

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO



BÁO CÁO ĐỒ ÁN CÔNG NGHỆ PHẦN MỀM

Đề tài:

ỨNG DỤNG TẶNG QUÀ

Giảng viên hướng dẫn: Thầy Huỳnh Xuân Phụng

Lớp: Công nghệ phần mềm

Sinh viên:

Nguyễn Lâm Gia Khang 18110132

Nguyễn Dương Đạt 18110092

Nguyễn Đình Phú 18110175

TP. Hồ Chí Minh, tháng 6, năm 2021

Mục lục

MỤC LỤC HÌNH ẢNH.....	2
NỘI DUNG	4
I. GIỚI THIỆU ĐỀ TÀI	4
1. Lý do chọn đề tài	4
2. Mục tiêu đề tài.....	4
3. Phương pháp thực hiện	4
II. QUÁ TRÌNH THỰC HIỆN.....	5
1. Tài liệu đặc tả yêu cầu	5
1.1. Phiên bản 1.0	5
1.2. Phiên bản 1.1	6
1.3. Phiên bản 2.0	9
1.4. Phiên bản 3.0	11
2. Kiến trúc hệ thống	13
2.1. Mô hình kiến trúc	13
2.2. Ứng dụng mô hình mvc	15
3. Sơ đồ lớp của hệ thống	19
4. Sơ đồ usecase.....	19
5. Sơ đồ tuần tự của hệ thống	21
5.1. Vẽ 3 sơ đồ tuần tự theo yêu cầu	21
5.2. Sơ đồ tuần tự trước báo cáo.....	23
6. Test case	51
7. Unit test	51
III. CHẠY THỬ NGHIỆM HỆ THỐNG.....	52
1. Thử nghiệm các API	52
1.1. Phiên bản 1.0	52
1.2. Phiên bản 1.1	54
1.3. Phiên bản 2.0	62
1.4. Phiên bản 3.0	70
2. Thử nghiệm hệ thống front-end	86
IV. SỬ DỤNG GITHUB	92
V.KẾT LUẬN	93
4.1. Đánh giá mức độ hoàn thành.....	93
4.2. Khó khăn	93
4.3. Hướng phát triển.....	93

Mục lục hình ảnh

Hình 2. 1: Mô hình mvc	14
Hình 2. 2: Front-end	14
Hình 2. 3: backend.....	15
Hình 2. 4: database	15
Hình 2. 5: DAO	16
Hình 2. 6: Service	16
Hình 2. 7: model	17
Hình 2. 8: controller	18
Hình 2. 9: view	19
Hình 2. 10: Sơ đồ lớp	19
Hình 2. 11: Usecase phiên bản trước báo cáo	20
Hình 2. 12: usecase phiên bản sau báo cáo	21
Hình 2. 13: chỉnh sửa gói quà.....	22
Hình 2. 14: Thêm gói quà.....	22
Hình 2. 15: thêm item.....	23
Hình 2. 16: Xóa mặt hàng trong kho	25
Hình 2. 17: Component diagram đăng ký	26
Hình 2. 18: Sign up.....	27
Hình 2. 19: Sign in.....	29
Hình 2. 20: xem lịch sử đơn hàng	32
Hình 2. 21: xem phản hồi	34
Hình 2. 22: Trả lời phản hồi	36
Hình 2. 23: xử lý đơn hàng.....	37
Hình 2. 24: báo cáo tình trạng đơn hàng	39
Hình 2. 25: xem trạng thái đơn hàng	41
Hình 2. 26: Xem mặt hàng	43
Hình 2. 27: quản lý đặt hàng	46
Hình 2. 28: quản lý hệ thống	48
Hình 2. 29: xem thống kê	51
Hình 2. 30: Unit test	52
Hình 2. 31: Unit test	52
Hình 3. 1: API đăng ký	53
Hình 3. 2: API đăng nhập	54
Hình 3. 3: API thêm thông tin khách hàng	55
Hình 3. 4: API chỉnh sửa thông tin khách hàng	56
Hình 3. 5: API lấy tất cả thông tin khách hàng	56
Hình 3. 6: API lấy thông tin khách hàng qua tên	57
Hình 3. 7: API xóa thông tin khách hàng	57
Hình 3. 8: API thêm phụ kiện vào một món quà bất kỳ.....	59
Hình 3. 9: API chỉnh sửa một phụ kiện nào đó	60
Hình 3. 10: API Lấy tất cả các đơn hàng kèm phụ kiện.....	60

Hình 3. 11: API lấy đơn hàng kèm phụ kiện theo mã đơn hàng	61
Hình 3. 12: Xóa đơn hàng kèm phụ kiện theo mã trong cơ sở dữ liệu.....	62
Hình 3. 13: API đặt quà	63
Hình 3. 14: API xem lịch sử đơn hàng qua mã đơn hàng	64
Hình 3. 15: API xem lịch sử đơn hàng qua tên tài khoản	64
Hình 3. 16: API lấy tất cả đơn hàng của khách hàng	65
Hình 3. 17: API lấy tất cả đơn hàng để xử lý	66
Hình 3. 18: API cập nhật xử lý đơn hàng	66
Hình 3. 19: API lấy tất cả các phản hồi	67
Hình 3. 20: API lấy phản hồi theo tên tài khoản	68
Hình 3. 21: Lấy phản hồi theo mã gói quà	68
Hình 3. 22: API phản hồi cho khách hàng.....	69
Hình 3. 23: API thêm phản hồi.....	70
Hình 3. 24: Thêm một mặt hàng mới	71
Hình 3. 25: Lấy tất cả các mặt hàng có sẵn trong kho	71
Hình 3. 26: Lấy thông tin mặt hàng qua mã mặt hàng	72
Hình 3. 27: chỉnh sửa thông tin mặt hàng	73
Hình 3. 28: Lấy số lượng mặt hàng	73
Hình 3. 29: Lấy tất cả các gói quà	74
Hình 3. 30: Lấy gói quà thông qua mã gói quà	75
Hình 3. 31: Tại một gói quà mới	76
Hình 3. 32: chỉnh sửa thông tin một gói quà	77
Hình 3. 33: API lấy số lượng gói quà.....	77
Hình 3. 34: API lấy tất cả các phụ kiện đi kèm.....	78
Hình 3. 35: API lấy phụ kiện đi kèm thông qua mã phụ kiện	79
Hình 3. 36: API thêm một phụ kiện	80
Hình 3. 37: API chỉnh sửa thông tin phụ kiện.....	81
Hình 3. 38: API lấy số lượng phụ kiện.....	81
Hình 3. 39: API lấy tất cả gói quà kèm mặt hàng đi kèm	82
Hình 3. 40: Lấy gói quà kèm mặt hàng thông qua mã gói quà	83
Hình 3. 41: Thêm gói quà kèm mặt hàng	84
Hình 3. 42: Chỉnh sửa gói quà kèm mặt hàng	85
Hình 3. 43: Xóa các mặt hàng đi kèm gói quà	85
Hình 3. 44: Trang đăng ký	86
Hình 3. 45: Trang đăng nhập	87
Hình 3. 46: trang thông tin khách hàng	87
Hình 3. 47: trang chủ	88
Hình 3. 48: trang đặt quà	88
Hình 3. 49: Trang đặt quà	89
Hình 3. 50: trang giao diện admin mục quản lý gói quà	89
Hình 3. 51: Trang thêm một gói quà mới	90
Hình 3. 52: Trang giao diện admin mục quản lý mặt hàng	90
Hình 3. 53: Trang thêm một mặt hàng mới	91

Hình 3. 54: Trang admin mục quản lý phụ kiện.....	91
Hình 3. 55: Trang thêm phụ kiện mới	92

NỘI DUNG

I. Giới thiệu đề tài

1. Lý do chọn đề tài

Chuyển đổi số đang là xu hướng của thế giới và đất nước chúng ta cũng không ngoại lệ. Việc chuyển đổi số giúp các công ty cũng như các cửa hàng có thể thích ứng trong mùa dịch covid-19. Tuy nhiên, có rất ít cửa hàng có thể chuyển đổi số thành công do các vấn đề về kinh phí cũng như nguồn nhân lực. Chính vì điều này nhóm chúng em xin phép chọn đề tài ứng dụng tặng quà nhằm mục đích cung cấp một bộ api có thể tích hợp và phát triển hệ thống cửa hàng bán quà tặng cũng như hỗ trợ một phần nhỏ trong việc chuyển đổi số của các cửa hàng.

2. Mục tiêu đề tài

Chương trình cung cấp một bộ api đơn giản, dễ sử dụng, đồng thời cũng cung cấp môi trường chạy thử và các hướng dẫn liên quan. Ngoài ra ứng dụng cũng cung cấp giao diện để việc kiểm thử trở nên dễ dàng hơn. Các tính năng hỗ trợ sẽ liên tục được cập nhật trong các phiên bản sắp tới.

3. Phương pháp thực hiện

Ứng dụng sử dụng các công nghệ mới trong việc xây dựng api, cũng như các công nghệ trong xây dựng cơ sở dữ liệu để có thể dễ dàng tích hợp với bất kỳ cơ sở dữ liệu nào. Ngoài ra nhóm cũng đã tham khảo một số ứng dụng tặng quà khác để giúp chương trình dễ dàng sử dụng hơn. Chương trình sử dụng giao diện web nên việc kiểm thử dễ dàng được sử dụng, đồng thời các api cũng có các mô tả để tiện cho việc tích hợp vào hệ thống. Bên cạnh đó hiện thực hóa trên web để có thể mang đến sự tiện lợi trong quá trình sử dụng. Thiết kế và thực hiện theo mô hình MVC và 3-tier để có thể nâng cấp và sửa chữa trong quá trình sử dụng.

II. Quá trình thực hiện

1. Tài liệu đặc tả yêu cầu

1.1. Phiên bản 1.0

Chapter	Description
Preface	Đây là tài liệu đặc tả dành cho ban lãnh đạo, các bên liên quan nhằm giúp cho mọi người hiểu hơn về dự án đang phát triển. Dự án sẽ ra mắt với những chức năng cơ bản và các chức năng mới sẽ liên tục được cập nhật và phát triển nhằm giúp hệ thống ổn định và phát triển lâu dài.
Introduction	Ứng dụng được triển khai trên website sẽ là một nền tảng trung gian kết nối các cửa hàng đồ lưu niệm, quà tặng, hoa, ... cũng như tất cả các cửa hàng có dịch vụ quà tặng. Hệ thống cũng sẽ kết nối với các bên vận chuyển trung gian như grab, be, ... để có thể tối ưu hóa tốc độ chuyển hàng.
Glossary	V1.0: phiên bản đầu tiên
User requirements definition	<p>Phiên bản 1.0 sẽ mang đến cho khách hàng các chức năng sau:</p> <ul style="list-style-type: none"> ⊕ Yêu cầu chức năng: <ol style="list-style-type: none"> 1. Mô tả: API có chức năng đăng ký và đăng nhập tài khoản. ⊕ Yêu cầu phi chức năng: <ol style="list-style-type: none"> 1. Sức chứa khoảng 1000 người dùng truy cập cùng lúc. 2. Chức năng linh hoạt cho việc đăng nhập giữa các actor khác nhau. 3. Khả năng bị tấn công. 4. Có khả năng khôi phục dữ liệu khi bị mất. 5. Toàn vẹn dữ liệu.

System architecture	Hệ thống sử dụng các công nghệ mới: spring boot, spring jpa data, jersey, vuejs, ...
System requirements specification	<p>Yêu cầu chức năng:</p> <ol style="list-style-type: none"> Sử dụng mô hình mvc + 3 tier để thiết kế. Phân loại người sử dụng (user và admin). <p>Yêu cầu phi chức năng</p> <ol style="list-style-type: none"> Khả năng bị tấn công (100 session thì có 1 session tấn công). Các dữ liệu đã được backup nên dễ dàng phục hồi. Giao diện hiện đại phù hợp với tất cả lứa tuổi.
System models	<pre> useCaseDiagram actor Guest actor User useCase SignUp useCase SignIn useCase Receive Guest --> SignUp User --> SignUp User --> SignIn User --> Receive </pre>
System evolution	<ol style="list-style-type: none"> Tích hợp thêm các chức năng đăng nhập bằng google, facebook, ...
Appendices	<p>Phiên bản phần mềm: v1.0.</p> <p>Hệ quản trị cơ sở dữ liệu: sql server.</p> <p>Công nghệ sử dụng: spring boot, jersey, spring data jpa, vuejs.</p> <p>Host: Heroku.</p>
Index	

1.2. Phiên bản 1.1

Chapter	Description
Preface	Đây là tài liệu đặc tả dành cho ban lãnh đạo, các bên liên quan nhằm giúp cho mọi người hiểu hơn về dự án đang phát triển. Dự án sẽ ra mắt với những chức năng cơ bản và các chức năng mới sẽ liên tục được cập nhật và phát triển nhằm giúp hệ thống ổn định và phát triển lâu dài.
Introduction	Ứng dụng được triển khai trên website sẽ là một nền tảng trung gian kết nối các cửa hàng đồ lưu niệm, quà tặng, hoa, ... cũng như tất cả các cửa hàng có dịch vụ quà tặng. Hệ thống cũng sẽ kết nối với các bên vận chuyển trung gian như grab, be, ... để có thể tối ưu hóa tốc độ chuyển hàng.
Glossary	V1.1: phiên bản thứ hai
User requirements definition	Phiên bản 1.1 sẽ mang đến cho khách hàng các chức năng sau:  Yêu cầu chức năng: 1. Mô tả: API được sử dụng để đăng ký thêm thông tin của khách hàng: POST, PUT, GET, DELETE 2. Mô tả: Khi đặt một đơn hàng chúng ta có thể quản lý các addOn đi kèm. AddOn là những món quà được đính kèm vào món quà chính. 1 món quà có thể có nhiều addOn.

	<ul style="list-style-type: none"> ✚ Yêu cầu phi chức năng: <ol style="list-style-type: none"> 1. Sức chứa khoảng 1000 người dùng truy cập cùng lúc. 2. Ít có khả năng bị tấn công. 3. Có khả năng khôi phục dữ liệu khi bị mất. 4. Toàn vẹn dữ liệu.
System architecture	Hệ thống sử dụng các công nghệ mới: spring boot, hibernate, ...
System requirements specification	<ul style="list-style-type: none"> ✚ Yêu cầu chức năng: <ol style="list-style-type: none"> 1. Sử dụng mô hình mvc + 3 tier để thiết kế. 2. Có khả năng mở rộng và phát triển. 3. Tính ổn định. <ul style="list-style-type: none"> ✚ Yêu cầu phi chức năng <ol style="list-style-type: none"> 1. Khả năng bị tấn công (100 session thì có 1 session tấn công). 2. Các dữ liệu đã được backup nên dễ dàng phục hồi. 3. Giao diện hiện đại phù hợp với tất cả lứa tuổi.
System models	
System evolution	
Appendices	<p>Phiên bản phần mềm: v1.1.</p> <p>Hệ quản trị cơ sở dữ liệu: sql server.</p> <p>Công nghệ sử dụng: spring boot, jersey, spring data jpa, vuejs.</p> <p>Host: của nhóm.</p>

Index	
-------	--

1.3. Phiên bản 2.0

Chapter	Description
Preface	Đây là tài liệu đặc tả dành cho ban lãnh đạo, các bên liên quan nhằm giúp cho mọi người hiểu hơn về dự án đang phát triển. Dự án sẽ ra mắt với những chức năng cơ bản và các chức năng mới sẽ liên tục được cập nhật và phát triển nhằm giúp hệ thống ổn định và phát triển lâu dài.
Introduction	Ứng dụng được triển khai trên website sẽ là một nền tảng trung gian kết nối các cửa hàng đồ lưu niệm, quà tặng, hoa, ... cũng như tất cả các cửa hàng có dịch vụ quà tặng. Hệ thống cũng sẽ kết nối với các bên vận chuyển trung gian như grab, be, ... để có thể tối ưu hóa tốc độ chuyển hàng.
Glossary	V2.0: phiên bản thứ ba
User requirements definition	<p>Phiên bản 2.0 sẽ mang đến cho khách hàng các chức năng sau:</p> <ul style="list-style-type: none">  Yêu cầu chức năng: <ol style="list-style-type: none"> 1. Mô tả: Sử dụng API để tiến hành đặt một đơn hàng và quản lý nó. 2. Mô tả: API giúp chúng ta xử lý, vận chuyển tất cả các đơn hàng đã đặt. 3. Mô tả: API sử dụng để ghi comment và giải quyết các comments. 4. Khách hàng có thể dễ dàng tìm kiếm được các món hàng mình mong muốn để tặng như là đồ lưu niệm, quà tặng, hoa, ... 5. Người dùng có thể dễ dàng xem thông tin về các đơn hàng đã đặt.

	<p>6. Quản lý về công việc, các đơn hàng được giao cho nhân viên này vận chuyển, thống kê cuối ngày, trả lại các món hàng chưa giao thành công.</p> <p>7. Giao đơn hàng cho các dịch vụ vận chuyển toàn quốc và quản lý chúng.</p> <ul style="list-style-type: none"> ✚ Yêu cầu phi chức năng: <ol style="list-style-type: none"> 1. Ít có khả năng bị tấn công. 2. Có khả năng khôi phục dữ liệu khi bị mất. 3. Toàn vẹn dữ liệu. 4. Giao hàng đúng giờ, nhanh hay chậm. 5. Cách thức giao hàng.
System architecture	Hệ thống sử dụng các công nghệ mới: spring boot, hibernate, ...
System requirements specification	<ul style="list-style-type: none"> ✚ Yêu cầu chức năng: <ol style="list-style-type: none"> 1. Sử dụng mô hình mvc + 3 tier để thiết kế. 2. Có khả năng mở rộng và phát triển. 3. Tính ổn định. <ul style="list-style-type: none"> ✚ Yêu cầu phi chức năng <ol style="list-style-type: none"> 1. Khả năng bị tấn công (100 session thì có 1 session tấn công). 2. Các dữ liệu đã được backup nên dễ dàng phục hồi. 3. Giao diện hiện đại phù hợp với tất cả lứa tuổi.

System models	<pre> useCaseDiagram actor Guest actor User useCase SignUp useCase SignIn useCase Order useCase ViewOrderHistories useCase Pay useCase View useCase ReceiveShipment useCase ReportShipmentState useCase ViewShipmentState Guest --> SignUp User --> SignUp User --> SignIn User --> Order User --> ViewOrderHistories User --> Pay User --> View User --> ViewShipmentState User --> ReceiveShipment User --> ReportShipmentState Order <--><<include>> View Order <--><<include>> Pay </pre>
System evolution	1. Tích hợp với các hệ thống giao hàng khác.
Appendices	Phiên bản phần mềm: v2.0. Hệ quản trị cơ sở dữ liệu: sql server. Công nghệ sử dụng: spring boot, jersey, spring data jpa, vuejs. Host: của nhóm.
Index	

1.4. Phiên bản 3.0

Chapter	Description
Preface	Đây là tài liệu đặc tả dành cho ban lãnh đạo, các bên liên quan nhằm giúp cho mọi người hiểu hơn về dự án đang phát triển. Dự án sẽ ra mắt với những chức năng cơ bản và các chức năng mới sẽ liên tục được cập nhật và phát triển nhằm giúp hệ thống ổn định và phát triển lâu dài.
Introduction	Ứng dụng được triển khai trên website sẽ là một nền tảng trung gian kết nối các cửa hàng đồ lưu niệm, quà tặng, hoa, ... cũng như tất cả các cửa hàng có dịch vụ quà tặng. Hệ thống cũng sẽ

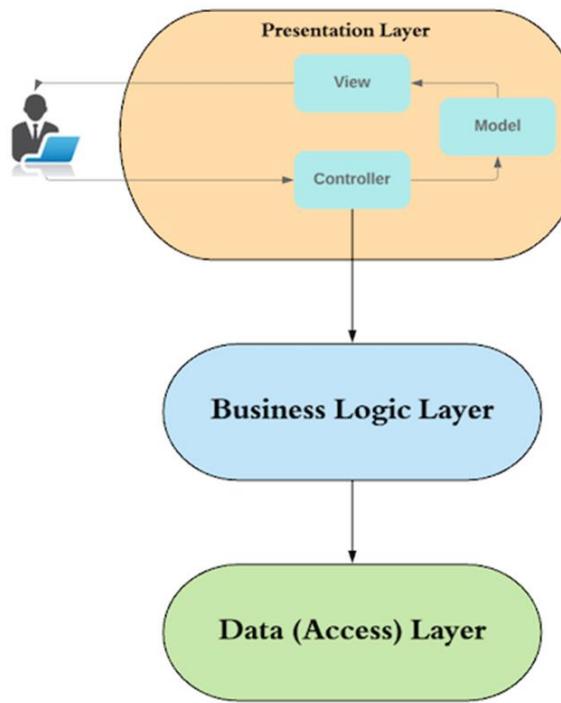
	kết nối với các bên vận chuyển trung gian như grab, be, ... để có thể tối ưu hóa tốc độ chuyển hàng.
Glossary	V3.0: phiên bản thứ tư
User requirements definition	<p>Phiên bản 3.0 sẽ mang đến cho khách hàng các chức năng sau:</p> <ul style="list-style-type: none"> + Yêu cầu chức năng: <ol style="list-style-type: none"> 1. Mô tả: Quản lý tất cả các mặt hàng trong kho. 2. Mô tả: Giftpack là API sử dụng để quản lý tất cả các gói quà trong kho. Một gói quà có thể có nhiều mặt hàng. 3. Mô tả: một đơn hàng có thể có một số phụ kiện đính kèm. API này được sử dụng để quản lý tất cả phụ kiện đó. 4. Mô tả: một gói quà thì bao gồm rất nhiều mặt hàng. Vì thế API được sử dụng để quản lý tất cả gói quà. 5. Admin có thể thống kê các đơn hàng đã và chưa hoàn tất, đơn hàng bị hủy. Tình trạng của từng đơn hàng. Thống kê đánh giá, doanh thu, doanh số. 6. Admin có thể thêm, xóa, sửa thông tin một sản phẩm bất kỳ. + Yêu cầu phi chức năng: <ol style="list-style-type: none"> 1. Ít có khả năng bị tấn công. 2. Có khả năng khôi phục dữ liệu khi bị mất. 3. Toàn vẹn dữ liệu.
System architecture	Hệ thống sử dụng các công nghệ mới: spring boot, hibernate, ...
System requirements specification	<ul style="list-style-type: none"> + Yêu cầu chức năng: <ol style="list-style-type: none"> 4. Sử dụng mô hình mvc + 3 tier để thiết kế. 5. Có khả năng mở rộng và phát triển. 6. Tính ổn định. + Yêu cầu phi chức năng <ol style="list-style-type: none"> 4. Khả năng bị tấn công (100 session thì có 1 session tấn công).

	<p>5. Các dữ liệu đã được backup nên dễ dàng phục hồi.</p> <p>6. Giao diện hiện đại phù hợp với tất cả lứa tuổi.</p>
System models	<pre> useCaseDiagram actor Guest actor User actor Admin useCase Sign up useCase Report Shipment State useCase Receive shipment useCase View shipment state useCase Sign in useCase Order useCase View useCase Pay useCase New Order Histories useCase View Complaint useCase Complaint settle useCase View stock statistics useCase Manage goods In stock useCase Manage System Guest --> Sign up User --> Sign up User --> Report Shipment State User --> Receive shipment User --> View shipment state User --> Sign in User --> Order User --> View User --> Pay User --> New Order Histories User --> View Complaint User --> Complaint settle User --> View stock statistics User --> Manage goods In stock User --> Manage System Admin --> Report Shipment State Admin --> View shipment state Admin --> Sign in Admin --> Order Admin --> View Admin --> Pay Admin --> New Order Histories Admin --> View Complaint Admin --> Complaint settle Admin --> View stock statistics Admin --> Manage goods In stock Admin --> Manage System Order <--> View : <<include>> Order <--> Pay : <<include>> View Complaint <--> Complaint settle : <<include>> </pre>
System evolution	<p>1. Tích hợp với các hệ thống giao hàng khác.</p>
Appendices	<p>Phiên bản phần mềm: v3.0.</p> <p>Hệ quản trị cơ sở dữ liệu: sql server.</p> <p>Công nghệ sử dụng: spring boot, jersey, spring data jpa, vuejs.</p> <p>Host: của nhóm.</p>
Index	

2. Kiến trúc hệ thống

2.1. Mô hình kiến trúc

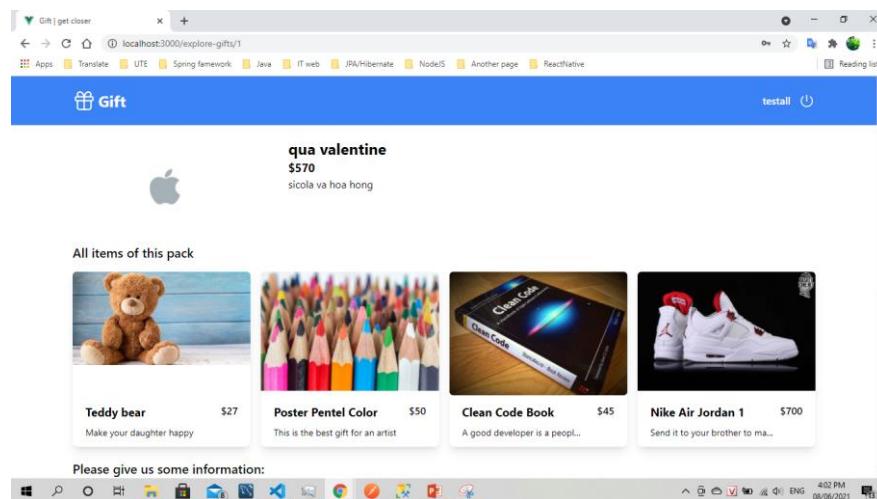
Ứng dụng tăng quà sử dụng mô hình kiến trúc mvc + 3 tier thuận lợi cho việc sửa lỗi, phát triển và nâng cấp các chức năng hệ thống.



Hình 2. 1: Mô hình mvc

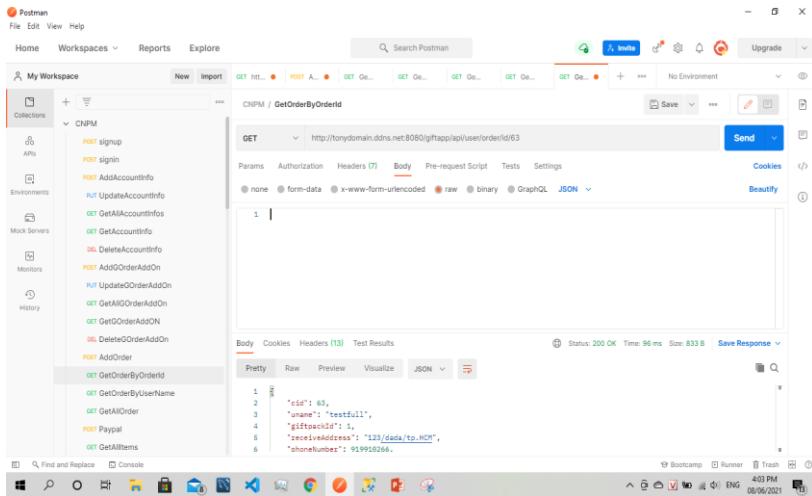
Ứng dụng được chia làm 3 phần: Front-end, backend và database giúp tách biệt và dễ dàng nâng cấp, sửa chữa khi cần thiết.

- Front-end: sử dụng html, css, javascript và vuejs để phát triển.



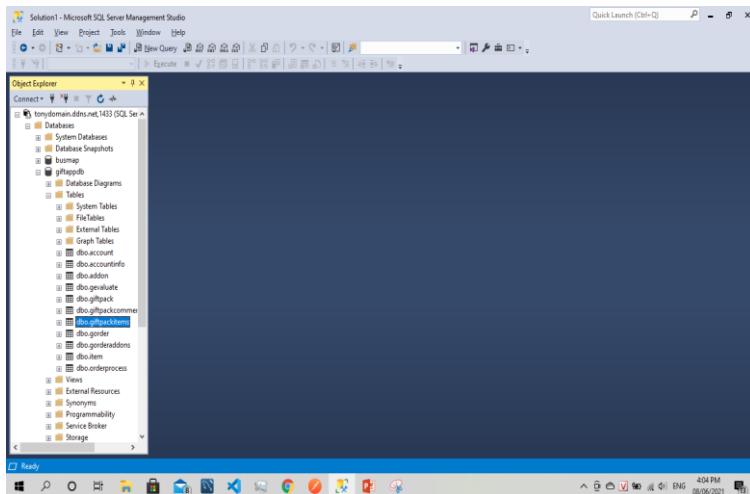
Hình 2. 2: Front-end

- Backend: Sử dụng ngôn ngữ java, thư viện jersey, spring boot, spring data jpa và kiểm thử bằng postman.



Hình 2. 3: backend

Database: sử dụng kĩ thuật ORM để phát triển.



