

**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 TUẤN ANH  
**TKHĐ:** ThS HOÀNG LÊ HẢI THANH  
**GVPB:** ThS NGUYỄN MINH TÂM  
ThS LÊ ĐÌNH THUẬN  
---o0o---

**SVTH1: NGUYỄN THỊ THU THẢO (2033732)**

TP. HỒ CHÍ MINH, THÁNG 5/2024

## 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 5/2024

Nguyễn Thị Thu Thảo

## 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 chỉ 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 tò mò 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 user, sửa thông tin user. Ngoài ra hệ thống còn có giao diện để admin 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 .....	13
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 .....	21
3.1.3    Material UI .....	23
3.2    Database sử dụng PostgreSQL .....	24
3.2.1    Giới thiệu về PostgreSQL .....	24
3.2.2    Ưu điểm .....	25
3.2.3    Nhược điểm .....	25
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 .....	32
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 admin.....	33
4.1.3	Nhu cầu người dùng .....	33
4.1.4	Nhu cầu khách vãng lai .....	33
4.2	Yêu cầu chức năng .....	33
4.2.1	Chức năng chung của khách và người dùng.....	33
4.2.2	Chức năng riêng dành cho người dùng có tài khoản .....	34
4.2.3	Các chức năng ở Administrator .....	34
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.....	36
4.3.4	Hệ thống hỗ trợ đa ngôn ngữ .....	37
4.3.5	Hiệu suất.....	37
4.3.6	Khả năng bảo trì .....	37
4.3.7	Hệ thống có giao diện dễ nhìn, đồng đều, vừa mắt, dễ sử dụng.....	37
4.4	Usecase diagram cho toàn hệ thống .....	38
4.5	Đặc tả Usecase.....	38
4.5.1	Usecases cho User .....	38
4.5.2	Use Cases chung cho Admin và User.....	51
4.5.3	Use Cases cho Admin.....	52
4.5.4	Use Cases chung cho User và Khách .....	54
4.6	Mô hình kiến trúc hệ thống .....	58
4.7	Layered architecture diagram .....	59
4.7.1	Cho User và khách.....	59
4.8	Sequence diagram cho các use case .....	60
4.8.1	Với actor Admin .....	60
4.8.2	Với actor User và Khách .....	61
4.9	Activity diagram (Lược đồ hoạt động) .....	62
4.10	Lược đồ quan hệ cơ sở dữ liệu (ERD).....	63

4.11	Class diagram .....	64
CHƯƠNG 5:	HIỆN THỰC BACK-END .....	67
5.1	Hiện thực database.....	67
5.2	Hiện thực api .....	68
5.2.1	Lớp controller và dto .....	68
5.2.2	Lớp Entity .....	69
5.2.3	Lớp Mapper và repository .....	70
5.2.4	Lớp service .....	71
5.3	Api xây dựng được trong dự án.....	72
CHƯƠNG 6:	HIỆN THỰC FRONT END .....	74
6.1	Hiện thực front -end .....	74
6.1.1	Thư mục app .....	74
6.1.2	Thư mục components .....	75
6.1.3	Thư mục hooks .....	77
6.1.4	Thư mục types .....	78
6.1.5	Thư mục utils.....	78
6.2	Logo trang web.....	78
6.2.1	Logo.....	78
6.3	Giao diện .....	79
6.3.1	Trang chủ.....	81
6.3.2	Trang tìm kiếm và hiển thị danh sách bài đăng .....	83
6.3.3	Trang chi tiết bài đăng .....	84
6.3.4	Trang giỏ hàng.....	87
6.3.5	Trang đặt hàng .....	88
6.3.6	Trang thanh toán vnpay .....	90
6.3.7	Trang xuất đơn hàng.....	91
6.3.8	Trang lấy đơn hàng.....	92
6.3.9	Trang đăng nhập .....	93
6.3.10	Trang đăng ký .....	93
6.3.11	Trang quản lý thông tin user.....	94
6.3.12	Trang quản lý đơn hàng của tôi .....	94
6.3.13	Quản lý đơn hàng của khách .....	96
6.3.14	Chi tiết đơn hàng .....	97
6.3.15	Quản lý bài đăng.....	97
6.3.16	Tạo bài đăng .....	98
6.3.17	Trạng đăng nhập admin .....	99
6.3.18	Trang quản lý admin.....	100
6.3.19	Trang đăng ký admin.....	100
6.4	Site map giao diện hệ thống .....	101

CHƯƠNG 7: KIỂM THỬ.....	102
7.1     Công cụ kiểm thử .....	102
7.1.1     Công cụ hỗ trợ kiểm thử postman .....	102
7.1.2     Công cụ hỗ trợ kiểm thử playwright.....	103
7.2     Kiểm thử api .....	104
7.2.1     Cách thực hiện .....	104
7.2.2     Kịch bản kiểm thử và kết quả kiểm thử.....	105
7.3     Kiểm thử tổng hợp.....	117
7.3.1     Cách thực hiện .....	117
7.3.2     Kịch bản kiểm thử .....	118
7.3.3     Report .....	122
KẾT LUẬN .....	124
TÀI LIỆU THAM KHẢO .....	125

## 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 user quản lý tài khoản .....	38
Hình 4.4:Module xem danh sách giỏ hàng .....	40
Hình 4.5:Module quản lý đơn hàng .....	42
Hình 4.6 Use case quản lý đơn hàng của khách .....	48
Hình 4.7:Module quản lý bài đăng .....	49
Hình 4.8:Module quản lý đơn hàng-admin .....	52
Hình 4.9:Mô hình kiến trúc hệ thống MVC .....	58
Hình 4.10: Layered architecture diagram cho user và khách .....	59
Hình 4.11:Sequence diagram cho admin .....	60
Hình 4.12:Sequence diagram cho user và khách .....	61
Hình 4.13: Activity diagram thuê và trả tài liệu .....	62
Hình 4.14:ERD cho toàn hệ thống.....	63
Hình 4.15: Classdiagram cho hệ thống.....	65
Hình 5.1: ERD cơ sở dữ liệu init.....	68
Hình 5.2: Cây thư mục lớp controller và dto của book .....	69
Hình 5.3: Cây thư mục lớp entity của book.....	70
Hình 5.4: Cây cấu trúc thư mục của job, mapper, repository của book .....	71
Hình 6.1:Phiên bản logo không chữ .....	79
Hình 6.2:Logo với tên dự án The Flying Bookstore .....	79
Hình 6.3:Trang chủ .....	81
Hình 6.4: Banner ở trang chủ .....	82
Hình 6.5:Section sách mới ra mắt ở trang chủ .....	82
Hình 6.6: Trang danh sách bài đăng và tìm kiếm tài liệu .....	83
Hình 6.7:Trang chi tiết bài đăng .....	84
Hình 6.8:Mô tả cuốn sách .....	85
Hình 6.9: Tab chi tiết cuốn sách .....	85
Hình 6.10: Section đặt thuê .....	86
Hình 6.11: Tab đánh giá .....	86
Hình 6.12:Trang giỏ hàng .....	87
Hình 6.13: Trang đặt hàng .....	88
Hình 6.14: Phần thông báo chuyển khoản trang đặt hàng .....	89
Hình 6.15: Chọn phương thức thanh toán .....	90
Hình 6.16: Tiến hành thanh toán .....	90
Hình 6.17: Trang xuất đơn hàng .....	91
Hình 6.18: Trang lấy đơn hàng .....	92
Hình 6.19: Trang đăng nhập .....	93
Hình 6.20: Trang đăng ký .....	93
Hình 6.21: Trang quản lý thông tin user.....	94
Hình 6.22: Trang quản lý đơn hàng của tôi .....	94
Hình 6.23: Đơn hàng có thể cancel .....	95

Hình 6.24: Đơn hàng có thể xác nhận thanh toán .....	95
Hình 6.25: Đơn hàng xác nhận lấy hàng .....	95
Hình 6.26: Đơn hàng xác nhận trả sách.....	96
Hình 6.27: Quản lý đơn hàng của khách .....	96
Hình 6.28: Chủ sách xác nhận nhận lại sách .....	96
Hình 6.29: Chi tiết đơn hàng .....	97
Hình 6.30: Quản lý bài đăng.....	97
Hình 6.31: Tạo bài đăng .....	98
Hình 6.32: Thêm mới sách .....	99
Hình 6.33: Trang đăng nhập admin .....	99
Hình 6.34: Site map giao diện hệ thống .....	101
Hình 7.1: Cấu trúc Postman.....	102
Hình 7.2: Logo playwright .....	103
Hình 7.3: Viết test case trong postman.....	104
Hình 7.4: Kết quả run testcase.....	105
Hình 7.5: Báo cáo auto test.....	123

## **DANH MỤC BẢNG BIỂU**

Bảng 5.1: Api thực hiện trong dự án .....	73
Bảng 7.1: Kịch bản kiểm thử và kết quả của kiểm thử thủ công.....	117
Bảng 7.2: Kịch bản kiểm thử tự động .....	121

## **GANTT CHART QUÁ TRÌNH LÀM VIỆC**

## CHƯƠNG 1: TỔNG QUAN

### 1.1 Giới thiệu đề tài

Hằng 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 đổi 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 admin của một page trên facebook. Page được tạo ra với mục đích trao đổi sách giữa các bạn sinh viên, học sinh. Khi các bạn sinh viên học xong một môn hoặc các bạn học sinh học xong một lớp, để tối ưu hóa chi phí thì sách đó sẽ được trao đổi giữa các bạn. Kết quả là mọi người đều đảm bảo có sách để học, các ghi chú của các bạn trước cũng được truyền lại cho các bạn học sau. Ban đầu page diễn ra trơn tru và ổn thoả, nhưng sau đó phát sinh vấn đề lớn bởi lượng công việc của admin quá nhiều. Admin 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 sách 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 học sinh/ sinh viên đem cho thuê, mượn tài liệu của mình. Đồng thời giúp quản lý quá trình cho thuê, mượn.

- Là nơi người dùng có thể thuê mượn tài liệu của người khác. Quản lý đơn hàng thuê mướn của mình.

### **1.3 Phạm vi đề tài**

Vì lý do hạn chế về thời gian và con người, nên 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 user của hệ thống thì mới có thể phát sinh đơn thuê với những user 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 giữa người chủ sở hữu và một user 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 người thuê, người cho thuê 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.
- Ngoài ra dự án cung cấp thông tin ban quản trị để người dùng có thể khiếu nại trực tiếp đến những kênh giao tiếp của ban quản trị như (gmail, số điện thoại).
- Dự án cung cấp chức năng về thông tin user, 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 admin đối với đơn hàng: với mục đích ban đầu là giảm bớt áp lực gánh nặng cho admin. Các chức năng trên đã giúp admin giảm tải việc tự đăng bài và tự liên hệ với

bên thuê và bên cho thuê. Giờ đây admin sẽ tập trung nhiều vào quản lý đơn hàng bằng giao diện duyệt đơn hàng của admin, không bị quá tải bởi những việc ngoài luồng khác.

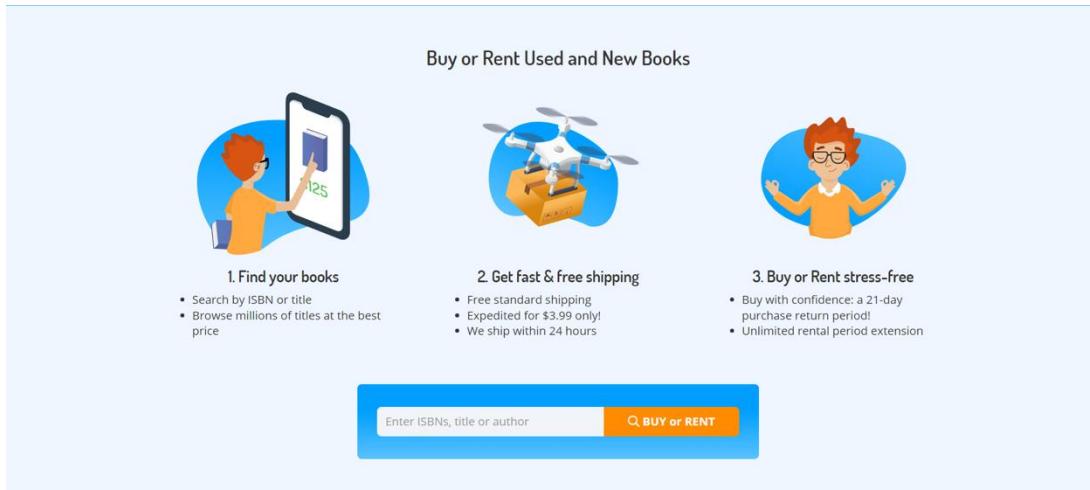
- Đè hạn chế việc mượn không trả thì hệ thống tiền hành dựa trên phí cọc, và phí phạt. Bên trong hệ thống ngày nhận sẽ được cập nhật lại khi người thuê xác nhận nhận hàng. Ngày trả là mốc giao kèo cứng với người cho thuê. Trả sớm thì sẽ được cập nhật lại ngày và tổng tiền thuê. Quá hạn sẽ không được cập nhật lại và bị tính phí phạt. Đây là mốc để hạn chế người thuê trả trễ.
- 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, lọt 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

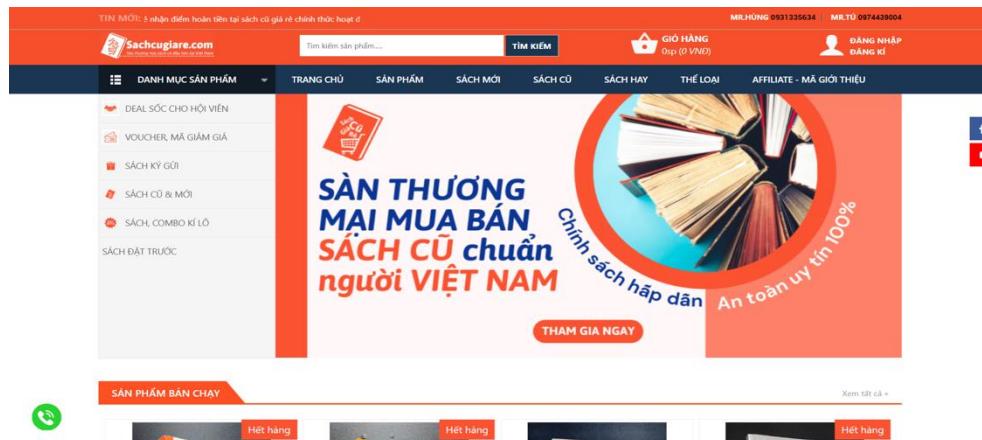
- Hệ thống giao diện được đầu tư kỹ càng, trực quan, dễ sử dụng.
- Sử dụng quét mã ISBN để tìm kiếm sách nhanh hơn, vì hệ thống đầu tư data sách thời gian thực đồ sộ có sẵn với mã ISBN.

#### 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.

TIN MỚI: tại nhà

MR.HÙNG 0931335634 | MR.TÚ 0974439004

Sachcugiare.com  
Sách Thương Mại Điện Tử Đầu Tiên Tự Việt Hóa

Tim kiếm sản phẩm....

TÌM KIẾM

GİỎ HÀNG  
1sp (7.500 VNĐ)

ĐĂNG NHẬP  
ĐĂNG KÝ

DANH MỤC SẢN PHẨM TRANG CHỦ SẢN PHẨM SÁCH MỚI SÁCH CŨ SÁCH HAY THẺ LOẠI AFFILIATE - MÃ GIỚI THIỆU

**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.

**ĐĂNG NHẬP**

Tên đăng nhập hoặc Email:

Mật khẩu:

Đăng nhập

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.

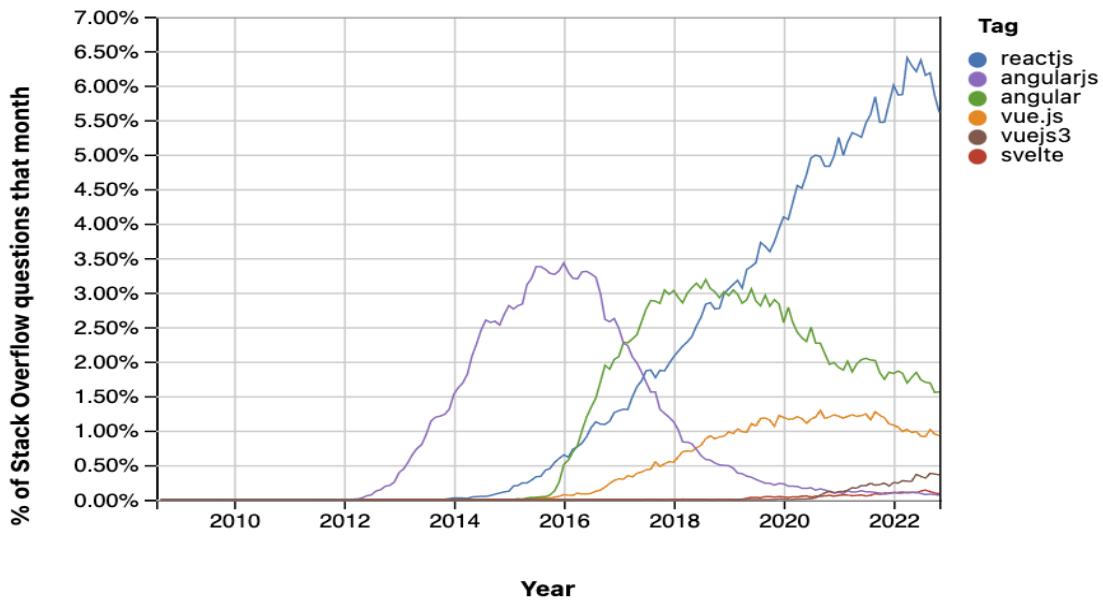
## CHƯƠNG 3: CÔNG NGHỆ SỬ DỤNG

### 3.1 Công nghệ ở Front-end

#### 3.1.1 ReactJS

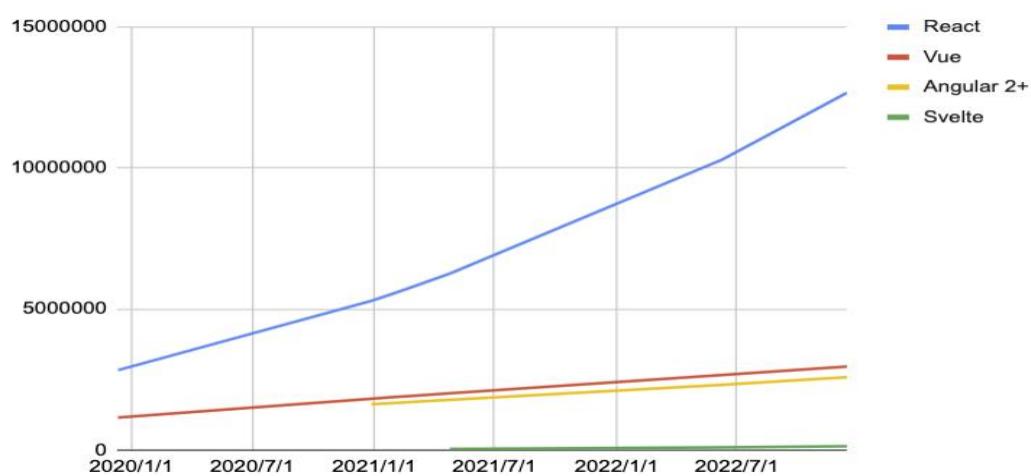
##### 3.1.1.1 Vì sao chọn ReactJS?

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 Svelte<sup>1</sup>.

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 dùng<sup>2</sup>. React ra đời từ Facebook vào năm 2013, tức là ra đời sau AngularJS. Ban đầu React sinh ra để dành cho các ứng dụng lớn dễ dàng quản lý và mở rộng. Mục tiêu chính của React là xây dựng website ở phía Frontend nhanh, đơn giản, hiệu năng cao và dễ dàng mở rộng. ReactJS có các tính năng nổi trội như sau:

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

### 3.1.1.3 Ưu điểm

- Declarative (khai báo): React giúp việc tạo giao diện người dùng trở nên dễ dàng hơn, nhanh chóng hơn. React sẽ cập nhật và hiển thị đúng các component được chỉ định một cách hiệu quả khi dữ liệu của bạn thay đổi. Declarative view làm cho code của bạn dễ dự đoán hơn và dễ debug hơn.
- Component - Based (Xây dựng dựa trên thành phần): Xây dựng các components được đóng gói để quản lý trạng thái của riêng chúng, sau đó kết hợp chúng để tạo các giao diện phức tạp. Vì logic thành phần được viết bằng JavaScript thay vì templates, nên bạn có thể dễ dàng truyền nhiều dạng data qua ứng dụng của mình và loại bỏ trạng thái khỏi DOM.

---

<sup>1</sup> 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>.

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

- Học 1 lần, viết được mọi nơi: React có thể render trên server dùng Node hoặc mobile app với React Native<sup>3</sup>. 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 Reddit<sup>4</sup>
- 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.<sup>5</sup>

#### *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 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

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

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

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

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*

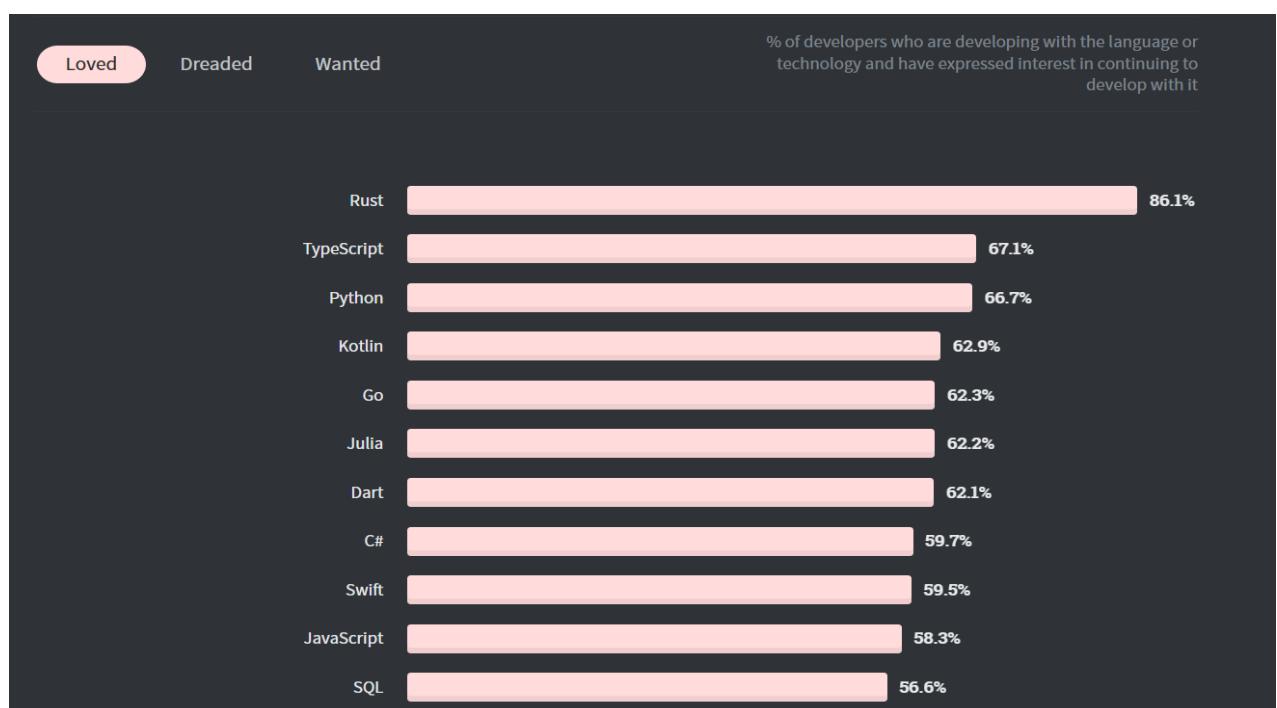
#### 3.1.2.1    *Khái niệm*

TypeScript là ngôn ngữ lập trình mạnh mẽ được xây dựng trên JavaScript, là công cụ lập trình tốt hơn Javascript<sup>6</sup>.

- 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.

