

BỘ THÔNG TIN VÀ TRUYỀN THÔNG  
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



**BÁO CÁO THỰC TẬP  
TỐT NGHIỆP ĐẠI HỌC**

**Đề tài:**

**“Xây dựng ứng dụng web tìm kiếm đặt mua sách trực tuyến”**

**Người hướng dẫn : Thạc sĩ Nguyễn Anh Hào**

**Sinh viên thực hiện : Trần Thư Đạt**

**Mã số sinh viên : N19DCCN036**

**Lớp : D19CQCNPM01-N**

**Khóa : 2019 - 2024**

**Hệ : ĐẠI HỌC CHÍNH QUY**

**TP. HCM, tháng 7 năm 2023**

BỘ THÔNG TIN VÀ TRUYỀN THÔNG  
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



**BÁO CÁO THỰC TẬP  
TỐT NGHIỆP ĐẠI HỌC**

**Đề tài:**

**“Xây dựng ứng dụng web tìm kiếm đặt mua sách trực tuyến”**

**Người hướng dẫn : Thạc sĩ Nguyễn Anh Hào**

**Sinh viên thực hiện : Trần Thư Đạt**

**Mã số sinh viên : N19DCCN036**

**Lớp : D19CQCNPM01-N**

**Khóa : 2019 - 2024**

**Hệ : ĐẠI HỌC CHÍNH QUY**

**TP. HCM, tháng 7 năm 2023**

## **LỜI CẢM ƠN**

Lời nói đầu tiên, em xin được gửi lời cảm ơn chân thành nhất đến các thầy cô trong Khoa Công nghệ Thông tin 2 đã tạo mọi điều kiện thuận lợi để em có thể thực hiện được bài báo cáo thực tập tốt nghiệp này. Đồng thời, em xin gửi lời cảm ơn chân thành đến thầy phụ trách hướng dẫn trực tiếp cho em-thạc sĩ Nguyễn Anh Hào, đã không ngần ngại những khó khăn, vất vả để luôn tích cực hỗ trợ em hết mình từ ngày bắt đầu cho đến hôm nay.

Trong quá trình thực hiện bài viết báo cáo, nếu như có nhiều những thiếu sót, em xin các thầy cô trong khoa có thể tiếp tục hỗ trợ, đóng góp ý kiến, phản hồi cũng như cung cấp thêm những hướng dẫn chi tiết cụ thể về mặt chuyên môn để em có thể thực hiện được nội dung được hoàn chỉnh nhất cho quá trình thực tập tốt nghiệp.

Thành phố Hồ Chí Minh, tháng 7/2023  
Sinh viên thực hiện

Trần Thu Đạt

# Mục lục

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI .....	1
1.1. Tên đề tài .....	1
1.2. Loại đề tài .....	1
1.3. Mục đích .....	1
1.4 Mục tiêu .....	1
CHƯƠNG 2: CƠ SỞ KHOA HỌC ĐỀ TÀI.....	2
2.1 Giới thiệu về Visual studio code.....	2
2.2 Giới thiệu về MySQL .....	2
2.3 Giới thiệu về Typescript : .....	3
2.4 Tổng quan về NestJS : .....	4
2.5 Tổng quan về TypeORM: .....	4
2.6 Giới thiệu về React JS .....	5
2.7 Giới thiệu về Restful.....	6
2.8 Tìm kiếm dựa trên mô hình Boolean .....	6
2.9 MapBox .....	7
CHƯƠNG 3: PHÂN TÍCH HỆ THỐNG.....	8
3.1 Bối cảnh/ hiện trạng của hệ thống .....	8
3.2 Lược đồ Usecase cho phân tích .....	11
3.3 Định nghĩa yêu cầu và ràng buộc đối với phần mềm .....	18
CHƯƠNG 4: THIẾT KẾ HỆ THỐNG.....	22
4.1. Lược đồ usecase cho thiết kế : .....	22
4.2 Thiết kế các thành phần của phần mềm.....	30
4.3. Thiết kế cơ sở dữ liệu và ràng buộc .....	46
4.4. Thiết kế dữ liệu .....	49
CHƯƠNG 5: CHƯƠNG TRÌNH .....	60
5.1 Giao diện trang chủ :.....	60
5.2 Giao diện tìm kiếm : .....	60

# BÁO CÁO TTTN ĐẠI HỌC

---

---

5.3 Giao diện chi tiết sách : .....	61
5.4. Giao diện giỏ hàng :.....	61
5.5. Giao diện khách xác nhận đơn hàng :.....	62
5.6. Giao diện đơn hàng :.....	62
5.7 Giao diện cho khách đánh giá sản phẩm : .....	63
5.8. Giao diện thêm thông tin nhận hàng :.....	63
5.9. Giao diện danh sách nhận hàng của khách hàng .....	64
5.10. Giao diện admin.....	64
CHƯƠNG 6: TỔNG KẾT .....	65
6.1 Kết quả đạt được :.....	65
6.2 Hạn chế của đề tài :.....	65
6.3 Hướng phát triển trong tương lai : .....	65
DANH MỤC TÀI LIỆU THAM KHẢO.....	66

**KÝ HIỆU CÁC CỤM TỪ VIẾT TẮT**

API: Application Programming Interface	Giao diện lập trình ứng dụng
ERD: Entity Relationship Diagram	Mô hình thực thể kết hợp
REST: Representational State Transfer	Một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web
UI: User Interface	Giao diện người dùng

# Danh mục các hình vẽ

Hình 1: Config kết nối TypeORM và Mysql .....	3
Hình 2: Config TypeORM ở module gốc .....	3
Hình 3: Biểu đồ cộng tác khi chưa có phần mềm .....	10
Hình 4: Biểu đồ cộng tác sau khi có phần mềm.....	11
Hình 5: Usecase khách hàng tìm sách .....	11
Hình 6: Usecase thêm sách vào giỏ hàng .....	12
Hình 7: Usecase khách hàng đặt sách .....	14
Hình 8: Usecase khách hàng đánh giá sản phẩm .....	15
Hình 9: Usecase khách thêm thông tin nhận hàng .....	16
Hình 10: Usecase nhân viên tạo khuyến mãi .....	17
Hình 11: Usecase nhân viên cập nhật trạng thái đơn hàng .....	18
Hình 12: Usecase khách hàng tìm sách .....	22
Hình 13: Lược đồ tuần tự cho Usecase tìm kiếm theo thuộc tính .....	22
Hình 14: Lược đồ tuần tự cho Usecase tìm kiếm theo tag .....	23
Hình 15: Lược đồ tuần tự cho Usecase tìm kiếm theo đánh giá .....	23
Hình 16: Usecase thêm sách vào giỏ hàng .....	24
Hình 17: Lược đồ tuần tự cho Usecase thêm sách vào giỏ hàng .....	24
Hình 18: Usecase khách hàng đặt sách .....	25
Hình 19: Lược đồ tuần tự cho Usecase khách hàng đặt sách .....	25
Hình 20: Lược đồ tuần tự cho Usecase xoá sách khỏi giỏ hàng .....	26
Hình 21: Usecase khách hàng đánh giá sản phẩm .....	26
Hình 22: Lược đồ tuần tự cho Usecase đánh giá sản phẩm .....	27
Hình 23: Usecase khách thêm thông tin nhận hàng .....	27
Hình 24: Lược đồ tuần tự cho Usecase thêm thông tin nhận hàng .....	28
Hình 25: Usecase nhân viên tạo khuyến mãi .....	28
Hình 26: Lược đồ tuần tự cho Usecase nhân viên tạo khuyến mãi theo loại khách hàng .....	29
Hình 27: Usecase nhân viên cập nhật trạng thái đơn hàng .....	29
Hình 28: Lược đồ tuần tự cho Usecase nhân viên cập nhật trạng thái đơn hàng .....	30
Hình 29: Form tìm kiếm theo thuộc tính của sách .....	33
Hình 30: Form lọc sách theo custom key hoặc đánh giá .....	34
Hình 31: Form hiển thị thông tin sách và thêm vào giỏ hàng .....	34
Hình 32: Form xác nhận thông tin đặt hàng .....	35
Hình 33: Form chi tiết đơn hàng .....	36
Hình 34: Form khách hàng đánh giá sản phẩm .....	37
Hình 35: Form khách hàng thêm thông tin nhận hàng .....	38
Hình 36: Form thêm khuyến mãi cho loại khách hàng .....	39
Hình 37: Form thay đổi trạng thái của đơn hàng .....	39
Hình 38: Mô hình ERD .....	48

---

---

Hình 39: Mô hình Diagram .....	50
Hình 40: Giao diện trang chủ.....	60
Hình 41: Giao diện tìm kiếm .....	60
Hình 42: Giao diện chi tiết sách.....	61
Hình 43: Giao diện giỏ hàng .....	61
Hình 44: Giao diện khách xác nhận đơn hàng.....	62
Hình 45: Giao diện đơn hàng .....	62
Hình 46: Giao diện khách hàng đánh giá sản phẩm.....	63
Hình 47: Giao diện thêm thông tin nhận sách.....	63
Hình 48: Giao diện danh sách thông tin nhận hàng .....	64
Hình 49: Giao diện admin.....	64

# Danh mục các bảng

<i>Bảng 1: Bảng công thức tính phí giao hàng .....</i>	8
<i>Bảng 2 : Đặc tả Usecase khách hàng tìm sách .....</i>	12
<i>Bảng 3: Đặc tả Usecase thêm sách vào giỏ hàng.....</i>	13
<i>Bảng 4: Đặc tả Usecase khách hàng đặt sách.....</i>	15
<i>Bảng 5: Đặc tả Usecase khách hàng đánh giá sản phẩm.....</i>	15
<i>Bảng 6: Đặc tả Usecase khách hàng thêm thông tin nhận hàng .....</i>	16
<i>Bảng 7: Đặc tả Usecase nhân viên thêm khuyến mãi.....</i>	17
<i>Bảng 8: Đặc tả Usecase nhân viên cập nhật trạng thái đơn hàng .....</i>	18
<i>Bảng 9: Yêu cầu từ môi trường nghiệp vụ (bussiness) .....</i>	19
<i>Bảng 10: Yêu cầu từ môi trường vận hành .....</i>	20
<i>Bảng 11: Yêu cầu từ môi trường phát triển .....</i>	21
<i>Bảng 12: Từ điển dữ liệu bảng Rate .....</i>	50
<i>Bảng 13: Từ điển dữ liệu bảng Author .....</i>	51
<i>Bảng 14: Từ điển dữ liệu bảng Book_type .....</i>	51
<i>Bảng 15: Từ điển dữ liệu bảng Publisher .....</i>	52
<i>Bảng 16: Từ điển dữ liệu bảng Compositions.....</i>	52
<i>Bảng 17: Từ điển dữ liệu bảng Book .....</i>	53
<i>Bảng 18: Từ điển dữ liệu bảng Customer_type .....</i>	53
<i>Bảng 19: Từ điển dữ liệu bảng Customer.....</i>	54
<i>Bảng 20: Từ điển dữ liệu bảng Receipt_information .....</i>	55
<i>Bảng 21: Từ điển dữ liệu bảng Staff.....</i>	55
<i>Bảng 22: Từ điển dữ liệu bảng Promotion .....</i>	56
<i>Bảng 23: Từ điển dữ liệu bảng Promotion_customer .....</i>	56
<i>Bảng 24: Từ điển dữ liệu bảng Promotion_book .....</i>	57
<i>Bảng 25: Từ điển dữ liệu bảng Status .....</i>	57
<i>Bảng 26: Từ điển dữ liệu bảng Cart .....</i>	57
<i>Bảng 27: Từ điển dữ liệu bảng Cart_detail .....</i>	58
<i>Bảng 28: Từ điển dữ liệu bảng Invoice .....</i>	58
<i>Bảng 29: Từ điển dữ liệu bảng Status_invoice .....</i>	59

# CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

**1.1. Tên đề tài :** Xây dựng ứng dụng web tìm kiếm và đặt mua sách trực tuyến.

**1.2. Loại đề tài :** Xây dựng một phần mềm ứng dụng.

## 1.3. Mục đích

Mục đích của đề tài xây dựng đề tài website tìm kiếm và đặt sách trực tuyến là tạo ra một nền tảng trực tuyến cho khách hàng tìm và đặt mua các đầu sách khác nhau.

Cụ thể, đề tài nhằm giúp người dùng tìm kiếm và đặt mua sách một cách dễ dàng và tiện lợi thông qua cung cấp một số chức năng sau :

- **Tìm kiếm dễ dàng :** Giúp người dùng có thể chọn được các đầu sách ưng ý với mong muốn của bản thân thông qua áp dụng mô hình tìm kiếm boolean thông qua nhiều tiêu chí, qua đó tăng khả năng khách hàng tìm được đầu sách mà họ ưng ý.
- **Đặt mua thuận tiện :** Người dùng có thể nhập địa điểm giao hàng. Website sẽ tính toán khoản cách và cộng với trọng lượng của đơn hàng và áp dụng các khuyến mãi nếu có và sau đó tính ra số tiền khách hàng phải trả.

## 1.4 Mục tiêu

Mục tiêu của đề tài website tìm kiếm và đặt sách trực tuyến là xây dựng một nền tảng trực tuyến hiệu quả và tiện lợi để giúp cho người dùng có thể tìm kiếm và đặt mua sách dễ dàng và nhanh chóng. Đồng thời giúp các nhân viên trong nhà sách có thể dễ dàng quản lý thông tin hơn.

Cụ thể đề tài nhằm đạt được các mục tiêu sau :

- **Tạo ra giao diện đơn giản và dễ sử dụng :** Thiết kế giao diện trực quan, dễ dàng sử dụng và thân thiện với người dùng, giúp người dùng tìm kiếm và đặt mua sách nhanh chóng và dễ dàng.
- **Cung cấp thông tin chi tiết và đầy đủ về các đầu sách :** Cập nhật đầy đủ các thông tin của sách để người dùng không bị bối rối có nên mua hay không.
- **Nâng cao trải nghiệm người dùng :** Đưa ra các tính năng và chức năng mới để nâng cao trải nghiệm người dùng giúp tìm kiếm đặt mua thuận lợi nhanh chóng.
- **Tìm kiếm chính xác và đặt mua dễ dàng đơn giản.**

# **CHƯƠNG 2: CƠ SỞ KHOA HỌC ĐỀ TÀI**

## **2.1 Giới thiệu về Visual studio code**

Visual Studio Code (VS Code hay VSC) là một trong những trình soạn thảo mã nguồn phổ biến nhất được sử dụng bởi các lập trình viên. Nhanh, nhẹ, hỗ trợ đa nền tảng, nhiều tính năng và là mã nguồn mở chính là những ưu điểm vượt trội khiến VS Code ngày càng được ứng dụng rộng rãi.

Visual Studio Code hỗ trợ chức năng debug, đi kèm với Git, có syntax highlighting, tự hoàn thành mã thông minh, snippets, và cài tiến mã nguồn. Nhờ tính năng tùy chỉnh, Visual Studio Code cũng cho phép người dùng thay đổi theme, phím tắt, và các tùy chọn khác.

## **2.2 Giới thiệu về MySQL**

MySQL là một hệ thống quản trị cơ sở dữ liệu mã nguồn mở (gọi tắt là RDBMS) hoạt động theo mô hình client-server. Với RDBMS là viết tắt của Relational Database Management System. MySQL được tích hợp apache, PHP. MySQL quản lý dữ liệu thông qua các cơ sở dữ liệu. Mỗi cơ sở dữ liệu có thể có nhiều bảng quan hệ chứa dữ liệu. MySQL cũng có cùng một cách truy xuất và mã lệnh tương tự với ngôn ngữ SQL.

MySQL là một sản phẩm cơ sở dữ liệu nguồn mở được tạo bởi MySQL AB, một công ty được thành lập năm 1995 tại Thụy Điển. Năm 2008, Sun Microsystems đã mua lại MySQL AB với giá xấp xỉ 1 tỷ đô la.

MySQL tương thích để chạy trên nhiều hệ điều hành, như Novell NetWare, Windows, Linux , nhiều loại UNIX (như Sun, Solaris, AIX và DEC, UNIX), FreeBSD. MySQL cũng cung cấp một phương tiện mà các máy khách có thể chạy trên cùng một máy tính với máy chủ hoặc trên một máy tính khác (giao tiếp qua mạng cục bộ hoặc Internet).

MySQL không yêu cầu quá nhiều công tác bảo trì bởi có tính ổn định cao. Do đó, nếu bạn phát triển các ứng dụng dựa trên MySQL, chi phí sở hữu sẽ thấp hơn so với các hệ thống quản trị dữ liệu khác.

Để kết nối MySQL với NestJs ta phải cài gói Nestjs TypeORM. Sau đó tạo một 1 file để config các cài đặt cho kết nối có dạng như sau :

```

const config: TypeOrmModuleOptions = {
  type: 'mysql',
  host: 'localhost', // Địa chỉ MySQL server
  port: 3306, // Cổng MySQL server
  username: 'your_username', // Tên đăng nhập MySQL
  password: 'your_password', // Mật khẩu MySQL
  database: 'your_database', // Tên cơ sở dữ liệu MySQL
  entities: [__dirname + '**/*.{entity{.ts,.js}}'], // Đường dẫn đến
  các entity trong ứng dụng
  synchronize: true, // Đồng bộ hóa cấu trúc cơ sở dữ liệu (chỉ sử
  dụng trong môi trường phát triển)
};

```

*Hình 1: Config kết nối TypeORM và Mysql*

Sau khi tạo file config ta thêm TypeORM module và cấu hình trong module gốc của ứng dụng NestJS.

```

@Module({
  imports: [
    TypeOrmModule.forRoot(config),
    // Các module khác trong ứng dụng
  ],
})

```

*Hình 2: Config TypeORM ở module gốc*

## 2.3 Giới thiệu về Typescript :

### 2.3.1 Tổng quan về Typescript :

TypeScript là một phiên bản cao hơn của JavaScript, được thiết kế để xây dựng các ứng dụng lớn và phức tạp. Nó kế thừa nhiều khái niệm từ Java và C#, TypeScript là ngôn ngữ tĩnh (Static typed) có nghĩa là nó nghiêm ngặt và có trật tự trái ngược với free-type. Nó còn được bổ sung thêm lớp hướng đối tượng mà điều này không có ở Javascript.

Typescript là một dự án kéo dài hơn 3 năm của Microsoft nhằm tạo ra một ngôn ngữ để mở rộng JavaScript. Giúp nó trở nên phù hợp hơn với những ứng dụng lớn, nhưng vẫn

quen thuộc với cấu trúc ngôn ngữ JS hiện tại để mọi người có thể học nhanh hơn. Trưởng nhóm dự án này là Anders Hejlsberg, cha đẻ của C#, Turbo Pascal và Delphi.

### **2.3.2 Lợi ích của việc sử dụng Typescript :**

- Dễ dàng hơn trong phát triển các dự án lớn, được hỗ trợ bởi các Javascript Framework lớn.
- Hầu hết các cú pháp hướng đối tượng đều được hỗ trợ bởi Typescript như kế thừa, đóng gói, constructor, abstract, interface, implement, override...
- Cách tổ chức code rõ ràng hơn, hỗ trợ cơ chế giúp kiến trúc hệ thống code hướng module, hỗ trợ namespace, giúp xây dựng các hệ thống lớn nơi mà nhiều lập trình viên có thể làm việc cùng nhau một cách dễ dàng hơn.
- Một lợi thế của Typescript nữa là mã nguồn mở vì vậy nó miễn phí và có cộng đồng hỗ trợ rất lớn.
- Typescript đang được sử dụng ở các Framework phổ biến như Angular, Nodejs, ReactJS, NestJS, React Native,...

### **2.4 Tổng quan về NestJS :**

NestJS là một NodeJS framework dùng để phát triển server-side applications hiệu quả và có thể mở rộng. NestJS là sự kết hợp bởi OOP(Object Oriented Programming), FP(Functional Programming), FRP(Functional Reactive Programming). NestJS sử dụng TypeScript để phát triển. Nó bắt buộc các nhà phát triển sử dụng một kiến trúc cụ thể bằng cách giới thiệu các module, provider và controller, đảm bảo ứng dụng highly scalable, testable và dễ dàng maintaince. Nest rất khắt khe và chặt chẽ trong việc xây dựng cấu trúc project. Với một nhà phát triển mới làm việc với NodeJS thì họ sẽ không có tư duy về architecture, scable hay maintainable thì trong quá trình phát triển có thể gây ra việc tạo ra kiến trúc code không chất lượng làm xáo trộn dự án. NestJS cung cấp cho chúng ta một kiến trúc cụ thể và rõ ràng để giải quyết vấn đề này.

### **2.5 Tổng quan về TypeORM:**

TypeORM là một ORM (Object-Relational Mapping) cho TypeScript và JavaScript, cho phép tương tác với cơ sở dữ liệu quan hệ bằng cách sử dụng các đối tượng TypeScript thay vì viết các câu lệnh SQL trực tiếp.

TypeORM hỗ trợ nhiều loại cơ sở dữ liệu, bao gồm PostgreSQL, MySQL, MariaDB, SQLite, Microsoft SQL Server và Oracle. Nó cung cấp nhiều tính năng như khả năng tạo bảng, tạo chỉ mục, tạo ràng buộc, quản lý cập nhật cơ sở dữ liệu và tìm kiếm.

TypeORM cũng cung cấp hỗ trợ cho các tính năng như kết nối đa cơ sở dữ liệu, quản lý phiên bản và các dịch vụ đám mây như Amazon RDS, Google Cloud SQL và Microsoft Azure SQL.

TypeORM có cú pháp đơn giản và rõ ràng, cho phép bạn định nghĩa các đối tượng và các mối quan hệ giữa chúng dễ dàng. Nó cũng hỗ trợ các tính năng như lazy loading và eager loading, giúp tối ưu hóa hiệu suất truy vấn cơ sở dữ liệu.

Tóm lại, TypeORM là một ORM rất mạnh mẽ và linh hoạt cho TypeScript và JavaScript, cho phép bạn cung cấp một cách tốt nhất để tương tác với cơ sở dữ liệu quan hệ.

## 2.6 Giới thiệu về React JS

ReactJS là một opensource được phát triển bởi Facebook, bản thân nó là một thư viện Javascript được dùng để xây dựng các tương tác với các thành phần trên website. Một trong những điểm nổi bật nhất của ReactJS đó là việc render dữ liệu không chỉ thực hiện được trên tầng Server mà còn ở dưới Client nữa. Nó cho phép ta xây dựng các ứng dụng web đơn trang (Single Page Applications - SPA) với hiệu suất cao, dễ bảo trì và tái sử dụng code.