Hình 2. 4: database

2.2. Ứng dụng mô hình mvc

- Data access layer: Tương tác với cơ sở dữ liệu.

```
■ accountInfoRepo.java  
■ accountRepo.java  
■ addonRepo.java  
■ gevaluateRepo.java  
■ giftpackCommentsRepo.java  
■ giftpackItemsRepo.java  
■ giftpackRepo.java  
■ gorderAddonsRepo.java  
■ gorderRepo.java  
■ itemRepo.java  
■ orderProcessRepo.java
```

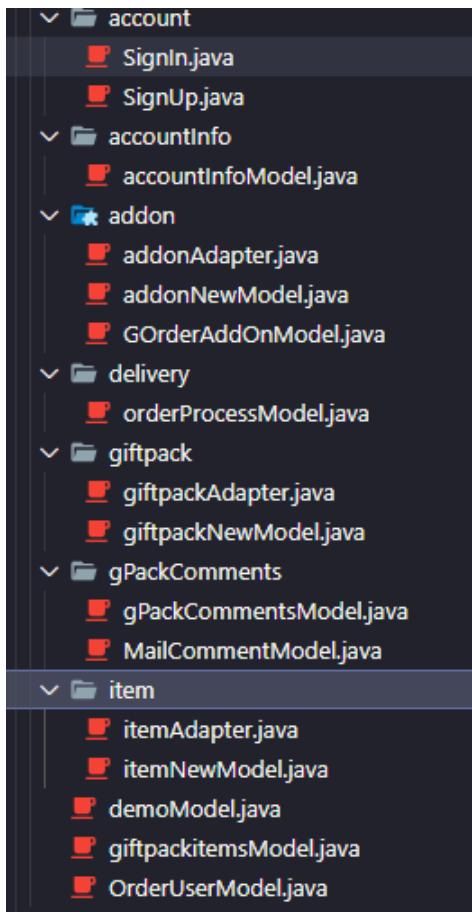
Hình 2. 5: DAO

- Business logic layer: xử lý các nghiệp vụ bài toán.

```
■ AccountInfoService.java  
■ accountService.java  
■ addonService.java  
■ deliveryService.java  
■ GiftpackItemsService.java  
■ giftpackService.java  
■ GOrderAddonService.java  
■ itemService.java  
■ MailService.java  
■ OrderService.java
```

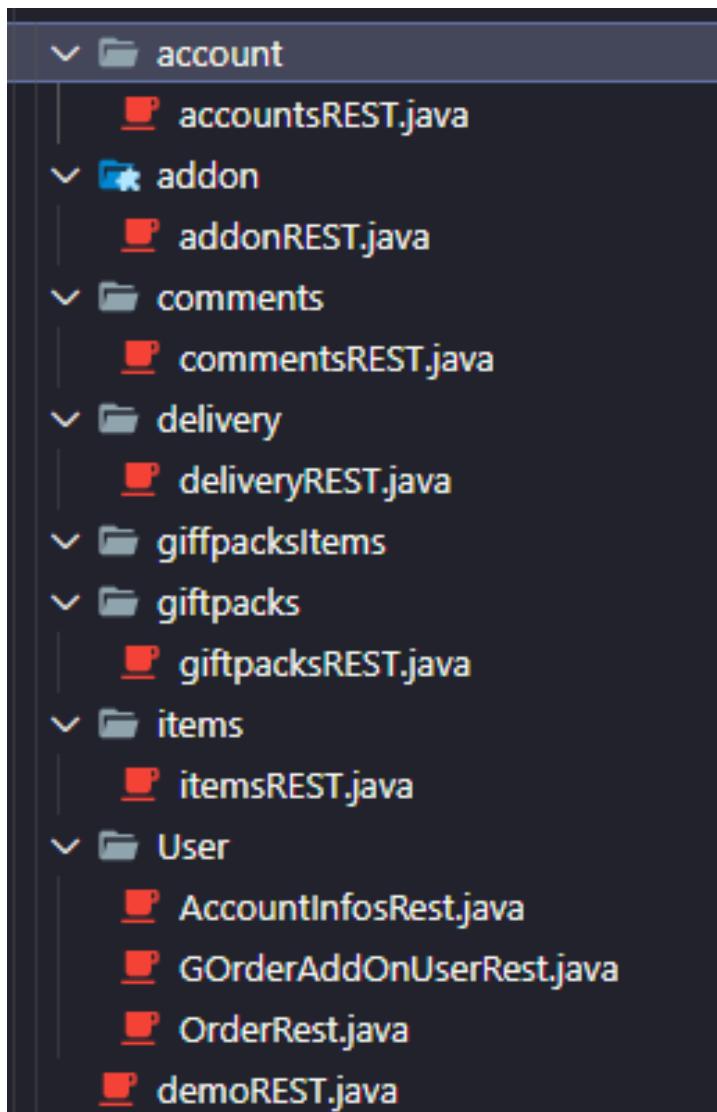
Hình 2. 6: Service

- Model: Khuôn mẫu cho việc hiển thị dữ liệu.



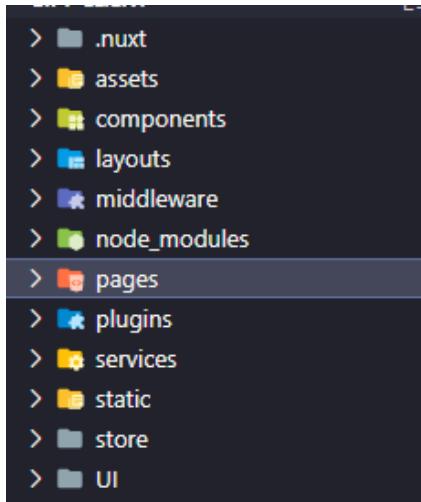
Hình 2. 7: *model*

- Controller: là cầu nối để thực hiện các phương thức khi nhận một yêu cầu từ client.



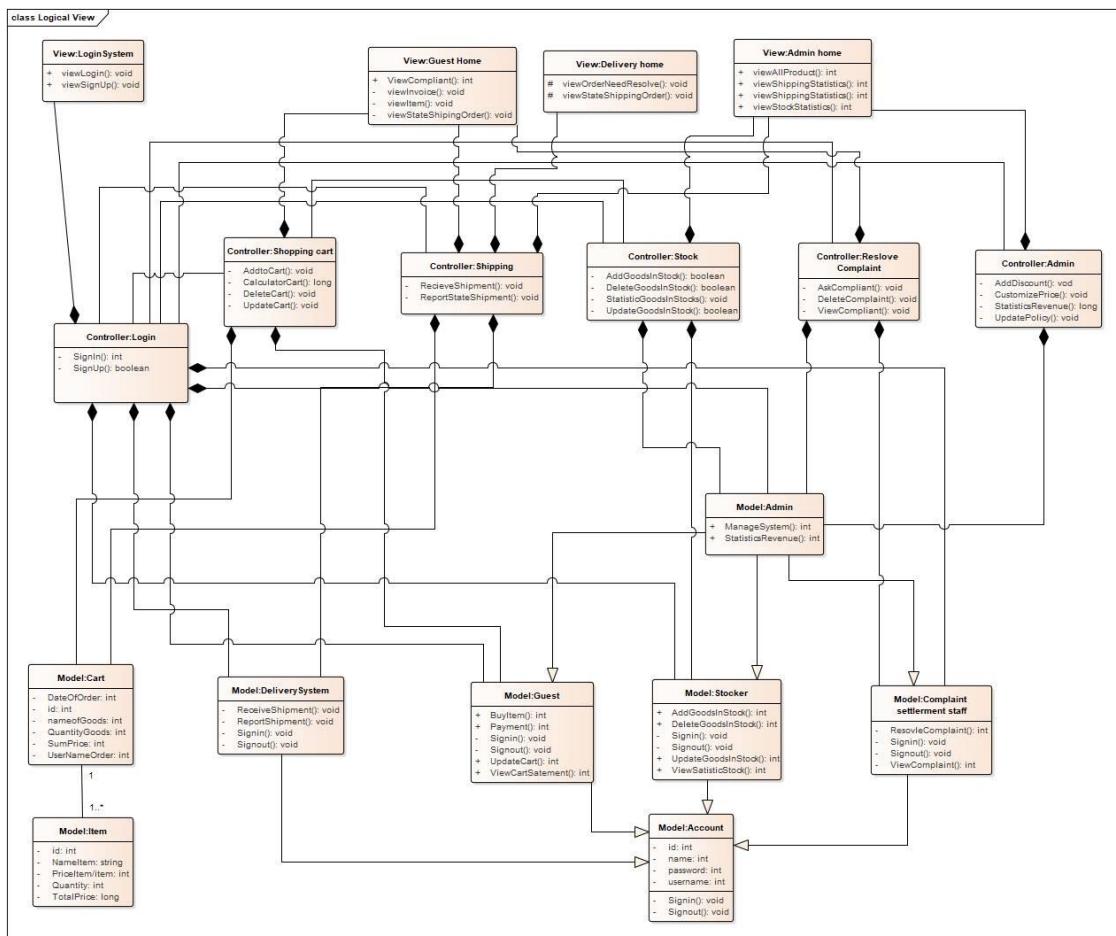
Hình 2. 8: controller

- View: Front-end



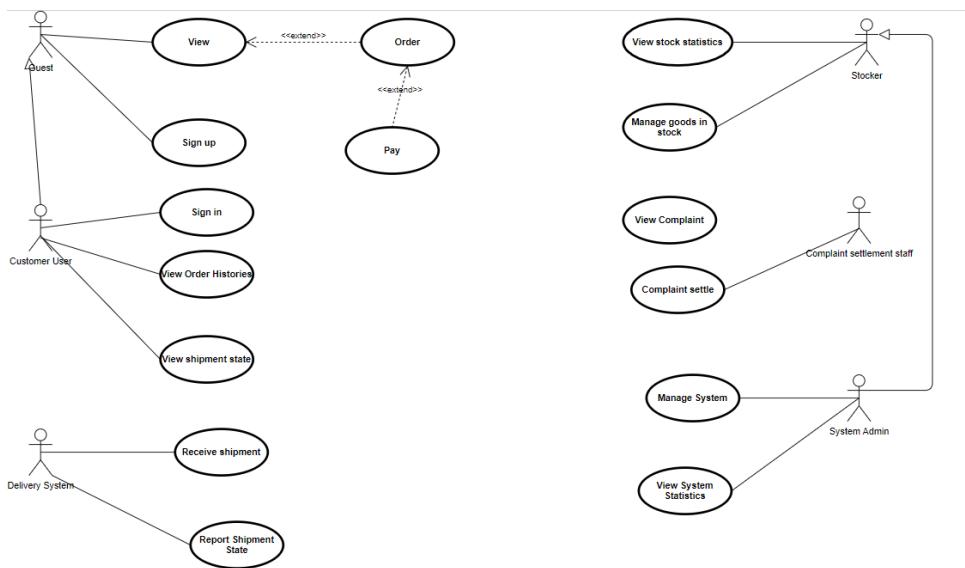
Hình 2. 9: view

3. Sơ đồ lớp của hệ thống



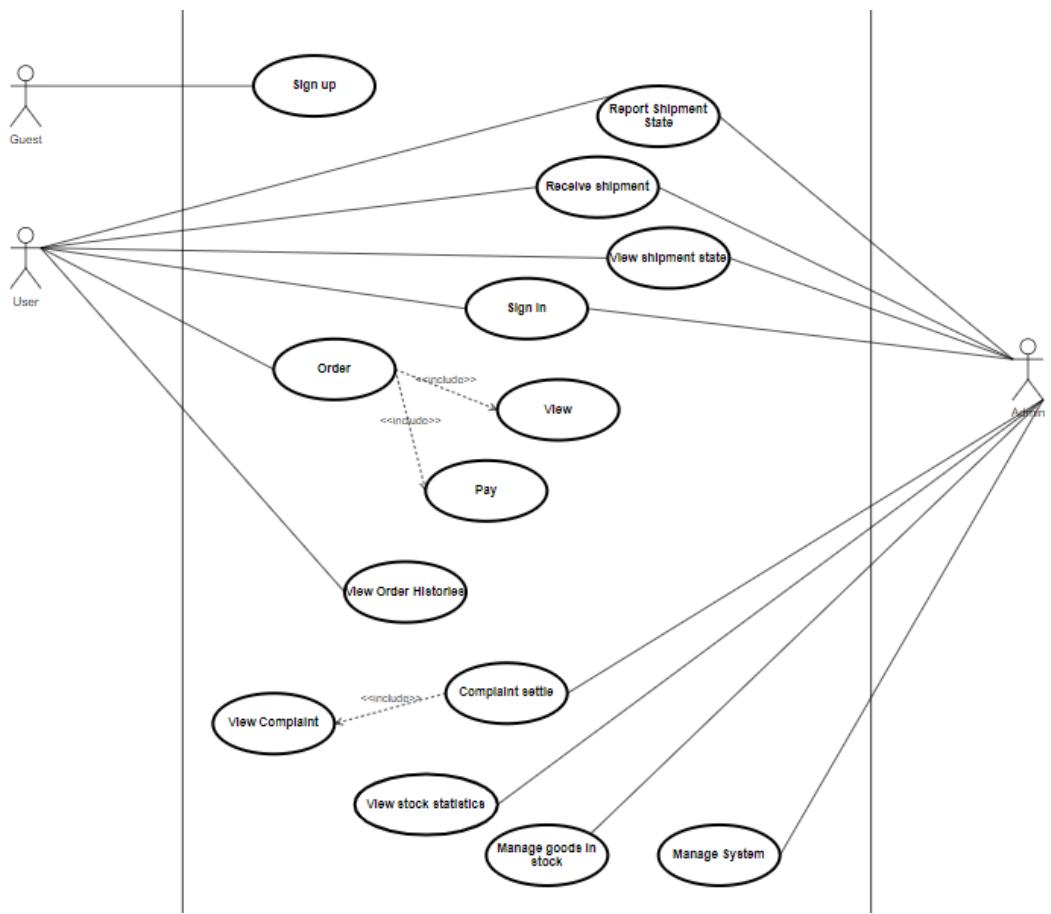
Hình 2. 10: Sơ đồ lớp

4. Sơ đồ usecase



Hình 2. 11: Usecase phiên bản trước báo cáo

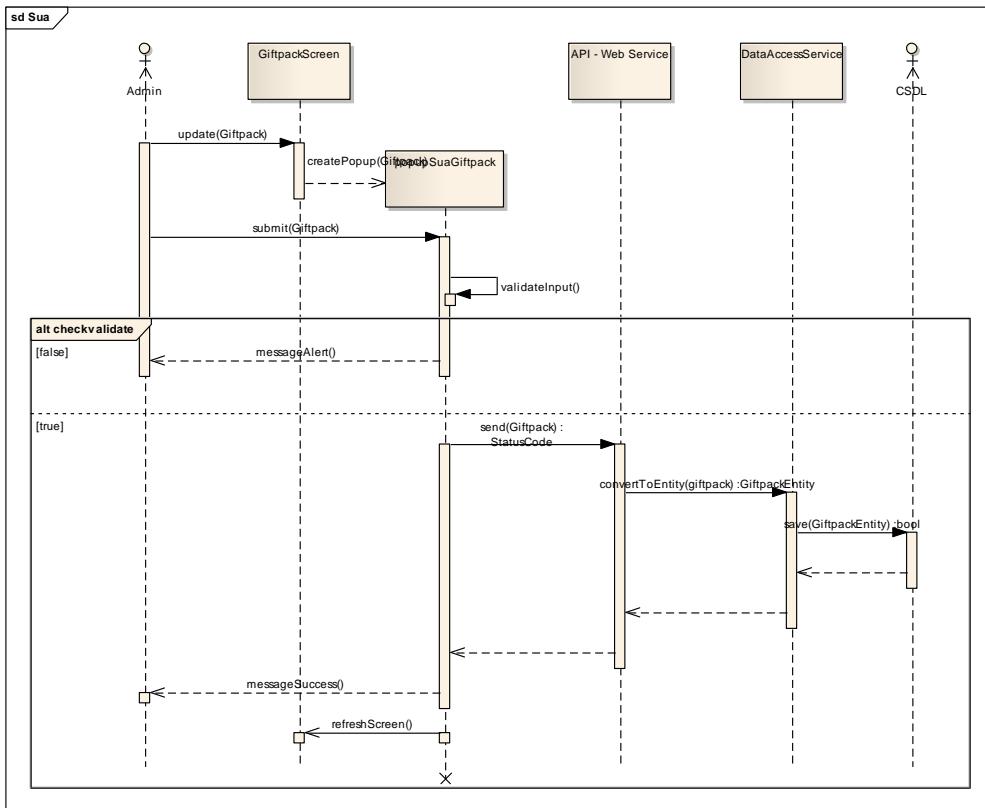
Note: nhóm em đã vẽ lại usecase theo yêu cầu.



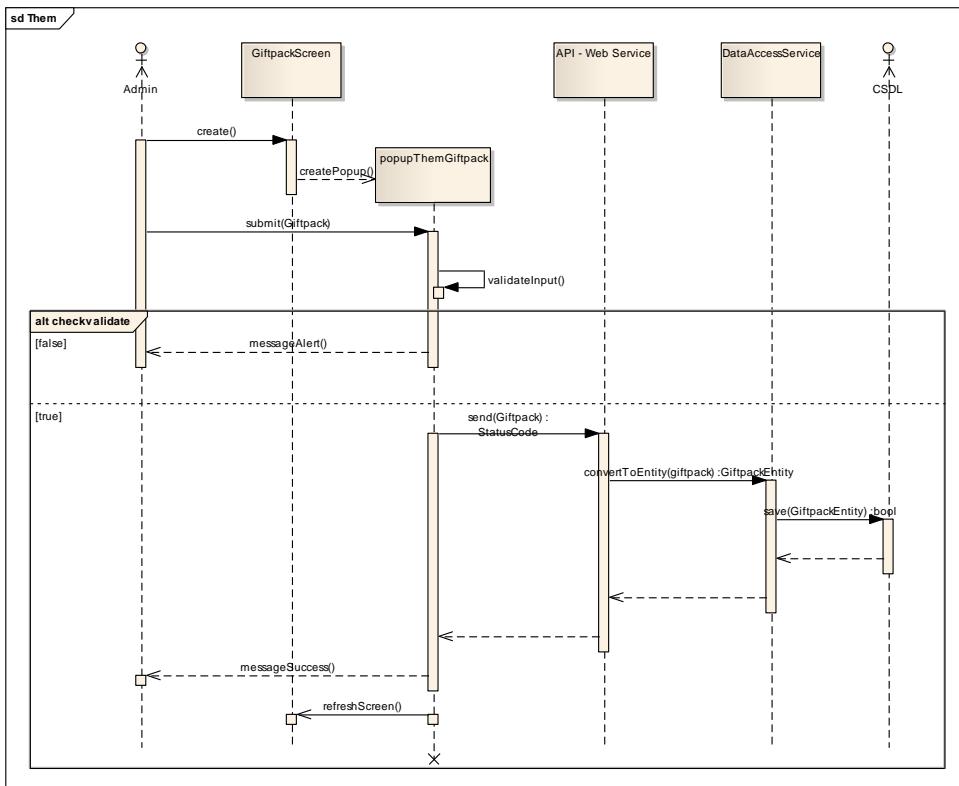
Hình 2. 12: usecase phiên bản sau báo cáo

5. Sơ đồ tuần tự của hệ thống

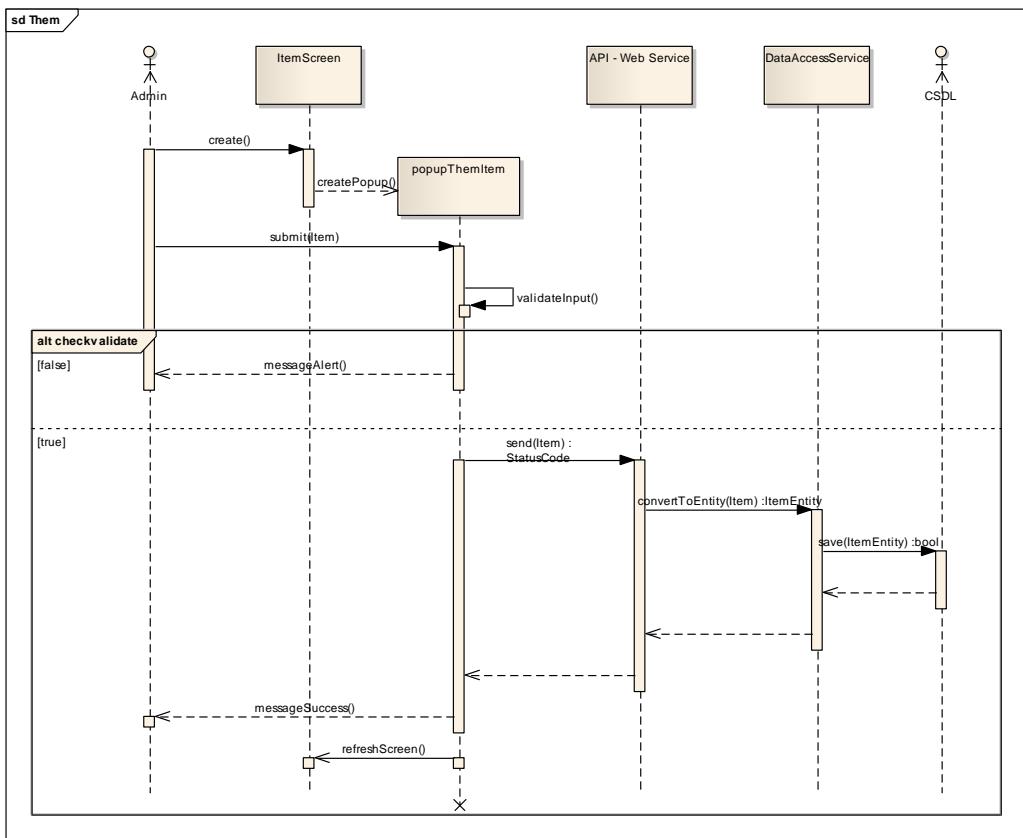
5.1. Vẽ 3 sơ đồ tuần tự theo yêu cầu



Hình 2. 13: chỉnh sửa gói quà



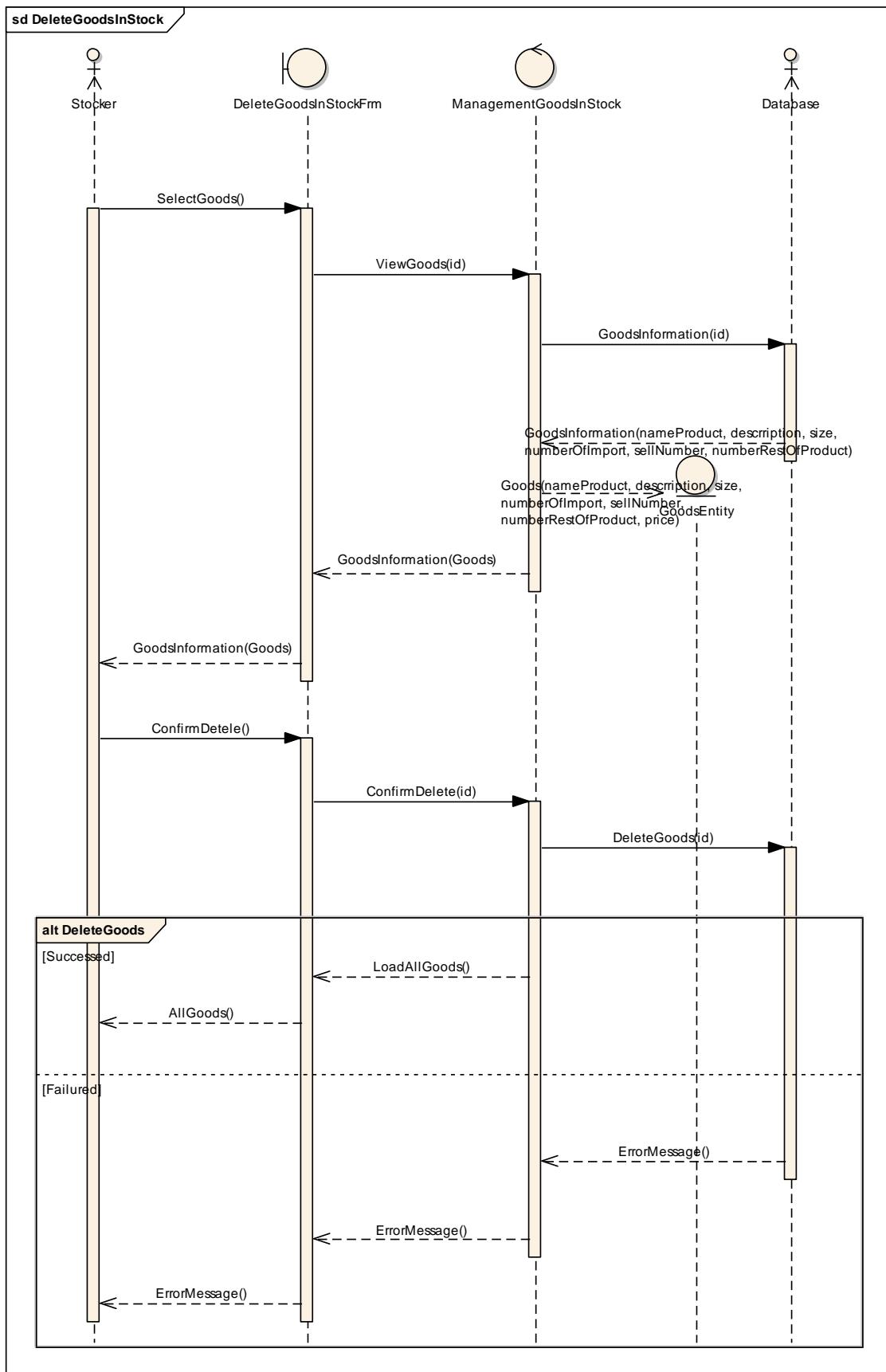
Hình 2. 14: Thêm gói quà



Hình 2. 15: thêm item

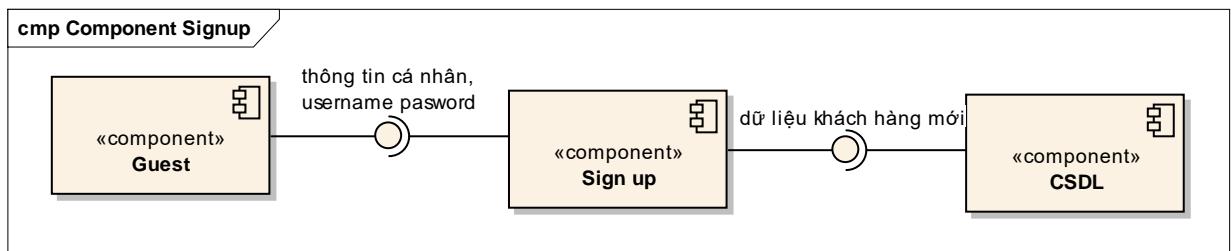
5.2. Sơ đồ tuần tự trước báo cáo

Name	Delete goods in stock
Brief description	Người thủ kho quản lý các công việc xóa một món hàng trong kho.
Actor(s)	Admin
Pre-conditions	Xác định chính xác món hàng cần xóa
Post-conditions	Sản phẩm sẽ được xóa khỏi danh sách sản phẩm
Flow of events	
Basic flow (Thành công)	<p>Người thủ kho thực hiện việc xóa sản phẩm trong kho như sau:</p> <ol style="list-style-type: none"> 1. Chọn sản phẩm cần xóa. 2. Hệ thống lấy thông tin của sản phẩm cần xóa và hiển thị. 3. Người thủ kho thực hiện xóa sản phẩm. 4. Hệ thống hiện bảng yêu cầu xác nhận xóa. 5. Hệ thống xóa khỏi danh sách sản phẩm. 6. Hiển thị lại danh sách mới.
Alternative flow (Thất bại)	<ol style="list-style-type: none"> 1. Nếu xảy ra các trường hợp khác hệ thống sẽ thông báo lỗi.
Extension point	Không có

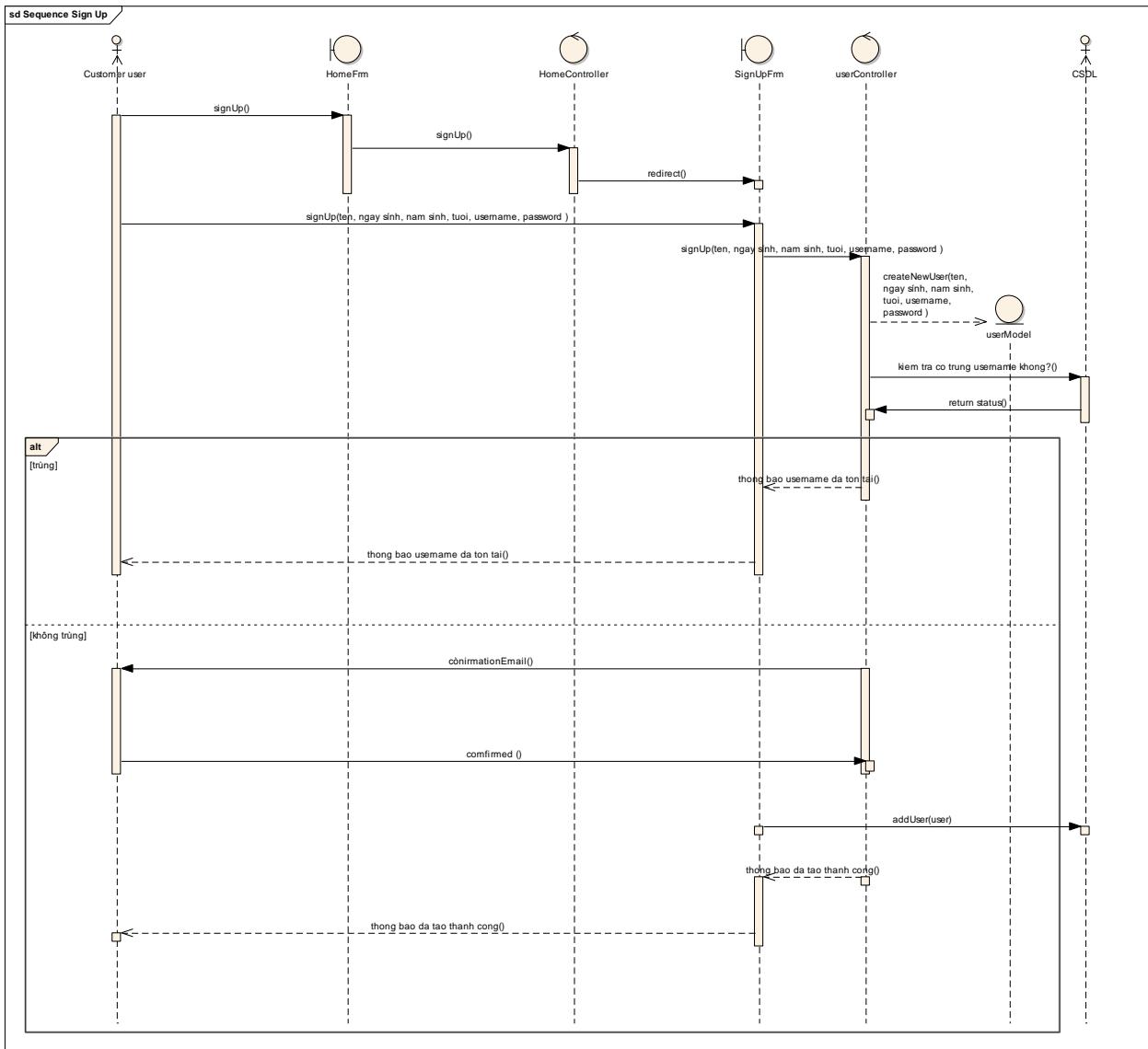


Hình 2. 16: Xóa mặt hàng trong kho

Name	Sign up
Brief description	Đăng ký tài khoản.
Actor(s)	Guest.
Pre-conditions	Vào trang web của nhóm.
Post-conditions	Có tài khoản.
Flow of events	
Basic flow (Thành công)	<ol style="list-style-type: none"> 1. Nhấn vào nút sign up. 2. Nhập các thông tin cần thiết như tên, địa chỉ... 3. Xác nhận và chờ meo xác nhận để tiếp tục. 4. Đăng ký thành công, “mong bạn hài lòng với dịch vụ của chúng tôi”.
Alternative flow (Thất bại)	<p>Thất bại có thể do không xác nhận mail, nhập thiếu thông tin.</p> <ol style="list-style-type: none"> 1. Khi nhập thiếu sẽ hiển thị thông báo ngay chõ thiếu. <p>Nếu trùng username thì thất bại</p> <ol style="list-style-type: none"> 1. Thông báo “username đã tồn tại”
Extension point	Không có

