

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



BÁO CÁO ĐỒ ÁN TỐT NGHIỆP

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

Ngành: KHOA HỌC MÁY TÍNH

HỘI ĐỒNG: KHMT-20CQ
GVHD: ThS MAI ĐỨC TRUNG
HDBV: TS TRƯƠNG THỊ THÁI MINH
TKHĐ: ThS LÊ ĐÌNH THUẬN
GVPB: ThS PHAN TRUNG HIẾU
---o0o---

SVTH1: NGUYỄN LÊ XUÂN HÒA (2033145)

TP. HỒ CHÍ MINH, THÁNG 12/2024

LỜI CAM ĐOAN

Nhóm em nhóm thực hiện đồ án tốt nghiệp "Xây dựng hệ thống thương mại điện tử cho thuê 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 đồ án tốt nghiệp là sự nghiên cứu và công trình của chính nhóm em dưới sự hướng dẫn của ThS 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 đồ án 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 ThS Mai Đức Trung - người hướng dẫn tận tâm và có sự chi bảo tận tâm trong quá trình nhóm em thực hiện đồ án tốt nghiệp " Xây dựng hệ thống thương mại điện tử cho thuê sách cũ giá hợp lý".

Nhóm em xin chân thành cảm ơn vì sự hỗ trợ và định hình ý kiến từ ThS 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 ThS Mai Đức Trung đã dành thời gian và công sức của mình để hướng dẫn nhóm em 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 nhóm em 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 nhóm em là “Xây dựng hệ thống thương mại điện tử cho thuê sách cũ giá hợp lý”. Tại đây nhóm em 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 đi thuê đồng thời quản lý việc đăng thuê của mình. Để đảm bảo mục tiêu đó nhóm em 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: tạo đơn hàng, xoá 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. Ngoài ra hệ thống còn có giao diện để Quản trị viên hệ thống dễ dàng quản lý thông tin đơn hàng.

MỤC LỤC

LỜI CAM ĐOAN	1
LỜI CẢM ƠN	2
TÓM TẮT ĐỀ TÀI	3
MỤC LỤC	4
DANH MỤC HÌNH ẢNH	8
DANH MỤC BẢNG BIỂU	10
GANTT CHART QUÁ TRÌNH LÀM VIỆC	11
CHƯƠNG 1: TỔNG QUAN	12
1.1 Giới thiệu đề tài	12
1.2 Mục tiêu đề tài	12
1.3 Phạm vi đề tài	12
CHƯƠNG 2: CÁC HỆ THỐNG LIÊN QUAN	15
2.1 BooksRun	15
2.1.1 Giới thiệu	15
2.1.2 Ưu điểm	15
2.1.3 Nhược điểm	15
2.2 Sách cũ giá rẻ	16
2.2.1 Giới thiệu	16
2.2.2 Ưu điểm	16
2.2.3 Nhược điểm	16
CHƯƠNG 3: CÔNG NGHỆ SỬ DỤNG	18
3.1 Công nghệ ở Front-end	18
3.1.1 ReactJS	18
3.1.2 TypeScript	20
3.1.3 Material UI	22
3.1.4 Zustand	24
3.2 Database sử dụng PostgreSQL	25
3.2.1 Giới thiệu về PostgreSQL	25
3.2.2 Ưu điểm	25
3.2.3 Nhược điểm	26
3.3 Công nghệ ở Back-end	26
3.3.1 Vai trò của back-end trong ứng dụng	26
3.3.2 Ngôn ngữ lập trình Java	26
3.3.3 SpringBoot	27
3.3.4 Spring Security	28
3.3.5 Hibernate	28
3.4 RESTful API	29

3.4.1	Khái niệm	29
3.4.2	Ưu điểm	30
3.4.3	Nhược điểm	31
3.5	Docker.....	31
3.5.1	Lịch sử và Mục đích	31
3.5.2	Ưu điểm của Docker.....	31
3.5.3	Các Thành Phần Chính của Docker:.....	32
3.5.4	Nhược điểm	32
	CHƯƠNG 4: PHÂN TÍCH THIẾT KẾ	33
4.1	Nhu cầu người dùng hệ thống	33
4.1.1	Actor	33
4.1.2	Nhu cầu Quản trị viên.....	33
4.1.3	Nhu cầu người dùng	33
4.1.4	Nhu cầu khách vãng lai	33
4.2	Yêu cầu chức năng	34
4.2.1	Chức năng chung của khách và người dùng.....	34
4.2.2	Chức năng riêng dành cho người dùng có tài khoản	34
4.2.3	Các chức năng ở Quản trị viên	35
4.3	Yêu cầu phi chức năng	35
4.3.1	Thiết kế responsive đa thiết bị	35
4.3.2	Thiết kế đa nền tảng.....	36
4.3.3	Hệ thống có tính bảo mật	37
4.3.4	Hiệu suất	37
4.3.5	Khả năng bảo trì	37
4.3.6	Hệ thống có giao diện dễ nhìn, đồng đều, vừa mắt, dễ sử dụng.....	37
4.4	Use case diagram cho toàn hệ thống.....	38
4.5	Đặc tả Use case.....	39
4.5.1	Use cases cho Người dùng	39
4.5.2	Use Cases chung cho Quản trị viên và Người dùng	55
4.5.3	Use Cases cho Quản trị viên.....	57
4.5.4	Use Cases chung cho Người dùng và Khách.....	59
4.6	Mô hình kiến trúc hệ thống.....	62
4.7	Layered architecture diagram	64
4.7.1	Cho Người dùng và khách	64
4.7.2	Cho Quản trị viên	67
4.8	Deploy diagram	68
4.9	Sequence diagram cho các use case	69
4.9.1	Với actor Quản trị viên	69
4.9.2	Với actor Người cho thuê và Người thuê sách	70

4.9.3	Với actor người mua sách và bán sách	71
4.10	Activity diagram (Lược đồ hoạt động)	72
4.10.1	Quản trị viên và Người dùng với chức năng Đăng nhập.....	72
4.10.2	Người dùng quản lý bài đăng	74
4.10.3	Người dùng thuê tài liệu	75
4.11	Lược đồ quan hệ cơ sở dữ liệu (ERD)	77
4.11.1	Lược đồ ERD vắn tắt.....	79
4.11.2	Lược đồ ERD đầy đủ	80
4.11.3	Lược đồ ERD implement.....	81
4.12	Class diagram	84
CHƯƠNG 5: HIỆN THỰC BACK-END.....		86
5.1	Hiện thực database.....	86
5.2	Phân tích source-code back-end	87
5.2.1	Thư mục gốc	88
5.2.2	Thư mục chính.....	89
5.3	Hiện thực api.....	90
5.3.1	Lớp controller và dto	90
5.3.2	Lớp Entity	91
5.3.3	Lớp Mapper và repository	92
5.3.4	Lớp service	93
5.4	Api xây dựng trong dự án	93
CHƯƠNG 6: HIỆN THỰC FRONT END		97
6.1	Phân tích source code front - end	97
6.1.1	Tổng quan thư mục front-end.....	97
6.1.2	Thư mục api.....	98
6.1.3	Thư mục app	98
6.1.4	Thư mục components	99
6.1.5	Thư mục hooks	103
6.1.6	Thư mục types	104
6.1.7	Thư mục utils.....	104
6.2	Bộ nhận diện thương hiệu	104
6.2.1	Logo.....	104
6.2.2	Màu sắc.....	105
6.2.3	Font chữ	106
6.2.4	Component chung.....	106
6.3	Giao diện.....	108
6.3.2	Trang chủ.....	110
6.3.3	Trang tìm kiếm và hiển thị danh sách bài đăng	113
6.3.4	Trang chi tiết bài đăng	116

6.3.5	Trang giỏ hàng.....	120
6.3.6	Trang đặt hàng	123
6.3.7	Trang xuất đơn hàng.....	127
6.3.8	Trang lấy đơn hàng.....	130
6.3.9	Trang đăng nhập	131
6.3.10	Trang đăng ký	133
6.3.11	Trang quản lý thông tin user.....	136
6.3.12	Trang quản lý đơn hàng thuê	138
6.3.13	Trang quản lý đơn hàng cho thuê	143
6.3.14	Trang quản lý đơn mua.....	144
6.3.15	Trang quản lý đơn bán.....	145
6.3.16	Chi tiết đơn hàng	145
6.3.17	Quản lý bài đăng.....	147
6.3.18	Tạo bài đăng	150
6.3.19	Trang quản lý voucher.....	153
6.3.20	Trang thêm voucher.....	155
6.3.21	Trang đăng nhập Quản trị viên	155
6.3.22	Trang quản lý đơn hàng thuê cho thuê Quản trị viên	157
6.3.23	Trang quản lý đơn hàng mua bán Quản trị viên	159
6.3.24	Trang quản lý voucher toàn sàn Quản trị viên.....	160
6.3.25	Trang thêm và sửa voucher toàn sàn Quản trị viên	162
6.3.26	Trang đăng ký Quản trị viên.....	164
6.4	Site map giao diện hệ thống	166
CHƯƠNG 7:	KIỂM THỬ.....	167
7.1	Công cụ kiểm thử.....	167
7.1.1	Công cụ hỗ trợ kiểm thử postman	167
7.1.2	Công cụ hỗ trợ kiểm thử playwright.....	168
7.2	Kiểm thử api.....	168
7.2.1	Cách thực hiện	168
7.2.2	Kịch bản kiểm thử và kết quả kiểm thử.....	170
7.3	Kiểm thử tổng hợp	182
7.3.1	Cách thực hiện	182
7.3.2	Kịch bản kiểm thử	183
7.3.3	Report	187
KẾT LUẬN	189
TÀI LIỆU THAM KHÁO	190

DANH MỤC HÌNH ẢNH

Hình 2.1: Giao diện BooksRun	15
Hình 2.2:Giao diện sách cũ giá rẻ.....	16
Hình 2.3:Giao diện đăng ký của Sách cũ giá rẻ.....	17
Hình 2.4:Giao diện đăng nhập của hệ thống Sách cũ giá rẻ	17
Hình 3.1: Đồ thị phần trăm việc sử dụng tag tên trên stackoverflow.....	18
Hình 3.2:Biểu đồ đường số lượng repository sử dụng framework.....	18
Hình 3.3:Biểu đồ phần trăm ngôn ngữ được yêu thích của Stackoverflow 2020.....	21
Hình 3.4:Nguyên tắc hoạt động của Restfull API	29
Hình 4.1:Biểu đồ độ phủ sóng của trình duyệt trên toàn cầu (10/10/2023)	36
Hình 4.2:Usecase diagram cho toàn bộ hệ thống	38
Hình 4.3:Module người dùng quản lý tài khoản.....	39
Hình 4.4:Module xem danh sách giỏ hàng.....	40
Hình 4.5:Module quản lý đơn hàng	46
Hình 4.6 Use case quản lý đơn hàng của khách	52
Hình 4.7:Module quản lý bài đăng	53
Hình 4.8:Module quản lý đơn hàng-Quản trị viên	57
Hình 4.9:Mô hình kiến trúc hệ thống MVC	62
Hình 4.10: Layered architecture diagram cho người dùng và khách	66
Hình 4.11:Sequence diagram cho Quản trị viên.....	69
Hình 4.12:Sequence diagram cho người dùng và khách	70
Hình 4.13: Activity diagram thuê và trả tài liệu	76
Hình 4.14:ERD cho toàn hệ thống.....	80
Hình 4.15: Classdiagram cho hệ thống.....	84
Hình 5.1: ERD cơ sở dữ liệu init.....	87
Hình 5.2: Cây thư mục lớp controller và dto của book	Error! Bookmark not defined.
Hình 5.3: Cây thư mục lớp entity của book.....	Error! Bookmark not defined.
Hình 5.4: Cây cấu trúc thư mục của job, mapper, repository của book	92
Hình 6.1:Phiên bản logo không chữ	105
Hình 6.2:Logo với tên dự án The Flying Bookstore	105
Hình 6.3:Trang chủ	110
Hình 6.4: Banner ở trang chủ	Error! Bookmark not defined.
Hình 6.5:Section sách mới ra mắt ở trang chủ	Error! Bookmark not defined.
Hình 6.6: Trang danh sách bài đăng và tìm kiếm tài liệu	113
Hình 6.7:Trang chi tiết bài đăng	116
Hình 6.8:Mô tả cuốn sách.....	Error! Bookmark not defined.
Hình 6.9: Tab chi tiết cuốn sách	Error! Bookmark not defined.
Hình 6.10: Section đặt thuê	Error! Bookmark not defined.
Hình 6.11: Tab đánh giá	118
Hình 6.12:Trang giỏ hàng.....	120
Hình 6.13: Trang đặt hàng	123
Hình 6.14: Phần thông báo chuyển khoản trang đặt hàng	126
Hình 6.15: Chọn phương thức thanh toán	Error! Bookmark not defined.
Hình 6.16: Tiến hành thanh toán	Error! Bookmark not defined.
Hình 6.17: Trang xuất đơn hàng.....	127
Hình 6.18: Trang lấy đơn hàng.....	130
Hình 6.19: Trang đăng nhập	131
Hình 6.20: Trang đăng ký.....	133
Hình 6.21: Trang quản lý thông tin user.....	136
Hình 6.22: Trang quản lý đơn hàng của tôi	138
Hình 6.23: Đơn hàng có thể cancel	139

Hình 6.24: Đơn hàng có thể xác nhận thanh toán	139
Hình 6.25: Đơn hàng xác nhận lấy hàng	139
Hình 6.26: Đơn hàng xác nhận trả sách.....	140
Hình 6.27: Quản lý đơn hàng của khách	143
Hình 6.28: Chủ sách xác nhận nhận lại sách	143
Hình 6.29: Chi tiết đơn hàng	145
Hình 6.30: Quản lý bài đăng.....	147
Hình 6.31: Tạo bài đăng	150
Hình 6.32: Thêm mới sách	151
Hình 6.33: Trang đăng nhập Quản trị viên	155
Hình 6.34: Site map giao diện hệ thống	166
Hình 7.1: Cấu trúc Postman.....	167
Hình 7.2: Logo playwright	168
Hình 7.3: Viết test case trong postman.....	169
Hình 7.4: Kết quả run testcase.....	170
Hình 7.5: Báo cáo auto test.....	188

DANH MỤC BẢNG BIỂU

Bảng 5.1: Api thực hiện trong dự án	93
Bảng 7.1: Kịch bản kiểm thử và kết quả của kiểm thử thủ công.....	182
Bảng 7.2: Kịch bản kiểm thử tự động	186

GANTT CHART QUÁ TRÌNH LÀM VIỆC

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 tròn 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 thuê 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 thuê và mua 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 bán và người thuê tìm được nhau thông qua thông tin chung đó. Từ đó phát sinh giao dịch.
- Về việc quá trình 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.
- Đè hạn chế việc mượn 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.
- Đè 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.

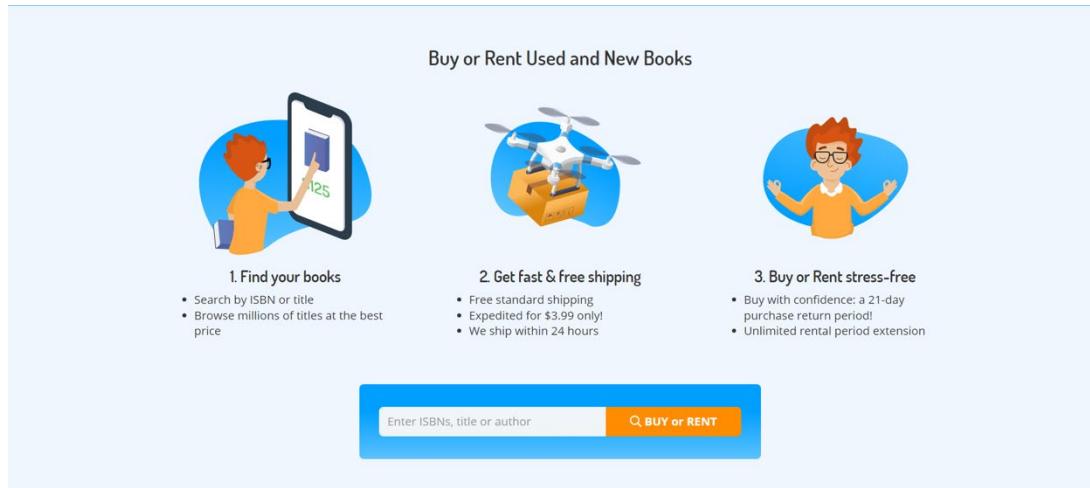
CHƯƠNG 2: CÁC HỆ THỐNG LIÊN QUAN

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 đề mục cho thuê sách. Hệ thống BooksRun còn đầu tư xây dựng app mobile cho người dùng dễ dàng thuê và mua sách cũ. Người dùng có thể quét mã vạch trên cuốn sách để hệ thống BooksRun tìm thấy thông tin sách đó một cách nhanh chóng.

Hình 2.1: Giao diện BooksRun



2.1.2 Ưu điểm

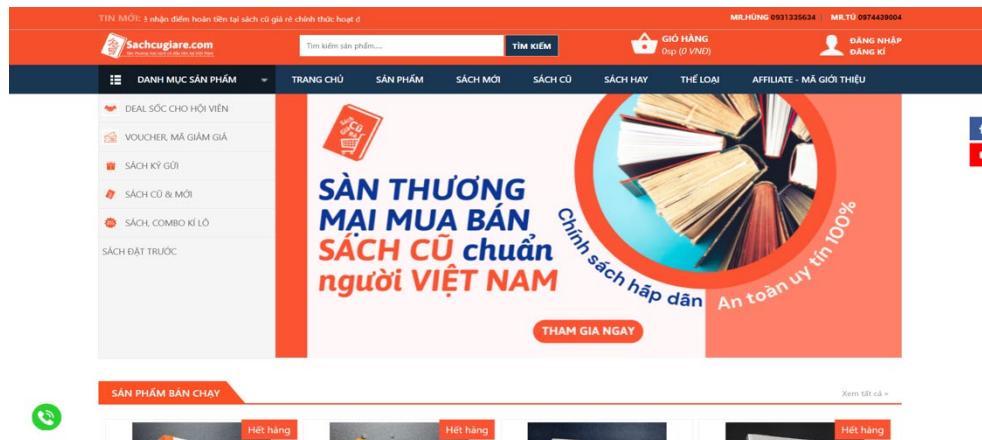
- Hệ thống giao diện được đầu tư kỹ càng, trực quan, dễ sử dụng.
- Sử dụng quét mã ISBN để tìm kiếm sách nhanh hơn, vì hệ thống đầu tư data sách thời gian thực đồ sộ có sẵn với mã ISBN.

2.1.3 Nhược điểm

Vì doanh nghiệp này theo mô hình B2C, tức là tự đứng ra cho thuê và mua bán luôn nên doanh nghiệp này thuộc dạng tập trung lại một chỗ, không tận dụng được ưu điểm của hệ thống nhóm em đang làm là mô hình C2C khi bất kì ai cũng có thể đứng ra cho thuê sách. Vậy nên đối với hệ thống của nhóm em, việc thuê một cuốn sách có thể ở bất kì đâu, giá cả cũng cạnh tranh vì thế có thể mang lại lợi ích lớn nhất cho người dùng.

2.2 Sách cũ giá rẻ

2.2.1 Giới thiệu



Hình 2.2: Giao diện sách cũ giá rẻ

Hệ thống Sách cũ giá rẻ có chức năng cho người dùng ký gửi sách, bán sách cũ và mới, bán sỉ sách, đồng thời hệ thống có chức năng bán sách cũ giá rẻ, giao diện đồng đều, hiển thị nội dung rõ ràng.

2.2.2 Ưu điểm

- Phân loại sách kỹ càng theo danh mục.
- Giao diện đồng đều.
- Hệ thống tương tác nhanh.
- Đầy đủ các chính sách đổi trả, bảo mật, dịch vụ, ...

2.2.3 Nhược điểm

Giao diện hệ thống chưa thật sự chuyên nghiệp về thiết kế. Chữ Đăng nhập và đăng kí ở gần nhau, dẫn đến dễ nhầm lẫn đây là 1 nút. Ở form đăng ký, những trường bắt buộc điền thì nên để dấu sao màu đỏ * để dễ hiểu rằng đây là trường bắt buộc, hoặc để chữ bắt buộc bên cạnh để người dùng biết trường này bắt buộc điền. Vậy nên giao diện không tối ưu UX.

The screenshot shows the header of the Sachcugiare.com website. At the top left is a 'TIN MỚI' section with a link to 'tại nhà'. On the right are contact numbers: MR.HÙNG 0931335634 and MR.TÚ 0974439004. The main navigation menu includes 'DANH MỤC SẢN PHẨM', 'TRANG CHỦ', 'SẢN PHẨM', 'SÁCH MỚI', 'SÁCH CŨ', 'SÁCH HAY', 'THẺ LOẠI', and 'AFFILIATE - MÃ GIỚI THIỆU'. A search bar with placeholder 'Tim kiem sản phẩm....' and a 'TÌM KIẾM' button are also present. On the far right, there's a shopping cart icon labeled 'GIỎ HÀNG 1sp (7.500 VNĐ)', a user profile icon, and links for 'ĐĂNG NHẬP' and 'ĐĂNG KÝ'.

ĐĂNG KÝ

Họ tên

Email

Điện thoại

Mật khẩu

Nhập lại mật khẩu

Đăng ký

Hình 2.3:Giao diện đăng ký của Sách cũ giá rẻ

Giao diện đăng nhập không có ô checkbox ghi nhớ thông tin đăng nhập, điều này làm khách hàng mất thời gian khi phải đăng nhập lại nhiều lần. Vậy nên giao diện không đầu tư UX.

The screenshot shows a login form titled 'ĐĂNG NHẬP'. It has two input fields: 'Tên đăng nhập hoặc Email:' and 'Mật khẩu:', both with placeholder text. Below the password field is a 'Đăng nhập' button. At the bottom left is a link 'Quên mật khẩu' with a question mark icon.

ĐĂNG NHẬP

Tên đăng nhập hoặc Email:

Mật khẩu:

Đăng nhập

⑦ Quên mật khẩu

Hình 2.4:Giao diện đăng nhập của hệ thống Sách cũ giá rẻ

Hệ thống này hiện tại chưa có chức năng cho thuê sách mà chỉ đang bán sách hoặc ký gửi sách. Với những cuốn sách hết hàng, hệ thống chưa lọc ra mà vẫn để trên mục sản phẩm bán chạy. Vậy nên không gian trang chủ bị chiếm mất một khoảng không gian để có thể hiển thị các cuốn sách bán chạy khác xuất hiện để tăng doanh thu. Hệ thống nhóm em sẽ không để các sản phẩm hiện tại đã có người thuê, hoặc sản phẩm hiện không cho thuê nữa lên trang đầu, mà ưu tiên hiển thị sản phẩm bán chạy, sản phẩm được người dùng khuyến đọc.

CHƯƠNG 3: CÔNG NGHỆ SỬ DỤNG

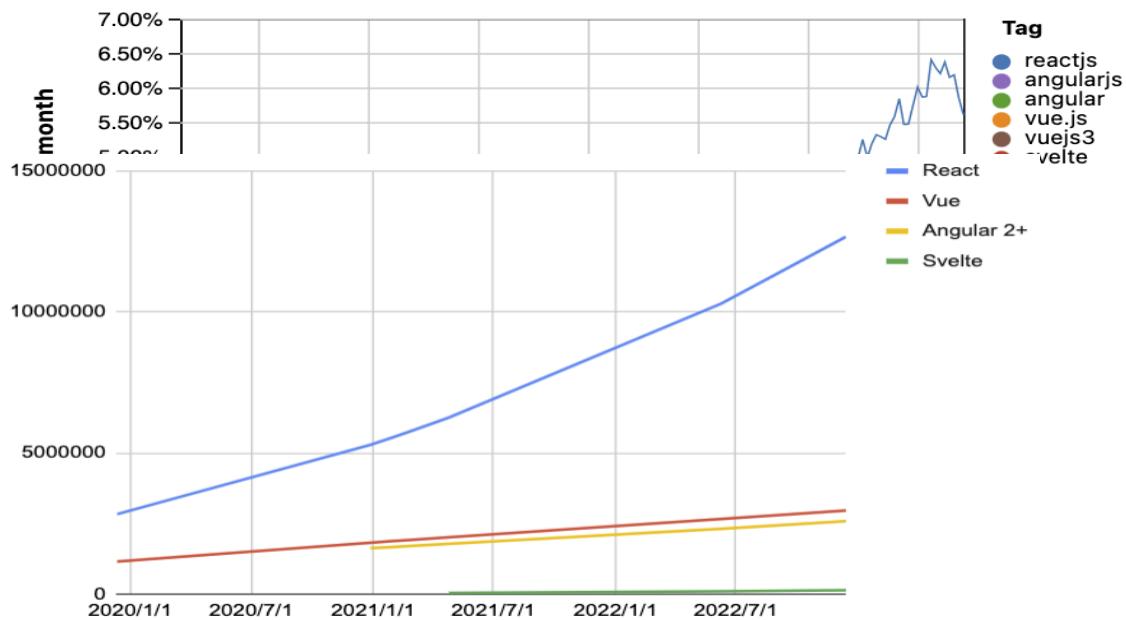
3.1 Công nghệ ở Front-end

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 3.1: 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 3.2: 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.

