

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**  
**CỞ SỞ TẠI THÀNH PHỐ HỒ CHÍ MINH**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO TỐT NGHIỆP ĐẠI HỌC**

**Môn: Phân tích, thiết kế và  
đảm bảo chất lượng phần mềm**

**ĐỀ TÀI:**

**XÂY DỰNG WEBSITE BÁN  
QUÀ TẶNG, PHỤ KIỆN**

**Người hướng dẫn : ThS. NGUYỄN ANH HÀO**  
**Sinh viên thực hiện : LƯU TIẾN PHÁT**  
**Mã số sinh viên : N18DCCN146**  
**Lớp : D18CQCP02\_N**  
**Khoá : 2018 - 2022**  
**Hệ : ĐẠI HỌC CHÍNH QUY**

**TP. HỒ CHÍ MINH – 12/2022**

## LỜI CẢM ƠN

Trước hết em xin bày tỏ lòng biết ơn sâu sắc tới giáo viên hướng dẫn ThS. Nguyễn Anh Hào đã tận tình giúp đỡ, hướng dẫn em rất nhiều trong suốt quá trình tìm hiểu nghiên cứu và hoàn thành môn học.

Em xin chân thành cảm ơn các thầy cô trong khoa Công Nghệ Thông Tin 2 - Học Viện Công Nghệ Bưu Chính Viễn Thông cơ sở tại Thành phố Hồ Chí Minh cũng như các thầy cô trong trường đã trang bị cho em những kiến thức cơ bản cần thiết để em có thể hoàn thành báo cáo.

Xin gửi lời cảm ơn đến bạn bè những người luôn bên em đã động viên và tạo điều kiện thuận lợi cho em, tận tình giúp đỡ chỉ bảo em những gì em còn thiếu sót trong quá trình làm báo cáo tốt nghiệp.

Cuối cùng em xin bày tỏ lòng biết ơn sâu sắc tới những người thân trong gia đình đã dành cho em sự quan tâm đặc biệt và luôn động viên em.

Vì thời gian có hạn, trình độ hiểu biết của bản thân còn nhiều hạn chế. Cho nên trong đề án không tránh khỏi những thiếu sót, em rất mong nhận được sự đóng góp ý kiến của tất cả các thầy cô giáo cũng như các bạn bè để đề án của em được hoàn thiện hơn.

Em xin chân thành cảm ơn!

Tp. Hồ Chí Minh, ngày 04 tháng 12 năm 2022  
Sinh viên thực hiện

Lưu Tiến Phát

[illegible]

## Mục Lục

1.	Tên đồ án .....	5
2.	Loại đồ án: .....	5
3.	Mục đích, ý nghĩa của đồ án: .....	5
II.	Phân tích hệ thống .....	5
1.	Ngữ cảnh của hệ thống .....	5
2.	Lược đồ usecase cho phân tích .....	7
3.	Yêu cầu/ràng buộc đối với phần mềm .....	14
a.	Yêu cầu từ môi trường nghiệp vụ (business) .....	14
b.	Yêu cầu từ môi trường vận hành (operation) .....	15
	• Các yêu cầu chức năng .....	15
	• Các yêu cầu chất lượng .....	16
	• Các yêu cầu từ môi trường phát triển (development) .....	17
III.	Thiết kế hệ thống phần mềm. ....	18
1.	Lược đồ usecase cho thiết kế. ....	18
1.1.	Usecase đăng ký .....	18
1.2.	Usecase refresh token .....	21
1.3.	Usecase thiết kế tìm kiếm .....	23
1.4.	Usecase đặt hàng .....	25
1.5.	Usecase thanh toán online .....	28
1.6.	Usecase đặt và giao hàng theo yêu cầu .....	31
1.7.	Usecase trả hàng .....	33
1.8.	Usecase cập nhật đơn hàng .....	35
2.	Cơ sở dữ liệu và ràng buộc .....	39
2.1.	Lược đồ ERD: .....	39
2.2.	Bảng: .....	39
1.9.	Ràng buộc .....	45
1.10.	Phân quyền .....	45

## I. Giới thiệu

1. **Tên đề án:** Xây dựng website bán quà tặng, phụ kiện.

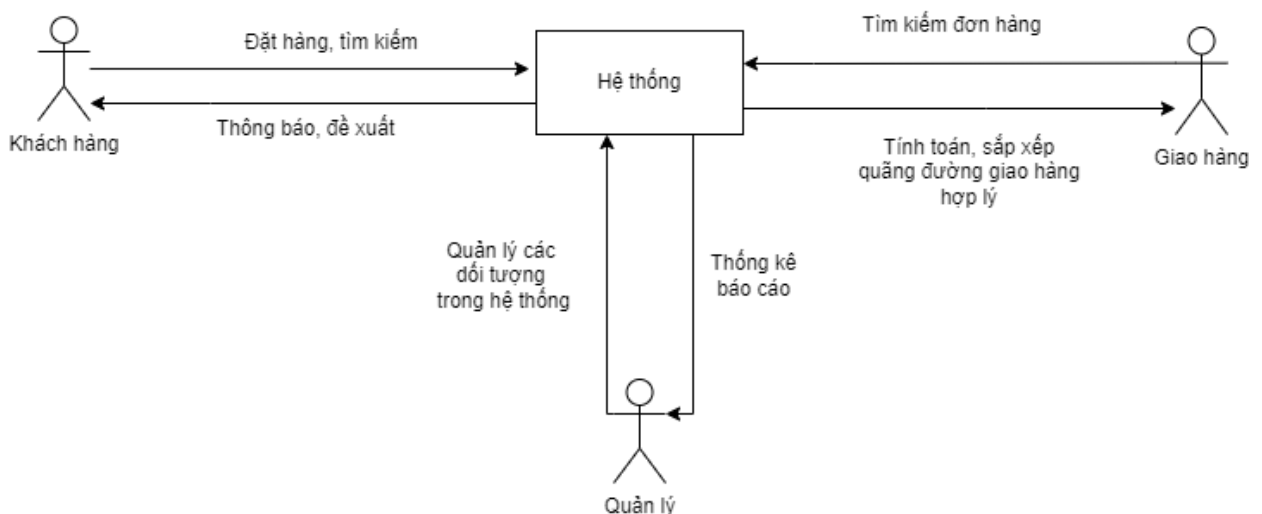
2. **Loại đề án:** Xây dựng một phần mềm ứng dụng.

3. **Mục đích, ý nghĩa của đề án:**

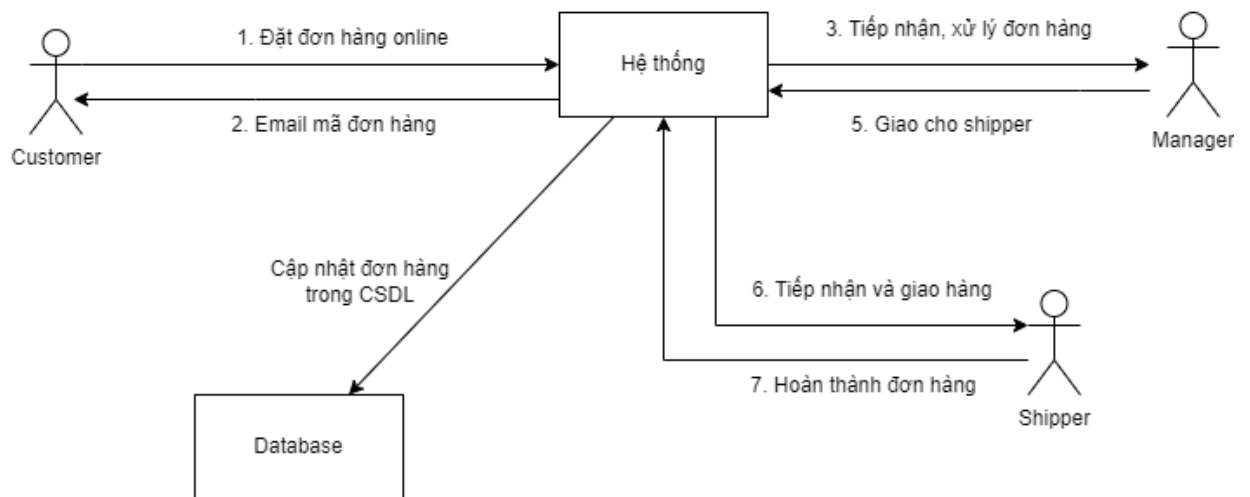
- **Mục đích:**
  - Ứng dụng dành cho khách hàng, giúp khách hàng dễ dàng tìm kiếm, đặt hàng và thanh toán online, ngoài ra khách hàng có thể đặt riêng theo yêu cầu, tự chọn quà tặng trên website, cửa hàng sẽ chuyển đến người nhận.
  - Ứng dụng quản lý dành cho người quản lý, dễ dàng theo dõi, nắm bắt thông tin các đối tượng trong cửa hàng.
  - Ứng dụng giao hàng dành cho người giao hàng trên các thiết bị di động, ứng dụng cho phép tìm kiếm đơn hàng, tìm đường đi tốt nhất.
- **Mục tiêu:** Tạo ra một hệ thống quản lý tất cả các đối tượng trong cửa hàng, giúp khách hàng, người quản lý, người giao hàng dễ dàng tìm kiếm, quản lý và liên hệ lẫn nhau. Và có yêu cầu về khả năng tự động khi khách hàng có thể đặt riêng quà tặng theo yêu cầu, địa điểm giao hàng vì vậy sẽ có cơ chế đưa ra gợi ý liên quan để tăng tốc độ cũng như hiệu suất cho cửa hàng.

## II. Phân tích hệ thống

### 1. Ngữ cảnh của hệ thống



Lược đồ cộng tác thể hiện sự cộng tác giữa Khách hàng, Quản lý, Giao hàng trong hệ thống ứng dụng thương mại kinh doanh mặt hàng quà tặng, phụ kiện.



Lược đồ thể hiện sự cộng tác giữa các đối tượng là khách hàng, quản lý và người giao hàng với hệ thống khi khách hàng đặt một đơn hàng mới. Hệ thống này tự động tiếp nhận, lưu trữ vào database và thông báo cho quản lý để xử lý. Sau khi xong sẽ tự động tính toán, sắp xếp shipper phù hợp với quãng đường là tối ưu nhất để tiết kiệm thời gian, chi phí đi lại.

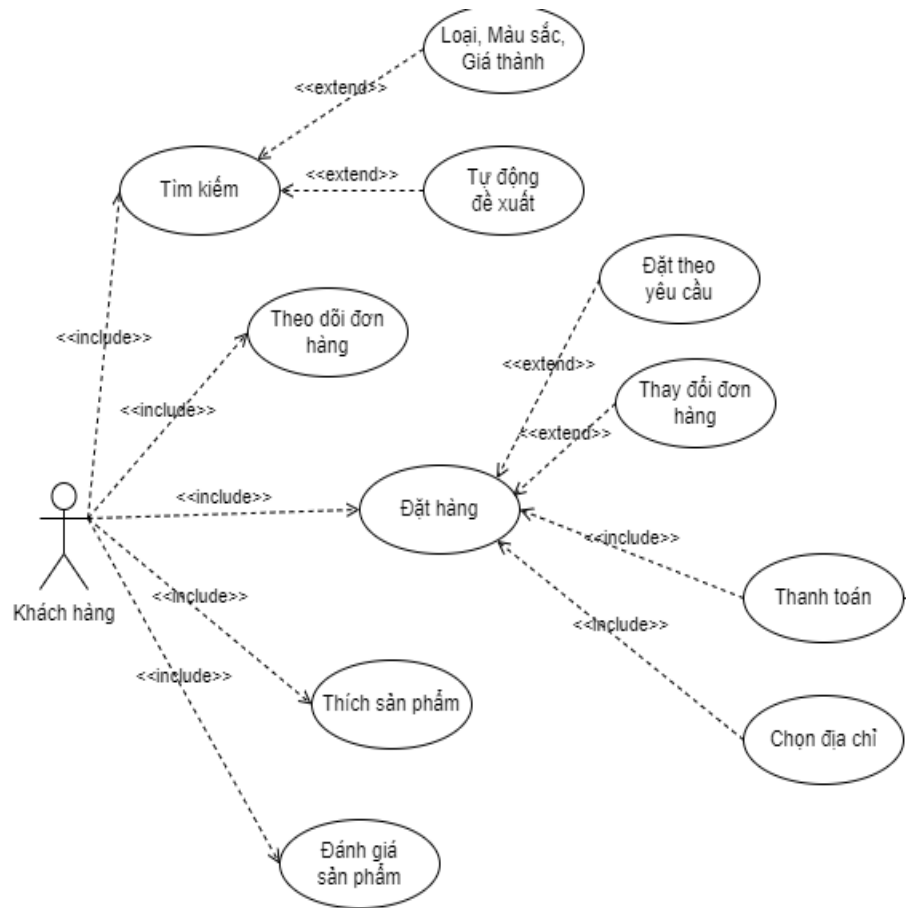
## 2. Lược đồ usecase cho phân tích

### Usecase U01: Lược đồ usecase của khách hàng.

Ý nghĩa: Mô tả các chức năng chính mà khách hàng có thể tương tác với website

Actors: Khách hàng.

Điều kiện: khách hàng truy cập vào website của cửa hàng.