### ❖ Một số thư viện, package khác:

- **React router dom(V6)**: là một thư viện định tuyến (routing) tiêu chuẩn trong React. Nó giữ cho giao diện của ứng dụng đồng bộ với URL trên trình duyệt. React-Router cho phép bạn định tuyến "luồng dữ liệu" (data flow) trong ứng dụng của bạn một cách rõ ràng. Nó tương đương với sự khảng định, nếu bạn có URL này, nó sẽ tương đương với Route này, và giao diện tương ứng.
- **Redux**: là một predictable state management tool cho các ứng dụng Javascript. Nó giúp bạn viết các ứng dụng hoạt động một cách nhất quán, chạy trong các môi trường khác nhau (client, server, and native) và dễ dàng để test. Redux ra đời lấy cảm hứng từ tư tưởng của ngôn ngữ Elm và kiến trúc Flux của Facebook. Do vậy Redux thường dùng kết hợp với React.
- **Axios**: Axios là một thư viện HTTP Client dựa trên Promise. Cơ bản thì nó cung cấp một API cho việc xử lý XHR (XMLHttpRequests).
- **AntDesign** : Ant Design là một bộ UI component được phát triển bởi Ant Financial (từng là một phần của Alibaba Group), cung cấp các thành phần giao diện người dùng để xây dựng ứng dụng web và mobile. Ant Design là một trong những bộ UI component phổ biến nhất cho React và được sử dụng rộng rãi trong cộng đồng phát triển web.

- Các style được sử dụng để xây dựng giao diện: SCSS, Style Component, TaiwindCSS.

## 2.7 Giới thiệu về Restful

**REST** là viết tắt của Representational State Transfer. **RESTful API** là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile...) khác nhau giao tiếp với nhau.

Chức năng quan trọng nhất của REST là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE...) và cách định dạng các URL cho ứng dụng web để quản các resource. RESTful không quy định logic code ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng, bất kỳ ngôn ngữ hoặc framework nào cũng có thể sử dụng để thiết kế một RESTful API.

## 2.8 Tìm kiếm dựa trên mô hình Boolean

Mô hình Boolean trong truy vấn ngôn ngữ tự nhiên là một phương pháp truy vấn cơ bản trong các hệ thống tìm kiếm thông tin và cơ sở dữ liệu. Mô hình này sử dụng các toán tử logic Boolean (AND, OR, NOT) để xác định các tài liệu phù hợp với câu hỏi được đặt ra.

Tìm kiếm boolean kèm trọng số cho mỗi tiêu chí là một phương pháp tìm kiếm thông tin trong các hệ thống tìm kiếm văn bản, trong đó mỗi tiêu chí tìm kiếm được gán một trọng số để xác định độ quan trọng của nó trong kết quả tìm kiếm.

Phương pháp tìm kiếm này thường được sử dụng trong các hệ thống tìm kiếm văn bản để cải thiện độ chính xác của kết quả tìm kiếm. Thay vì sử dụng các phép toán boolean đơn giản như AND, OR và NOT để kết hợp các tiêu chí tìm kiếm, mỗi tiêu chí tìm kiếm được gán một trọng số để xác định độ quan trọng của nó trong kết quả tìm kiếm.

Cụ thể, khi thực hiện tìm kiếm theo phương pháp này, mỗi tiêu chí tìm kiếm sẽ được gán một trọng số. Trọng số này thường được xác định dựa trên độ quan trọng của tiêu chí tìm kiếm đối với người dùng hoặc dựa trên sự xuất hiện của từ khóa trong tài liệu. Sau đó, hệ thống tìm kiếm sẽ tính điểm cho mỗi tài liệu dựa trên số lượng và trọng số của các tiêu chí tìm kiếm mà tài liệu chứa.

Ví dụ, trong trang web tìm kiếm và đặt sách, khách có thể tìm kiếm sách theo các tiêu chí tên sách, tác giả, thể loại, nhà xuất bản. Ta sẽ gán trọng số cho các tiêu chí dựa theo độ quan trọng, tiêu chí càng quan trọng thì trọng số càng cao. Ví dụ ta gán trọng số cao cho tên sách và thể loại vì đó là hai tiêu chí mà khách hàng quan tâm. Kết quả trả về sẽ được sắp xếp theo tổng số của các tiêu chí từ cao xuống thấp.

## 2.9 MapBox

Mapbox là một công ty cung cấp các dịch vụ bản đồ trực tuyến và các công cụ để phát triển các ứng dụng liên quan đến địa lý. Mapbox cung cấp cho các nhà phát triển các API, SDK và các công cụ để tạo ra các bản đồ tùy chỉnh. Mapbox được sử dụng rộng rãi trong các ứng dụng web và mobile, bao gồm cả các ứng dụng về điều hướng, du lịch, giáo dục, và các ứng dụng địa lý khác. Các ứng dụng sử dụng Mapbox có thể sử dụng các tính năng như định vị GPS, hiển thị thông tin địa lý, tìm kiếm địa điểm, và hướng dẫn điều hướng. Ta sử dụng API của MapBox để tính được khoảng cách giao hàng từ đó ta tính được chi phí để vận chuyển.

# CHƯƠNG 3: PHÂN TÍCH HỆ THỐNG

## 3.1 Bối cảnh/ hiện trạng của hệ thống

### 3.1.1 Bài toán thực tế

Nhà sách Thành Nghĩa hiện tại có dịch vụ cho phép khách hàng đặt mua sách mà không cần đến cửa hàng thông qua việc khách hàng nhắn tin hoặc gọi điện cho nhà sách và cung cấp thông tin cho nhân viên nhà sách. Khách hàng cung cấp các thông tin về các đầu sách muốn mua, số lượng và địa chỉ giao hàng.

Sau khi nhận được thông tin về vị trí giao hàng, nhân viên sẽ kiểm tra xem khoảng cách từ nhà sách đến nơi đặt là bao nhiêu km(kilomet) :

- Nếu như quá 35 km thì nhà sách sẽ không giao hàng. Nhân viên gọi lại và thông báo cho khách hàng để khách.
- Nếu như trong phạm vi 35 km thì nhà sách sẽ trực tiếp đi giao. Phí giao hàng được tính theo công thức theo hình dưới đây :

Khoảng cách	Phí giao hàng	Thêm 0.5kg
0km-15km	20 nghìn đồng nếu tổng giá sách dưới 300 nghìn đồng. Miễn phí nếu tổng giá sách giá trị từ 300 nghìn đồng trở lên và trọng lượng đơn hàng dưới 0.6kg	Thêm 3 nghìn đồng
16km-25km	30 nghìn đồng nếu như nếu tổng giá sách dưới 600 nghìn đồng. Miễn phí nếu nếu tổng giá sách có giá trị từ 600 nghìn đồng trở lên và trọng lượng đơn hàng dưới 1.1kg	Thêm 5 nghìn đồng
25km-35km	40 nghìn đồng nếu tổng giá sách dưới 1 triệu đồng. Miễn phí nếu nếu tổng giá sách có giá trị từ 1 triệu đồng trở lên và trọng lượng đơn hàng dưới 1.6kg	Thêm 7 nghìn đồng

Bảng 1: Bảng công thức tính phí giao hàng

Nếu nhu thoả mãn khoảng cách giao hàng dưới 35 km, nhân viên sẽ kiểm tra về đầu sách mà khách hàng muốn mua trong bảng excel của nhà sách:

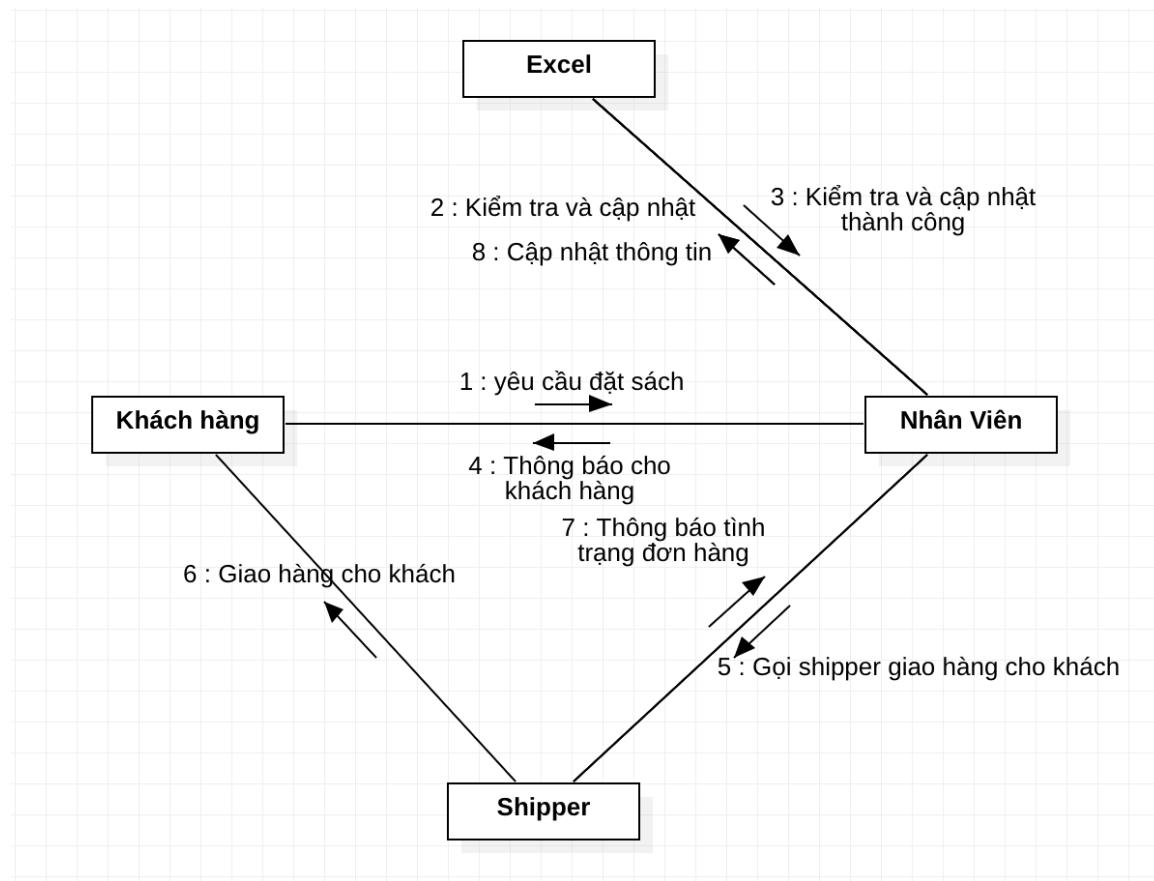
- Trường hợp đầu sách đó có và đủ số lượng, nhân viên kiểm tra loại khách hàng (thân thiết, vip, ...) để áp dụng các khuyến mãi. Nhân viên sẽ gọi lại cho khách hàng để thông báo hoá đơn và đặt cọc. Sau đó nhân viên sẽ cập nhật lại số lượng đầu sách đó ở excel và làm hoá đơn cho đơn hàng của khách hàng đặt sách. Lưu đơn hàng của khách đó đã cọc bao nhiêu hay đã thanh toán hết vào excel. Sau đó, gọi cho nhân viên giao hàng cho khách.
- Trường hợp đầu sách đó không tồn tại hoặc số lượng không đủ cho khách hàng, nhân viên sẽ gọi thông báo cho khách hàng.

Sau khi khách nhận hàng, nhân viên vào excel cập nhật lại đơn hàng đó là đã thanh toán.

### **3.1.2 Hiện trạng**

Hiện tại nhà sách đang quản lý dịch vụ dưới dạng quản lý trực tiếp, việc quản lý thông qua excel. Tuy nhiên hình thức này có những hạn chế như khó khăn trong việc quản lý và truy xuất thông tin, không đảm bảo tính linh hoạt, khó khăn trong việc quản lý, theo dõi đơn hàng, lợi nhuận. Và rất nhiều thao tác thủ công của nhân viên có thể gây ra các sai sót ảnh hưởng tới nhà sách.

### 3.1.2.1 Biểu đồ cộng tác hiện trạng trước khi sử dụng phần mềm

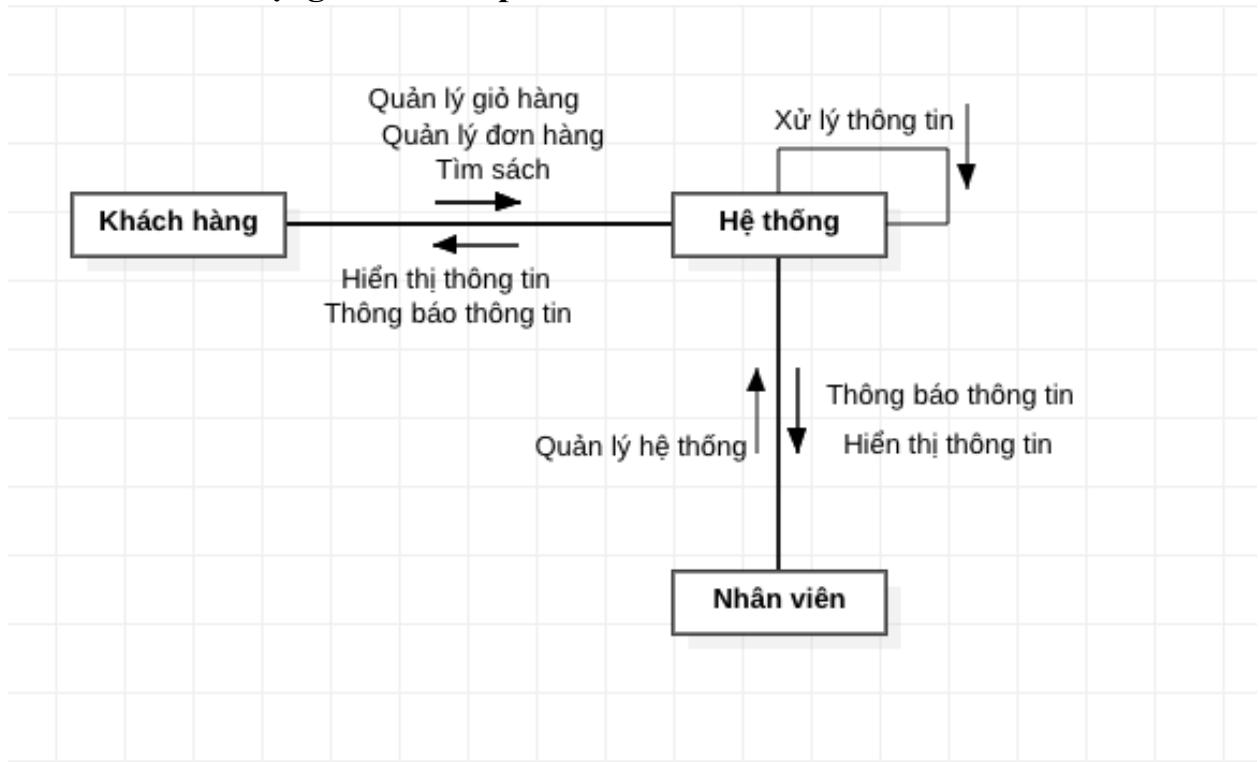


Hình 3: Biểu đồ cộng tác khi chưa có phần mềm

### 3.1.3 Đề xuất

Xây dựng một website tiện dụng, thân thiện với người dùng để khách hàng có thể tiết kiệm thời gian trong quá trình tìm kiếm và đặt mua. Thay vì phải gọi cho nhân viên cung cấp rất nhiều thông tin và phải đợi nhân viên tính toán tìm kiếm và gọi lại cho mình, thì với website này người dùng sau khi tìm kiếm chọn các đầu sách mình mong muốn thì hệ thống sẽ tự động tính toán phí giao hàng cho người dùng mà không cần phải chờ đợi. Đồng thời tránh sai sót đến từ nhân viên vì nó là tự động hoàn toàn.

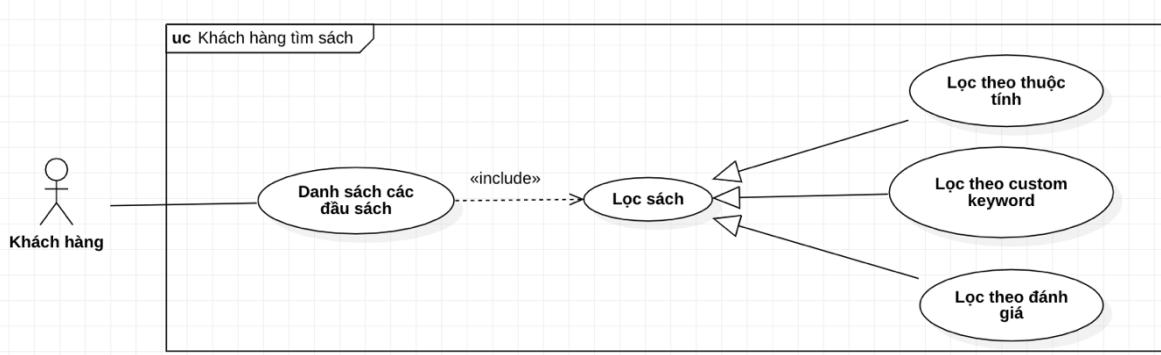
### 3.1.3.1 Biểu đồ cộng tác khi có phần mềm



Hình 4: Biểu đồ cộng tác sau khi có phần mềm

## 3.2 Lược đồ Usecase cho phân tích

### 3.2.1 Usecase khách hàng tìm sách



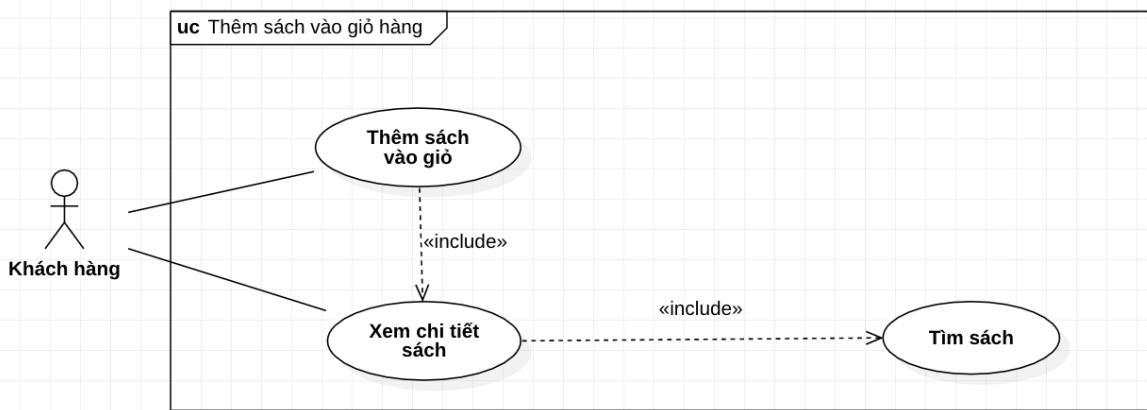
Hình 5: Usecase khách hàng tìm sách

USECASE NAME	Khách hàng tìm sách
--------------	---------------------

<b>USECASE ID</b>	UC-01
<b>SCENARIO</b>	Khách hàng tìm sách trên website.
<b>ACTOR</b>	Khách hàng.
<b>DESCRIPTION</b>	Khi khách hàng muốn tìm được đầu sách mà họ mong muốn để mua trên website.
<b>TRIGGER</b>	Khách hàng tìm kiếm sách.
<b>PRECONDITION</b>	Khách hàng nhập thông tin vào bộ lọc của hệ thống.
<b>POSTCONDITION</b>	Khách hàng tìm được đầu sách mong muốn từ website.
<b>MAIN FLOW</b>	<ol style="list-style-type: none"> <li>Khách hàng nhấn vào thanh tìm kiếm và nhập từ khoá mà mình muốn tìm.</li> <li>Hệ thống hiển thị các tùy chọn tìm kiếm theo thuộc tính của sách, custom key hoặc theo lượt bán.</li> <li>Hệ thống kiểm tra thông tin và lọc theo thông tin mà người dùng cung cấp.</li> <li>Hệ thống chuyển đến giao diện những đầu sách thỏa mãn từ bộ lọc của khách hàng.</li> </ol>
<b>EXCEPTIONS</b>	1a. Người dùng không nhập thông tin mà bấm lọc, hệ thống sẽ hiển thị thông báo yêu cầu người dùng nhập thông tin. Quay lại [Main Flow 1]

Bảng 2 : Đặc tả Usecase khách hàng tìm sách

### 3.2.2 Usecase thêm sách vào giỏ hàng

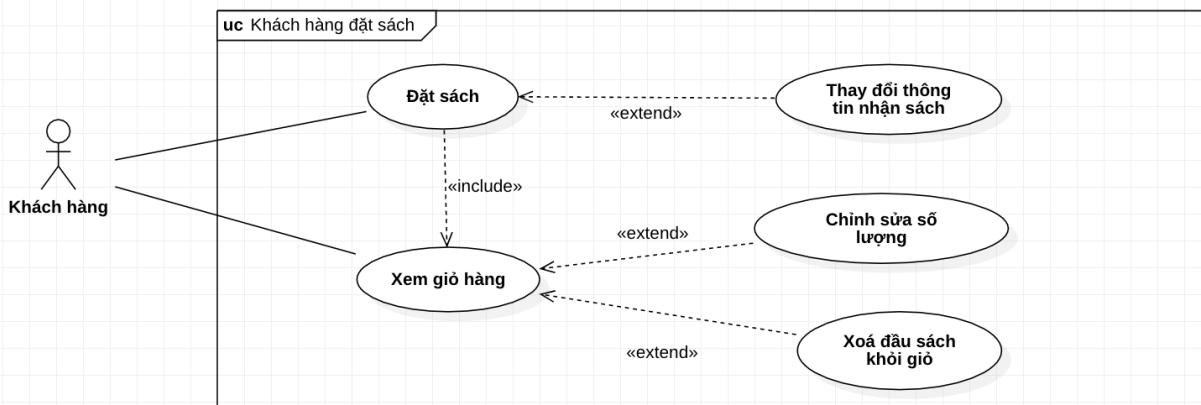


Hình 6: Usecase thêm sách vào giỏ hàng

<b>USECASE NAME</b>	Thêm sách vào giỎ hàng.
<b>USECASE ID</b>	UC-02
<b>SCENARIO</b>	Khách hàng thêm sách vào giỏ hàng trên website.
<b>ACTOR</b>	Khách hàng.
<b>DESCRIPTION</b>	Khi khách hàng tìm được đầu sách ưng ý, khách hàng thêm sách vào giỏ hàng.
<b>TRIGGER</b>	Khách hàng thực hiện chức năng thêm sách vào giỏ hàng.
<b>PRECONDITION</b>	Khách hàng đăng nhập vào hệ thống.
<b>POSTCONDITION</b>	Thêm đầu sách vào giỏ hàng cho khách hàng.
<b>MAIN FLOW</b>	<ol style="list-style-type: none"> <li>1. Khách hàng tìm kiếm sách (UC-01).</li> <li>2. Khách hàng bấm vào đầu sách để xem chi tiết thông tin từ những đầu sách đã lọc được từ UC-01.</li> <li>3. Hệ thống chuyển đến trang chi tiết cho khách hàng.</li> <li>4. Khách hàng chọn số lượng và bấm thêm vào giỏ hàng.</li> <li>5. Hệ thống kiểm tra thông tin về số lượng có hợp lệ hay không và thông báo cho khách hàng.</li> <li>6. Thêm đầu sách vào giỏ hàng cho khách hàng.</li> </ol>
<b>EXCEPTIONS</b>	<p>5a. Khi người dùng chưa đăng nhập thì hệ thống yêu cầu đăng nhập trước.      6a. Đăng nhập vào hệ thống      Quay lại [Main Flow 5].</p> <p>5b. Khi người dùng chọn số lượng mà số lượng đó không đủ vì đã có người đặt trước. Hệ thống hiển thị thông báo và cập nhật giao diện hiển thị số lượng hiện tại. Quay lại [Main Flow 4].</p>

