra status trả về khi thành công	Status trả về là 200 (thành công)
88			Kiểm tra status trả về khi không cho phép đổi status	Status trả về là 423 message: Error update status
89			Kiểm tra status trả về khi thành công thử với các status: RETURNED, RETURNING, DELIVERED, USER_PAID	Status trả giống với status truyền vào trong param khi thành công
90	Voucher Session	GetVoucherById	Kiểm tra status trả về khi thành công	Status trả về là 200 (thành công)
91			Kiểm tra thời gian phản hồi	Thời gian phản hồi nhỏ hơn 500ms
92			Kiểm tra Content-Type header	Content-Type header là application/json
93			Kiểm tra các trường dữ liệu bắt buộc	Dữ liệu trả về đầy đủ các trường mong muốn
94			Kiểm tra giá trị hợp lệ của VoucherType	VoucherType là giá trị hợp lệ
95	DeleteVoucher		Kiểm tra status trả về khi thành công	Status trả về là 204 (không có nội dung)
96			Kiểm tra thời gian phản hồi	Thời gian phản hồi nhỏ hơn 500ms
97			Kiểm tra body phản hồi	Body phản hồi là null
98	AddVoucher		Kiểm tra status trả về khi thành công	Status trả về là 200 (thành công)
99			Kiểm tra schema của phản hồi	Phản hồi phù hợp với schema mong muốn
100			Kiểm tra thời gian phản hồi	Thời gian phản hồi nằm trong phạm vi chấp nhận được
101		UpdateVoucher	Kiểm tra status trả về khi thành công	Status trả về là 200 (thành công)

10 2	GetAllVouchers	Kiểm tra thời gian phản hồi	Thời gian phản hồi nhỏ hơn 200ms	Pass
10 3		Kiểm tra Content-Type header	Content-Type header là application/json	Pass
10 4		Kiểm tra schema của phản hồi	Phản hồi chứa đầy đủ các thuộc tính yêu cầu	Pass
10 5		Kiểm tra status trả về khi thành công	Status trả về là 200 (thành công)	Pass
10 6		Kiểm tra thời gian phản hồi	Thời gian phản hồi nhỏ hơn 500ms	Pass
10 7		Kiểm tra các trường dữ liệu bắt buộc	Dữ liệu trả về đầy đủ các trường mong muốn	Pass
10 8		Kiểm tra giá trị của trường Id	Id là số nguyên không âm	Pass
10 9		GetShop VoucherById	Kiểm tra status trả về khi thành công	Pass
11 0	DeleteShopVoucher	Kiểm tra thời gian phản hồi	Thời gian phản hồi nhỏ hơn 500ms	Pass
11 1		Kiểm tra Content-Type header	Content-Type header là application/json	Pass
11 2		Kiểm tra các trường dữ liệu bắt buộc	Dữ liệu trả về đầy đủ các trường mong muốn	Pass
11 3		Kiểm tra giá trị hợp lệ của VoucherType	VoucherType là giá trị hợp lệ	Pass
11 4		Kiểm tra status trả về khi thành công	Status trả về là 204 (không có nội dung)	Pass
11 5	VoucherShop	Kiểm tra thời gian phản hồi	Thời gian phản hồi nhỏ hơn 500ms	Pass
11 6		Kiểm tra body phản hồi	Body phản hồi là null	Pass
11 7		AddShopVoucher	Kiểm tra status trả về khi thành công	Pass
11 8	UpdateShopVoucher	Kiểm tra schema của phản hồi	Phản hồi phù hợp với schema mong muốn	Pass
11 9		Kiểm tra thời gian phản hồi	Thời gian phản hồi nằm trong phạm vi chấp nhận được	Pass
12 0		Kiểm tra status trả về khi thành công	Status trả về là 200 (thành công)	Pass
12 1		Kiểm tra thời gian phản hồi	Thời gian phản hồi nhỏ hơn 200ms	Pass
12 2		Kiểm tra Content-Type header	Content-Type header là application/json	Pass
12 3		Kiểm tra schema của phản hồi	Phản hồi chứa đầy đủ các thuộc tính yêu cầu	Pass
12 4		Kiểm tra status trả về khi thành công	Status trả về là 200 (thành công)	Pass

12 5	GetAllShopVouchers	Kiểm tra thời gian phản hồi	Thời gian phản hồi nhỏ hơn 500ms	Pass
12 6		Kiểm tra các trường dữ liệu bắt buộc	Dữ liệu trả về đầy đủ các trường mong muốn	Pass
12 7		Kiểm tra giá trị của trường Id	Id là số nguyên không âm	Pass
12 8	Sales Order	GetSales OrderById	Kiểm tra status trả về khi thành công	Status trả về là 200 (thành công)
12 9			Kiểm tra thời gian phản hồi	Thời gian phản hồi nhỏ hơn 200ms
13 0			Kiểm tra schema của phản hồi	Phản hồi chứa đầy đủ các trường bắt buộc
13 1		GetSales OrdersBySeller	Kiểm tra status trả về khi thành công	Status trả về là 200 (thành công)
13 2			Kiểm tra thời gian phản hồi	Thời gian phản hồi nhỏ hơn 500ms
13 3			Kiểm tra Content-Type header	Content-Type header là application/json
13 4			Kiểm tra các trường dữ liệu bắt buộc	Dữ liệu trả về đầy đủ các trường mong muốn

Bảng 89: Kích bản kiểm thử và kết quả kiểm thử

Nhóm API	Pass	Fail	Total
User	10	0	10
Genre	7	0	7
Book	10	0	10
Listing	22	0	22
Copy	5	0	5
Review	3	0	3
Lease Order	22	0	22
Sale Order	8	0	8
Voucher Session	23	0	23
Voucher Shop	23	0	23
Tổng cộng	133	0	133

Bảng 90: Kết quả của kiểm thử tổng hợp theo tên api unit test

7.3 Kiểm thử tổng hợp

7.3.1 Cách thực hiện

Trước khi thực hiện test ta cần cài Playwright bằng cách sử dụng npm hoặc yarn. Ở đây chúng ta sẽ sử dụng npm để chạy lệnh sau. Lệnh này sẽ cài đặt phiên bản mới nhất cho playwright vào thư mục hiện hành.

npm init playwright@latest

Sau đó thêm các thiết lập cho init source code như sau:

thao@thaos-MacBook-Pro test_fe % npm init playwright@latest

Need to install the following packages:

create-playwright@1.17.132

Ok to proceed? (y) y

Getting started with writing end-to-end tests with Playwright:

Initializing project in '.'

- ✓ Do you want to use TypeScript or JavaScript? · JavaScript
- ✓ Where to put your end-to-end tests? · tests
- ✓ Add a GitHub Actions workflow? (y/N) · true
- ✓ Install Playwright browsers (can be done manually via 'npx playwright install')? (Y/n) · true

Initializing NPM project (npm init -y)...

Sau khi có source code thì tiến hành viết các testcase vào trong thư mục tests

7.3.2 Kịch bản kiểm thử

Tên hệ thống		The Flying Bookstore		45 cases
Link hệ thống		https://the-flying-bookstore.vercel.app		
Thông tin tài khoản		dev15/123456		
Mã TC	Tên test case	Tiền điều kiện	Các bước thực hiện	Kết quả mong muốn
Login (2 cases)				
Login	Kiểm tra login thành công	Có account dev15/123456 trong database	1. Vào link website 2. Nhấn nút đăng nhập 3. Nhập thông tin username, password 4. Nhấn nút đăng nhập	Đăng nhập thành công
Logout	Kiểm tra logout thành công	Đăng nhập thành công acc dev15/123456	1. Nhấn icon account 2. Nhấn icon Đăng xuất	Đăng xuất thành công
Giỏ hàng (4 cases)				
GH_1	Kiểm tra page giỏ hàng	Đăng nhập thành công acc dev15/123456	1. Truy cập vào link website với trạng thái đăng nhập 2. Click vào icon chúc năng giỏ hàng trên thanh điều hướng	Điều hướng thành công sang trang giỏ hàng. Trang giỏ hàng chứa các nội dung: header: Giỏ hàng của bạn tab: sách thuê, sách mua button: quay lại trang chủ Hiện giỏ hàng đang trống

GH_2	Thêm sản phẩm thuê vào giỏ hàng	Đăng nhập thành công acc dev15/123456	<ol style="list-style-type: none"> Truy cập vào link website với trạng thái đăng nhập Nhập tên sản phẩm vào input tìm kiếm Click icon search Click vào bài đăng tìm được Thêm ngày bắt đầu, ngày kết thúc thuê sách Click nút "Đặt thuê ngay" 	<p>Sản phẩm được thêm vào giỏ hàng tab Sách thuê thành công.</p> <p>Hiển thị thông tin sản phẩm bao gồm: người cho thuê, thời gian thuê, địa chỉ, số điện thoại, số ngày thuê, tiền thuê, tiền cọc, tổng cộng, tổng tiền thanh toán.</p> <p>Nút thanh toán có thể click</p>
GH_3	Thêm sản phẩm mua vào giỏ hàng	Đăng nhập thành công acc dev15/123456	<ol style="list-style-type: none"> Truy cập vào link website với trạng thái đăng nhập Nhập tên sản phẩm vào input tìm kiếm Click icon search Click vào bài đăng tìm được Click nút "Mua ngay" 	<p>Sản phẩm được thêm vào giỏ hàng sách mua thành công.</p> <p>Hiển thị thông tin sản phẩm bao gồm: người bán, địa chỉ, số điện thoại, giá gốc, giá giảm trực tiếp, khuyến mãi từ người bán, tổng tiền thanh toán.</p> <p>Nút "Đặt mua hàng" có thể click</p>
GH_4	Kiểm tra tab voucher sản phẩm mua	Đăng nhập thành công acc dev15/123456	<ol style="list-style-type: none"> Truy cập vào link website với trạng thái đăng nhập Nhập tên sản phẩm vào input tìm kiếm Click icon search Click vào bài đăng tìm được Click nút "Mua ngay" 	<p>Hiển thị thành công tab voucher của shop và voucher toàn sàn trong giỏ hàng.</p> <p>Nút "bỏ chọn", "Chọn hoặc nhập mã khác" có thể click</p>
Trang checkout (8 cases)				
CO_1	Kiểm tra thông tin sách trang checkout	Đăng nhập thành công acc dev15/123456	<ol style="list-style-type: none"> Truy cập vào link website với trạng thái đăng nhập Nhập tên sản phẩm vào input tìm kiếm Click icon search Click vào bài đăng tìm được Thêm ngày bắt đầu, ngày kết thúc thuê sách Click nút "Đặt thuê ngay" 	<p>Hiển thị thành công sách thuê bao gồm: hình ảnh, tên sách, số lượng, giá thuê, phí cọc.</p>

			7. Click nút "Đặt thuê hàng"	
CO_2	Kiểm tổng quan trang checkout	Đăng nhập thành công acc dev15/123456	1. Truy cập vào link website với trạng thái đăng nhập 2. Nhập tên sản phẩm vào input tìm kiếm 3. Click icon search 4. Click vào bài đăng tìm được 5. Thêm ngày bắt đầu, ngày kết thúc thuê sách 6. Click nút "Đặt thuê ngay" 7. Click nút "Đặt thuê hàng"	Hiển thị thành công thông tin đặt đơn thuê bao gồm các thành phần: Thông tin sản phẩm, thông tin đặt hàng, Thông tin đặt thuê, thanh toán. Hiển thị thành công quy trình thuê sách. Các nút có thể click: quay lại, tạo đơn hàng.
CO_3	Tạo đơn hàng thuê	Đăng nhập thành công acc dev15/123456	1. Truy cập vào link website với trạng thái đăng nhập 2. Nhập tên sản phẩm vào input tìm kiếm 3. Click icon search 4. Click vào bài đăng tìm được 5. Thêm ngày bắt đầu, ngày kết thúc thuê sách 6. Click nút "Đặt thuê ngay" 7. Click nút "Đặt thuê hàng" 8. Click nút "Tạo đơn hàng"	Tạo thành công đơn hàng. Hiển thị thông tin đơn hàng vừa tạo gồm các nội dung: bài đăng, số lượng, giá thuê, cọc, thông tin đặt hàng, mã đơn hàng, ngày đặt thuê, người cho thuê, thời gian thuê, địa chỉ người cho thuê, số điện thoại người cho thuê, trạng thái, phương thức thanh toán, số ngày thuê, tiền thuê, tiền cọc. Các nút có thể click: quản lý đơn hàng, tiếp tục mua sắm.
CO_4	Kiểm tra thông tin đặt đơn thuê trang checkout	Đăng nhập thành công acc dev15/123456	1. Truy cập vào link website với trạng thái đăng nhập 2. Nhập tên sản phẩm vào input tìm kiếm 3. Click icon search 4. Click vào bài đăng tìm được 5. Thêm ngày bắt đầu, ngày kết thúc thuê sách 6. Click nút "Đặt thuê ngay" 7. Click nút "Đặt thuê hàng"	Hiển thành công thông tin đặt thuê bao gồm: người cho thuê, thời gian thuê, địa chỉ, số điện thoại, số ngày thuê, tiền thuê, tiền cọc, tổng cộng, tổng tiền thanh toán.

CO_5	Kiểm tra thông tin đặt đơn mua trang checkout	Đăng nhập thành công acc dev15/123456	1. Truy cập vào link website với trạng thái đăng nhập 2. Nhập tên sản phẩm vào input tìm kiếm 3. Click icon search 4. Click vào bài đăng tìm được 5. Click nút "Mua ngay" 6. Click nút "Đặt mua hàng"	Hiển thành công thông tin đặt mua bao gồm: người cho thuê, thời gian thuê, địa chỉ, số điện thoại, số ngày thuê, tiền thuê, tiền cọc, tổng cộng, tổng tiền thanh toán.
CO_6	Kiểm tra thông tin payment trang checkout	Đăng nhập thành công acc dev15/123456	1. Truy cập vào link website với trạng thái đăng nhập 2. Nhập tên sản phẩm vào input tìm kiếm 3. Click icon search 4. Click vào bài đăng tìm được 5. Thêm ngày bắt đầu, ngày kết thúc thuê sách 6. Click nút "Đặt thuê ngay" 7. Click nút "Đặt thuê hàng"	Hiển thành công thông tin payment.
CO_7	Kiểm tra thông tin voucher đơn mua trang checkout	Đăng nhập thành công acc dev15/123456	1. Truy cập vào link website với trạng thái đăng nhập 2. Nhập tên sản phẩm vào input tìm kiếm 3. Click icon search 4. Click vào bài đăng tìm được 5. Thêm ngày bắt đầu, ngày kết thúc thuê sách 6. Click nút "Đặt thuê ngay" 7. Click nút "Đặt thuê hàng"	Hiển thành công thông tin voucher đơn mua gồm voucher shop và voucher toàn sàn.
CO_8	Kiểm tra thông tin user trang checkout	Đăng nhập thành công acc dev15/123456	1. Truy cập vào link website với trạng thái đăng nhập 2. Nhập tên sản phẩm vào input tìm kiếm 3. Click icon search 4. Click vào bài đăng tìm được 5. Thêm ngày bắt đầu,	Hiển thành công thông tin user gồm: họ và tên lót, email, ngày sinh, số điện thoại, địa chỉ. Nút có thể click: Xác nhận

			ngày kết thúc thuê sách 6. Click nút "Đặt thuê ngay" 7. Click nút "Đặt thuê hàng"	
Chi tiết bài đăng (7 cases)				
CTBĐ_1	Kiểm tra thông tin copy trang chi tiết bài đăng	Đăng nhập thành công acc dev15/123456	1. Truy cập vào link website với trạng thái đăng nhập 2. Nhập tên sản phẩm vào input tìm kiếm 3. Click icon search 4. Click vào bài đăng tìm được	Hiển thị thành công thông tin copy bao gồm: ảnh cuốn sách, tiêu đề sách, tác giả, nhà xuất bản, ngày xuất bản. Nút xem trước sách có thể click.
CTBĐ_2	Kiểm tra thông tin listing trang chi tiết bài đăng	Đăng nhập thành công acc dev15/123456	1. Truy cập vào link website với trạng thái đăng nhập 2. Nhập tên sản phẩm vào input tìm kiếm 3. Click icon search 4. Click vào bài đăng tìm được	Hiển thị thành công thông tin listing bao gồm: ngày xuất bản, kích thước, isbn, số trang, tiền cọc, phần trăm hu hại, địa chỉ cho thuê, phí phạt trả trước, giá gốc, giá bán, địa chỉ mua sách.
CTBĐ_3	Kiểm tra xem trước vài trang cuốn sách	Đăng nhập thành công acc dev15/123456	1. Truy cập vào link website với trạng thái đăng nhập 2. Nhập tên sản phẩm vào input tìm kiếm 3. Click icon search 4. Click vào bài đăng tìm được 5. Click nút "Xem trước sách"	Hiển thị thành công popup xem trước sách. Các nút hiển thị thành công và có thể click: return back, zoom-in, zoom-out
CTBĐ_4	Kiểm tra thông tin chủ sách trang chi tiết bài đăng	Đăng nhập thành công acc dev15/123456	1. Truy cập vào link website với trạng thái đăng nhập 2. Nhập tên sản phẩm vào input tìm kiếm 3. Click icon search 4. Click vào bài đăng tìm được 5. Click nút "Xem trước sách"	Hiển thị thành công thông tin chủ sách bao gồm: số sách hiện có, số sách đang cho thuê
CTBĐ_5	Kiểm tra Tab thông tin đặt thuê trang chi tiết bài đăng	Đăng nhập thành công acc dev15/123456	1. Truy cập vào link website với trạng thái đăng nhập 2. Nhập tên sản phẩm vào input tìm kiếm 3. Click icon search	Hiển thị thông tin đặt thuê bao gồm: Nhập ngày nhận ngày trả, thông tin giá thuê, số ngày thuê, tổng tiền thuê, tiền cọc, tổng