#### 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

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

Trong State of JS<sup>7</sup> 2020 TypeScript đã có 78% số người sử dụng trả lời, trong đó 93% cho biết họ sẽ sử dụng lại nó.<sup>8</sup>

- TypeScript cho phép chỉ định các loại data: JavaScript không quy ước các loại data được truyền vào, do vậy làm cho developer phải mất thời gian xem tài liệu và dự đoán loại data. Nhưng với TypeScript, cho phép chỉ định các loại data được truyền vào trong code và có khả năng báo lỗi khi các loại data không khớp.
- Đối với ứng dụng nhỏ hoặc trang web đơn giản, TypeScript có thể sẽ hơi cồng kềnh. Nhưng đối với dự án lớn, việc phải đảm bảo không có lỗi trong code được xử lý bằng thủ công hoặc viết automation test khá mất thời gian. Ngoài ra dự án có đến hàng trăm hay hàng ngàn file, việc thay đổi một vài đoạn code nhỏ có thể gây ra sự cố lên hàng chục file khác. Vậy nên việc xác thực kết nối giữa mọi phần trong toàn bộ source code có thể tiêu tốn rất nhiều thời gian. Những ngôn ngữ kiểm tra được kiểu của data như TypeScript có thể tự động xử lý việc đó và cung cấp phản hồi tức thì trong quá trình code<sup>9</sup>.

#### 3.1.2.3 *Ưu điểm*

- Static typing (kiểu 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.<sup>10</sup>
- 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.

---

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

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

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

<sup>10</sup> 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/>.

- 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 hiểu<sup>11</sup>. Với những dự án gấp gáp việc bỏ qua type nhiều sẽ làm giảm giá trị của TypeScript, vậy nên developer nên cân nhắc về thời gian và giá trị nhận được khi sử dụng TypeScript.

### **3.1.3 Material UI**

#### **3.1.3.1 Khái niệm**

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

#### **3.1.3.2 Vì sao chọn Material UI?**

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

Ngoài ra về số lượng Starred (người yêu thích) trên Github thì Material UI vượt trội hơn cả nghìn sao so với đối thủ cùng lĩnh vực là Ant Design. Với Material UI dẫn đầu trong số các 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 Github<sup>12</sup>.

#### **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

---

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

<sup>12</sup> 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>.

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 nhất<sup>13</sup>.

#### 3.1.3.4 Nhược điểm

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

## 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,

---

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

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.<sup>14</sup>

### 3.2.2 **Ưu điểm**

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

### 3.2.3 **Nhược điểm**

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

---

<sup>14</sup> postgresqltutorial.com. 2024. What is PostgreSQL?. Truy cập từ:

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

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

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

### **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) để chạy trên máy ảo Java Virtual Machine (JVM). Nhờ đó, bất kỳ thiết bị nào có thể cài đặt JVM thì có thể chạy được các ứng dụng viết bằng Java.

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

##### **3.3.2.2 Ưu điểm**

- **Đa nhiệm và linh hoạt:** Java hỗ trợ đa nhiệm và quản lý bộ nhớ tự động, giúp tăng cường hiệu suất và linh hoạt trong quản lý tài nguyên.
- **Độ bảo mật cao:** Java có cơ sở bảo mật mạnh mẽ với kiến trúc sandbox giúp ngăn chặn việc thực thi mã độc hại.

- Chạy trên môi trường ảo Java (JVM): Ứng dụng Java có thể chạy trên mọi nền tảng hỗ trợ JVM mà không cần sửa đổi.
- Khả năng mở rộng và tích hợp dễ dàng: Java có nhiều thư viện và framework mạnh mẽ giúp dễ dàng tích hợp với các công nghệ khác.
- Xử lý đồng thời (Concurrency) Tốt: Java cung cấp các công cụ mạnh mẽ cho xử lý đồng thời, giúp xử lý nhiều yêu cầu đồng thời mà không gặp vấn đề.

### 3.3.2.3 Nhược điểm

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

## 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.
- 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.

---

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

### *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.<sup>17</sup>

## **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.

### *3.3.5.2 Ưu Điểm*

- Ánh xạ đối tượng - dữ liệu: Cho phép ánh xạ giữa các đối tượng Java và cơ sở dữ liệu quan hệ một cách tự động, giảm sự trùng lặp và tăng tính bảo trì.

---

<sup>17</sup> 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/>

- 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.

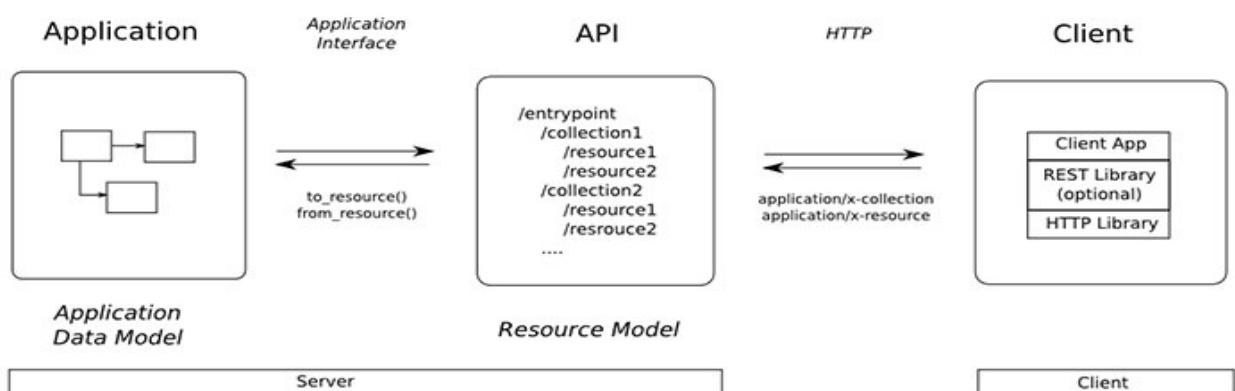
### 3.3.5.3 Nhược Điểm

- 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.<sup>18</sup>



Hình 3.4: Nguyên tắc hoạt động của Restfull API

<sup>18</sup> 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.<sup>19</sup>

## 3.5 Docker

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

### 3.5.1 Lịch sử và Mục đích

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

---

<sup>19</sup> 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>.

máy của tôi" - khi một ứng dụng chạy tốt trên máy phát triển nhưng gặp lỗi khi chuyển sang máy sản xuất.

### **3.5.2        *Ưu điểm của Docker***

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

- User: 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.
- Admin (quản trị viên hệ thống): Những người quản trị hệ thống, quản lý đơn hàng, duyệt đơn hàng.

#### 4.1.2 Nhu cầu admin

- Đăng nhập, đăng xuất.
- Quản lý được đơn hàng: Duyệt đơn hàng

#### 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 đi thuê.
- 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 cho hệ thống, từ thông tin đó hệ thống sẽ cho ra danh sách những tài liệu hiện đang được đăng cho thuê 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. 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.

#### **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 cập nhật 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 bài đăng vào giỏ hàng, xóa bài đăng 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 trễ trả sách. Chức năng trả sách, nhận sách.
- Xem đơn hàng 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 đăng bài: Người dùng cung cấp thông tin về tài liệu về hình ảnh, lời giới thiệu, giá cả, ... lên hệ thống để tạo bài đăng.
- Quản lý bài đăng: Xem được danh sách sách đang đăng của mình. Có thể xoá bài đăng. Có thể xem chi tiết bài đăng: Thông tin sách đã cho thuê như thời hạn, người thuê sách, thông tin sách, giá cả, ...
- Hệ thống thanh toán: Thanh toán tiến hành bằng nhiều hình thức online hoặc COD.

#### **4.2.3 Các chức năng ở Administrator**

- Đăng nhập, đăng xuất admin.
- Quản lý đơn hàng: Xem danh sách đơn hàng, duyệt đơn hàng, thay đổi trạng thái đơn hàng.

## 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: Laptop bao nhiêu inch, mobile bao nhiêu inch, đối với browser nào như thế nào, hệ thống chạy tốt với browser version bao nhiêu trỏ lên.
- 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:
  - Máy tính bảng nhỏ hơn, như Amazon Kindle (600×800) và Barnes & Noble Nook (600×1024), được đặt ở chế độ nằm đứng.
  - Chiều dài từ 576px đến 768px
- Màn hình kích thước siêu nhỏ:
  - Điện thoại Iphone 4 và Iphone 5.

---

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

- Chiều dài từ 320px đến 576px.



Hình 4.1: Biểu đồ phu 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 Browerslist<sup>21</sup> yêu cầu mặc định của website hiện nay bao gồm:
- Phiên bản trình duyệt được chọn có độ phủ sóng lớn hơn 0,5% theo thống kê sử dụng toàn cầu.
- Là 2 phiên bản mới nhất cho mỗi trình duyệt, được thống kê ở bảng dưới.
- Không bao gồm các trình duyệt đã chết: Các trình duyệt không hỗ trợ hoặc cập nhật chính thức trong 24 tháng. Hiện tại là IE 11, IE\_Mob 11, BlackBerry 10, BlackBerry 7, Samsung 4, Opera Mobile 12.1 và tất cả các phiên bản của Baidu.
- Bao gồm Firefox ESR: Là bản phát hành hỗ trợ mở rộng mới nhất của Firefox.

#### 4.3.3 Hệ thống có tính bảo mật

- Website có chứng chỉ SSL, SSL (Secure Sockets Layer) là một tiêu chuẩn an ninh công nghệ toàn cầu. Đảm bảo an toàn dữ liệu và các giao dịch trên website thông qua việc mã hóa các thông tin nhạy cảm trong giao dịch trực tuyến.
- Website sử dụng HTTPS, giao thức của HTTPS được sử dụng để giao tiếp an toàn qua mạng máy tính và được sử dụng rộng rãi trên Internet. Thiết kế giao thức của https là để chống lại mọi sự tấn công nên bảo mật thông tin sẽ tốt hơn.

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

- Hệ thống yêu cầu người dùng và administrator 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 administrator 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      *Hệ thống hỗ trợ đa ngôn ngữ***

Cụ thể hệ thống sẽ sử dụng ngôn ngữ phổ biến nhất hiện nay là tiếng tiếng Việt. Tương lai phát triển cho cả tiếng Anh.

#### **4.3.5      *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.

#### **4.3.6      *Khả năng bảo trì***

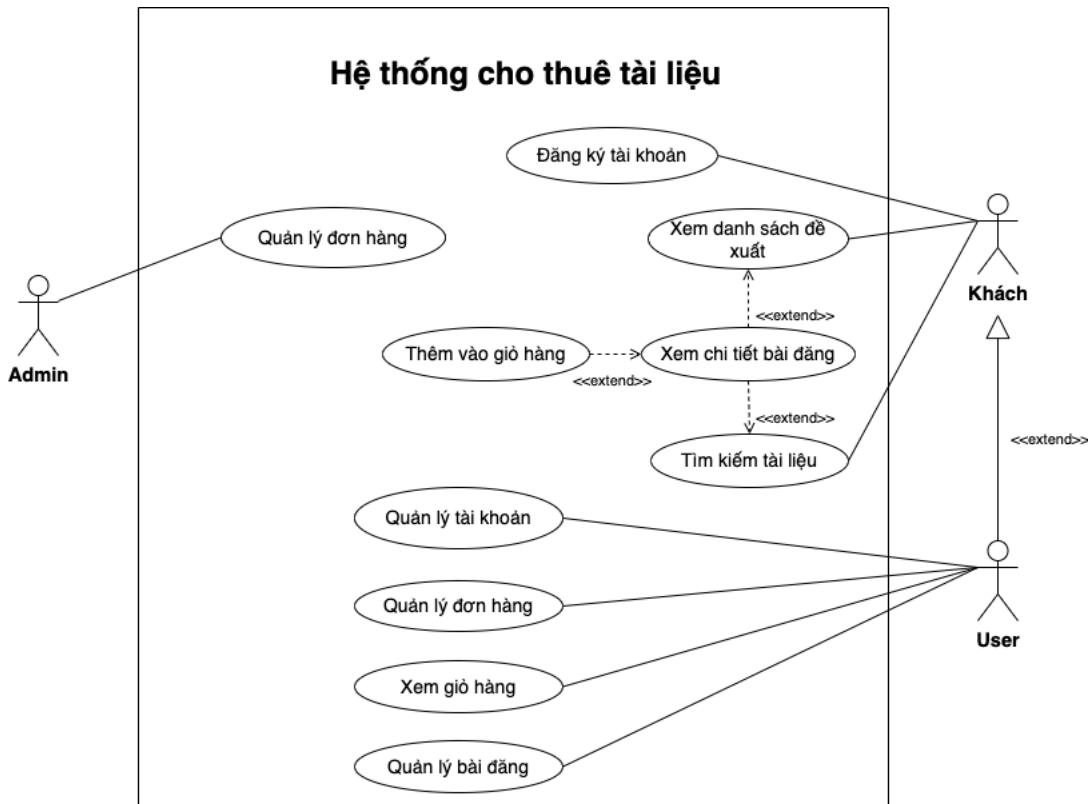
Hệ thống thực hiện bảo trì vào ngày cuối tháng, thời gian bảo trì không vượt quá 8 tiếng.

#### **4.3.7      *Hệ thống có giao diện dễ nhìn, đồng đều, vừa mắt, dễ sử dụng***

- Tuân thủ theo các quy tắc thiết kế cơ bản của Gestalt như sau:
  - Sự tương đồng (similarity)
  - Sự tiếp nối (continuation)
  - Sự gần bên (proximity)
  - Sự đóng kín (closure)
  - Chính-phụ (figure-ground)
  - Sự tối giản (prägnanz)
- Sử dụng phương pháp nghiên cứu phương thức thiết kế UX (UX Research Method):
  - Kiểm tra khả năng sử dụng (Usability Testing): Kiểm tra tính khả dụng để tạo ra một sản phẩm làm cho khách hàng mua ngay từ cái nhìn/ lần sử dụng đầu tiên.
  - Phỏng vấn (Interviews): Nhà nghiên cứu gặp gỡ trực tiếp người dùng để thảo luận sâu hơn về trải nghiệm của người dùng với những chức năng hiện có của website
  - Feedback khách hàng: Người dùng sau khi sử dụng xong một chức năng mới trên website có thể sẽ có biểu mẫu đánh giá xuất hiện để người dùng đánh giá về chức năng mới đó.

- Sử dụng hình ảnh trực quan sinh động với chất lượng cao và dung lượng nhẹ, người dùng chỉ được upload file tối đa 5MB để không bị vượt quá giới hạn lưu trữ của database.
- Dùng google analytics để phân tích Báo cáo thu nạp người dùng.

#### 4.4 Usecase diagram cho toàn hệ thống

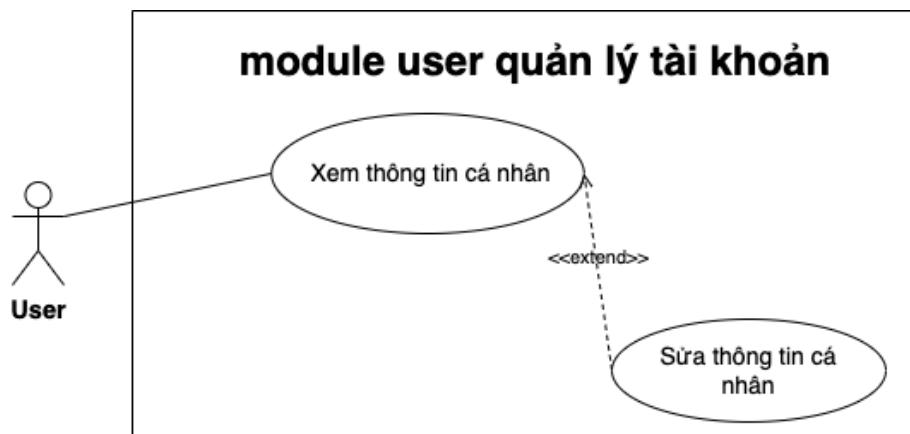


Hình 4.2: Usecase diagram cho toàn bộ hệ thống

#### 4.5 Đặc tả Usecase

##### 4.5.1 Usecases cho User

###### 4.5.1.1 User Quản lý tài khoản



Hình 4.3: Module user quản lý tài khoản

- User xem thông tin cá nhân

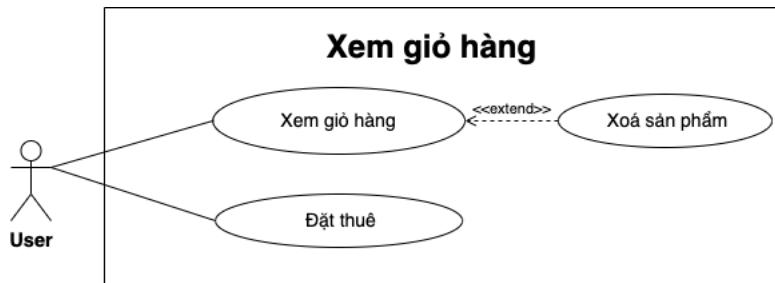
<b>Usecase name</b>	User xem thông tin cá nhân
<b>Actor</b>	User
<b>Description</b>	User muốn xem thông tin cá nhân của mình
<b>Preconditions</b>	User đã đăng nhập thành công User đang ở trang chủ trang web
<b>Postconditions</b>	User xem thông tin cá nhân thành công Hệ thống hiển thị thông tin cá nhân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

- User sửa thông tin cá nhân

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

	thành công của User lên trang thông tin cá nhân
<b>Alternative Flow</b>	Không có
<b>Exception</b>	Không có

#### 4.5.1.2 User Xem giỏ hàng



Hình 4.4:Module xem danh sách giỏ hàng

- **Xem giỏ hàng**

<b>Usecase name</b>	User xem giỏ hàng
<b>Actor</b>	User
<b>Description</b>	User muốn xem sản phẩm đã thêm vào giỏ hàng
<b>Preconditions</b>	User đã đăng nhập thành công User đang ở trang chủ website
<b>Postconditions</b>	User xem sản phẩm trong giỏ hàng thành công Hệ thống hiển thị thông tin sản phẩm trong giỏ hàng
<b>Normal Flow</b>	<ol style="list-style-type: none"> <li>1. User bấm vào icon giỏ hàng trên thanh menu</li> <li>2. Hệ thống hiển thị trang giỏ hàng</li> <li>3. Hệ thống hiển thị sản phẩm đã thêm vào giỏ hàng</li> <li>4. User thấy sản phẩm đã thêm vào giỏ hàng</li> </ol>
<b>Exception</b>	Exception tại bước 3: 3a. Không có sản phẩm trong giỏ hàng của user 4a. Hệ thống hiển thị hình ảnh thay thế khi không có sản phẩm trong giỏ hàng
<b>Alternative Flow</b>	Không có

- **Xóa sản phẩm**

<b>Usecase name</b>	User Xóa sản phẩm
<b>Actor</b>	User
<b>Description</b>	User muốn xóa sản phẩm khỏi giỏ hàng

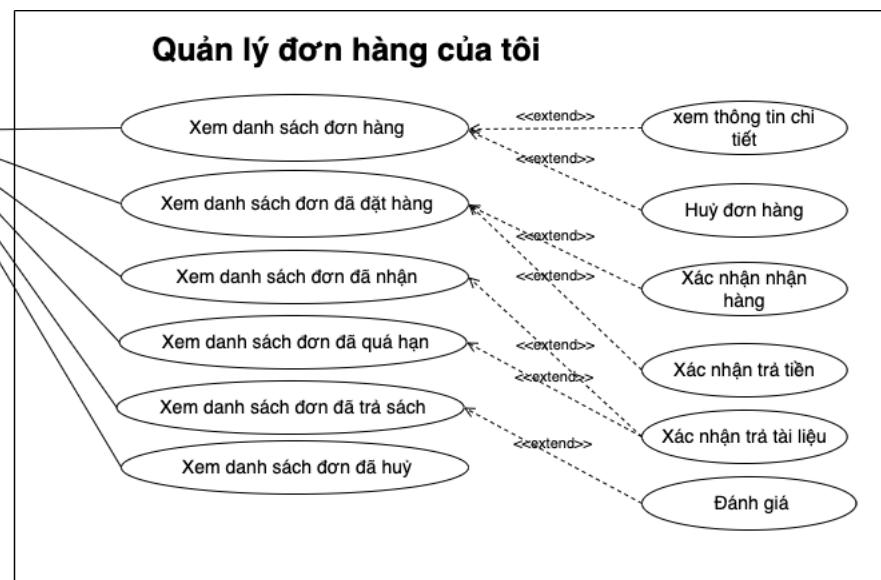
<b>Preconditions</b>	User đã đăng nhập thành công User đang ở trang giỏ hàng Có sản phẩm trong giỏ hàng
<b>Postconditions</b>	User đã xóa thành công sản phẩm khỏi giỏ hàng Sản phẩm bị xoá thành công khỏi giỏ hàng
<b>Normal Flow</b>	1. User chọn nút “thùng rác” để xoá sản phẩm đó khỏi giỏ hàng 2. Hệ thống xoá sản phẩm khỏi giỏ hàng của người dùng
<b>Alternative Flows</b>	Không có
<b>Exception</b>	Không có

- **Đặt thuê**

<b>Usecase name</b>	Đặt thuê
<b>Actor</b>	User
<b>Description</b>	User đặt thuê sản phẩm
<b>Preconditions</b>	User đang ở trang giỏ hàng User đã đăng nhập thành công Có sản phẩm user muốn thuê trong giỏ hàng
<b>Postconditions</b>	User đặt thuê thành công cho đơn hàng Hệ thống cập nhật đơn hàng vào cơ sở dữ liệu
<b>Normal Flow</b>	<ol style="list-style-type: none"> <li>1. Tại trang giỏ hàng &gt;&gt; User bấm nút “Đặt thuê hàng”</li> <li>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</li> <li>3. User nhập thông tin đặt thuê &gt;&gt; bấm xác nhận</li> <li>4. Hệ thống cập nhật thông tin người đặt hàng. Hệ thống hiển thị thông báo nhắc “chọn thanh toán rồi tạo đơn hàng”</li> <li>5. User kiểm tra thông tin đặt hàng &gt;&gt; user chọn phương thức thanh toán vnpay</li> <li>6. Hệ thống loại bỏ hidden của nút tạo đơn hàng</li> <li>7. User bấm “tạo đơn hàng”</li> <li>8. Hệ thống hiển thị trang thanh toán đơn hàng</li> <li>9. User chọn phương thức thanh toán &gt;&gt; tiến hành thanh toán</li> <li>10. Hệ thống xử lý thanh toán thành công</li> <li>11. Hệ thống tiến hành lưu đơn hàng vào hệ thống trạng thái đã thanh toán.</li> <li>12. Hệ thống hiển thị trang check-out với đơn hàng đã được tạo thành công</li> </ol>
<b>Exception</b>	Exception 1 tại bước 9: 9a. Người dùng bấm nút “quay lại”

	10a. Hệ thống hiển thị pop-up “huỷ thanh toán” 11a. User bấm “xác nhận huỷ” 12a. Hệ thống quay lại tạo đơn hàng
<b>Alternative Flows</b>	<p>Alternative 1: tiếp tục với bước 5:</p> <ol style="list-style-type: none"> <li>5. User kiểm tra thông tin đặt hàng &gt;&gt; user chọn phương thức thanh toán cod</li> <li>6. Hệ thống loại bỏ hidden của nút tạo đơn hàng</li> <li>7. User bấm “tạo đơn hàng”</li> <li>8. Hệ thống tiến hành lưu đơn hàng vào hệ thống trạng thái đã đặt hàng.</li> <li>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”</li> </ol> <p>Alternative 2: tiếp tục với bước 5:</p> <ol style="list-style-type: none"> <li>5. User kiểm tra thông tin đặt hàng &gt;&gt; user chọn phương thức thanh toán chuyển khoản</li> <li>6. Hệ thống hiển thị thông tin, QR code để người dùng chuyển khoản. Hệ thống loại bỏ hidden của nút tạo đơn hàng</li> <li>7. User bấm “tạo đơn hàng”</li> <li>8. Hệ thống tiến hành lưu đơn hàng vào hệ thống trạng thái đã đặt hàng.</li> <li>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”</li> </ol>