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:

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 user interfaces*. Truy cập từ: <https://legacy.reactjs.org/>.

- 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

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 3.3: Biểu đồ phần trăm 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 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 bổ 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.
- Bổ 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.

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>.

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 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 đe 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 đe 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

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/>.

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 tiến định kỳ.

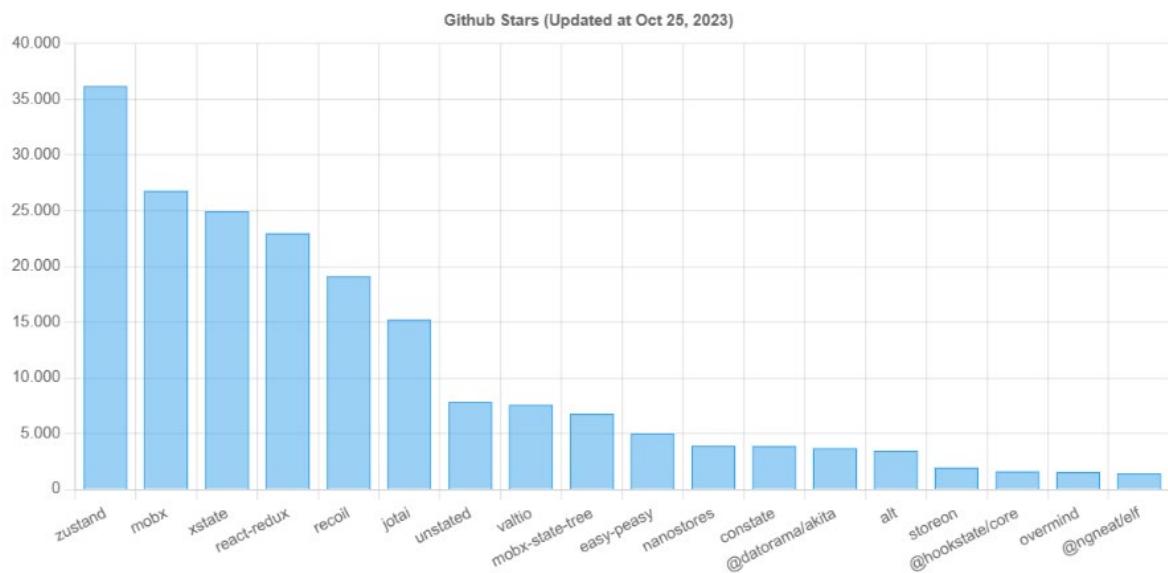
3.1.4 Zustand

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.

3.1.4.2 Tại sao chọn Zustand?



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 thích 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 nhóm 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 developer React 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.

¹⁴ postgresqltutorial.com. 2024. What is PostgreSQL?. Truy cập từ: <https://www.postgresqltutorial.com/postgresql-getting-started/what-is-postgresql/>

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.
- 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 kỹ 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)

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-ve-cac-he-quan-tri-co-so-du-dieu-quan-he-aRBvXWMbkWE>

để chạy trên máy ảo Java 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.
- **Phần mềm khó duyệt:** JVM có thể làm tăng kích thước của ứng dụng, làm cho quá trình duyệt và tải trang chậm chạp hơn.¹⁶

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 nhóm em chọn Spring Boot là vì nó hỗ trợ cho mô hình MVC mà nhóm em 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.

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

- 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 mô-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.
- 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.

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/>

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 qua 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.

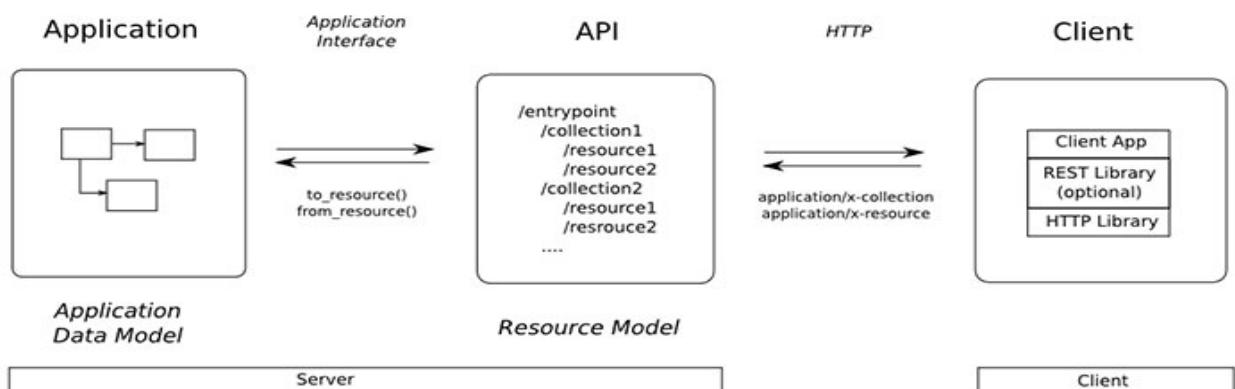
3.3.5.3 *Nhược Điểm*

- 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



Hình 3.4: Nguyên tắc hoạt động của Restfull API

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/>.

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 trùm 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 hoá cho dự án.

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>.

- 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 trực chia sẻ chung kernel và chỉ sử dụng những thành phần cần thiết.
- 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 bất 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.
- Bảo mật: Các container sẽ chia ra dùng chung kernel của hệ điều hành chủ, nên 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 cho thuê.
- 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.

4.1.4 Nhu cầu khách vãng lai

- Đăng ký tài khoản.
- Tìm tài liệu.
- Thấy được bài đăng các tài liệu đang đăng cho thuê.
- Thêm bài đăng vào giỏ hàng được nhưng muốn thuê tài liệu 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 người đã thuê mướn trước. Thêm sản phẩm vào giỏ hàng.
- Chức năng đăng ký tài khoản.
- Xem giỏ hàng và thêm bài đăng 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.
- 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 bài 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 4.120 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:

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

- Máy tính bảng nhỏ hơn, như Amazon Kindle (600×800) và Barnes & Noble Nook (600×1024), được đặt ở chế độ nằm đúng.
- 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.



Hình 4.1: 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 Browerslist²¹ 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.

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

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 viênistrator tạo mật khẩu có độ phức tạp cao, cụ thể như sau:
 - 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 viênistrator 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

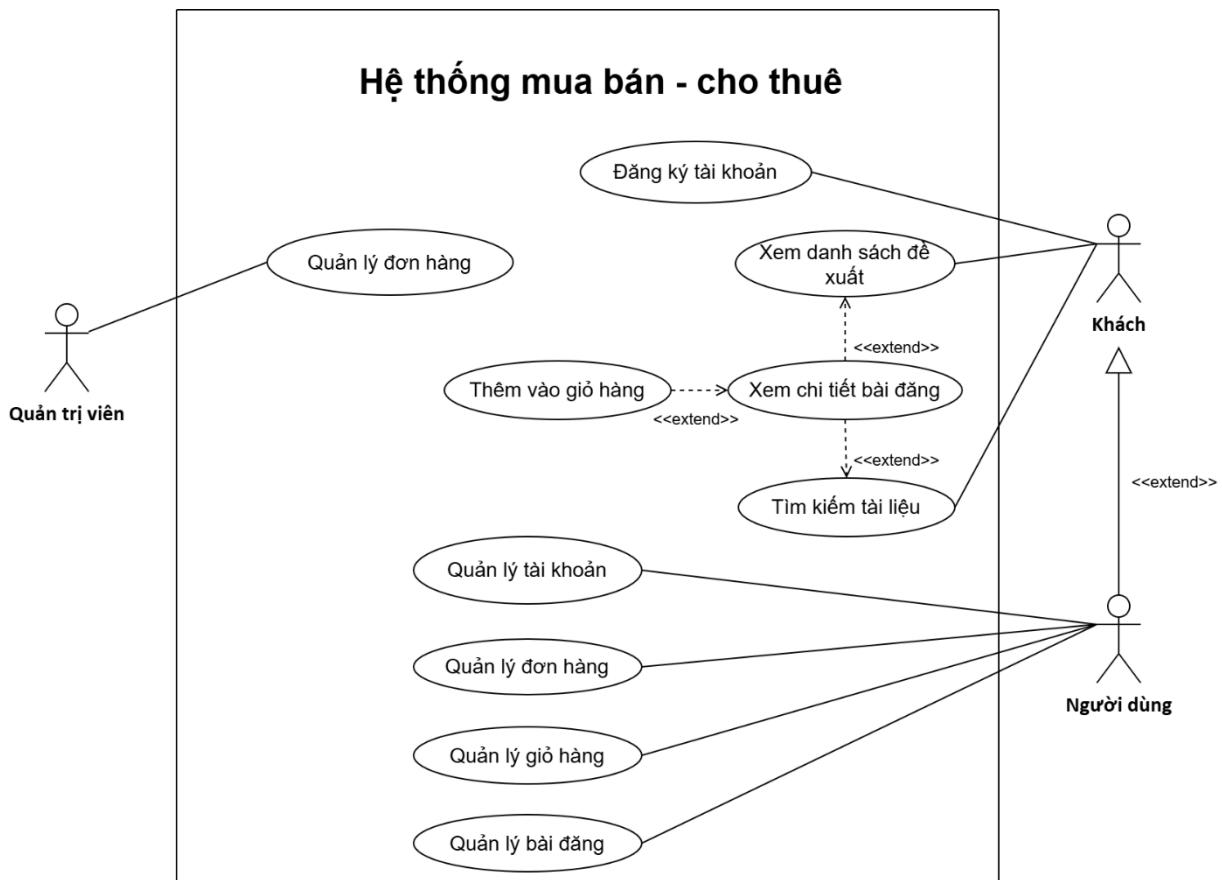
Hệ thống thực hiện bảo trì vào ngày cuối tháng, thời gian bảo trì không vượt quá 8 tiếng.

4.3.6 HỆ THỐNG CÓ GIAO DIỆN DỄ NHÌN, ĐỒNG ĐỀU, VÀ 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 bộ hệ thống

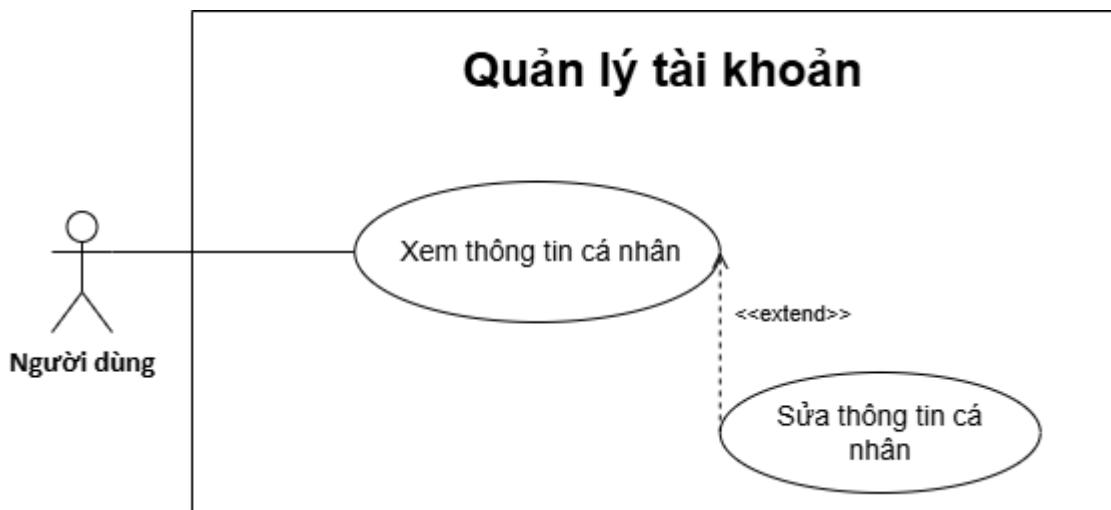


Hình 4.2: Use case diagram cho toàn bộ hệ thống

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 4.3:Module người dùng quản lý tài khoản

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

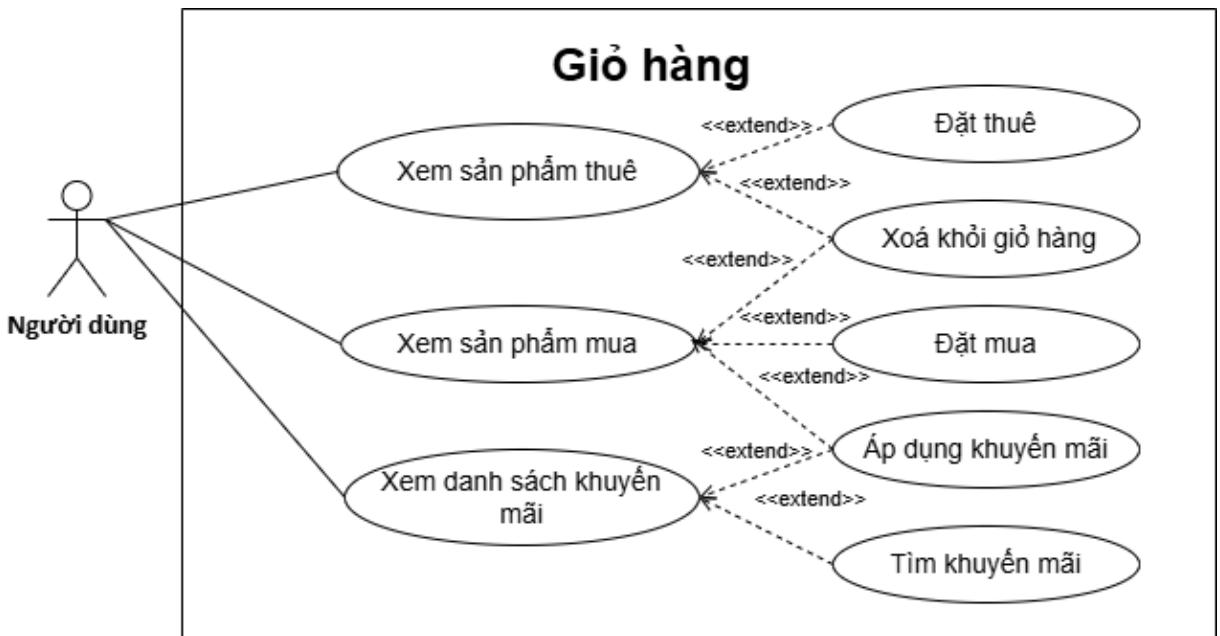
Use case name	Người dùng xem thông tin cá nhân
Actor	Người dùng
Description	Người dùng muốn xem thông tin cá nhân của mình
Preconditions	Người dùng đã đăng nhập thành công Người dùng đang ở trang chủ trang web
Postconditions	Người dùng xem thông tin cá nhân thành công Hệ thống hiển thị thông tin cá nhân
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

- **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

4.5.1.2 Người dùng Xem giỏ hàng



Hình 4.4: 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

- 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

- 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"> Người dùng chọn nút “thùng rác” để xoá sản phẩm đó khỏi giỏ hàng Hệ thống xoá sản phẩm khỏi giỏ hàng của người dùng
Alternative Flows	Không có
Exception	Không có

- **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"> Hệ thống tính toán và chọn khuyến mãi giảm nhiều nhất cho người dùng ở ngay trang giỏ hàng. Người dùng bấm vào “Chọn hoặc nhập mã khác” Hệ thống hiển thị modal danh sách khuyến mãi 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 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

- **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"> Người dùng chọn khuyến mãi khác với khuyến mãi hiện tại hệ thống đã tự động chọn cho người dùng 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

- **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"> Người dùng tìm khuyến mãi theo tên hoặc theo mã 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 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

- **Đặt thuê**

Use case name	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	<p>1. Tại trang giỏ hàng >> Người dùng bấm nút “Đặt thuê hàng”</p> <p>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</p> <p>3. Người dùng nhập thông tin đặt thuê >> bấm xác nhận</p> <p>4. Hệ thống cập nhật thông tin người đặt hàng. Hệ thống hiển thị thông báo nhắc “chọn thanh toán rồi tạo đơn hàng”</p> <p>5. Người dùng kiểm tra thông tin đặt hàng >> người dùng chọn phương thức thanh toán vnpay</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 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</p>

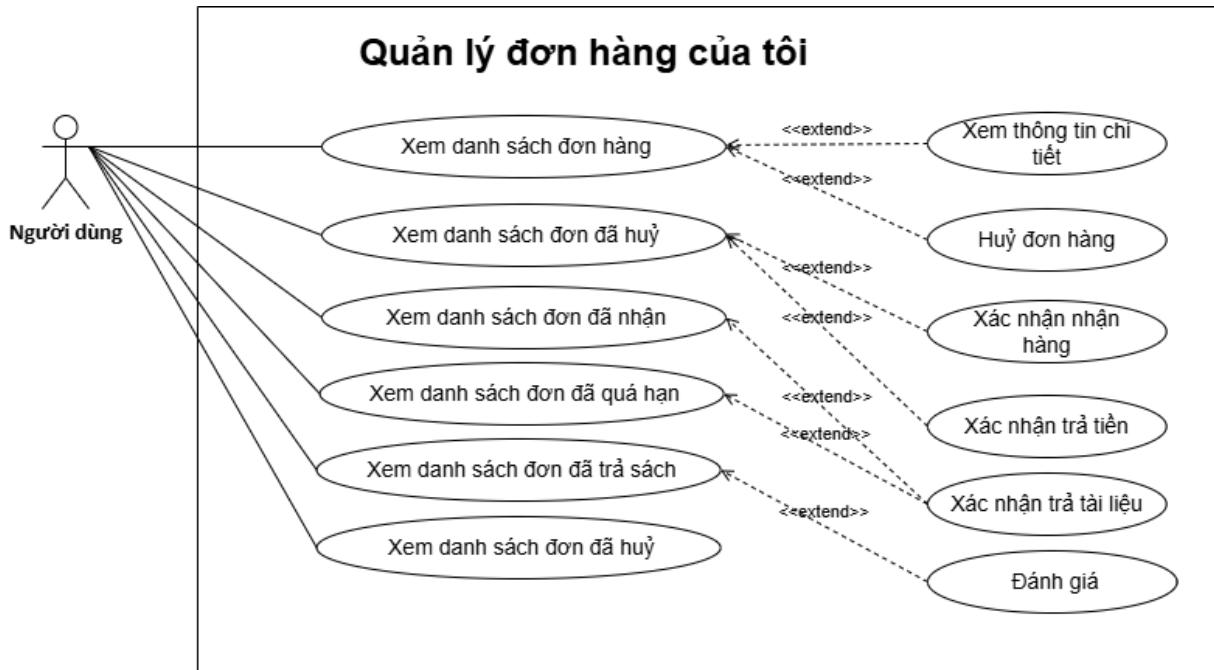
	<p>khoản. Hệ thống loại bỏ hidden của nút tạo đơn hàng</p> <ol style="list-style-type: none"> 7. Người dùng bấm “tạo đơn hàng” 8. Hệ thống tiến hành lưu đơn hàng vào hệ thống trạng thái đã đặt hàng. 9. Hệ thống hiển thị trang check-out với đơn hàng đã được tạo thành công với thông báo nhắc “Bạn hãy nhanh chóng thanh toán theo phương thức chuyển khoản để lấy hàng”
--	--

- **Đặt mua**

Use case name	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 chuyển khoản. Hệ thống loại bỏ hidden của nút tạo đơn hàng 7. Người dùng bấm “tạo đơn hàng”

	<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>
--	--

4.5.1.3 Người dùng Quản lý đơn hàng



Hình 4.5: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

- 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

- 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

- 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	1. Người dùng bấm vào tab “ĐÃ QUÁ HẠN” 2. Hệ thống hiển thị trang quản lý đơn hàng ở tab đã quá hạn 3. Hệ thống hiển thị danh sách các đơn hàng đã quá hạn
Alternative Flows	Không có
Exception	Không có

- **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	1. Người dùng bấm vào tab “ĐÃ TRẢ SÁCH” 2. Hệ thống hiển thị trang quản lý đơn hàng ở tab đã trả sách 3. Hệ thống hiển thị danh sách các đơn hàng đã trả sách
Alternative Flows	Không có
Exception	Không có

- **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"> 1. Người dùng bấm vào tab “ĐÃ HUỶ” 2. Hệ thống hiển thị trang quản lý đơn hàng ở tab đã huỷ 3. Hệ thống hiển thị danh sách các đơn đã huỷ
Alternative Flows	Không có
Exception	Không có

- **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 chuyển khoản
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	<ol style="list-style-type: none"> 1. Người dùng bấm vào nút “đã trả tiền” ở đơn hàng chuyển khoản 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

- **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	<ol style="list-style-type: none"> 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

- **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

- **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

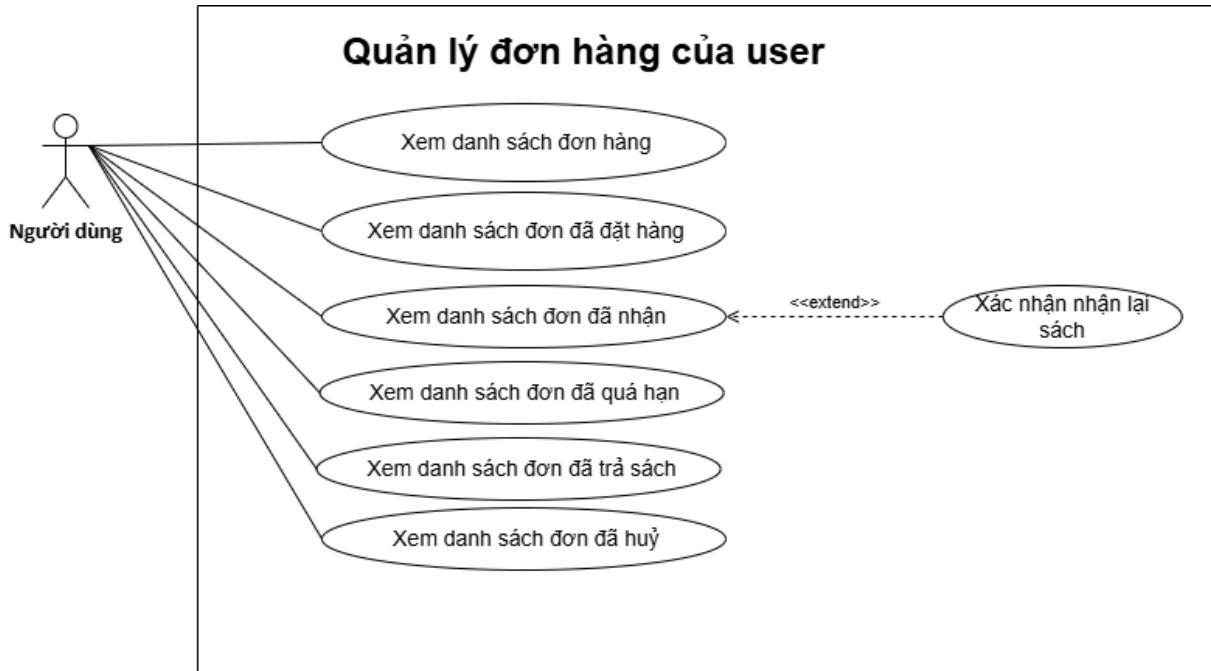
- **Huỷ đơn hàng**

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 thẻ 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