			4. Click vào bài đăng tìm được 5. Click nút "Xem trước sách"	tiền. Các nút hiển thị và có thể click: đặt thuê ngay, calendar chọn ngày
CTBD_6	Kiểm tra Tab thông tin đánh giá sản phẩm trang chi tiết bài đăng	Đăng nhập thành công acc dev15/123456	1. Truy cập vào link website với trạng thái đăng nhập 2. Nhập tên sản phẩm vào input tìm kiếm 3. Click icon search 4. Click vào bài đăng tìm được 5. Click nút "Xem trước sách" 6. Click tab "Đánh giá"	Hiển thị thông tin đánh giá sản phẩm bao gồm: icon review, các tab chọn xem tất cả, 5 sao đến 1 sao. Các tab từ 1-5 sao có thể click.
CTBD_7	Kiểm tra Tab mua sách trang chi tiết bài đăng	Đăng nhập thành công acc dev15/123456	1. Truy cập vào link website với trạng thái đăng nhập 2. Nhập tên sản phẩm vào input tìm kiếm 3. Click icon search 4. Click vào bài đăng tìm được 5. Click nút "Xem trước sách" 6. Click tab "Đánh giá"	Hiển thị thông tin đặt mua bao gồm: giá mua. Nút mua sách hiển thị thành công và có thể click.

Trang chủ (5 cases)

TC_1	Kiểm tra banner trên trang chủ	Đăng nhập thành công acc dev15/123456	1. Truy cập vào trang chủ dưới link website với trạng thái đăng nhập	Hiển thị thành công thông tin banner trên trang chủ bao gồm: Hình ảnh, thông tin tiếp thị, nút thuê ngay và nút xem chi tiết
TC_2	Kiểm tra carousel trên trang chủ	Đăng nhập thành công acc dev15/123456	1. Truy cập vào trang chủ dưới link website với trạng thái đăng nhập 2. Scroll đến carousel	Hiển thị thành công carousel trên trang chủ
TC_3	Kiểm tra footer trên trang chủ	Đăng nhập thành công acc dev15/123456	1. Truy cập vào trang chủ dưới link website với trạng thái đăng nhập 2. Scroll đến footer	Hiển thị thành công footer trên trang chủ và thông tin của footer bao gồm: logo, thông tin facebook, youtube, insta, map, thông tin liên hệ

TC_4	Kiểm tra header trên trang chủ	Đăng nhập thành công acc dev15/123456	1. Truy cập vào trang chủ dưới link website với trạng thái đăng nhập	Hiển thị thành công header trên trang chủ và thông tin của header bao gồm: button logo trang web, thanh tìm kiếm, các icon giỏ hàng, user, quản lý
TC_5	Kiểm tra statistic-list trên trang chủ	Đăng nhập thành công acc dev15/123456	1. Truy cập vào trang chủ dưới link website với trạng thái đăng nhập 2. Scroll đến statistic-list	Hiển thị thành công statistic-list trên trang chủ và thông tin của statistic-list.

Quản lý (13 cases)

QLCT_1	Kiểm tra trang quản lý đơn cho thuê	Đăng nhập thành công acc dev15/123456	1. Truy cập vào trang chủ dưới link website với trạng thái đăng nhập 2. Click icon quản lý 3. Click quản lý đơn cho thuê	Hiển thị thành công trang đơn hàng cho thuê. Hiển thị thành công các tab trong đơn hàng cho thuê bao gồm: Tất cả, đã đặt hàng, đã nhận, đã quá hạn, đã trả sách, đã huỷ. Nút tải lại có thể click
QLCT_2	Kiểm tra trang quản lý đơn thuê	Đăng nhập thành công acc dev15/123456	1. Truy cập vào trang chủ dưới link website với trạng thái đăng nhập 2. Click icon quản lý 3. Click quản lý đơn thuê	Hiển thị thành công trang đơn hàng thuê. Hiển thị thành công các tab trong đơn hàng cho thuê bao gồm: Tất cả, đã đặt hàng, đã nhận, đã quá hạn, đã trả sách, đã huỷ. Nút tải lại có thể click
QLBD_1	Kiểm tra trang quản lý bài đăng cho cả đơn thuê và đơn bán	Đăng nhập thành công acc dev15/123456	1. Truy cập vào trang chủ dưới link website với trạng thái đăng nhập 2. Click icon quản lý bài đăng	Hiển thị thành công trang quản lý bài đăng. Hiển thị thành công các tab trong quản lý bài đăng: Bài đăng bán và thuê, bài đăng chỉ thuê, bài đăng chỉ bán. Các nút có thể click: Thêm bài đăng
QLBD_2	Kiểm tra trang quản lý bài đăng cho	Đăng nhập thành công acc dev15/123456	1. Truy cập vào trang chủ dưới link website với trạng thái đăng nhập	Hiển thị thành công trang quản lý bài đăng tab "bài đăng chỉ bán"

	đơn chỉ mua bán		nhập 2. Click icon quản bài đăng 3. Click tab " bài đăng chỉ bán"	Hiển thị thành công danh sách bài đăng bao gồm các thông tin trong danh sách: id, tên bài đăng, tác giả, giá gốc, giá bán
QLBD_3	Kiểm tra trang quản lý bài đăng cho đơn chỉ thuê	Đăng nhập thành công acc dev15/123456	1. Truy cập vào trang chủ dưới link website với trạng thái đăng nhập 2. Click icon quản bài đăng 3. Click tab " bài đăng chỉ thuê"	Hiển thị thành công trang quản lý bài đăng tab "bài đăng chỉ thuê" Hiển thị thành công danh sách bài đăng bao gồm các thông tin trong danh sách: id, tên bài đăng, tác giả, giá thuê, tiền cọc, tiền phạt
QLDMB_1	Kiểm tra trang quản lý đơn bán	Đăng nhập thành công acc dev15/123456	1. Truy cập vào trang chủ dưới link website với trạng thái đăng nhập 2. Click icon quản lý 3. Click item "Đơn bán"	Hiển thị thành công trang quản lý đơn bán Hiển thị thành công các tab của trang quản lý đơn bán bao gồm: "tất cả, đã đặt hàng, đã nhận, đã huỷ". Nút tải lại có thể click
QLDMB_2	Kiểm tra trang quản lý đơn mua	Đăng nhập thành công acc dev15/123456	1. Truy cập vào trang chủ dưới link website với trạng thái đăng nhập 2. Click icon quản lý 3. Click item "Đơn mua"	Hiển thị thành công trang quản lý đơn mua Hiển thị thành công các tab của trang quản lý đơn mua bao gồm: "tất cả, đã đặt hàng, đã nhận, đã huỷ". Nút tải lại có thể click
QLTTU_1	Kiểm tra trang quản lý thông tin user	Đăng nhập thành công acc dev15/123456	1. Truy cập vào trang chủ dưới link website với trạng thái đăng nhập 2. Click icon user 3. Click item "Chào {tên user}"	Hiển thị thành công trang quản thông tin user Hiển thị thành công thông tin user Nút "Xác nhận" có thể click
QLTV_1	Kiểm tra trang thêm mới voucher	Đăng nhập thành công acc dev15/123456	1. Truy cập vào trang chủ dưới link website với trạng thái đăng nhập 2. Click icon quản lý 3. Click tab "Quản lý voucher" 4. Click nút "thêm voucher"	Hiển thị thành công trang tạo voucher. Hiển thị thông tin cần cung cấp: tên voucher, mã voucher, ngày bắt đầu khuyến mãi, ngày kết thúc khuyến mãi, minimum value, discount percentage.

				Nút "Tạo voucher" có thể click
QLTV_2	Kiểm tra trang quản lý voucher	Đăng nhập thành công acc dev15/123456	1. Truy cập vào trang chủ dưới link website với trạng thái đăng nhập 2. Click icon quản lý 3. Click tab "Quản lý voucher"	Hiển thị thành công trang quản lý voucher có các tab: tất cả voucher, voucher theo giá tiền, voucher theo phần trăm. Hiển thị danh sách voucher gồm các thông tin: tên, mã, thời gian hiệu lực, giá trị tối thiểu, loại voucher, số tiền giảm. Nút "Thêm voucher" có thể click
QLTTBD_1	Kiểm tra trang tạo bài đăng_ đầu sách	Đăng nhập thành công acc dev15/123456	1. Truy cập vào trang chủ dưới link website với trạng thái đăng nhập 2. Click icon user 3. Chọn item "Chào {tên user}" 3. Click nút "Thêm bài đăng"	Hiển thị thành công trang tạo bài đăng Hiển thị thông tin cần thêm: Tiêu đề, tác giả, thể loại, nhà xuất bản, số trang, kích thước, isbn, ngày phát hành, ngôn ngữ. Nút "chọn sách" có thể click.
QLTTBD_2	Kiểm tra trang tạo bài đăng_ tài liệu	Đăng nhập thành công acc dev15/123456	1. Truy cập vào trang chủ dưới link website với trạng thái đăng nhập 2. Click icon user 3. Chọn item "Chào {tên user}" 3. Click nút "Thêm bài đăng" 4. Cuộn đến tài liệu	Hiển thị thành công trang tạo bài đăng Hiển thị thông tin cần thêm: phần trăm hư hại. Nút "upload ảnh" có thể click.
QLTTBD_3	Kiểm tra trang tạo bài đăng_ Bài đăng	Đăng nhập thành công acc dev15/123456	1. Truy cập vào trang chủ dưới link website với trạng thái đăng nhập 2. Click icon user 3. Chọn item "Chào {tên user}" 3. Click nút "Thêm bài đăng" 4. Cuộn đến bài đăng	Hiển thị thành công trang tạo bài đăng Hiển thị thông tin cần thêm: mô tả trạng thái sách, chọn dịch vụ, thuê và bán, tiền cọc, giá thuê theo ngày, phí phạt trả trễ theo ngày, giá bán. Nút "upload ảnh" có thể click.
Trang tìm kiếm (6 cases)				

TK_1	Kiểm tra tìm kiếm theo thẻ loại	Đăng nhập thành công acc dev15/123456	1. Truy cập vào trang chủ dưới link website với trạng thái đăng nhập 2. Click icon search 3. Click checkbox thẻ loại học thuật	Hiển thị thành công danh sách bài đăng sách theo thẻ loại học thuật. Checkbox thẻ loại học thuật đã được đánh dấu checked
TK_2	Kiểm tra hoạt động nút Xoá bộ lọc	Đăng nhập thành công acc dev15/123456	1. Truy cập vào trang chủ dưới link website với trạng thái đăng nhập 2. Click icon search 3. Click checkbox thẻ loại học thuật 4. Click nút "Xoá bộ lọc"	Xoá bộ lọc thành công. Trước khi xoá bộ lọc trang kiểm hiển thị thành công danh sách sách theo thông tin lọc. Sau khi xoá bộ lọc thì trang tìm kiếm hiển thị thành công danh sách toàn bộ sách.
TK_3	Kiểm tra hoạt động nút dropdown	Đăng nhập thành công acc dev15/123456	1. Truy cập vào trang chủ dưới link website với trạng thái đăng nhập 2. Click icon search 3. Click nút dropdown "danh mục" lần 1 4. Click nút dropdown "danh mục" lần 2	Sau khi click dropdown lần 1 thì danh sách thẻ loại ẩn thành công. Sau khi click dropdown lần 2 thì danh sách thẻ loại hiển thị thành công.
TK_4	Kiểm tra trang tìm kiếm	Đăng nhập thành công acc dev15/123456	1. Truy cập vào trang chủ dưới link website với trạng thái đăng nhập 2. Click icon search	Hiển thị thành công sidebar lọc theo. Nút "xoá bộ lọc" có thể click. Hiển thị thành công danh mục thẻ loại.
TK_5	Kiểm tra chọn bài đăng theo dịch vụ	Đăng nhập thành công acc dev15/123456	1. Truy cập vào trang chủ dưới link website với trạng thái đăng nhập 2. Click icon search 3. Click nút để chọn dịch vụ 4. Click item "Chỉ sách thuê" 5. Click nút để chọn dịch vụ 6. Click item "Chỉ sách mua"	Sau khi chọn item sách chỉ thuê thì danh sách sách chỉ thuê được hiển thị thành công. Sau khi chọn item sách chỉ mua thì danh sách sách chỉ mua được hiển thị thành công.
TK_6	Kiểm tra tìm theo từ khoá	Đăng nhập thành công acc dev15/123456	1. Truy cập vào trang chủ dưới link website với trạng thái đăng nhập 2. Click icon search	Hiển thị thành công danh sách phù hợp với từ khoá tìm kiếm.

			3. Thêm từ khoá vào input tìm kiếm 4. Click icon search	
--	--	--	---	--

Bảng 91: Kịch bản kiểm thử tự động

7.3.3 Báo cáo tổng hợp kết quả



Hình 86: Tổng hợp kết quả test case

Dựa vào hình trên, dưới đây là bảng tổng hợp kết quả test case và kết luận cho báo cáo:

Loại kết quả	Số lượng	Chi tiết
Đã chạy (All)	45	Tổng số test case được thực thi.
Passed	45	Test case chạy thành công.
Failed	0	Test case bị lỗi.
Flaky	0	Test case không ổn định.
Skipped	4	Test case bị bỏ qua.

Hình 87: Bảng tổng hợp kết quả test case

Kết luận

Tất cả 45 test case đã chạy đều thành công (Passed), không có lỗi (Failed) hoặc bất kỳ test case nào không ổn định (Flaky).

Có 4 test case bị bỏ qua (Skipped) nhưng có lý do hợp lý để bỏ qua.

Thời gian thực thi toàn bộ test case: 15.7 phút, kết thúc lúc 10:18:22 AM ngày 08/12/2024.

