|  |
| --- |
| **BỘ THÔNG TIN VÀ TRUYỀN THÔNG**  TRẦN THƯ ĐẠT MSSV: N19DCCN036 CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM 2019 – 2024 Lớp: D19CQCNPM01-N  **HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**  ---    **ĐỒ ÁN THỰC TẬP TỐT NGHIỆP**  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 7/2023  TP. HCM  2023  20… |

|  |
| --- |
| **BỘ THÔNG TIN VÀ TRUYỀN THÔNG**  **HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**  ---    **ĐỒ ÁN THỰC TẬP TỐT NGHIỆP**  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 7/2023 |

**ĐỀ CƯƠNG**

**Nội dung lý thuyết :**

* Tìm hiểu bối cảnh và mong muốn chung về công cụ tìm kiếm đặt mua sách.
* Tìm hiểu các kỹ thuật, công nghệ hỗ trợ xây dựng ứng dụng: mô hình Boolean (để tìm kiếm), ReactJS, SpringBoot, RESTapi, MySQL.

**Nội dung thực hành :**

Phân tích, thiết kế và xây dựng ứng dụng, có các chức năng:

1.Tìm kiếm sách dựa trên mô hình Boolean:

a) theo bộ tiêu chí (tên sách, tác giả, thể loại, nhà xuất bản, năm xuất bản)

b) theo tag (custom keywords, là các từ khóa cho nội dung của sách),

c) theo số lượng mua và đánh giá sao. Mỗi tiêu chí tìm kiếm được khách cho trọng số. Kết quả tìm kiếm được xếp ưu tiên theo cách tính của mô hình.

2.Đặt mua sách: có giới hạn phạm vi bán, tiền mua sách có ưu đãi theo chính sách phân loại khách hàng (thân thiết, vip,..) và có tính phí giao hàng (theo trọng lượng và khu vực).

**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 Thư Đạt

**Mục lục**

[CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI 8](#_Toc142421839)

[1.1. Tên đề tài 8](#_Toc142421840)

[1.2. Loại đề tài 8](#_Toc142421841)

[1.3. Mục đích 8](#_Toc142421842)

[1.4 Mục tiêu 8](#_Toc142421843)

[CHƯƠNG 2: CƠ SỞ KHOA HỌC ĐỀ TÀI 10](#_Toc142421844)

[2.1 Giới thiệu về Visual studio code 10](#_Toc142421845)

[2.2 Giới thiệu về MySQL 10](#_Toc142421846)

[2.3 Giới thiệu về Typescript : 11](#_Toc142421847)

[2.4 Tổng quan về NestJS : 12](#_Toc142421848)

[2.3.4 Tổng quan về TypeORM: 12](#_Toc142421849)

[2.4 Giới thiệu về React JS 13](#_Toc142421850)

[2.5 Giới thiệu về Restful 14](#_Toc142421851)

[2.6 Tìm kiếm dựa trên mô hình Boolean 14](#_Toc142421852)

[2.7 MapBox 15](#_Toc142421853)

[CHƯƠNG 3: PHÂN TÍCH HỆ THỐNG 16](#_Toc142421854)

[3.1 Bối cảnh/ hiện trạng của hệ thống 16](#_Toc142421855)

[3.2 Lược đồ usecase cho phân tích 20](#_Toc142421856)

[3.3 Định nghĩa yêu cầu và ràng buộc đối với phần mềm 26](#_Toc142421857)

[CHƯƠNG 4: THIẾT KẾ HỆ THỐNG 29](#_Toc142421858)

[4.1. Lược đồ usecase cho thiết kế : 29](#_Toc142421859)

[4.2 Thiết kế các thành phần của phần mềm 32](#_Toc142421860)

[4.3. Thiết kế cơ sở dữ liệu và ràng buộc 44](#_Toc142421861)

[4.4. Thiết kế dữ liệu 47](#_Toc142421862)

[DANH MỤC TÀI LIỆU THAM KHẢO 49](#_Toc142421863)

**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 bảng, sơ đồ, hình**

[Hình 1. Usecase khách hàng tìm sách 20](#_Toc140872841)

[Hình 2. Usecase khách hàng đặt sách 22](#_Toc140872842)

[Hình 3. Usecase khách hàng huỷ đơn 23](#_Toc140872843)

[Hình 4. Usecase nhân viên tạo khuyến mãi 25](#_Toc140872844)

[Hình 5. Usecase nhân viên tạo hoá đơn 26](#_Toc140872845)

[Hình 6. Nhân viên giao hàng **Error! Bookmark not defined.**](#_Toc140872846)

# 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.

# 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 cải 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

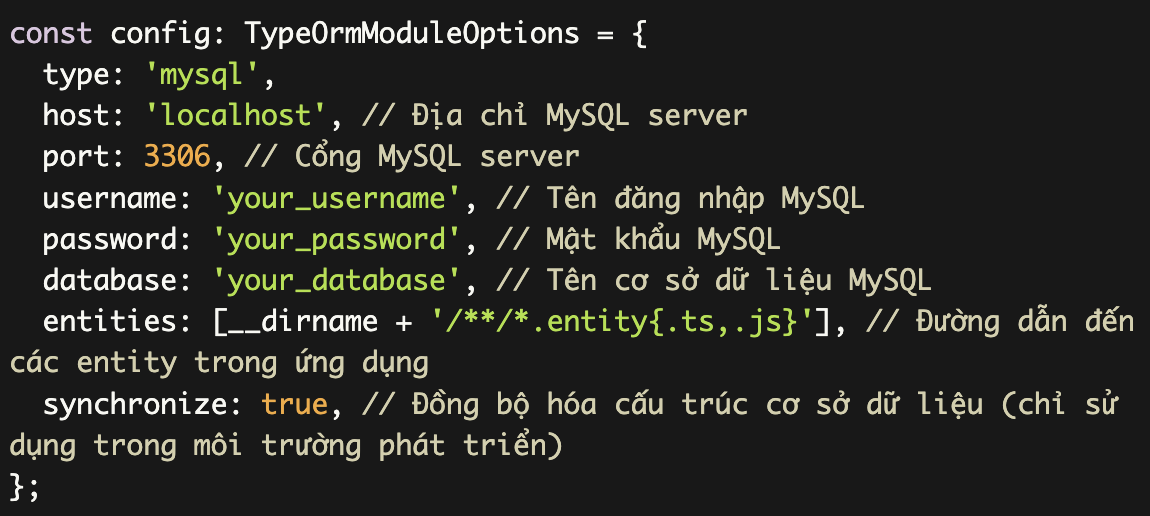
**MySQL là một hệ thống quản trị cơ sở dữ liệu mã nguồn mở (gọi tắt là RDBMS) hoạt động theo mô hình client-server**. Với **RDBMS** là viết tắt của **Relational Database Management System**. **MySQL** được **tích hợp apache**, **PHP**. **MySQL** quản lý dữ liệu thông qua các cơ sở dữ liệu. Mỗi cơ sở 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 :



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



## 2.3 Giới thiệu về Typescript :

### 2.3.1 Tổng quan về Typescript :

[**TypeScript**](https://topdev.vn/blog/typescript-la-gi/) 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 dàng để giải quyết vấn đề này.

## 2.3.4 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.

## 2.4 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 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 khẳng 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.

## 2.5 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.

## 2.6 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 trọng số của các tiêu chí từ cao xuống thấp.

## 2.7 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.

# 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. | Thêm 3 nghìn đồng |
| Miễn phí nếu tổng giá sách giá trị từ 300 nghìn đồng trở lên và trọng lượng đơn hàng dưới 0.6kg |
| 16km-25km | 30 nghìn đồng nếu như nếu tổng giá sách dưới 600 nghìn đồng. | Thêm 5 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 |
| 25km-35km | 40 nghìn đồng nếu tổng giá sách dưới 1 triệu đồng. | Thêm 7 nghìn 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 |

Nếu như 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:

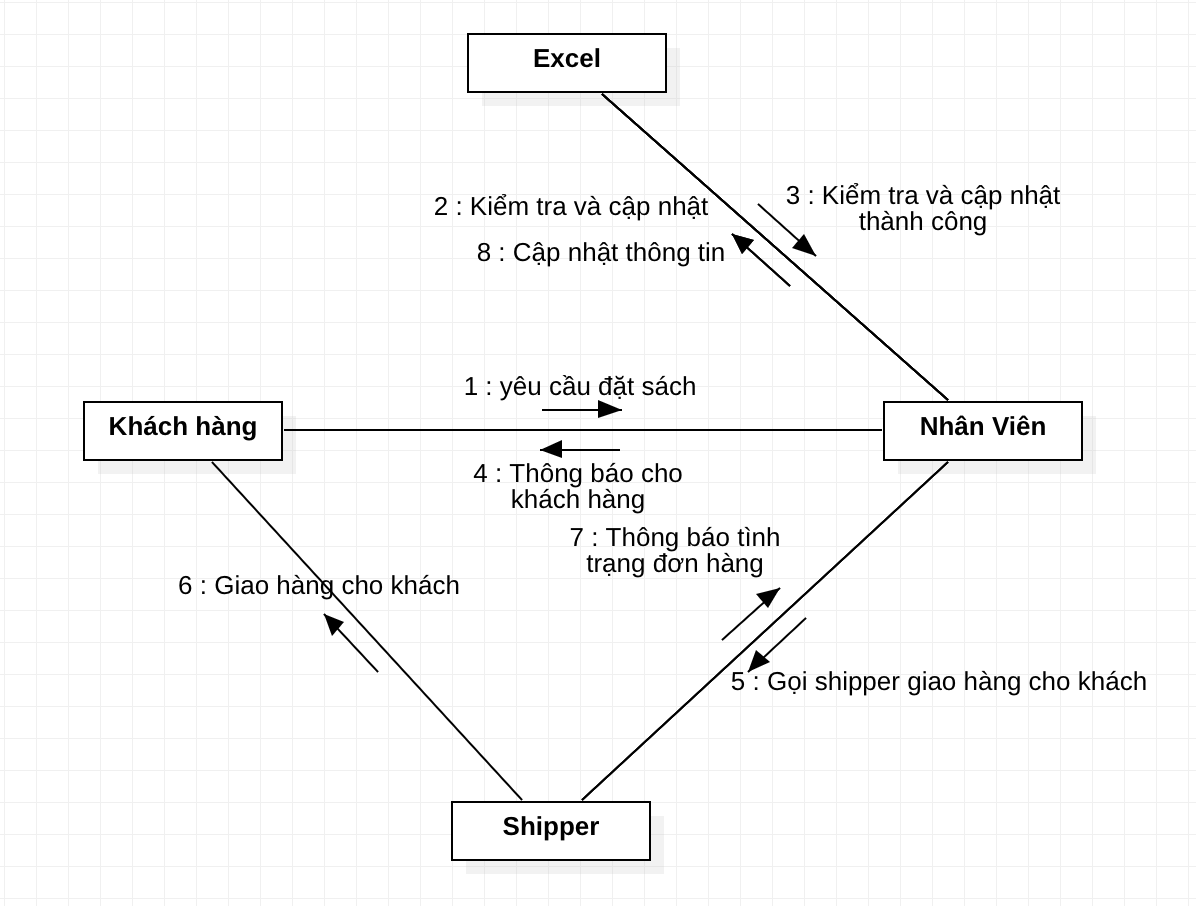
* Trường hợp đầu sách đó có và đủ số lượng, nhân viên kiểm tra loại khách hàng (thân thiết, vip, …) để áp dụng các khuyến mãi. Nhân viên sẽ gọi lại cho khách hàng để thông báo hoá đơn và đặt cọc. Sau đó nhân viên sẽ cập nhật lại số lượng đầu sách đó ở excel và làm hoá đơn cho đơn hàng của khách hàng 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 dích 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 số ảnh hưởng tới nhà sách.