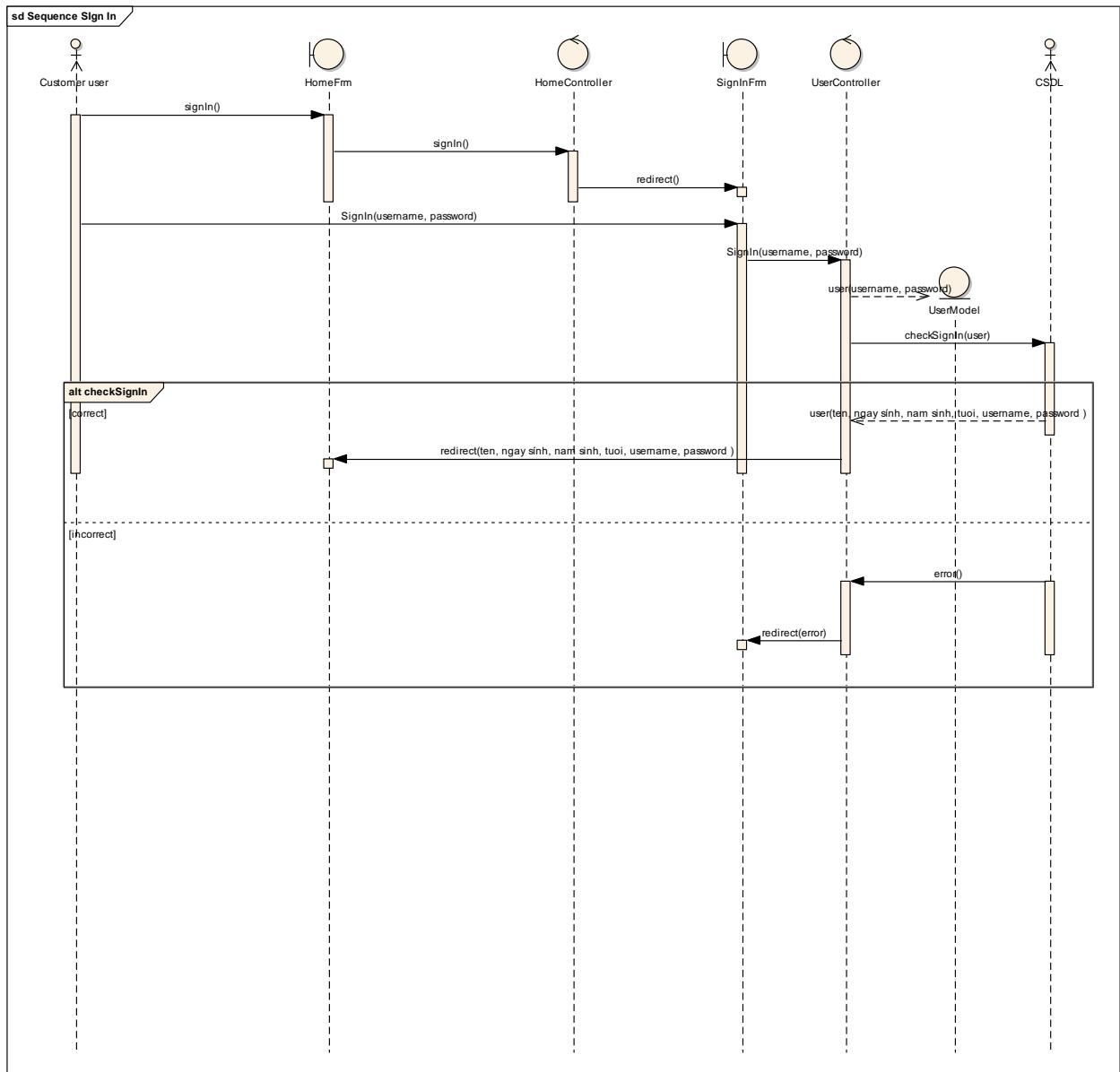


Hình 2. 17: Component diagram đăng ký

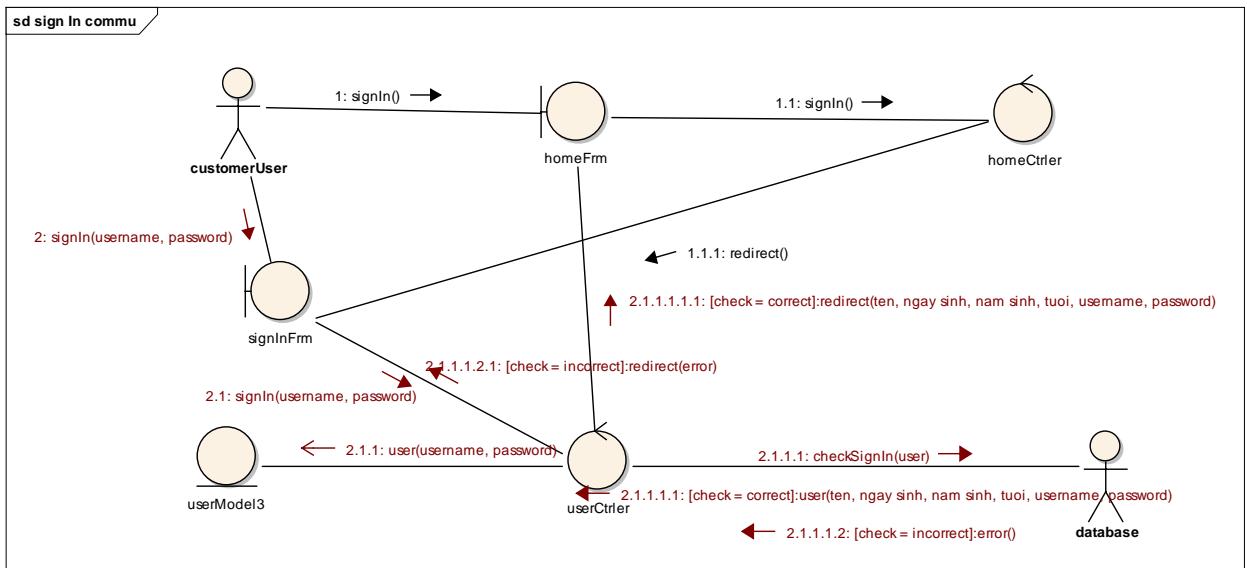


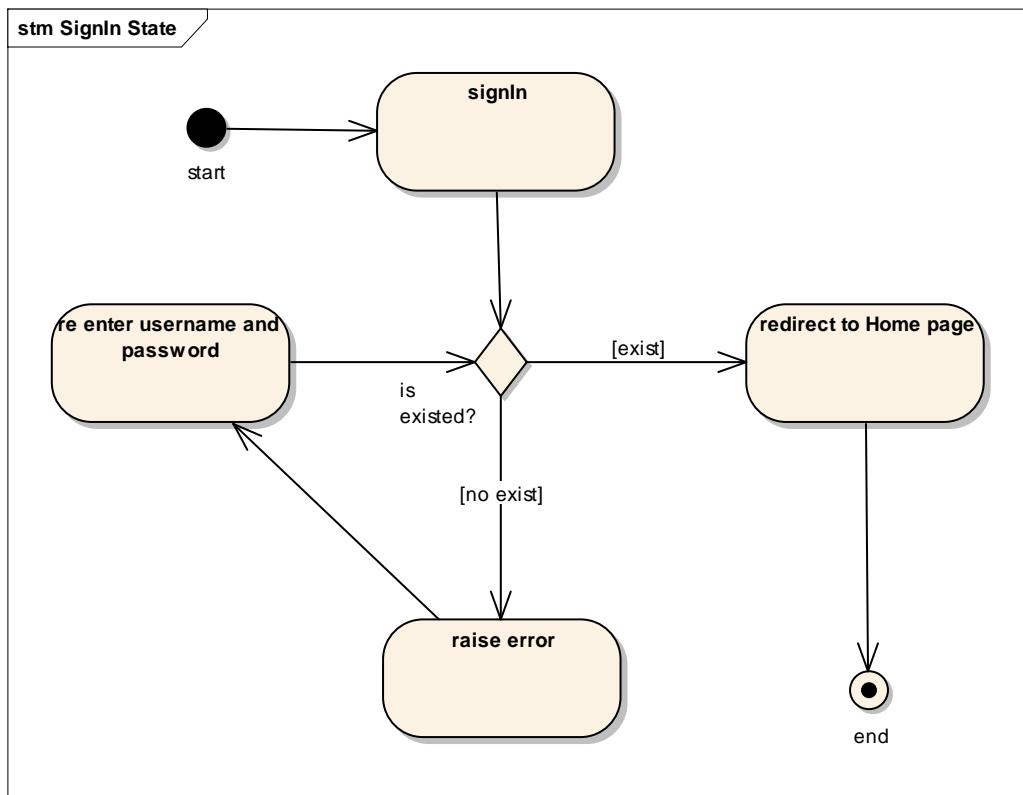
Hình 2. 18: Sign up

Name	Sign in
Brief description	Đăng nhập vào trang để sử dụng nhiều dịch vụ hơn
Actor(s)	Customer user
Pre-conditions	Có tài khoản
Post-conditions	Đăng nhập hoàn tất.
Flow of events	
Basic flow (Thành công)	<p>Đăng nhập theo các bước:</p> <ol style="list-style-type: none"> 1. Người dùng nhập tài khoản mật khẩu vào. 2. Nhấn “đăng nhập”. 3. Chuyển hướng đến trang chủ.
Alternative flow (Thất bại)	<ol style="list-style-type: none"> 1. Thông báo sai tên tài khoản hoặc mật khẩu.
Extension point	Không có

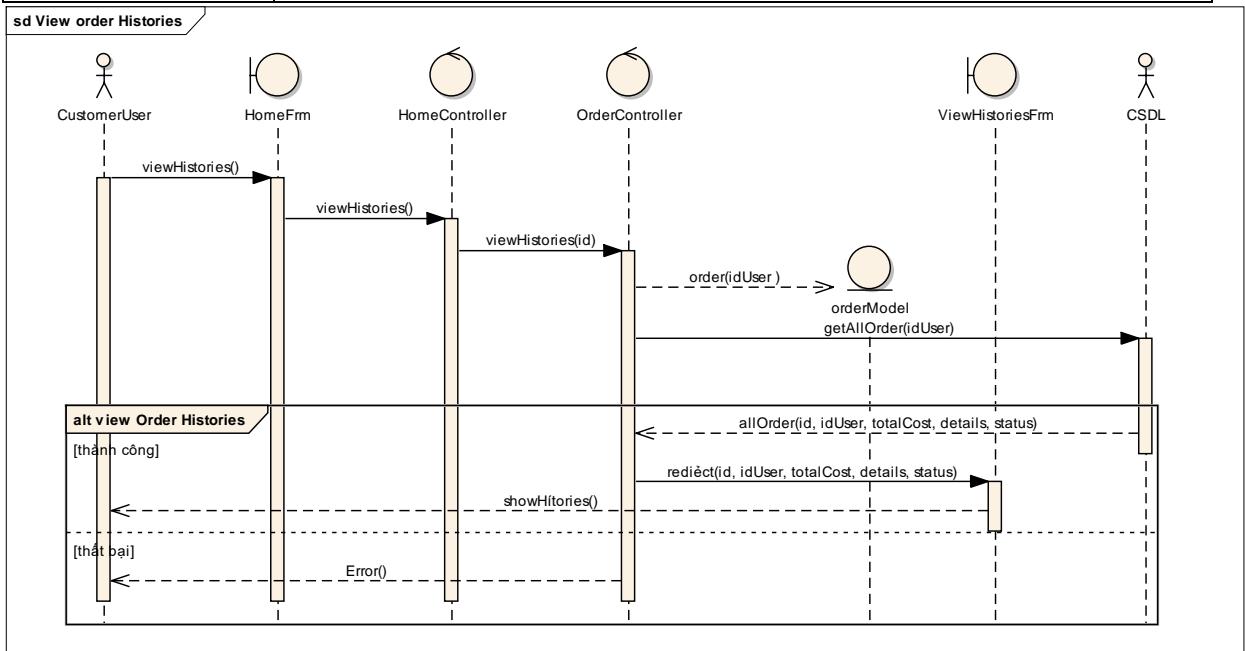


Hình 2. 19: Sign in

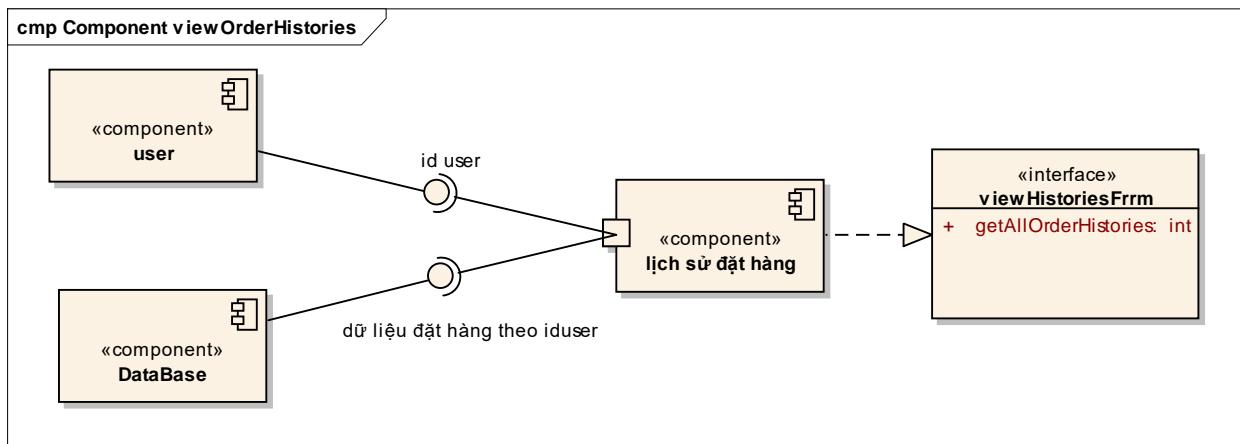




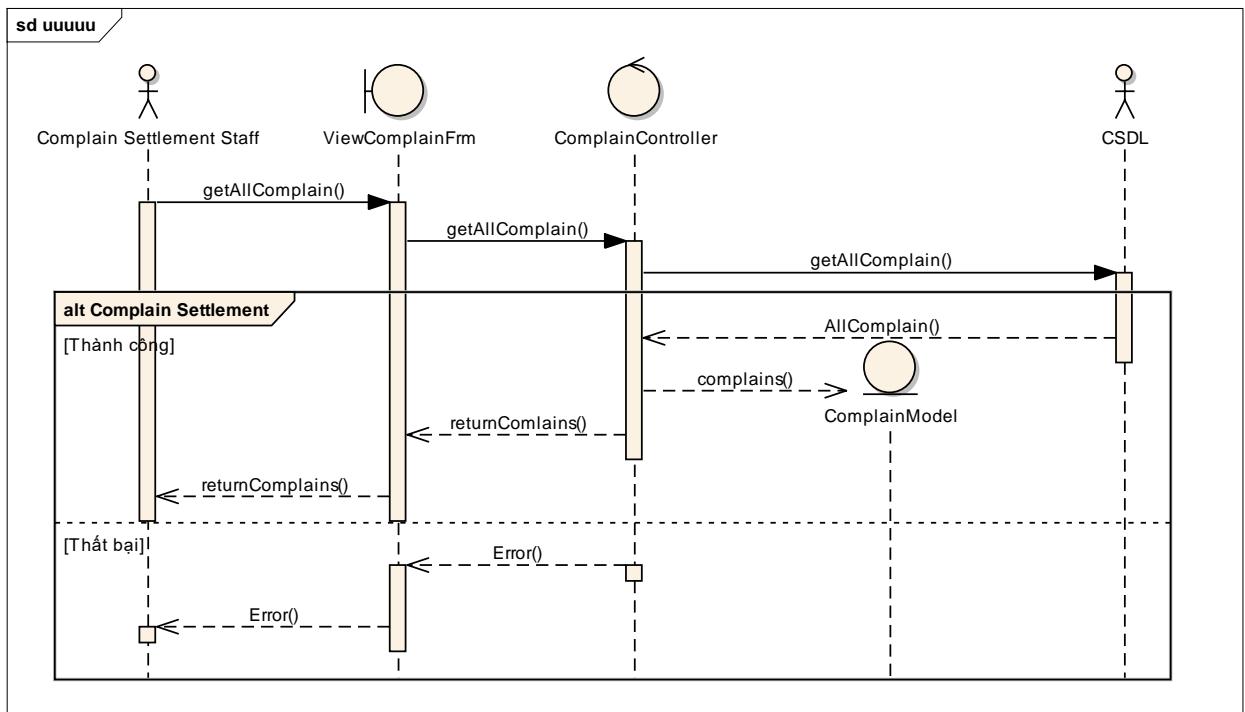
Name	View orders histories
Brief description	Khách xem lại lịch sử đặt hàng của mình.
Actor(s)	Customer user.
Pre-conditions	Đã đăng nhập.
Post-conditions	Hiển ra danh sách sản phẩm đã đặt.
Flow of events	
Basic flow (Thành công)	<ol style="list-style-type: none"> Khách có nhu cầu xem lại trong quá khứ mình đã làm gì với đồng tiền của mình. Vào phần lịch sử để xem. Hiển ra danh sách các sản phẩm đã đặt theo thứ tự từ gần đến xa.
Alternative flow (Thất bại)	<ol style="list-style-type: none"> Thông báo khách “hệ thống đang xảy ra lỗi mong quý khách thông cảm”.
Extension point	Không có



Hình 2. 20: xem lịch sử đơn hàng

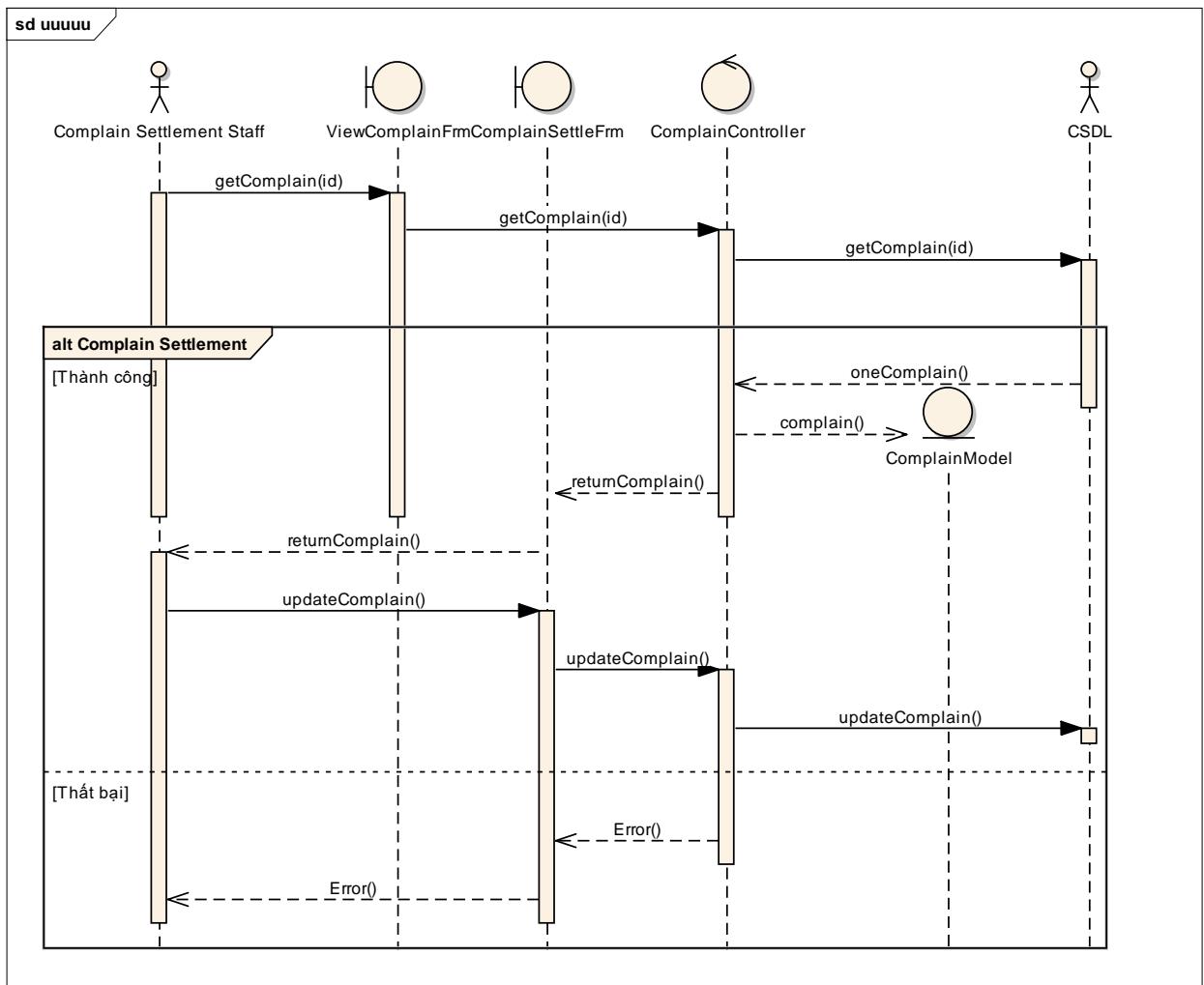


Name	View complaint
Brief description	Nhân viên chăm sóc khách hàng sẽ xem xét các khiếu nại.
Actor(s)	Complaint settlement staff
Pre-conditions	Thông tin phản hồi từ khách hàng
Post-conditions	Hiển thị các thông tin phản hồi từ khách hàng
Flow of events	
Basic flow (Thành công)	<ol style="list-style-type: none"> Hệ thống lấy dữ liệu phản hồi từ khách hàng để hiển thị. Lọc các phản hồi thành các mức độ (1->5). Hiển thị thành danh sách.
Alternative flow (Thất bại)	<ol style="list-style-type: none"> Nếu xảy ra các trường hợp khác hệ thống sẽ thông báo lỗi.
Extension point	Không có



Hình 2. 21: xem phản hồi

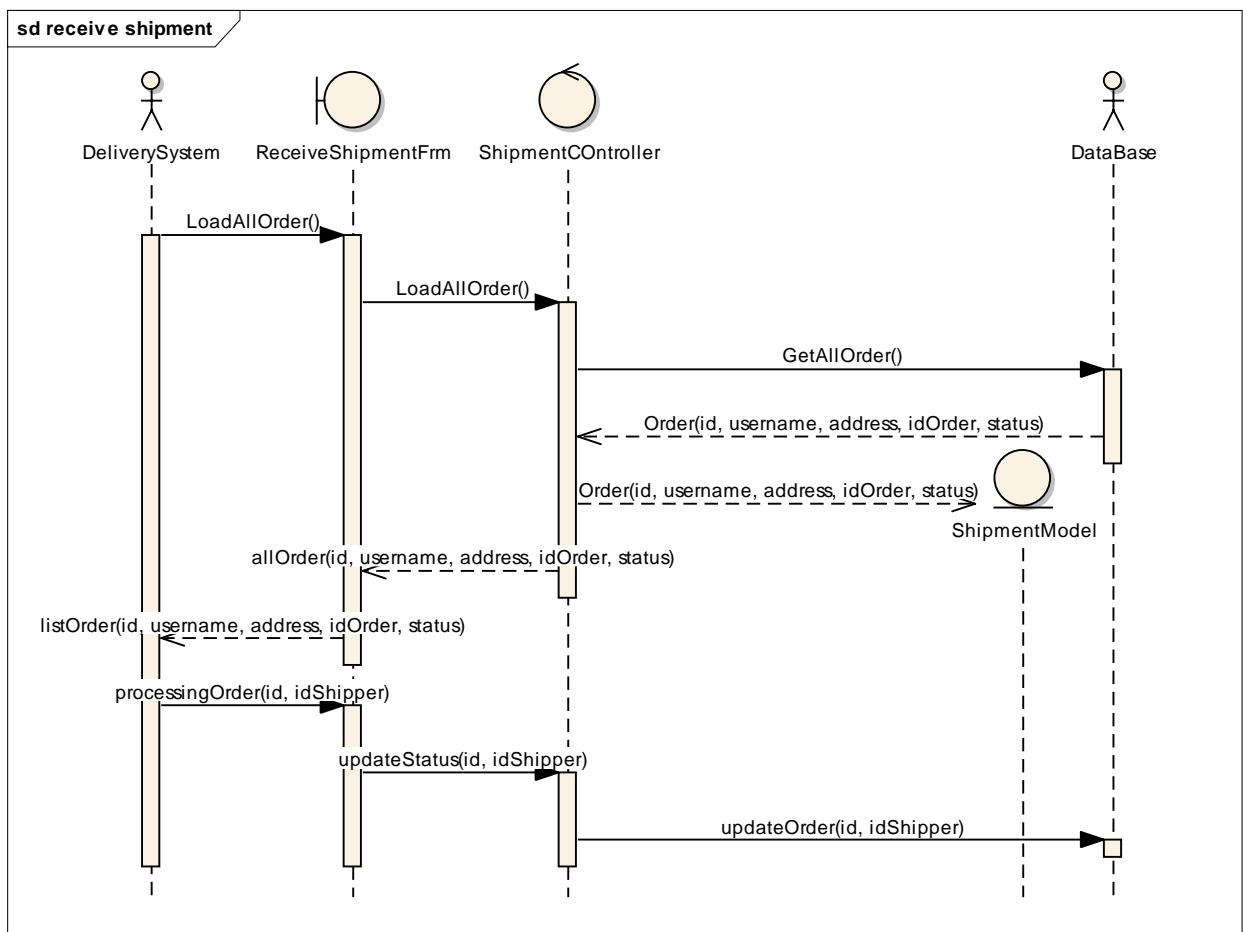
Name	Complain Settlement
Brief description	Nhân viên chăm sóc khách hàng sẽ giải quyết các khiếu nại.
Actor(s)	Complaint settlement staff
Pre-conditions	Thông tin phản hồi từ khách hàng
Post-conditions	Giải quyết phản hồi từ khách hàng
Flow of events	
Basic flow (Thành công)	<ol style="list-style-type: none"> 1. Từ danh sách phản hồi chọn 1 khiếu nại 2. Hệ thống dựa theo id để hiện chi tiết khiếu nại 3. Giải quyết khiếu nại
Alternative flow (Thất bại)	<ol style="list-style-type: none"> 1. Nếu xảy ra các trường hợp khác hệ thống sẽ thông báo lỗi.
Extension point	Không có



Hình 2. 22: Trả lời phản hồi

Name	Receive shipment
Brief description	Người quản lý hệ thống giao hàng phân công nhiệm vụ giao hàng cho người giao hàng
Actor(s)	Delivery system
Pre-conditions	Có người đặt hàng.
Post-conditions	Xử lý hoàn tất việc giao hàng.
Flow of events	
Basic flow (Thành công)	<ol style="list-style-type: none"> Tải lên tất cả đơn hàng để xử lý. Hệ thống sẽ yêu cầu phân phối người giao hàng. Khi đó hệ thống sẽ phân một shiper trong nhóm đang rảnh để giao hàng.

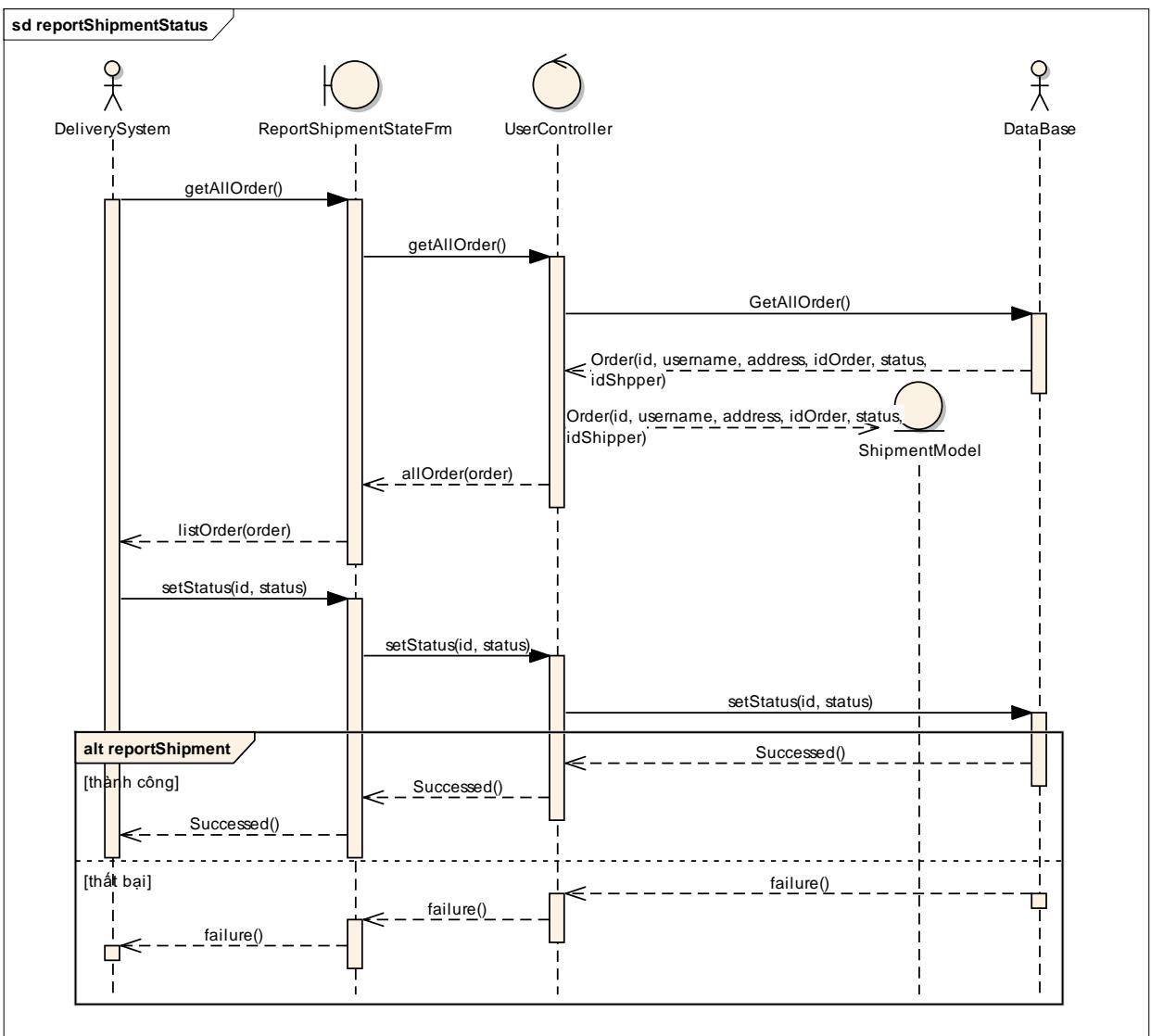
	4. Nếu ai cũng đang giao thì sẽ chờ khi có người rảnh thì sẽ phân phối.
Alternative flow (Thất bại)	Không có.
Extension point	Không có.