*Lược đồ use case các chức năng dành cho khách hàng*

### Usecase U02: **Lược đồ usecase đặt hàng online.**

Ý nghĩa: Mô tả khi khách hàng thực hiện đặt hàng và thanh toán online

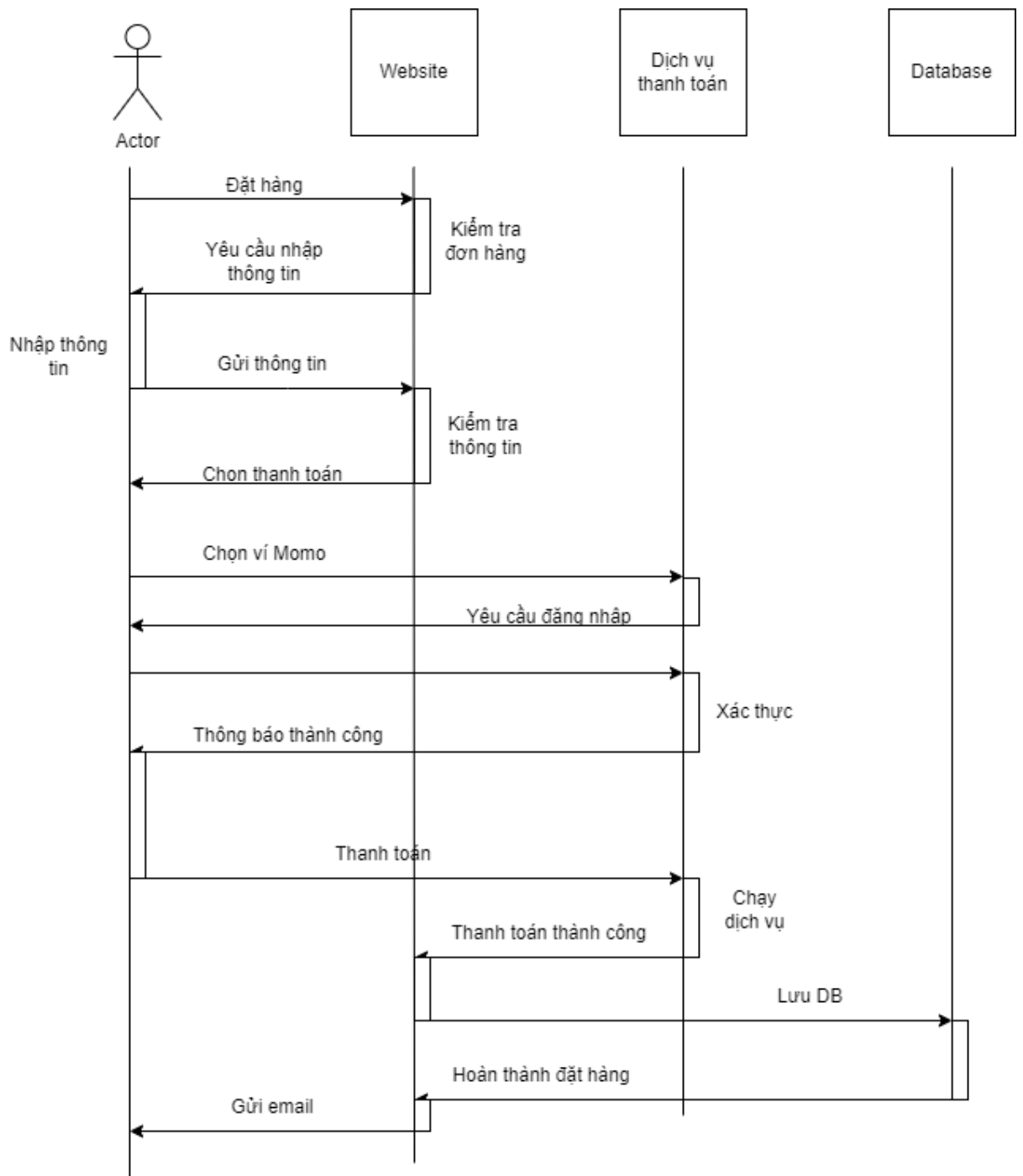
Actors: Khách hàng, Xác thực, Dịch vụ e-banking, Dịch vụ ví,

Điều kiện: khách hàng đặt hàng và thanh toán online trên website của cửa hàng



*Lược đồ use case khách hàng đặt hàng và thanh toán online*





*Lược đồ tuần tự thể hiện khách hàng đặt hàng và thanh toán online.*

### Usecase U03: Nhân viên quản lý đơn đặt hàng

Ý nghĩa: Tiếp nhận và thay đổi trạng thái đơn hàng để dễ dàng nắm bắt thông tin đơn hàng của khách hàng.

Actors: Khách hàng, Dịch vụ e-banking, Dịch vụ ví, Xác thực.

Điều kiện: Khi khách hàng đã đặt hàng thành công và hệ thống đã ghi nhận đơn hàng của khách hàng.



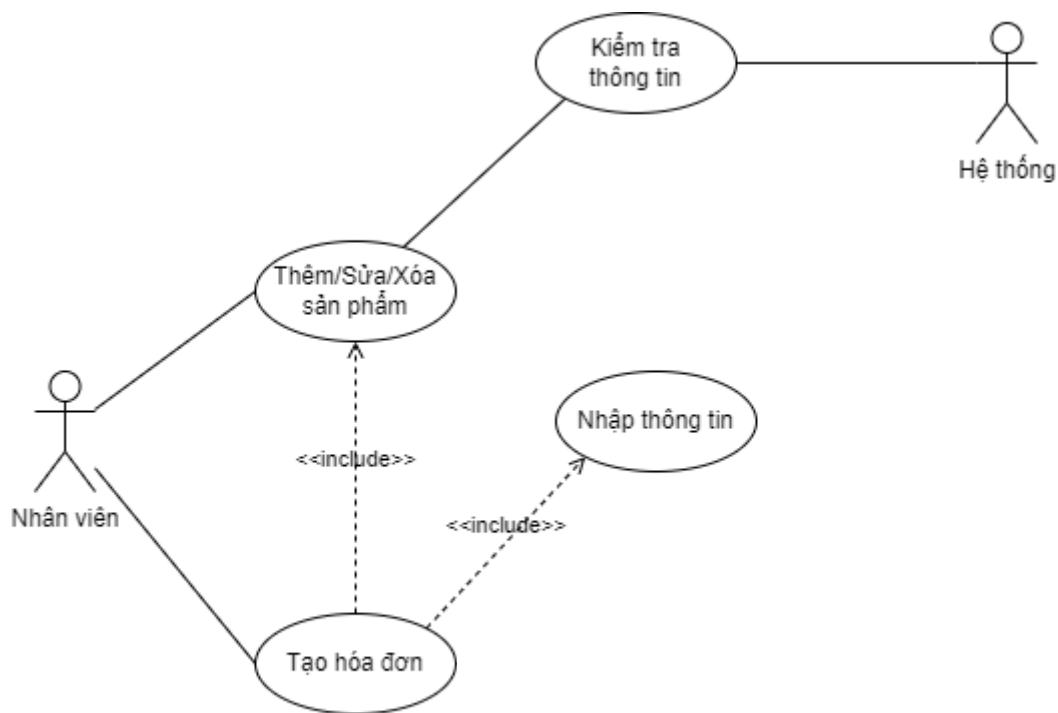
*Lược đồ usecase nhân viên quản lý đơn hàng*

#### Usecase U04: Nhân viên quản lý hóa đơn nhập/xuất của cửa hàng

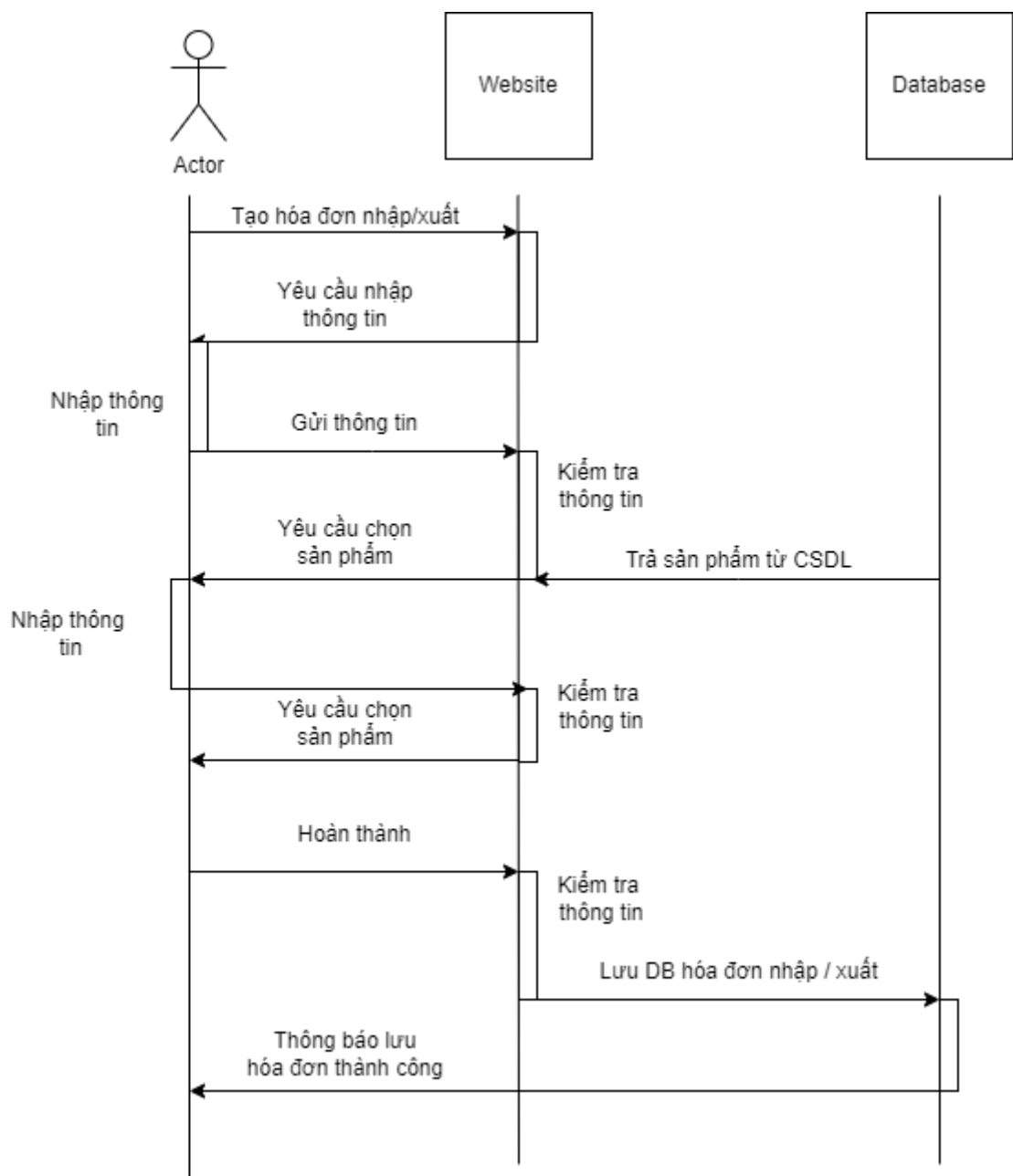
Ý nghĩa: Số lượng, mẫu mã của sản phẩm được thay đổi liên tục hoặc thanh lý, loại bỏ hàng tồn kho, cần một hệ thống quản lý nhập xuất để cập nhật những thay đổi của hàng hóa. Điều này giúp cho thông tin của sản phẩm không bị lỗi thời, tránh gây thất thoát, hao hụt, dễ thống kê, báo cáo.

Actors: Nhân viên, Hệ thống

Điều kiện: Khi cửa hàng nhập/xuất hàng hóa và hệ thống cần cập nhật thông tin cho hoạt động nhập/xuất đó.