#### 4.5.1.3 User Quản lý đơn hàng của tôi



Hình 4.5:Module quản lý đơn hàng

- **Xem danh sách đơn hàng**

<b>Usecase name</b>	User xem danh sách đơn hàng
<b>Actor</b>	User
<b>Description</b>	User muốn xem danh sách các đơn hàng
<b>Preconditions</b>	User đã đăng nhập thành công User đang ở trang chủ
<b>Postconditions</b>	User xem danh sách đơn hàng thành công Hệ thống hiển thị thông tin danh sách đơn hàng thành công
<b>Normal Flow</b>	<ol style="list-style-type: none"> <li>1. User bấm vào biểu tượng icon quản lý &gt;&gt; Đơn hàng của tôi</li> <li>2. Hệ thống hiển thị trang quản lý đơn hàng ở tab tất cả</li> <li>3. Hệ thống hiển thị danh sách các đơn thuê của user</li> </ol>
<b>Alternative Flows</b>	Không có
<b>Exception</b>	Không có

- **Xem danh sách đơn hàng đã đặt**

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

- **Xem danh sách đơn hàng đã nhận**

<b>Usecase name</b>	User xem danh sách đơn đã nhận
<b>Actor</b>	User
<b>Description</b>	User muốn xem danh sách các đơn hàng đã nhận
<b>Preconditions</b>	User đã đăng nhập thành công User đang ở trang quản lý đơn hàng của tôi

<b>Postconditions</b>	User xem danh sách đơn hàng đã nhận Hệ thống hiển thị thông tin danh sách đơn hàng đã nhận thành công
<b>Normal Flow</b>	1. User 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 thuê đã nhận của user
<b>Alternative Flows</b>	Không có
<b>Exception</b>	Không có

- **Xem danh sách đơn hàng đã quá hạn**

<b>Usecase name</b>	User xem danh sách đơn đã quá hạn
<b>Actor</b>	User
<b>Description</b>	User muốn xem danh sách các đơn hàng đã quá hạn
<b>Preconditions</b>	User đã đăng nhập thành công User đang ở trang quản lý đơn hàng của tôi
<b>Postconditions</b>	User 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
<b>Normal Flow</b>	1. User 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 thuê đã quá hạn của user
<b>Alternative Flows</b>	Không có
<b>Exception</b>	Không có

- **Xem danh sách đơn hàng đã trả sách**

<b>Usecase name</b>	User xem danh sách đơn đã trả sách
<b>Actor</b>	User
<b>Description</b>	User muốn xem danh sách các đơn hàng đã trả sách
<b>Preconditions</b>	User đã đăng nhập thành công User đang ở trang quản lý đơn hàng của tôi
<b>Postconditions</b>	User 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
<b>Normal Flow</b>	1. User 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 thuê đã trả sách của user
<b>Alternative Flows</b>	Không có
<b>Exception</b>	Không có

- **Xem danh sách đơn hàng đã huỷ**

<b>Usecase name</b>	User xem danh sách đơn đã huỷ
<b>Actor</b>	User
<b>Description</b>	User muốn xem danh sách các đơn hàng đã huỷ
<b>Preconditions</b>	User đã đăng nhập thành công User đang ở trang quản lý đơn hàng của tôi
<b>Postconditions</b>	User xem danh sách đơn hàng đã trả huỷ Hệ thống hiển thị thông tin danh sách đơn hàng đã huỷ thành công
<b>Normal Flow</b>	<ol style="list-style-type: none"> <li>1. User bấm vào tab “ĐÃ HUỶ”</li> <li>2. Hệ thống hiển thị trang quản lý đơn hàng ở tab đã huỷ</li> <li>3. Hệ thống hiển thị danh sách các đơn thuê đã huỷ của user</li> </ol>
<b>Alternative Flows</b>	Không có
<b>Exception</b>	Không có

- **Xác nhận đã trả tiền**

<b>Usecase name</b>	User xác nhận đã trả tiền cọc
<b>Actor</b>	User
<b>Description</b>	User trả tiền cọc cho một đơn hàng chuyển khoản
<b>Preconditions</b>	User đã đăng nhập thành công User đang ở trang quản lý đơn hàng của tôi >> tab đã đặt hàng
<b>Postconditions</b>	User xác nhận đã trả cọc thành công Hệ thống lưu trạng thái xác nhận
<b>Normal Flow</b>	<ol style="list-style-type: none"> <li>1. User bấm vào nút “đã trả tiền” ở đơn hàng chuyển khoản trong danh sách đã đặt</li> <li>2. Hệ thống lưu thông tin xác nhận và ẩn nút “đã trả tiền” ở đơn hàng”</li> </ol>
<b>Alternative Flows</b>	Không có
<b>Exception</b>	Không có

- **Đánh giá đơn hàng**

<b>Usecase name</b>	User đánh giá đơn hàng
<b>Actor</b>	User
<b>Description</b>	User đánh giá đơn hàng đã trả sách
<b>Preconditions</b>	User đã đăng nhập thành công User đang ở trang quản lý đơn hàng của tôi tab đã trả hoặc tab tắt cả
<b>Postconditions</b>	User xem đánh giá đơn hàng thành công Hệ thống lưu thông tin đánh giá đơn hàng thành công
<b>Normal Flow</b>	<ol style="list-style-type: none"> <li>1. User bấm vào nút đánh giá đơn hàng ở đơn hàng đã trả</li> <li>2. Hệ thống hiển thị pop up đánh giá đơn hàng</li> <li>3. User nhập thông tin số điểm đánh giá và mô tả &gt;&gt; Đánh giá</li> <li>4. Hệ thống cập nhật đánh giá đơn hàng thành công. Hệ thống quay lại tab trang quản lý</li> </ol>
<b>Alternative Flows</b>	Không có
<b>Exception</b>	Exception 1: Tiếp tục bước 3: 3a. User bấm nút “huỷ” 4b. Hệ thống quay lại tab trang quản lý

- Xác nhận trả tài liệu

<b>Usecase name</b>	User xác nhận trả tài liệu
<b>Actor</b>	User
<b>Description</b>	User muốn trả tài liệu đang thuê
<b>Preconditions</b>	User đã đăng nhập thành công Đơn hàng đang ở trạng thái đã nhận sách thuê hoặc quá hạn User đang ở trang quản lý đơn hàng của tôi >> tab đã nhận, đã quá hạn, hoặc tắt cả
<b>Postconditions</b>	User xác nhận trả tài liệu cho thuê thành công Hệ thống tiến hành lưu dữ liệu trên cơ sở dữ liệu thành công
<b>Normal Flow</b>	<ol style="list-style-type: none"> <li>1. User bấm nút “trả tài liệu” ở đơn hàng</li> <li>2. Hệ thống tiến hành cập nhật thông tin vào cơ sở dữ liệu &gt;&gt; hiển thị thay đổi lên trang web</li> </ol>
<b>Exception</b>	Không có
<b>Alternative Flows</b>	Không có

- Xác nhận nhận hàng

<b>Usecase name</b>	User xác nhận nhận tài liệu
---------------------	-----------------------------

<b>Actor</b>	User
<b>Description</b>	User xác nhận nhận tài liệu từ chủ thuê
<b>Preconditions</b>	User đã đăng nhập thành công Đơn hàng đang ở trạng thái đã thanh toán User đang ở trang quản lý đơn hàng của tôi >> tab đã đặt hàng hoặc tất cả
<b>Postconditions</b>	User xác nhận nhận tài liệu thành công Hệ thống tiến hành lưu dữ liệu trên cơ sở dữ liệu thành công
<b>Normal Flow</b>	1. User 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
<b>Exception</b>	Không có
<b>Alternative Flows</b>	Không có

- **Huỷ đơn hàng**

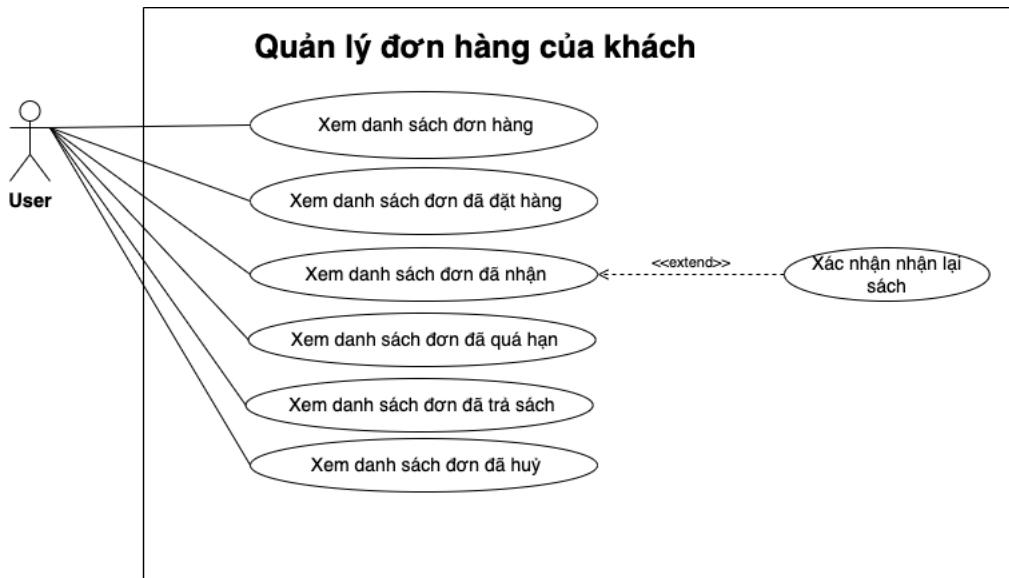
<b>Usecase name</b>	User xác nhận huỷ đơn hàng
<b>Actor</b>	User
<b>Description</b>	User xác nhận huỷ đơn hàng
<b>Preconditions</b>	User đã đăng nhập thành công User đang ở trang quản lý đơn hàng của tôi Đơn huỷ là đơn đang ở trạng thái chờ thanh toán
<b>Postconditions</b>	User huỷ đơn thành công Hệ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

- **Xem thông tin chi tiết đơn hàng**

<b>Usecase name</b>	User xem chi tiết đơn hàng
<b>Actor</b>	User
<b>Description</b>	User xem chi tiết đơn hàng
<b>Preconditions</b>	User đã đăng nhập thành công User đang ở trang quản lý đơn hàng của tôi

<b>Postconditions</b>	User xem chi tiết đơn hàng thành công Hệ thống hiển thị chi tiết đơn hàng thành công
<b>Normal Flow</b>	1. User 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
<b>Exception</b>	Không có
<b>Alternative Flows</b>	Không có

#### 4.5.1.4 User quản lý đơn hàng của khách

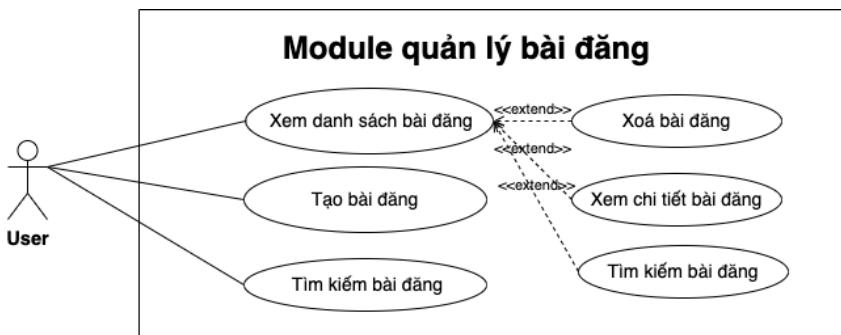


Hình 4.6 Use case quản lý đơn hàng của khách

- Use case Xác nhận nhận lại sách

<b>Use case name</b>	User xác nhận nhận lại sách
<b>Actor</b>	User
<b>Description</b>	User xác nhận nhận lại sách
<b>Preconditions</b>	User đã đăng nhập thành công User đang ở trang quản lý đơn hàng của khách tab đã đặt hàng hoặc tab đã nhận
<b>Postconditions</b>	User xác nhận nhận lại sách thành công Hệ thống tiến hành lưu dữ liệu trên cơ sở dữ liệu thành công
<b>Normal Flow</b>	1. User 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
<b>Exception</b>	Không có
<b>Alternative Flows</b>	Không có

#### 4.5.1.5 User quản lý bài đăng



Hình 4.7:Module quản lý bài đăng

- Xem danh sách bài đăng

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

- Tạo bài đăng

<b>Use case name</b>	Tạo bài đăng
<b>Actor</b>	User
<b>Description</b>	User tạo bài đăng
<b>Preconditions</b>	User đã đăng nhập thành công User đang ở trang quản lý bài đăng
<b>Postconditions</b>	User đăng bài đăng thành công Hệ thống lưu bài đăng vào cơ sở dữ liệu và hiển thị trên trang web
<b>Normal Flow</b>	1. Tại trang bài đăng của website, User bấm vào nút thêm bài đăng 2. Hệ thống hiển thị form đăng bài

	<p>3. Ở mục tìm sách User chọn thêm mới sách &gt;&gt; điền thông tin thêm mới &gt;&gt; bấm nút “tạo sách”</p> <p>4. Hệ thống hiển thị tạo sách thành công</p> <p>5. Ở mục tài liệu user thêm thông tin &gt;&gt; bấm “tạo tài liệu”</p> <p>6. Hệ thống hiển thị tạo tài liệu thành công</p> <p>7. Ở mục bài đăng user thêm thông tin &gt;&gt; bấm “tạo bài đăng”</p> <p>8. Hệ thống hiển thị tạo bài đăng thành công. Hệ thống hiển thị trang chi tiết bài đăng</p>
<b>Exception</b>	Không có
<b>Alternative Flows</b>	<p>Alternative 1:</p> <p>Tiếp tục bước 3</p> <p>3a. Ở mục tìm sách User chọn sách &gt;&gt; bấm nút “chọn sách”</p> <p>4a. Hệ thống hiển thị chọn sách thành công</p> <p>Tiếp tục bước 5</p>

- **Tìm kiếm bài đăng**

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

- **Xoá bài đăng**

<b>Use case name</b>	User xoá bài đăng
<b>Actor</b>	User
<b>Description</b>	User muốn xoá một bài đăng
<b>Preconditions</b>	User đã đăng nhập thành công User đang ở trang quản lý bài đăng
<b>Postconditions</b>	User xoá bài đăng thành công Hệ thống xoá bài đăng khỏi cơ sở dữ liệu

<b>Normal Flow</b>	<ol style="list-style-type: none"> <li>1. User bấm nút thùng rác cuối bài đăng muốn xoá</li> <li>2. Hệ thống hiển thị pop-up bạn có muốn xoá bài đăng không</li> <li>3. User bấm “đồng ý”</li> <li>4. Hệ thống hiển thị xoá bài đăng thành công</li> </ol>
<b>Exception</b>	<p>Exception 1: Tại bước 3:</p> <p>3a. User bấm “Hủy” 4b. Hệ thống quay lại trang quản lý bài đăng</p>
<b>Alternative Flows</b>	Không có

#### 4.5.2 Use Cases chung cho Admin và User

##### 4.5.2.1 Đăng nhập

<b>Use case name</b>	Đăng nhập
<b>Actor</b>	User/Admin
<b>Description</b>	User/Admin cần đăng nhập vào tài khoản hiện có để sử dụng hệ thống
<b>Preconditions</b>	User/Admin đã đăng ký tài khoản trước đó trong hệ thống Với tài khoản Admin, thì cần được hệ thống phân quyền Admin
<b>Postconditions</b>	User/Admin đăng nhập thành công Hệ thống ghi nhận hoạt động đăng nhập của User/Admin vào database
<b>Normal Flow</b>	<ol style="list-style-type: none"> <li>1. Tại trang chủ của website, User/Admin nhấn nút “Đăng nhập/Đăng ký”</li> <li>2. Hệ thống hiển thị form đăng nhập</li> <li>3. User/Admin nhập tài khoản, mật khẩu</li> <li>4. User/Admin bấm nút “đăng nhập”</li> <li>5. Hệ thống xác thực thông tin đăng nhập của User/Admin</li> <li>6. Hệ thống ghi nhận hoạt động đăng nhập thành công</li> <li>7. Hệ thống hiển thị trang chủ sau khi đăng nhập</li> </ol>
<b>Exception</b>	<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>
<b>Alternative Flows</b>	Không có

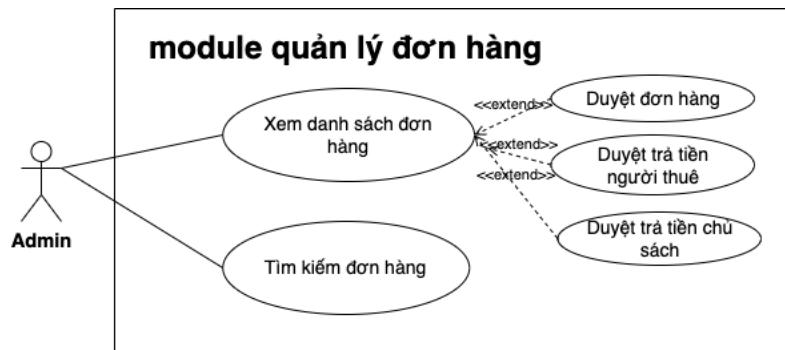
##### 4.5.2.2 Đăng xuất

<b>Use case name</b>	Đăng xuất
----------------------	-----------

<b>Actor</b>	User/Admin
<b>Description</b>	User/Admin cần đăng xuất khỏi tài khoản hiện có
<b>Preconditions</b>	User/Admin thì đã đăng nhập vào tài khoản hiện có
<b>Postconditions</b>	User/Admin đăng xuất thành công Hệ thống ghi nhận hoạt động đăng xuất của User/Admin vào database
<b>Normal Flow</b>	<ol style="list-style-type: none"> <li>1. User/Admin nhấp vào icon “Tài khoản”</li> <li>2. Hệ thống hiển thị danh sách các nút bấm</li> <li>3. User/Admin nhấp nút “Đăng xuất”</li> <li>4. Hệ thống xác thực thông tin đăng xuất của User/Admin</li> <li>5. Hệ thống ghi nhận hoạt động đăng xuất thành công</li> <li>6. Hệ thống chuyển đến giao diện trang chủ</li> </ol>
<b>Alternative Flows</b>	Không có
<b>Exception</b>	Không có

#### 4.5.3 Use Cases cho Admin

##### 4.5.3.1 Admin quản lý đơn hàng



Hình 4.8:Module quản lý đơn hàng-admin

- Xem danh sách đơn hàng

<b>Usecase name</b>	Xem danh sách đơn hàng
<b>Actor</b>	Admin
<b>Description</b>	Admin xem danh sách các đơn hàng
<b>Preconditions</b>	Admin đã đăng nhập bằng tài khoản admin
<b>Postconditions</b>	Admin xem được danh sách đơn hàng Hệ thống hiển thị danh sách đơn hàng
<b>Normal Flow</b>	<ol style="list-style-type: none"> <li>1. Admin truy cập vào trang quản lý đơn hàng</li> <li>2. Hệ thống hiển thị trang quản lý đơn hàng, danh sách đơn hàng</li> </ol>

<b>Exception</b>	Không có
<b>Alternative Flows</b>	Không có

- **Duyệt đơn hàng**

<b>Usecase name</b>	Tìm kiếm đơn hàng
<b>Actor</b>	Admin
<b>Description</b>	Admin tìm kiếm những đơn hàng muốn tìm
<b>Preconditions</b>	Admin đã đăng nhập bằng tài khoản admin
<b>Postconditions</b>	Admin duyệt được đơn hàng Thay đổi đơn hàng được cập nhật lưu trữ trong cơ sở dữ liệu
<b>Normal Flow</b>	<ol style="list-style-type: none"> <li>1. Admin truy cập vào trang quản lý đơn hàng</li> <li>2. Hệ thống hiển thị trang quản lý đơn hàng</li> <li>3. Admin bấm vào nút duyệt đơn</li> <li>4. Hệ thống tiến hành thay đổi trạng thái đơn hàng</li> </ol>
<b>Exception</b>	Không có
<b>Alternative Flows</b>	Không có

- **Duyệt trả tiền người thuê**

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

- **Duyệt trả tiền người bán**

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

#### 4.5.4 Use Cases chung cho User và Khách

##### 4.5.4.1 Đăng ký tài khoản

<b>Use case name</b>	Đăng ký tài khoản
<b>Actor</b>	Khách
<b>Description</b>	Khách vãng lai cần đăng ký một tài khoản mới để đăng nhập hệ thống
<b>Preconditions</b>	Khách vãng lai muốn đăng ký tài khoản trên hệ thống
<b>Postconditions</b>	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
<b>Normal Flow</b>	<ol style="list-style-type: none"> <li>1. Khách nhấp nút “Đăng ký”</li> <li>2. Hệ thống hiển thị trang đăng ký chứa form đăng ký</li> <li>3. Khách điền thông tin cá nhân được yêu cầu</li> <li>4. Khách bấm nút “submit”</li> <li>5. Hệ thống xác thực thông tin tài khoản mới, kiểm tra tài khoản nếu phù hợp để đăng ký và lưu trữ tài khoản mới vào cơ sở dữ liệu</li> <li>6. Hệ thống quay lại đăng nhập</li> </ol>
<b>Exception</b>	Không có
<b>Alternative Flows</b>	Không có

##### 4.5.4.2 Xem danh sách đề xuất

<b>Use case name</b>	Xem danh sách đề xuất
<b>Actor</b>	User/Khách
<b>Description</b>	User/Khách xem danh sách bài đăng
<b>Preconditions</b>	User/Khách ở trang chủ của trang web
<b>Postconditions</b>	User/Khách xem được danh sách đề xuất thành công Hệ thống hiển thị danh sách đề xuất
<b>Normal Flow</b>	1. Tại trang chủ của website, User/Khách thấy danh sách đề xuất
<b>Exception</b>	Không có
<b>Alternative Flows</b>	Không có

#### 4.5.4.3 Chi tiết bài đăng

<b>Use case name</b>	Xem chi tiết bài đăng
<b>Actor</b>	User/Khách
<b>Description</b>	User/Khách xem chi tiết bài đăng
<b>Preconditions</b>	User/Khách đang ở trang chủ hoặc trang danh sách tài liệu, hoặc danh sách tài liệu từ tìm kiếm tài liệu
<b>Postconditions</b>	User/Khách xem được chi tiết thông tin về tài liệu thành công Hệ thống hiển thị thông tin tài liệu
<b>Normal Flow</b>	1. Tại danh sách tài liệu, User/Khách nhấn vào bài đăng muốn xem thông tin chi tiết 2. Hệ thống hiển thị thông tin chi tiết bài đăng
<b>Exception</b>	Không có
<b>Alternative Flows</b>	Không có

#### 4.5.4.4 Tìm kiếm tài liệu

<b>Use case name</b>	Tìm kiếm tài liệu
<b>Actor</b>	User/Khách
<b>Description</b>	User/Khách tìm kiếm tài liệu với từ khóa
<b>Preconditions</b>	User/Khách cần có từ khóa về tài liệu để dễ kiểm User/Khách đang ở trang chủ
<b>Postconditions</b>	User/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
<b>Normal Flow</b>	<ol style="list-style-type: none"> <li>1. User/Khách thấy nút tìm kiếm</li> <li>2. Hệ thống hiển thị ô input để user/khách nhập từ khoá</li> <li>3. User/Khách được nhập từ khóa</li> <li>4. User/Khách nhấn nút tìm kiếm</li> <li>5. Hệ thống hiển thị danh sách tài liệu có chứa từ khóa tìm kiếm của User/Khách</li> </ol>
<b>Exception</b>	<p>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”</p>
<b>Alternative Flows</b>	<p>Alternative1 :</p> <ol style="list-style-type: none"> <li>1. User/khách nhấn icon thẻ loại bên trái input tìm kiếm</li> <li>2. Hệ thống hiển thị danh sách thẻ loại</li> <li>3. User nhán chọn thẻ loại</li> <li>4. Hệ thống hiển thị danh sách tài liệu thuộc thẻ loại tìm kiếm</li> </ol>

#### 4.5.4.5 Thêm vào giỏ hàng

<b>Use case name</b>	Thêm vào giỏ hàng
<b>Actor</b>	User/Khách
<b>Description</b>	User/Khách thêm vào giỏ hàng tài liệu mà mình muốn
<b>Preconditions</b>	User/Khách cần ở trang chi tiết bài đăng của tài liệu mà User/Khách muốn thêm vào
<b>Postconditions</b>	User/Khách thêm tài liệu vào giỏ hàng thành công Hệ thống hiển thị bài đăng có trong giỏ hàng
<b>Normal Flow</b>	<ol style="list-style-type: none"> <li>1. User/Khách đang ở trang chi tiết bài đăng của tài liệu</li> <li>2. User/Khách thêm ngày nhận ngày trả</li> <li>3. Hệ thống active nút “đặt thuê ngay” &gt;&gt; user/khách bấm nút đó</li> <li>4. Hệ thống thêm bài đăng vào giỏ hàng thành công</li> </ol>
<b>Exception</b>	Không có
<b>Alternative Flows</b>	Không có