Hình 2. 23: xử lý đơn hàng

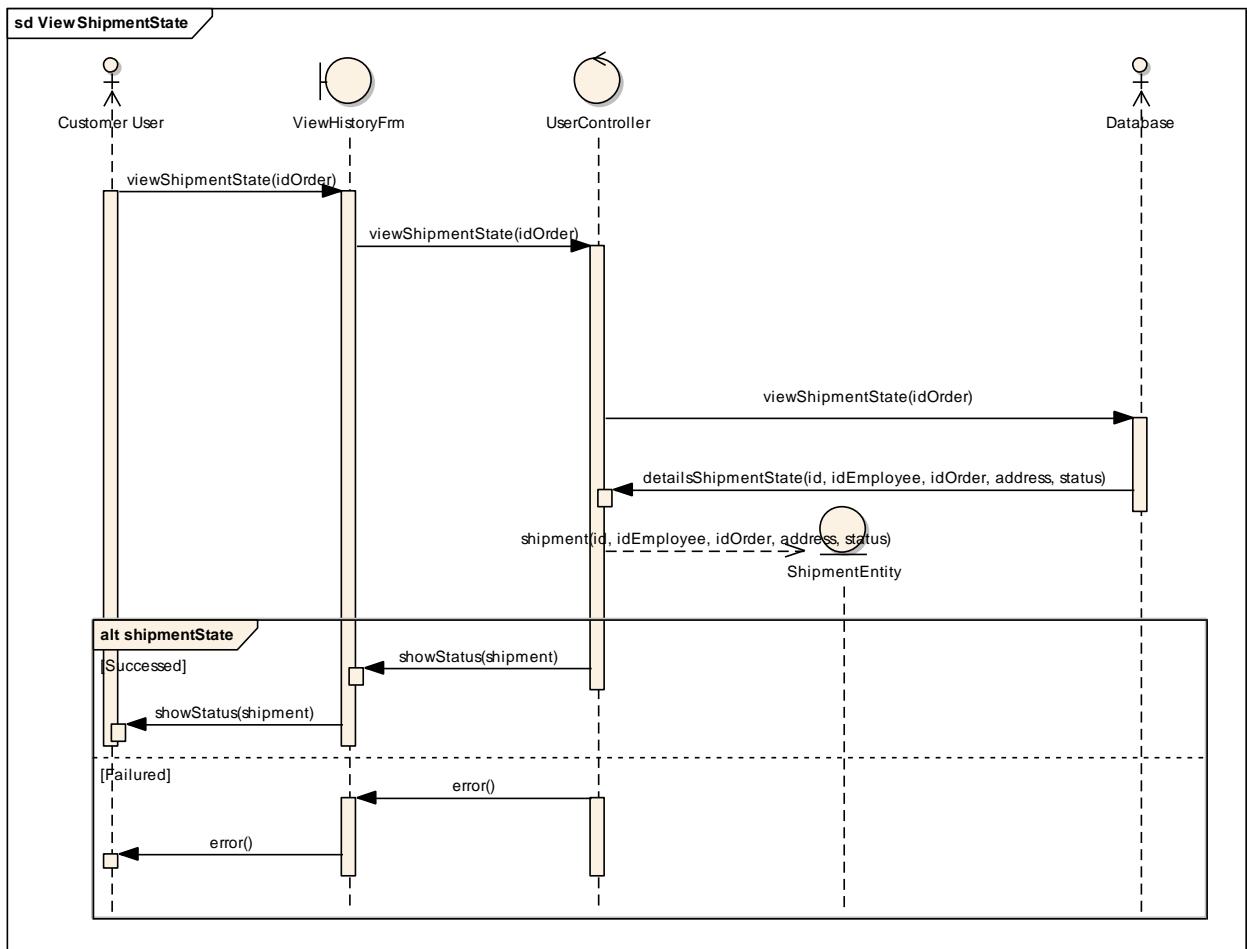
Name	Report shipment state
Brief description	Báo cáo tình trạng đơn hàng.
Actor(s)	Delivery system
Pre-conditions	Shipper nhận hàng
Post-conditions	Báo cáo tình trạng đơn hàng.
Flow of events	
Basic flow (Thành công)	<ol style="list-style-type: none"> 1. Khi khách đặt thì state là “đang chuẩn bị”. 2. Khi shipper vừa nhận đơn hàng thì state là đang giao. 3. Khi shipper đến và giao được cho khách sẽ chụp hình minh chứng là đã giao. 4. Cập nhập trạng thái thành “đã giao”. 5. Cập nhập dữ liệu lên lịch sử hàng của customer.
Alternative flow (Thất bại)	<p>Khi giao hàng mà khách không lấy:</p> <ol style="list-style-type: none"> 1. Cung cấp thông tin và hệ với bộ phận quản lý để giải quyết. <p>Khi trạng thái “đã giao” mà khách chưa nhận được hàng:</p>

	1. Liên hệ với tổng đài chăm sóc khách hàng để được giải quyết.
Extension point	Không có.



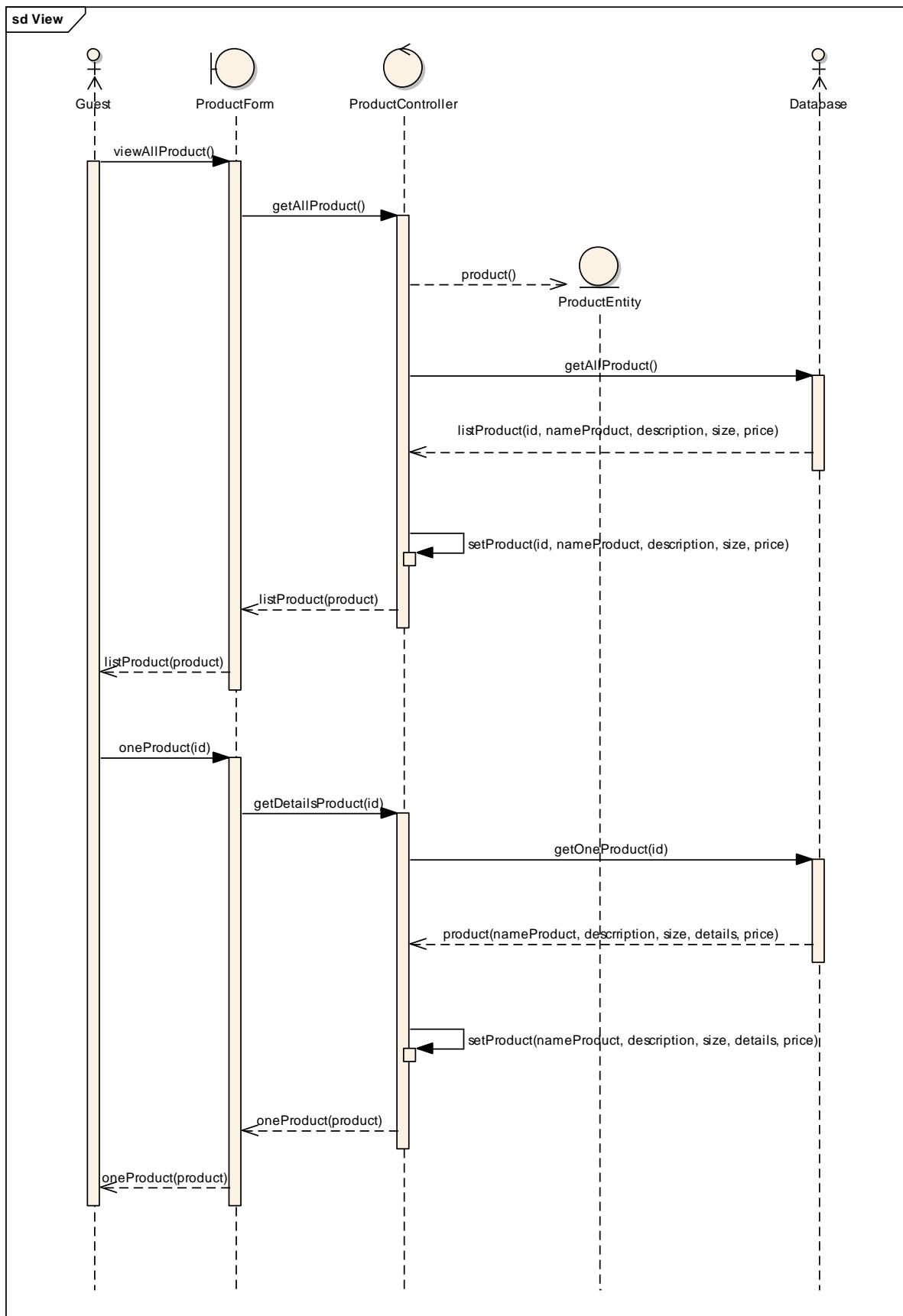
Hình 2. 24: báo cáo tình trạng đơn hàng

Name	View shipment state
Brief description	Khách hàng có thể xem tình trạng quá trình vận chuyển của mình.
Actor(s)	Customer user
Pre-conditions	Có đơn hàng
Post-conditions	Tình trạng vận chuyển đơn hàng.
Flow of events	
Basic flow (Thành công)	<ol style="list-style-type: none"> 1. Khách hàng chọn món hàng muốn xem. 2. Hệ thống truy cập cơ sở dữ liệu lấy chi tiết tình trạng vận chuyển. 3. Trả lại thông tin tình trạng vận chuyển.
Alternative flow (Thất bại)	<ol style="list-style-type: none"> 1. Thông báo lỗi.
Extension point	Không có.



Hình 2. 25: xem trạng thái đơn hàng

Name	View
Brief description	Khách hàng có thể xem sản phẩm cần mua.
Actor(s)	User & customer
Pre-conditions	
Post-conditions	Thông tin sản phẩm
Flow of events	
Basic flow (Thành công)	<ol style="list-style-type: none"> 1. Hệ thống tải tất cả các sản phẩm trong kho. 2. Khách hàng chọn vào một sản phẩm. 3. Hệ thống tìm chi tiết sản phẩm trong kho. 4. Hiện thông tin chi tiết sản phẩm
Alternative flow (Thất bại)	<ol style="list-style-type: none"> 1. Thông báo lỗi.
Extension point	Không có.

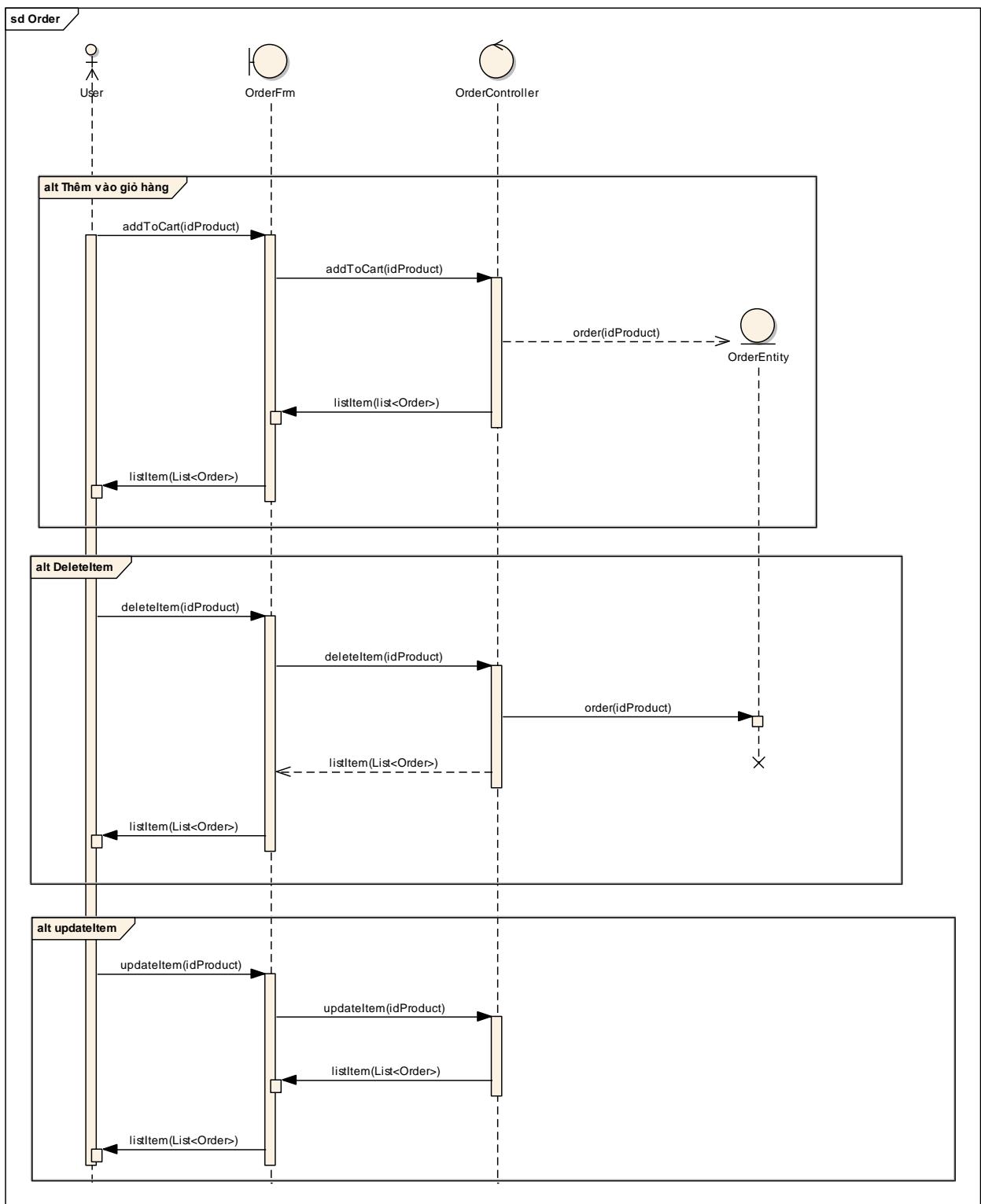


Hình 2. 26: Xem mặt hàng

Name	Order (add)
Brief description	Khách hàng có thể thêm vào giỏ hàng
Actor(s)	Customer
Pre-conditions	Sản phẩm
Post-conditions	Danh sách sản phẩm
Flow of events	
Basic flow (Thành công)	1. Chọn sản phẩm muốn mua. 2. Hệ thống sẽ thêm món hàng vào giỏ hàng. 3. Hệ thống trả về danh sách giỏ hàng.
Alternative flow (Thất bại)	
Extension point	Không có.

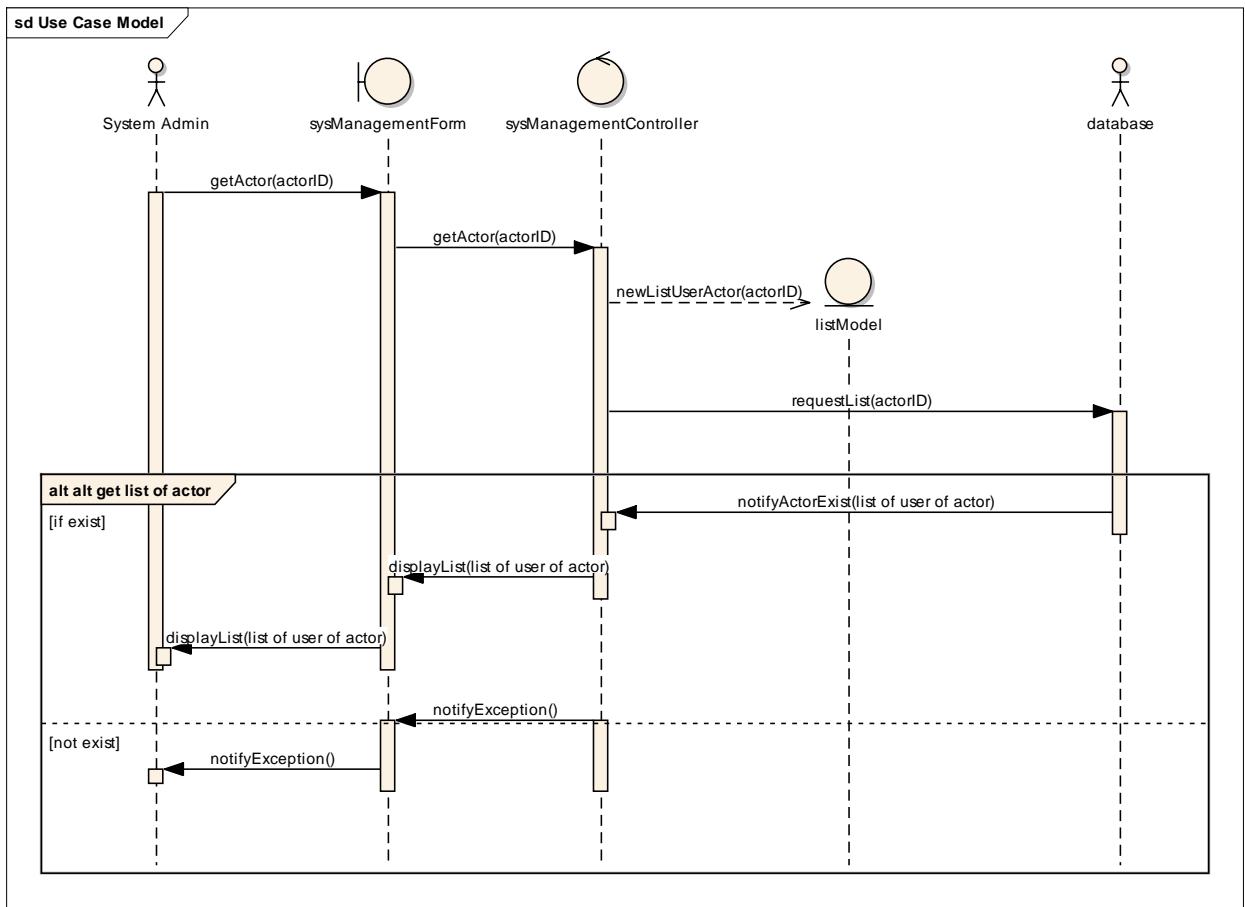
Name	Order (remove)
Brief description	Khách hàng có thể xóa món hàng
Actor(s)	Customer
Pre-conditions	Sản phẩm
Post-conditions	Danh sách sản phẩm
Flow of events	
Basic flow (Thành công)	1. Chọn sản phẩm muốn xóa. 2. Hệ thống sẽ xóa món hàng vào giỏ hàng. 3. Hệ thống trả về danh sách giỏ hàng.
Alternative flow (Thất bại)	
Extension point	Không có.

Name	Order (update)
Brief description	Khách hàng có thể cập nhật món hàng
Actor(s)	Customer
Pre-conditions	Sản phẩm
Post-conditions	Danh sách sản phẩm
Flow of events	
Basic flow (Thành công)	<ol style="list-style-type: none"> 1. Chọn sản phẩm muốn cập nhật. 2. Hệ thống sẽ cập nhật món hàng vào giỏ hàng. 3. Hệ thống trả về danh sách giỏ hàng.
Alternative flow (Thất bại)	
Extension point	Không có.

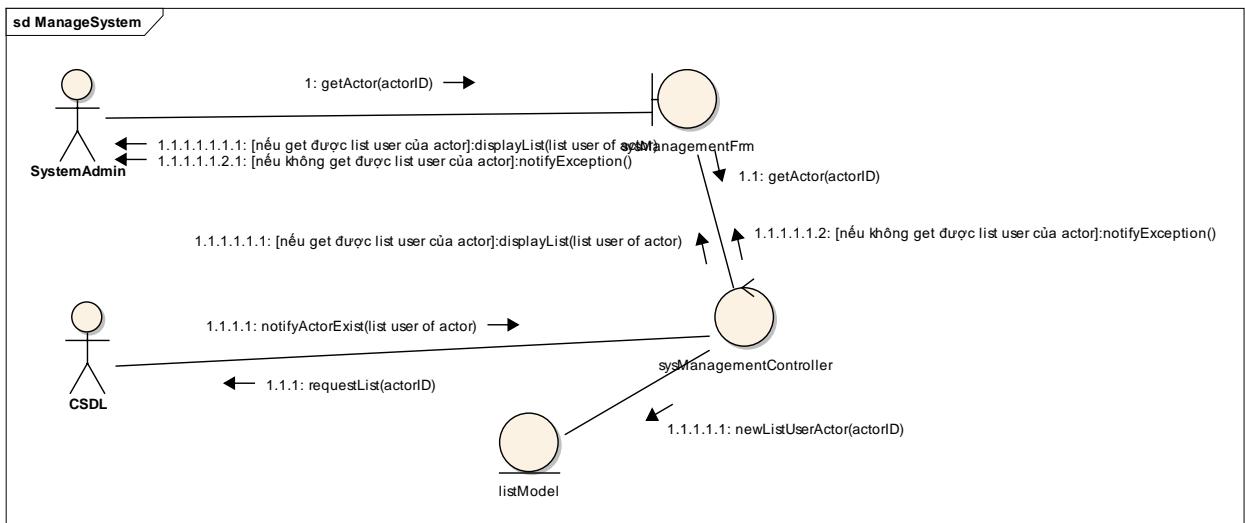


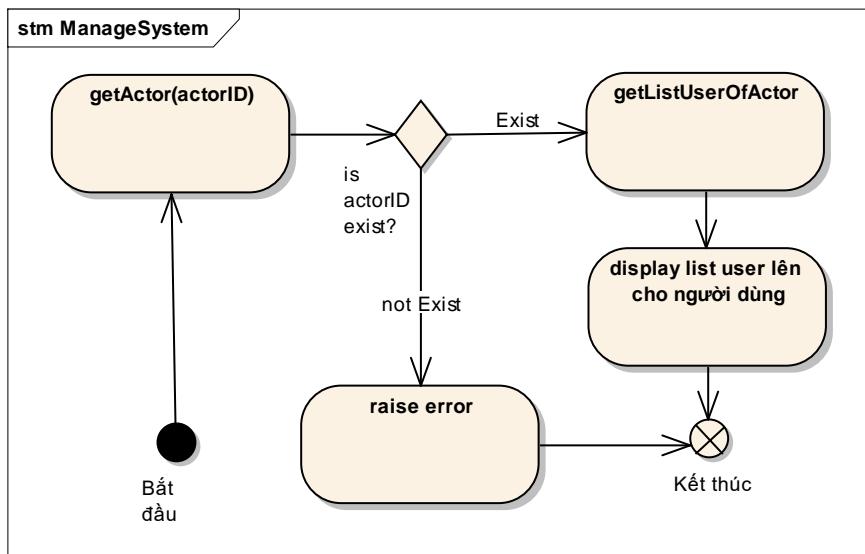
Hình 2. 27: quản lý đặt hàng

Name	Manage System
Brief description	Admin quản lý 4 actor còn lại.
Actor(s)	System Admin
Pre-conditions	Không
Post-conditions	Khi admin dùng chức năng này, chọn 1 trong 4 actor muốn quản lý và hệ thống sẽ gửi về danh sách người dùng ứng với actor đó.
Flow of events	
Basic flow (Thành công)	<p>Admin sử dụng chức năng manage system:</p> <ol style="list-style-type: none"> Chọn 1 trong 4 actor được hiển thị trên màn hình. Hệ thống trả về danh sách người dùng ứng với actor đã chọn.
Alternative flow (Thất bại)	Nếu hệ thống bị lỗi thì xuất ra nguyên nhân bị lỗi đồng lời hiển thị nút quay về trang trước.
Extension point	Không có

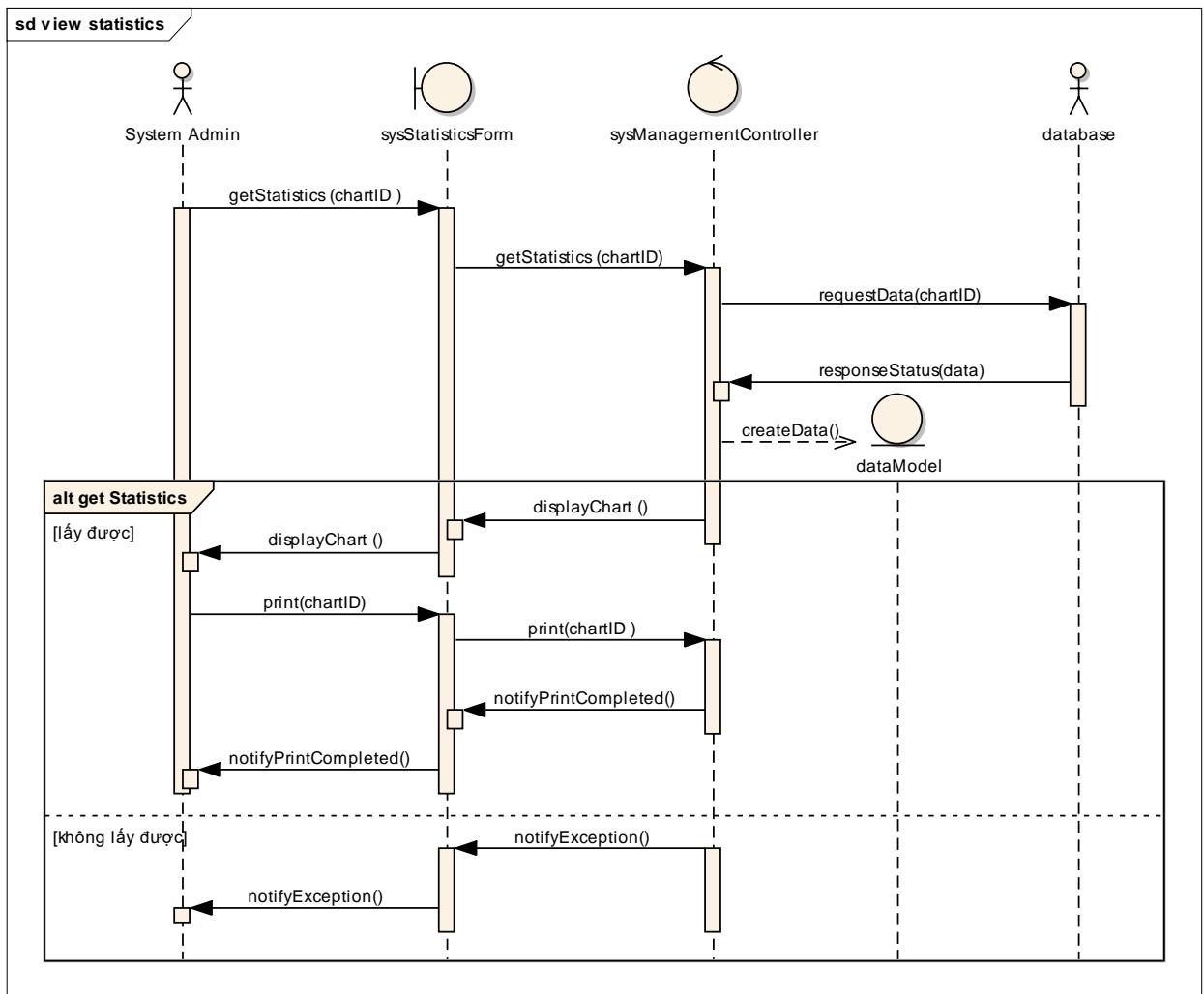


Hình 2. 28: quản lý hệ thống





Name	View system statistics
Brief description	Admin xem đồ thị biểu hiện việc kinh doanh của hệ thống
Actor(s)	System Admin
Pre-conditions	Không
Post-conditions	Khi người thủ kho sử dụng chức năng thống kê thì hệ thống sẽ hiển thị tất cả các thông tin về doanh thu, số lượng hàng bán được ... theo dạng biểu đồ.
Flow of events	
Basic flow (Thành công)	<p>Admin sử dụng chức năng thống kê:</p> <ol style="list-style-type: none"> 1. Hệ thống lấy các thông tin như tên sản phẩm, số lượng bán, số lượng nhập, số lượng còn lại, giá bán, doanh thu của tất cả các mặt hàng. 2. Hiển thị thành 1 bảng và các biểu đồ thống kê. 3. Có thể lọc doanh thu của sản phẩm theo tháng, quý, năm. 4. Xuất ra bảng excel để in ấn hoặc báo cáo khi cần.
Alternative flow (Thất bại)	Nếu hệ thống bị lỗi thì xuất ra nguyên nhân bị lỗi đồng thời hiển thị nút quay về trang trước.
Extension point	Không có



Hình 2. 29: xem thông kê

6. Test case

Các test case cho các phiên bản đã được cập nhật tại github.