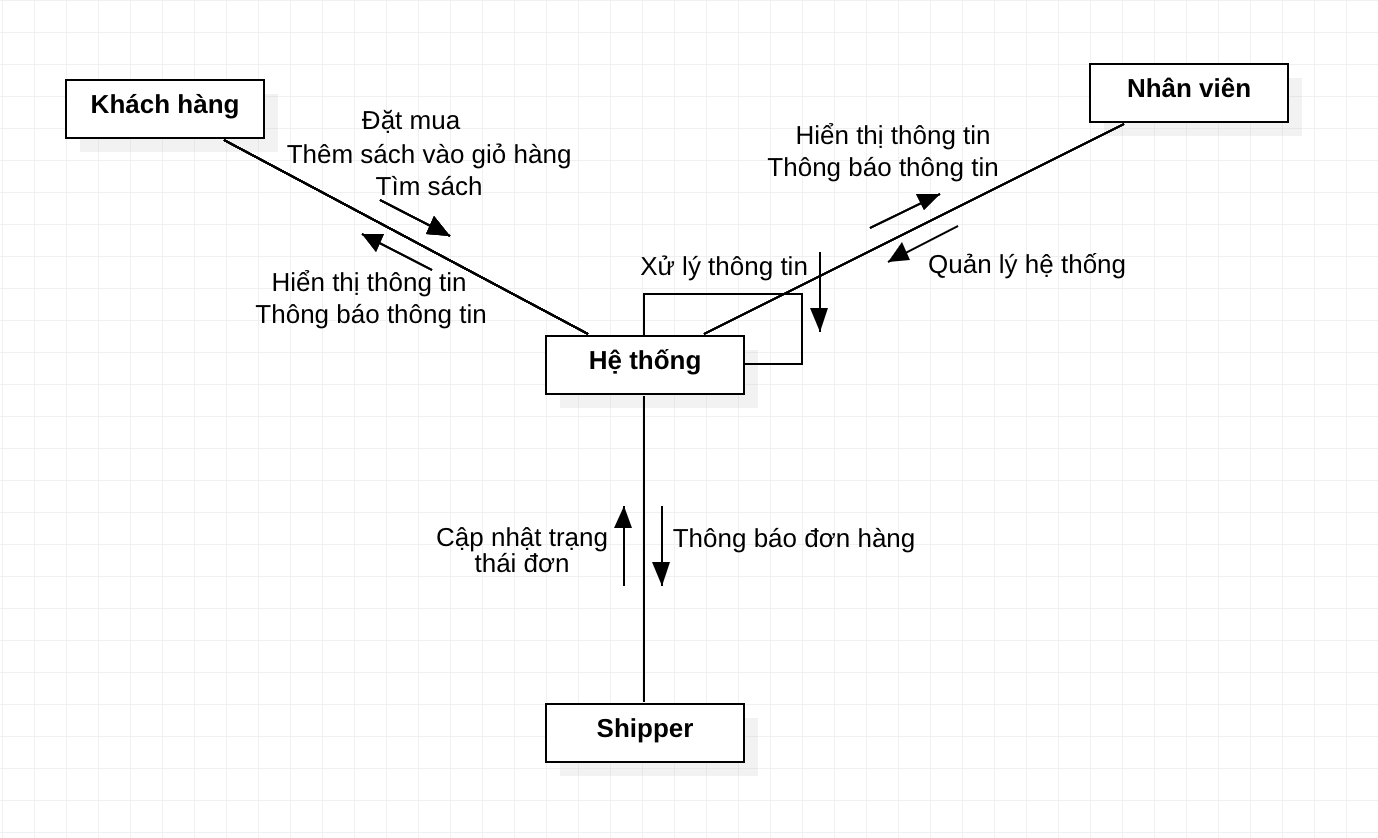
### 3.1.2.1 Biểu đồ cộng tác hiện trạng trước khi sử dụng phần mềm



### 3.1.3 Đề xuất

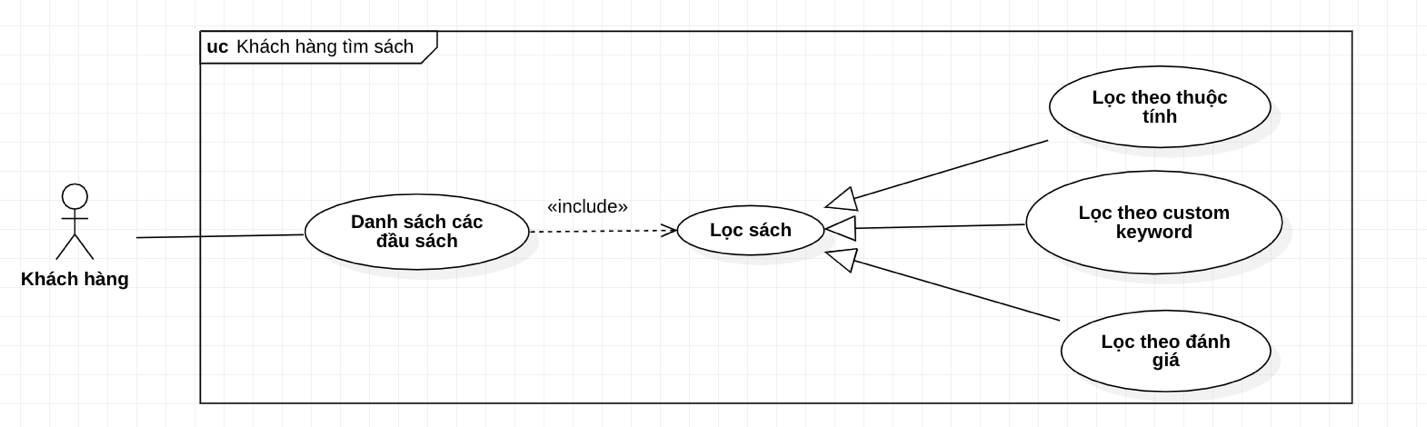
Xây dựng một website tiện dụng, thân thiện với người dùng để khách hàng có thể tiết kiệm thời gian trong quá trình tìm kiếm và đặt mua. Thay vì phải gọi cho nhân viên cung cấp rất nhiều thông tin và phải đợi nhân viên tính toán tìm kiếm và gọi lại cho mình, thì với website này người dùng sau khi tìm 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