Lược đồ UseCase tạo hóa đơn nhập/xuất hàng hóa vào hệ thống



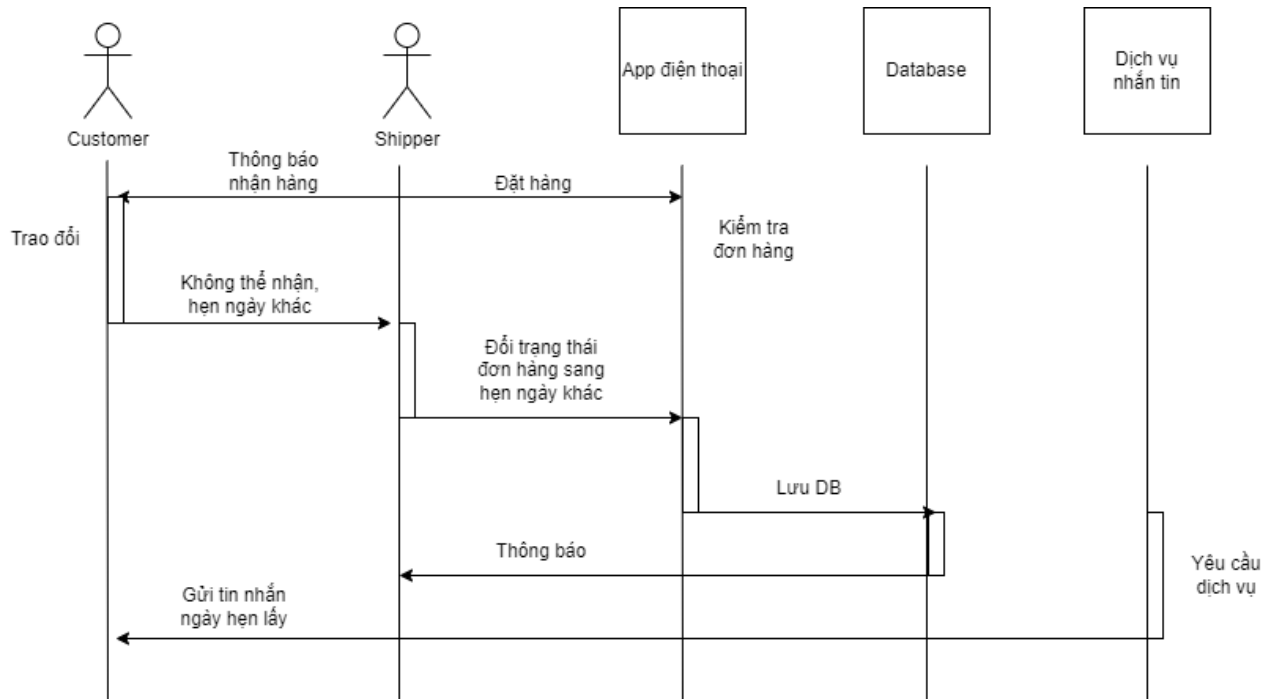
Lược đồ hoạt động quá trình tạo hóa đơn nhập/xuất

### Usecase U05: Shipper hẹn ngày giao khi khách hàng không thể nhận hàng hôm nay

Ý nghĩa: Thể hiện các trình tự hoạt động của người giao hàng khi giao hàng, khi hủy đơn hàng vì sự cố hoặc khi khách hàng muốn chọn giao ngày hôm khác.

Actors: Khách hàng, Shipper

Điều kiện: khách hàng hiện tại không có mặt để nhận hàng, xác nhận chuyển sang ngày khác.



Lược đồ tuần tự hẹn ngày giao khi khách hàng không thể nhận trong hôm nay.

### 3. Yêu cầu/ràng buộc đối với phần mềm

#### a. Yêu cầu từ môi trường nghiệp vụ (business)

UseCase	ID2	Nội dung yêu cầu	Stack-Holder
U01	F01	<b>Cửa hàng có cam kết về chất lượng của mặt hàng</b> Những các loại phụ kiện có hóa đơn nguồn gốc sản xuất rõ ràng, những mặt hàng “tươi” có hạn sử dụng và phải thanh lý khi quá hạn	Chủ cửa hàng
	F02	<b>Cửa hàng phải cho biết chi phí mua hàng</b> Giá được niêm yết trên mỗi sản phẩm	Chủ cửa hàng
	F03	<b>Có giấy phép kinh doanh</b> Phải có đầy đủ giấy tờ pháp ký, cho phép kinh doanh của nhà nước, trên trang web hiển thị huy hiệu Đã báo bộ Công thương	Chủ cửa hàng
	F04	<b>Cam kết về bảo hành cho sản phẩm</b> Thông tin về thời gian, chính sách bảo hành đề được in trên website của cửa hàng	Chủ cửa hàng
U02	F01	<b>Khách hàng được phép đặt hàng, hủy đơn hàng.</b> Cho phép khách hàng đặt hàng online, thanh toán online, hủy đơn hàng.	Khách hàng
	F02	<b>Khách hàng đổi, trả hàng</b> Nếu có lý do phù hợp với chính sách của cửa hàng thì cửa hàng sẽ cho phép đổi, trả hàng. Thông tin đổi, trả hàng được ghi rõ khi khách hàng mua hàng.	Khách hàng
	F03	<b>Bình luận, Đánh giá sản phẩm</b> Khách hàng được đánh giá sản phẩm, đăng ảnh thực tế trong bình luận nhưng sẽ có kiểm soát nội dung.	Khách hàng
	F04	<b>Quản lý đơn hàng</b> Khách hàng xem lại lịch sử đặt hàng, xem các sản phẩm đã thích, theo dõi đơn hàng của mình,..	Khách hàng
U03	F01	<b>Quản lý</b> Nhân viên quản lý quản lý các đối tượng trong cửa hàng (thêm, sửa, xóa,...) tùy theo chức vụ của nhân viên đó được giao	Nhân viên
	F02	<b>Nhân viên ngưng đơn hàng, hủy đơn hàng</b> Khi sản phẩm có vấn đề, nhân viên sẽ nói chuyện với khách hàng tạm ngưng đơn hàng hoặc hủy.	Nhân viên
	F03	<b>Thông kê, báo cáo, biểu đồ</b> Nhân viên thấy được tình hình doanh thu theo thời gian, thông kê sản phẩm nhờ vào biểu đồ, và có thể in ra báo cáo.	Nhân viên

U04	F01	<b>Hẹn ngày giao lại</b> Khách hàng có lý do đặc biệt, thì có thể hẹn giao ngày khác, gửi tin nhắn hoặc email đến để thông báo ngày hẹn với khách hàng;	Người giao hàng
-----	-----	--	-----------------

**b. Yêu cầu từ môi trường vận hành (operation)**

- Các yêu cầu chức năng

UseCase	ID	Nội dung yêu cầu	Stack-Holder
U01	F01	<b>Phân quyền</b> Giới hạn chức năng của ứng dụng cho từng loại tài khoản, mục đích sử dụng	Hệ thống
	F02	<b>Bảo mật</b> Sử dụng nhiều lớp bảo mật cho hệ thống, dùng JWT để bảo mật thông tin đăng nhập của người dùng vào hệ thống, sử dụng Spring-Security để giúp API được bảo mật hơn. Tránh được các tấn công mạng như DDoS hay SQL injection, bảo vệ thông tin người dùng,	Hệ thống
	F03	<b>Kiểm tra dữ liệu nhập</b> Trước khi lưu dữ liệu thì đều được kiểm tra tính đúng đắn của dữ liệu nhập vào tránh nhầm lẫn, sai sót	Hệ thống
	F04	<b>Các trợ giúp chức năng</b> Nhấn vào các biểu tượng dấu hỏi nhỏ sẽ hiển thị ra các hướng dẫn giúp người sử dụng	Hệ thống
	F05	<b>Thông báo</b> Hiển thị thông báo, lưu ý, cảnh báo trên ứng dụng hoặc gửi email về thông báo đó	Hệ thống
U02	F01.01	<b>Yêu cầu về đặt hàng</b> Chỉ duyệt đơn hàng khi hàng đang có sẵn trong hệ thống. Nếu hàng hết quá 30 ngày thì sẽ hủy đơn hàng và thông báo cho khách hàng.	Hệ thống
	F01.02	<b>Thông tin đơn hàng</b> Trạng thái đơn hàng được cập nhật bởi cửa hàng, khách hàng có thể xem để cập nhật thông tin.	Hệ thống
	F02	<b>Nhập/xuất hàng hóa</b> Nếu hàng hóa hết nhưng có vẫn có khách đã đặt hàng trước đó thì sẽ thông báo yêu cầu nhập thêm trên hệ thống, Một số sản phẩm có hạn sử dụng giới hạn như hoa tươi, bánh ngọt sẽ được yêu cầu thanh lý nếu quá ngày.	Hệ thống

	F03	<b>Yêu cầu về bình luận</b> Các bình luận có nội dung không liên quan sẽ được hệ thống quét các từ khóa xấu và khóa lại, đợi yêu cầu từ người dùng.	Hệ thống
U03	F01	<b>Sắp xếp, tính toán trình tự giao hàng</b> Để giúp shipper có thể tối ưu hóa lộ trình giao hàng, hệ thống sẽ lấy thông tin giao hàng và tính toán lộ trình cho shipper một cách tốt nhất, giảm chi phí đi lại	Hệ thống

- **Các yêu cầu chất lượng**

UseCase	ID	Nội dung yêu cầu	Stack-Holder
U01	NF01.01	<b>Performance efficiency: hệ thống</b> <i>Time-behavior:</i> $\leq 2$ seconds / transaction <i>Resource utilization:</i> cpu I5, hdd 1T, mem 16G <i>Capacity:</i> $\sim 50$ transactions / day	Hệ thống
	NF01.02	<b>Maintainability: hệ thống</b> Reusability: <b>yes</b> , Modifiability: <b>yes</b> , Testability: <b>yes</b>	Hệ thống
	NF02	<b>Authentication: hệ thống</b> Sử dụng OAuth2 để đăng nhập vào hệ thống, thông tin đăng nhập được lưu trữ dưới dạng token và lưu vào trong cookie của trình duyệt. Có giới hạn thời gian sống.	Hệ thống
U02	NF01	<b>Security Checkout: website</b> Không để lộ lọt thông tin thanh toán của khách hàng, không lưu trữ thông tin nhạy cảm, sử dụng các chứng chỉ hàng đầu để khách hàng yên tâm thanh toán online.	Hệ thống
	NF02	<b>Responsive UI: website</b> Website mua bán có thể thay đổi dựa theo kích thước màn hình.	Hệ thống
	NF03	<b>Easy-to-use: website</b> <ul style="list-style-type: none"> <li>• Giao diện hài hòa đẹp mắt, sử dụng font hiện đại, color palette để phối màu, hình ảnh sản phẩm rõ nét, các chức năng chính được bố trí hợp lý.</li> <li>• Giao diện quản lý được tổ chức hợp lý, bảng số liệu và biểu đồ thể hiện trực quan, dễ hiểu. sử dụng các biểu tượng để trình bày sinh động, tránh nhàm chán.</li> </ul>	Hệ thống
	NF04	<b>Performace: website</b> Thời gian load: $< 10s$ phụ thuộc vào kết nối mạng Thời gian phản hồi: $< 5s$ phụ thuộc vào kết nối mạng.	Hệ thống
	NF05	<b>Caching: website</b>	Hệ thống



		Sử dụng <b>cache</b> để lưu lại hình ảnh trên browser, đỡ mất thời gian load lại khi mạng chậm.	
U03	NF01	<b>Compatibility: app</b> Android 7.0 trở lên, IOS 11 trở lên	Hệ thống
	NF02	<b>Performance: app</b> Dung lượng: < 50MB Thời gian phản hồi < 10s tùy thuộc vào kết nối mạng. Không được có hiện tượng giật, lag.	Hệ thống
	NF03	<b>No Crashing: app</b> App không được có hiện tượng crashing và yêu cầu phải có internet mới có thể thực hiện các chức năng chính	Hệ thống

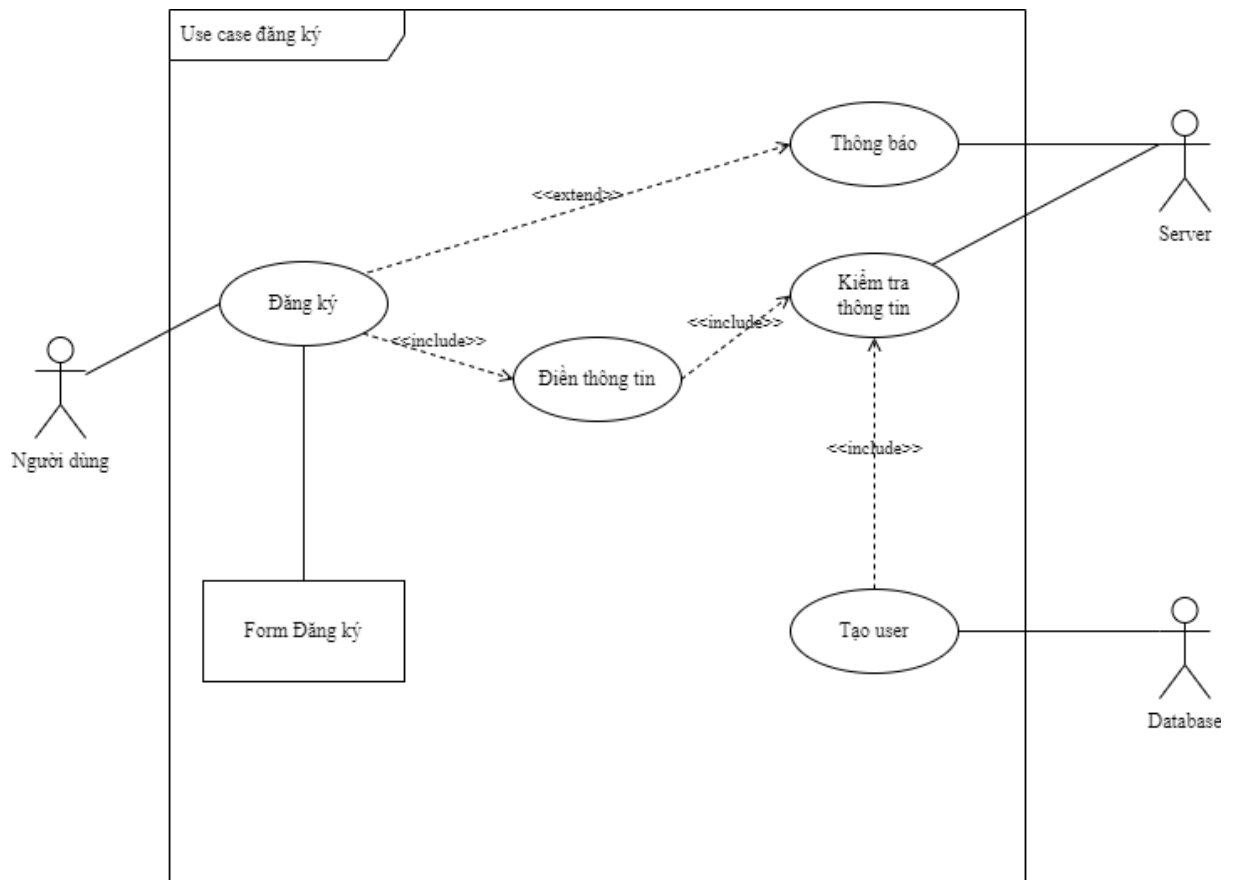
- Các yêu cầu từ môi trường phát triển (development)

ID	Đối tượng	Nội dung yêu cầu	Stack-Holder
TA01	Application	Website: sử dụng trình duyệt mới nhất, không hỗ trợ IE	Khách hàng
TA02	Application	App điện thoại: Android 7 trở lên, IOS 11 trở lên.	Shipper
TD01	Application	Server: Intel, Windows 10 Server, CPU Xeon, HDD 5TB, RAM 32GB	Dev team
TD02	Software	Viết code trên Visual Studio Code đối với website, Eclipse đối với API và Andoird Studio đối với app trên điện thoại.	Dev team
TD03	Software	Dùng ReactJs cho webiste, Spring Boot cho API và Flutter để tạo app điện thoại.	Dev team
TD04	Software	Dùng Selenium để viết các test case tự động, sử dụng Microsoft Excel.	Tester team

### III. Thiết kế hệ thống phần mềm.

#### 1. Lược đồ usecase cho thiết kế.

##### 1.1. Usecase đăng ký



*Lược đồ usecase thiết kế cho đăng ký*

##### a. Usecase 01: Đăng ký

- Vai trò: Người dùng đăng ký tài khoản để có thể đăng nhập vào hệ thống.
- Tương tác: với form Đăng ký và Api Đăng ký.