#### 4.5.4.6 Xem giỏ hàng

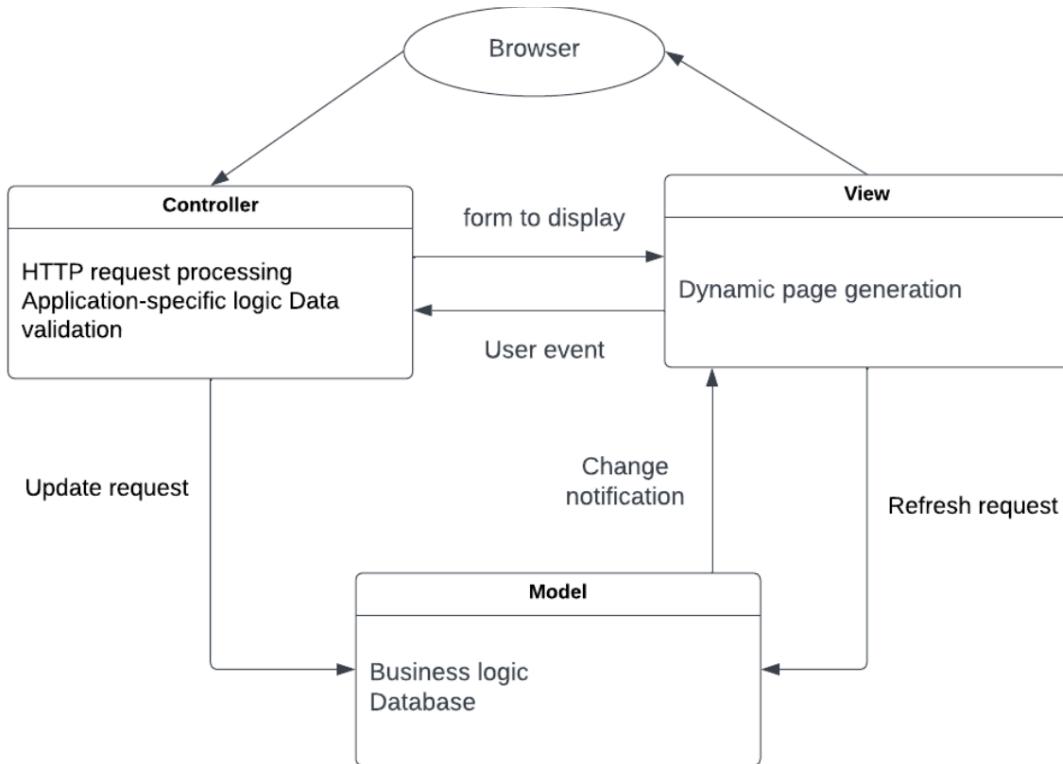
<b>Use case name</b>	Xem giỏ hàng
<b>Actor</b>	User/Khách
<b>Description</b>	User/Khách xem được giỏ hàng

<b>Preconditions</b>	User/Khách ở trang chủ
<b>Postconditions</b>	User/Khách xem giỏ hàng thành công Hệ thống hiển thị bài đăng trong giỏ hàng
<b>Normal Flow</b>	1. User/Khách bấm icon giỏ hàng 2. Hệ thống hiển thị trang giỏ hàng và đơn hàng trong giỏ hàng
<b>Exception</b>	Exception 1: tại bước 2: 2b. Không có bài đăng trong giỏ hàng. Hệ thống hiển thị “giỏ hàng trống”
<b>Alternative Flows</b>	Không có

#### 4.5.4.7 Xoá bài đăng khỏi giỏ hàng

<b>Use case name</b>	Xoá bài đăng khỏi giỏ hàng
<b>Actor</b>	User/Khách
<b>Description</b>	User/Khách xoá bài đăng khỏi giỏ hàng
<b>Preconditions</b>	User/Khách ở trang đơn hàng
<b>Postconditions</b>	User/Khách xoá bài đăng khỏi giỏ hàng thành công Hệ thống xoá bài đăng khỏi giỏ hàng
<b>Normal Flow</b>	1. User/khách bấm icon thùng rác bên cạnh đơn hàng 2. Hệ thống xoá đơn hàng khỏi giỏ hàng thành công
<b>Exception</b>	Không có
<b>Alternative Flows</b>	Không có

#### 4.6 Mô hình kiến trúc hệ thống

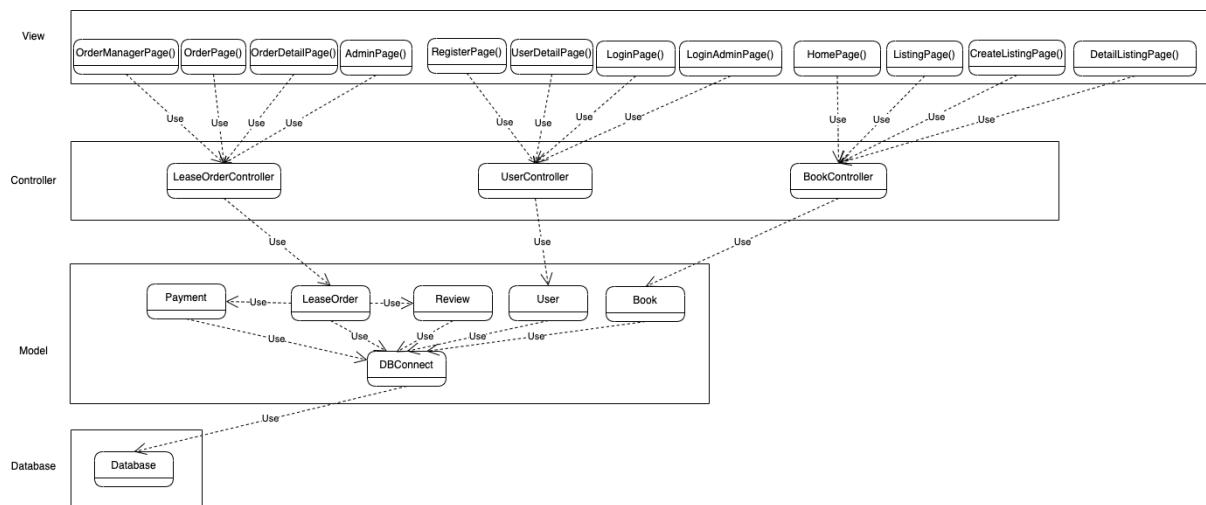


Hình 4.9:Mô hình kiến trúc hệ thống MVC

- 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 user 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 User và khách



Hình 4.10: Layered architecture diagram cho user và khách

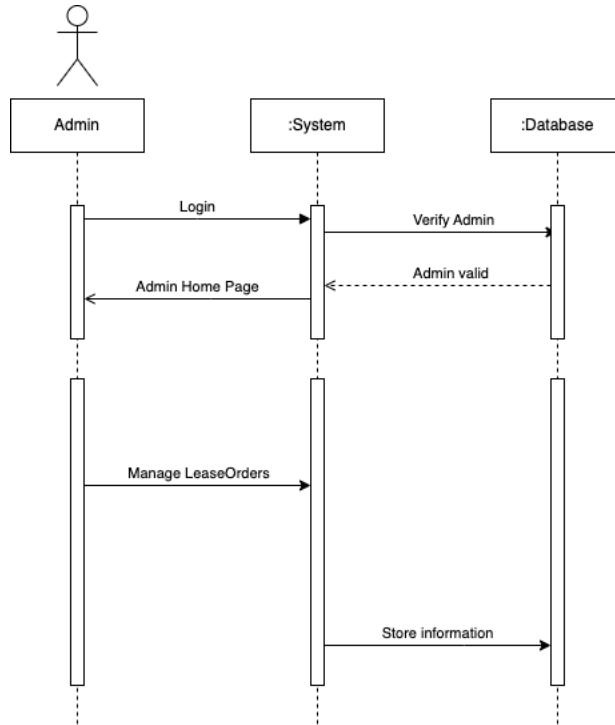
Layered architecture diagram được team em sử dụng theo mô hình MVC.

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
  - LeaseOrderController 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.
  - UserController quản lý các chức năng UI liên qua đến user: đăng nhập, đăng ký, lấy dữ liệu liên quan đến user.
  - BookController quản lý các chức năng liên quan đến listing, book, copy: tạo listing, lấy thông tin listing.
- 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.

## 4.8 Sequence diagram cho các use case

### 4.8.1 Với actor Admin

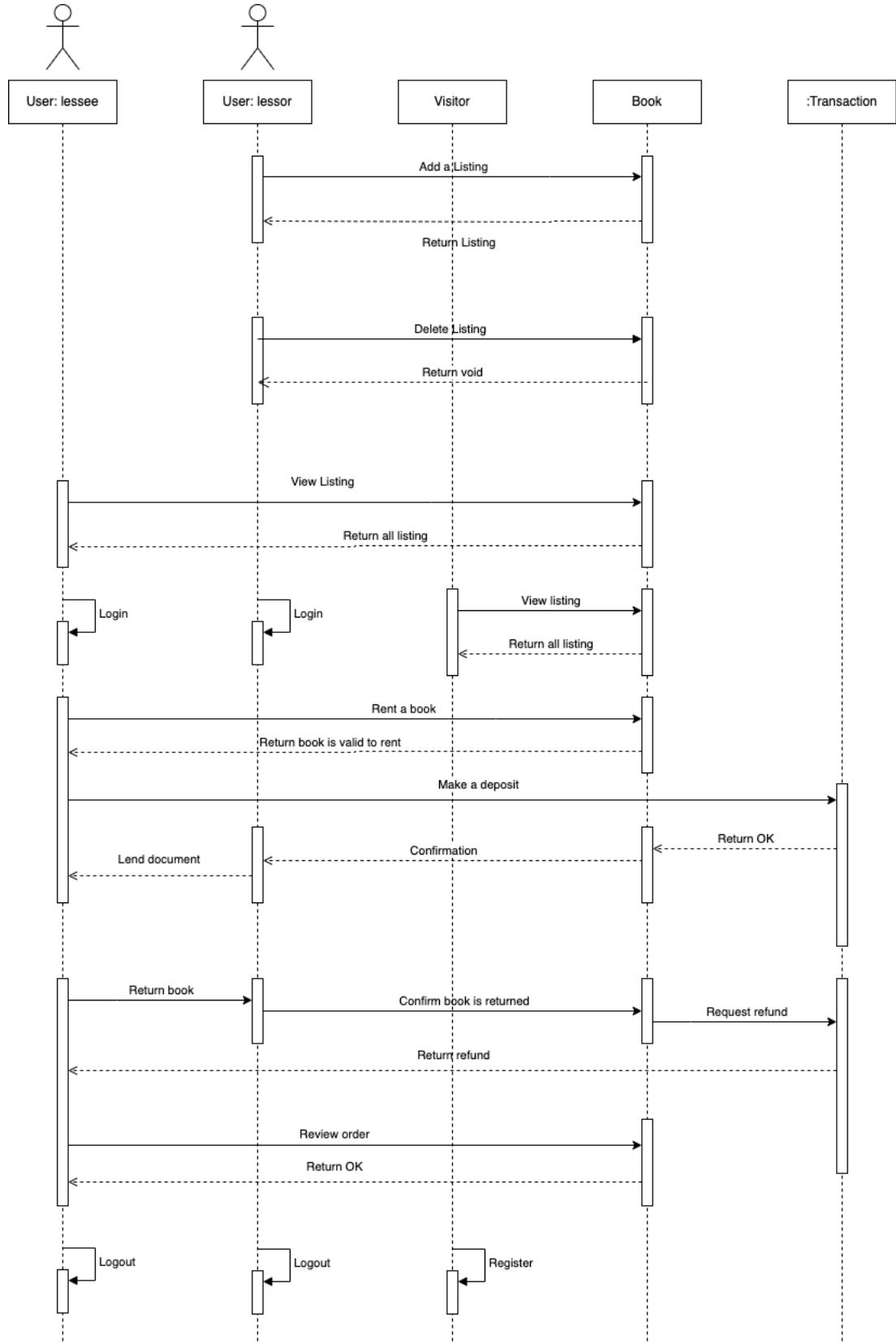


Hình 4.11: Sequence diagram cho admin

Lược đồ trình tự của admin được hiểu như sau:

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

#### 4.8.2 Với actor User và Khách



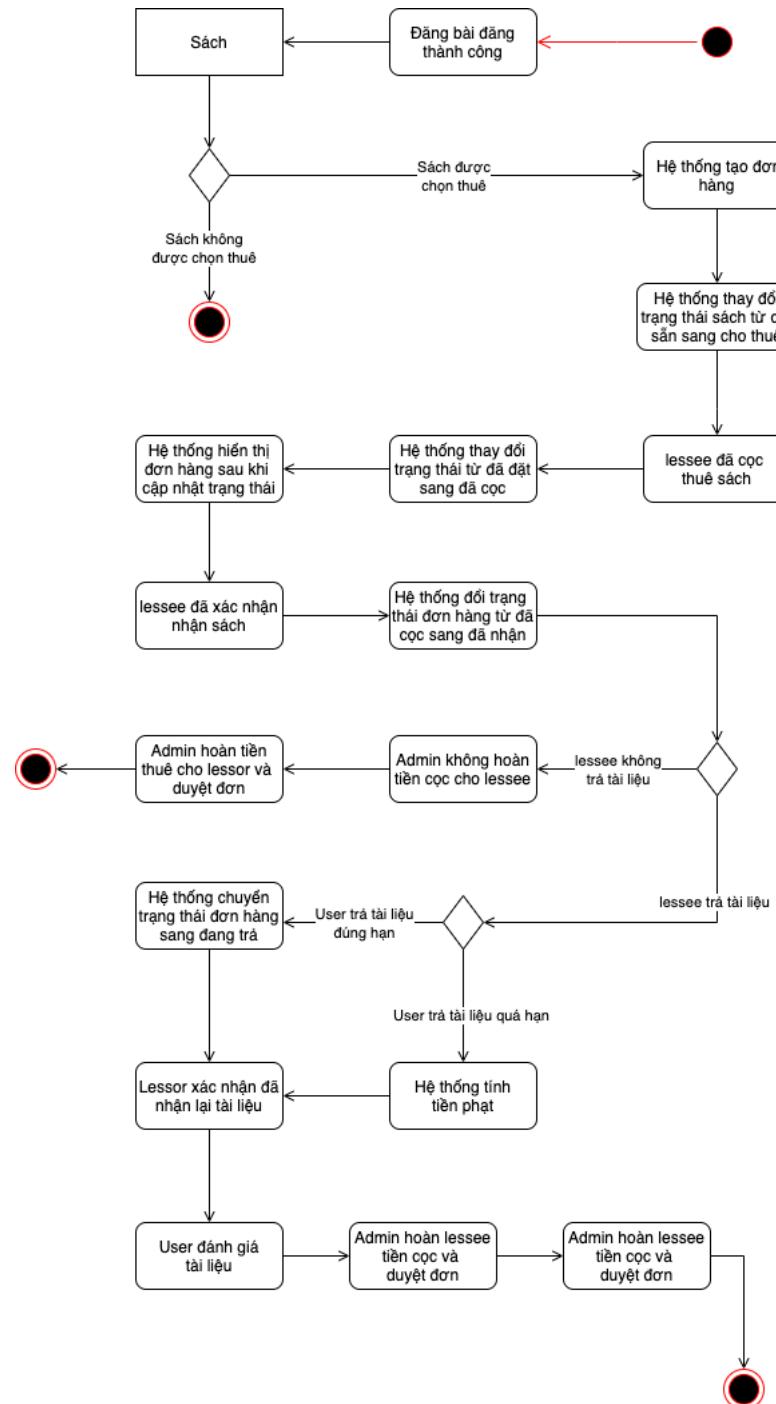
Hình 4.12: Sequence diagram cho user và khách

Lược đồ trình tự cho hoạt động của user và khách 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.

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

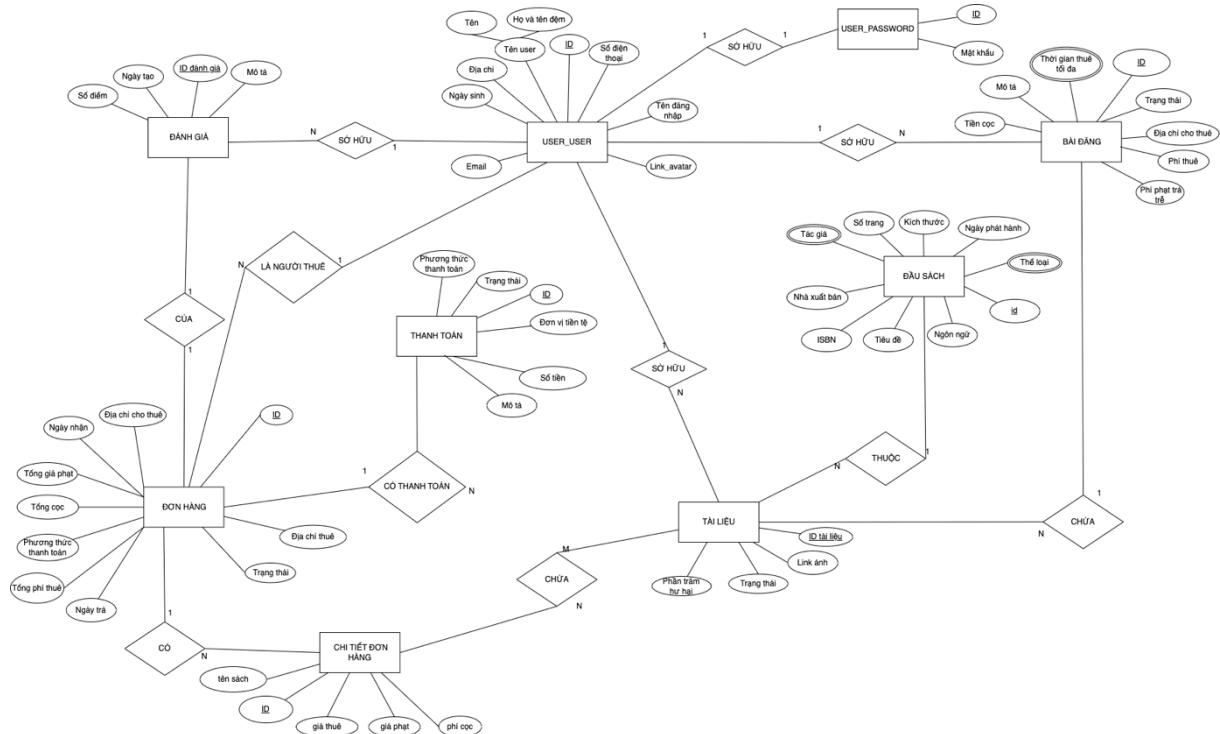
#### 4.9 Activity diagram (Lược đồ hoạt động)



Hình 4.13: Activity diagram thuê và trả tài liệu

Lược đồ hoạt động thuê và trả tài liệu trên hệ thống được hiểu như sau: Sau khi chủ sách tạo bài đăng sách thành công nếu sách không được thuê thì kết thúc. Nếu sách được thuê bởi lessee thì hệ thống tạo đơn hàng và thay đổi trạng thái của sách từ sẵn có thành đã cho thuê. Sau đó lessee tiến hành đóng tiền cọc thuê sách: nếu lessee không đóng cọc thuê sách thì đơn hàng sẽ bị cancel và sách sẽ quay trở về trạng thái có sẵn, nếu lessee có đóng cọc thuê sách thì đơn hàng thay đổi từ đã đặt hàng sang đã cọc sách. Sau đó lessee tiến hành bước nhận sách. Nếu lessee đã nhận sách và xác nhận nhận được sách, hệ thống thay đổi trạng thái đơn hàng từ đã cọc sang đã nhận sách. Tiếp theo là bước trả sách, nếu lessee không trả sách thì admin của hệ thống sẽ không trả cọc lại cho lessee và sau đó tiến hành trả tiền thuê cho lessor. Nếu lessee trả sách có hai trường hợp xảy ra nếu đúng hạn thì tiếp bước đánh giá tài liệu, nếu trễ hạn thì hệ thống tính phí trễ hạn và đến bước đánh giá tài liệu. Sau cùng, admin thanh toán lại tiền cho lessee và lessor. Kết thúc hoạt động.

#### 4.10 Lược đồ quan hệ cơ sở dữ liệu (ERD)



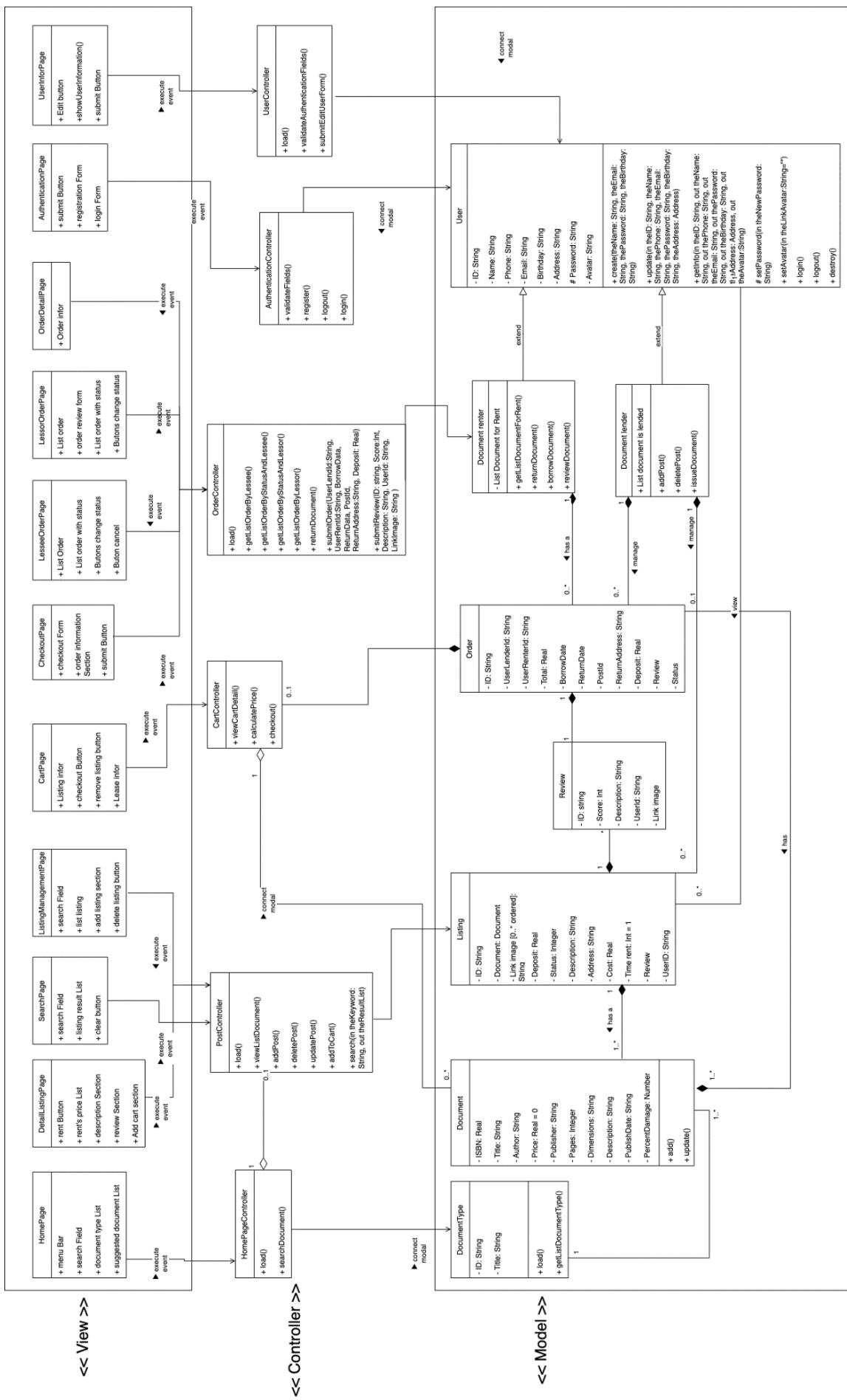
Hình 4.14: ERD cho toàn hệ thống

Lược đồ quan hệ cơ sở dữ liệu gồm các entity: user\_user, user\_password, bài đăng, tài liệu, đầu sách, đánh giá, đơn hàng, chi tiết đơn hàng, thanh toán.

