

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC CẦN THƠ**  
**KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**



**LUẬN VĂN TỐT NGHIỆP**  
**NGÀNH CÔNG NGHỆ THÔNG TIN**

**Đề tài**

**XÂY DỰNG SÀN THƯƠNG MẠI ĐIỆN TỬ**  
**PHÂN HỆ: SỬ DỤNG CÔNG NGHỆ DJANGO**  
**THIẾT KẾ BACK-END VÀ CÔNG NGHỆ FLUTTER**  
**THIẾT KẾ ỨNG DỤNG MOBILE**

**(Building an e-commerce platform model Using Django to develop  
Back-end and Flutter to develop Mobile Application)**

*Giảng viên hướng dẫn:*

**ThS. NGUYỄN MINH TRUNG**

*Sinh viên thực hiện:*

**TRẦN TRỌNG NHÂN**

**MSSV: B1809493**

**KHÓA 44**

**Cần Thơ, 12/2022**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC CẦN THƠ**  
**KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**



**LUẬN VĂN TỐT NGHIỆP**  
**NGÀNH CÔNG NGHỆ THÔNG TIN**

**Đề tài**

**XÂY DỰNG SÀN THƯƠNG MẠI ĐIỆN TỬ**  
**PHÂN HỆ: SỬ DỤNG CÔNG NGHỆ DJANGO**  
**THIẾT KẾ BACK-END VÀ CÔNG NGHỆ FLUTTER**  
**THIẾT KẾ ỨNG DỤNG MOBILE**

**(Building an e-commerce platform model Using Django to develop**  
**Back-end and Flutter to develop Mobile Application)**

Giảng viên hướng dẫn:

**ThS. NGUYỄN MINH TRUNG**

Sinh viên thực hiện:

**TRẦN TRỌNG NHÂN**

**MSSV: B1809493**

**KHÓA 44**

**Cần Thơ, 12/2022**

TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG  
KHOA CÔNG NGHỆ THÔNG TIN

**XÁC NHẬN CHỈNH SỬA LUẬN VĂN  
THEO YÊU CẦU CỦA HỘI ĐỒNG**

Tên luận văn (tiếng Việt và tiếng Anh): XÂY DỰNG SÀN THƯƠNG MẠI  
ĐIỆN TỬ PHÂN HỆ SỬ DỤNG CÔNG NGHỆ DJANGO THIẾT KẾ BACK-END  
VÀ CÔNG NGHỆ FLUTTER THIẾT KẾ ỨNG DỤNG MOBILE / BUILDING AN  
E-COMMERCE PLATFORM MODEL USING DJANGO TO DEVELOP BACK-  
END AND FLUTTER TO DEVELOP MOBILE APPLICATION

Họ tên sinh viên: Trần Trọng Nhân

MASV: B1809493

Mã lớp: DI18V7A4

Đã báo cáo tại hội đồng ngành: Công nghệ thông tin

Ngày báo cáo: 12/12/2022

Hội đồng báo cáo gồm:

- |    |                         |                   |
|----|-------------------------|-------------------|
| 1. | T.S. Thái Minh Tuấn     | Chủ tịch hội đồng |
| 2. | T.S. Bùi Võ Quốc Bảo    | Thành viên        |
| 3. | Th.S. Nguyễn Minh Trung | Thư ký            |

Luận văn đã được chỉnh sửa theo góp ý của Hội đồng.

*Cần Thơ, ngày ..... tháng ..... năm 20...*

**Giáo viên hướng dẫn**

*(Ký và ghi họ tên)*

## LỜI CẢM ƠN

Lời nói đầu, em xin được phép cảm ơn Trường Đại học Cần Thơ, Khoa Công Nghệ Thông Tin và Truyền Thông đã tạo điều kiện cho phép em được thực hiện đề tài này.

Em xin cảm ơn gia đình, bạn bè đã ủng hộ và đồng hành cùng em trong suốt quá trình học tập tại trường. Em xin chân thành cảm ơn thầy cô đã giúp đỡ, gợi ý và hướng dẫn em để có thể hoàn thành được đề tài này. Đặc biệt, em xin gửi lời cảm ơn sâu sắc đến thầy Trung, người đã tận tình bỏ thời gian hướng dẫn cũng như hết sức tạo điều kiện giúp em hoàn thiện đề tài “Xây dựng sàn thương mại điện tử - phân hệ: Sử dụng công nghệ Django để thiết kế Back-end và công nghệ Flutter để thiết kế ứng dụng mobile”.

Với kinh nghiệm còn hạn chế, luận văn này chắc chắn sẽ có những thiếu sót. Em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của thầy cô để em có điều kiện bổ sung, nâng cao kiến thức của mình, phục vụ tốt hơn cho công tác thực tế sau này.

Em xin chân thành cảm ơn!

Trần Trọng Nhân

Lớp Công nghệ thông tin và truyền thông A4 K44

## NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

This image shows a full page of white paper with horizontal blue ruling lines. The lines are evenly spaced and run across the width of the page, providing a template for handwriting practice or general writing. There are no margins, text, or other markings on the page.

Cần Thơ, ngày tháng năm 2022

(Chữ ký của giảng viên)

## NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

[illegible]

Cần Thơ, ngày tháng năm 2022

(Chữ ký của giảng viên)

# MỤC LỤC

<b>PHẦN 1: GIỚI THIỆU .....</b>	<b>1</b>
1. ĐẶT VẤN ĐỀ .....	1
2. LỊCH SỬ GIẢI QUYẾT VẤN ĐỀ .....	1
3. MỤC TIÊU ĐỀ TÀI .....	2
4. ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU .....	2
4.1. Đối tượng nghiên cứu .....	2
4.2. Phạm vi nghiên cứu: .....	2
4.3. Phương pháp nghiên cứu: .....	3
5. NỘI DUNG NGHIÊN CỨU .....	3
<b>PHẦN 2: NỘI DUNG .....</b>	<b>4</b>
<b>CHƯƠNG 1: MÔ TẢ HỆ THỐNG .....</b>	<b>4</b>
1. MÔ TẢ ĐỀ TÀI .....	4
2. Các chức năng chính của hệ thống .....	4
3. Môi trường vận hành .....	5
4. Các ràng buộc về thực thi và thiết kế .....	5
<b>CHƯƠNG 2: THIẾT KẾ VÀ CÀI ĐẶT GIẢI PHÁP .....</b>	<b>7</b>
1. CƠ SỞ LÝ THUYẾT .....	7
MÔ HÌNH CLIENT - SERVER .....	7
MÔ HÌNH REST .....	7
1.1 Python .....	8
1.2 Django Framework .....	9
1.3. Django Rest Framework .....	10
1.4. Dart .....	10
1.5. Flutter .....	11

1.6. OpenStreetMap .....	12
1.7. PostgreSQL .....	12
1.8. Docker .....	13
2. THIẾT KẾ DỮ LIỆU HỆ THỐNG .....	13
2.1. Usecase diagram .....	13
2.1.1. Use case diagram của khách vãng lai .....	13
2.1.2. Use case diagram của khách hàng .....	14
2.1.3. Use case của đại lý .....	15
2.1.4. Use case của người giao hàng .....	16
2.1.5. Use case của người quản trị .....	17
2.2. Mô hình dữ liệu mức quan niệm (CDM) .....	18
2.3. Mô hình dữ liệu mức vật lý (PDM) .....	19
2.4. Các yêu cầu ràng buộc cần thiết khi xử lý dữ liệu .....	20
3. CÁC CHỨC NĂNG CHÍNH TRONG HỆ THỐNG .....	21
3.1. Chức năng mua và thanh toán giỏ hàng .....	21
3.2. Chức năng cập nhật trạng thái đơn hàng .....	22
3.3. Chức năng đánh giá sản phẩm .....	23
4. CÁC API CỦA HỆ THỐNG .....	23
4.1. Về sản phẩm .....	24
4.1.1. Xem sản phẩm .....	24
4.1.2. Xem chi tiết một sản phẩm .....	25
4.1.3. Thêm sản phẩm .....	25
4.1.4. Sửa sản phẩm .....	26
4.1.5. Xóa sản phẩm .....	27
4.1.6. Active sản phẩm .....	28
4.2. Giỏ hàng và đặt hàng .....	29



4.2.1. Thêm sản phẩm vào giỏ hàng .....	29
4.2.2. Đặt hàng .....	32
4.2.3. Thay đổi trạng thái đơn hàng .....	33
4.3. Đánh giá .....	34
4.3.1. Đánh giá sản phẩm .....	34
4.3.2. Active đánh giá .....	35
4.3.3. Xem đánh giá của sản phẩm .....	36
4.3.4. Phản hồi đánh giá .....	36
4.4. Shipper .....	38
4.4.1. Xem tất cả đơn hàng được chỉ định .....	38
4.4.2. Thêm ghi chú giao hàng .....	39
5. GIAO DIỆN CỦA APP NGƯỜI GIAO HÀNG .....	39
5.1. Trang chủ .....	39
5.2. Quản lý đơn hàng được chỉ định .....	40
5.3. Xem tất cả địa chỉ của đơn hàng .....	44
<b>PHẦN 3: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....</b>	<b>46</b>
1. KẾT LUẬN .....	46
1.1. Kết quả đạt được .....	46
1.2. Hạn chế .....	46
2. HƯỚNG PHÁT TRIỂN .....	47
<b>PHỤ LỤC .....</b>	<b>48</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>52</b>

## DANH MỤC HÌNH

Hình 1. Sơ đồ tổng quan về hệ thống (Mô hình Client-Server) (Nguồn: fptcloud.com)	7
Hình 2. Mô hình REST API (Nguồn: tma.vn) .....	8
Hình 3. Ngôn ngữ lập trình python (Nguồn: python.org) .....	8
Hình 4. Framework Django (Nguồn: djangoproject.com) .....	9
Hình 5. Django Rest Framework (Nguồn: django-rest-framework.org) .....	10
Hình 6. Ngôn ngữ lập trình Dart(Nguồn: dart.dev) .....	10
Hình 7. Flutter Framework (Nguồn: flutter.dev) .....	11
Hình 8. OpenStreetMap (Nguồn: openstreetmap.org) .....	12
Hình 9. PostgreSQL (Nguồn: kinsta.com) .....	12
Hình 10. Docker (Nguồn: docker.com) .....	13
Hình 11. Use case diagram của khách vãng lai .....	13
Hình 12. Use diagram của khách hàng .....	14
Hình 13. Use case của đại lý .....	15
Hình 14. Use case của người giao hàng .....	16
Hình 15. Use case của người quản trị .....	17
Hình 16. Mô hình dữ liệu mức quan niệm .....	18
Hình 17. Mô hình dữ liệu mức vật lý .....	19
Hình 18. Lưu đồ mua và thanh toán giỏ hàng .....	21
Hình 19. Lưu đồ chức năng cập nhật trạng thái đơn hàng .....	22
Hình 20. Lưu đồ chức năng đánh giá sản phẩm .....	23
Hình 21. Kết quả lấy tất cả sản phẩm .....	24
Hình 22. Xem chi tiết một sản phẩm .....	25

Hình 23. Đại lý thêm sản phẩm .....	26
Hình 24. Sửa sản phẩm .....	27
Hình 25. Xóa sản phẩm .....	28
Hình 26. Active sản phẩm .....	29
Hình 27. Thêm sản phẩm vào giỏ hàng .....	30
Hình 28. Xem thông tin giỏ hàng .....	31
Hình 29. Đặt hàng .....	32
Hình 30. Thay đổi trạng thái đơn hàng .....	33
Hình 31. Đánh giá sản phẩm .....	34
Hình 32. Active đánh giá .....	35
Hình 33. Xem đánh giá của sản phẩm .....	36
Hình 34. Phản hồi đánh giá .....	37
Hình 35. Xem các đơn hàng được chỉ định .....	38
Hình 36. Thêm ghi chú giao hàng .....	39
Hình 37. Trang chủ của app người giao hàng .....	40
Hình 38. Danh sách đơn hàng được chỉ định .....	41
Hình 39. Xem chi tiết của đơn hàng số 20 .....	42
Hình 40. Xem chỉ đường từ người giao hàng đến đơn hàng .....	43
Hình 41. Thay đổi trạng thái của đơn hàng .....	44
Hình 42. Xem tất cả địa chỉ của các đơn hàng .....	45
Hình 43. Cập nhật apt của Ubuntu .....	48
Hình 44. Thêm khóa GPG của Docker .....	49
Hình 45. Câu lệnh set up Docker .....	49

Hình 46. Câu lệnh cài đặt Docker .....	49
Hình 47. Cấu hình file Dockerfile .....	50
Hình 48. Cấu hình file docker-compose.yml .....	50
Hình 49. Build container .....	51
Hình 50. Liệt kê các container đang chạy hiện tại .....	51

## DANH MỤC TỪ VIẾT TẮT

Từ viết tắt	Viết đầy đủ
VPS	Vitural Private Server
OSM	Open Street Map
REST	Representational State Transfer
API	Application Programming Interface
Param	Parameter

## TÓM TẮT

Với sự phát triển không ngừng của công nghệ kèm theo nhu cầu được mua sắm sản phẩm của người dân cũng như là bán sản phẩm của các thương nhân. Thì việc làm sao để có thể mua, bán hàng một cách dễ dàng cũng là một vấn đề cần xem xét trong thời buổi hiện tại.

Nhìn thấy được việc này, chúng em đã quyết định thực hiện đề tài “Xây dựng sàn thương mại điện tử” để có thể mang đến một môi trường, nơi mà người mua và người bán có thể mua, bán sản phẩm một cách dễ dàng, thuận tiện. Hướng đến việc xây dựng website đơn giản, dễ sử dụng với người dùng. Website được xây dựng trên nền tảng Django làm Back-end và Angular làm Front-end, quản lý bởi hệ quản trị cơ sở dữ liệu PostgreSQL, website mong muốn sẽ đáp ứng được mọi nhu cầu của người dùng. Hệ thống sẽ giúp khách hàng có cái nhìn tổng quát về danh sách sản phẩm, cùng với đó là bộ công cụ tìm kiếm giúp khách hàng có thể tìm được sản phẩm một cách nhanh chóng. Website có hỗ trợ quản lý sản phẩm, quản lý tài khoản, quản lý bài viết, quản lý đơn hàng,... từ đó giúp cho người bán cũng như là người quản trị có thể cập nhật, thay đổi và xem xét một cách dễ dàng để phục vụ cho khách hàng cũng như tăng doanh thu cho hệ thống.

Hệ thống đã được cài đặt và hoạt động thành công trên nền tảng website cũng như là app giao hàng ở nền tảng mobile. Với những công nghệ được sử dụng và những tính năng của hệ thống, mong rằng đề tài sẽ có thể phát triển thêm để có thể áp dụng vào thực tiễn cũng như giúp ích được cho các doanh nghiệp và khách hàng.

## ABSTRACT

As technology continues to evolve, there is a need to buy products from people and sell products from merchants. Then how to be able to buy and sell easily is also a problem at present.

Seeing this, we decided to work on the topic of "Building an e-commerce platform" to create an environment where buyers and sellers can buy and sell products easily and convenient. Towards building a website that is simple and easy to use for users. The website using Django as the Back-end and Angular as the Front-end, managed by the PostgreSQL database management system, the website hopes to meet all the needs of users. The system will help customers have an overview of the product list, along with a search tool to help customers find products quickly. The website supports product management, account management, rating management, order management, etc.. helping sellers as well as administrators to update, change and review easily to serve customers as well as increase revenue for the system.

The system has been successfully installed and operated on the website platform as well as the delivery app on the mobile platform. With the technologies used and the features of the system, we hope that the topic can be further developed so that it can be applied in practice as well as helpful for businesses and customers.

## PHẦN 1: GIỚI THIỆU

### 1. ĐẶT VẤN ĐỀ

Trong thời buổi hiện tại, nhu cầu mua sắm của con người không dừng lại ở việc sản phẩm phải đẹp, chất lượng hay là giá cả phải chăng, mà còn phải là mua một cách thuận tiện cũng như không tốn nhiều thời gian.

Nắm bắt được những nhu cầu đó, thương mại điện tử ra đời như là một điều tất yếu trong bối cảnh hiện tại, nơi mà công nghệ phát triển một cách mạnh mẽ. Các website thương mại điện tử xuất hiện ngày càng nhiều, càng lúc càng thân thiện và dễ dàng tương tác với khách hàng, chỉ cần vài cú click chuột thôi là có thể mua được món hàng ưng ý trong khi không cần phải đi ra ngoài lựa chọn.

Bên cạnh đó, người bán hàng cũng có thể sử dụng website thương mại điện tử để có thể quản lý kho hàng, quản lý nhập hàng cũng như là xem thống kê của cửa hàng theo các mốc thời gian cụ thể như ngày, tuần, tháng, quý.... Chỉ cần là nơi nào có internet thì người bán có thể quản lý cửa hàng của mình bất cứ lúc nào.

Vì những lý do nêu trên, chúng em quyết định thực hiện đề tài Xây dựng sàn thương mại điện tử để có thể hiểu thêm về nghiệp vụ cũng như là thỏa mãn được sự tò mò đối với một ngành dịch vụ không mới cũng chẳng bao giờ cũ này.

### 2. LỊCH SỬ GIẢI QUYẾT VẤN ĐỀ

Cùng với sự phát triển không ngừng của công nghệ, thương mại điện tử cũng đang phát triển không ngừng để có thể trở thành một phương thức giao dịch phổ biến của người dùng khi mua sắm. Trước đề tài này, chúng ta có thể kể đến những website thương mại điện tử nổi tiếng như **alibaba.com**, **amazon.com**, **taobao.com**,.... Trong nước thì chúng ta có thể kể đến những website như **lazada.vn**, **tiki.vn**, **shopee.vn**,...

Những website vừa kể đến đây là những trang chuyên cung cấp những sản phẩm mà khách hàng có thể đặt mua trực tuyến trên mạng. Mỗi trang web đều có lượt truy cập rất cao trong mỗi tháng, cụ thể là hơn 2 tỷ người đối với **amazon.com**, hơn 90 triệu người đối với **alibaba.com**. Ở Việt Nam thì là hơn 100 triệu người đối với **shopee.vn** và hơn 15 triệu người với **tiki.vn** (Theo thống kê của website



[www.similarweb.com](http://www.similarweb.com)). Và tại khoa Công nghệ thông tin và truyền thông của trường đại học Cần Thơ cũng đã có rất nhiều anh chị khóa trước đã nghiên cứu cũng như xây dựng các mô hình thương mại điện tử.

### **3. MỤC TIÊU ĐỀ TÀI**

Xây dựng một trang web đơn giản, dễ dàng sử dụng đối với mọi lứa tuổi, có công cụ tìm kiếm sản phẩm nhanh và chính xác là mục tiêu hàng đầu của chúng em khi thực hiện đề tài **“Xây dựng sàn thương mại điện tử”** này.

Đối với người bán thì website phải giúp người bán dễ dàng quản lý, cập nhật sản phẩm cũng như quản lý kho hàng của mình. Ngoài ra hệ thống còn phải giúp người bán thống kê lợi nhuận, doanh thu theo tuần, tháng, năm một cách nhanh chóng, đơn giản và chính xác.

Đối với người giao hàng, app giao hàng phải đáp ứng nhu cầu xem thông tin của đơn hàng, giao diện đơn giản và dễ sử dụng. Có chỉ đường một cách chính xác và không được phản hồi yêu cầu quá lâu.

Còn đối với người quản trị, hệ thống cần phải đáp ứng đủ nhu cầu quản trị bao gồm quản lý tài khoản đăng ký, quản lý được sản phẩm nào được đăng ký,...Cũng như là thống kê được tỉ lệ tham quan web trung bình của người dùng và tỉ lệ quay lại của người dùng.