Kết quả này chứng minh rằng hệ thống đã vượt qua các kịch bản kiểm thử được thiết kế, đảm bảo độ tin cậy và tính chính xác trong các chức năng đã kiểm tra.

7.3.4 Báo cáo từng testcase

✓ auth.bookstore.setup.js		
✓ authenticate setupBookstore	11.0s	
auth.bookstore.setup.js:11		
✓ bookstore/Card/VerifyCardPage.bookstore.spec.js		
✓ verify detail listing page bookstore	19.7s	
bookstore/Card/VerifyCardPage.bookstore.spec.js:9		
✓ bookstore/Card/VerifyCardTabLease.bookstore.spec.js		
✓ verify detail listing page bookstore	20.1s	
bookstore/Card/VerifyCardTabLease.bookstore.spec.js:9		
✓ bookstore/Card/VerifyCardTabSaleOrder.bookstore.spec.js		
✓ verify tab order sale order bookstore	24.8s	
bookstore/Card/VerifyCardTabSaleOrder.bookstore.spec.js:9		
✓ bookstore/Card/VerifyTabSaleOrderVoucher.bookstore.spec.js		
✓ verify tab order sale order bookstore	17.6s	
bookstore/Card/VerifyTabSaleOrderVoucher.bookstore.spec.js:9		
✓ bookstore/checkoutPage/VerifyBookInfor.bookstore.spec.js		
✓ verify check out page for leaseOrder bookstore	21.5s	
bookstore/checkoutPage/VerifyBookInfor.bookstore.spec.js:9		
✓ bookstore/checkoutPage/VerifyCheckOutLeaseOrder.bookstore.spec.js		
✓ verify check out page for leaseOrder bookstore	30.5s	
bookstore/checkoutPage/VerifyCheckOutLeaseOrder.bookstore.spec.js:9		
✓ bookstore/checkoutPage/VerifyExportLeaseOrder.bookstore.spec.js		
✓ verify check out page for leaseOrder bookstore	25.3s	
bookstore/checkoutPage/VerifyExportLeaseOrder.bookstore.spec.js:9		
✓ bookstore/checkoutPage/VerifyOrderLeaseInfor.bookstore.spec.js		
✓ verify check out page for leaseOrder bookstore	26.2s	
bookstore/checkoutPage/VerifyOrderLeaseInfor.bookstore.spec.js:9		
✓ bookstore/checkoutPage/VerifyOrderSaleInfor.bookstore.spec.js		
✓ verify check out page for leaseOrder bookstore	21.4s	
bookstore/checkoutPage/VerifyOrderSaleInfor.bookstore.spec.js:9		
✓ bookstore/checkoutPage/VerifyPaymentInfor.bookstore.spec.js		
✓ verify check out page for leaseOrder bookstore	20.1s	
bookstore/checkoutPage/VerifyPaymentInfor.bookstore.spec.js:9		
✓ bookstore/checkoutPage/VerifyTabSaleOrderVoucher.bookstore.spec.js		
✓ verify tab order sale order bookstore	22.3s	
bookstore/checkoutPage/VerifyTabSaleOrderVoucher.bookstore.spec.js:9		
✓ bookstore/checkoutPage/VerifyUserOrderInfor.bookstore.spec.js		
✓ verify check out page for leaseOrder bookstore	21.8s	
bookstore/checkoutPage/VerifyUserOrderInfor.bookstore.spec.js:9		

Hình 88: Kết quả testcase Playwright

✓ bookstore/checkoutPage/VerifyCheckOutLeaseOrder.bookstore.spec.js	
✓ verify check out page for leaseOrder bookstore	30.5s
bookstore/checkoutPage/VerifyCheckOutLeaseOrder.bookstore.spec.js:9 ▶ ☰	
✓ bookstore/checkoutPage/VerifyExportLeaseOrder.bookstore.spec.js	
✓ verify check out page for leaseOrder bookstore	25.3s
bookstore/checkoutPage/VerifyExportLeaseOrder.bookstore.spec.js:9 ▶ ☰	
✓ bookstore/checkoutPage/VerifyOrderLeaseInfor.bookstore.spec.js	
✓ verify check out page for leaseOrder bookstore	26.2s
bookstore/checkoutPage/VerifyOrderLeaseInfor.bookstore.spec.js:9 ▶ ☰	
✓ bookstore/checkoutPage/VerifyOrderSaleInfor.bookstore.spec.js	
✓ verify check out page for leaseOrder bookstore	21.4s
bookstore/checkoutPage/VerifyOrderSaleInfor.bookstore.spec.js:9 ▶ ☰	
✓ bookstore/checkoutPage/VerifyPaymentInfor.bookstore.spec.js	
✓ verify check out page for leaseOrder bookstore	20.1s
bookstore/checkoutPage/VerifyPaymentInfor.bookstore.spec.js:9 ▶ ☰	
✓ bookstore/checkoutPage/VerifyTabSaleOrderVoucher.bookstore.spec.js	
✓ verify tab order sale order bookstore	22.3s
bookstore/checkoutPage/VerifyTabSaleOrderVoucher.bookstore.spec.js:9 ▶ ☰	
✓ bookstore/checkoutPage/VerifyUserOrderInfor.bookstore.spec.js	
✓ verify check out page for leaseOrder bookstore	21.8s
bookstore/checkoutPage/VerifyUserOrderInfor.bookstore.spec.js:9 ▶ ☰	
✓ bookstore/DetailListing/VerifyDetailBook.bookstore.spec.js	
✓ verify detail tab in detail listing page bookstore	15.4s
bookstore/DetailListing/VerifyDetailBook.bookstore.spec.js:9 ▶ ☰	
✓ bookstore/DetailListing/VerifyDetailListing.bookstore.spec.js	
✓ verify detail listing page bookstore	15.6s
bookstore/DetailListing/VerifyDetailListing.bookstore.spec.js:9 ▶ ☰	
✓ bookstore/DetailListing/VerifyReviewBook.bookstore.spec.js	
✓ verify detail listing page bookstore	16.5s
bookstore/DetailListing/VerifyReviewBook.bookstore.spec.js:9 ▶ ☰	
✓ bookstore/DetailListing/VerifyTabBookOwner.bookstore.spec.js	
✓ verify tab order sale order bookstore	15.2s
bookstore/DetailListing/VerifyTabBookOwner.bookstore.spec.js:9 ▶ ☰	
✓ bookstore/DetailListing/VerifyTabLeaseOrder.bookstore.spec.js	
✓ verify detail listing page bookstore	21.9s
bookstore/DetailListing/VerifyTabLeaseOrder.bookstore.spec.js:9 ▶ ☰	

Hình 89: Kết quả testcase Playwright

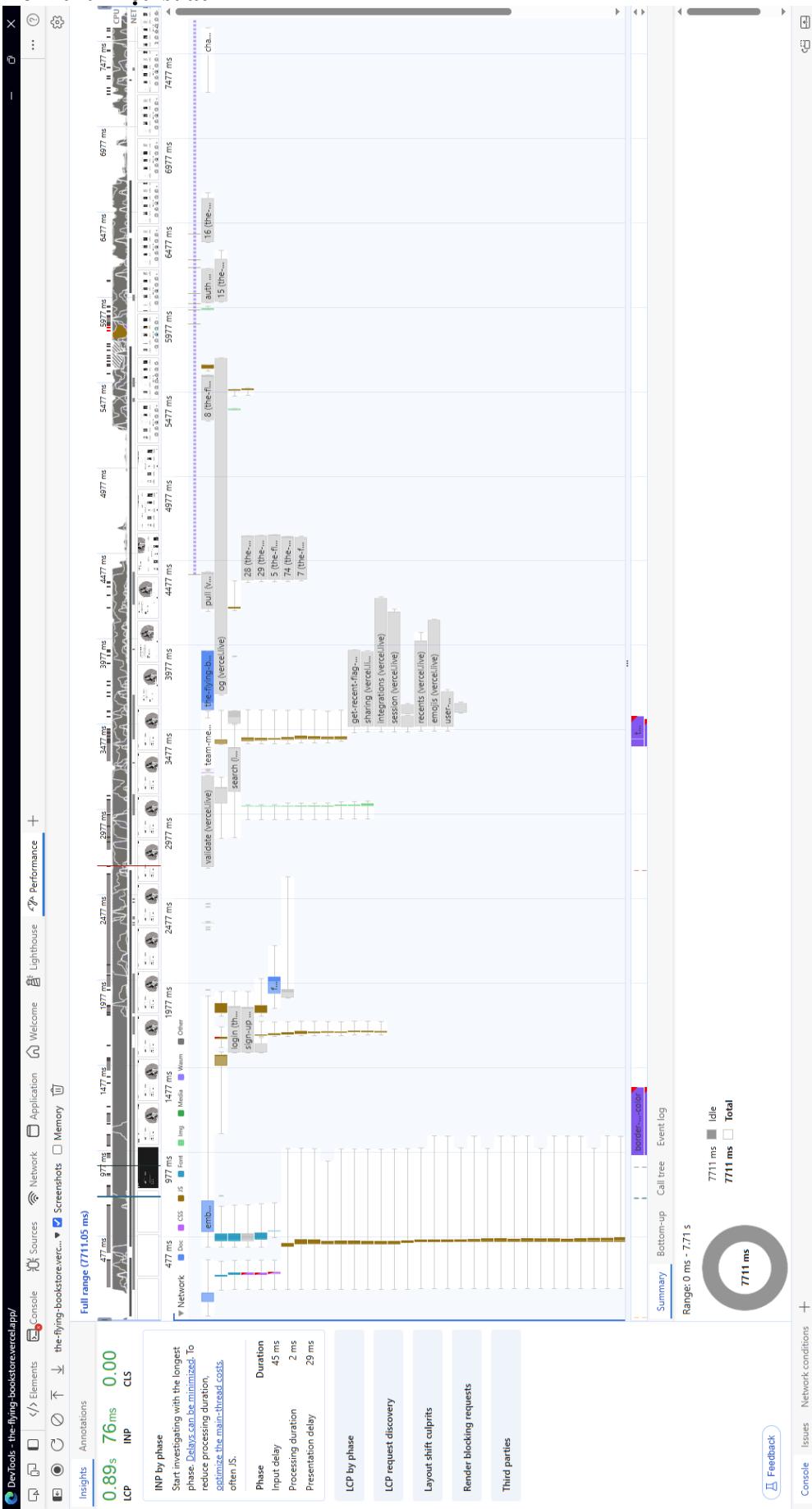
✓ bookstore/DetailListing/VerifyTabReview.bookstore.spec.js		
✓ verify detail listing page bookstore	bookstore/DetailListing/VerifyTabReview.bookstore.spec.js:9	▶ ⏳
15.6s		
✓ bookstore/DetailListing/VerifyTabSaleOrder.bookstore.spec.js		
✓ verify tab order sale order bookstore	bookstore/DetailListing/VerifyTabSaleOrder.bookstore.spec.js:9	▶ ⏳
21.5s		
✓ bookstore/homePage/banner.bookstore.spec.js		
✓ verify banner bookstore	bookstore/homePage/banner.bookstore.spec.js:9	▶ ⏳
9.5s		
✓ bookstore/homePage/carousel.bookstore.spec.js		
✓ verify carousel bookstore	bookstore/homePage/carousel.bookstore.spec.js:9	▶ ⏳
18.7s		
✓ bookstore/homePage/footer.bookstore.spec.js		
✓ verify banner bookstore	bookstore/homePage/footer.bookstore.spec.js:9	▶ ⏳
15.3s		
✓ bookstore/homePage/header.bookstore.spec.js		
✓ verify header bookstore	bookstore/homePage/header.bookstore.spec.js:10	▶ ⏳
10.8s		
✓ bookstore/homePage/statistic-list.bookstore.spec.js		
✓ verify statistic list bookstore	bookstore/homePage/statistic-list.bookstore.spec.js:9	▶ ⏳
10.2s		
✓ bookstore/logout.bookstore.spec.js		
✓ verify banner bookstore	bookstore/logout.bookstore.spec.js:9	▶ ⏳
11.2s		
✓ bookstore/management/manageLeaseOrder/VerifyDonChoThue.bookstore.spec.js		
✓ verify tab order sale order bookstore	bookstore/management/manageLeaseOrder/VerifyDonChoThue.bookstore.spec.js:9	▶ ⏳
14.1s		
✓ bookstore/management/manageLeaseOrder/VerifyDonThue.bookstore.spec.js		
✓ verify tab order sale order bookstore	bookstore/management/manageLeaseOrder/VerifyDonThue.bookstore.spec.js:9	▶ ⏳
15.7s		
✓ bookstore/management/manageListing/VerifyManageListing.bookstore.spec.js		
✓ verify tab order sale order bookstore	bookstore/management/manageListing/VerifyManageListing.bookstore.spec.js:9	▶ ⏳
11.4s		
✓ bookstore/management/manageListing/VerifyManageListingPurchase.bookstore.spec.js		
✓ verify tab order sale order bookstore	bookstore/management/manageListing/VerifyManageListingPurchase.bookstore.spec.js:9	▶ ⏳
15.7s		
✓ bookstore/management/manageListing/VerifyManageListingRent.bookstore.spec.js		
✓ verify tab order sale order bookstore	bookstore/management/manageListing/VerifyManageListingRent.bookstore.spec.js:9	▶ ⏳
12.8s		

Hình 90: Kết quả testcase Playwright

✓ bookstore/management/manageSaleOrder/VerifyDonBan.bookstore.spec.js	
✓ verify tab order sale order bookstore	14.4s
bookstore/management/manageSaleOrder/VerifyDonBan.bookstore.spec.js:9 ▶ ☰	
✓ bookstore/management/manageSaleOrder/VerifyDonMua.bookstore.spec.js	
✓ verify tab order sale order bookstore	13.4s
bookstore/management/manageSaleOrder/VerifyDonMua.bookstore.spec.js:9 ▶ ☰	
✓ bookstore/management/manageUserInfor/VerifyPersonallInfor.bookstore.spec.js	
✓ verify tab order sale order bookstore	11.7s
bookstore/management/manageUserInfor/VerifyPersonallInfor.bookstore.spec.js:9 ▶ ☰	
✓ bookstore/management/manageVoucher/VerifyCreateVoucher.bookstore.spec.js	
✓ verify tab order sale order bookstore	25.4s
bookstore/management/manageVoucher/VerifyCreateVoucher.bookstore.spec.js:9 ▶ ☰	
✓ bookstore/management/manageVoucher/VerifyVoucherPage.bookstore.spec.js	
✓ verify tab order sale order bookstore	2.8m
bookstore/management/manageVoucher/VerifyVoucherPage.bookstore.spec.js:9 ▶ ☰	
✓ bookstore/management/ThemBaiDang/VerifyCreateBook.bookstore.spec.js	
✓ verify tab order sale order bookstore	17.8s
bookstore/management/ThemBaiDang/VerifyCreateBook.bookstore.spec.js:9 ▶ ☰	
✓ bookstore/management/ThemBaiDang/VerifyCreateCopy.bookstore.spec.js	
✓ verify tab order sale order bookstore	19.0s
bookstore/management/ThemBaiDang/VerifyCreateCopy.bookstore.spec.js:9 ▶ ☰	
✓ bookstore/management/ThemBaiDang/VerifyCreateListing.bookstore.spec.js	
✓ verify tab order sale order bookstore	15.5s
bookstore/management/ThemBaiDang/VerifyCreateListing.bookstore.spec.js:9 ▶ ☰	
✓ bookstore/searchPage/Filter/VerifyCheckboxGenre.bookstore.spec.js	
✓ verify dropdown "bộ lọc" bookstore	10.9s
bookstore/searchPage/Filter/VerifyCheckboxGenre.bookstore.spec.js:9 ▶ ☰	
✓ bookstore/searchPage/Filter/VerifyClearFilter.bookstore.spec.js	
✓ verify xoá bộ lọc bookstore	12.6s
bookstore/searchPage/Filter/VerifyClearFilter.bookstore.spec.js:9 ▶ ☰	
✓ bookstore/searchPage/Filter/VerifyDropdownFilter.bookstore.spec.js	
✓ verify dropdown "bộ lọc" bookstore	11.9s
bookstore/searchPage/Filter/VerifyDropdownFilter.bookstore.spec.js:9 ▶ ☰	
✓ bookstore/searchPage/Filter/VerifyFillter.bookstore.spec.js	
✓ verify presence of filter bookstore	12.8s
bookstore/searchPage/Filter/VerifyFillter.bookstore.spec.js:9 ▶ ☰	
✓ bookstore/searchPage/Listing/VerifyListingList.bookstore.spec.js	
✓ verify danh sách listing bookstore	20.0s
bookstore/searchPage/Listing/VerifyListingList.bookstore.spec.js:9 ▶ ☰	
✓ bookstore/searchPage/Listing/VerifySearch.bookstore.spec.js	
✓ verify dropdown "bộ lọc" bookstore	12.7s
bookstore/searchPage/Listing/VerifySearch.bookstore.spec.js:9 ▶ ☰	