Bảng 3: Đặc tả Usecase thêm sách vào giỏ hàng

### 3.2.3 Usecase khách hàng đặt sách



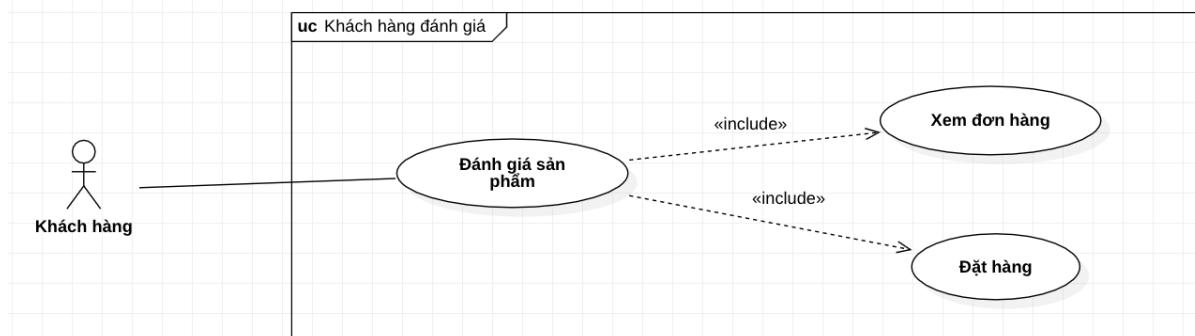
Hình 7: Usecase khách hàng đặt sách

<b>USECASE NAME</b>	Khách hàng đặt sách
<b>USECASE ID</b>	UC-03
<b>SCENARIO</b>	Khách hàng đặt sách trên website.
<b>ACTOR</b>	Khách hàng.
<b>DESCRIPTION</b>	Khi khách hàng muốn đặt sách qua website, khách hàng chọn sách cần mua và tiền hàng gửi yêu cầu đặt hàng.
<b>TRIGGER</b>	Khách hàng thực hiện gửi yêu cầu đặt hàng.
<b>PRECONDITION</b>	Khách hàng đăng nhập vào hệ thống.
<b>POSTCONDITION</b>	Thêm hóa đơn cho khách hàng với trạng thái là chờ duyệt.
<b>MAIN FLOW</b>	<ol style="list-style-type: none"> <li>Khách hàng bấm vào giỏ hàng, xem thông tin giỏ hàng.</li> <li>Hệ thống hiển thị giao diện giỏ hàng với từng đầu sách và số lượng.</li> <li>Khách hàng xác nhận số lượng và xác nhận nơi giao hàng.</li> <li>Hệ thống hiển thị số tiền mà khách hàng phải trả.</li> <li>Khách hàng bấm nút xác nhận đặt hàng.</li> <li>Hệ thống kiểm tra và thông báo về cho khách hàng.</li> </ol>
<b>EXCEPTIONS</b>	1a. Khách hàng chưa đăng nhập, hệ thống yêu cầu đăng nhập. Tiếp tục [Main Flow 1].

	6a. Khi số lượng không đủ vì đã có người đặt trước. Hệ thống hiển thị thông báo và cập nhật giao diện hiển thị số lượng hiện tại. Quay lại [Main Flow 3].
--	---

Bảng 4: Đặc tả Usecase khách hàng đặt sách

### 3.2.4 Khách hàng đánh giá sản phẩm

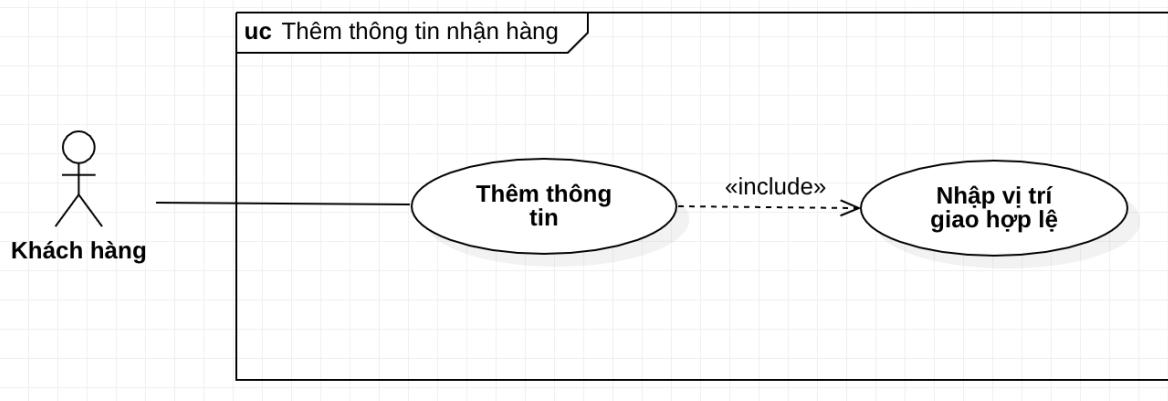


Hình 8: Usecase khách hàng đánh giá sản phẩm

<b>USECASE NAME</b>	Khách hàng đánh giá sản phẩm
<b>USECASE ID</b>	UC-04
<b>SCENARIO</b>	Khách hàng đánh giá sản phẩm trên website.
<b>ACTOR</b>	Khách hàng.
<b>DESCRIPTION</b>	Khi khách hàng muốn đánh giá những cuốn sách trong đơn hàng mà mình đặt.
<b>TRIGGER</b>	Khách hàng thực thi chức năng đánh giá đơn hàng.
<b>PRECONDITION</b>	1. Khách hàng đã đăng nhập vào hệ thống. 2. Đơn hàng đó đang ở trạng thái đã giao.
<b>POSTCONDITION</b>	Đơn hàng được khách đánh giá thành công.
<b>MAIN FLOW</b>	1. Khách hàng bấm vào mục đơn hàng của mình trên giao diện. 2. Hệ thống hiển thị giao diện các đơn hàng đã đặt của người dùng. 3. Khách hàng chọn đơn hàng và đầu sách muốn đánh giá. Chỉ những đơn hàng mà trạng thái là đã giao mới có nút đánh giá. 4. Khách hàng nhập bình luận và số sao cho sách trong đơn hàng. 5. Hệ thống thực hiện kiểm tra và thực thi. Sau đó hiển thị thông báo.
<b>EXCEPTIONS</b>	

Bảng 5: Đặc tả Usecase khách hàng đánh giá sản phẩm

### 3.2.5 Khách hàng thêm thông tin nhận hàng.

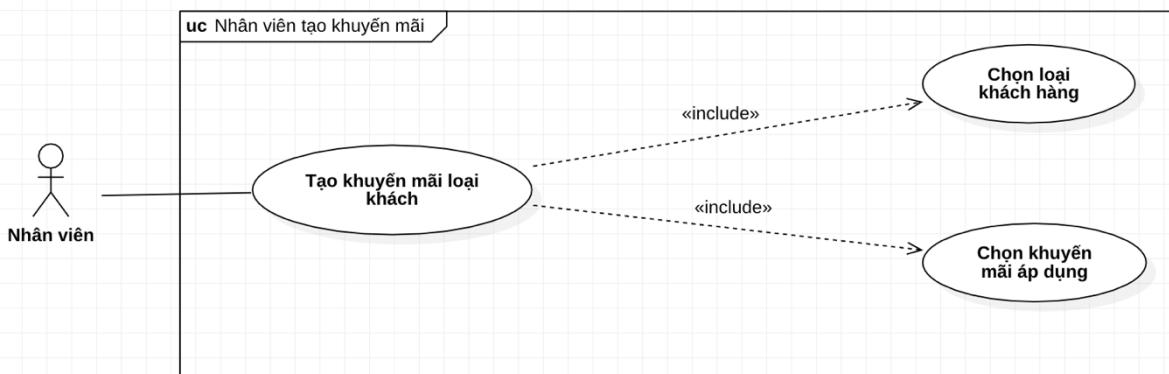


Hình 9: Usecase khách hàng thêm thông tin nhận hàng

<b>USECASE NAME</b>	Thêm thông tin nhận hàng
<b>USECASE ID</b>	UC-05
<b>SCENARIO</b>	Khách hàng thêm vị trí nhận hàng trên website.
<b>ACTOR</b>	Khách hàng.
<b>DESCRIPTION</b>	Khi khách hàng muốn thêm một vị trí khác để nhận hàng.
<b>TRIGGER</b>	Khách hàng thực thi chức năng thêm thông tin nhận hàng.
<b>PRECONDITION</b>	Khách hàng đã đăng nhập vào hệ thống.
<b>POSTCONDITION</b>	Thêm vị trí giao hàng mới thành công.
<b>MAIN FLOW</b>	<ol style="list-style-type: none"> <li>Khách hàng vào mục thông tin cá nhân.</li> <li>Hệ thống chuyển đến trang có giao diện là thông tin của khách hàng bao gồm cả thông tin nhận hàng.</li> <li>Khách hàng nhấn vào nút thêm thông tin nhận hàng.</li> <li>Hệ thống mở giao diện nhập thông tin cho khách hàng.</li> <li>Khách hàng nhập đầy đủ thông tin.</li> <li>Hệ thống kiểm tra thông tin mà khách hàng nhập. Nếu hợp lệ lưu vào cơ sở dữ liệu và thông báo cho khách hàng.</li> </ol>
<b>EXCEPTIONS</b>	<ol style="list-style-type: none"> <li>Nếu như chọn vị trí cách nhà sách hơn 35km thì hệ thống sẽ thông báo lỗi cho khách hàng và yêu cầu chọn giao nơi khác. Quay lại [Main Flow 5]</li> </ol>

Bảng 6: Đặc tả Usecase khách hàng thêm thông tin nhận hàng

### 3.2.6 Nhân viên tạo khuyến mãi

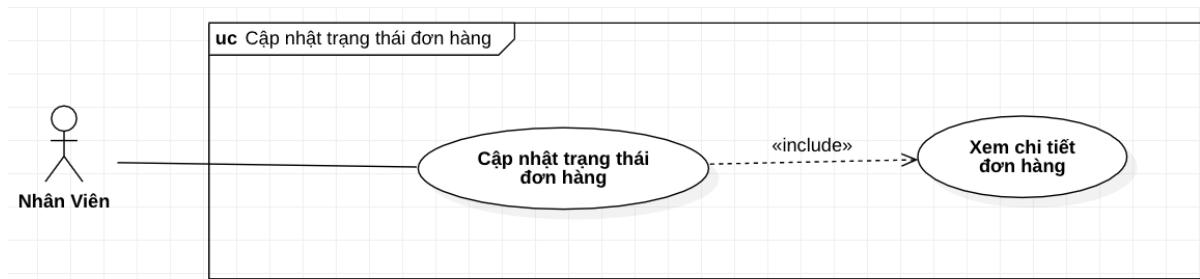


Hình 10: Usecase nhân viên tạo khuyến mãi

<b>USECASE NAME</b>	Nhân viên tạo khuyến mãi.
<b>USECASE ID</b>	UC-06
<b>SCENARIO</b>	Nhân viên tạo khuyến mãi cho khách hàng.
<b>ACTOR</b>	Nhân viên.
<b>DESCRIPTION</b>	Khi nhà sách muốn tạo ra một đợi khuyến mãi giảm giá cho khách hàng theo loại khách hàng.
<b>TRIGGER</b>	Nhân viên thực thi chức năng tạo khuyến mãi trên hệ thống.
<b>PRECONDITION</b>	Nhân viên đăng nhập vào hệ thống thành công.
<b>POSTCONDITION</b>	Thêm khuyến mãi thành công.
<b>MAIN FLOW</b>	<ol style="list-style-type: none"> <li>Nhân viên chọn vào mục quản lý khuyến mãi và chọn thêm khuyến mãi khách hàng.</li> <li>Hệ thống hiển thị giao diện thêm khuyến mãi tạo theo loại khách hàng.</li> <li>Nhân viên nhập đầy đủ thông tin và bấm vào nút tạo khuyến mãi.</li> <li>Hệ thống kiểm tra tính hợp lệ của thông tin mà nhân viên nhập, sau đó lưu khuyến mãi vào cơ sở dữ liệu. Hiển thị thông báo cho nhân viên.</li> </ol>
<b>EXCEPTIONS</b>	<p>4a. Dữ liệu nhân viên nhập không thỏa mãn ví dụ nhập phần trăm khuyến mãi lớn hơn 100. Ngày bắt đầu khuyến mãi ở trước hiện tại,... Hệ thống thông báo cho nhân viên lỗi sai.  [Quay lại Main Flow 3]</p>

Bảng 7: Đặc tả Usecase nhân viên thêm khuyến mãi

### 3.2.7 Nhân viên cập nhật trạng thái đơn hàng :



Hình 11: Usecase nhân viên cập nhật trạng thái đơn hàng

<b>USECASE NAME</b>	Nhân viên cập nhật trạng thái đơn.
<b>USECASE ID</b>	UC-07
<b>SCENARIO</b>	Nhân viên đổi trạng thái trong trường hợp có đơn hàng mới và đã giao thành công.
<b>ACTOR</b>	Nhân viên.
<b>DESCRIPTION</b>	Có đơn hàng từ khách hàng đặt, nhân viên thay đổi trạng thái của đơn hàng từ chờ duyệt sang đã duyệt. Khi đã giao hàng thì nhân viên cập nhật là đã giao.
<b>TRIGGER</b>	Nhân viên thực thi chức năng cập nhật trạng thái.
<b>PRECONDITION</b>	Nhân viên đăng nhập thành công vào hệ thống.
<b>POSTCONDITION</b>	Đơn hàng được chuyển đổi trạng thái sang đã duyệt hoặc đã giao hoặc đã huỷ.
<b>MAIN FLOW</b>	<ol style="list-style-type: none"> <li>Nhân viên bấm vào mục quản lý đơn hàng.</li> <li>Hệ thống chuyển sang giao diện các đơn hàng hiện có của hệ thống.</li> <li>Nhân viên chọn vào đơn hàng cần thay đổi trạng thái.</li> <li>Hệ thống hiển thị chi tiết đơn hàng đó.</li> <li>Nhân viên đổi trạng thái đơn hàng.</li> <li>Hệ thống thông báo lỗi hoặc thành công cho nhân viên.</li> </ol>
<b>EXCEPTIONS</b>	

Bảng 8: Đặc tả Usecase nhân viên cập nhật trạng thái đơn hàng

## 3.3 Định nghĩa yêu cầu và ràng buộc đối với phần mềm

### 3.3.1 Yêu cầu từ môi trường nghiệp vụ (business)

UseCase	Req-ID	Nội dung yêu cầu	Stack-Holder
UC-01	B01.1	Khách hàng có thể tìm kiếm sách thông qua hệ thống bán sách trực tuyến.	Khách hàng
	B01.2	Khách hàng cần cung cấp thông tin về cuốn sách mong muốn như tên sách, tác giả, thể loại, tag, ...	Khách hàng
	B01.3	Hệ thống cần hiển thị danh sách các đầu sách liên quan đến những tiêu chí mà khách hàng cung cấp.	Chủ nhà sách
UC-02	B02.1	Khách hàng cần cung cấp thông tin về số lượng sách.	Khách hàng
	B02.2	Hệ thống cần hiển thị chi tiết thông tin về đầu sách mà khách hàng đã chọn.	Chủ nhà sách
	B02.3	Hệ thống cần xác nhận là đã thêm thành công vào giỏ hàng chưa và thông báo cho khách.	Chủ nhà sách
UC-03	B03.1	Hệ thống cần giữ sách trong thời gian đặt và đảm bảo tính khả dụng của đầu sách đó cho khách hàng.	Chủ nhà sách.
	B03.2	Khi đặt sách hệ thống cần hiển thị rõ ràng số tiền của sách là bao nhiêu, phí giao hàng là bao nhiêu và khoảng thời gian dự kiến giao hàng.	Chủ nhà sách
UC-04	B04.1	Đánh giá đơn khách hàng cần nhập đánh giá đúng với thực trạng.	Khách hàng.
UC-05	B05.1	Khách hàng cần cung cấp chính xác thông tin giao hàng.	Khách hàng.
	B05.2	Hệ thống cần có chức năng kiểm tra khoảng cách từ nhà sách đến nơi giao hàng.	Chủ nhà sách
UC-06	B06.1	Hệ thống cần chỉ rõ các thông tin khuyến mãi như khuyến mãi cho đối tượng khách hàng nào, đầu sách nào, bao nhiêu phần trăm, ngày bắt đầu, ngày kết thúc.	Chủ nhà sách.
	B06.2	Nhân viên nhập đúng thông tin mà chủ nhà sách đưa ra.	Nhân viên.
UC-07	B07.1	Nhân viên cần xem xét chi tiết đơn hàng có thực tế hay không. Ví dụ như trường hợp khách đặt quá nhiều cần liên hệ với khách để xác thực.	Nhân viên

Bảng 9: Yêu cầu từ môi trường nghiệp vụ (business)

### 3.3.2 Yêu cầu từ môi trường vận hành (operation)

UseCase	Req-ID	Nội dung yêu cầu	Stack-Holder
UC-01	F01.1	Hệ thống cần được thiết kế với giao diện đơn giản và dễ sử dụng.	Chủ nhà sách.
	F01.2	Hệ thống cần phải xác thực và kiểm tra thông tin của sách để đảm bảo tính chính xác.	Chủ nhà sách.

UC-02	F02.1	Hệ thống cần đảm bảo về tính chính xác để khách hàng thêm sách vào giỏ với số lượng hợp lệ	Chủ nhà sách.
UC-03	F03.1	Hệ thống cần hỗ trợ chức năng thay đổi thông tin nhận hàng.	Chủ nhà sách
	F03.2	Hệ thống cung cấp tính năng xem lại đơn hàng, bao gồm trạng thái đơn hàng, thông tin về sách đã đặt, và thông tin vận chuyển.	Chủ nhà sách
UC-04	F04.1	Hệ thống nên cung cấp một quy trình rõ ràng và dễ dàng để khách hàng có thể đánh giá đơn hàng.	Chủ nhà sách
	F04.2	Hệ thống nên gửi thông báo xác nhận đã đánh giá đơn hàng cho khách hàng sau khi quy trình thành công.	Chủ nhà sách
UC-05	F05.1	Hệ thống cần cung cấp một giao diện dễ sử dụng cho khách hàng để thêm thông tin nhận hàng.	Chủ nhà sách
	F05.2	Hệ thống có chức năng kiểm tra khoảng cách vị trí mà khách hàng cung cấp.	Chủ nhà sách
UC-06	F06.1	Hệ thống cần cung cấp một giao diện dễ sử dụng cho nhân viên để thêm khuyến mãi.	Chủ nhà sách
	F06.2	Hệ thống nên có cơ chế để quản lý danh sách khách hàng và loại khách hàng	Chủ nhà sách
	F06.3	Hệ thống nên gửi thông báo xác nhận cho nhân viên sau khi khuyến mãi đã được thêm thành công	Chủ nhà sách
	F06.4	Hệ thống cần lưu trữ thông tin về khuyến mãi để có thể xem xét và phân tích sau này	Chủ nhà sách
UC-07	F07.1	Hệ thống cần cung cấp một giao diện dễ sử dụng cho nhân viên để cập nhật trạng thái đơn hàng.	Chủ nhà sách
	F07.2	Thông tin trạng thái đơn hàng được nhân viên cập nhật nên được xử lý bởi hệ thống.	Chủ nhà sách
	F07.3	Hệ thống cần lưu trữ thông tin về trạng thái đơn hàng để có thể xem xét và phân tích sau này.	Chủ nhà sách

Bảng 10: Yêu cầu từ môi trường vận hành

### 3.3.2 Yêu cầu từ môi trường phát triển (development)

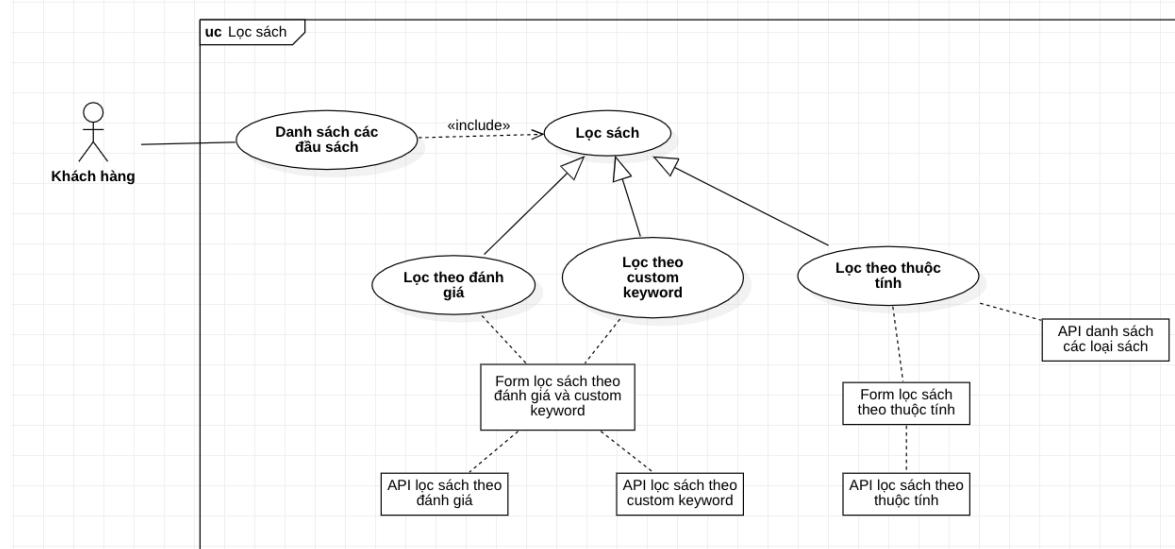
ID	Đối tượng	Nội dung yêu cầu	Stack-Holder
TA01	Application	Thiết bị của khách hàng có kết nối mạng.	Khách hàng
TD01	Software	Viết code trên phần mềm Visual Studio Code.	Dev team
TD02	Software	Dùng NestJs, Entity TypeORM, ReactJS, MySql.	Dev team