## 3.2 Lược đồ usecase cho phân tích

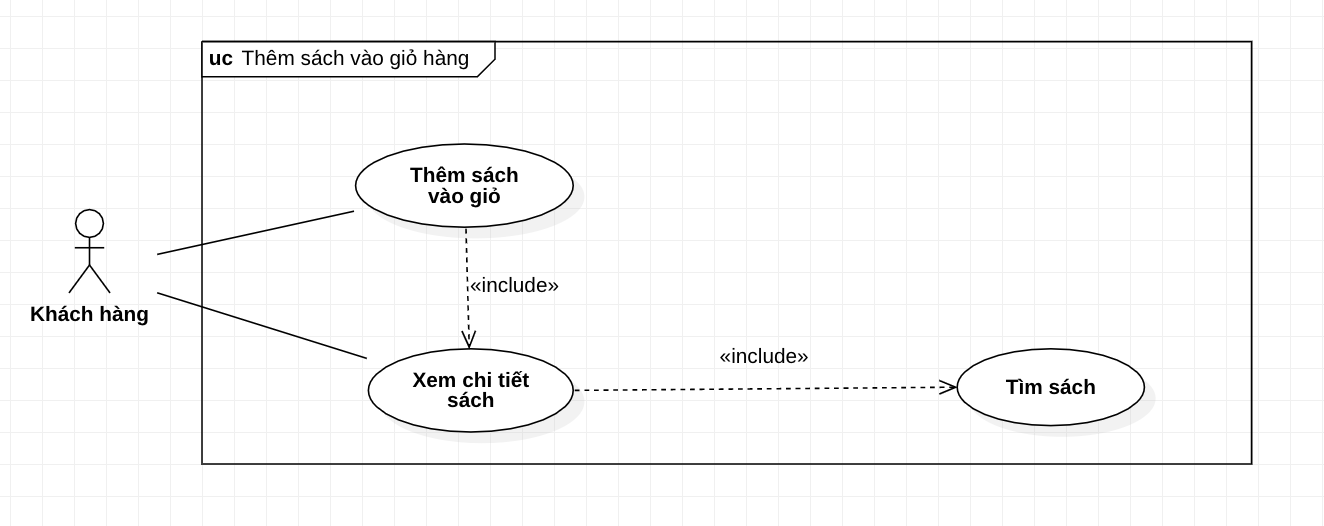
### 3.2.1 Usecase khách hàng tìm sách



Hình . Usecase khách hàng tìm sách

|  |  |
| --- | --- |
| **USECASE NAME** | Khách hàng tìm sách |
| **USECASE ID** | UC-01 |
| **SCENARIO** | Khách hàng tìm sách trên website. |
| **ACTOR** | Khách hàng. |
| **DESCRIPTION** | Khi khách hàng muốn tìm được đầu sách mà họ mong muốn để mua trên website. |
| **TRIGGER** | Khách hàng tìm kiếm sách. |
| **PRECONDITION** | Khách hàng nhập thông tin vào bộ lọc của hệ thống. |
| **POSTCONDITION** | Khách hàng tìm được đầu sách mong muốn từ website. |
| **MAIN FLOW** | 1. Khách hàng nhấn vào thanh tìm kiếm.  2. Hệ thống hiện lên giao diện cho người dùng bộ lọc theo các tiêu chí.  3. Khách hàng điền thông tin cho bộ lọc.  4. Hệ thống kiếm tra thông tin và lọc theo thông tin mà người dùng cung cấp.  5. Hệ thống chuyển đến giao diện những đầu sách thoả mãn từ bộ lọc của khách hàng. |
| **EXCEPTIONS** | 4a. 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 3] |

### 3.2.2 Usecase thêm sách vào giỏ hàng



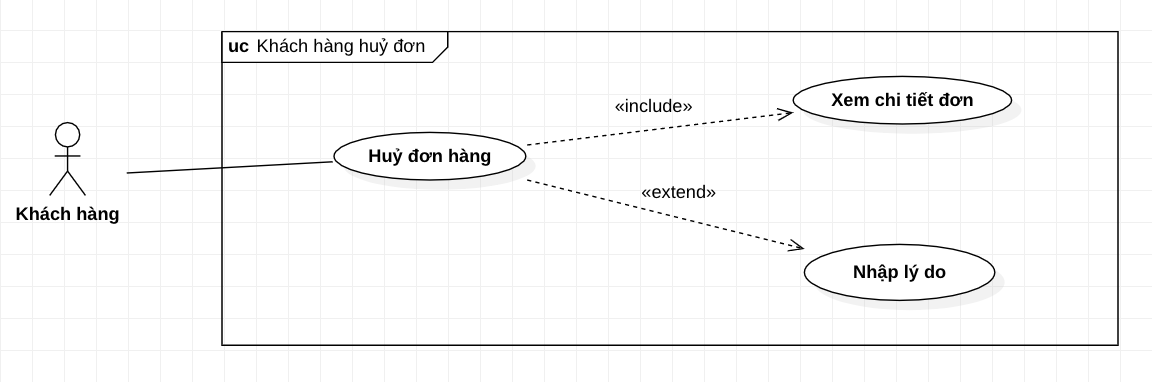
|  |  |
| --- | --- |
| **USECASE NAME** | Thêm sách vào giỏ hàng. |
| **USECASE ID** | UC-02 |
| **SCENARIO** | Khách hàng thêm sách vào giỏ hàng trên website. |
| **ACTOR** | Khách hàng. |
| **DESCRIPTION** | Khi khách hàng tìm được đầu sách ưng ý, khách hàng thêm sách vào giỏ hàng. |
| **TRIGGER** | Khách hàng thực hiện chức năng thêm sách vào giỏ hàng. |
| **PRECONDITION** | Khách hàng đăng nhập vào hệ thống. |
| **POSTCONDITION** | Thêm đầu sách vào giỏ hàng cho khách hàng. |
| **MAIN FLOW** | 1. Khách hàng tìm kiếm sách (UC-01).  2. Khách hàng bấm vào đầu sách để xem chi tiết thông tin từ những đầu sách đã lọc được từ UC-01.  3. Hệ thống chuyển đến trang chi tiết cho khách hàng.  4. Khách hàng chọn số lượng và bấm thêm vào giỏ hàng.  5. Hệ thống kiểm tra thông tin về số lượng có hợp lệ hay không và thông báo cho khách hàng.  6. Thêm đầu sách vào giỏ hàng cho khác hàng. |
| **EXCEPTIONS** | 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].  5b. Khi người dùng chọn số lượng mà số lượng đó không đủ vì đã có người đặt trước. Hê thống hiển thị thông báo và cập nhật giao diện hiển thị số lượng hiện tại. Quay lại [Main Flow 4]. |

### 3.2.3 Usecase khách hàng đặt sách

Hình . Usecase khách hàng đặt sách

|  |  |
| --- | --- |
| **USECASE NAME** | Khách hàng đặt sách |
| **USECASE ID** | UC-03 |
| **SCENARIO** | Khách hàng đặt sách trên website. |
| **ACTOR** | Khách hàng. |
| **DESCRIPTION** | Khi khách hàng muốn đặt sách qua website, khách hàng chọn sách cần mua và tiền hàng gửi yêu cầu đặt hàng. |
| **TRIGGER** | Khách hàng thực hiện gửi yêu cầu đặt hàng. |
| **PRECONDITION** | Khách hàng đăng nhập vào hệ thống. |
| **POSTCONDITION** | Thêm hoá đơn cho khách hàng với trạng thái là chờ duyệt. |
| **MAIN FLOW** | 1. Khách hàng bấm vào giỏ hàng, xem thông tin giỏ hàng.  2. Hệ thống hiển thị giao diện giỏ hàng với từng đầu sách và số lượng.  3. Khách hàng xác nhận số lượng và xác nhận nơi giao hàng.  4. Hệ thống hiển thị số tiền mà khách hàng phải trả.  5. Khách hàng bấm nút xác nhận đặt hàng.  6. Hệ thống kiểm tra và thông báo về cho khách hàng. |
| **EXCEPTIONS** | 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. Hê thống hiển thị thông báo và cập nhật giao diện hiển thị số lượng hiện tại. Quay lại [Main Flow 3]. |

### 3.2.4 Khách hàng huỷ đơn



Hình . Usecase khách hàng huỷ đơn

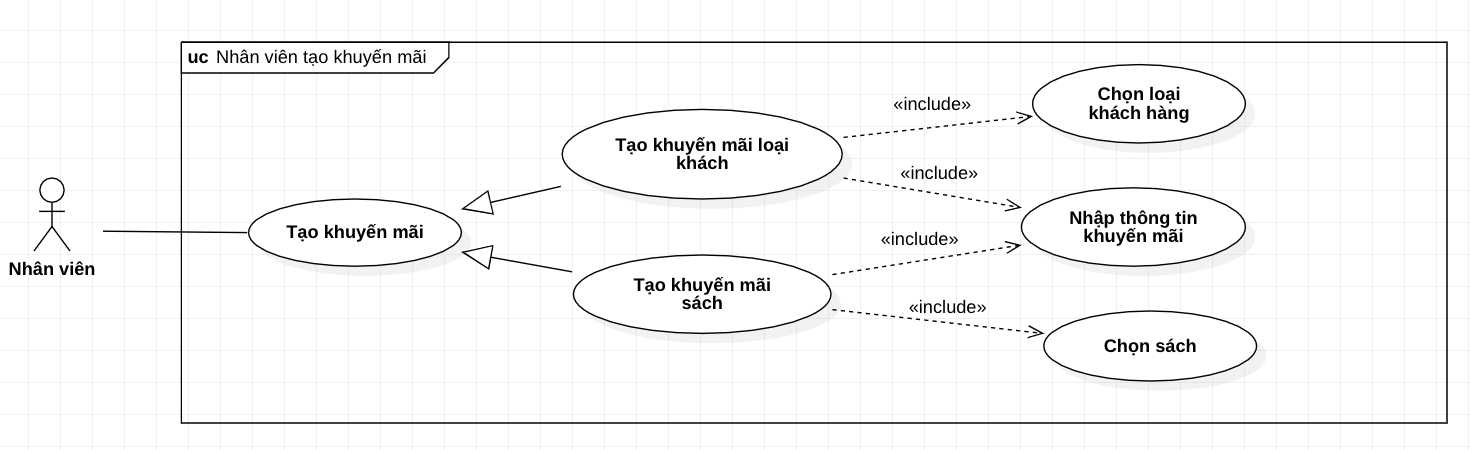
|  |  |
| --- | --- |
| **USECASE NAME** | Khách hàng huỷ đơn |
| **USECASE ID** | UC-04 |
| **SCENARIO** | Khách hàng huỷ đơn trên website. |
| **ACTOR** | Khách hàng. |
| **DESCRIPTION** | Khi khách hàng muốn huỷ đơn hàng mà mình đặt. |
| **TRIGGER** | Khách hàng thực thi chức năng huỷ đơn hàng. |
| **PRECONDITION** | 1.Khách hàng đã đăng nhập vào hệ thống.  2. Đơn hàng đó đang ở trạng thái chờ duyệt. |
| **POSTCONDITION** | Đơn hàng được huỷ thành công. |
| **MAIN FLOW** | 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 muốn hủy. Chỉ những đơn hàng mà trạng thái là chờ duyệt mới có nút huỷ đơn.  4. Khách hàng bấm huỷ đơn.  5. Hệ thống thực hiện kiểm tra và huỷ đơn cho khách. Sau đó hiển thị thông báo. |
| **EXCEPTIONS** | 5a. Khi nhân viên vừa ấn xác nhận đơn hàng, đơn hàng chuyển qua trạng thái đang giao mà giao diện người dùng chưa cập nhật. Hệ thống sẽ hiển thị thông báo đơn hàng đang giao không được huỷ.  Kết thúc Usecase |