Link: <https://github.com/giakhangnguyenlam/WebsiteTangQua/tree/master/TestCase>

7. Unit test

```

File Edit Selection View Go Run Terminal Help
TestGiftpackItemService.java - BackEnd - Visual Studio Code
gitapp > src > test > java > com > swcat > giftapp > DAO > TestGiftpackItemService.java > TestGiftpackItemService > testGetAllGiftpackItems()
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44

@MockBean
private giftpackitemsRepo giftpackitemsRepo;

@InjectMocks
private GiftpackItemsService giftpackItemsService;

@Test
Run Test | Debug Test | ✓
public void testGetAllGiftpackItems(){
    List<giftpackitems> listGiftpack = new ArrayList<>();
    listGiftpack.add(new giftpackitems(1, 1, 1));
    listGiftpack.add(new giftpackitems(2, 2, 2));

    Mockito.when(giftpackitemsRepo.findAll()).thenReturn(listGiftpack);

    List<giftpackitems> giftpackitems = giftpackItemsService.getAllGiftpackItems();

    assertEquals(listGiftpack.get(1).getCID(), giftpackitems.get(1).getCID());
}

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Filter (e.g. text, exclude) Launch Java Tests - 775

18:37:13.548 [main] INFO org.springframework.test.context.support.AnnotationConfigContextLoaderUtils - Could not detect default configuration classes for test class [com.swcat.giftapp.DAO.TestGiftpackItemService]: TestGiftpackItemService does not declare any static, non-private, non-final, nested classes annotated with @Configuration.

18:37:13.883 [main] DEBUG org.springframework.test.context.support.ActiveProfilesUtils - Could not find an annotation declaration for active profiles.

Spaces: 4 UTF-8 CRLF Java Go Live JavaSE-11 6:38 PM 11/06/2021

Hình 2. 30: Unit test

All	Passed	Time
1	0	0.13s

...swcat.giftapp.DAO.TestGiftpackItemService

testGetAllGiftpackItems N/A

Java Test Report - BackEnd - Visual Studio Code

TestGiftpackItemService.java 1, U gitpackitemsRESTest.java 5, U

```

File Edit Selection View Go Run Terminal Help
TestGiftpackItemService.java - BackEnd - Visual Studio Code
gitapp > src > test > java > com > swcat > giftapp > DAO > TestGiftpackItemService.java > TestGiftpackItemService > testGetAllGiftpackItems()
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39

@ExtendWith(SpringExtension.class)
public class TestGiftpackItemService {

    @MockBean
    private giftpackitemsRepo giftpackitemsRepo;

    @InjectMocks
    private GiftpackItemsService giftpackItemsService;

    @Test
    Run Test | Debug Test | ✓
    public void testGetAllGiftpackItems(){
        List<giftpackitems> listGiftpack = new ArrayList<>();
        listGiftpack.add(new giftpackitems(1, 1, 1));
        listGiftpack.add(new giftpackitems(2, 2, 2));

        Mockito.when(giftpackitemsRepo.findAll()).thenReturn(listGiftpack);

        List<giftpackitems> giftpackitems = giftpackItemsService.getAllGiftpackItems();

        assertEquals(listGiftpack.get(1).getCID(), giftpackitems.get(1).getCID());
    }
}


```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Filter (e.g. text, exclude) Launch Java Tests - 775

18:37:13.548 [main] INFO org.springframework.test.context.support.AnnotationConfigContextLoaderUtils - Could not detect default configuration classes for test class [com.swcat.giftapp.DAO.TestGiftpackItemService]: TestGiftpackItemService does not declare any static, non-private, non-final, nested classes annotated with @Configuration.

18:37:13.883 [main] DEBUG org.springframework.test.context.support.ActiveProfilesUtils - Could not find an annotation declaration for active profiles.

Spaces: 4 UTF-8 CRLF Java Go Live JavaSE-11 6:39 PM 11/06/2021

Hình 2. 31: Unit test

III. Chạy thử nghiệm hệ thống

1. Thủ nghiệm các API

1.1. Phiên bản 1.0

===== <API:Account> =====

Mô tả: API có chức năng đăng ký và đăng nhập tài khoản.

Tạo tài khoản: <http://tonydomain.ddns.net:8080/giftapp/api/accounts/signup> (POST)

Yêu cầu: nhập username, password

```
{  
    "username": "catuser",  
    "password": "123meow"  
}
```

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'My Workspace' containing various API collections like 'CNPM / signup', 'APIs', 'Environments', 'Mock Servers', 'Monitors', and 'History'. The main workspace shows a 'POST signup' request under the 'CNPM / signup' collection. The 'Body' tab is selected, showing the JSON payload:

```
{"username": "vietbaicao",  
 "password": "123"}
```

The response panel at the bottom shows a status of 201 Created. The bottom right corner of the window displays system information: Bootcamp, ENG, 11:30 AM, 11/06/2021.

Hình 3. 1: API đăng ký

Đăng nhập dành cho user: <http://tonydomain.ddns.net:8080/giftapp/api> (POST)

Yêu cầu: nhập username, password, accountType

```
{  
    "username": "catuser",  
    "password": "123meow",  
    "accountType": 1  
}
```

accountType

- 1: normal user

- 0: admin user

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'My Workspace' containing collections like 'CNPMP' and 'POST signin'. The main area shows a 'POST' request to 'http://tonydomain.ddns.net:8080/giftapp/api/accounts/signin'. The 'Body' tab is selected, displaying a JSON payload:

```

1   {
2     "username": "vietbaocao",
3     "password": "123",
4     "accountType": 1
5   }

```

Below the body, the response status is 200 OK. At the bottom, there's a taskbar with various icons.

Hình 3. 2: API đăng nhập

1.2. Phiên bản 1.1

===== <API: AccountInfo> =====

Mô tả: API được sử dụng để đăng ký thêm thông tin của khách hàng.

Thêm thông tin khách hàng: <http://tonydomain.ddns.net:8080/giftapp/api/accountinfos> (POST)

Thêm vào các thông tin sau: uname, disname, uaddress, phonenum, email

```
{
  "uname" : "testfull",
  "disname" : "khangtest2",
  "uaddress" : "daylaaddresss",
  "phonenum" : 919910266,
  "email" : "giakhangnguyen115@gmail.com"
}
```

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'My Workspace' containing collections like 'CNPM' which has 'POST signup' and 'POST signin'. Below that are sections for APIs, Environments, Mock Servers, Monitors, and History. The main workspace shows a 'POST / AddAccountInfo' request. The 'Body' tab is selected, showing a JSON payload:

```

1 {
2   "uname": "vietbaocoao",
3   "disname": "khangtest2",
4   "uaddress": "daylaaddress",
5   "phonenum": 919918266,
6   "email": "giakhangnguyen115@gmail.com"
7 }
8

```

Below the body, the response details are shown: Status: 201 Created, Time: 143 ms, Size: 565 B. The response body contains the message: 'Lưu thành công accountInfoService'.

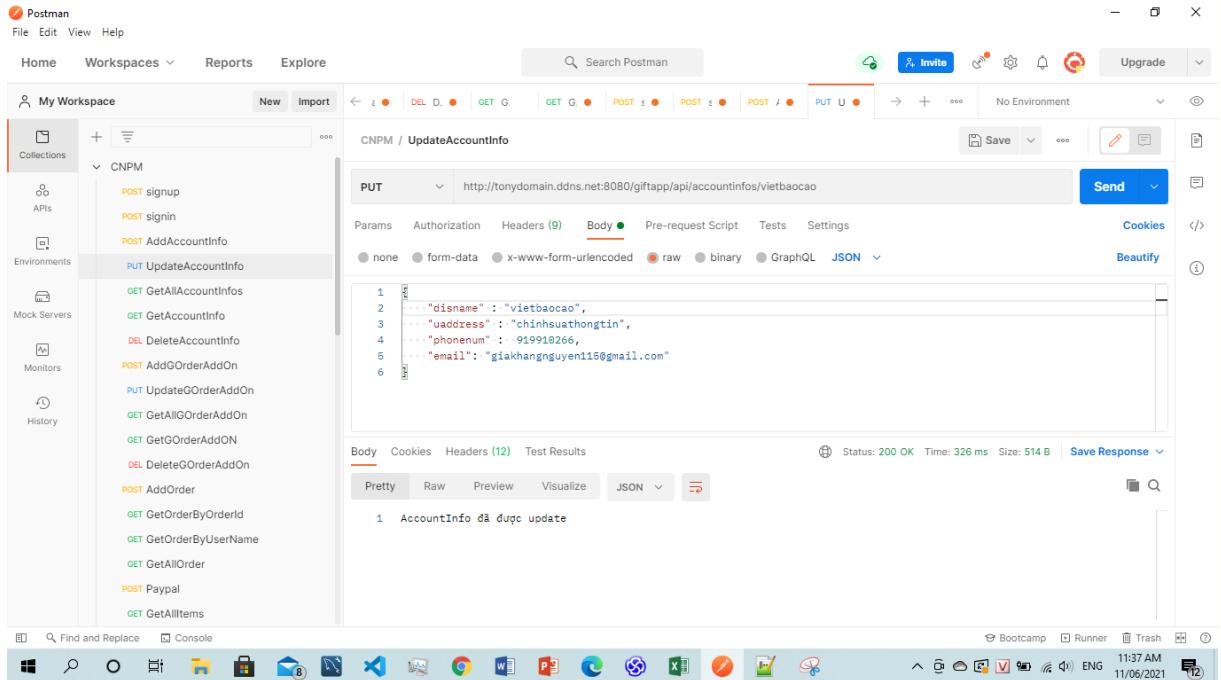
Hình 3. 3: API thêm thông tin khách hàng

Chỉnh sửa thông tin khách hàng:

<http://tonydomain.ddns.net:8080/giftapp/api/accountinfos/{uname}> (PUT)

Chỉnh sửa các thông tin sau: disname, uaddress, phonenum, email

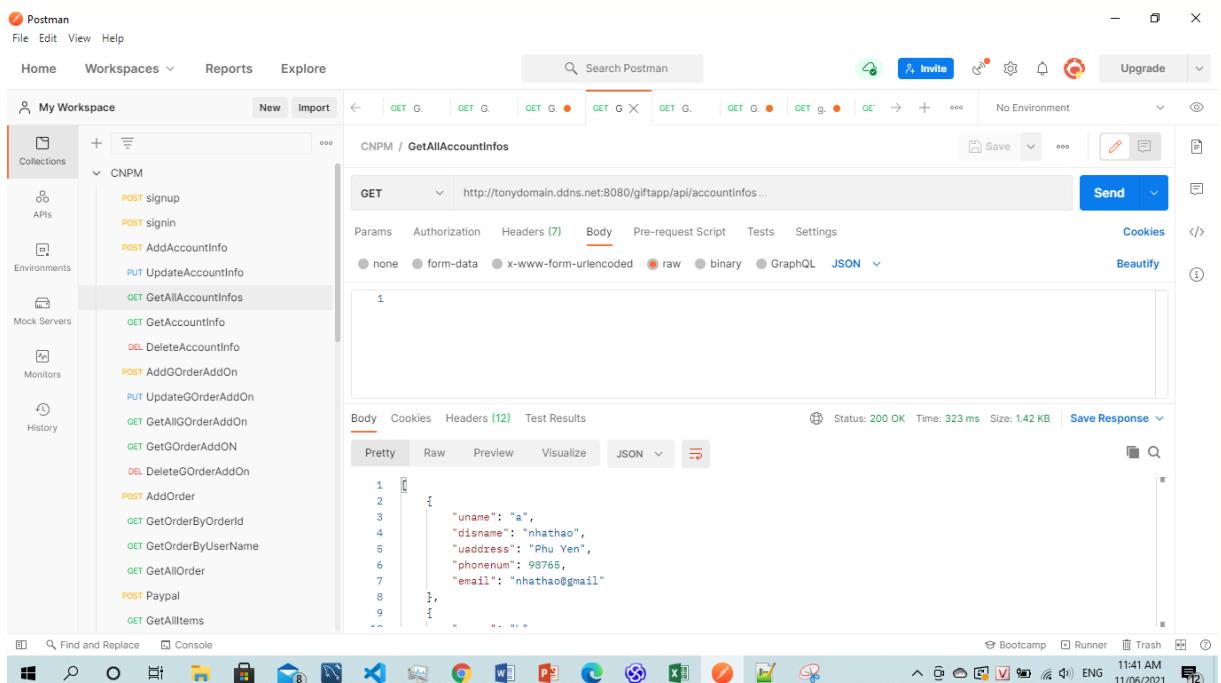
```
{
  "disname": "khangtest3",
  "uaddress": "chinhhsuathongtin",
  "phonenum": 919910266,
  "email": "giakhangnguyen115@gmail.com"
}
```



Hình 3. 4: API chỉnh sửa thông tin khách hàng

Lấy tất cả thông tin khách hàng:

<http://tonydomain.ddns.net:8080/giftapp/api/accountinfos> (GET)



Hình 3. 5: API lấy tất cả thông tin khách hàng

Lấy thông tin một khách hàng qua uname:

<http://tonydomain.ddns.net:8080/giftapp/api/accountinfos/{uname}> (GET)

The screenshot shows the Postman interface with a collection named 'CNPM'. A GET request is made to the endpoint `http://tonydomain.ddns.net:8080/giftapp/api/accountinfos/baobao`. The response status is 200 OK, and the response body is a JSON object:

```

1
2   "uname": "baobao",
3   "disname": "baobao",
4   "uaddress": "123/abc/abc",
5   "phonenum": "919918266",
6   "email": "giakhangnguyen115@gmail.com"
7

```

Hình 3. 6: API lấy thông tin khách hàng qua tên

Xóa tài khoản của khách hàng:

<http://tonydomain.ddns.net:8080/giftapp/api/accountinfos/{uname}> (DELETE)

The screenshot shows the Postman interface with a collection named 'CNPM'. A DELETE request is made to the endpoint `http://tonydomain.ddns.net:8080/giftapp/api/accountinfos/khangtest2`. The response status is 404 Not Found, and the response body is:

```

1 không tìm thấy AccountInfo

```

Hình 3. 7: API xóa thông tin khách hàng

===== <API: GAddOnOrders> =====

Mô tả: Khi đặt một đơn hàng chúng ta có thể quản lý các addOn đi kèm. AddOn là những món quà được đính kèm vào món quà chính. 1 món quà có thể có nhiều addOn.

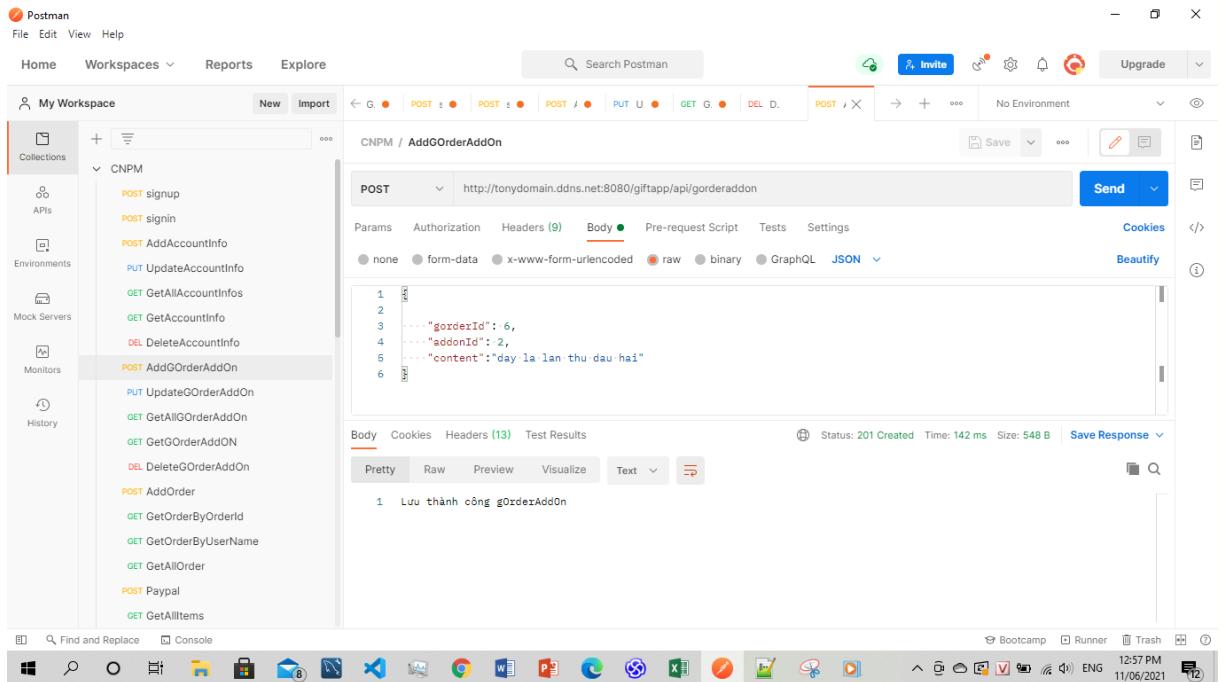
Thêm addOn vào một món quà bất kỳ:

<http://tonydomain.ddns.net:8080/giftapp/api/gorderaddon> (POST)

Note: Chức năng này dành cho người quản lí khi người dùng muốn thêm 1 cái j đó nữa.

Yêu cầu: gorderId, addonId, content

```
{  
    "gorderId": 6,  
    "addonId": 2,  
    "content": "day la lan thu dau hai"  
}
```



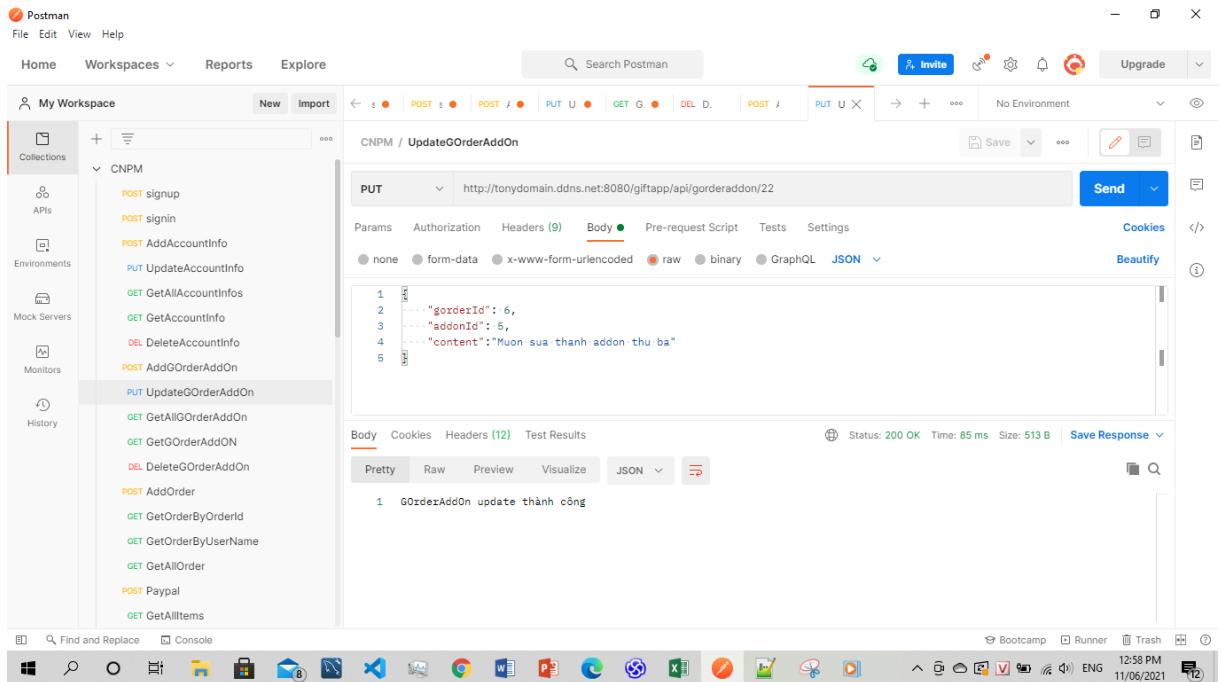
Hình 3. 8: API thêm phụ kiện vào một món quà bất kỳ

Update một addOn nào đó mà người dùng muốn sửa:

<http://tonydomain.ddns.net:8080/giftapp/api/gorderaddon/{cid}> (PUT)

Yêu cầu: gorderId, addonId, content

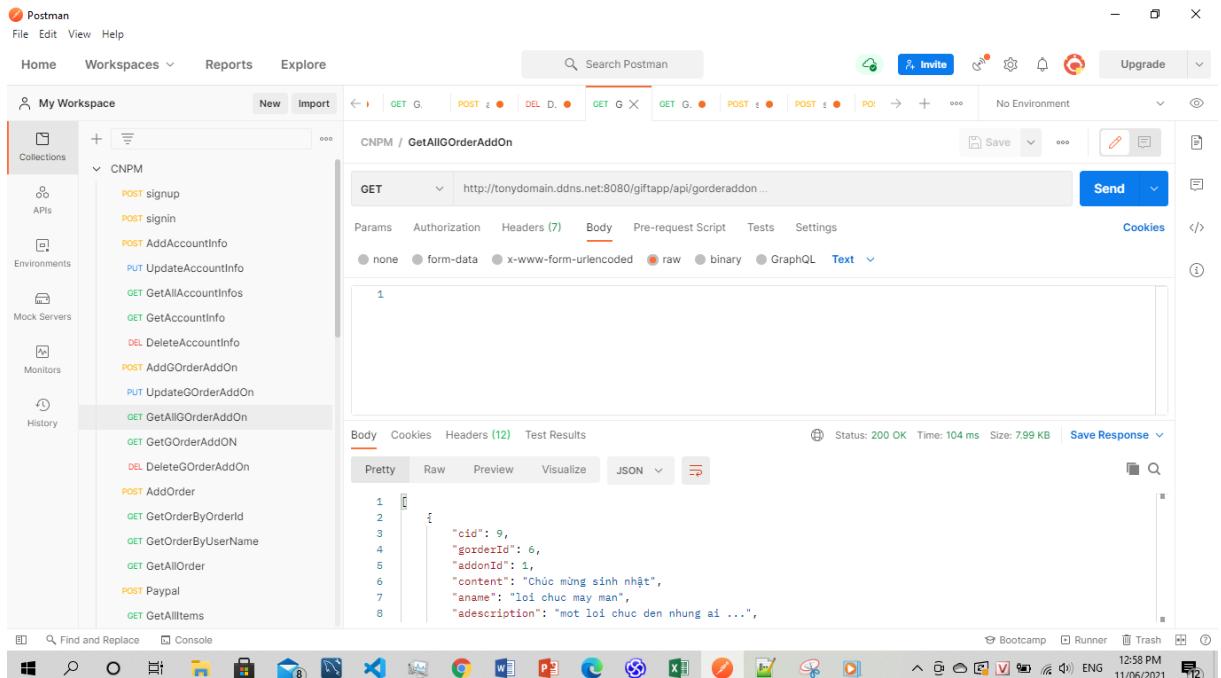
```
{
  "gorderId": 6,
  "addonId": 5,
  "content": "Muon sua thanh addon thu ba"
}
```



Hình 3. 9: API chỉnh sửa một phụ kiện nào đó

Lấy tất cả các goderaddon trong database:

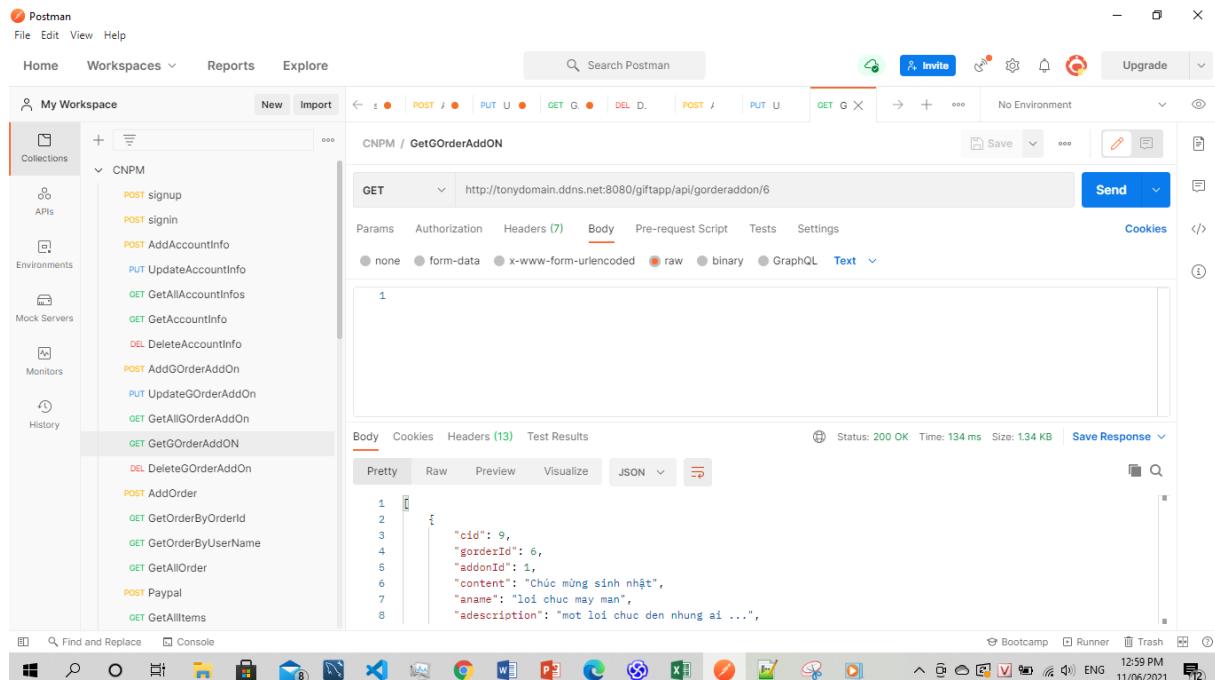
<http://tonydomain.ddns.net:8080/giftapp/api/gorderaddon> (GET)



Hình 3. 10: API Lấy tất cả các đơn hàng kèm phụ kiện

Lấy gorderaddOn theo mã đơn hàng:

<http://tonydomain.ddns.net:8080/giftapp/api/gorderaddon/6> (GET)



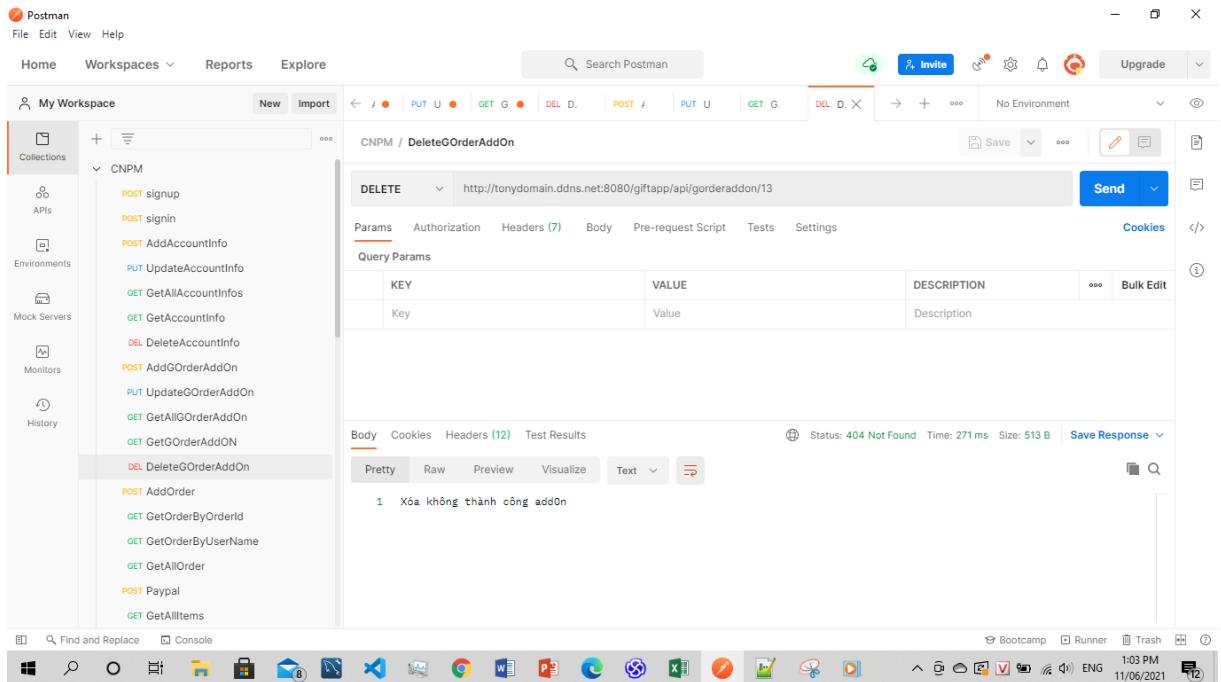
The screenshot shows the Postman application interface. On the left, there's a sidebar with 'My Workspace' containing collections like 'CNPM' and various API endpoints. The main area shows a 'GetGOrderAddON' request with the URL 'http://tonydomain.ddns.net:8080/giftapp/api/gorderaddon/6'. The 'Body' tab is selected, showing the response content. The response status is 200 OK, and the response body is a JSON object:

```
1
2
3   "cid": 9,
4   "gorderID": 6,
5   "addonID": 1,
6   "content": "Chúc mừng sinh nhật",
7   "name": "Lời chúc may mắn",
8   "adescription": "một lời chúc đến những ai ...",
```

Hình 3. 11: API lấy đơn hàng kèm phụ kiện theo mã đơn hàng

Xóa gorderaddOn theo mã gorderaddOn:

<http://tonydomain.ddns.net:8080/giftapp/api/gorderaddon/13> (DELETE)



Hình 3. 12: Xóa đơn hàng kèm phụ kiện theo mã trong cơ sở dữ liệu

1.3. Phiên bản 2.0

===== <API: Order> =====

Mô tả: Sử dụng API để tiến hành đặt một đơn hàng và quản lý nó.

Đặt một đơn hàng: <http://tonydomain.ddns.net:8080/giftapp/api/user/order> (POST)

Yêu cầu: uname, giftpackId, receiveAddress, phoneNumber, dateArrive, greetingCardContent, addonId, content

```
{
  "uname" : "testfull",
  "giftpackId": 1,
  "receiveAddress" : "123/dada/tp.HCM",
  "phoneNumber" : 919910266,
  "dateArrive" : "23-05-2021",
  "greetingCardContent" : "Happy birthday",
  "addonId": [1,2],
}
```

"content" : ["Chúc mừng sinh nhật", "Nhạc sinh nhật vui vẻ"]

}

The screenshot shows the Postman interface. On the left, there's a sidebar with 'My Workspace' containing collections like 'APIs', 'Environments', 'Mock Servers', 'Monitors', and 'History'. The main area shows a 'POST' request to 'http://tonydomain.ddns.net:8080/giftapp/api/user/order'. The 'Body' tab is selected, showing JSON input:

```
3 ... "giftpackId": 1,
4 ... "receiveAddress" : "123/dada/tp.HCM",
5 ... "phoneNumber" : 919919266,
6 ... "dateArrive" : "23-05-2021",
7 ... "greetingCardContent" : "Happy birthday",
8 ... "addOnId": [1, 2, 3, 4],
9 ... "content" : ["Chúc mừng sinh nhật", null, null, null]
10 ...
```

The response status is 201 Created, and the message is 'Đã tạo đơn hàng thành công'.

Hình 3. 13: API đặt quà

Xem lịch sử đơn hàng: <http://tonydomain.ddns.net:8080/giftapp/api/user/order/id/{cid}>
(GET)

Note: cid chính là mã đơn hàng gửi kèm trong thư

The screenshot shows the Postman interface with the following details:

- Collection:** CNPM
- Request:** GET /user/order/id/69
- Body:** JSON (Pretty)
- Response:**

```

1
2   "cid": 69,
3   "uname": "vietbaicao",
4   "giftpackId": 1,
5   "receiveAddress": "123/dada/tp.HCM",
6   "phoneNumber": 919910266,
7   "dateOrder": "11-06-2021",
8   "dateArrive": "23-05-2021",

```

Hình 3. 14: API xem lịch sử đơn hàng qua mã đơn hàng

Xem toàn bộ lịch sử đặt hàng:

<http://tonydomain.ddns.net:8080/giftapp/api/user/order/uname/{uname}> (GET)

The screenshot shows the Postman interface with the following details:

- Collection:** CNPM
- Request:** GET /user/order/uname/baicao
- Body:** Text (Pretty)
- Response:**