- User\_user chứa thông tin cá nhân của user như họ tên, địa chỉ, ngày sinh, số điện thoại, email, user name.
- User\_password chứa thông tin mật khẩu của user
- Bài đăng chứa thông tin cơ bản bài đăng như id, mô tả, phí cọc, phí thuê, mô tả, trạng thái.

- Tài liệu chứa thông tin cơ bản của một bản tài liệu như phần trăm hư hại, trạng thái, hình ảnh tài liệu.
- Đầu sách chứa thông tin cơ bản đầu sách như thể loại, ngày phát hành, kích thước, số trang, tác giả, nhà xuất bản, isbn, tiêu đề.
- Đánh giá chứa đánh giá của khách thuê về đơn hàng mình thuê.
- Đơn hàng chứa thông tin đơn thuê sách như ngày nhận, ngày trả, địa chỉ cho thuê, tổng cọc, phương thức thanh toán, địa chỉ cho thuê và trạng thái đơn hàng.
- Chi tiết đơn hàng chứa chi tiết copy như tên sách, giá thuê, giá phạt và phí cọc.
- Thanh toán chứa toàn bộ thông tin thanh toán của hệ thống.

#### **4.11 Class diagram**



Hình 4.15: Classdiagram cho hệ thống

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

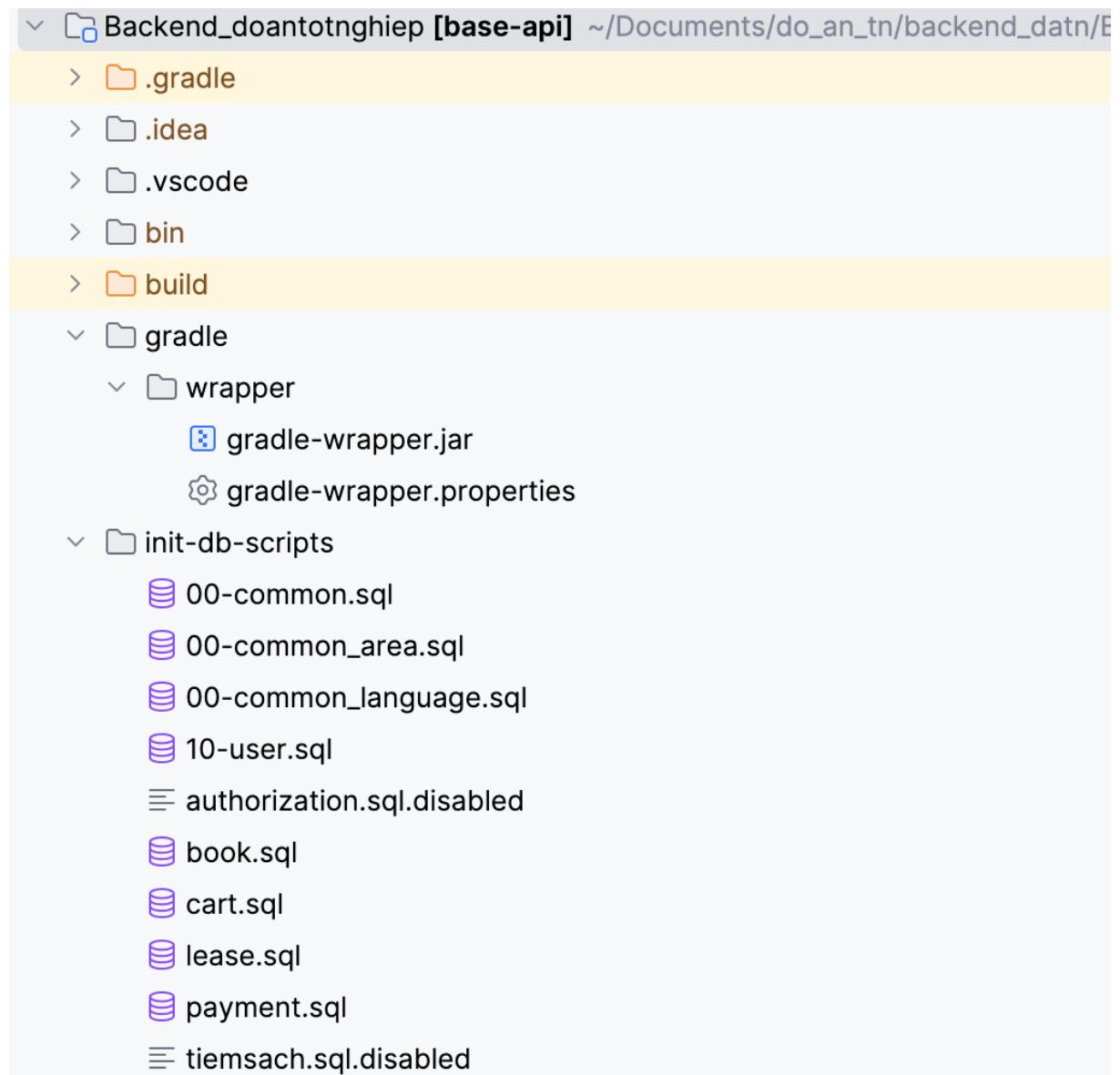
- Lớp view chứa các trang front-end của hệ thống: Trang homepage, thanh navbar, danh sách đề xuất, các thông tin giới thiệu khác. Trang chi tiết bài đăng chứa thông tin chi tiết bài đăng và nút “đặt thuê” để thêm bài đăng vào giỏ hàng. Trang tìm kiếm có chứa sidebar thể loại, thanh tìm kiếm và danh sách tìm kiếm, nút clear để loại bỏ lọc danh sách. Trang quản lý bài đăng có tìm bài đăng, danh sách bài đăng của user, 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 user chứa thông tin user 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, AuthentiationController, UserController
- Lớp model chứa các lớp thực hiện tác vụ logic nghiệp vụ. Bao gồm các model: thể loại, tài liệu, đánh giá đơn hàng, đơn hàng, bài đăng, user.

## CHƯƠNG 5: HIỆN THỰC BACK-END

### 5.1 Hiện thực database

Database được config trong thư mục application-dev.yaml

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



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

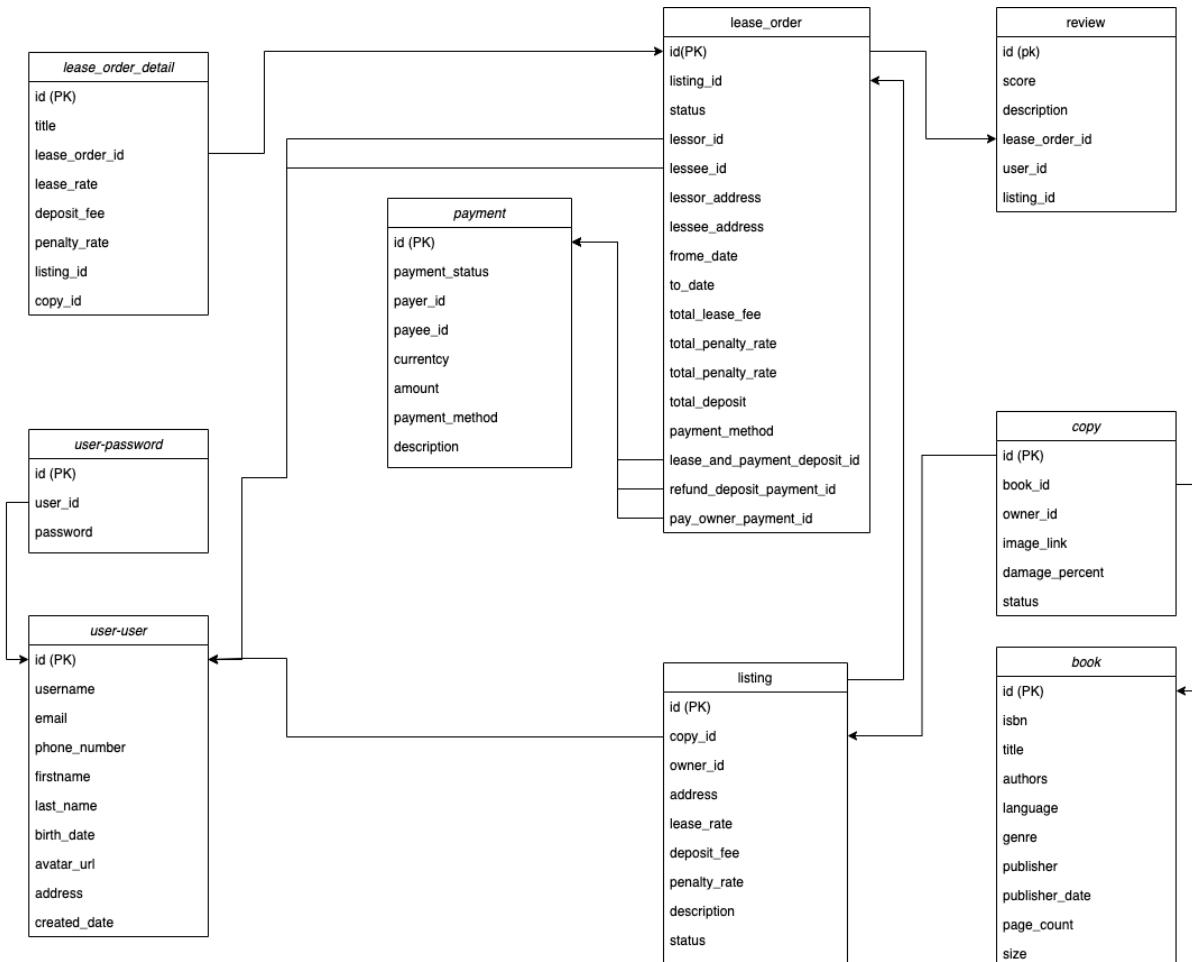
- Thư mục 10-user.sql : Chứa script sql khởi tạo bảng user-user , bảng user-password và các câu lệnh insert data user.
- Thư mục book.sql: Chứa script sql khởi tạo bảng và dữ liệu các bảng liên quan đến sách và bài đăng: bảng book, bảng copy, bảng listing.

- Thư mục lease.sql: Chứa script khởi tạo bảng và dữ liệu ban đầu cho bảng liên quan đến đơn hàng lease-order, lease-order-detail, review.
- Thư mục payment.sql : Chứa script khởi tạo bảng và dữ liệu ban đầu cho bảng payment.

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 Hiện thực api

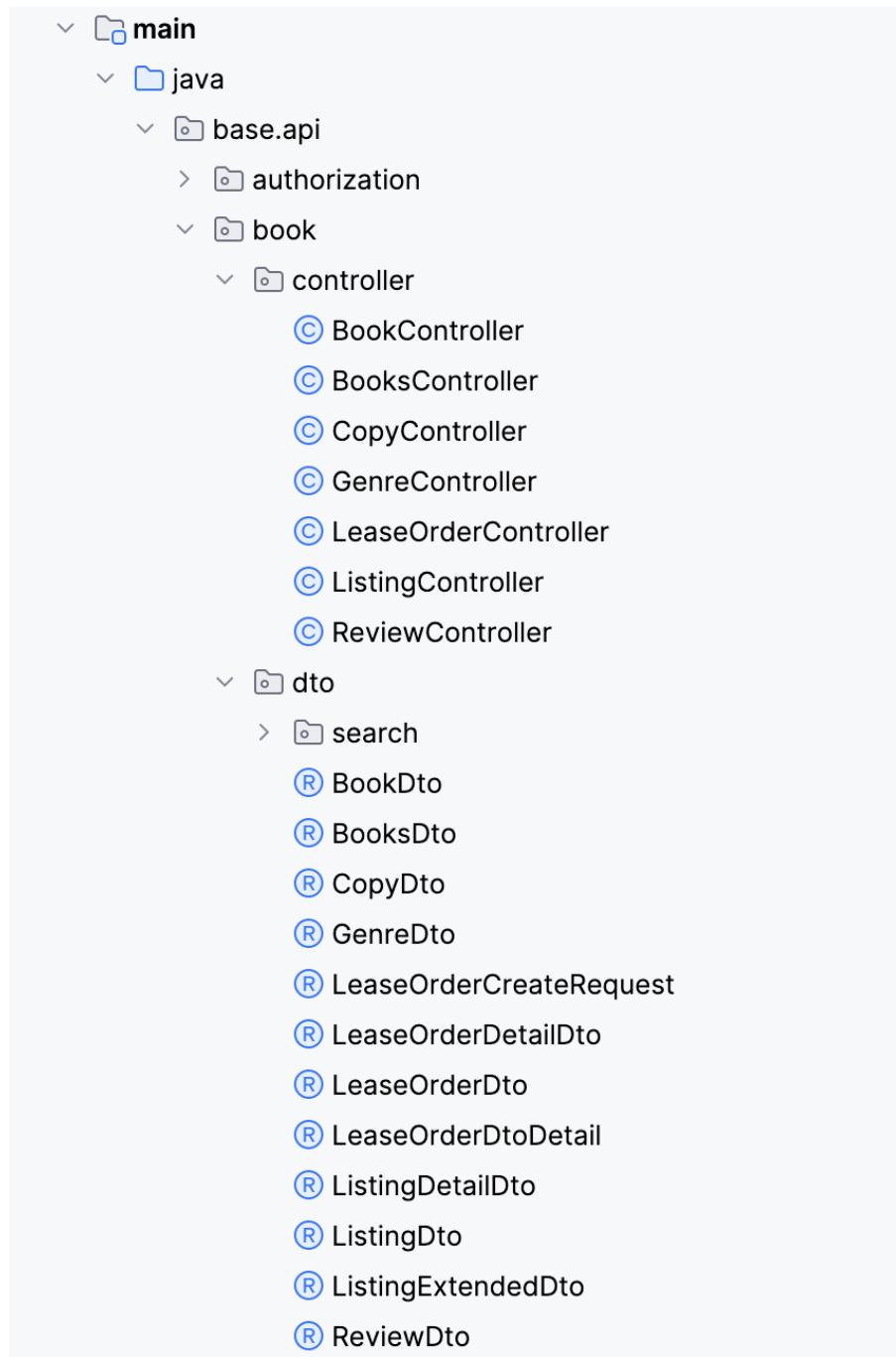
Các api trong dự án được tiến hành xây dựng thông qua các lớp: controller, service, mapper, repository, dto. Mỗi lớp có chức năng, nhiệm vụ riêng.

### 5.2.1 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.



Hình 5.2: Cây thư mục lớp controller và dto của book

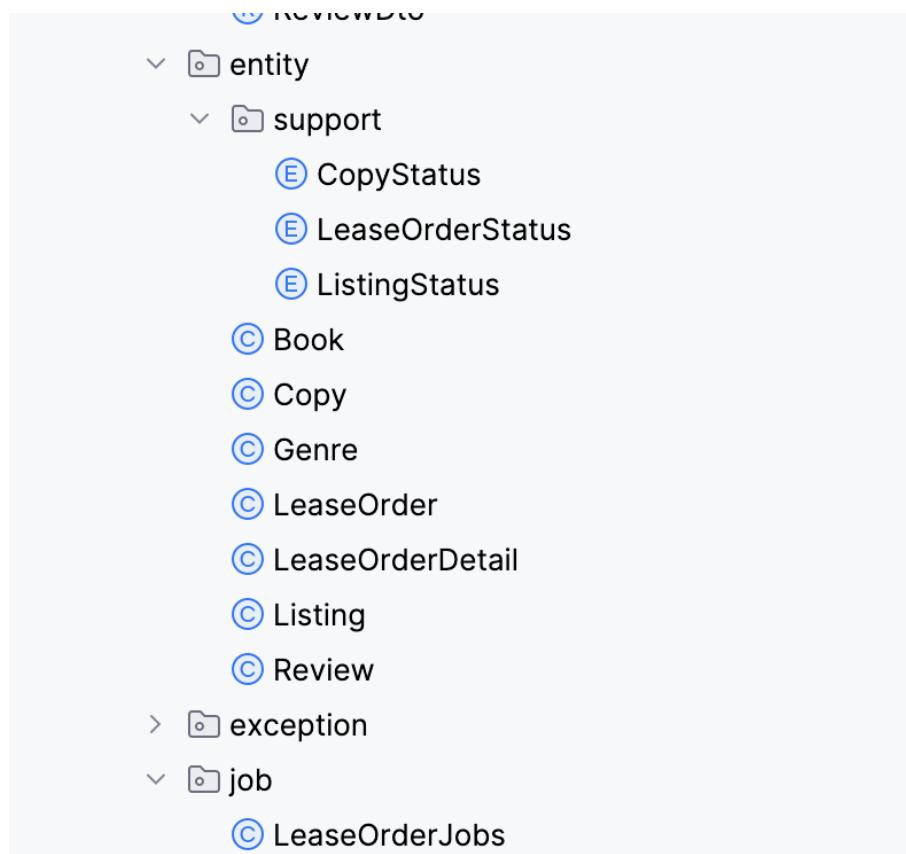
Lớp DTO chứa các mẫu dữ liệu sẵn để truyền dữ liệu giữa các lớp, tầng của api. DTO nhóm em xây dựng ở đây được sử dụng để chuyển dữ liệu từ lớp service lên lớp controller, và ngược lại. DTO được xây dựng ở đây nhằm mục đích tách biệt khỏi thực thể entity, khi cấu trúc thực thể thay đổi thì không ảnh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

### 5.2.2 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 thức 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.



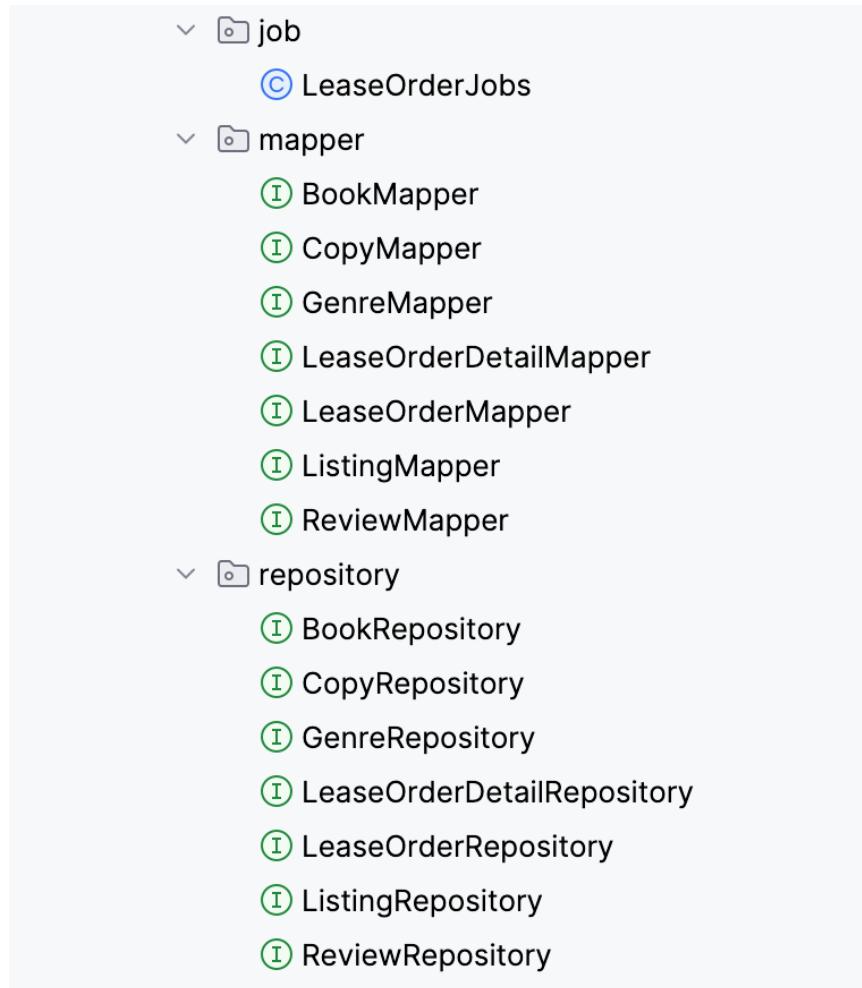
Hình 5.3: Cây thư mục lớp entity của book

### 5.2.3 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.



Hình 5.4: Cây cấu trúc thư mục của job, mapper, repository của book

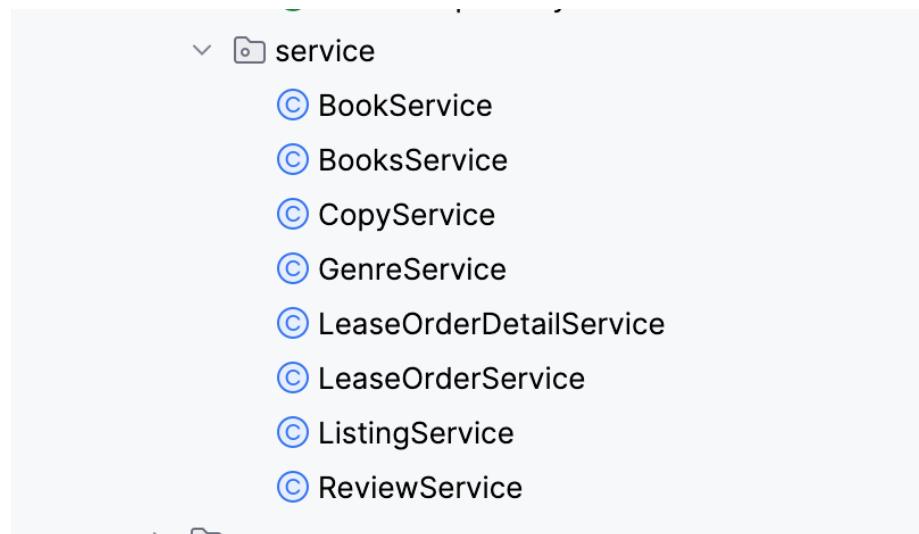
Lớp Job trong dự án này chứa các hàm chạy tự động trong dự án, giúp update cơ sở dữ liệu bởi những nghiệp vụ riêng.

#### 5.2.4 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.



### 5.3 API xây dựng được trong dự án

API dự án sách cũ giá hợp lý			
Nhóm api	tên api	api	loại api
user	Đăng ký	localhost:8082/api/user/register	post
	Đăng nhập	localhost:8082/api/user/login	post
	Lấy thông tin user	localhost:8082/api/user/myInfo	get
	Lấy user theo id	localhost:8082/api/user/{id}	get
	Sửa thông tin user	localhost:8082/api/user/{id}	put
book	Lấy tất cả book	localhost:8082/api/book	get
	Lấy book theo id	localhost:8082/api/book/{id}	get
	Tìm book theo title	localhost:8082/api/book/search/{title}	get
	Tạo book	localhost:8082/api/book	post
listing	Tìm kiếm listing	http://localhost:8082/api/listing/search?size={size}&genre={genre}&page={page}&title={title}	get
	Lấy listing theo id	localhost:8082/api/listing/detailListing/{id}	get
	Tạo listing	localhost:8082/api/listing	post
	Lấy listing theo user	localhost:8082/api/listing/search/byOwnerId/{id}	get
	Xoá listing	localhost:8082/api/listing/delete/{id}	delete

	Lấy chi tiết listing	localhost:8082/api/listing/detailListing/{id}	get
	Tìm listing theo user và title	localhost:8082/api/listing/search/byOwnerIdAndName?title={title}&ownerId={id}	get
copy	Tạo copy	localhost:8082/api/copy	post
review	Tạo review	localhost:8082/api/review	post
Lease order	Tạo đơn hàng	localhost:8082/api/leaseOrder	post
	Lấy danh sách đơn hàng theo người cho thuê	localhost:8082/api/leaseOrder/search/lessor/{lessorId}	get
	Lấy danh sách đơn hàng theo người thuê	localhost:8082/api/leaseOrder/search/lessee/{lesseeId}	get
	Lấy danh sách đơn hàng theo người cho thuê và status	localhost:8082/api/leaseOrder/search/lessor/status/{lessorId}?status={status}	get
	Lấy danh sách đơn hàng theo người thuê và status	localhost:8082/api/leaseOrder/search/lessee/status/{lesseeId}?status={status}	get
	Thay đổi trạng thái đơn hàng	localhost:8082/api/leaseOrder/edit/status?status={status}&id={id}	get
	Lấy đơn hàng theo id	localhost:8082/api/leaseOrder/{id}	get

Bảng 5.1: Api thực hiện trong dự án

## CHƯƠNG 6: HIỆN THỰC FRONT END

### 6.1 Hiện thực front-end

#### 6.1.1 Thư mục app

```
app/
  |- (auth) /
  |  |- login/
  |  |  |- [...login]/
  |  |  |  page.tsx
  |  |  |  Login.css
  |  |  |  Login.scss
  |  |- sign-up/
  |  |  |- [...sign-up]/
  |  |  |  page.tsx
  |- (manager) /
  |  |- customer-order/
  |  |  page.tsx
  |  |- manager-post/
  |  |  |- add-post/
  |  |  |  |- layout.tsx
  |  |  |  page.tsx
  |  |  |- layout.tsx
  |  |  page.tsx
  |- my-order/
  |  |- [...status]/
  |  |  page.tsx
  |  |  layout.tsx
  |- order/
  |  |- [...order]/
  |  |  page.tsx
  |  |  layout.tsx
  |- profile/
  |  page.tsx
  |  layout.tsx
  |- cart/
  |  |- layout.tsx
  |  page.tsx
  |- checkout/
  |  |- [...checkout]/
  |  |  page.tsx
  |  |  layout.tsx
  |- detail/
  |  |- [...detail]/
  |  |  |- layout.tsx
  |  |  page.tsx
  |- search/
  |  |- layout.tsx
  |  page.tsx
  |- error.tsx
  |- globals.css
  |- layout.tsx
  |- loading.tsx
  |- page.tsx
```

Trong thư mục app này sẽ chứa các thành phần đại diện cho các trang (pages) chính của ứng dụng. Gồm : trang đăng nhập, đăng xuất, đăng ký, các trang quản lý: quản lý bài đăng, quản lý đơn hàng khách, quản lý đơn hàng của tôi, trang thanh toán, trang tìm bài đăng, hiệu ứng loading, trang chi tiết bài đăng, trang giỏ hàng, trang quản lý thông tin user.