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

### 

|  |  |
| --- | --- |
| **USECASE NAME** | Thêm thông tin nhận hàng |
| **USECASE ID** | UC-05 |
| **SCENARIO** | Khách hàng thêm vị trí nhận hàng trên website. |
| **ACTOR** | Khách hàng. |
| **DESCRIPTION** | Khi khách hàng muốn thêm một vị trí khác để nhận hàng. |
| **TRIGGER** | Khách hàng thực thi chức năng thêm thông tin nhận hàng. |
| **PRECONDITION** | Khách hàng đã đăng nhập vào hệ thống. |
| **POSTCONDITION** | Thêm vị trí giao hàng mới thành công. |
| **MAIN FLOW** | 1. Khách hàng vào mục thông tin cá nhân.  2. Hệ thống chuyển đến trang có giao diện là thông tin của khách hàng bao gồm cả thông tin nhận hàng.  3. Khách hàng nhấn vào nút thêm thông tin nhận hàng.  4. Hệ thống mở giao diện nhập thông tin cho khách hàng.  5. Khách hàng nhập đầy đủ thông tin.  6. 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. |
| **EXCEPTIONS** | 6a. 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 khách.  Quay lại [Main Flow 5] |

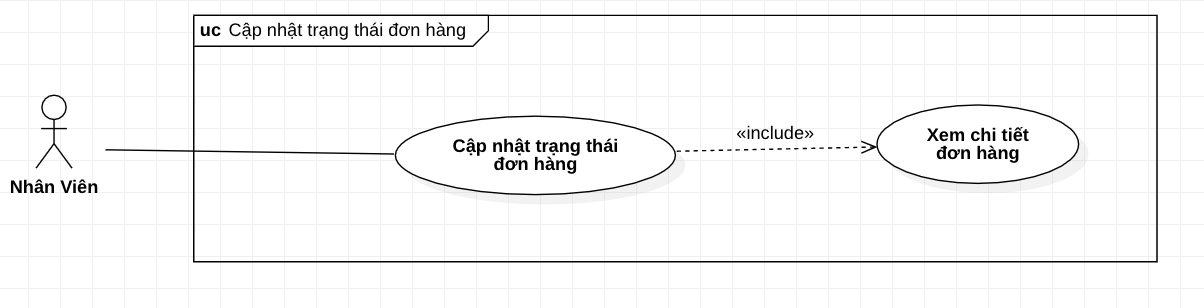
### 3.2.6 Nhân viên tạo khuyến mãi



Hình . Usecase nhân viên tạo khuyến mãi

|  |  |
| --- | --- |
| **USECASE NAME** | Nhân viên tạo khuyến mãi. |
| **USECASE ID** | UC-06 |
| **SCENARIO** | Nhân viên tạo khuyến mãi cho khách hàng. |
| **ACTOR** | Nhân viên. |
| **DESCRIPTION** | Khi nhà sách muốn tạo ra một đợi khuyến mãi giảm giá cho khách hàng theo loại khách hàng hoặc theo từng đầu sách. |
| **TRIGGER** | Nhân viên thực thi chức năng tạo khuyến mãi trên hệ thống. |
| **PRECONDITION** | Nhân viên đăng nhập vào hệ thống thành công. |
| **POSTCONDITION** | Thêm khuyến mãi thành công. |
| **MAIN FLOW** | 1. Nhân viên chọn vào mục quản lý khuyến mãi và chọn thêm khuyến mãi trên giao diện hệ thống.  2. Hệ thống hiển thị giao diện thêm khuyến mãi với 2 lựa chọn là tạo theo loại khách hàng hoặc theo đầu sách.  3. Nhân viên nhập đầy đủ thông tin và bấm vào nút tạo khuyến mãi.  4. 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. |
| **EXCEPTIONS** | 4a. Dữ liệu nhân viên nhập không thoả 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] |

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



Hình . Usecase nhân viên tạo hoá đơn

|  |  |
| --- | --- |
| **USECASE NAME** | Nhân viên cập nhật trạng thái đơn. |
| **USECASE ID** | UC-07 |
| **SCENARIO** | Nhân viên đổi trạng thái trong trường hợp có đơn hàng mới và đã giao thành công. |
| **ACTOR** | Nhân viên. |
| **DESCRIPTION** | 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. |
| **TRIGGER** | Nhân viên thực thi chức năng cập nhật trạng thái. |
| **PRECONDITION** | Nhân viên đăng nhập thành công vào hệ thống. |
| **POSTCONDITION** | Đơn hàng được chuyển đổi trạng thái sang đã duyệt hoặc đã giao hoặc đã huỷ. |
| **MAIN FLOW** | 1. Nhân viên bấm vào mục quản lý đơn hàng.  2. Hệ thống chuyển sang giao diện các đơn hàng hiện có của hệ thống.  3. Nhân viên chọn vào đơn hàng cần thay đổi trạng thái.  4. Hệ thống hiển thị chi tiết đơn hàng đó.  5. Nhân viên đổi trạng thái đơn hàng.  6. Hệ thống thông báo lỗi hoặc thành công cho nhân viên. |
| **EXCEPTIONS** |  |

## 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 | Nhân viên xác nhận yêu cầu đặt sách của khách hàng. | Nhân viên |
| B03.2 | 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. |
| B03.3 | Khi đặt sách hệ thống cần hiển thị rõ ràng số tiền của sách là bao nhiêu, phí ship là bao nhiêu và khoảng thời gian dự kiến giao hàng. | Chủ nhà sách |
| UC-04 | B04.1 | Huỷ đơn khách hàng nhập lý do huỷ đơn để có vấn đề thì nhà sách sẽ khách phục. | Khách hàng. |
| UC-05 | B05.1 | Khách hàng cần cung cấp chính xác thông tin giao hàng. | Khách hàng. |
| 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 |
| UC-08 | B08.1 | Phải giao đúng địa chỉ mà hệ thống hiển thị, cung cấp. | Nhân viên. |

### 3.3.2 Yêu cầu từ môi trường vận hàng (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. |
| F02.2 | Hệ thống có chức năng tự động tính tiền sách đã áp dụng khuyến mãi. | Chủ nhà sách. |
| Chưa đủ, bổ sung sau | | | |

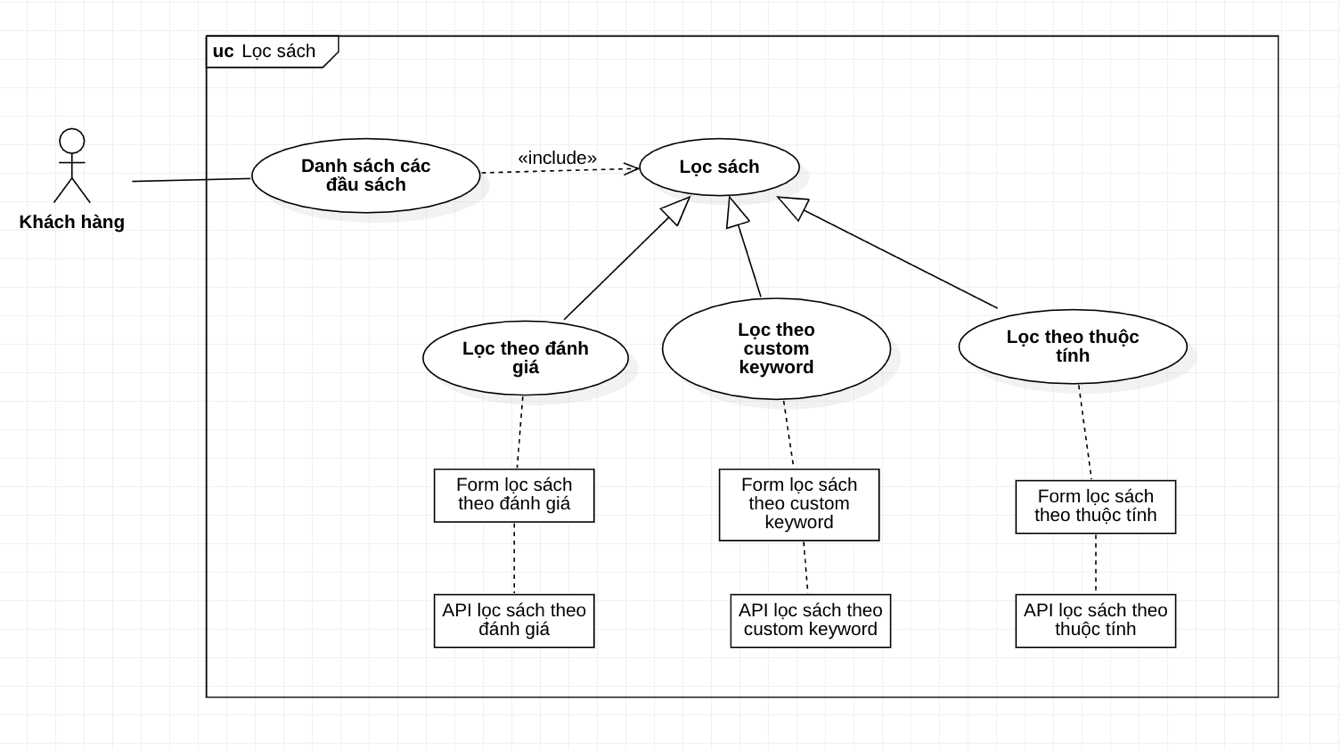
### 3.3.2 Yêu cầu từ môi trường vận hàng (operation)