**b. Form 01: Đăng ký**

- Hình của form:

ĐĂNG NHẬP

ĐĂNG KÝ

Tên đăng nhập (\*)

Họ (\*)

Tên (\*)

Điện thoại (\*)

Email (\*)

Địa chỉ (\*)

Mật khẩu của bạn (\*)

Nhập lại mật khẩu (\*)

☐ Hiện mật khẩu

ĐĂNG KÝ

- Giao tiếp với user: Người dùng điền thông tin cá nhân (*họ, tên, số điện thoại, địa, chỉ, email, tên đăng nhập, mật khẩu*).
- Các control:

Tên control	Input	Output	Xử lý
submit	- Họ, tên - Số điện thoại - Địa chỉ - Email - Tên đăng nhập. - Mật khẩu.	- Nếu tạo tài khoản thành công chuyển sang trang đăng nhập. - Nếu tạo thất bại sẽ thông báo	- Nhận thông tin form gửi đến và kiểm tra dữ liệu gửi. 1. Nếu thông tin đúng → Gọi API đăng ký, gửi thông tin đăng ký → Lưu thông tin người dùng vào database. → Phản hồi thành công về trang đăng ký 2. Nếu thông tin không chính xác sẽ thông báo lỗi trên form đăng ký.

- Xác thực:

Nội dung	Bắt buộc	Xử lý
Họ	Có	- Kiểm tra nội dung rỗng
Tên	Có	- Kiểm tra nội dung rỗng
Số điện thoại	Không	- Kiểm tra số điện thoại yêu cầu phải đủ 10 số, và có định dạng phù hợp với số điện thoại ở Việt Nam.
Địa chỉ	Không	
Tên đăng nhập	Có	- Kiểm tra nội dung rỗng - Kiểm tra username đã bị trùng với người khác.
Mật khẩu	Có	- Kiểm tra nội dung rỗng - Kiểm tra có định dạng mật khẩu ( <i>mật khẩu phải ít nhất 5 ký tự và chứa ít nhất một chữ số trong đó</i> ).

#### c. Api/Service: Đăng ký

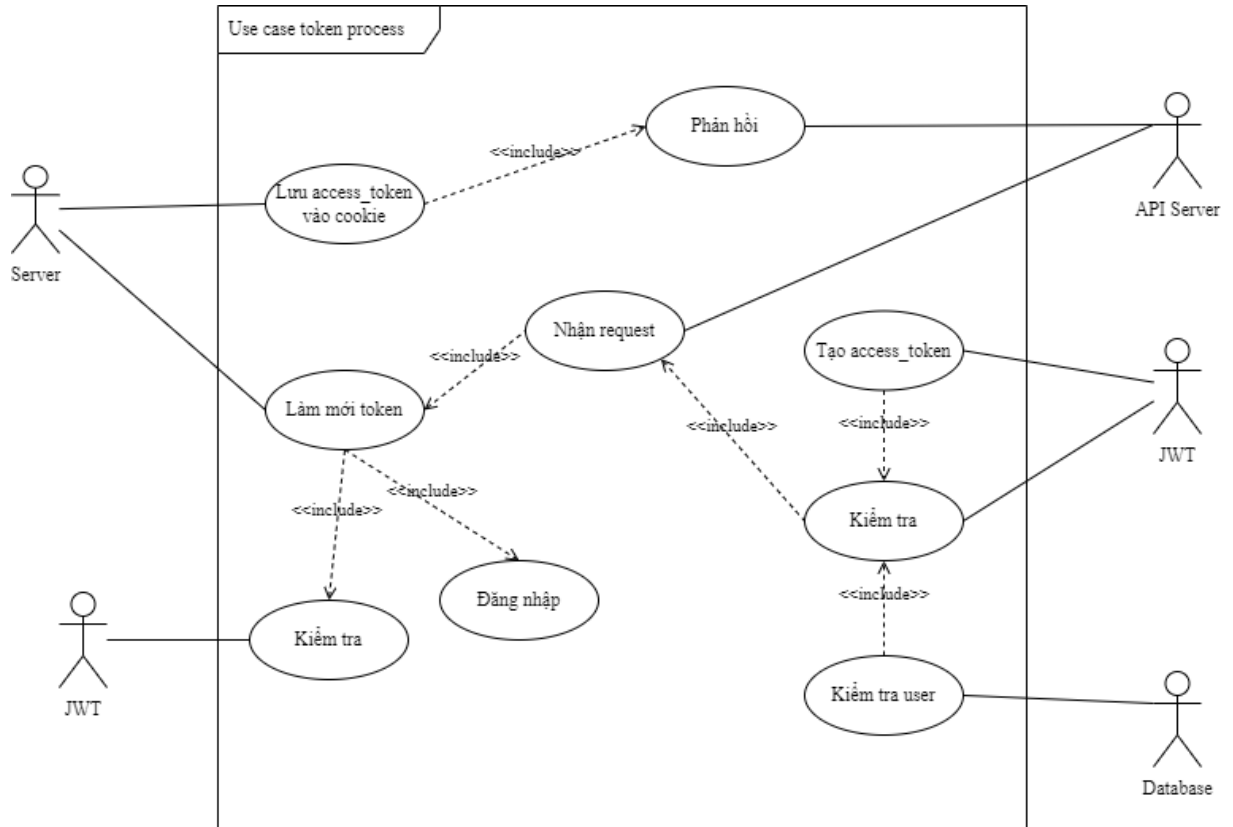
- Inputs: *họ, tên, số điện thoại, địa, chỉ, email, tên đăng nhập, mật khẩu*.
- Outputs: *thông báo, http code*
- Xử lý:
  - Nhận được dữ liệu được gửi lên từ client, tiếp tục xác thực các thông tin vừa nhận được. Kiểm tra *username* có bị trùng hay từng được sử dụng bởi người dùng khác.
  - Nếu *username* không bị trùng, lưu thông tin user vào database. → Gửi lại cho client *http code (success), tin nhắn*.
  - Nếu *username* bị trùng trả về *http code (not implemented), tin nhắn*.

Http Code	Ý nghĩa	Tin nhắn
200	Thành công	- Đăng ký user thành công
501	API không thể xử lý request	- <i>username</i> đã được sử dụng. - Không thể đăng ký user này

#### d. DBMS Procedure

- Inputs: *họ, tên, số điện thoại, địa, chỉ, email, tên đăng nhập, mật khẩu*.
- Outputs: *userId*.
- Tables: *user, role, user\_roles*.

## 1.2. Usecase refresh token



### a. Usecase 02: refresh token

- Vai trò: Server làm mới access\_token khi hết hạn.
- Tương tác: API refresh token

### b. Form 02: refresh token

- Hình của form: không có
- Giao tiếp với server: server yêu cầu làm mới access\_token.
- Các control:

Tên control	Input	Output	Xử lý
Refresh token	- refresh_token	- access_token mới	- Khi access_token hết hạn → Tiến hành kiểm tra refresh_token → Nếu refresh_token hết hạn → Đăng xuất và yêu cầu người dùng đăng nhập lại.  → Nếu refresh_token còn thời hạn thì gọi API yêu cầu lấy access_token khác.

			→ Sau khi lấy được access_token sẽ lưu và cookie của trình duyệt.
<b>Validation</b>	- refresh_token, - access_token	- <i>valid hoặc not valid</i>	- Dịch vụ jwt sẽ luôn luôn kiểm tra access_token mỗi khi chuyển sang trang mới để kiểm tra thời hạn của access_token. - Khi access_token hết hạn, hệ thống sẽ kiểm tra thời gian của refresh_token, → Trong thời hạn thì sẽ sử dụng token này để yêu cầu một access_token khác. → Trả về <i>true/false</i>

**Thời hạn token:** kể từ lúc đăng nhập

Loại token	Thời hạn
refresh_token	- 1 tháng
access_token	- 2 tuần

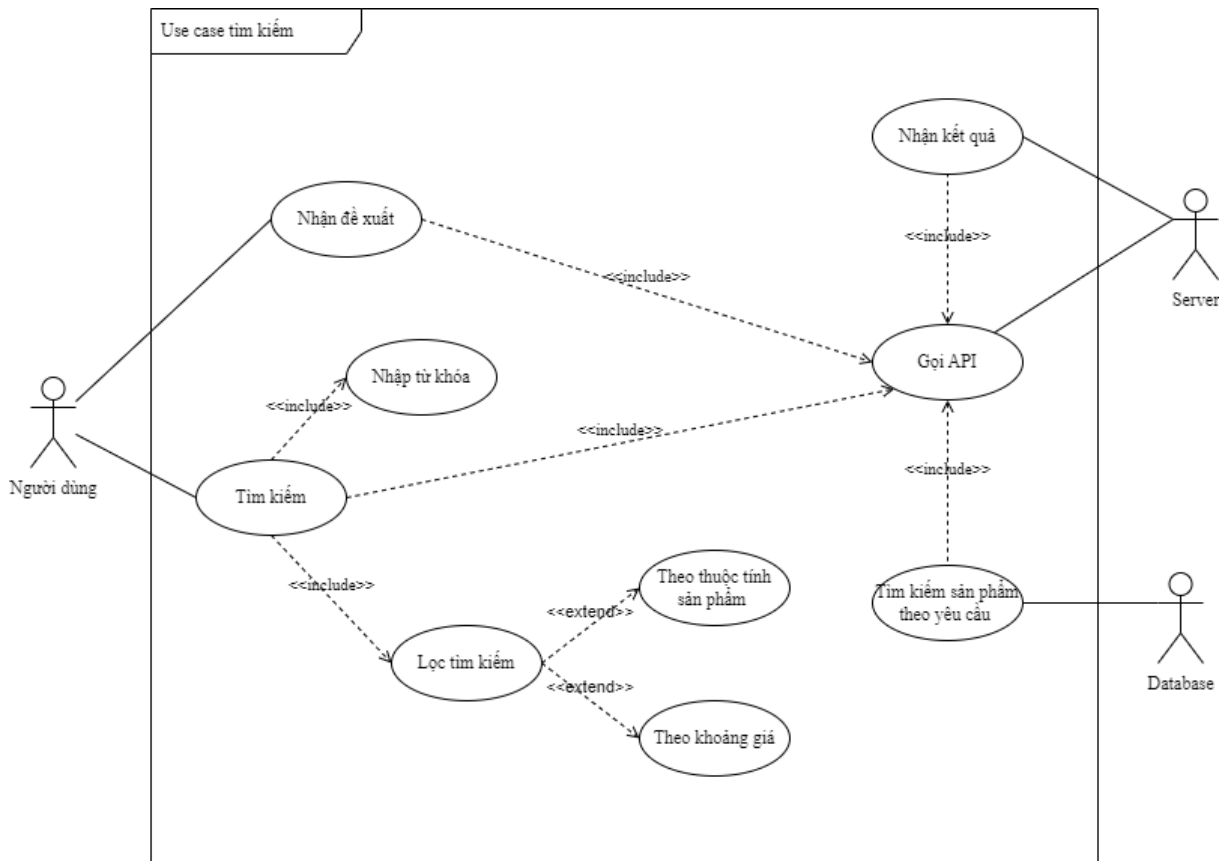
**c. Api/Service: refresh\_token**

- Inputs: *refresh\_token*
- Outputs: *thông báo, http code, access\_token*
- Xử lý:
  - Sau khi nhận được request và refresh\_token từ client → JWT sẽ tiến hành kiểm tra nội dung của token đó (*userId, expiration date*).
  - Nếu token không hợp lệ → Gửi về client lỗi và thông tin lỗi.
  - Nếu token hợp lệ → Tạo một access\_token khác và gửi lại cho client.