```

1
2   {
3     "cid": 68,
4     "uname": "baicao",
5     "giftpackId": 1,
6     "receiveAddress": "123/abc/abc",
7     "phoneNumber": 919910277,
8     "dateOrder": "09-06-2021",

```

Hình 3. 15: API xem lịch sử đơn hàng qua tên tài khoản

Lấy tất cả đơn hàng của khách hàng:

<http://tonydomain.ddns.net:8080/giftapp/api/user/order> (GET)

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'My Workspace' containing collections like 'CNPM' which has various API endpoints listed. The main area shows a 'GET' request to 'http://tonydomain.ddns.net:8080/giftapp/api/user/order'. The 'Body' tab is selected, showing a JSON response with the following content:

```
1
2
3
4
5
6
7
8
{
  "cid": 6,
  "uname": "testfull",
  "giftackId": 1,
  "receiveAddress": "123/dada/tp.HCM",
  "phoneNumber": 919910266,
  "dateOrder": "07-06-2021",
}
```

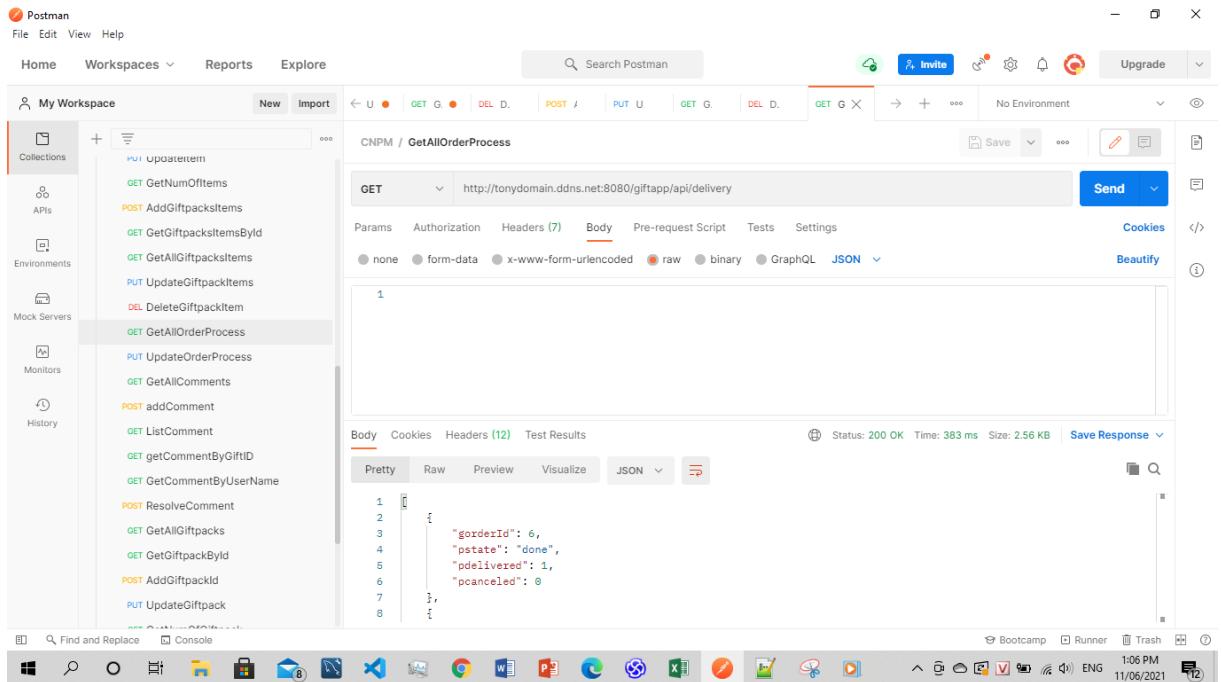
The status bar at the bottom indicates 'Status: 200 OK'.

Hình 3. 16: API lấy tất cả đơn hàng của khách hàng

===== <API: OrderProcess> =====

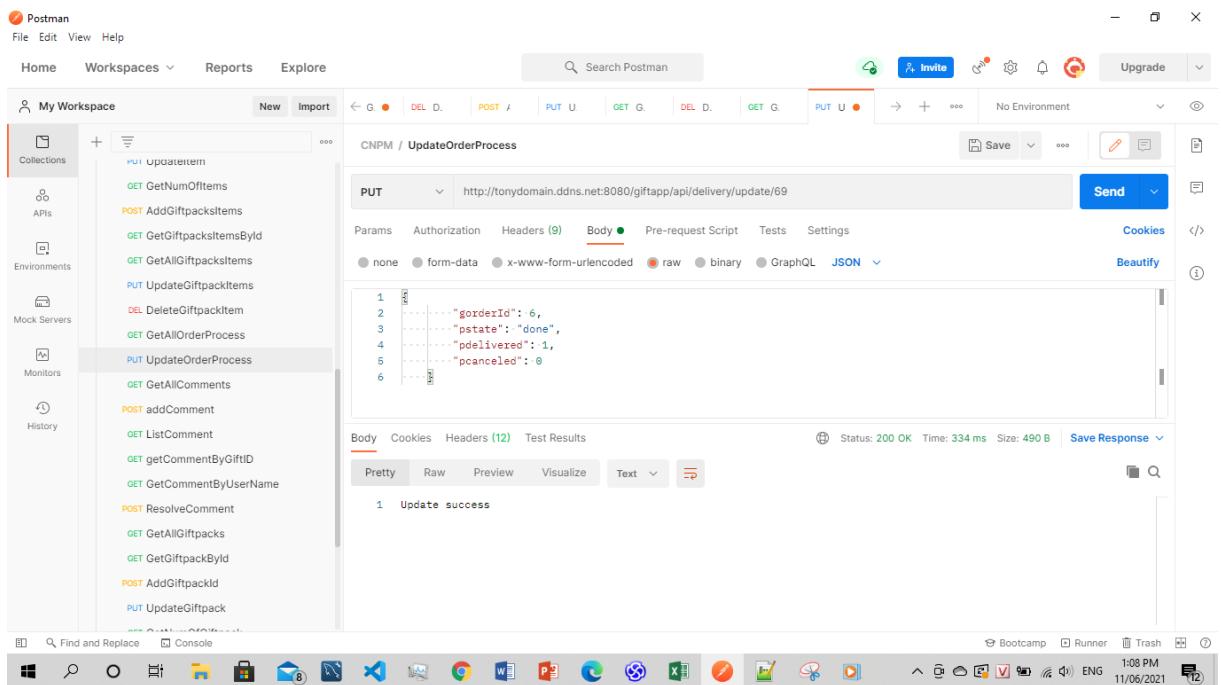
Mô tả: API giúp chúng ta xử lý, vận chuyển tất cả các đơn hàng đã đặt

Lấy tất cả đơn hàng để xử lý: <http://tonydomain.ddns.net:8080/giftapp/api/delivery> (GET)



Hình 3. 17: API lấy tất cả đơn hàng để xử lý

Cập nhật tình trạng một đơn hàng: <http://localhost:8080/api/delivery/update/5> (PUT)



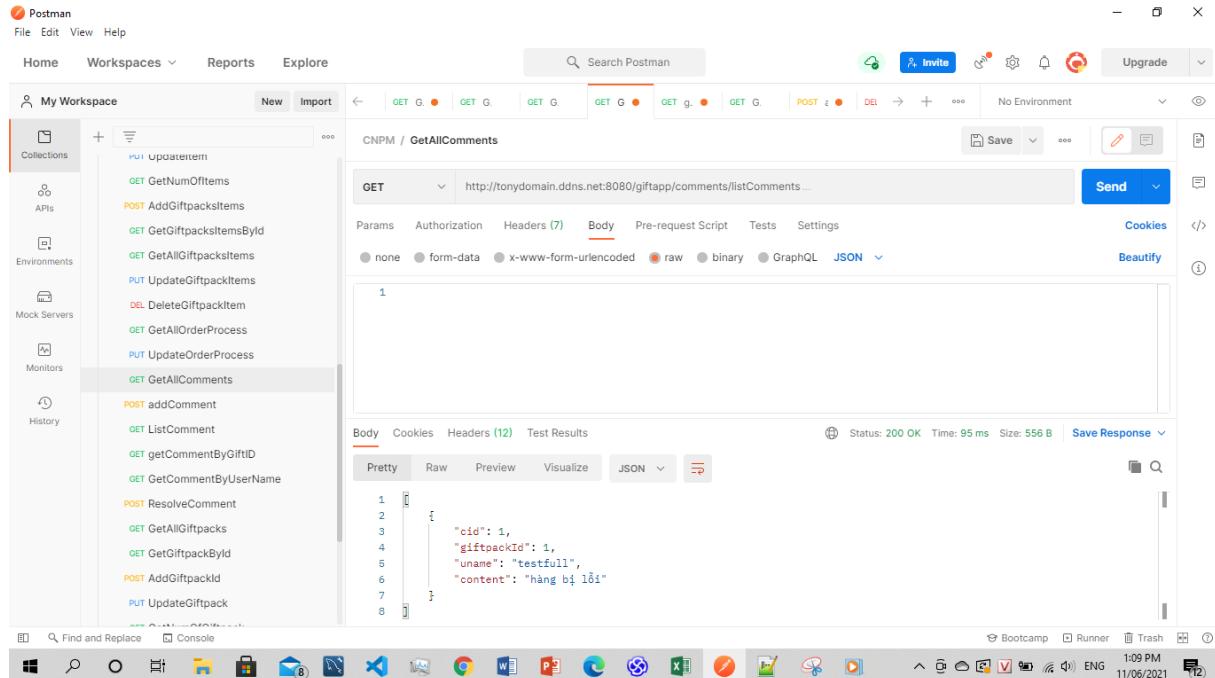
Hình 3. 18: API cập nhật xử lý đơn hàng

===== <API: GiftPackComments> =====

Mô tả: API sử dụng để ghi comment và giải quyết các comments

Lấy tất cả các comments:

<http://tonydomain.ddns.net:8080/giftapp/api/comments/listComments> (GET)



The screenshot shows the Postman application interface. On the left, there's a sidebar with 'My Workspace' containing various API collections and endpoints. The main area shows a 'GET / GetAllComments' request. The 'Body' tab of the request section is selected, showing the URL 'http://tonydomain.ddns.net:8080/giftapp/comments/listComments ...'. The response body is displayed in a JSON viewer, showing a single comment object:

```
1
2
3
4
5
6
7
8
1: [
  {
    "cid": 1,
    "giftpackId": 1,
    "uname": "testfull",
    "content": "hàng bị lỗi"
  }
]
```

Hình 3. 19: API lấy tất cả các phản hồi

Lấy comments theo tên người dùng: <http://tonydomain.ddns.net:8080/giftapp/api/comments/getCommentsByUname/{uname}> (GET)

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'My Workspace' containing various API endpoints like 'GetCommentByUserName'. The main area shows a GET request to 'http://tonydomain.ddns.net:8080/giftapp/comments/getCommentsByUserName/testfull'. The response status is 200 OK, and the JSON body is displayed as:

```

1  {
2   "cid": 1,
3   "giftpackId": 1,
4   "uname": "testfull",
5   "content": "hàng bị lỗi"
6 }

```

Hình 3. 20: API lấy phản hồi theo tên tài khoản

Lấy comments theo mã gói quà: <http://tonydomain.ddns.net:8080/giftapp/api/comments/getCommentsByGiftID/{id}>

Note: id chính là mã gói quà

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'My Workspace' containing various API endpoints like 'getCommentByGiftID'. The main area shows a GET request to 'http://tonydomain.ddns.net:8080/giftapp/comments/getCommentsByGiftID/1...'. The response status is 200 OK, and the JSON body is displayed as:

```

1  {
2   "cid": 1,
3   "giftpackId": 1,
4   "uname": "testfull",
5   "content": "hàng bị lỗi"
6 }

```

Hình 3. 21: Lấy phản hồi theo mã gói quà

Giải quyết các khiếu nại:

<http://tonydomain.ddns.net:8080/giftapp/comments/resolveComment> (POST)

Yêu cầu: email, content, commentResolve, giftpackId

```
{  
    "email": "giakhangnguyen115@gmail.com",  
    "content": "Gift very good",  
    "commentResolve": "Thanks you for buy Gift",  
    "giftpackId": 1  
}
```

The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar lists various API endpoints under the 'CNPM / ResolveComment' collection. The 'ResolveComment' endpoint is selected and highlighted in orange. In the main workspace, a POST request is configured with the URL <http://tonydomain.ddns.net:8080/giftapp/comments/resolveComment>. The 'Body' tab is selected, showing the JSON payload from the previous code block. The 'Send' button is visible at the top right of the request configuration area.

Hình 3. 22: API phản hồi cho khách hàng

Thêm phản hồi: <http://tonydomain.ddns.net:8080/giftapp/comments/addComment> (POST)

Yêu cầu: giftpackId, uname, content

```
{  
    "giftpackId": 10,  
    "uname": "testfull",
```

"content": "hàng bị lỗi"

}

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'My Workspace' containing various API endpoints like 'GetNumOfItems', 'AddGiftpacksItems', etc. The main area shows a POST request to 'http://tonydomain.ddns.net:8080/giftapp/comments/addComment'. The 'Body' tab is selected, showing the JSON input: { "giftpackId": 10, "uname": "testfull", "content": "hàng bị lỗi" }. Below the request, the response status is 200 OK with a success message: 'Add Comment sucess'. The bottom status bar shows the time as 11:12 PM and date as 11/06/2021.

Hình 3. 23: API thêm phản hồi

1.4. Phiên bản 3.0

===== <API: Items> =====

Mô tả: Quản lý tất cả các mặt hàng trong kho.

Thêm một mặt hàng mới: <http://tonydomain.ddns.net:8080/giftapp/api/items> (POST)

Yêu cầu: itemName, itemDescription

{

```
"itemName": "smart phone",
"itemDescription": "mot cai dien thoai",
"thumbnailImg": "https://www.apple.com/ac/structured-
data/images/open_graph_logo.png?201810040211",
"price": 750,
"activeState": 1
```

{}

Hình 3. 24: Thêm một mặt hàng mới

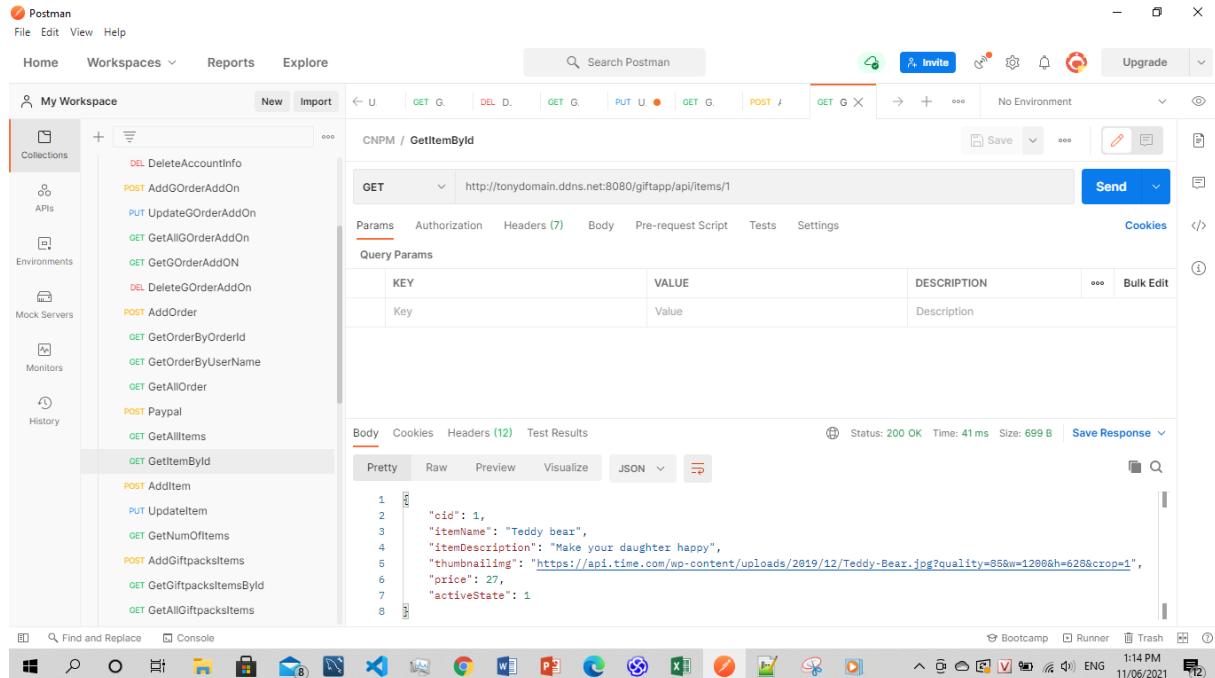
Lấy tất cả các mặt hàng có sẵn trong kho:

<http://tonydomain.ddns.net:8080/giftapp/api/items> (GET)

Hình 3. 25: Lấy tất cả các mặt hàng có sẵn trong kho

Lấy mặt hàng theo mã hàng: <http://tonydomain.ddns.net:8080/giftapp/api/items/{cid}>
(GET)

Note: cid chính là mã hàng hóa.



The screenshot shows the Postman application interface. On the left, there's a sidebar with 'My Workspace' containing various API endpoints. The main area shows a 'CNPMP / GetItemByld' collection with a single 'GET' request to 'http://tonydomain.ddns.net:8080/giftapp/api/items/1'. The 'Params' tab is selected, showing a 'Query Params' table with a single row: 'Key' (Value) and 'Value' (Description). Below the table, the 'Body' tab is selected, showing the JSON response:

```
1  {
2    "cid": 1,
3    "itemName": "Teddy bear",
4    "itemDescription": "Make your daughter happy",
5    "thumbnailimg": "https://api.time.com/wp-content/uploads/2019/12/Teddy-Bear.jpg?quality=85&w=1200&h=628&crop=1",
6    "price": 27,
7    "activeState": 1
8  }
```

Hình 3. 26: Lấy thông tin mặt hàng qua mã mặt hàng

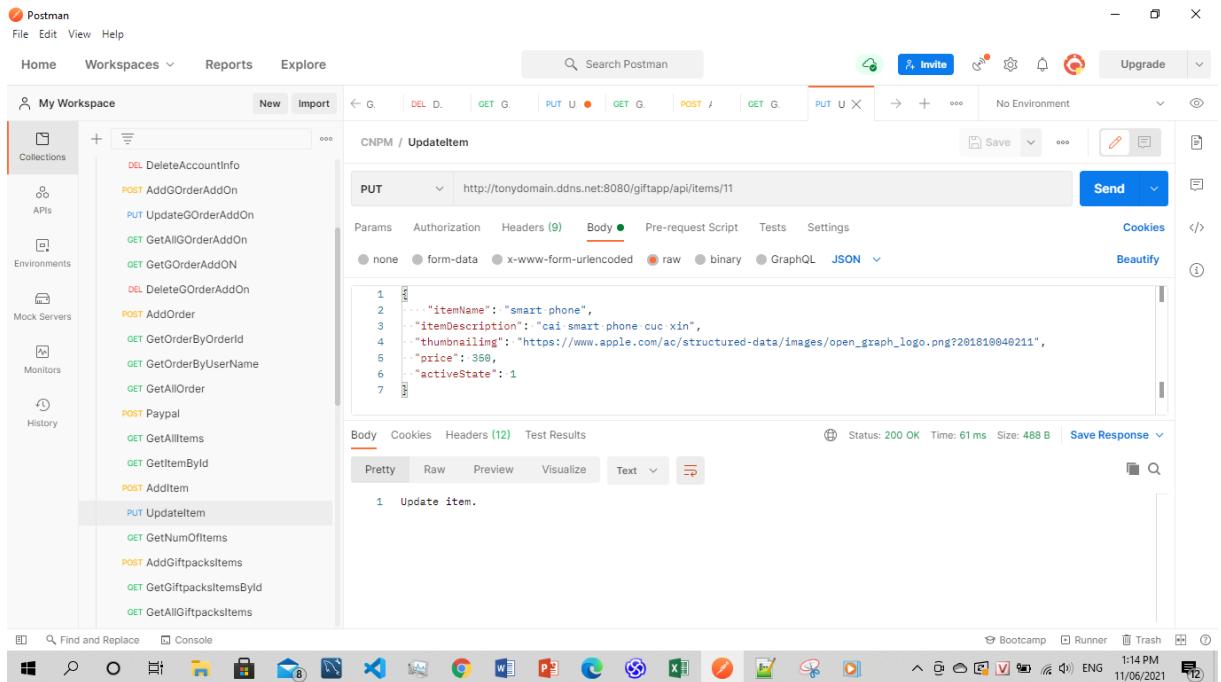
Chỉnh sửa thông tin bằng mã sản phẩm:

<http://tonydomain.ddns.net:8080/giftapp/api/items/{cid}>

Note: cid chính là mã hàng hóa.

Yêu cầu: itemName, itemDescription, thumbnailimg, price, activeState

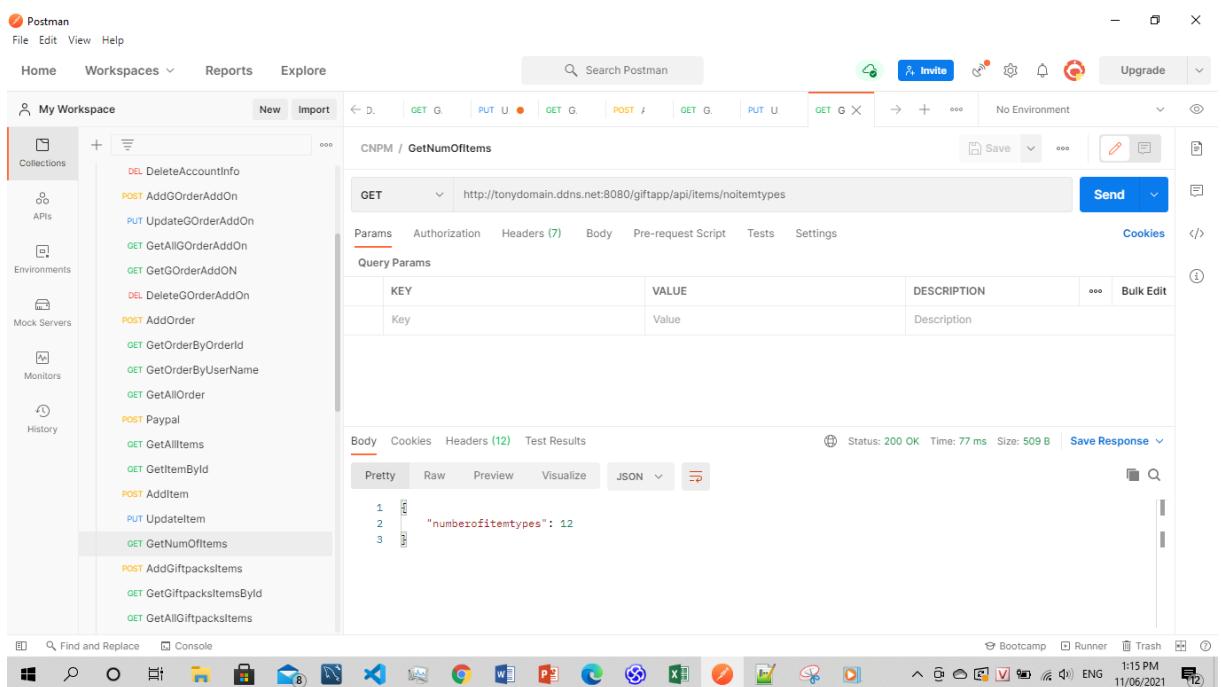
```
{
  "itemName": "smart phone",
  "itemDescription": "cai smart phone cuc xin",
  "thumbnailimg": "https://www.apple.com/ac/structured-
data/images/open_graph_logo.png?201810040211",
  "price": 350,
  "activeState": 1
}
```



Hình 3. 27: chỉnh sửa thông tin mặt hàng

Lấy số lượng sản phẩm:

<http://tonydomain.ddns.net:8080/giftapp/api/items/noitemtypes> (GET)



Hình 3. 28: Lấy số lượng mặt hàng

===== <API: Giftpack> =====

Mô tả: Giftpack là API sử dụng để quản lý tất cả các gói quà trong kho. Một gói quà có thể có nhiều mặt hàng.

Lấy tất cả thông tin các gói quà: <http://tonydomain.ddns.net:8080/giftapp/api/giftpacks> (GET)

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'My Workspace' containing various API endpoints. The main area shows a 'CNP / GetAllGiftpacks' collection with a single GET request to 'http://tonydomain.ddns.net:8080/giftapp/api/giftpacks'. The 'Params' tab is selected, showing a 'Query Params' table with a single entry 'Key' and 'Value'. Below the table, the 'Body' tab is selected, showing a JSON response:

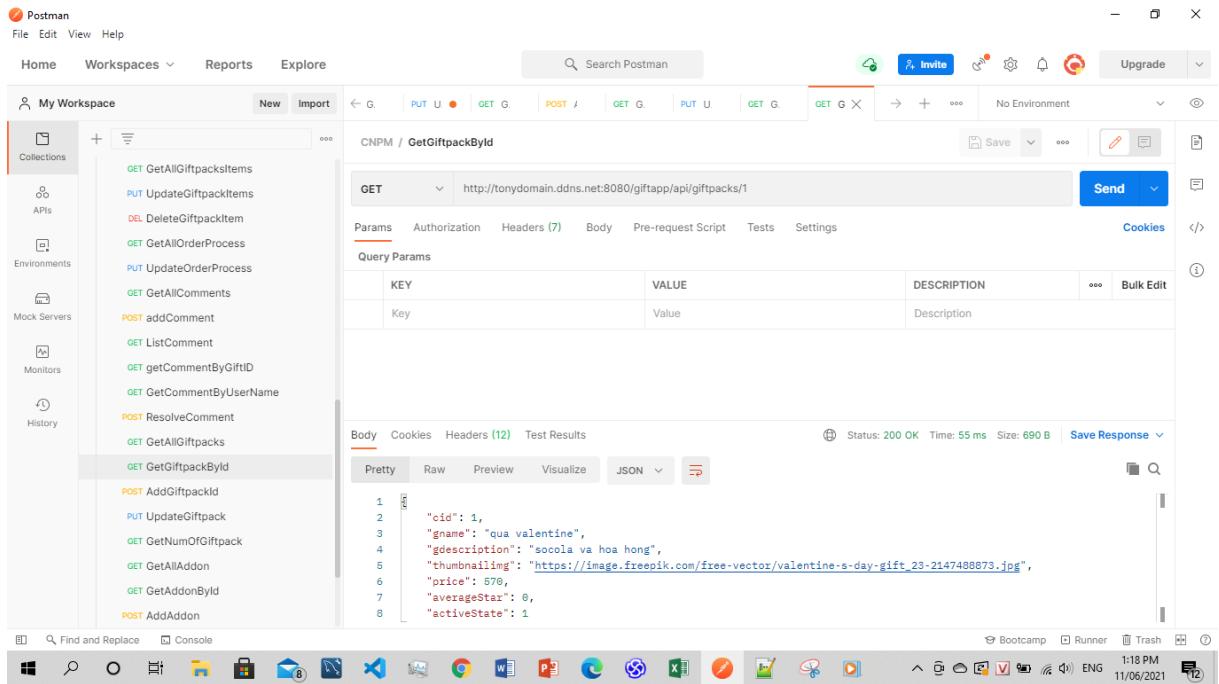
```
1  {
2   "cid": 1,
3   "gname": "qua valentine",
4   "gdescription": "socola va hoa hong",
5   "thumbnailImg": "https://image.freepik.com/free-vector/valentine-s-day-gift_23-2147488873.jpg",
6   "price": 578,
7   "averageStar": 0,
```

Hình 3. 29: Lấy tất cả các gói quà

Lấy thông tin một gói quà thông qua mã gói quà:

<http://tonydomain.ddns.net:8080/giftapp/api/giftpacks/{cid}> (GET)

Note: cid chính là mã gói quà.

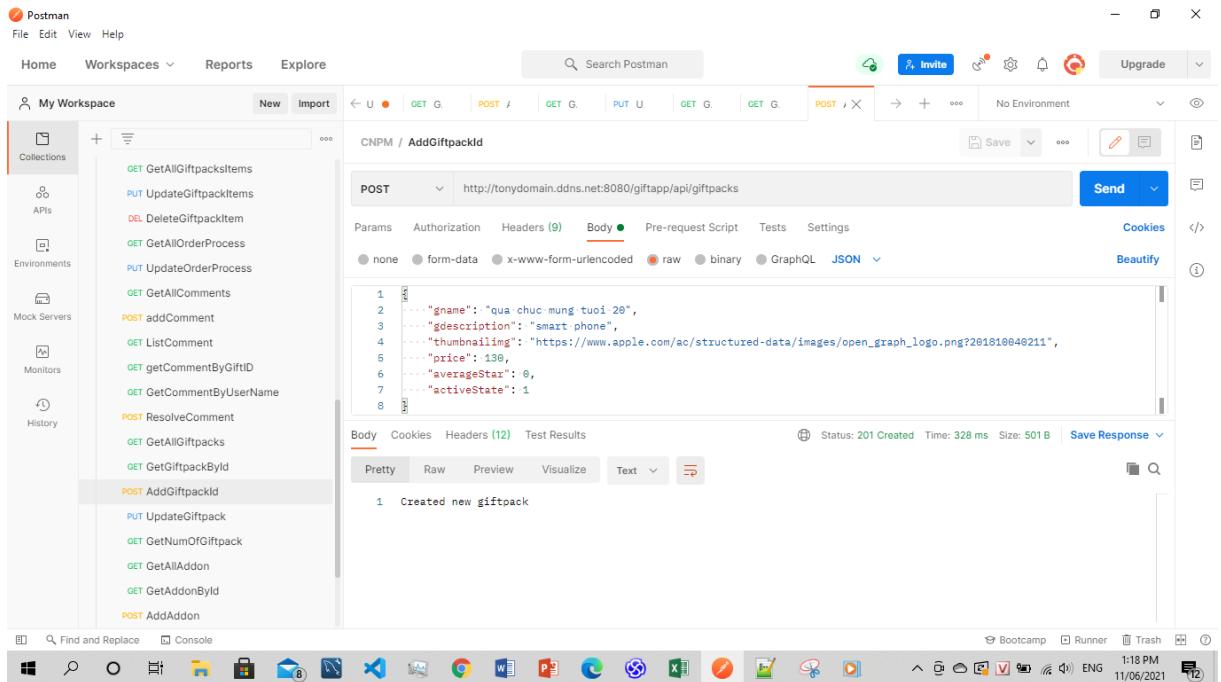


Hình 3. 30: Lấy gói quà thông qua mã gói quà

Tạo ra một gói quà mới: <http://tonydomain.ddns.net:8080/giftapp/api/giftpacks> (POST)

Yêu cầu: gname, gdescription, thumbnailimg, price, averageStar, activeState