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |

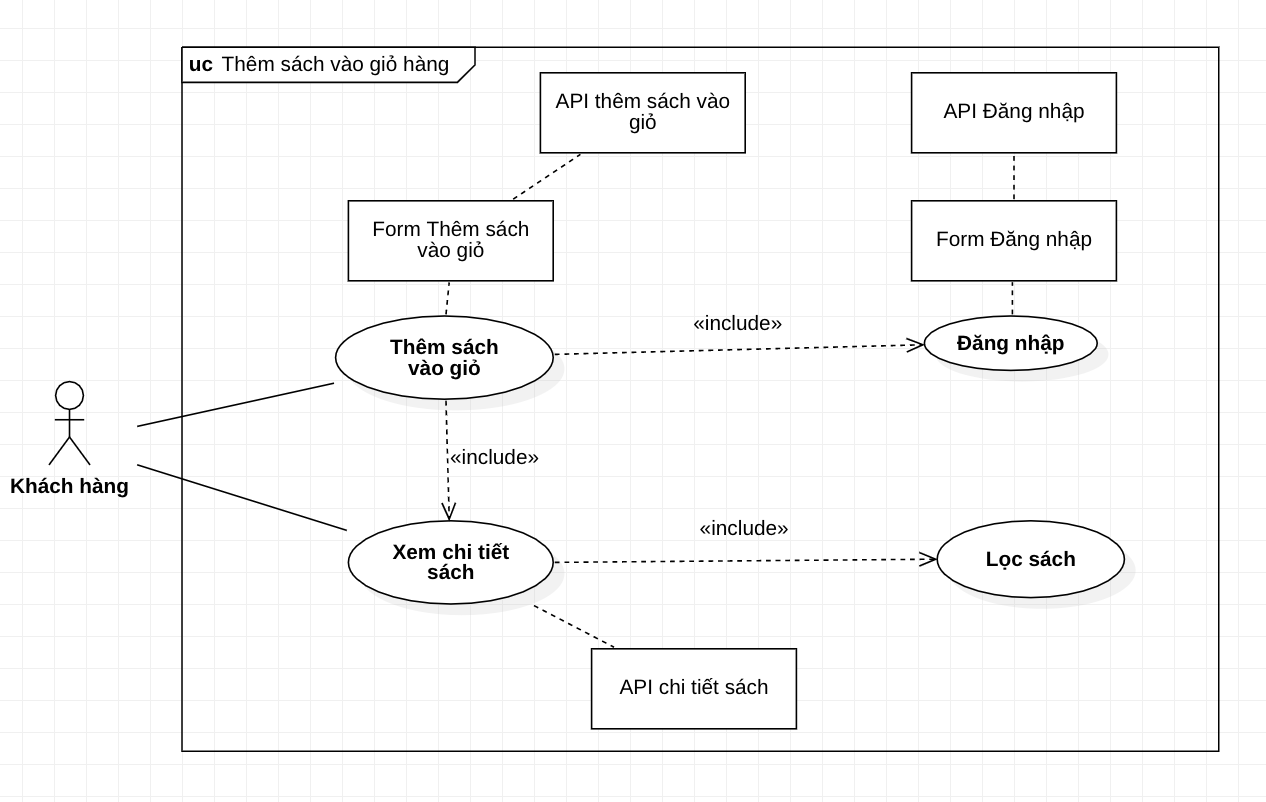
# CHƯƠNG 4: THIẾT KẾ HỆ THỐNG

## 4.1. Lược đồ usecase cho thiết kế :

### 4.1.1 Usecase khách hàng tìm sách :



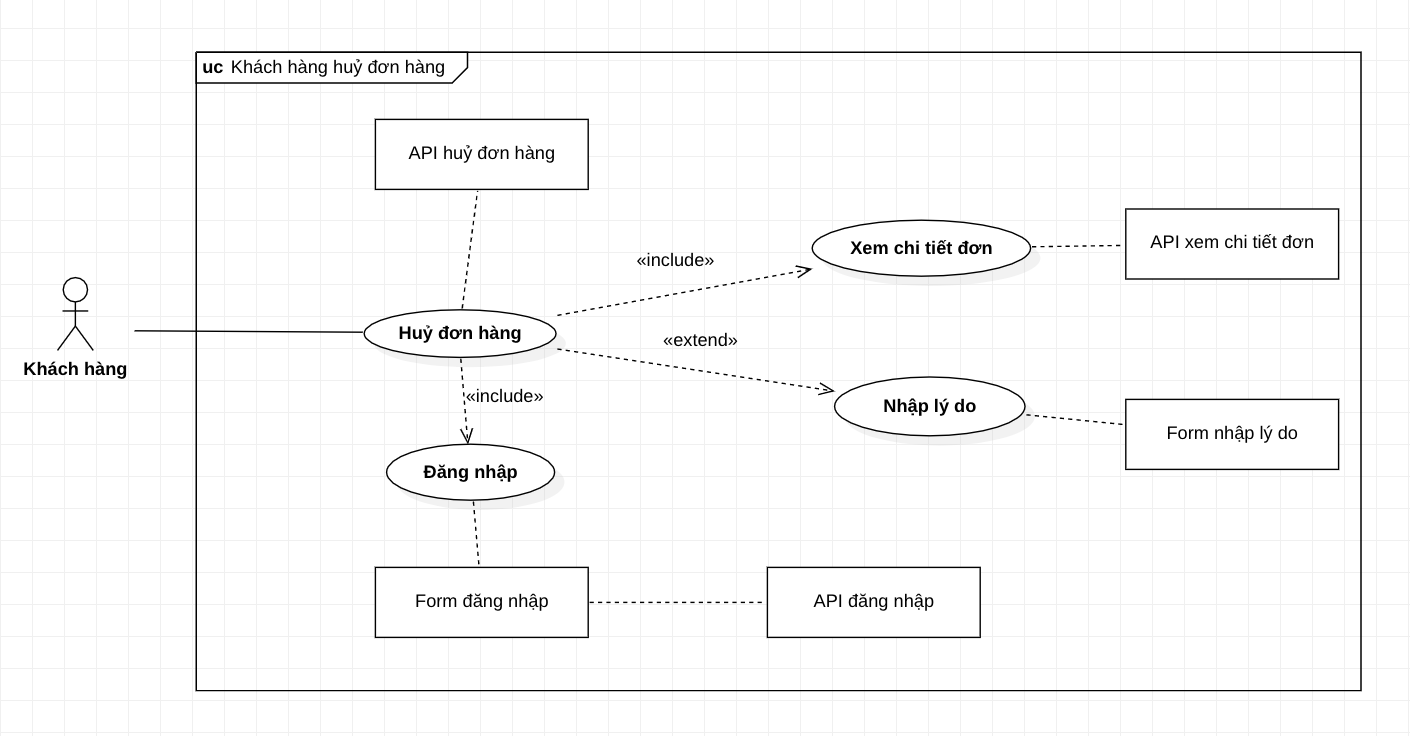
4.1.2 Usecase khách hàng thêm sách vào giỏ hàng :



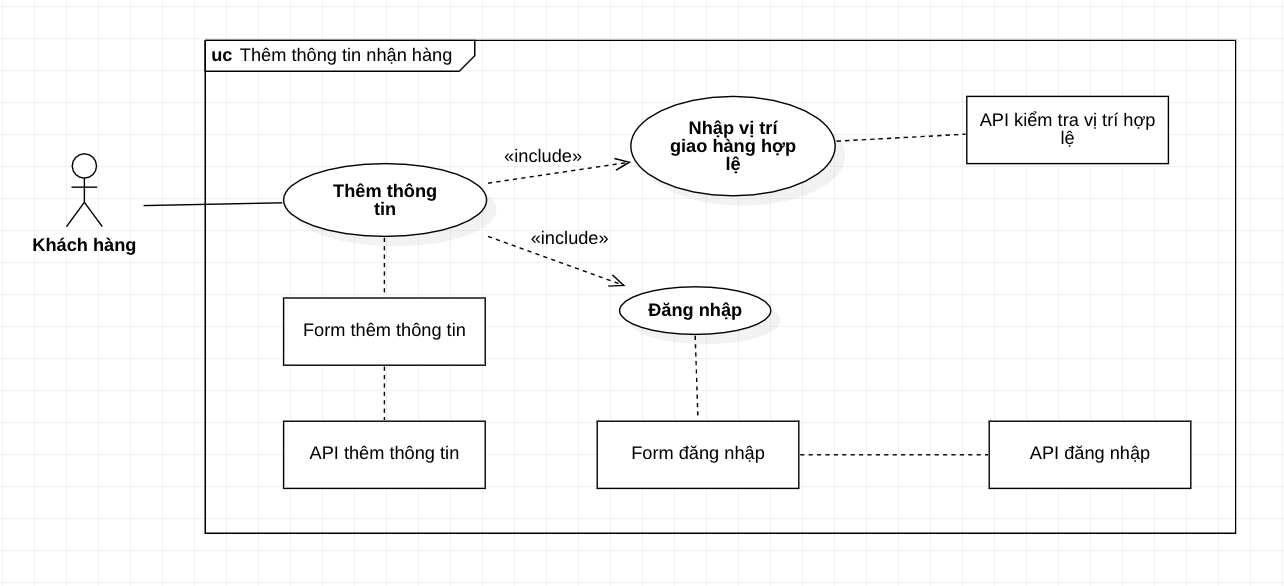
### 4.1.3 Usecase khách hàng đặt sách :