**d. DBMS Procedure**

- Inputs: *userId*
- Outputs: thông tin user
- Tables: user, role, user\_roles.

### 1.3. Usecase thiết kế tìm kiếm



#### a. Usecase 03: Thiết kế tìm kiếm

- Vai trò: Người dùng tìm kiếm theo yêu cầu, để tìm ra sản phẩm ưng ý nhất.
- Tương tác: với form Tìm kiếm và API tìm kiếm

#### b. Form 03: Tìm kiếm

- Hình của form:

- Giao tiếp với user: Người dùng điền từ khóa vào ô tìm kiếm, hoặc sử dụng bộ lọc tìm kiếm (theo thuộc tính hoặc khoảng giá).

- Các control:

Tên control	Input	Output	Xử lý
<b>submit</b>	<ul style="list-style-type: none"> <li>- Từ khóa</li> <li>- Màu sắc</li> <li>- Loại phụ kiện</li> <li>- Kích thước</li> <li>- Giới tính</li> <li>- Khoảng giá</li> </ul>	- Sản phẩm chứa đầy đủ thông tin mà người dùng yêu cầu.	<ul style="list-style-type: none"> <li>- Với mỗi input sẽ tương trưng cho một param. Hệ thống sẽ gọi API tìm kiếm sản phẩm, truyền vào những thuộc tính do người dùng yêu cầu.</li> <li>→ Trả về các sản phẩm có thông tin trùng khớp với yêu cầu người dùng</li> </ul>

- Xác thực:

Nội dung	Rỗng	Xử lý
Từ khóa	Không	- Rỗng → Trả về tất cả sản phẩm
Thuộc tính	Không	- Trả về sản phẩm chỉ có thuộc tính được yêu cầu
Khoảng giá	Không	- Trả về sản phẩm trong khoảng giá yêu cầu (tính cả giá sau khi áp khuyến mãi).

#### c. Api/Service: Tìm kiếm

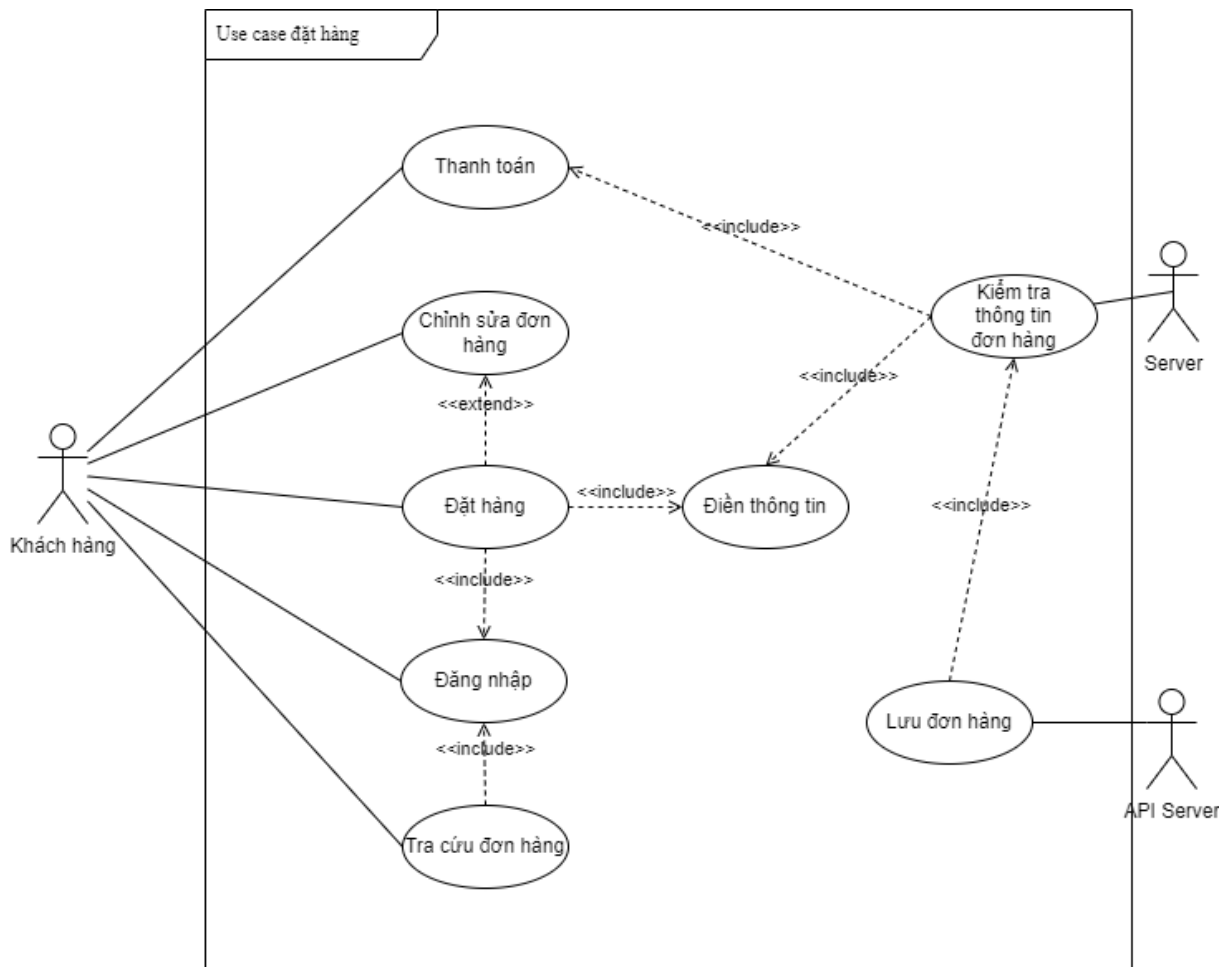
- Inputs: *từ khóa, thuộc tính sản phẩm, khoảng giá*
- Outputs: *danh sách sản phẩm phù hợp với yêu cầu.*
- Xử lý:
  - Thứ tự ưu tiên: *từ khóa, khoảng giá, thuộc tính.*

#### d. DBMS Procedure

- Inputs: *từ khóa, thuộc tính sản phẩm, khoảng giá*
- Outputs: *danh sách sản phẩm phù hợp với yêu cầu.*
- Tables: product, product\_detail



## 1.4. Usecase đặt hàng



### a. Usecase 04: Đặt hàng

- Vai trò: Cho phép người dùng đặt hàng
- Tương tác: với form Đặt hàng và API đặt hàng

## b. Form 04: Giỏ hàng

- Hình của form:

### 1. THÔNG TIN NGƯỜI NHẬN

Họ tên: (\*)

Điện thoại: (\*)



Email: (\*)

Địa chỉ: (\*)

Ghi chú:

### 2. THANH TOÁN

☐ COD: Thanh toán khi nhận hàng

### 3. THÔNG TIN GIỎ HÀNG

Tên	Số lượng	Thành tiền
Kính mắt gọng nhựa Double B Đơn giá: 100.000 đ	1	100.000 đ
Gối sưởi Baby bear gấu bịt mắt sạc điện 27X17 Đơn giá: 270.000 đ	1	270.000 đ
Tạm tính		370.000 đ
Phí vận chuyển		0 đ
Tổng cộng		370.000 đ

Thanh toán

- Giao tiếp với user: Người dùng chọn sản phẩm và lưu lại trong giỏ hàng, người dùng yêu cầu đã chọn hình thức thanh toán.
- Các control:

Tên control	Input	Output	Xử lý
<b>Đặt hàng</b>	<ul style="list-style-type: none"> <li>Giỏ hàng</li> <li>Hình thức thanh toán</li> </ul>	<ul style="list-style-type: none"> <li>Xác nhận đặt hàng từ hệ thống.</li> </ul>	<ul style="list-style-type: none"> <li>Lấy thông tin đặt hàng. Tiến hành kiểm tra thông tin khách hàng nhập <ul style="list-style-type: none"> <li>→ Thông tin chưa hợp lệ, yêu cầu nhập lại.</li> <li>→ Thông tin hợp lệ, tiến hành lấy hình thức thanh toán của khách hàng.</li> </ul> </li> <li>1. COD: xác nhận đặt hàng thành công.</li> <li>2. Dịch vụ thanh toán online: <ul style="list-style-type: none"> <li>→ Sau khi thanh toán online thành công, xác nhận đặt hàng thành công</li> </ul> </li> </ul>

Kiểm tra:

Nội dung	Rỗng	Xử lý
Giỏ hàng	Không	- Rỗng → Yêu cầu chọn sản phẩm
Thông tin giao hàng	Không	- Rỗng → Yêu cầu nhập lại thông tin
Thanh toán	Không	- Rỗng → Yêu cầu chọn hình thức thanh toán

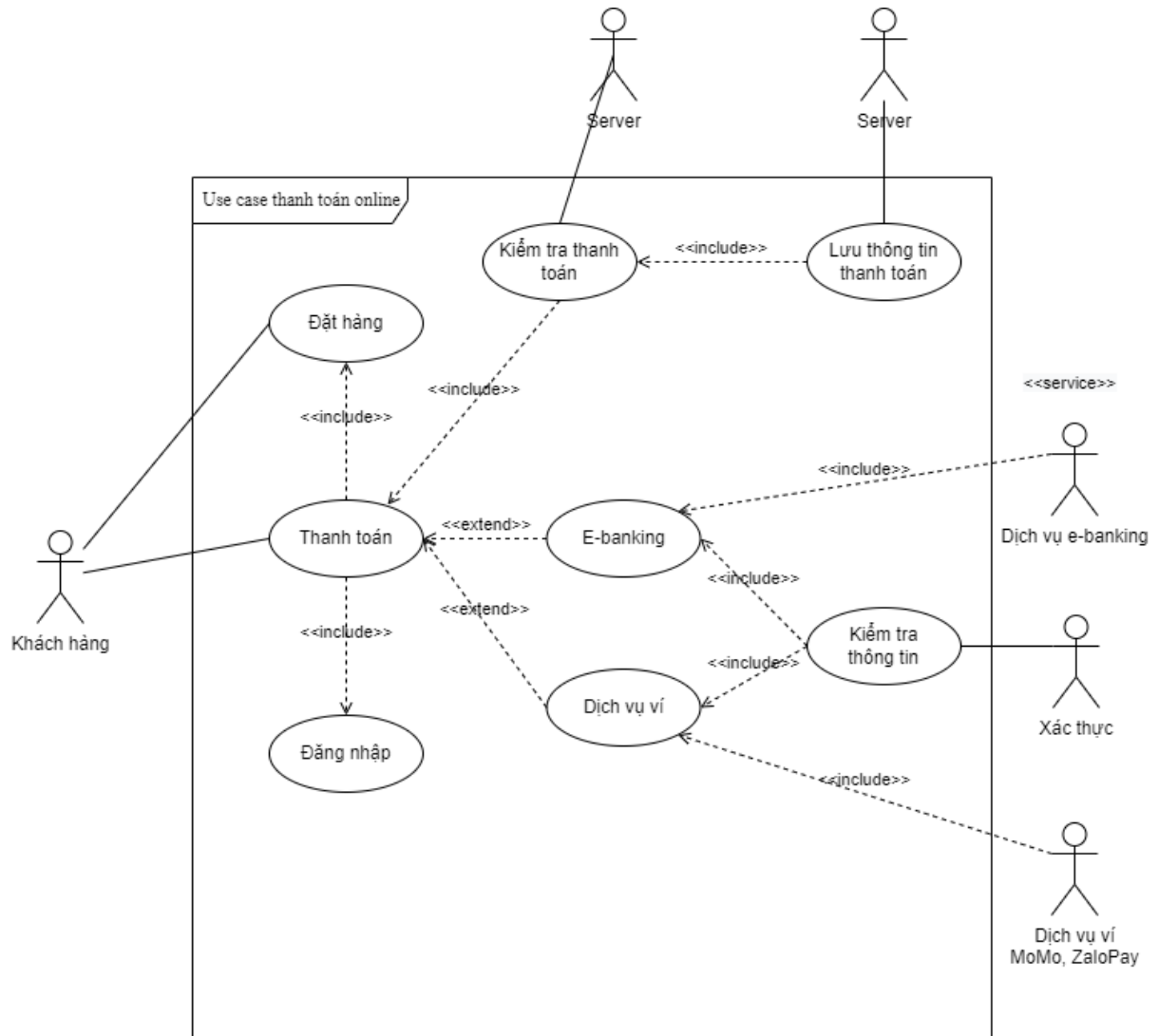
**c. Api/Service: Đặt hàng**

- Inputs: *đơn hàng, chi tiết đơn hàng, chi tiết thanh toán*
- Outputs: *thông báo, http code,*
- Xử lý:
  - Nhận dữ liệu đơn hàng được gửi tới từ client, thông tin đơn hàng sẽ được kiểm tra, xác thực thông tin → lưu dữ liệu vào trong database → Hệ thống sẽ gửi mail chứa thông tin đơn hàng khách hàng. Gửi phản hồi đã đặt hàng thành công cho phía client.