### **4. ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU**

#### **4.1. Đối tượng nghiên cứu**

Tìm hiểu các hệ thống sàn thương mại điện tử trên mạng để nắm bắt xu thế và nghiên cứu xây dựng ứng dụng, hệ thống trong lĩnh vực thương mại điện tử đã có trên thị trường hiện nay.

Tìm hiểu nhu cầu của người bán, người mua, người giao hàng cũng như là người quản trị trong website sàn thương mại điện tử.

#### **4.2. Phạm vi nghiên cứu:**

Đề tài tập trung giải quyết những vấn đề cơ bản trong sàn thương mại điện tử như mua sản phẩm, đăng bán sản phẩm, quản trị tài khoản người dùng, quản trị sản

phẩm được đăng bán, quản lý vận chuyển cũng như thay đổi trạng thái của đơn hàng. Được thực hiện từ ngày 08/08/2022 đến ngày 08/12/2022.

#### **4.3. Phương pháp nghiên cứu:**

Tìm hiểu và thiết kế cơ sở dữ liệu để có thể áp dụng vào hệ thống cũng như là áp dụng vào thực tế. Thêm vào đó là có thể dễ dàng bảo trì và nâng cấp sau này.

Tìm hiểu ngôn ngữ lập trình python, áp dụng Framework Django để xây dựng Back-end, tìm hiểu ngôn ngữ lập trình dart, áp dụng Framework Flutter để xây dựng ứng dụng mobile. Sử dụng hệ quản trị cơ sở dữ liệu PostgreSQL để xây dựng cơ sở dữ liệu, sử dụng Docker làm môi trường ảo để có thể deploy Back-end lên máy chủ ảo VPS (Virtual Private Server - VPS).

### **5. NỘI DUNG NGHIÊN CỨU**

Để xây dựng một website sàn thương mại điện tử một cách hoàn chỉnh, cần tìm hiểu và nghiên cứu các phần sau:

- Tìm hiểu kiến trúc mô hình Client-Server.
- Tìm hiểu mô hình REST ở phía back-end.
- Tìm hiểu và sử dụng ngôn ngữ Python cũng như framework Django để thiết kế ứng dụng phía Back-end.
- Tìm hiểu và sử dụng ngôn ngữ Dart cũng như framework Flutter để thiết kế ứng dụng mobile.
- Tìm hiểu hệ quản trị cơ sở dữ liệu PostgreSQL.
- Tìm hiểu nền tảng Docker để deploy Back-end.
- Tìm hiểu, nghiên cứu về quy trình quản lý sản phẩm để có thể xây dựng cơ sở dữ liệu hợp lý cho website.

## **PHẦN 2: NỘI DUNG**

### **CHƯƠNG 1: MÔ TẢ HỆ THỐNG**

#### **1. MÔ TẢ ĐỀ TÀI**

BNZone - “**Website Sàn thương mại điện tử**” là hệ thống bán các sản phẩm không giới hạn loại hình của sản phẩm dành cho các cửa hàng vừa và nhỏ, đáp ứng nhu cầu quản lý hàng hóa, quản lý xuất nhập kho,... Mỗi sản phẩm sẽ có những thông tin cơ bản như tên sản phẩm, hình ảnh của sản phẩm, giá tiền, những thông số chi tiết của sản phẩm, biến động giá của sản phẩm. Ngoài ra, khách hàng còn có thể xem đánh giá của người khác về sản phẩm định mua để từ đó có thể đưa ra lựa chọn mua hàng một cách hợp lý. Khách hàng cũng có thể đưa ra đánh giá cho những sản phẩm mà mình đã mua.

Hệ thống giúp cho người bán (từ bây giờ sẽ gọi là đại lý) nhận được thông báo về những đơn hàng mới, xem được những bình luận của khách hàng cũng như là độ hài lòng của họ. Có thể phản hồi được những đánh giá của khách hàng, giúp đại lý và khách hàng có thể tương tác được với nhau.

Trong hệ thống còn có người giao hàng, đây là khâu cuối cùng của thương mại điện tử tính từ lúc người dùng đặt hàng, đến đại lý xác nhận và gửi cho kho vận chuyên, thì người vận chuyển có nhiệm vụ đưa đơn hàng của khách đã đặt đến tay của khách hàng.

Còn về người quản trị, hệ thống có thể giúp được người quản trị website quản lý được tài khoản của khách hàng và đại lý, quản lý được bình luận của người dùng và quản lý được sản phẩm mà những đại lý đang bán.

#### **2. Các chức năng chính của hệ thống**

##### **- Khách vắng lai**

Là khách hàng chưa đăng ký tài khoản, họ chỉ có thể xem tên, giá và số lượng của sản phẩm. Có thể tìm kiếm được sản phẩm và không có quyền thêm sản phẩm vào giỏ hàng, cũng như là mua sản phẩm.

#### **- Khách hàng**

Là khách hàng đã đăng ký tài khoản, họ có thể xem tên, giá và số lượng của sản phẩm cũng như là xem được trang chi tiết của sản phẩm. Có thể tìm kiếm sản phẩm và thêm sản phẩm vào giỏ hàng cũng như là mua hàng, xem được trạng thái và lịch sử của đơn hàng. Cập nhật thông tin tài khoản. Có thể đưa ra ý kiến về một sản phẩm mà họ đã mua và phản hồi bình luận về một sản phẩm nào đó.

#### **- Đại lý**

Là người bán có tài khoản đại lý được người quản trị kích hoạt. Có thể xem thông tin chi tiết của sản phẩm, có thể phản hồi lại bình luận của khách hàng, thêm hàng hóa, thêm phiếu nhập để nhập hàng hóa.

#### **- Người giao hàng**

Là người có nhiệm vụ giao hàng đã đặt của khách hàng đến địa chỉ mà khách hàng đã chọn, có thể xem bản đồ, xem thông tin của đơn hàng nhưng không xem được sản phẩm của đơn hàng, là người thay đổi trạng thái cuối cùng của đơn hàng (giao thành công, hủy đơn).

#### **- Admin**

Là người quản trị của trang web có chức năng quản lý tài khoản, quản lý sản phẩm và quản lý đánh giá của khách hàng. Người quản trị cũng là người sẽ chỉ định vận chuyển cho đơn hàng của đại lý.

### **3. Môi trường vận hành**

- Website hoạt động tốt trên các trình duyệt Google Chrome, Cốc Cốc, Edge.
- Sử dụng Docker để deploy back-end lên VPS.
- Yêu cầu kết nối Internet.

### **4. Các ràng buộc về thực thi và thiết kế**

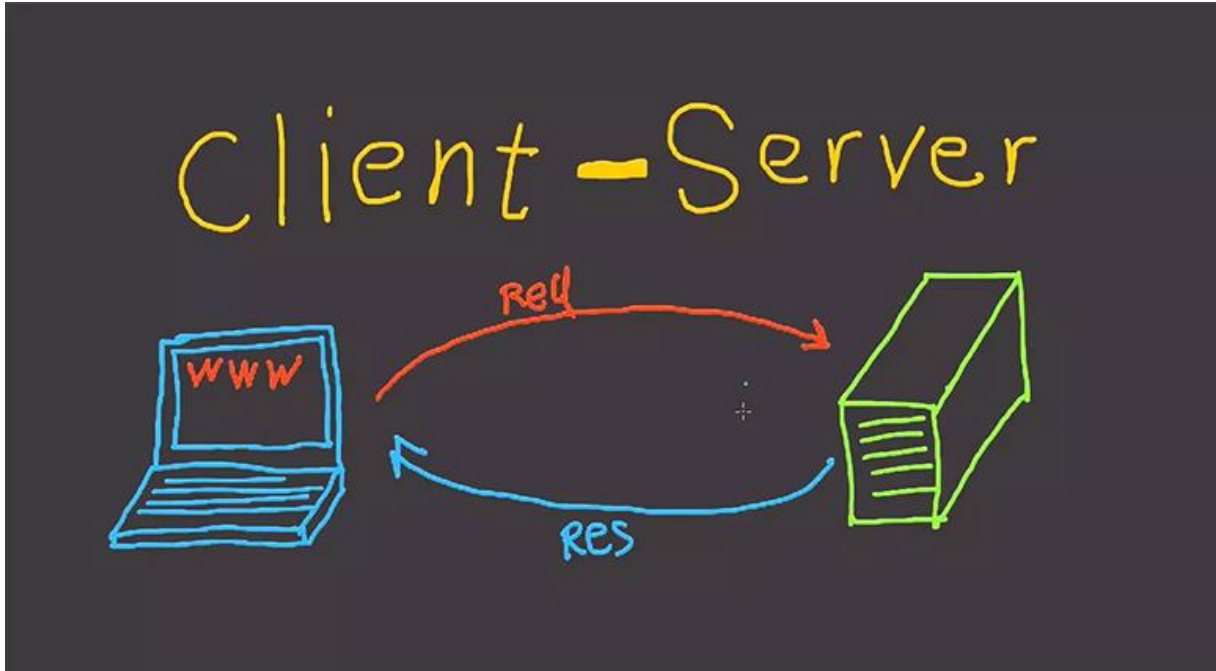
- Thời gian chờ phản hồi không quá 10 giây.
- Lập trình bằng ngôn ngữ python, dart.
- Sử dụng Django Framework.

- Sử dụng Flutter Framework cho ứng dụng điện thoại.
- Sử dụng Docker.
- Dùng PostgreSQL để làm cơ sở dữ liệu.
- Hệ thống đáp ứng khả năng an toàn, bảo mật thông tin.
- Cho phép ngăn chặn và từ chối một số truy cập không hợp lệ.
- Website phải dễ sử dụng, bảo trì và nâng cấp.
- Công cụ vận hành: Google Chrome.
- Công cụ hỗ trợ lập trình: Visual Code

## CHƯƠNG 2: THIẾT KẾ VÀ CÀI ĐẶT GIẢI PHÁP

### 1. CƠ SỞ LÝ THUYẾT

#### MÔ HÌNH CLIENT - SERVER



Hình 1. Sơ đồ tổng quan về hệ thống (Mô hình Client-Server) (Nguồn: fptcloud.com)

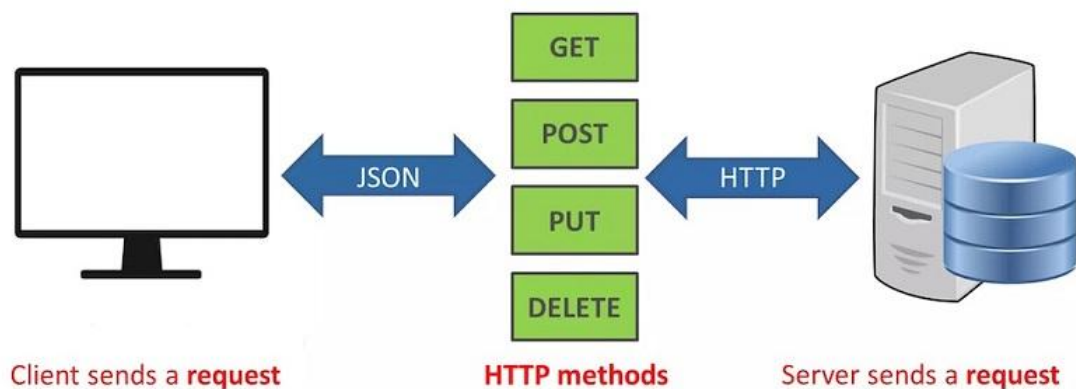
Nguyên tắc hoạt động của mô hình **Client-Server**:

- **Client**: là máy tính đóng vai trò là máy khách. Đối với client, chúng sẽ không cung cấp tài nguyên, dữ liệu gì đến những máy khác mà chỉ sử dụng tài nguyên, dữ liệu được gửi từ **server**.

- **Server**: là máy tính đóng vai trò là máy chủ. Đối với server, chúng sẽ cung cấp dữ liệu, tài nguyên đến những máy tính khác khi được gửi yêu cầu. **Server** cũng đóng vai trò giúp cho các hoạt động trên máy khách được diễn ra trơn tru, hiệu quả hơn.

#### MÔ HÌNH REST

Về phía back-end, ở đây chúng em sẽ xây dựng API theo mô hình REST để có thể dễ dàng trao đổi dữ liệu giữa client và server.



Hình 2. Mô hình REST API (Nguồn: tma.vn)

REST API còn gọi là RESTful API, là một giao diện để lập trình ứng dụng, được tạo ra bởi nhà khoa học máy tính Roy Fielding.

REST API bao gồm 2 thành phần chính:

- **API (Application Programming Interface)** dịch ra là giao diện lập trình ứng dụng, giao diện này sẽ được người lập trình sử dụng.

- **REST (REpresentational State Transfer)** là một đại diện cho sự chuyển đổi dữ liệu. Mỗi một request **REST API** đều sẽ không mang trạng thái trước đó của nó, như vậy, để trao đổi **state** với nhau, thì chúng phải buộc thông qua các **resources** và những **resource** này là đại diện cho sự thay đổi dữ liệu (Nguồn: tma.vn)

## 1.1 Python



Hình 3. Ngôn ngữ lập trình python (Nguồn: python.org)

Python là ngôn ngữ lập trình được sử dụng rộng rãi trong các ứng dụng web, phát triển phần mềm, khoa học dữ liệu và máy học (Machine Learning - ML) được ra mắt lần đầu tiên vào năm 1991, do Guido van Rossum tạo ra. Python được thiết kế với

ưu điểm là dễ đọc, dễ học, dễ nhớ. Và python cũng là ngôn ngữ phổ biến 3 trong năm 2021 (do trang **stackoverflow** thống kê).

## 1.2 Django Framework



*Hình 4. Framework Django (Nguồn: [djangoproject.com](https://djangoproject.com))*

Django là một web framework mã nguồn mở được viết bằng Python, được thiết kế để đáp ứng nhu cầu phát triển website an toàn và dễ dàng bảo trì. Được ra mắt vào năm 2005, do Django Software Foundation phát triển.

### **Ưu điểm:**

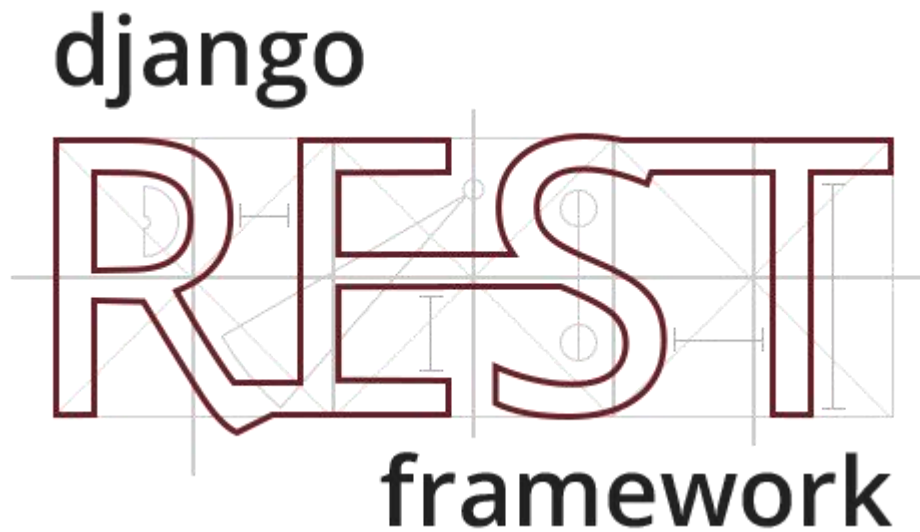
- Đơn giản và nhanh chóng.
- Có tính bảo mật cao.
- Phù hợp với mọi dự án website.
- Có cộng đồng người dùng rộng lớn.
- Có kho tài liệu thư viện được cập nhật một cách chính xác và liên tục.

### **Nhược điểm:**

- Chỉ định url bằng các quy tắc biểu thức không dễ để thực hiện (đặc biệt đối với người mới bắt đầu tìm hiểu).
- Mẫu lỗi không phải là mẫu lỗi thông báo, đôi khi mất rất nhiều thời gian để có thể tìm ra được lỗi do nó không chỉ định đến dòng lỗi nào mà chỉ về những dòng lỗi trong thư viện của chính nó.
- Đôi khi hơi cồng kềnh đối với những dự án nhỏ.



### 1.3. Django Rest Framework



*Hình 5. Django Rest Framework (Nguồn: [django-rest-framework.org](https://django-rest-framework.org))*

Django Rest framework là một bộ công cụ mạnh mẽ và linh hoạt trong việc xây dựng một ứng dụng web API.

### 1.4. Dart



*Hình 6. Ngôn ngữ lập trình Dart(Nguồn: [dart.dev](https://dart.dev))*

Dart là ngôn ngữ lập trình web do Google phát triển được ra mắt vào năm 2012.

## 1.5. Flutter



*Hình 7. Flutter Framework (Nguồn: flutter.dev)*

Flutter là một framework dùng để phát triển ứng dụng web và mobile, sử dụng ngôn ngữ Dart. Được Google phát triển vào năm 2017. Đây là một dự án mã nguồn mở và hoàn toàn miễn phí.

### **Ưu điểm:**

- Nhanh chóng.
- Có “Hot reload” chức năng giúp thực hiện các thay đổi đối với mã code và xem được kết quả ngay lập tức.
- Tùy chỉnh toàn bộ và kết xuất nhanh.
- Đa nền tảng, một lần code có thể chạy lên nhiều nền tảng khác nhau như web, android, ios.

### **Nhược điểm:**

- Còn khá mới, đôi khi thiếu tài liệu tham khảo.
- Ứng dụng khi build ra khá nặng.

## 1.6. OpenStreetMap



*Hình 8. OpenStreetMap (Nguồn: [openstreetmap.org](http://openstreetmap.org))*

OpenStreetMap(OSM) là dịch vụ bản đồ thế giới mở hoàn toàn miễn phí cho các ứng dụng sử dụng bản đồ hoặc chỉ đường. Ở Việt Nam có các ứng dụng lớn sử dụng OSM như là nền tảng cho ứng dụng của mình như BusMap, Grab, ...

## 1.7. PostgreSQL

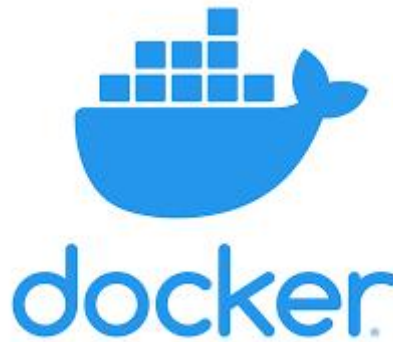


*Hình 9. PostgreSQL (Nguồn: [kinsta.com](http://kinsta.com))*

PostgreSQL là hệ quản trị cơ sở dữ liệu mã nguồn mở được khoa điện toán của đại học California tại Berkeley phát triển. PostgreSQL được xây dựng theo chuẩn

SQL99. Người dùng có thể sử dụng tự do, chỉnh sửa và phân bổ theo nhiều hình thức khác nhau. So với các hệ quản trị cơ sở dữ liệu khác thì PostgreSQL không yêu cầu quá cao về việc bảo trì do tính ổn định, có thể phát triển nhiều ứng dụng khác nhau của nó.