```
{
  "gname": "qua chuc mung tuoi 20",
  "gdescription": "smart phone",
  "thumbnailimg": "https://www.apple.com/ac/structured-
data/images/open_graph_logo.png?201810040211",
  "price": 130,
  "averageStar": 0,
  "activeState": 1
}
```



Hình 3.31: Tạo một gói quà mới

Cập nhật một món quà: <http://tonydomain.ddns.net:8080/giftapp/api/giftpacks/12>
(PUT)

Yêu cầu: gname, gdescription, thumbnailimg, price, averageStar, activeState

```
{
  "gname": "qua chuc mung tuoi 20",
  "gdescription": "nhieu smart phone",
  "thumbnailimg": "https://www.apple.com/ac/structured-
data/images/open_graph_logo.png?201810040211",
  "price": 1300,
  "averageStar": 0,
  "activeState": 1
}
```

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'My Workspace' containing various API endpoints like 'GetAllGiftpackItems', 'UpdateGiftpackItems', etc. The main area shows a 'PUT / http://tonydomain.ddns.net:8080/giftapp/api/giftpacks/12' request. The 'Body' tab is selected, showing a JSON payload:

```

1
2   ...
3     "gname": "qua chuc mung tuoi 23",
4     "gdescription": "nhieu smart phone",
5     "thumbnailimg": "https://www.apple.com/ac/structured-data/images/open_graph_logo.png?201810040211",
6     "price": 1300,
7     "averageStar": 0,
8     "activeState": 1

```

The response status is 200 OK with a time of 43 ms and a size of 492 B. Below the response, the body is displayed as 'Updated giftpack'.

Hình 3. 32: chỉnh sửa thông tin một gói quà

Lấy số lượng gói quà có trong kho:

<http://tonydomain.ddns.net:8080/giftapp/api/giftpacks/nogiftpacktypes> (GET)

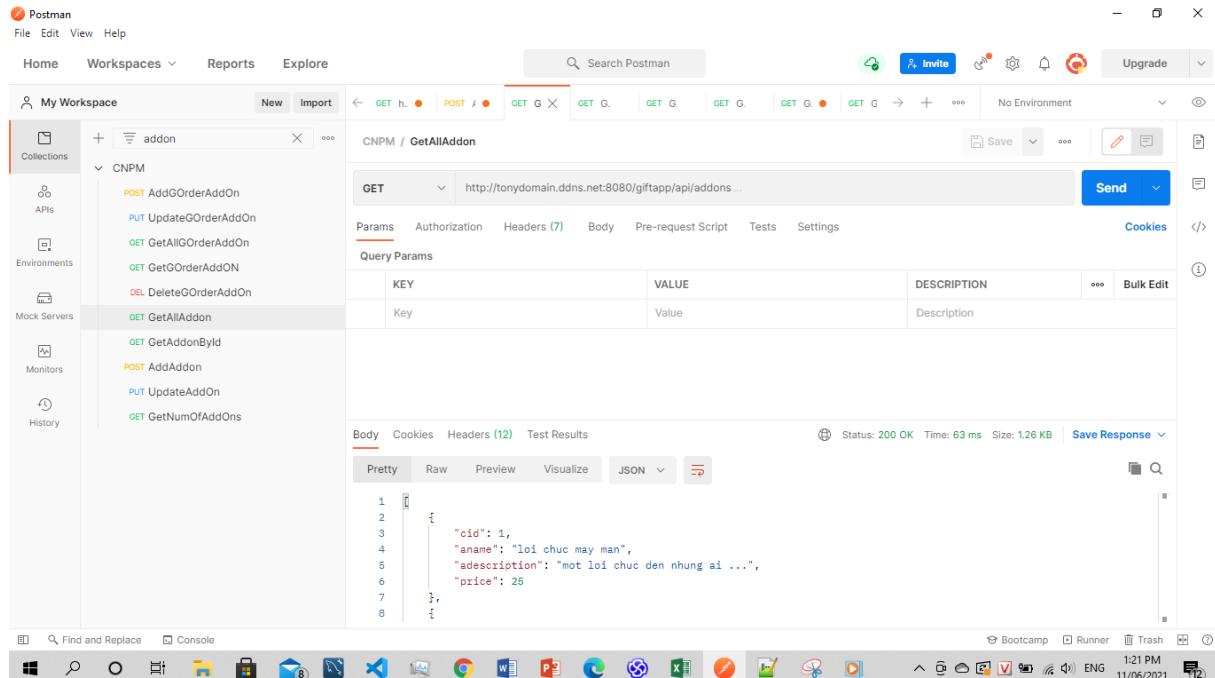
The screenshot shows the Postman application interface. On the left, there's a sidebar with 'My Workspace' containing various API endpoints like 'GetAllGiftpackItems', 'UpdateGiftpackItems', etc. The main area shows a 'GET / http://tonydomain.ddns.net:8080/giftapp/api/giftpacks/nogiftpacktypes' request. The 'Body' tab is selected, showing the response message: 'This request does not have a body'. The response status is 200 OK with a time of 70 ms and a size of 512 B. Below the response, the body is displayed as 'numberofgiftpacktypes': 8.

Hình 3. 33: API lấy số lượng gói quà

===== <API: AddOn> =====

Mô tả: một đơn hàng có thể có một số phụ kiện đính kèm. API này được sử dụng để quản lý tất cả phụ kiện đó.

Lấy tất cả các phụ kiện đi kèm: <http://tonydomain.ddns.net:8080/giftapp/api/addons> (GET)



The screenshot shows the Postman application interface. On the left, there's a sidebar with 'My Workspace' containing collections like 'CNPM' and 'Addon'. Under 'Addon', several endpoints are listed: POST AddOrderAddOn, PUT UpdateOrderAddOn, GET GetAllOrderAddOn, GET GetOrderAddON, DEL DeleteOrderAddOn, GET GetAllAddon, GET GetAddonByld, POST AddAddon, PUT UpdateAddOn, and GET GetNumOfAddOns. The main area shows a 'GET' request to 'http://tonydomain.ddns.net:8080/giftapp/api/addons...'. The 'Body' tab of the request details shows a JSON object:

```
1  {
2   "cid": 1,
3   "name": "loi chuc may man",
4   "adeescription": "mot loi chuc den nhung ai ...",
5   "price": 25
6 },
7 }
```

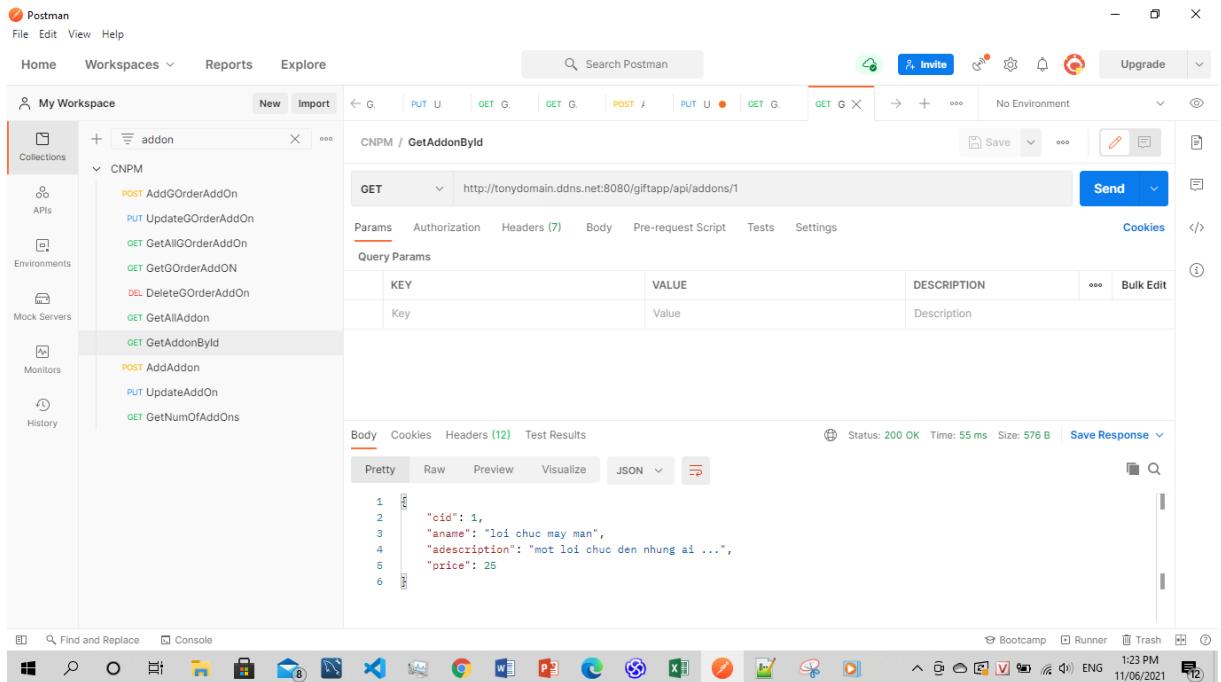
The response tab shows a successful 200 OK status with a response time of 63 ms and a size of 1.26 KB. The response body is identical to the request body above. The bottom of the screen shows the Windows taskbar with various pinned icons.

Hình 3. 34: API lấy tất cả các phụ kiện đi kèm

Lấy phụ kiện đi kèm theo mã phụ kiện:

<http://tonydomain.ddns.net:8080/giftapp/api/addons/{cid}> (GET)

Note: cid chính là mã phụ kiện.

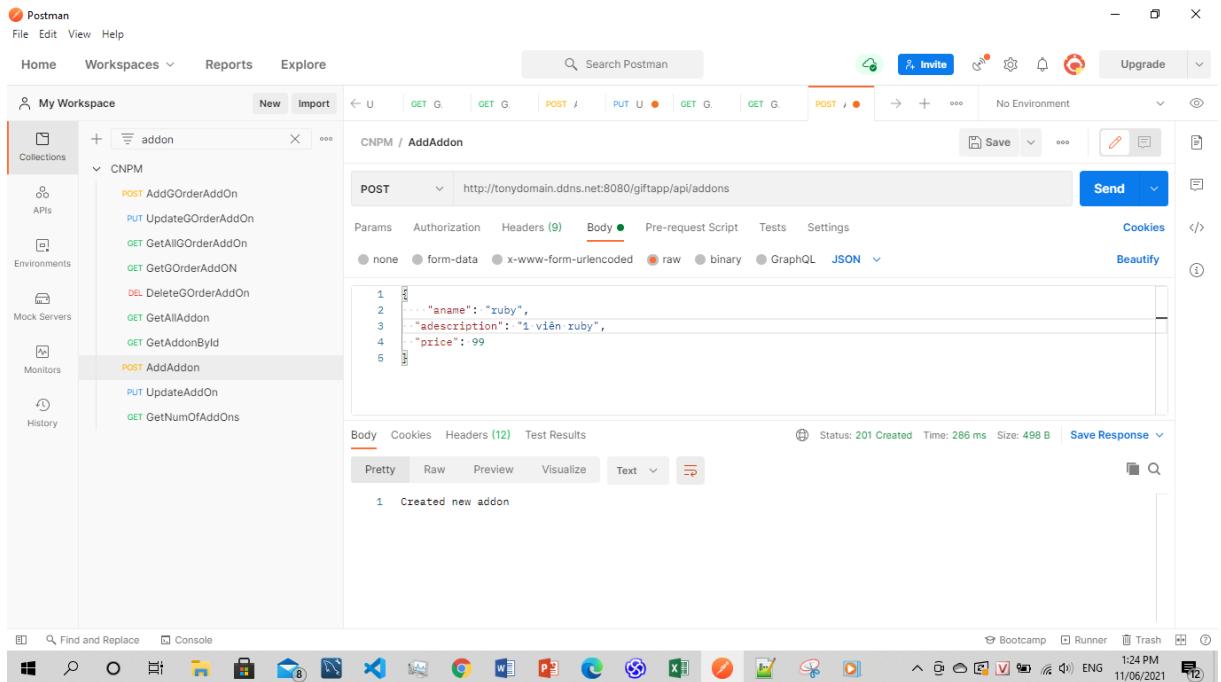


Hình 3. 35: API lấy phụ kiện đi kèm thông qua mã phụ kiện

Thêm một phụ kiện mới: <http://tonydomain.ddns.net:8080/giftapp/api/addons> (POST)

Yêu cầu: aname, adescription, price

```
{
  "aname": "flower",
  "adescription": "999 doa hoa hong",
  "price": 99
}
```



Hình 3. 36: API thêm một phụ kiện

Cập nhật một phụ kiện: <http://tonydomain.ddns.net:8080/giftapp/api/addons/{cid}>
(PUT)

Note: cid chính là mã phụ kiện

Yêu cầu: aname, adescription, price

```
{
  "aname": "flower",
  "adescription": "999 doa hoa hong, 1 thiệp",
  "price": 199
}
```

Postman

File Edit View Help

Home Workspaces Reports Explore

My Workspace

Collections

APIs

Environments

Mock Servers

Monitors

History

PUT G. POST / PUT U ● GET G. POST / PUT U X No Environment

Search Postman

PUT http://tonydomain.ddns.net:8080/giftapp/api/addons/9

Params Authorization Headers (9) Body Body

none form-data x-www-form-urlencoded raw binary GraphQL JSON

1: {
2: "name": "flower",
3: "description": "999 doa hoa hong, 1 thiệp",
4: "price": 199
5: }

Send Cookies Beautify

Status: 200 OK Time: 52 ms Size: 490 B Save Response

Pretty Raw Preview Visualize Text

1 Updated addon!

Find and Replace Console

Bootcamp Runner Trash

1:25 PM 11/06/2021

Hình 3. 37: API chỉnh sửa thông tin phụ kiện

Lấy số lượng phụ kiện:

<http://tonydomain.ddns.net:8080/giftapp/api/addons/noaddontypes> (GET)

Postman

File Edit View Help

Home Workspaces Reports Explore

My Workspace

Collections

APIs

Environments

Mock Servers

Monitors

History

POST G. POST / PUT U ● GET G. GET G. POST / POST U GET G. X No Environment

Search Postman

GET http://tonydomain.ddns.net:8080/giftapp/api/addons/noaddontypes

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (12) Test Results

Pretty Raw Preview Visualize JSON

1: {
2: "noaddontypes": 11
3: }

Save Response

Find and Replace Console

Bootcamp Runner Trash

1:25 PM 11/06/2021

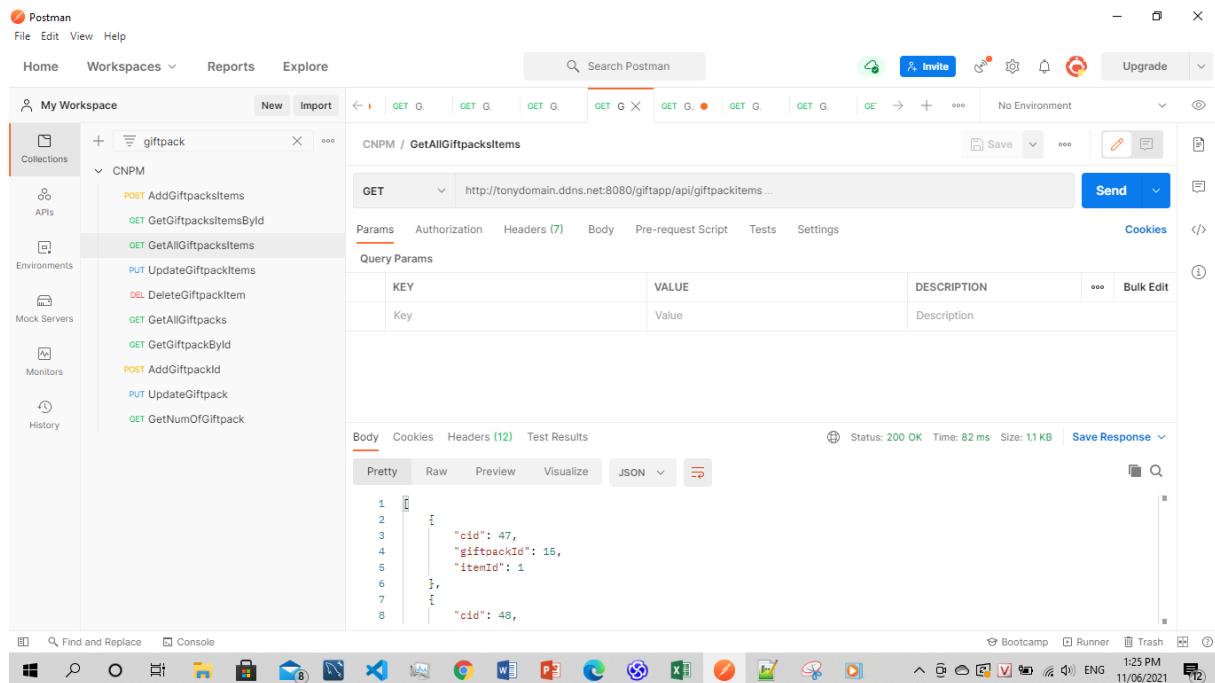
Hình 3. 38: API lấy số lượng phụ kiện

===== <API: GiftpackItems> =====

Mô tả: một gói quà thì bao gồm rất nhiều mặt hàng. Vì thế API được sử dụng để quản lý tất cả gói quà.

Lấy tất cả các gói quà kèm mặt hàng:

<http://tonydomain.ddns.net:8080/giftapp/api/giftpackitems> (GET)



The screenshot shows the Postman interface with the following details:

- Collection:** CNPM / GetAllGiftpacksItems
- Method:** GET
- URL:** http://tonydomain.ddns.net:8080/giftapp/api/giftpackitems
- Query Params:** Key
- Response Body (Pretty JSON):**

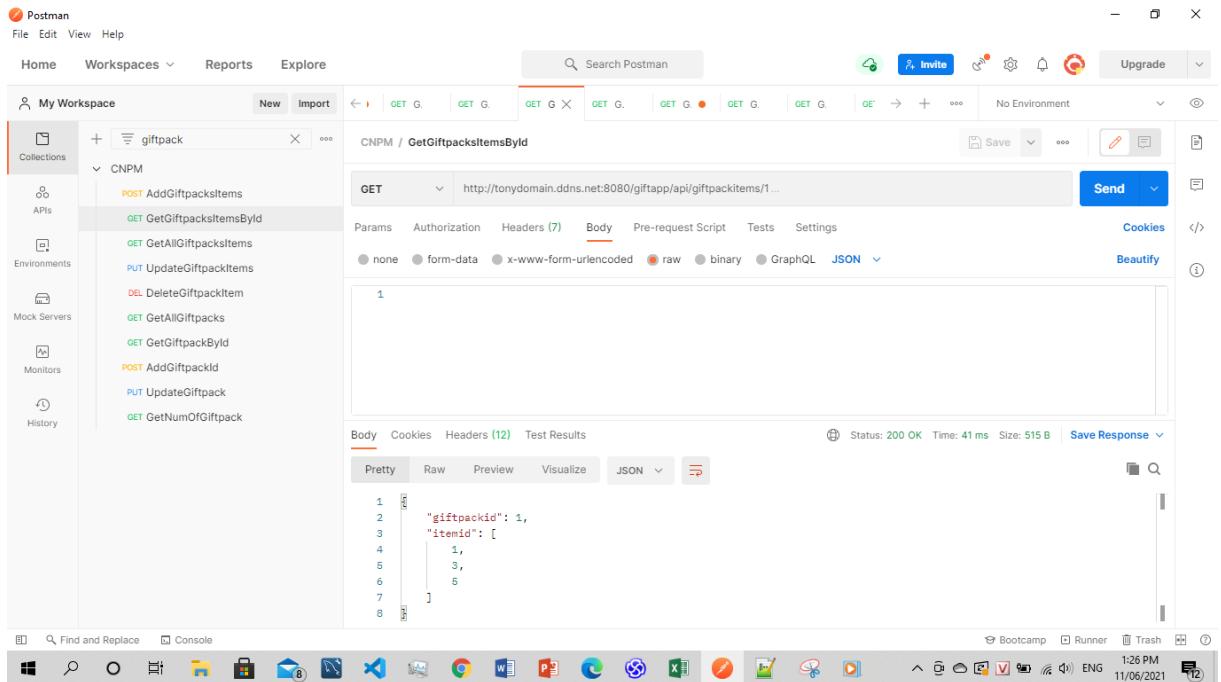
```
1  [
2   {
3     "cid": 47,
4     "giftpackId": 15,
5     "itemId": 1
6   },
7   {
8     "cid": 48,
```

Hình 3. 39: API lấy tất cả gói quà kèm mặt hàng đi kèm

Lấy các mặt hàng của một gói quà:

<http://tonydomain.ddns.net:8080/giftapp/api/giftpackitems/{cid}> (GET)

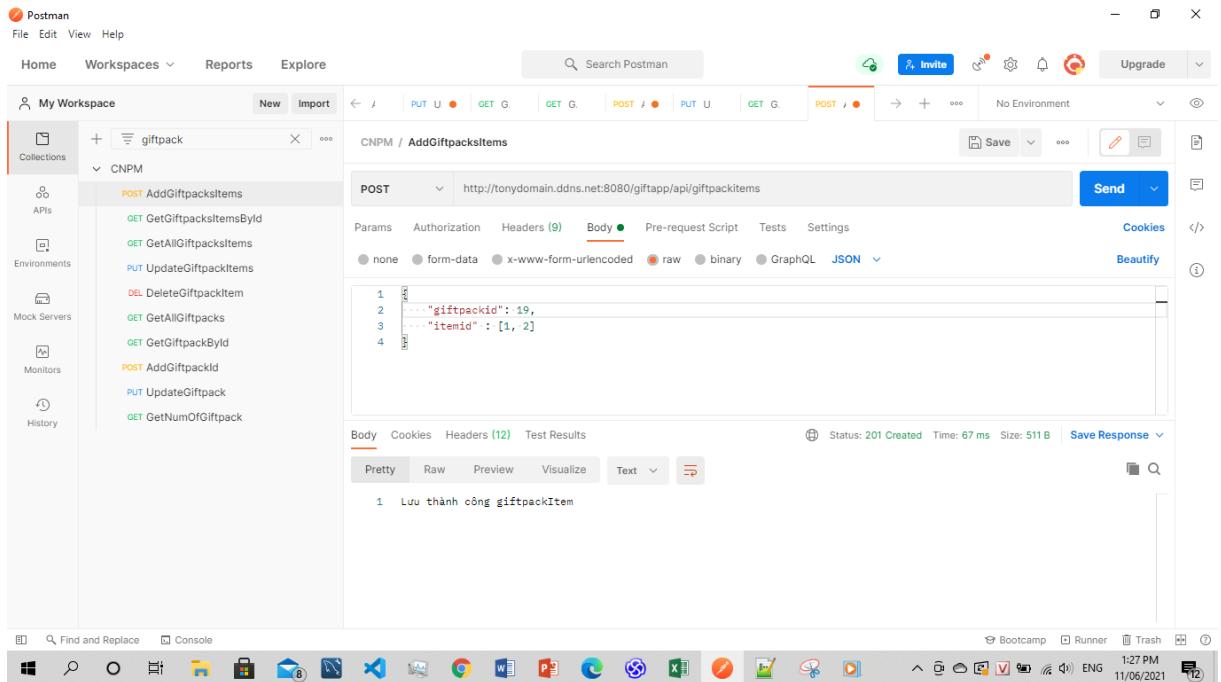
Note: cid chính là mã gói quà.



Hình 3. 40: Lấy gói quà kèm mặt hàng thông qua mã gói quà

Thêm một gói quà có nhiều mặt hàng: <http://localhost:8080/api/giftpackitems> (POST)

```
{
  "giftpackid": 1,
  "itemid" : [1, 2]
}
```



Hình 3. 41: Thêm gói quà kèm mặt hàng

Chỉnh sửa một gói quà kèm mặt hàng: <http://tonydomain.ddns.net:8080/giftapp/api/giftpackitems/{cid}>

Note: cid chính là mã của gói quà kèm mặt hàng được lưu trong db

Yêu cầu: giftpackId, itemId

```
{
  "giftpackId": 1,
  "itemId": 2
}
```

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** My Workspace, Collections (selected), APIs, Environments, Mock Servers, Monitors, History.
- Request URL:** http://tonydomain.ddns.net:8080/giftapp/api/giftpackitems/2
- Method:** PUT
- Body (JSON):**

```

1
2 ...
3 "giftpackId": 1,
4 ...
      