**d. DBMS Procedure**

- Inputs: *đơn hàng, chi tiết đơn hàng, chi tiết thanh toán, chi tiết dịch vụ*
- Outputs: *orderId, message*
- Tables: *order, order\_detail, transaction* (trường hợp khách hàng đó chọn hình thức thanh toán online)

## 1.5. Usecase thanh toán online



### e. Usecase 05: thanh toán online

- Vai trò: Người dùng sử dụng các dịch vụ thanh toán online để thực hiện giao dịch
- Tương tác: với API do dịch vụ thanh toán online đó cung cấp

**f. Form 05: Thanh toán online**

- Giao tiếp với user: Người dùng lựa chọn hình thức thanh toán online và thực hiện thanh toán.
- Các control:

Tên control	Input	Output	Xử lý
<b>Thanh toán online</b>	- Các yêu cầu của hình thức thanh toán online đó.	- Xác nhận thanh toán của dịch vụ	- Sử dụng các API, dịch vụ do nhà phát hành dịch vụ thanh toán đó phát hành để xử lý.  - Lưu lại thông tin thanh toán online vào hệ thống để lưu trữ.
<b>Thanh toán dịch vụ e-banking</b>	- Thông tin thanh toán do người dùng cung cấp của ngân hàng đó	- Xác nhận thanh toán của ngân hàng	- Kiểm tra thông tin ngân hàng do người dùng nhập: số tài khoản, tên người dùng, ngân hàng, ...  - Gửi thông tin đó sử dụng API do ngân hàng cung cấp, và nhận về thông báo xác nhận đã thanh toán của ngân hàng.
<b>Thanh toán dịch vụ ví</b>	- Thông tin thanh toán do người dùng cung cấp phù hợp yêu cầu của dịch vụ ví (ZaloPay, MoMo,...)	- Xác nhận thanh toán của dịch vụ ví	- Kiểm tra thông tin do người dùng nhập đầy đủ với thông tin mà dịch vụ ví đó yêu cầu  - Gửi thông tin đó sử dụng API do dịch vụ ví cung cấp, và nhận về thông báo xác nhận đã thanh toán của dịch vụ ví đó.

**Kiểm tra**

Nội dung	Rỗng	Xử lý
Chọn hình thức thanh toán	Không	- Rỗng → Yêu cầu chọn hình thức thanh toán
Thông tin thanh toán	Không	- Đối với từng loại hình thức thanh toán khác nhau sẽ có kiểm tra thông tin nhập phù hợp với yêu cầu của dịch vụ thanh toán online:  + Dịch vụ e-banking (sử dụng thẻ ngân hàng): số tài khoản, tên người dùng, ngân hàng, ...

		+ Dịch vụ ví (ZaloPay, MoMo): đầy đủ thông tin mà dịch vụ ví đó yêu cầu khách hàng để thực hiện thanh toán.
--	--	---

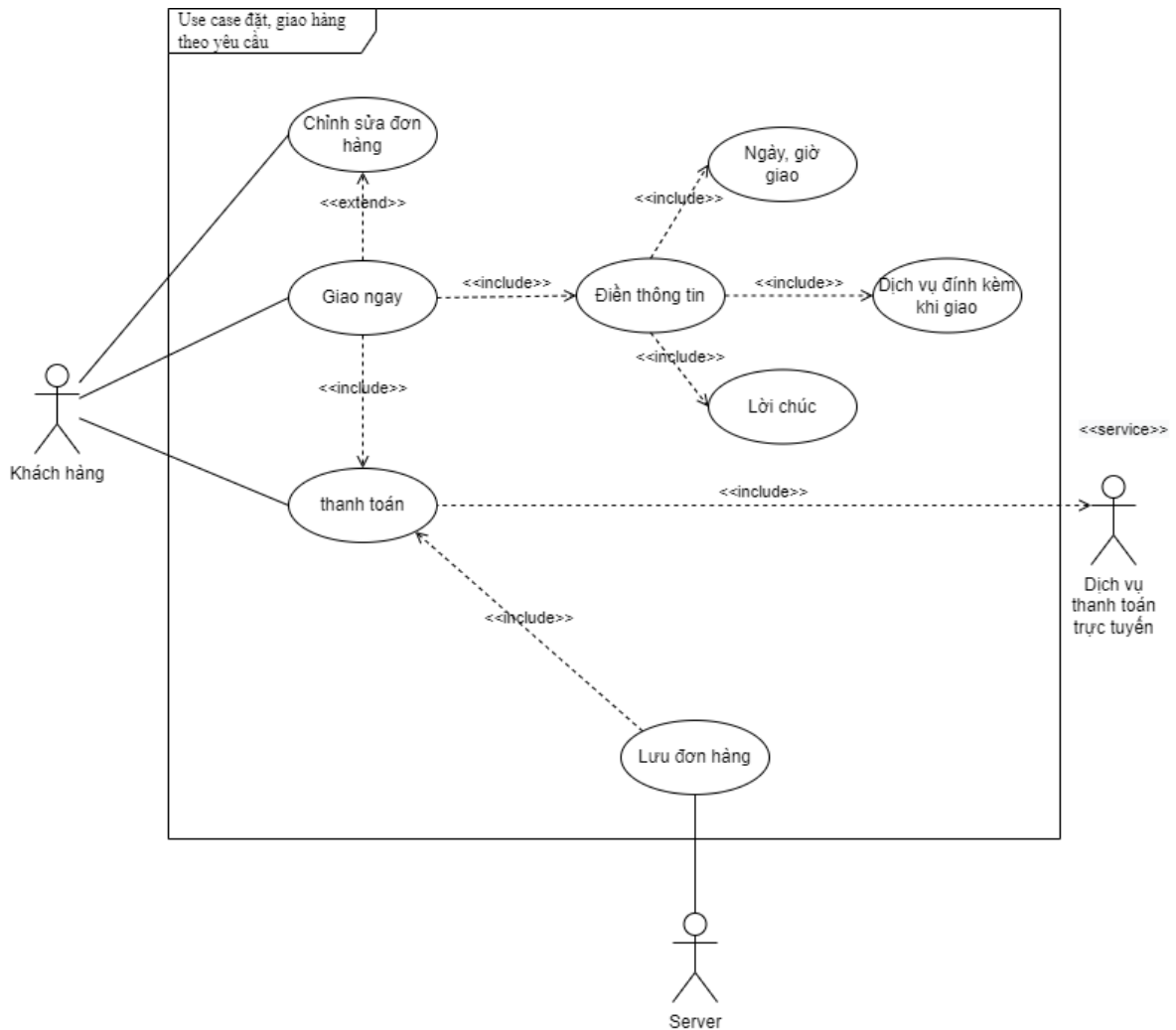
**g. Api/Service: Lưu thông tin thanh toán**

- Inputs: *thông tin thanh toán*.
- Outputs: *thông báo, http code*,
- Xử lý:
  - Nhận dữ liệu đơn hàng được gửi tới từ client, thông tin thanh toán sẽ được kiểm tra sau khi kiểm tra thông tin đơn hàng → lưu dữ liệu vào trong database → Hệ thống sẽ phản hồi thông báo việc lưu thông tin thanh toán vào database

**h. DBMS Procedure**

- Inputs: *chi tiết thanh toán*
- Outputs: transactionId, message
- Tables: transaction

## 1.6. Usecase đặt và giao hàng theo yêu cầu



### a. Usecase 06: Đặt hàng, giao hàng theo yêu cầu

- Vai trò: Người dùng đặt hàng và được lựa chọn các dịch vụ khi giao hàng.
- Tương tác: với form Đặt hàng và API đặt, giao hàng theo yêu cầu

### b. Form 06: Giỏ hàng

- Giao tiếp với user: Người dùng chọn sản phẩm và lưu lại trong giỏ hàng, tiến hành đặt giao ngay, chọn ngày, giờ giao, các dịch vụ đi kèm, yêu cầu thanh toán online.
- Các control:

Tên control	Input	Output	Xử lý
<b>Đặt hàng</b>	- Giỏ hàng	- Xác nhận đặt hàng từ hệ thống.	- Lấy thông tin đặt hàng. Tiến hành kiểm tra thông tin khách hàng nhập → Thông tin chưa hợp lệ, yêu cầu nhập lại.

			→ Thông tin hợp lệ, yêu cầu khách hàng chọn thanh toán online: → Sau khi thanh toán online thành công, xác nhận đặt hàng thành công
<b>Thanh toán online</b>	- Các yêu cầu của hình thức thanh toán online đó.	- Xác nhận thanh toán của dịch vụ	- Sử dụng các API, dịch vụ do nhà phát hành dịch vụ thanh toán đó phát hành để xử lý.

**c. Api/Service: Đặt, giao hàng giao theo yêu cầu**

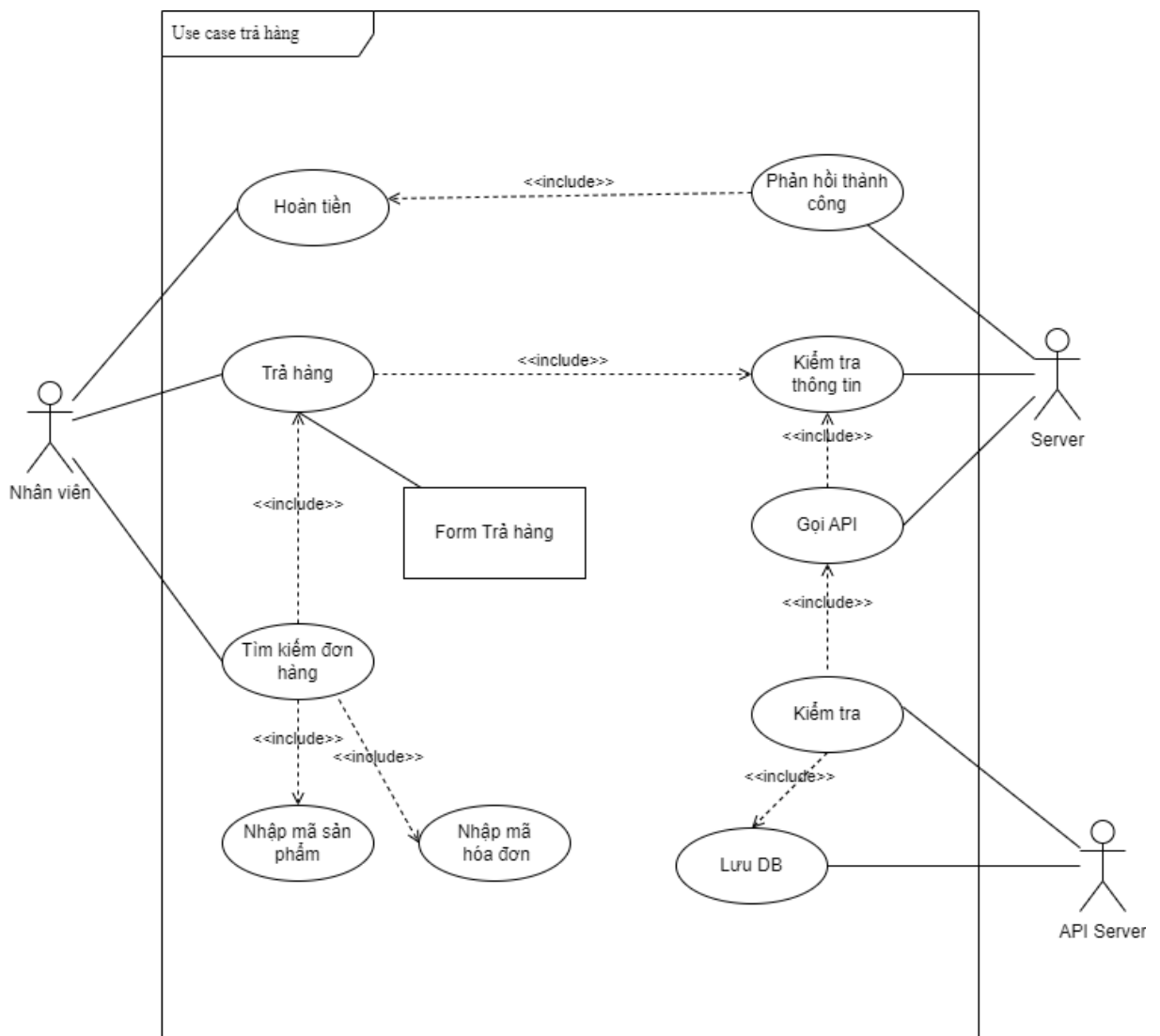
- Inputs: *đơn hàng, chi tiết đơn hàng, chi tiết thanh toán*
- Outputs: *thông báo, http code.*
- Xử lý:
  - Nhận dữ liệu đơn hàng được gửi tới từ client, thông tin đơn hàng sẽ được kiểm tra, xác thực thông tin → lưu dữ liệu vào trong database → Hệ thống sẽ gửi mail chứa thông tin đơn hàng khách hàng. Gửi phản hồi đã đặt hàng thành công cho phía client.

**d. DBMS Procedure**

- Inputs: *đơn hàng, chi tiết đơn hàng, chi tiết thanh toán*
- Outputs: *orderId, message*
- Tables: *order, order\_detail, transaction.*



## 1.7. Usecase trả hàng



**a. Usecase 07: usecase trả hàng**

- Vai trò: Khách hàng trả hàng, đổi hàng tại cửa hàng
- Tương tác: Tương tác với form Return và Return API.

**b. Form 07: Form Return**

- Giao tiếp với user: nhân viên tạo đơn trả hàng, và thực hiện trả hàng cho khách hàng.
- Các control:

Tên control	Input	Output	Xử lý
<b>Đổi hàng</b>	<ul style="list-style-type: none"> <li>- mã đơn hàng</li> <li>- mã sản phẩm</li> </ul>	- Cho phép trả hàng và hoàn trả tiền lại cho khách hàng.	- Sau khi nhận được mã đơn hàng và mã sản phẩm từ khách hàng tiến hành xác thực kiểm tra thông tin vừa nhập bằng API getOrder.