## 1.8. Docker



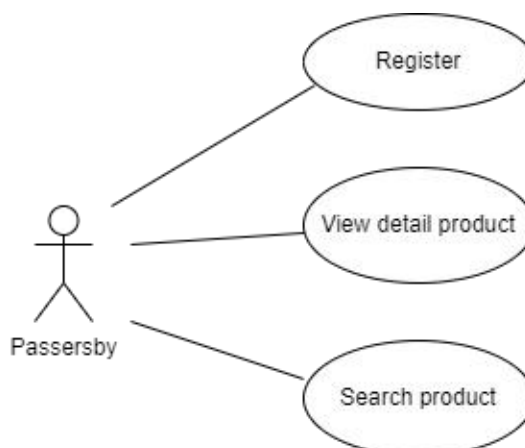
Hình 10. Docker (Nguồn: [docker.com](https://docker.com))

Docker là một nền tảng cung cấp cách để xây dựng, triển khai và thực thi ứng dụng dễ dàng hơn bằng cách sử dụng các container (trên nền tảng ảo hóa) được viết bằng ngôn ngữ Go, ra mắt vào năm 2013 và được phát triển bởi nhóm Docker.

## 2. THIẾT KẾ DỮ LIỆU HỆ THỐNG

### 2.1. Usecase diagram

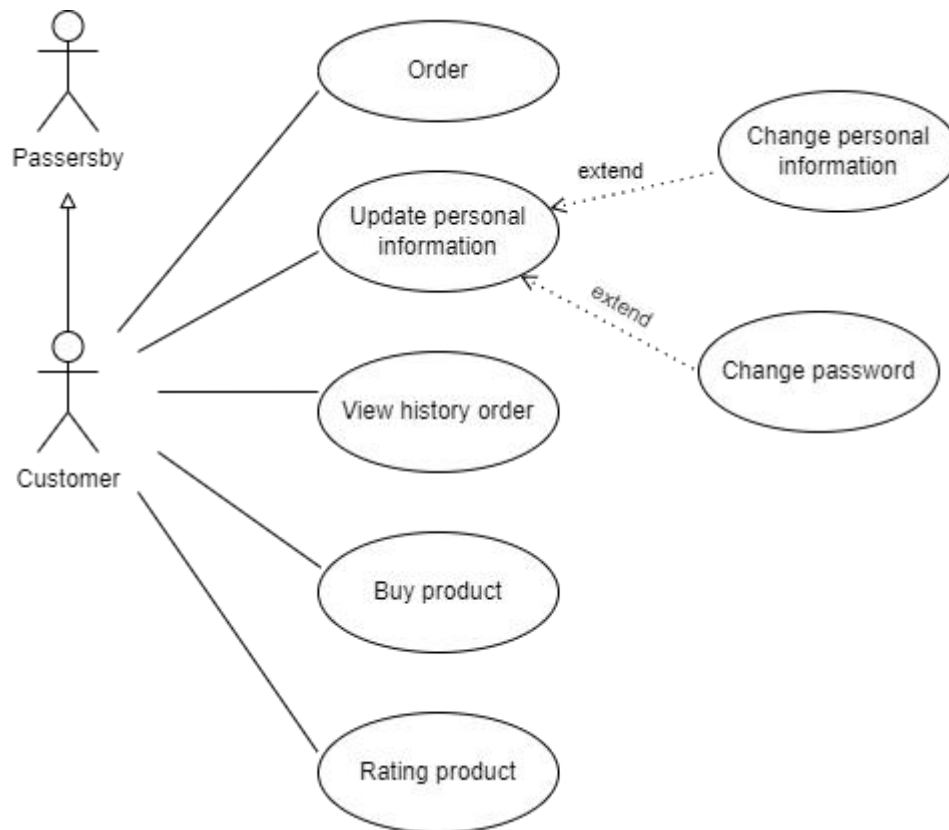
#### 2.1.1. Use case diagram của khách vãng lai



Hình 11. Use case diagram của khách vãng lai

Sơ đồ use case của khách vãng lai, trong đó khách vãng lai chỉ có thể xem chi tiết sản phẩm và tìm kiếm sản phẩm, ngoài ra không được thực hiện bất cứ việc gì trong website.

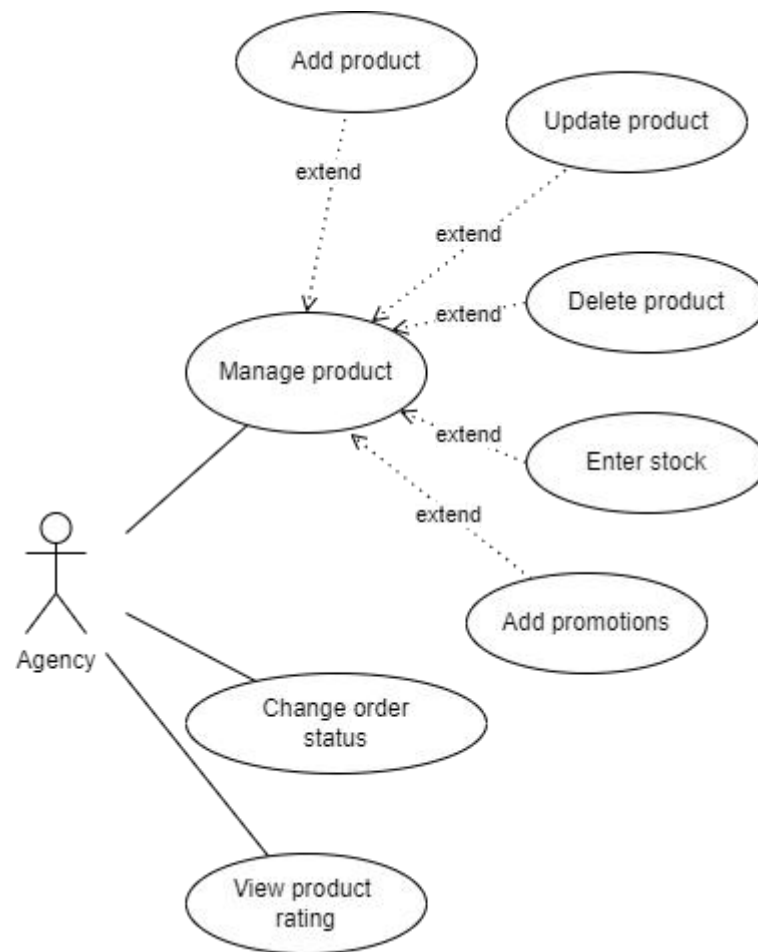
### 2.1.2. Use case diagram của khách hàng



Hình 12. Use diagram của khách hàng

Khách hàng (là khách vãng lai sau khi đăng ký tài khoản và đăng nhập vào hệ thống) có toàn bộ những chức năng của khách vãng lai, thêm vào đó là những chức năng như xem thông tin cá nhân, thay đổi thông tin cá nhân, có giỏ hàng, có thể đặt hàng, xem lịch sử đặt hàng cũng như tình trạng đơn hàng, có thể xem đánh giá sản phẩm cũng như đánh giá những sản phẩm đã mua trước đó.

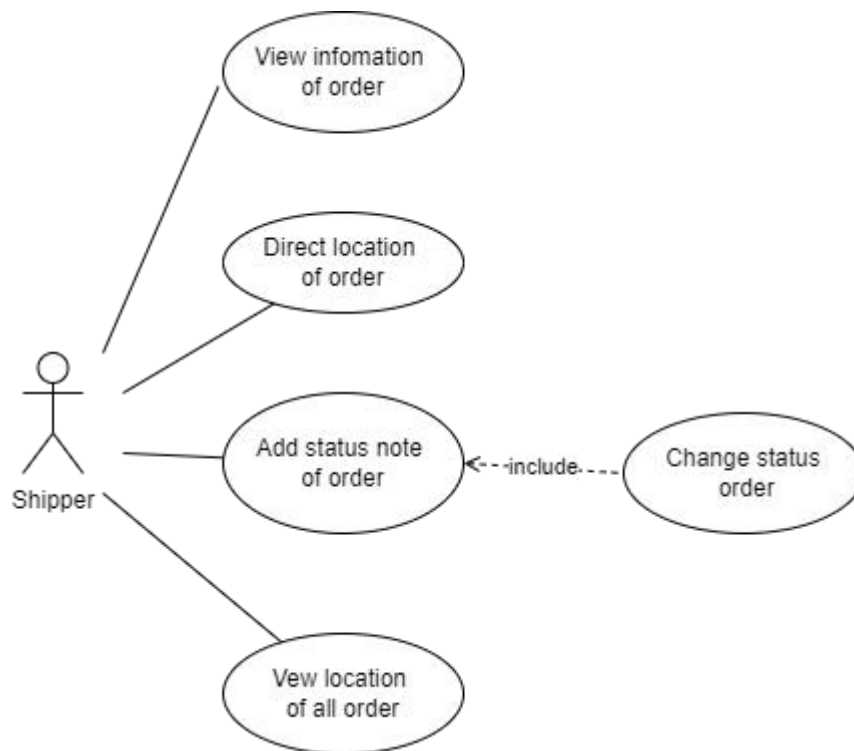
### 2.1.3. Use case của đại lý



Hình 13. Use case của đại lý

Đại lý có quyền thêm, sửa và xóa sản phẩm. Có thể tạo phiếu nhập để thêm số lượng sản phẩm vào trong kho và thêm chương trình khuyến mãi có thời gian cho một sản phẩm nào đó. Có thể thay đổi trạng thái đơn hàng nhưng không được thay đổi trạng thái cuối cùng của đơn hàng (giao hàng thành công và đơn hàng bị hủy). Có thể xem được đánh giá của sản phẩm, cũng như là thống kê bình luận của sản phẩm.

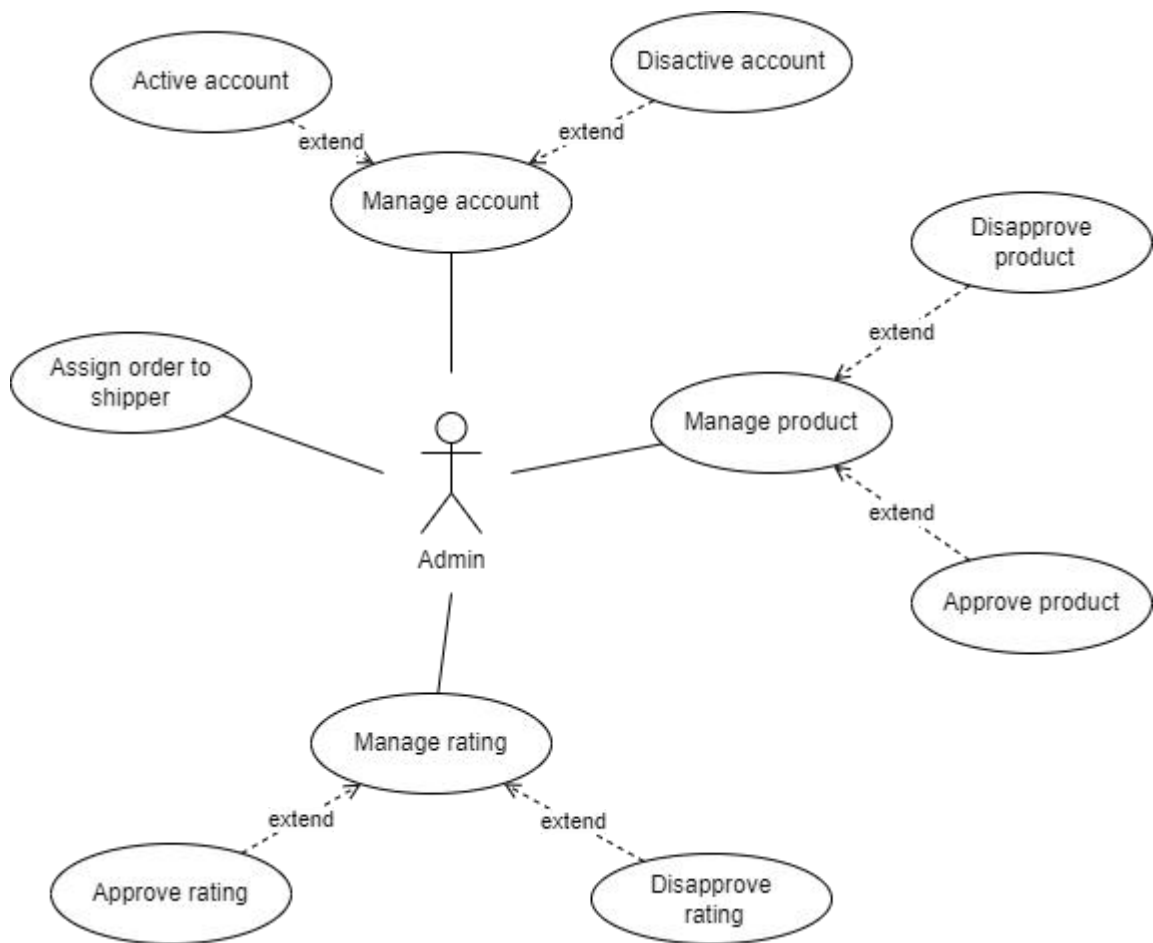
#### 2.1.4. Use case của người giao hàng



Hình 14. Use case của người giao hàng

Người giao hàng có các chức năng như xem thông tin của đơn hàng, có thể sử dụng bản đồ chỉ dẫn từ vị trí hiện tại đến vị trí của đơn hàng, thêm ghi chú trạng thái cho đơn hàng và thay đổi trạng thái cuối của đơn hàng (hủy đơn hàng, giao hàng thành công).

### 2.1.5. Use case của người quản trị

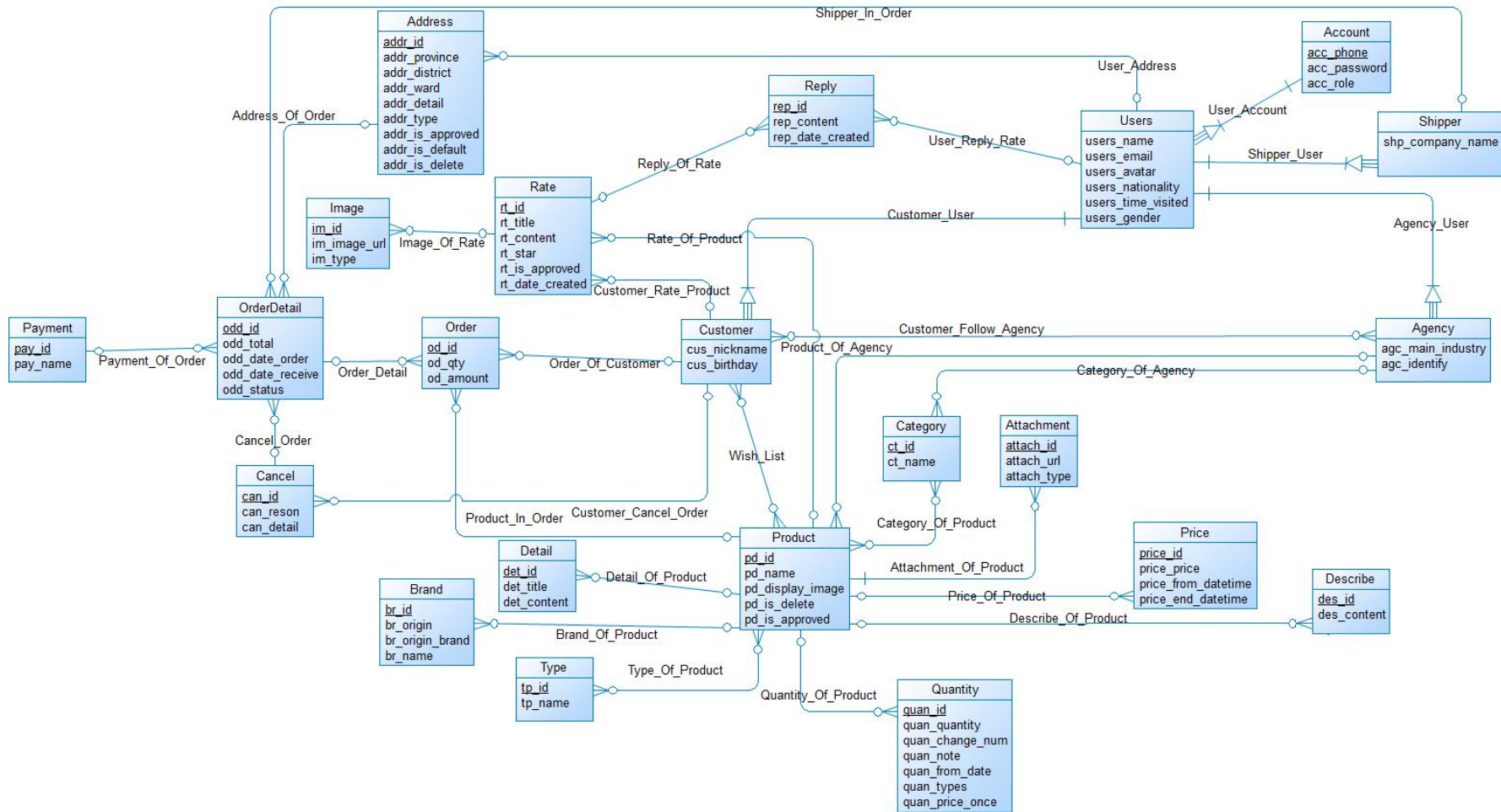


Hình 15. Use case của người quản trị

Người quản trị có quyền quyết định xem đại lý có được phép hoạt động hay không bằng cách kích hoạt tài khoản đại lý, sản phẩm muốn được xuất hiện trên trang chủ phải được cho phép bởi người quản trị, đối với bình luận cũng vậy. Sau khi kích hoạt tài khoản, sản phẩm (không bao gồm bình luận) đại lý sẽ nhận được tin nhắn tự động từ người quản trị nhằm thông báo về sự thay đổi. Người quản trị có thể chỉ định shipper nào gửi đơn đặt hàng cho khách hàng.