Hình 91: Kết quả testcase Playwright

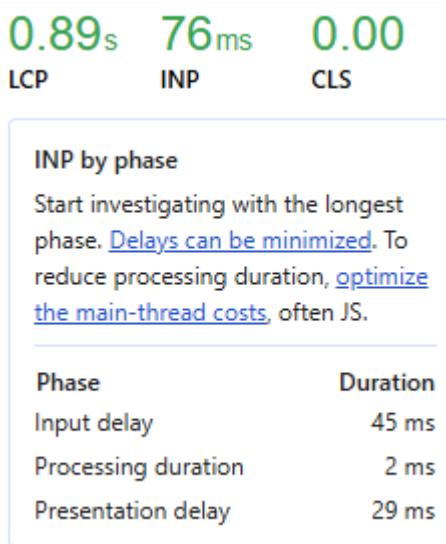
7.4 Kiểm thử hiệu suất



Hình 92: Phân tích chi tiết từ Chrome DevTools

7.4.1 Phân tích chi tiết từ Chrome DevTools

7.4.1.1 Các thông số đo lường chính



Hình 93: Các thông số đo lường chính:

- **LCP (Largest Contentful Paint): 0.89s**
 - Thời gian khá nhanh, phù hợp để cải thiện trải nghiệm người dùng.
 - Mọi tài nguyên ảnh hưởng đến LCP (như hình ảnh lớn hoặc phông chữ) cần tối ưu hóa thêm.
- **INP (Interaction to Next Paint): 76ms**
 - Đây là mức rất tốt (mức lý tưởng < 200ms), cho thấy thời gian phản hồi tương tác ngắn.
- **CLS (Cumulative Layout Shift): 0**
 - Không có sự thay đổi bố cục ngoài ý muốn trong khi trang tải, đây là điểm cộng.

7.4.1.2 Các yêu cầu mạng (Network Requests)

- Có rất nhiều request HTTP (bao gồm tài nguyên từ vercel.live và các endpoint như login, validate, session).
- Thời gian xử lý kéo dài:
 - Có nhiều khoảng thời gian trống hoặc gián đoạn giữa các request (idle).
 - Request get-recent-files và validate xuất hiện trễ (khoảng sau 2-3 giây), có thể làm tăng thời gian tải toàn bộ ứng dụng.

7.4.1.3 Tài nguyên Render-blocking

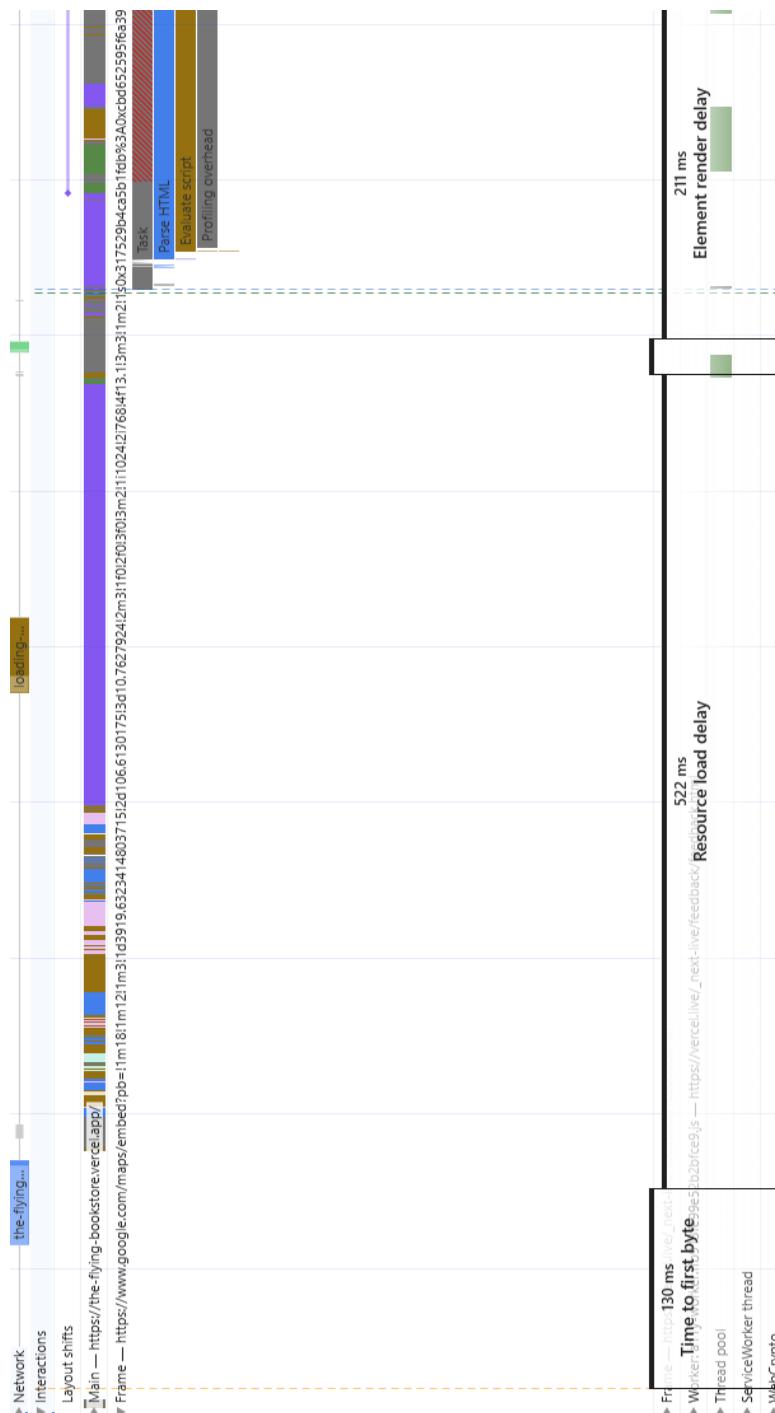
- Một số tài nguyên JavaScript có vẻ đang chặn quá trình render.
- Những đoạn mã third parties cũng xuất hiện trong call stack, điều này có thể ảnh hưởng đến hiệu suất nếu không được tối ưu hóa (ví dụ: tải song song hoặc tải trì hoãn - lazy loading).

7.4.1.4 Vấn đề về truy vấn API từ Backend (Java)

- Các endpoint quan trọng (như session hoặc get-recent-files) mất thời gian đáng kể, có thể liên quan đến:

- Hiệu suất xử lý bên server (Java).
 - Latency giữa backend và server Next.js (Front-end).

7.4.1.5 Chi tiết từng giai đoạn



Hình 94: LCP theo giai đoạn

- Trong hình ta thấy
 - TTFB mất 130ms, nằm trong mức khá tốt nhưng vẫn có thể cải thiện để đạt dưới 100ms.
 - Resource load delay (522ms): Phần này hiển thị rằng các tài nguyên lớn hoặc phức tạp có thể đang bị chậm. Trong hình ảnh, có thể thấy loading và layout task nằm trong giai đoạn này. Có thể tối ưu các tài nguyên bằng cách preload hoặc sử dụng lazy-load cho tài nguyên không cần thiết ngay lập tức.
 - Element render delay (211ms): Đây là phần khá cao và có thể do CSS hoặc JavaScript render-blocking.
- **Third-party scripts:** Có thể thấy trong biểu đồ, có một khung chứa iframe từ Google Maps. Tài nguyên nhúng bên thứ ba (hệ thống có nhúng Google Maps) góp phần gây chậm trong quá trình tải.

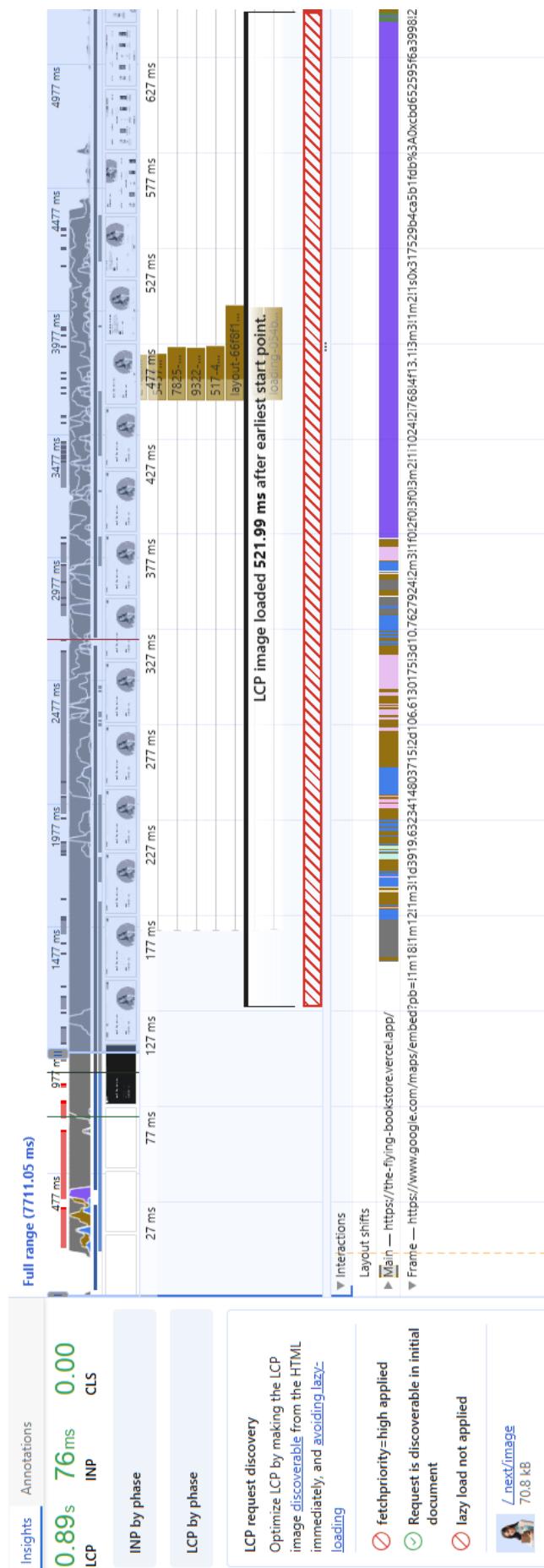
7.4.1.6 Các giai đoạn trong LCP (Largest Contentful Paint)

- Time to First Byte (TTFB - 15%): TTFB chiếm 15% tổng thời gian LCP, nghĩa là server phản hồi chậm ở mức nhẹ. Điều này có thể do:
 - Server cần tối ưu hóa hơn (giảm độ trễ của phản hồi đầu tiên).
 - Có thể cải thiện thông qua việc sử dụng CDN hoặc cải thiện cấu hình máy chủ.
- Resource Load Delay (59%):
 - Đây là phần chiếm tỷ lệ lớn nhất (59%), nghĩa là quá trình tải các tài nguyên bị chậm. Các nguyên nhân tiềm năng:
 - Tài nguyên lớn (hình ảnh, font chữ, hoặc script) chưa được tối ưu hóa.
 - Tài nguyên chưa được sử dụng các kỹ thuật như nén (Gzip, Brotli) hoặc cache.
 - Có thể cải thiện bằng cách:
 - Nén và tối ưu hóa các tệp tĩnh (hình ảnh, CSS, JS).
 - Kiểm tra các third-party scripts gây chậm trễ.
- Resource Load Duration (3%):
 - Chiếm tỷ lệ nhỏ (3%), điều này cho thấy việc tải tài nguyên sau khi bắt đầu có ít vấn đề. Tuy nhiên, cần kiểm tra các tài nguyên cụ thể để chắc chắn.
- Element Render Delay (24%):
 - Phần tử lớn nhất mất 24% thời gian để render sau khi tài nguyên đã tải. Điều này có thể do:
 - Layout hoặc CSS chưa được tối ưu hóa.
 - Font chữ lớn hoặc tài nguyên blocking làm trì hoãn quá trình vẽ (render).

Cách cải thiện:

- Giảm việc sử dụng font chữ lớn hoặc chưa tải trước (preload font).
- Tránh việc layout phức tạp ảnh hưởng đến quá trình render.

7.4.1.7 LCP Request Discovery



Hình 95 LCP Request Discovery

- Kết quả trả về
 - Hình ảnh được tải:
 - Đường dẫn: /_next/image.
 - Kích thước: 70.8 kB (không quá lớn, nhưng cần tối ưu hóa thêm nếu có thể).
 - fetchpriority = high đã được áp dụng:
 - Điều này đảm bảo hình ảnh được ưu tiên tải trước, góp phần giảm thời gian LCP.
- Request is discoverable in the initial document: Hình ảnh LCP đã được chỉ định trong HTML ban đầu, đây là thực hành tốt, tránh làm trì hoãn việc tải.
- Lazy load not applied: Lazy-loading không được áp dụng cho hình ảnh này, phù hợp vì hình ảnh LCP cần tải ngay lập tức, không bị trì hoãn.

7.4.2 Đề xuất cải thiện

7.4.2.1 Front-end (Next.js)

- **Tối ưu hóa LCP:**
 - Giảm kích thước tài nguyên liên quan đến hình ảnh lớn (nếu có), sử dụng Image Optimization của Next.js.
 - Tận dụng preload hoặc lazy-load cho các thành phần không quan trọng trong lần render đầu tiên.
- **Tối ưu hóa các tài nguyên JS/CSS:**
 - Giảm kích thước bundle bằng cách cắt bỏ các thư viện không cần thiết (tree-shaking).
 - Dùng `next/script` để lazy-load các đoạn mã JavaScript bên thứ ba.
 - Nén hình ảnh (sử dụng WebP hoặc nén bằng công cụ như TinyPNG).
 - Sử dụng preload hoặc prefetch cho các tài nguyên quan trọng.
 - Kiểm tra third-party scripts (Google Maps iframe) để giảm thiểu tác động của chúng.
 - Kiểm tra và đơn giản hóa CSS, tránh sử dụng các quy tắc phức tạp ảnh hưởng đến layout hoặc paint.
- **Caching và Prefetch:**
 - Sử dụng `getStaticProps` hoặc `getServerSideProps` để giảm tải số lượng request API trong runtime.
 - Thêm caching headers (HTTP Cache) với tài nguyên tĩnh như hình ảnh, phông chữ.

7.4.2.2 Server

Cải thiện server:

- Sử dụng CDN để giảm độ trễ TTFB.
- Tối ưu hóa máy chủ để trả về tài nguyên nhanh hơn.

7.4.2.3 Back-end (Java)

- **Giảm độ trễ của API:**

- Kiểm tra thời gian phản hồi của các endpoint validate, session, và get-recent-files.
- Xem xét các cơ chế caching (Redis/Memcached) hoặc query tuning ở tầng database.
- **Gộp các request API:** Nếu các endpoint trả về dữ liệu liên quan, xem xét gộp lại thành một API duy nhất để giảm số lượng request.
- **Tối ưu latency backend:** Đảm bảo backend có thời gian phản hồi <100ms (là lý tưởng). Tìm hiểu nguyên nhân các request mất thời gian dài (ví dụ: xử lý logic hoặc truy vấn database chậm).