			→ Nếu thông tin không trùng khớp thì sẽ không cho đổi hàng → Nếu thông tin trùng khớp thì cho phép thực hiện trả hàng, nhân viên sẽ trả lại tiền cho khách hàng.
--	--	--	---

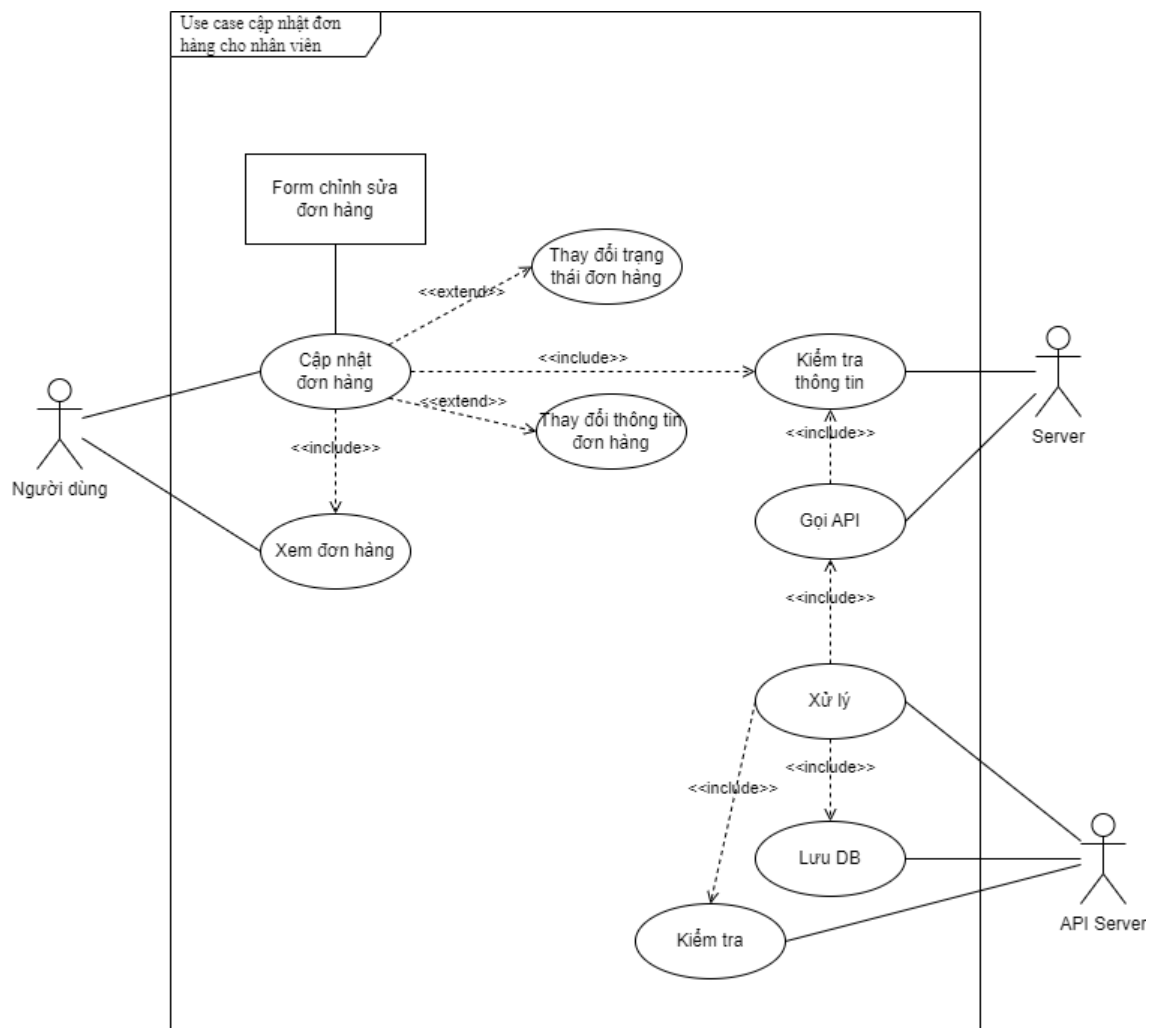
**c. Api/Service: API return**

- Inputs: *mã đơn hàng, mã sản phẩm*
- Outputs: *thông báo, http code*
- Xử lý:
  - Kiểm tra mã đơn hàng và mã sản phẩm được gửi lên từ client có hợp lệ, sản phẩm có tồn tại trong đơn hàng hay không.
  - Nếu sản phẩm có tồn tại trong đơn hàng thì cho phép trả hàng, lưu thông tin trả hàng vào trong database, trả về success code và thông báo nhân viên hoàn tiền lại cho khách hàng.
  - Nếu sản phẩm hoặc đơn hàng không hợp lệ thì trả về error code và thông báo đơn hàng không hợp lệ

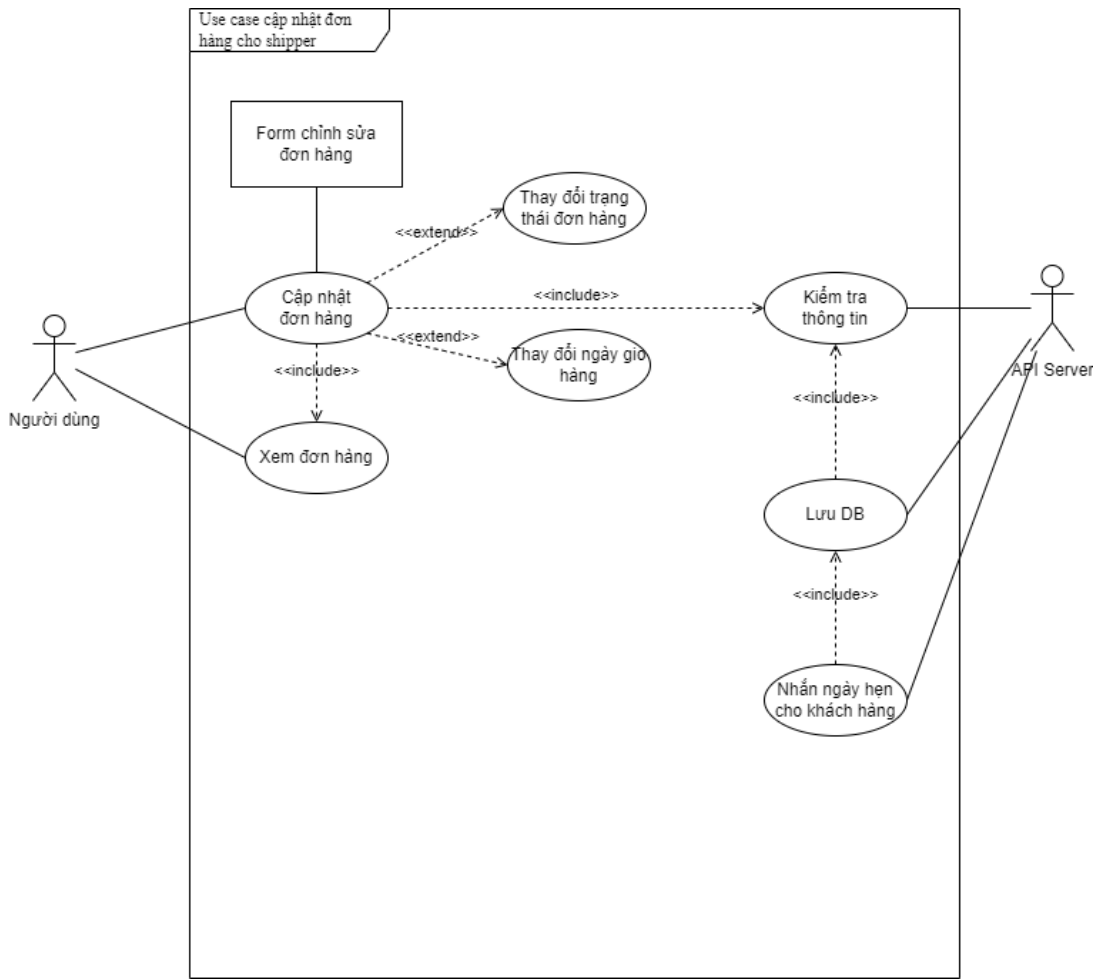
**d. DBMS Procedure**

- Inputs: *orderId, productId*
- Outputs: đánh dấu sản phẩm trong đơn hàng sẽ được hoàn tiền.
- Tables: order, order\_detail, transaction.

## 1.8. Usecase cập nhật đơn hàng



*Lược đồ usecase cập nhật đơn hàng của nhân viên*



Lược đồ usecase cập nhật đơn hàng của shipper

**a. Usecase 08: cập nhật đơn hàng**

- Vai trò: Nhân viên thay đổi trạng thái đơn hàng, shipper thay đổi ngày giao hàng
- Tương tác: API cập nhật đơn hàng, form cập nhật đơn hàng

**b. Form 08: cập nhật đơn hàng**

- Hình của form:

**Thông tin đơn hàng**

- Ngày đặt: 14/11/2021
- Giờ đặt: 00:00:00
- Tổng số sản phẩm: 8
- Tổng thành tiền: 440.000 đ

**Thông tin người đặt**

- Họ tên: phat luu
- Địa chỉ: tphcm
- Số điện thoại: 0796919429
- Ghi chú:

**Danh sách các sản phẩm**

STT	#	TÊN SẢN PHẨM	LOẠI	ĐƠN VỊ TÍNH	SL TỒN	SL YÊU CẦU	GIÁ	THÀNH TIỀN
1	P1944	Thắt lưng dây nịt kim loại mắt xích basic 110cm - Gold	Quà tặng	chiếc	9	1	70.000 đ	70.000 đ
2	P1944	Thắt lưng dây nịt kim loại mắt xích basic 110cm - Gold	Quà tặng	chiếc	9	1	70.000 đ	70.000 đ
3	P1945	Khẩu trang MJ Astronaut space happy	Khẩu trang	cái	8	2	40.000 đ	80.000 đ
4	P1944	Thắt lưng dây nịt kim loại mắt xích basic 110cm - Gold	Quà tặng	chiếc	9	1	70.000 đ	70.000 đ
5	P1944	Thắt lưng dây nịt kim loại mắt xích basic 110cm - Gold	Quà tặng	chiếc	9	1	70.000 đ	70.000 đ
6	P1945	Khẩu trang MJ Astronaut space happy	Khẩu trang	cái	8	2	40.000 đ	80.000 đ

Trạng thái đơn hàng

Hủy đơn hàng

Đơn hàng mới đến

Khách hàng đã xác nhận

Đang xử lý

Đã hoàn thành

- Giao tiếp với user: nhân viên trực tiếp cập nhật trạng thái đơn hàng trên form, shipper thay đổi ngày giờ giao hàng.
- Các control:

Tên control	Input	Output	Xử lý
Đơn hàng mới đến	Trạng thái đơn hàng	Trạng thái được thay đổi	- Đơn hàng mới được khởi tạo, trạng thái của đơn hàng là 0.
Hủy đơn hàng	Trạng thái đơn hàng	Trạng thái được thay đổi	- Đơn hàng bị hủy, trạng thái của đơn hàng là -1.
Khách hàng đã xác nhận	Trạng thái đơn hàng	Trạng thái được thay đổi	- Đơn hàng được xác nhận, trạng thái của đơn hàng là 1.
Đang xử lý	Trạng thái đơn hàng	Trạng thái được thay đổi	- Đơn hàng đang trong quá trình xử lý (đóng gói, xuất kho), trạng thái của đơn hàng là 2.
Đang giao hàng	Trạng thái đơn hàng	Trạng thái được thay đổi	- Đơn hàng đang được giao, trạng thái của đơn hàng là 3.
Đã hoàn thành	Trạng thái đơn hàng	Trạng thái được thay đổi	- Đơn hàng đang được giao hàng, trạng thái của đơn hàng là 4.
Cập nhật thời gian giao	Ngày/ giờ mới để giao	Thời gian giao hàng được thay đổi.	- Nhận được thời gian hẹn giao của khách hàng. Kiểm tra thời gian giao là hợp lệ. → Gọi API cập nhật đơn hàng để thay đổi thời gian giao hàng, → Gửi tin nhắn đến số điện thoại khách hàng ngày, giờ hẹn, lý do.

Kiểm tra:

Nội dung	Rỗng	Xử lý
Trạng thái	Không	- Đối với các trạng thái lớn hơn 0, chỉ có thể thay đổi theo chiều tăng dần chứ không thể đổi theo chiều ngược lại, - Khi đã hủy đơn hàng thì không thể thay đổi trạng thái khác được.
Ngày/giờ	Không	- Ngày giờ hẹn phải thời gian trong tương lai và không được quá một tuần.