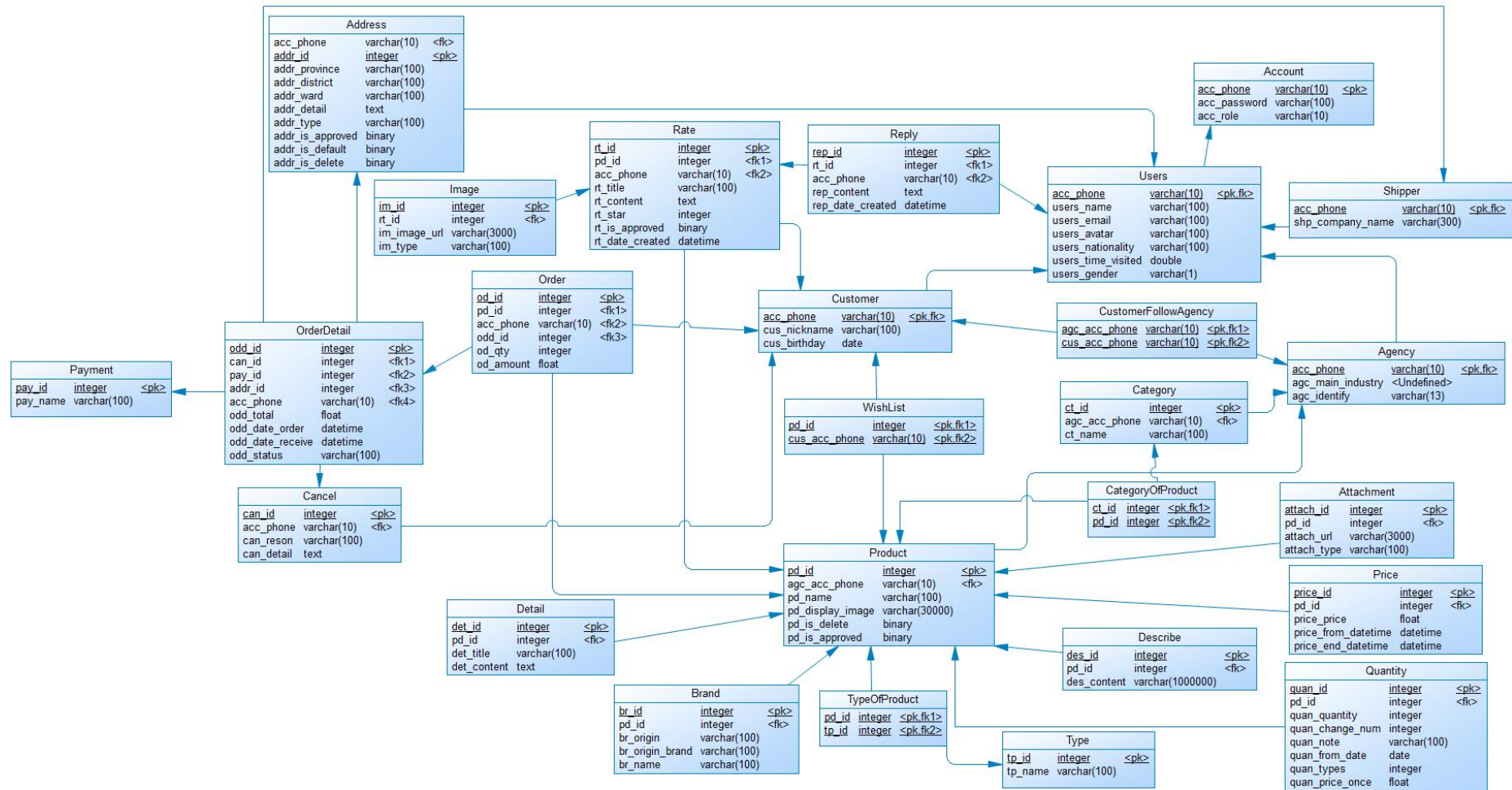


## 2.2. Mô hình dữ liệu mức quan niệm (CDM)



Hình 16. Mô hình dữ liệu mức quan niệm

### 2.3. Mô hình dữ liệu mức vật lý (PDM)



Hình 17. Mô hình dữ liệu mức vật lý

## 2.4. Các yêu cầu ràng buộc cần thiết khi xử lý dữ liệu

### - Giỏ hàng (Order):

+ Thêm hoặc cập nhật số lượng: Số lượng sản phẩm phải lớn hơn hoặc bằng số lượng sản phẩm còn trong kho.

### - Đánh giá sản phẩm (Rate):

+ Thêm đánh giá sản phẩm: Khách hàng phải mua sản phẩm trước đó và sản phẩm phải ở trong đơn hàng có trạng thái “giao hàng thành công” thì mới được phép thêm đánh giá. Mỗi khách hàng chỉ được đánh giá một sản phẩm một lần, không được đánh giá nhiều lần trên cùng một sản phẩm.

+ Duyệt đánh giá: Đánh giá của khách hàng sẽ được đưa đến cho người quản trị duyệt, sau khi được chấp thuận thì sẽ được hiển thị trên hệ thống.

### - Đơn hàng (OrderDetail):

+ Đổi trạng thái đơn hàng: Đại lý có thể đổi trạng thái đơn hàng một cách tuần tự “Đang chờ duyệt” -> “Đã duyệt” -> “Gửi cho vận chuyển”, sau đó người quản trị sẽ giao đơn hàng cho người giao hàng. Người giao hàng sẽ có thể đổi trạng thái đơn hàng là “Đơn hàng giao thành công” hoặc “Đơn hàng giao không thành công” và phải chụp lại bằng chứng rõ ràng khi giao thành công cũng như là ghi lại ghi chú. Sau khi giao hàng thành công sẽ gửi email thông báo đến khách hàng.

+ Địa chỉ đơn hàng: Đơn hàng chỉ có một địa chỉ.

+ Đại lý trong đơn hàng: Nếu giỏ hàng có nhiều đại lý thì khi đặt hàng phải tách các sản phẩm của từng đại lý thành một đơn hàng riêng biệt, mỗi đơn hàng là một tập hợp những sản phẩm có chung một đại lý.

### 3. CÁC CHỨC NĂNG CHÍNH TRONG HỆ THỐNG

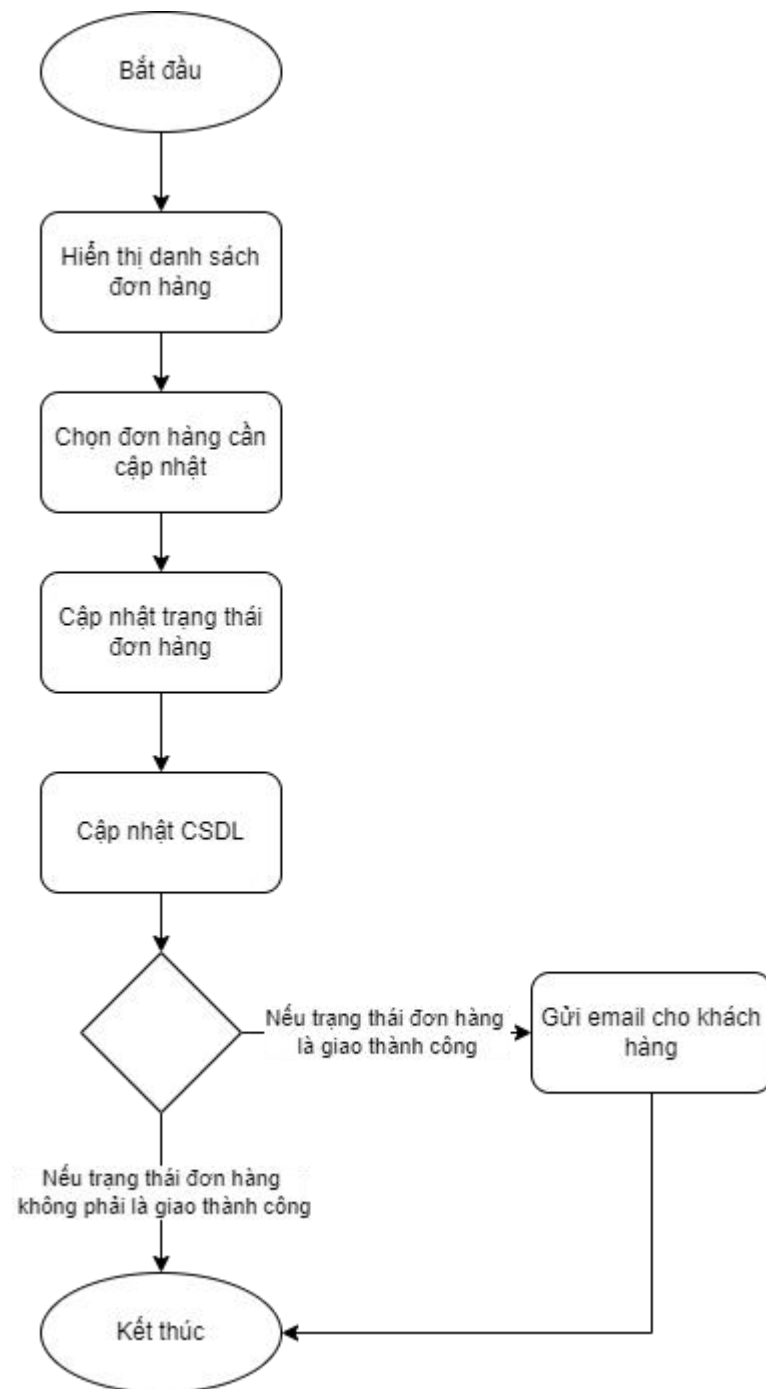
#### 3.1. Chức năng mua và thanh toán giỏ hàng



Hình 18. Lưu đồ mua và thanh toán giỏ hàng

Khi khách hàng đã chọn sản phẩm và lưu vào giỏ hàng, khách hàng có thể chỉnh sửa, cập nhật lại giỏ hàng về số lượng cũng như là xóa các mặt hàng mà khách hàng không muốn mua hiện có trong giỏ hàng. Sau đó khách hàng có thể tiến hành đặt hàng bằng cách nhập thông tin về phương thức thanh toán cũng như là địa chỉ nhận hàng. Cuối cùng, sau khi nhập hết thông tin cần thiết, khách hàng có thể tiến hành đặt hàng.

### 3.2. Chức năng cập nhật trạng thái đơn hàng

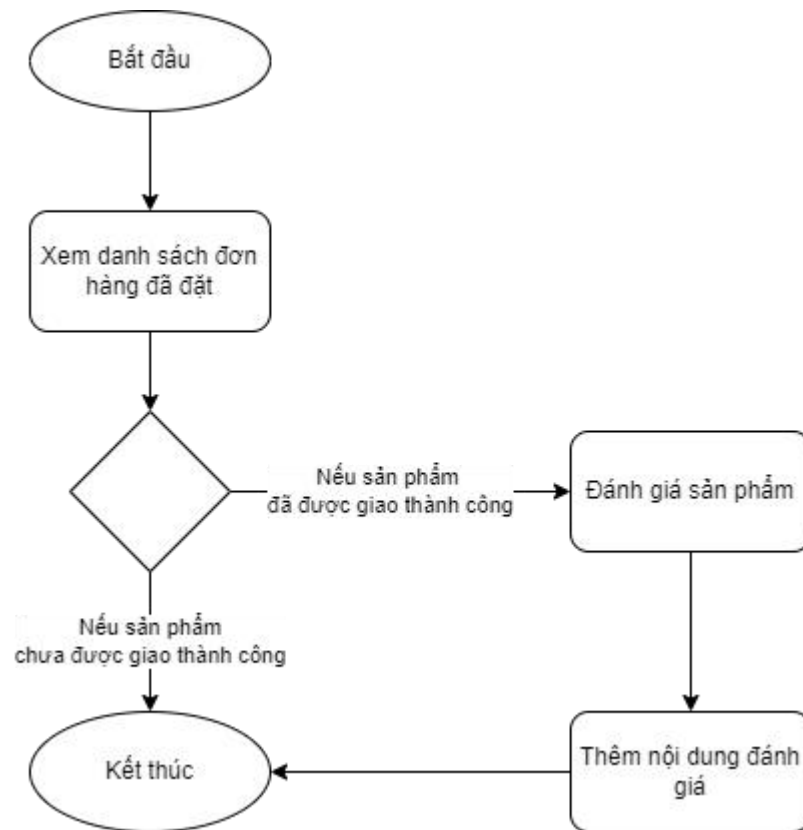


Hình 19. Lưu đồ chức năng cập nhật trạng thái đơn hàng

Sau khi người dùng đặt hàng thành công, danh sách đơn hàng sẽ được cập nhật. Đại lý sau khi tiếp nhận đơn hàng sẽ duyệt và gửi qua bên vận chuyển, sau khi được gửi qua bên vận chuyển, người quản trị có thể chỉ định người giao hàng nào được giao đơn hàng đến khách hàng. Trạng thái đơn hàng được cập nhật là tuần tự, không được

thay đổi thứ tự của trạng thái, các trạng thái bao gồm “Chờ duyệt” -> “Đã duyệt” -> “Gửi cho vận chuyển” -> “Đang vận chuyển” -> “Giao đơn hàng thành công”. Sau khi giao hàng thành công thì khách hàng sẽ nhận được email tự động từ người quản trị.

### 3.3. Chức năng đánh giá sản phẩm



Hình 20. Lưu đồ chức năng đánh giá sản phẩm

Sản phẩm sau khi được giao thành công thì khách hàng có thể tiến hành đánh giá bằng việc cho sản phẩm số sao từ 1 đến 5 và thêm tiêu đề, nội dung cho phần đánh giá của mình. Khách hàng chỉ được đánh giá cùng một sản phẩm 1 lần duy nhất, không được phép sửa đổi cũng như là xóa đánh giá. Sau khi đánh giá xong, khách hàng sẽ chờ cho đến khi người quản trị chấp nhận đánh giá cho phép hiển thị ở hệ thống.

## 4. CÁC API CỦA HỆ THỐNG

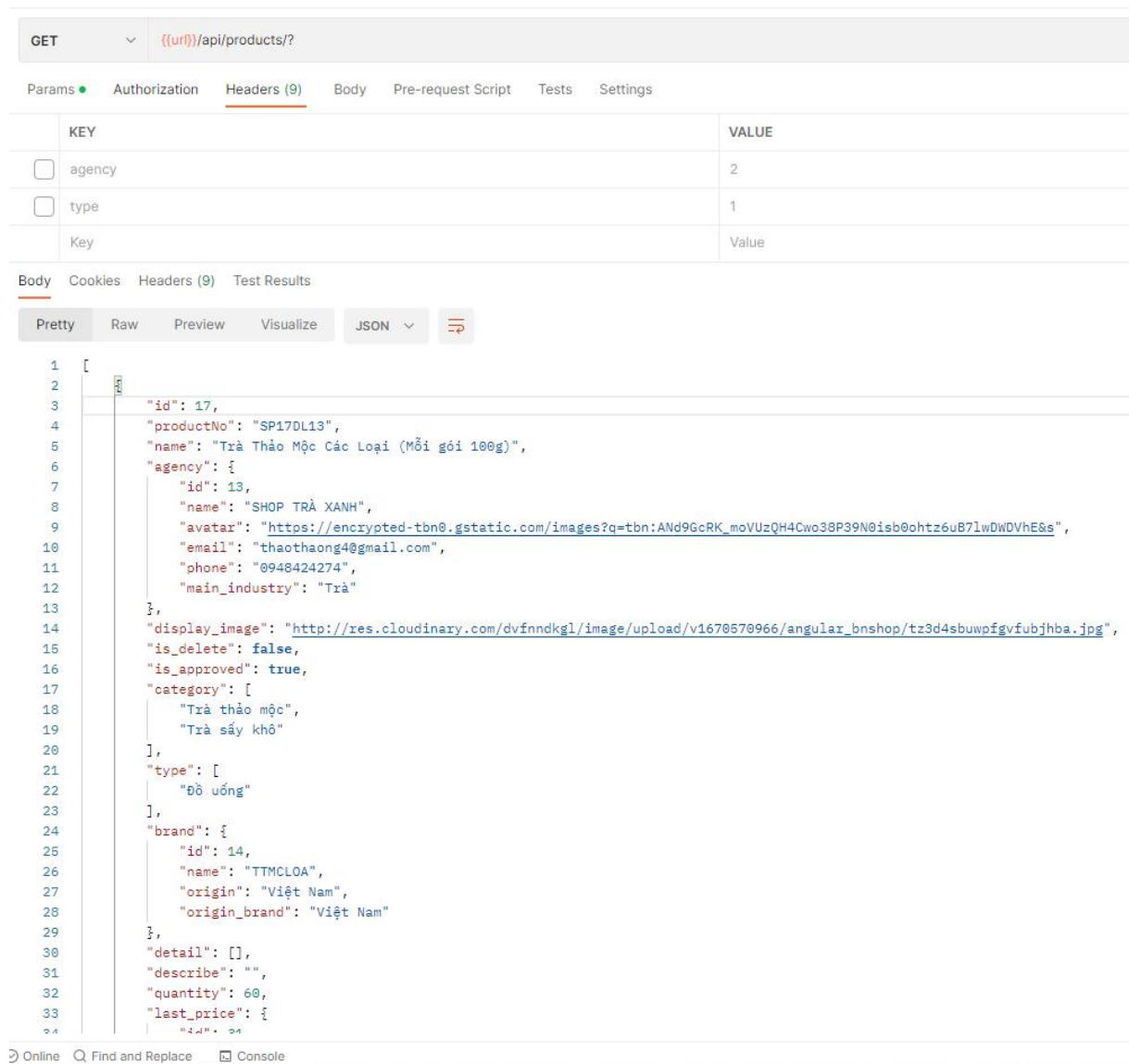
Ở phần kết quả của api, em sẽ sử dụng chương trình Postman để thể hiện kết quả.

## 4.1. Về sản phẩm

Tất cả người dùng đều có quyền gửi phương thức **GET** cho các api thuộc về sản phẩm, còn những phương thức còn lại thì chỉ có **đại lý** và **người quản trị** mới được quyền sử dụng.

### 4.1.1. Xem sản phẩm

Sử dụng phương thức **GET** để gọi api có url là **/api/products/**



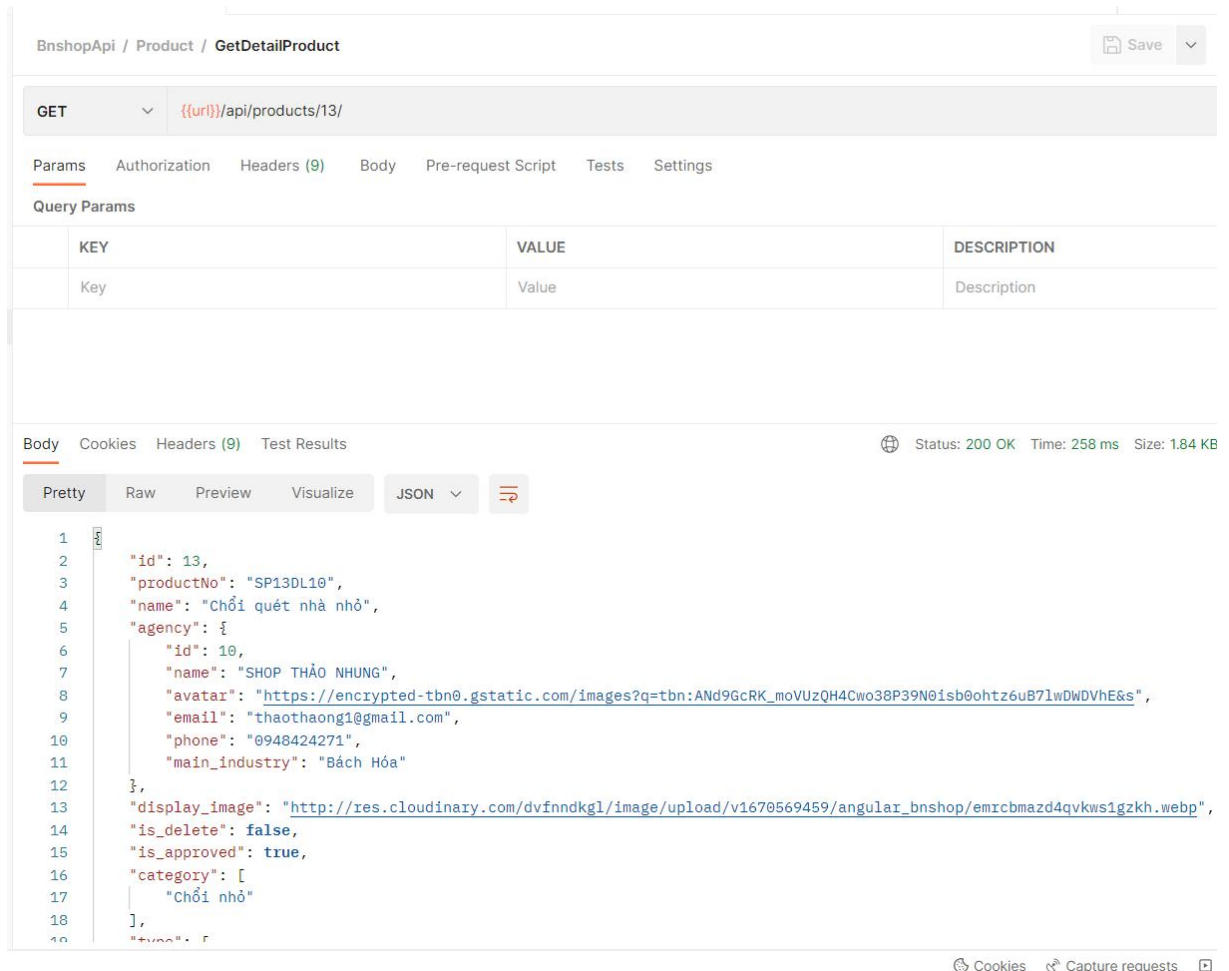
Hình 21. Kết quả lấy tất cả sản phẩm

Kết quả trả về sẽ là một mảng chứa thông tin của các sản phẩm trong hệ thống. Trong đó khi truyền param **agency** là **id** của người dùng đại lý thì kết quả sẽ trả về những sản phẩm của **agency** đó, còn **type** là để lấy sản phẩm theo tình trạng như là

chưa được kích hoạt, đã kích hoạt hoặc đã xóa. Nếu không truyền param sẽ truyền về toàn bộ sản phẩm.