```
- Response:** Placeholder with an astronaut icon and the text "Hit send to get a response".
- Bottom Bar:** Windows taskbar with various pinned icons.

Hình 3. 42: Chỉnh sửa gói quà kèm mặt hàng

Xóa một mặt hàng trong gói quà:

<http://tonydomain.ddns.net:8080/giftapp/api/giftpackitems/{cid}>

Note: cid chính là mã của gói quà kèm mặt hàng được lưu trong db

The screenshot shows the Postman interface with the following details:

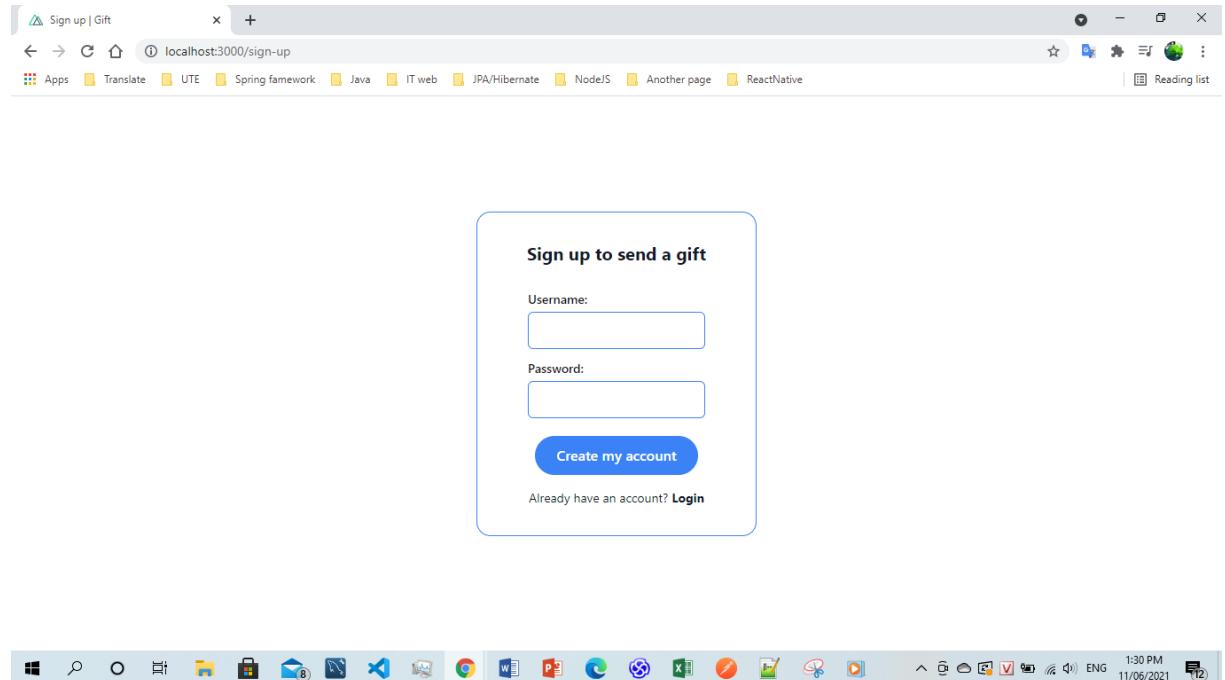
- Left Sidebar:** My Workspace, Collections (selected), APIs, Environments, Mock Servers, Monitors, History.
- Request URL:** http://tonydomain.ddns.net:8080/giftapp/api/giftpackitems/19
- Method:** DELETE
- Query Params:**

KEY	VALUE	DESCRIPTION	Bulk Edit
Key	Value	Description	
- Body:** Placeholder with the message "Xóa giftpackItem thành công".
- Status:** 200 OK, Time: 86 ms, Size: 506 B
- Bottom Bar:** Windows taskbar with various pinned icons.

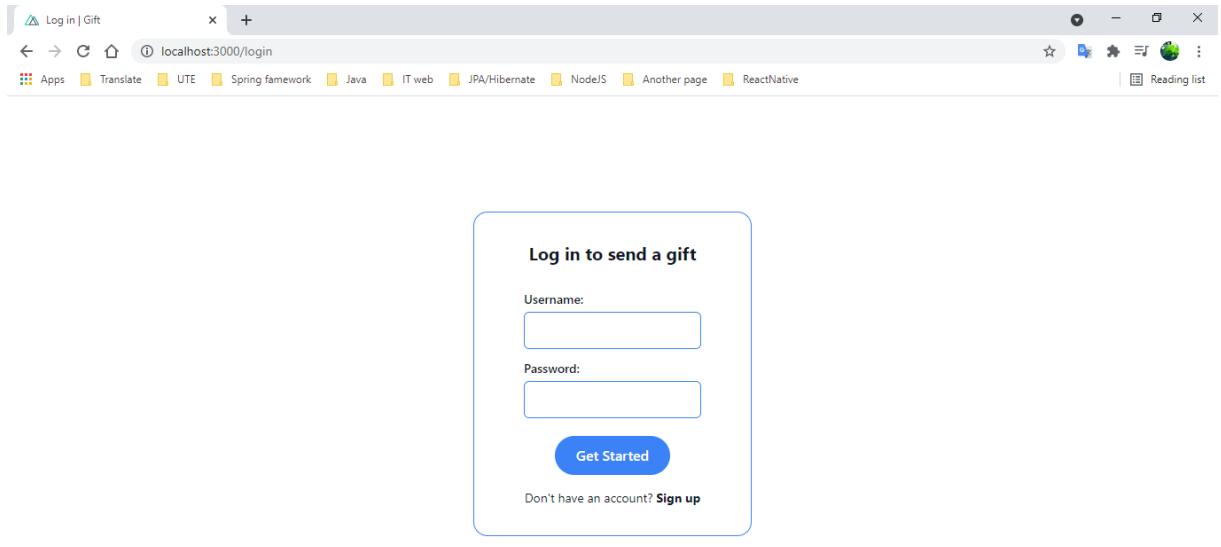
Hình 3. 43: Xóa các mặt hàng đi kèm gói quà

2. Thử nghiệm hệ thống front-end

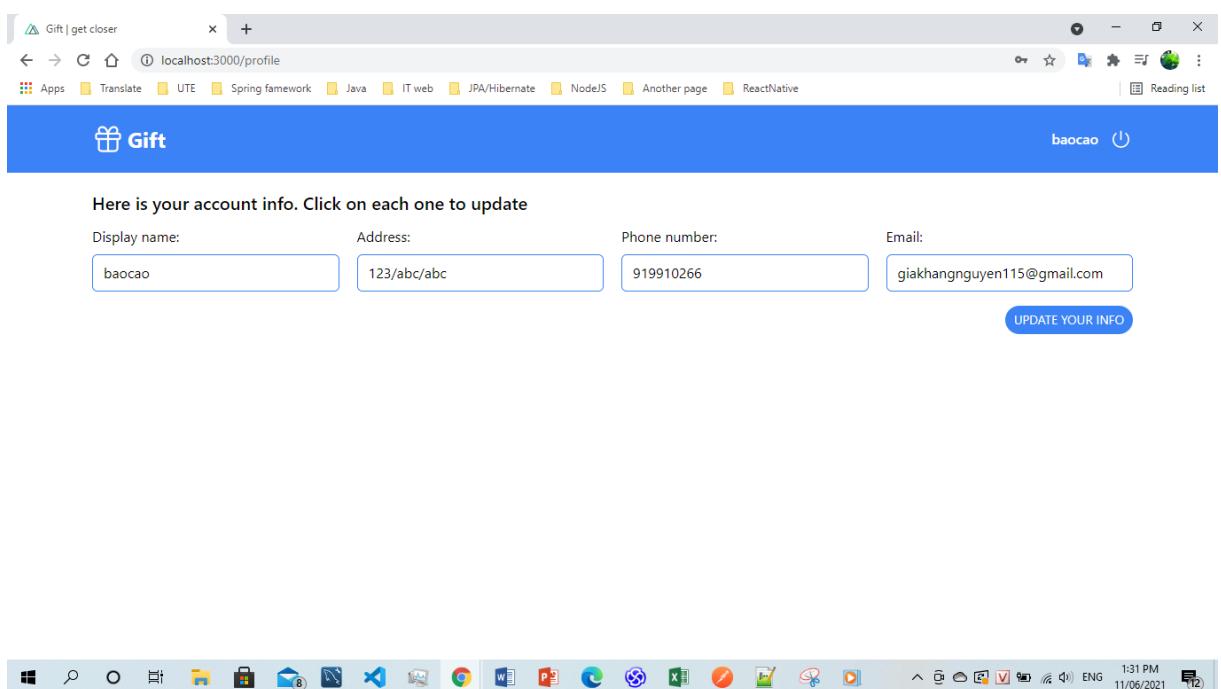
Ghi chú: một trang front end có thể tích hợp rất nhiều API.



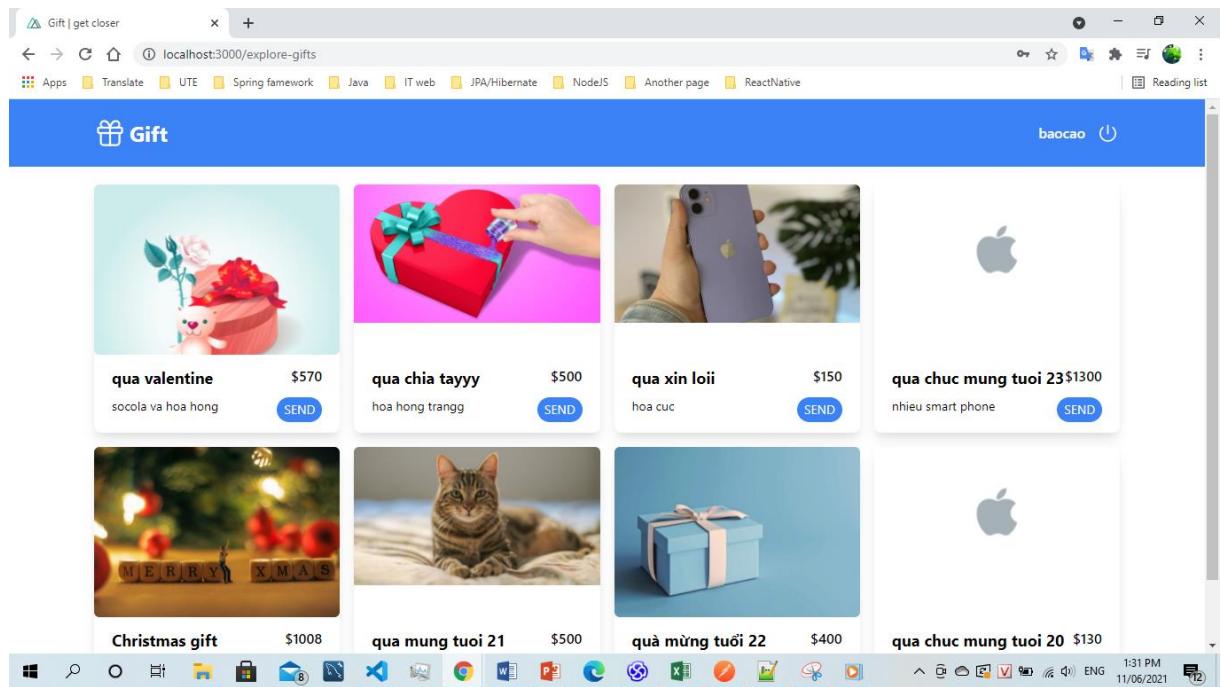
Hình 3. 44: Trang đăng ký



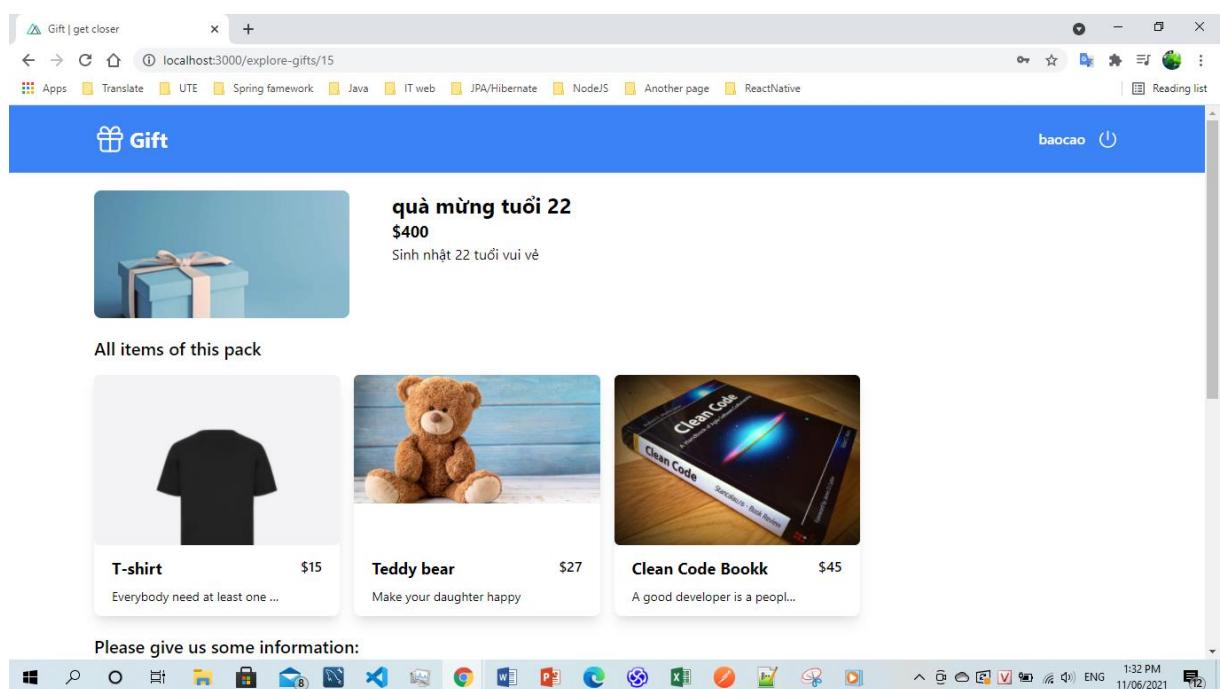
Hình 3. 45: Trang đăng nhập



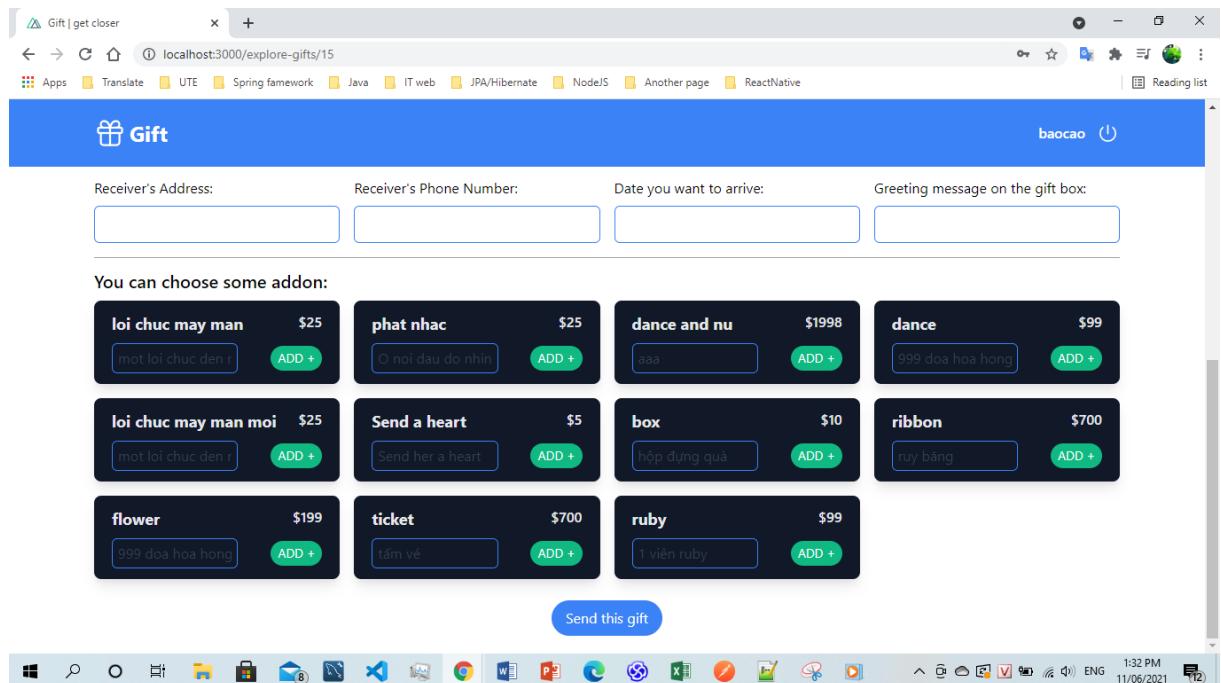
Hình 3. 46: trang thông tin khách hàng



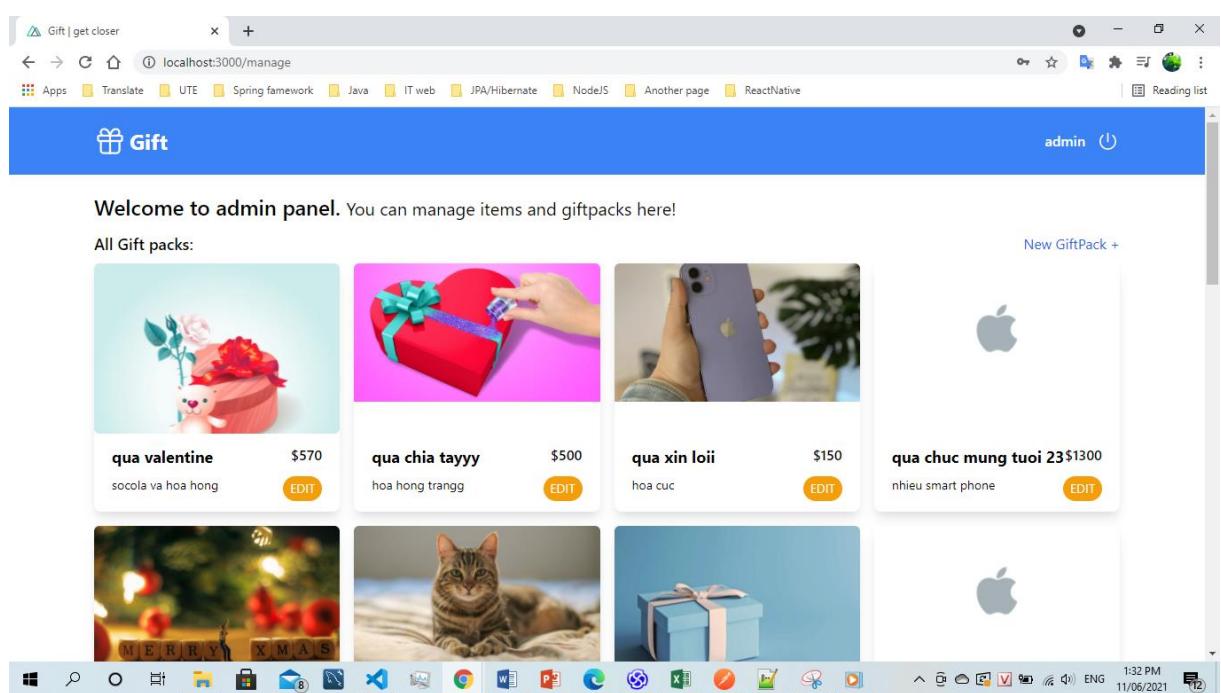
Hình 3. 47: trang chủ



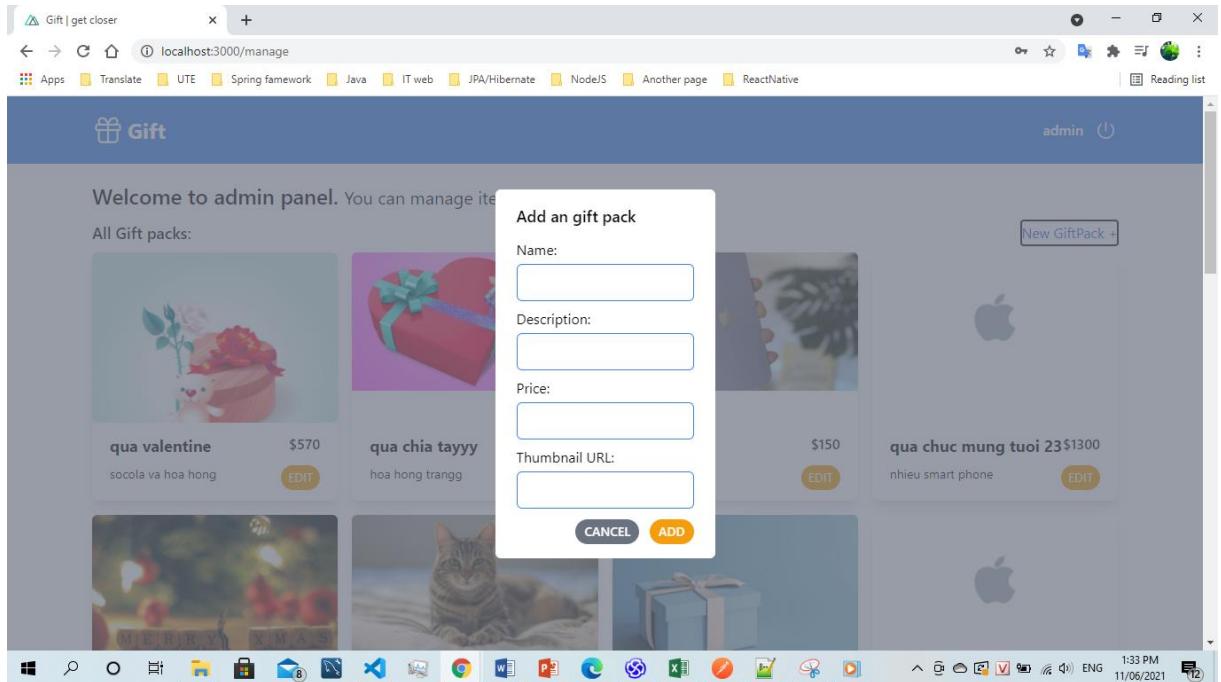
Hình 3. 48: trang đặt quà



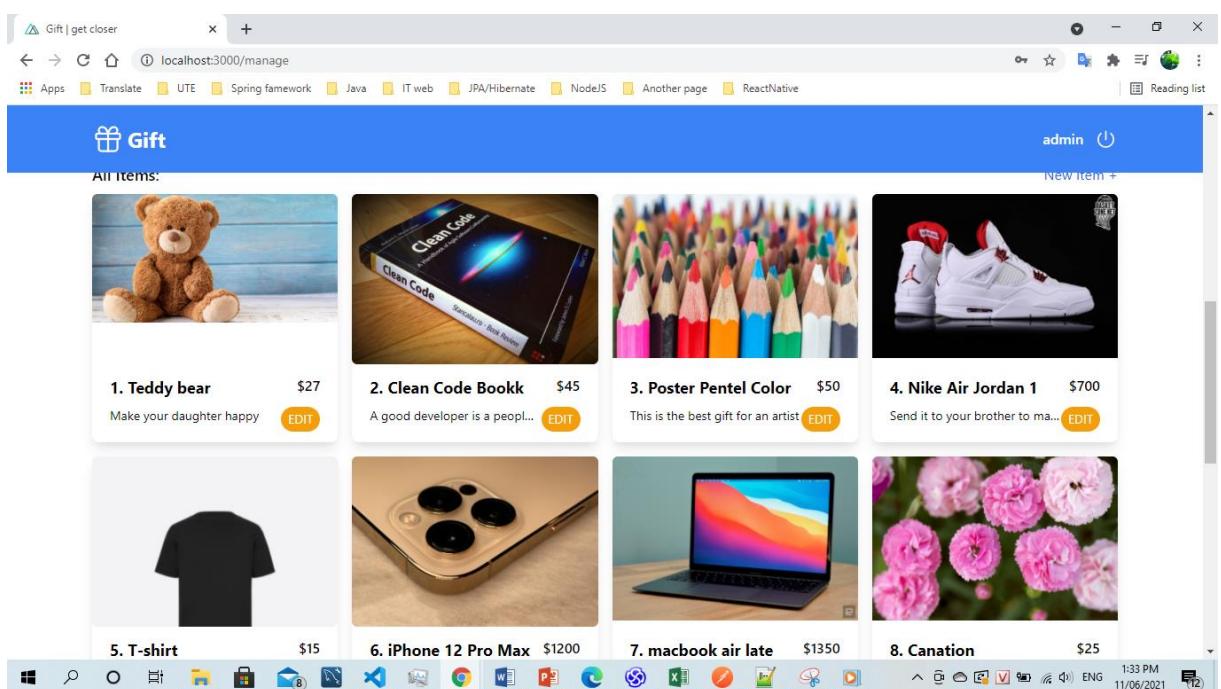
Hình 3. 49: Trang đặt quà



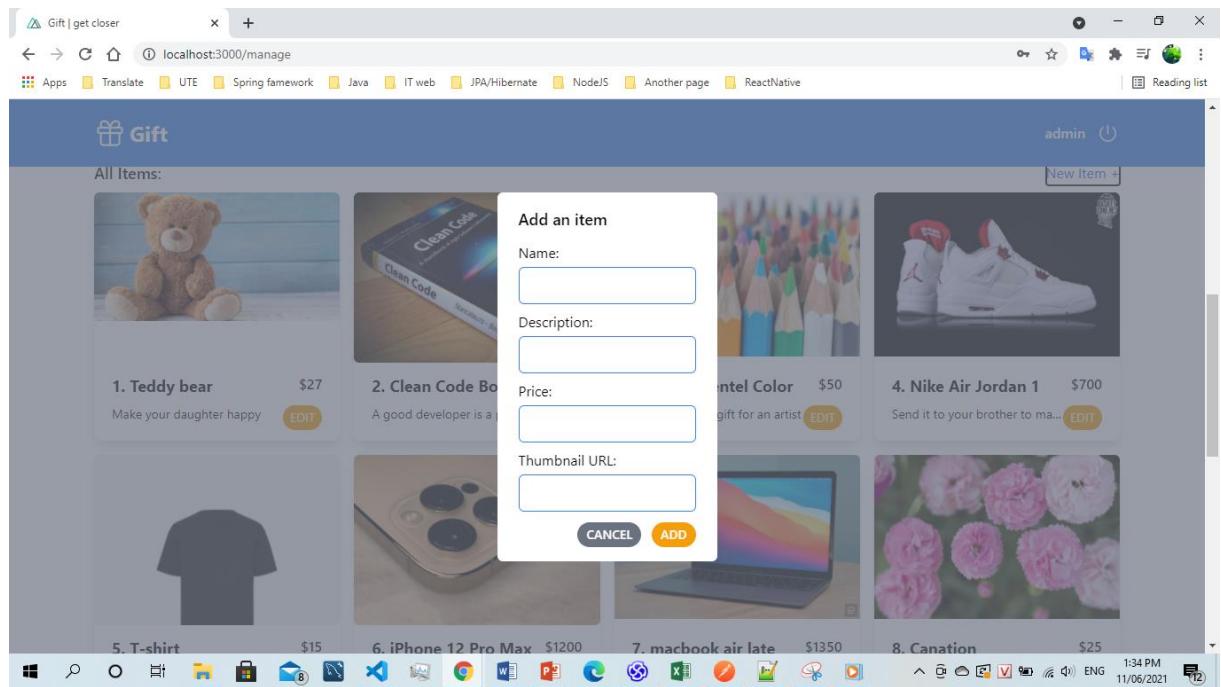
Hình 3. 50: trang giao diện admin mục quản lý gói quà



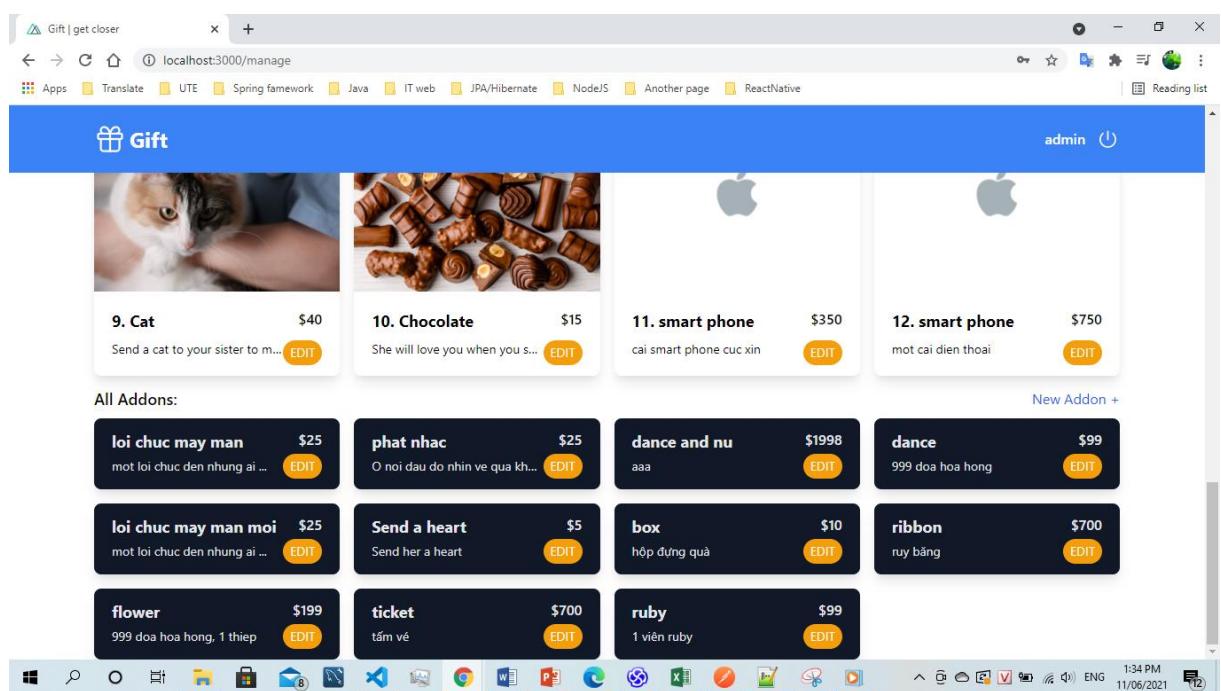
Hình 3. 51: Trang thêm một gói quà mới



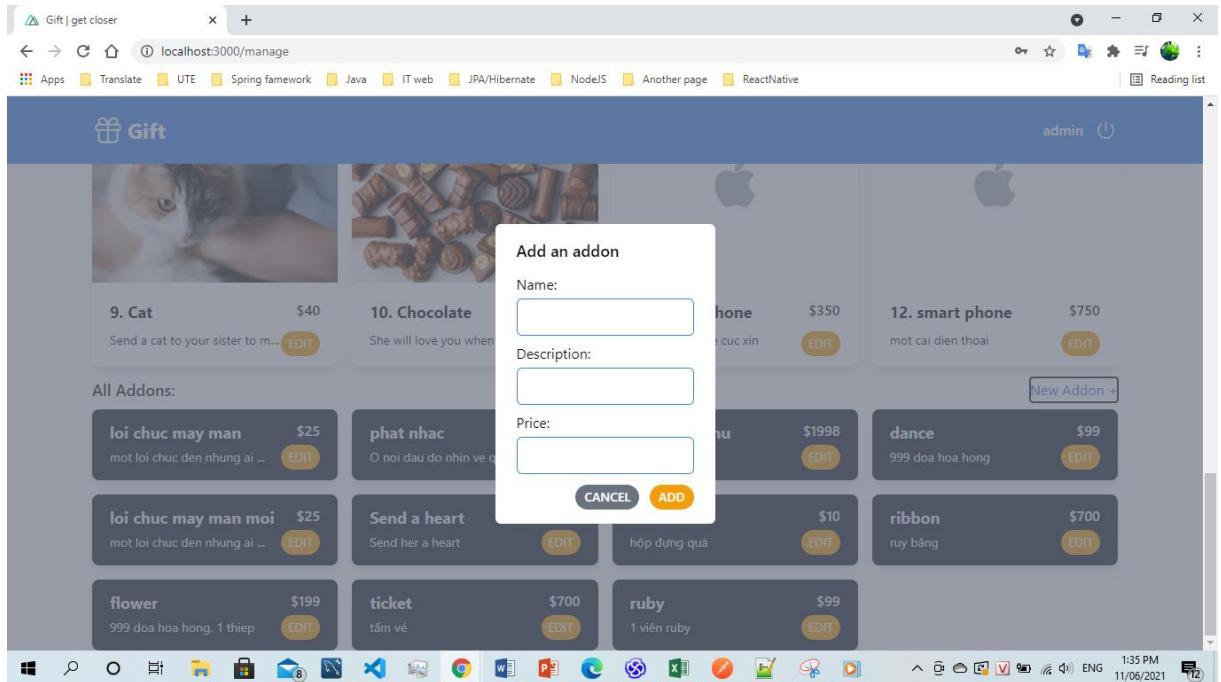
Hình 3. 52: Trang giao diện admin mục quản lý mặt hàng



Hình 3. 53: Trang thêm một mặt hàng mới



Hình 3. 54: Trang admin mục quản lí phụ kiện



Hình 3. 55: Trang thêm phụ kiện mới

IV. Sử dụng github

Link: <https://github.com/giakhangnguyenlam/WebsiteTangQua>

Backend:

<https://github.com/giakhangnguyenlam/WebsiteTangQua/tree/master/BackEnd>

Database:

https://github.com/giakhangnguyenlam/WebsiteTangQua/tree/master/DB_Exported

Front-end:

<https://github.com/giakhangnguyenlam/WebsiteTangQua/tree/master/FrontEnd/gift-client>

Thiết kế:

https://github.com/giakhangnguyenlam/WebsiteTangQua/tree/master/System_Design

Testcase:

<https://github.com/giakhangnguyenlam/WebsiteTangQua/tree/master/TestCase>

Document:

<https://github.com/giakhangnguyenlam/WebsiteTangQua/tree/master/document>

README:

<https://github.com/giakhangnguyenlam/WebsiteTangQua/blob/master/README.md>

Hỗ trợ test postman:

<https://github.com/giakhangnguyenlam/WebsiteTangQua/tree/master/Supporting%20test%20postman>

V.Kết luận

4.1. Đánh giá mức độ hoàn thành

Nhóm thực hiện tự đánh giá rằng nhóm đã cơ bản hoàn thành các yêu cầu của đồ án. Tất cả các tài liệu về api cũng như các hình ảnh kiểm thử đã được bổ sung đầy đủ. Qua đồ án nhóm cũng đã học thêm được nhiều kiến thức mới như viết api, phương pháp thiết kế một ứng dụng cũng như các kiến thức liên quan đến việc làm việc nhóm, xử lý các tình huống xung đột trên github cũng như có cái nhìn khái quát một ứng dụng được hình thành như thế nào. Tuy nhiên trong quá trình làm đồ án cũng không thể tránh những thiếu sót, nhóm sẽ tiếp tục hoàn thiện và phát triển trong những dự án tiếp theo.

4.2. Khó khăn

Trong quá trình làm việc nhóm chưa thể giải quyết vấn đề bảo mật ứng dụng. Ứng dụng chưa hoàn thiện Front-end. Việc sử dụng github cũng gặp đôi chút khó khăn khi xảy ra xung đột. Kiến thức api của nhóm cũng chưa hoàn thiện dẫn đến những khó khăn nhất trong việc giải quyết các tình huống phát sinh khi xây dựng ứng dụng. Đồng thời tác động của dịch bệnh nên nhóm chỉ có thể giao tiếp qua google meet nên việc triển khai ứng dụng trở nên khó khăn hơn rất nhiều.

4.3. Hướng phát triển

Trong thời gian tiếp theo, nhóm thực hiện sẽ cố gắng cải thiện web với giao diện cũng như những tính năng mới hơn, đáp ứng đầy đủ yêu cầu của người dùng hơn.

Các vấn đề còn lại của trang web, nhóm sẽ tiến hành tìm kiếm tài liệu để học hỏi nhiều hơn để tích luỹ kiến thức về việc tạo lập một trang web hơn. Song song với việc

trên, nhóm thực hiện sẽ tiến hành kiểm tra lại trang web và tìm cách nâng cấp để làm giảm đi những vấn đề còn hạn chế của phần mềm.

Nhóm sẽ tiếp tục phát triển web theo hướng viết api và sẽ có giao cung cấp những phiên bản mới hơn trong thời gian tới đồng thời cũng có kiến thức để ứng dụng việc bảo mật vào web để hoàn thiện ứng dụng.