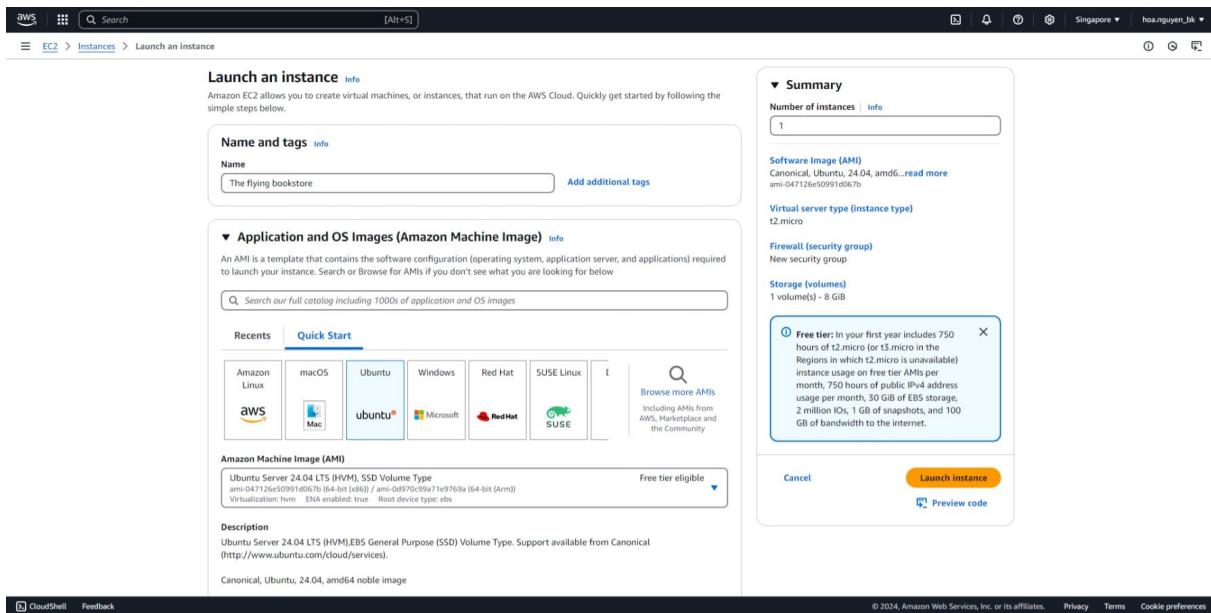
CHƯƠNG 8: TRIỂN KHAI HỆ THỐNG

8.1 Triển khai Back-end

8.1.1 Tạo instance trên EC2

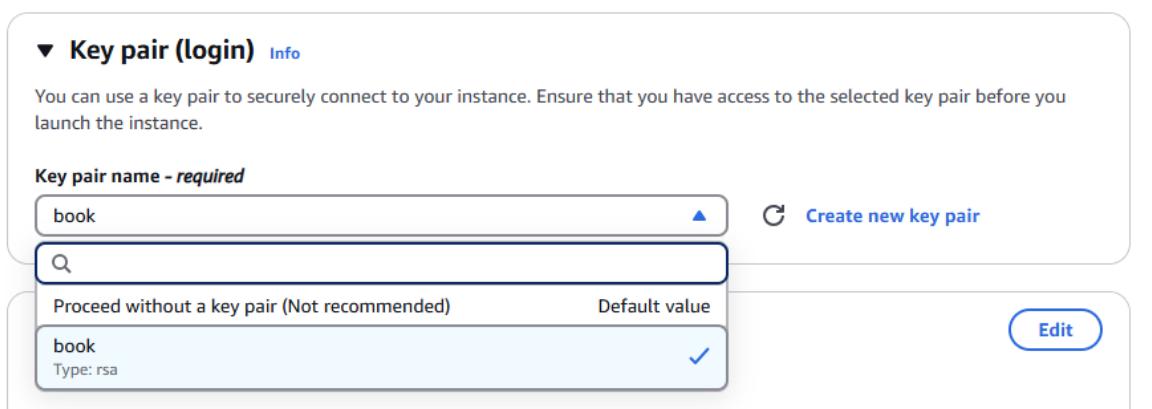
Để triển khai Back-end, ta sử dụng Amazon EC2 để deploy source code backend từ Github.

Bước 1: Ta launch một instance bên trong EC2 tên là “The Flying Bookstore” và sử dụng hệ điều hành Ubuntu trên máy ảo của AWS. Vì đang sử dụng bậc miễn phí nên ta cần thận không sử dụng quá 750 giờ mỗi tháng, nếu không sẽ bị tính phí vào thẻ ta đã đăng ký dùng để mở tài khoản.



Hình 96: Launch một instance bên trong EC2 tên là “The flying bookstore”

Bước 2: Tạo hoặc chọn Key pair có sẵn để ở máy cá nhân, ta có thể đăng nhập một cách bảo mật vào máy ảo của ta



Hình 97: Tạo hoặc chọn Key pair có sẵn

Key Pair là cặp khóa (Private Key và Public Key) được sử dụng để đăng nhập bảo mật vào instance trên AWS. Khi ta tạo hoặc chọn Key Pair:

- **Public Key** được lưu trên máy ảo (EC2 instance).
- **Private Key** được lưu ở máy tính cá nhân của ta và phải được giữ an toàn, vì ta sẽ sử dụng nó để xác thực SSH khi kết nối với instance.

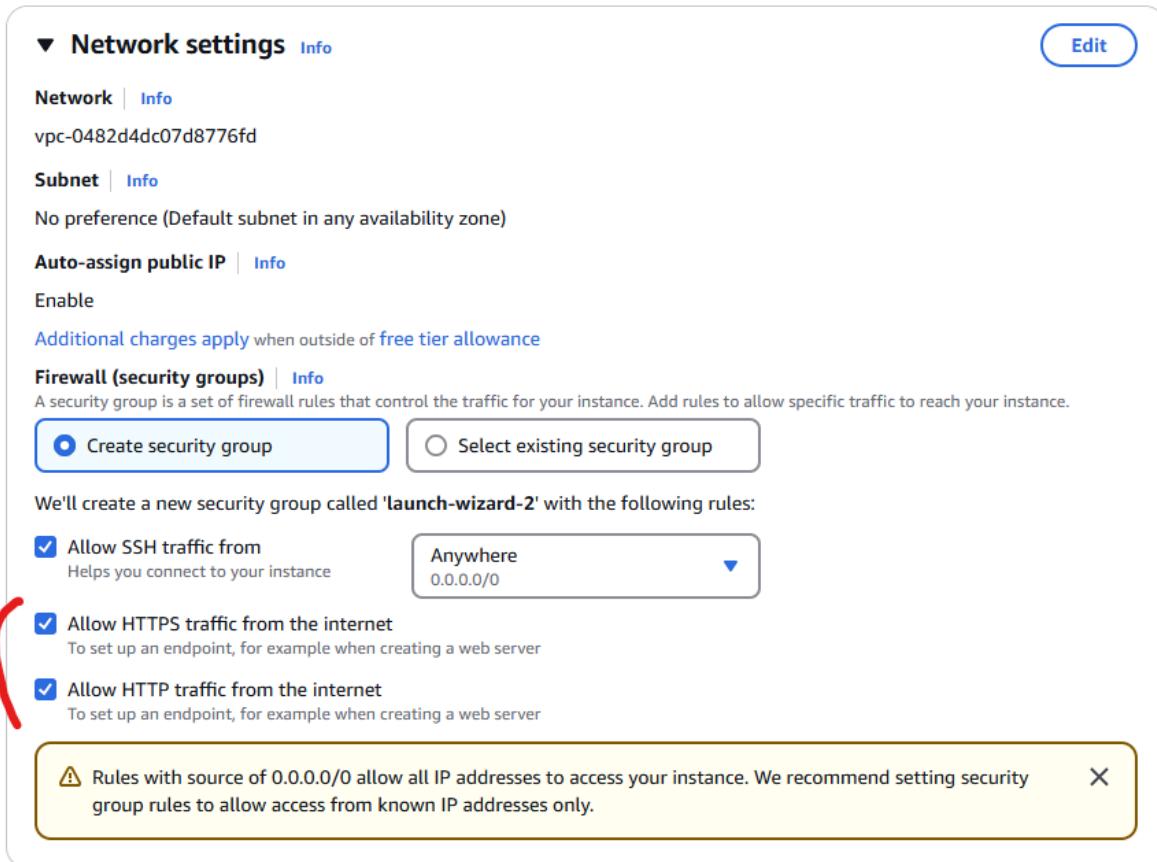
Trong giao diện ta có thể thấy:

- **"Key pair name - required"**: Đây là nơi ta chọn hoặc nhập tên Key Pair của mình.
- **"Create new key pair"**: Nếu ta không có Key Pair nào, AWS cho phép tạo một Key Pair mới. File chứa Private Key (.pem) sẽ được tải về máy tính ta.

Lưu ý:

- File .pem rất quan trọng. Nếu mất file này, ta sẽ không thể truy cập vào instance qua SSH trừ khi ta thực hiện các bước khôi phục phức tạp.
- Hãy chắc chắn rằng Key Pair được lưu trữ an toàn.

Bước 3: Tick chọn 2 ô Allow HTTP và HTTPS traffic



Hình 98: Tick chọn 2 ô Allow HTTP và HTTPS traffic

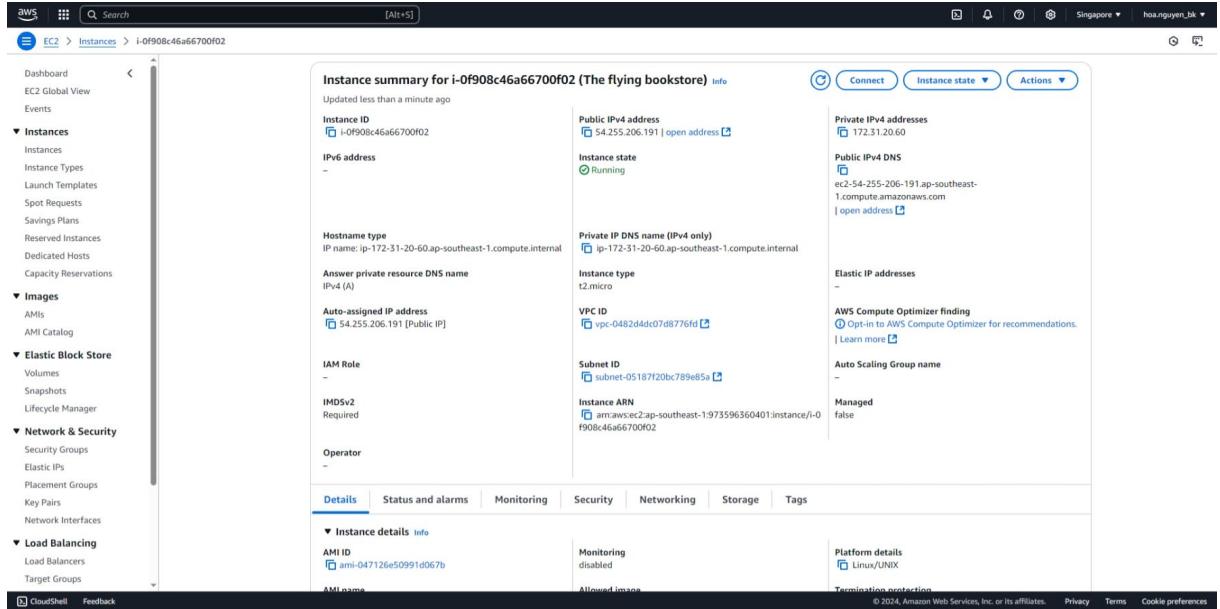
Hai ô này liên quan đến việc cấu hình **Security Group**, là một tập hợp các quy tắc tường lửa (firewall rules) dùng để kiểm soát lưu lượng mạng đến và đi của instance.

- **"Allow HTTP traffic from the internet"**: Cho phép kết nối qua giao thức HTTP (port 80) từ mọi IP trên Internet. Điều này cần thiết nếu muốn tạo một endpoint hoặc chạy một trang web không mã hóa (HTTP), ở đây chính là ta muốn thử kết nối bằng localhost từ source front-end của ta.
- **"Allow HTTPS traffic from the internet"**: Cho phép kết nối qua giao thức HTTPS (port 443) từ mọi IP trên Internet. Điều này cần thiết nếu triển khai trang web có hỗ trợ mã hóa SSL/TLS. Quan trọng nhất là khi ta deploy back-end lên ta cần phải tạo thêm sub domain cho back-end để back-end có thể có giao thức HTTPS thì vercel từ front-end mới cho phép kết nối qua.

Lưu ý về bảo mật:

- Quy tắc 0.0.0.0/0 cho phép tất cả IP trên Internet truy cập vào các port được bật. Điều này là mặc định để thuận tiện cho truy cập công cộng, nhưng có thể tiềm ẩn nguy cơ bảo mật nếu không được quản lý cẩn thận.
- Nếu bạn biết cụ thể IP nào sẽ truy cập (ví dụ, máy tính bạn), bạn có thể giới hạn nguồn kết nối bằng cách thay 0.0.0.0/0 thành dải IP hoặc IP cố định.

Sau 3 bước này ta bấm “Launch instance” và ta đã thành công tạo một instance trên EC2 của AWS. Kết quả sau khi bấm Launch Instance:



Hình 99: Instance tạo thành công

- **Tạo thành công EC2 Instance:**

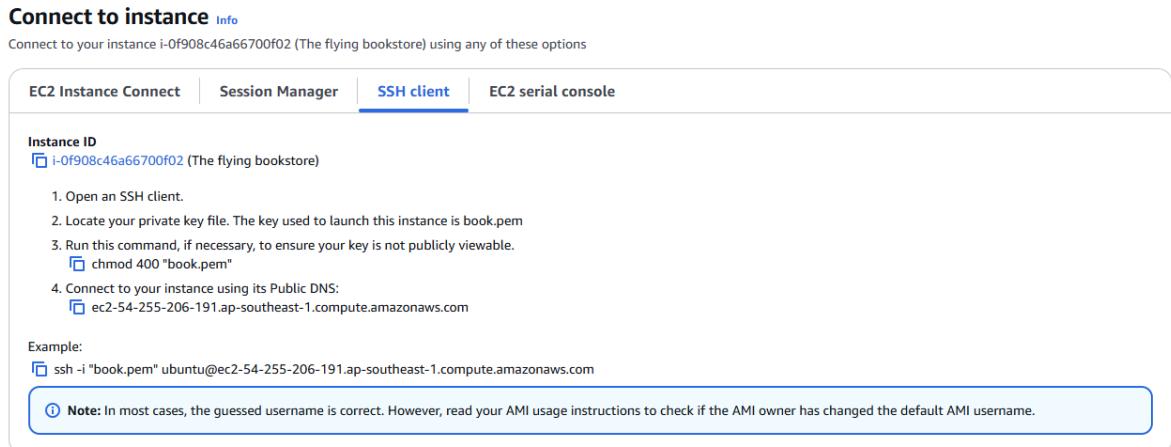
- AWS sẽ tạo một máy ảo dựa trên các cấu hình bạn đã chọn bao gồm: hệ điều hành Ubuntu, key pair “book” và security group.
- Bạn sẽ được chuyển đến trang quản lý EC2, nơi bạn có thể theo dõi trạng thái của instance. Hiện tại instance của tác giả là “Running” vì instance lúc tác giả chụp màn hình còn đang hoạt động. Hiện tại thì tác giả đã tắt vì hết thời gian sử dụng miễn phí.