#### 4.1.2. Xem chi tiết một sản phẩm

Sử dụng phương thức GET để gọi api có url là `/api/products/<id>/`



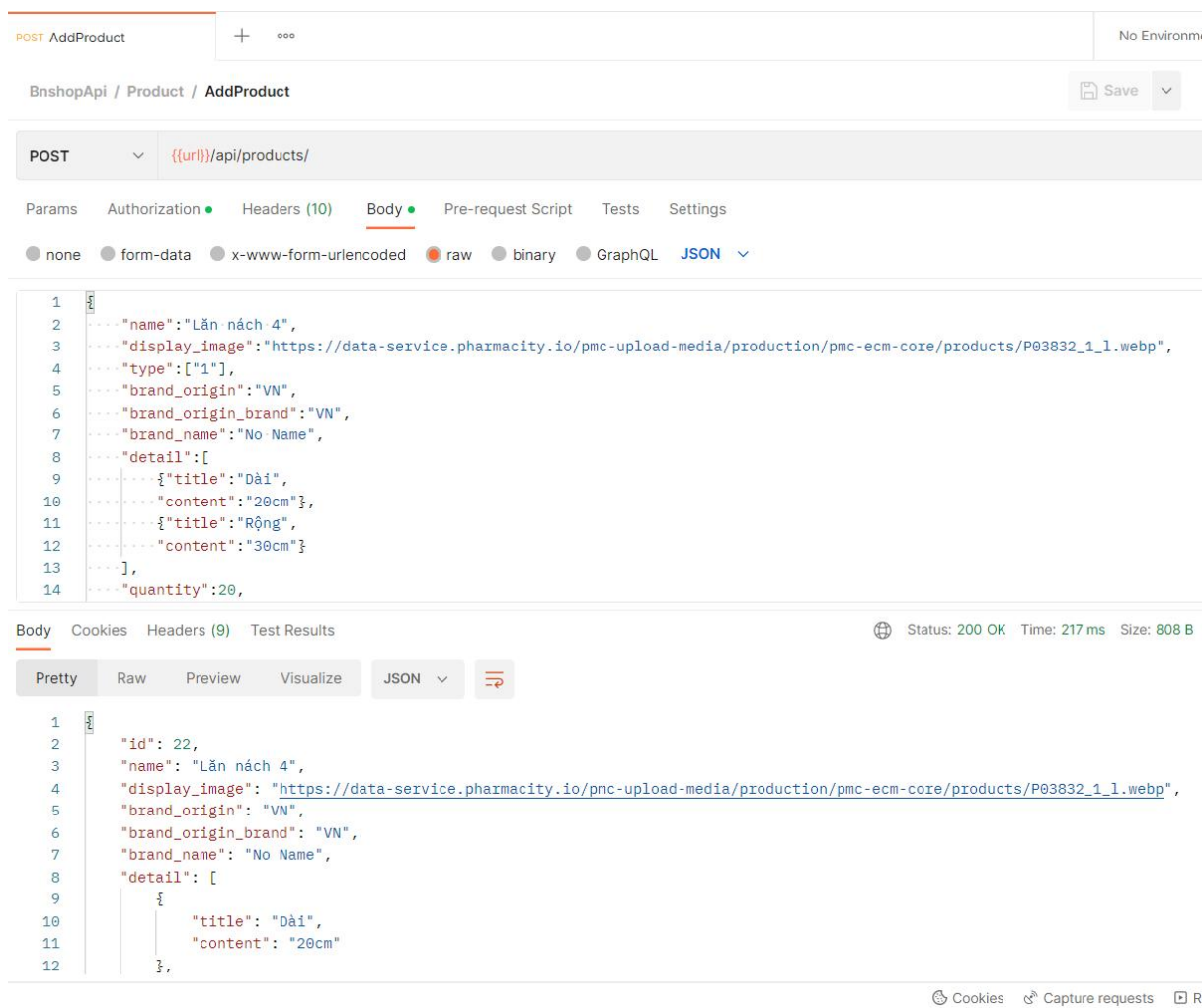
Hình 22. Xem chi tiết một sản phẩm

Kết quả trả về sẽ là một **object** là thông tin chi tiết của sản phẩm có id là `<id>`.

#### 4.1.3. Thêm sản phẩm

Api này chỉ có đại lý mới có quyền sử dụng với phương thức là **POST** và url là `/api/products/`





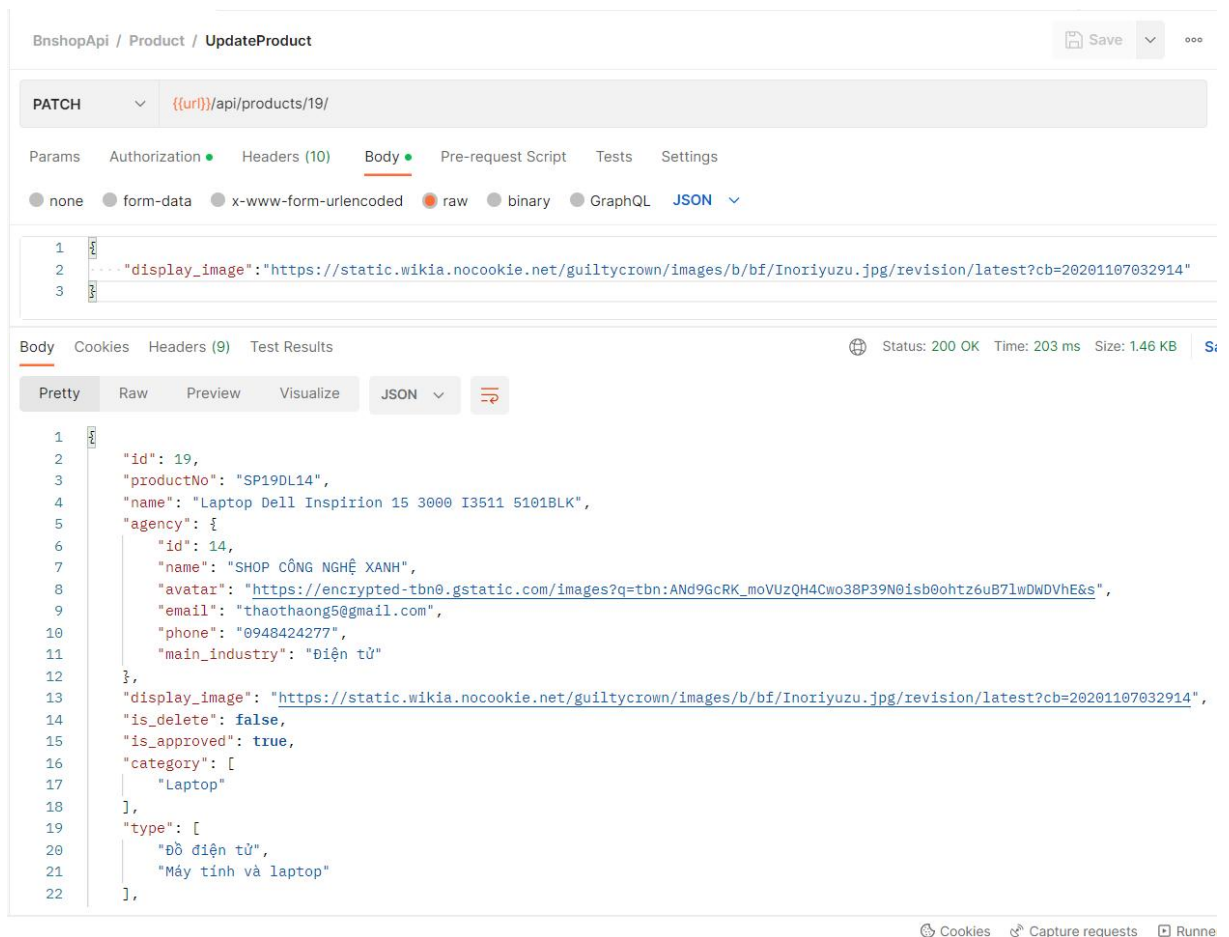
Hình 23. Đại lý thêm sản phẩm

Sau khi đại lý thêm sản phẩm thành công, hệ thống sẽ trả về thông tin sản phẩm đúng với những thông tin sản phẩm mà đại lý đã nhập trước đó kèm với id của sản phẩm.

#### 4.1.4. Sửa sản phẩm

Đây vẫn là một api dành cho đại lý, với phương thức là **PATCH** và url là `/api/products/`. Tuy nhiên, đại lý chỉ có thể sửa đổi những thông tin như sau:

- Giá của sản phẩm.
- Danh mục của sản phẩm.
- Hình giới thiệu của sản phẩm.
- Mô tả của sản phẩm.

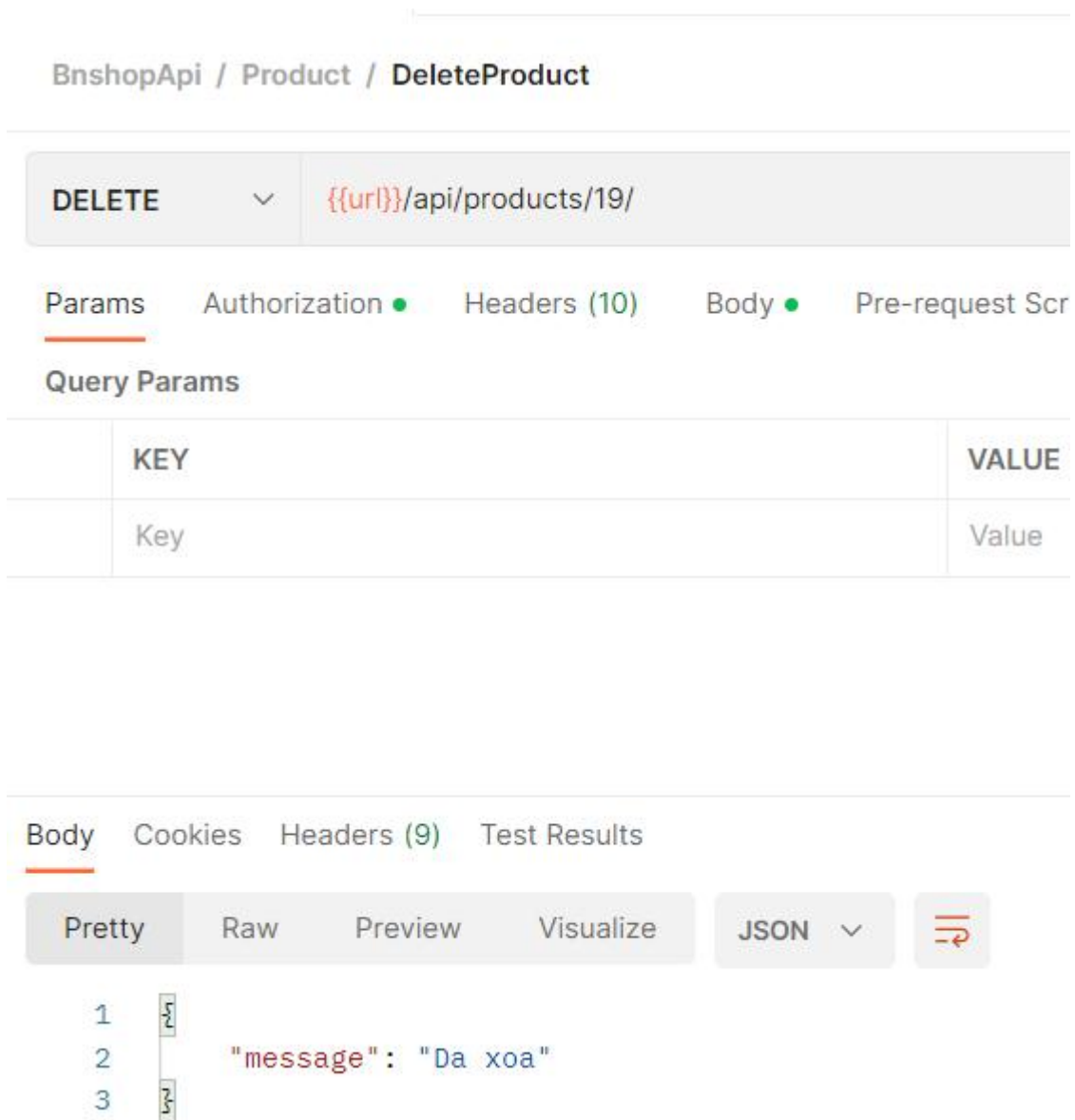


Hình 24. Sửa sản phẩm

Sau khi thay đổi thành công, hệ thống sẽ trả về thông tin của sản phẩm sau khi sửa đổi bởi đại lý.

#### 4.1.5. Xóa sản phẩm

Đại lý có thể xóa sản phẩm của chính mình khi gửi yêu cầu với phương thức **DELETE** với url là **/api/products/<id>/** với **<id>** là id của sản phẩm.

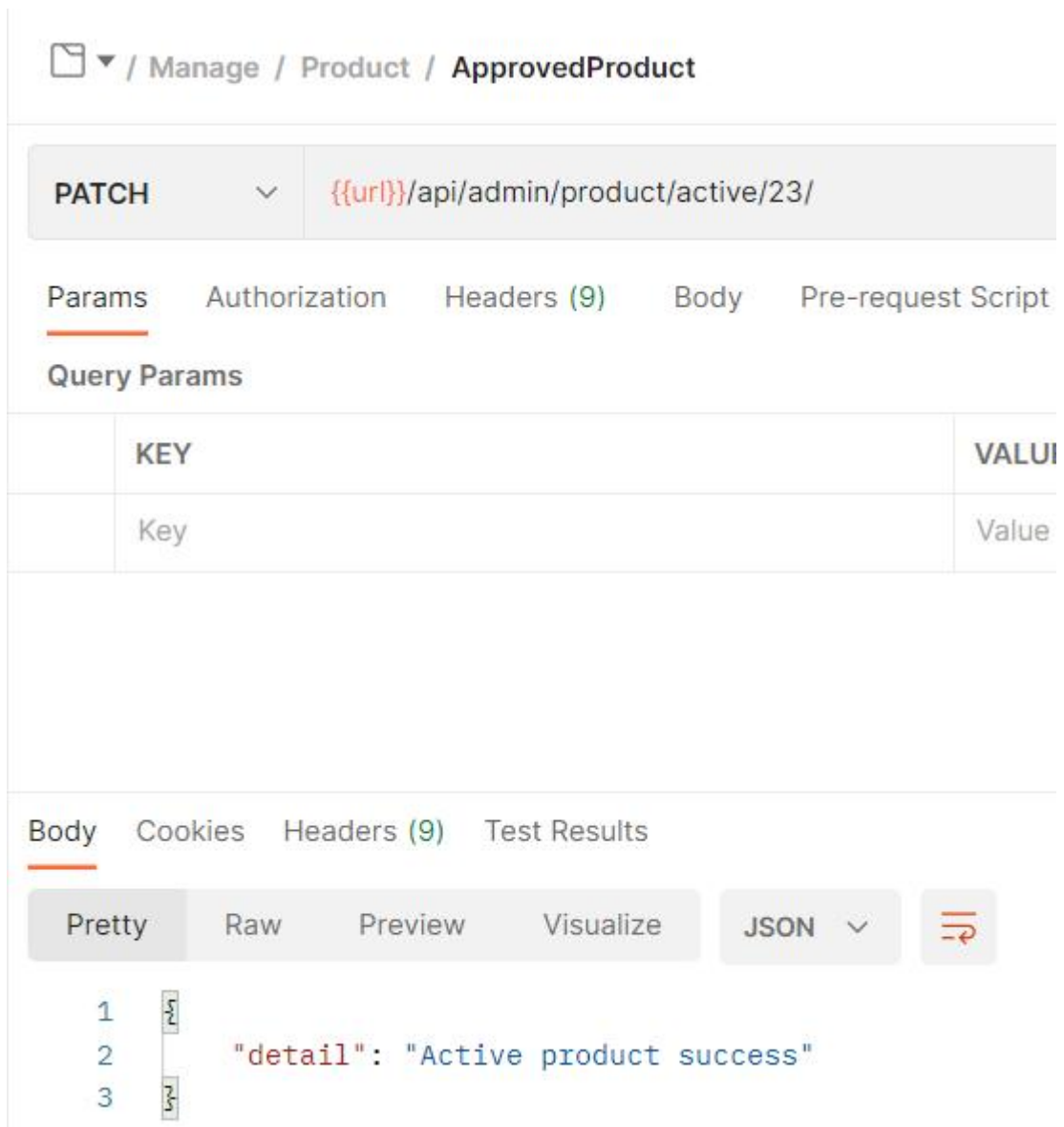


Hình 25. Xóa sản phẩm

Nếu xóa sản phẩm thành công, hệ thống sẽ trả về thông báo là đã xóa cho đại lý.