### 6.1.2 *Thư mục components*

```
components/
├── auth/
│   ├── FormLogin.tsx
│   └── FormSignin.tsx
├── cart/
│   ├── Cart.css
│   ├── Cart.scss
│   ├── CartInfo.tsx
│   ├── CartInfoRent.tsx
│   ├── CartItem.tsx
│   └── CartTotal.tsx
├── checkout/
│   ├── BookItem.tsx
│   ├── FormInputText.tsx
│   ├── HowToRent.tsx
│   ├── InfoRent.tsx
│   ├── Order.tsx
│   ├── OrderInfoRent.tsx
│   ├── OrderTotal.tsx
│   ├── Pay.tsx
│   ├── PaymentStatus.tsx
│   ├── Step.css
│   ├── Step.scss
│   ├── Step1.tsx
│   ├── Step2.tsx
│   ├── Step3.tsx
│   └── parseUrlParams.ts
├── common/
│   └── CommonAlert.tsx
├── createPost/
│   ├── AccordionCustom.tsx
│   ├── CreateBook.tsx
│   ├── CreateDocument.tsx
│   ├── CreatePost.tsx
│   ├── FindBookAutocomplete.tsx
│   ├── GenreAutocomplete.tsx
│   ├── InputListing.tsx
│   ├── ModalSearchBook.tsx
│   └── dummy.ts
├── detail/
│   ├── BookGallery.css
│   ├── BookGallery.scss
│   ├── BookGallery.tsx
│   ├── BookInfo.tsx
│   ├── DocumentDetail.tsx
│   ├── DocumentInfo.tsx
│   └── Owner.tsx
```

```
|   └── Quality.tsx
|   └── RateBook.tsx
|   └── RateComment.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
|       ├── NewComing/
|       |   ├── NewBook.css
|       |   ├── NewBook.scss
|       |   ├── NewBook.tsx
|       |   └── NewComingList.tsx
|       ├── Statistic/
|       |   ├── BookCardCarousel.tsx
|       |   ├── G13ZDV5U.jpg
|       |   ├── Statistic.css
|       |   ├── Statistic.jsx
|       |   └── Statistic.scss
|       ├── HomeBanner.tsx
|       └── PromoteSection.tsx
|   └── managerPost/
|       ├── DeletePostModal.tsx
|       └── column.tsx
|   └── nav/
|       ├── AlertSignOut.tsx
|       ├── BookMenu.tsx
|       ├── Navbar.tsx
|       └── SearchBar.tsx
|   └── order/
|       ├── CancelModal.tsx
|       ├── CustomTabPanel.tsx
|       ├── DetailOrder.tsx
|       ├── HeaderOrder.tsx
|       ├── ListOrder.tsx
|       ├── MenuProfile.tsx
|       ├── NoData.tsx
|       ├── OrderFooter.tsx
|       └── RateModel.tsx
```

```

|   ┌ column.tsx
|   ┌ contentAlert.tsx
|   ┌ scrollButton/
|   ┌   ┌ ScrollButton.tsx
|   ┌ search/
|   ┌   ┌ Filter.tsx
|   ┌   ┌ HeaderListBook.tsx
|   ┌   ┌ ListSearchBook.tsx
|   ┌   ┌ SearchedBook.module.css
|   ┌   ┌ SearchedBook.tsx

```

Là thư mục chứa các component hình thành nên các trang chính của front end. Như những viên gạch xây nên từng phòng của một tòa nhà, mỗi component đại diện cho những viên gạch đó.

Hệ thống gồm có các component:

- Xác thực: form đăng nhập, form đăng ký
- Giỏ hàng: Thông tin đặt hàng, thông tin thuê, thông tin thuê, thông tin sách thuê, thông tin giá cả.
- Thông tin thanh toán: thông tin cuốn sách, thông báo khi thuê sách, các trường phải điền, xác nhận thông tin điền, đơn hàng, ....
- Thông tin chi tiết bài đăng: thông tin cuốn sách, chi tiết cuốn sách, review, form chọn ngày, ...
- Footer của web, các thành phần của homepage, danh sách bài đăng, xoá bài đăng
- Các thành phần thanh navbar, các thành phần trang đặt hàng, nút cuộn lên, các thành phần của thanh tìm kiếm.

### **6.1.3      *Thư mục hooks***

```

hooks/
|   ┌ alert.tsx
|   ┌ cart.tsx
|   ┌ chooseBook.tsx
|   ┌ genre.ts
|   ┌ listNewBook.tsx
|   ┌ order.tsx
|   ┌ search.tsx
|   ┌ step.ts
|   ┌ user.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.
- chooseBook.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.
- user.tsx: hook quản lý thông tin và token của người dùng.

#### **6.1.4      *Thư mục types***

```
types/
├── book.ts
├── cart.ts
├── category.ts
├── checkout.ts
├── common.ts
├── form.ts
├── order.ts
└── page.ts
    └── user.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, user.

#### **6.1.5      *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    Logo trang web**

#### **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

### 6.3 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. Thanh toán Vnpay
7. Xuất đơn hàng
8. Lấy đơn hàng
9. Trang đăng nhập
10. Trang đăng ký
11. Quản lý thông tin user

12. Quản lý đơn hàng của tôi
13. Quản lý đơn hàng của khách
14. Quản lý bài đăng
15. Chi tiết đơn hàng
16. Tạo bài đăng
17. Trang đăng ký admin
18. Trang đăng nhập admin
19. Trang quản lý admin

### 6.3.1 Trang chủ

The Flying Bookstore

BÁN CHẠY

## Danh nhân Việt Nam

Đoàn Kế Thiện #Văn học Việt Nam

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

**Thuê ngay** Xem chi tiết →

**5.000đ/ngày** 71.000đ Giảm 20%

Giao hàng nhanh

Bảo mật thanh toán

Chất lượng đỉnh nhất

Chính sách rõ ràng

Sách mới ra mắt

Từ 24/2 đến 19/05

Houses of Stone 14.000đ/ngày <a href="#">Thêm vào giỏ hàng</a>	To Die for 8.000đ/ngày <a href="#">Thêm vào giỏ hàng</a>	Vanishing Point 2.300đ/ngày <a href="#">Thêm vào giỏ hàng</a>	Hide and Seek 2.200đ/ngày <a href="#">Thêm vào giỏ hàng</a>	The Many Lives & Secret Sorrows of Josephine B. 1.800đ/ngày <a href="#">Thêm vào giỏ hàng</a>

Danh mục sản phẩm

- Kịch
- Kinh tế học
- Giáo dục
- Môi trường
- Sử thi

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

Nhập mail tại đây

Liên hệ

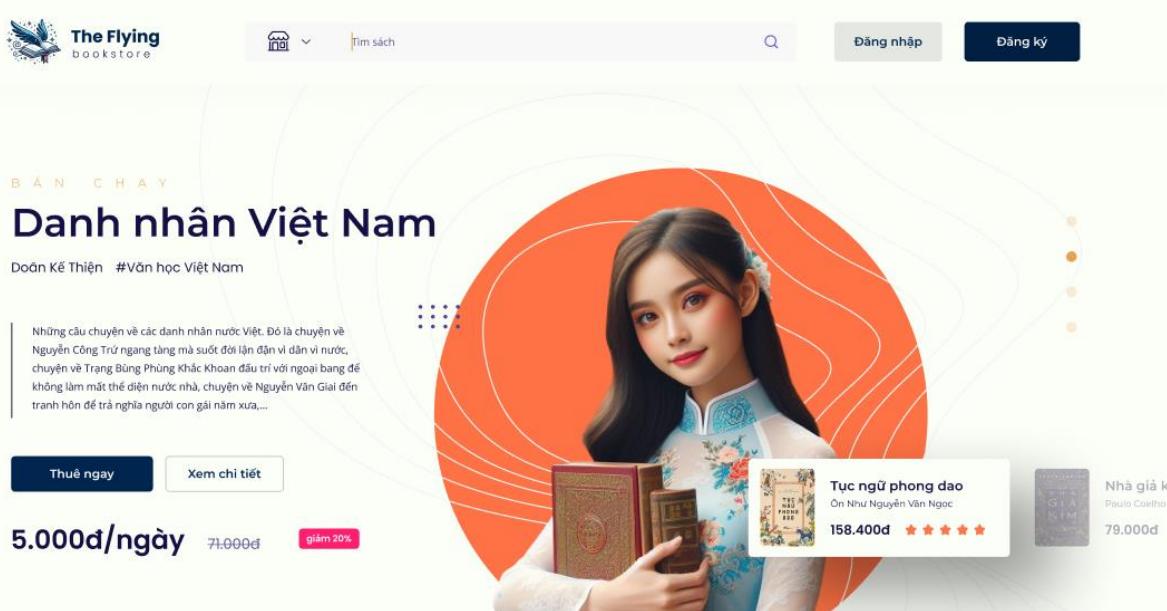
171/11 Trường Phước Phan, Bình Hưng Hòa, Bình Tân, TP.HCM  
+123 345 123  
support@flying bookstore

The Flying Bookstore là nơi cung cấp cho thuê, mua bán tài liệu của bạn. Đồng thời giúp quản lý các tài liệu minh chứng thuê, mua.

Hình 6.3:Trang chủ

Trang chủ đơn hàng chứa thanh navbar, banner, thông tin bài đăng đề xuất và footer trang web chứa thông tin liên hệ của ban quản trị hệ thống. Từ đây thì ta có thể vào trang chi tiết bài đăng khi nhấp vào mỗi bài đăng. Có thể tìm kiếm bài đăng dựa vào các thanh tìm kiếm và icon tìm kiếm theo thể loại. Đồng thời có thể truy cập vào các trang quản lý bằng những icon bên phải màn hình.

#### 6.3.1.1 Banner



Hình 6.4: Banner ở trang chủ

Ở banner ngay đầu trang có lấy một cuốn sách là Danh nhân Việt Nam, với hình ảnh người mẫu ở đây được tạo bằng Creator AI.

#### 6.3.1.2 Section sách mới ra mắt



Hình 6.5: Section sách mới ra mắt ở trang chủ

Danh sách đề xuất có tiêm thêm hiệu ứng bangling chuyên làm cho trang web trở nên bắt mắt và sinh động hơn.

### 6.3.2 Trang tìm kiếm và hiển thị danh sách bài đăng

The Flying bookstore

Tìm sách...

XÓA BỘ LỌC

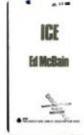
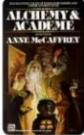
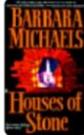
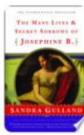
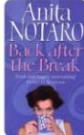
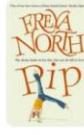
**Lọc theo**

**Tất cả sách**

Hiển thị 995 cuốn sách hiện có

Danh mục

- Học thuật
- Hồi ký
- Kinh điển
- Hải kịch
- truyện tranh
- Văn hóa
- 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 ng่าน
- Xã hội
- Xã hội học
- Các môn thể thao
- Hồi hộp
- Sách giáo khoa
- Giật gân

			
Ice 1.600đ/ngày	Terminal 1.400đ/ngày	Alchemy & Academe 2.000đ/ngày	Houses of Stone 1.400đ/ngày
			
To Die for 800đ/ngày	Vanishing Point 2.300đ/ngày	Hide and Seek 2.200đ/ngày	The Many Lives & Secret S... 1.800đ/ngày
			
Springwater Wedding 1.700đ/ngày	Tales of Mithgar 1.000đ/ngày	Let the Drum Speak 2.300đ/ngày	Back After the Break 600đ/ngày
			
Pip 2.400đ/ngày	The Last Witness 2.900đ/ngày	Garnethill 2.800đ/ngày	Into the Darkness 1.300đ/ngày
			
Pride and Prejudice 2.300đ/ngày	Emma 2.000đ/ngày	Sense and Sensibility 2.400đ/ngày	The House of Scrofa 2.000đ/ngày

1 2 3 4 5 ... 50 >

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

Nhập mail tại đây



#### Liên hệ

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

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



Hình 6.6: Trang danh sách bài đăng và tìm kiếm tài liệu

Trang tìm kiếm tài liệu truy cập bằng cách tìm kiếm tài liệu. Tại đây người dùng có thể lựa chọn tìm kiếm bằng cách đánh từ khoá vào ô input sau đó bấm tìm kiếm hoặc tìm kiếm theo thể loại sách hoặc cả hai. Ngoài ra nút “xoá bộ lọc” giúp xoá bộ lọc tìm kiếm trước đó.

### 6.3.3 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, title, author, publisher, and publication date. It also displays the price (2.200đ/ngày), availability (Hiện vẫn còn hàng), and a 'Đặt thuê' (Rent) section. The 'Đặt thuê' section allows users to enter a pickup date (19/05/2024) and a return date (Ngày trả), set a rental price (Giá thuê: 2.200đ/ngày), and calculate the total cost (Tổng tiền: 0). A note indicates that the user has not chosen a return date yet. Below this is a 'Chủ sách' (Book owner) section showing statistics: 989 books available and 1 book currently rented out. At the bottom, there's a 'Sách mới ra mắt' (New books) section with five book covers and their details.

**Thông tin chung về sách**

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

**Thông tin về sách thuê**

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

**Sách mới ra mắt**

Pride and Prejudice	Emma	Sense and Sensibility	The House of Scorta	Ice
2.300đ/ngày	200đ/ngày	2.400đ/ngày	2.000đ/ngày	1600đ/ngày
<a href="#">Thêm vào giỏ hàng</a>	<a href="#">Thêm vào giỏ hàng</a>	<a href="#">Thêm vào giỏ hàng</a>	<a href="#">Thêm vào giỏ hàng</a>	<a href="#">Thêm vào giỏ hàng</a>

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

Hình 6.7: Trang chi tiết bài đăng

Trang chi tiết bài đăng gồm các phần mô tả cuốn sách với hình ảnh, mô tả, thông tin tác giả nhà xuất bản và giá thuê, chi tiết chung cuốn sách về những chi tiết cụ thể hơn như ngày xuất bản size, tiền cọc phần trăm hư hại, giá cọc,... . Phần đánh giá hiển thị các đánh giá của người dùng, và phần đặt thuê trong đó có chọn ngày mượn ngày trả, hệ thống sẽ tính tiền tạm cho bạn xem, đồng thời nút “đặt thuê ngay” sẽ không còn bị mờ để bạn có thể thêm bài đăng vào đơn hàng. Chi tiết từng phần được thể hiện như hình dưới đây:

**Hide and Seek**

Tác giả: Fern Michaels

Nhà xuất bản: Oxford University Press

Ngày xuất bản: 01-01-2008

Giá thuê: 2.200đ/ngày

Hiện vẫn còn hàng

Hình 6.8: Mô tả cuốn sách

Chi tiết	Đánh giá
Ngày xuất bản	01/05/2022
Kích thước	14.5 x 20.5 cm
ISBN	121341381648 (ISBN13: 121341381648)
Số trang	408 trang
Tiền cọc	78.000đ
Số lượng sách hiện có	3 cuốn
Mô tả về sách thuê	Sách mới mua tháng 6/2023
Địa chỉ cho thuê	123 Nguyễn Trọng Tuyển, phường 12, quận Bình Thạnh
Thể loại	Sách tôn giáo & triết học

Hình 6.9: Tab chi tiết cuốn sách

## Đặt thuê

Thuê theo ngày 300đ/ngày ^

Ngày nhận  
18/05/2024

Ngày trả

Giá thuê 300đ/ngày

Số ngày thuê 0

Tổng tiền thuê 0

Tiền cọc 224.100đ

**Tổng tiền** 0

! Bạn cần chọn ngày trả

Hình 6.10: Section đặt thuê

CHI TIẾT ĐÁNH GIÁ

5 ★★★★★

2 bài đánh giá

Xem tất cả 5 sao 4 sao 3 sao 2 sao 1 sao

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

 **Ấn danh**  
Sách rất tốt  
★★★★★ 5 sao | 19/05/2024

Hình 6.11: Tab đánh giá

### 6.3.4 Trang giỏ hàng

The screenshot shows the shopping cart page of The Flying Bookstore. At the top, there's a header with the store logo, a search bar, and a user account icon. Below the header, the title "Giỏ hàng của bạn" (Your Cart) is displayed. The main content area shows a single item in the cart: "ACADEME ANNE McCAFFREY" by Anne McCaffrey, with the subtitle "Alchemy and Academe". It includes details like quantity (1), rental period (19/05/2024 - 23/05/2024), and total price (120.300đ). Below the cart summary, there's a section titled "Thông tin đặt hàng" (Order Information) with fields for the borrower's name (Braun Tracy), rental duration (4 days), and delivery address (Đường Kim Giang, Phường Kim Giang, Quận Thanh Xuân, Hà Nội). The total amount is listed as 120.300đ. At the bottom of this section is a yellow "ĐẶT THUÊ HÀNG" (Place Order) button. A dark banner at the bottom encourages users to enter their email for special gifts.

Hình 6.12: Trang giỏ hàng

Trang giỏ hàng chứa bài đăng và thông tin đặt hàng. Ngoài ra có nút xoá để xoá bài đăng khỏi giỏ hàng. Và nút đặt thuê để đặt thuê hàng.

### 6.3.5 Trang đặt hàng

The screenshot shows the booking information section of the website. It includes fields for customer details (Name, Email, Phone, Birthdate), payment information (Number of days, Payment method), and delivery address. There are also sections for book details and payment methods.

Thông tin đặt thuê		Thông tin thanh toán	
Họ và tên lót *	Nguyễn Lê Xuân	Số ngày thuê	4
Tên *	Hòa	Tiền thuê	8.000đ
Email *	hoa.nguyenlexuan@gmail.com	Tiền cọc	120.300đ
Ngày sinh *	12/12/2000	Tổng cộng	120.300đ
Số điện thoại *	0905907362		
Địa chỉ *	123 Nơ Trang Long, phường 12, quận Bình Thạnh, thành phố Hồ Chí Minh	XÁC NHẬN	
			Bạn sẽ đưa 120.300đ cho công ty Flying Bookstore tại địa chỉ 171/11 Trường Phuoc Phan, quận Bình Tân, tp.HCM
<b>Thông tin đặt hàng</b>		<b>Thanh toán</b>	
Người cho thuê	Braun Tracy	Dựa tiền trực tiếp	
Thời gian thuê	19/05/2024 - 23/05/2024	Chuyển khoản	
Địa chỉ	Đường Kim Giang, Phường Kim Giang, Quận Thanh Xuân, Hà Nội	Thanh toán VNPay	
Số điện thoại	0905 907 362		
<b>QUAY LẠI GIỎ HÀNG</b>		<b>TAO ĐƠN HÀNG</b>	
<b>Quy trình thuê sách</b> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p><b>Đến nhận sách</b></p> <p>Sau khi thanh toán thành công, bạn có thể chọn 1 trong 2 cách sau: - Ban 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.</p> </div> <div style="text-align: center;"> <p><b>Đọc sách</b></p> <p>Gìn giữ sách cẩn thận bạn nhé. Hãy nhớ sau bạn còn có người khác đọc nữa, nên hãy tránh việc vứt hay làm rách sách bạn nhé.</p> </div> <div style="text-align: center;"> <p><b>Trả sách</b></p> <p>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 sẽ, 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.</p> </div> </div>			
<p>Nhập mail để nhận được những quà tặng đặc biệt nhé </p> <input type="text" value="Nhập mail tại đây"/>			
<div style="display: flex; justify-content: space-between;"> <div style="width: 30%;"> <p><b>The Flying bookstore</b></p> <p>The Flying Bookstore là nơi có thể giúp bạn học sinh/sinh viên đếm cho thuê, mượn tài liệu của mình. Đồng thời giúp quản lý các tài liệu mình cho thuê, mượn.</p> <p><a href="#">Facebook</a> <a href="#">Twitter</a> <a href="#">Instagram</a></p> </div> <div style="width: 40%;"> <p><b>Liên hệ</b></p> <p>171/11 Trường Phuoc Phan, Bình Hưng Hòa, Bình Tân, TP.HCM +123 345 123 <a href="mailto:support@flying.bookstore">support@flying.bookstore</a></p> <p></p> </div> </div>			

Hình 6.13: Trang đặt hàng

Trang đặt hàng chứa thông tin người đặt, thông tin chủ sách, thông tin thanh toán và chọn phương thức thanh toán. Có 3 phương thức thanh toán chính COD, vnpay, chuyển khoản.