- **Public IP và HTTP Link:**

- Mỗi instance được cung cấp một Public IP Address hoặc Public DNS nếu tính năng Auto-assign Public IP được bật (mà tác giả đã chọn trong bước 3).
- Link của tác giả có dạng: <http://ec2-54-255-206-191.ap-southeast-1.compute.amazonaws.com>

- **Truy cập instance qua trình duyệt:** Bạn có thể dùng trình duyệt truy cập trực tiếp vào link trên.

8.1.2 Kết nối và đưa source code back-end lên EC2



Hình 100: Hướng dẫn kết nối tới instance của EC2

Để kết nối SSH đến instance EC2 từ Windows, bạn có thể làm theo các bước sau:

Bước 1: Cài đặt SSH Client

Trên Windows 10 hoặc 11: Bạn có thể sử dụng Windows Terminal hoặc Command Prompt. SSH đã được tích hợp sẵn.

Bước 2: Đảm bảo file private key (book.pem) đúng định dạng và quyền trên Windows Terminal hoặc Command Prompt.

Sao chép file `book.pem` vào một thư mục, ví dụ: `C:\ec2`

Mở PowerShell hoặc Command Prompt, chạy lệnh sau để cấp quyền đọc:

```
icacls "C:\ec2\book.pem" /inheritance:r /grant:r "%username%:R"
```

Bước 3: Kết nối tới EC2 Instance

- Mở Command Prompt hoặc Windows Terminal.
- Di chuyển tới thư mục chứa file `book.pem`, ví dụ:

```
cd C:\ec2
```

- Chạy lệnh SSH:

```
ssh -i "book.pem" ec2-user@ec2-54-255-206-191.ap-southeast-1.compute.amazonaws.com
```

8.2 Triển khai Database

8.2.1 Cài đặt Docker

Bước 1: Cập nhật hệ thống

```
sudo apt-get update
```

Bước 2: Cài đặt các gói cần thiết

```
sudo apt-get install -y ca-certificates curl
```

Bước 3: Thiết lập keyrings để thêm kho Docker

```
sudo install -m 0755 -d /etc/apt/keyrings
```

```
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
```

```
sudo chmod a+r /etc/apt/keyrings/docker.asc

Bước 4: Thêm kho Docker vào danh sách nguồn của Apt
echo \

"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu \
$(./etc/os-release && echo "$VERSION_CODENAME") stable" | \

sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

Bước 5: Cập nhật lại danh sách gói

```
sudo apt-get update
```

Bước 6: Cài đặt Docker và các plugin

```
sudo apt-get install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin
docker-compose-plugin
```

Lệnh này cài đặt Docker Community Edition (CE) cùng các công cụ cần thiết như **docker-ce-cli** (giao diện dòng lệnh), **containerd.io** (runtime quản lý container), và các plugin mở rộng. Các plugin như **docker-buildx-plugin** giúp build container trên nhiều kiến trúc, còn **docker-compose-plugin** hỗ trợ quản lý ứng dụng phức tạp với nhiều container qua file `docker-compose.yml`. Tùy chọn `-y` tự động chấp nhận cài đặt mà không yêu cầu xác nhận thủ công.

8.2.2 Thiết lập mạng Docker

Tạo một mạng Docker để kết nối các container

```
sudo docker network create book-network
```

8.2.3 Cài đặt và chạy PostgreSQL

Bước 1: Tải về image PostgreSQL

```
sudo docker pull postgres
```

Bước 2: Chạy container PostgreSQL

```
sudo docker run --name postgres --network book-network \
-e POSTGRES_USER=postgres \
-e POSTGRES_PASSWORD=password \
-p 5432:5432 \
-d postgres
```

Giải thích tham số:

- **--name postgres:** Đặt tên container là postgres.
- **--network book-network:** Kết nối container vào mạng book-network.
- **-e POSTGRES_USER=postgres:** Đặt tên người dùng PostgreSQL là postgres.
- **-e POSTGRES_PASSWORD=password:** Đặt mật khẩu là password.
- **-p 5432:5432:** Mở cổng 5432 của máy chủ để truy cập PostgreSQL.
- **-d postgres:** Chạy container ở chế độ nền với image postgres.

8.2.4 Thiết lập và chạy backend

Bước 1: Clone dự án backend

```
git clone https://github.com/Minh718/Backend_doantotnghiep.git
```

Bước 2: Di chuyển đến thư mục dự án

```
cd Backend_doantotnghiep
```

Bước 3: Xây dựng Docker image cho backend

```
sudo docker build -t bookc2c
```

Bước 4: Tạo file .env

Trong thư mục dự án, tạo file .env và thêm nội dung sau:

```
POSTGRES_HOST=postgres  
DB_NAME=postgres  
POSTGRES_USERNAME=postgres  
POSTGRES_PASSWORD=password
```

Giải thích biến môi trường:

- **POSTGRES_HOST:** Tên container PostgreSQL (kết nối qua mạng Docker).
- **DB_NAME:** Tên database.
- **POSTGRES_USERNAME:** Tên người dùng PostgreSQL.
- **POSTGRES_PASSWORD:** Mật khẩu PostgreSQL.

Bước 5: Chạy container backend

```
sudo docker run -d -p 8080:8080 \  
--name bookc2c \  
--network book-network \  
--env-file .env \  
bookc2c
```

Giải thích tham số:

- **-p 8080:8080:** Mở cổng 8080 để truy cập ứng dụng.
- **--network book-network:** Kết nối container vào mạng book-network.
- **--env-file .env:** Cung cấp file .env làm biến môi trường.

8.2.5 Kiểm tra hệ thống

Kiểm tra trạng thái các container

```
sudo docker ps
```

Kiểm tra kết nối PostgreSQL. Bạn có thể sử dụng công cụ như **DBeaver**, **pgAdmin**, hoặc lệnh **psql** để kết nối tới PostgreSQL:

Host: localhost

Port: 5432

User: postgres

Password: password

Ghi chú:

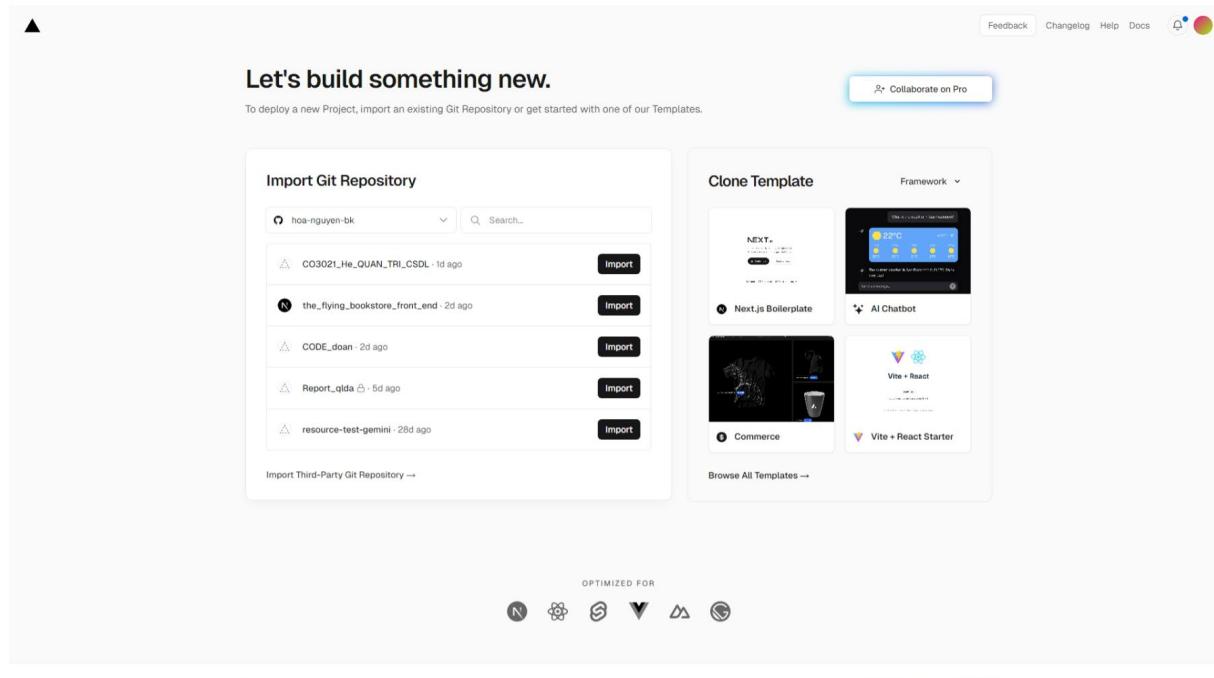
- Đảm bảo bạn có quyền truy cập Internet để tải các image Docker.
- Kiểm tra logs nếu xảy ra lỗi:

```
sudo docker logs <container_name>
```

8.3 Triển khai Front-end

Bước 1: Kết nối với Vercel

- **Đăng nhập Vercel:** Truy cập vercel.com và đăng nhập bằng GitHub.
- **Tạo mới Project:**
 - Nhấn **New Project** → Chọn repository chứa dự án.
 - Cấp quyền Vercel truy cập repository.



Hình 101: Tạo mới Project Vercel

Bước 2: Cấu hình và triển khai

- **Thiết lập build:**
 - Tự động nhận dạng framework **Next.js**.
 - Build command: **next build**.
 - Output directory: **.next**.
- **Triển khai:**
 - Nhấn **Deploy** để bắt đầu build và triển khai.
 - Sau khi hoàn tất, bạn sẽ nhận được link tên miền từ Vercel.
 - **Environment Variables:** Thiết lập biến môi trường trong tab **Settings** của dự án như sau, tạo tên biến môi trường **NEXT_PUBLIC_API_URL** và lấy tên miền EC2 đã thiết lập ở trên đưa vào

`NEXT_PUBLIC_API_URL = http://ec2-54-255-206-191.ap-southeast-1.compute.amazonaws.com`

The screenshot shows the 'Environment Variables' section of the Vercel Project Settings. On the left, a sidebar lists various project settings: General, Domains, Environment Variables (selected), Git, Integrations, Deployment Protection, Functions, Data Cache, Cron Jobs, Log Drains, Security, and Advanced. The main area is titled 'Environment Variables' and contains the following details:

- In order to provide your Deployment with Environment Variables at Build and Runtime, you may enter them right here, for the Environment of your choice.**
- A new Deployment is required for your changes to take effect.**
- Create new** and **Link Shared Environment Variables** buttons.
- Sensitive** checkbox: If enabled, you and your team will not be able to read the values after creation.
- Environments** dropdown: Production (selected).
- Key** and **Value** input fields for adding environment variables. An example entry is 'e.g. CLIENT_KEY'.
- Add Another** button.
- Import .env** button: or paste the .env contents above.
- Save** button.
- Automatically expose System Environment Variables** checkbox.

Below this, a table lists existing environment variables:

	Key	Value	Environment	Last Updated	Actions
1	NEXT_PUBLIC_API_URL	https://...	Development, Preview, Production	Updated 32d ago	...
2	FILE_CLOUD_KEY	-----	Development, Preview, Production	Added 74d ago	...

Shared Environment Variables section: No Shared Environment Variables are linked to this app. Shared Environment Variables are set at the Team level and are inherited by this Project.

Hình 102: Thiết lập biến môi trường

Bước 3: Cập nhật & quản lý

- Tự động cập nhật:** Khi push code lên GitHub nhánh main, Vercel sẽ tự động build và cập nhật phiên bản mới.
- Kiểm tra trạng thái:** Quản lý dự án tại trang dashboard của Vercel.

student060632fpt's projects Hobby the-flying-bookstore-2b9z

Project Deployments Analytics Speed Insights Logs Observability Firewall Storage Settings

Deployments

Continuous generated from hoa-nguyen-bk/the_flying_bookstore_front_end

Upgrade to Pro for 2x more CPUs and faster builds

All Branches... Select Date Range All Environments Status 5/6

Deployment ID	Status	Duration	Commit	Description	Last Updated	Actions
qhbngdot	Ready	1m 34s (2d ago)	main c364a9 Merge pull request #17 from hoa-n...	feat/main	2d ago by hoa-nguyen-bk	...
hdesbtq8c	Ready	1m 36s (2d ago)	feat/add-voucher 549f761 Refactor: remove unused imports a...	feat/add-voucher	2d ago by hoa-nguyen-bk	...
llafg7gav	Ready	2m 12s (2d ago)	main 5135e12 Merge pull request #16 from hoa-n...	main	2d ago by hoa-nguyen-bk	...
e10eppbi6	Ready	2m 16s (2d ago)	feat/voucher 3cee74c Feat: add Vercel Analytics and Spe...	feat/voucher	2d ago by hoa-nguyen-bk	...
a4drfdn0j	Ready	1m 30s (1d ago)	feat/voucher 7751985 Feat: update DocumentDetail and ...	feat/voucher	1d ago by hoa-nguyen-bk	...
3ki6my2xk	Ready	1m 27s (1d ago)	feat/voucher 71079cc Feat: update CreatePost compone...	feat/voucher	1d ago by hoa-nguyen-bk	...
gbliwhbm8	Ready	1m 33s (1d ago)	feat/voucher aa7372f Fix: update hidden price or hidden r...	feat/voucher	1d ago by hoa-nguyen-bk	...
2u4b4thfu	Error	1m 12s (1d ago)	feat/voucher 45d05d2 Feat: create listing success	feat/voucher	1d ago by hoa-nguyen-bk	...
i7hjxw52i	Error	1m 21s (1d ago)	feat/voucher 86a5d2f Feat: done manage order for seller	feat/voucher	1d ago by hoa-nguyen-bk	...
38uer76w0	Error	1m 40s (1d ago)	feat/voucher 68c787e Feat: Manage buy order done	feat/voucher	1d ago by hoa-nguyen-bk	...
21apmtgms	Ready	1m 35s (1d ago)	main 303391b Merge pull request #15 from hoa-n...	main	1d ago by hoa-nguyen-bk	...
ioq83zbfc	Ready	1m 28s (1d ago)	feat/voucher c8b843d Feat: finish build show all order buy	feat/voucher	1d ago by hoa-nguyen-bk	...
hbhooduy	Error	1m 15s (1d ago)	feat/voucher 440ef52 Feat: Refactor image service and I...	feat/voucher	1d ago by hoa-nguyen-bk	...
8i6fbfeed	Ready	1m 31s (1d ago)	feat/voucher 46ab952 Feat: Refactor error handling and I...	feat/voucher	1d ago by hoa-nguyen-bk	...
r4g1xq6jf	Ready	1m 22s (1d ago)	feat/voucher 539ce15 Feat: Fix formatting of review desc...	feat/voucher	1d ago by hoa-nguyen-bk	...
fh36x7tal	Error	46s (1d ago)	feat/voucher a1b0979 Feat: Update review description for...	feat/voucher	1d ago by hoa-nguyen-bk	...
3gnd8uoym	Ready	2m 5s (1d ago)	feat/voucher 465d118 Feat: Enhance book detail page wit...	feat/voucher	1d ago by hoa-nguyen-bk	...
jx14xqafe	Ready	1m 32s (1d ago)	feat/voucher 56bd9dc Feat: Add order management featu...	feat/voucher	1d ago by hoa-nguyen-bk	...
mk3qm5nj6	Error	1m 11s (2d ago)	feat/voucher c91ea2 Feat: Implement voucher creation f...	feat/voucher	2d ago by hoa-nguyen-bk	...
jxjns53fj	Error	1m 10s (2d ago)	feat/voucher 3ac893f Feat: Add voucher session and sho...	feat/voucher	2d ago by hoa-nguyen-bk	...

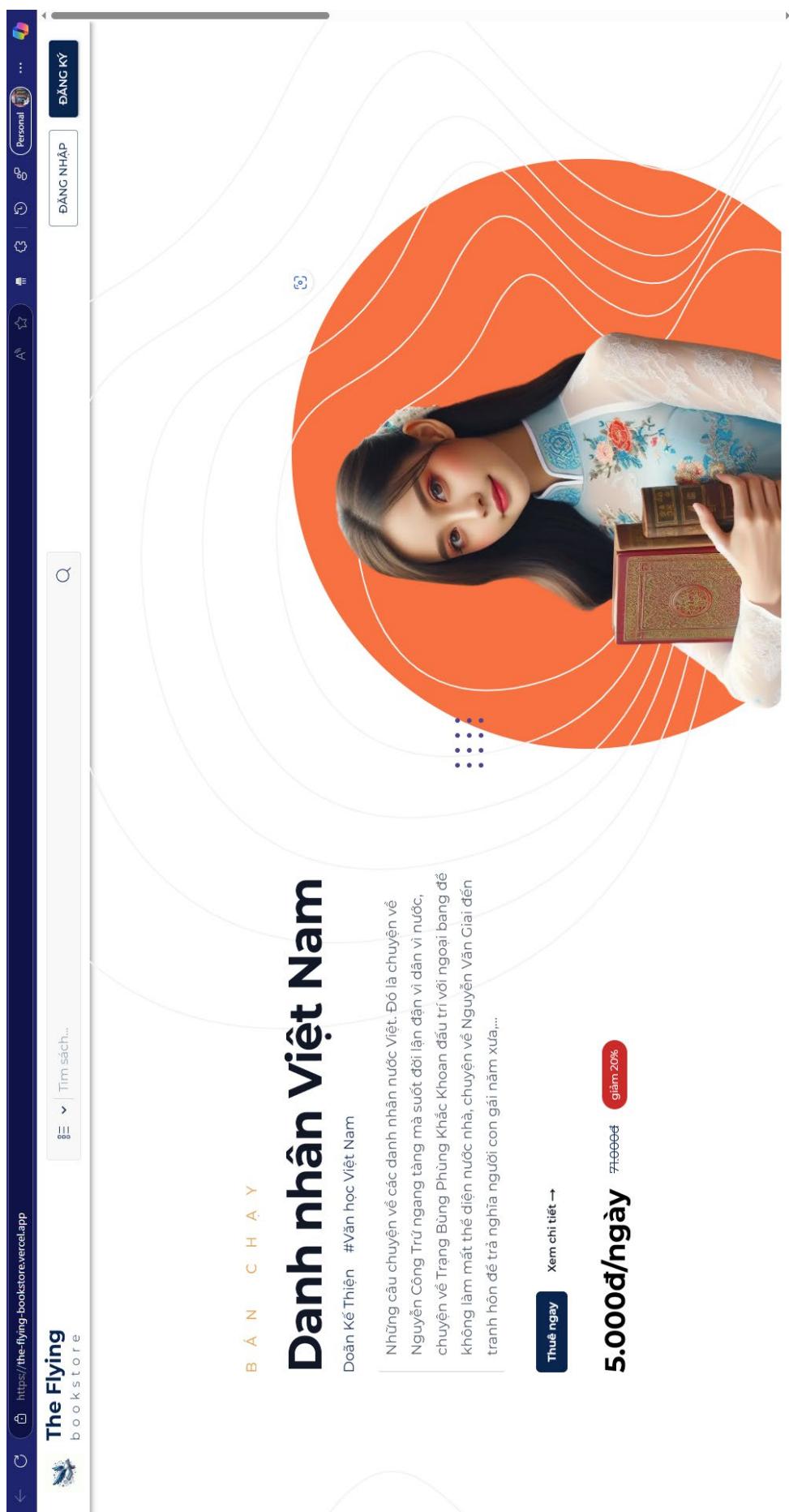
Load More

Home Docs Guides Help Contact Legal

© 2024, Vercel Inc.

● Loading status...

Hình 103: Quản lý deployment tại trang dashboard của Vercel



Hình 104: Deploy Front-end thành công

CHƯƠNG 9: KẾT LUẬN

9.1 Kết quả đạt được so với giai đoạn 2 – Đề án tốt nghiệp

Nhóm đã hoàn thành các nhiệm vụ giai đoạn 3 hơn so với giai đoạn 2 bao gồm:

- Sửa các lỗi chưa kịp sửa ở giai đoạn 2. Nặng nhất là lỗi không triển khai được hệ thống
- Bổ sung các trang hướng dẫn theo lời của giảng viên góp ý ở giai đoạn phản biện và bảo vệ giai đoạn 2
- Thêm các trang mới và chức năng mới bao gồm:
 - Hướng dẫn về tình trạng sách và chính sách bảo mật của trang
 - Voucher toàn sàn
 - Voucher theo người bán
 - Chức năng chính là mua bán sách
- Triển khai hệ thống lên Vercel và Aws
- Kiểm thử hiệu suất trên DevTools
- Tìm hiểu các hệ thống chức năng tương tự giai đoạn 3
- Thiết kế kiến trúc hệ thống phù hợp giai đoạn 3
- Thiết kế các prototype của hệ thống phù hợp giai đoạn 3
- Hoàn thiện việc phân tích và thiết kế hệ thống phù hợp giai đoạn 3
- Hiện thực đầy đủ các tính năng của hệ thống của giai đoạn 3
- Tiến hành kiểm thử kỹ hơn giai đoạn 2 và đánh giá hệ thống giai đoạn 2 và 3.

9.2 Khả năng mở rộng của dự án

- Tích hợp thêm chức năng cho Quản trị viên để nâng cao khả năng làm việc và giảm thiểu thời gian tiêu tốn như là Tự động gợi ý voucher phù hợp với tình trạng hệ thống hiện tại và tối ưu cho khách hàng. Đồng thời Quản trị viên có thể nhìn thấy báo cáo doanh thu hàng tuần, hàng tháng, hàng năm từ các đơn mua và thuê của hệ thống
- Tương tự với người bán và người cho thuê có thể thấy được báo cáo hàng tuần, hàng tháng, hàng năm từ các đơn mua và đơn thuê của mình, từ đó đưa ra được những chiến lược kinh doanh sach tốt hơn cho tương lai.
- Tích hợp chức năng giao tiếp giữa người dùng trong hệ thống như nhắn tin.
- Làm mua bán và cho thuê Ebook theo thời gian cụ thể từ ngày mây đến ngày mây hoặc theo tháng, theo năm.
- Xài microservice, tích hợp Elastic Search để tối ưu hóa kết quả tìm kiếm và cho ra kết quả nhanh hơn.
- Chức năng gợi ý sách dựa trên bộ lọc cộng tác:
 - Gợi ý sách theo sở thích người dùng (user-based collaborative filter)
 - Gợi ý sách dựa trên đặc điểm sách (item-based collaborative filter)
- Cơ chế thưởng coin, phạt coin khi người dùng viết review hoặc trả trễ sách

- Tích hợp với hệ thống thanh toán Blockchain với các smart contract để khi người mua hoặc người thuê đã chuyển tiền cọc hoặc tiền mua thì smart contract duyệt trạng thái và thông báo cho người cho thuê hoặc người bán có thể chuyển hàng mà không cần đến Quản trị viên kiểm duyệt.