*Bảng 11: Yêu cầu từ môi trường phát triển*

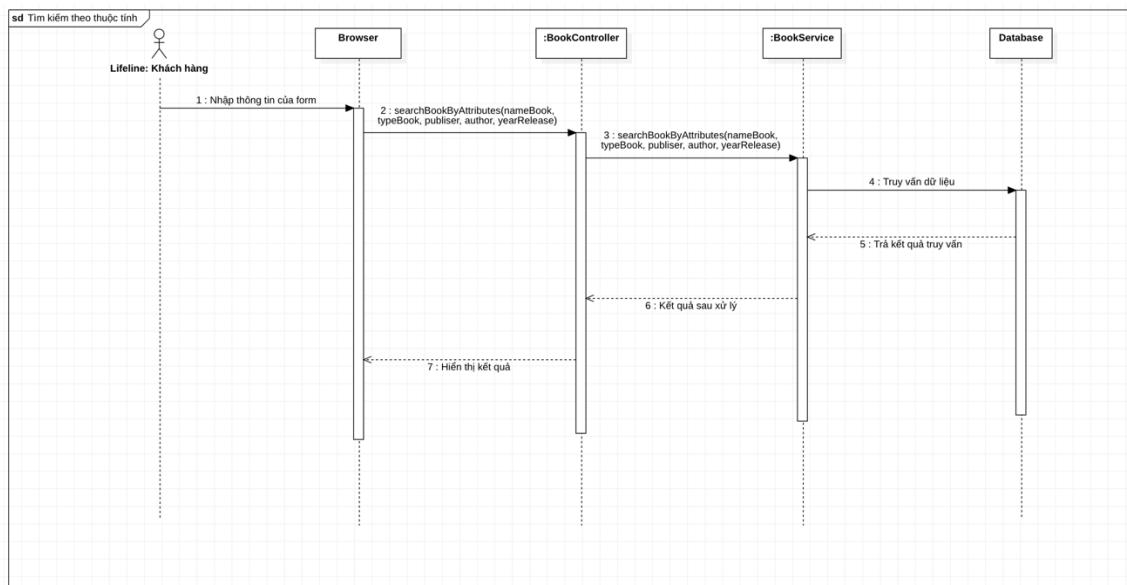
# CHƯƠNG 4: THIẾT KẾ HỆ THỐNG

## 4.1. Lược đồ usecase cho thiết kế :

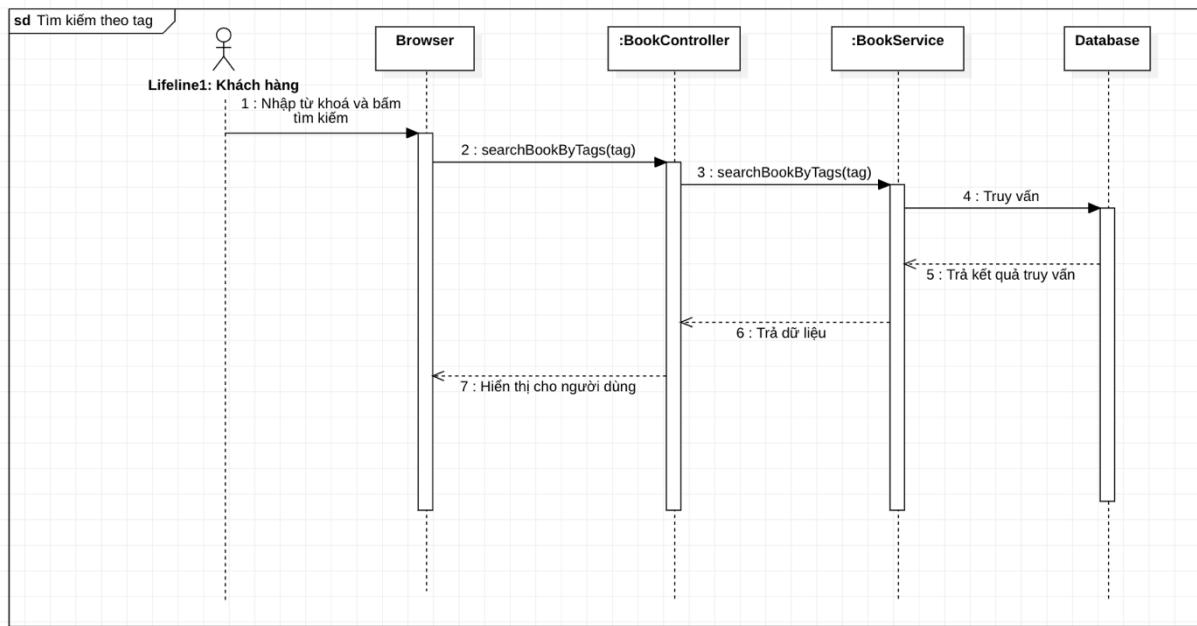
### 4.1.1 Usecase khách hàng tìm sách :



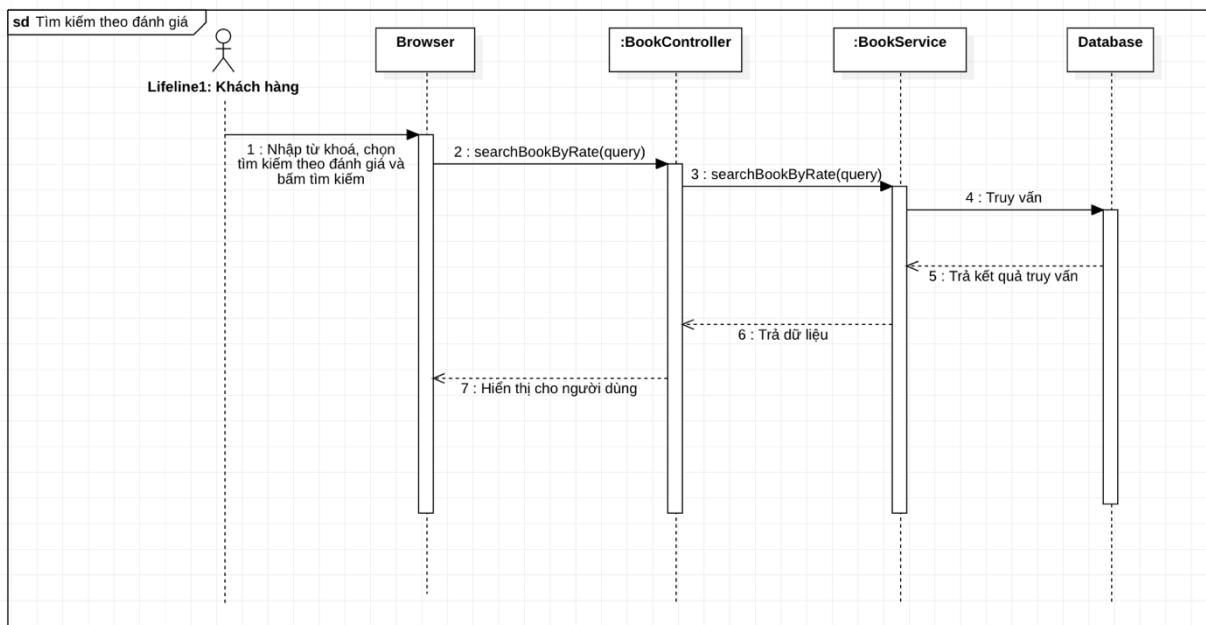
Hình 12: Usecase khách hàng tìm sách



Hình 13: Lược đồ tuần tự cho Usecase tìm kiếm theo thuộc tính

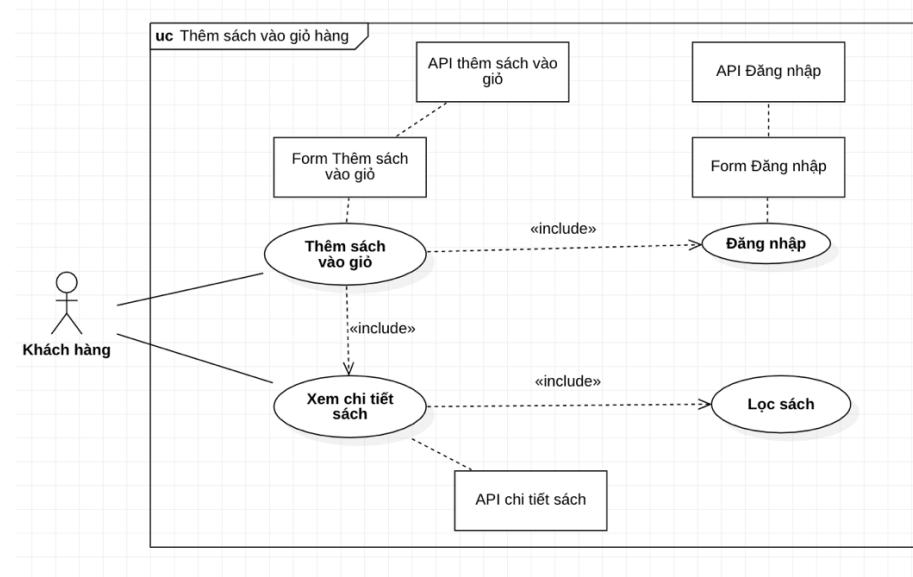


Hình 14: Lược đồ tuần tự cho Usecase tìm kiếm theo tag

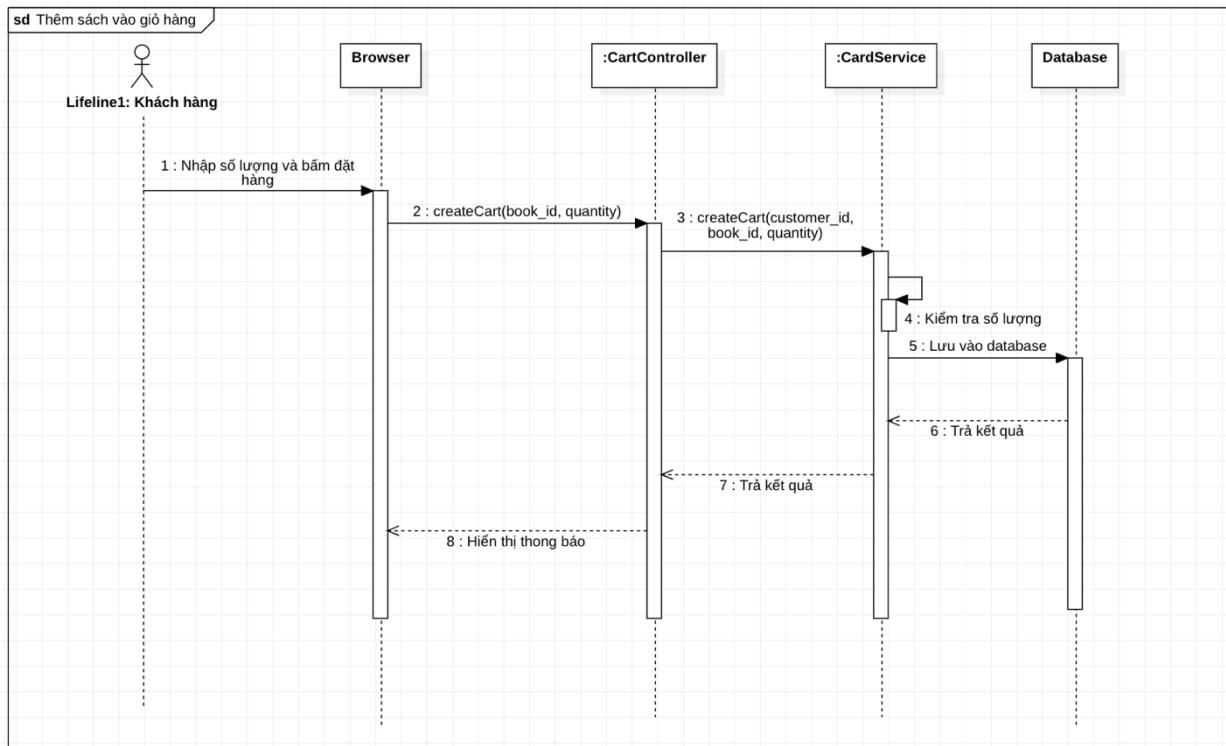


Hình 15: Lược đồ tuần tự cho Usecase tìm kiếm theo đánh giá

#### 4.1.2 Usecase khách hàng thêm sách vào giỏ hàng :

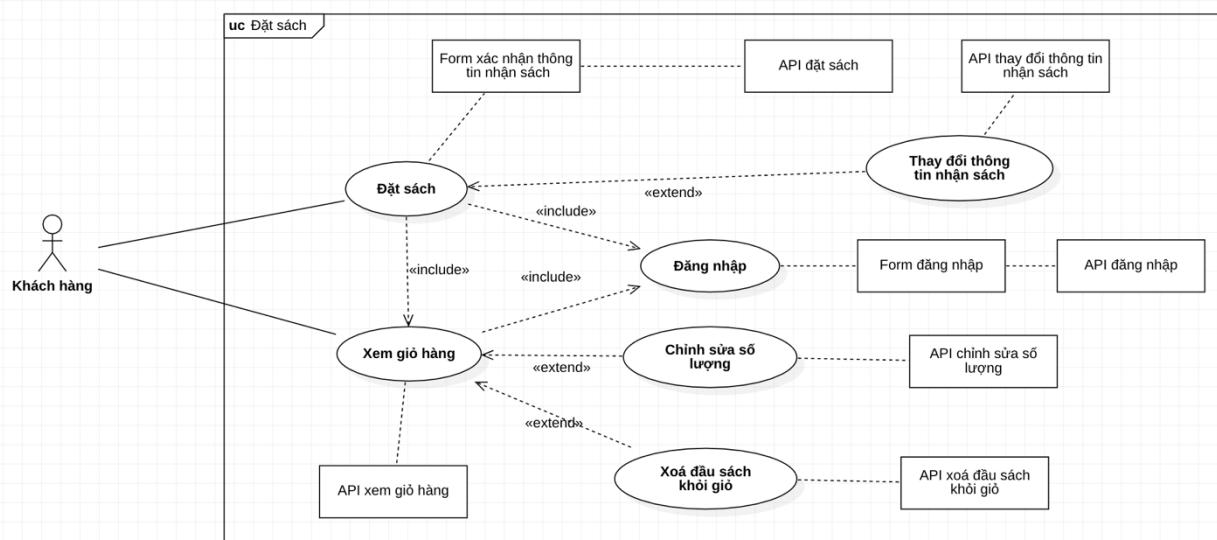


Hình 16: Usecase thêm sách vào giỏ hàng

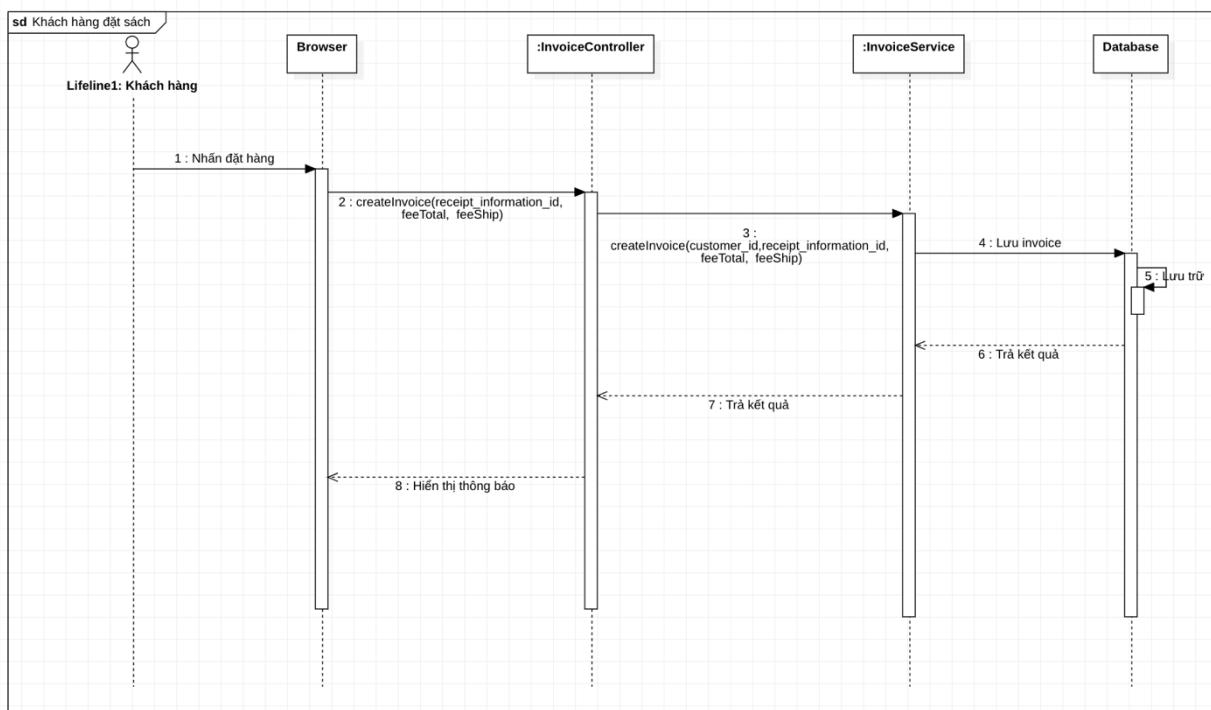


Hình 17: Lược đồ tuần tự cho Usecase thêm sách vào giỏ hàng

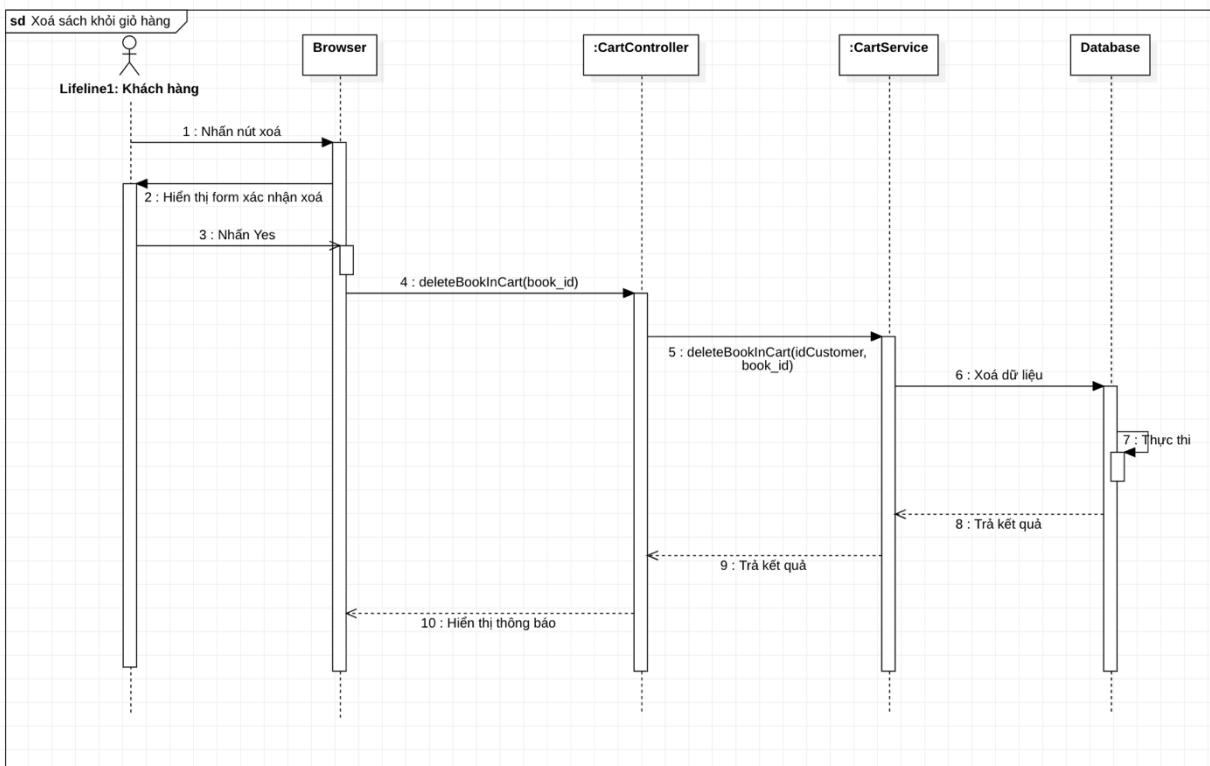
#### 4.1.3 Usecase khách hàng đặt sách :