### 

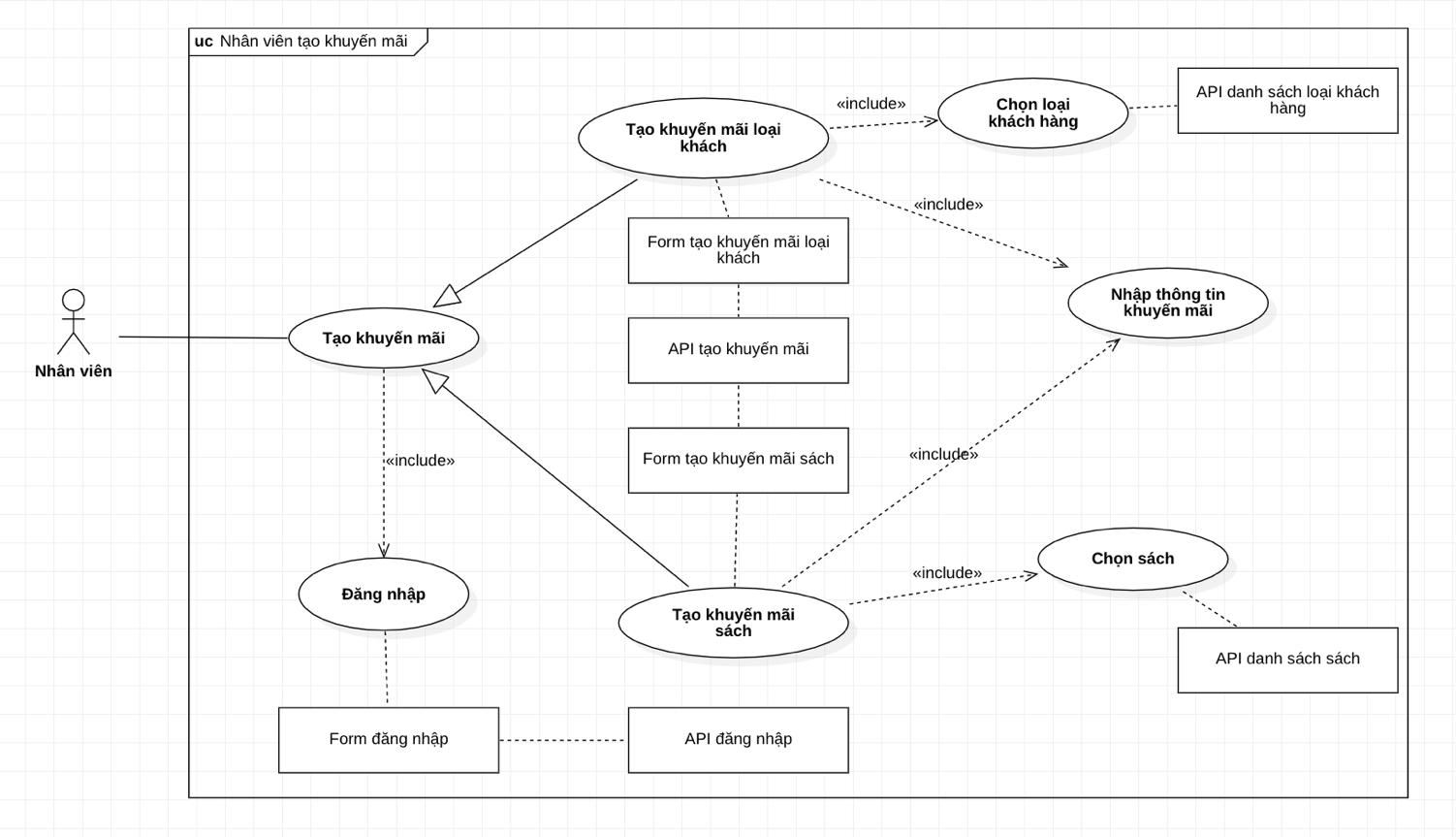
### 4.1.4 Usecase khách hàng huỷ đơn hàng :



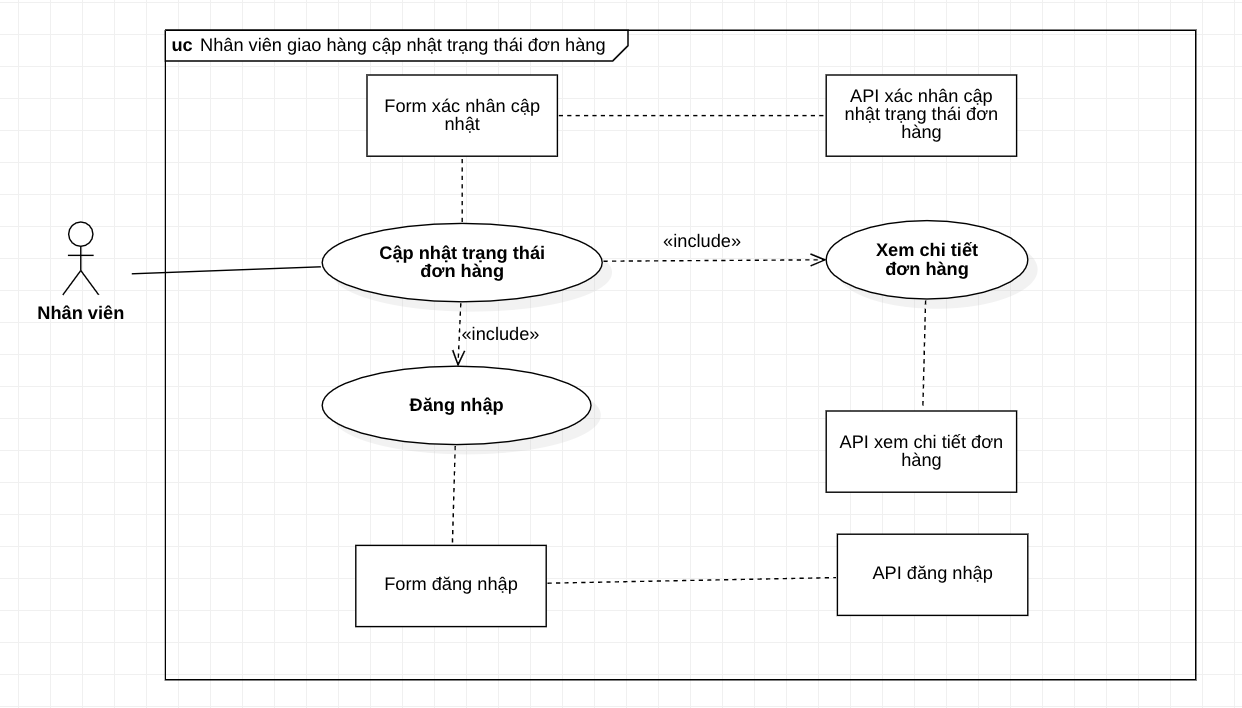
### 4.1.5 Usecase khách hàng thêm thông tin nhận hàng :



4.1.6 Usecase nhân viên tạo khuyến mãi :



### 4.1.7 Usecase nhân viên cập 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 | Form :   * + - * Form lọc sách theo các thuộc tính của sách(F01)       * Form lọc sách theo custom keywords(F02)       * Form lọc theo số lượng mua và đánh giá sao(F03)   API :   * + - * API lọc sách theo các thuộc tính của sách(API01)       * API lọc sách theo custom keyword(API02)       * API lọc sách theo số lượng mua và đánh giá (API03) |

- UC-02 / Khách hàng

|  |  |
| --- | --- |
| Vai trò | Tác nhân sử dụng usecase này để thêm sách vào giỏ hàng |
| Tương tác | Form :   * + - * Form thêm sách vào giỏ hàng(F04)   API :   * + - * API xem chi tiết sách(API04)       * API thêm sách vào giỏ hàng (API05) |

- UC-03 / Khách hàng

|  |  |
| --- | --- |
| Vai trò | Tác nhân sử dụng usecase này để đặt sách |
| Tương tác | Form :   * + - * Form xác nhận thông tin đặt sách(F05)   API :   * + - * API thay đổi thông tin nhận sách(API06)       * API xem giỏ hàng (API07)       * API xoá đầu sách khỏi giỏ hàng(API08)       * API chỉnh sửa số lượng đầu sách(API09)       * API đặt sách (API10) |

- UC-04 / Khách hàng

|  |  |
| --- | --- |
| Vai trò | Tác nhân sử dụng usecase này để huỷ đơn hàng |
| Tương tác | Form :   * + - * Form nhập lý do (F06)   API :   * + - * API xem chi tiết đơn hàng (API11)       * API huỷ đơn hàng (API12) |

-UC-05 / Khách hàng

|  |  |
| --- | --- |
| Vai trò | Tác nhân sử dụng usecase này để thêm thông tin nhận hàng |
| Tương tác | Form :   * + - * Form thêm thông tin(F07)   API :   * + - * API thêm thông tin nhận hàng(API13)       * API kiểm tra vị trí hợp lệ(API14) |

-UC-06 / Nhân viên

|  |  |
| --- | --- |
| Vai trò | Tác nhân sử dụng usecase này để thêm khuyến mãi |
| Tương tác | Form :   * + - * Form tạo khuyến mãi theo loại khách hàng(F08)       * Form tạo khuyến mãi theo đầu sách(F09)   API :   * + - * API danh sách loại khách hàng(API15)       * API danh sách các đầu sách (API16)       * API tạo khuyến mãi(API17) |

-UC-07 / Nhân viên

|  |  |
| --- | --- |
| Vai trò | Tác nhân sử dụng usecase này để thay đổi trạng thái đơn hàng |
| Tương tác | Form :   * + - * Form cập nhật trạng thái.(F10)   API :   * + - * API xem chi tiết đơn hàng (API11)       * API xác nhận cập nhật trạng thái đơn hàng(API18) |

### 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 | * + - * Textbox tìm kiếm tên sách.       * Textbox tên tác giả.       * Textbox năm xuất bản.       * Dropdown thể loại : cho phép chọn loại sách.       * Dropdown nhà xuất bản : cho phép chọn nhà xuất bản.       * Button tìm kiếm : Gửi và yêu cầu trả lại các đầu sách ở màn hình hiện tại. |

- UC-01/F02

|  |  |
| --- | --- |
| Tên Form | Form lọc sách theo custom keywords |
| Giao tiếp với user | Khách hàng |
| Control | * + - * Textbox nhập custom keyword.       * Button tìm kiếm : Gửi và yêu cầu trả lại các đầu sách ở màn hình hiện tại. |

- UC-01/F03

|  |  |
| --- | --- |
| Tên Form | Form lọc sách theo lượng mua và đánh giá sao |
| Giao tiếp với user | Khách hàng |
| Control | * + - * Textbox nhập tên sách.       * Dropdown chọn chế độ : theo lượng mua hoặc theo đánh giá sao.       * Button tìm kiếm : Gửi và yêu cầu trả lại các đầu sách ở màn hình hiện tại. |