- **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

4.5.1.4 Người dùng quản lý đơn hàng của khách

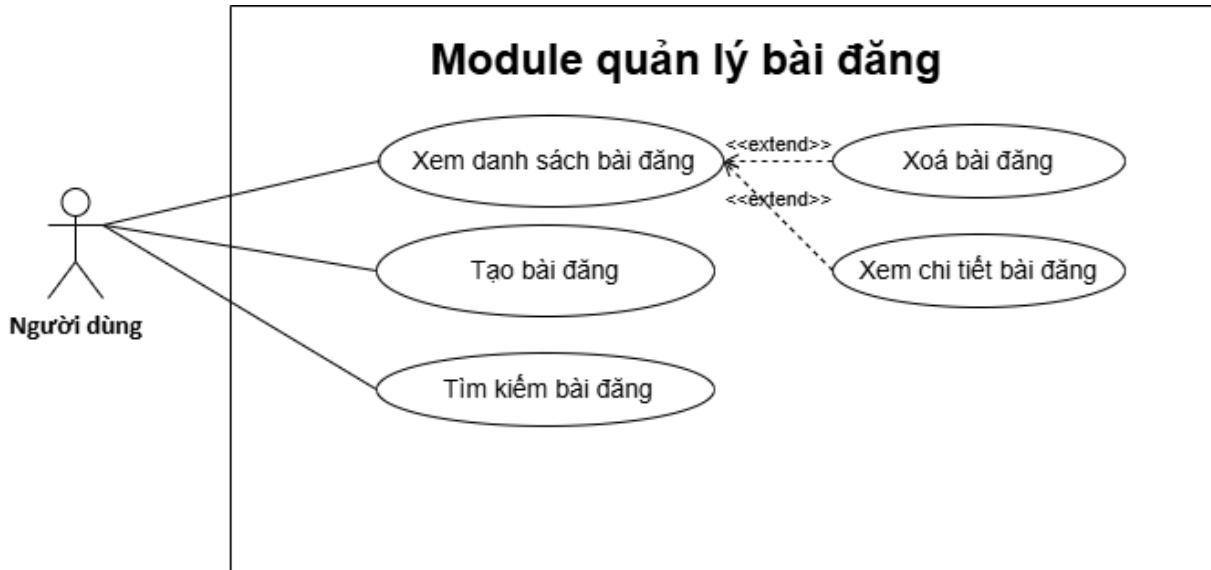


Hình 4.6 Use case quản lý đơn hàng của khách

- 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

4.5.1.5 Người dùng quản lý bài đăng



Hình 4.7:Module quản lý bài đăng

- **Xem danh sách bài đăng**

Use case name	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

- **Tạo bài đăng**

Use case name	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>

- **Tìm kiếm bài đăng**

Use case name	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

- **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

4.5.2 Use Cases chung cho Quản trị viên và Người dùng

4.5.2.1 Đăng nhập

Use case name	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 6. Hệ thống ghi nhận hoạt động đăng nhập thành công

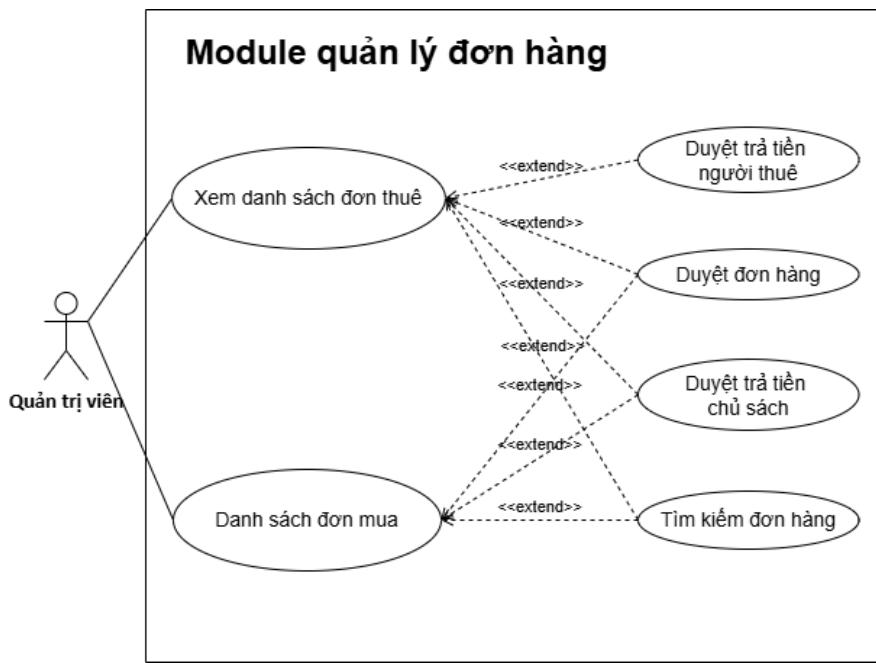
	7. Hệ thống hiển thị trang chủ sau khi đăng nhập
Exception	<p>Exception 1: Tại bước 5</p> <p>5a. Kiểm tra thông tin đăng nhập chưa đúng 6a. Hệ thống hiển thị lại form đăng nhập, và thông báo mật khẩu hoặc tên đăng nhập chưa đúng Quay lại và tiếp tục với bước 3</p>
Alternative Flows	Không có

4.5.2.2 Đăng xuất

Use case name	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 nhấp vào icon “Tài khoản” Hệ thống hiển thị danh sách các nút bấm Người dùng/Quản trị viên nhấp 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

4.5.3 Use Cases cho Quản trị viên

4.5.3.1 Quản trị viên quản lý đơn hàng



Hình 4.8:Module quản lý đơn hàng-quản trị viên

- Xem danh sách đơn thuê

Use case name	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

- Xem danh sách đơn hàng mua

Use case name	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

- **Duyệt đơn hàng**

Use case name	Tìm kiếm 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

- **Duyệt trả tiền người thuê**

Use case name	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	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

- **Duyệt trả tiền người bán**

Use case name	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"> Quản trị viên truy cập vào trang quản lý đơn hàng Hệ thống hiển thị trang quản lý đơn hàng Quản trị viên bấm vào nút duyệt trả tiền người bán Hệ thống tiến hành thay đổi trạng thái đơn hàng
Exception	Không có
Alternative Flows	Không có

4.5.4 Use Cases chung cho Người dùng và Khách

4.5.4.1 Đăng ký tài khoản

Use case name	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"> Khách nhấp nút “Đăng ký” Hệ thống hiển thị trang đăng ký chưa form đăng ký Khách điền thông tin cá nhân được yêu cầu Khách bấm nút “Tạo tài khoản” 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

4.5.4.2 Xem danh sách đề xuất

Use case name	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	1. Tại trang chủ của website, Người dùng/Khách thấy danh sách đề xuất
Exception	Không có
Alternative Flows	Không có

4.5.4.3 Chi tiết bài đăng

Use case name	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	1. Tại danh sách tài liệu, Người dùng/Khách nhấp 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

4.5.4.4 Tìm kiếm tài liệu

Use case name	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 kiểm 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"> 1. Người dùng/Khách thấy nút tìm kiếm 2. Hệ thống hiển thị ô input để người dùng/khách nhập từ khoá 3. Người dùng/Khách được nhập từ khóa 4. Người dùng/Khách nhấn nút tìm kiếm 5. 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"> 1. Người dùng/khách nhấn icon thẻ loại bên trái input tìm kiếm 2. Hệ thống hiển thị danh sách thẻ loại 3. Người dùng nhấn chọn thẻ loại 4. Hệ thống hiển thị danh sách tài liệu thuộc thẻ loại tìm kiếm

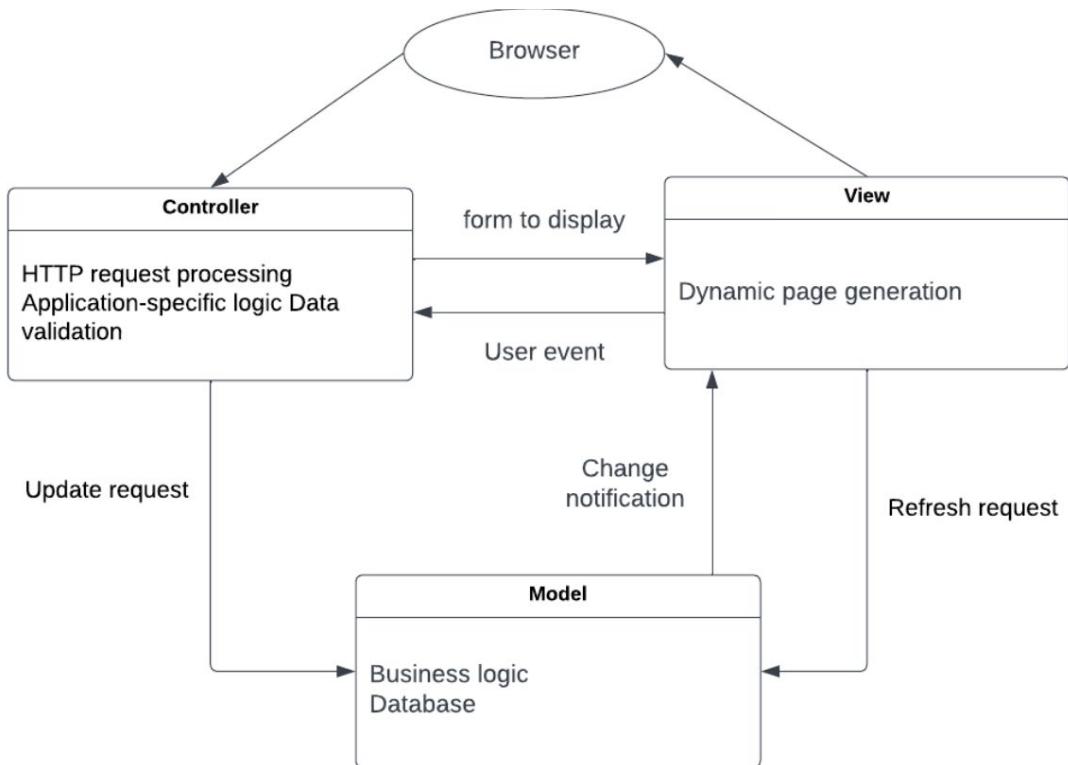
4.5.4.5 Thêm vào giỏ hàng

Use case name	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

4.5.4.6 Xem giỏ hàng

Use case name	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

4.6 Mô hình kiến trúc hệ thống



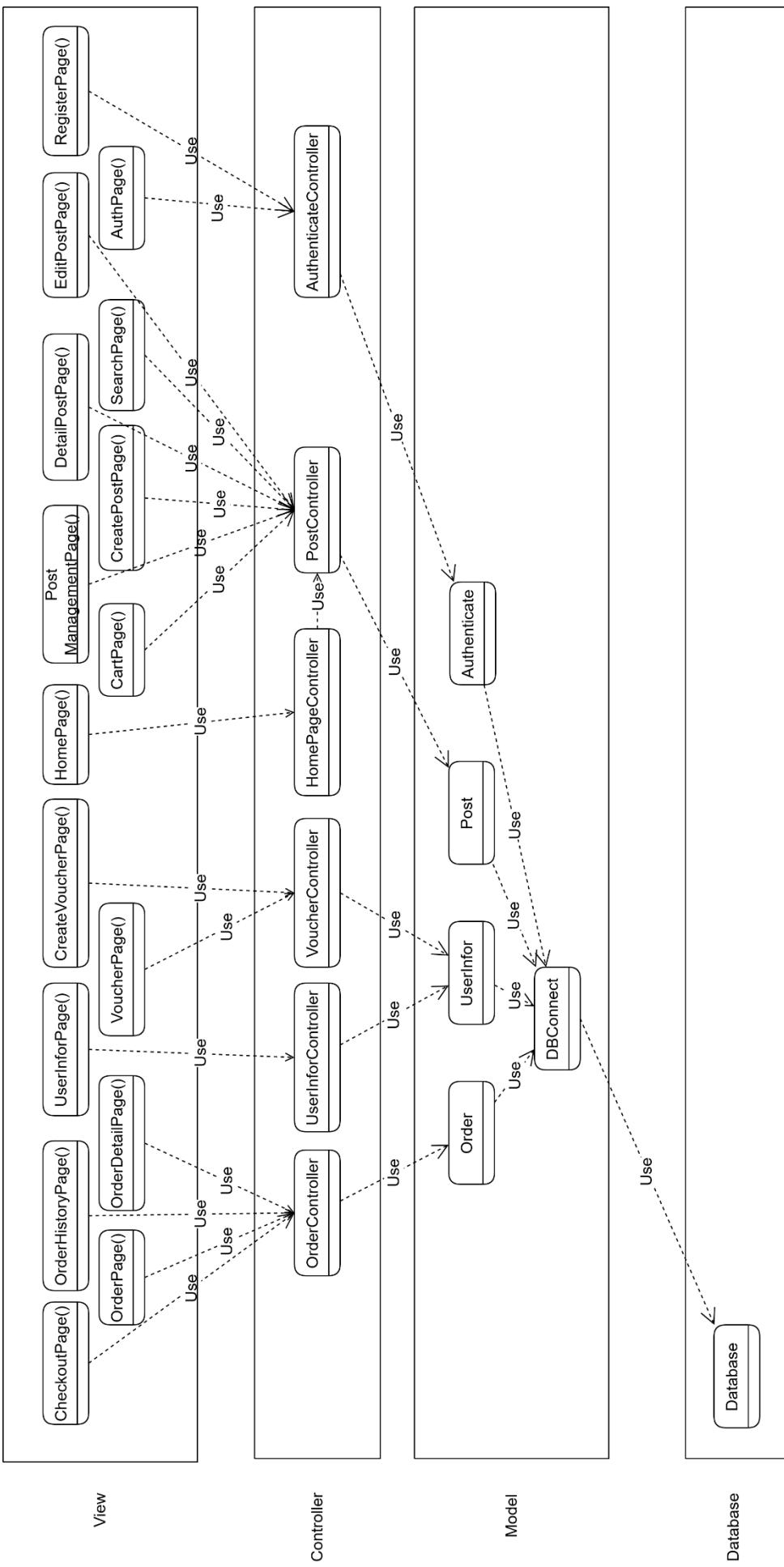
Hình 4.9: Mô hình kiến trúc hệ thống MVC

- 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*



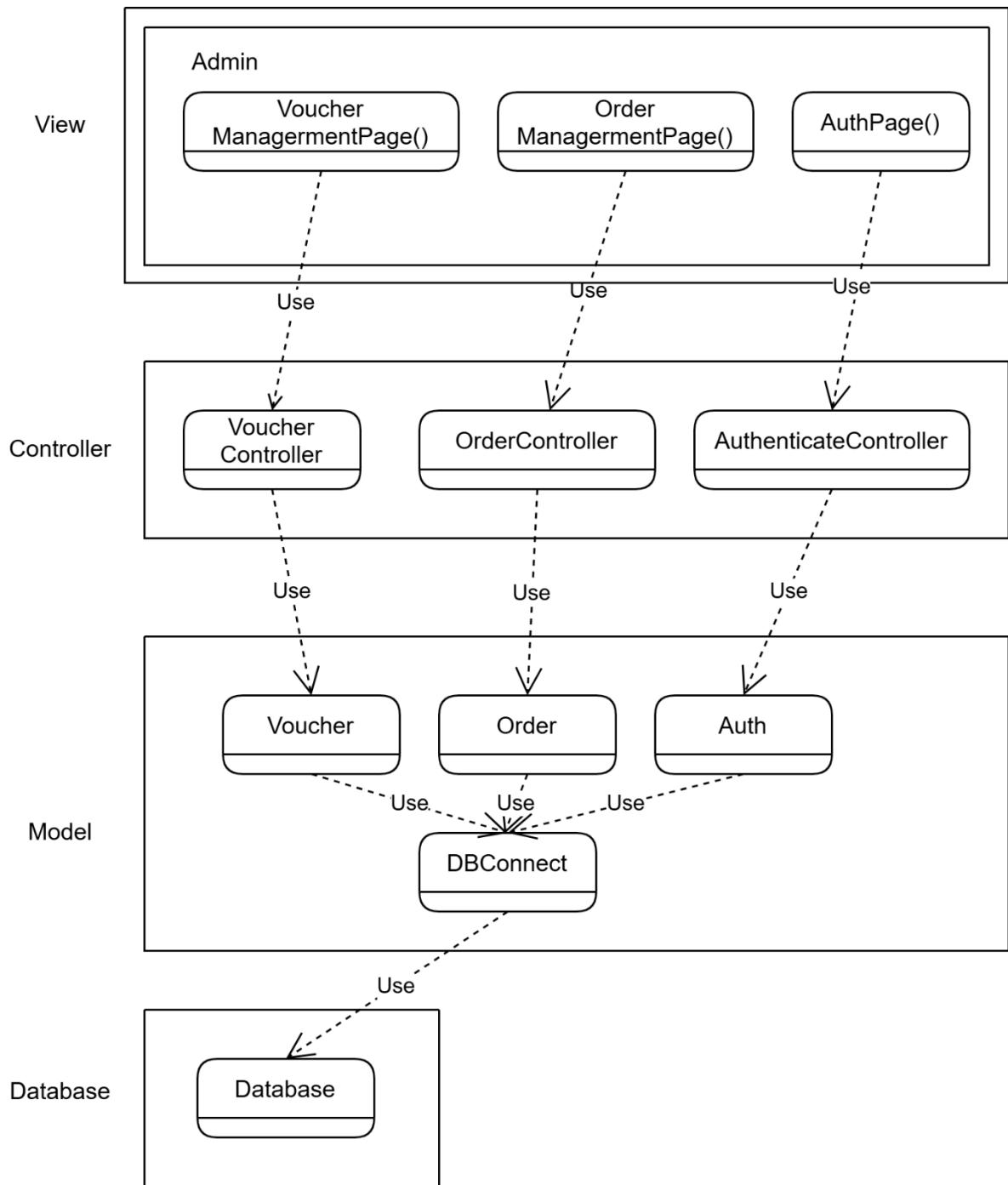
Hình 4.10: Layered architecture diagram cho người dùng và khách

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

Mô tả chi tiết các lớp như sau:

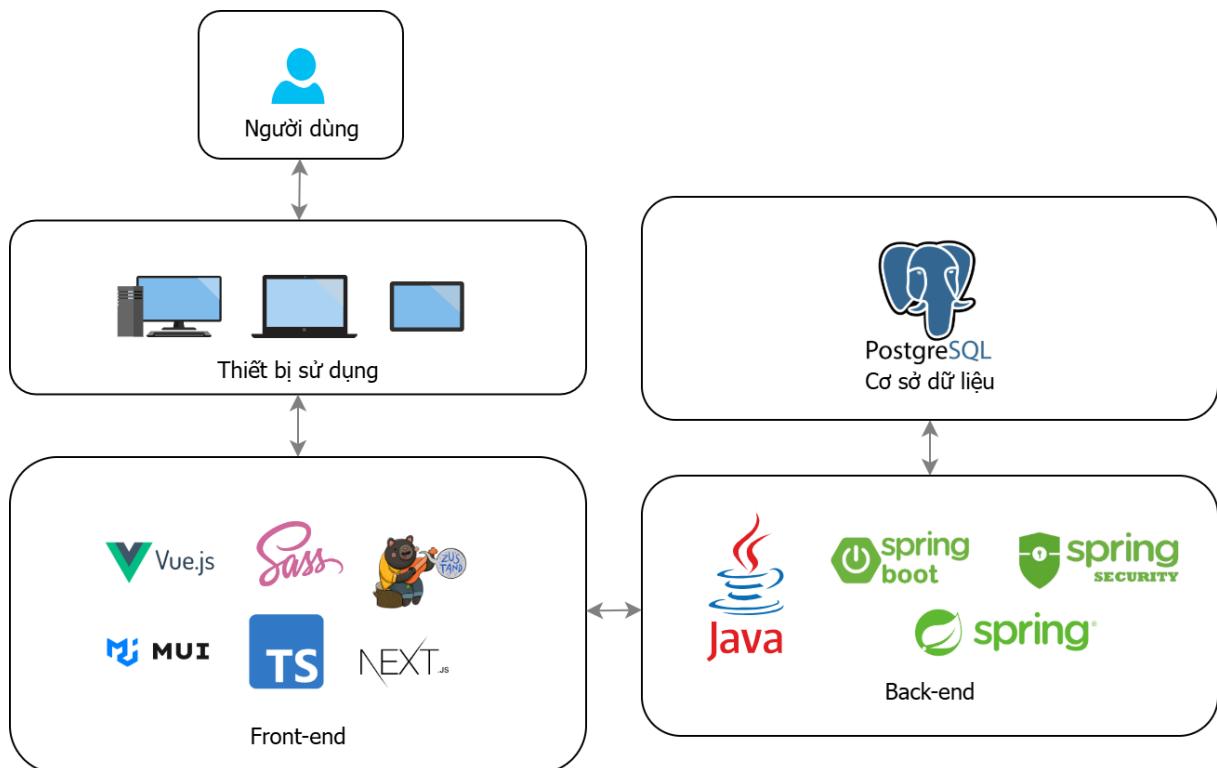
- Lớp View: là nền tảng website hiển thị giao diện người dùng trực quan, mỗi đối tượng trong View đại diện cho một đến nhiều trang trong website thực hiện chức năng hiển thị thông tin và các chức năng người dùng tương tác với hệ thống.
- Lớp Controller: cung cấp các chức năng về hiển thị và quản lý cho 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 4.11: Layered architecture diagram cho quản trị viên

4.8 Deploy diagram



Hình 4.12: Deploy diagram

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:

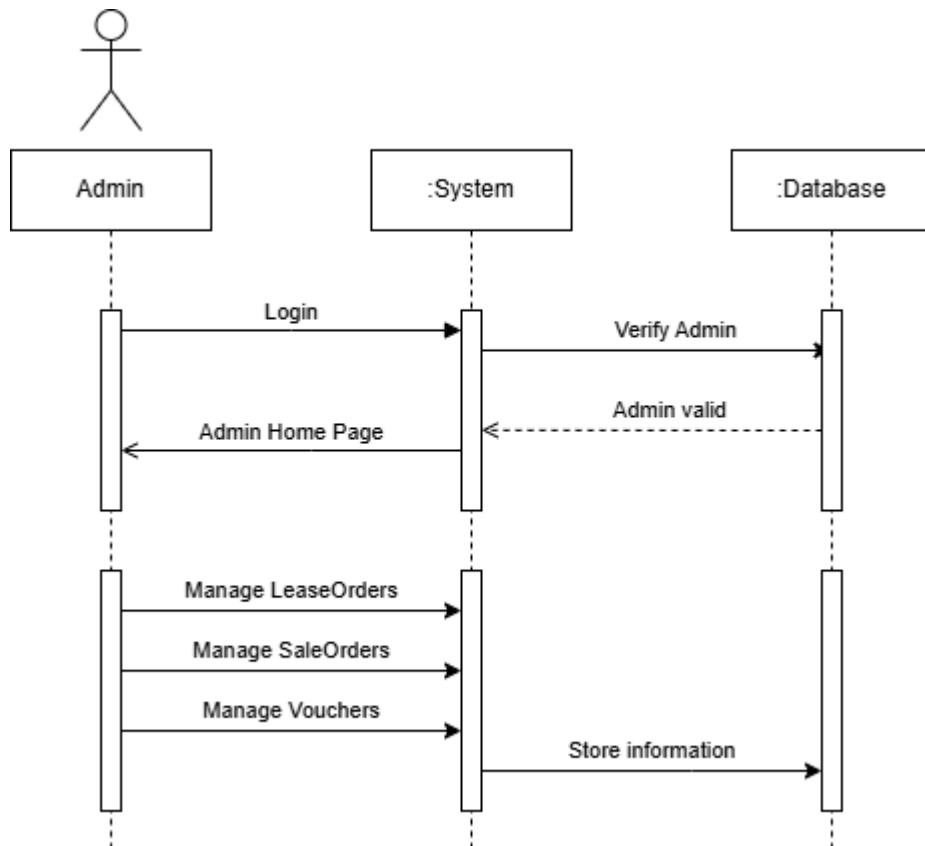
- PostgreSQL: Hệ quản trị cơ sở dữ liệu quan hệ được sử dụng để lưu trữ và quản lý dữ liệu.