Hình 18: Usecase khách hàng đặt sách

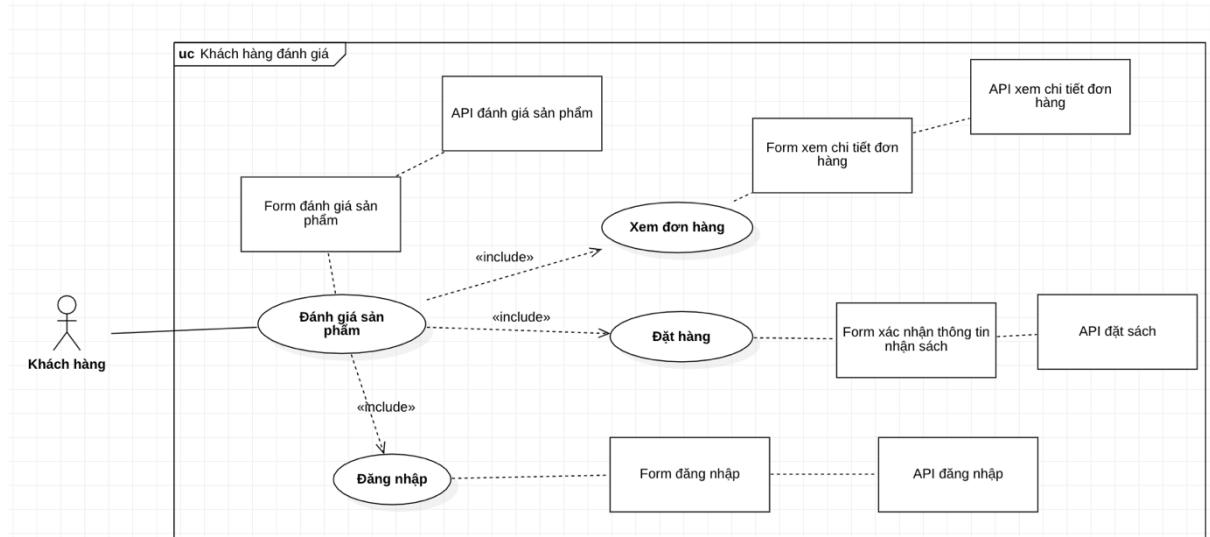


Hình 19: Lược đồ tuần tự cho Usecase khách hàng đặt sách

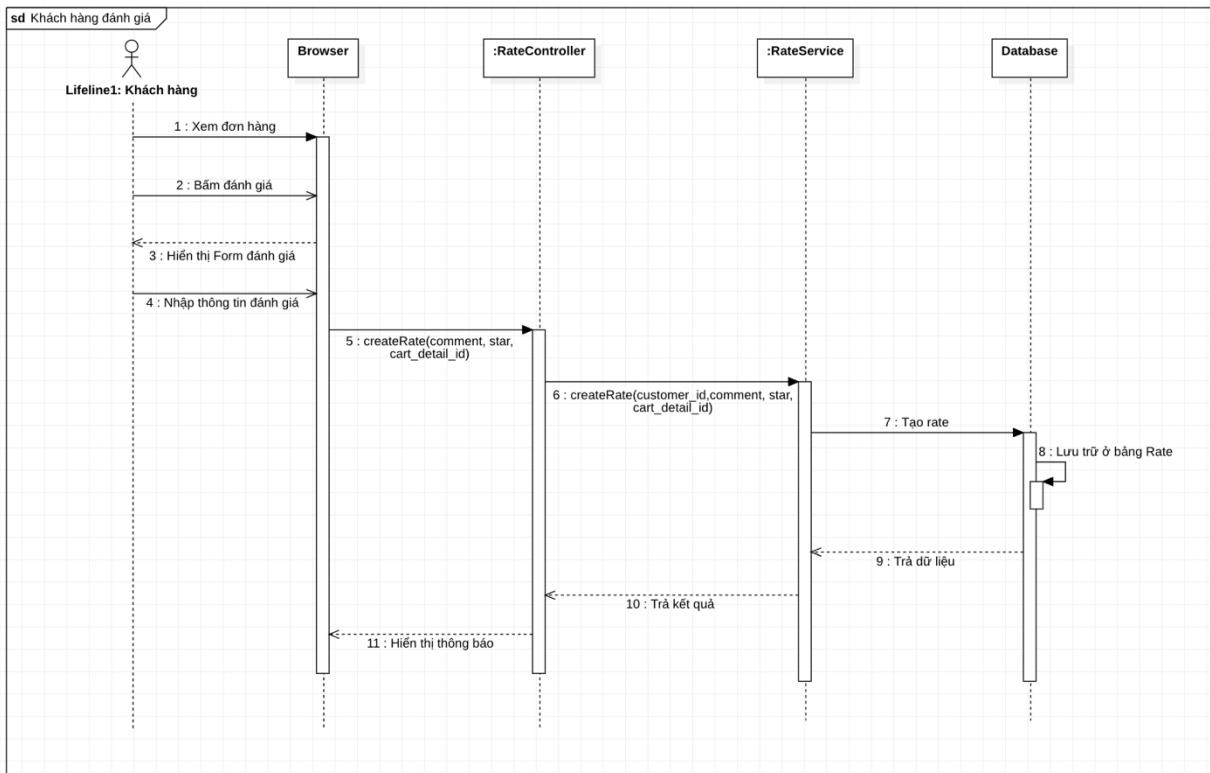


Hình 20: Lược đồ tuần tự cho Usecase xoá sách khỏi giỏ hàng

#### 4.1.4 Usecase khách hàng đánh giá sản phẩm :

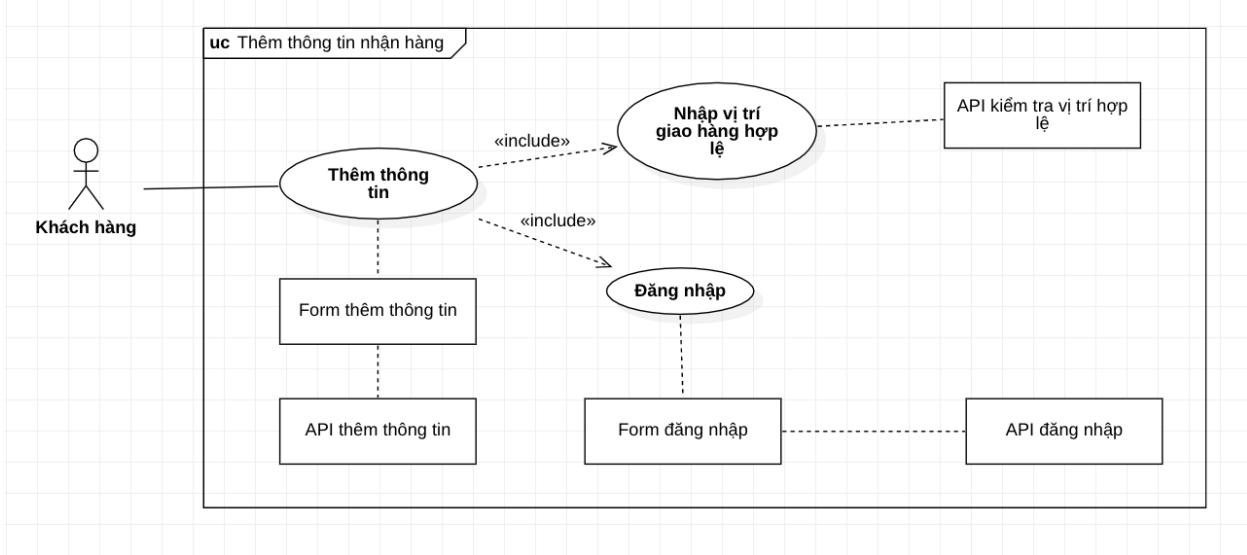


Hình 21: Usecase khách hàng đánh giá sản phẩm

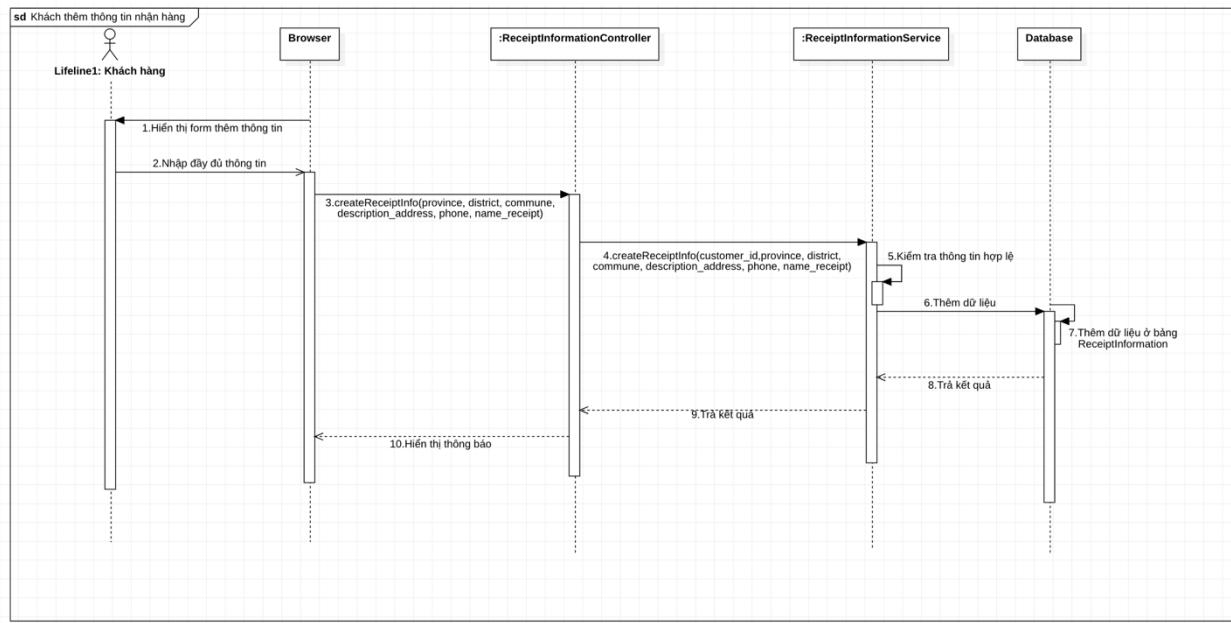


Hình 22: Lược đồ tuần tự cho Usecase đánh giá sản phẩm

#### 4.1.5 Usecase khách hàng thêm thông tin nhận hàng :

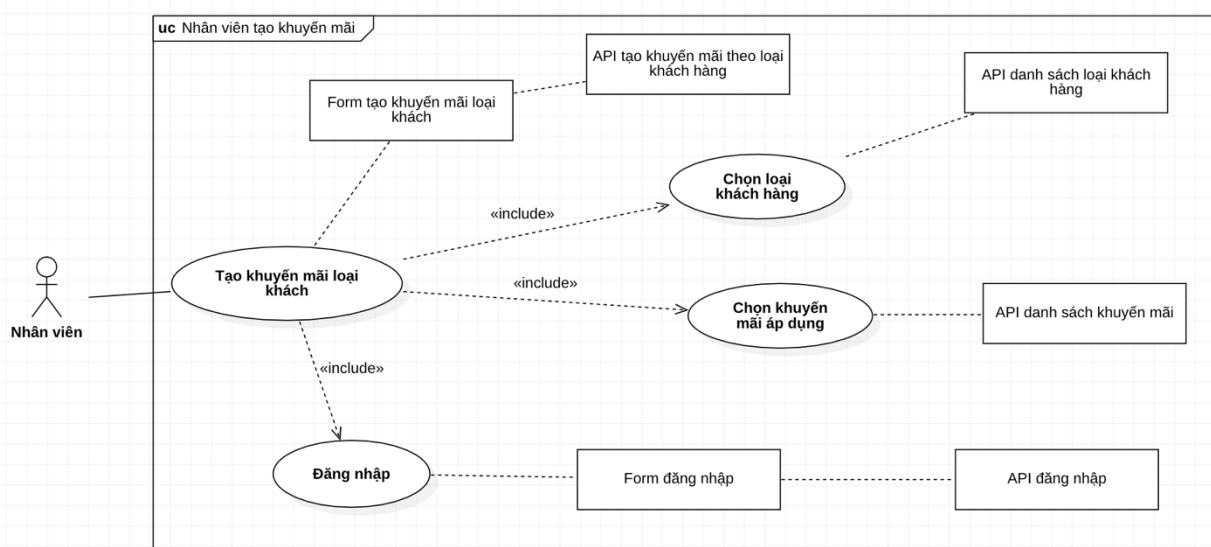


Hình 23: Usecase khách thêm thông tin nhận hàng

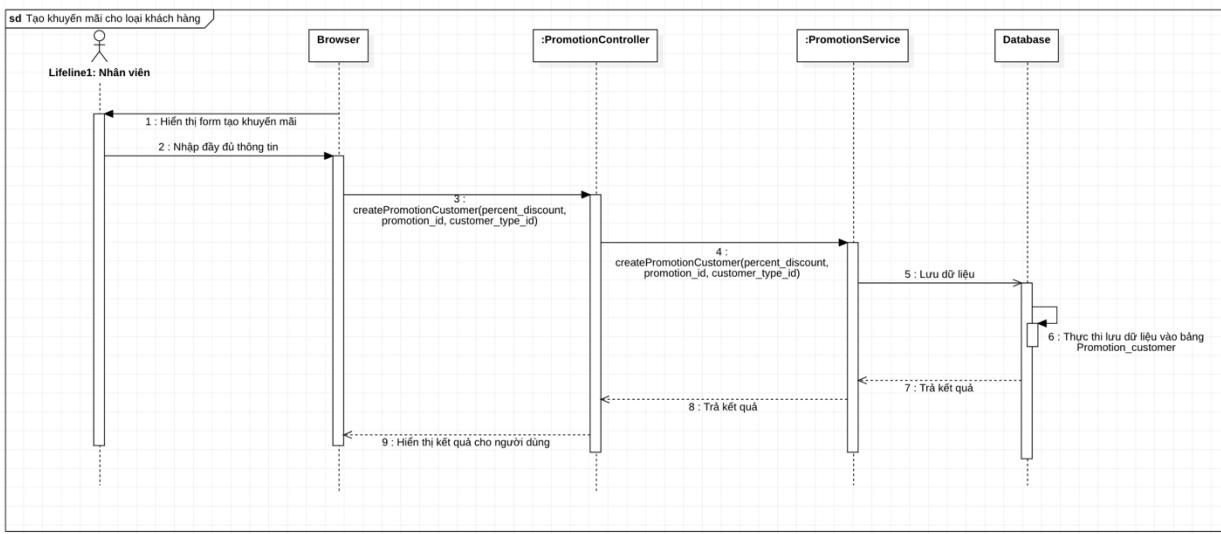


Hình 24: Lược đồ tuần tự cho Usecase thêm thông tin nhận hàng

#### 4.1.6 Usecase nhân viên tạo khuyến mãi :

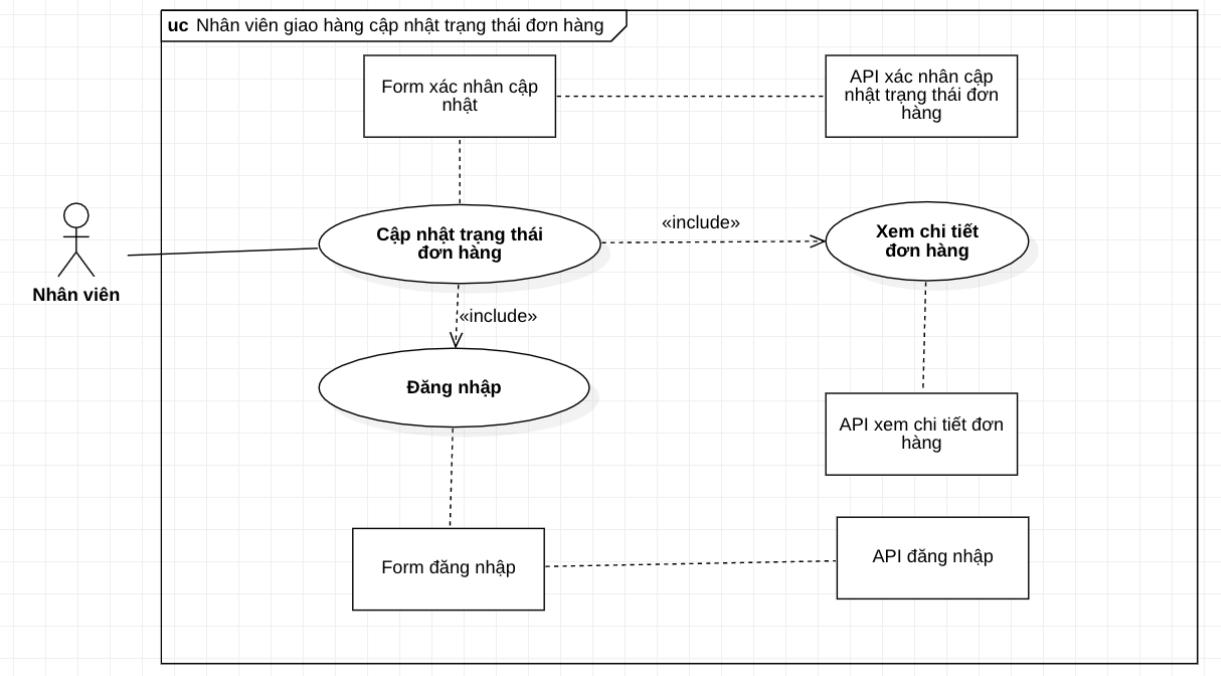


Hình 25: Usecase nhân viên tạo khuyến mãi

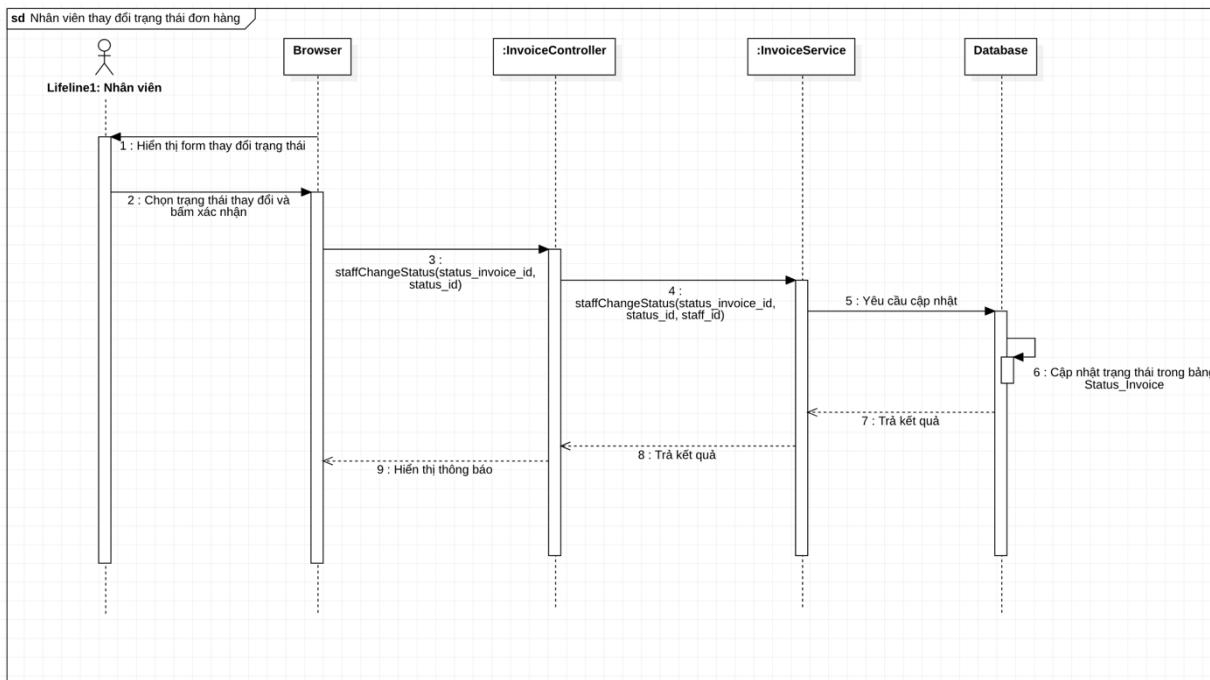


Hình 26: Lược đồ tuần tự cho Usecase nhân viên tạo khuyến mãi theo loại khách hàng

#### 4.1.7 Usecase nhân viên cập nhật trạng thái đơn hàng :



Hình 27: Usecase nhân viên cập nhật trạng thái đơn hàng



Hình 28: Lược đồ tuần tự cho Usecase nhân viên cập nhật trạng thái đơn hàng

## 4.2 Thiết kế các thành phần của phần mềm

### a. UsecaseID/Actor Name

- UC-01 / Khách hàng

Vai trò	Tác nhân sử dụng usecase này để lọc ra các đầu sách
Tương tác	<p>Form :</p> <ul style="list-style-type: none"> <li>- Form lọc sách theo thuộc tính(F01)</li> <li>- Form lọc sách theo custom key hoặc đánh giá(F02)</li> </ul> <p>API :</p> <ul style="list-style-type: none"> <li>- API lọc sách theo các thuộc tính của sách(API01)</li> <li>- API lọc sách theo custom keyword(API02)</li> <li>- API lọc sách theo số lượng mua và đánh giá (API03)</li> <li>- API lấy danh sách các loại sách(API04)</li> </ul>

- UC-02 / Khách hàng

Vai trò	Tác nhân sử dụng usecase này để thêm sách vào giỏ hàng
---------	--

Tương tác	<p>Form :</p> <ul style="list-style-type: none"> <li>- Form thêm sách vào giỏ hàng(F03)</li> </ul> <p>API :</p> <ul style="list-style-type: none"> <li>- API xem chi tiết sách(API05)</li> <li>- API thêm sách vào giỏ hàng (API06)</li> </ul>
-----------	--

- UC-03 / Khách hàng

Vai trò	Tác nhân sử dụng usecase này để đặt sách
Tương tác	<p>Form :</p> <ul style="list-style-type: none"> <li>- Form xác nhận thông tin đặt sách(F04)</li> </ul> <p>API :</p> <ul style="list-style-type: none"> <li>- API thay đổi thông tin nhận sách(API07)</li> <li>- API xem giỏ hàng (API08)</li> <li>- API xoá đầu sách khỏi giỏ hàng(API09)</li> <li>- API chỉnh sửa số lượng đầu sách(API10)</li> <li>- API đặt sách (API11)</li> </ul>

- UC-04 / Khách hàng

Vai trò	Tác nhân sử dụng usecase này để đánh giá đơn hàng
Tương tác	<p>Form :</p> <ul style="list-style-type: none"> <li>- Form xem chi tiết đơn hàng.(F05)</li> <li>- Form đánh giá sản phẩm(F06)</li> <li>- Form xác nhận thông tin nhận sách(F04)</li> </ul> <p>API :</p> <ul style="list-style-type: none"> <li>- API xem chi tiết đơn hàng (API12)</li> <li>- API đánh giá sản phẩm (API13)</li> <li>- API đặt sách (API11)</li> </ul>

-UC-05 / Khách hàng

Vai trò	Tác nhân sử dụng usecase này để thêm thông tin nhận hàng
Tương tác	<p>Form :</p> <ul style="list-style-type: none"> <li>- Form thêm thông tin(F07)</li> </ul> <p>API :</p> <ul style="list-style-type: none"> <li>- API kiểm tra vị trí hợp lệ(API14)</li> <li>- API thêm thông tin nhận hàng(API15)</li> </ul>

-UC-06 / Nhân viên

Vai trò	Tác nhân sử dụng usecase này để thêm khuyến mãi
Tương tác	<p>Form :</p> <ul style="list-style-type: none"> <li>- Form tạo khuyến mãi theo loại khách hàng(F08)</li> </ul> <p>API :</p> <ul style="list-style-type: none"> <li>- API danh sách loại khách hàng(API16)</li> <li>- API danh sách khuyến mãi(API17)</li> <li>- API tạo khuyến mãi theo loại khách hàng(API18)</li> </ul>

-UC-07 / Nhân viên

Vai trò	Tác nhân sử dụng usecase này để thay đổi trạng thái đơn hàng
Tương tác	<p>Form :</p> <ul style="list-style-type: none"> <li>- Form xác nhận cập nhật trạng thái.(F9)</li> </ul> <p>API :</p> <ul style="list-style-type: none"> <li>- API xem chi tiết đơn hàng (API06)</li> <li>- API xác nhận cập nhật trạng thái đơn hàng(API19)</li> </ul>

### b. UseCaseID/ Form ID

- UC-01/F01

Tên Form	Form lọc sách theo các thuộc tính của sách
Giao tiếp với user	Khách hàng
Control	<ul style="list-style-type: none"> <li>- Textbox tên sách.</li> <li>- Dropdown Loại sách.</li> <li>- Textbox tên tác giả.</li> <li>- Textbox tên nhà xuất bản.</li> <li>- Textbox năm ra mắt.</li> <li>- Button tìm kiếm : Gửi và yêu cầu trả lại các đầu sách ở màn hình hiện tại.</li> </ul>
Xử lý	<ul style="list-style-type: none"> <li>- Gọi đến API lấy danh sách các loại sách cho dữ liệu ở Dropdown.</li> <li>- Sau khi điền bấm tìm kiếm, hệ thống gọi đến API lọc sách theo thuộc tính của sách.</li> </ul>