#### 4.1.6. Active sản phẩm

Api này dành cho **người quản trị** khi kiểm duyệt sản phẩm, sau khi kiểm duyệt thành công thì người quản trị có quyền cấp phép cho sản phẩm để xuất hiện trên sàn thương mại. Api này được gọi với phương thức là **PATCH** và url là **/api/admin/product/active/<id>** với **<id>** là id của sản phẩm



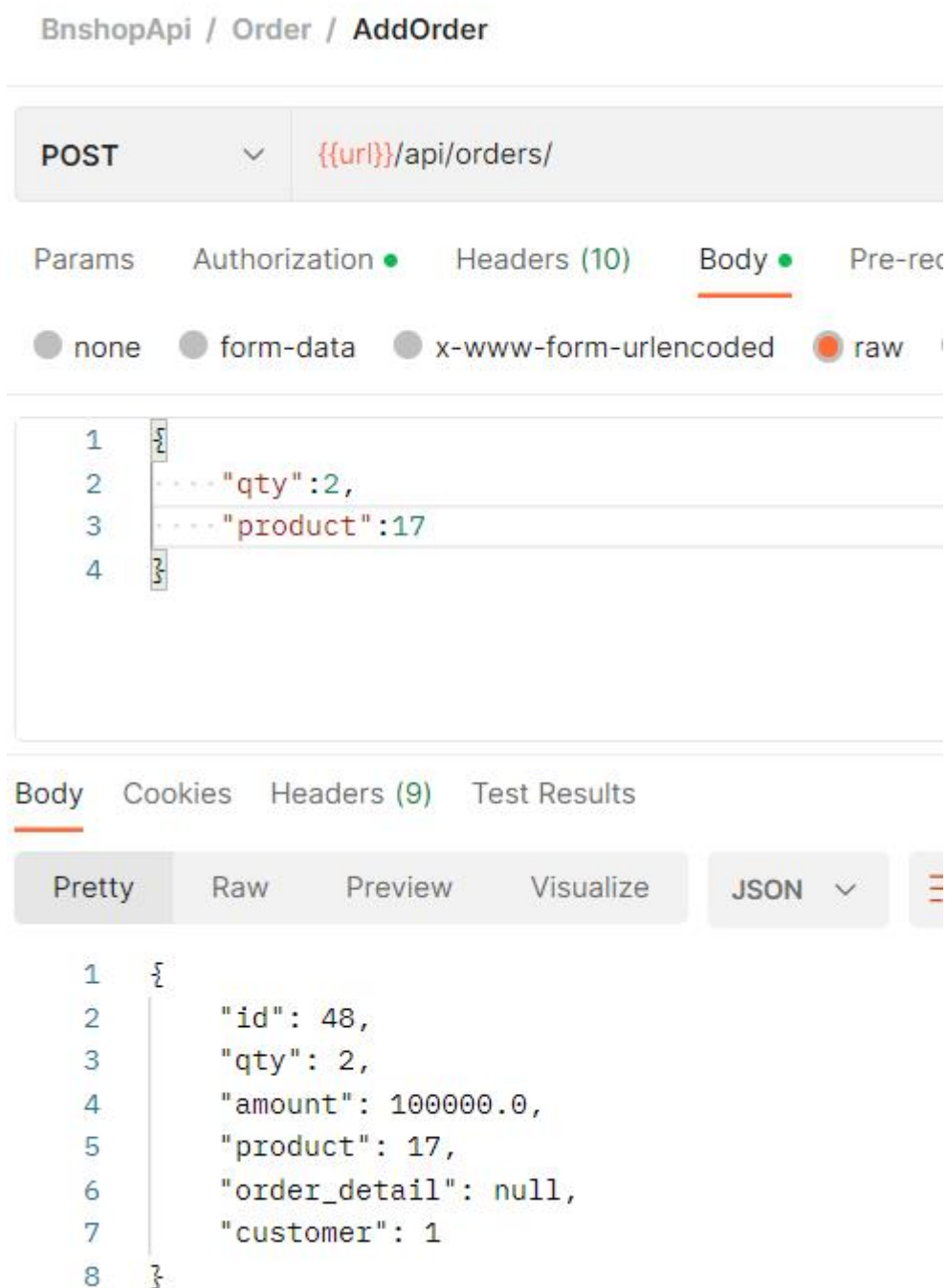
Hình 26. Active sản phẩm

Nếu sản phẩm được cấp phép thành công, hệ thống sẽ trả về thông báo cấp phép thành công cho người quản trị.

## 4.2. Giỏ hàng và đặt hàng

### 4.2.1. Thêm sản phẩm vào giỏ hàng

Khách hàng có thể gọi api với phương thức là **POST** với url là **/api/orders/** để thêm sản phẩm vào giỏ hàng.



*Hình 27. Thêm sản phẩm vào giỏ hàng*

Sau khi thêm vào giỏ hàng thành công, khách hàng có thể xem lại thông tin của đơn hàng vừa đặt. Cũng như là có thể xem lại danh sách sản phẩm trong giỏ hàng của mình với url là **/api/orders/** cùng với phương thức **GET**.

BnshopApi / Order / GetAllOrder

GET

▼

{{url}}/api/orders/

Params

Authorization ●

Headers (8)

Body

Pre-request Script

Query Params

	KEY	VALUE
	Key	Value

Body

Cookies

Headers (9)

Test Results

Pretty

Raw

Preview

Visualize

JSON ▼

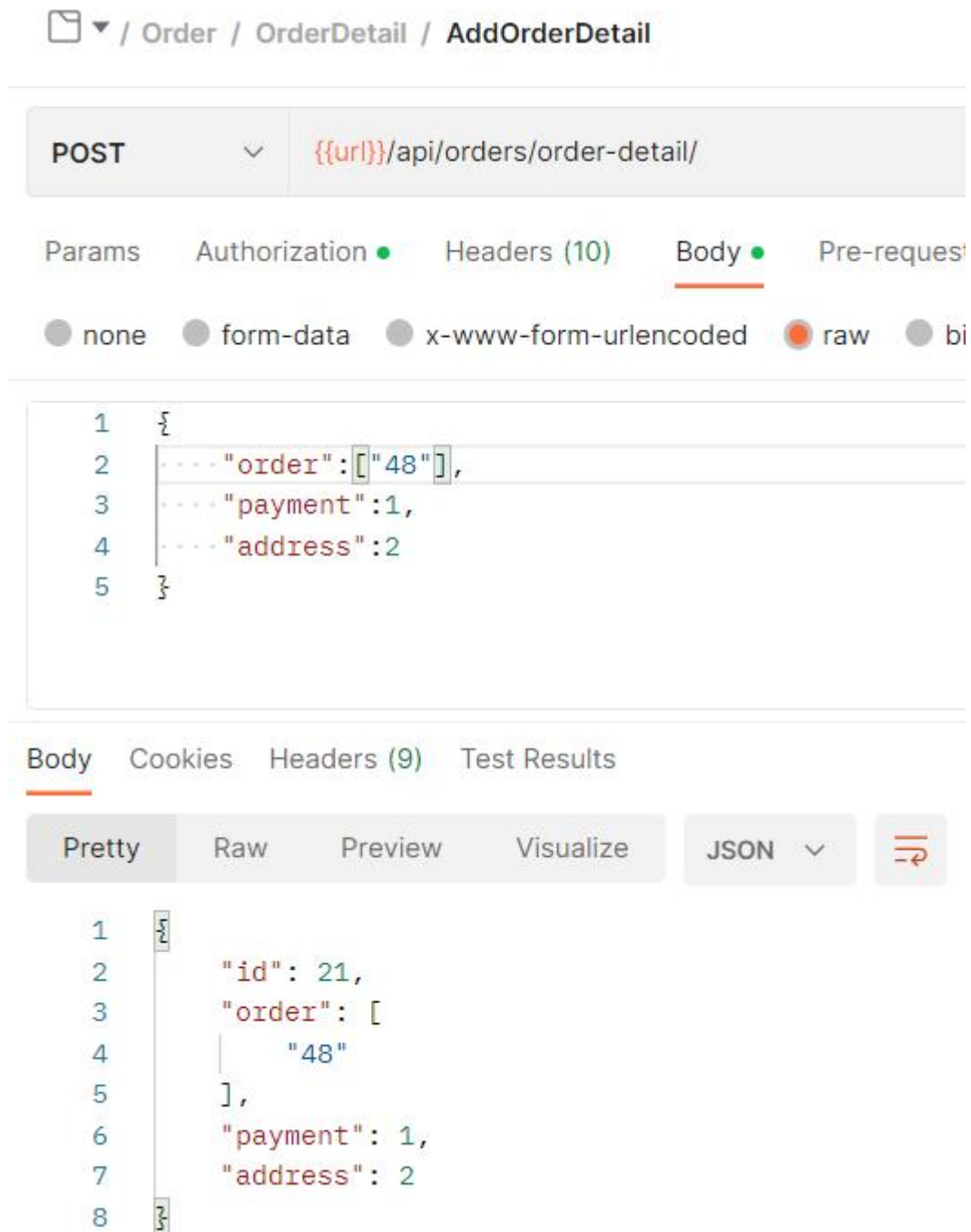
↺

```
1  [
2    {
3      "id": 48,
4      "qty": 2,
5      "amount": 100000.0,
6      "product": 17,
7      "order_detail": null,
8      "customer": 1
9    }
10 ]
```

Hình 28. Xem thông tin giỏ hàng

#### 4.2.2. Đặt hàng

Sau khi thêm sản phẩm vào giỏ hàng, người dùng có thể chọn sản phẩm mình muốn đặt hàng và tiến hành đặt hàng với url là **/api/orders/order-detail/** với phương thức **POST**.

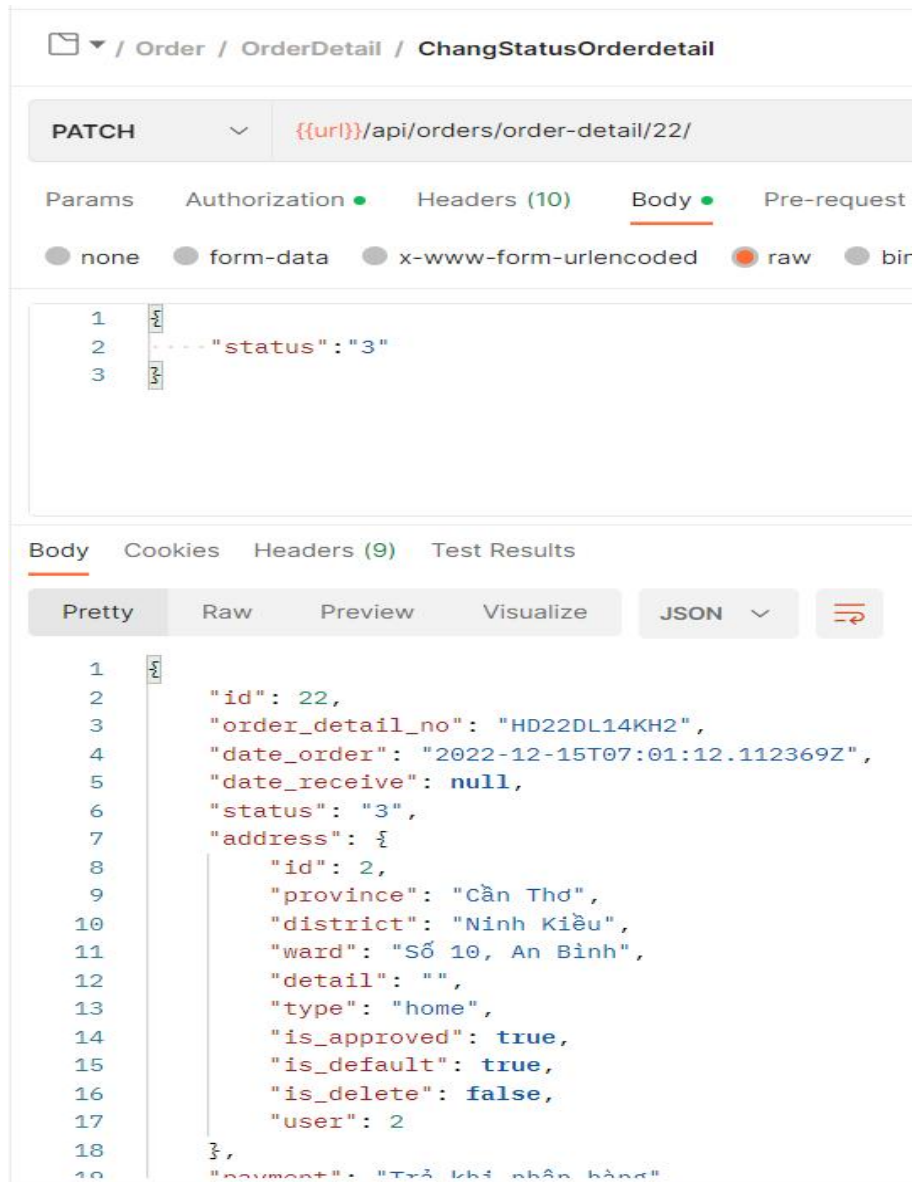


Hình 29. Đặt hàng

Truyền body với **order** là danh sách id của đơn hàng trong giỏ hàng, **payment** là phương thức trả tiền và **address** là địa chỉ giao đến.

#### 4.2.3. Thay đổi trạng thái đơn hàng

Api được gọi khi đại lý duyệt đơn hàng của khách hàng, trạng thái sẽ được thay đổi tuần tự ở front-end. Với api này, đại lý sẽ gọi url là **/api/orders/order-detail/<id>/** với phương thức **POST**.



Hình 30. Thay đổi trạng thái đơn hàng

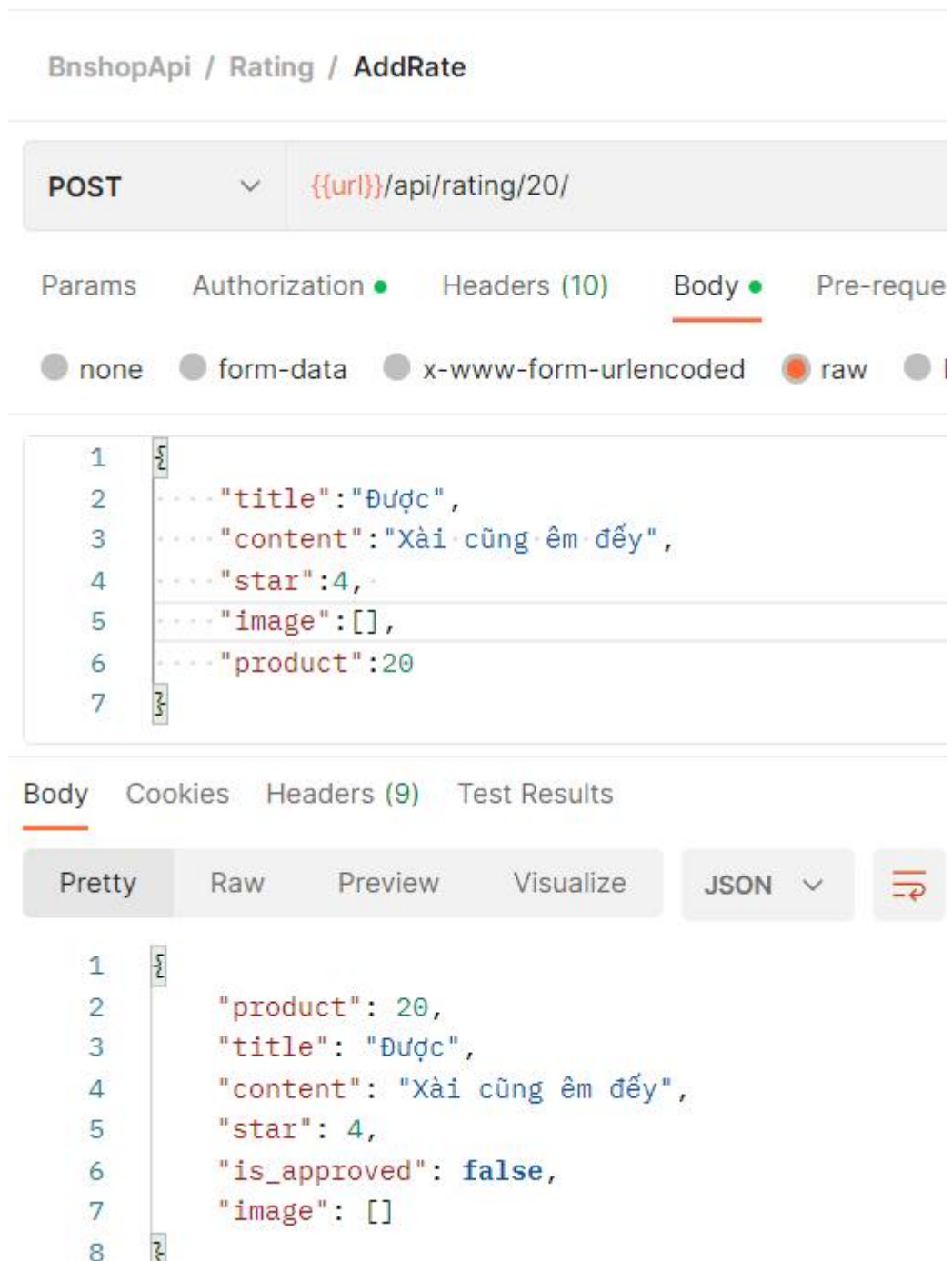
Sau khi thay đổi trạng thái đơn hàng thành công, hệ thống sẽ trả về thông tin chi tiết của đơn hàng đó cho đại lý.



## 4.3. Đánh giá

### 4.3.1. Đánh giá sản phẩm

Khách hàng sau khi nhận hàng thành công có thể tiến hành đánh giá sản phẩm thông qua url `/api/rating/<id>/` với phương thức là **POST**, trong đó `<id>` là id của sản phẩm.

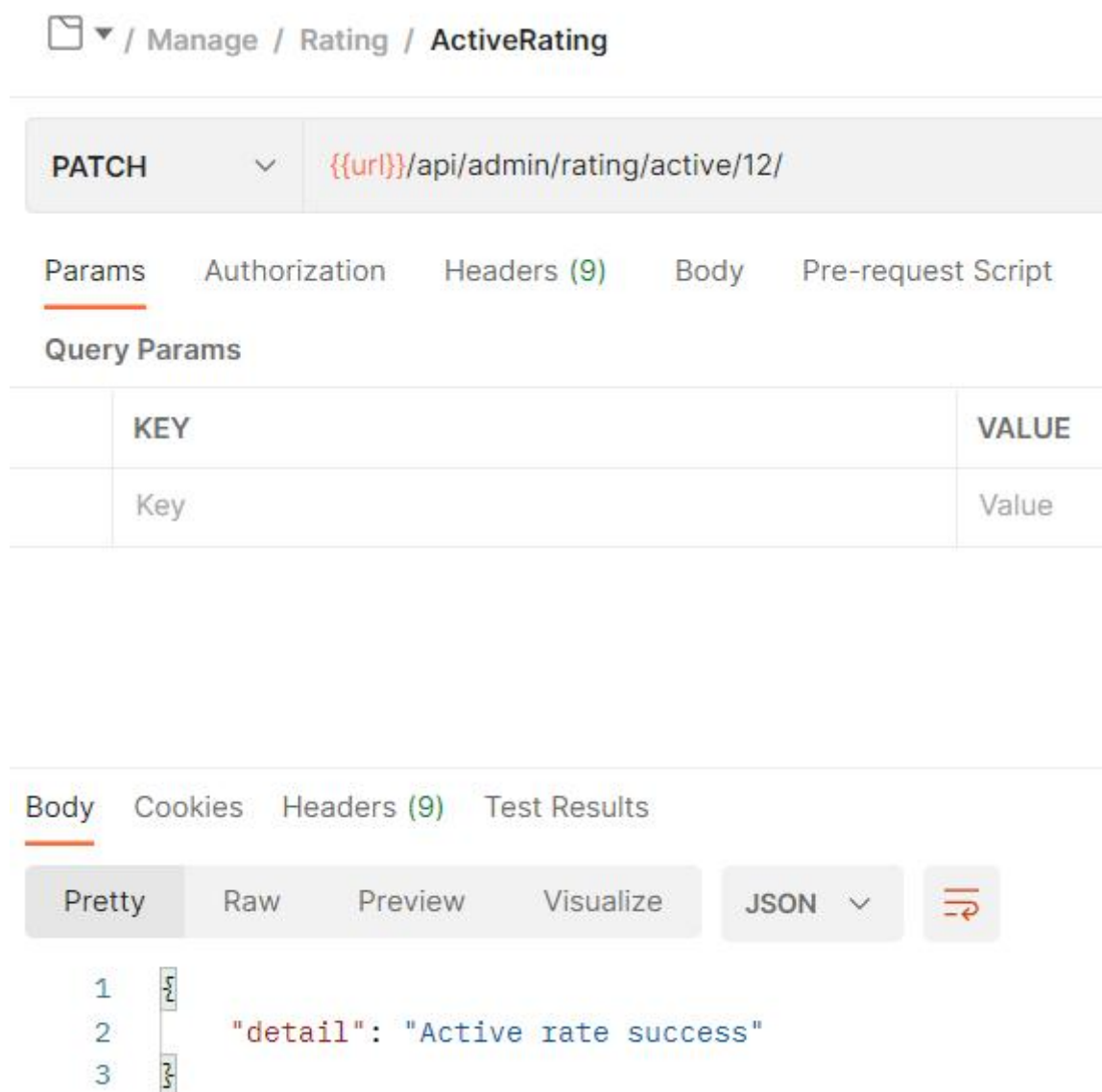


Hình 31. Đánh giá sản phẩm

Khi đánh giá thành công, khách hàng sẽ nhận được thông tin đánh giá mình vừa thực hiện.