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

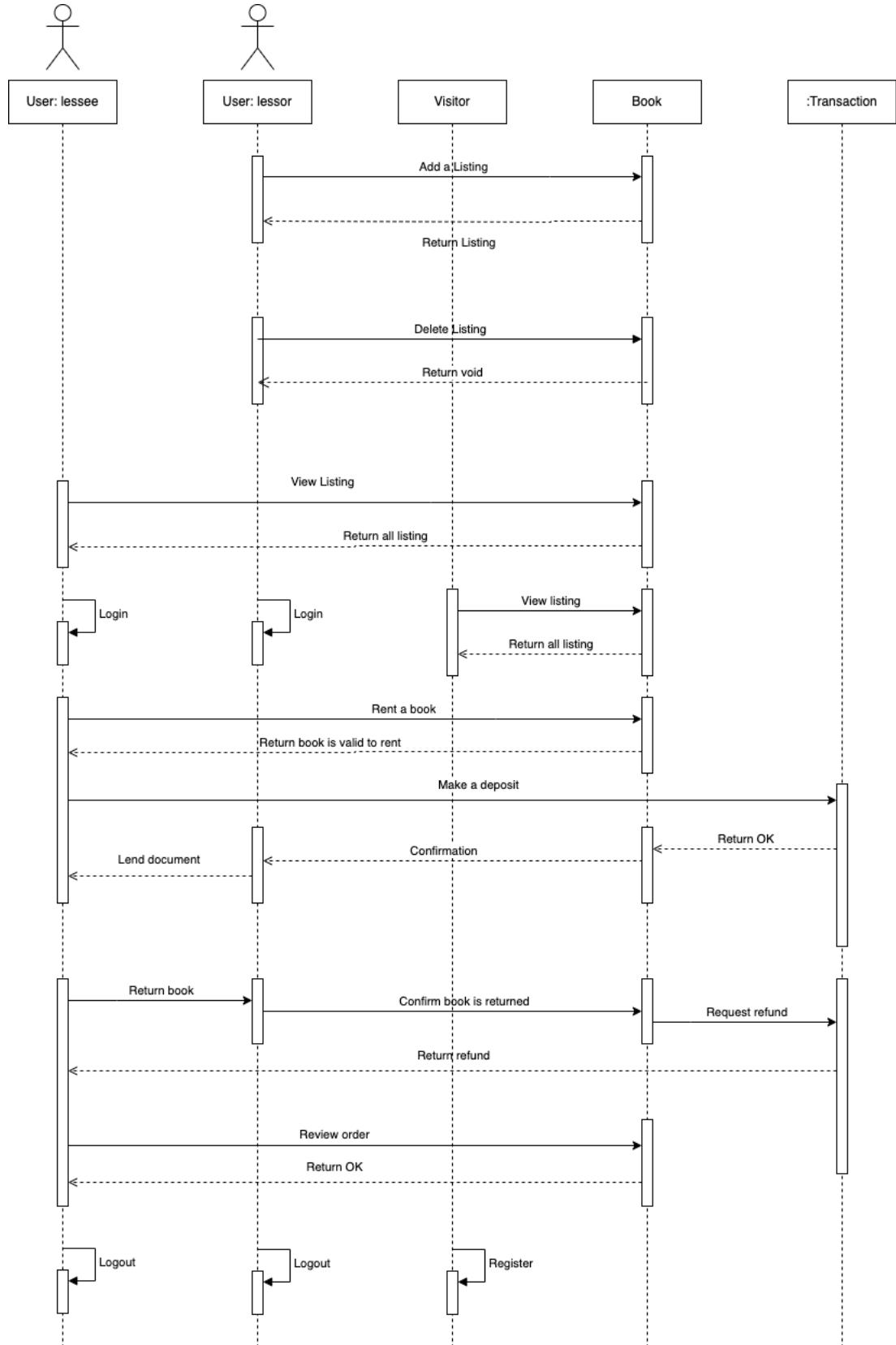


Hình 4.11: Sequence diagram cho Quản trị viên

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



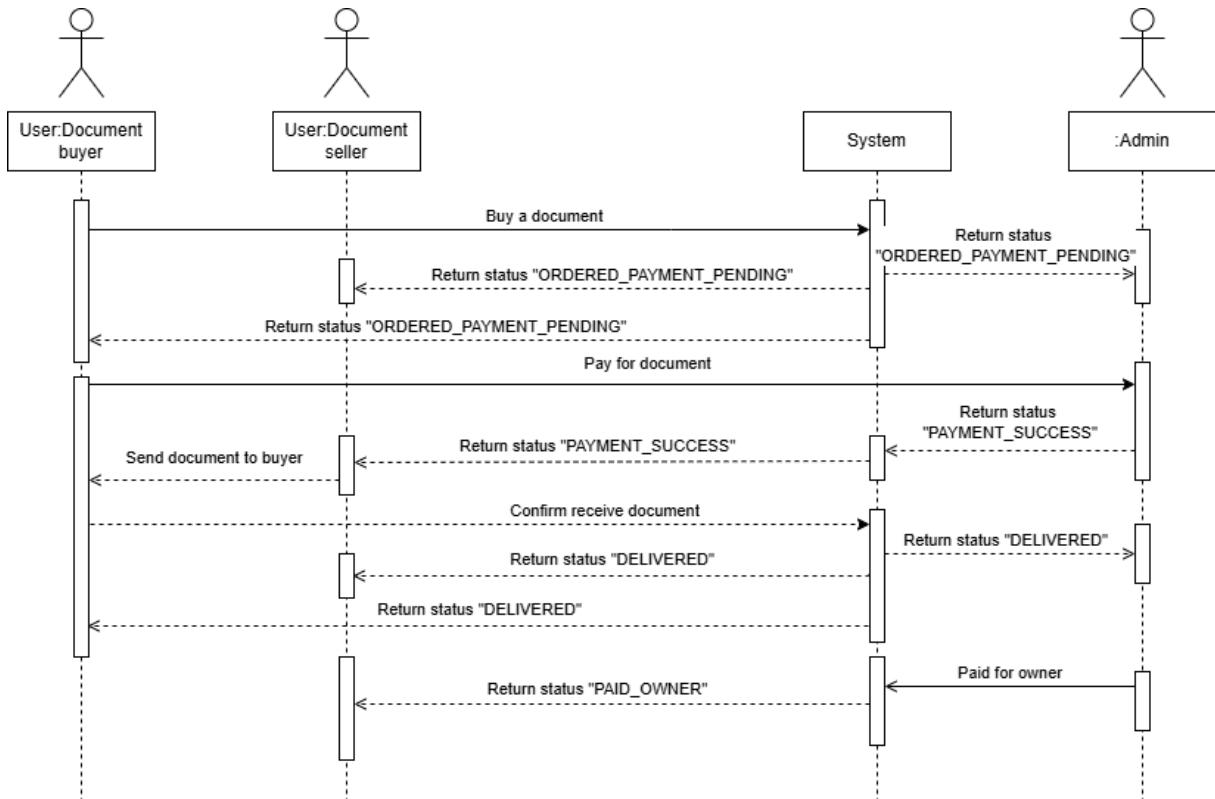
Hình 4.12: Sequence diagram cho người dùng và khách thuê sách

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



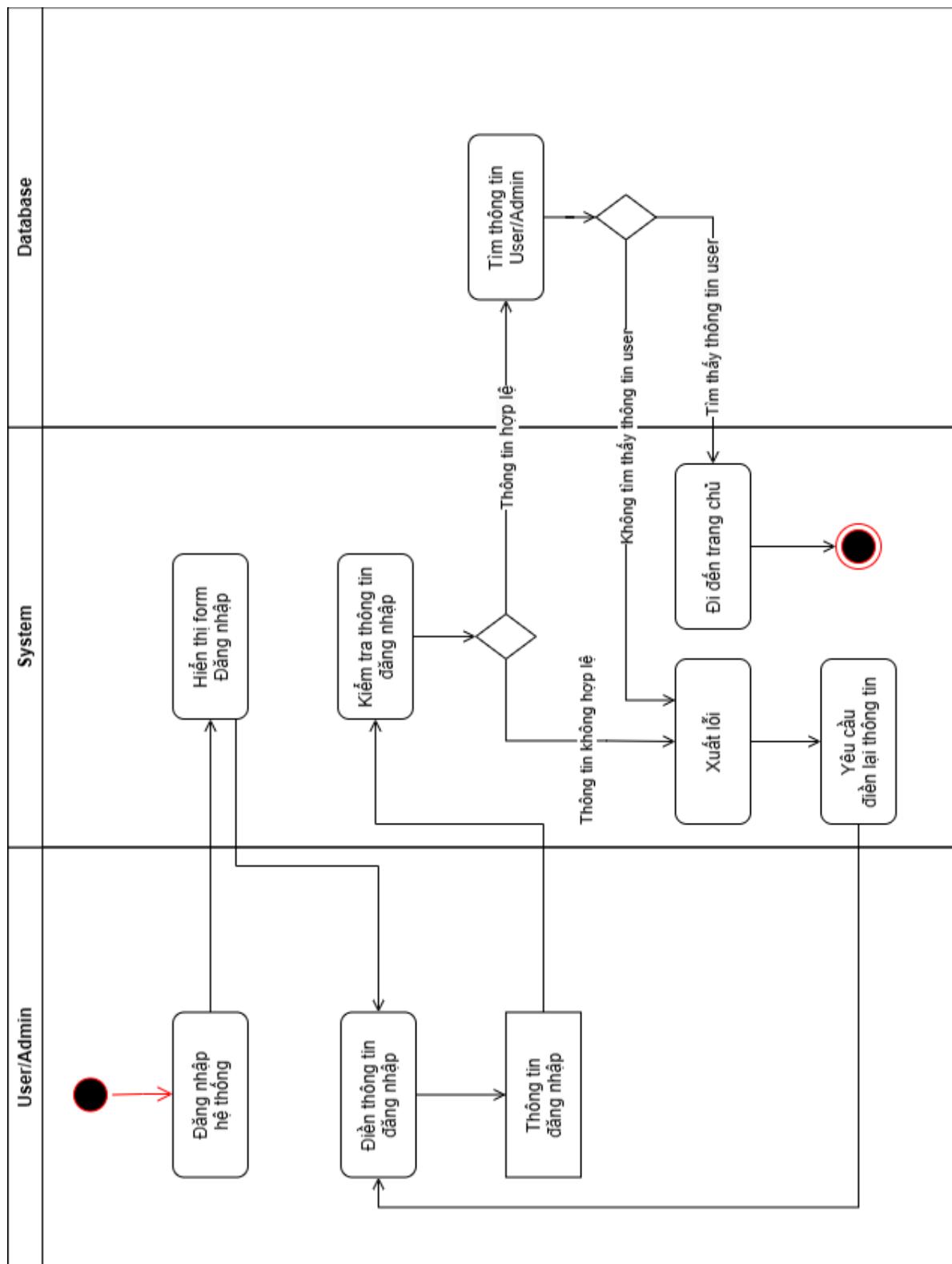
Hình 4.13: Sequence diagram với actor người mua và bán sách

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 14: 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.

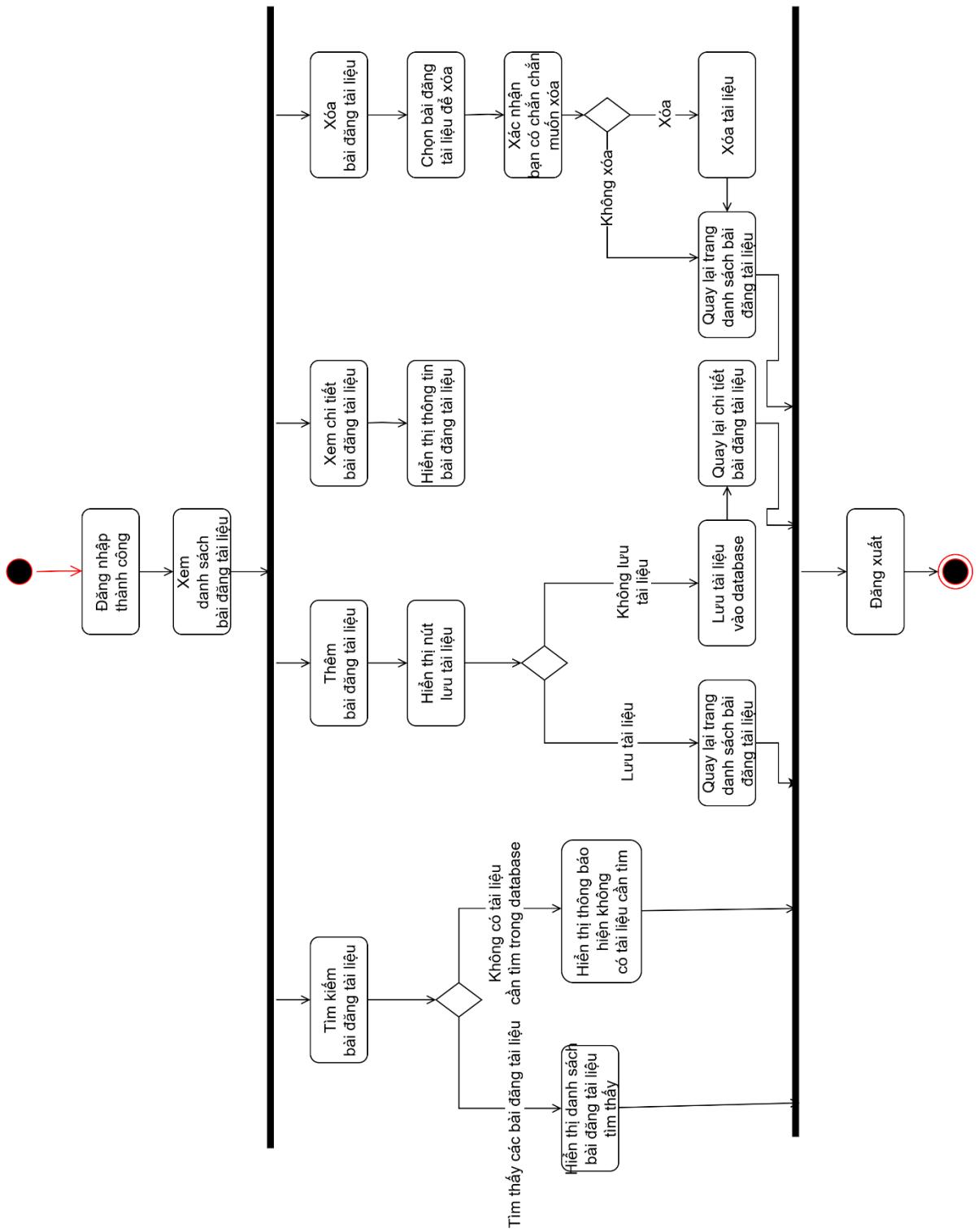
2. Nhập thông tin đăng nhập:

- 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:
 - **Nếu thông tin đăng nhập không hợp lệ:**
 - Hệ thống thông báo lỗi và yêu cầu người dùng nhập lại thông tin.
 - **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:
 - **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.
 - **Nếu tìm thấy thông tin:**
 - Tiến hành chuyển đến bước tiếp theo.
6. **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.
7. **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



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:
 - Nếu tìm thấy bài đăng: Hệ thống hiển thị danh sách các bài đăng phù hợp với tiêu chí tìm kiếm.
 - 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:
 - Hệ thống hiển thị giao diện thêm bài đăng mới và nút lưu tài liệu.
 - Người dùng có thể quyết định:
 - **Không lưu tài liệu:** Hệ thống quay lại trang danh sách bài đăng tài liệu.
 - **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:
 - Hệ thống hiển thị thông tin chi tiết bài đăng tài liệu.
 - 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:
 - Hệ thống hiển thị yêu cầu xác nhận việc xóa.
 - Nếu người dùng xác nhận xóa: Hệ thống thực hiện xóa bài đăng khỏi cơ sở dữ liệu.
 - Nếu người dùng không xác nhận: Hệ thống quay lại trang danh sách bài đăng tài liệu.

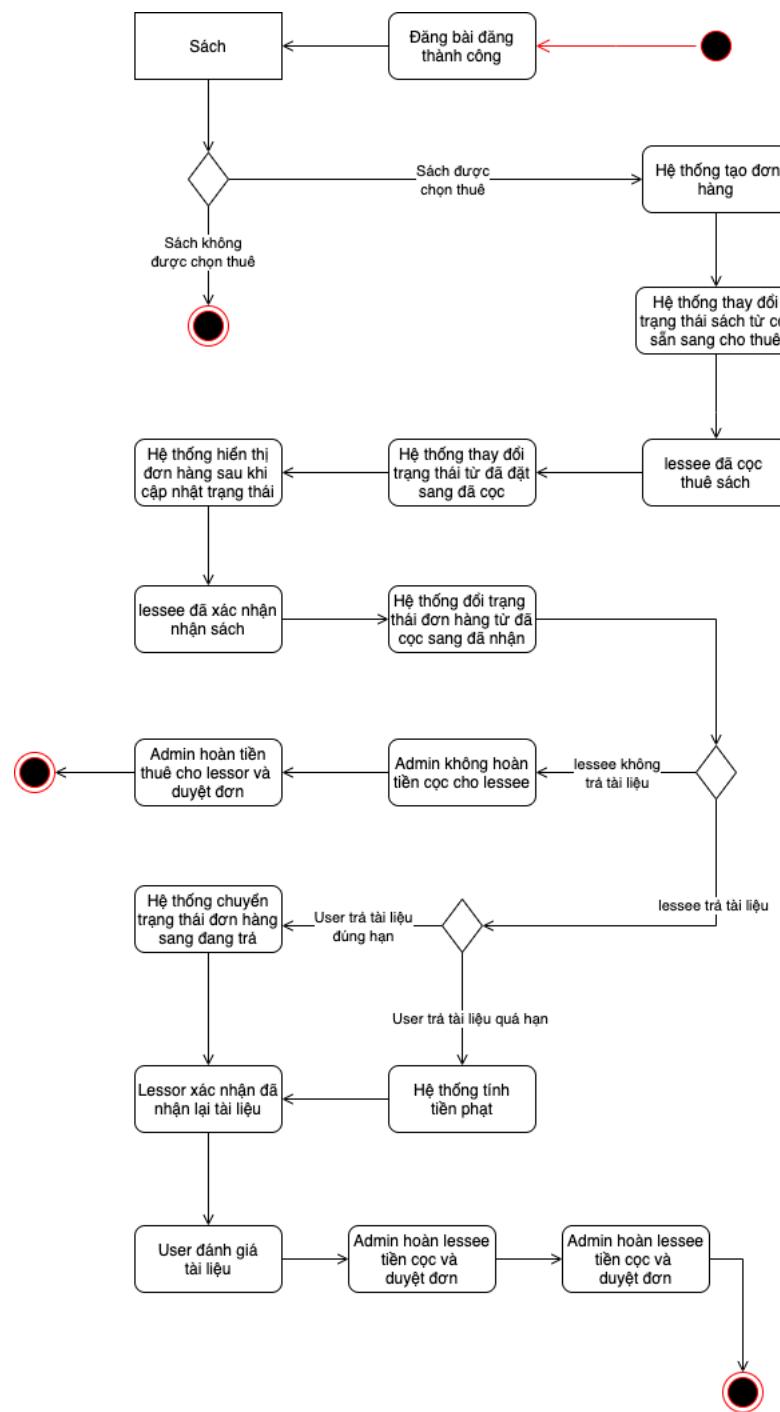
6. Kết thúc:

- 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 4.16: 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:

- Nếu sách không được thuê: Kết thúc hoạt động.
- Nếu sách được thuê bởi người thuê:
 - Hệ thống tạo đơn hàng.
 - 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:
 - Nếu không đóng cọc:
 - Đơn hàng bị hủy.

- Sách quay về trạng thái *sẵn có*.
- **Nếu đóng cọc thành công:**
 - 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:

- **Nếu không trả sách:**
 - Quản trị viên không hoàn trả tiền cọc cho người thuê.
 - Quản trị viên tiến hành thanh toán tiền thuê sách cho chủ sách.
- **Nếu trả sách:**
 - **Trả sách đúng hạn:**
 - Tiến hành bước đánh giá tài liệu.
 - **Trả sách trễ hạn:**
 - Hệ thống tính phí trễ hạn.
 - 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.

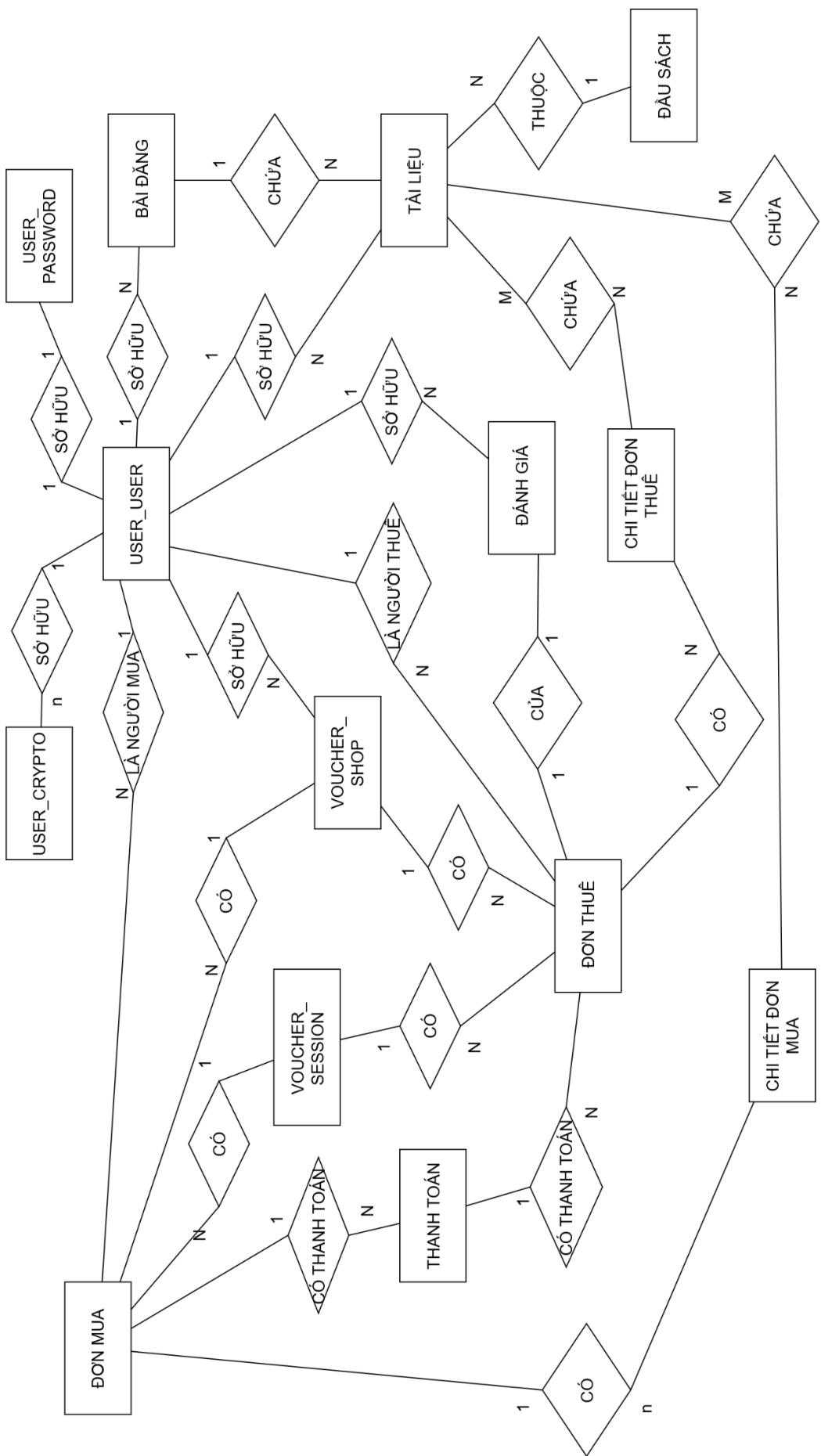
4.11 Lược đồ quan hệ cơ sở dữ liệu (ERD)

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_session và voucher_shop

- **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 chúa 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** chúa voucher của người bán chúa id, giá trị tối thiểu, mức giảm theo phần trăm, loại voucher, mức giảm theo tiền, tên, ngày bắt đầu, ngày kết thúc.