- UC-02/F04

|  |  |
| --- | --- |
| Tên Form | Form thêm sách vào giỏ hàng |
| Giao tiếp với user | Khách hàng |
| Control | * + - * Textbox số lượng : chỉ cho phép nhập số.       * Button thêm vào giỏ : Gửi yêu cầu và thông báo lại màn hình hiện tại. |

- UC-03/F05

|  |  |
| --- | --- |
| Tên Form | Form xác nhận thông tin đặt sách |
| Giao tiếp với user | Khách hàng |
| Control | * + - * Button huỷ : quay về trang trước đó.       * Button đặt hàng: xác nhận thông tin đặt sách và hiển thị thông báo. |

- UC-04/F06

|  |  |
| --- | --- |
| Tên Form | Form nhập lý do |
| Giao tiếp với user | Khách hàng |
| Control | * + - * Textbox nhập lý do huỷ đơn.       * Button huỷ : quay về trang trước đó.       * Button đặt hàng: xác nhận huỷ đơn và hiển thị thông báo. |

- UC-05 / F07

|  |  |
| --- | --- |
| Tên Form | Form thêm thông tin nhận hàng |
| Giao tiếp với user | Khách hàng |
| Control | * + - * Dropdown chọn tỉnh.       * Dropdown chọn huyện.       * Dropdown chọn xã.       * Textbox nhập chi tiết.       * Textbox nhập số điện thoại.       * Textbox nhập tên người nhận.       * Button đặt hàng : Lưu thông tin và hiển thị thông báo. |

- UC-06 / F08

|  |  |
| --- | --- |
| Tên Form | Form tạo khuyến mãi theo loại khách hàng |
| Giao tiếp với user | Nhân viên |
| Control | * + - * Dropdown chọn loại khách hàng.       * Textbox nhập phần trăm giảm giá.       * Button đặt hàng : Lưu thông tin và hiển thị thông báo. |

- UC-06 / F09

|  |  |
| --- | --- |
| Tên Form | Form tạo khuyến mãi theo đầu sách |
| Giao tiếp với user | Nhân viên |
| Control | * + - * Dropdown chọn đầu sách.       * Textbox nhập phần trăm giảm giá.       * Button đặt hàng : Lưu thông tin và hiển thị thông báo. |

- UC-07 / F10

|  |  |
| --- | --- |
| Tên Form | Form cập nhật trạng thái |
| Giao tiếp với user | Nhân viên |
| Control | * + - * Dropdown chọn trạng thái đơn hàng.       * Button đặt hàng : Lưu thông tin và hiển thị thông báo. |

### c. API / Service ID

API lọc sách theo các thuộc tính của sách(API01)

* + - * Url : /book/search-attribute
      * Method : POST
      * Input :

|  |  |  |  |
| --- | --- | --- | --- |
| Tên | Kiểu dữ liệu | Bắt buộc | Mô tả |
| nameBook | String |  | Tìm kiếm theo tên |
| nameAuthor | String |  | Tìm kiếm theo tên tác giả |
| typeBook | String |  | Loại sách |
| publisher | String |  | Nhà xuất bản |
| releaseYear | Int |  | Năm xuất bản |

* + - * Output : Dữ liệu các đầu sách thoả để trả về view hiển thị cho người dùng.
      * Xử lý :
        + Gọi tới service bookService và sử dụng method searchBookByAttribute.
        + Kết quả trả về dạng Array chứa các Object là các đầu sách thoả mãn được sắp xếp giảm dần của tổng trọng số.

API lọc sách theo custom keyword(API02)

* + - * Url : /book/search-custom-keyword
      * Method : POST
      * Input :

|  |  |  |  |
| --- | --- | --- | --- |
| Tên | Kiểu dữ liệu | Bắt buộc | Mô tả |
| Custom keywords | Array[String] | X | Mảng các string là các custom keyword |

* + - * Output : Dữ liệu các đầu sách thoả mãn để trả về view hiển thị cho người dùng.
      * Xử lý :
        + Gọi tới service bookService và sử dụng method searchBookByCustomKeyWord.
        + Kết quả trả về dạng Array chứa các Object là các đầu sách thoả 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 |
|  |  |  |  |

* + - * Output :
      * Xử lý :

API xem chi tiết sách(API04)

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

API thêm sách vào giỏ hàng (API05)

* + - * 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 xác nhận thông tin nhận sách(API06)

* + - * 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 (API07)

* + - * Url : /cart
      * Method : GET
      * Output : Là array chứa các object là các đầu sách và số lượng đặt.
      * Xử lý :
        + Gọi tới CartService và sử dụng method GetCartOfCustomer.

API xoá đầu sách khỏi giỏ hàng(API08)

* + - * Url :
      * Method :
      * Query param :

|  |  |  |  |
| --- | --- | --- | --- |
| Tên | Kiểu dữ liệu | Bắt buộc | Mô tả |
|  |  |  |  |

* + - * Input :

|  |  |  |  |
| --- | --- | --- | --- |
| Tên | Kiểu dữ liệu | Bắt buộc | Mô tả |
|  |  |  |  |

* + - * Output :
      * Xử lý :

API chỉnh sửa số lượng đầu sách(API09)

* + - * Url : /cart-detail
      * Method : PATCH
      * Input :

|  |  |  |  |
| --- | --- | --- | --- |
| Tên | Kiểu dữ liệu | Bắt buộc | Mô tả |
| BookId | String | X | Mã sách trong giỏ hàng |
| Quantity | Int | X | số lượng mua |

* + - * Output : Trả về true nếu thành công và false nếu thất bại cho View hiển thị cho người dùng.
      * Xử lý :
        + Gọi tới cartDetailService sử dụng method updateCartDetail

API đặt sách (API10)

* + - * 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 (API11)

* + - * Url : /invoice
      * Method : GET
      * Output : Danh sách các đầu sách và số lượng mà khách hàng đặt.
      * Xử lý :
        + Gọi đến InvoiceService và sử dụng method getInvoiceStatusOfCustomer.

API huỷ đơn hàng (API12)

* + - * Url :
      * Method :
      * Query param :

|  |  |  |  |
| --- | --- | --- | --- |
| Tên | Kiểu dữ liệu | Bắt buộc | Mô tả |
|  |  |  |  |

* + - * Input :

|  |  |  |  |
| --- | --- | --- | --- |
| Tên | Kiểu dữ liệu | Bắt buộc | Mô tả |
|  |  |  |  |

* + - * Output :
      * Xử lý :

API thêm thông tin nhận hàng(API13)

* + - * 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 :
        + Gọi tới service receiptInfomation và sử dụng method createReceiptInfomation.
        + Tạo receipt information mới.

API danh sách loại khách hàng(API14)

* + - * Url : /receipt-information/check-distance
      * Method : GET
      * Query param :

|  |  |  |  |
| --- | --- | --- | --- |
| Tên | Kiểu dữ liệu | Bắt buộc | Mô tả |
| address | String | X | Địa chỉ cần kiểm tra |

* + - * Output : True hoặc false
      * Xử lý :
        + Gọi tới service ReceiptInformation và sử dụng method checkDistance.

API danh sách loại khách hàng(API15)

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

API danh sách các đầu sách (API16)

* + - * Url : /book
      * Method : GET
      * Output : Array chứa các object là các đầu để trả về view hiển thị.
      * Xử lý :
        + Gọi tới service bookService và sử dụng method getAllBook.
        + Kết quả trả về là array chứa các object là các đầu sách.

API tạo khuyến mãi(API17)

* + - * Url : /promotion
      * Method : POST
      * Input :

|  |  |  |  |
| --- | --- | --- | --- |
| Tên | Kiểu dữ liệu | Bắt buộc | Mô tả |
| Name | String | X | Tên khuyến mãi |
| startDate | Datetime | X | Ngày bắt đầu khuyến mãi |
| endDate | Datetime | X | Ngày kết thúc khuyến mãi |
| Reason | String | X | Khuyến mãi nhân dịp |
| staffId | String | X | Người tạo khuyến mãi |
| Type | String | X | Là khuyến mãi theo đầu sách hay theo loại sách hàng. |
| bookId | String | X | Mã đầu sách áp dụng khuyến mãi |
| customerTypeID | String | X | Mã loại khách hàng áp dụng khuyến mãi |

* + - * Output : Hiển thị thông báo trạng thái.
      * Xử lý :
        + Gọi đến service promotionService và sử dụng phương thức createPromotion.

API xem danh sách shipper (API18)

* + - * Url : /staff
      * Method : GET
      * Query param :

|  |  |  |  |
| --- | --- | --- | --- |
| Tên | Kiểu dữ liệu | Bắt buộc | Mô tả |
| Role | String | X | Role shipper để lấy danh sách người giao hàng |

* + - * Output : Là 1 Array chứa các object là các shipper để hiển thị ra view.
      * Xử lý :
        + Gọi đến service staffService sau đó sử dụng method getShipper.