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

## Thanh toán



### Bước 1: Thanh toán

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



Cách 2: Bạn chuyển khoản theo thông tin sau:

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

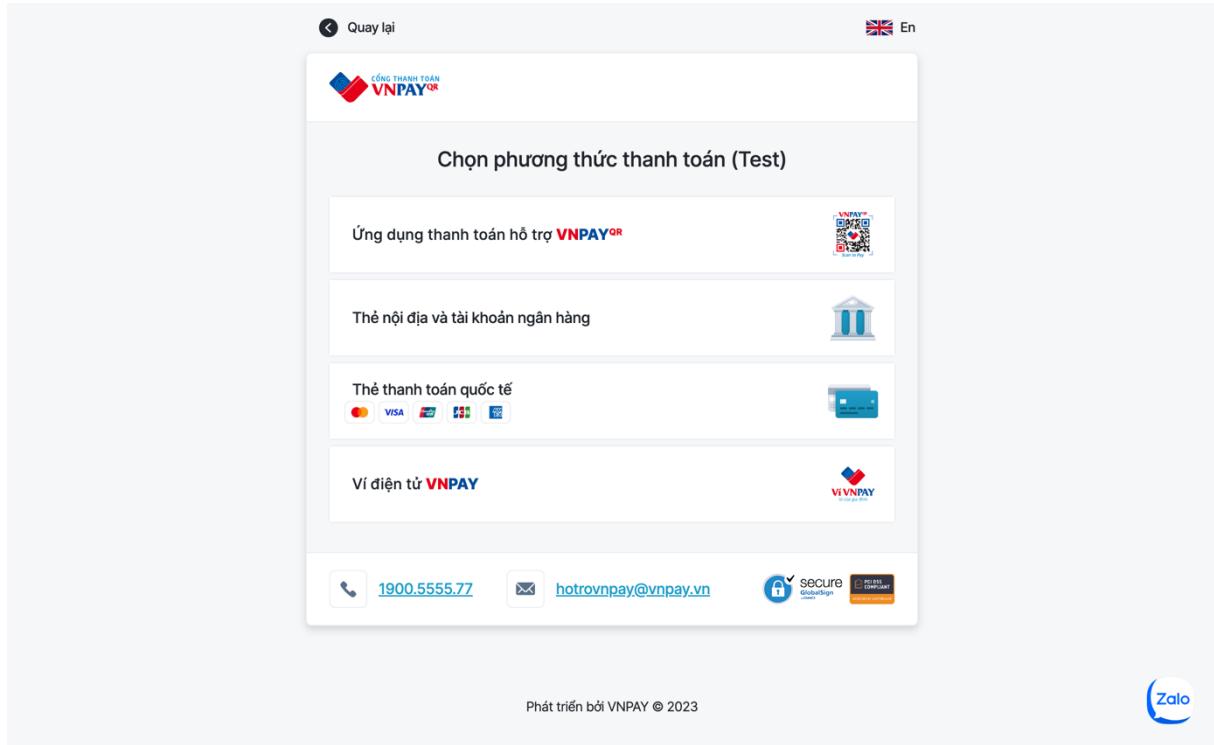
### Bước 2: Tin nhắn chuyển khoản theo cú pháp:

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

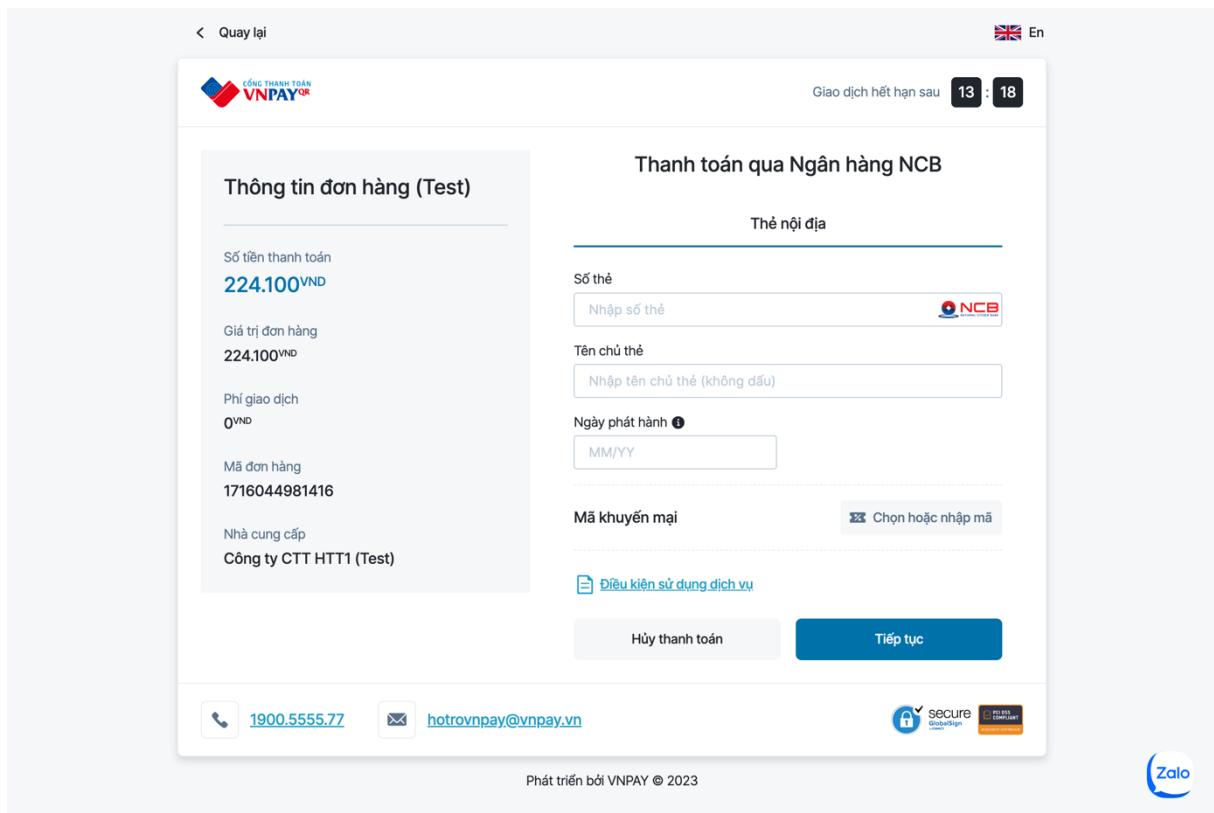
### Bước 3: Chờ quản trị xác nhận

*Hình 6.14: Phần thông báo chuyển khoản trang đặt hàng*

### 6.3.6 Trang thanh toán vnpay



Hình 6.15: Chọn phương thức thanh toán



Hình 6.16: Tiến hành thanh toán

### 6.3.7 Trang xuất đơn hàng

The screenshot shows the order creation process on The Flying Bookstore website. At the top, there's a navigation bar with the logo 'The Flying bookstore', a search bar, and icons for account, cart, and order status. Below the navigation is a breadcrumb trail: 'Điện thoại' → 'Xuất đơn hàng'. The main content area displays a success message: 'Đơn hàng được tạo thành công!' (Order created successfully!). It shows the book details: 'Alchemy and Academe' by Anne McCaffrey, quantity 1, price 2.000đ. Below this is a table of order information:

	Thông tin đặt hàng	Chờ thanh toán	
Mã đơn hàng	20	Trạng thái	
Ngày đặt thuê	19/05/2024	Phương thức thanh toán	
Người cho thuê	Braun Tracy	Chuyển khoản	
Thời gian thuê	19/05/2024 - 23/05/2024	Số ngày thuê	5
Địa chỉ	Đường Kim Giang, Phường Kim Giang, Quận Thanh Xuân, Hà Nội	Tiền thuê	8.000đ
Số điện thoại	0905 907 362	Tiền cọc	120.300đ
		Tổng cộng	128.300đ

A note below the table says: 'Bạn hãy nhanh chóng thanh toán theo phương thức Chuyển khoản để lấy hàng' (Please pay quickly via Transfer to get the item). At the bottom, there are buttons for 'Tiếp tục mua sắm' (Continue shopping), 'Quản lý đơn hàng' (Manage order), and 'Xác nhận lấy hàng' (Confirm pickup).

**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: 'Giữ sách cẩn thận bạn nhé.  
Hãy nhớ sau bạn còn có người khác đọc nữa, nên hãy tránh việc viết vẽ hay làm rách sách bạn nhé.'
- Trả sách**: Includes an icon of a calendar with a checkmark. 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 chủ sách nhận lại được sách, số tiền còn lại sẽ được hoàn về tài khoản của bạn. Nếu bạn cần thêm mua cuốn sách đó luôn chỉ với 1 click.'

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

**Liên hệ**

- Địa chỉ: 171/11 Trường Phước Phan, Bình Hưng Hòa, Bình Tân, TP.HCM
- Số điện thoại: +123 345 123
- Email: support@flying bookstore

**Map**: A Google Map showing the location of 'Tiệm Giường Huyện Cảnh' at 171/11 Trường Phước Phan, Bình Hưng Hòa, Bình Tân, TP.HCM.

Hình 6.17: Trang xuất đơn hàng

Trang xuất 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 và nút xác nhận để đơn hàng có thể chuyển sang trạng thái tiếp theo là xác nhận đã lấy hàng.

### 6.3.8 Trang lấy đơn hàng

The screenshot shows the order summary page for a user named Braun Tracy. The page includes the following details:

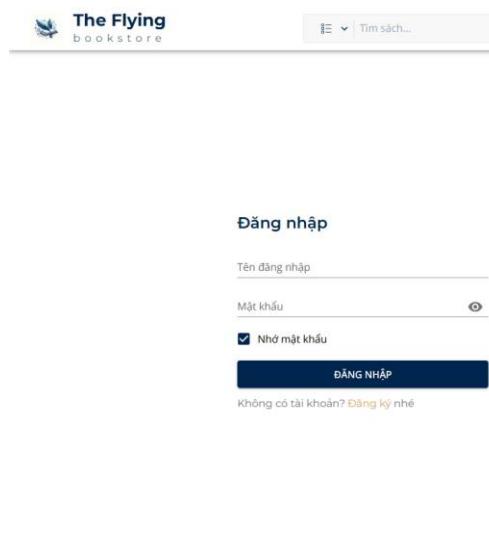
- Order Summary:**
  - Book: Alchemy and Academe by Anne McCaffrey
  - Quantity: 1
  - Delivery Address: Đường Kim Giang, Phường Kim Giang, Quận Thanh Xuân, Hà Nội
  - Phone Number: 0905 907 362
  - Delivery Date: 19/05/2024
  - Delivery Status: Đã lấy hàng (Already picked up)
  - Total Price: 122.300đ
- Order Details Table:**

Mã đơn hàng	20	Trạng thái	Dã 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
- Delivery Instructions:**
  - Đến nhận sách:** Cẩn thận bạn nhé! Hãy nhớ sau ban còn có người khác đọc nữa, nên hay tránh việc viết về làm rách sách bạn nhé.
  - Đọc sách:** Cẩn giữ sách cẩn thận bạn nhé!
  - Trả sách:** Vui lòng trả cuốn sách đúng hạn, ban có thể đặt trả sách nhưng thời gian ship cũng không được quá thời gian trả sách. Sau khi sách nhận lại được sạch, sẽ tiền cọc lại sẽ được hoàn về tài khoản của bạn. Nếu ban cần thêm thời gian, ban có thể gia hạn thêm hoặc mua cuốn sách đó luôn chỉ với 1 click.
- Newsletter Sign-up:** Nhập mail để nhận được những quà tặng đặc biệt nhé! (Input field: Nhập mail tại đây, Button: Đăng ký).
- Contact Information:**
  - The Flying Bookstore logo
  - Address: 177/11 Trương Phuoc Phan, Bình Hưng Hòa, Bình Tân, TP.HCM
  - Phone: +123 345 123
  - Email: support@flying bookstore
  - Social media links: Facebook, YouTube, Instagram
- Map:** A map showing the location of The Flying Bookstore at 177/11 Trương Phuoc Phan, Bình Hưng Hòa, Bình Tân, TP.HCM.

Hình 6.18: 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.

### 6.3.9 Trang đăng nhập



The screenshot shows the login page of 'The Flying bookstore'. At the top, there is a logo for 'The Flying bookstore' and a search bar with the placeholder 'Tim sách...'. On the right side, there are two buttons: 'ĐĂNG NHẬP' (Login) and 'ĐĂNG KÝ' (Sign Up). The main form is titled 'Đăng nhập' (Login). It contains fields for 'Tên đăng nhập' (Username) and 'Mật khẩu' (Password), both with eye icon password helpers. A checked checkbox labeled 'Nhớ mật khẩu' (Remember password) is present. Below the fields is a blue 'ĐĂNG NHẬP' button. At the bottom of the form, there is a link 'Không có tài khoản? Đăng ký nhanh' (Don't have an account? Quick sign up).

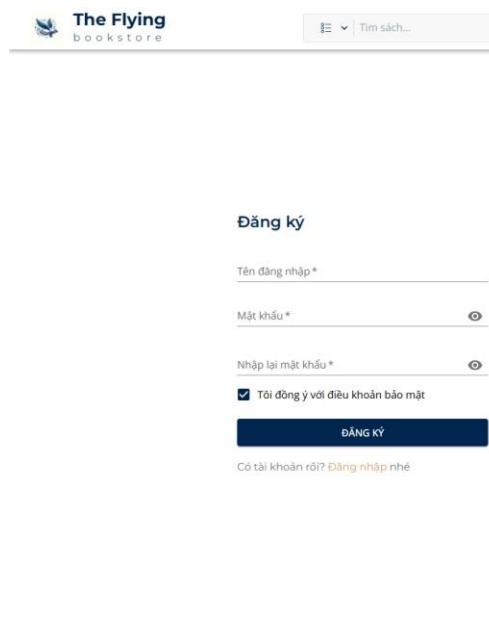


A detailed illustration of a young girl with brown hair, wearing a green dress and a red tie, sitting cross-legged on the floor of a magical library. She is reading a book titled 'Sách Lý'. The room is filled with tall bookshelves packed with books. Several white owls are perched on ledges and flying around. A large, glowing celestial globe sits on a stand in front of her. The ceiling features a stained-glass window showing a starry sky.

Hình 6.19: 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.

### 6.3.10 Trang đăng ký



The screenshot shows the registration page of 'The Flying bookstore'. At the top, there is a logo for 'The Flying bookstore' and a search bar with the placeholder 'Tim sách...'. On the right side, there are two buttons: 'ĐĂNG NHẬP' (Login) and 'ĐĂNG KÝ' (Sign Up). The main form is titled 'Đăng ký' (Sign Up). It contains fields for 'Tên đăng nhập\*' (Username\*), 'Mật khẩu\*' (Password\*), and 'Nhập lại mật khẩu\*' (Re-enter password\*). A checked checkbox labeled 'Tôi đồng ý với điều khoản bảo mật' (I agree with the privacy policy) is present. Below the fields is a blue 'ĐĂNG KÝ' button. At the bottom of the form, there is a link 'Có tài khoản rồi? Đăng nhập nhanh' (Don't have an account? Quick sign up).



A detailed illustration of a grand, sunlit library. The room is filled with tall, arched bookshelves filled with books. Sunlight streams in through a large, circular glass window in the ceiling, illuminating the green plants and vines that grow on the walls and shelves. In the center of the room, there is a red sofa and a small round table. The floor is made of polished wood.

Hình 6.20: Trang đăng ký

### 6.3.11 Trang quản lý thông tin user

Hình 6.21: Trang quản lý thông tin user

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

### 6.3.12 Trang quản lý đơn hàng của tôi

Hình 6.22: Trang quản lý đơn hàng của tôi

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

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

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

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
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 6.23: Đơn hàng có thẻ 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
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

☒

ĐÃ TRẢ TIỀN

☒

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

Hình 6.24: Đơn hàng có thẻ 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
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

☒

ĐÃ NHẬN ĐƯỢC  
HÀNG

Tổng tiền thuê: 13.500đ  
Tổng tiền cọc: 247.000đ  
Tổng tiền: 247.000đ

Hình 6.25: 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 6.26: Đơn hàng xác nhận trả sách

### 6.3.13 Quản lý đơn hàng của khách

Quản lý đơn hàng của khách																																												
<a href="#">TẤT CẢ</a> <a href="#">ĐÃ ĐẶT HÀNG</a> <a href="#">ĐÃ NHẬN</a> <a href="#">ĐÃ QUÁ HẠN</a> <a href="#">ĐÃ TRẢ SÁCH</a> <a href="#">ĐÃ HỦY</a>																																												
Bạn hiện có 7 đơn hàng																																												
<table border="1"> <thead> <tr> <th>Id đơn hàng</th><th>Người thuê</th><th>Thời gian thuê</th><th>Trạng thái chủ sách</th><th>Xem chi tiết</th> </tr> </thead> <tbody> <tr> <td>#16</td><td>hoa hoa</td><td>11/05 - 13/05/2024</td><td>Chủ sách đã đưa hàng</td><td><a href="#">Xem chi tiết</a></td> </tr> <tr> <th>Id bài đăng</th><th>Tên sách</th><th>Thời gian thuê</th><th>Tiền cọc</th><th>Tiền thuê</th></tr> <tr> <td>2003</td><td>tổ tiên</td><td>2 ngày</td><td>120,000</td><td>1,200</td></tr> </tbody> </table> <p><input checked="" type="checkbox"/> Đã giao sách cho người thuê</p> <p>Tổng tiền thuê: 12.000đ Tổng tiền cọc: 120.000đ Tổng tiền: 120.000đ</p> <table border="1"> <thead> <tr> <th>Id đơn hàng</th><th>Người thuê</th><th>Thời gian thuê</th><th>Trạng thái chủ sách</th><th>Xem chi tiết</th> </tr> </thead> <tbody> <tr> <td>#13</td><td>adf adf</td><td>13/05 - 13/05/2024</td><td>Chủ sách đã lấy lại sách</td><td><a href="#">Xem chi tiết</a></td> </tr> <tr> <th>Id bài đăng</th><th>Tên sách</th><th>Thời gian thuê</th><th>Tiền cọc</th><th>Tiền thuê</th></tr> <tr> <td>2000</td><td>From Potter's Field</td><td>1 ngày</td><td>12,312,123</td><td>22</td></tr> </tbody> </table> <p><input checked="" type="checkbox"/> Bạn chờ admin trả lại tiền thuê trong 3-5 ngày làm việc nhé</p> <p>Tổng tiền thuê: 22đ Tổng tiền cọc: 12,312,123đ Tổng tiền: 12,312,123đ</p>					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	<a href="#">Xem chi tiết</a>	Id bài đăng	Tên sách	Thời gian thuê	Tiền cọc	Tiền thuê	2003	tổ tiên	2 ngày	120,000	1,200	Id đơn hàng	Người thuê	Thời gian thuê	Trạng thái chủ sách	Xem chi tiết	#13	adf adf	13/05 - 13/05/2024	Chủ sách đã lấy lại sách	<a href="#">Xem chi tiết</a>	Id bài đăng	Tên sách	Thời gian thuê	Tiền cọc	Tiền thuê	2000	From Potter's Field	1 ngày	12,312,123	22
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	<a href="#">Xem chi tiết</a>																																								
Id bài đăng	Tên sách	Thời gian thuê	Tiền cọc	Tiền thuê																																								
2003	tổ tiên	2 ngày	120,000	1,200																																								
Id đơn hàng	Người thuê	Thời gian thuê	Trạng thái chủ sách	Xem chi tiết																																								
#13	adf adf	13/05 - 13/05/2024	Chủ sách đã lấy lại sách	<a href="#">Xem chi tiết</a>																																								
Id bài đăng	Tên sách	Thời gian thuê	Tiền cọc	Tiền thuê																																								
2000	From Potter's Field	1 ngày	12,312,123	22																																								

Hình 6.27: Quản lý đơn hàng của khách

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

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

Bạn đã được nhận lại sách chưa? ĐÃ NHẬN LẠI SÁCH

Tổng tiền thuê: 1.300đ  
Tổng tiền cọc: 79.200đ  
Tổng tiền: 79.200đ

Hình 6.28: Chủ sách xác nhận nhận lại sách

### 6.3.14 Chi tiết đơn hàng

Mã đơn hàng	Trạng thái	Chỗ thanh toán
20	Bài đăng	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
Dịa chỉ	Đường Kim Giang, Phường Kim Giang, Quận Thanh Xuân, Hà Nội	Tiền cọc
Số điện thoại	0905 907 362	Tổng cộng

Hình 6.29: Chi tiết đơn hàng

### 6.3.15 Quản lý bài đăng

CỘT	BỘ LỌC	BỘ GIẢN	XUẤT	
<input type="checkbox"/> Id bài đăng	<input type="checkbox"/> Tên bài đăng	<input type="checkbox"/> Tác giả	<input type="checkbox"/> Giá thuê	<input type="checkbox"/> Tiền cọc
<input type="checkbox"/> 2	From Pott...	Patricia Daniels C...	2.800đ/ngày	414.800đ/ngày
<input type="checkbox"/> 858	Terminal	Colin Forbes	1.400đ/ngày	262.400đ/ngày
<input type="checkbox"/> 37	To Die for	Janet Neel	800đ/ngày	336.400đ/ngày
<input type="checkbox"/> 13	Vanishing ...	John Nichol	2.300đ/ngày	251.000đ/ngày
<input type="checkbox"/> 14	Hide and ...	Fern Michaels	2.200đ/ngày	230.900đ/ngày

Hình 6.30: Quản lý bài đăng

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

### 6.3.16 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 6.31: 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

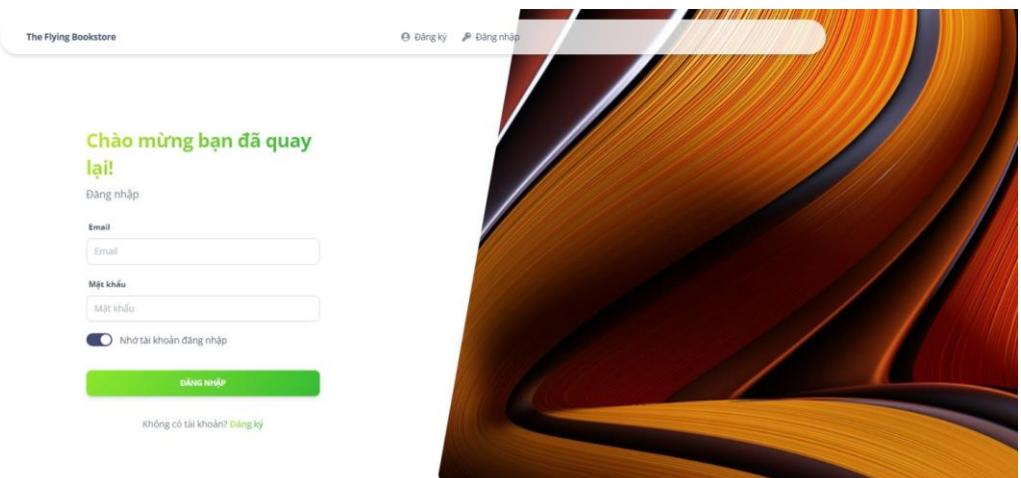
Tim sách \*

Thêm mới sách| X ▾

Tiêu đề *	Tác giả *	
Thể loại		
Nhà xuất bản *	Số trang *	Kích thước
0		
ISBN *	Ngày phát hành *	Ngôn ngữ *
<span>TẠO SÁCH</span>		

Hình 6.32: Thêm mới sách

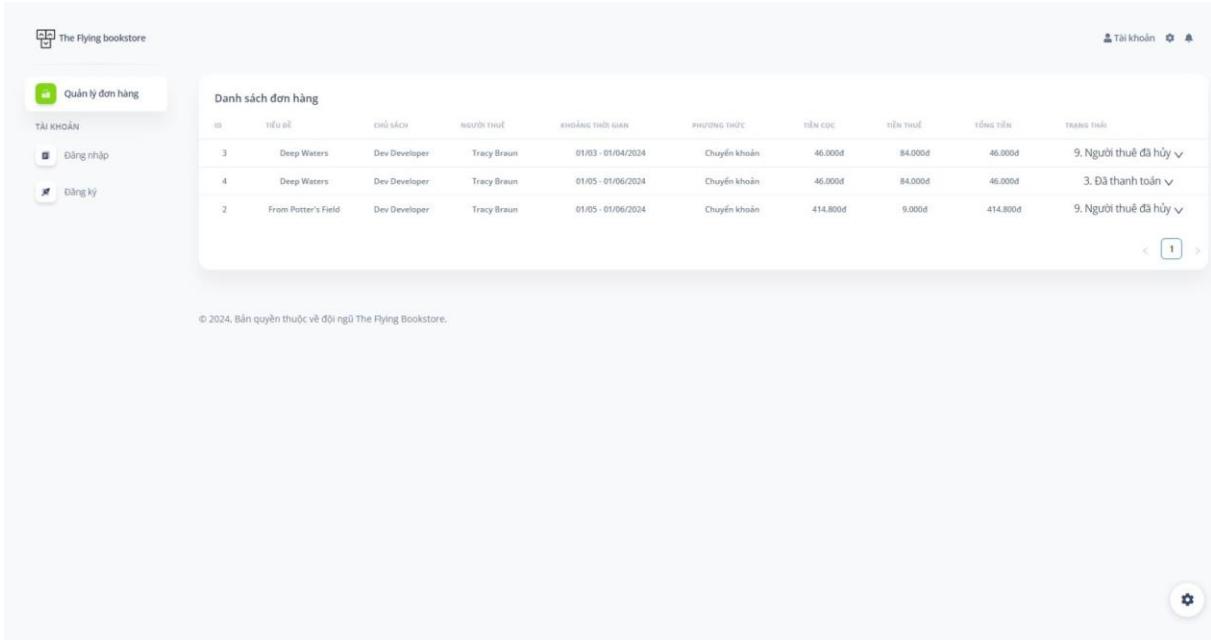
### 6.3.17 Trang đăng nhập admin



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

Hình 6.33: Trang đăng nhập admin

### 6.3.18 Trang quản lý admin



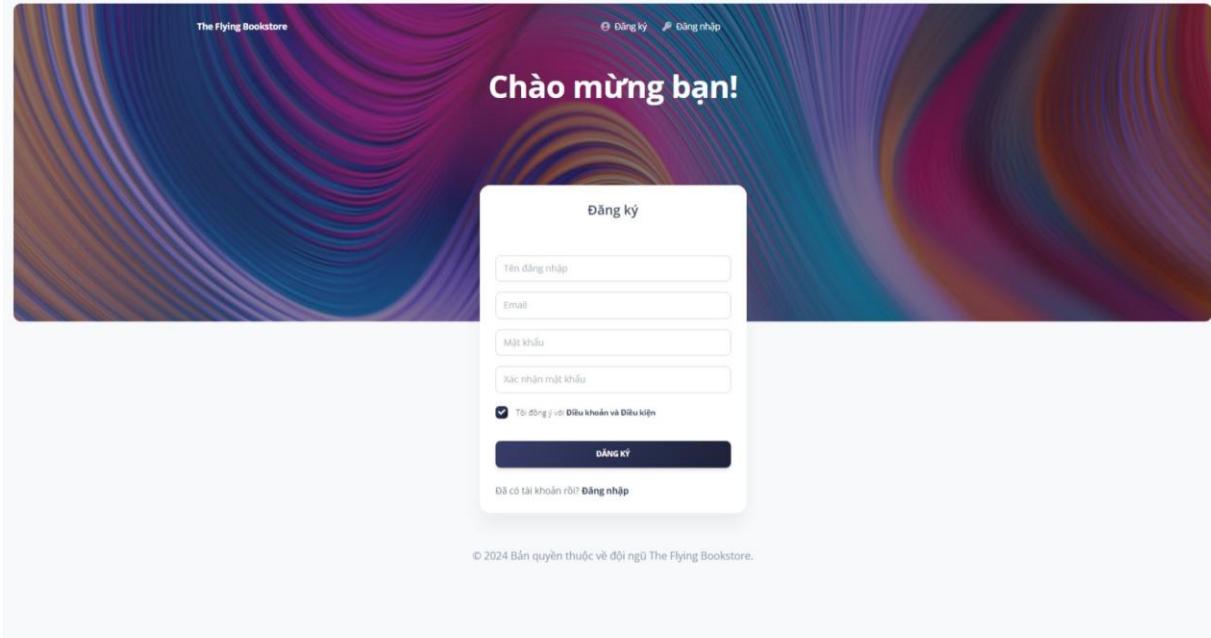
The screenshot shows the 'Quản lý đơn hàng' (Order Management) section of the admin dashboard. It includes a sidebar with 'TÀI KHOẢN' and buttons for 'Đăng nhập' and 'Đăng ký'. The main area displays a table titled 'Danh sách đơn hàng' (List of Orders) with columns: ID, TÊN SÁCH, CHỦ SÁCH, NGƯỜI THUÊ, KHÔNG THỜI GIAN, PHƯƠNG THỨC, TIỀN CỘC, TIỀN THUÊ, TỔNG TIỀN, and TRẠNG THÁI. The table contains three rows of order data. At the bottom, there is a copyright notice: '© 2024. Bản quyền thuộc về đội ngũ The Flying Bookstore.' and a settings gear icon.