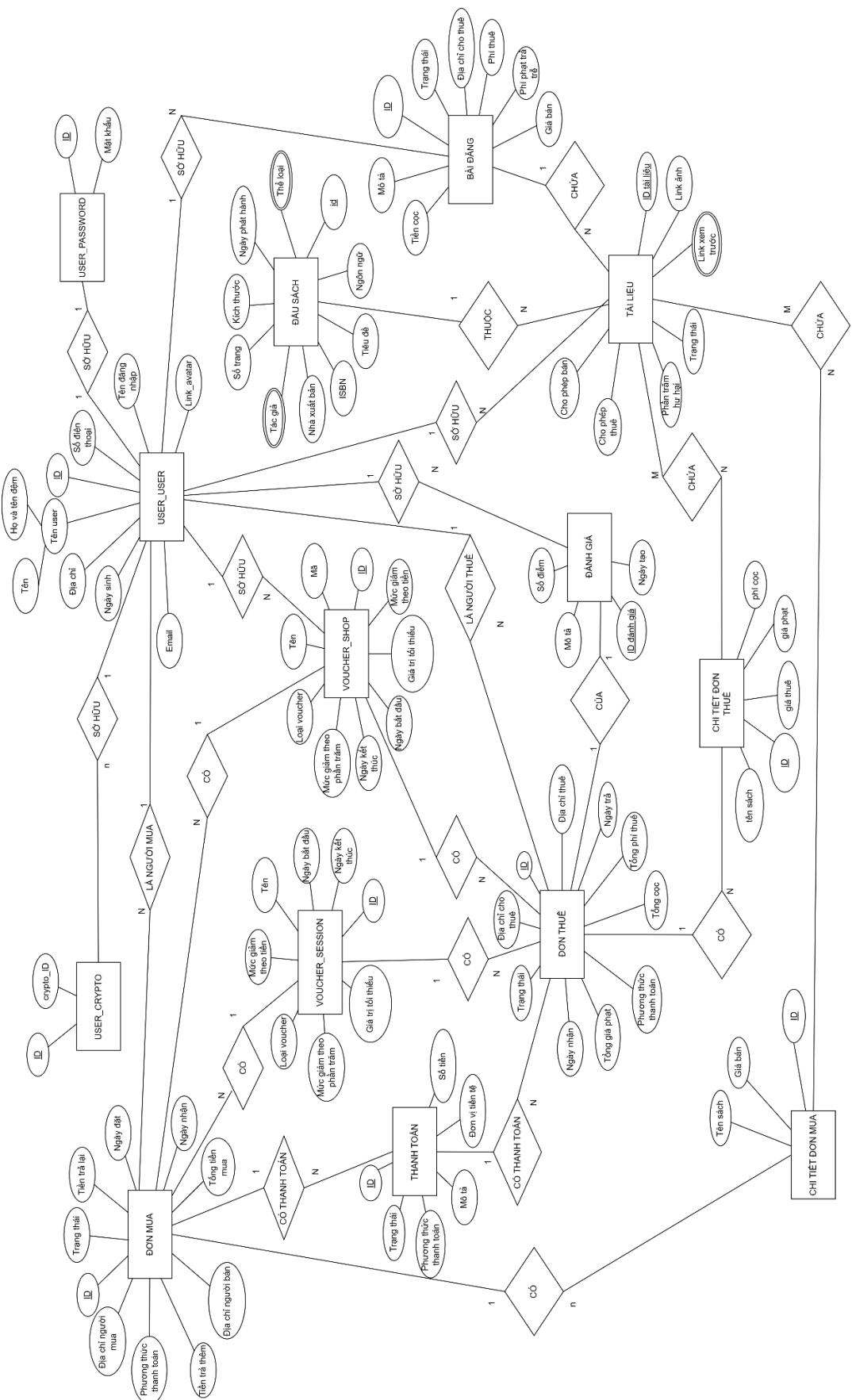
4.11.1 Lược đồ ERD vắn tắt



Hình 17

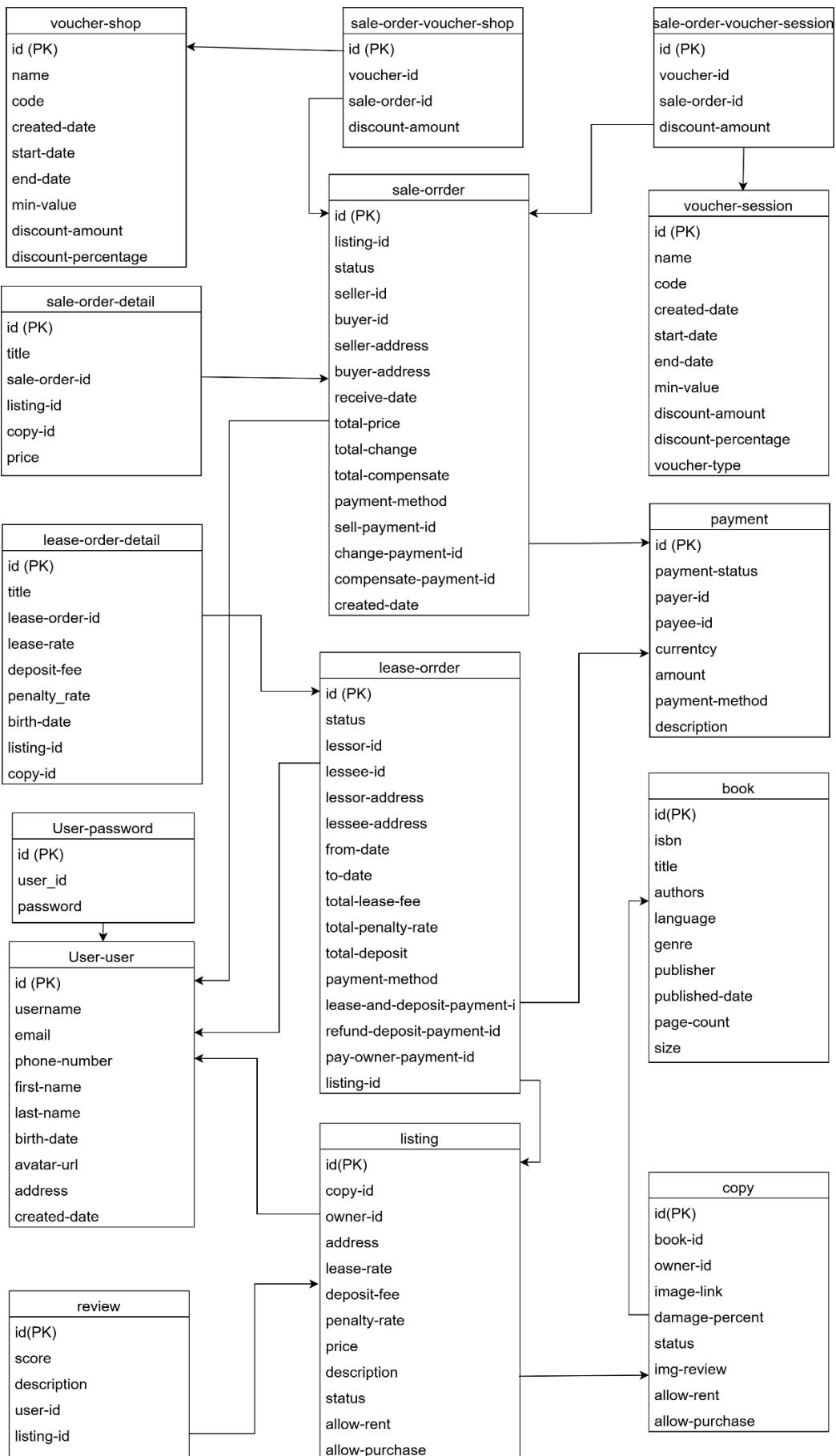
Lược đồ ERD vắn tắt

4.11.2 Lược đồ ERD đầy đủ



Hình 4.18: ERD cho toàn hệ thống

4.11.3 Lược đồ ERD implement



Hình 19: Lược đồ ERD implement

1. Các thực thể chính và quan hệ

User và User-password

- **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 và Copy

- **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 và Sale-order-detail

- **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 và Lease-order-detail

- **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 và Sale-order-voucher-shop

- **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.

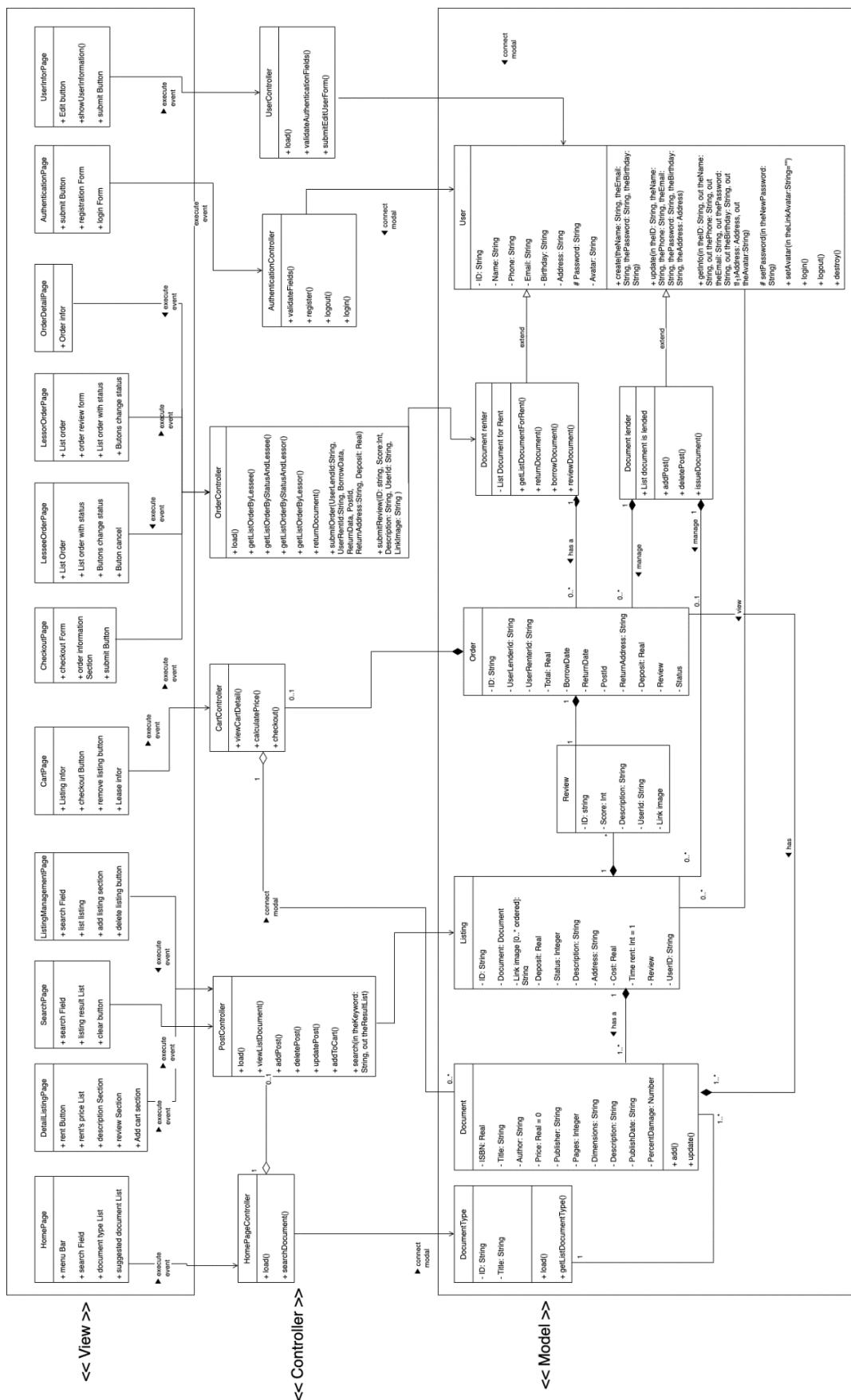
Voucher-session và Sale-order-voucher-session

- **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 4.20: Classdiagram 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, Người dùngController
- 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.yalm

```
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 innit 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`.

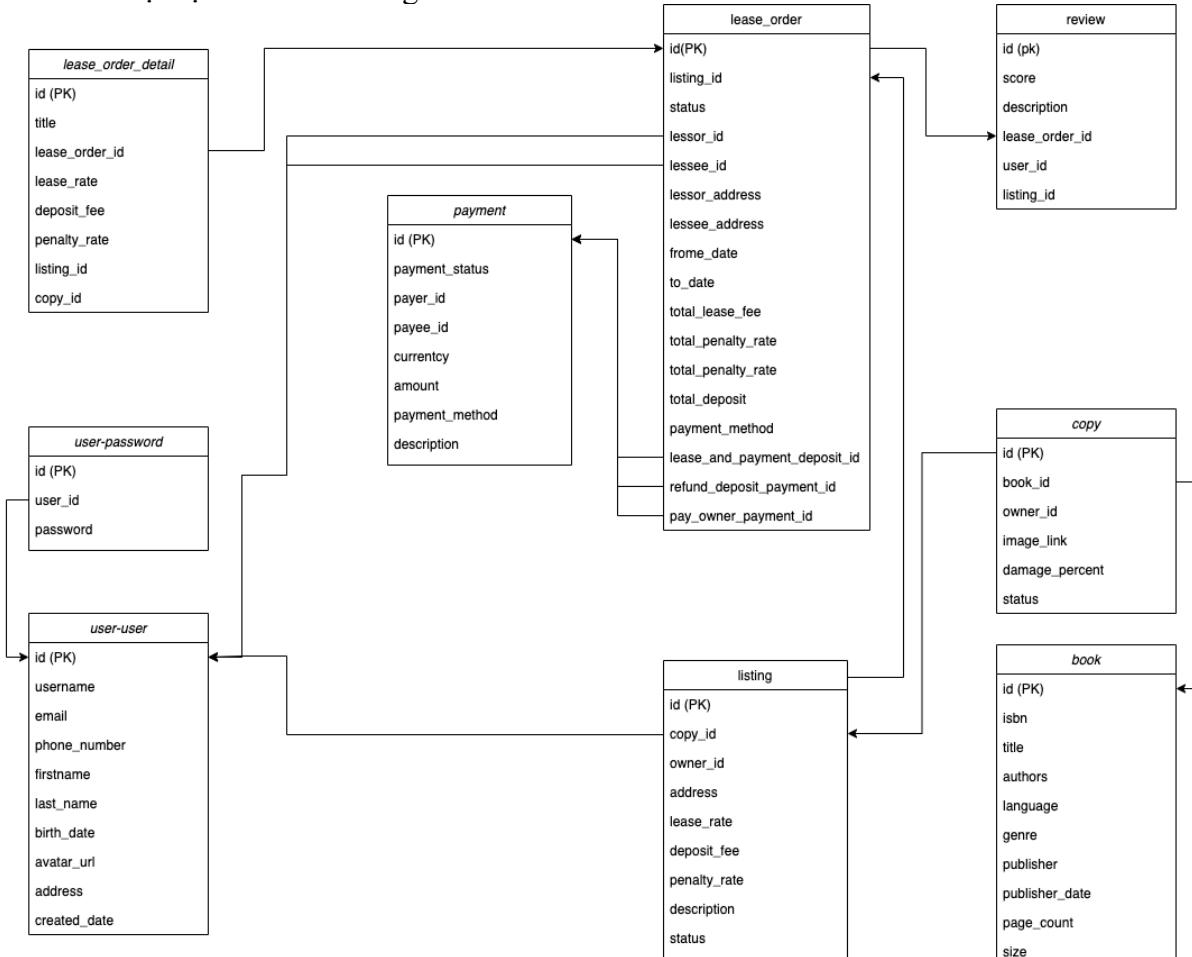
- **voucher.sql:**

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`

Cơ sở dữ liệu tạo thành các bảng như sau:



Hình 5.1: ERD cơ sở dữ liệu init

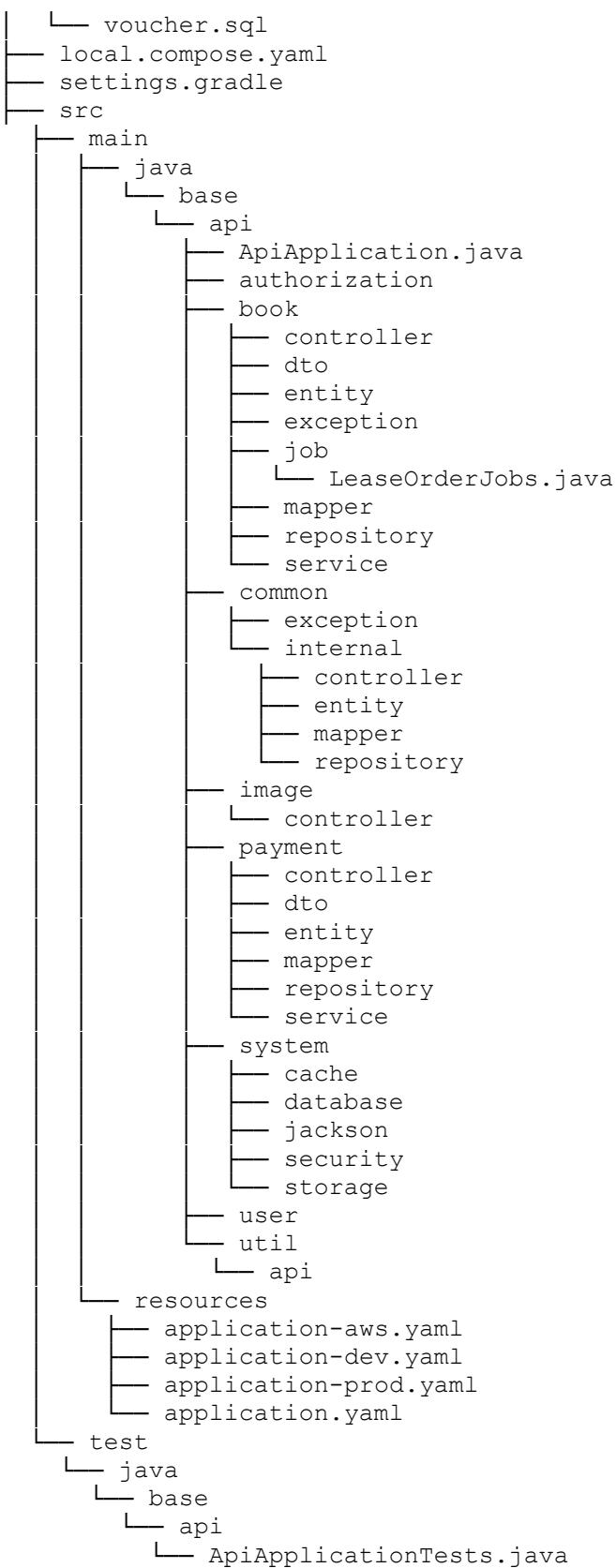
5.2 Phân tích source-code back-end

Sử dụng <https://github.com/structure-codes/cli> để lấy cấu trúc source dễ nhìn đắt 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

```



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:
 - voucher.sql: Script SQL để tạo bảng hoặc dữ liệu liên quan đến voucher.

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*

java/

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 *Thu mục test*

- Chứa các bài kiểm thử cho ứng dụng:
 - **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 nhóm em 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 entity đại diện cho cấu trúc dữ liệu trao đổi qua api, còn DTO 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 thể 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 nhóm em 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 bị 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	Sale order	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 5.1: Api thực hiện 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 phân 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 chuyền 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 *Thu 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
```

6.1.3 *Thu mục app*

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

```

Trong thư mục app này sẽ chứa các thành phần đại diện cho các trang (pages) chính của ứng dụng.