## Tìm kiếm nâng cao

Tên sách:

Loại sách:

Du lịch

Tên tác giả:

Tên nhà xuất bản:

Năm ra mắt:

Tìm kiếm

Hình 29: Form tìm kiếm theo thuộc tính của sách

- UC-01/F02

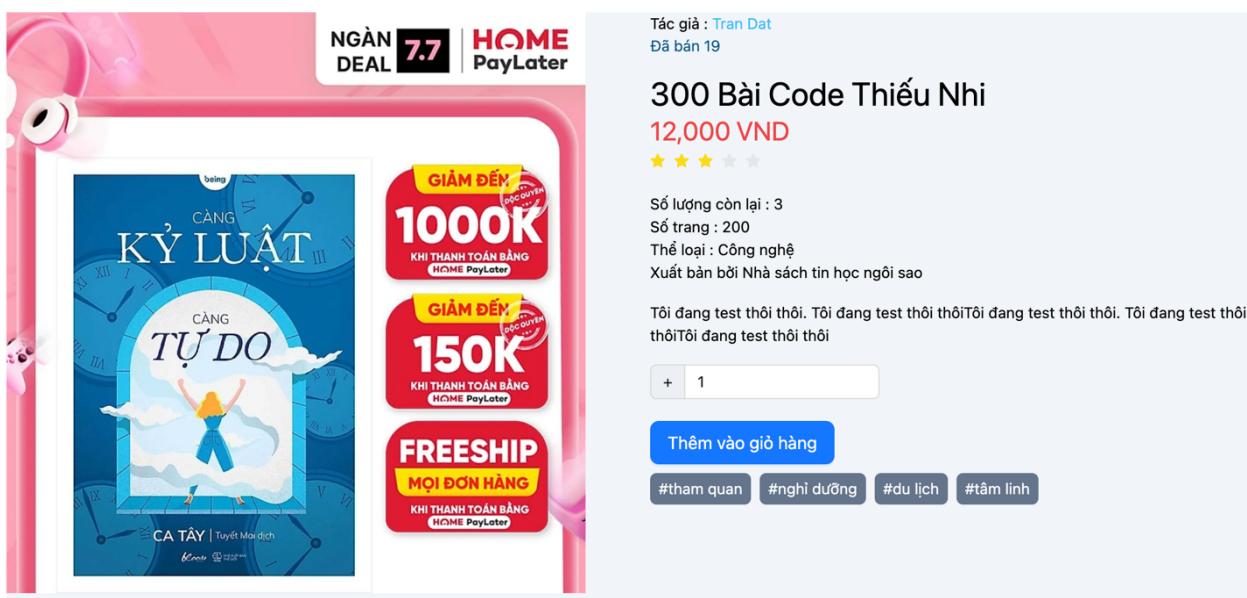
Tên Form	Form lọc sách theo custom key hoặc đánh giá
Giao tiếp với user	Khách hàng
Control	<ul style="list-style-type: none"> <li>- Textbox từ khoá để search.</li> <li>- Checkbox Lọc theo tag.</li> <li>- Checkbox Lọc theo lượt mua.</li> <li>- Button tìm kiếm : Gửi và yêu cầu trả lại các đầu sách ở màn hình hiện tại.</li> </ul>
Xử lý	<ul style="list-style-type: none"> <li>- Nếu check vào “Lọc theo tag” hệ thống sẽ gọi API lọc theo custom key để xử lý.</li> <li>- Nếu check vào “Lọc theo lượt mua” hệ thống sẽ gọi API lọc theo đánh giá để xử lý.</li> </ul>

The screenshot shows the Tiki search bar with a placeholder 'Bạn cần tìm gì?' and a blue 'Tìm kiếm' button. Below the search bar are two filter buttons: 'Lọc theo tag' (selected) and 'Lọc theo lượt mua'.

Hình 30: Form lọc sách theo custom key hoặc đánh giá

- UC-02/F03

Tên Form	Form thêm sách vào giỏ hàng
Giao tiếp với user	Khách hàng
Control	<ul style="list-style-type: none"> <li>- Textbox số lượng : chỉ cho phép nhập số.</li> <li>- Button thêm vào giỏ : Gửi yêu cầu và thông báo lại màn hình hiện tại.</li> </ul>
Xử lý	- Lấy dữ liệu số lượng và đầu sách để làm input cho API thêm sách vào giỏ hàng.



Hình 31: Form hiển thị thông tin sách và thêm vào giỏ hàng

- UC-03/F04

Tên Form	Form xác nhận thông tin đặt sách
Giao tiếp với user	Khách hàng

Control	- Button đặt hàng: xác nhận thông tin đặt sách và hiển thị thông báo.
Xử lý	- Sử dụng API đặt sách để xử lý đặt hàng. Sau khi xử lý xong đơn hàng được tạo ra trong cở sở dữ liệu ở bảng Invoice.

**Xác nhận đặt hàng**

X

**Thông tin nhận hàng**

Địa chỉ nhận hàng : số 1 võ văn ngân,Linh Chiểu, Thủ Đức, Hồ Chí Minh

Người nhận hàng : Tran Thu Dat

Số điện thoại : 0941374589

**Danh sách đặt hàng**

300 bài code thiếu nhi : 2 (quyển) \* 12,000 = 24,000 (VND)

Tổng : 24,000 (VND)

**Chi tiết thanh toán**

Tiền sản phẩm : 24,000 (VND)

Khoảng cách : 2.78 (km)

Trọng lượng : 0.7 (kg)

Phí vận chuyển : 20,000 (VND)

Khuyến mãi : 15(%)

Tổng hóa đơn : **44,000 37,400 (VND)****Đặt hàng**

Hình 32: Form xác nhận thông tin đặt hàng

- UC-04/F05

Tên Form	Form xem chi tiết đơn hàng
----------	----------------------------

Giao tiếp với user	Khách hàng
Control	
Xử lý	<ul style="list-style-type: none"> <li>- Sử dụng API xem chi tiết đơn hàng để lấy dữ liệu hiển thị lên giao diện cho người dùng.</li> </ul>

Đơn hàng 4

	<b>300 Bài Code Thiếu Nhi</b> Nặng : 1.75 kg	Số lượng : 5 Thành Tiền : 60,000 VND	Phí vận chuyển 20,000 VND Tổng số tiền (đã khuyến mãi) 32,000 VND
Trạng thái : Chờ duyệt Nơi giao : số 1 võ văn ngân, Linh Chiểu, Thủ Đức, Hồ Chí Minh Ngày đặt : 12/8/2023		Người nhận : Tran Thu Dat Số điện thoại người nhận : 0941374589	

Hình 33: Form chi tiết đơn hàng

- UC-04/F06

Tên Form	Form đánh giá sản phẩm
Giao tiếp với user	Khách hàng
Control	<ul style="list-style-type: none"> <li>- TextArea : Nhập bình luận về sản phẩm.</li> <li>- CheckBox : nhập số sao cho sản phẩm.</li> <li>- Button đánh giá: xác nhận đánh giá và hiển thị thông báo.</li> </ul>
Xử lý	<ul style="list-style-type: none"> <li>- Lấy dữ liệu đánh giá và số sao để truyền vào input cho API13 xử lý.</li> <li>- Sử dụng API đánh giá sản phẩm (API13). Sau khi xử lý đánh giá và số sao sẽ được lưu vào cơ sở dữ liệu bảng Cart_detail.</li> </ul>

The screenshot shows a modal window titled "Đánh giá sản phẩm". It contains a text area labeled "Đánh giá của bạn" with a placeholder "Nhập đánh giá của bạn" and a character count "0 / 300". Below this is a rating section labeled "Chất lượng sản phẩm" with five yellow stars. A blue button labeled "Đánh giá" is at the bottom right. The background of the modal has a dark grey gradient.

Hình 34: Form khách hàng đánh giá sản phẩm

- UC-05 / F07

Tên Form	Form thêm thông tin nhận hàng
Giao tiếp với user	Khách hàng
Control	<ul style="list-style-type: none"> <li>- Dropdown chọn tỉnh.</li> <li>- Dropdown chọn huyện.</li> <li>- Dropdown chọn xã.</li> <li>- Textbox nhập chi tiết.</li> <li>- Textbox nhập số điện thoại.</li> <li>- Textbox nhập tên người nhận.</li> <li>- Button nhận hàng : Lưu thông tin và hiển thị thông báo.</li> </ul>
Xử lý	<ul style="list-style-type: none"> <li>- Lấy tỉnh, quận, phường và địa chỉ chi tiết để làm input cho API kiểm tra vị trí hợp lệ(API15).</li> <li>- Lấy tên người nhận, số điện thoại, tỉnh, quận, phường và địa chỉ chi tiết để làm input cho API thêm thông tin nhận hàng(API14). Sau khi xử lý thông tin nhận hàng mới được lưu vào cơ sở dữ liệu bảng ReceiptInformation.</li> </ul>

**Thêm địa chỉ giao hàng**

\* Tên người nhận:

\* Số điện thoại người nhận:

\* Tỉnh thành:  ▾

\* Quận/huyện:  ▾

\* Phường/xã:  ▾

\* Chi Tiết địa chỉ:

0 / 200

**Thêm thông tin**

*Hình 35: Form khách hàng thêm thông tin nhận hàng*

- UC-06 / F08

Tên Form	Form tạo khuyến mãi theo loại khách hàng
Giao tiếp với user	Nhân viên
Control	<ul style="list-style-type: none"> <li>- Dropdown chọn loại khách hàng.</li> <li>- Dropdown chọn loại khuyến mãi để áp dụng.</li> <li>- Textbox nhập phần trăm giảm giá.</li> <li>- Button tạo khuyến mãi : Lưu thông tin và hiển thị thông báo.</li> </ul>
Xử lý	<ul style="list-style-type: none"> <li>- Sử dụng API danh sách khuyến mãi(API17) để dùng cho dữ liệu của dropdown chọn khuyến mãi.</li> <li>- Sử dụng API danh sách loại khách hàng(API16) để dùng cho dữ liệu của dropdown chọn loại khách hàng.</li> <li>- Lấy dữ liệu mã khuyến mãi, loại khách hàng và phần trăm giảm giá làm input cho API18 xử lý.</li> <li>- Sử dụng API tạo khuyến mãi theo loại khách hàng(API18). Sau khi hoàn thành sẽ thêm dữ liệu khuyến mãi vào bảng Promotion_customer.</li> </ul>

The screenshot shows a software interface for managing discounts. On the left, there's a sidebar with options like 'Quản lý sách', 'Quản lý khuyến mãi' (selected), 'Tạo khuyến mãi', 'Khuyến mãi cho...', and 'Quản lý đơn hàng'. The main area displays a list of existing discounts, including one for 'Khuyến mãi tháng 9' with a 10% discount from 25/08/2023 to 05/09/2023. A modal window titled 'Thêm khuyến mãi cho khách hàng' (Add discount for customer type) is open, containing fields for 'Chọn khuyến mãi áp dụng:' (Select discount application), 'Chọn loại khách hàng:' (Select customer type), and 'Phần trăm giảm giá:' (Percentage discount). A blue button labeled 'Thêm khuyến mãi' (Add discount) is at the bottom right of the modal.

Hình 36: Form thêm khuyến mãi cho loại khách hàng

- UC-07 / F9

Tên Form	Form cập nhật trạng thái
Giao tiếp với user	Nhân viên
Control	<ul style="list-style-type: none"> <li>- Dropdown chọn trạng thái đơn hàng.</li> <li>- Button Xác nhận : Lưu thông tin và hiển thị thông báo.</li> </ul>
Xử lý	<ul style="list-style-type: none"> <li>- Lấy thông tin từ API xem chi tiết đơn hàng để hiển thị các thông tin của đơn hàng mà khách hàng đặt.</li> <li>- Lấy dữ liệu từ dropdown chọn trạng thái để truyền input cho API cập nhật trạng thái đơn hàng xử lý. Sau khi xử lý sẽ cập nhật trạng thái vào cơ sở dữ liệu.</li> </ul>

The screenshot shows a software interface for managing orders. On the left, there's a sidebar with options like 'Quản lý khuyến mãi' (selected), 'Quản lý đơn hàng' (selected), and 'Tạo khuyến mãi'. The main area displays a list of orders with columns for 'Ngày đặt' (Order date), 'Đơn hàng' (Order), 'Tổng tiền' (Total price), and 'Trạng thái' (Status). A modal window titled 'Đơn hàng' (Order) is open, containing sections for 'Thông tin đơn hàng' (Order information), 'Danh sách đặt hàng' (Order items), 'Chi tiết thanh toán' (Payment details), and 'Thay đổi trạng thái' (Change status). The 'Thay đổi trạng thái' section includes a dropdown for selecting a status and a blue 'Xác nhận' (Confirm) button. The background shows a list of orders with various statuses like 'Đã giao' (Delivered) and 'Chờ duyệt' (Pending review).

Hình 37: Form thay đổi trạng thái của đơn hàng

**c. API / Service ID****API lọc sách theo các thuộc tính của sách(API01)**

- Url : /book/search-attribute-advanced
- Method : POST
- Input :

Tên	Kiểu dữ liệu	Bắt buộc	Mô tả
bookName	String		Chuỗi liên quan đến tên sách
typeBook	String		Tên loại sách
author	String		Chuỗi liên quan đến tác giả
publisher	String		Chuỗi liên quan đến nhà xuất bản
yearRelease	Number		Năm ra mắt

- Output : Dữ liệu các đầu sách thỏa để trả về view hiển thị cho người dùng.
- Xử lý :
  - o Gọi tới service bookService và sử dụng method searchBookByAttribute. Đầu tiên gán trọng số cho các tiêu chí càng quan trọng gán trọng số càng cao, ví dụ bookName gán 5, typeBook gán 4 ... Sử dụng phép AND và OR kiểm tra các chuỗi người dùng cung cấp có liên quan đến thông tin của sách để tính điểm cho từng đầu sách từ cơ sở dữ liệu. Có liên quan thì cộng điểm. Sau đó những cuốn sách nào có điểm lớn hơn 0 sẽ được sắp xếp và trả về cho người dùng.
  - o Kết quả trả về dạng Array chứa các Object là các đầu sách thỏa mãn được sắp xếp giảm dần của tổng trọng số.

**API lọc sách theo custom keyword(API02)**

- Url : /book/search-tag
- Method : POST
- Input :

Tên	Kiểu dữ liệu	Bắt buộc	Mô tả
tag	String	X	Chuỗi các tag liên quan

- Output : Dữ liệu các đầu sách thỏa mãn để trả về view hiển thị cho người dùng.
- Xử lý :
  - o Gọi tới service bookService và sử dụng method searchBookByTags. Phương thức này sẽ tách chuỗi tag mà người dùng cung cấp ra thành các từ riêng biệt. Sau đó sử dụng phép AND để kiểm tra có liên quan đến tag của sách từ cơ sở dữ liệu không. Có liên quan thì cộng điểm. Sau đó những cuốn sách nào có điểm lớn hơn 0 sẽ được sắp xếp và trả về cho người dùng.

- Kết quả trả về dạng Array chứa các Object là các đầu sách thỏa mãn được sắp xếp giảm dần của tổng trọng số.

### API lọc sách theo số lượng mua và đánh giá (API03)

- Url : /book/search-rate
- Method : POST
- Input :

Tên	Kiểu dữ liệu	Bắt buộc	Mô tả
query	String	X	Các từ liên quan đến sách

- Output : Dữ liệu các đầu sách thỏa mãn để trả về view hiển thị cho người dùng.
- Xử lý :
  - Gọi tới service bookService và sử dụng method searchBookByRate. Phương thức này dùng phép AND để kiểm tra query có liên quan đến các thuộc tính của sách trong cơ sở dữ liệu không. Sau đó kết quả sẽ được sắp xếp theo lượt mua và số sao.
  - Kết quả trả về dạng Array chứa các Object là các đầu sách thỏa mãn được sắp xếp giảm dần của tổng trọng số.

### API lấy danh sách các loại sách(API04)

- Url : /book-type
- Method : GET
- Output : Dữ liệu trả về tất cả các loại sách hiện có.
- Xử lý :
  - Gọi tới service bookTypeService và sử dụng method getAllTypeBook.

### API xem chi tiết sách(API05)

- Url : /book/:id
- Method : GET
- Query param :

Tên	Kiểu dữ liệu	Bắt buộc	Mô tả
Id	String	X	Là Id của đầu sách cần xem chi tiết

- Output : Là Object chứa dữ liệu của đầu sách để trả về view cho người dùng
- Xử lý :
  - Gọi tới service bookService và sử dụng method getBookById. Phương thức này sẽ lấy dữ liệu từ bảng Book.

- Kết quả trả về là Object chứa dữ liệu đầu sách.

### API thêm sách vào giỏ hàng (API06)

- Url : /cart
- Method : POST
- Input :

Tên	Kiểu dữ liệu	Bắt buộc	Mô tả
book_id	string	X	Là id của sách cần thêm vào giỏ hàng
quantity	Number	X	Là số lượng mà ta muốn mua

- Output : Thêm sách vào giỏ hàng, thông báo cho người dùng.
- Xử lý :
  - Gọi tới service cartService sử dụng method createCart.
  - Sau khi gọi tới sẽ thêm sách và số lượng vào bảng cart.

### API thay đổi thông tin nhận sách(API07)

- Url : /receipt-information
- Method : PATCH
- Input :

Tên	Kiểu dữ liệu	Bắt buộc	Mô tả
receipt_infomation_id	String	X	Mã thông tin nhận sách.

- Output :
  - Trả về true nếu thay đổi thành công và false nếu chưa thành công. View hiển thị cho người dùng biết
- Xử lý :
  - Gọi tới ReceiptInfomationService và sử dụng method setDefault. Trường is\_default của thông tin có receipt\_infomation\_id sẽ được set bằng true và các thông tin còn lại trường is\_default sẽ set bằng false.

### API xem giỏ hàng (API08)

- Url : /cart
- Method : GET
- Input : customer\_id lấy từ Json web token.
- Output : Là array chứa các object là các đầu sách và số lượng đặt.
- Xử lý :

- Gọi tới CartService và sử dụng method GetCartOfCustomer.
- Lọc cart có isCompleted là false và có customer\_id trùng với đầu vào.

### API xoá đầu sách khỏi giỏ hàng(API09)

- Url : /cart/:book\_id
- Method : DELETE
- Query param :

Tên	Kiểu dữ liệu	Bắt buộc	Mô tả
Book_id	String	X	Mã đầu sách cần xoá
Customer_id	String	X	Mã khách hàng

- Output : Mã thành công hay thất bại.
- Xử lý :
  - Gọi tới CartService và sử dụng method deleteBookInCart. Method truy vấn tìm ra dữ liệu trong bảng CartDetail và xoá nó khỏi cơ sở dữ liệu.

**API cập nhật số lượng đầu sách(API10) :** Dùng chung với API06 thêm sách vào giỏ hàng

### API đặt sách (API11)

- Url : /invoice
- Method : POST
- Input :

Tên	Kiểu dữ liệu	Bắt buộc	Mô tả
receipt_infomation_id	String	X	Id của thông tin nhận hàng
feeTotal	Number	X	Tổng số tiền cần trả
feeShip	Number	X	Phí giao hàng

- Output : Thêm đơn đặt hàng. Hiển thị thông báo.
- Xử lý :
  - Gọi tới InvoiceService và sử dụng method createInvoice.
  - Tạo hoá đơn, tạo chi tiết hoá đơn với trạng thái chờ duyệt.

### API xem chi tiết đơn hàng (API12)

- Url : /invoice
- Method : GET

- Output : Danh sách các đầu sách và số lượng mà khách hàng đặt.
- Xử lý :
  - o Gọi đến InvoiceService và sử dụng method getInvoiceStatusOfCustomer.

### API đánh giá sản phẩm (API13)

- Url : /rate
- Method : POST
- Input :

Tên	Kiểu dữ liệu	Bắt buộc	Mô tả
comment	String	X	Bình luận của khách cho sản phẩm.
star	Number	X	Số sao mà khách cho sản phẩm.
cart_detail_id	String	X	Mã chi tiết giỏ hàng
customer_id	String	X	Mã khách hàng đánh giá sản phẩm.

- Output : Thông báo đánh giá thành công hoặc thất bại.
- Xử lý :
  - o Gọi đến RateService và sử dụng method createRate. Method này tạo rate, cập nhật số sao và tổng đánh giá cho Book.

### API kiểm tra thông tin vị trí hợp lệ (API14):

- Url: <https://api.mapbox.com/directions/v5/mapbox/driving/:address1/:address2>
- Method : GET
- Params :

Tên	Kiểu dữ liệu	Bắt buộc	Mô tả
Address1	String	X	Toạ độ vị trí nhà sách
Address2	String	X	Toạ độ vị trí khách hàng

- Output : trả về khoảng cách giữa 2 địa điểm.

### API thêm thông tin nhận hàng(API15)

- Url : /receipt-infomation
- Method : POST
- Input :

Tên	Kiểu dữ liệu	Bắt buộc	Mô tả
Province	String	X	Tên tỉnh
District	String	X	Tên quận/huyện
Commune	String	X	Tên phường/xã
descriptionAddress	String	X	Số nhà, tên đường
Phone	String	X	Số điện thoại người nhận
nameReceipt	String	X	Tên người nhận hàng

- Output : Hiển thị trạng thái cho người dùng.
- Xử lý :
  - o Gọi tới service receiptInfomation và sử dụng method createReceiptInfomation.
  - o Tạo receipt information mới. Thêm receipt information này vào bảng ReceiptInformation

### API danh sách loại khách hàng(API16)

- Url : /customer-type
- Method : GET
- Output : Array chứa các object là các loại khách hàng để trả về view hiển thị.
- Xử lý :
  - o Gọi tới service customerTypeService và sử dụng method getAllCustomerType. Method này truy vấn bảng CustomerType.
  - o Kết quả trả về là array chứa các object là các type customer.

### API danh sách khuyến mãi(API17)

- Url : /promotion
- Method : GET
- Output : Array chứa các object là các khuyến mãi để trả về view hiển thị.
- Xử lý :
  - o Gọi tới service promotionService và sử dụng method getAllPromotion. Method này truy vấn bảng Promotion.
  - o Kết quả trả về là array chứa các object là các promotion của cửa hàng đã tạo.

### API thêm khuyến mãi theo loại khách hàng(API18)

- Url : /promotion/customer-type
- Method : POST
- Input :

Tên	Kiểu dữ liệu	Bắt buộc	Mô tả
Percent_discount	Number	X	Phần trăm giảm giá
Promotion_id	String	X	Mã khuyến mãi
Customer_type_id	String	X	Mã loại khách hàng áp dụng khuyến mãi.

- Output : Khuyến mãi vừa tạo.
- Xử lý :
  - o Gọi tới service promotionService và sử dụng method createPromotionCustomer. Tạo thêm dữ liệu trong bảng promotionCustomer.