#### 4.3.2. Active đánh giá

Sau khi khách hàng đánh giá, thì đánh giá không được xuất hiện liền trên hệ thống mà phải thông qua người quản trị, người quản trị sẽ kiểm duyệt nội dung, nếu không có từ gì vi phạm quy tắc cộng đồng thì sẽ tiến hành active đánh giá qua url **/api/admin/rating/** với phương thức **PATCH**.

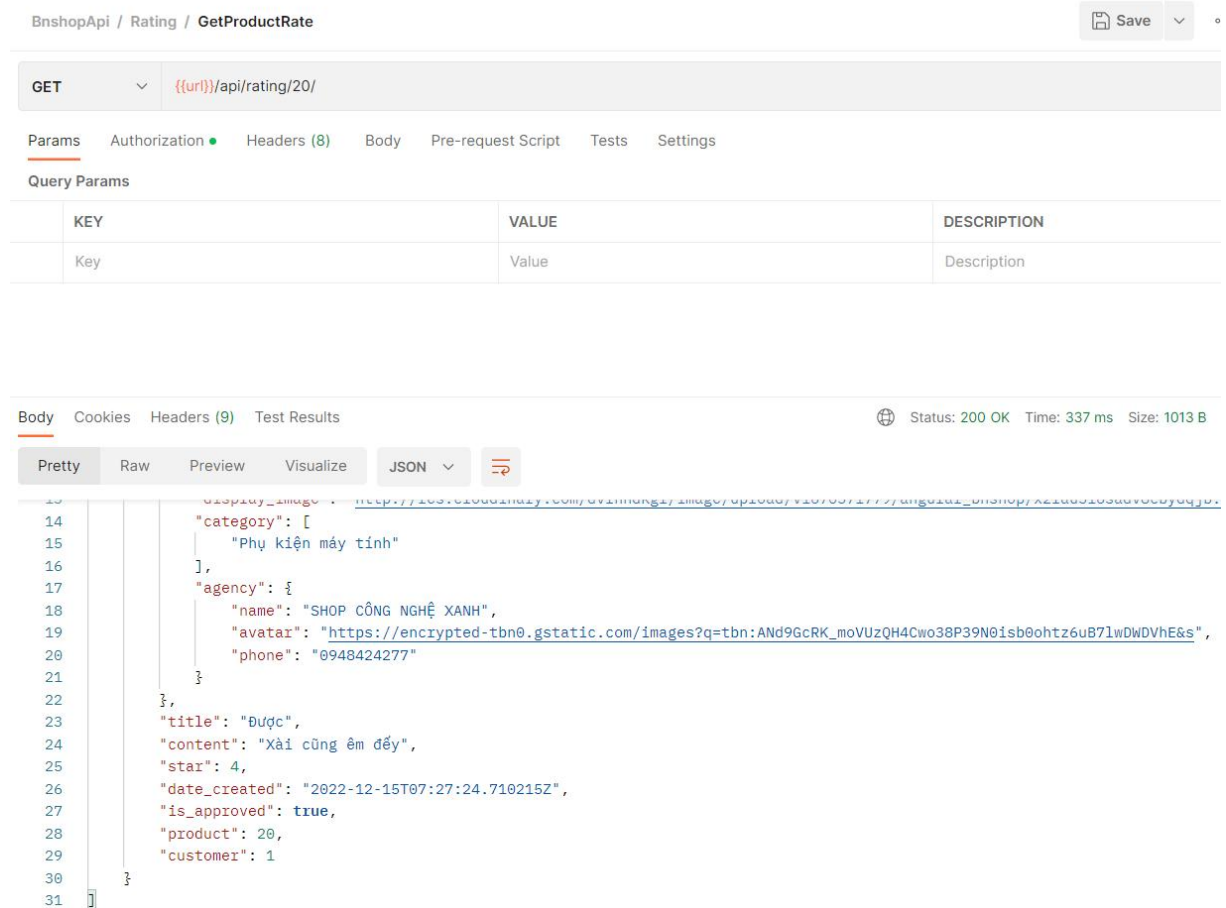


Hình 32. Active đánh giá

Nếu active đánh giá thành công, sẽ trả về thông báo như trong hình.

### 4.3.3. Xem đánh giá của sản phẩm

Khách hàng có thể xem đánh giá của sản phẩm cụ thể nào đó thông qua url **/api/rating/<id>/** với phương thức **GET**. Trong đó <id> chính là id của sản phẩm.

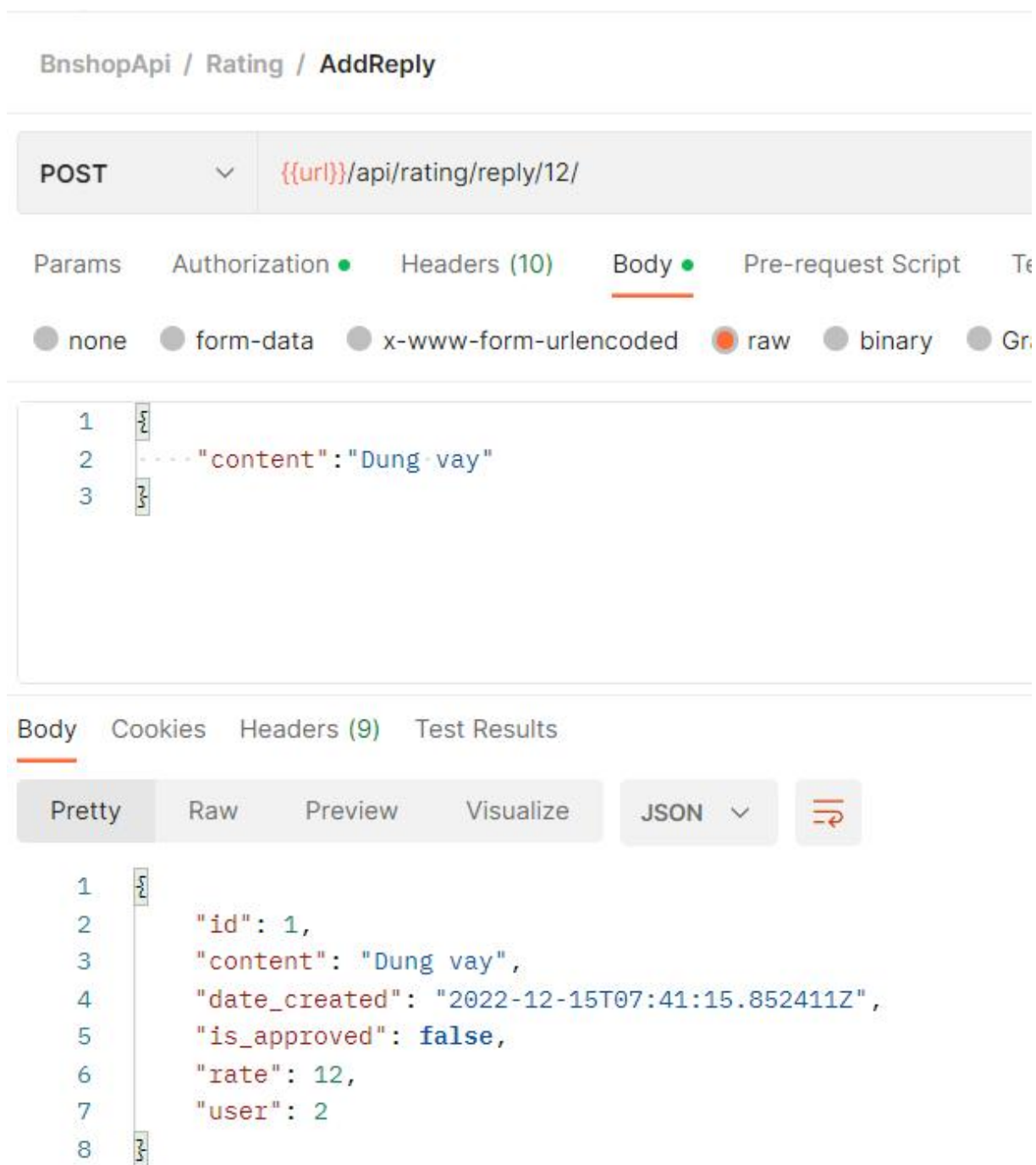


Hình 33. Xem đánh giá của sản phẩm

Kết quả sẽ trả về danh sách các đánh giá cũng như là phản hồi của người dùng đối với sản phẩm.

### 4.3.4. Phản hồi đánh giá

Người dùng khi thấy được đánh giá của sản phẩm thì có thể phản hồi đánh giá đó thông qua url **/api/rating/reply/<id>/** với phương thức **POST**. Trong đó, <id> là id của đánh giá.



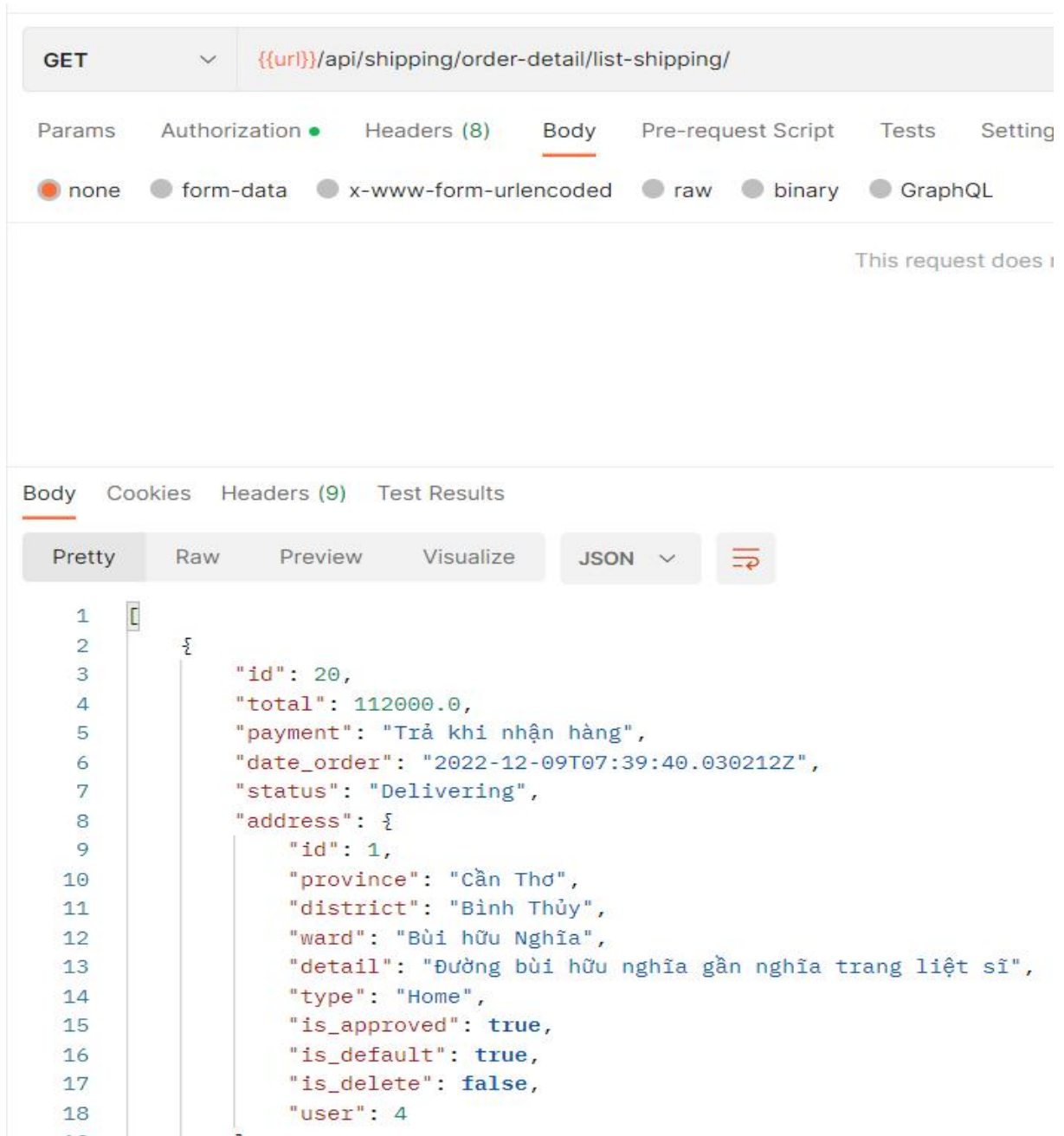
Hình 34. Phản hồi đánh giá

Sau khi phản hồi thành công, hệ thống sẽ trả về kết quả là phản hồi vừa mới tạo của người dùng.

## 4.4. Shipper

### 4.4.1. Xem tất cả đơn hàng được chỉ định

Shipper có thể xem những đơn hàng được chỉ định cho mình thông qua url **/api/shipping/order-detail/list-shipping/** với phương thức **GET**.

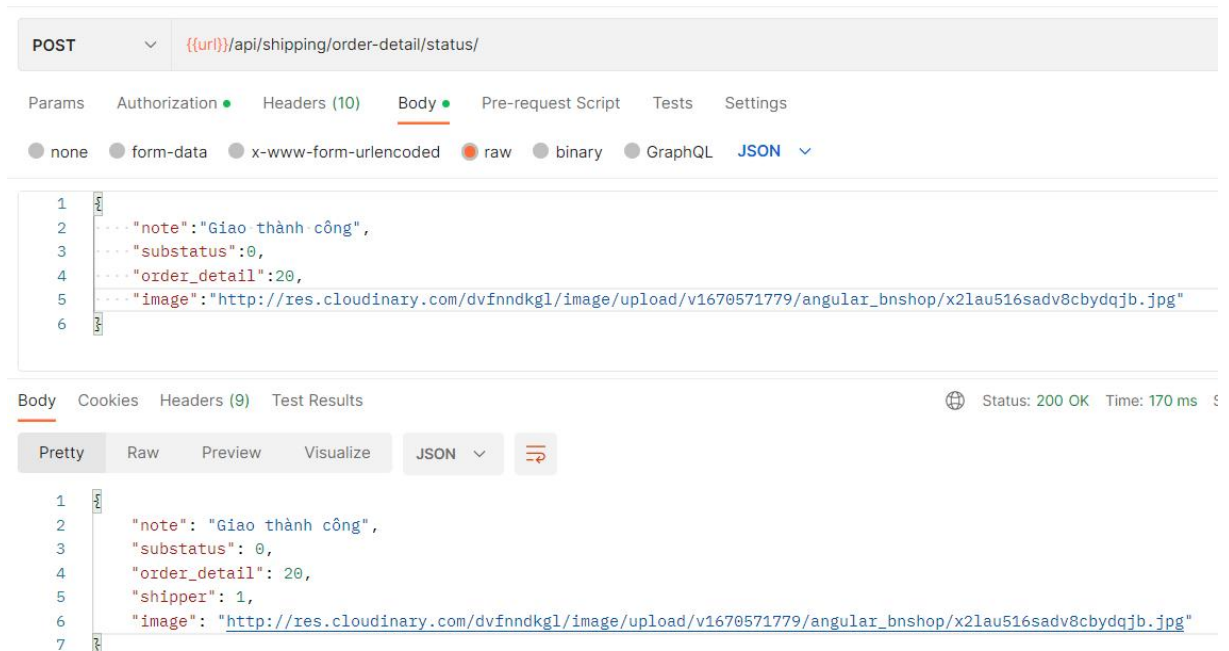


Hình 35. Xem các đơn hàng được chỉ định

Sau khi thực hiện thành công, hệ thống sẽ trả về danh sách các đơn hàng được chỉ định tới shipper.

#### 4.4.2. Thêm ghi chú giao hàng

Khi shipper đưa đơn hàng đến địa chỉ của khách hàng, shipper cần phải ghi chú lại ghi chú khi khách nhận hàng cũng như là lý do không thể giao hàng được cho khách thông qua url **/api/shipping/order-detail/status/** với phương thức **POST**.



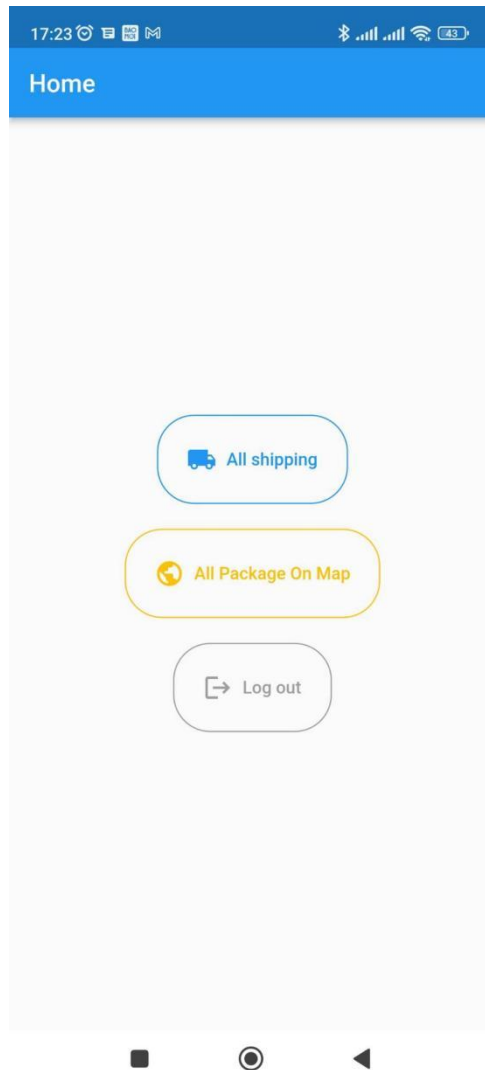
Hình 36. Thêm ghi chú giao hàng

Nếu giao hàng thành công (với `substatus = 0`) thì đơn hàng sẽ tự động chuyển trạng thái về đã nhận.

## 5. GIAO DIỆN CỦA APP NGƯỜI GIAO HÀNG

### 5.1. Trang chủ

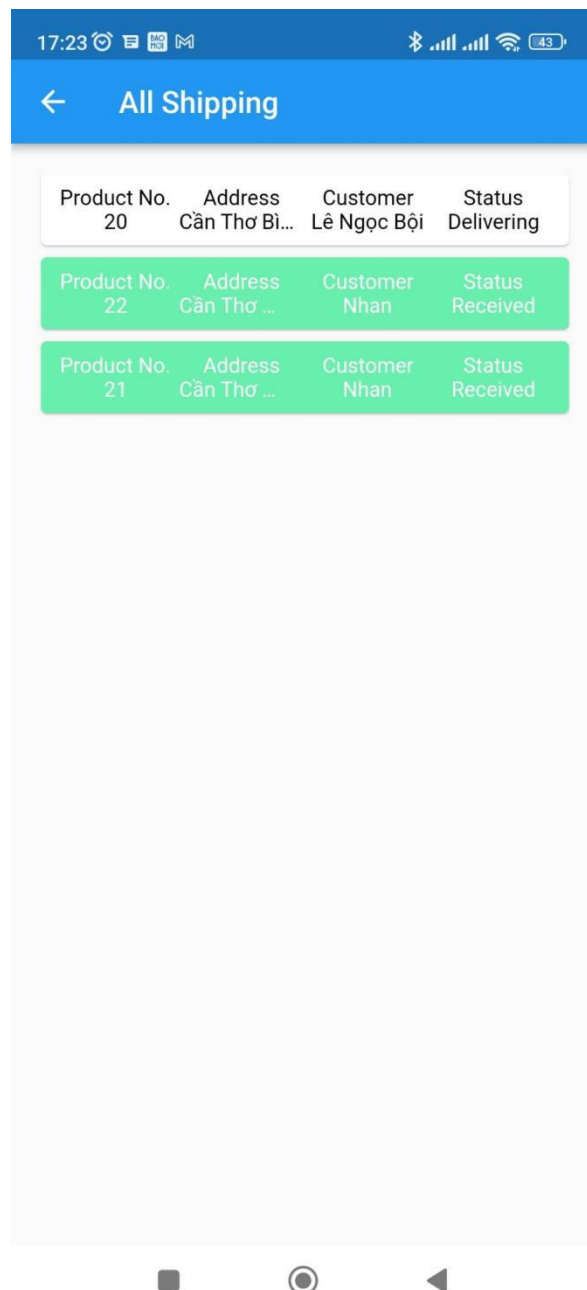
Sẽ hiện thị 2 chức năng chính bao gồm: xem danh sách tất cả đơn hàng được chỉ định, xem tất cả các địa điểm của các đơn hàng.