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
    └── nhat-ky-am-ap-con-duong-di-dao-conduongdidao-jacket.jpg
    └── 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
    └── scrollButton
        └── ScrollButton.tsx
    └── search
        ├── Filter.tsx
        ├── HeaderListBook.tsx
        ├── ListSearchBook.tsx
        └── SearchedBook.module.css
    └── voucher
        └── column.tsx

```

auth/

- Chứa các component liên quan đến xác thực người dùng:
 - FormLogin.tsx, FormSignup.tsx: 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: 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: 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
├── cart.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 6.1: Phiên bản logo không chữ

Sau đây là phiên bản logo có chữ:



Hình 6.2: Logo với tên dự án The Flying Bookstore

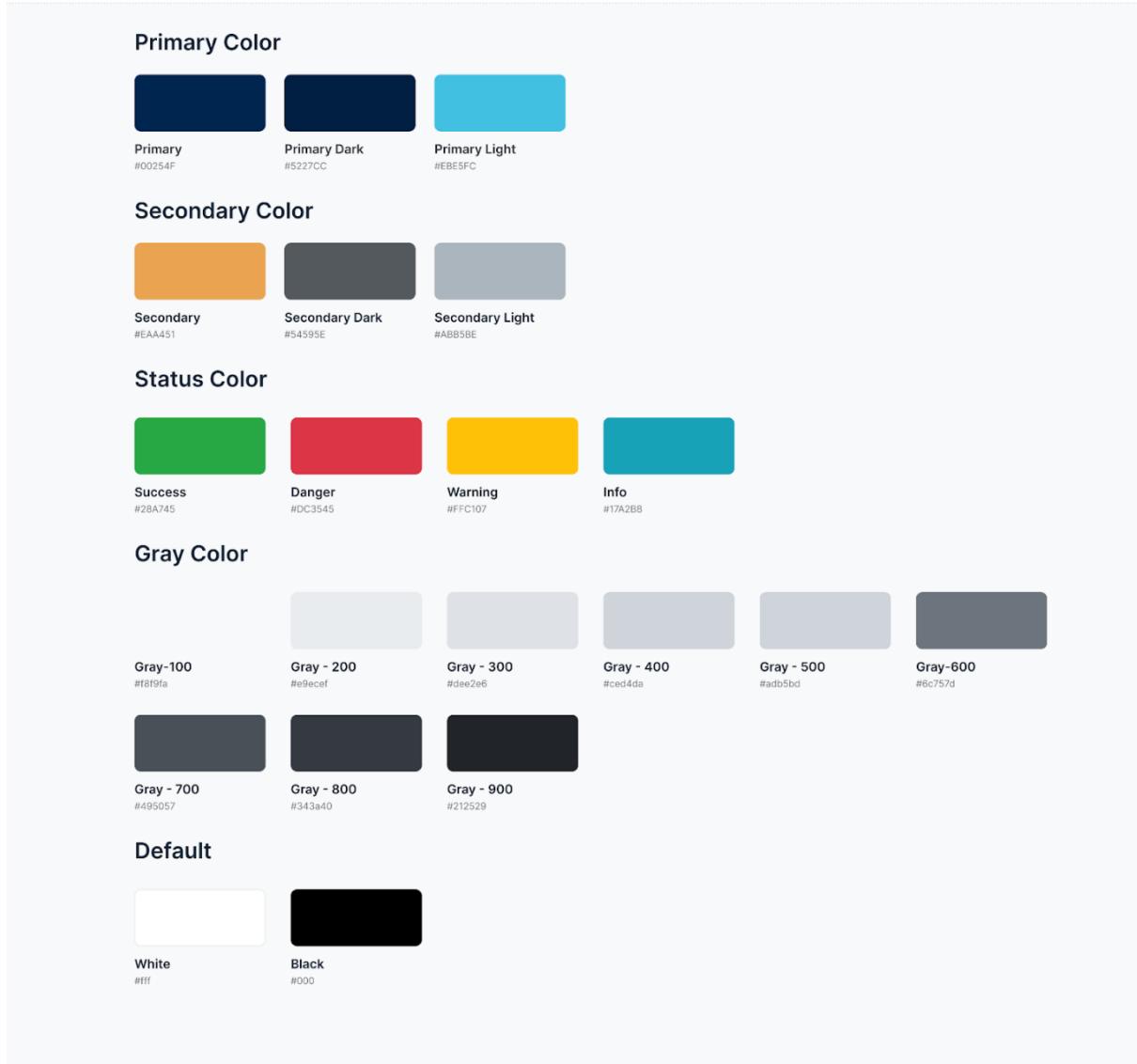
6.2.2 Màu sắc

Màu sắc áp dụng trong logo được áp dụng trên toàn bộ website và tạo thành các biến, trong đó em có bản 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: Bảng hệ thống màu

6.2.3 *Font chữ*

Font chữ chính: Montserrat

Font chữ phụ: Open sans

6.2.4 *Component chung*

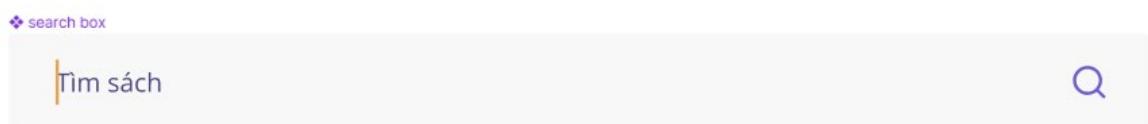
6.2.4.1 *Header*

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



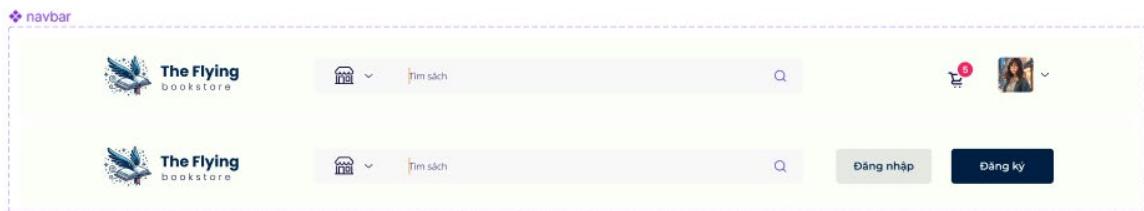
Hình: Component list menu

Em tạo thêm search box component cho riêng ô search



Hình: component search

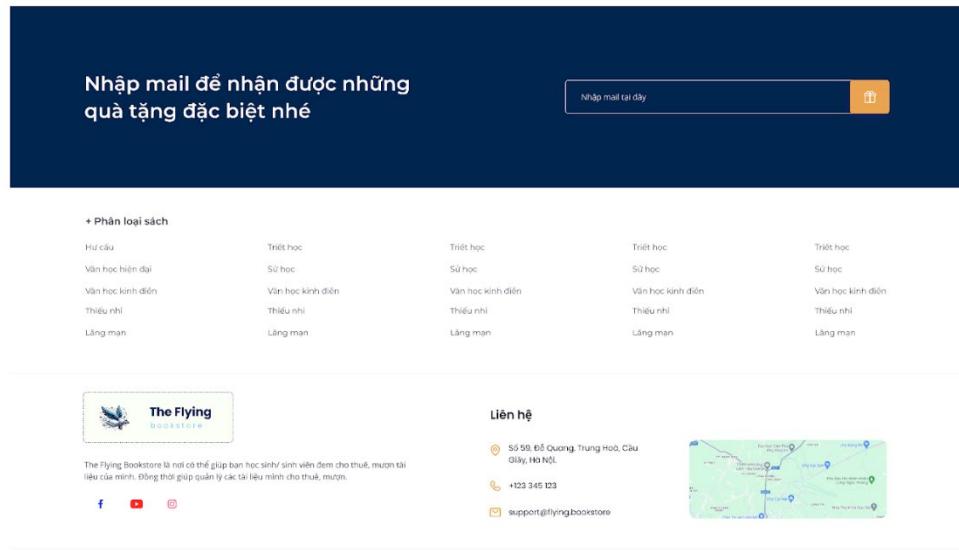
Gom tất cả component từ trên lại, em có một header với navbar hoàn chỉnh như sau, có 2 trạng thái cho navbar này là đăng nhập rồi (ở trên) và chưa đăng nhập (ở dưới)



Hình: 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: Footer component

6.3 Giao diện

Các bộ phận 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

*6.3.1.1 *Tổng quan**

*6.3.1.2 *Quy tắc và Ràng buộc**

*6.3.1.3 *Mô tả chi tiết**

*6.3.1.4 *Quy trình xử lý**

*6.3.1.5 *API Mô tả**

6.3.2 Trang chủ

The screenshot shows the homepage of The Flying Bookstore. At the top, there's a navigation bar with icons for search, cart, and account. The main header features a large image of a woman in a traditional Vietnamese ao dai holding books, with the text "BÁN CHẠY" and "Danh nhân Việt Nam". Below this, there's a section for "Doanh Kế Thiện #Văn học Việt Nam" with a brief description and a "Thuê ngay" button. A prominent offer "5.000đ/ngày" is displayed. To the right of the main image are four service icons: "Giao hàng nhanh", "Bảo mật thanh toán", "Chất lượng đỉnh nhất", and "Chính sách rõ ràng". Below these are sections for "Sách mới ra mắt" (with books like "Ice" by Ed McBain, "Companions" by Susan Sizemore, "Alchemy and Academe" by Anne McCaffrey, "Houses of Stone" by Barbara Michaels, and "Hide and Seek" by Fern Michaels) and "Danh mục sản phẩm" (with categories like Giật gân, Học thuật, Hồi ký, Kinh điển, Hài kịch, and truyện tranh). A dark blue footer bar at the bottom encourages users to enter their email for special offers.

Hình 6.3: Trang chủ

6.3.2.1 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.

6.3.2.2 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.

6.3.2.3 Mô tả chi tiết

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

6.3.2.4 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.

6.3.2.5 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:
 - o size (int, bắt buộc): Số lượng kết quả mỗi trang.
 - o page (int, bắt buộc): Trang hiện tại.
 - o title (String, optional): Tiêu đề sách.
 - o allowRent (int, bắt buộc): Cho phép thuê (1 là true, 0 là false).
 - o allowPurchase (int, bắt buộc): Cho phép mua (1 là true, 0 là false).

6.3.3 Trang tìm kiếm và hiển thị danh sách bài đăng

The Flying bookstore

XÓA BỎ LỌC

Sách theo thể loại Hồi ký

Hiển thị 40 cuốn sách hiện có

Sách mua và thuê

Danh mục

- Học thuật
- Hồi ký
- Kinh điển
- Hải kịch
- truyện tranh
- Văn hoá
- Kịch
- Kinh tế học
- Giáo dục
- Môi trường
- Sứ thi
- Truyền 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 mạn
- Khoa học
- Truyền ngắn
- Xã hội
- Xã hội học
- Các môn thể thao
- Hồi hộp
- Sách giáo khoa
- Giật gân

Castaway

Red Dust

Hang Time

Broken

Dannii

The Game - out of print

31 Songs

Jane Austen

At the Center of the Storm

Botham

A Parrot in the Pepper Tree

Bringing the House Down

When Heaven and Earth C...

Without Mercy

Broken Music

Every Secret Thing

It was on Fire when I Lay D...

Talking Straight

A Blessing over Ashes

Don Cherry's Hockey Stor...

Thuê: 200đ/ngày
Mua: 313.670đ

Thuê: 1200đ/ngày
Mua: 82.950đ

Thuê: 500đ/ngày
Mua: 343.210đ

Thuê: 950đ/ngày
Mua: 95.480đ

Thuê: 1600đ/ngày
Mua: 232.330đ

Thuê: 2.600đ/ngày
Mua: 277.200đ

Thuê: 1.600đ/ngày
Mua: 306.180đ

Thuê: 2.300đ/ngày
Mua: 24.031đ

Thuê: 300đ/ngày
Mua: 138.460đ

Thuê: 2.800đ/ngày
Mua: 64.330đ

Thuê: 1.100đ/ngày
Mua: 320.740đ

Thuê: 1.300đ/ngày
Mua: 147.280đ

Thuê: 2.600đ/ngày
Mua: 261.100đ

Thuê: 500đ/ngày
Mua: 29.680đ

Thuê: 1.700đ/ngày
Mua: 171.640đ

Thuê: 500đ/ngày
Mua: 772.200đ

Thuê: 1.000đ/ngày
Mua: 192.080đ

Thuê: 2.500đ/ngày
Mua: 259.140đ

Thuê: 100đ/ngày
Mua: 32.060đ

Thuê: 2.500đ/ngày
Mua: 174.930đ

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

Nhập mail tại đây

Lien he

+84 11 Trương Phước Phan, Bình Hưng Hòa, Bình Tân, TP.HCM

+84 123 345 123

support@flying.bookstore

Chinh sach

Chinh sach bao mat

Huong dan ve tinh trang sach

Tiệm Giuong Huyền Cảnh ~
Kiem bau do ien han

Tiệm Giuong Huyền Cảnh cho thuê đ...

Cô Lộ

Phan Văn Đồng

Chợ Lớn

KHU PHỐ 16

Phim tài

Dữ liệu bản đồ ©2024 - Điều khoản

Hình 6.4: Trang danh sách bài đăng và tìm kiếm tài liệu

6.3.3.1 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.

6.3.3.2 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.

6.3.3.3 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/loc.

4	Phân trang	Điều hướng giữa các trang kết quả tìm kiếm/ lọc.	Button group	integer	Trang 1		Hiển thị số trang và nút điều hướng (Trước/Tiếp).
---	------------	--	--------------	---------	---------	--	---

6.3.3.4 Quy trình xử lý

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

6.3.3.5 API Mô tả

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

6.3.4 Trang chi tiết bài đăng

The screenshot shows a detailed book listing for 'Hide and Seek' by Fern Michaels. The page includes the book cover, author information, publication details, and a summary. It also displays rental information (2.200đ/ngày), a rental calculator, and a sidebar for recommended books.

Thông tin chung về sách	Đặt thuê
Ngày xuất bản: 01/01/2008	Thuê theo ngày: 2.200đ/ngày
Kích thước: 15.5x20.5	Ngày nhận: 19/05/2024
ISBN: 1558746218	Ngày trả:
Số trang: 316	Giá thuê: 2.200đ/ngày
Thông tin về sách thuê	Số ngày thuê: 0
Tiền cọc: 230.900đ	Tổng tiền thuê: 0
Phản trám hư hại: 9	Tiền cọc: 230.900đ
Địa chỉ cho thuê: 123 Nguyễn Trọng Tuyển, phường 12, quận Bình Thạnh	Tổng tiền: 0
Phi phạt trả trễ: 2.500đ/ngày	Bạn cần chọn ngày trả

Sách mới ra mắt

Sách	Giá	Thao tác
Pride and Prejudice	2.300đ/ngày	<input type="button" value="Thêm vào giỏ hàng"/>
Emma	200đ/ngày	<input type="button" value="Thêm vào giỏ hàng"/>
Sense and Sensibility	2.400đ/ngày	<input type="button" value="Thêm vào giỏ hàng"/>
The House of Scorta	2.000đ/ngày	<input type="button" value="Thêm vào giỏ hàng"/>
Ice	1600đ/ngày	<input type="button" value="Thêm vào giỏ hàng"/>

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

Địa chỉ: Tiệm Giuong Huyền Cảnh - ..., Xem bản đồ lớn hơn

Hình 6.5: Trang chi tiết bài đăng

6.3.4.1 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.

6.3.4.2 Quy tắc và Ràng buộc

1. Khách có thể xem thông tin tài liệu mà không cần đăng nhập.
2. 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.**
3. Hiển thị nút hành động "Đặt thuê ngay" hoặc "Mua ngay" nếu tài liệu còn khả dụng.
4. Nếu tài liệu không khả dụng, hiển thị thông báo "Sách hiện không có sẵn".

6.3.4.3 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.

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 'ĐÁNH GIÁ' (the latter being underlined). Below this is a summary section with a 5-star rating icon, '2 bài đánh giá', and buttons for 'Xem tất cả' and '5 sao' (highlighted), along with other buttons for 4, 3, 2, and 1 star. Below this is a list of two reviews, each with a small profile picture of an angel holding a book, the title 'Ấn danh', the text 'Sách đẹp, văn còn mới', the rating '5 sao', and the date '13/05/2024'. The reviews are identical.

Hình 6.6: 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 (ân danh nếu không công khai).	Textview				Hiển thị tên người đánh giá hoặc để “Ân danh” nếu không hiển thị.

6.3.4.4 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:
 - **Thuê:** Nhấn vào "Đặt thuê ngay".
 - **Mua:** Nhấn vào "Mua ngay".
 - Nếu tài liệu không khả dụng, khách chỉ có thể xem thông tin.

6.3.4.5 API Mô tả

Api lấy thông tin chi tiết tài liệu

- **Endpoint:** /api/listing/detailListing/3
- Phương thức: GET
- **Mô tả:** Lấy thông tin chi tiết của một tài liệu.
- Tham số đầu vào: Không yêu cầu.

Api đánh giá sách

- **API:** Lấy danh sách đánh giá sách
- **Endpoint:** /api/review
- Phương thức: GET
- **Mô tả:** Lấy danh sách đánh giá chi tiết của sách.

6.3.5 Trang giỏ hàng

Giỏ hàng của bạn

SÁCH THUÊ SÁCH MUA

Sách	Số lượng	Giá
At the Center of the Storm George Tenet	1	138,460đ

Thông tin đặt hàng

Thông tin	Giá trị	Thành tiền
Người bán	Developer Dev	59.340đ
Địa chỉ	123 Nguyễn Trọng Tuyển, phường 12, quận Bình Thạnh	10.000đ
Số điện thoại	0905 907 362	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

Hình 6.7: Trang giỏ hàng

6.3.5.1 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.

6.3.5.2 Quy tắc và Ràng buộc

1. Xem giỏ hàng:

- Hiển thị đầy đủ thông tin sách trong giỏ hàng bao gồm: tên sách, số lượng, giá, tổng giá trị.

2. 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.

3. 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.

4. 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.

6.3.5.3 Mô tả chi tiết

STT	Tên	Mô tả	Loại điều khiển	Kiểu dữ liệu	Giá trị mặc định	Bắt buộc	Quy tắc
1	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.

6.3.5.4 Quy trình xử lý

1. Xem giỏ hàng:

- Người dùng truy cập giỏ hàng để xem danh sách sách đã thêm.
- Hệ thống hiển thị các sách kèm thông tin chi tiết (tên, giá, số lượng).

2. Cập nhật giỏ hàng:

- Người dùng có thể thay đổi số lượng sách bằng cách nhập trực tiếp hoặc chọn tăng/giảm.
- Hệ thống tự động cập nhật tổng giá trị của giỏ hàng.

3. Xóa sách:

- Khi người dùng nhấn nút **Xóa**, sách sẽ bị loại bỏ khỏi giỏ.
- Hệ thống tính lại tổng tiền thanh toán.

4. Áp dụng mã giảm giá:

- Người dùng nhập mã giảm giá và nhấn **Áp dụng**.
- Hệ thống kiểm tra mã giảm giá hợp lệ, sau đó cập nhật tổng tiền với số tiền giảm.

5. Xác nhận đặt hàng:

- Người dùng kiểm tra lại thông tin đặt hàng (địa chỉ, số điện thoại, tên người nhận).
- Nhấn nút **Đặt mua hàng** để xác nhận.

6.3.5.5 API Mô tả

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:** keyword (string, bắt buộc): Từ khóa dùng để tìm kiếm voucher session.

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:** keyword (string, bắt buộc): Từ khóa dùng để tìm kiếm voucher session.

6.3.6 Trang đặt hàng

The screenshot shows the checkout process on The Flying Bookstore website. The top navigation bar includes the logo, search bar, and account icons. The main form is divided into several sections:

- Thông tin đặt mua** (Purchase Information):

Họ và tên lót *	Nguyễn Lê Xuân	Tên *	Hoa
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):

	At the Center of the Storm George Tenet	Số lượng: 1	138.460đ	197.800đ
--	---	-------------	----------	----------
- Mã giảm giá** (Discount Codes):

Khuyến Mãi từ người bán	Có thể chọn 1
Giảm 10.000đ Cho đơn hàng từ 50.000đ	<input type="checkbox"/> BỎ CHỌN
Chọn hoặc nhập mã khác	
Khuyến Mãi toàn sàn	Có thể chọn 1
Giảm 50.000đ Cho đơn hàng từ 400.000đ	<input type="checkbox"/> BỎ CHỌN
Chọn hoặc nhập mã khác	
- Thanh toán** (Payment):

Dựa tiền trực tiếp	Chuyển khoản	VNPAY
Bạn sẽ đưa Од cho công ty Flying Bookstore tại địa chỉ 171/11 Trường Phước Phan, quận Bình Tân, tp.HCM		
- Quy trình thuê sách** (Book Rental Process):

Đến nhận sách	Đọc sách	Trả sách
Sau khi thanh toán thành công, bạn có thể chọn 1 trong 2 cách sau: - Bạn hãy đến địa chỉ thuê sách và đưa mã đơn hàng chủ sách để lấy sách bạn nhé. - Bạn 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.	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.	Vui lòng trả cuốn sách đúng hạn, bạn có thể đặt ship trả sách nhưng thời gian ship cũng 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 gia hạn thêm hoặc mua cuốn sách đó luôn chỉ với 1 click.
- Nhập mail để nhận được những quà tặng đặc biệt nhé!** (Enter email to receive special gifts):

Hình 6.8: Trang đặt hàng

6.3.6.1 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

6.3.6.2 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.

6.3.6.3 Mô tả chi tiết

STT	Tên	Mô tả	Loại điều khiển	Kiểu dữ liệu	Giá trị mặc định	Bắt buộc	Quy tắc
1	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.
----	--------------	---------------------	--------	--	--	---	---

6.3.6.4 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.

6.3.6.5 API Mô tả

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:
 - status (string): Trạng thái của đơn hàng.
 - listingId (number): ID của danh sách liên quan đến đơn hàng.
 - buyerAddress (string): Địa chỉ của người mua.
 - paymentMethod (string): Phương thức thanh toán cho đơn hàng.
 - VoucherShopId (string): ID của voucher shop.
 - VoucherSessionId (string): ID của voucher session.

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:
 - id (number): ID duy nhất của người dùng.
 - username (string): Tên đăng nhập của người dùng.
 - email (string): Địa chỉ email của người dùng.
 - phoneNumber (string): Số điện thoại của người dùng.
 - firstName (string): Tên của người dùng.

- lastName (string): Họ của người dùng.
- birthDate (string): Ngày sinh của người dùng.
- avatarUrl (string): URL ảnh đại diện của người dùng.
- address (string): Địa chỉ của người dùng.
- password (string): Mật khẩu của người dùng.

Đối với thanh toán chuyên khoản giao diện cung cấp thêm thông tin chuyên 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: PGD 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 6.9: Phần thông báo chuyển khoản trang đặt hàng

6.3.7 Trang xuất đơn hàng

The screenshot displays two main sections of the website:

- Top Section (Order Confirmation):**
 - Đơn hàng được tạo thành công!**
 - Bài đăng**: At the Center of the Storm by George Tenet.
 - Số lượng: 1**, **Giá bán: 138.460đ**
 - Thông tin đặt hàng** table:

Mã đơn hàng	14	Phương thức thanh toán	COD
Ngày đặt mua	30/11/2024	Giá bán	138.460đ
Người bán	Developer Dev	Khuyến mãi từ người bán	10.000đ
Địa chỉ người bán	123 Nguyễn Trọng Tuyển, phường 12, quận Bình Thạnh	Khuyến mãi từ The Flying Bookstore	50.000đ
Số điện thoại người bán	0905 907 362	Tổng tiền thanh toán	78.460đ
Trạng thái	Chờ thanh toán		
 - Thao tác**: **Bạn hãy nhanh chóng thanh toán theo phương thức COD để lấy hàng**.
 - Buttons**: Tiếp tục mua sắm, Quản lý đơn hàng, Xác nhận lấy hàng.
- Bottom Section (Rental Process):**
 - Quy trình thuê sách**:
 - Đến nhận sách**: **Đến nhận sách**
 - Đọc sách**: **Gìn giữ sách cẩn thận bạn nhé.** Hãy nhớ bạn còn có người khác đọc nữa, nên tránh việc viết vẽ hay làm rách sách bạn nhé.
 - Trả sách**: **Vui lòng trả cuốn sách đúng hạn, bạn có thể đặt ship trả sách** nhưng thời gian ship cũng 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 tăng thêm hoặc mua cuốn sách đó luôn chỉ với 1 click.
- Footer (Dark Blue Bar):**
 - Nhập mail để nhận được những quà tặng đặc biệt nhé!**
 - Input Field**: Nhập mail tại đây.
 - Buttons**: and .
- Contact Information:**
 - 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.
- Map**: A map showing the location of "Tiệm Giường Huyền Cảnh" on a street in HCMC.

Hình 6.10: Trang xuất đơn hàng

6.3.7.1 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.

6.3.7.2 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.

6.3.7.3 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

6.3.7.4 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.

6.3.7.5 API Mô tả

API: Lấy thông tin đơn hàng bán

- **Endpoint:** /api/SaleOrder/1
- **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:**
 - Không có.

6.3.8 Trang lấy đơn hàng

The screenshot shows the 'Lấy thành công đơn hàng!' (Order pickup successful!) page. At the top, there's a search bar with 'Tim sách...' and a magnifying glass icon. Below it are three buttons: 'Diễn thông tin' (Information), 'Xuất đơn hàng' (Create order), and 'Lấy hàng' (Pick up). The main content area displays a book cover for 'Alchemy & Academe' by Anne McCaffrey. The book details are: 'Alchemy and Academe' by Anne McCaffrey, Số lượng: 1, Giá thuê: 2.000đ, Cọc: 120.300đ. Below this is a table for 'Thông tin đặt hàng' (Order information) with columns for icons, data, and status.

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

At the bottom left is a button 'Tiếp tục mua sắm' (Continue shopping) and at the bottom right is a button 'Quản lý đơn hàng của tôi' (Manage my orders).

Quy trình thuê sách

- Đến nhận sách**: Includes an icon of a location pin and a magnifying glass. Text: Sau khi thanh toán thành công, bạn có thể chọn 1 trong 2 cách sau:
- Bạn hãy đến địa chỉ thuê sách và đưa mã đơn hàng chủ sách để lấy sách bạn nhé.
- Bạn 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 an open book with a heart. Text: Cố gắng sách cần thận bạn nhé.
Hãy nhớ sau bạn còn có người khác đọc nữa, nên hay tránh việc viết vẽ làm rách sách bạn nhé.
- Trả sách**: Includes an icon of a calendar with a red circle. Text: Vui lòng trả cuốn sách đúng hạn, bạn có thể đặt ship trả sách nhưng thời gian ship cũng không được quá thời gian trả sách. Sau khi sách nhận lại được sạch, số tiền còn lại sẽ được hoàn về tài khoản của bạn. Nếu bạn cần thêm thời gian, bạn có thể ghi hạn thêm hoặc mua cuốn sách đó luôn chỉ với 1 click.

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

Liên hệ

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

Map

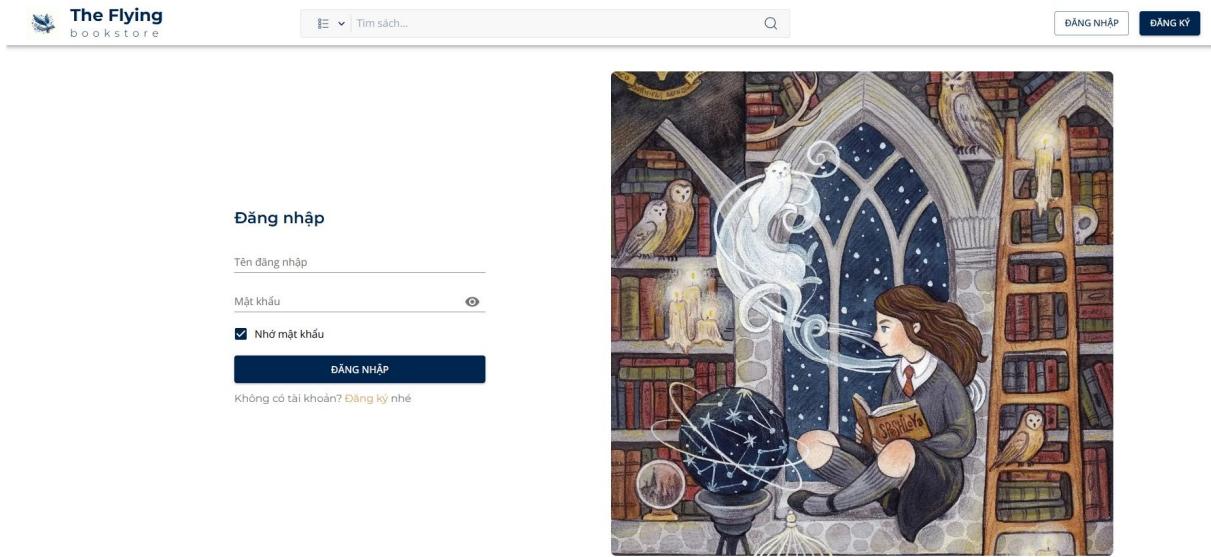
The Flying Bookstore là nơi có thể giúp bạn học sinh/sinh viên đem cho thuê, mượn tài liệu của mình. Đồng thời giúp quản lý các tài liệu mình cho thuê, mượn.

[Nhập mail tại đây](#)

Hình 6.11: 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.

6.3.9 Trang đăng nhập



Hình 6.12: Trang đăng nhập

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

6.3.9.1 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.

6.3.9.2 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.

6.3.9.3 Mô tả chi tiết

61	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

6.3.9.4 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:
 - Nếu thông tin hợp lệ:
 - Hiển thị thông báo Đăng nhập thành công.
 - Chuyển hướng đến trang chính.
 - Nếu thông tin không hợp lệ:
 - Hiển thị thông báo lỗi chi tiết (Ví dụ: "Tên đăng nhập không tồn tại").

6.3.9.5 API Mô tả

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
----------	--------------	----------	----

6.3.10 Trang đăng ký

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

Đăng ký

Tên đăng nhập

Mật khẩu

Nhập lại mật khẩu *

Độc Đáo

Tôi đồng ý với Điều khoản bảo mật

ĐĂNG KÝ

Có tài khoản rồi? [Đăng nhập](#) nhé!

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

hoaminzy

Tiệm Guacamole Cánh ... Xem bản đồ lớn hơn

Tiệm Guacamole Cánh - cho thuê áo

Phim tắt: Dữ liệu bản đồ ©2024 - Điều khoản - Báo cáo một lỗi bản đồ

Hình 6.13: Trang đăng ký

6.3.10.1 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.

6.3.10.2 Quy tắc và Ràng buộc

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

6.3.10.3 Mô tả chi tiết

61	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

6.3.10.4 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").

6.3.10.5 API Mô tả

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

6.3.11 Trang quản lý thông tin user

The screenshot shows a user profile management interface. On the left, there's a sidebar with icons for account settings, order management, and other features. The main area is titled 'Thông tin cá nhân' (Personal Information). It contains several input fields: 'Họ và tên lót' (Name) with value 'Nguyễn Lê Xuân', 'Tên' (Name) with value 'Hòa', 'Email' with value 'hoa.nguyenlexuan@gmail.com', 'Ngày sinh' (Date of Birth) with value '12/12/2000', 'Số điện thoại' (Phone Number) with value '0905907362', and 'Địa chỉ' (Address) with value '123 Nơ Trang Long, phường 12, quận Bình Thạnh, thành phố Hồ Chí Minh'. At the bottom are two buttons: 'XÁC NHẬN' (Confirm) and 'Hủy' (Cancel).

Hình 6.14: Trang quản lý thông tin user

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

6.3.11.1 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.

6.3.11.2 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).

6.3.11.3 Mô tả chi tiết

STT	Tên	Mô tả	Loại điều khiển	Kiểu dữ liệu	Giá trị mặc định	Bắt buộc	Quy tắc
1	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	Textbox	varchar(255)	Lấy từ cơ sở dữ liệu	X	Phải là email hợp lệ.

		người dùng					
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.

6.3.11.4 Quy trình xử lý

Bước 1: Người dùng truy cập vào trang "Cài đặt tài khoản".

Bước 2: Các trường thông tin cá nhân của người dùng được hiển thị với giá trị hiện tại từ cơ sở dữ liệu.

Bước 3: Người dùng có thể chỉnh sửa các trường thông tin mong muốn như Tên, Email, Ngày sinh, Số điện thoại và Địa chỉ.

Bước 4: Sau khi hoàn thành việc chỉnh sửa, người dùng nhấn nút "Xác nhận" để lưu thay đổi.

Bước 5: Hệ thống kiểm tra tính hợp lệ của các thông tin:

- Nếu hợp lệ, thông tin sẽ được cập nhật và hiển thị thông báo thành công.
- Nếu có lỗi, hệ thống sẽ thông báo và yêu cầu người dùng chỉnh sửa lại các trường sai sót.

6.3.11.5 API Mô tả

API: Cập nhật thông tin người dùng

- **Endpoint:** /api/user/2
- **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

6.3.12 Trang quản lý đơn hàng thuê

Id đơn hàng	Chủ sách	Thời gian thuê	Trạng thái người thuê	Xem chi tiết	
#12	Developer Dev	13/05 - 13/05/2024	Đã trả sách	Xem chi tiết	
44	With No One As ...	1 ngày	486,500	900	487,400

Id đơn hàng	Chủ sách	Thời gian thuê	Trạng thái người thuê	Xem chi tiết	
#20	Braun Tracy	19/05 - 19/05/2024	Đã lấy hàng	Xem chi tiết	
7	Alchemy and Aca...	1 ngày	120,300	2,000	122,300

Hình 6.15: Trang quản lý đơn hàng của tôi

Chứa danh sách đơn hàng của người dùng 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 nhấn 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:

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
Vui lòng thanh toán đơn hàng trong 24 giờ Tổng tiền thuê: 2.000đ Tổng tiền cọc: 120.300đ Tổng tiền: 120.300đ 					

Hình 6.16: Đơn hàng có thể cancel

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
Vui lòng thanh toán đơn hàng trong 24 giờ, nếu chuyển khoản thành công, bạn hãy nhấn nút Đã trả tiền Tổng tiền thuê: 2.000đ Tổng tiền cọc: 120.300đ Tổng tiền: 120.300đ 					

Hình 6.17: 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 cộng
320	Bad Samaritans	5 ngày	247,000	13,500	260,500
Vui lòng chỉ nhấn "đã nhận được hàng" khi đơn hàng đã được giao đến bạn và bạn đã nhận được hàng Tổng tiền thuê: 13.500đ Tổng tiền cọc: 247.000đ Tổng tiền: 247.000đ 					

Hình 6.18: 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
320	Bad Samaritans	5 ngày	247,000	13,500

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 6.19: Đơn hàng xác nhận trả sách

6.3.12.1 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.

6.3.12.2 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.

6.3.12.3 Mô tả chi tiết

	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.

6.3.12.4 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:
 - o **Chờ thanh toán:** Người dùng có thể nhấn "Đã Trả Tiền" để xác nhận thanh toán cọc.
 - o **Đã trả sách:** Người dùng có thể nhấn "Đánh Giá" để đánh giá dịch vụ thuê sách.
5. Nếu muốn xem chi tiết hơn về từng đơn hàng, người dùng nhấn vào nút "Xem chi tiết".

6.3.12.5 API Mô tả

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

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

lesseeId number ID của người thuê sách Có

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 Mô tả

id number ID của đơn thuê cần chỉnh sửa Có

Tên tham số Loại dữ liệu Mô tả

Bắt buộc

status	string	Trạng thái mới của đơn thuê	Có
--------	--------	-----------------------------	----

6.3.13 Trang quản lý đơn hàng cho thuê

Id đơn hàng	Người thuê	Thời gian thuê	Trạng thái chủ sách	Xem chi tiết
#16	hoa hoa	11/05 - 13/05/2024	Chủ sách đã đưa hàng	Xem chi tiết
Id bài đăng	Tên sách	Thời gian thuê	Tiền cọc	Tiền thuê
2003	tổ tiên	2 ngày	120,000	1,200
Tổng tiền thuê: 1.200đ Tổng tiền cọc: 120.000đ Tổng tiền: 120.000đ				

Id đơn hàng	Người thuê	Thời gian thuê	Trạng thái chủ sách	Xem chi tiết
#13	ádf ádf	13/05 - 13/05/2024	Chủ sách đã lấy lại sách	Xem chi tiết
Id bài đăng	Tên sách	Thời gian thuê	Tiền cọc	Tiền thuê
2000	From Potter's Field	1 ngày	12,312,123	22
Tổng tiền thuê: 22đ Tổng tiền cọc: 12.312.123đ Tổng tiền: 12.312.123đ				

Hình 6.20: 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	Người thuê	Thời gian thuê	Trạng thái chủ sách	Xem chi tiết
#12	Developer Dev	16/05 - 16/05/2024	Chủ sách chờ nhận sách	Xem chi tiết
Id bài đăng	Tên sách	Thời gian thuê	Tiền cọc	Tiền thuê
22	Into the Darkness	0 ngày	79,200	1,300
Tổng cọc: 79.200đ				
Tổng tiền thuê: 1.300đ Tổng tiền cọc: 79.200đ Tổng tiền: 79.200đ				

Hình 6.21: Chủ sách xác nhận nhận lại sách

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

6.3.14 Trang quản lý đơn mua

The Flying bookstore

Tim sách...

TẤT CẢ ĐÃ ĐẶT HÀNG ĐÃ NHẬN ĐÃ HỦY

Bạn hiện có 4 đơn hàng

TÀI LIỆU

- Cài đặt tài khoản
- Đơn thuê
- Đơn cho thuê
- Đơn mua
- Đơn bán
- Quản lý voucher
- Quản lý bài đăng
- Thêm bài đăng

Quản lý đơn mua

ĐÃ TRẢ TIỀN

Vui lòng thanh toán đơn hàng trong 24 giờ, nếu chuyển khoản thành công, bạn hãy nhấn nút Đã trả tiền

ĐÃ TRẢ TIỀN

Gía bán: 235.480đ
Tổng tiền: 235.480đ

ĐÃ HỦY

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

Gía bán: 175.700đ
Tổng tiền: 175.700đ

Chủ sách đã đưa hàng

Vui lòng chỉ nhận "đã nhận được sách" khi đơn hàng đã được giao đến bạn và bạn đã kiểm tra xong hàng

ĐÃ GIAO SÁCH CHO NGƯỜI MUA

Gía bán: 161.630đ
Tổng tiền: 161.630đ

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
+123 345 123
support@flying bookstore

Chính sách

Chính sách bảo mật
Hướng dẫn về tình trạng sách

6.3.15 Trang quản lý đơn bán

The screenshot shows the 'Quản lý đơn bán' (Order Management) section of the website. At the top, there are tabs for 'Tất cả' (All), 'ĐÃ ĐẶT HÀNG' (Placed), 'ĐÃ NHẬN' (Received), and 'ĐÃ HỦY' (Cancelled). Below this, it says 'Bạn hiện có 1 đơn hàng' (You currently have 1 order). The order details are listed in a table:

ID đơn hàng	Người mua	Trạng thái	Người mua	Xem chi tiết		
#5	Romaguera Nigel	Chờ thanh toán				
Id bài đánh...	Tên sách	Thời gian đặt	Giá gốc	Giá bán	Tổng cộng	
4	Foundling	12/11/2024	383.700	268.590	268.590	⋮

At the bottom right, it shows 'Giá bán: 268.590đ' and 'Tổng tiền: 268.590đ'. A note below the table says 'Người mua mới đặt hàng, chờ người mua thanh toán'.

A dark blue banner with white text: 'Nhập mail để nhận được những quà tặng đặc biệt nhé!' (Enter your email to receive special gifts!). Below the banner, there is a search bar with the placeholder 'hoaminzy' and a magnifying glass icon. To the left, there is a logo for 'The Flying bookstore'.

6.3.16 Chi tiết đơn hàng

The screenshot shows the 'Chi tiết đơn hàng #20' (Order Detail #20) page. On the left, there is a sidebar with navigation links: 'Cài đặt tài khoản', 'Quản lý đơn hàng của tôi', 'Quản lý đơn hàng của khách', 'Quản lý bài đăng', and 'Thêm bài đăng'. The main content area displays the following information:

Chi tiết đơn hàng #20

Bài đăng

Sản phẩm	Số lượng	Giá thuê:
Alchemy and Academe Anne McCaffrey	1	2.000đ Cọc: 120.300đ

Thông tin đặt hàng

Mã đơn hàng	Trạng thái	Chờ thanh toán
20	Đang chờ	COD
Ngày đặt thuê	19/05/2024	Phương thức thanh toán
Người cho thuê	Braun Tracy	Số ngày thuê
Thời gian thuê	19/05/2024 - 19/05/2024	Tiền thuê
Địa chỉ	Đường Kim Giang, Phường Kim Giang, Quận Thanh Xuân, Hà Nội	Tiền cọc
Số điện thoại	0905 907 362	Tổng cộng

Hình 6.22: Chi tiết đơn hàng

6.3.16.1 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.

6.3.16.2 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.

6.3.16.3 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/dơn 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

6.3.16.4 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.

6.3.16.5 API Mô tả

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	Mô tả	Bắt buộc
-------------	--------------	-------	----------

leaseOrderId number ID của đơn thuê cần truy xuất Có

API: Lấy danh sách đơn thuê (Quản trị viên)

- Endpoint:** /api/leaseOrder/Quản trị viên
- 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.

6.3.17 Quản lý bài đăng

ID bài đăng	Tên bài đăng	Tác giả	Giá thuê	Tiền cọc	Tiền phạt
2	From Pott...	Patricia Daniels C...	2.800đ/ngày	414.800đ/ngày	4.600đ/ngày
858	Terminal	Colin Forbes	1.400đ/ngày	262.400đ/ngày	2.900đ/ngày
37	To Die for	Janet Neel	800đ/ngày	336.400đ/ngày	3.700đ/ngày
13	Vanishing ...	John Nichol	2.300đ/ngày	251.000đ/ngày	2.700đ/ngày
14	Hide and ...	Fern Michaels	2.200đ/ngày	230.900đ/ngày	2.500đ/ngày

Hình 6.23: 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 khoá vào ô tìm kiếm thì danh sách bài đăng sẽ lọc theo từ khoá tìm kiếm.

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.

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.

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

4. Quy trình thực hiện

- 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:
 - o Nhấn vào biểu tượng **Xem chi tiết** để xem đầy đủ thông tin bài đăng.
 - o Nhấn vào biểu tượng **Chỉnh sửa** để cập nhật thông tin bài đăng.
 - o Nhấn vào biểu tượng **Xóa** để xóa bài đăng (nếu không có đơn hàng liên quan).

5. API Mô tả

API: Lấy danh sách bài đăng

- o **Phương thức:** GET
- o **Endpoint:** /api/posts
- o **Tham số đầu vào:**

Tên tham số	Loại dữ liệu	Mô tả	Bắt buộc
-------------	--------------	-------	----------

user_id	Số nguyên	ID của người dùng	Có
---------	-----------	-------------------	----

API: Xóa danh sách (Delete Listing)

- **Endpoint:** /api/listing/delete/{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
-----------	--------	--------------------------	----

6.3.18 Tạo bài đăng

Chào Dev

- Cài đặt tài khoản
- Quản lý đơn hàng của tôi
- Quản lý đơn hàng của khách
- Quản lý bài đăng
- Thêm bài đăng

Đăng bài mới

< Sách

Tìm sách *	From Potter's Field
Tiêu đề *	From Potter's Field
Tác giả *	Patricia Daniels Cornwell
Thể loại	Bí ẩn, Viễn tưởng, Bí ẩn,, Giật gân, Giật gân,, Hồi hộp, Bí ẩn,, Bí ẩn,,,
Nhà xuất bản *	Oxford University Press
Số trang *	340
Kích thước	15.5x20.5
ISBN *	2005018
Ngày phát hành *	1995-01-01
Ngôn ngữ *	en

CHỌN SÁCH

< Tài liệu

Phần trăm hư hại *

UPLOAD FILE

No image

TẠO TÀI LIỆU

< Bài đăng

Mô tả trạng thái sách *

Tiền cọc *	Giá thuê theo ngày *	Phi phạt trả trễ theo ngày *
------------	----------------------	------------------------------

Địa chỉ cho thuê *

123 Nguyễn Trọng Tuyển, phường 12, quận Bình Thạnh

TẠO BÀI ĐĂNG

Hình 6.24: 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.

< Sách

Tìm sách *

Thêm mới sách | X

Tiêu đề *	Tác giả *	
Thể loại		
Nhà xuất bản *	Số trang *	Kích thước
ISBN *	Ngày phát hành *	Ngôn ngữ *

TẠO SÁCH

Hình 6.25: Thêm mới sách

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.

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).

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

STT	Tên	Mô tả	Kiểm soát	Loại dữ liệu	Giá trị mặc định	Bắt buộc	Quy tắc
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

4. Quy trình thực hiện

- 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.

5. API Mô tả

API: Tạo danh sách mới (Create Listing)

- 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

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

6.3.19 Trang quản lý voucher

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



Liên hệ

171/11 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



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.

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.

3. Mô tả chi tiết

1. 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 Voucher	Mã Voucher	Thời Gian	Giá Trị	Loại Voucher	Số Tiền Giảm	Phần Trăm Giảm
			Tối Thiểu Lực	Tối Thiểu	Voucher		

Tên chương trình

- khuyến mãi (VD: Shop Anniversary).
- Mã voucher duy nhất.

STT	Tên Voucher	Mã Voucher	Thời Gian Hiệu Lực	Giá Trị Tối Thiểu	Loại Voucher	Số Tiền Giảm	Phần Trăm Giảm
3	Thời gian voucher có hiệu lực (Từ ngày - Đến ngày).						
4	Giá trị tối thiểu đơn hàng áp dụng.						
5	Loại voucher: Giảm giá theo số tiền hoặc phần trăm.						
6	Giá trị giảm giá cụ thể (theo số tiền hoặc phần trăm).						

2. Xóa Voucher

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

4. Quy trình thực hiện

1. Xem Voucher

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

2. Xóa Voucher

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

5. API Mô tả

1. API: Lấy danh sách voucher

- Phương thức:** GET
- Endpoint:** /api/seller/vouchers
- Tham số đầu vào:**

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

2. API: Xóa voucher

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

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

6.3.20 Trang thêm voucher

6.3.21 Trang đăng nhập Quản trị viên

Hình 6.26: Trang đăng nhập Quản trị viên

1.1. Tổng quan

- Mục đích:** Quản trị viên 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:** Quản trị viên.
- Điều kiện tiên quyết:** Quản trị viên đã có tài khoản trong hệ thống.
- Kết quả mong muốn:** Quản trị viên đăng nhập thành công và được chuyển hướng tới trang quản lý.

1.2. 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.

1.3. Mô tả chi tiết

STT	Tên	Mô tả	Điều khiển	Kiểu dữ liệu	Giá trị mặc định	Bắt buộc	Quy tắc
1	Tên tài khoản	Tên tài khoản Quản trị viên để đăng nhập	Textbox	varchar(255)	X		Người dùng nhập tài khoản đã đăng ký.
2	Mật khẩu truy cập hệ thống	Mật khẩu để truy cập hệ thống	Password textbox	varchar(255)	X		Người dùng nhập mật khẩu hợp lệ.

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

1.4. Quy trình

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

1.5. API Đăng nhập

- Phương thức:** POST
- URL:** /api/user/login
- Payload:**