**c. Api/Service: cập nhật đơn hàng**

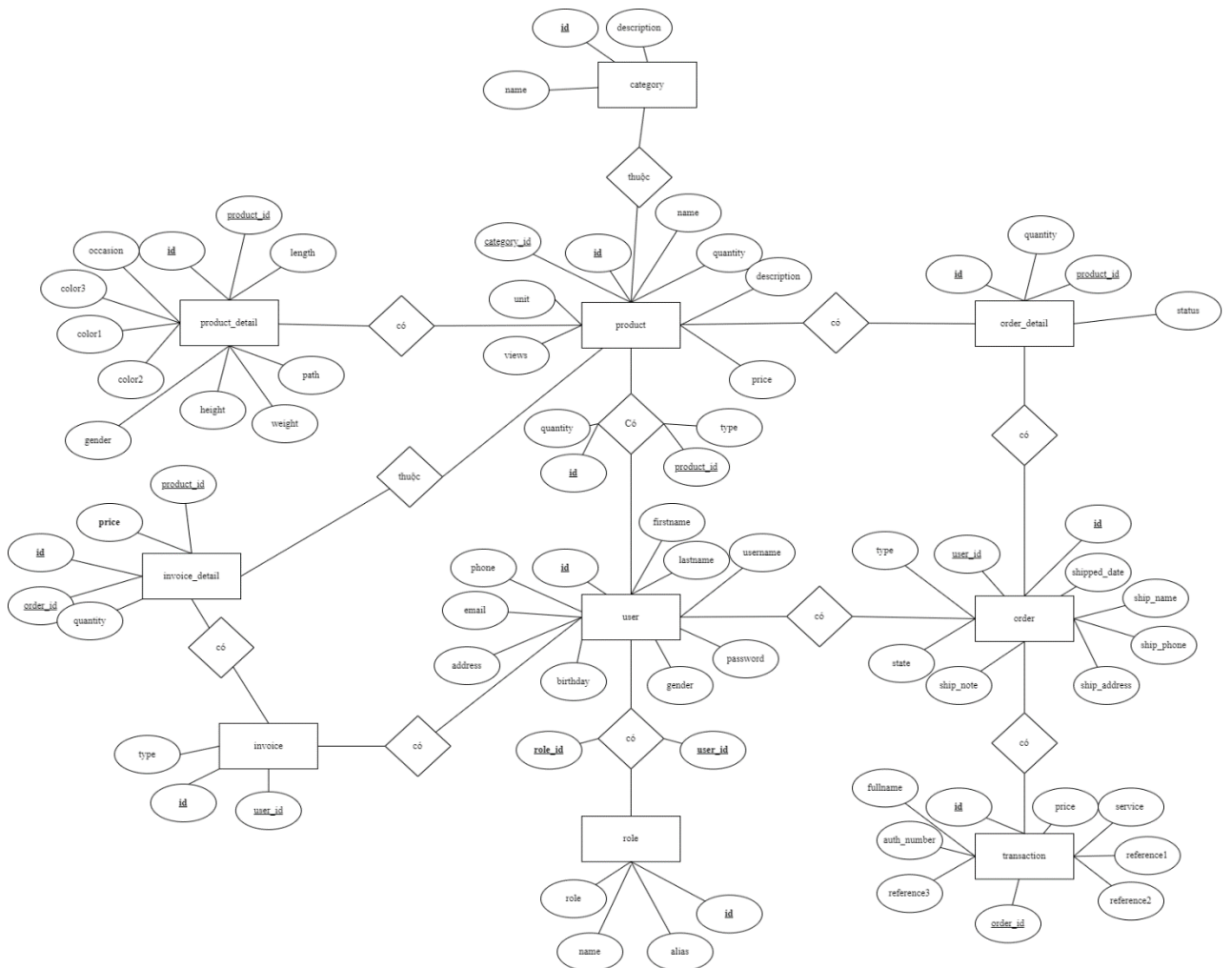
- Inputs: *trạng thái, ngày giờ hẹn, orderId*
- Outputs: *thông báo, http code.*
- Xử lý:
  - Nhận được yêu cầu thay đổi thông tin đơn hàng từ client, kiểm tra thông tin được thay đổi có hợp lệ hay không.
    - Nếu thông tin hợp lệ, tiến hành cập nhật đơn hàng đó trên database, trả về client success code và thông báo thay đổi thành công.
    - Nếu thông tin không hợp lệ thì sẽ trả về client error code và thông báo không thể cập nhật đơn hàng.

**d. DBMS Procedure**

- Inputs: *trạng thái, ngày giờ hẹn, orderId*
- Outputs: *order yêu cầu được cập nhật*
- Tables: *order*

## 2. Cơ sở dữ liệu và ràng buộc

### 2.1. Lược đồ ERD:



Lược đồ ERD

### 2.2. Bảng:

- **category** (id, name, description, created\_date, modified\_date)
- **product** (id, name, quantity, unit, price, description, views, category\_id, created\_date, modified\_date)
- **product\_detail** (id, product\_id, color1, color2, color3, height, weight, length, gender, occasion)
- **user** (id, firstname, lastname, username, password, birthday, gender, email, phone, address, created\_date, modified\_date)
- **order** (id, user\_id, shipped\_date, ship\_name, ship\_phone, ship\_address, ship\_note, state, type, created\_date, modified\_date)
- **order\_detail** (id, order\_id, product\_id, quantity, created\_date, modified\_date)
- **invoice\_type** (id, name, description, created\_date, modified\_date)
- **invoice** (id, user\_id, invoice\_type\_id, created\_date, modified\_date)
- **invoice\_detail** (id, invoice\_id, product\_id, quantity, price, created\_date, modified\_date)

- **transaction** (id, order\_id, fullname, auth\_number, price, service, reference1, reference2, reference3, created\_date, modified\_date)
- **role** (id, name, alias, created\_date, modified\_date)
- **user\_roles** (user\_id, roles\_id, created\_date, modified\_date)
- **user\_products** (id, user\_id, product\_id, quantity, type, created\_date, modified\_date)
- **Bảng category**

	Cột	Kiểu dữ liệu	Nội dung	Ghi Chú
1	id	BIGINT, PK	mã danh mục	
2	name	VARCHAR(50)	tên	
3	description	VARCHAR(50)	mô tả	
4	created_date	DATETIME	ngày tạo	
5	modified_date	DATETIME	ngày sửa	

- **Bảng product**

	Cột	Kiểu dữ liệu	Nội dung	Ghi Chú
1	id	BIGINT, PK	mã sản phẩm	
2	name	VARCHAR(50)	tên	
3	quantity	INT	số lượng	
4	unit	VARCHAR(50)	đơn vị	
5	price	FLOAT	đơn giá	
6	description	VARCHAR(50)	mô tả	
7	views	INT	lượt xem	
8	category_id	BIGINT, FK	mã danh mục	
9	created_date	DATETIME	ngày tạo	
10	modified_date	DATETIME	ngày sửa	

- **Bảng product\_detail**

	Cột	Kiểu dữ liệu	Nội dung	Ghi Chú
1	id	BIGINT, PK	mã	
2	product_id	BIGINT, PK	mã sản phẩm	



2	color1	VARCHAR(50)	màu chủ đạo 1	
3	color2	VARCHAR(50)	màu chủ đạo 2	
4	color3	VARCHAR(50)	màu chủ đạo 3	
3	height	FLOAT	số lượng	
4	weight	FLOAT	đơn vị	
5	length	FLOAT	chiều dài	
6	gender	FLOAT	thích hợp với giới tính	
7	occasion	VARCHAR(50)	Thích hợp cho dịp lễ	

- **Bảng user**

	Cột	Kiểu dữ liệu	Nội dung	Ghi Chú
1	id	BIGINT, PK	mã người dùng	
2	firstname	VARCHAR(50)	tên	
3	lastname	VARCHAR(50)	họ	
4	username	VARCHAR(255)	tên đăng nhập	
5	password	VARCHAR(255)	mật khẩu	
6	birthday	DATE	ngày sinh	
7	gender	INT	giới tính	
8	email	VARCHAR(45)	email	
9	phone	VARCHAR(10)	điện thoại	
10	address	VARCHAR(255)	địa chỉ	
11	created_date	DATETIME	ngày tạo	
12	modified_date	DATETIME	ngày sửa	

- **Bảng user\_products**

	Cột	Kiểu dữ liệu	Nội dung	Ghi Chú
1	id	BIGINT, PK	mã	
2	userId	BIGINT, FK	mã người dùng	

3	productId	BIGINT, FK	mã sản phẩm	
4	quantity	INT	Số lượng	
5	type	VARCHAR(50)	Tloại	
6	created_date	DATETIME	ngày tạo	
7	modified_date	DATETIME	ngày sửa	

- **Bảng order**

	Cột	Kiểu dữ liệu	Nội dung	Ghi Chú
1	id	BIGINT, PK	mã đơn hàng	
2	user_id	BIGINT, FK	mã user	
3	shipped_date	DATETIME	ngày giao	
4	ship_name	VARCHAR(50)	tên giao hàng	
5	ship_phone	VARCHAR(10)	số điện thoại	
6	ship_address	VARCHAR(255)	địa chỉ giao	
7	ship_note	VARCHAR(50)	ghi chú	
8	state	INT	trạng thái	
10	type	VARCHAR(50)	Loại order	
11	order_date	DATETIME	ngày tạo	
12	created_date	DATETIME	ngày tạo	
13	modified_date	DATETIME	ngày sửa	

- **Bảng order\_detail**

	Cột	Kiểu dữ liệu	Nội dung	Ghi Chú
1	id	BIGINT, PK	mã chi tiết đơn hàng	
2	order_id	BIGINT, FK	mã đơn	
3	product_id	BIGINT, FK	mã sản phẩm	
4	quantity	INT	số lượng	
4	status	VARCHAR(50)	trạng thái	

5	created_date	DATETIME	ngày tạo	
6	modified_date	DATETIME	ngày sửa	

- **Bảng invoice\_type**

	Cột	Kiểu dữ liệu	Nội dung	Ghi Chú
1	id	BIGINT, PK	mã loại hóa đơn	
2	name	VARCHAR(45)	tên loại hóa đơn	
3	created_date	DATETIME	ngày tạo	
4	modified_date	DATETIME	ngày sửa	

- **Bảng invoice**

	Cột	Kiểu dữ liệu	Nội dung	Ghi Chú
1	id	BIGINT, PK	mã hóa đơn	
2	user_id	BIGINT, FK	mã user	
3	invoice_type_id	BIGINT, FK	mã loại hóa đơn	
4	created_date	DATETIME	ngày tạo	
5	modified_date	DATETIME	ngày sửa	

- **Bảng invoice\_detail**

	Cột	Kiểu dữ liệu	Nội dung	Ghi Chú
1	id	BIGINT, PK	mã chi tiết hóa đơn	
2	invoice_id	BIGINT, FK	mã đơn	
3	product_id	BIGINT, FK	mã sản phẩm	
4	quantity	INT	số lượng	
5	price	FLOAT	giá tiền	
6	created_date	DATETIME	ngày tạo	
7	modified_date	DATETIME	ngày sửa	

- **Bảng transaction**

	Cột	Kiểu dữ liệu	Nội dung	Ghi Chú
1	id	BIGINT, PK	mã transaction	
2	order_id	BIGINT, FK	mã đơn hàng	
3	fullname	VARCHAR(50)	Tên người thanh toán	
4	auth_number	VARCHAR(50)	Mã xác thực	
5	price	FLOAT	Số tiền thanh toán	
6	service	VARCHAR(255)	Tên dịch vụ thanh toán	
7	reference1	VARCHAR(255)	Mã số 1	Cột để lưu trữ những giá trị đặc biệt không có trong bảng
8	reference2	VARCHAR(255)	Mã số 2	
9	reference3	VARCHAR(255)	Mã số 3	
10	created_date	DATETIME	ngày tạo	
11	modified_date	DATETIME	ngày sửa	

- **Bảng role**

	Cột	Kiểu dữ liệu	Nội dung	Ghi Chú
1	id	BIGINT, PK	mã quyền	
2	name	VARCHAR(255)	tên quyền	
3	alias	VARCHAR(255)	tên khác	

- **Bảng user\_roles**

	Cột	Kiểu dữ liệu	Nội dung	Ghi Chú
1	user_id	BIGINT, FK	mã người dùng	
2	roles_id	BIGINT, FK	mã quyền	
3	created_date	DATETIME	ngày tạo	
4	modified_date	DATETIME	ngày sửa	

### 1.9. Ràng buộc

Bảng 1	Bảng 2	Relationship	Bảng phụ
user	roles	n - n	user_roles
user	order	1 - n	
user	invoice	1 - n	
user	product	n - n	user_products
category	product	1 - n	
product	product_detail	1 - n	
order	order_detail	1 - n	
order	transaction	1 - 1	
invoice_type	invoice	1 - n	
invoice	invoice_detail	1 - n	

### 1.10. Phân quyền

Tên	alias	Chi tiết
ROLE_MANAGER	Manager	<ul style="list-style-type: none"> <li>- Có quyền như SALER.</li> <li>- Cho phép tạo tài khoản cho các quyền SALER và SHIPPER</li> <li>- Cho phép nhập/xuất các sản phẩm.</li> </ul>
ROLE_SALER	Saler	<ul style="list-style-type: none"> <li>- Tiếp nhận đơn đặt hàng, mua bán hàng, đổi trả hàng.</li> </ul>
ROLE_SHIPPER	Shipper	<ul style="list-style-type: none"> <li>- Truy cập vào ứng dụng giao hàng.</li> </ul>
ROLE_CUSTOMER	Customer	<ul style="list-style-type: none"> <li>- Đăng ký trên trang web của cửa hàng, chỉ có thể sử dụng ở trang bán hàng.</li> </ul>