*Hình 37. Trang chủ của app người giao hàng*

## **5.2. Quản lý đơn hàng được chỉ định**

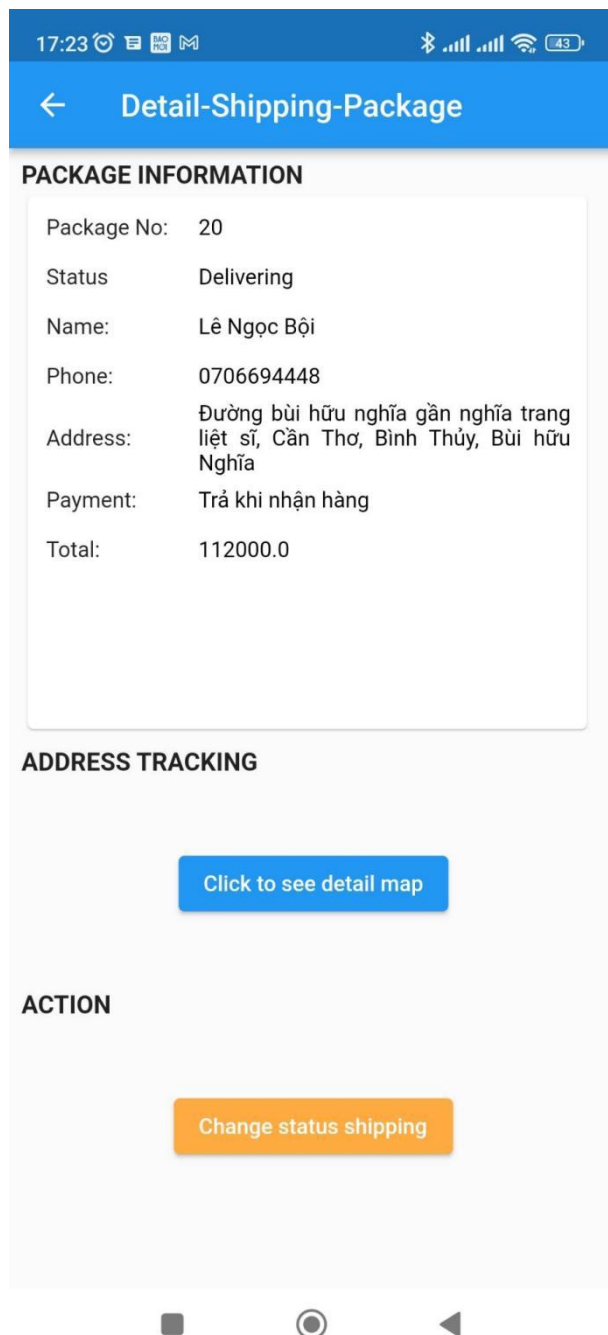
Trong chức năng này, người giao hàng sẽ có thể xem được danh sách những đơn hàng được chỉ định cho mình cũng như có thể thay đổi trạng thái đơn hàng và xem chỉ đường đến địa chỉ giao hàng của đơn hàng.



*Hình 38. Danh sách đơn hàng được chỉ định*

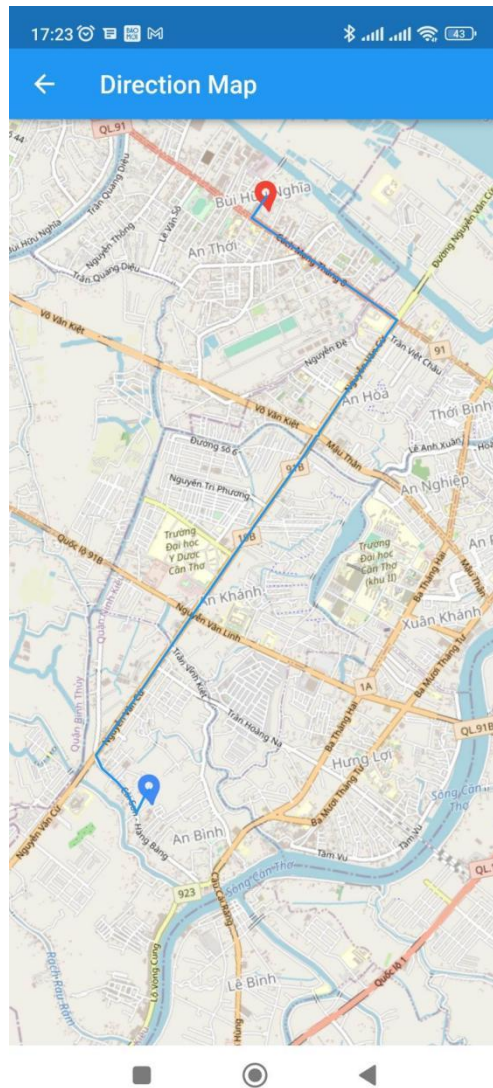
Sau khi vào danh sách đơn hàng được chỉ định, người giao hàng có thể click vào một trong các trường hiển thị để xem chi tiết của đơn hàng.



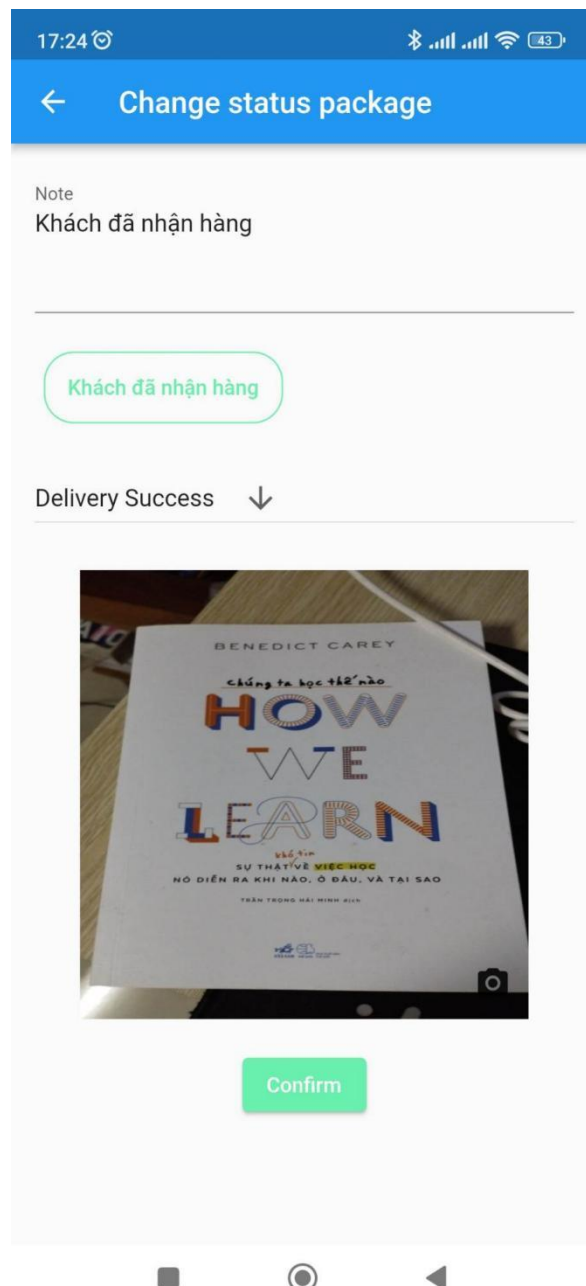


*Hình 39. Xem chi tiết của đơn hàng số 20*

Tại đây, người giao hàng có thể xem chỉ dẫn đường đi đến đơn hàng hoặc là thay đổi trạng thái của đơn hàng.



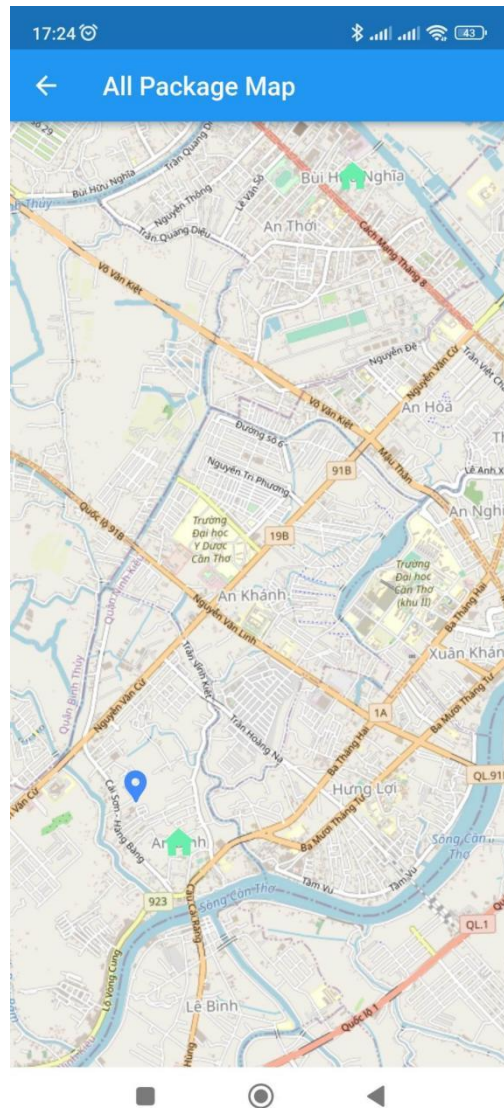
Hình 40. Xem chỉ đường từ người giao hàng đến đơn hàng



Hình 41. Thay đổi trạng thái của đơn hàng

### 5.3. Xem tất cả địa chỉ của đơn hàng

Chức năng này giúp người giao hàng có các nhìn tổng quát về vị trí của các đơn hàng được chỉ định, cũng như có thể sắp xếp đường đi từ trước để có thể tối ưu quãng đường vận chuyển.



Hình 42. Xem tất cả địa chỉ của các đơn hàng

## PHẦN 3: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 1. KẾT LUẬN

#### 1.1. Kết quả đạt được

Thiết kế và xây dựng thành công “**Sàn thương mại điện tử phân hệ sử dụng Django để thiết kế Back-end và Flutter để thiết kế ứng dụng mobile**” với hầu hết những chức năng đã đề ra. Đối tượng sử dụng website có năm nhóm chính: khách vãng lai, khách hàng, đại lý, người quản trị và người giao hàng

Khách vãng lai đã có thể tìm kiếm sản phẩm theo tên, xem chi tiết sản phẩm, xem đánh giá sản phẩm nhưng bị giới hạn không được cho thêm đồ vào giỏ hàng cũng như là thanh toán, không được phép bình luận cũng như phản hồi bình luận của các khách hàng khác.

Khách hàng ngoại trừ những chức năng được thừa kế của khách vãng lai thì khách hàng có thể bình luận sản phẩm mà mình đã mua trước đó, cũng như là cho sản phẩm vào giỏ hàng và thanh toán đơn đặt hàng.

Đại lý đã có thể quản lý sản phẩm (thêm, sửa và xóa sản phẩm), có thể tạo phiếu nhập để nhập hàng, đại lý có thể thay đổi trạng thái đơn đặt hàng của khách hàng mà thuộc về những sản phẩm của đại lý, nhưng không được thay đổi trạng thái cuối cùng của đơn đặt hàng.

Người giao hàng đã có thể xem được thông tin của đơn hàng mà mình được giao, xem chỉ đường từ nơi hiện tại đến địa chỉ của khách hàng, và người giao hàng là người mà sẽ thay đổi trạng thái cuối cùng của đơn hàng.

Người quản trị đã có thể quản lý tài khoản, quản lý đánh giá và quản lý đơn hàng của toàn bộ hệ thống, đơn hàng cũng được người quản trị giao cho người giao hàng để đưa đến cho khách hàng.

#### 1.2. Hạn chế

Tuy đạt được những mục tiêu đã đề ra, nhưng hệ thống vẫn còn một số hạn chế nhất định như: giao diện vẫn còn đơn giản quá mức, những chức năng như thống kê chưa thể đưa lên giao diện.

## **2. HƯỚNG PHÁT TRIỂN**

Với những gì đạt được và những hạn chế thì hướng phát triển đề tài của chúng em là thêm chức năng cho hệ thống ( chatbot, tìm kiếm bằng giọng nói, xem hình ảnh 3D, ... ), phát triển ứng dụng chính lên điện thoại di động, chỉnh sửa lại giao diện cho đặc sắc hơn, tăng cường hiệu năng, bảo mật cho hệ thống, thêm chức năng gợi ý đường đi tối ưu cho app giao hàng.

## PHỤ LỤC

### CÀI ĐẶT DOCKER ĐỂ CHẠY BACK-END VÀ DATABASE

#### 1. Môi trường cài đặt:

- Hệ điều hành: Ubuntu 22.04
- Ram: 1GB
- CPU: 1 Core
- Ổ cứng: 30GB
- Có kết nối internet

#### 2. Các bước cài đặt

Cập nhật lại **index package** của apt và cài đặt **packages** cho phép apt sử dụng một **responsitory** qua HTTPS

```
hen@bnshop: ~  
hen@bnshop:~$ sudo apt-get update  
Hit:1 http://archive.ubuntu.com/ubuntu jammy InRelease  
Hit:2 http://security.ubuntu.com/ubuntu jammy-security InRelease  
Hit:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease  
Hit:4 http://archive.ubuntu.com/ubuntu jammy-backports InRelease  
Reading package lists... Done  
hen@bnshop:~$ sudo apt-get install \  
> ca-certificates \  
> curl \  
> gnupg \  
> lsb-release  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
lsb-release is already the newest version (11.1.0ubuntu4).  
lsb-release set to manually installed.  
ca-certificates is already the newest version (20211016ubuntu0.22.04.1).  
ca-certificates set to manually installed.  
curl is already the newest version (7.81.0-1ubuntu1.6).  
gnupg is already the newest version (2.2.27-3ubuntu2.1).  
gnupg set to manually installed.  
The following packages were automatically installed and are no longer required:  
  libflashrom1 libftdi1-2  
Use 'sudo apt autoremove' to remove them.  
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.  
hen@bnshop:~$
```

Hình 43. Cập nhật apt của Ubuntu



Tiếp theo là thêm khóa GPG của Docker

```
hen@bnshop: ~  
hen@bnshop:~$ sudo mkdir -p /etc/apt/keyrings  
hen@bnshop:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg  
hen@bnshop:~$
```

*Hình 44. Thêm khóa GPG của Docker*

Thêm một vài câu lệnh để set up Docker

```
hen@bnshop: ~  
hen@bnshop:~$ echo \  
> "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \  
> $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

*Hình 45. Câu lệnh set up Docker*

Cài đặt Docker Engine, Containerd và Docker Compose

```
hen@bnshop: ~  
hen@bnshop:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following packages were automatically installed and are no longer required:  
  libflashrom1 libftdi1-2
```

*Hình 46. Câu lệnh cài đặt Docker*

Cấu hình file Dockerfile và docker-compose.yml



```

hen@bnshop: ~/BNShop/backend/bnShopApi
# pull official base image
FROM python:3.10.8-alpine

RUN apk update
RUN apk add build-base
RUN apk add --no-cache gcc
RUN apk add --no-cache libpq-dev
RUN apk add --no-cache python3-dev
RUN apk add --no-cache postgresql-dev
RUN apk add --no-cache musl-dev

# set work directory
WORKDIR /usr/src/bnShopApi/

# set environment variables
ENV PYTHONDONTWRITEBYTECODE 1
ENV PYTHONUNBUFFERED 1

# install dependencies
RUN pip install --upgrade pip
COPY ./requirements.txt .
RUN pip install -r requirements.txt

# copy project
COPY . .

```

Hình 47. Cấu hình file Dockerfile

```

hen@bnshop: ~/BNShop/backend/bnShopApi
version: '3.8'

services:
  web:
    build: ./
    command: python manage.py runserver 0.0.0.0:8000
    volumes:
      - ../usr/src/bnShopApi/
    ports:
      - 8000:8000
    depends_on:
      - db
  db:
    image: postgres:15
    volumes:
      - postgres_data:/var/lib/postgresql/data
    environment:
      - POSTGRES_USER=postgres
      - POSTGRES_PASSWORD=12345678
      - POSTGRES_DB=bnshop
  redis:
    image: redis:alpine
    container_name: redis
    ports:
      - 6379:6379
    healthcheck:
      test: ["CMD", "redis-cli", "ping"]
      interval: 2s
      timeout: 3s
      retries: 10
volumes:
  postgres_data:

```

Hình 48. Cấu hình file docker-compose.yml

Build container cho web, database và redis (để sử dụng websocket)

```
hen@bnshop: ~/BNShop/backend/bnShopApi
hen@bnshop:~/BNShop/backend/bnShopApi$ sudo docker-compose up -d --build
Building web
Sending build context to Docker daemon 708.1kB
Step 1/15 : FROM python:3.10.8-alpine
> 0d0305f5d056
```

Hình 49. Build container

Sau khi build xong thì kiểm tra lại xem container có được build thành công hay là không

```
hen@bnshop: ~/BNShop/backend/bnShopApi
hen@bnshop:~/BNShop/backend/bnShopApi$ sudo docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
781bb4d27c03	bnshopapi_web	"python manage.py ru..."	30 seconds ago	Up 27 seconds	0.0.0.0:8000->8000/tcp, :::8000->8000/tcp	bnshopapi_web_1
53db1c1da3e3	redis:alpine	"docker-entrypoint.s..."	33 seconds ago	Up 30 seconds (healthy)	0.0.0.0:6379->6379/tcp, :::6379->6379/tcp	redis
c2c9f044abdd	postgres:15	"docker-entrypoint.s..."	33 seconds ago	Up 30 seconds	5432/tcp	bnshopapi_db_1

```
hen@bnshop:~/BNShop/backend/bnShopApi$
```

Hình 50. Liệt kê các container đang chạy hiện tại

Sau khi build container thành công thì có thể truy cập server Django với port là 8000, database PostgreSQL với port là 5432 và Redis với port là 6379.

## TÀI LIỆU THAM KHẢO

- [1] George Reese (2000), *Database Programming with JDBC and Java, Second Edition*, O'Reilly & Associates, <https://web.archive.org/web/20120505084022/http://java.sun.com/developer/Books/jdbc/ch07.pdf>, truy cập vào ngày 09/09/2022.
- [2] Tham khảo Django, <https://www.djangoproject.com/>
- [3] Tham khảo Flutter, <https://flutter.dev/>
- [4] Tham khảo PostgreSQL, <https://www.postgresql.org/>
- [5] Tham khảo Docker, <https://www.digitalocean.com/>