```
json
{
  "username": "ten_dang_nhap",
  "password": "mat_khau"
}
```

- 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.

6.3.22 Trang quản lý đơn hàng thuê cho thuê Quản trị viên

ID	TÊN SÁCH	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 ✓

Hình 6.34: Trang quản lý Quản trị viên

Chứa 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.

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.

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.

3. Mô tả chi tiết

STT	Tên	Mô tả	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	Người thuê đã trả tiền	Xác nhận người thuê đã thanh toán tiền thuê	Button				Hiển thị khi đơn hàng đang ở trạng thái “Chờ người thuê thanh toán”.

5. Quy trình

- Quản trị viên truy cập vào trang “Quản lý đơn thuê” từ menu.
- Hệ thống hiển thị danh sách các đơn thuê sách, bao gồm tất cả các chi tiết cần thiết về từng đơn thuê.
- 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.
- 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:
 - 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.
 - Đã trả cọc cho người thuê:** Hiển thị khi cọc được trả lại cho người thuê sau khi đơn hàng hoàn tất.

- Hệ thống có phân trang cho danh sách đơn hàng để giúp quản trị viên dễ dàng duyệt qua các đơn hàng khác nhau.

5. API Quản lý đơn hàng

- Phương thức:** GET
- URL:** /api/orders/Quản trị viên
- Mô tả:** Lấy danh sách tất cả các đơn hàng thuê sách cho quản trị viên.
- Phương thức:** POST
- URL:** /api/order/{orderId}/confirm-payment
- 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”.

6.3.23 Trang quản lý đơn hàng mua bán Quản trị viên

ID	TIÊU ĐỀ	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 Kertzmann	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	

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

6.3.24 Trang quản lý voucher toàn sàn Quản trị viên

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

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

1. Tổng quan

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

2. Quy tắc và Ràng buộc

- Quản trị viên 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 Quản trị viên 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.

3. Mô tả chi tiết

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

4. Quy trình

- Quản trị viên đăng nhập** vào hệ thống với tài khoản có quyền truy cập chức năng quản lý voucher.
- Quản trị viên vào mục "Quản lý voucher"** từ menu chính.
- 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.
- Để xóa một voucher, **Quản trị viên nhấn nút "Xóa"** tại dòng voucher tương ứng.
- Hệ thống hiển thị **hộp thoại xác nhận** với thông tin cụ thể về voucher sẽ xóa.
- Sau khi Quản trị viên xác nhận, hệ thống **xóa voucher** và cập nhật lại danh sách trên giao diện.
- Thay đổi được ghi lại** trong hệ thống để đảm bảo khả năng kiểm tra sau này.

6.3.25 Trang thêm và sửa voucher toàn sàn Quản trị viên

The Flying bookstore

Tạo voucher mới

Tên voucher

Mã voucher

Ngày bắt đầu

Ngày kết thúc

Giá trị tối thiểu

Giá trị tối thiểu

Loại voucher

Giảm giá bằng số tiền

Số tiền giảm

Số tiền giảm

XÁC NHẬN

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

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:**
 - Quản trị viên 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.

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.

3. Mô tả chi tiết

1. Thêm Voucher

STT	Tên	Mô tả	Kiểm soát	Loại 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

STT	Tên	Mô tả	Kiểm soát	Loại dữ liệu	Giá trị mặc định	Bắt buộc	Quy tắc
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	Giá trị giảm giá	Number field	Số	N/A	Có	Số dương

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.

4. Quy trình thực hiện

1. Thêm Voucher

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

2. Sửa Voucher

- Quản trị viên truy cập trang **Quản lý voucher** và chọn voucher cần chỉnh sửa.
- Thực hiện chỉnh sửa các thông tin cần thay đổi.
- Nhấn nút **Xác nhận**.
- Hệ thống kiểm tra:
 - Nếu thông tin hợp lệ:
 - Cập nhật thông tin trong cơ sở dữ liệu.
 - Hiển thị thông báo **Cập nhật voucher thành công**.
 - Nếu thông tin không hợp lệ:
 - Hiển thị thông báo lỗi chi tiết (Ví dụ: "Ngày bắt đầu không hợp lệ").

5. API Mô tả

1. API: Thêm voucher mới

- Phương thức:** POST
- Endpoint:** /api/voucher/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

2. API: Sửa voucher

- Phương thức:** PUT

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

6.3.26 Trang đăng ký Quản trị viên

Hình 6.35: Trang đăng ký Quản trị viên

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

6.3.26.1 Tổng quan

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

6.3.26.2 Quy tắc và Ràng buộc

1. **Kiểm tra vai trò Quản trị viên khi đăng ký:**
 - Quản trị viên cần nhập mã xác thực hoặc email đặc biệt để được cấp quyền.
2. **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.
3. **Phân quyền tự động:**
 - Sau khi đăng ký, tài khoản được gán vai trò Quản trị viên trong cơ sở dữ liệu.

6.3.26.3 Mô tả chi tiết

STT	Tên Trường	Mô tả	Kiểu điều khiển	Kiểu dữ liệu	Giá trị mặc định	Bắt buộc	Quy tắc
1	Tên đăng nhập	Tên đăng nhập của Quản trị viên	Textbox	varchar(50)		X	Quản trị viên 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 quản trị viên	Textbox	nvarchar(255)		X	Email phải đúng định dạng, duy nhất trong hệ thống.
3	Mật khẩu	Mật khẩu bảo vệ tài khoản quản trị viên	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 Quản trị viên	Mã xác thực để hệ thống nhận biết người đăng ký là Quản trị viên	Textbox	nvarchar(50)		X	Mã xác thực phải đúng với mã hệ thống đã cung cấp trước đó.
6	Đồng ý điều khoản	Checkbox xác nhận Quản trị viên đồng ý với các điều khoản và điều kiện sử dụng hệ thống	Checkbox	Boolean	False	X	Quản trị viên 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 Quản trị viên vào cơ sở dữ liệu, đồng thời gửi email xác nhận (nếu yêu cầu).

6.3.26.4 Quy trình xử lý

□ Kiểm tra thông tin nhập liệu:

- Hệ thống kiểm tra tính hợp lệ của tất cả các trường bắt buộc.
- Báo lỗi nếu trường nào không hợp lệ (hiển thị thông báo cụ thể cho từng lỗi).

□ Xử lý mã xác thực Quản trị viên:

- Hệ thống kiểm tra mã xác thực Quản trị viên với cơ sở dữ liệu.
- Nếu mã không đúng, từ chối đăng ký và hiển thị thông báo lỗi.

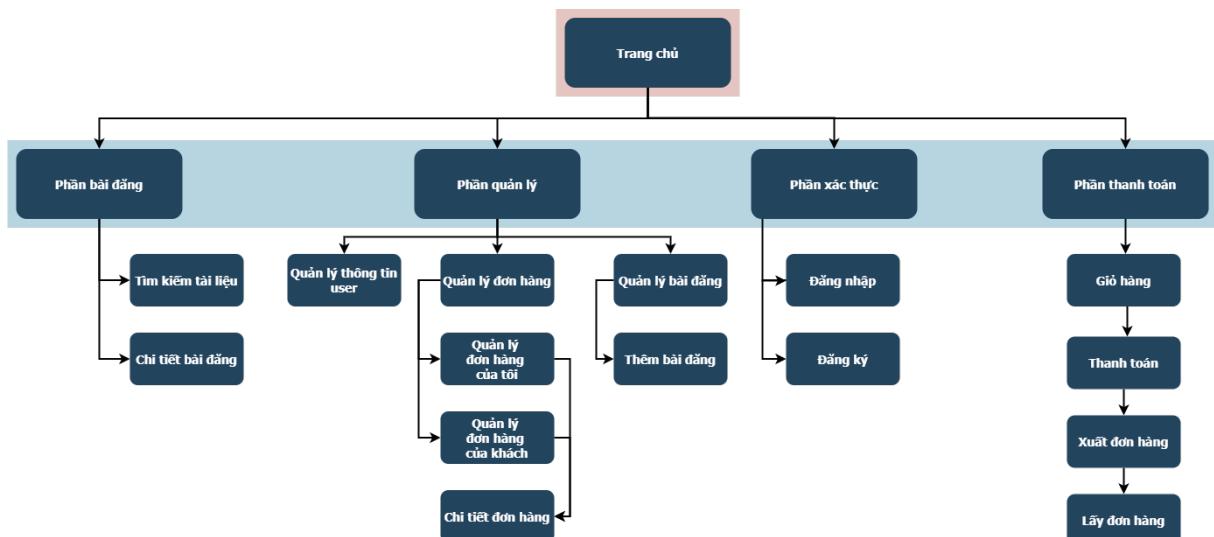
□ Gửi email xác nhận (tuỳ chọn):

- Sau khi đăng ký thành công, hệ thống có thể gửi email xác nhận đến email của Quản trị viên.

□ Phân quyền:

- Tài khoản sau khi đăng ký sẽ tự động được gán vai trò **Quản trị viên**.

6.4 Site map giao diện hệ thống



Hình 6.27: Site map giao diện hệ thống

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 7.1: 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 7.2: Logo playwright



Một số đặc điểm chính của Playwright:

- Hỗ trợ nhiều trình duyệt: Playwright không chỉ hỗ trợ một trình duyệt duy nhất mà còn hỗ trợ Chromium, Firefox và WebKit. Điều này cho phép bạn kiểm thử ứng dụng web của mình trên nhiều trình duyệt khác nhau từ một mã nguồn duy nhất.
- API mạnh mẽ và linh hoạt: Playwright cung cấp một API để 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

7.2 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 viết trong tab Test của postman.

The screenshot shows the Postman interface with a GET request to `localhost:8082/api/book/3`. The `Tests` tab is selected, displaying the following JavaScript code:

```
1 pm.test("Status code is 200", function(){
2     pm.response.to.have.status(200);
3 });
4
5 pm.test("Status code is 404", function(){
6     pm.response.to.have.status(404);
7 });
8
9 pm.test("check data", function(){
10    var jsonResponse = pm.response.json();
11    pm.expect(jsonResponse.isbn).to.eql("195153448");
12    pm.expect(jsonResponse.title).to.eql("Wallis");
13    pm.expect(jsonResponse.authors[0]).to.eql("Anne Edwards");
14    pm.expect(jsonResponse.languageCode).to.eql("en");
15    pm.expect(jsonResponse.genre[0]).to.eql("Historical");
16    pm.expect(jsonResponse.publisher).to.eql("Oxford University Press");
17    pm.expect(jsonResponse.publishedDate).to.eql("1991-01-01");
18    pm.expect(jsonResponse.pageCount).to.eql(478);
19    pm.expect(jsonResponse.size).to.eql("15.5x20.5");
20 });
21
22 pm.test("check data", function(){
23    var jsonResponse = pm.response.json();
24    pm.expect(jsonResponse.id).to.eql(3);
25 });
26
27
```

Below the code, the status bar shows `200 OK 42 ms 679 B` and `Save as example`.

Hình 7.3: Viết test case trong postman

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 a test script in the Tests tab. The script contains assertions for status code, JSON response fields, and specific values like author name and genre. Below the script, the Test Results section shows 3/3 assertions passed: Status code is 200, check data, and check data.