#### API Cập nhật trạng thái đơn hàng(API19)

- Url : invoice/status-invoice
- Method : PATCH
- Input :

Tên	Kiểu dữ liệu	Bắt buộc	Mô tả
Staff_id	String	X	Mã nhân viên cập nhật trạng thái đơn hàng
Status_id	String	X	Mã trạng thái
Status_invoice_id	String	X	Mã trạng thái đơn hàng.

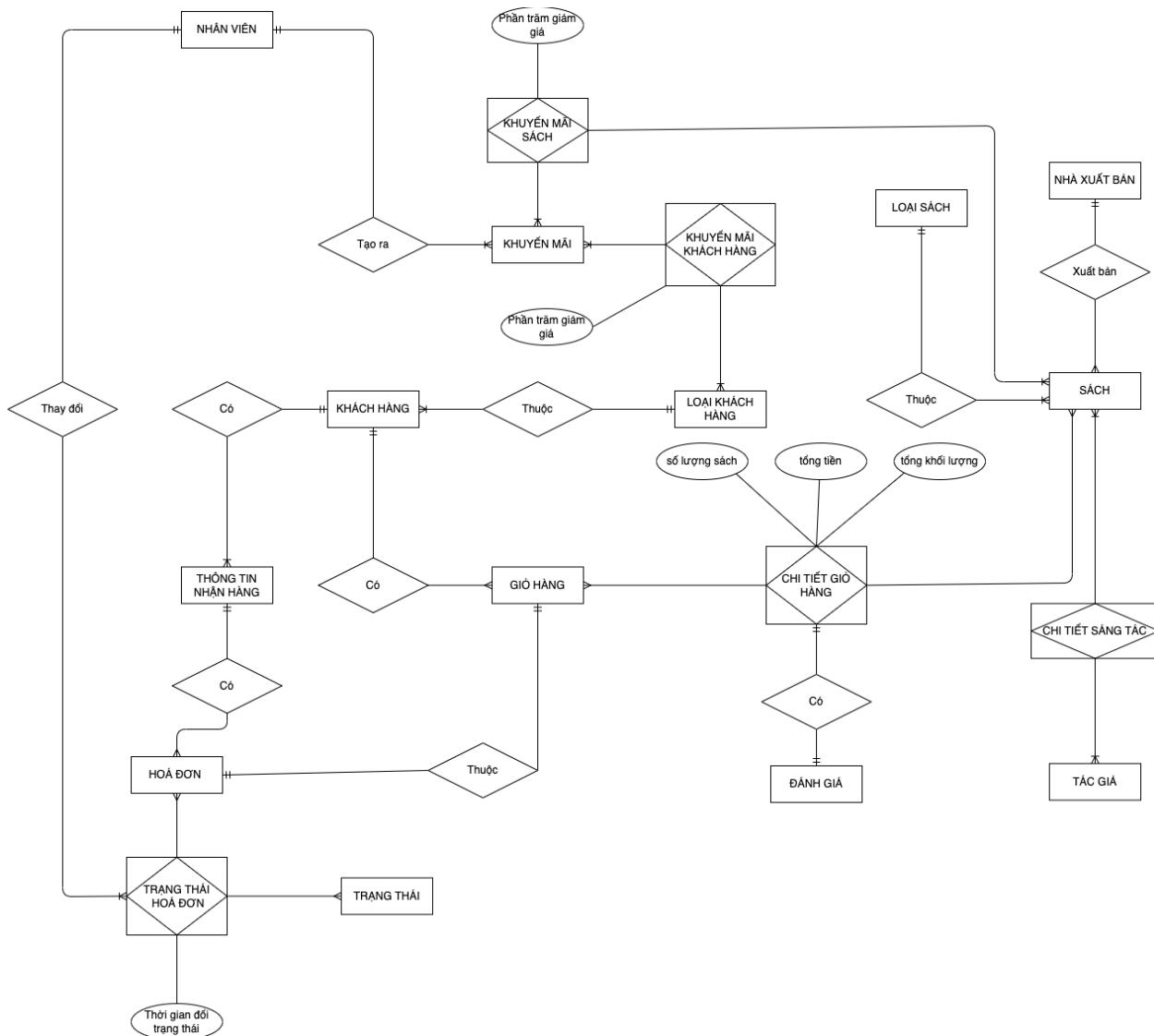
- Output : Trả về cập nhật thành công hay thất bại.
- Xử lý :
  - o Gọi đến server InvoiceService và sử dụng method staffChangeStatus.
  - o Kiểm tra trạng thái, nhân có tồn tại không. Sau đó tiến hành cập nhật.

#### 4.3. Thiết kế cơ sở dữ liệu và ràng buộc

#### 4.3.1. Mô hình thực thể ERD

- Nhân viên (mã nhân viên, họ, tên, giới tính, địa chỉ, ngày sinh, số điện thoại, email, avatar, mật khẩu, quyền).
- Khách hàng (mã khách hàng, họ, tên, giới tính, ngày sinh, số điện thoại, email, avatar, mật khẩu, quyền).
- Loại khách hàng (mã loại, tên loại).
- Thông tin nhận hàng (mã thông tin, tỉnh, huyện, xã, chi tiết địa chỉ, số điện thoại, tên người nhận, mặc định).
- Khuyến mãi (mã khuyến mãi, tên khuyến mãi, thời điểm bắt đầu, thời điểm kết thúc, lý do khuyến mãi).
- Sách (mã sách, tên sách, hình ảnh, số trang, giá, năm phát hành, số lượng còn, số lượng đã bán, cân nặng, tổng số đánh giá, tổng số sao).
- Loại sách(mã loại sách, tên loại sách).
- Đánh giá (mã đánh giá, bình luận, số sao).
- Tác giả (mã tác giả, họ, tên, giới tính, ngày sinh, số điện thoại, địa chỉ, email).
- Nhà xuất bản (mã nhà xuất bản, tên, địa chỉ, số điện thoại, email).
- Giỏ hàng(mã giỏ hàng, đã đặt).
- Hóa đơn (mã hóa đơn, ngày lập hóa đơn, tổng số tiền, phí giao hàng).
- Trạng thái hóa đơn (mã trạng thái, tên trạng thái, chi tiết).

#### 4.3.2. Mô hình thực thể kết hợp ERD



Hình 38: Mô hình ERD

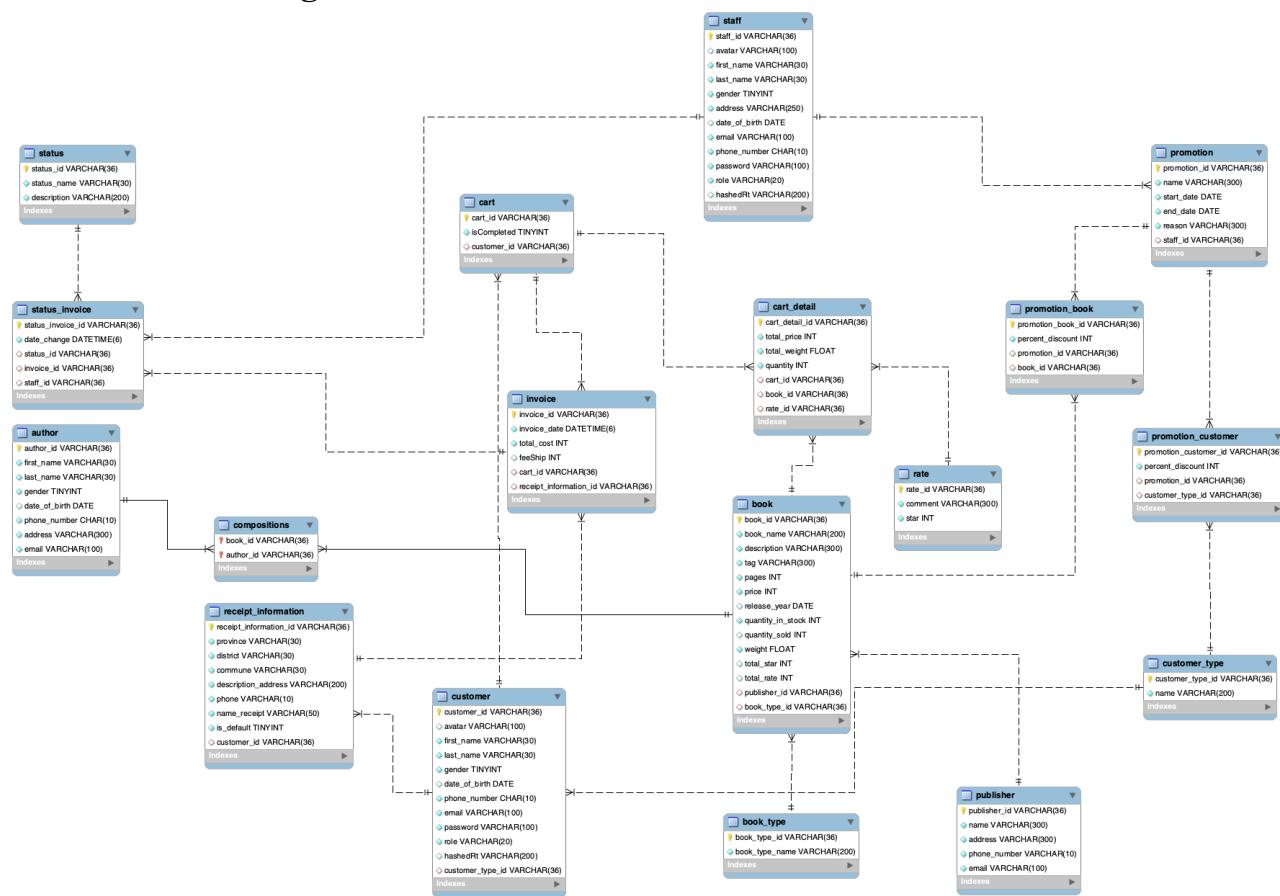
#### 4.3.3. Mô hình dữ liệu quan hệ từ ERD

- Nhân viên (mã nhân viên, họ, tên, giới tính, địa chỉ, ngày sinh, số điện thoại, email, avatar, mật khẩu, quyền).
- Khách hàng (mã khách hàng, họ, tên, giới tính, ngày sinh, số điện thoại, email, avatar, mật khẩu, quyền, **mã loại khách hàng**).
- Loại khách hàng (mã loại, tên loại).
- Thông tin nhận hàng (mã thông tin, tỉnh, huyện, xã, chi tiết địa chỉ, số điện thoại, tên người nhận, mặc định, **mã khách hàng**).
- Khuyến mãi (mã khuyến mãi, tên khuyến mãi, thời điểm bắt đầu, thời điểm kết thúc, lý do khuyến mãi, **mã nhân viên tạo khuyến mãi**).

- Khuyến mãi sách (mã khuyến mãi sách, mã khuyến mãi, mã sách, phần trăm giảm giá).
- Khuyến mãi theo loại khách (mã khuyến mãi loại khách hàng, mã khuyến mãi, mã loại khách hàng, phần trăm giảm giá).
- Sách (mã sách, tên sách, hình ảnh, chi tiết sách, tag, số trang, giá, năm phát hành, số lượng còn, số lượng đã bán, cân nặng, tổng số đánh giá, tổng số sao, **mã loại sách**, **mã nhà xuất bản**).
- Loại sách(mã loại sách, tên loại sách).
- Dánh giá (mã đánh giá, bình luận, số sao).
- Tác giả (mã tác giả, họ, tên, giới tính, ngày sinh, số điện thoại, địa chỉ, email).
- Sáng tác (mã sách, mã tác giả).
- Nhà xuất bản (mã nhà xuất bản, tên nhà xuất bản, địa chỉ, số điện thoại, email).
- Giỏ hàng(mã giỏ hàng, đã đặt hàng, **mã khách hàng**).
- Chi tiết giỏ hàng (mã chi tiết giỏ hàng, mã giỏ hàng, mã sách, tổng tiền, tổng cân nặng, số lượng, **mã đánh giá**).
- Hoá đơn (mã hoá đơn, ngày lập hoá đơn, tổng số tiền, phí giao hàng, **mã giỏ hàng**, **mã thông tin nhận hàng**).
- Trạng thái (mã trạng thái, tên trạng thái, chi tiết).
- Trạng thái hoá đơn (mã trạng thái hoá đơn, mã trạng thái, mã hoá đơn, ngày thay đổi, **mã nhân viên thay đổi trạng thái**).

#### 4.4. Thiết kế dữ liệu

#### 4.4.1. Mô hình Diagram



Hình 39: Mô hình Diagram

#### 4.4.2. Từ điển dữ liệu

Rate(rate\_id, comment, star, book\_id, customer\_id)

STT	Tên trường	Kiểu	Diễn giải	Khóa duy nhất	Khóa chính	Khóa ngoại	Not Null
1	rate_id	varchar(36)	Mã đánh giá		X		X
2	comment	varchar(300)	Bình luận				X
3	star	int	Số sao				X

Bảng 12: Từ điển dữ liệu bảng Rate

**Author**(author\_id, first\_name, last\_name, gender, date\_of\_birth, phone\_number, address, email)

STT	Tên trường	Kiểu	Diễn giải	Khóa duy nhất	Khóa chính	Khóa ngoại	Not Null
1	author_id	varchar(36)	Mã tác giả		X		X
2	first_name	varchar(30)	Họ				X
3	last_name	varchar(30)	Tên				X
4	gender	tinyint	Giới tính				X
5	date_of_birth	date	Ngày sinh				
6	phone_number	char(10)	Số điện thoại	X			X
7	address	varchar(300)	Địa chỉ				X
8	email	varchar(300)	Email	X			X

Bảng 13: Từ điển dữ liệu bảng Author

**Book\_type**(book\_type\_id, booking\_type\_name)

STT	Tên trường	Kiểu	Diễn giải	Khóa duy nhất	Khóa chính	Khóa ngoại	Not Null
1	booking_type_id	varchar(36)	Mã loại sách		X		X
2	booking_type_name	varchar(200)	Tên loại sách	X			X

Bảng 14: Từ điển dữ liệu bảng Book\_type

**Publisher**(publisher\_id, name, address, phone\_number, email)

STT	Tên trường	Kiểu	Diễn giải	Khóa duy nhất	Khóa chính	Khóa ngoại	Not Null
1	publisher_id	varchar(36)	Mã nhà xuất bản		X		X

2	name	varchar(300)	Tên nhà xuất bản	X			X
3	address	varchar(300)	Địa chỉ				X
4	phone_number	varchar(10)	Số điện thoại	X			X
5	email	varchar(100)	Email	X			X

Bảng 15: Từ điển dữ liệu bảng Publisher

**Compositions(book\_id, author\_id)**

STT	Tên trường	Kiểu	Diễn giải	Khóa duy nhất	Khóa chính	Khóa ngoại	Not Null
1	book_id	varchar(36)			X	X	X
2	author_id	varchar(36)			X	X	X

Bảng 16: Từ điển dữ liệu bảng Compositions

**Book(book\_id, book\_name, description, tag, pages, price, release\_year, quantity\_stock, quantity\_in\_stock, weight, publisher\_id, book\_type\_id)**

STT	Tên trường	Kiểu	Diễn giải	Khóa duy nhất	Khóa chính	Khóa ngoại	Not Null
1	book_id	varchar(36)	Mã sách		X		X
2	book_name	varchar(200)	Tên sách	X			X
3	description	varchar(300)	Mô tả				X
4	tag	varchar(300)					X
5	pages	int	Số trang				X
6	price	int	Giá				X
7	release_year	date	Năm phát hành				
8	quantity_sold	int	Số sách đã bán				X

9	quantity_in_stock	int	Số sách còn lại				X
10	weight	float	Cân nặng				X
11	publisher_id	varchar(36)	Mã nhà xuất bản			X	X
12	book_type_id	varchar(36)	Mã loại sách			X	X
13	total_star	Int	Tổng số sao mà người mua đánh giá				X
14	Total_rate	int	Tổng số đánh giá của người mua				X

Bảng 17: Từ điển dữ liệu bảng Book

**Customer\_type**(customer\_type\_id, name)

STT	Tên trường	Kiểu	Diễn giải	Khóa duy nhất	Khóa chính	Khóa ngoại	Not Null
1	customer_type_id	varchar(36)	Mã loại khách hàng		X		X
2	name	varchar(200)	Tên loại khách hàng	X			X

Bảng 18: Từ điển dữ liệu bảng Customer\_type

**Customer**(customer\_id, avatar, first\_name, last\_name, gender, date\_of\_birth, phone\_number, email, password, role ,hashedRt, customer\_type\_id)

STT	Tên trường	Kiểu	Diễn giải	Khóa duy nhất	Khóa chính	Khóa ngoại	Not Null
1	customer_id	varchar(36)	Mã khách hàng		X		X
2	avatar	varchar(100)	Ảnh đại diện				
3	first_name	varchar(30)	Họ				X
4	last_name	varchar(30)	Tên				X

5	gender	tinyint	Giới tính				X
6	date_of_birth	date	Ngày sinh				
7	phone_number	char(10)	Số điện thoại	X			X
8	email	varchar(100)	Email	X			X
9	password	varchar(100)	Mật khẩu				X
10	role	varchar(20)	Quyền				X
11	hashedRt	varchar(200)	Mã refresh token				X
12	customer_type_id	varchar(36)	Mã loại khách hàng		X		X

Bảng 19: Từ điển dữ liệu bảng Customer

**Receipt\_information**(receipt\_information\_id, province, district, commune, description\_address, phone, name\_receipt, is\_default, customer\_id)

STT	Tên trường	Kiểu	Điễn giải	Khóa duy nhất	Khóa chính	Khóa ngoại	Not Null
1	receipt_information_id	varchar(36)	Mã người nhận		X		X
2	province	varchar(30)	Tên tỉnh				X
3	district	varchar(30)	Tên huyện				X
4	commune	varchar(30)	Xã, phường				X
5	description_address	varchar(200)	Địa chỉ chi tiết				X
6	phone	varchar(10)	Số điện thoại				X
7	name_receipt	varchar(50)	Tên người nhận				X
8	is_default	tinyint	Mặc định				
9	customer_id	varchar(36)	Mã khách hàng		X		X

Bảng 20: Từ điển dữ liệu bảng Receipt\_information

**Staff**(staff\_id, avatar, first\_name, last\_name, gender, address, date\_of\_birth, email, phone\_number, password, role, hashedRt)

STT	Tên trường	Kiểu	Diễn giải	Khóa duy nhất	Khóa chính	Khóa ngoại	Not Null
1	staff_id	varchar(36)	Mã nhân viên		X		X
2	avatar	varchar(100)	Ảnh đại diện				X
3	first_name	varchar(30)	Họ				X
4	last_name	varchar(30)	Tên				X
5	address	varchar(250)	Địa chỉ				X
6	gender	tinyint	Giới tính				X
7	date_of_birth	date	Ngày sinh				
8	phone_number	char(10)	Số điện thoại	X			X
9	email	varchar(100)	Email	X			X
10	password	varchar(100)	Mật khẩu				X
11	role	varchar(20)	Quyền				X
12	hashedRt	varchar(200)	Mã refresh token				X

Bảng 21: Từ điển dữ liệu bảng Staff

**Promotion**(promotion\_id, name, start\_date, end\_date, reason, staff\_id)

STT	Tên trường	Kiểu	Diễn giải	Khóa duy nhất	Khóa chính	Khóa ngoại	Not Null
1	promotion_id	varchar(36)	Mã khuyến mãi		X		X

2	name	varchar(300)	Tên khuyến mãi				X
3	start_date	date	Ngày kết thúc				X
4	end_date	date	Ngày bắt đầu				X
5	reason	varchar(300)	Lý do				X
6	staff_id	varchar(36)	Mã nhân viên tạo khuyến mãi			X	X

Bảng 22: Từ điển dữ liệu bảng Promotion

**Promotion\_customer**(promotion\_customer\_id, percent\_discount, promotion\_id, customer\_type\_id)

STT	Tên trường	Kiểu	Diễn giải	Khóa duy nhất	Khóa chính	Khóa ngoại	Not Null
1	promotion_customer_id	varchar(36)	Mã khuyến mãi khách hàng		X		X
2	percent_discount	int	Phần trăm giảm giá				X
3	promotion_id	varchar(36)	Mã khuyến mãi			X	X
4	customer_type_id	varchar(36)	Mã loại khách			X	X

Bảng 23: Từ điển dữ liệu bảng Promotion\_customer

**Promotion\_book**(promotion\_book\_id, percent\_discount, promotion\_id, book\_id)

STT	Tên trường	Kiểu	Diễn giải	Khóa duy nhất	Khóa chính	Khóa ngoại	Not Null
1	promotion_book_id	varchar(36)	Mã khuyến mãi sách		X		X

2	percent_discount	int	Phần trăm giảm giá				X
3	promotion_id	varchar(36)	Mã khuyến mãi			X	X
4	book_id	varchar(36)	Mã sách			X	X

Bảng 24: Từ điển dữ liệu bảng Promotion\_book

**Status(status\_id, status\_name, description)**

STT	Tên trường	Kiểu	Diễn giải	Khóa duy nhất	Khóa chính	Khóa ngoại	Not Null
1	status_id	varchar(36)	Mã trạng thái		X		X
2	status_name	varchar(30)	Tên trạng thái		X		X
3	description	varchar(200)	Mô tả				X

Bảng 25: Từ điển dữ liệu bảng Status

**Cart(cart\_id, isCompleted, customer\_id)**

STT	Tên trường	Kiểu	Diễn giải	Khóa duy nhất	Khóa chính	Khóa ngoại	Not Null
1	cart_id	varchar(36)	Mã giỏ hàng		X		X
2	isCompleted	tinyint	Hoàn thành đặt hàng				X
3	customer_id	varchar(36)	Mã khách hàng			X	X

Bảng 26: Từ điển dữ liệu bảng Cart

**Cart\_detail(cart\_detail\_id, total\_price, total\_weight, quantity, cart\_id, book\_id)**

STT	Tên trường	Kiểu	Diễn giải	Khóa duy nhất	Khóa chính	Khóa ngoại	Not Null
1	cart_detail_id	varchar(36)	Mã chi tiết giỏ hàng		X		X

2	total_price	int	Tổng giá				X
3	total_weight	float	Tổng cân nặng				X
4	quantity	int	Số lượng				X
5	cart_id	varchar(36)	Mã giỏ hàng			X	X
6	book_id	varchar(36)	Mã sách			X	X
7	rate_id	Varchar(36)	Mã đánh giá			X	

Bảng 27: Tùy biến dữ liệu bảng Cart\_detail

**Invoice**(invoice\_id, invoice\_date, total\_cost, feeShip, cart\_id, receipt\_information\_id)

STT	Tên trường	Kiểu	Diễn giải	Khóa duy nhất	Khóa chính	Khóa ngoại	Not Null
1	invoice_id	varchar(36)	Mã hóa đơn		X		X
2	invoice_date	datetime	Ngày tạo hóa đơn				X
3	total_cost	int	Tổng giá				X
4	feeShip	int	Phí vận chuyển				X
5	cart_id	varchar(36)			X	X	
6	receipt_infor mation_id	varchar(36)	Mã biên lai		X	X	