API duyệt đơn(API18)

* + - * Url :
      * Method :
      * Query param :

|  |  |  |  |
| --- | --- | --- | --- |
| Tên | Kiểu dữ liệu | Bắt buộc | Mô tả |
|  |  |  |  |

* + - * Input :

|  |  |  |  |
| --- | --- | --- | --- |
| Tên | Kiểu dữ liệu | Bắt buộc | Mô tả |
|  |  |  |  |

* + - * Output :
      * Xử lý :

API xác nhận cập nhật trạng thái đơn hàng(API19)

* + - * Url :
      * Method :
      * Query param :

|  |  |  |  |
| --- | --- | --- | --- |
| Tên | Kiểu dữ liệu | Bắt buộc | Mô tả |
|  |  |  |  |

* + - * Input :

|  |  |  |  |
| --- | --- | --- | --- |
| Tên | Kiểu dữ liệu | Bắt buộc | Mô tả |
|  |  |  |  |

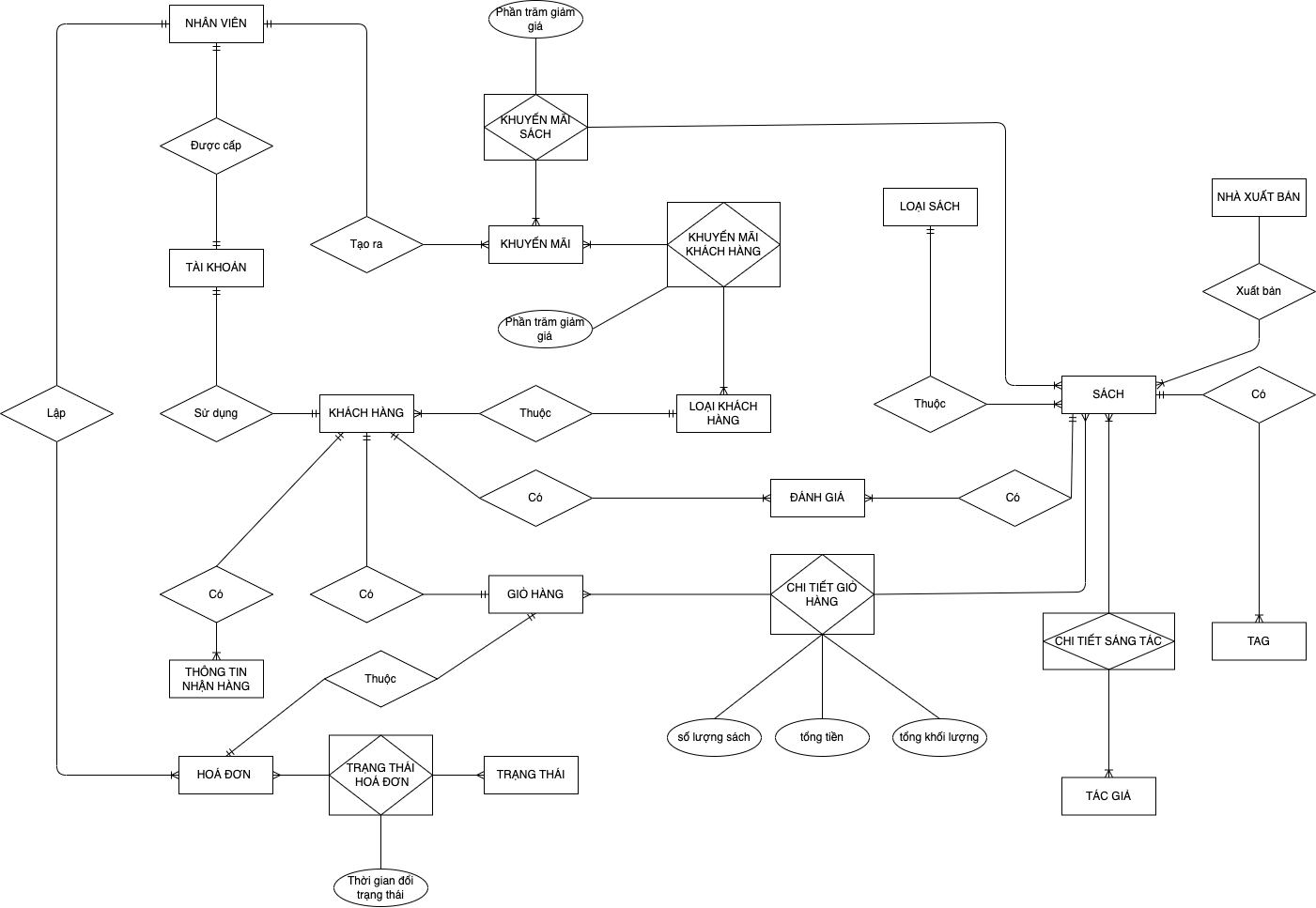
* + - * Output :
      * Xử lý :

## 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).
* Khách hàng (mã khách hàng, họ, ngày sinh, số điện thoại, email, avatar).
* Tài khoản (mã tài khoản, mật khẩu).
* 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, sách mới, cân nặng).
* Loại sách(mã loại sách, tên loại sách).
* Đánh giá (mã đánh giá, bình luận, số sao).
* Tag (mã tag, nội dung).
* Tác giả (mã tác giả, họ, tên, giới tính, sinh nhật, 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).
* Hoá đơn (mã hoá đơn, ngày lập hoá đơn, tổng số tiền, thuế).
* Trạng thái hoá đơn (mã trạng thái, tên trạng thái, chi tiết).

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



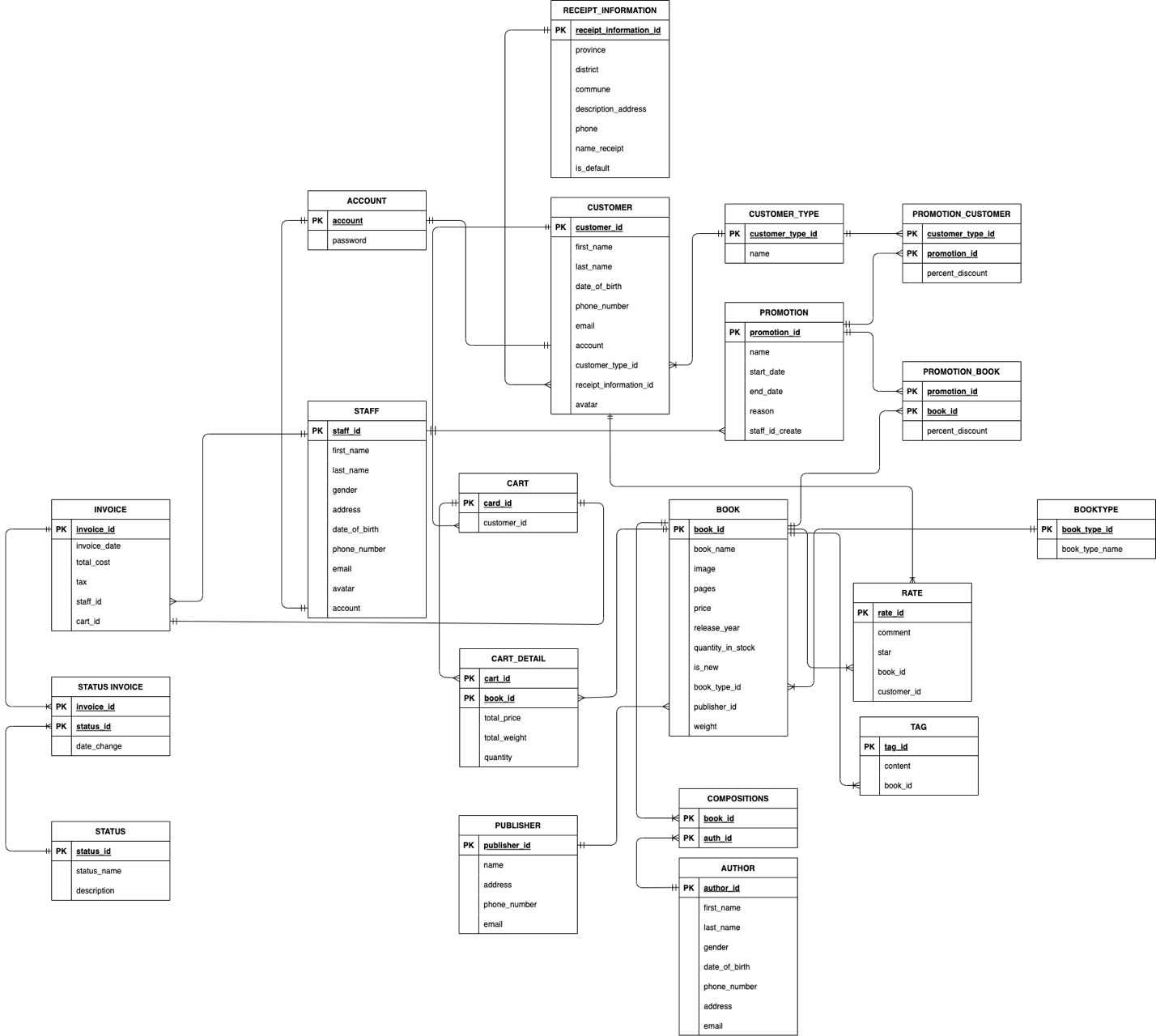
Hình 4..2 Mô hình ERD

### 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, mã tài khoản).
* Khách hàng (mã khách hàng, họ, ngày sinh, số điện thoại, email, avatar, tài khoản).
* Tài khoản (mã tài khoản, mật khẩu).
* 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, mã nhân viên tạo khuyến mãi).
* Khuyễn mãi sách (mã khuyến mãi, mã sách, phần trăm giảm giá).
* Khuyễn mãi theo loại khách (mã khuyến mãi, mã loại khách hàng, phần trăm giảm giá).
* Sách (mã sách, tên sách, hình ảnh, số trang, giá, năm phát hành, số lượng còn, sách mới, cân nặng, mã loại sách, mã nhà xuất bản).
* Loại sách(mã loại sách, tên loại sách).
* Đánh giá (mã đánh giá, bình luận, số sao, mã sách, mã khách hàng).
* Tag (mã tag, nội dung, mã sách).
* Tác giả (mã tác giả, họ, tên, giới tính, sinh nhật, số điện thoại, địa chỉ, email).
* Sáng tác (mã sách, mã tác giả).
* Nhà xuất bản (mã nhà xuất bản, tên, địa chỉ, số điện thoại, email).
* Giỏ hàng(mã giỏ hàng).
* Chi tiết giỏ hàng (mã giỏ hàng, mã sách, tổng tiền, tổng cân nặng, số lượng).
* Hoá đơn (mã hoá đơn, ngày lập hoá đơn, tổng số tiền, thuế, mã nhân viên lập hoá đơn).
* Trạng thái (mã trạng thái, tên trạng thái, chi tiết).
* Trạng thái hoá đơn (mã trạng thái, mã hoá đơn, ngày thay đổi).

## 4.4. Thiết kế dữ liệu

### 4.4.1. Mô hình Diagram



Hình 4..1 Mô hình database diagram

### 4.4.2. Từ điển dữ liệu

# 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/>