```

1 pm.test("Status code is 200", function(){
2     pm.response.to.have.status(200);
3 });
4
5 // pm.test("Status code is 404", function(){
6 //     pm.response.to.have.status(404);
7 // });
8
9 pm.test("check data", function(){
10    var jsonResponse = pm.response.json();
11    pm.expect(jsonPath).to.be.a.string();
12    pm.expect(jsonPath).to.be.a.string();
13    pm.expect(jsonPath).to.eql("Anne Edwards");
14    pm.expect(jsonPath).to.eql("en");
15    pm.expect(jsonPath).to.eql("Historical");

```

Body Cookies Headers (14) Test Results (3/3) 200 OK 42 ms 679 B Save as example ...

All Passed Skipped Failed ⌂

PASS Status code is 200
PASS check data
PASS check data

Hình 7.4: Kết quả run testcase

7.2.2 Kịch bản kiểm thử và kết quả kiểm thử

Kịch bản kiểm thử api dự án xây dựng hệ thống cho thuê sách cũ giá hợp lý				
tool: postman				
Nhóm api	Tên api	testcase thực hiện	Kết quả mong muốn	kết quả
user	Register	Kiểm tra status trả về	Status trả về là 200 (thành công)	pass
		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

		Kiểm tra id trả về	Có tạo được id người dùng và kết quả thành công trả về id của user	pass
		Kiểm tra đăng ký trùng pass	Status trả về 409 (đăng ký không thành công)	pass
		Kiểm tra trùng username	Status trả về 409 (đăng ký không thành công)	pass
		Kiểm tra message trả về khi lỗi đăng ký	Trả về message "Error creating new người dùng because reasons"	pass
		Kiểm tra không khai báo password	Trả về message "Error creating new người dùng because reasons" Status trả về 409	pass
	login	Kiểm tra status trả về	Status trả về là 200 (thành công)	pass
		Kiểm tra token trả về	Có trả về token	pass
		Kiểm tra sai mật khẩu	Status trả về là 401 Không có trường token nào được trả về	pass
		Kiểm tra sai username	Status trả về là 401 Không có trường token nào được trả về	pass
genre	getAllGenre	Kiểm tra status trả về	Status trả về là 200 (thành công khi đăng ký thành công)	pass
		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
book	searchBookBy Title	Kiểm tra status trả về	Status trả về là 200 (thành công)	pass
		Kiểm tra kết quả trả về với các từ khoá khác nhau (tim kiêm 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
		Kiểm tra kết quả trả về với từ khoá đặc biệt @, . ,#,123,...	Dữ liệu trả về list gồm đầy đủ các trường như mong muốn gồm: - isbn	pass

		(tìm kiếm thành công)	<ul style="list-style-type: none"> - title - authors - languageCode - genre - publisher - publishedDate - pageCount - size 	
		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
GetBookById		Kiểm tra status trả về	Status trả về là 200 (thành công)	pass
		Kiểm tra các trường data trả về	Dữ liệu trả về list gồm đầy đủ các trường như mong muốn gồm: <ul style="list-style-type: none"> - isbn - title - authors - languageCode - genre - publisher - publishedDate - pageCount - size 	pass
		Kiểm tra khi id không có trong cơ sở dl	Status trả về là 404	pass
		Kiểm tra id dữ liệu trả về	Trả về đúng id truyền vào (id có trong cơ sở dữ liệu)	pass
CreateBook		Kiểm tra status trả về	Status trả về là 200 (thành công)	pass
		Kiểm tra các trường data trả về	Dữ liệu trả về list gồm đầy đủ các trường như mong muốn gồm: <ul style="list-style-type: none"> - isbn - title - authors - languageCode - genre - publisher - publishedDate - pageCount - size 	pass
		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	pass

			<p>gồm:</p> <ul style="list-style-type: none"> - isbn - title - authors - languageCode - genre - publisher - publishedDate - pageCount - size 	
		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
	getAllBook	Kiểm tra status trả về khi thành công	Status trả về là 200 (khi thành công)	pass
		Kiểm tra dữ liệu trả về khi thành công	Dữ liệu trả về list gồm đầy đủ các trường như mong muốn gồm: <ul style="list-style-type: none"> - id - isbn - title - authors - languageCode - genre - publisher - publishedDate - pageCount - size 	pass
listing	getDetailListin gById	Kiểm tra status trả về	Status trả về là 200 (khi thành công)	pass
		Kiểm tra dữ liệu listing trả về	Trả về đầy đủ và đúng các trường mong muốn gồm : <ul style="list-style-type: none"> - id - user - address - leaseRate - depositFee - penaltyRate - description - listingStatus - copy - book - review - bookOwned - bookLeasing 	pass
		Kiểm tra id dữ liệu trả về	Trả về đúng id truyền vào	pass

	Kiểm tra status nếu không tìm thấy dữ liệu	Trả về status 204 (No Content)	pass
listing search	Kiểm tra status trả về	Status trả về là 200 (thành công)	pass
	Kiểm tra dữ liệu khi không param size phân trang	<p>Status trả về là 200 (thành công)</p> <p>size của một trang bằng 20 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
	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
	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</p>	pass

		<p>param truyền vào "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 	
	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 hiển thị trang 0 là trang đầu tiên</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
	Kiểm tra tìm dữ liệu với param title page size và page number	<p>Status trả về là 200 (thành công)</p> <p>size của một trang bằng 20 hiển thị trang 0 là trang đầu tiên</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 	pass

		<ul style="list-style-type: none"> - penaltyRate - description - listingStatus - copy - book - review - bookOwned - bookLeasing 	
	Kiểm tra tìm dữ liệu với param genre	<p>Status trả về là 200 (thành công)</p> <p>size của một trang bằng 20 hiển thị trang 0 là trang đầu tiên</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
	Kiểm tra tìm dữ liệu khi không có param title và genre	<p>Status trả về là 200 (thành công)</p> <p>size của một trang bằng 20 hiển thị trang 0 là trang đầu tiên</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

		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
		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
listing search by id and title		Kiểm tra status trả về	Status trả về là 200 (thành công)	pass
		Kiểm tra dữ liệu khi request title bởi nhiều từ khoá: @, . ,#,123,... (tìm kiếm 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 - penaltyRate - description - listingStatus - copy - book - review - bookOwned - bookLeasing	pass

		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
create listing		Kiểm tra status trả về khi tạo thành công	Status trả về là 200 (thành công)	pass
		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
		Kiểm tra id trả khi tạo thành công	Có trả về trường id khi tạo thành công	pass
copy	create copy	Kiểm tra status trả về khi tạo thành công	Status trả về là 200 (thành công)	pass
		Kiểm tra id trả khi tạo thành công	Có trả về trường id khi tạo thành công	pass
		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
	get copy by id	Kiểm tra status trả về khi thành công	Status trả về là 200 (thành công)	pass
		Kiểm tra dữ liệu khi tạo thành công	Trả về đầy đủ các trường mong muốn gồm: - id - score - description - leaseOrderId - userId - listingId - createdDate	pass

			Dữ liệu trả về giống với dữ liệu truyền từ body request	
get copy by ownerid		Kiểm tra status trả về khi thành công	Status trả về là 200 (thành công)	pass
		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
review	create review	Kiểm tra status trả về khi tạo thành công	Status trả về là 200 (thành công)	pass
		Kiểm tra id trả về khi tạo thành công	Có trả về trường id khi tạo thành công	pass
		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
leaseOrder	create new order	Kiểm tra status trả về khi tạo thành công	Status trả về là 200 (thành công)	pass
		Kiểm tra id trả về khi tạo thành công	Có trả về trường id khi tạo thành công	pass
		Kiểm tra dữ liệu trả về khi thành công	Trả về các trường sau đúng với dữ liệu body request truyền vào: - listingId - status - lesseeId - lesseeAddress - fromDate - toDate - paymentMethod	pass
		Kiểm tra tạo thành công lease order details	Trả về có leaseOrderdetail với các trường mong muốn đó là : - id - title	pass

		<ul style="list-style-type: none"> - listingId - leaseRate - depositFee - penaltyRate 	
	Kiểm tra tạo thành công lease order details với các loại method payment (COD, VNPAY, BANK_TRANSFE R),	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	pass
	Kiểm tra tạo thành công lease order details với ngày tạo là ngày hiện tại	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	pass
	Kiểm tra tạo thành công lease order details, tạo được leaseAndDepositPay mentId	Trả về có leaseOrderdetail với các trường mong muốn như tạo thành như tạo thành công. Trả về có giá trị của trường leaseAndDepositPaymentId	pass
	Kiểm tra tạo thành công lease order details, kiểm tra tổng tiền phạt ban đầu	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.	pass
	Kiểm tra tạo thành công lease order details, kiểm tra tổng phí thuê	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 bussiness logic	pass
	Kiểm tra tạo thành công lease order details, kiểm tra tiền cọc	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	pass
get lease order by lessor id	Kiểm tra status trả về khi thành công	Status trả về là 200 (thành công)	pass
	Kiểm tra dữ liệu trả ra khi thành công	Trả về các trường sau trong mỗi phần tử của list: - leaseOrder	pass

			- listing - lessor	
		Kiểm tra lessor id trả về khi tìm kiếm thành công có giá trị	Phần tử thứ nhất trong dãy trả về có giá trị lessor id bằng với lessor id truyền vào	pass
		Kiểm tra khi tìm kiếm thành công không có giá trị	Trả về list rỗng	pass
get lease order by lesse id	Kiểm tra status trả về khi thành công	Kiểm tra status trả về là 200 (thành công)	pass	
		Trả về các trường sau trong mỗi phần tử của list: - leaseOrder - listing - lessor	pass	
	Kiểm tra lesse id trả về khi tìm kiếm thành công có giá trị	Phân tử thứ nhất trong dãy trả về có giá trị lesse id bằng với lesse id truyền vào	pass	
		Trả về list rỗng	pass	
get lease order by lessor id and status	Kiểm tra status trả về khi thành công	Status trả về là 200 (thành công)	pass	
		Trả về các trường sau trong mỗi phần tử của list: - leaseOrder - listing - lessor	pass	
	Kiểm tra lessor id trả về khi tìm kiếm thành công có giá trị	Phân tử thứ nhất trong dãy trả về có giá trị lessor id bằng với lessor id truyền vào	pass	
		Trả về list rỗng	pass	
	Kiểm tra với từng status lựa chọn	Trả về đúng status theo business logic	pass	
get lease order by lesse id and status	Kiểm tra status trả về khi thành công	Status trả về là 200 (thành công)	pass	
		Trả về các trường sau trong mỗi phần tử của list: - leaseOrder	pass	

			- listing - lessor	
	Kiểm tra lesse id trả về khi tìm kiếm thành công có giá trị	Phần tử thứ nhất trong dãy trả về có giá trị lesse id bằng với lesse id truyền vào	pass	
	Kiểm tra khi tìm kiếm thành công không có giá trị	Trả về list rỗng	pass	
	Kiểm tra với từng status lựa chọn	Trả về đúng status theo business logic	pass	
get leaseOrderById	Kiểm tra status trả về khi thành công	Status trả về là 200 (thành công)	pass	
	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	pass	
	Kiểm tra trường id trả về khi thành công	Trả về id giống với id truyền vào khi thành công	pass	
change status	Kiểm tra status trả về khi thành công	Status trả về là 200 (thành công)	pass	
	Kiểm tra status trả về khi không cho phép đổi status	Status trả về là 423 message: Error update status	pass	
	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	pass	

Bảng 7.1: Kịch bản kiểm thử và kết quả của kiểm thử thủ công

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

Kịch bản kiểm thử tổng hợp dự án xây dựng hệ thống cho thuê sách cũ giá hợp lý				
tool: playwright				
Mã TC	Tên test case	Tiền điều kiện	Các bước thực hiện	Kết quả mong muốn
LOGIN				
đăng ký	đăng ký	truy cập trang web tại địa chỉ: http://localhost:3000/	1. Nhấn nút "Đăng ký" trên thanh navbar 2. Điền tên đăng nhập, mật khẩu, nhập lại mật khẩu 3. Nhấn nút "Đăng ký"	Đăng ký thành công tài khoản
đăng nhập	đăng nhập	Đăng nhập acc dev1/123456 thành công	1. Nhấn nút "Đăng nhập" trên thanh navbar 2. Điền username, password 3. Click nút "Đăng nhập"	Đăng nhập tài khoản thành công
đăng xuất	đăng xuất	Đăng nhập acc dev1/123456 thành công	1. Nhấn avatar user 2. Chọn đăng xuất	Đăng xuất thành công
chức năng user				
xem thông tin user	Xem thông tin user	Đăng nhập acc dev1/123456 thành công	1. Nhấn avatar user 2. Chọn "chào + username"	Xem thông tin người dùng thành công
Chỉnh sửa thông tin user	Chỉnh sửa thông tin user	Đăng nhập acc dev1/123456 thành công	1. Nhấn avatar user 2. Chọn "chào + username" 3. Thêm thông tin và trường thông tin muốn thay	Thay đổi thông tin người dùng thành công

			đổi 4. Nhấn nút "xác nhận"	
--	--	--	-------------------------------	--

chức năng bài đăng

tìm kiếm bài đăng theo title	Tìm kiếm bài đăng theo title	Đăng nhập acc dev1/123456 thành công Hoặc không cần đăng nhập	1. Điền thông tin keyword vào input tìm kiếm 2. Click nút tìm kiếm	Hệ thống hiển thị danh sách tìm kiếm thành công
tìm kiếm bài đăng theo genre	Tìm kiếm bài đăng theo genre	Đăng nhập acc dev1/123456 thành công Hoặc không cần đăng nhập	1. Nhấn nút mũi tên xuống bên trái input tìm kiếm 2. Chọn thể loại muốn tìm kiếm	Hệ thống hiển thị danh sách tìm kiếm thành công
xem chi tiết bài đăng	Xem chi tiết bài đăng	Đăng nhập acc dev1/123456 thành công Hoặc không cần đăng nhập	1. Tìm kiếm bài đăng 2. Nhấn vào bài đăng	Hệ hiển thị chi tiết bài đăng thành công
thêm bài đăng vào giỏ hàng	Thêm bài đăng vào giỏ hàng	Đăng nhập acc dev1/123456 thành công Hoặc không cần đăng nhập	1. Tìm kiếm bài đăng 2. Nhấn vào bài đăng 3. Chọn ngày nhận - ngày trả 4. Nhấn nút "Đặt thuê sách"	Thêm đơn hàng vào giỏ hàng thành công
xoá bài đăng khỏi giỏ hàng	Xoá bài đăng khỏi giỏ hàng	Đăng nhập acc dev1/123456 thành công Hoặc không cần đăng nhập	1. Tìm kiếm bài đăng 2. Nhấn vào bài đăng 3. Chọn ngày nhận - ngày trả 4. Nhấn nút "Đặt thuê sách" 5. Nhấn icon thùng rác bên cạnh bài đăng	Xoá bài đăng khỏi giỏ hàng thành công
đặt thuê	Tạo đơn hàng	Đăng nhập acc dev3/123456 thành công	1. Tìm kiếm bài đăng 2. Nhấn vào bài đăng 3. Chọn ngày nhận - ngày trả 4. Nhấn nút "Đặt thuê sách" 5. Nhấn nút "Đặt thuê hàng" 6. Sửa thông tin đặt thuê 7. Bấm xác nhận 8. Chọn phương thức thanh toán >> thao tác thanh toán 9. Nhấn nút "tạo đơn hàng"	Tạo đơn hàng thành công

chức năng quản lý bài đăng

xem danh sách bài đăng	Xem danh sách bài đăng trong	Đăng nhập acc dev1/123456 thành công	1. Nhấn icon quản lý bài đăng trên thanh navbar	Xem danh sách bài đăng trong quản lý bài
------------------------	------------------------------	--------------------------------------	---	--

	quản lý bài đăng			đăng thành công
chọn book	Chọn book trong tạo bài đăng	Đăng nhập acc dev1/123456 thành công	1. Nhấn icon quản lý bài đăng trên thanh navbar 2. Nhấn nút thêm bài đăng 3. Trong mục sách nhấn nút tìm sách và chọn sách 4. Bấm nút "chọn sách"	Chọn sách trong tạo bài đăng thành công
tạo copy	Tạo copy trong tạo bài đăng	Đăng nhập acc dev1/123456 thành công	1. Nhấn icon quản lý bài đăng trên thanh navbar 2. Nhấn nút thêm bài đăng 3. Trong mục sách nhấn nút tìm sách và chọn sách 4. Bấm nút "chọn sách" 5. Trong mục tài liệu >> thêm phần trăm hư hại >> thêm hình ảnh 6. Bấm nút tạo tài liệu	Hệ thống tạo tài liệu trong bài đăng thành công
tạo bài đăng	Tao bài đăng	Đăng nhập acc dev1/123456 thành công	1. Nhấn icon quản lý bài đăng trên thanh navbar 2. Nhấn nút thêm bài đăng 3. Trong mục sách nhấn nút tìm sách và chọn sách 4. Bấm nút "chọn sách" 5. Trong mục tài liệu >> thêm phần trăm hư hại >> thêm hình ảnh 6. Bấm nút tạo tài liệu 7. Tại mục bài đăng thêm mô tả trạng thái, tiền cọc, giá thuê và phí phạt, địa chỉ cho thuê 8. Nhấn nút "tạo bài đăng"	Hệ thống tạo bài đăng thành công

chức năng quản lý đơn hàng của tôi

xem danh sách đơn hàng	Xem danh sách đơn hàng của tôi	Đăng nhập acc dev1/123456 thành công	1. Nhấn icon quản lý đơn hàng chọn đơn hàng của tôi	Xem danh sách đơn hàng của tôi thành công
xác nhận nhận sách	Xác nhận nhận sách từ chủ thuê	Đăng nhập acc dev1/123456 thành công	1. Nhấn icon quản lý đơn hàng chọn đơn hàng của tôi 2. Ở đơn hàng trạng thái lấy hàng >> Nhấn vào nút "Đã nhận được hàng"	Xác nhận nhận hàng thành công, đơn hàng chuyển trạng thái đã nhận hàng
xác nhận trả cọc	Xác nhận trả tiền	Đăng nhập acc dev1/123456 thành công	1. Nhấn icon quản lý đơn hàng chọn đơn hàng của tôi	Xác nhận đã trả tiền

	cọc cho hệ thống		2. Ở đơn hàng trạng thái chờ thanh toán >> Nhấn vào nút "Đã trả tiền"	cọc thành công
xác nhận trả sách	Xác nhận trả sách cho chủ sách	Đăng nhập acc dev1/123456 thành công	1. Nhấn icon quản lý đơn hàng chọn đơn hàng của tôi 2. Ở đơn hàng trạng thái đã lấy hàng >> Nhấn vào nút "Đã trả sách"	Xác nhận đã trả sách thành công
đánh giá đơn hàng	Đánh giá đơn hàng	Đăng nhập acc dev1/123456 thành công	1. Nhấn icon quản lý đơn hàng chọn đơn hàng của tôi 2. Ở đơn hàng trạng thái đã trả sách >> Nhấn vào nút "Đánh giá đơn hàng" 3. Thêm mô tả và cho số điểm >> bấm "đồng ý"	Hệ thống đánh giá đơn hàng thành công

chức năng quản lý đơn hàng của khách

xem danh sách đơn hàng	Xem danh sách đơn hàng của tôi	Đăng nhập acc dev1/123456 thành công	1. Nhấn icon quản lý đơn hàng chọn đơn hàng của khách	Xem danh sách đơn hàng của khách
xác nhận lại nhận sách	Kiểm tra xuất word mẫu đơn vị	Đăng nhập acc vt_anhnh/123456aA@ thành công	1. Nhấn icon quản lý đơn hàng chọn đơn hàng của tôi 2. Ở đơn hàng trạng thái đang trả sách >> Nhấn vào nút "Đã nhận lại sách"	Xác nhận đã nhận lại sách thành công

Bảng 7.2: Kịch bản kiểm thử tự động

7.3.3 Report

Project: chromium		All 22	Passed 22	Failed 0	Flaky 0	Skipped 0
		5/17/2024, 4:17:50 PM Total time: 4.2m				
✓	test-1.spec.ts					
✓	đăng nhập					9.0s
	test-1.spec.ts:3					
✓	test-10.spec.ts					
✓	them bai dang vao gio hang					6.2s
	test-10.spec.ts:3					
✓	test-11.spec.ts					
✓	dat thue					23.7s
	test-11.spec.ts:3					
✓	test-12.spec.ts					
✓	xem danh sach bai dang- quan ly bai dang					8.5s
	test-12.spec.ts:3					
✓	test-13.spec.ts					
✓	chon book					10.4s
	test-13.spec.ts:3					
✓	test-14.spec.ts					
✓	tao copy					9.2s
	test-14.spec.ts:3					
✓	test-15.spec.ts					
✓	tao bai dang					9.8s
	test-15.spec.ts:3					
✓	test-16.spec.ts					
✓	xem danh sach don hang cua toi					8.8s
	test-16.spec.ts:3					
✓	test-17.spec.ts					
✓	xac nhan nhan sach					9.1s
	test-17.spec.ts:3					
✓	test-18.spec.ts					
✓	xac nhan tra coc					9.4s
	test-18.spec.ts:3					
✓	test-19.spec.ts					
✓	xac nhan tra lai sach					9.2s
	test-19.spec.ts:3					
✓	test-2.spec.ts					
✓	đăng ký					14.0s
	test-2.spec.ts:3					

✓ test-20.spec.ts	
✓ danh gia don hang	10.0s
test-20.spec.ts:3	
✓ test-21.spec.ts	
✓ xem danh sach don hang cua khach	9.4s
test-21.spec.ts:3	
✓ test-22.spec.ts	
✓ xac nhan nhan lai sach	10.0s
test-22.spec.ts:3	
✓ test-3.spec.ts	
✓ xem thong tin user	10.0s
test-3.spec.ts:3	
✓ test-4.spec.ts	
✓ dang xuat	10.6s
test-4.spec.ts:3	
✓ test-5.spec.ts	
✓ sua thong tin user	18.7s
test-5.spec.ts:3	
✓ test-6.spec.ts	
✓ tim kiem bai dang theo title	8.8s
test-6.spec.ts:3	
✓ test-7.spec.ts	
✓ tim kiem bai dang theo genre	8.1s
test-7.spec.ts:3	
✓ test-8.spec.ts	
✓ xem chi tiet bai dang	13.7s
test-8.spec.ts:3	
✓ test-9.spec.ts	
✓ xoa bai dang khoi gio hang	21.9s
test-9.spec.ts:3	

Hình 7.5: Báo cáo auto test

KẾT LUẬN

Nhóm đã hoàn thành các nhiệm vụ được đề đúng thời hạn ra bao gồm:

- Tìm hiểu các hệ thống tương tự và xác định yêu cầu nghiệp vụ của hệ thống
- Tìm hiểu các công nghệ hiện có trên thị trường để phát triển hệ thống
- Thiết kế kiến trúc hệ thống
- Thiết kế các prototype của hệ thống
- Tìm hiểu các công nghệ có thể dùng trong hệ thống
- Hoàn thiện việc phân tích và thiết kế hệ thống.
- Hiện thực đầy đủ các tính năng của hệ thống.
- Tiến hành kiểm thử và đánh giá hệ thống.

Khả năng mở rộng của dự án:

- Có thể tích hợp việc bán đứt sách ngoài việc cho thuê.
- Tích hợp thêm các chức năng cho Quản trị viên để nâng cao khả năng làm việc và giảm thiểu thời gian tiêu tốn.
- Tích hợp các chức năng giao tiếp giữa người dùng trong hệ thống như nhắn tin.

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>.