Bảng 28: Tùy biến dữ liệu bảng Invoice

**Status\_invoice**(status\_invoice\_id, date\_change status\_id, invoice\_id, staff\_id)

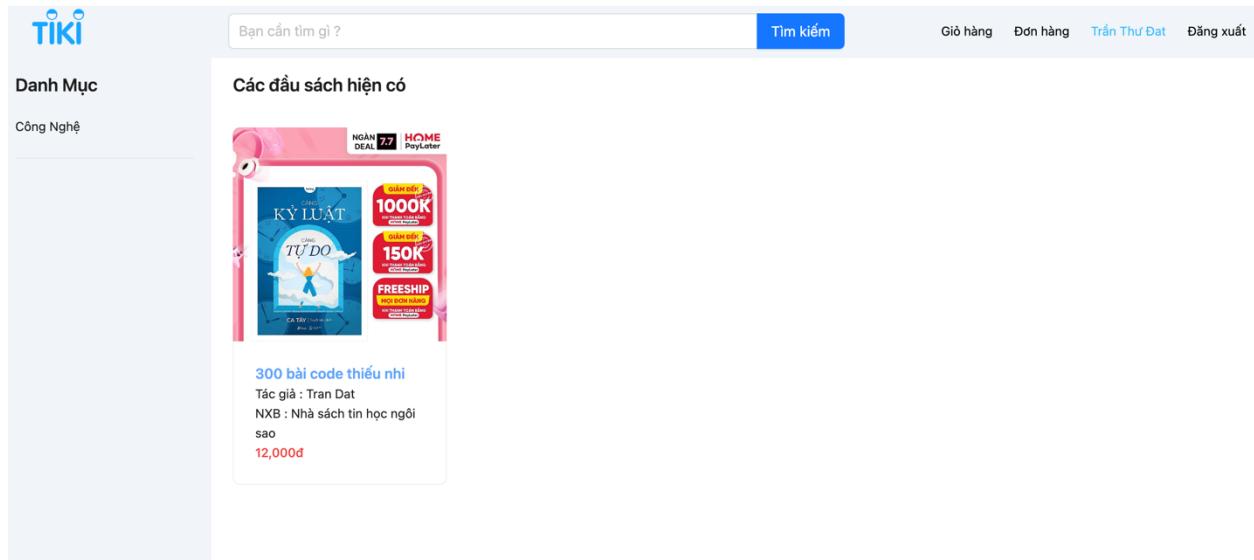
STT	Tên trường	Kiểu	Diễn giải	Khóa duy nhất	Khóa chính	Khóa ngoại	Not Null
1	status_invoice_ id	varchar(36)	Mã trạng thái hóa đơn		X		X

2	date_change	datetime	Ngày thay đổi trạng thái				X
3	invoice_id	varchar(36)	Mã hóa đơn			X	X
4	staff_id	varchar(36)	Mã nhân viên			X	
5	status_id	varchart(36)	Mã trạng thái			X	X

Bảng 29: Từ điển dữ liệu bảng Status\_invoice

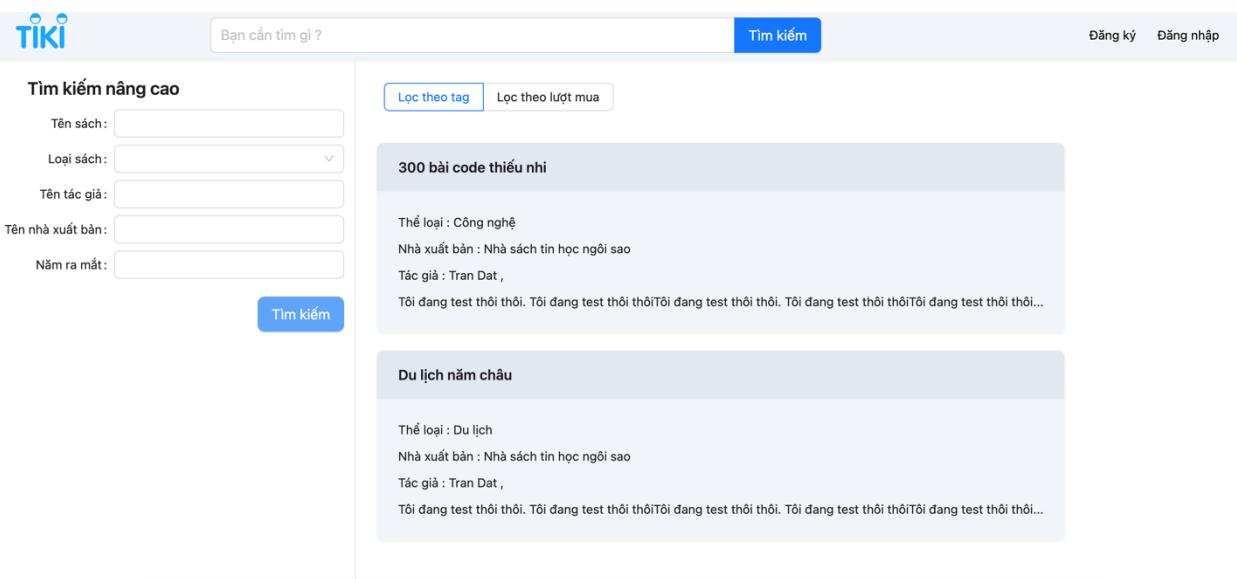
# CHƯƠNG 5: CHƯƠNG TRÌNH

## 5.1 Giao diện trang chủ :



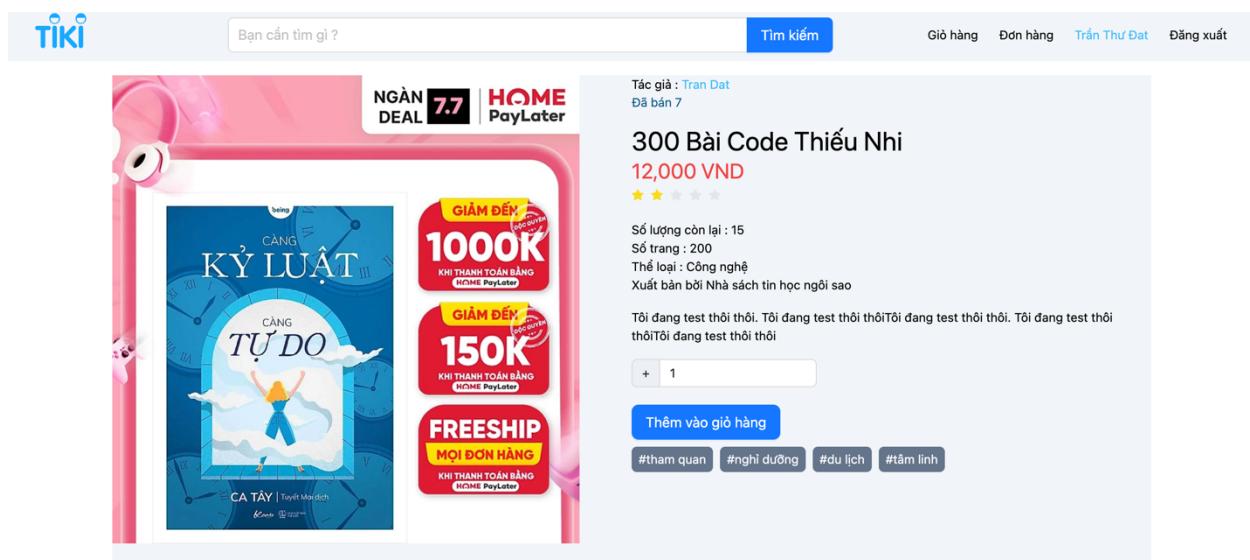
Hình 40: Giao diện trang chủ

## 5.2 Giao diện tìm kiếm :



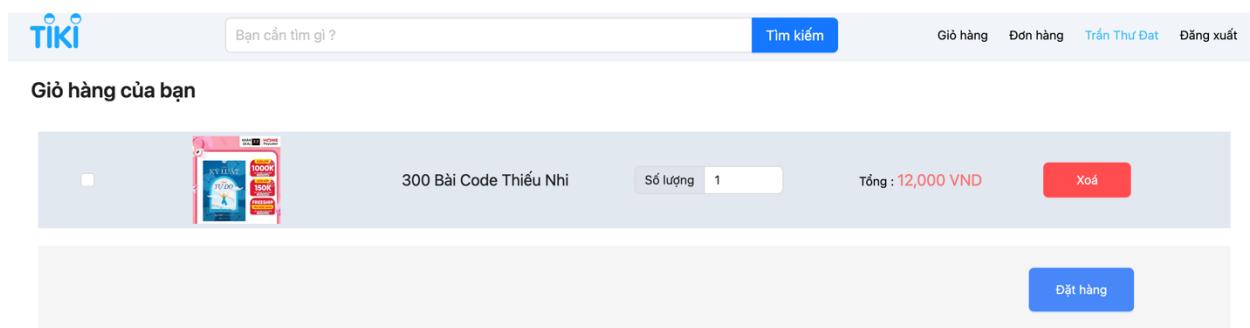
Hình 41: Giao diện tìm kiếm

### 5.3 Giao diện chi tiết sách :



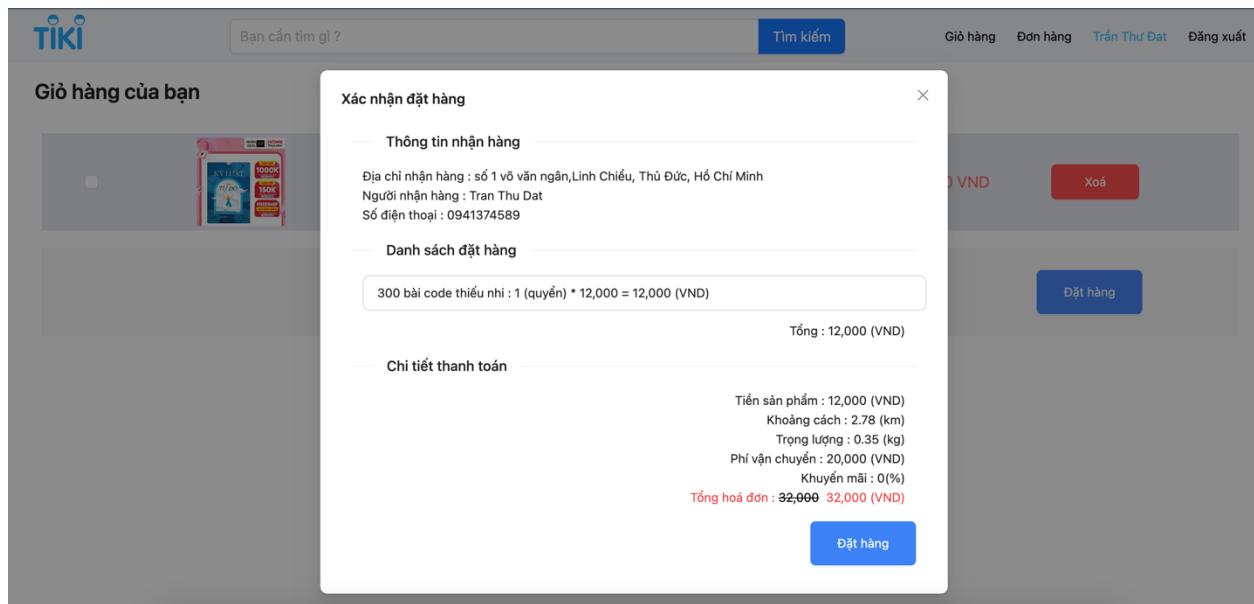
Hình 42: Giao diện chi tiết sách

### 5.4. Giao diện giỏ hàng :



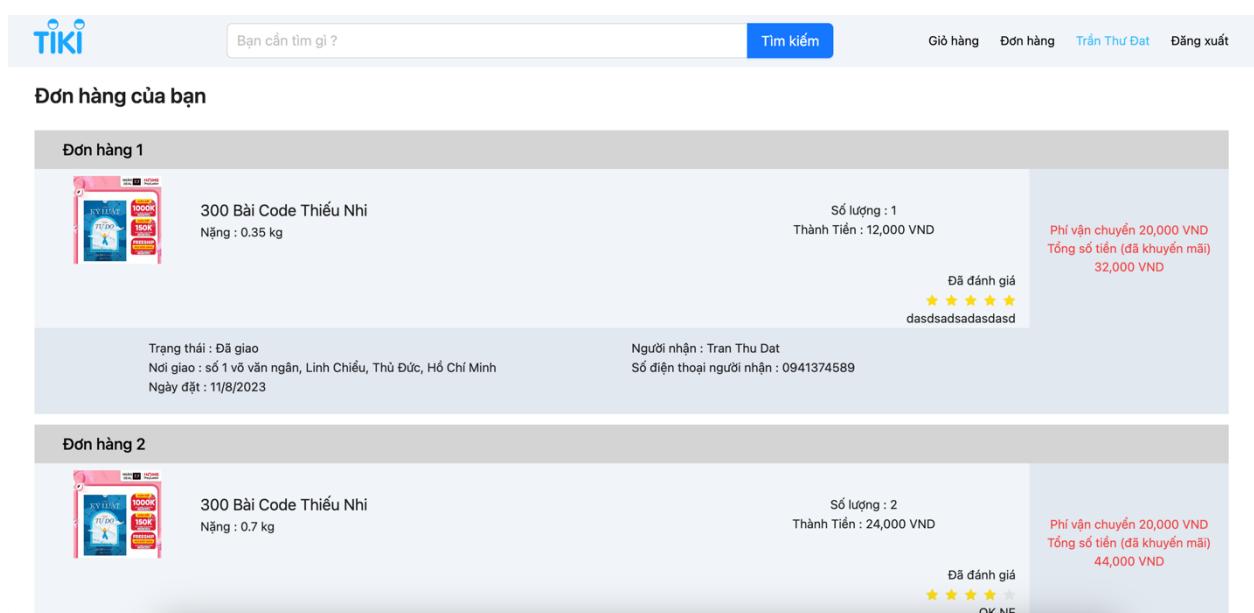
Hình 43: Giao diện giỏ hàng

## 5.5. Giao diện khách xác nhận đơn hàng :



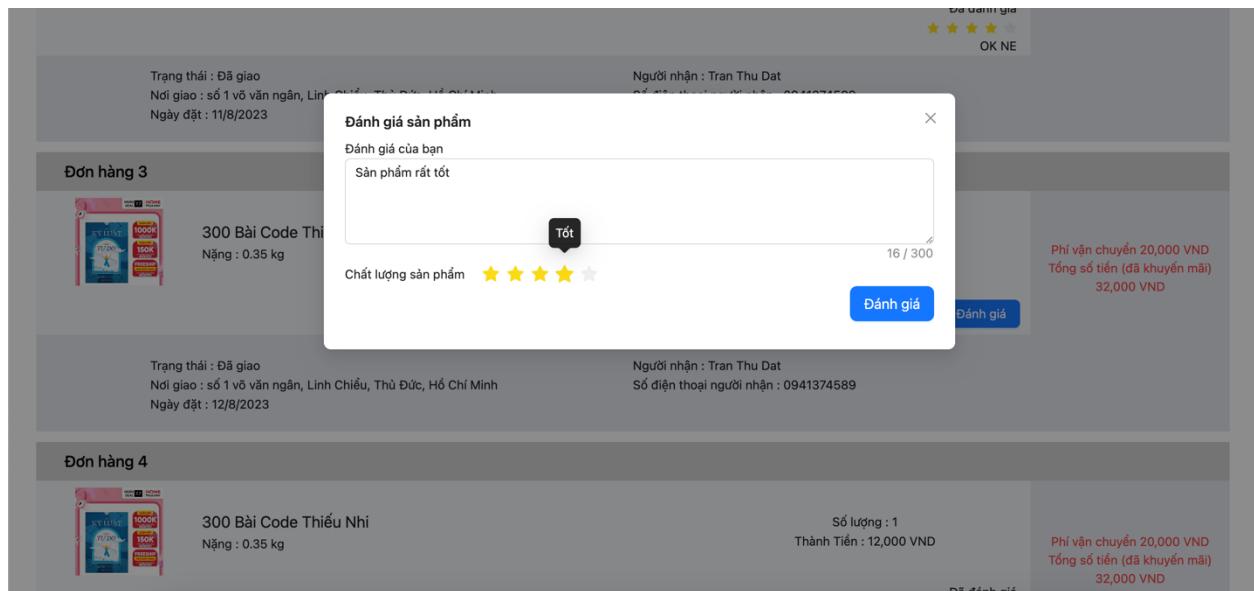
Hình 44: Giao diện khách xác nhận đơn hàng

## 5.6. Giao diện đơn hàng :



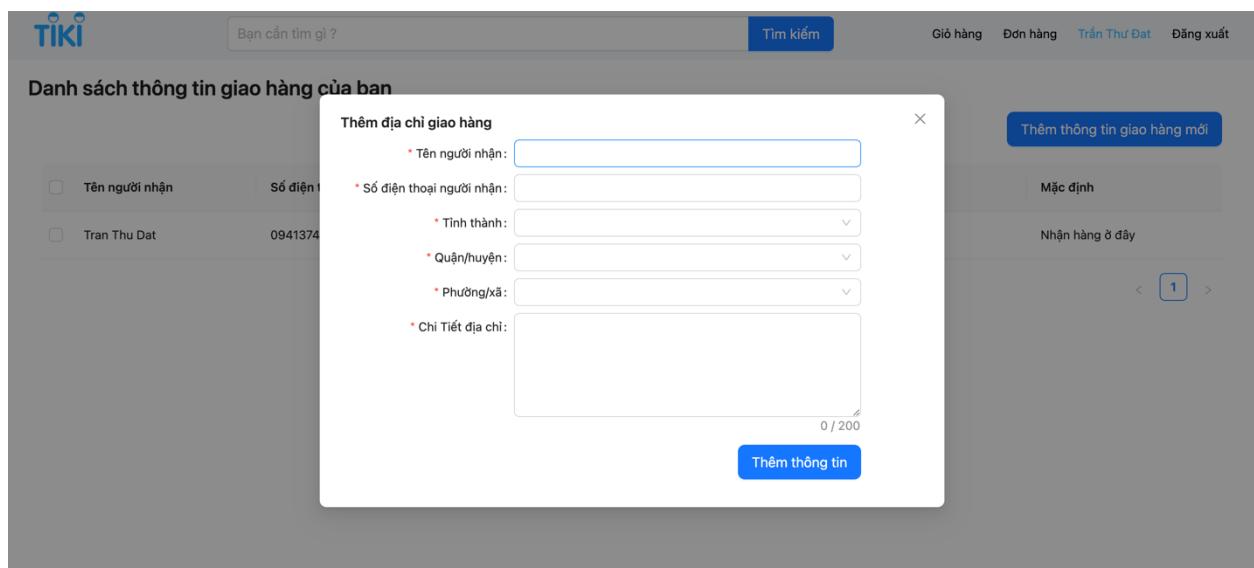
Hình 45: Giao diện đơn hàng

## 5.7 Giao diện cho khách đánh giá sản phẩm :



Hình 46: Giao diện khách hàng đánh giá sản phẩm

## 5.8. Giao diện thêm thông tin nhận hàng :



Hình 47: Giao diện thêm thông tin nhận sách

## 5.9. Giao diện danh sách nhận hàng của khách hàng

<input type="checkbox"/> Tên người nhận	Số điện thoại	Tỉnh	Huyện	Xã	Chi tiết	Mặc định
<input type="checkbox"/> Tran Thu Dat	0941374589	Hồ Chí Minh	Thủ Đức	Linh Chiểu	số 1 võ văn ngân	Nhận hàng ở đây
<input type="checkbox"/> Nguyễn Sơn Bá	0123456672	Hồ Chí Minh	Thủ Đức	An Khánh	số 11 thái thuận	

< 1 >

Hình 48: Giao diện danh sách thông tin nhận hàng

## 5.10. Giao diện admin

Ngày đặt hàng	Người nhận hàng	Địa chỉ giao hàng	Phí vận chuyển	Tổng tiền	Trạng thái
11/08/2023	Tran Thu Dat	số 1 võ văn ngân, Linh Chiểu, Thủ Đức, Hồ Chí Minh	20000(VND)	32000(VND)	Đã giao
11/08/2023	Tran Thu Dat	số 1 võ văn ngân, Linh Chiểu, Thủ Đức, Hồ Chí Minh	20000(VND)	44000(VND)	Đã giao
12/08/2023	Nguyễn Sơn Bá	số 11 thái thuận , An Khánh, Thủ Đức, Hồ Chí Minh	20000(VND)	44000(VND)	Đã giao
12/08/2023	Tran Thu Dat	số 1 võ văn ngân, Linh Chiểu, Thủ Đức, Hồ Chí Minh	20000(VND)	32000(VND)	Chờ duyệt
12/08/2023	Tran Thu Dat	số 1 võ văn ngân, Linh Chiểu, Thủ Đức, Hồ Chí Minh	20000(VND)	32000(VND)	Đã giao

< 1 2 >

Hình 49: Giao diện admin

## CHƯƠNG 6: TỔNG KẾT

### 6.1 Kết quả đạt được :

- Biết cách phân tích, thiết kế một cơ sở dữ liệu từ bài toán thực tế.
- Biết phân tích các chức năng và đưa các chức năng từ lý thuyết đến bản demo.
- Xây dựng được giao diện phục vụ nhu cầu tìm kiếm và đặt hàng của khách.
- Tính đúng đắn : Thực hiện đầy đủ yêu cầu người dùng.
- Tính tiện dụng : Giao diện đơn giản dễ sử dụng.
- Tính tương thích : Có thể chạy trên các nền tảng khách nhau.

### 6.2 Hạn chế của đề tài :

- Chương trình chỉ phân quyền về mức ứng dụng : chưa đảm bảo tính bảo mật về cơ sở dữ liệu.
- Chưa tích hợp được thanh toán qua ví điện tử, thẻ ngân hàng ...
- Dữ liệu cơ sở dữ liệu còn bị thụ động khó cập nhật thông tin khi dữ liệu thực tế bị thay đổi.

### 6.3 Hướng phát triển trong tương lai :

- Xem xét, cải thiện về mặt giao diện và hiệu năng.
- Khắc phục những nhược điểm đang tồn tại trong hệ thống.
- Hoàn thiện chương trình một cách tối ưu nhất để đem vào sử dụng thực tế và cho người dùng sử dụng một cách thuận tiện nhất

## DANH MỤC TÀI LIỆU THAM KHẢO

1. Kiến thức cơ bản về MySQL: <https://dev.mysql.com/doc/>
2. Kiến thức về NestJS: <https://docs.nestjs.com/>
3. Kiến thức về TypeORM : <https://typeorm.io/>
4. Kiến thức về ReactJS: <https://reactjs.org/>