ID	TÊN SÁCH	CHỦ SÁCH	NGƯỜI THUÊ	KHÔNG THỜI GIAN	PHƯƠNG THỨC	TIỀN CỘC	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 admin

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.

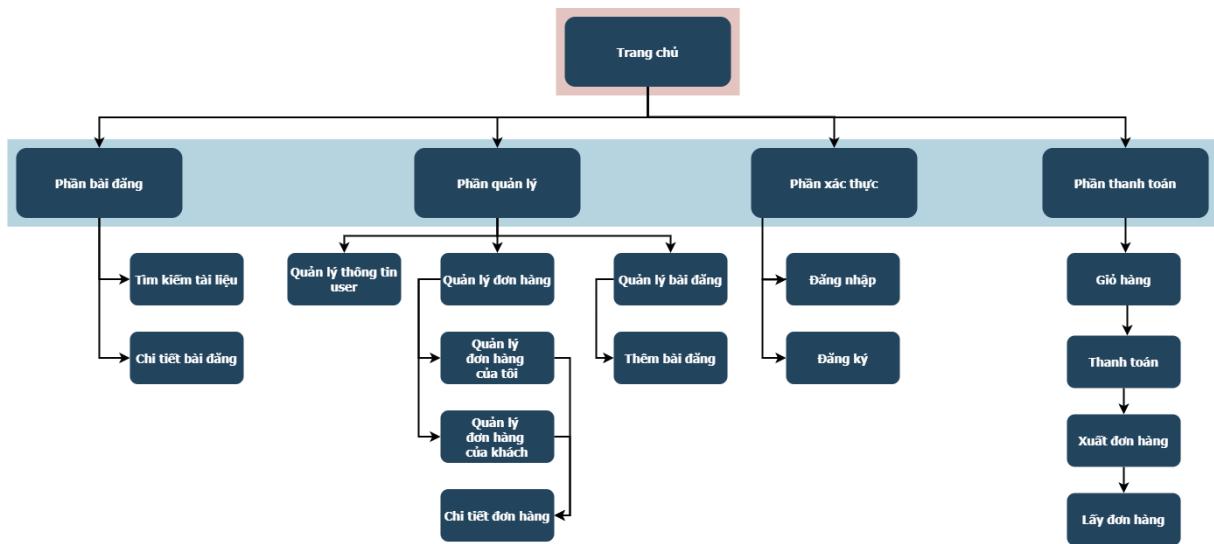
### 6.3.19 Trang đăng ký admin



The screenshot shows the 'Đăng ký' (Registration) page. It features a large purple and blue abstract background with the text 'Chào mừng bạn!' (Welcome!). The registration form is centered, containing fields for 'Tên đăng nhập', 'Email', 'Mật khẩu', and 'Xác nhận mật khẩu'. There is also a checkbox for accepting terms and conditions: 'Tôi đồng ý với Điều khoản và Điều kiện' (I agree with the Terms and Conditions). A 'ĐĂNG KÝ' button is at the bottom, and a link 'Đã có tài khoản rồi? Đăng nhập' (Already have an account? Log in) is below it. At the bottom, there is a copyright notice: '© 2024 Bản quyền thuộc về đội ngũ The Flying Bookstore.'

Hình 6.35: Trang đăng ký admin

#### 6.4 Site map giao diện hệ thống



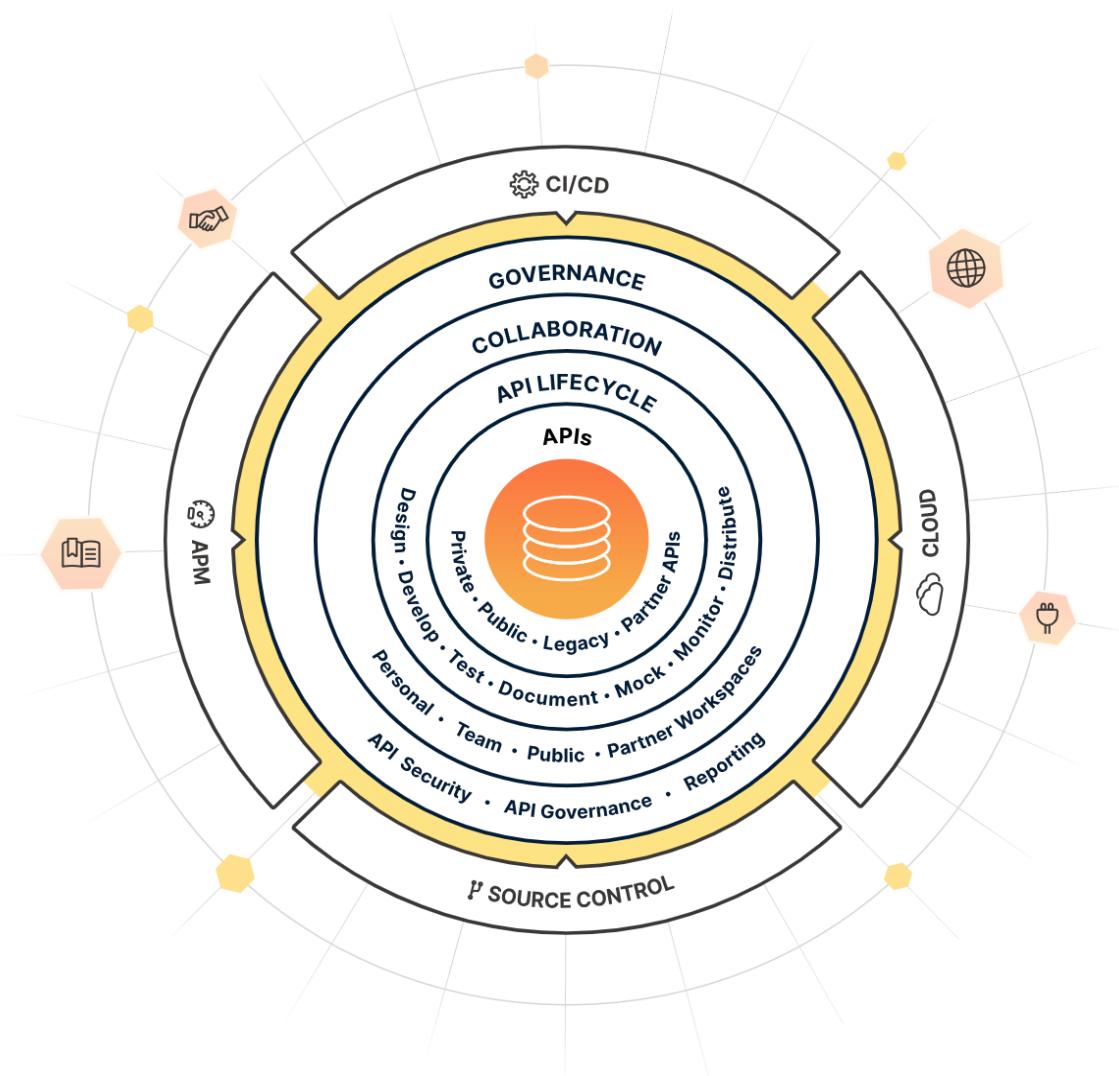
Hình 6.34: Site map giao diện hệ thống

## 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:

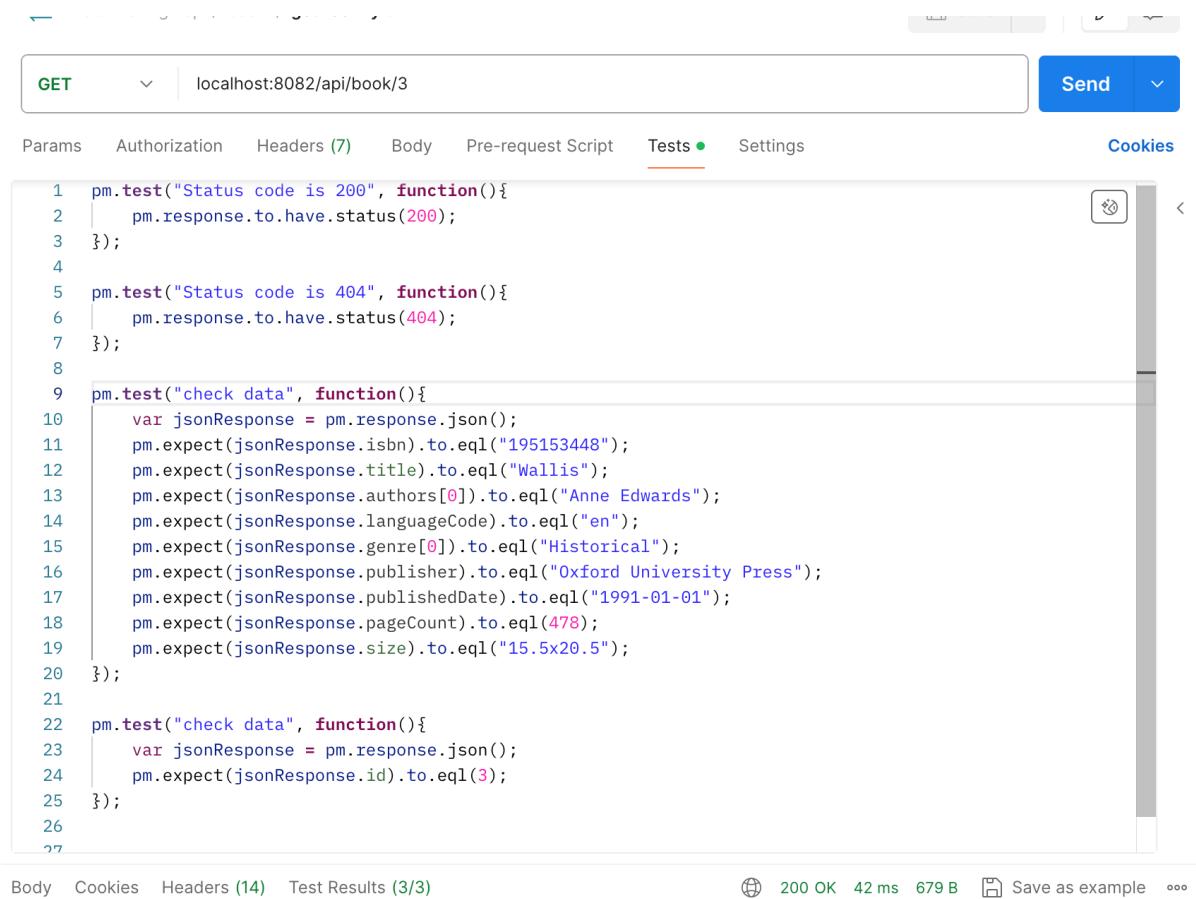
- Hỗ trợ nhiều trình duyệt: Playwright không chỉ hỗ trợ một trình duyệt duy nhất mà còn hỗ trợ Chromium, Firefox và WebKit. Điều này cho phép bạn kiểm thử ứng dụng web của mình trên nhiều trình duyệt khác nhau từ một mã nguồn duy nhất.
- API mạnh mẽ và linh hoạt: Playwright cung cấp một API để sử dụng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



```
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
```

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 for a GET request to `localhost:8082/api/book/3`. The 'Tests' tab is selected, displaying a JavaScript code block for assertions:

```

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['id']).to.eql("105453448");
12    pm.expect(jsonPath['local var']).to.eql("is");
13    pm.expect(jsonPath['authors[0]']).to.eql("Anne Edwards");
14    pm.expect(jsonPath['languageCode']).to.eql("en");
15    pm.expect(jsonPath['genre[0]']).to.eql("Historical");
}

```

Below the tests, the 'Test Results' section shows three green 'PASS' status indicators:

- PASS Status code is 200
- PASS check data
- PASS check data

At the bottom, the results are summarized as 200 OK, 42 ms, and 679 B.

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 user 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 user because reasons"	pass
		Kiểm tra không khai báo password	Trả về message "Error creating new user 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	searchBookByTitle	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 (tìm kiếm thành công)	Dữ liệu trả về list gồm đầy đủ các trường như mong muốn gồm: - isbn - title - authors - languageCode - genre - publisher - publishedDate - pageCount - size	pass
		Kiểm tra kết quả trả về với từ khoá đặc biệt @, . ,#,123,... (tìm kiếm thành công)	Dữ liệu trả về list gồm đầy đủ các trường như mong muốn gồm: - isbn - title - authors - languageCode - genre - publisher	pass

			- 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: - 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: - 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 gồm: - isbn - title - authors - languageCode	pass

			- 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: - id - isbn - title - authors - languageCode - genre - publisher - publishedDate - pageCount - size	pass
listing	getDetailListingById	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 : - 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	Status trả về là 200 (thành công) size của một trang bằng 20 hiển thị trang 0 là trang đầu tiên "Trả về đầy đủ và đúng các trường mong muốn gồm : - id - user - address - leaseRate - depositFee - penaltyRate - description - listingStatus - copy - book - review - bookOwned - bookLeasing	pass
	Kiểm tra dữ liệu khi có param size phân trang	Status trả về là 200 (thành công) size của một trang bằng size param truyền vào hiển thị trang 0 là trang đầu tiên "Trả về đầy đủ và đúng các trường mong muốn gồm : - id - user - address - leaseRate - depositFee - penaltyRate - description - listingStatus - copy - book - review - bookOwned - bookLeasing	pass
	Kiểm tra dữ liệu khi có param page phân trang không có param page phân trang	Status trả về là 200 (thành công) size của một trang bằng size param truyền vào trang hiển thị là trang của param truyền vào "Trả về đầy đủ và đúng các trường mong muốn trong content gồm : - id	pass

		<ul style="list-style-type: none"> <li>- user</li> <li>- address</li> <li>- leaseRate</li> <li>- depositFee</li> <li>- penaltyRate</li> <li>- description</li> <li>- listingStatus</li> <li>- copy</li> <li>- book</li> <li>- review</li> <li>- bookOwned</li> <li>- bookLeasing</li> </ul>	
	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"> <li>- id</li> <li>- user</li> <li>- address</li> <li>- leaseRate</li> <li>- depositFee</li> <li>- penaltyRate</li> <li>- description</li> <li>- listingStatus</li> <li>- copy</li> <li>- book</li> <li>- review</li> <li>- bookOwned</li> <li>- bookLeasing</li> </ul>	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"> <li>- id</li> <li>- user</li> <li>- address</li> <li>- leaseRate</li> <li>- depositFee</li> <li>- penaltyRate</li> <li>- description</li> <li>- listingStatus</li> <li>- copy</li> <li>- book</li> </ul>	pass

		<ul style="list-style-type: none"> <li>- review</li> <li>- bookOwned</li> <li>- bookLeasing</li> </ul>	
	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"> <li>- id</li> <li>- user</li> <li>- address</li> <li>- leaseRate</li> <li>- depositFee</li> <li>- penaltyRate</li> <li>- description</li> <li>- listingStatus</li> <li>- copy</li> <li>- book</li> <li>- review</li> <li>- bookOwned</li> <li>- bookLeasing</li> </ul>	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"> <li>- id</li> <li>- user</li> <li>- address</li> <li>- leaseRate</li> <li>- depositFee</li> <li>- penaltyRate</li> <li>- description</li> <li>- listingStatus</li> <li>- copy</li> <li>- book</li> <li>- review</li> <li>- bookOwned</li> <li>- bookLeasing</li> </ul>	pass
	Kiểm tra tìm dữ liệu khi có cả hai 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</p>	pass

			<p>trường mong muốn trong content gồm:</p> <ul style="list-style-type: none"> <li>- id</li> <li>- user</li> <li>- address</li> <li>- leaseRate</li> <li>- depositFee</li> <li>- penaltyRate</li> <li>- description</li> <li>- listingStatus</li> <li>- copy</li> <li>- book</li> <li>- review</li> <li>- bookOwned</li> <li>- bookLeasing</li> </ul>	
		Kiểm tra dữ liệu khi không tìm thấy	<p>Status trả về là 200 (thành công)</p> <p>size của một trang bằng 20</p> <p>hiển thị trang 0 là trang đầu tiên</p> <p>content là một list rỗng</p>	pass
	listing search by id and title	Kiểm tra status trả về	<p>Status trả về là 200 (thành công)</p>	pass
		Kiểm tra dữ liệu khi request title bởi nhiều từ khoá: @, . ,#,123,... (tìm kiếm thành công)	<p>Status trả về là 200 (thành công)</p> <p>size của một trang bằng 20</p> <p>hiển thị trang 0 là trang đầu tiên</p> <p>Trả về đầy đủ và đúng các trường mong muốn trong content gồm:</p> <ul style="list-style-type: none"> <li>- id</li> <li>- user</li> <li>- address</li> <li>- leaseRate</li> <li>- depositFee</li> <li>- penaltyRate</li> <li>- description</li> <li>- listingStatus</li> <li>- copy</li> <li>- book</li> <li>- review</li> <li>- bookOwned</li> <li>- bookLeasing</li> </ul>	pass
		Kiểm tra dữ liệu khi không tìm thấy	<p>Status trả về là 200 (thành công)</p> <p>size của một trang bằng 20</p> <p>hiển thị trang 0 là trang đầu tiên</p> <p>content là một list rỗng</p>	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 Dữ liệu trả về giống với dữ liệu truyền từ body request	pass
	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	pass

			<ul style="list-style-type: none"> <li>- score</li> <li>- description</li> <li>- leaseOrderId</li> <li>- userId</li> <li>- listingId</li> <li>- createdDate</li> </ul> <p>Dữ liệu trả về giống với dữ liệu truyền từ body request</p>	
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	<p>Trả về đầy đủ các trường mong muốn gồm:</p> <ul style="list-style-type: none"> <li>- id</li> <li>- score</li> <li>- description</li> <li>- leaseOrderId</li> <li>- userId</li> <li>- listingId</li> <li>- createdDate</li> </ul> <p>Dữ liệu trả về giống với dữ liệu truyền từ body request</p>	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	<p>Trả về các trường sau đúng với dữ liệu body request truyền vào:</p> <ul style="list-style-type: none"> <li>- listingId</li> <li>- status</li> <li>- lesseeId</li> <li>- lesseeAddress</li> <li>- fromDate</li> <li>- toDate</li> <li>- paymentMethod</li> </ul>	pass
		Kiểm tra tạo thành công lease order details	<p>Trả về có leaseOrderdetail với các trường mong muốn đó là :</p> <ul style="list-style-type: none"> <li>- id</li> <li>- title</li> <li>- listingId</li> <li>- leaseRate</li> <li>- depositFee</li> <li>- penaltyRate</li> </ul>	pass
		Kiểm tra tạo thành công lease order details với các loại	Trả về có leaseOrderdetail với các trường mong muốn như tạo thành	pass

		method payment (COD, VNPay, BANK_TRANSFER),	như tạo thành công. Và trả về method payment đúng với method payment truyền vào từ request body	
		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 leaseAndDepositPaymentId	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 business 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	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 - listing - lessor	pass
	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	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 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
get lease order by lessor id and status	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 - listing - lessor	pass
	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 and status	Kiểm tra với từng status lựa chọn	Trả về đúng status theo business logic	pass
	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 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	Kiểm tra status trả về là 200 (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	Trả về id giống với id truyền vào khi thành công	pass
change status	Kiểm tra status trả về khi thành công	Kiểm tra status trả về là 200 (thành công)	Status trả về là 200 (thành công)	pass
	Kiểm tra status trả về khi không cho phép đổi status	Kiểm tra status trả về là 423 message: Error update 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	Kiểm tra status trả về giống với status truyền vào trong param khi thành công	Status trả về giống với status truyền vào trong param khi thành công	pass

Bảng 7.1: Kích 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

<b>Kịch bản kiểm thử tổng hợp dự án xây dựng hệ thống cho thuê sách cũ giá hợp lý</b>				
tool: playwright				
Mã TC	Tên test case	Tiền điều kiện	Các bước thực hiện	Kết quả mong muốn
<b>LOGIN</b>				
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
<b>chức năng user</b>				
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 user 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 đổi 4. Nhấn nút "xác nhận"	Thay đổi thông tin user thành công
<b>chức năng bài đăng</b>				

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 gio hàng	Thêm bài đăng vào gio 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 gio hàng thành công
xoá bài đăng khỏi gio hàng	Xoá bài đăng khỏi gio 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 gio 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 quản lý bài đăng	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 đăng thành công
chọn book	Chọn book	Đăng nhập acc dev1/123456 thành công	1. Nhấn icon quản lý bài đăng trên thanh navbar	Chọn sách trong tạo

	trong tạo bài đăng		2. Nhấn nút thêm bài đăng 3. Trong mục sách nhấn nút tìm sách và chọn sách 4. Bấm nút "chọn sách"	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	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 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 cọc cho hệ thống	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 chờ thanh toán >> Nhấn vào nút "Đã trả tiền"	Xác nhận 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
<b>chức năng quản lý đơn hàng của khách</b>				
xem danh sách đơn hàng	Xem danh sách đơn hàng của tôi	Đăng nhập acc dev/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 admin để nâng cao khả năng làm việc và giảm thiểu thời gian tiêu tốn.
- Tích hợp các chức năng giao tiếp giữa user 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 user interfaces*. Truy cập từ: <https://legacy.reactjs.org/>.
3. legacy.reactjs.org. 2023. *React – A JavaScript library for building user interfaces*. Truy cập từ: <https://legacy.reactjs.org/>.
4. react.dev. 2023. *React Community – React*. Truy cập từ: <https://react.dev/community>.
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>.