- Làm trên Mobile app để có thể áp dụng chức năng quét mã ISBN, scan sách không rõ ISBN mà vẫn thấy rõ sách
- Tích hợp hệ thống giao hàng vận chuyển bên thứ ba để người dùng đỡ bước giao tiếp và tìm hệ thống vận chuyển.

9.3 Kết luận

Dự án sàn thương mại điện tử cho việc thuê mượn, mua bán tài liệu học tập được thiết kế nhằm giải quyết nhu cầu quản lý và tối ưu hóa việc sử dụng tài liệu trong cộng đồng học sinh, sinh viên. Với các chức năng tập trung vào tính năng giao dịch C2C, hệ thống đã thành công trong việc tạo ra một môi trường trực tuyến hỗ trợ việc trao đổi tài liệu hiệu quả, giảm chi phí và tránh lãng phí tài nguyên.

Qua việc cung cấp giao diện thân thiện, các chức năng quản lý đơn hàng rõ ràng và tích hợp các biện pháp bảo đảm như phí cọc và quản lý hạn trả, hệ thống đã góp phần giảm tải áp lực cho quản trị viên, đồng thời tăng cường sự chủ động cho người dùng trong quá trình giao dịch. Ngoài ra, các tính năng như voucher khuyến mãi và đánh giá người dùng không chỉ tạo thêm giá trị cho hệ thống mà còn nâng cao trải nghiệm người dùng và khuyến khích sự tương tác giữa các thành viên.

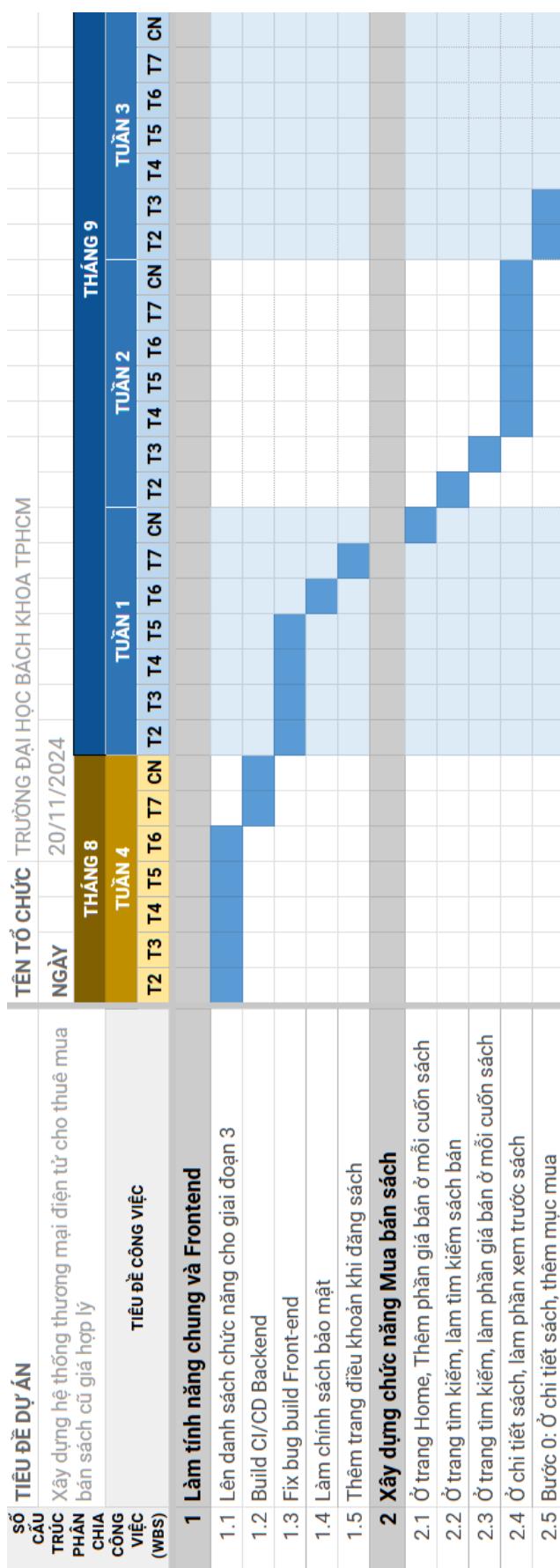
Mặc dù dự án vẫn còn những hạn chế cần cải thiện, nhưng những nền tảng đã xây dựng là một bước tiến quan trọng trong việc áp dụng công nghệ vào quản lý và chia sẻ tài liệu học tập. Với các định hướng phát triển trong tương lai, chúng tôi tin rằng hệ thống sẽ tiếp tục được tối ưu hóa và trở thành một giải pháp tiên phong trong lĩnh vực này, góp phần thúc đẩy sự phát triển của cộng đồng học tập tại Việt Nam và xa hơn nữa.

TÀI LIỆU THAM KHẢO

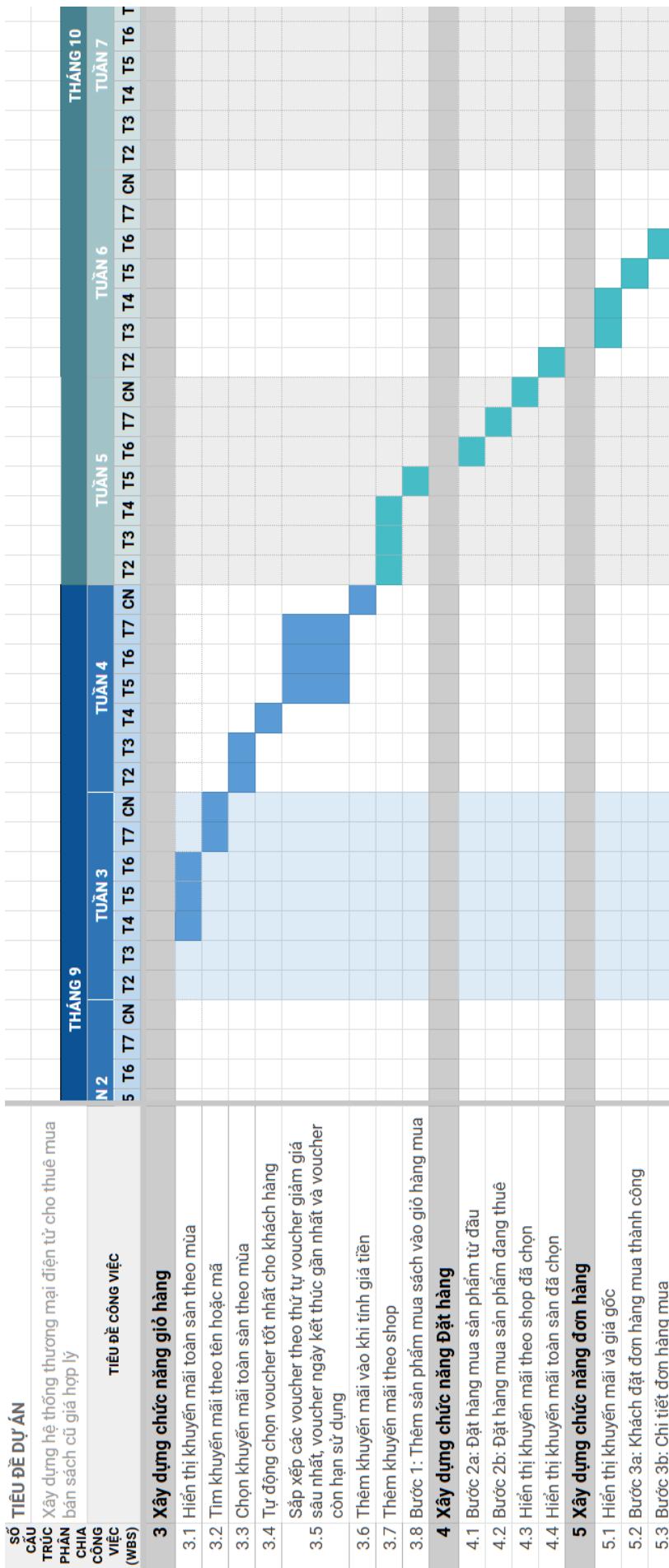
1. gist.github.com. 2023. *Front-end frameworks popularity (React, Vue, Angular and Svelte)*. Truy cập từ: <https://gist.github.com/tkrotoff/b1caa4c3a185629299ec234d2314e190#github-repositories-that-depend-on>.
2. legacy.reactjs.org. 2023. *React – A JavaScript library for building người dùng interfaces*. Truy cập từ: <https://legacy.reactjs.org/>.
3. legacy.reactjs.org. 2023. *React – A JavaScript library for building người dùng interfaces*. Truy cập từ: <https://legacy.reactjs.org/>.
4. react.dev. 2023. *React Community – React*. Truy cập từ: <https://react.dev/community>.
5. react.dev. 2023. *React*. Truy cập từ: <https://react.dev/>.
6. www.typescriptlang.org. 2023. *TypeScript: JavaScript With Syntax For Types..* Truy cập từ: <https://www.typescriptlang.org/>.
7. 2020.stateofjs.com. 2023. *State of JS 2020: JavaScript Flavors*. Truy cập từ: <https://2020.stateofjs.com/en-US/technologies/javascript-flavors/>.
8. www.typescriptlang.org. 2023. *TypeScript: JavaScript With Syntax For Types..* Truy cập từ: <https://www.typescriptlang.org/>.
9. www.typescriptlang.org. 2023. *TypeScript: Why does TypeScript exist?*. Truy cập từ: <https://www.typescriptlang.org/why-create-typescript>.
10. www.altexsoft.com. 2023. *Pros and Cons of TypeScript: When and Why It's Better than Plain JS | AltexSoft*. Truy cập từ: <https://www.altexsoft.com/blog/typescript-pros-and-cons/>.
11. itjet.io. 2023. *What is TypeScript? Benefits of TypeScript and Disadvantages | itjet*. Truy cập từ: <https://itjet.io/blog/what-is-typescript>.
12. github.com. 2023. *mui/material-ui: MUI Core: Ready-to-use foundational React components, free forever. It includes Material UI, which implements Google's Material Design ..* Truy cập từ: <https://github.com/mui/material-ui>.
13. mui.com. 2023. *Material UI: React components based on Material Design*. Truy cập từ: <https://mui.com/material-ui/>.
14. postgresqltutorial.com. 2024. *What is PostgreSQL?*. Truy cập từ: <https://www.postgresqltutorial.com/postgresql-getting-started/what-is-postgresql/>
15. viblo.asia. 2024. Kiến thức cơ bản về các hệ quản trị cơ sở dữ liệu quan hệ. Truy cập từ: <https://viblo.asia/p/kien-thuc-co-ban-vecac-he-quan-tri-co-so-du-lieu-quan-he-aRBvXWMbkWE>
16. javatpoint.com. 2023. *Advantages and disadvantages of Java| TopDev*. Truy cập từ: <https://www.javatpoint.com/advantages-and-disadvantages-of-java>
17. bambooagile.eu. 2021. Pros and Cons of Using Spring Boot| bambooagile. Truy cập từ: <https://bambooagile.eu/insights/pros-and-cons-of-using-spring-boot/>
18. topdev.vn. 2019. *RESTful API là gì? Cách thiết kế RESTful API | TopDev*. Truy cập từ: <https://topdev.vn/blog/restful-api-la-gi/>.
19. www.thepowermba.com. 2023. *REST API: what it is, how it works, advantages and disadvantages*. Truy cập từ: <https://www.thepowermba.com/en/blog/rest-api-what-it-is>.
20. getbootstrap.com. 2023. *Overview Bootstrap*. Truy cập từ: <https://getbootstrap.com/docs/4.1/layout/overview/>.
21. browserslist.ist. 2023. *Browserslist*. Truy cập từ: <https://browserslist.ist/#q=defaults®ion=VN>.

PHỤ LỤC

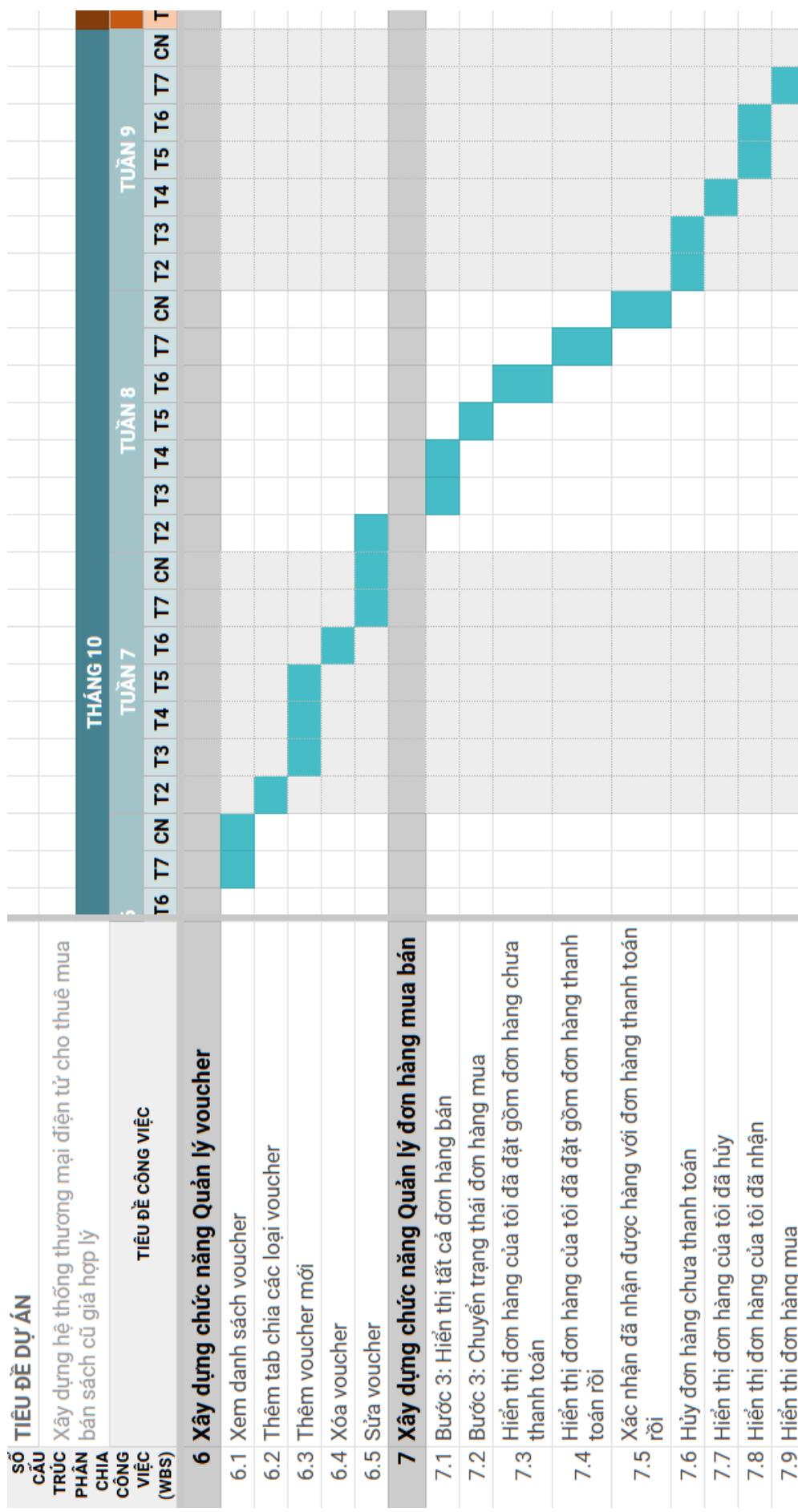
1. Gantt chart



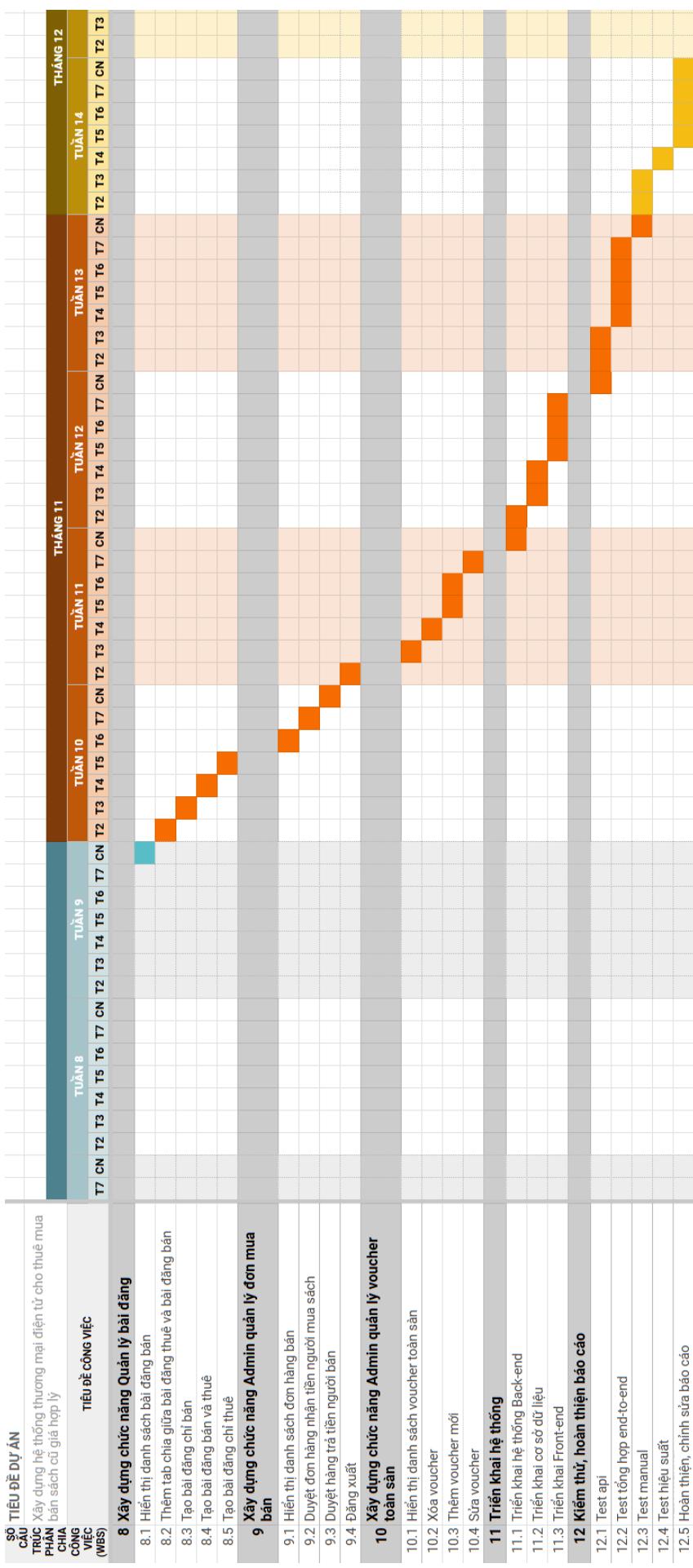
Hình 105: Gantt chart làm tính năng chung và Frontend



Hình 106: Gantt chart xây dựng chức năng Mua bán sách



Hình 107: Gantt chart xây dựng chức năng Quản lý voucher



Hình 108: Gantt chart xây dựng chức năng Quản lý bài đăng

2. Danh mục chức năng

Số thứ tự	Phân theo trang	Chi tiết	Front-end	Back-end	Merge api
1	Trang Home	Thêm phần giá bán ở mỗi cuốn sách	Xong	Xong	Xong
2	Trang tìm kiếm	Tìm kiếm sách bán	Xong	Xong	Xong
3		Thêm phần giá bán ở mỗi cuốn sách	Xong	Xong	Xong
4	Trang chi tiết bài đăng	Bước 0: Ở chi tiết sách, thêm mục mua	Xong	Xong	Xong
5		Chức năng xem trước vài trang cuốn sách	Xong	Xong	Xong
6	Trang giỏ hàng	Hiển thị khuyến mãi toàn sàn theo mùa	Xong	Xong	Xong
7		Tìm khuyến mãi theo tên hoặc mã	Xong	Xong	Xong
8		Chọn khuyến mãi toàn sàn theo mùa	Xong	Không cần	Xong
9		Tự động chọn voucher tốt nhất cho khách hàng	Xong	Không cần	Xong
10		Sắp xếp các voucher theo thứ tự voucher giảm giá sâu nhất, voucher ngày kết thúc gần nhất và voucher còn hạn sử dụng	Xong	Không cần	Xong
11		Thêm khuyến mãi vào khi tính giá tiền	Xong	Không cần	Xong
12		Thêm khuyến mãi theo shop	Xong	Không cần	Xong
13		Bước 1: Thêm sản phẩm mua sách vào giỏ hàng mua	Xong	Không cần	Xong
14		Bước 2a: Đặt hàng mua sản phẩm từ đầu	Xong	Xong	Xong
15		Bước 2b: Đặt hàng mua sản phẩm đang thuê	Xong	Xong	Xong
16		Hiển thị khuyến mãi theo shop đã chọn	Xong	Xong	Xong
17		Hiển thị khuyến mãi toàn sàn đã chọn	Xong	Xong	Xong
18	Xây dựng trang Tạo đơn hàng thành công	Hiển thị khuyến mãi và giá gốc	Mới xong	Mới xong	Mới xong
19		Bước 3a: Khách đặt đơn hàng mua thành công	Xong	Xong	Mới xong
20	Xây dựng trang Chi tiết đơn hàng	Bước 3b: Chi tiết đơn hàng mua	Mới xong	Xong	Mới xong
21	Xây dựng trang Quản lý voucher	Xem danh sách voucher	Xong	Xong	Xong
22		Thêm tab chia các loại voucher	Xong	Xong	Xong
23		Thêm voucher mới	Mới xong	Xong	Mới xong
24		Xóa voucher	Mới xong	Xong	Mới xong
25		Sửa voucher	Mới xong	Xong	Mới xong

26	Xây dựng trang Quản lý đơn hàng mua bán	Bước 3: Hiển thị tất cả đơn hàng bán	Xong	Xong	Xong
27		Bước 3: Chuyển trạng thái đơn hàng mua	Xong	Xong	Xong
28		Hiển thị đơn hàng của tôi đã đặt gồm đơn hàng chưa thanh toán	Xong	Xong	Xong
29		Hiển thị đơn hàng của tôi đã đặt gồm đơn hàng thanh toán rồi	Xong	Xong	Xong
30		Xác nhận đã nhận được hàng với đơn hàng thanh toán rồi	Xong	Xong	Xong
31		Hủy đơn hàng chưa thanh toán	Xong	Xong	Xong
32		Hiển thị đơn hàng của tôi đã hủy	Xong	Xong	Xong
33		Hiển thị đơn hàng của tôi đã nhận	Xong	Xong	Xong
34		Hiển thị đơn hàng mua	Xong	Xong	Xong
35		Hiển thị danh sách bài đăng bán	Xong	Xong	Xong
36	Xây dựng trang tạo bài đăng bán	Thêm tab chia giữa bài đăng thuê và bài đăng bán	Xong	Xong	Xong
37		Tạo bài đăng chỉ bán	Xong	Xong	Xong
38		Tạo bài đăng bán và thuê	Xong	Xong	Xong
39		Tạo bài đăng chỉ thuê	Xong	Xong	Xong
40	Trang Bảo mật	Hiển thị điều khoản bảo mật khi bấm vào Footer	Xong	Không cần	Xong
41		Hiển thị điều khoản bảo mật ở Đăng ký	Xong	Không cần	Xong
42	Trang Hướng dẫn về tình trạng sách	Hiển thị trang hướng dẫn tình trạng sách khi bấm vào Footer	Xong	Không cần	Xong
43		Hiển thị tình trạng đăng sách khi vào trang đăng sách	Xong	Không cần	Xong
44	Xây dựng trang Admin quản lý đơn hàng	Hiển thị danh sách đơn hàng bán	Xong	Xong	Mới xong
45		Duyệt đơn hàng nhận tiền người mua sách	Xong	Xong	Mới xong
46		Duyệt hàng trả tiền người bán	Xong	Xong	Mới xong
47		Đăng xuất	Xong	Không cần	Xong
48	Trang Admin quản lý voucher toàn sàn	Hiển thị danh sách voucher toàn sàn	Xong	Xong	Xong
49		Xóa voucher	Xong	Xong	Xong
50	Trang Admin thêm voucher mới	Thêm voucher mới	Xong	Xong	Xong
51		Sửa voucher	Xong	Xong	Xong

Bảng 92: Danh mục chức năng